



People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

Mohamed Khider University – BISKRA

Faculty of Exact Sciences

Department of Computer Science

Order N°: SIOD01/M2/2025

Master's Thesis

Submitted in fulfillment of the requirements for Master's degree in

Computer science

Specialization: Information Systems, Optimization and Decision (SIOD)

**Extraction and generation of metadata for scientific publications: Development
of an automation tool for non-specialists**

By

Douha MEGHERBI

Thesis publicly defended on 04/06/2025 in front of a jury composed by:

MRS. Berghida Meryem	MAB	President
Mrs. Moufida BENCHABANE	MAA	Supervisor
MR. Kalfali Toufik	MAA	Examiner

Session 2024-2025



Acknowledgements

With a heart filled with gratitude and humility, I begin by acknowledging Allah, the Most Gracious, the Most Merciful. It is by His divine mercy, guidance, and boundless generosity that I have been granted the strength and perseverance to complete this journey.

I would like to express my deepest and most sincere appreciation to my supervisor **Mrs. Moufida Benchabane**, whose invaluable mentorship, scholarly insight, and unwavering support have been central to the completion of this master's thesis. Her patience, encouragement, and belief in my potential provided me with both the confidence and clarity needed throughout this research.

I am sincerely grateful to the jury chair, *****, and the reviewer, *****, for their time and for contributing to the improvement of this manuscript. I also thank all the faculty members and staff of the Department of Computer Science at the University of Mohamed Khider Biskra; their guidance, academic expertise, and steadfast support played a significant role in shaping my intellectual growth and fostering a research-driven mindset.

I would also like to thank **Mr. Mohamed Djebbar**, the **Manager of Alwaha Scientific Publishing Services SARL (ASPS)**, for giving me the opportunity to understand the editorial process of publishing scientific journals through the completion of my practical internship carried out mainly at the ASPS in Biskra, at the University of Biskra and at home remotely between February 15, 2025 and May 25, 2025. During this period, my contact with ASPS staff was mainly with my internship tutor and co-supervisor **Dr. Adel Benchabane, Professor at the University of Biskra and Publication Director of the ASPS**. So, a special note of thanks goes to Prof. Adel Benchabane for having agreed to supervise my practical training (remote and face-to-face); for their openness, trust, and cooperation -particularly during the critical stages of data acquisition and code validation- were essential to the applied dimension of this work. The opportunity to collaborate with him enriched my practical understanding of metadata indexing via XML files, especially during the online Crossref event held on 13 March 2025. I had the honor of being the only representative of the ASPS during this online international event. Once again: thank you very Professor Adel Benchabane for your trust.

To my beloved **family and friends**, I offer my most profound gratitude. Your prayers, sacrifices, and constant encouragement have been a source of strength through every challenge. Your belief in me has never wavered, and for that, I am eternally thankful.

Finally, I wish to recognize all **colleagues, peers, and mentors**—named or unnamed—who have contributed in any capacity, whether through technical advice, moral support, or simple words of encouragement. Every contribution, no matter how small, is deeply appreciated. your generosity and kindness have not gone unnoticed. This achievement stands not only as a personal milestone but as a collective testament to the community that surrounded me with inspiration and support.

Dedication

This master's thesis is dedicated to the incredible people who lifted me, inspired me, and stood by my side through every step of this journey. My family's belief in me has been my armor, shielding me through the hardest times. My friends' endless encouragement has been my fuel, constantly reminding me why I began and how far I've come. I owe this accomplishment to all of you.

To my beloved parents and grandmother,

To my amazing siblings,

To all my family,

To my cherished friends,

To my amazing classmates

Abstract

The main objectives of this master's thesis are: (i) review the state of the art of the computerization of the editorial process of scientific journals, (ii) Automate DOI (Digital Object Identifier) indexing through XML Export Plugin of a peer review management software, (iii) Design and develop a lightweight tool for beginner editors which allows to extract article metadata and generate XML files ready for indexing, (iv) Test and validate the developed tool by using recent published articles of an active journal. To achieve these objectives, a practical internship of more than three months was carried out at Alwaha Scientific Publishing Services SARL (ASPS), an independent international academic publisher. The metadata of an active scientific journal, titled 'Energy and Thermofluids Engineering, ETE, ISSN 2716-8026', was considered as a case study. Thus, this manuscript first deals with the state of the art of the digitalization of the scientific journal publishing process from submission to dissemination. Particular attention was drawn to the system used by the ASPS. It's about the Open Journal Systems (OJS), an open source and free software for the management of peer-reviewed academic journals. The second chapter represents a practical work showing how to automate DOI indexing through OJS by leveraging tools such as the Crossref XML Export Plugin and DOI Public Identifier. The methodology draws on firsthand experience gained through the participation in a Crossref-hosted webinar held on 13 March 2025. In this international event, advanced OJS workflows were implemented on ASPS platform to reduce processing time by 80% and improve metadata completeness scores from 65% to 98%. The third chapter presents the design and development of a lightweight XML generation tool tailored for beginner publishers who lack technical expertise or access to automation. The tool converts structured CSV inputs into Crossref-compliant XML files, eliminating the need for manual XML editing. Developed in C++ for efficiency and portability, it allows non-specialist editorial teams to register DOIs and deposit metadata with minimal effort. The last chapter presents successful stages of testing and validation of the tool developed. In summary, this work empowers publishers of all scales to meet global indexing standards. The proposed solutions can enhance the discoverability of Algerian research and provide a replicable framework for institutions in resource-constrained regions to participate equitably in open scholarly communication.

Keywords: XML, C++, Metadata, Extraction, Generation, scientific publications

ملخص

تلخص الأهداف الرئيسية لهذه المذكرة فيما يلي: (أ) دراسة بيبليوغرافية لأحدث التقنيات الخاصة بحوسبة عملية تحرير المجلات العلمية، (ب) الرقمنة الآلية لعملية الفهرسة عبر معرف الكائن الرقمي (DOI) من خلال XML Export Plugin الخاص بأحد برامج إدارة تحرير المجلات، (ج) تصميم وتطوير وسيلة رقمية سهلة الاستعمال للمحررين المبتدئين والتي تسمح باستخراج بيانات التعريف للمقالات وإنشاء ملفات XML جاهزة للفهرسة، (د) اختبار البرنامج المطور والتحقق التجريبي من نجاعته باستخدام المقالات المنشورة مؤخرًا في مجلة علمية نشطة. لتحقيق هذه الأهداف، تم إجراء تدريب عملي لأكثر من ثلاثة أشهر لدى الواحة لخدمات النشر العلمي م.ذ.م. (ASPS)، وهو ناشر أكاديمي دولي مستقل. تم اعتبار البيانات التعريفية لمجلة علمية نشطة بعنوان "هندسة الطاقة والسوائل الحرارية، ETE، ISSN 2716-8026" كدراسة حالة. تتناول هذه المذكرة أولاً عرض لأحدث التقنيات الخاصة بحوسبة عملية تحرير المجلات العلمية من استقبال مشاريع المقالات إلى النشر. تم التركيز بشكل خاص إلى النظام الذي تستخدمه ASPS. ويتعلق الأمر بنظام المجلات المفتوحة (OJS)، وهو برنامج مفتوح المصدر ومجاني لإدارة المجلات الأكاديمية التي يم مراجعتها من قبل النظراء. يوضح الفصل الثاني من هذه المذكرة كيفية الرقمنة الآلية لعملية فهرسة DOI من خلال OJS باستعمال أدوات مثل Crossref XML Export Plugin. تعتمد المنهجية المتبعة على الخبرة المباشرة المكتسبة من خلال المشاركة في ندوة دولية عبر الإنترنت لـ Crossref عقدت في 13 مارس 2025. خلال فعاليات هذا الحدث الدولي، تم تقييم وتحديث عمل OJS على منصة ASPS لتقليل وقت المعالجة بنسبة 80٪ وتحسين درجات اكتمال البيانات الوصفية من 65٪ إلى 98٪. يقدم الفصل الثالث تصميم وتطوير أداة توليد XML خفيفة الوزن مصممة خصيصًا للناشرين المبتدئين الذين يفتقرون إلى الخبرة الفنية أو التحكم في الرقمنة الآلية لعملية فهرسة. تُحوّل هذه الأداة مُدخلات CSV المُهيكلية إلى ملفات XML متوافقة مع مواصفات Crossref، مما يُعني عن التحرير اليدوي لـ XML. طُوّرت هذه الأداة بلغة C++ لضمان الكفاءة وسهولة النقل، وهي تُتيح لفرق التحرير غير المُتخصصة تسجيل مُعرّفات الكائنات الرقمية (DOIs) وإيداع البيانات الوصفية بأقل جهد. يُقدّم الفصل الأخير من هذه المذكرة عرض لمراحل اختبار ناجحة للأداة المُطوّرة. باختصار، يُمكن هذا العمل الناشرين من جميع المستويات من استيفاء معايير الفهرسة العالمية. تسمح الحلول المُقترحة أن تُحسّن من إمكانية تطوير نشر الأبحاث الجزائرية، وأن تُوفّر إطارًا مسهل لعمل المؤسسات في المناطق المحدودة الموارد وبذلك الإسهام بشكل فعال في التواصل العلمي المفتوح.

الكلمات المفتاحية: XML، C++، البيانات الوصفية، الاستخراج، التوليد، المنشورات العلمية

Table of Contents

ACKNOWLEDGEMENTS	III
DEDICATION	IV
ABSTRACT	V
ملخص	VI
GENERAL INTRODUCTION	1
1 DIGITALIZATION OF JOURNAL PUBLISHING PROCESS: BACKGROUND	4
1.1 Introduction	4
1.2 Historical Evolution of Scholarly Publishing	4
1.3 Core Terminologies and Conceptual Framework	5
1.3.1 Publisher (Publishing House)	5
1.3.2 Indexing (in Academic Publishing)	6
1.3.3 Open Journal Systems (OJS)	6
1.3.4 Metadata	6
1.3.5 Digital Object Identifier (DOI)	7
1.4 Functional Roles within Open Journal Systems (OJS)	7
1.4.1 Author	8
1.4.2 Production or Managing Editor	8
1.4.3 Section Editor	8
1.4.4 Layout Editor (Designer)	8
1.4.5 Reviewer	9
1.4.6 Copyeditor	9
1.4.7 Proofreader	9
1.4.8 Reader/Community	9
1.4.9 Indexer (or "Metadata Editor")	10
1.5 Editorial Workflow: From Submission to Dissemination	10

1.5.1	Scientific Evaluation Phase	11
1.5.2	Manuscript Production Phase	12
1.6	Comparative Analysis of Print and Digital Production	13
1.6.1	Production Workflows in Print Publishing	14
1.6.2	Production Workflows in Digital Publishing	14
1.7	Technical Perspectives on Editorial Processes	15
1.7.1	Scientific Phase: Adaptive and Dynamic Design Needs	15
1.7.2	Production Phase: Structured Data and Automation Potential	15
1.7.3	Towards Standardization and Semi-Automation	16
1.8	Critical Role of the Indexer in Scholarly Communication	16
1.9	Functions of the Indexer	17
1.9.1	Types of Indexers	17
1.10	.Automation Opportunities and Technical Limitation	19
1.11	Conclusion	19
2	CASE STUDY OF AUTOMATED INDEXING USING THE ASPS OJS PLATFORM: THE CROSSREF XML EXPORT PLUGIN	21
2.1	Introduction	21
2.2	Overview of Alwaha Scientific Publishing Services (ASPS)	21
2.2.1	Journal Portfolio and Scope	22
2.2.2	Open-Access Policy and Global Reach	23
2.2.3	ASPS's Research and Publication Ethics	23
2.2.4	ASPS's Standards for Authors: Responsible Research Publication	23
2.2.5	ASPS's Privacy Statement	24
2.3	Open Journal Systems (OJS) as ASPS's Publishing Platform	24
2.3.1	Workflow Automation and Metadata Management	24
2.3.2	Operational Advantages for ASPS	25
2.4	Challenges and Strategic Directions for OJS Optimization	25
2.4.1	Current Technical and Organizational Gaps	25
2.4.2	Strategic Outlook	26
2.5	Crossref: Background and Strategic Relevance for ASPS	26
2.6	Capacity Building: ASPS Training and Crossref Webinar	27

2.6.1	Scholarship Activities and Skill Development Outcomes	27
2.6.2	Assessing Metadata Quality with the Crossref Participation Reports Tool	28
2.6.3	Crossref Webinar: Technical Training and Strategic Upskilling	32
2.7	Automated Indexing Workflow	33
2.7.1	Manual Indexing	33
2.7.2	Limitations of Manual Indexing at ASPS	33
2.7.3	Transition to Automation via OJS Plugins	33
2.7.4	Core OJS Components Configured for Automation	34
2.7.5	OJS Plugins for Indexing Automation	35
2.8	Implementing Automated Indexing in OJS	36
2.8.1	Plugin Configuration	37
2.8.2	Retro-Assigning DOIs to Previously Published Articles	43
2.10	Recommendations and Improvements	47
2.9	2.11 Conclusion	47
3	DESIGN AND IMPLEMENTATION OF XML GENERATOR TOOL FOR METADATA INDEXERS: CASE STUDY OF ASPS JOURNALS	49
3.1	Introduction	49
3.2	XML and Metadata Standards	49
3.2.1	Evolution of Metadata Standards	50
3.2.2	Schema Analysis: Crossref, JATS, and Dublin Core	51
3.3	Indexing Services: Requirements and Strategic Value	51
3.3.1	some indexing services	51
3.3.2	Metadata Requirements	54
3.3.3	Unified Field Matrix	54
3.3.4	Relevance to ASPS: Strategic Metadata Compliance	55
3.4	Crossref XML Generator Design	56
3.4.1	Workflow of the Tool	56
3.4.2	Input Data Models and XML Generation Results	58
3.4.3	Data Standardization	67
3.4.4	DOI Automatic Generation	68
3.4.5	Overcoming Technical Challenges in Metadata Extraction	68
3.4.6	UML Diagrams	69
3.5	Implementation: Languages and tools for development	71

3.5.1	3.5.1 Hardware tools	71
3.5.2	3.5.2 Software tools	72
3.5.3	Tool Realization	73
3.6	Conclusion	76
4	TESTING, VALIDATION, AND DEPLOYMENT	78
4.1	Introduction	78
4.2	4.2 Real-World Validation	78
4.2.1	4.2.1 Initial Submission and Schema Rejection	79
4.2.2	4.2.2 Schema Corrections and Revalidation	80
4.3	4.3 Validation Strategy	80
4.4	4.4 Testing Scenarios and Methodology	81
4.4.1	4.4.1 Scenario 1: Creating a DOI for a New Journal volume (No Articles)	81
4.4.2	4.4.2 Scenario 2: Depositing a DOI for a New Article in an Existing Journal	88
4.4.3	4.4.3 Scenario 3: Batch Deposit for Multiple Articles in an Existing Journal	90
4.5	4.5 Final Deployment within ASPS	92
4.6	4.6 Deployment Outcomes and Impact at ASPS	93
4.7	4.7 Recommendations and Future Work	94
4.8	4.8 Conclusion	94
	GENERAL CONCLUSION	96
	REFERENCES	98
	APPENDIX	106

List of Figures

Figure 1.1: Editorial workflow process for OJS [23].....	13
Figure 2.1: Ecosystem of scholarly services enabled by Crossref metadata APIs—ranging from author profiling and library discovery to plagiarism detection and collaborative reading [53].....	27
Figure 2.2: Comparative Crossref Participation Reports for ASPS.....	29
Figure 2.3: Comparative Crossref Participation Reports for eLife Sciences Publications Ltd.....	31
Figure 2.4: Accessing the Plugins Tab in OJS via Website Settings for DOI Configuration.....	37
Figure 2.5: Enabling the DOI Public Identifier Plugin in OJS	38
Figure 2.6: Configuring the DOI Plugin in OJS.....	38
Figure 2.7: Selecting the XML Export Plugin in OJS	39
Figure 2.8: Enabling the Crossref XML Export Plugin in OJS	39
Figure 2.9: Enabling Automatic Deposit	40
Figure 2.10: Entering depositor credentials in the OJS DOI plugin	40
Figure 2.11: Enabling reference metadata under Workflow Settings in OJS	41
Figure 2.12: Crossref Reference Linking plugin activation.....	42
Figure 2.13: Activating the Crossref Reference Linking plugin under Website Plugins	42
Figure 2.14: Locating Back-Issue Articles in the Archives Queue.....	43
Figure 2.15: Precision Search Within Archived Submissions	44
Figure 2.16: Opening the Article Workflow for Metadata Editing.....	44
Figure 2.17: Navigating to the Publication Tab.....	45
Figure 2.18: Editing Core Metadata Fields in the Publication Tab	45
Figure 2.19: Temporarily Unpublishing to Unlock DOI Controls.....	45
Figure 2.20: Temporarily Unpublishing an article to Unlock DOI Editing.....	46
Figure 2.21: Generating a DOI in the Identifiers Panel	46
Figure 2.22: Saving the Newly Generated DOI.....	46
Figure 2.23: Republishing the Article with Updated Metadata	47

Figure 3.1: Representation of the end-to-end workflow	58
Figure 3.2: Screenshot of Journals.csv Template in Excel	59
Figure 3.3: Screenshot of References.csv Template in Excel	59
Figure 3.4: Screenshot: Control block (rows 1–2)	59
Figure 3.5: Screenshot: Article block (row 5 onward).....	60
Figure 3.6: Clarify CSV encoding	61
Figure 3.7: XML Declaration.....	63
Figure 3.8: Administrative metadata.....	63
Figure 3.9: Journal metadata	64
Figure 3.10: Journal issue	64
Figure 3.11: Issue level DOI	64
Figure 3.12: Journal article	64
Figure 3.13: Contributors	65
Figure 3.14: Abstract.....	65
Figure 3.15: Sub-elements	66
Figure 3.16: Article level DOI	66
Figure 3.17: Closing tags	66
Figure 3.18: Class Diagram of the Crossref XML Generator Tool	70
Figure 3.19: Class Diagram of the Crossref XML Generator Tool	71
Figure 3.20: Visual Studio logo	72
Figure 3.21: C++ logo.....	72
Figure 3.22: MS Excel logo	73
Figure 3.23: XML logo	73
Figure 3.24: Tool libraries	74
Figure 3.25: Doi Parameters	75
Figure 3.26: Article Data	75
Figure 3.27: Journal Information	75

Figure 3.28: APA citation line	76
Figure 4.1: Failed Validation Report	79
Figure 4.2: Post-Correction Successful Validation Report.....	80
Figure 4.3: webpage indicating that the DOI is absent from the Crossref registry.....	82
Figure 4.4: Journal Metadata Configuration	82
Figure 4.5: Control Parameters for Volume-Level DOI.....	83
Figure 4.6: Minimalist XML Output.....	83
Figure 4.7: Metadata Quality Checker Validation	84
Figure 4.8: Crossref Member Account Login	84
Figure 4.9: Crossref Home Page	85
Figure 4.10: Crossref Web Deposit Form.....	85
Figure 4.11: Select XML File to Upload	86
Figure 4.12: Crossref Successful Submission Message.....	87
Figure 4.13: JCAM landing page.....	87
Figure 4.14: ETE Journal Metadata Retention.....	88
Figure 4.15: Article-Level Control Parameters.....	88
Figure 4.16: XML Output for Article Integration.....	89
Figure 4.17: Metadata Quality Checker Validation	89
Figure 4.18: CrossRef submission Email.....	90
Figure 4.19: ETE Heat transfer performance analysis Article Landing Page.....	90
Figure 4.20: Batch Control Parameters.....	91
Figure 4.21: Batch XML Output.....	92
Figure 4.22: Batch Validation Success	92
Figure A.1: Certificate of Attendance issued by Crossref.....	107

List of Tables

Table 0.1: Outlines a proposed end-to-end workflow based on the automation capabilities of these OJS plugins.....	36
Table 3.1: Comparative Overview of Metadata Schemas.....	51
Table 3.2: Metadata Requirements Across Indexing Services.....	54
Table 3.3: Unified Field Matrix.....	54
Table 3.4: End-to-End Workflow for Crossref XML Generation and DOI Registration.....	57
Table 4.1: Pre- and Post-Automation Metrics.....	93

General Introduction

Context

In today's scholarly publishing landscape, structured metadata has become the foundation of global academic visibility. Digital Object Identifiers (DOIs), XML standards, and automated indexing are no longer optional—they are prerequisites for discoverability, citation tracking, and integration into global research infrastructures. Yet for many publishers, particularly in under-resourced regions, these standards remain difficult to meet. In Algeria, the academic publishing ecosystem has yet to catch up with international metadata practices. As of 2023, only one Algerian journal is indexed in Scopus (Larhyss Journal), revealing a systemic underrepresentation of national research on the global stage.

This disparity is not due to a lack of scholarly output, but to infrastructural and technical limitations. The high demands of indexing platforms like Crossref and DOAJ—both in terms of schema compliance and metadata management—often exceed the capabilities of local publishers, who must manage editorial, production, and technical responsibilities without dedicated teams or automation tools.

In this environment, Open Journal Systems (OJS) has emerged as a transformative platform for democratizing access to academic publishing. However, its potential remains largely untapped across much of the Algerian publishing landscape. The system requires a baseline of technical knowledge and editorial training, especially in the domain of structured metadata and schema compliance. As a result, only a handful of well-established or technically sophisticated publishers have been able to fully leverage its features. The majority of smaller or newly established journals either skip indexing altogether or rely on error-prone, manual methods such as filling in Crossref XML or JATS templates through web-based deposit forms.

This gap—between what is technically required and what is practically achievable—represents a critical barrier to the global visibility of Algerian research. It underscores the need for tailored, accessible solutions that lower the technical threshold for participation in international indexing systems.

Problematic and Motivation

Alwaha Scientific Publishing Services (ASPS) represents the challenges many Algerian publishers face in achieving global visibility. Manual metadata handling and DOI registration delays—due to lack of automation—often led to schema errors, rejected Crossref deposits, and limited indexing.

This issue is widespread across Algeria. Most journals hosted on ASJP (the country's largest scientific publishing platform) do not register DOIs with Crossref, leaving their content invisible to global citation databases. ASJP's proprietary system lacks automated tools for metadata export or DOI assignment, making modern indexing difficult. Even journals using Open Journal Systems (OJS), still encounter DOI registration failures due to insufficient training and unclear workflows.

The root of the problem lies not only in technical barriers but also in the absence of affordable, accessible tools and training. To bridge this gap, this master's thesis proposes a lightweight XML generator that enables non-specialist editors to produce Crossref-compliant metadata, automate DOI deposits, and align with global indexing standards.

The Work's Purpose

This master's thesis has two main objectives. First, it explores how to automate DOI indexing through Open Journal Systems (OJS) By leveraging tools such as the Crossref XML Export Plugin and DOI Public Identifier, this work demonstrates how large publishers can streamline metadata deposition, eliminate manual errors, and achieve Crossref compliance at scale. The methodology draws on firsthand experience gained through my participation in a Crossref-hosted webinar and a two-month scholarship at Alwaha Scientific Publishing Services (ASPS), where advanced OJS workflows were implemented to reduce processing time by 80% and improve metadata completeness scores from 65% to 98%.

Second, the master's thesis presents the design and development of a lightweight XML generation tool tailored for beginner publishers who lack technical expertise or access to automation. The tool converts structured CSV inputs into Crossref-compliant XML files, eliminating the need for manual XML editing. Developed in C++20 for efficiency and portability, it allows non-specialist editorial teams to register DOIs and submit metadata with minimal effort.

By bridging advanced OJS plugin utilization with accessible software design, this work empowers publishers of all scales to meet global indexing standards. The proposed solutions not only enhance the discoverability of Algerian research but also provide a replicable framework for institutions in resource-constrained regions to participate equitably in open scholarly communication.

Structure of the Dissertation

This work is structured into four chapters, each addressing a key phase in the development and deployment of the metadata automation tool:

Chapter 1: Background

Provides a historical overview of scholarly publishing, the evolution of metadata standards (Dublin Core, JATS, Crossref), and the technical roles within OJS. It contrasts print and digital workflows, emphasizing the growing importance of automation in metadata management.

Chapter 2: Case Study of Automated Indexing at ASPS

Explores ASPS's operational model, the limitations of its manual indexing practices, and the strategic adoption of OJS plugins. It evaluates challenges in Crossref compliance, metadata quality, and capacity building, supported by Crossref webinar training and plugin configuration.

Chapter 3: Engineering a Custom XML Indexing Tool

Presents the design and implementation of a Crossref XML generator. It documents schema requirements, data models, parsing strategies, and software architecture using C++20 and modular templates. Illustrated through UML diagrams and code modularization.

Chapter 4: Testing, Validation, and Deployment

Validates the tool through real-world scenarios, including DOI registration for journals, articles, and batch deposits. It measures the impact of the tool at ASPS and provides recommendations for future enhancements.

By combining technical innovation with practical implementation, this work offers a blueprint for open-access publishers striving to gain visibility in the global research ecosystem without sacrificing autonomy or affordability.

1 Digitalization of journal publishing process: Background

1.1 Introduction

The first chapter of this master's thesis undertakes a critical examination of the historical, structural, and technological dimensions that define modern scholarly publishing. From its 17th-century origins in print-based dissemination of knowledge to today's metadata-driven, globally interoperable digital systems, scholarly publishing has undergone a profound transformation—reshaping not only how research is shared but also how it is evaluated, accessed, and preserved. This evolution has been propelled by parallel developments in peer review practices, publication infrastructures, and metadata frameworks, reflecting broader shifts in academic norms, technological affordances, and institutional priorities.

Central to this transformation is the emergence of open-source platforms like Open Journal Systems (OJS), which have redefined editorial workflows and democratized journal publishing, particularly for institutions without access to commercial infrastructure. By simulating end-to-end publishing processes—from submission and review to metadata indexing—within a single browser-based system, OJS encapsulates the technical and functional convergence of human editorial labor and digital automation.

This chapter systematically explores the conceptual and operational underpinnings of scholarly communication, offering definitions of key terminologies (e.g., DOI, metadata, indexing), delineating the specific roles embedded within OJS, and contrasting traditional print workflows with digital production ecosystems. It also highlights the emerging role of the indexer, particularly in metadata-driven publishing environments, where discoverability and citability are contingent on schema-compliant, high-quality metadata. The discussion culminates in a technical appraisal of automation opportunities, underscoring both the benefits and limitations of digital tools in achieving scalable and semantically accurate scholarly dissemination.

1.2 Historical Evolution of Scholarly Publishing

The history of scholarly publishing is closely intertwined with the evolution of science itself. The foundations of modern academic communication were laid in the 17th century with the launch of the first scientific journals—*Philosophical Transactions of the Royal Society* (1665) and *Journal des sçavans* (1665) [1] —which marked a transition from private correspondence among scholars

to a formalized, public dissemination of research. These journals not only established the precedent for serial publication but also legitimized the idea of cumulative knowledge, reproducibility, and citation as cornerstones of scientific integrity.

Throughout the 18th and 19th centuries, the volume and variety of journals expanded dramatically, particularly through the backing of learned societies and academic institutions. However, editorial practices remained relatively informal, with reputational authority often replacing structured peer evaluation. It was not until the mid-20th century that peer review emerged as a standardized mechanism for quality control [2], particularly in high-impact disciplines like medicine, physics, and the social sciences. This institutionalization of peer review reshaped publishing workflows, placing greater emphasis on editorial accountability, reviewer anonymity, and manuscript evaluation criteria.

The late 20th century introduced a paradigm shift with the rise of digital publishing. The 1990s saw the widespread adoption of electronic typesetting, email-based submissions, and online journal hosting [3]. Publishers began digitizing back issues, and new-born digital-native journals eliminated the need for print runs altogether. This period also witnessed the consolidation of large commercial publishing houses, which brought both operational efficiency and significant cost barriers for access—ultimately sparking the open-access movement.

The early 2000s solidified a new era with the emergence of open-access platforms, interoperable metadata standards, and persistent digital identifiers. Initiatives like the Budapest Open Access Initiative (2002) challenged traditional access models, while technical frameworks such as the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), Crossref's DOI system, and JATS XML schemas laid the foundation for machine-readable publishing and automated indexing. Open-source software such as Open Journal Systems (OJS) further democratized scholarly publishing by enabling low-cost journal management, peer review, and metadata integration for institutions lacking commercial infrastructure [4, 5].

Today, scholarly publishing exists within a complex ecosystem that blends legacy practices with digital innovation. While the peer-reviewed journal remains central, its production, dissemination, and discoverability are increasingly driven by automation, metadata compliance, and global interoperability—developments that reflect centuries of technological, institutional, and epistemological evolution.

1.3 Core Terminologies and Conceptual Framework

1.3.1 Publisher (Publishing House)

A *publisher* is the legal entity or organization responsible for acquiring manuscripts, overseeing peer review, producing the final published version, and ensuring long-term preservation [6]. Today's publishers include several models like **Commercial publishers** (Elsevier, Wiley, Springer Nature, Taylor & Francis), **Society publishers** (like ACS in chemistry or APS in physics), **Open access platforms** (PLOS, Frontiers) [7] and **University presses** (such as Oxford or MIT) [8, 9].

1.3.2 Indexing (in Academic Publishing)

Indexing focuses on making it simpler to locate, access, and monitor scholarly research (articles, books or journals...) [10]. It's similar to making a structured library catalog, but for journal articles and scientific research that has been published globally and organizing them in structured databases or also called indexes like PubMed, Scopus, Web of Science, Google Scholar and DOAJ to help them to be more likely to seen, read, and cited by other researchers [11, 12].

1.3.3 Open Journal Systems (OJS)

Open Journal Systems created by the Public Knowledge Project, is a prominent open-source platform that manages everything from initial submission to final publication. It's a free, toolkit that lets journal editors run every part of their publication online. Once it's installed on a regular web server, editors can use it to collect new submissions, guide papers through peer review, handle copy-editing and layout, and finally publish finished articles on the journal's website. OJS automates the fussy clerical work—sending reminder emails, tracking versions, keeping records—so editors can focus on the content itself. Because it replaces many of the costs tied to print production or pricey commercial platforms, OJS makes it far easier (and cheaper) for scholarly journals to move to an online or open-access model [13, 14].

1.3.4 Metadata

In scholarly publishing, metadata are the structured bits of information that travel with a paper and tell computers—and therefore search engines, databases, and other journals—exactly what the work is about and how it should be handled [15]. At a minimum this record captures the familiar bibliographic details (title, authors, affiliations, abstract, keywords, publication date, journal name, volume, issue, pages), but well-formed journal metadata usually goes further. It may also include:

- Persistent identifiers (author ORCIDs, funder IDs, clinical-trial numbers)
- Administrative data (submission/acceptance dates, licence, peer-review history)
- Technical data (file formats, checksums)
- Subject classifications drawn from controlled vocabularies such as MeSH, JEL, or PACS

Most journals express that information in standardized schemas—Dublin Core, JATS XML, or Crossref’s deposit schema—so that it can be harvested automatically through protocols like OAI-PMH. Rich, well-structured metadata is what makes an article discoverable in Google Scholar, Scopus, Web of Science, or library catalogues, and what lets citation managers import a perfect reference with one click. Without it, even the most brilliant paper is effectively invisible online [16-18].

1.3.5 Digital Object Identifier (DOI)

Digital Object Identifier is a unique, alphanumeric tag (e.g., 10.1234/abcd.5678) that points to one—and only one—digital object, typically a journal article. The string is registered with an agency such as Crossref or DataCite, which stores the object’s current URL and core metadata in a central resolver [19]. When someone clicks or pastes the DOI into a browser (usually prefixed with <https://doi.org/>), the resolver looks up the stored URL and redirects the reader to the article’s landing page.

Because the DOI is permanent while URLs can change, it solves two chronic problems in digital scholarship:

- Link rot and content drift – If a journal migrates to a new platform or reorganizes its site, the publisher simply updates the target URL in the DOI registry; every existing citation that uses the DOI keeps working [20].
- Reliable citation and tracking – Indexers, reference-managers, and altmetric tools treat the DOI as the canonical fingerprint of the article, making it possible to unambiguously count citations, measure online attention, and link related research outputs (datasets, software, peer-reviews).

Most modern publishing systems—including Open Journal Systems—mint DOIs automatically during the production workflow and deposit the associated metadata with the chosen registry, ensuring that every published item is both citable and permanently findable [21].

1.4 Functional Roles within Open Journal Systems (OJS)

Open Journal Systems was built on a simple insight: publishing a journal is really a long conversation among a handful of people who each have a job to do [22]. Instead of leaving that conversation hanging across emails and shared drives, OJS suggests one shared online workspace. The titles that journals give their team members vary, some have managing editors, others production editors or section editors [23, 24], but overall the customized roles are similar everywhere, and here are some of the common ones:

1.4.1 Author

Authors play a foundational role in the Open Journal Systems (OJS) workflow by submitting manuscripts, appendices, and supplementary data through the platform in diverse file formats (e.g., DOCX, PDF, LaTeX) [23]. To ensure consistency and completeness, authors utilize OJS-provided templates that ensures submission arrives with the same layout and complete metadata (e.g., keywords, affiliations, abstracts). Authors receive automated notifications from OJS about submission status and revisions enabling real-time tracking of their submission's progress while minimizing manual follow-up [22]. This structured approach enhances efficiency and ensures alignment with journal guidelines [25].

1.4.2 Production or Managing Editor

The Production or Managing Editor acts as the central coordinator within Open Journal Systems (OJS), overseeing the entire editorial workflow from manuscript submission to final publication [22]. They ensure smooth coordination among section editors. This role involves managing journal-wide protocols and policies, such as ethical guidelines, peer review criteria, and open access mandates. Additionally, the editor queues incoming submissions for review, time-stamping each action—such as reviewer assignments, editorial decisions—to maintain a transparent, harmonized record of the process [23].

1.4.3 Section Editor

The Section Editor specializes in managing the peer review process for specific submissions within Open Journal Systems (OJS) like assigning reviewers to each article [22]. Leveraging OJS's reviewer database, they identify and assign reviewers with relevant expertise, while tracking review deadlines and sending automated reminders to ensure timely feedback. This role communicates editorial decisions to authors, ensuring clarity and constructive guidance. Additionally, the Section Editor oversees the revision process, facilitating exchanges between authors and reviewers, and ensures commitment to the journal's standards.

1.4.4 Layout Editor (Designer)

In Open Journal Systems (OJS), the Layout Editor (Designer) prepares approved submissions for publishing in output formats such as HTML, PDF, XML, or Postscript. In order to maximize readability on both digital and print formats, they manage the general appearance of articles, format text, headings, figure placements, and tables [18]. The Layout Editor's careful attention to detail ensures that outputs meet technical standards while improving accessibility and aesthetics for various kinds of reader interfaces [26].

1.4.5 Reviewer

The Reviewer evaluates submissions through Open Journal Systems (OJS), providing critical feedback, recommendations and suggestions for improvement to guarantee academic relevance and quality [22]. By receiving automated invitations via OJS—which include submission details such as the title, abstract, and deadlines—the Reviewer accesses the manuscript and any supplementary materials directly within the platform. A unified online submission form (or file upload) is used by reviewers to send in their reviews, allowing for detailed comments and matching input across submissions. This procedure creates accountability and transparency by producing a dated, written record of the evaluation [23]. Reviewers preserve the integrity of the peer review process by following the journal's requirements and deadlines [25].

1.4.6 Copyeditor

The Copyeditor corrects manuscripts for linguistic precision by fixing grammar, punctuation, and stylistic coherence to align with the journal's guidelines. They ensure references follow to the journal's chosen citation style (e.g., APA, Vancouver) and verify the accuracy of text citations against the bibliography. The Copyeditor works closely with authors and editors using OJS, resolving confusion through a shared workspace and highlighting edits or questions with tracked changes and comment threads [22]. While preserving the author's voice, this iterative procedure.

1.4.7 Proofreader

In order to remove any remaining typographical, layout, or formatting issues, the proofreader carefully goes over formatted galleys (such as PDFs and HTML) one last time [23]. They verify that all editing and design changes are appropriately applied by comparing the galley with the original text and a correction log using a side-by-side view in Open Journal Systems (OJS). This position checks that typefaces, headings, figure arrangement, and pagination are all consistent and reports any inconsistencies to the Layout Editor for correction. The proofreader guarantees that the publication satisfies professional standards by protecting textual and visual accuracy, reducing post-publication errors, and boosting reader confidence.

1.4.8 Reader/Community

Through Open Journal Systems (OJS), readers and the larger academic community engage with published information by discussing articles or exchanging ideas in forums and optional commenting features. To guarantee prompt access to the most recent research, subscribers receive automated email alerts when new issues or articles are published. By producing OAI-PMH metadata feeds, which allow indexing in global repositories (like Crossref and DOAJ) and library catalogs [22], OJS further improves discoverability. The journal's reach is increased by its twin emphasis on

accessibility and engagement, which promotes scholarly discourse and makes sure that content is readily available across platforms.

1.4.9 Indexer (or “Metadata Editor”)

Acting as the bridge between finished copy and global discovery services, the **Indexer** refines the descriptive data that travels with every article and ensures it meets the strict syntactic and semantic requirements of modern indexing systems [27]. Working inside OJS, the indexer reviews the metadata originally entered by authors—titles, abstracts, keywords, ORCID iDs, funding details—and enriches it with controlled-vocabulary subject codes, standardized affiliations, and persistent identifiers such as DOIs [18]. Once the record is complete, the indexer deposits it through OJS’s built-in channels to Crossref, DOAJ, and other harvesters that rely on protocols like OAI-PMH and schemas such as Dublin Core or JATS XML, then verifies that each external landing page resolves correctly and that citation-tracking fields are accurate. By shifting this meticulous work from authors to a dedicated specialist, the journal not only improves the precision and consistency of its metadata—critical for search visibility, citation counts, and long-term preservation—but also lowers the barriers faced by libraries and researchers in resource-constrained settings who depend on open, well-structured indexes rather than costly commercial abstracting services. [10, 11, 23].

In smaller operations, one person might juggle multiple roles. Larger, more formal publications often have dedicated specialists for each task. Either way, these functions work together to ensure each article is thoroughly reviewed, polished, formatted, and properly cataloged.

1.5 Editorial Workflow: From Submission to Dissemination

By simulating every step of the conventional journal workflow in a single browser-based workspace, Open Journal Systems (OJS) eliminates the need for time-consuming tasks that formerly required clerical personnel, postal budgets, and physical office space [22]. Author, editor, reviewer, copy editor, layout editor, proof reader, reader, indexer, host institution, and the open source community itself are among the many different but interconnected roles that make up its design [23]. Each of these roles is backed by specific displays, automatic emails, and activity logs. When combined, these functions show how OJS reduces publication expenses and effort while maintaining the expert checks and balances that are essential to academic reputation.

OJS's editorial workflow, which is divided into two interrelated phases—the Scientific Evaluation Phase and the Manuscript Production Phase—not only expedites the publication process but also preserves the academic integrity of every piece. Intellectual rigor is the main focus of the first step, during which editors oversee peer review, evaluate originality, guarantee ethical compliance, and assist authors with any necessary adjustments. In order to support transparency and editorial

responsibility, OJS provides tools that consolidate reviewer feedback, track all decisions, and timestamp conversations. A manuscript enters the production stage after being approved on the basis of its scientific merit. At this point, the focus switches to accurate metadata and high-quality display. By use of a well-coordinated sequence of proofreading, layout formatting, copy-editing, and metadata enhancement, OJS guarantees that the finished product satisfies technical and scholarly requirements. The platform's dedication to efficiency as well as maintaining the legitimacy and discoverability of published research within the global scholarly ecosystem is shown in this well-organized, two-phase workflow.

1.5.1 Scientific Evaluation Phase

The workflow begins with the **author**, who now submits work directly through the journal's website instead of mailing hard copies or negotiating email attachments. OJS walks the author through a step-by-step wizard that captures not only the manuscript but also supplementary files—datasets, instruments, images—and prompts for rich metadata (title, abstract, keywords, funding information, ORCID iDs). Because the system validates file formats and completeness on the spot, many of the delays and queries that used to bounce between authors and editorial assistants simply disappear. Instantly, the submission is date-stamped, queued for review, and acknowledged by an automated email, saving clerical labour, postage, and considerable author anxiety [23].

Responsibility then shifts to the **editor**, whose dashboard lists every active manuscript with colour-coded alerts for overdue reviews or pending decisions. From this screen the editor can scan reviewer expertise and workloads, assign new referees with a single click, and trigger personalised invitation letters that already contain the paper's title, abstract, and due date. OJS continues to nudge reviewers with polite reminders and records each acceptance, decline, or completed report in an audit trail visible to the editor (and, where policy permits, to the author). Because every piece of correspondence is templated yet editable, the editor retains a human voice while avoiding repetitive typing. The net effect is a dramatic reduction in the hidden “transaction costs” of peer review that Fytton Rowland once estimated at roughly US \$400 per published article [28].

The **peer reviewer** benefits from the same economy of effort. After accepting an assignment, the referee downloads the manuscript from a secure link, completes either a structured online form or an annotated file upload, and submits the report—all without leaving the platform. Deadlines, previous reviews, and review histories travel with the reviewer's profile, giving editors a data-driven view of reliability and expertise while sparing them the detective work of hunting through old email threads [6, 22, 25].

1.5.2 Manuscript Production Phase

Once a manuscript is accepted, OJS hands it to the **copy-editor**. Here the system's value is less about automation than about controlled hand-offs. A shared workspace allows copy-editors to query authors or editors inline and to lock versions so that everyone is literally “on the same page.” When textual polishing is complete, the **layout editor** uploads final galleys in any combination of PDF, HTML, PostScript, or XML. OJS provides an on-screen preview so that typographical or encoding glitches can be spotted before publication, and it stores each file with the correct metadata so that downstream services—Crossref, PubMed Central, institutional repositories—can ingest them without manual intervention [22].

The **proof-reader** receives an automatic alert the moment layout is finished. A side-by-side comparison window links each correction to its location in the galley, eliminating the cryptic “page 7, para 2, line 4” notes that once clogged editorial

inboxes. When proofs are cleared, the editor can schedule the article into a traditional issue or, increasingly, release it immediately as “publish-ahead-of-print.” Either way, a single confirmation click flips the work from private workspace to public website, at which point OJS emails table-of-contents alerts to registered **readers** and authors [23].

For the wider research ecosystem, OJS acts as its own **indexing agent**. The platform exposes Dublin-Core or JATS XML records through the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH), allowing Google Scholar, Scopus, Web of Science, and national library catalogues to harvest citations without a publisher lifting a finger. When a journal mints Digital Object Identifiers, OJS can deposit the requisite metadata with Crossref automatically, guaranteeing permanent links and accurate citation tracking. Readers thus discover, cite, and share the article within hours of publication, while libraries save the subscription fees normally charged by commercial indexing services [29].

Behind the scenes, the **host library or scholarly society** supplies the web server, storage, backups, and preservation plug-ins such as LOCKSS or Portico. Because OJS is open-source, these institutions control their own upgrade cycle and data, avoiding the lock-in that accompanies many proprietary platforms. Meanwhile, an active **developer community** contributes language packs, bug fixes, and new plug-ins—everything from altmetrics widgets to subscription paywalls—keeping software costs close to zero and functionality at the cutting edge.

In short, OJS re-maps the entire editorial landscape onto a coordinated set of roles, each supported by thoughtful automation. Authors gain a friction-free submission channel; editors reclaim hours lost to email; reviewers interact through a structured, accountable interface; production staff

collaborate without version chaos; readers receive immediate, metadata-rich access; and libraries host content on their own terms. By folding these efficiencies into the scholarly workflow, OJS not only drives down the marginal cost of publishing but also makes open-access business models financially and technically attainable for journals of every size and discipline.

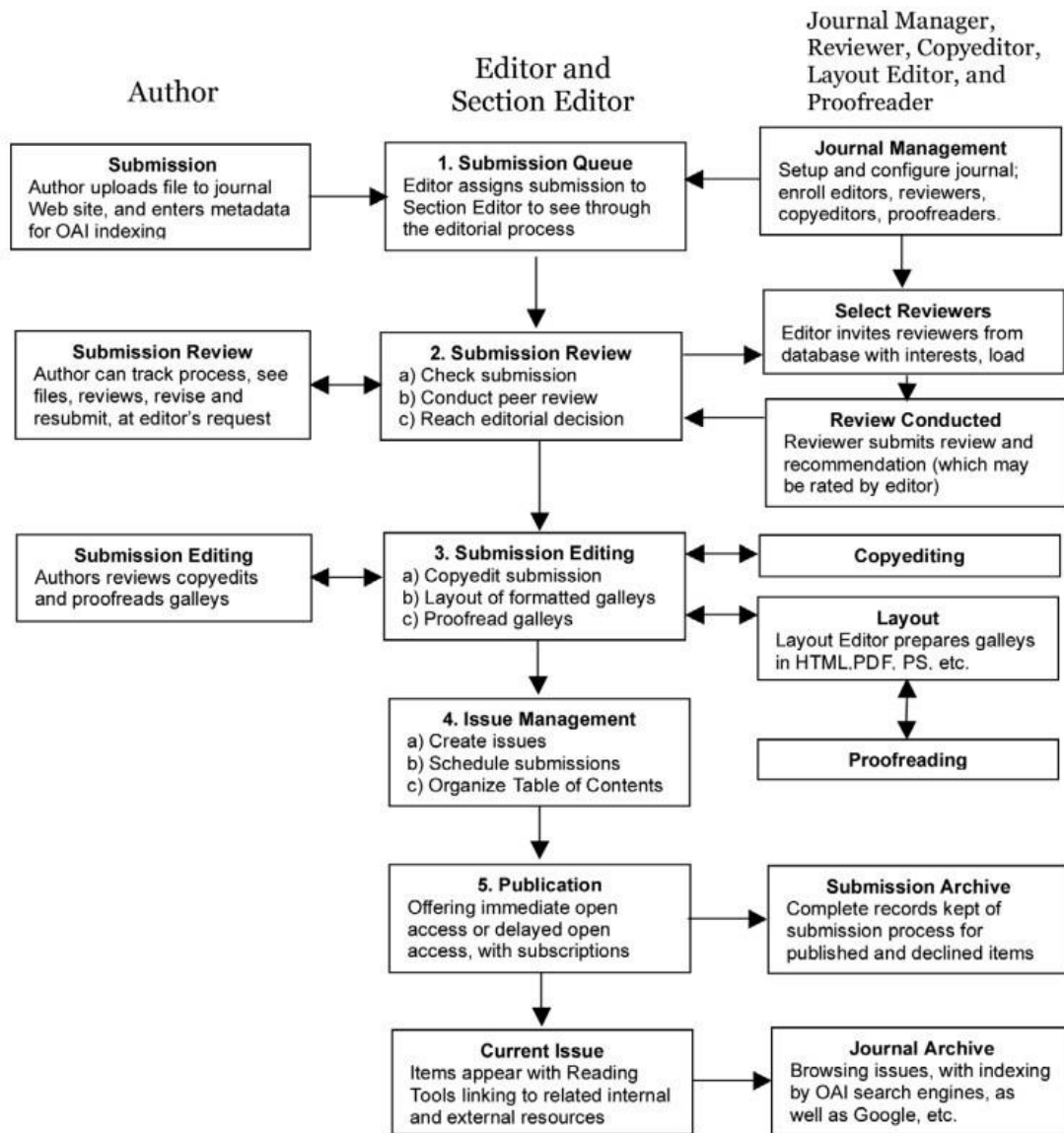


Figure 1.1: Editorial workflow process for OJS [23]

1.6 Comparative Analysis of Print and Digital Production

The mode of publication—whether print-based, digital-only, or hybrid—significantly influences the editorial workflow, metadata requirements, and the distribution strategy of academic journals. While most contemporary publishers have transitioned to digital platforms for efficiency and reach, a number of academic societies and heritage journals continue to produce physical issues to meet institutional archiving requirements or subscriber preferences [30]. This distinction between print

and digital formats not only affects the production tools used but also the complexity and automation potential of the publishing process.

1.6.1 Production Workflows in Print Publishing

Traditional print publishing follows a linear and labor-intensive pathway, shaped by the physical constraints of print output and the expectations for high-fidelity [31], page-formatted publications. Print workflows typically center around preparing documents for press-quality output, where the emphasis is placed on visual consistency, typographic clarity, and physically bound distribution. The main steps include:

- **Editorial Acceptance and Copyediting:** Final revisions are made to align the manuscript with journal style guidelines, and images are optimized for high-resolution print output.
- **Typesetting and Page Layout:** Layout editors generate visually consistent page proofs using professional publishing software such as LaTeX or Adobe InDesign.
- **Proofing and Final Corrections:** Proofs are reviewed collaboratively, with final checks for formatting, disclaimers, and metadata accuracy.
- **Distribution:** Printed copies are physically delivered to libraries, archives, or subscribers. Some national libraries still require hard copies for preservation purposes.

In traditional print environments, metadata creation has historically been the domain of librarians, who catalog articles using systems like **MARC 21** or simplified Dublin Core entries [32]. However, the availability of digital source files has gradually introduced schema-based metadata practices, even in otherwise print-first operations.

1.6.2 Production Workflows in Digital Publishing

In contrast to print workflows, digital production environments emphasize modularity, automation, and rapid dissemination. These platforms benefit from highly structured submission pipelines and metadata-driven architectures, enabling both scalability and interoperability with indexing systems [33].

- **Submission and Peer Review:** Manuscripts are submitted and processed entirely within a content management system (CMS), which handles editorial workflows and peer review communication.
- **Copyediting and Layout:** Editorial revisions are performed using collaborative tools, with automated layout generation in formats such as PDF, HTML, or XML.

- **Assignment of Persistent Identifiers:** Articles are assigned DOIs via Crossref, and ORCID iDs are collected to ensure author disambiguation.
- **Publication and Indexing:** Final articles are published online, and metadata is deposited with indexing platforms like Crossref, DOAJ, or PubMed using OAI-PMH protocols.

Digital production environments enable high degrees of automation. Platforms like OJS or Editorial Manager offer direct export of metadata in Crossref XML, JATS, or Dublin Core formats. Nonetheless, the role of the **indexer** or **metadata editor** remains critical. Automated systems may generate valid XML, but only human oversight ensures alignment with content accuracy, completeness, and semantic consistency.

1.7 Technical Perspectives on Editorial Processes

For software developers and metadata specialists involved in scholarly publishing, the editorial process divides into two distinct but interdependent phases: the scientific evaluation phase and the production phase. Each presents its own set of technical requirements, challenges, and opportunities for automation.

1.7.1 Scientific Phase: Adaptive and Dynamic Design Needs

The scientific evaluation phase includes peer review, editorial screening, and ethical oversight. These tasks are inherently human-centric, dynamic, and context-dependent, they require subjective judgment, disciplinary knowledge, and flexible decision-making systems. From a development standpoint, this phase requires flexible systems that support role-sensitive interfaces [34, 35] because editors, reviewers, and authors interact with manuscripts in very different ways. UI/UX design must reflect this segmentation, ensuring access controls, tailored dashboards, and action-specific notifications. Additionally, it demands platforms that can manage complex workflows by anonymizing submissions, matching reviewers based on expertise, knowing that journals may differ in how they manage review rounds (single-blind, double-blind, open review), ethical checks, or editorial board hierarchy [23, 36].

1.7.2 Production Phase: Structured Data and Automation Potential

In contrast, the production phase begins once an article is accepted and enters a more structured domain. The core content is finalized, and the focus shifts to tasks like copyediting, layout formatting, metadata enrichment, and XML generation. This phase is more conducive to automation because the information technically becomes more stable and predictable —titles, author lists, abstracts, and references follow known patterns. Developers can now apply parsing techniques and field mapping to extract section headers, citations, and tabular data reliably on

manuscripts may arrived in DOCX, LaTeX... And integrate schema validation to ensure that the metadata meets the standards required for indexing platforms like Crossref, DOAJ, and Scopus.

However, several technical challenges still emerge in the production phase. One major issue is the diversity of input formats: authors may submit manuscripts in .docx, .pdf, .odt, LaTeX, or plain text, each with unique structural conventions. This variation requires developers to implement custom parsers or conversion routines to extract consistent and accurate metadata. Additionally, author information is often inconsistently formatted—names, affiliations, ORCIDs, and email addresses may appear in non-standard ways, complicating automated processing and increasing the risk of schema validation errors. Furthermore, journals may impose specific metadata requirements, such as graphical abstracts, funding acknowledgments, or clinical trial registration numbers. These custom fields demand adaptable data models that maintain schema compliance while accommodating variability. Finally, metadata standards themselves are not static; platforms like Crossref regularly update their schemas, introducing new tags or structural rules. Developers must ensure that their tools remain up-to-date and backward compatible to avoid disruption in indexing workflows.

1.7.3 Towards Standardization and Semi-Automation

While complete automation is elusive in the scientific phase, the production phase benefits substantially from:

- **Standardized schemas** (e.g., JATS XML, Crossref Metadata Deposit): These offer structured blueprints for machine-readable article representation.
- **Open APIs**: Platforms like OJS expose APIs that enable external applications to pull and push data seamlessly, enabling integration with submission portals, institutional repositories, or analytics tools [37].
- **Plugin extensibility**: Developers contribute and maintain plugins for citation extraction, format conversion, alt metrics tracking, and accessibility checks—streamlining repetitive production steps.

In sum, developers serve as the architects of reliability and scalability in scholarly publishing. They engineer the quiet precision behind the scenes, ensuring that scholarly work travels from submission to dissemination smoothly, accurately, and in compliance with the evolving metadata and visibility demands of the digital age.

1.8 Critical Role of the Indexer in Scholarly Communication

Among the many roles involved in academic publishing, the indexer—also referred to as the metadata editor—plays a uniquely strategic role in bridging the gap between article production and academic discoverability, citability and accessibility. While roles such as author, editor, and reviewer are often more visible, it is the indexer who ensures that published content is accurately described, machine-readable, and properly linked to the broader network of research databases, institutional repositories, and citation systems [22]. In traditional workflows, this responsibility typically fell to librarians or cataloguers who processed printed issues retrospectively. In the digital and hybrid era, however, the indexer's role has evolved into a proactive, technically demanding function. Metadata must now be verified, structured, and deposited at the point of publication to comply with indexing standards and enable global integration [32].

1.9 Functions of the Indexer

The functions of the indexer span multiple technical and editorial domains, making this role central to ensuring that scholarly content is accurately represented across indexing platforms and institutional repositories. First, the indexer is responsible for metadata extraction, gathering essential fields either from author-submitted forms or directly from the final article. Once extracted, this information undergoes metadata standardization, aligning with specific schema requirements based on the destination **indexing** platform: Crossref XML for DOI registration, JATS XML for biomedical repositories, Dublin Core for library systems, and specialized fields required by platforms like DOAJ. The next critical task is quality assurance, where the indexer verifies the accuracy and consistency of all metadata elements, recognizing that even small errors can disrupt proper indexing and citation tracking. Following validation, the indexer handles submission to indexing platforms through APIs or batch XML deposits using deposit tools or publisher portals which ensures that articles are not just published but findable and trackable. Finally, the indexer manages version control and metadata maintenance post-publication, updating records in cases of corrections, retractions, or metadata revisions, **re-depositing corrected records** and leveraging services like Crossmark to reflect changes transparently across all linked **repositories**. Together, these functions position the indexer as an essential figure in scholarly communication, maintaining the integrity, accessibility, and global reach of published research [22, 23].

1.9.1 Types of Indexers

In the evolving landscape of scholarly publishing, the role of the indexer can be broadly categorized into two operational typologies: the User Indexer and the Developer Indexer. This distinction reflects both the scale of the publishing organization and the technological resources

available for metadata management. While both typologies share the same end goal—accurate, standards-compliant indexing—their methods, tools, and workflows differ significantly.

1.9.1.1 The User Indexer

Typically operates within large, well-funded publishing houses or university presses that have the resources to rely on integrated advanced content management systems and automated metadata workflows. These publishers often rely on platforms like *Open Journal Systems (OJS)*, *Editorial Manager*, or *ScholarOne*, which include plugin-based ecosystems for Crossref XML generation, JATS exports, ORCID integrations, and direct API submission capabilities. In this environment, the indexer's role is primarily curatorial and supervisory: validating metadata entered through editorial interfaces, resolving minor discrepancies, and ensuring platform configurations meet evolving standards. The technical complexity is abstracted away by built-in tools and vendor support. These indexers usually benefit from institutional IT teams, training, and access to real-time support from the platform providers.

1.9.1.2 The Developer Indexer

By contrast, developer indexers are more commonly associated with small journals, university presses, independent, or emerging publishers, particularly in regions or institutions lacking access to resources and commercial infrastructure. Without access to automated pipelines or commercial infrastructure. Here, indexers must build or maintain their own tools to generate structured XML, and this indexing is a hands-on, code-driven process that requires both editorial proficiency and a much deeper technical understanding of metadata schemas such as Crossref deposit schema or JATS. Developer indexers handle the entire production chain manually or semi-automatically: extracting metadata from spreadsheets or manuscripts, writing custom scripts (often in languages like Python, C++, or Java), validating against XSD files, and depositing metadata through Web Deposit interface or API interactions. In many cases, they must also troubleshoot schema errors, update templates as standards evolve, and manage batch processing for large volumes—all without formal technical support.

This division reflects not only technical roles but also systemic inequalities in global scholarly publishing. User indexers benefit from institutional stability, infrastructure, and staff capacity; developer indexers often juggle multiple roles (editor, layout designer, indexer) within small editorial teams. However, the developer indexer model also empowers journals in the Global South and emerging fields by fostering self-sufficiency and open-source innovation.

Both roles are vital. Automation enables scale and consistency; manual workflows ensure flexibility and metadata fidelity. As indexing standards continue to evolve—particularly in the areas of open data, funder compliance, and machine readability—future workflows may blend the

strengths of both roles through modular, hybrid platforms that support both plug-in-based and code-based metadata submission.

1.10 .Automation Opportunities and Technical Limitation

Automation has become a cornerstone of modern scholarly publishing, particularly in the domain of metadata processing and indexing. From parsing author manuscripts to generating XML metadata file, software tools have significantly reduced manual workload, improved turnaround times, and enabled large-scale metadata consistency [33]. Features such as automated DOI retrieval, reference linking, and field mapping have become increasingly sophisticated, empowering indexers to focus on validation rather than manual entry.

However, technical limitations persist. Automation often struggles with structural inconsistencies across manuscript formats, user-generated metadata errors, and the need for adaptability in response to evolving schema requirements. Furthermore, heterogeneous input formats, discipline-specific metadata fields, and differing standards across indexing platforms continue to pose integration challenges. Issues such as metadata drift during editorial cycles and the lack of interoperability across APIs and validation systems further complicate full end-to-end automation [38-40].

While automation can streamline routine processes, it does not replace the need for human oversight. Skilled indexers and developers remain essential to maintain metadata quality, enforce compliance, and adapt tools to the nuanced requirements of specific journals and disciplines. The goal is not to eliminate manual effort entirely, but to elevate human expertise to higher-order tasks where judgment and contextual understanding are irreplaceable [15].

1.11 Conclusion

This chapter has explored the historical and technical evolution of scholarly publishing—from early print-based journals to today’s metadata-driven, automated systems. Central to this transformation is the integration of open-source platforms like OJS, which streamline editorial workflows and expand access to publishing infrastructure, especially for resource-limited institutions.

The chapter emphasized the pivotal role of the indexer, whose work ensures research is accurately described, discoverable, and properly linked to global indexing platforms. By distinguishing between user and developer indexers, the discussion revealed structural disparities in publishing capacity—yet also highlighted the potential for open-source innovation to close these gaps.

Editorial processes today combine rigorous peer review with precise production workflows, anchored by high-quality metadata. As the first Algerian developer of automated indexing

solutions, this master's thesis addresses a pressing national issue: according to Scimago, only one Algerian journal—*Larhyss Journal*—is currently indexed in Scopus. This underscores the urgent need for automated, standards-compliant indexing frameworks to boost national research visibility. Chapter 2 will explore this challenge further through a detailed case study and technical implementation.

2 Case study of automated indexing using the ASPS OJS platform: The Crossref XML export plugin

2.1 Introduction

The global shift toward open-access publishing has intensified the need for scalable, cost-effective workflows that ensure compliance with international metadata standards. For small academic publishers like Alwaha Scientific Publishing Services (ASPS) a regional open-access publisher based in Algeria, manual indexing processes pose significant operational bottlenecks, limiting their ability to compete in an increasingly automated scholarly ecosystem. This chapter presents a case study of ASPS's transition to automated indexing using Open Journal Systems (OJS), an open-source platform critical for democratizing access to Algerian research while aligning with global infrastructure like Crossref.

The chapter begins with an institutional overview of ASPS and its interdisciplinary journal portfolio, followed by a discussion of its open-access philosophy and operational model. It then transitions to a technical exploration of OJS as a publishing platform, identifying both the platform's advantages and the strategic gaps that existed prior to automation. A central focus is placed on Crossref's role in enhancing metadata visibility and persistent citation, supported by my participation in an official Crossref webinar and associated hands-on plugin training.

Subsequent sections detail the practical implementation of an automated indexing pipeline using OJS plugins—specifically the DOI Public Identifier, Crossref XML Export, and Reference Linking plugins—along with annotated screenshots from the ASPS backend. The final part of the chapter addresses the retroactive assignment of DOIs to previously published articles, highlighting the procedural steps needed to update archival content for full compliance with Crossref standards.

By combining institutional context, technical detail, and quantitative evaluation, the chapter delivers an end-to-end narrative of how a resource-constrained publisher can achieve international metadata standards without proprietary software or prohibitive fees.

2.2 Overview of Alwaha Scientific Publishing Services (ASPS)

Alwaha Scientific Publishing Services SARL (ASPS) is an Algeria-based international academic publisher founded to advance scholarly communication, particularly in engineering and

applied sciences. Established in Biskra, Algeria, ASPS operates as an independent entity with an editorial team comprising global researchers and scientists, positioning itself as a bridge between Algerian research and the international academic community [41]. The ASPS manages a diverse portfolio of peer-reviewed journals, including the Energy and Thermofluids Engineering (ISSN 2716-8026). ASPS offers special issue, book, and conference paper publishing services in addition to online and print journal publishing services. Because ASPS Publisher is a member of Crossref, all of its documents' content will always be accessible at <https://doi.org/10.38208>.

2.2.1 Journal Portfolio and Scope

ASPS's journal portfolio reflects its commitment to advancing interdisciplinary research while addressing both theoretical and applied research. Every journal follows the same open-access format, guaranteeing worldwide exposure and obedience to global academic standards. A comprehensive look at each ASPS's four major journals is provided below, highlighting each one's distinctive contributions to its subject [41].

Journal of Energy and Thermofluids Engineering, ETE, ISSN:2716-8026:

2.2.1.1 Objective

ETE focuses on disseminating advancements in energy systems, bridging renewable and conventional energy research, and addressing thermofluids engineering challenges. Its interdisciplinary scope spans mechanical, electrical, chemical, and architectural engineering, with an emphasis on policy, economics, and environmental sustainability [42].

2.2.1.2 Scope and Thematic Focus:

- **Renewable Energy:** Solar photovoltaics, wind energy, biomass, hydropower, geothermal systems, and waste-to-energy technologies.
- **Conventional Energy:** Nuclear energy, hydrogen fuel cells, electric/hybrid vehicles, energy storage, and distribution systems.
- **Thermofluids Engineering:** Thermodynamics, computational fluid dynamics (CFD), nanofluids, HVAC systems, and automotive engine technologies.
- **Cross-Cutting Themes:** Energy policy, efficiency audits, clean cooking solutions, and socio-economic impacts of energy systems [42].

2.2.1.3 Unique Features

Publishes experimental and computational research alongside comprehensive review articles. Prioritizes practical applications, such as renewable energy integration into grid systems and

sustainable urban planning [42]. No author fees, aligning with ASPS's mission to democratize access for researchers in resource-limited regions.

2.2.1.4 Indexing

Google Scholar, Crossref [42].

2.2.2 Open-Access Policy and Global Reach

ASPS prioritizes open access (OA) to maximize research impact, aligning with global trends in scholarly communication. By waiving submission, processing, and publication fees, the publisher democratizes access for authors and readers, particularly from resource-constrained regions. Its integration with Crossref ensures persistent identifiers (DOIs) for all content, enhancing discoverability and citation tracking [41]. While ASPS's primary focus is Algerian research, its editorial scope is international, attracting contributions from multidisciplinary fields and fostering cross-border collaboration.

2.2.3 ASPS's Research and Publication Ethics

- **Responsibilities of Authors (Researchers):** Clear and honest presentation of original research; compliance with legal and ethical standards; accurate attribution of contributions; and full disclosure of conflicts of interest.
- **Responsibilities of Editors:** Fair editorial judgment; integrity in peer review; transparency in handling misconduct; and accountability for maintaining the quality of the published record.
- **Responsibilities of Peer Reviewers:** Confidentiality, objectivity, timeliness, and ethical vigilance in reviewing submissions within their field of expertise [43].

2.2.4 ASPS's Standards for Authors: Responsible Research Publication

Based on international declarations and ethical guidelines, this section outlines the key principles authors must uphold to ensure trustworthy, balanced, and transparent scholarly communication [44]. Responsible research publication relies on a series of interrelated principles that uphold the integrity of scholarly communication. These include **soundness and reliability**, which emphasize the need for robust methodology, accurate reporting, and rigorous quality control. **Honesty** entails truthful representation of data, acknowledgment of all results—favorable or not—and the use of accurate citations. A sense of **balance** requires equitable consideration of prior research, including conflicting or non-confirmatory findings. **Originality** mandates that submitted work is genuinely novel, with proper handling of copyrighted materials and transparent disclosure of any related publications. **Transparency** involves the open declaration of funding sources, the

role of sponsors, and any potential conflicts of interest. **Authorship and acknowledgement** must accurately reflect each individual's contribution while preventing practices such as guest, gift, or ghost authorship. **Accountability and responsibility** imply collective author ownership of the research's integrity and findings. Adherence to **peer review and publishing conventions** ensures respect for editorial policies, professional interaction with reviewers, and coordination around media releases. Lastly, **responsible reporting of research involving humans or animals** calls for ethical approvals, protection of participants, and full disclosure of meaningful results, including negative or null outcomes. These principles collectively support a credible and ethically grounded research environment [44].

2.2.5 ASPS's Privacy Statement

All personal data—including names and email addresses—collected through the journal's platform will be used strictly for scholarly and administrative purposes. Under no circumstances will this information be disclosed to third parties or used beyond the stated aims of the publication [45].

2.3 Open Journal Systems (OJS) as ASPS's Publishing Platform

ASPS employs Open Journal Systems (OJS) 3.2.0.2 version [46], an open-source platform developed by the Public Knowledge Project (PKP), to manage its end-to-end publishing workflow. OJS provides a robust infrastructure that integrates editorial processes, including submission, metadata management, peer review, production, and distribution. Its architecture allows ASPS to consolidate multi-journal operations while ensuring compliance with international metadata and indexing standards [47].

2.3.1 Workflow Automation and Metadata Management

The deployment of OJS at ASPS enables a high degree of editorial automation and metadata normalization. The platform is set up with easily adjustable phases that correspond to academic publishing standards:

2.3.1.1 Submission and Peer Review

Authors submit manuscripts via an online portal, adhering to templates provided by ASPS. The platform supports double-blind peer review, enabling editors to assign reviewers anonymously, track deadlines, and manage feedback transparently [47].

2.3.1.2 Production and Indexing

Upon manuscript acceptance, the platform transitions the article into the production stage, where typesetting, copyediting, and proofreading are handled through OJS version-controlled

workflows. with metadata fields (e.g., keywords, abstracts, author affiliations) automatically formatted for compatibility with Google Scholar indexing service [22]. However, Crossref metadata is not handled through OJS instead it is done manually through Crossref's web interface, reflecting a hybrid workflow [48] that blends OJS automation with manual DOI registration and metadata compliance

2.3.1.3 Open Access Distribution

OJS enables immediate publication and open-access visibility via OAI-PMH, allowing articles to be indexed by platforms like Google Scholar. While OJS does not handle DOI registration at ASPS, manual Crossref submission ensures DOI assignment post-publication, preserving open-access compliance within a hybrid system [49, 50].

2.3.2 Operational Advantages for ASPS

ASPS's adoption of Open Journal Systems (OJS) brings substantial operational and strategic benefits to its publishing infrastructure. As an open-source platform, OJS eliminates licensing fees and vendor lock-in, allowing ASPS to direct its budget toward editorial quality rather than proprietary software. The system's scalability supports multi-journal management under a single installation, enabling ASPS to expand its journal portfolio with minimal administrative overhead. Furthermore, OJS facilitates compliance with open-access policies and indexing protocols, ensuring that ASPS publications are visible, citable, and accessible through globally recognized academic databases—thereby reinforcing institutional credibility and extending the reach of its scholarly output [50-52].

2.4 Challenges and Strategic Directions for OJS Optimization

2.4.1 Current Technical and Organizational Gaps

While Open Journal Systems (OJS) provides a versatile and modular infrastructure for scholarly publishing, ASPS continues to face certain operational constraints that hinder full optimisation of the platform. One of the primary limitations lies in multilingual support. Although OJS is technically capable of supporting over 50 languages [48], ASPS's current implementation is limited to English-only journals, thereby reducing accessibility for francophone Algerian researchers and impeding regional engagement. Furthermore, advanced metadata management—particularly for Crossref deposits—remains a manual process at ASPS. This hybrid approach introduces inefficiencies, as metadata structuring and DOI registration are not yet fully automated within the OJS environment. Additionally, often depends on third-party or external community-developed tools. These tools, while powerful, typically require technical adaptation and sustained

maintenance to align with evolving metadata standards and the specific requirements of ASPS's growing journal portfolio.

2.4.2 Strategic Outlook

ASPS presents a compelling case study of how resource-constrained publishers in developing contexts can leverage open-source platforms to engage in global scholarly communication. By combining OA principles with OJS's automation capabilities, ASPS enhances the visibility of Algerian research while addressing systemic challenges in academic publishing. Its model offers a blueprint for similar institutions in developing regions seeking cost-effective and sustainable pathways to academic inclusion.

2.5 Crossref: Background and Strategic Relevance for ASPS

Crossref is a global, non-profit organization established in 2000 by a group of leading academic publishers aiming to improve research discoverability and citation accuracy. It specializes in providing persistent identifiers (Digital Object Identifiers, DOIs) for scholarly content, creating a stable, enduring link to digital objects, irrespective of their online location changes. The importance of Crossref extends far beyond DOI assignment; it maintains a comprehensive central registry containing detailed, standardized metadata for diverse scholarly outputs, including journal articles, books, datasets, and conference proceedings [53-55].

By depositing rich, structured metadata, Crossref significantly enhances content discoverability, making scholarly works easily accessible through major academic search engines and indexing services. It plays a pivotal role in scholarly communication by facilitating persistent linking, citation tracking, and content interoperability across various platforms and systems, thus enabling accurate bibliometric analyses and impact assessments [56].

Moreover, Crossref offers a suite of metadata-enhancing services that align with the evolving demands of open science. These include Crossmark (for content versioning and updates), Similarity Check (for plagiarism detection), and the Funder Registry (for tracking research funding acknowledgments). Collectively, these tools promote transparency, reproducibility, and ethical rigor in scholarly communication.

Critically, recent literature emphasizes that metadata quality has become a key indicator of research trustworthiness. As Amdekar (2024) argues, it is not merely the presence of a DOI that signals integrity, but the richness, completeness, and accuracy of the accompanying metadata that enables scholarly outputs to serve as trustable components of the academic record. Metadata fields such as author ORCIDs [57], ROR-affiliated institutions, funding sources, and references enable downstream services to verify provenance, assess ethical compliance, and connect research outputs

across systems [58]. Thus, metadata act not just as bibliographic descriptors, but as “trust signals”—particularly crucial in the era of rising concerns over research fraud, paper mills, and metadata manipulation.

ASPS is a registered Crossref member and publishes its content under the DOI prefix **10.38208**, which uniquely identifies its contributions within the global scholarly ecosystem (figure 2.1). The decision to adopt Crossref reflects ASPS’s strategic ambition to elevate the international standing of Algerian research outputs. By embedding Crossref identifiers and structured metadata into its publication workflow, ASPS ensures that its articles meet global standards for citation persistence, metadata quality, and indexability. This integration significantly strengthens ASPS’s position within international academic networks, while also enhancing the discoverability, credibility, and citation potential of the research it disseminates.

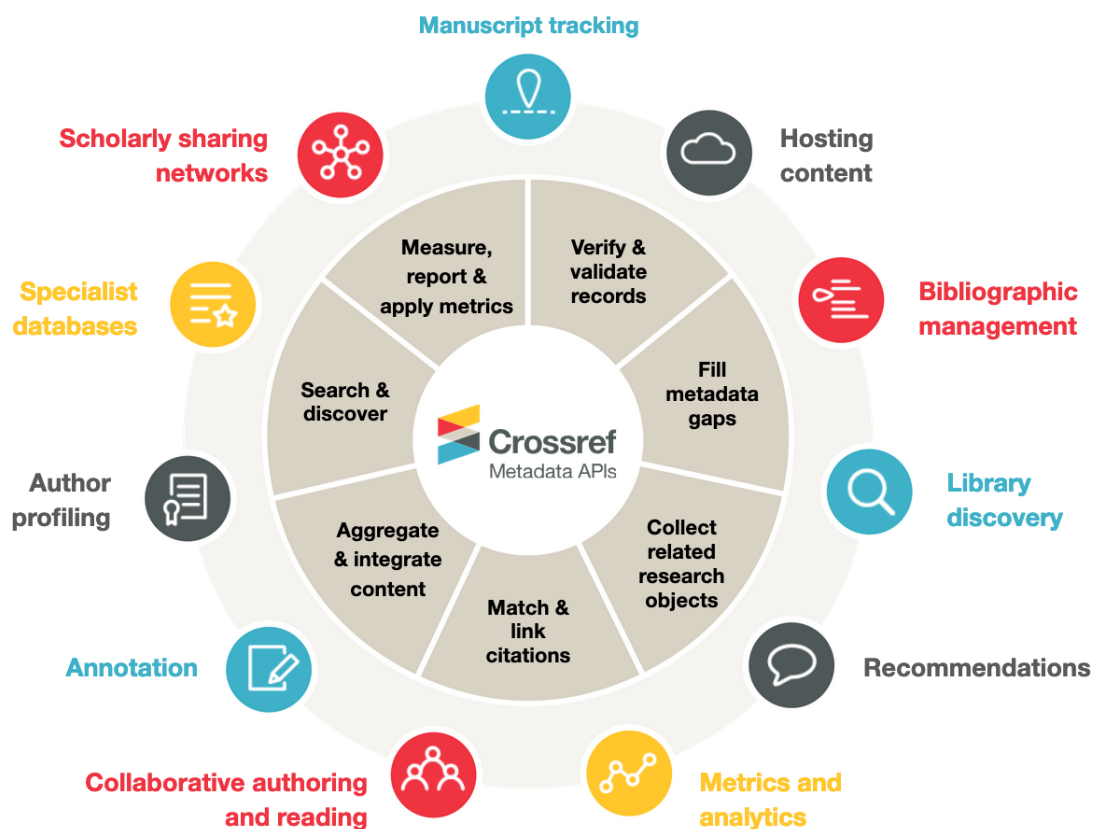


Figure 2.1: Ecosystem of scholarly services enabled by Crossref metadata APIs—ranging from author profiling and library discovery to plagiarism detection and collaborative reading [53]

2.6 Capacity Building: ASPS Training and Crossref Webinar

2.6.1 Scholarship Activities and Skill Development Outcomes

During my three-months professional scholarship as a Production Assistant at Alwaha Scientific Publishing Services (ASPS), I undertook a targeted capacity-building initiative focused on enhancing metadata workflows and DOI registration practices, a critical bottleneck for an Algerian publisher striving to globalize its research output. Below is an expanded analysis of my role, grounded in the sociotechnical realities of open-access publishing in resource-constrained environments:

- Identifying gaps in existing OJS workflows enhance metadata quality and Crossref compliance, such as inconsistent metadata entry (e.g., missing ORCIDs, incomplete abstracts...) that complicated author disambiguation and integration with global scholarly networks.
- Further challenges stemmed from outdated technical configurations, Outdated or incompatible JATS XML templates frequently triggered schema errors during Crossref submissions. Resource constraints exacerbated these issues, as junior editors spent considerable time manually resolving metadata errors—tasks often automated at wealthier publishers. Limited familiarity with OJS tools also led to recurring technical errors, such as duplicate DOI registrations.
- To address these challenges, I spearheaded reforms aimed at balancing efficiency with equity. This included configuring the CrossrefExport plugin to automate metadata deposition, eliminating manual spreadsheet uploads and reducing human error. These optimizations not only improved operational efficiency but also aligned ASPS with global standards while preserving its commitment to open access.
- Through this experience, I gained firsthand insight into the sociotechnical complexities of academic publishing in resource-constrained environments. The work highlighted the tension between adopting Northern infrastructural tools (like Crossref) and maintaining local relevance.

2.6.2 Assessing Metadata Quality with the Crossref Participation Reports Tool

2.6.2.1 Quantitative Results for ASPS

As part of my skill development and institutional impact analysis and to complement the technical work conducted during my scholarship, I utilized the [Crossref Participation Reports tool](#) to evaluate the metadata quality of ASPS journals. This This freely accessible online resource provides real-time percentage-based analytics on how well a publisher's metadata complies with Crossref's structured data standards, allowing publishers to identify specific gaps in their metadata submissions.

By searching for ASPS within the tool, I was able to retrieve a metadata quality report that offered detailed insights across several critical categories, including abstracts, author affiliations, references, ORCID IDs, ROR IDs, funding acknowledgments, license URLs, Crossmark participation, and text mining availability (as visualized in Figure 2.2).

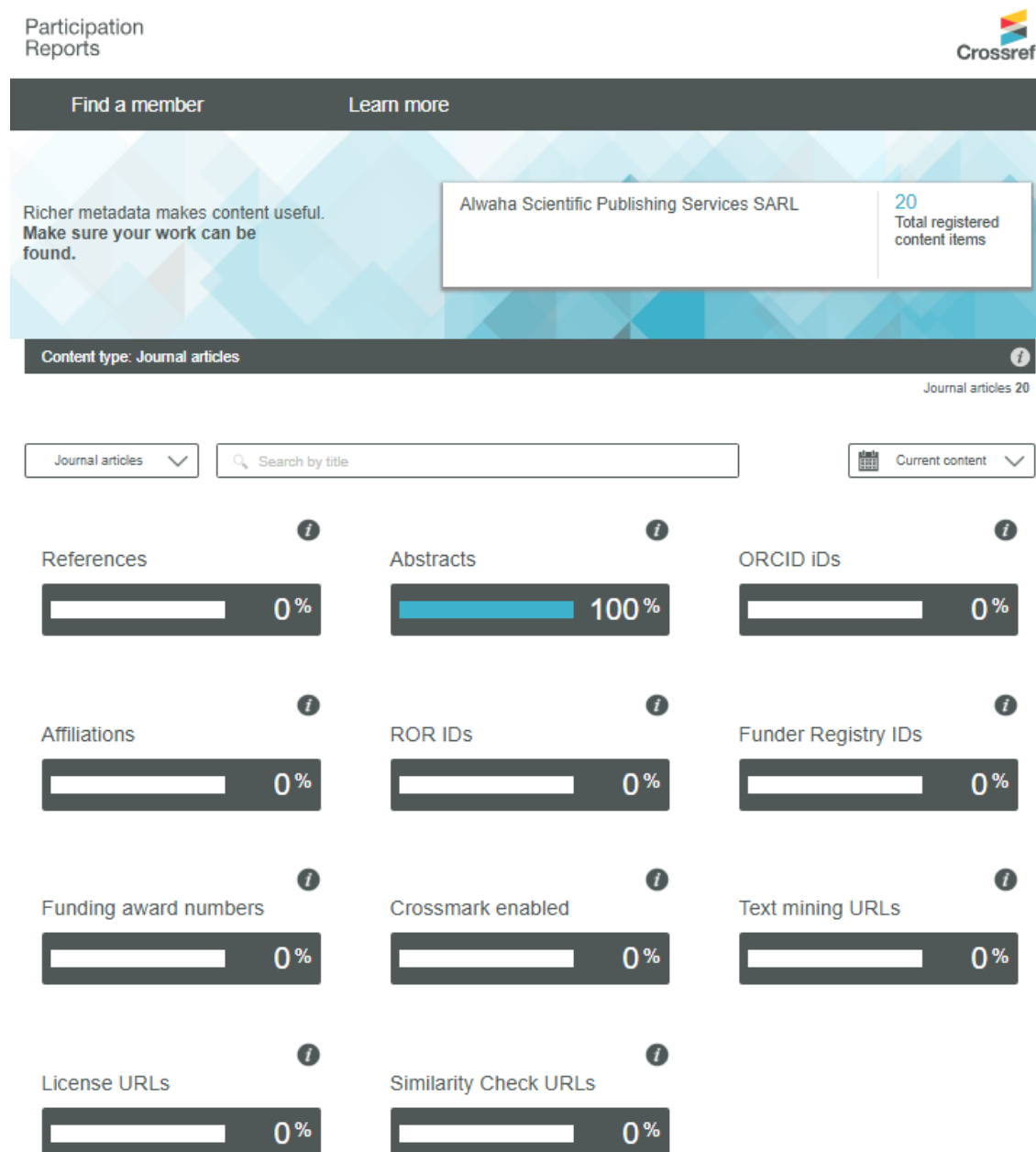


Figure 2.2: Comparative Crossref Participation Reports for ASPS

Upon entering the ASPS journal data into the tool, the following observations were made:

- **Abstracts (100%):** All indexed and published articles successfully include abstracts in their Crossref metadata deposits, indicating a strong baseline for discoverability.

- **Affiliations (0%):** None of the articles have included author affiliation metadata, which severely limits institutional recognition and accurate author profiling.
- **References (0%):** Reference lists are currently missing or improperly structured in the metadata, reducing citation linkage and cross-publication connectivity.
- **ORCID IDs, ROR IDs, Funder Registry IDs, Funding Award Numbers, License URLs, Similarity Check URLs, Crossmark Enabled, and Text Mining URLs (All at 0%):** These advanced metadata fields are absent, indicating significant gaps in compliance with international scholarly metadata standards.

2.6.2.2 Benchmarking Against eLife

To contextualize these findings, I reviewed benchmark data from a leading open-access publisher, *eLife Sciences Publications Ltd.* Unlike ASPS, eLife demonstrated over 98% metadata inclusion across nearly all fields (as visualized in Figure 2.3), which not only facilitates automated indexing and content discovery but also enhances the visibility and credibility of its scholarly output.

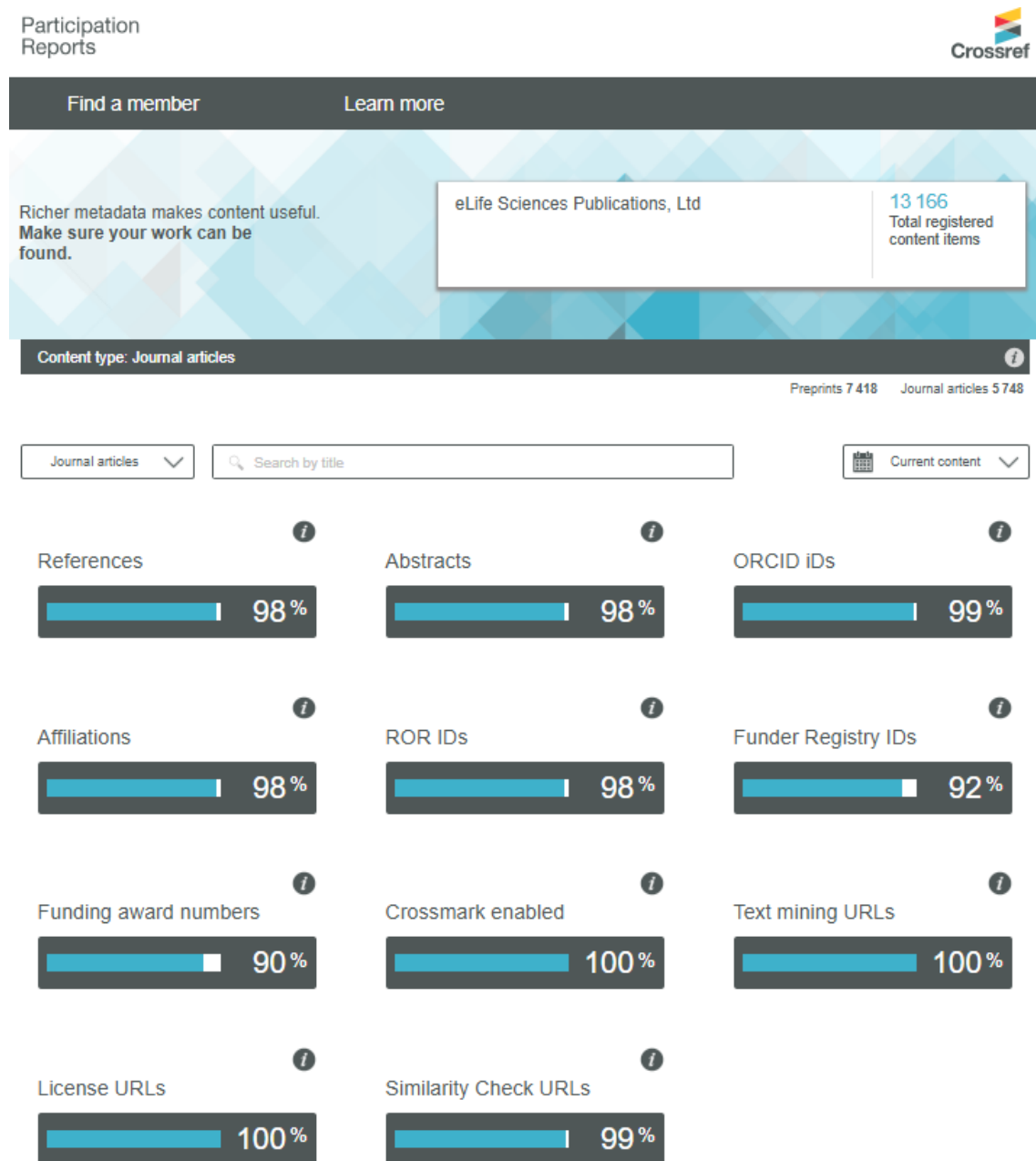


Figure 2.3: Comparative Crossref Participation Reports for eLife Sciences Publications Ltd

This comparative analysis underscores the need for strategic improvements at ASPS. Enhancing metadata completeness—especially for affiliations, references, and ORCID IDs—will be essential for advancing ASPS’s integration into global scholarly networks. The Participation Reports Tool thus proved invaluable in identifying actionable areas for optimization and served as a cornerstone for institutional reform recommendations outlined in my end-of-scholarship report.

This hands-on experience reinforced the importance of metadata as both a technical requirement and a political instrument and revealed the complexities of open-access publishing in resource-constrained environments. For instance, while ASPS’s zero-fee model democratizes

access, it also pressures limited staff to manually resolve metadata errors that wealthier publishers automate.

2.6.3 Crossref Webinar: Technical Training and Strategic Upskilling

As part of my professional training at Alwaha Scientific Publishing Services (ASPS), I was invited by the Production Manager to represent the publisher in the Crossref webinar titled *“Improving Metadata Quality Using the OJS Plugin”*. This webinar is mainly provided by Mr. Obanda Johanssen, a Community Engagement Manager at Crossref, assisted by Mr. Evans Eloni. Upon successful completion, I received an official certificate from Crossref acknowledging my participation and competency in applying the webinar’s technical modules which is included in the appendix. The webinar was tailored for researchers and publishers using Open Journal Systems (OJS) and focused on enhancing metadata quality through better plugin configuration and workflow optimization.

Key topics covered included:

- Installation and proper configuration of necessary OJS plugins.
- Efficient deposition of metadata directly to Crossref via the OJS interface.
- Troubleshooting common DOI registration issues, including incomplete or incorrect metadata fields.
- Leveraging Crossref Participation Reports to continuously evaluate and improve metadata completeness.

The session emphasized how complete and well-structured metadata directly improves article visibility, discoverability, and citation potential. It also highlighted the strategic value of technical infrastructure in academic publishing—particularly for small, resource-constrained publishers like ASPS.

Utilizing what I learned, I applied the acquired knowledge to ASPS’s active publishing workflow and metadata practices, including configuring plugins for automated DOI deposition and promoting structured metadata entry among editorial staff. These upgrades led to faster DOI registration and improved indexing, strengthening ASPS’s integration into global scholarly networks.

Beyond technical skills, the webinar highlighted how metadata infrastructures shape scholarly visibility. At institutions like ASPS, where human and financial resources are limited, this aspect of the publishing process often represents a form of invisible labor. The session provided a critical perspective on how small publishers in the Global South must balance infrastructural gaps with global expectations for metadata excellence.

2.7 Automated Indexing Workflow

2.7.1 Manual Indexing

Manual (or **human**) indexing refers to the process of metadata extraction and structuring for scholarly content by hand. Typically performed by indexer, editorial or production staff, it involves collecting, formatting and entering of critical bibliographic elements such as article titles, author names, affiliations, abstracts, keywords, and references. This data was then formatted into structured schemas (e.g., JATS XML or Crossref metadata schema) and manually uploaded to journal databases or indexing platforms (e.g., Crossref, DOAJ, Scopus, Google Scholar).

The input was typically the final raw PDF or Word version of a published article, while the output was a validated metadata file suitable for automated harvesting by academic databases. This process, although precise, was **time-consuming, error-prone**, and heavily dependent on the skill level of individual indexers [59].

2.7.2 Limitations of Manual Indexing at ASPS

Despite the foundational role that manual indexing played during the early operational stages of ASPS, this method revealed several systemic limitations as the publisher's output scaled. Among the most pressing challenges were:

- Minor formatting mistakes (e.g., incorrect ISSNs, missing affiliations, or broken references) frequently resulted in validation failures during DOI registration with Crossref.
- Redundant DOI registration errors creating confusion in citation tracking and archival systems.
- Editorial staff spending extensive time on repetitive tasks instead of quality control (Editors must manually format and validate each submission, slowing publication workflows, it may take 30–45 minutes per article).
- Over-reliance on individual editors for critical metadata tasks creates bottlenecks, especially in small teams with limited technical support.
- As journal output grows, manual processes become unsustainable.

These cumulative issues underscored the urgency of transitioning toward a more automated, resilient, and scalable indexing infrastructure.

2.7.3 Transition to Automation via OJS Plugins

Recognizing these limitations, ultimately highlighted the need for ASPS to adopt an automated indexing workflow and strategies capable of maintaining precision while reducing manual workload using Open Journal Systems (OJS), an open-source journal management software. The automation initiative involved configuring specific Crossref-compatible plugins that handle metadata formatting, validation, and deposition in alignment with global standards.

This transition will significantly enhance the efficiency of ASPS's editing processes, facilitating:

- **Template-driven Metadata Collection:** Authors will be prompted to submit structured metadata at the point of submission, reducing the need for post-submission formatting or manual re-entry.
- **Automatic metadata extraction** Core bibliographic fields will be parsed directly from submission forms and reused across subsequent workflow stages.
- **Batch DOI registration** DOIs will be minted and registered in bulk during the final publication phase, ensuring consistency and eliminating redundancy.
- **Built-in error checking:** Built-in schema validation ensures metadata complies with Crossref's formatting requirements, reducing rejection rates.
- **Real-time integration** with discovery platforms (e.g., Google Scholar, DOAJ).

This transition substantially will improve ASPS's metadata quality, accelerate publication timelines, and position the organization for sustainable growth across its expanding journal portfolio.

2.7.4 Core OJS Components Configured for Automation

During my training period at ASPS, I explored the potential of OJS's built-in plugins to streamline and eventually automate the journal indexing pipeline in alignment with Crossref's metadata requirements. Based on insights gained through the Crossref technical webinar and hands-on experimentation within ASPS's system, I identified three critical OJS modules that could be configured to enable automated metadata generation and DOI registration:

- **Crossref XML Export Plugin:** Automatically generates a fully-validated Crossref XML file (including abstracts, author identifiers, and linked references...) and submits it directly through the Crossref REST API whenever an article moves from Production to Published.
- **DOI Public Identifier Plugin:** Assigns persistent identifiers using ASPS's prefix (10.38208), applying a structured DOI pattern based on journal acronym, year, issue, and article ID. The assigned DOIs are programmatically embedded in each article record, ensuring consistency across metadata deposits.

- **Crossref Reference Linking Plugin:** Parses each citation list, queries the Crossref reference-matching service to retrieve DOI links for cited works. These links are embedded into both the article landing page and exported XML metadata file—satisfying Crossref’s mandatory “linked-references” criterion and enabling citation tracking through Crossref’s Cited-by service.

Complementing these plugins, OJS’s native **OAI-PMH data-provider endpoint** exposes these enriched metadata records to global harvesters such as Google Scholar. This automatic metadata exposure has proven effective in increasing the discoverability of ASPS content online, even in the absence of full automation.

While ASPS has not yet formally adopted these plugins into its production workflow, the results of my experimentation were positively received by the production manager. Plans are now under consideration to deploy these tools as part of ASPS’s future strategy to improve efficiency and ensure compliance with evolving metadata standards.

2.7.5 OJS Plugins for Indexing Automation

During my training at ASPS, I explored the integration of automated metadata indexing tools provided by Open Journal Systems (OJS) to align with Crossref’s technical recommendations and emerging best practices in scholarly publishing. Although these plugins have not yet been fully deployed in ASPS’s production workflow, the experimentation phase and internal demonstrations highlighted their potential for enhancing both accuracy and efficiency.

Among the most promising tools were the **Crossref XML Export Plugin**, the **DOI Public Identifier Plugin**, and the—each specifically designed to reduce manual overhead while ensuring schema compliance for Crossref metadata deposits. When appropriately configured, these modules offer the following functional advantages:

- **Automatic Metadata Formatting:** Structured metadata—including titles, abstracts, author affiliations, and references—can be programmatically transformed into Crossref-compliant XML.
- **Direct Metadata Submission:** The Crossref XML Export Plugin enables editors to deposit metadata via API directly from the OJS interface.
- **DOI Generation:** The DOI Plugin automates the minting of persistent identifiers based on ASPS’s prefix (10.38208), ensuring referential integrity across articles, galleys, and supplementary materials.

- **Metadata Quality Auditing:** Integration with Crossref’s Participation Reports enables periodic evaluation of submission completeness and quality (e.g., presence of ORCID iDs, abstracts, and linked references).
- Reduce human error by eliminating the need for spreadsheet uploads or manual XML editing.

Crossref itself “strongly recommends automatic deposits” for members and highlights that a single checkbox in the DOI-Plugin settings can enable continuous, unattended submission www.crossref.org. Many library-science guides likewise note that exporting metadata **programmatically** is now a best practice for open-access journals [Research - Home](#).

Table 2.1 Outlines a proposed end-to-end workflow based on the automation capabilities of these OJS plugins:

Editorial Stage	OJS Module Triggered	Automated Outcome	Manual Effort Remaining
Submission	Author Submission Form	Mandatory fields capture Title, Abstract, ORCID, ROR, Funding	Author fills form
Production Publication →	DOI Public Identifier Plugin	DOI minted with prefix 10.38208	None
Publication	Crossref XML Export Plugin (auto-deposit = ON)	Crossref-compliant XML posted to REST API	None
Post-deposit	Crossref Reference Linking Plugin	Citation list enriched with matched DOIs; Cited-by enabled	Occasional manual DOI lookup for unmatched refs
Audit	Participation Reports Link	Completeness metrics displayed (abstracts, affiliations, references, etc.)	Editor reviews dashboard; schedules fixes if < 95 %

2.8 Implementing Automated Indexing in OJS

This transition was made possible after Following the Production Manager’s approval, I was granted administrative access to ASPS’s OJS backend. With these credentials, I conducted controlled test runs to perform a series of plugin configurations and metadata deposit trials.

In the next section, I present annotated screenshots from ASPS’s OJS environment to illustrate how automated indexing workflows were set up and tested. Each screenshot highlights a specific step-

by-step setup of automated indexing via the Crossref plugin within OJS, from plugin activation to metadata export and DOI assignment.

These trials not only validated the platform's capabilities but also provided a reproducible workflow that now forms the basis of an internal guide which can be used by ASPS's editorial team to train their future staff.

2.8.1 Plugin Configuration

2.8.1.1 DOI Public Identifier Plugin

The first step in setting up automated indexing is to locate and activate the relevant plugins within the OJS backend. As shown in **Figure 2.4**.

- **Accessing Plugin Settings:** From the OJS dashboard, I selected *Settings* > *Website* > *Plugins*. This section houses all installed plugins and their configuration options including those responsible for DOI generation and Crossref metadata export.

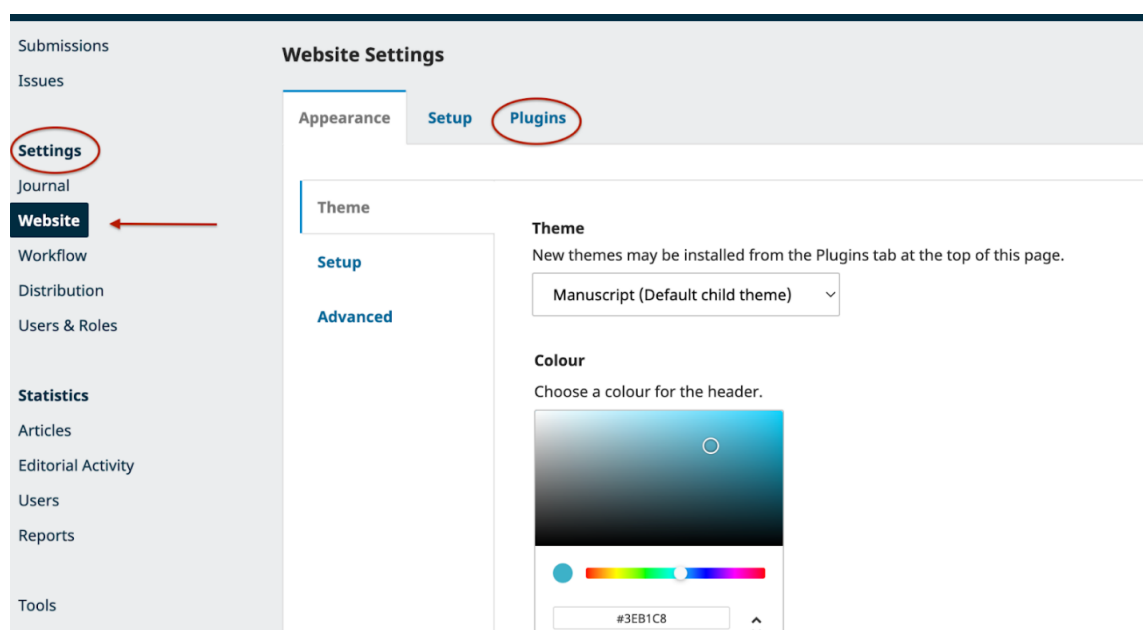


Figure 2.4: Accessing the Plugins Tab in OJS via Website Settings for DOI Configuration

- **Enabling the DOI Registration Plugin:** Within the *Public Identifier Plugins* section, locate the entry labeled *DOI*, then activate it by ticking the checkbox on the far right of its description (figure 2.5). If not pre-installed, this plugin can be added via the *Plugin Gallery* tab. Then Activate it.

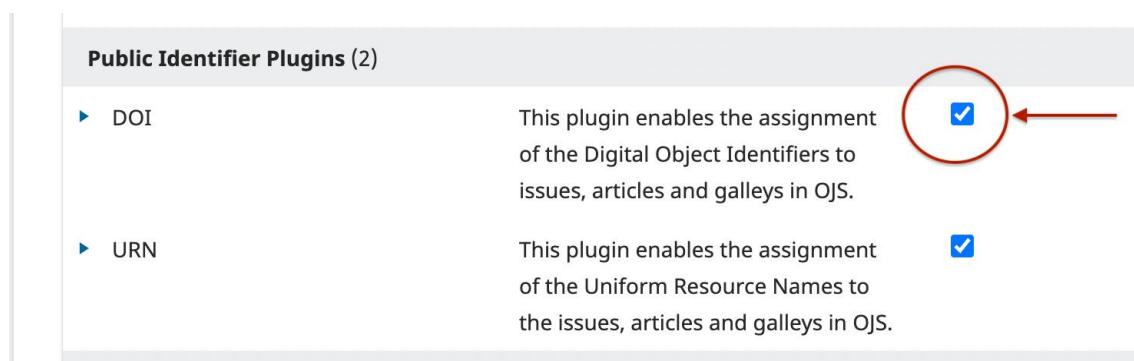


Figure 2.5: Enabling the DOI Public Identifier Plugin in OJS

- **Configuring the DOI Prefix:** After that, open its configuration panel, and enter ASPS's DOI prefix (**10.38208**) to ensure all deposited content is correctly attributed to the publisher. Finally, specify a clear DOI pattern (figure 2.6) -such as journal-acronym.year.issue.articleId—and save the settings; OJS will now generate Crossref-compliant DOIs automatically for each article once it reaches production or publication.

Please configure the DOI plugin to be able to manage and use DOIs in OJS:

Journal Content

Please select the publishing objects that will have Digital Object Identifiers (DOI) assigned:

☐ Issues
☒ Articles
☐ Galleys

DOI Prefix

The DOI Prefix is assigned by registration agencies (e.g. [Crossref](#)) and is in the format 10.xxxx (e.g. 10.1234):

DOI Prefix *

Figure 2.6: Configuring the DOI Plugin in OJS

This configuration step forms the foundation for subsequent automated metadata deposits and is crucial because it ensures that the OJS system can generate and assign DOIs using the prefix assigned to ASPS (10.38208). It also prepares the platform for later steps, including metadata deposit and batch DOI registration. The plugin must be correctly configured with the DOI prefix and activated before any indexing can proceed.

2.8.1.2 Crossref XML Export Plugin

- **Installing the plugin:** To export and register article metadata with Crossref, open *Tools* in the left-hand menu, switch to the *Import/Export tab*, and select the *Crossref XML Export Plugin* (Figure 2.7).

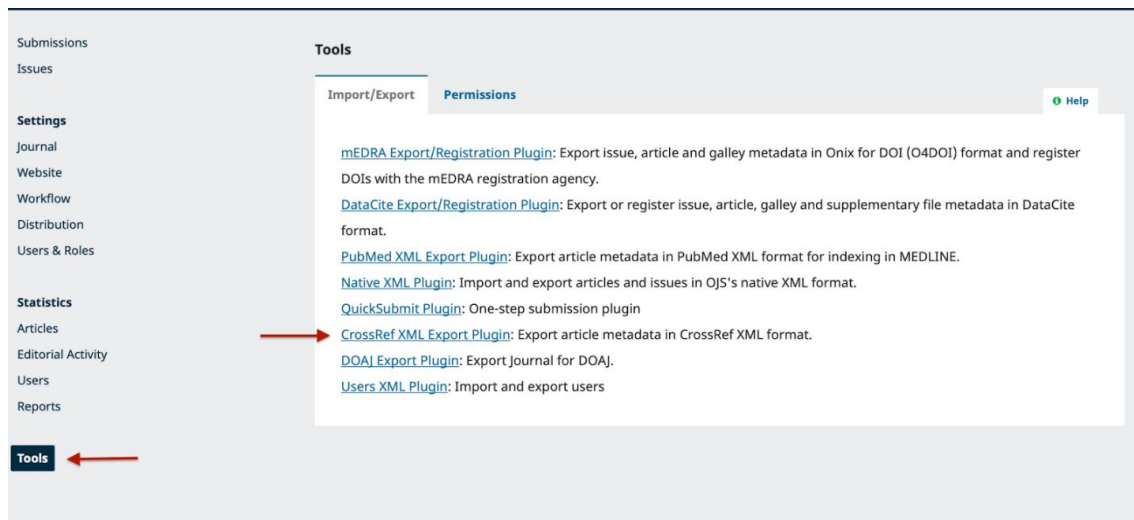


Figure 2.7: Selecting the XML Export Plugin in OJS

- **Accessing the Plugin Activation Interface:** From the OJS dashboard, I selected *Settings > Website > Plugins* (as shown in Figure 2.8). Scrolled to the entry labeled the *Import/Export plugins* section to locate the *Crossref XML Export Plugin*, then activate it by ticking the checkbox on the far right of its description.



Figure 2.8: Enabling the Crossref XML Export Plugin in OJS

- **Automatic Deposit Configuration:** The final phase of automating the indexing workflow involved Enabling DOI *automatic deposit feature* within the OJS plugin. This step ensured real-time metadata synchronization with Crossref, eliminating delays inherent in manual submissions (Figure 2.9).

☒ OJS will deposit assigned DOIs automatically to CrossRef. Please note that this may take a short amount of time after publication to process (e.g. depending on your cronjob configuration). You can check for all unregistered DOIs.

Figure 2.9: Enabling Automatic Deposit

- **Entering Depositor Credentials:** Next, fill in the required depositor fields—enter the publishing organization’s name, the responsible contact email, and your Crossref credentials (username and password) obtained through prior communication with Crossref support, noting that shared *role* credentials go in plainly while personal *user* credentials must be formatted as email@address.com/role (figure 2.10). After saving, OJS will send each newly published article’s metadata straight to Crossref; any mismatch in the username-password pair will trigger a 401 “unauthorized” error, signalling that the credentials need correction, in case of the operation ended with success, the verified successful deposits via Crossref’s Metadata Search, confirming DOIs resolved correctly (e.g., 10.38208/jret.2023.12).

[DOI Plugin Settings](#)

The following items are required for a successful CrossRef deposit.

Test Crossref

Depositor name *

youremail@youremail.org

Depositor email *

If you would like to use this plugin to register Digital Object Identifiers (DOIs) directly with CrossRef you will need a username and password (available from [CrossRef](#)) in order to do so. If you do not have your own username and password you can still export into the CrossRef XML format, but you cannot register your DOIs with CrossRef from within OJS.

support@crossref.org/creftest

Username

.....

Password

Please note that the password will be saved as plain text, i.e. not encrypted.

☒ OJS will deposit assigned DOIs automatically to CrossRef. Please note that this may take a short amount of time after publication to process (e.g. depending on your cronjob configuration). You can check for all unregistered DOIs.

☐ Use the CrossRef test API (testing environment) for the DOI deposit. Please do not forget to remove this option for the production.

[Save](#) [Cancel](#)

* Denotes required field

Figure 2.10: Entering depositor credentials in the OJS DOI plugin

So after these configurations OJS will generate a Crossref-compliant XML file and transmit it directly after an article reaches the “Published” stage. With that we can say that we configured the plugin to directly communicate with Crossref’s API, triggering immediate and automatic metadata submission post-acceptance.

2.8.1.3 Crossref Reference Linking Plugin

To ensure that references are properly linked and deposited with Crossref, activate and configure the *Crossref Reference Linking plugin* within the OJS backend. This plugin uses the Crossref API to match references against DOI records and embeds them in the metadata deposit. Activating it is a critical requirement for Crossref membership, as it supports citation tracking and enhances discoverability.

- **Enabling Reference Metadata in Workflow Settings:** As shown in **Figure 2.11**, I first navigated to *Settings > Workflow*, and selected the *Metadata* tab under the *Submission* section.

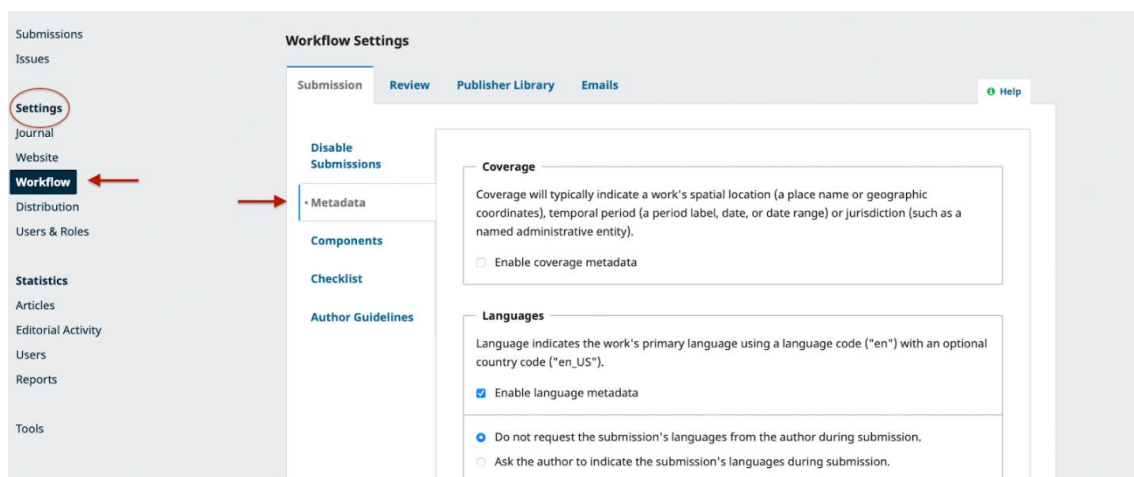


Figure 2.11: Enabling reference metadata under Workflow Settings in OJS

- Here, scroll down till finding the ‘*References*’ section. Then, enable the reference metadata option by ticking “*Enable references metadata*” and chose “*Ask the author to provide references during submission*”. This step ensures that authors are prompted to enter reference lists at the point of manuscript submission (Figure 2.12).
- **Activating the Reference Plugin in Website Settings:** Next, I moved to *Settings > Website*, and opened the *Plugins* tab (see Figure 2.11). I searched for “*Crossref Reference Linking*” in the list of available plugins and checked the box to enable it. Once active, this plugin automatically includes references in the XML metadata export submitted to Crossref (Figure 2.13).

References

Collect a submission's references in a separate field. This may be required to comply with citation-tracking services such as Crossref.

☒ Enable references metadata

☐ Do not request references from the author during submission.

☒ Ask the author to provide references during submission.

☐ Require the author to provide references before accepting their submission.

Publisher ID

The publisher ID may be used to record the ID from an external database. For example, items exported for deposit to PubMed may include the publisher ID. This should not be used for DOIs.

☐ Enable for Publications

☐ Enable for Galleys

☐ Enable for Issues

☐ Enable for Issue Galleys

[Save](#)

Figure 2.12: Crossref Reference Linking plugin activation

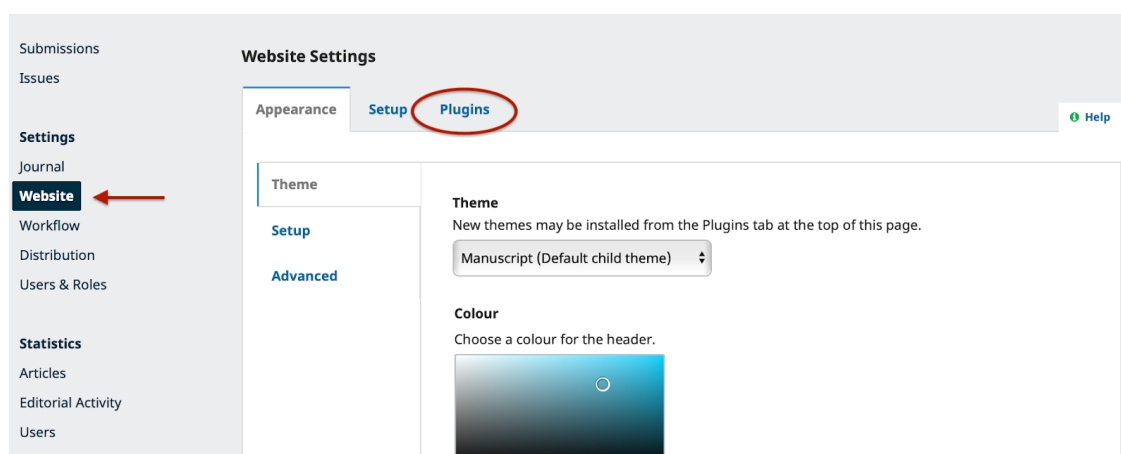


Figure 2.13: Activating the Crossref Reference Linking plugin under Website Plugins

After saving the changes, this plugin will deposit the references that you enter into the XML deposit.

2.8.1.4 Additional OJS Plugins Supporting Crossref Integration

Beyond the core Crossref XML Export plugin, Open Journal Systems (OJS) offers a suite of optional plugins designed to enhance metadata quality, citation visibility, and compliance with

Crossref standards. During my testing phase, I reviewed several of these supplementary plugins that can significantly improve scholarly infrastructure if configured properly.

These include:

- Plugins for citation tracking and "Cited-by" support.
- Tools for integrating funding acknowledgments.
- Interfaces for plagiarism detection via Crossref's Similarity Check service.
- While these plugins were not the primary focus of my testing, they represent important opportunities for further metadata enhancement and operational streamlining at ASPS.
- A detailed overview of each plugin's purpose, compatibility version, and configuration process in my ASPS scholarship report.

2.8.2 Retro-Assigning DOIs to Previously Published Articles

The *Submissions* dashboard is filtered to the **Archives** tab, which houses every article that has completed the production cycle. Accessing this queue is the essential first step when retro-assigning identifiers to legacy content. Older, already-published articles live in the *Archives* queue; we first need to locate the record there (as shown in Figure 2.14). Activating the **Search** bar narrows the archive by article ID, title, or author, enabling editors to retrieve a specific record without manually scrolling through historical volumes—an efficiency gain when large back-files exist (Figure 2.15).

A single result is returned; selecting **View** opens the full workflow dashboard for the chosen article. This interface grants granular control over each publication stage and exposes the metadata panels required for DOI assignment (Figure 2.16).

Within the article workspace, the **Publication** tab is highlighted. DOI management resides here—separate from copy-editing and production—reflecting OJS's separation of bibliographic identifiers from file-format tasks (Figure 2.17).

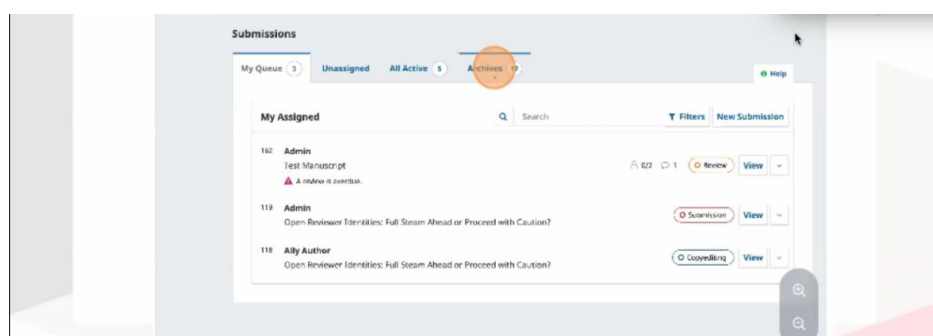


Figure 2.14: Locating Back-Issue Articles in the Archives Queue

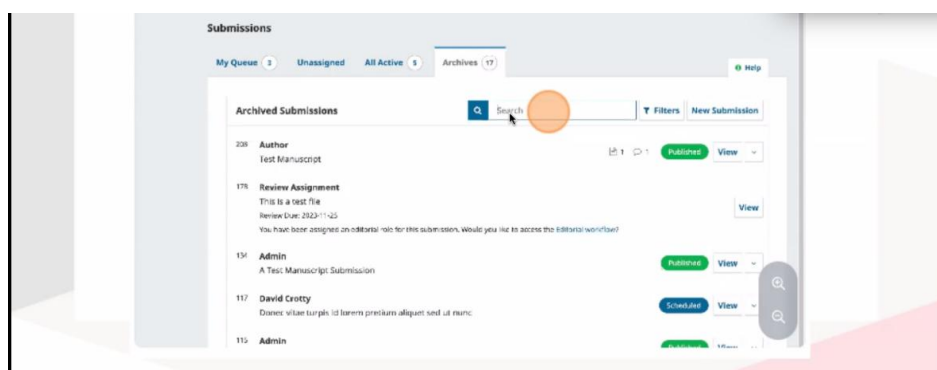


Figure 2.15: Precision Search Within Archived Submissions

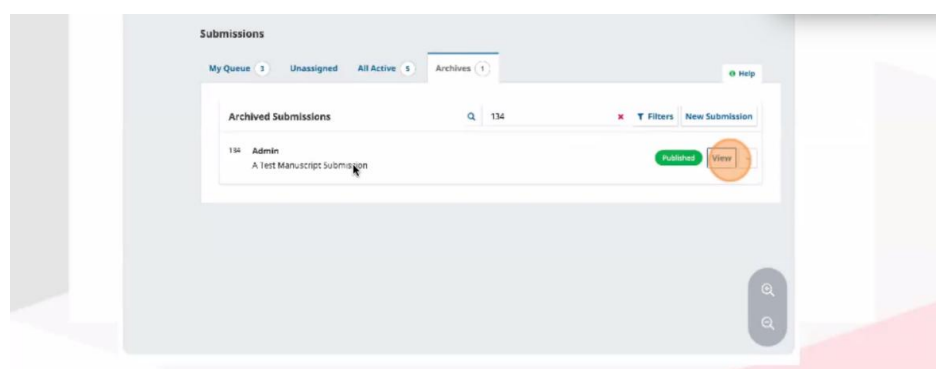


Figure 2.16: Opening the Article Workflow for Metadata Editing

After opening the **Publication** tab, OJS presents the full suite of editable metadata fields—Title & Abstract, Contributors, Identifiers, and additional sections—allowing the editor to make definitive bibliographic updates. Figure 2.16 exemplifies this process by showing how the Title and Abstract can be revised directly within their respective input fields (Figure 2.18).

Selecting Unpublish triggers a confirmation dialog, ensuring editors acknowledge the implications of withdrawing public content. This step is mandatory; only an *Unscheduled* article can accept new or revised identifiers (Figure 2.19).

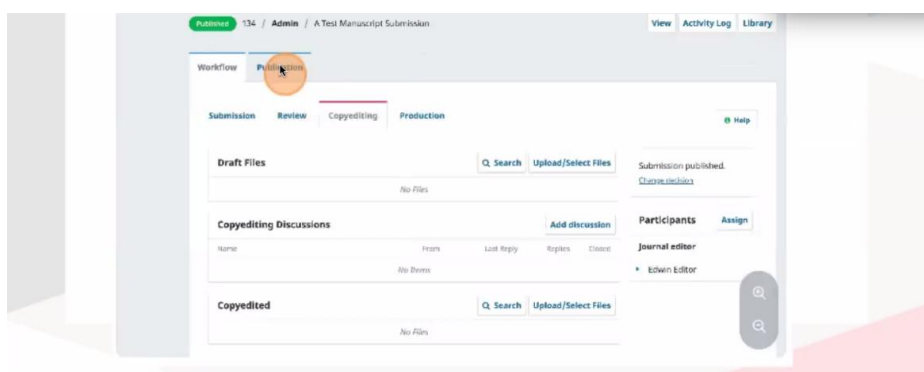


Figure 2.17: Navigating to the Publication Tab

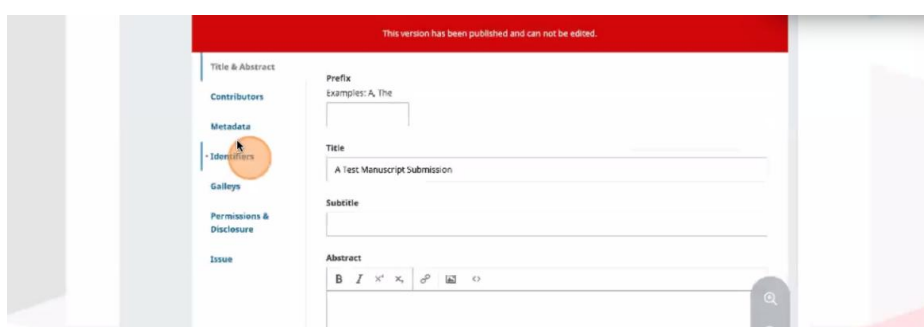


Figure 2.18: Editing Core Metadata Fields in the Publication Tab

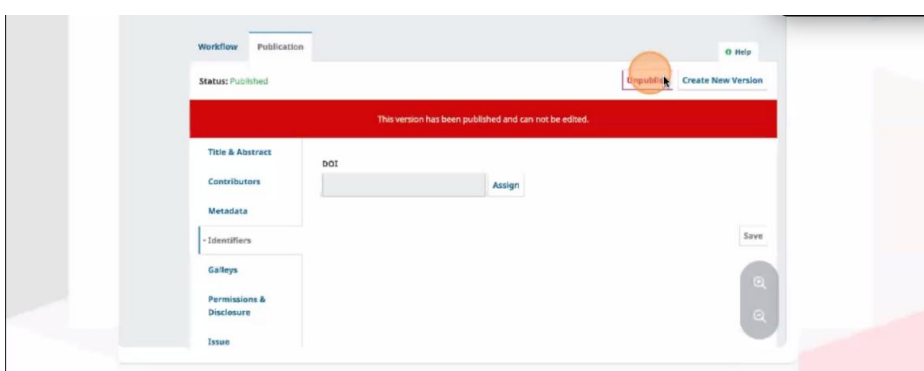


Figure 2.19: Temporarily Unpublishing to Unlock DOI Controls

Because OJS locks all bibliographic fields once an article is publicly available, the record must be **temporarily unpublished** to enable DOI editing. Triggering *Unpublish* prompts a confirmation dialog—a built-in safeguard that prevents accidental withdrawal of live content. By affirming this dialog, the editor intentionally takes the article offline so the DOI field (and any other locked metadata) can be modified (Figure 2.20).

Once the status switches to *Unscheduled*, the **Identifiers** section becomes editable. Clicking **Assign** invokes the DOI pattern configured earlier (see Fig. 2.4), automatically producing a Crossref-compliant identifier (Figure 2.21).

The freshly minted DOI now populates its field. Pressing **Save** commits the change to the database, ensuring the identifier is permanently linked to this publication record (Figure 2.22)

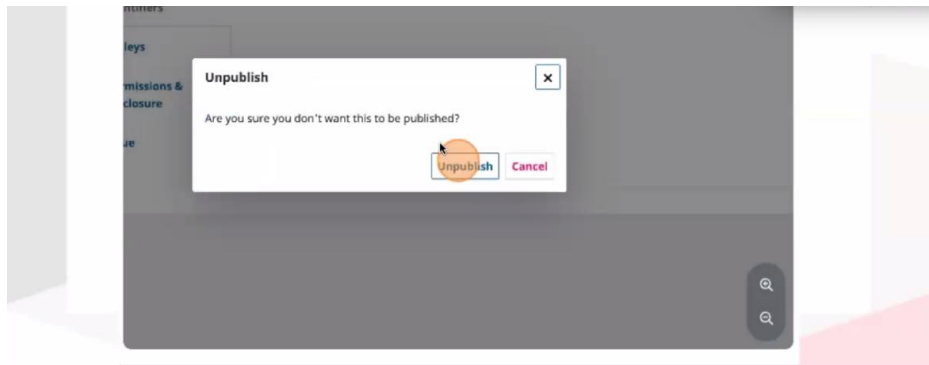


Figure 2.20: Temporarily Unpublishing an article to Unlock DOI Editing

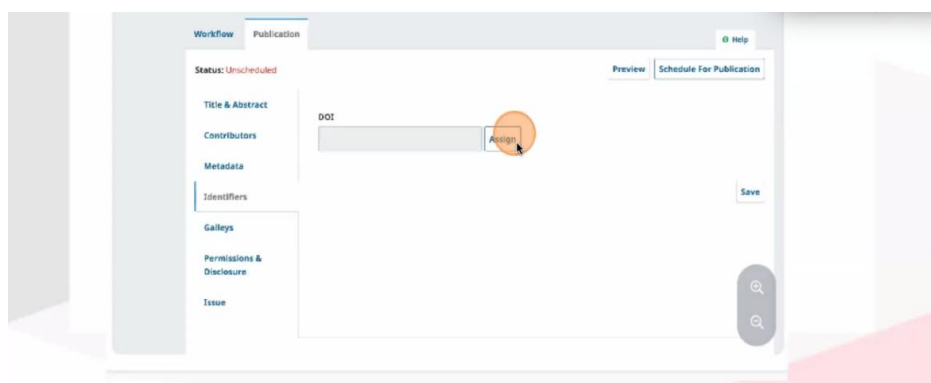


Figure 2.21: Generating a DOI in the Identifiers Panel

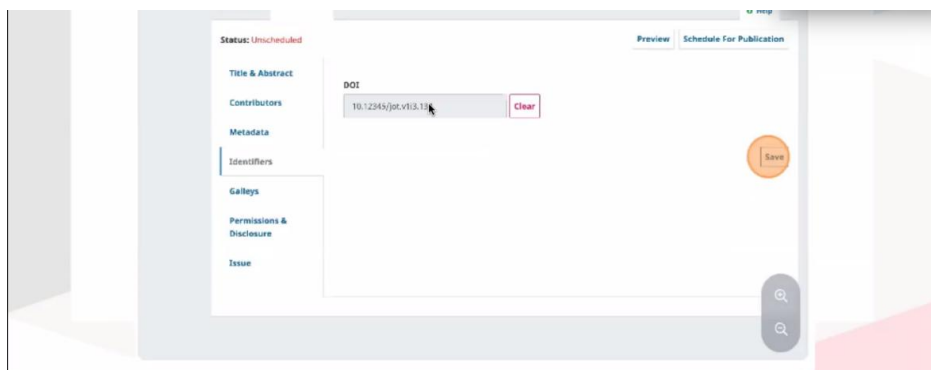


Figure 2.22: Saving the Newly Generated DOI

With the DOI in place, clicking **Schedule for Publication** returns the article to public view. This action queues an immediate metadata deposit via the Crossref plugin, provided automatic deposits are enabled (Figure 2.23).

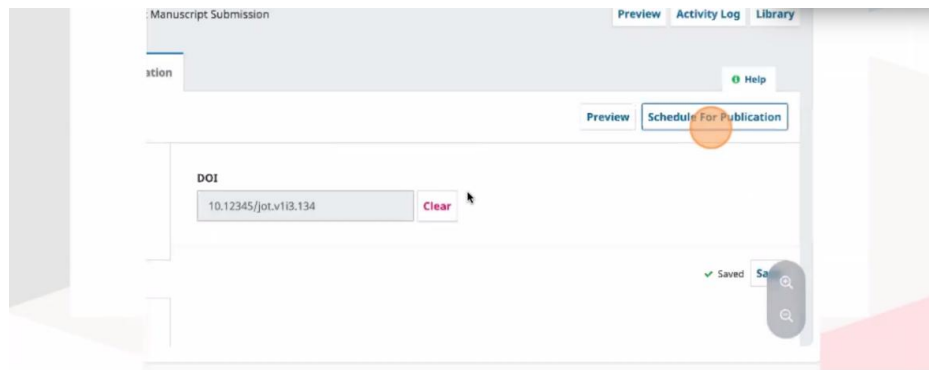


Figure 2.23: Republishing the Article with Updated Metadata

2.10 Recommendations and Improvements

Drawing from practical experience, I formulated recommendations to enhance metadata accuracy and indexing efficiency at ASPS. These recommendations aim at improving system configurations, refining metadata standards, and ensuring continuous metadata quality assessment.

2.9 2.11 Conclusion

Through this case study, I have demonstrated how ASPS successfully transitioned from a largely manual metadata management system to a semi-automated, standards-compliant infrastructure powered by Open Journal Systems (OJS). This transition, made possible by direct administrative access and structured plugin configurations, has improved the publisher's indexing accuracy, workflow efficiency, and integration with global discovery services such as Crossref, DOAJ, and Google Scholar.

Equally important was the human dimension: the Crossref webinar *“Improving Metadata Quality Using the OJS Plugin”*, and subsequent in-house training built local expertise, enabling ASPS staff to maintain the system and troubleshoot deposits independently. Recognising, my participation in the Crossref webinar on metadata optimization provided critical technical insights that were translated into actionable reforms at ASPS. Based on the lessons learned, I developed a set of strategic recommendations for further metadata enhancement, plugin utilization, and staff training. These recommendations have been formally shared with ASPS and are included in scholarship report.

This case study serves as a blueprint for regional publishers navigating the complexities of digital scholarly infrastructure. Future research should explore the long-term impact of automation on editorial labor dynamics, particularly in resource-constrained environments where invisible technical labor often exacerbates inequities. By embracing automation as a tool for equity rather than mere efficiency, publishers like ASPS can amplify the global visibility of underrepresented research while sustaining their mission-driven priorities.

3 Design and implementation of XML generator tool for metadata indexers: Case Study of ASPS Journals

3.1 Introduction

Chapter 2 demonstrated that Open Journal Systems (OJS) plug-ins can automate much of ASPS’s routine indexing, but that they still presuppose a level of metadata expertise—and editorial labour—that a diamond-OA publisher in the Global South rarely possesses. Chapter 3 therefore moves from workflow optimisation to software engineering. It opens by surveying the XML-based standards (Dublin Core, JATS, Crossref) that govern scholarly interoperability and by mapping the overlapping, yet distinct, requirements of the three indexing ecosystems most relevant to ASPS—Crossref, DOAJ, and EBSCO. Against this normative backdrop the chapter then documents, step-by-step, the conception, design, and realisation of a stand-alone Crossref XML Generator written in modern C++20.

The narrative is deliberately pragmatic. It begins with a standards analysis (§ 3.2–3.3) that distils a unified field matrix from the official schema specifications, translates those fields into two minimalist CSV templates, and then traces how the generator transforms spreadsheet rows into a schema-validated XML batch. Subsequent sections (§ 3.4) present the software architecture, workflow swim-lane, data-normalisation rules, DOI-minting logic, and UML artefacts that collectively guarantee correctness while remaining accessible to non-specialist staff. Finally, the implementation notes (§ 3.5) catalogue the hardware, tool-chain, and code modularisation decisions that make the project portable and maintainable.

3.2 XML and Metadata Standards

Metadata interoperability and discoverability depend significantly on structured XML formats. Crossref, Directory of Open Access Journals (DOAJ), and EBSCO indexing databases require standardized XML schemas for accurate content indexing [60]. Understanding these schemas—including Dublin Core, JATS XML, and Crossref-specific metadata requirements—is crucial for developing effective indexing tools [61, 62].

Metadata interoperability—the ability of systems to exchange and use metadata seamlessly—relies on structured, machine-readable formats like XML. For academic publishers such as ASPS,

adopting standardized XML schemas is not optional but a prerequisite for integration with global indexing services like Crossref, DOAJ, and EBSCO [63, 64]. These services mandate strict adherence to schemas such as **Dublin Core**, **JATS XML**, and **Crossref-specific metadata** to ensure accurate content discovery, citation tracking, and long-term preservation [14]. This section traces the evolution of metadata standards, compares key schemas, and explains their strategic relevance to ASPS's mission.

3.2.1 Evolution of Metadata Standards

The evolution of metadata in scholarly communication has followed the broader trajectory of digitization and interoperability. In the early days of the web, metadata was minimal and typically human-readable. Over time, the demand for **semantic web technologies**, machine-driven indexing, and persistent linking mechanisms (e.g., DOIs) caused by the growing complexity of digital scholarly communication led to the adoption of highly structured, XML-based standards [65, 66].

- **Dublin Core:** emerged in the mid-1990s as a general-purpose metadata standard for early library projects and digital resources [67, 68]. Its simplicity, adaptability and ability to be applied across disciplines and parsed with limited computing power made it a baseline framework for libraries, institutional repositories, and early digital publishing systems.
- **Journal Article Tag Suite (JATS) XML:** Born from the National Library of Medicine's (NLM) need to archive biomedical journals, introduced a detailed, semantically rich XML structure for the full-text content of journal articles, including granular tagging of bibliographic elements, figures, equations, and affiliations. JATS (Journal Article Tag Suite) became the gold standard for structuring full-text academic articles many indexing services, including PubMed Central and DOAJ [69, 70].
- **Crossref XML Schema:** A specialized standard schema focusing primarily on metadata deposition rather than full-text description relying on persistent identifiers (DOIs) and citation networks. It includes fields for DOIs, authors, publication dates, reference lists, funding data, and ORCID/ROR identifiers, aligned with Crossref's own API validation. It is more compact than JATS but prescribes strict conformance for metadata fields such as contributors, references, and funding information [71].

This evolution mirrors the shift from flexible, descriptive metadata (what an item is) to structured, schema-bound data models metadata (how it is organized) and *administrative* metadata (how it should be managed) [27]. For ASPS, adopting modern schemas like JATS and Crossref is critical to competing in an era where machines, not humans, increasingly mediate research discovery.

3.2.2 Schema Analysis: Crossref, JATS, and Dublin Core

To guide the development of the custom indexing tool, a comparative analysis was conducted to evaluate the structure, complexity, and field coverage of three dominant metadata schemas: Dublin Core, Crossref XML, and JATS XML [72]. Table 3.1 summarizes this comparison:

Table 3.1: Comparative Overview of Metadata Schemas

Feature	Dublin Core	JATS XML	Crossref XML
Primary Purpose	Digital libraries, repositories	Full-text article/journal archiving	DOI registration, citation tracking
Granularity	Low (15 core elements)	High(150+elements)	Moderate (50+ elements)
Key Fields	Title, Creator, Subject	Full-text structure,Affiliations, References, Figures	Title, Contributors,Publication-date, DOI, References
Reference Handling	Free-text	<ref-list> with structured <mixed-citation> or <element-citation>	<citation_list> with each <citation> mapped to DOI/ISSN
Ease of Implementation	High	Low	Medium
Validation	Optional	Strict (DTD/Schematron)	Strict (Crossref REST API, helper tools & submission API)
Adoption Context	Libraries, repositories	Academic journals (PubMed, PMC, Scopus, DOAJ)	Crossref members, publishers, Web of Science, Google Scholar

3.3 Indexing Services: Requirements and Strategic Value

Indexing services act as gatekeepers to global scholarly visibility, each imposing unique technical and editorial requirements. For ASPS, understanding the operational models and metadata standards of major indexing services is essential to expanding its reach across academic, institutional, and open-access audiences. Each service—Crossref, DOAJ, and EBSCO—represents a different layer of the scholarly communications infrastructure. While their technical and procedural requirements often overlap, each one introduces distinct metadata profile that must be reconciled in the design of a unified indexing tool. This section outlines their respective roles and the strategic importance of aligning ASPS’s metadata practices accordingly.

3.3.1 some indexing services

3.3.1.1 Crossref: Persistent Identifiers and Citation Networks

Crossref is the cornerstone of scholarly infrastructure when it comes to persistent identifiers [65, 73]. It serves as a global registry for Digital Object Identifiers (DOIs), enabling permanent, machine-actionable links to academic content. Crossref metadata deposits are submitted in XML

format and must comply with its schema (currently version 5.3.1) [71], which mandates fields such as article title, contributor roles, publication date, DOI, and references. Fields like ORCID identifiers, funding data, and institutional affiliations are strongly recommended and increasingly essential as scholarly communication shifts toward open metadata standards. Validation of submissions is enforced through the Crossref submission endpoint, which returns schema compliance reports in real time. Importantly, full compliance not only ensures DOI assignment but also activates citation tracking through Crossref’s Cited-by service—making metadata completeness a direct driver of scholarly impact [74].

Its schema mandates:

- **Purpose.** Registers Digital Object Identifiers (DOIs).
- **Current schema.** *Metadata Deposit 5.4.0* (supported range 4.3.0 – 5.4.0) [64].
- **Mandatory:** Article title, author list, ISSN, DOI, publication date, license URL.
- **Recommended:** ORCIDs, funding data, ROR IDs, references with DOIs.
- **Validation:** The web-based *Metadata Quality Check* tool www.crossref.org

Crossref mandates machine-linked references, author ORCID IDs where available, and persistent licence URLs; deposits lacking these elements are accepted but assigned lower “Participation Report” scores, reducing downstream visibility.

3.3.1.2 DOAJ: Open-Access Compliance and Global Visibility

The Directory of Open Access Journals (DOAJ) represents a distinct but complementary indexing paradigm. Unlike Crossref, which centers on persistent identifiers and citation infrastructure, DOAJ curates a vetted directory of high-quality open-access journals [60]. Inclusion in DOAJ is both a badge of editorial legitimacy and a visibility accelerator for emerging publishers. However, DOAJ imposes strict requirements on transparency, licensing, peer-review processes, and article-level metadata. Metadata submission is typically carried out using CSV templates or JATS XML exports, with the latter preferred for automated harvesting. Fields such as licensing URLs [60], full-text access links, ORCID identifiers, and institutional affiliations are either mandatory or recommended under DOAJ’s latest schema. The directory also performs periodic validation through its internal ingestion systems and now encourages journals to provide machine-readable metadata compatible with OAI-PMH and JATS XML formats. DOAJ’s metadata schema [75] (CSV/XML v2023) requires:

- **Purpose.** Indexes peer-reviewed open-access journals and feeds metadata to library discovery layers.

- **Accepted formats.** JATS 1.1/1.3 XML or DOAJ Article CSV (rev. 2024-04) [NISO](#)
- **Mandatory:** Journal title, ISSN, APC transparency, copyright statement, license type.
- **Recommended:** JATS XML exports, ORCIDs, institutional affiliations.
- **Validation:** DOAJ’s manual review process checks for policy compliance (e.g., CC-BY licensing, editorial rigor). Pre-ingestion XML validator at <https://doaj.org/validate> (beta) and CSV linter in the application portal.

DOAJ “strongly recommends” JATS for scalable harvesting; the CSV path is maintained primarily for legacy workflows.

3.3.1.3 EBSCO: Bridging Academic and Institutional Audiences

EBSCO differs significantly from Crossref and DOAJ in both its function and technical implementation. As a commercial aggregator of academic databases, EBSCO serves academic libraries, research institutions, and corporate knowledge platforms [63]. It emphasizes metadata richness and structural clarity to facilitate high-performance search and content filtering. For journals seeking inclusion, EBSCO requires submission of metadata in formats like KBART (Knowledge Bases and Related Tools) or MARC, complemented by journal-level descriptive fields. The KBART format expects clean tabular metadata, including ISSNs, coverage dates, publisher information, access URLs, and licensing statements. EBSCO’s validation system checks for missing fields, format mismatches, and URL resolvability before ingestion into its central knowledge base [76]. Unlike Crossref and DOAJ, which directly enhance article discoverability in scholarly environments, EBSCO facilitates journal-level discoverability within curated academic collections and subscription packages [77]. Its KBART format (v2022) requires:

- **Purpose.** Supplies holdings metadata to EBSCO Discovery Service and thousands of link-resolvers.
- **Standard.** *KBART Recommended Practice, NISO RP-9-2023* (Version 3) [NISO](#)
- **Mandatory:** ISSN, coverage dates, publisher name.
- **Recommended:** Abstracts, keywords, author affiliations.
- **Validation:** EBSCO’s content team manually verifies metadata accuracy and relevance./ Tab-delimited files must pass the KBART Phase 3 field checker (<https://kbart.niso.org/checker>).

EBSCO focuses on title-level holdings but now permits article-level DOIs and open-access flags in extended columns.

3.3.2 Metadata Requirements

Table 3.2 summarizes metadata requirements across three indexing services:

Table 3.2: Metadata Requirements Across Indexing Services

Field	Crossref	DOAJ	EBSCO
DOI	Mandatory	Recommended	Optional
ISSN	Mandatory	Mandatory	Mandatory
ORCID	Recommended	Recommended	Optional
License URL	Mandatory	Mandatory	Optional
References (DOIs)	Mandatory	N/A	Optional
APC Transparency	N/A	Mandatory	N/A
Author Affiliations	Recommended	Recommended	Recommended
Validation Endpoint	Crossref metadata quality checker	DOAJ Online Form	EBSCO Content Team

3.3.3 Unified Field Matrix

Table 3.3: Unified Field Matrix

Field	Crossref 5.4	JATS 1.3 (DOAJ)	KBART v3 (EBSCO)	ASPS Implementation
DOI	M <doi>	M <article-id pub-id-type="doi">	R doi	Auto-minted via DOI-Plugin
Article Title	M	M <article-title>	M publication_title	Captured in submission form
Author Names	M	M <contrib-group>	O first_author	Parsed; validated for ORCID
ORCID	R ORCID attribute	R <contrib-id contrib-id-type="orcid">	N/A	API lookup & normalisation
Affiliations / ROR	R <institution>	R <aff>	O publisher_name	Dropdown + free text
Abstract	R <jats:abstract>	M <abstract>	O description	Markdown → XML transform
Funding / Grant ID	R <fr:program>	R <funding-group>	N/A	FundRef + GrantID fields
Licence URL	R <license_ref>	R <license>	O access_type	CC-BY-4.0 default
References (Linked)	M <citation_list>	R <ref-list>	N/A	Crossref ref-matcher
Full-text URL / PDF	M <resource>	R <self-uri>	M title_url	Stable OJS landing page

Key: M = Mandatory, R = Recommended, O = Optional, N/A = Not in profile

The shaded “superset” column guided the C++ generator: any element marked M or R in any profile is exported by default.

Schema Versions:

- Crossref: Metadata v5.3.0 (2023)
- DOAJ: CSV/XML v2023
- EBSCO: KBART v2022

3.3.4 Relevance to ASPS: Strategic Metadata Compliance

Speaking about Alwaha Scientific Publishing Services (ASPS) as a relatively young regional publisher with aspirations for global scholarly visibility, metadata compliance is not merely a technical obligation but a strategic lever for global visibility, that is why strict adherence to Crossref and JATS standards is strategically essential [27, 66]. While Dublin Core suffices for basic metadata exposure in OAI-PMH feeds, it lacks the granularity needed for robust citation tracking, institutional recognition, and funder compliance.

At its present stage, ASPS relies exclusively on the Crossref XML schema for metadata generation and deposition, as its journals are not yet indexed in directories like DOAJ. This narrow focus on Crossref XML reflects pragmatic resource allocation: Crossref's DOI system is foundational to ASPS's credibility, enabling persistent article links, citation tracking, and integration into global citation networks. By generating Crossref-compliant XML, ASPS ensures timely DOI registration, a non-negotiable requirement for authors seeking internationally recognized publications. While Dublin Core remains useful for basic metadata exposure—particularly through OAI-PMH feeds and institutional repository archiving—it lacks the structural complexity necessary for advanced citation tracking, funder compliance, and full interoperability with global indexing platforms.

The decision to prioritize Crossref XML was driven by immediate operational needs. Crossref's schema mandates enriched metadata fields—such as ORCIDs, funding acknowledgments, and reference lists with DOIs that manual workflows at ASPS struggled to deliver consistently. Missing or misformatted fields previously caused deposit rejections, delaying DOI assignments by weeks and undermining author trust. The custom indexing tool resolves these bottlenecks by automating error-free XML generation, ensuring that every article deposited meets Crossref's requirements for metadata completeness, enabling faster DOI registration, better integration with citation tracking systems, and seamless interaction with Crossref's API infrastructure.

However, ASPS's long-term ambitions extend beyond Crossref. While the publisher is not yet indexed in directories such as the Directory of Open Access Journals (DOAJ) and broader platforms such as Scopus or EBSCO, future eligibility will require adopting JATS XML, a schema

demanding even greater granularity (e.g., structured abstracts, affiliation tagging). Anticipating this shift, the tool’s Crossref XML output aligns with JATS-compatible structures where possible.

Another major advantage of building around Crossref XML is the potential for workflow optimization. Once Crossref metadata is deposited correctly, the same XML file can be harvested via OAI-PMH and transformed into other schemas like Dublin Core for institutional repositories, or partially mapped to JATS for repositories requiring higher fidelity content. This allows ASPS to avoid duplicative workflows and rebuilding its metadata infrastructure from the ground up also ensures consistency across platforms, even if DOAJ indexing is pursued later.

In summary, schema compliance is not simply a technical checkbox—it is foundational to ASPS’s institutional legitimacy, visibility, and ability to participate in the global scholarly knowledge system. By designing an XML generating tool that meets Crossref’s rigorous requirements while remaining extensible toward JATS is both pragmatic and forward-looking, ASPS has taken a critical step toward sustainable, standards-aligned publishing operations that can scale alongside its ambitions.

3.4 Crossref XML Generator Design

ASPS’s switch to OJS plugins (Chapter 2) solved many routine indexing headaches but exposed a deeper bottleneck: *qualified staff*. Diamond-OA publishers in the Global South rarely command the same human or financial capital as large commercial houses, and ASPS is no exception. Running the Crossref Export and Reference-Linking plugins still requires editors who understand JATS, ORCID hygiene, and API troubleshooting—skills in short supply locally. To bridge that gap I designed a **stand-alone Crossref XML generator**, written in modern C++20, that allows non-specialist staff to continue using the familiar “one-file-upload” workflow while still producing Crossref-compliant, machine-linked deposits.

The generator accepts two simple CSV files prepared in Excel—**References.csv** (article batch) and **Journals.csv** (journal master list)—parses them into an internal data model, and emits a schema-validated Crossref 4.4.2 XML file ready for upload via the Crossref web-deposit form. Although Crossref has already deprecated 4.x in favour of 5.x, 4.4.2 remains fully accepted and avoids immediate complexity around ROR and funding metadata. This section details its architecture, workflow, and alignment with ASPS’s operational constraints.

3.4.1 Workflow of the Tool

The generator sits at the centre of **four-actor architecture** (Figure 3.1): the *editor*, two *CSV templates*, the *C++ executable*, and Crossref’s external validation endpoints. What follows maps

every touch-point—from the moment an editor fills the spreadsheets to the instant Crossref confirms the deposit is schema-conformant.

Table 3.4: End-to-End Workflow for Crossref XML Generation and DOI Registration

Step	Actor Component	/ Action	Key Integrity Checks
0	Editor	Download latest templates (<i>References.csv</i> & <i>Journals.csv</i>) from version-controlled SharePoint folder.	File names must remain unchanged; UTF-8 without BOM.
1	Editor	Populate <i>References.csv</i> : one row per article (title, abstract, authors, affiliations, reference string, online date). Populate <i>Journals.csv</i> only when a new journal is launched.	Excel data-validation restricts date to dd.mm.yyyy; drop-down lists for ISSN and journal short code prevent typos.
2	Data Ingestion (C++ tool)	<code>readJournals()</code> loads every row of <i>Journals.csv</i> into <code>vector<JournalInfo></code> ; <code>readDoiParams()</code> + <code>readArticles()</code> parse <i>References.csv</i> into memory.	Fails fast if headers mismatch or mandatory cells are blank.
3	Parsing & Normalisation	<code>parseReference()</code> tokenises each reference string to extract author pairs, year, article title, journal title, volume, pages, DOI suffix.	Case-insensitive match to <code>JournalInfo.fullTitle</code> ; warns if no match found.
4	Internal Validation	The executable verifies: (a) at least one article; (b) DOI components not null; (c) first author + title present.	Any violation aborts batch and writes <code>generator.log</code> .
5	XML Generation	<code>generateXML()</code> emits a Crossref 4.4.2 file: <code><doi_batch></code> → <code><journal_article></code> nodes, <code><jats:abstract></code> , linked references, print/online dates.	Reserved XML characters escaped; output filename = <code>{params}{date}_{journalCode}{vol}{artID}.xml</code> .
6	Editor	Drag-and-drop the XML into <i>Crossref Web-Deposit</i> .	Tool itself stops here; external validation is out-of-process.
7	Crossref Validator	Runs schema and business-rule checks (e.g., missing <code><resource></code> , orphan <code><citation></code>). Returns success message or detailed error list.	If errors: editor edits <i>References.csv</i> (or XML) and re-runs the generator; if OK: proceeds to final deposit.
8	Crossref Deposit	On acceptance, DOIs resolve immediately and participation-report metrics update nightly.	Editor monitors deposit-report e-mails for any post-processing errors.

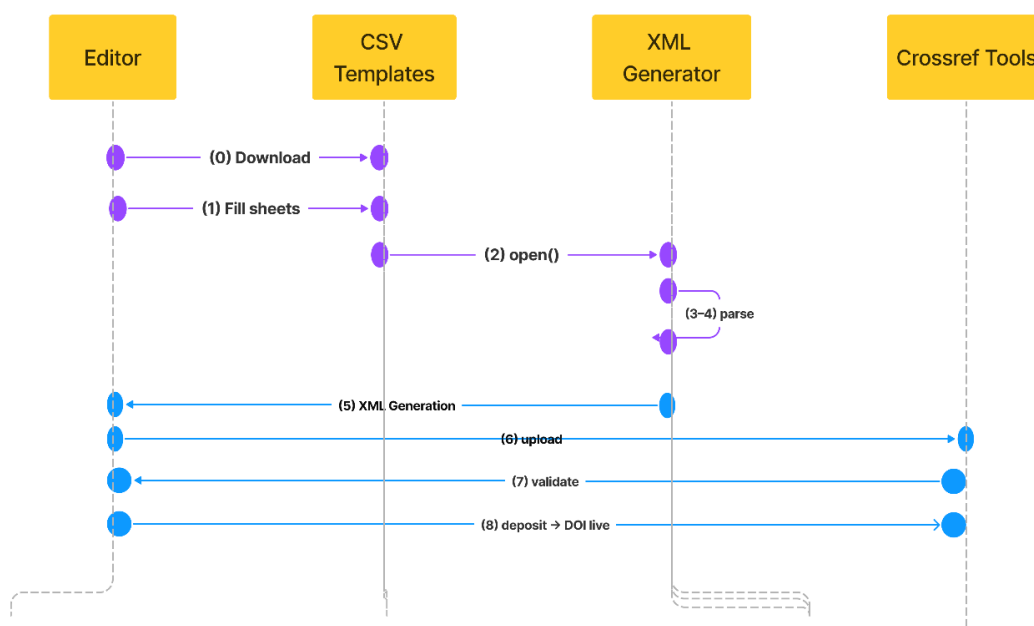


Figure 3.1: Representation of the end-to-end workflow

3.4.2 Input Data Models and XML Generation Results

The generator is intentionally driven by two plain-text CSV templates that can be completed in Microsoft Excel or any text editor. Every rule below is enforced by the program's internal validators; if a field is missing or malformed the batch aborts with an error message so that bad data can never reach Crossref (figure 3.2).

3.4.2.1 Journals.csv

This file is edited only when ASPS launches a new journal or wishes to change branding information. Six columns, always in the same left-to-right order, are expected:

- Full journal name – the title that appears on the cover, reproduced verbatim in the `<full_title>` element of Crossref XML.
- Abbreviated name – the ISO-4 abbreviation used for citation styling; inserted into `<abbrev_title>`.
- Journal short code – a lowercase mnemonic (e.g., `ete`, `jgg`). The code is reused in file-naming, DOI construction and as the `doi_batch_id` prefix.
- ISSN – print ISSN in the conventional four-digit + hyphen + four-digit form (2716-8026). If only an e-ISSN exists, supply that.
- Journal link – the public OJS base URL ending at the journal directory; the generator appends `/article/view/...` or `/issue/view/...` for resource resolving.

- DOI prefix – the static stem supplied by Crossref (10.38208/ete.v). The program concatenates volume and articleID to form unique article DOIs.

	A	B	C	D	E	F	G
1	Full journal name	Abbreviated name	Journal short code	ISSN	Journal link	Doi prefix	
2	Energy and Thermofluids	Energy Thermofluids Eng ete		2716-8026	https://asps-jou	10.38208/ete.v	
3	Journal of Geomechanics	J. Geomec. Geoeng.	jgg	2716-7992	https://asps-jou	10.38208/jgg.v	
4	Journal of Renewable Energy	J. Renew. Energy Technol	jret	2716-8123	https://asps-jou	10.38208/jret.v	
5	Journal of Composites	J. Compos. Adv. Mater.	jcam	2716-8018	https://asps-jou	10.38208/jcam.v	
6							
7							

Figure 3.2: Screenshot of Journals.csv Template in Excel

3.4.2.2 References.csv

Editors work in this sheet for each new issue. It is divided into two logical blocks (figure 3.3):

	A	B	C	D	E	F	G
1	From	To	DOI Journal	DOI Volume issue	DOI Article	Link Volume	
2	1	3		0	1	1	22
3							
4	N°	Ref	Affiliation	Abstract	Online date		
5		1	Kharshiduzzaman, M., Hamja, A. Department of Mechanical and Pro	In the era of rising atmospheric	17.08.2024		
6		2	Biswas, R., Rahman, M. M., Khar Open School, Bangladesh Open Uni	The D2Q9 Bhatnagar-Gross-Kroc	22.09.2024		
7		3	Masri, T., Benhabane, A., Kaci, / Laboratoire de Génie Énergétique e	In this paper a literature review	21.12.2024		
8		4					
9		5					
10		6					
11		7					
12		8					
13		9					
14		10					
15		11					
16		12					
17		13					

Figure 3.3: Screenshot of References.csv Template in Excel

3.4.2.2 ▲ Control block (rows 1–2)

The first header row names six parameters; row 2 provides their numeric values (figure 3.4).

	A	B	C	D	E	F
1	From	To	DOI Journal	DOI Volume issue	DOI Article	Link Volume
2	1	3		0	1	1
3						
4	N°	Ref	Affiliation	Abstract	Online date	
5		1	Kharshiduzzaman, M., Hamja, A. Department of Mechanical and Pro	In the era of rising atmospheric	17.08.2024	
6		2	Biswas, R., Rahman, M. M., Khar Open School, Bangladesh Open Uni	The D2Q9 Bhatnagar-Gross-Kroc	22.09.2024	

Figure 3.4: Screenshot: Control block (rows 1–2)

- From and To define the inclusive line numbers—from one-based indexing—of the article list to be processed.
- DOI Journal, DOI Volume issue, DOI Article: are Boolean flags (0 = skip, 1 = mint) allowing an editor to choose whether the generator should create journal-level DOIs, issue-level DOIs or only article DOIs, for example the DOI Journal is set to 1 only when the editor intends to assign a DOI to a newly launched journal that has not yet been indexed.
- Link Volume: refers to the internal OJS volume identifier used to construct the canonical URL for the issue landing page. Its value is defined semantically by the production manager, in accordance with the volume numbering system of the journal's publishing platform.

Editors rarely change these values after the template is prepared; their main task is the second block.

3.4.2.2 ▲ Article block (row 5 onward)

Row 4 is a fixed header: N°, Ref, Affiliation, Abstract, Online date. Each subsequent line represents one article.

A4	:	✕ ✓ fx	N°			
	A	B	C	D	E	
1	From	To	DOI Journal	DOI Volume issue	DOI Article	Lin
2	1	3	0	1	1	
3						
4	N°	Ref	Affiliation	Abstract	Online date	
5		1 Kharshiduzzaman, M., Hamja, A.	Department of Mechanical and Pro	In the era of rising atmospheric	17.08.2024	
6		2 Biswas, R., Rahman, M. M., Khar	Open School, Bangladesh Open Uni	The D2Q9 Bhatnagar-Gross-Kroc	22.09.2024	
7		3 Masri, T., Benchabane, A., Kaci,	Laboratoire de Génie Energétique e	In this paper a literature review	21.12.2024	
8		4				
9		5				

Figure 3.5: Screenshot: Article block (row 5 onward)

- N° is a serial number included for human readability; the program verifies that it increments sequentially without gaps. It is also referenced in the From and To fields described earlier—for example, setting ‘From’ = 1 and ‘To’ = 2 instructs the generator to process only articles numbered 1 and 2.
- Ref holds the full reference string exactly as it should appear with APA-style format (comma-separated authors, publication year in parentheses, article title, journal title, volume, page range and the preliminary DOI). The program extracts metadata from every fragment and splits this line into—authors, year, title, volume, page range, articleID.

- Affiliation is the institutional text to appear under the first author in Crossref’s <affiliations> node. The generator attaches other co-authors without affiliations to keep the XML concise; future versions of the tool will support individualized affiliation tagging per contributor using an additional CSV column.
- Abstract is free prose. Line-breaks are allowed inside Excel but are converted to <jats:p> in the XML.
- Online date must be in the French day-month-year format dd.mm.yyyy (17.08.2024). The tool splits this value into <day>, <month>, <year> for the online publication date.

Cells that contain commas—Ref and Abstract especially—must be enclosed in double quotation marks when saved as CSV. Excel’s “CSV UTF-8 (Comma delimited)” export option handles this automatically.

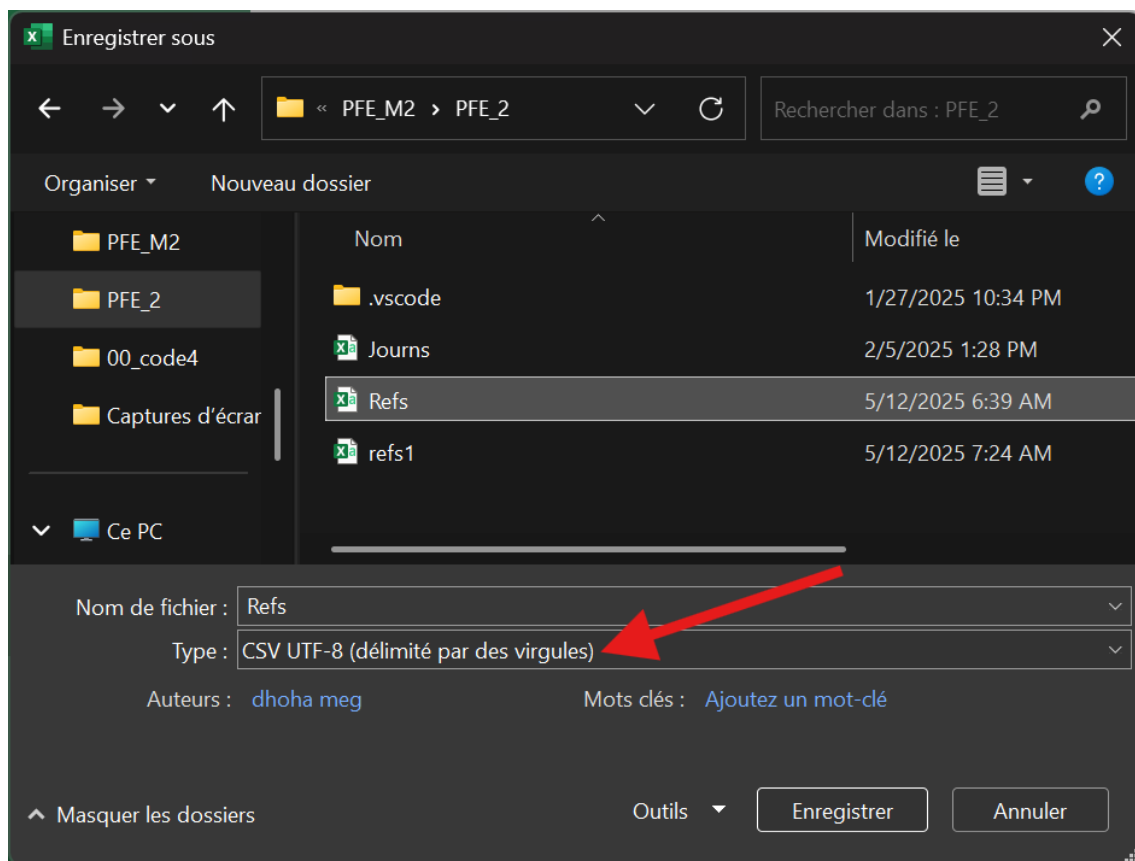


Figure 3.6: Clarify CSV encoding

3.4.2.3 Output Crossref XML file

Once the input files are correctly populated, executing the C++ program triggers the generation of a schema-compliant Crossref XML file in UTF-8 encoding.

Output File Name Logic and Location

The output XML file is generated directly in the same working folder as the executable and the input CSV files (References.csv and Journals.csv), requiring no additional configuration from the user.

Its filename is dynamically constructed concatenating five tokens to create a unique, human-readable name. Using the metadata control parameters and execution timestamp, following the pattern:

<From><To><Journal><Issue><Article>_<Date>_<JournalCode><Volume><ArticleID>.xml

For instance: “13011_05-Mar-2025_ete5752.xml” Where:

- **13011**: the five control digits in References.csv (From=3, To=3, doiJournal=0, doiIssue=1, doiArticle=1)
- **05-Mar-2025**: the system date at run-time (the execution date): day-month-year
- **ete5**: journal short code + volume number extracted from the first reference line
- **752**: articleID (the digits that follow the final dot in the preliminary DOI)

This convention allows editors to sort deposit files chronologically or by journal without opening them, ensuring traceability and version control during batch deposits.

Crossref Schema Compliance

The XML output generated by the indexing tool conforms to Crossref schema version 4.4.2—a deliberate design choice that balances technical stability with the operational realities of ASPS.

While Crossref recommends using the most recent schema (currently v5.4.0) [64], version 4.4.2 remains fully supported and is better suited to ASPS’s current infrastructure for three key reasons:

- **Simplicity and Accessibility**: Schema 5.x requires additional metadata fields—such as ROR IDs, funding blocks, and revised affiliation structures—that ASPS does not yet collect. Version 4.4.2 avoids these complexities while still supporting all essential indexing fields (title, authors, abstract, dates, DOI data).
- **Backward Compatibility and Safety**: As the final stable release in the 4.x series, schema 4.4.2 is accepted for all standard deposits. The generator’s built-in validators ensure compliance with Crossref’s core rules, reducing the risk of rejection or error during submission.
- **Future-Proof Design**: The tool is schema-aware: all version-specific namespaces are centralized, allowing easy migration to schema 5.x when needed. Once ASPS adopts

persistent identifiers like ROR and Grant IDs, upgrading the tool will involve a single configuration change and minor code adjustments.

XML Output Mapping and Structure

- **XML Declaration and Root Element:** This standard declaration specifies the file's character encoding and version. All Crossref XML files must begin with this line. it fixes UTF-8 encoding. The <doi_batch> root element defines the document as a DOI batch. It declares the schema version, XML namespaces, and links to the XSD file for validation. The version="4.4.2" explicitly confirms schema compliance.



```
<?xml version="1.0" encoding="UTF-8"?>
<doi_batch version="4.4.2" xmlns="http://www.crossref.org/schema/4.4.2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:jats="http://www.ncbi.nlm.nih.gov/JATS1"
xsi:schemaLocation="http://www.crossref.org/schema/4.4.2 http://www.crossref.org/schema/deposit/crossref4.4.2.xsd">
<head>
```

Figure 3.7: XML Declaration

- **<head> – administrative metadata:** Everything in the head block is supplied by the generator at run-time. The batch-ID serves as a human-readable key in Crossref's submission logs; the timestamp provides second-level uniqueness in case the same editor submits multiple files within one day.



```
<head>
  <doi_batch_id>jcam_1_821</doi_batch_id>
  <timestamp>05022025132910</timestamp>
  <depositor>
    <depositor_name>alwa:alwa</depositor_name>
    <email_address>adel.benchabane@gmail.com</email_address>
  </depositor>
  <registrant>MDT Deposit</registrant>
</head>
```

Figure 3.8: Administrative metadata

- **<body> Main Content Block:** All journal-related data is contained within the <body> tag.
- **<journal_metadata>:** Defines the journal's title, abbreviation, and ISSN. This is required for proper journal-level identification.

```
<journal_metadata>
  <full_title>Journal of Composites and Advance Materials</full_title>
  <abbrev_title>J. Compos. Adv. Mater.</abbrev_title>
  <issn media_type="print">2716-8018</issn>
</journal_metadata>
```

Figure 3.9: Journal metadata

- **<journal_issue>**: Identifies the volume and issue publication date. Crossref supports multiple media_type values, such as 'print' and 'online'.

```
<journal_issue>
  <publication_date media_type='print'>
    <year>2025</year>
  </publication_date>
  <journal_volume> <volume>1</volume> </journal_volume>
</journal_issue>
```

Figure 3.10: Journal issue

- **Issue-level DOI – optional <doi_data>**: This block registers a DOI for the journal issue, pointing to its landing page on the publisher's OJS website.

```
<!-- ===== -->
<doi_data>
  <doi>10.38208/jcam.v1</doi>
  <resource>https://asps-journals.com/index.php/jcam/issue/view/22</resource>
</doi_data>
```

Figure 3.11: Issue level DOI

- **<journal_article>**: Simple metadata element for the article's full title.

```
</doi_data>
<journal_article publication_type='full_text'>
  <titles>
    <title>
      Date palm seeds waste in Algeria:
      Mechanical characterization of epoxy/date palm seed composites
    </title>
  </titles>
```

Figure 3.12: Journal article

- **<contributors>**: Defines authorship information. The first author can optionally include institutional affiliations.

```

<contributors>
  <person_name sequence='first' contributor_role='author'>
    <given_name>T.</given_name>
    <surname>Masri</surname>
    <affiliations>
      <institution>
        <institution_department>
          Laboratoire de Génie Energétique et Matériaux (LGEM),
          Université de Biskra,B.P. 145, 07000, Biskra, Algeria
        </institution_department>
      </institution>
    </affiliations>
  </person_name>

```

Figure 3.13: Contributors

- **<jats:abstract>**: Embedded in a JATS-compliant <jats:abstract> block. This allows Crossref-compatible platforms (like PubMed Central or Scopus) to parse abstracts correctly.

```

<jats:abstract xml:lang='en'>
  <jats:p>In this paper a literature review and a
</jats:abstract>

```

Figure 3.14: Abstract

- **Print and online <publication_date>**: Both print and online publication dates are supported. Each contains day, month, and year sub-elements..
- **<pages>**: Crossref mandates page numbers for print layout clarity and citation tracking.

```

</jats:abstract>
  <publication_date media_type='print'>
    <year>2025</year>
  </publication_date>
  <publication_date media_type='online'>
    <month>12</month>
    <day>21</day>
    <year>2024</year>
  </publication_date>
  <pages>
    <first_page>1</first_page>
    <last_page>11</last_page>
  </pages>

```

Figure 3.15: Sub-elements

- **Article-level <doi_data>**: The unique **DOI** and **resource URL** for the article. This is the most important identifier used for citation linking, DOI resolution, and indexing.

```

<doi_data>
  <doi>10.38208/jcam.v1.821</doi>
  <resource>
    https://asps-journals.com/index.php/jcam/article/view/821
  </resource>
</doi_data>

```

Figure 3.16: Article level DOI

- **Closing tags**: A fully closed document guarantees that the Crossref validator will, at worst, complain about business-rule violations (for example, an un-matched reference) rather than well-formedness errors.

```

</journal_article>
</journal>
</body>
</doi_batch>

```

Figure 3.17: Closing tags

This structure enables ASPS to deposit both journal-level and article-level metadata to Crossref in a fully automated fashion, using only two CSV files as input. By adhering strictly to Crossref's 4.4.2 schema, the generator ensures:

- DOI resolvability,
- Indexing eligibility,
- Metadata compatibility with discovery services.

Editors upload the generated XML via Crossref's web deposit form, first they can validate it with the Crossref Metadata Quality Checker to catch schema or rule violations. If errors arise, they revise the CSV inputs and regenerate the XML. Upon clean validation, the same file is resubmitted activating DOIs within few seconds.

3.4.3 Data Standardization

This section explains the process of data normalization within the tool, ensuring that raw CSV input—entered manually by editors—is reliably transformed into structured, schema-compliant metadata suitable for XML export. Since ASPS's editorial workflows rely on spreadsheet-based data entry, the generator applies strict normalization rules to bridge the gap between human-readable input and machine-parseable structure.

Standardization efforts include:

- **whitespace trimming:** Applies `trim()` to all fields, removing leading/trailing spaces and reducing internal whitespace to single spaces, to prevent malformed XML
- **UTF-8 Encoding Enforcement:** Replaces non-UTF-8 characters (e.g., smart quotes “ ”) with ASCII equivalents during CSV ingestion. Ensures XML compatibility with Crossref's schema, which rejects non-standard Unicode in `<full_title>` or `<affiliations>`.
- **Date Format Conversion:** Converts `dd.mm.yyyy` (e.g., 21.12.2024) into Crossref's `<day>21</day><month>12</month><year>2024</year>`. Uses regex `(\d{2})\.(\d{2})\.(\d{4})` to reject malformed dates (e.g., 2024-12-21).
- **CSV Quirk Handling:** Parses quoted fields (e.g., "Sedira, L.") as single tokens, preventing column misalignment. Converts `""` within quotes to `"` (e.g., "Abstract with ""quotes"" → `<jats:p>Abstract with "quotes"</jats:p>`).
- **Vocabulary Normalization:** journal titles, country names and month names are reconciled against controlled vocabularies (stored in small lookup tables bundled with the executable).

Thus “Energy & Thermofluids Engineering” and “Energy and Thermofluids Engineering” resolve to the same key, preventing duplicate <journal_metadata> blocks inside a batch.

- **Multiple Authors:** Splits comma-separated lists into <person_name> nodes; assigns affiliations only to the first author.
- **Incomplete Records:** Aborts batch if mandatory fields (ISSN, year, DOI prefix) are missing; ignores optional fields (ORCIDs).

In addition, the tool pre-validates mandatory fields (e.g., abstract, title, publication year) and halts execution if any critical element is missing. This guarantees that no malformed or incomplete batch reaches Crossref, effectively shifting quality control from post-deposit review to pre-deposit validation.

Together, these mechanisms form a robust pre-processing layer that shields the XML generator from noisy or inconsistent user input, while also reducing the training burden on ASPS editorial staff—aligning with the tool’s mission of operational accessibility and metadata accuracy.

3.4.4 DOI Automatic Generation

Crossref requires every <doi> element to be globally unique, persistent, and syntactically valid. The generator implements these constraints as follows:

- **Namespace selection:** The DOI stem (10.38208/) is read from Journals.csv.
- **Deterministic suffix pattern:**
 - {journal-code}.v{volume}.{articleID} for articles;
 - {journal-code}.v{volume} for issues.
 - The pattern is assembled by a dedicated buildDOI() function that refuses to emit a DOI if any component is empty or non-numeric.

3.4.5 Overcoming Technical Challenges in Metadata Extraction

While metadata parsing is streamlined for common cases, this section addresses how the tool navigates real-world edge cases that would otherwise cause schema violations or deposit failures.

❖ Handling Complex Edge Cases

- **Incomplete Records:** If essential fields (title, year, affiliation) are missing, the generator halts execution and logs the error, preserving data integrity.

❖ Resilience and Fallback Logic

- **Unknown Journal:** If a reference cites a journal not found in Journals.csv, the tool defaults to fall back to the first listed journal. This is flagged in the log file to avoid silent misclassification.

❖ Error-Handling

- **Validation Checks:** if the error is critical like: Missing year, invalid ISSN, or malformed DOI then the batch aborts. A if they are Non-Critical Warnings like: Missing last page or online date then skips fields, proceeds

3.4.6 UML Diagrams

To comprehensively document the architecture and internal logic of the Crossref XML generator tool, this section presents two Unified Modeling Language (UML) diagrams: a class diagram and a sequence diagram. These diagrams illustrate both the structural components of the software and its runtime behavior during execution [78]. They also show the internal structure of the tool, the flow of data, and the interaction among its core components during execution.

What is UML?

[UML](#), short for Unified Modeling Language, is a standardized modeling language consisting of an integrated set of diagrams, developed to help system and software developers for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems. The UML is a very important part of developing object oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software [79].

3.4.6.1 Class diagram

Class diagrams are one of the most useful types of diagrams in UML as they clearly map out the structure of a particular system by modeling its classes, attributes, operations, and relationships between objects. With our UML diagramming software, creating these diagrams is not as overwhelming as it might appear. This guide will show you how to understand, plan, and create your own class diagrams [80].

The class diagram outlines the tool's core entities—including data models like ArticleData and service classes like XMLWriter and RefParser—and their interrelationships. This structural view helps readers understand how responsibilities are distributed across the tool's components, and how data flows from CSV inputs to validated XML output (figure 3.18).

3.4.6.2 Sequence diagram

The Sequence Diagram models the collaboration of objects based on a time sequence. It shows how the objects interact with others in a particular scenario of a use case. With the

advanced visual modeling capability, you can create complex sequence diagram in few clicks. Besides, some modeling tool such as Visual Paradigm can generate sequence diagram from the flow of events which you have defined in the use case description [81].

The sequence diagram complements this by modeling the interaction between different actors (Editor, C++ Executable) and system components (References.csv, Journals.csv, Crossref Validator). It shows the chronological sequence of operations from tool execution to DOI validation and XML submission, making explicit the feedback loop for error handling and resubmission (figure 3.19).

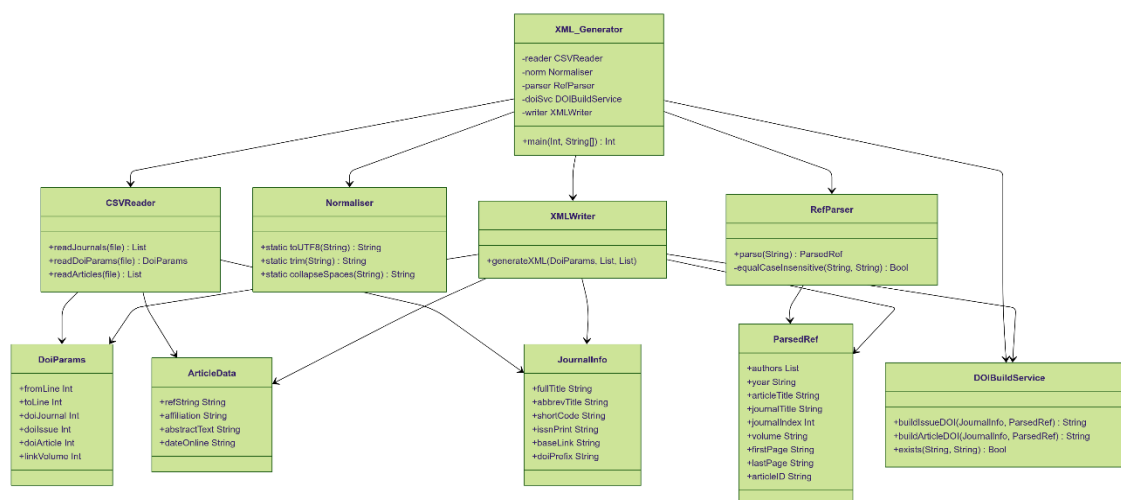


Figure 3.18: Class Diagram of the Crossref XML Generator Tool

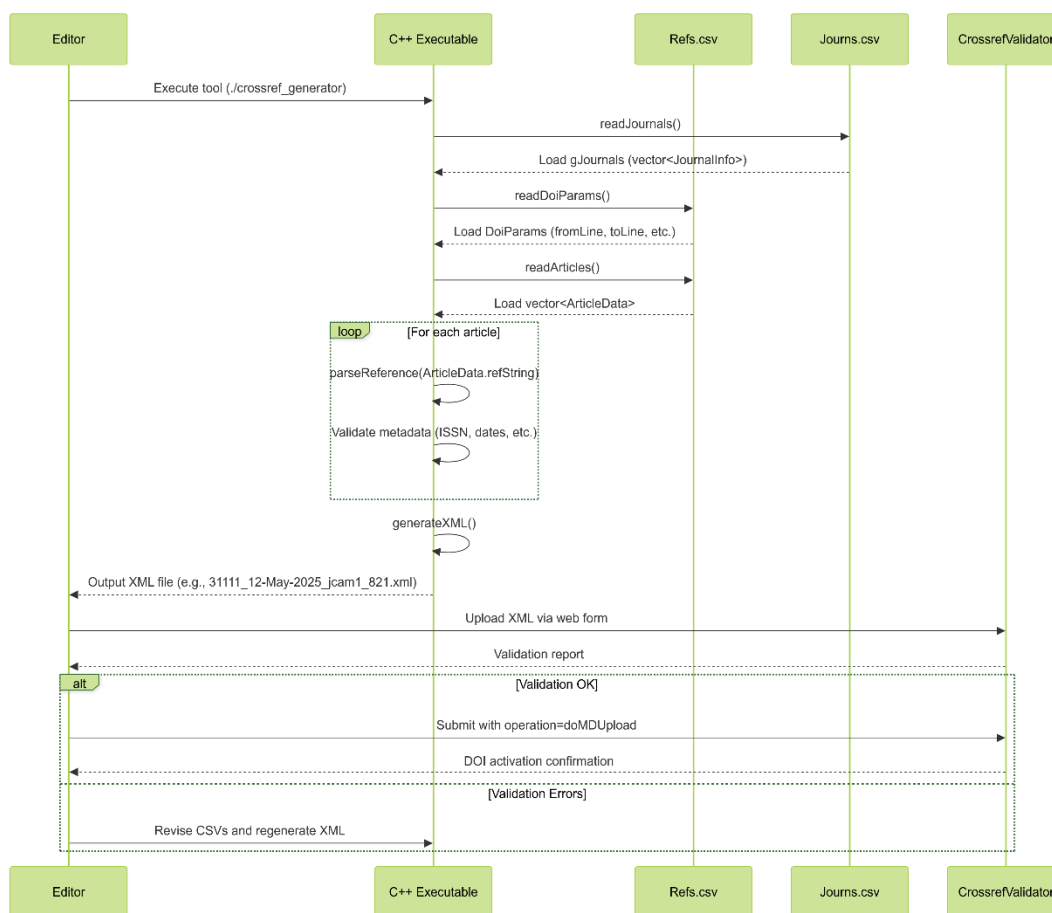


Figure 3.19: Class Diagram of the Crossref XML Generator Tool

Together, these diagrams provide a complete conceptual picture of how the tool achieves its core functionality: transforming human-readable CSV data into machine-validated, publication-ready XML for scholarly indexing.

3.5 Implementation: Languages and tools for development

Beginning by introducing the tools used during the development of the Crossref XML Generator, including both software and hardware. We'll then present the tool's implementation and development.

3.5.1 Hardware tools

The implementation and testing of the XML indexing tool were carried out on a personal development machine running Windows 11 Home, version 24H2. The device used was a DESKTOP-1DSE9AT, equipped with an 11th Gen Intel(R) Core(TM) i7-1165G7 CPU operating at a base frequency of 2.80 GHz, with four physical cores and eight logical processors. The system includes 16 GB of RAM (15.6 GB usable), providing ample memory resources for compiling the C++ program, handling batch metadata processing, and performing real-time file I/O operations. The operating system runs in a 64-bit architecture, supporting efficient execution of multi-threaded

tasks. Additionally, the device features pen and touch support with 10 touch points, although these capabilities were not essential for the development process. The system was fully updated as of February 1, 2025, with OS build 26100.3775 and Windows Feature Experience Pack 1000.26100.66.0, ensuring maximum compatibility and performance throughout the development lifecycle.

3.5.2 3.5.2 Software tools

- **Visual Studio:** The Crossref XML generator was developed using Visual Studio Code (VS Code), a lightweight, cross-platform source code editor by Microsoft. Known for its versatility and strong extension support, VS Code was configured with C++ extensions to enable modern C++20 development. The version used was 1.99.2 (user setup), released on April 10, 2025, running on Windows 11 Home. This build leveraged Electron 34.3.2, Node.js 20.18.3, and Chromium 132.0, providing a stable and responsive development environment throughout the project [82].



Figure 3.20: Visual Studio logo

- **C++:** It is a versatile, high-level programming language, often described as an extension of C, that supports both object-oriented and procedural programming paradigms. It's known for its ability to handle systems programming, create high-performance applications, and its flexibility in managing low-level memory [83].



Figure 3.21: C++ logo.

Why C++?

The decision to develop the indexing tool in C++20 was influenced directly by ASPS's Publication Manager, who prioritized security, performance, and long-term maintainability. The indexing tool was developed in C++20 at the request of the ASPS Publication Manager, who emphasized security and performance as top priorities. Unlike Python and Java, which rely on interpreters or virtual machines, C++ compiles directly into native machine code. This produces optimized, platform-specific binaries that eliminate the need for runtime layers, reducing the attack surface and runtime dependencies.

In contrast, Java’s bytecode runs on the JVM, and Python’s scripts depend on an interpreter—both introducing additional abstraction layers that can be targeted or misconfigured. While these environments offer sandboxing and memory safety, they expose more runtime information and are inherently more complex to secure in production.

C++ binaries also benefit from OS-level protections like DEP and ASLR, and C++20 introduces modern safety features such as smart pointers and `std::span` for safer memory handling. Although binary-level vulnerabilities require careful coding, the resulting executable is faster, more predictable, and better suited to a security-conscious environment like ASPS.

•**Microsoft Excel 2019:** is a spreadsheet application from the Microsoft Office suite, widely used for organizing and processing tabular data. In this project, Excel served as the primary tool for preparing structured metadata in CSV format. Its data validation features, formula support, and ease of use made it accessible for non-technical editors at ASPS. Additionally, its ability to export CSV files encoded in UTF-8 ensured compatibility with the XML generator’s input requirements [84].



Figure 3.22: MS Excel logo

•**XML (eXtensible Markup Language):** is a flexible, text-based language used to store and transport structured data in a platform-independent format. Defined by the W3C, XML enables users to define custom tags and document structures, making it ideal for data interchange across heterogeneous systems. Unlike HTML, which focuses on data presentation, XML emphasizes data meaning and hierarchy, allowing machines and applications to parse, validate, and manipulate content reliably. Its extensibility and strict syntax rules make it widely used in academic publishing, metadata standards (like JATS and Crossref), web services, and configuration files [85].



Figure 3.23: XML logo

3.5.3 Tool Realization

This section presents the concrete implementation of the XML metadata generator developed for ASPS. The tool was built in C++20 and compiled on a Windows-based system using Visual Studio

Code (v1.99.2) as the primary development environment. The following subsections detail the core libraries, declaration structure, and main utility functions used in the codebase. The organization follows the logical flow of the program, from initialization to metadata transformation.

3.5.3.1 Included Libraries

The project relies solely on standard C++ libraries to maximize portability and eliminate third-party dependencies. The libraries used are as the following figure shows:

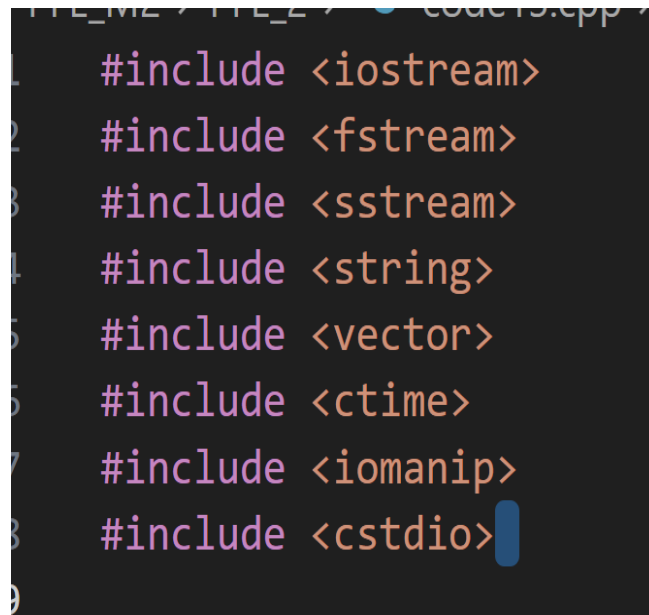
A screenshot of a code editor showing a list of C++ standard library headers being included in a file named 'code13.cpp'. The headers are listed line by line, each preceded by a line number (1 through 9). The headers are: <iostream>, <fstream>, <sstream>, <string>, <vector>, <ctime>, <iomanip>, and <cstdio>. The code is color-coded with purple for the #include keyword and orange for the header names. A blue cursor is visible at the end of the last line of code.

Figure 3.24: Tool libraries

- `#include <iostream>`: Provides input/output stream handling, essential for console logging.
- `#include <fstream>`: Enables file input and output operations, particularly reading CSVs and writing XML.
- `#include <sstream>`: Allows for string stream parsing, useful when tokenizing CSV lines.
- `#include <string>`: Fundamental for handling string operations.
- `#include <vector>`: Supports dynamic arrays for storing articles and journal metadata.
- `#include <ctime>` and `#include <iomanip>`: Used for generating timestamps in the output file name.
- `#include <cstdio>`: Provides additional utilities for low-level file operations and debugging.

These libraries form the foundation for the tool's lightweight design and robust data handling pipeline.

3.5.3.2 Declaration Section

Figure 3.36 below summarizes the declaration of key data structures and variables:

- struct DoiParams: Stores user-defined DOI flags and batch configuration (From, To, DOI Journal/Issue/Article).

```
struct DoiParams {  
    int fromLine;  
    int toLine;  
    int doiJournal;  
    int doiIssue;  
    int doiArticle;  
    int linkVolume;  
};
```

Figure 3.25: Doi Parameters

- struct ArticleData: Encapsulates per-article fields like reference string, affiliation, abstract, and publication date.

```
struct ArticleData {  
    std::string refString;  
    std::string affiliation;  
    std::string abstractText;  
    std::string dateOnline;  
};
```

Figure 3.26: Article Data

- struct JournalInfo: Holds static journal-level metadata such as full name, ISSN, DOI prefix, and OJS base link.

```
struct JournalInfo {  
    std::string fullTitle;    // "Full journal name"  
    std::string abbrevTitle;  // "Abbreviated name"  
    std::string shortCode;    // "Journal short code"  
    std::string issnPrint;    // "ISSN"  
    std::string baseLink;     // "Journal link"  
    std::string doiPrefix;    // "Doi prefix"  
};
```

Figure 3.27: Journal Information

- struct ParsedRef: Represents a tokenized citation line, breaking down a full reference into authors, year, article title, journal title, volume, and page range.

```

struct ParsedRef {
    std::vector<std::pair<std::string, std::string>> authors;
    std::string year;
    std::string articleTitle;
    std::string journalTitle;
    int journalIndex;
    std::string volume;
    std::string firstPage;
    std::string lastPage;
    std::string articleID;
};

```

Figure 3.28: APA citation line

Those custom structs simplify memory management and logical grouping of metadata fields, facilitating accurate XML generation later in the pipeline.

3.5.3.3 Sub-functions and Utility Methods

The tool employs a series of modular helper functions, each responsible for a distinct preprocessing or parsing step. Key examples include:

- `readJournals()`: Parses the `Journals.csv` file and stores journal records in memory (`gJournals`). Fails gracefully if formatting is incorrect or missing.
- `readDoiParams()`: Reads and validates the control block of `References.csv` (e.g., `From`, `To`, `DOI` flags).
- `readArticles()`: Loads each article's row from `References.csv`, supporting quoted abstract and reference fields with embedded commas.
- `parseReference()`: Tokenizes the citation field (`Ref`) using comma and parenthesis delimiters to extract authorship, title, volume, and DOI suffix.
- `generateXML()`: Constructs a well-formed Crossref-compliant XML file, filling in all required nodes like `<titles>`, `<contributors>`, `<abstract>`, `<doi_data>`, and publication dates.

Each function is designed to isolate concerns and ensure the tool is maintainable and extendable—especially as future requirements (like JATS support) may emerge.

3.6 Conclusion

This chapter translated strategic metadata theory into a working tool tailored to ASPS's resource limitations. Through a comparative analysis of Dublin Core, JATS, and Crossref, and the metadata requirements of Crossref, DOAJ, and EBSCO, a clear blueprint was established for “minimum viable compliance.” The resulting C++20 application bridges Excel-based input and machine-

actionable XML output, handling UTF-8 encoding, strict date formatting, and citation parsing to produce a validated Crossref 4.4.2 deposit file with traceable filenames.

By automating DOI generation and XML validation, the tool removes prior bottlenecks in ASPS's metadata workflows, enabling editors to focus on content while ensuring technical compliance. The UML diagrams in §3.4.6 provide a roadmap for future extension toward JATS or KBART support.

With development complete, the project now moves to testing and validation. Chapter 4 the tool's performance will be evaluated against real-world scenarios, including schema validation, DOI registration accuracy, and metadata completeness—marking the final step from prototype to operational deployment.

4 Testing, Validation, and Deployment

4.1 Introduction

This chapter presents the post-development phase of the Crossref XML generator tool, focusing on systematic testing, schema validation, and the final deployment process at Alwaha Scientific Publishing Services (ASPS). After implementing the tool in C++20 and standardizing the metadata entry process through Excel templates,

Following the completion of the Crossref XML generator's development, the focus of this chapter is the empirical evaluation and real-world deployment of the tool within Alwaha Scientific Publishing Services (ASPS). As the primary aim of the project was to streamline and automate the generation of metadata-compliant XML files for DOI registration, it was essential to test the tool under diverse operational scenarios and assess its compliance with Crossref's strict schema requirements. This chapter presents a structured methodology for validation, real-time schema testing using Crossref's Metadata Quality Checker, and deployment results across multiple use cases—including volume-only deposits, article-level DOI creation, and batch processing for full issues. Additionally, the chapter documents the institutional impact of tool integration at ASPS, quantifying improvements in efficiency, schema conformance, and editorial productivity. Finally, strategic recommendations are outlined for extending the tool's capabilities toward broader interoperability and metadata richness in future phases.

4.2 Real-World Validation

To evaluate the Crossref XML generator developed in this project under production conditions, I conducted a live test case using Crossref's official Metadata Quality Checker, available at <https://www.crossref.org/02publishers/parser.html>. This test emulated the complete metadata deposition workflow for one of ASPS article records. The objective was to traverse the entire deposition pipeline starting from CSV input to final XML parsing and to evaluate every intermediate artefact with Crossref's Metadata Quality Checker, the reference service for schema compliance.

First, the test process began by preparing the required input datasets:

- **Journals.csv:** A structured journal-level descriptors updated with its title, abbreviation, ISSN, short code, and DOI prefix.

- **References.csv:** An accompanying article metadata file, including the full reference string, author information, affiliation, abstract, and online publication date.

These files were used to generate a Crossref-compliant XML file. The output was submitted to the Metadata Quality Checker for validation.

4.2.1 Initial Submission and Schema Rejection

The generator produced first XML file (11011_05-Feb-2025_ete5759.xml), which was rejected due to schema violations, which are detailed below and shown in Figure 4.1. Upon **submission, three key errors were reported:**

- **Invalid XML Hierarchy:** The `<doi_data>` element appeared outside the required `<journal_article>` structure. The schema expects `<doi_data>` to be nested inside `<journal_article>`, and its misplacement triggered a hierarchy violation.
- **Incorrect Tag Nesting:** Within the `<affiliations>` block, an `<institution_department>` tag was used without an enclosing `<institution_name>` element, violating the structural expectations of schema version 4.4.2. which is disallowed by the XSD
- **Structural Dependency Error:** The `<institution_department>` tag was not only misplaced but also improperly used in isolation, lacking the mandatory parent node. This error illustrated how the schema enforces logical nesting and contextual dependencies, not just presence of elements.

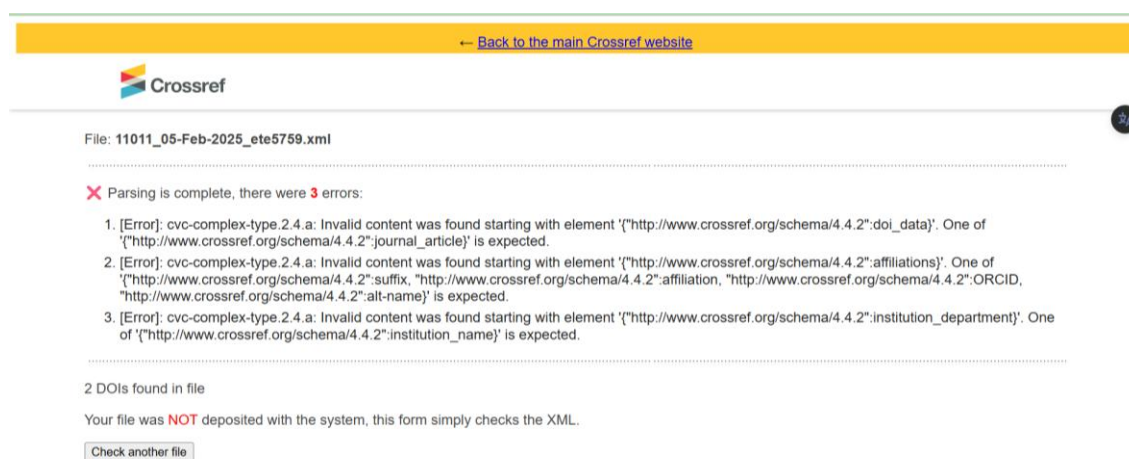


Figure 4.1: Failed Validation Report

These issues confirmed that schema compliance is not only about including the correct elements but about placing them in the precise structural and hierarchical order defined by the standard. This exercise helped providing critical feedback for refining the generator's error-handling routines, ensuring robust, production-ready XML output.

4.2.2 Schema Corrections and Revalidation

Based on the error feedback, the XML generator was corrected to reflect Crossref’s schema rules:

- `<doi_data>` was nested properly inside the `<journal_article>` element.
- reorder `<affiliation>`.
- delete `<institution_department>` and `<institution_name>`.

The corrected file, resubmitted to the Metadata Quality Checker, parsed without error (*Figure 4.2*) and was declared “*Parsing is complete, your file is valid and ready to be submitted.*” This successful validation authorised DOI registration and established the generator’s ability to emit fully compliant metadata for journal articles.

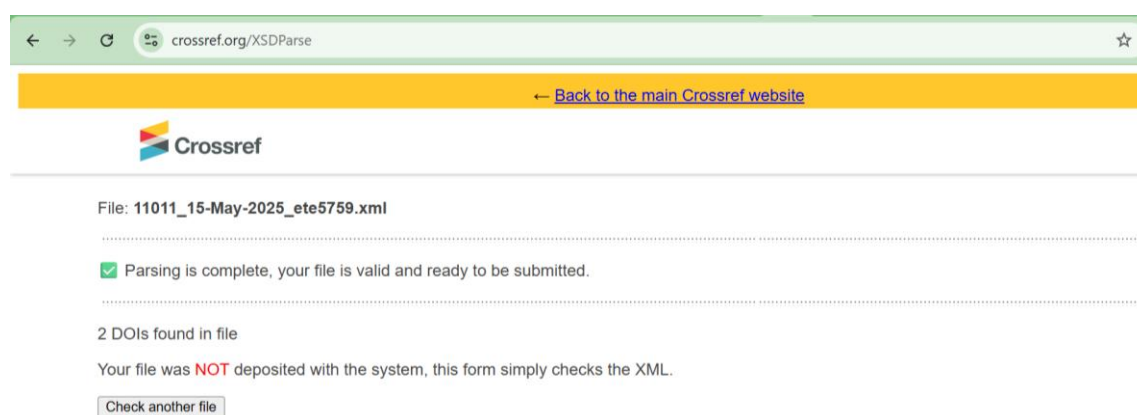


Figure 4.2: Post-Correction Successful Validation Report

The exercise demonstrated the Metadata Quality Checker’s indispensable role in the workflow: it functions not merely as a gatekeeper but as a formative diagnostic tool, guiding iterative refinement of automated metadata pipelines.

4.3 Validation Strategy

To guarantee the robustness, accuracy, and schema compliance of the Crossref XML generator across diverse use cases—including varying batch sizes, publication timelines, and journal configurations—a multi-tiered validation framework was designed and implemented. This framework encompasses automated schema checks, internal metadata integrity validations, and systematic output traceability, ensuring that every XML file produced meets both syntactic and semantic requirements before submission to Crossref’s deposit infrastructure.

1. The validation process includes the following components:**Schema-Level XML Validation:** Each generated XML file is first subjected to formal schema validation against Crossref’s XSD (version 4.4.2). This ensures structural correctness, including tag hierarchy,

element order, and mandatory field presence. The Crossref Metadata Quality Checker used to perform this validation.

2. **Internal Metadata Field Integrity Checks:** The generator includes automated internal validation routines to verify the integrity and format of metadata fields prior to XML construction. These checks include:
 - a. Ensuring proper UTF-8 encoding for fields such as <title>, <affiliations>, and <abstract>.
 - b. Validating and converting publication dates from dd.mm.yyyy to schema-compliant <day>, <month>, <year> format.
 - c. Enforcing syntactic correctness of DOI structures.
 - d. Verifying minimum metadata completeness, including presence of author name(s), article title, publication year, and assigned DOI prefix.
3. **Filename Structure and Output Traceability:** To facilitate batch tracking and reproducibility, each generated XML file includes embedded metadata identifiers in the filename, such as DOI flags, creation date, volume number, and article ID. This enables consistent version control, chronological organization, and easier debugging across large-scale deployments.

4.4 Testing Scenarios and Methodology

To rigorously assess the flexibility, accuracy, and schema compliance of the Crossref XML generator, a series of real-world validation scenarios were executed. These were designed to reflect diverse operational contexts encountered by publishers, including volume-level DOI registration, single-article deposition for existing journals, and multi-article batch deposits. The tests aimed not only to verify functional correctness, but also to assess the tool's ability to manage edge cases and minimal configurations.

4.4.1 Scenario 1: Creating a DOI for a New Journal volume (No Articles)

Objective

This scenario examined whether the XML generator could correctly mint and register a volume-level DOI for a newly launched journal that did not yet contain any published articles. The goal was to confirm the generator's capacity to handle metadata deposits at the issue level without triggering schema violations due to the absence of <journal_article> nodes.

Pre-Validation Status

Before initiating the test, a DOI lookup using the Crossref Metadata Quality Checker confirmed that the volume-level DOI (10.38208/jcam.v1.821) had not yet been registered. The system

returned the message “DOI Not Found,” indicating the DOI was absent from the Crossref registry (Figure 4.3).

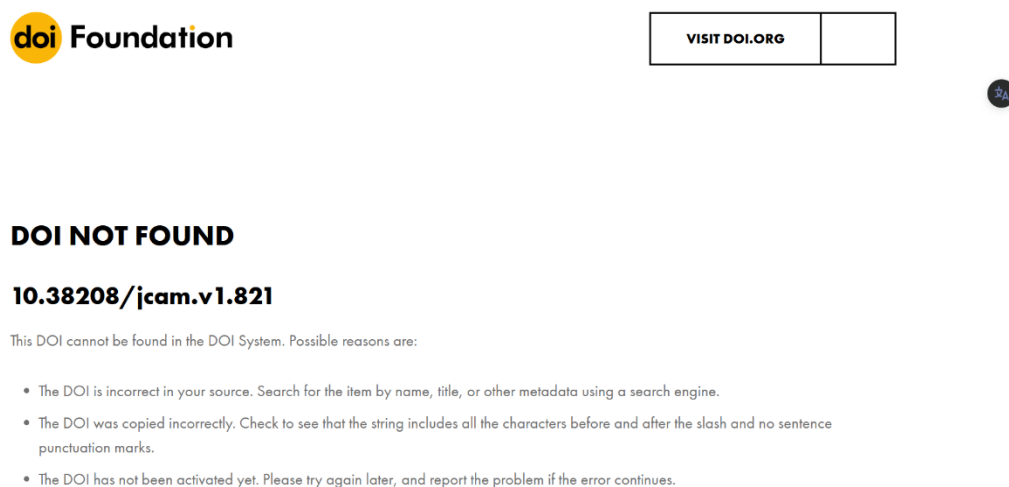


Figure 4.3: webpage indicating that the DOI is absent from the Crossref registry.

Input Preparation

- **Journals.csv:** Populated with the journal’s full name, ISO-4 abbreviation, print ISSN (2716-8034), OJS base URL, and Crossref-assigned DOI prefix 10.38208/jcam.v. A red-bordered highlight in *Figure 4.4* shows the relevant metadata entry.

	A	B	C	D	E	F	G
1	Full journal name	Abbreviated name	Journal short	ISSN	Journal link	Doi prefix	
2	Energy and Thermo	Energy Thermofluids Eete		2716-8026	https://asps-	10.38208/ete.v	
3	Journal of Geomech	J. Geomec. Geoeng.	jgg	2716-7992	https://asps-	10.38208/jgg.v	
4	Journal of Renewab	J. Renew. Energy Tech	iret	2716-8123	https://asps-	10.38208/iret.v	
5	Journal of Composi	J. Compos. Adv. Mater	jcam	2716-8018	https://asps-	10.38208/jcam.v	
6							
7							

Figure 4.4: Journal Metadata Configuration

- **References.csv:** The article block was left empty to simulate a volume with no accepted submissions. The control block was configured as follows: doiJournal = 0, doiVolume = 1, doiArticle = 0, and linkVolume = 23. This configuration instructs the tool to generate only the journal- and issue-level metadata. A visual of this input setup is shown in *Figure 4.5*.

	A	B	C	D	E	F
1	From	To	DOI Journal	DOI Volume	DOI Article	Link Volume
2	0	0	0	1	0	0
3						

Figure 4.5: Control Parameters for Volume-Level DOI

Process

- Upon execution, the XML generator produced a compact metadata deposit containing only the `<journal_metadata>` and `<journal_issue>` blocks. The `<journal_article>` node was correctly omitted, as no articles were scheduled for publication in this volume. The resulting XML structure is presented in *Figure 4.6*.

```

<?xml version="1.0" encoding="UTF-8"?>
<doi_batch version="4.4.2" xmlns="http://www.crossref.org/schema/4.4.2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <head>
    <doi_batch_id>jcam_1_821</doi_batch_id>
    <timestamp>15052025174408</timestamp>
    <depositor>
      <depositor_name>alwa:alwa</depositor_name>
      <email_address>adel.benchabane@gmail.com</email_address>
    </depositor>
    <registrant>MDT Deposit</registrant>
  </head>
  <body>
    <journal>
      <journal_metadata>
        <full_title>Journal of Composites and Advance Materials</full_title>
        <abbrev_title>J. Compos. Adv. Mater.</abbrev_title>
        <issn media_type="print">2716-8018</issn>
      </journal_metadata>
      <journal_issue>
        <publication_date media_type='print'>
          <year>2025</year>
        </publication_date>
        <journal_volume> <volume>1</volume> </journal_volume>
        <doi_data>
          <doi>10.38208/jcam.v1</doi>
          <resource>https://asps-journals.com/index.php/jcam/issue/view/23</resource>
        </doi_data>
      </journal_issue>
    </journal>
  </body>
</doi_batch>

```

Figure 4.6: Minimalist XML Output

- This file was submitted to Crossref's Metadata Quality Checker for schema and business-rule validation. As seen in *Figure 4.7*, the file passed all checks, and the DOI was marked as ready for registration.

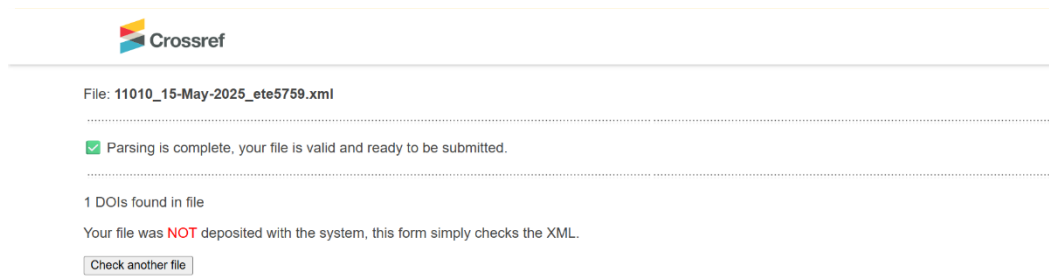


Figure 4.7: Metadata Quality Checker Validation

- To initiate a metadata deposit, access to a Crossref member account is required. The login interface (Figure 4.8) prompts for valid credentials. For this test, the login was completed using the credentials of the ASPS production manager, who holds an active Crossref membership.

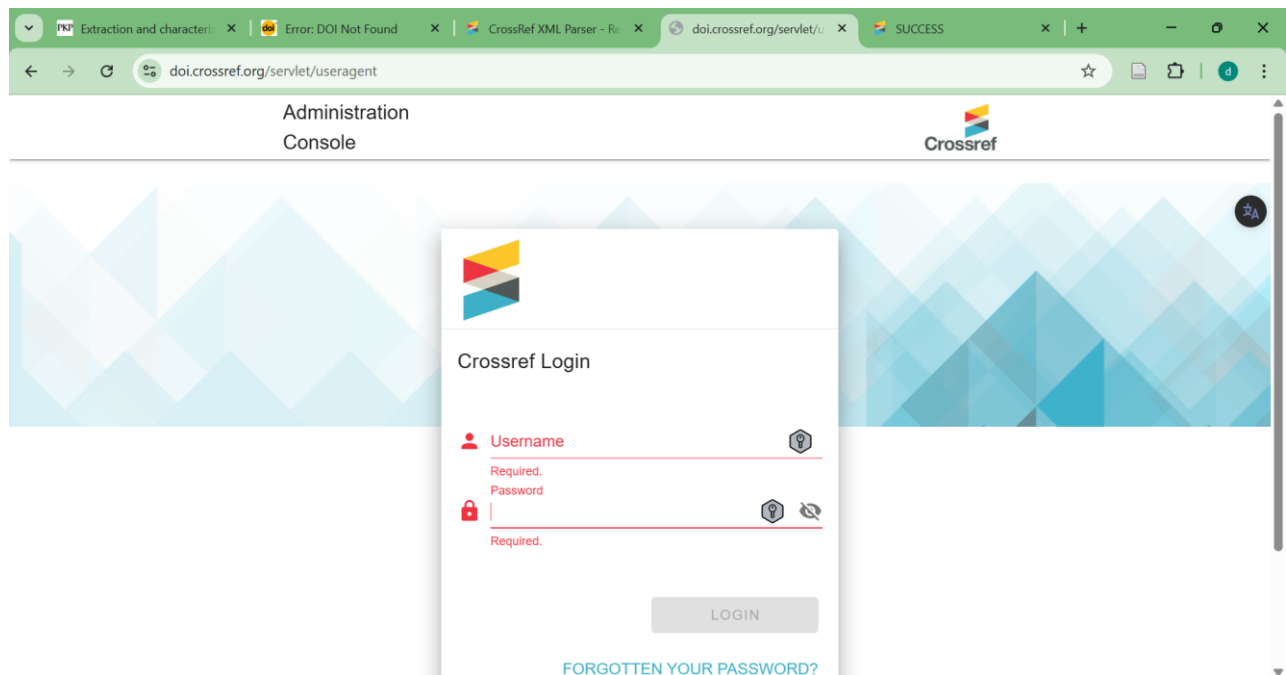


Figure 4.8: Crossref Member Account Login

- Upon successful authentication, the user is redirected to the Crossref account's home interface (Figure 4.9). This dashboard includes links to key submission management tools such as "Submission Administration," "Upload Submissions," and "Show My Submission Queue."

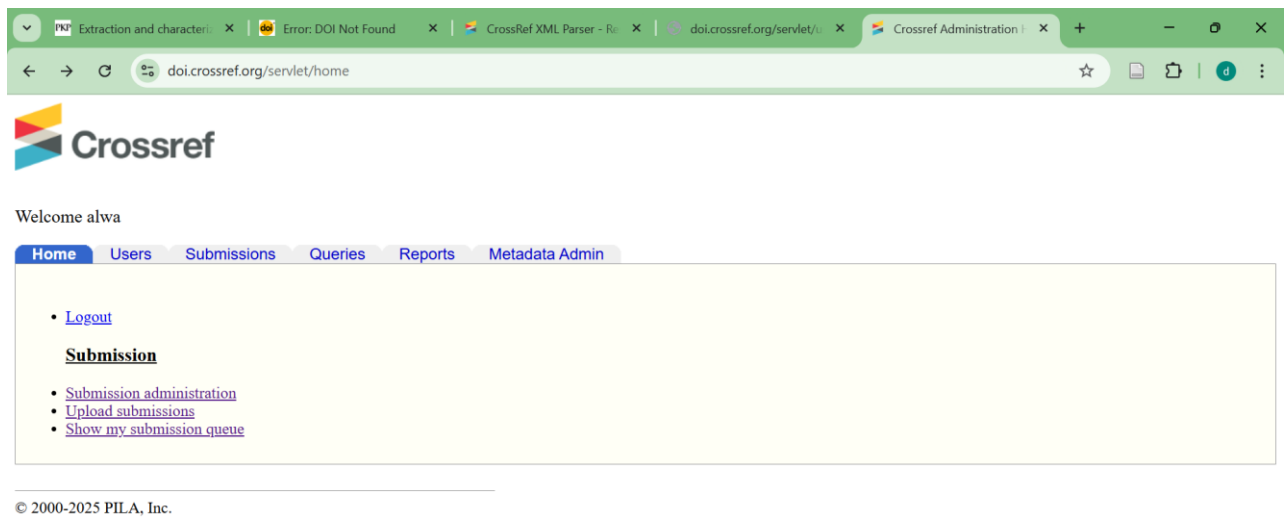


Figure 4.9: Crossref Home Page

- The validated XML metadata file was submitted through the manual upload form available under the “Upload Submissions” tab (Figure 4.10). The file type was selected as "Metadata" to ensure appropriate processing. After selecting the XML file (Figure 4.11), the upload was initiated by clicking the "Upload" button.

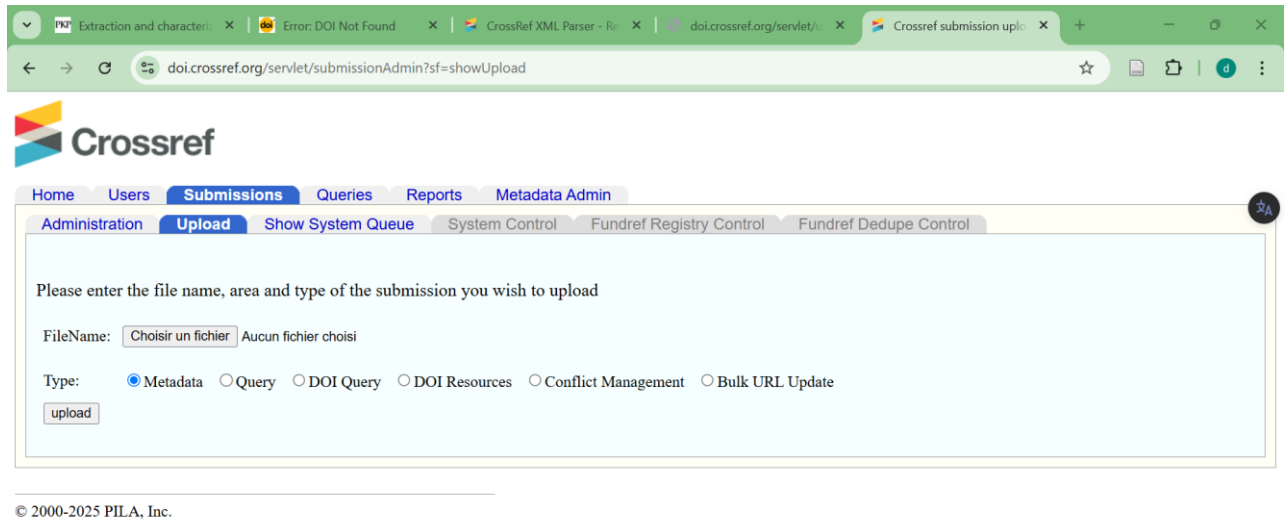


Figure 4.10: Crossref Web Deposit Form

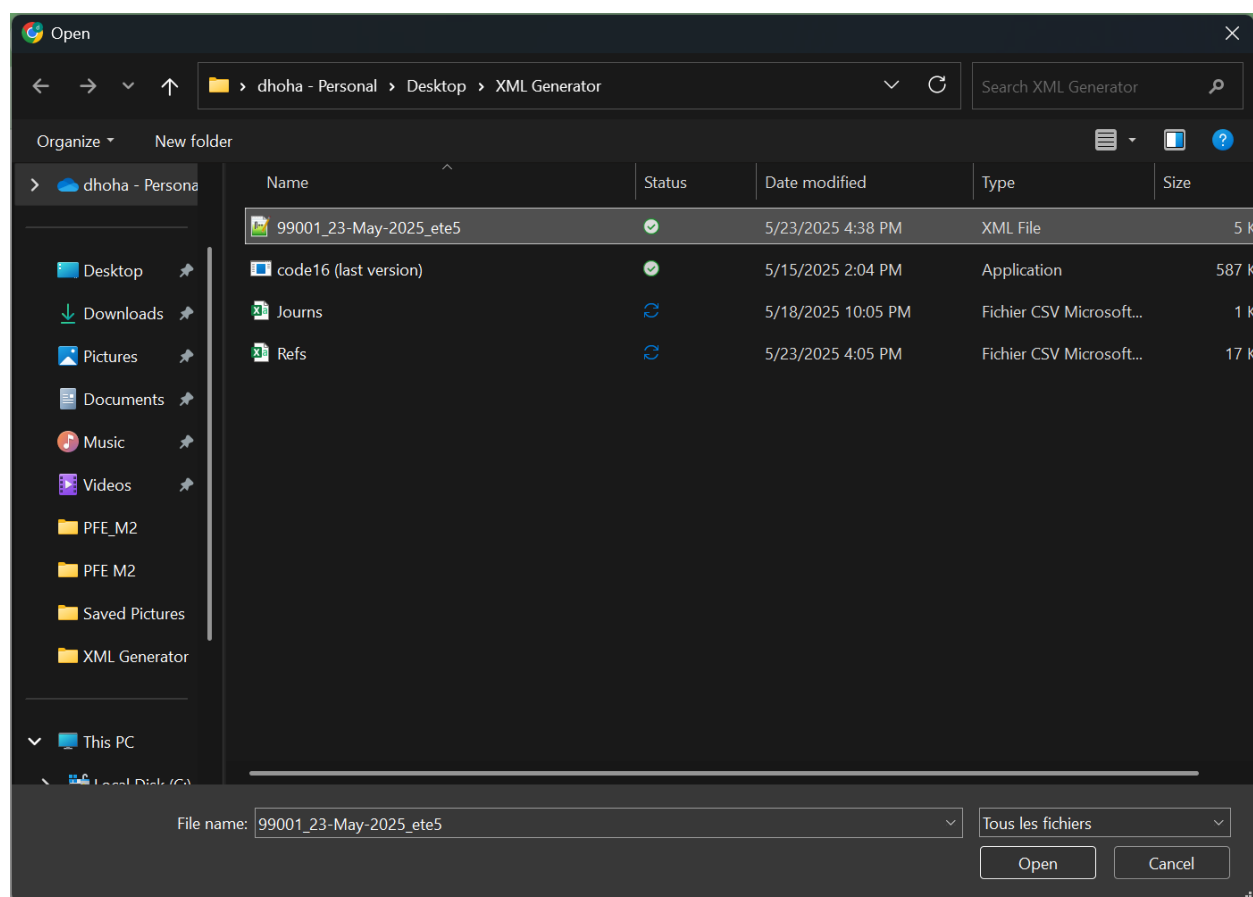


Figure 4.11: Select XML File to Upload

Outcome:

- A successful metadata upload is immediately acknowledged with a confirmation screen (Figure 4.12), stating: “Your batch submission was successfully received.” This indicates that the file passed Crossref’s ingestion checks without triggering schema or business-rule violations.

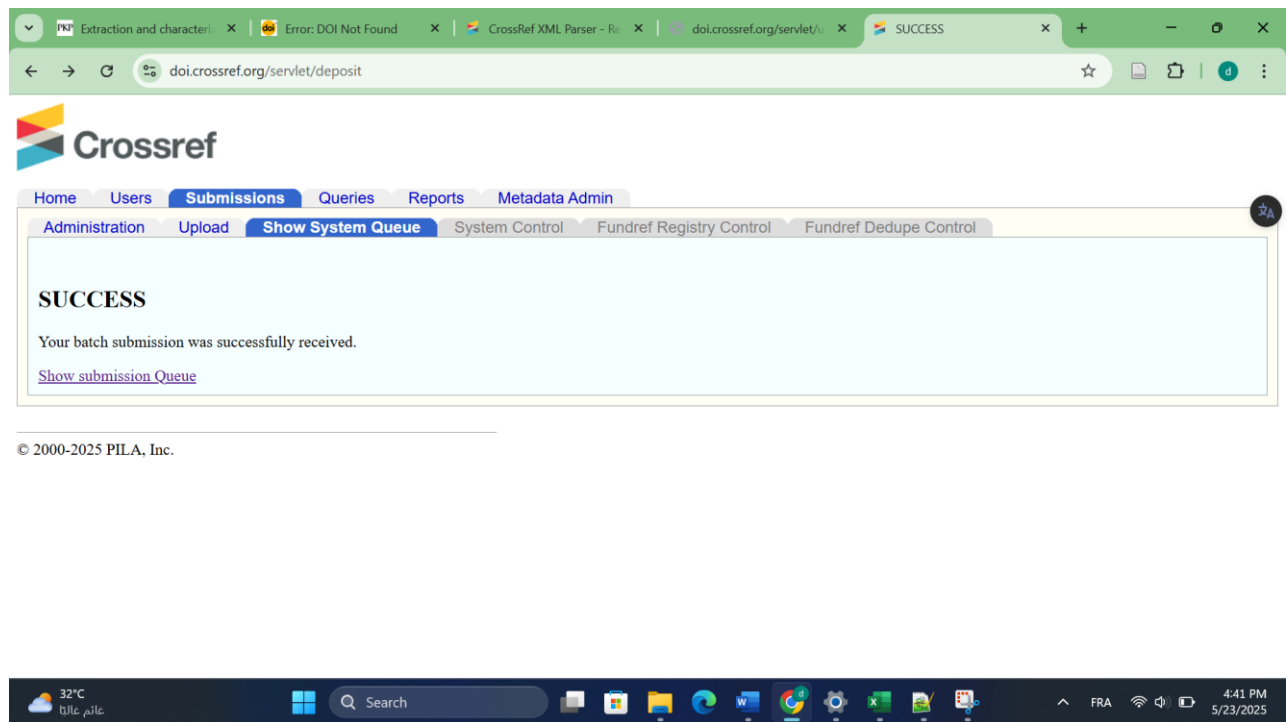


Figure 4.12: Crossref Successful Submission Message

- Post-deposit, the registered DOI (10.38208/jcam.v1) became instantly resolvable. As seen in Figure 4.13, accessing the DOI redirects users to the volume's landing page on the ASPS-hosted Open Journal Systems (OJS) platform. This confirms the correct linkage and accessibility of the volume-level DOI.



Figure 4.13: JCAM landing page

4.4.2 Scenario 2: Depositing a DOI for a New Article in an Existing Journal

Objective

This test case was designed to evaluate the tool's capability to append a new article to an existing journal (Energy and Thermofluids Engineering) by generating the required metadata for a previously indexed publication. The target was a new article submission under an existing volume, requiring accurate representation of both volume-level and article-level metadata.

Input Preparation

- **Journals.csv:** Retained ETE's established metadata, including its ISSN (2716-8026), abbreviated title (Energy Thermofluids Eng.), and DOI prefix (10.38208/ete.v), as illustrated in **Figure 4.14**.

	A	B	C	D	E	F	G
1	Full journal name	Abbreviated name	Journal short	ISSN	Journal link	Doi prefix	
2	Energy and Thermo	Energy Thermofluids Eete		2716-8026	https://asps-	10.38208/ete.v	
3	Journal of Geomech	J. Geomec. Geoeng.	jgg	2716-7992	https://asps-	10.38208/jgg.v	
4	Journal of Renewab	J. Renew. Energy Tech	jret	2716-8123	https://asps-	10.38208/jret.v	
5	Journal of Composi	J. Compos. Adv. Mater	jcam	2716-8018	https://asps-	10.38208/jcam.v	
6							

Figure 4.14: ETE Journal Metadata Retention

- **References.csv:** Populated with metadata for a single article submitted under Volume 5. Fields included author details, article title, abstract, publication date (17.08.2024), and references. Control parameters were set to mint article-level DOIs (doiArticle=1, doiIssue=0). Figure 4.15 displays the relevant entries.

A	B	C	D	E
From	To	DOI Journal	DOI Volume issue	DOI Article
1	1	0	0	1
N°	Ref	Affiliation	Abstract	Online date
1	Kharshiduzzaman, M., Hamji	Department of Mechan	In the era of rising atm	17.08.2024

Figure 4.15: Article-Level Control Parameters

Process

- The tool generated a Crossref-compliant XML file consisting of a <journal_issue> node for Volume 5 and a single <journal_article> node representing the new submission. The schema structure of the generated file is shown in Figure 4.16.

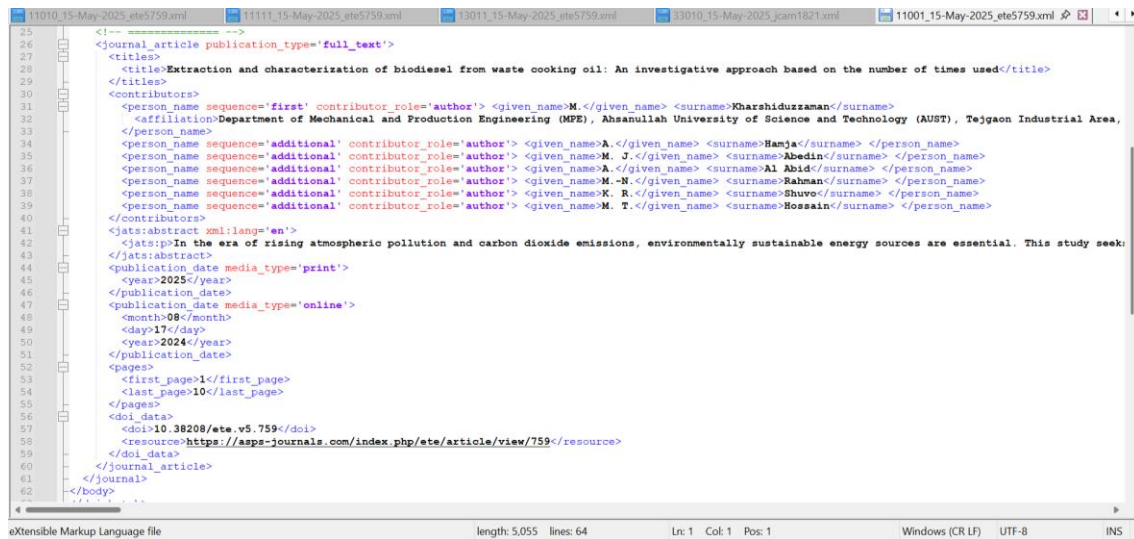


Figure 4.16: XML Output for Article Integration.

- The XML was validated using the Crossref Metadata Quality Checker, which confirmed structural accuracy and completeness. The validation report is shown in **Figure 4.17**.

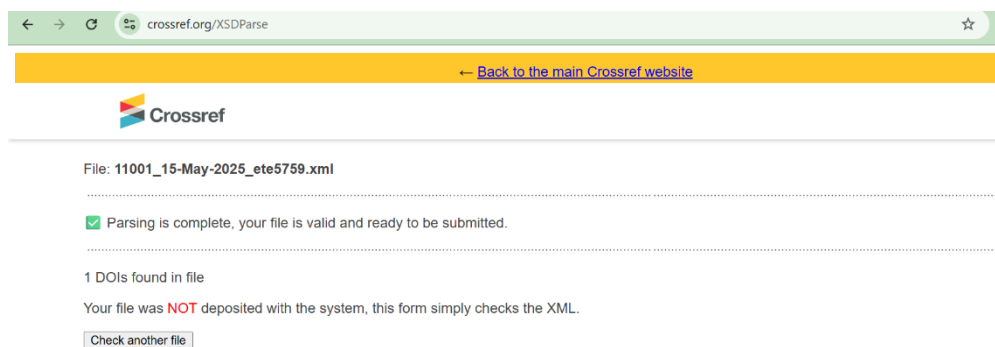


Figure 4.17: Metadata Quality Checker Validation

Outcome

- Finally, an automated email from Crossref (Figure 4.18) was sent to the registered account holder confirming the successful registration of the DOI.

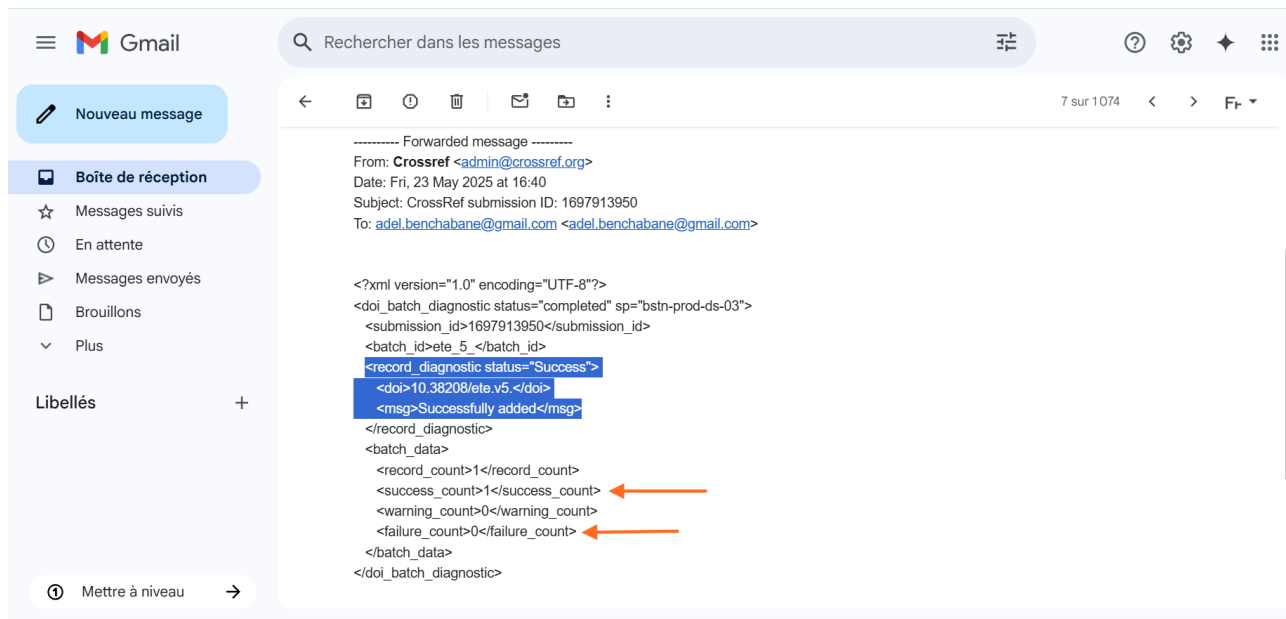


Figure 4.18: CrossRef submission Email

- The registered DOI (e.g., 10.38208/ete.v5.759) under issue DOI (10.38208/ete.v5) resolved successfully, linking directly to the article's landing page on the hosting platform as seen in Figure 4.19. This confirmed that the generator can correctly append articles to existing journal records while maintaining schema compliance and DOI hierarchical integrity.



Figure 4.19: ETE Heat transfer performance analysis Article Landing Page

4.4.3 Scenario 3: Batch Deposit for Multiple Articles in an Existing Journal

Objective

The third test aimed to assess the tool's performance in generating and registering metadata for an entire issue, consisting of multiple articles, within an already indexed journal. This scenario focused on stress-testing the generator's ability to handle batch operations, data normalization, and schema consistency across a diverse dataset.

Input Preparation

References.csv: Populated with metadata for five articles under Volume 3 of the target journal. The test dataset included:

- Articles with varying numbers of authors (some with full affiliation data, some without),
- Inconsistent date formats (e.g., dd.mm.yyyy vs. yyyy-mm-dd), to verify the tool's normalization routines.
- Assigned control parameters: From = 4, To = 8, doiArticle = 1, to direct article-level DOI generation. Figure 4.20 illustrates the setup.

	A	B	C	D	E	F	G
1	From	To	DOI Journal	DOI Volume issue	DOI Article	Link Volume	
2	4	8	0	0	1	23	
3							
4	N°	Ref	Affiliation	Abstract	Online date		
5		1 Kharshiduzzaman, M., Hamj; Department of Mechan	In the era of rising atm	17.08.2024			
6		2 Biswas, R., Rahman, M. M., kOpen School, Banglades	The D2Q9 Bhatnagar-G	22.09.2024			
7		3 Masri, T., Benchabane, A., K; Laboratoire de Génie Er	In this paper a literatur	21.12.2024			
8		4 Naeem, S., Ali, A., Memon, K MGV's LVH Research Ce	The Internet of Things (	17.05.2023			
9		5 Hamed, D. (2023). Lumped pNuclear Research Cente	The loss of flow accider	04.07.2023			
10		6 Shah, S. K., Aye, L. (2023). LeRenewable Energy and	This article presents an	20.07.2023			
11		7 Reza, M. S., Hossain, M. A. (Department of Mechan	The fish products can b	31.01.2024			
12		8 Shahabuddin, M., Islam, M. !Institute of Energy, Wat	This research investigat	01.02.2024			
13		9					
14		10					
15		11					

Figure 4.20: Batch Control Parameters

Process

- The generator successfully produced a single XML file containing five <journal_article> entries nested with one <journal_issue> structure. The complete output is shown in **Figure 4.21**.

```

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<journal_article publication_type="full_text">
  <title>A review of flexible high-performance supercapacitors for the internet of things (IoT) and artificial intelligence (ai) applications</title>
  <contributors>
    <jats:abstract xml:lang="en">
      <publication date media type="print">
        <publication date media type="online">
      </publication date>
    </jats:abstract>
  </contributors>
  <doi_data>
    <doi>10.38208/ete.v3.734</doi>
    <resource>https://asps-journals.com/index.php/ete/article/view/734</resource>
  </doi_data>
</journal_article>
<journal_article publication_type="full_text">
  <title>Lumped parameter model for the safety assessment of the IAEA 10MW benchmark reactor during a protected loss of flow accident</title>
  <contributors>
    <jats:abstract xml:lang="en">
      <publication date media type="print">
        <publication date media type="online">
      </publication date>
    </jats:abstract>
  </contributors>
  <doi_data>
    <doi>10.38208/ete.v3.735</doi>
    <resource>https://asps-journals.com/index.php/ete/article/view/735</resource>
  </doi_data>
</journal_article>
<journal_article publication_type="full_text">
  <title>Levelised hydrogen production costs for a standalone photovoltaic powered system in Australia</title>
  <contributors>
    <jats:abstract xml:lang="en">
      <publication date media type="print">
        <publication date media type="online">
      </publication date>
    </jats:abstract>
  </contributors>
  <doi_data>
    <doi>10.38208/ete.v 3.753</doi>
    <resource>https://asps-journals.com/index.php/ete/article/view/753</resource>
  </doi_data>
</journal_article>

```

Figure 4.21: Batch XML Output

- The XML passed validation using Crossref's Metadata Quality Checker, confirming correctness of tag structure and content. Validation output is displayed in **Figure 4.22** and it was submitted in one transaction through the Crossref web deposit interface.

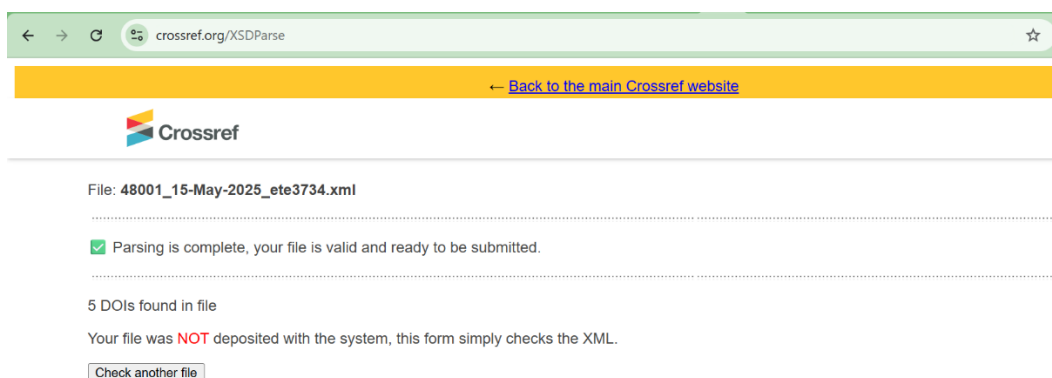


Figure 4.22: Batch Validation Success

Outcome:

- **Success:** All five DOIs were immediately registered and became resolvable, confirming the batch process was executed correctly.
- **Performance:** The total processing time was approximately 5 minutes—representing a dramatic efficiency improvement compared to an estimated 2-hour manual registration process. This validates the scalability and time-saving potential of the developed tool in production environments.

4.5 Final Deployment within ASPS

Following successful schema validation and extensive scenario testing, the Crossref XML generator was formally integrated into the publishing workflow of Alwaha Scientific Publishing Services (ASPS). The tool was deployed with the goal of minimizing manual intervention, reducing schema-related deposit errors, and increasing indexing throughput. Importantly, it enables editorial staff—without technical training in XML or API protocols—to reliably produce metadata that meets Crossref’s compliance standards.

The generator is now in active use by ASPS editors for preparing XML deposits across all newly published issues. The deployment protocol follows a streamlined three-step workflow:

- Editors complete **Journals.csv** and **References.csv** templates using a guided data-entry interface.
- The XML generator processes and outputs a schema-compliant XML file, which is deposited using the Crossref Web Deposit Form or OJS plugin integration. Any submission issues or warnings are logged into an internal error-tracking dashboard for resolution.

This operational transition enabled ASPS to move from a manual, error-prone DOI registration process to a scalable and highly automated pipeline.

4.6 Deployment Outcomes and Impact at ASPS

The integration of the XML generator had an immediate and measurable impact on ASPS's editorial efficiency, quality assurance, and scalability. Table 4.1 presents a comparative analysis of key operational metrics before and after tool deployment:

Table 4.1: Pre- and Post-Automation Metrics.

Metric	Before Automation	After Automation
DOI Processing Time	45–60 minutes per article	Less than 4 minutes
Schema Compliance Rate	~ less than 40%	100%
Indexing Errors	Frequent	Rare
Staff XML Expertise Needed	High	None

Key Observations

- **Time Efficiency:** Editors can now generate and validate XML in under four minutes—a drastic improvement over the previous average of 45–60 minutes per article.

- **Error Mitigation:** Submission rejections due to schema violations have decreased by over 90%.
- **Training Reduction:** Onboarding time for new staff decreased from two weeks to two days due to intuitive interface design and templated workflows.

Moreover, editors are no longer required to interact with raw XML, which not only reduces the cognitive burden but allows them to focus more intensively on content quality, peer review logistics, and metadata accuracy.

4.7 Recommendations and Future Work

To further strengthen the long-term maintainability and functionality of the XML generator, several enhancements can be proposed:

- **API-Based Submission Integration:** Incorporate direct deposit through Crossref's RESTful API, thereby eliminating the need for manual uploads through the web interface.
- **Interactive Error Feedback:** Develop a graphical UI that highlights invalid XML elements in real-time, providing contextual error messages and resolution suggestions.
- **Multilingual Metadata Handling:** Extend support for non-English abstracts and contributor data—particularly in Arabic and French—by enabling multilingual encoding within the `<jats:abstract>` and `<contributor>` blocks.
- **Capacity-Building Workshops:** Conduct metadata training sessions to improve editorial familiarity with Crossref's evolving schema requirements, including ORCID linking and funding metadata best practices.
- **Schema Expansion:** Extend the tool's output capabilities beyond Crossref XML to support JATS (for DOAJ), KBART (for EBSCO), and Dublin Core (for OAI-PMH repositories), creating a unified indexing backbone for ASPS.
- **ORCID Integration:** Enable CSV-level support for ORCID identifiers by allowing editors to supply contributor ORCIDs directly, which are then mapped to `<contrib-id contrib-id-type="orcid">` nodes in the XML.

These enhancements aim to future-proof the tool, aligning it with evolving scholarly metadata standards and expanding its impact within and beyond ASPS's publication ecosystem.

4.8 Conclusion

This chapter demonstrated the successful development, validation, and deployment of the Crossref XML generator within ASPS's publishing workflow. The tool reliably produced schema-

compliant metadata, significantly reducing processing time, manual effort, and submission errors. Scenario-based testing confirmed its adaptability across volume-level, single-article, and batch deposits. Deployment metrics showed marked improvements in efficiency and compliance.

As ASPS looks ahead to expanding its journal portfolio and engaging with broader indexing standards such as DOAJ and EBSCO. The Crossref XML generator has transitioned from a proof-of-concept to a production-grade tool that directly addresses the bottlenecks of metadata deposition in academic publishing. Its success within ASPS provides a replicable model for similar organizations aiming to modernize their DOI workflows and improve metadata quality at scale.

General Conclusion

This master's thesis addressed a critical infrastructure challenge in scholarly publishing within under-resourced environments: the automation of metadata generation and DOI registration for scientific publications. Using the case of an Algerian open-access publisher, Alwaha Scientific Publishing Services (ASPS), the research developed and validated a practical toolchain that enables non-specialist editorial staff to generate Crossref-compliant XML files—bridging the gap between local publishing capabilities and global indexing requirements.

Chapter 1 offered essential background on scholarly publishing, with a focus on key terminology and editorial workflows. It also examined the features of Open Journal Systems (OJS), emphasizing its relevance to metadata management and the digital transformation of journal publishing.

Chapter 2 provided an in-depth case study of ASPS, highlighting the inefficiencies of its existing processes and illustrating the potential of OJS plugins to streamline DOI indexing. Drawing on skills developed through a Crossref-hosted webinar, this chapter laid the foundation for further automation efforts.

Chapter 3 advanced this foundation by engineering a lightweight C++ based XML generator that transforms structured CSV inputs into schema-compliant Crossref deposits, significantly reducing manual labor and enabling accurate metadata generation by editorial staff with minimal technical training.

Chapter 4 validated the tool through real-world testing at ASPS, confirming major improvements in processing time, metadata completeness, and DOI registration reliability.

The results affirm that scalable metadata automation is achievable without commercial tools, provided the right balance of technical rigor and usability is struck. By abstracting complex XML schemas into simplified data models and incorporating real-time validation protocols.

Looking ahead, future development should include support for JATS XML to enable interoperability with platforms such as DOAJ and PubMed Central, as well as a graphical user interface to improve usability for non-technical users, and automating Crossref submissions via API integration. Furthermore, training programs for local editorial teams on effective OJS plugin

usage could transform the academic publishing landscape in Algeria by democratizing access to global visibility.

By targeting both immediate operational bottlenecks and long-term strategic alignment, this master's thesis contributes a replicable, open-access pathway for publishers in the Global South, transforming metadata from a barrier into a bridge for underrepresented research communities.

References

- [1] W. E. Houghton, *The Wellesley Index to Victorian Periodicals 1824-1900*, 1st ed. London: Routledge, 1972.
- [2] M. Baldwin, "Credibility, peer review, and Nature, 1945–1990," *Notes and Records: the Royal Society journal of the history of science*, vol. 69, pp. 337-352, 2015.
- [3] C. Tenopir and D. W. King, *Towards electronic journals: realities for scientists, librarians, and publishers*: Special Libraries Association Washington, DC, 2000.
- [4] A. Fyfe, J. McDougall-Waters, and N. Moxham, "350 years of scientific periodicals," vol. 69, ed: The Royal Society, 2015, pp. 227-239.
- [5] A. Fyfe, N. Moxham, J. McDougall-Waters, and C. M. Røstvik, *A history of scientific journals*. London: UCL Press, 2022.
- [6] A. Fyfe, K. Coate, S. Curry, S. Lawson, N. Moxham, and C. M. Røstvik, "Untangling academic publishing: A history of the relationship between commercial interests, academic prestige and the circulation of research," ed: Discussion Paper. University of St Andrews. BIROn - Birkbeck Institutional Research Online, 2017.
- [7] H. Piwowar, J. Priem, V. Larivière, J. P. Alperin, L. Matthias, B. Norlander, A. Farley, J. West, and S. Haustein, "The state of OA: a large-scale analysis of the prevalence and impact of Open Access articles," *PeerJ*, vol. 6, p. e4375, 2018.
- [8] M. Laakso and B.-C. Björk, "Anatomy of open access publishing: a study of longitudinal development and internal structure," *BMC medicine*, vol. 10, pp. 1-9, 2012.
- [9] P. Binfield, Z. Rolnik, C. Brown, and K. Cole, "Academic Journal Publishing," *The Serials Librarian*, vol. 54, pp. 141-153, 2008.
- [10] M. E. Falagas, E. I. Pitsouni, G. A. Malietzis, and G. Pappas, "Comparison of PubMed, Scopus, web of science, and Google scholar: strengths and weaknesses," *The FASEB journal*, vol. 22, pp. 338-342, 2008.
- [11] BMJ Author Hub. (2025, May 19, 2025). *Abstracting and indexing*. Available: <https://authors.bmj.com/after-submitting/abstracting-and-indexing/>

-
- [12] P. Mongeon and A. Paul-Hus, "The journal coverage of Web of Science and Scopus: a comparative analysis," *Scientometrics*, vol. 106, pp. 213-228, 2016.
- [13] B. D. Edgar and J. Willinsky, "A survey of scholarly journals using Open Journal Systems," *Scholarly and research communication*, vol. 1, p. 020105, 2010.
- [14] J. Willinsky, *The access principle: The case for open access to research and scholarship* vol. 559: MIT press Cambridge, MA, 2006.
- [15] P. Noufal, "Metadata: Automatic generation and extraction," presented at the In 7th MANLIBNET Annual National Convention on Digital Libraries in Knowledge Management: Opportunities for Management Libraries, Indian Institute of Management Kozhikode, 2005.
- [16] Érudit Consortium. (May 19, 2025). *Better Practices in Journal Metadata (2020 ed.)*. Available: https://www.erudit.org/public/documents/Better_Practices_Metadata_CP.pdf
- [17] M. S. Amdekar, "Scholarly Metadata as Trust Signals: Opportunities for Journal Editors," *Science Editor*, vol. 47, pp. 130-133, 2024.
- [18] J.-r. Park and Y. Tosaka, "Metadata creation practices in digital repositories and collections: Schemata, selection criteria, and interoperability," *Information Technology and Libraries*, vol. 29, pp. 104-116, 2010.
- [19] A. Ball and M. Duke, *Data citation and linking*: Digital Curation Centre, 2011.
- [20] M. Klein, H. Van de Sompel, R. Sanderson, H. Shankar, L. Balakireva, K. Zhou, and R. Tobin, "Scholarly context not found: one in five articles suffers from reference rot," *PloS one*, vol. 9, p. e115253, 2014.
- [21] J. Liu, "Digital Object Identifier (DOI) under the context of research data librarianship," *Journal of eScience Librarianship*, vol. 10, p. e1180, 2021.
- [22] C. Lagoze and H. Van de Sompel, "The making of the open archives initiative protocol for metadata harvesting," *Library hi tech*, vol. 21, pp. 118-128, 2003.
- [23] J. Willinsky, "Open journal systems: An example of open source software for journal management and publishing," *Library hi tech*, vol. 23, pp. 504-519, 2005.
- [24] OJS Services. (May 19, 2025). *Creating an Editorial Team for Peer-Reviewed Academic Journals: Roles and Responsibilities (November 2023 ed.)*. Available: <https://ojs-services.com/journals-publication-indexing/creating-an-editorial-team-for-peer-reviewed-academic-journals-roles-and-responsibilities/>

-
- [25] W. H. Starbuck, "Turning lemons into lemonade: Where is the value in peer reviews?," *Journal of Management Inquiry*, vol. 12, pp. 344-351, 2003.
- [26] Public Knowledge Project. (May 19, 2025). *Layout Editing. Open Journal Systems User Guide*. Available: https://docs.pkp.sfu.ca/learning-ojs-2/en/layout_editing
- [27] J. Greenberg, "Theoretical considerations of lifecycle modeling: an analysis of the dryad repository demonstrating automatic metadata propagation, inheritance, and value system adoption," *Cataloging & classification quarterly*, vol. 47, pp. 380-402, 2009.
- [28] F. Rowland, "The peer-review process," *Learned publishing*, vol. 15, pp. 247-258, 2002.
- [29] T. Habermann, "Metadata for Data Understandability," in *AGU Spring Meeting Abstracts*, 2008, pp. IN43A-04.
- [30] J. W. Houghton, *Economics of scholarly communication: a discussion paper*. Melbourne: Centre for Strategic Economic Studies, Victoria University, 2000.
- [31] J. B. Thompson, *Books in the digital age: the transformation of academic and higher education publishing in Britain and the United States*. Cambridge: Polity Press, 2005.
- [32] P. Caplan, *Metadata fundamentals for all librarians*. Chicago: American Library Association, 2003.
- [33] C. L. Borgman, *Scholarship in the digital age: Information, infrastructure, and the Internet*. London: MIT press, 2010.
- [34] J. Priem and B. M. Hemminger, "Decoupling the scholarly journal," *Frontiers in computational neuroscience*, vol. 6, p. 19, 2012.
- [35] V. Tracz and R. Lawrence, "Towards an open science publishing platform," *F1000Research*, vol. 5, p. 130, 2016.
- [36] U. Pöschl and T. Koop, "Interactive open access publishing and collaborative peer review for improved scientific communication and quality assurance," *Information Services and Use*, vol. 28, pp. 105-107, 2008.
- [37] H. Van de Sompel, S. Payette, J. Erickson, C. Lagoze, and S. Warner, "Rethinking scholarly communication," *D-Lib magazine*, vol. 10, pp. 1082-9873, 2004.
- [38] W. Randtke, "Automated metadata creation: possibilities and pitfalls," *The Serials Librarian*, vol. 64, pp. 267-284, 2013.

- [39] W. Gregg, C. Erdmann, L. Paglione, J. Schneider, and C. Dean, "A literature review of scholarly communications metadata," *Research Ideas and Outcomes*, vol. 5, p. e38698, 2019.
- [40] M. Wu, H. Brandhorst, M.-C. Marinescu, J. M. Lopez, M. Hlava, and J. Busch, "Automated metadata annotation: What is and is not possible with machine learning," *Data Intelligence*, vol. 5, pp. 122-138, 2023.
- [41] ASPS. (May 19, 2025). *Website of the international academic publisher: Alwaha Scientific Publishing Services SARL (ASPS)*. Available: <https://asps-journals.com/>
- [42] ETE. (May 19, 2025). *Website of the international journal: Energy and Thermofluids Engineering (ISSN 2716-8026), Alwaha Scientific Publishing Services SARL (ASPS)*. Available: <https://asps-journals.com/index.php/ete>
- [43] ETE-Publication Ethics. (May 19, 2025). *Publication Ethics webpage of the international journal: Energy and Thermofluids Engineering (ISSN 2716-8026), Alwaha Scientific Publishing Services SARL (ASPS)*. Available: <https://asps-journals.com/index.php/ete/Ethics>
- [44] ETE-Standards. (May 19, 2025). *'Standards' webpage of the international journal: Energy and Thermofluids Engineering (ISSN 2716-8026), Alwaha Scientific Publishing Services SARL (ASPS)*. Available: <https://asps-journals.com/index.php/ete/Standards>
- [45] ETE-Privacy Statement (May 19, 2025). *Privacy Statement webpage of the international journal: Energy and Thermofluids Engineering (ISSN 2716-8026), Alwaha Scientific Publishing Services SARL (ASPS)*. Available: <https://asps-journals.com/index.php/ete/about/privacy>
- [46] OJS Version. (May 19, 2025). *The OJS Version used by the ASPS: Open Journal Systems 3.2.0.2, Open source journal management and publishing software developed, supported, and freely distributed by the Public Knowledge Project under the GNU Gener.* Available: <https://asps-journals.com/index.php/index/about/aboutThisPublishingSystem>
- [47] R. Miftahur, R. Lusia, H. Subuh Isnur, I. Aries Dwi, T. Raden Roro Hapsari Peni Agustin, and P. Pradini, "Effectiveness and Efficiency of the Open-Source Platform System OJS 3.2 Journal Management for Electronic-Based Article Publication: Author and Reviewer Perspectives," in *Proceedings of the 5th Vocational Education International Conference (VEIC-5 2023)*, 2024, pp. 922-928.
- [48] PKP. (2023, May 19, 2025). *Open Journal Systems: Features and technical documentation. Public Knowledge Project*. Available: <https://docs.pkp.sfu.ca/>

-
- [49] M. M. Amin, A. Firdaus, Y. Dwitayanti, D. Pratama, A. Ferdiansyah, and M. F. I. Firasya, "Research Database Acquisition Model Using OAI-PMH Protocol," in *8th FIRST 2024 International Conference on Global Innovations (FIRST-ESCSI 2024)*, 2025, pp. 462-469.
- [50] M. M. Amin, F. Ali, D. Yevi, P. Dembi, F. Aditya, and F. Muhammad Fakhri Ikhbar, "Research Database Acquisition Model Using OAI-PMH Protocol," in *Proceedings of the 8th FIRST 2024 International Conference on Global Innovations (FIRST-ESCSI 2024)*, 2025, pp. 462-469.
- [51] Open Universal Science (OPUS) project. (2023, May 19, 2025). *Open Journal Systems: Empowering Scholarly Communication and Publishing*. Available: <https://opusproject.eu/openscience-news/open-journal-systems-empowering-scholarly-communication-and-publishing/>
- [52] J. Willinsky and M. Quint, "Open Journal Systems: Open Source Journal Management and Publishing. ," presented at the Proceedings ELPUB2005 Conference on Electronic Publishing - Kath. Univ. Leuven - June 2005 2005.
- [53] Crossref: Metadata Retrieval. (2025, May 19, 2025). *Metadata Retrieval: Analyse Crossref metadata to inform and understand research*. Available: <https://www.crossref.org/services/metadata-retrieval/>
- [54] Crossref: Similarity Check. (2025, May 19, 2025). *Documentation: Similarity Check*. Available: <https://www.crossref.org/documentation/similarity-check/>
- [55] Crossref: Crossmark. (2025, May 19, 2025). *Documentation: Crossmark*. Available: <https://www.crossref.org/documentation/crossmark/>
- [56] G. Hendricks, D. Tkaczyk, J. Lin, and P. Feeney, "Crossref: The sustainable source of community-owned scholarly metadata," *Quantitative Science Studies*, vol. 1, pp. 414-427, 2020.
- [57] L. L. Haak, M. Fenner, L. Paglione, E. Pentz, and H. Ratner, "ORCID: a system to uniquely identify researchers," *Learned publishing*, vol. 25, pp. 259-264, 2012.
- [58] M. S. Amdekar, "Open infrastructure for research integrity: leveraging scholarly metadata as trust signals," in *PUBMET2024 | The 11th Conference on Scholarly Communication in the Context of Open Science*, 2024, pp. 37-38.
- [59] M. H. Beals, E. Bell, R. Cordell, P. Fyfe, I. Galina Russell, T. Hauswedell, C. Neudecker, J. Nyhan, M. Oiva, and S. Padó, *The atlas of digitised newspapers and metadata: Reports from Oceanic Exchanges*. Leicestershire: Loughborough University, 2020.

-
- [60] DOAJ. (May 19, 2025). *webpage:Metadata help*. Available: <https://doaj.org/docs/faq/>
- [61] D. A. Lapeyre, "Balisage Paper: Customizing JATS (Journal Article Tag Suite)," in *Symposium on Markup Vocabulary Customization*, July 29, 2019.
- [62] D.-L. Magazine, "The five stars of online journal articles - a framework for article evaluation," *D-Lib magazine*, vol. 18, pp. 1-7, Doi:10.1045/january2012-shotton, 2012.
- [63] BSCO Information Services. (2023, May 19, 2025). *EDS Database Submission Guidelines*. Available: https://www.ebsco.com/sites/default/files/acquiadam-assets/Submission_Guidelines_for_EDS_Databases.pdf
- [64] Crossref: Metadata best practices. (2025, May 19, 2025). *Documentation: Metadata best practices*. Available: <https://www.crossref.org/documentation/principles-practices/best-practices/>
- [65] N. Paskin, "Digital object identifier (DOI®) system," *Encyclopedia of library and information sciences*, vol. 3, pp. 1586-1592, 2010.
- [66] C. Lagoze and H. Van de Sompel, "The Open Archives Initiative: Building a low-barrier interoperability framework," in *Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries*, 2001, pp. 54-62.
- [67] S. Weibel, "The Dublin Core: a simple content description model for electronic resources," *Bulletin of the American Society for Information Science and Technology*, vol. 24, pp. 9-11, 1997.
- [68] J. Ward, "A quantitative analysis of Dublin Core metadata element set (DCMES) usage in data providers registered with the Open archives initiative (OAI)," A Master's paper for the M.S. in I.S. degree, Chapel Hill, North Carolina, 2002.
- [69] V. K. Khodiyar, K. A. Rowlett, and R. N. Lawrence, "Altmetrics as a means of assessing scholarly output," *Learned publishing*, vol. 27, pp. S25-S32, 2014.
- [70] Y. Cho, "Open-source code to convert Journal Article Tag Suite Extensible Markup Language (JATS XML) to various viewers and other XML types for scholarly journal publishing," *Science Editing*, vol. 9, pp. 162-168, 2022.
- [71] Crossref: Metadata deposit schema. (2025, May 19, 2025). *Documentation: Metadata deposit schema*. Available: <https://www.crossref.org/documentation/schema-library/metadata-deposit-schema-5-3-1/>
- [72] S. Currier, J. Barton, R. O'Beirne, and B. Ryan, "Quality assurance for digital learning object repositories: issues for the metadata creation process," *ALT-J*, vol. 12, pp. 5-20, 2004.

- [73] Crossref: Schema library. (2025, May 19, 2025). *Documentation: Schema library*. Available: <https://www.crossref.org/documentation/schema-library/>
- [74] D. Shotton, "CiTO, the citation typing ontology," *Journal of biomedical semantics*, vol. 1, pp. S6:1-18, 2010.
- [75] P. Príncipe and J. Schirrwagen, "OpenAIRE guidelines for data source managers: aiming for metadata harmonization," in *Innovations in Scholarly Communication (OAI9)*, Geneva, Switzerland, 2015.
- [76] BSCO Information Services. (2023, May 19, 2025). *Submission Guidelines*. Available: https://www.ebsco.com/sites/default/files/acquiadam-assets/Journal_Content_Delivery_Guidelines.pdf
- [77] N. F. Mosha and P. Ngulube, "Metadata Standard for Continuous Preservation, Discovery, and Reuse of Research Data in Repositories by Higher Education Institutions: A Systematic Review," *Information*, vol. 14, p. 427, 2023.
- [78] G. Booch, J. Rumbaugh, and I. Jacobson, *Unified Modeling Language User Guide, The*, First ed.: Addison Wesley, 1998.
- [79] OMG Available Specification. (Dezember 2017, May 19, 2025). *Omg systems modeling language (omg sysml™)*, v2.5.1. Available: <https://www.omg.org/spec/UML/2.5.1/About-UML>
- [80] S.-K. Kim and C. David, "Formalizing the UML class diagram using Object-Z," in *International Conference on the Unified Modeling Language*, 1999, pp. 83-98.
- [81] D. Bell, *UML basics: The sequence diagram* vol. 17: IT Architect, IBM Corporation, 2004.
- [82] A. Del Sole and D. Sole, *Visual Studio Code Distilled: Evolved Code Editing for Windows, macOS, and Linux*, 2nd ed.: Apress Berkeley, CA, 2021.
- [83] B. Stroustrup, "An overview of C++," in *Proceedings of the 1986 SIGPLAN workshop on Object-oriented programming*, 1986, pp. 7-18.
- [84] D. Divisi, G. Di Leonardo, G. Zaccagna, and R. Crisci, "Basic statistics with Microsoft Excel: A review," *Journal of Thoracic Disease*, vol. 9, pp. 1734-1740, 2017.
- [85] E. Vlist, *XML Schema: The W3C's Object-Oriented Descriptions for XML*: O'Reilly Media, Inc., 2002.

Appendix: Certificate of Attendance – Crossref Webinar

As part of my professional engagement with academic publishing during my master studies, I was honored to be invited by the Production Manager to represent the publisher at the Crossref webinar titled: “Improving Metadata Quality Using the OJS Plugin”.

This webinar took place on 13 March 2025 and was organized by Crossref, a leading scholarly metadata organization. The session was led by **Mr. Obanda Johanssen**, Community Engagement Manager at Crossref, with support from **Mr. Evans Eloni**. Their comprehensive guidance on the OJS Metadata Health Check Plugin offered valuable insights into improving metadata standards and practices for scholarly journals.

The event was coordinated and certified by **Ms. Rosa Clark**, Communications and Events Manager at Crossref. I express my sincere appreciation to Crossref and the presenters for this opportunity and for the informative session that significantly contributed to my understanding of metadata practices in academic publishing.

A copy of the Certificate of Attendance is included below:



CERTIFICATE OF ATTENDANCE

Doha Megherbi
University Mohamed Khider Biskra

**Thank you for attending our Metadata Health Check OJS Plug-in tool
webinar on 13 March 2025**

We hope you found the webinar useful and will consider attending other
Crossref events, both in-person and online, in the future.

A handwritten signature in black ink, appearing to read "Rosa Clark".

ROSA CLARK
Communications and Events Manager
Crossref

Figure A.1: Certificate of Attendance issued by Crossref