Mohamed Khider University of Biskra
Faculty of Exact Sciences
Department of Computer Science

MASTER THESIS

Speciality : Master - Artificial Intelligence - Ref : P1379/2026

TITLE

Design and Development of a Domain-Specific Language Model as a Private Tutor Using LLM Adaptation Techniques

Prepared and defended by

Belkacem achraf ZAGHEZ

Defense date : 07/06/2026

DEFENSE COMMITTEE

Full Name	Grade	Role
Samir BOUREKKACHE	Prof	**President**
Laid KAHLOUL	Prof	**Supervisor**
Farah FEKRAOUI	MCB	**Examiner**
Mohamed DJELLAB	Prof	**Incubator Representative**

Academic Year : 2025/2026

Acknowledgment

"Gratitude is not only the greatest of virtues, but the parent of all others."

First and foremost, I would like to express my deepest gratitude to **Almighty God** for granting me the strength, patience, and knowledge to complete this work.

I would like to sincerely thank my supervisor, **Prof. Laid Kahloul**, for his invaluable guidance, continuous support, and insightful advice throughout the development of this project. His expertise and encouragement were essential in shaping this work.

I also extend my sincere appreciation to my co-supervisor, **Dr. Imane Youkana**, for her support, constructive feedback, and helpful suggestions, which greatly contributed to the improvement of this project.

I would like to warmly thank **Prof. Rachida Saouli** for her generous assistance and guidance during the project. Her support was extremely helpful and greatly appreciated.

Finally, I would like to express my heartfelt thanks to my **beloved mother** for her endless love, sacrifices, and unwavering support. Her encouragement has always been my greatest motivation.

I also extend my deepest gratitude to my entire family for their constant love, continuous backing, and presence throughout my life. My sincere appreciation goes to all the teachers who have taught, guided, and inspired me across every stage of my academic journey, helping shape the person I am today. Lastly, I deeply dedicate this work to the blessed memory of my late grandparents, who are no longer with us, but whose enduring light paved the way and guided my steps to reach this milestone.

Belkacem Achraf ZAGHEZ
May 2026

Abstract

This project presents **DeepBac**, an advanced intelligent tutoring platform engineered specifically for Algerian Baccalaureate students. General-purpose Large Language Models (LLMs) consistently struggle with localized educational contexts due to factual hallucinations, a lack of strict curriculum alignment, and the inability to process complex trilingual code-switching (Modern Standard Arabic, French, and Darja). To address these critical gaps, this work proposes a hybrid AI architecture that integrates a robust Retrieval-Augmented Generation (RAG) pipeline with a Deep Knowledge Tracing (DKT) module. The system utilizes a dual-database paradigm employing MySQL for relational metadata and Qdrant for dense semantic vector embeddings ensuring that all generated pedagogical responses are strictly grounded in the official national curriculum. Furthermore, an LSTM based neural network continuously tracks and predicts student mastery levels, dynamically steering the LLM to provide adaptive scaffolding and personalized difficulty adjustments. Empirical evaluations using an automated LLM-as-a-Judge framework demonstrated that the RAG enabled architecture achieved a near perfect factual faithfulness score of 9.89/10, effectively eliminating out-of-domain hallucinations. Additionally, the predictive DKT module recorded an R^2 of 0.9116, confirming its high precision in cognitive modeling. Ultimately, beta deployment results indicated overwhelming user acceptance, proving that domain specific LLM adaptation can successfully and safely democratize personalized exam preparation.

Keywords: Large Language Models (LLM), Retrieval-Augmented Generation (RAG), Deep Knowledge Tracing (DKT), Code-Switching, Algerian Baccalaureate, Adaptive Learning.

Résumé

Ce projet présente **DeepBac**, une plateforme de tutorat intelligent avancée, conçue spécifiquement pour les étudiants du Baccalauréat algérien. Les grands modèles de langage (LLMs) généralistes rencontrent des difficultés dans les contextes éducatifs locaux en raison des hallucinations factuelles, du manque d'alignement avec les programmes stricts et de l'incapacité à traiter l'alternance codique trilingue (Arabe standard, Français et Darja). Pour combler ces lacunes, ce travail propose une architecture d'IA hybride qui intègre un pipeline robuste de Génération Augmentée par la Recherche (RAG) avec un module de Suivi Approfondi des Connaissances (DKT). Le système utilise un paradigme à double base de données employant MySQL pour les métadonnées relationnelles et Qdrant pour les plongements vectoriels sémantiques garantissant que toutes les réponses générées sont strictement ancrées dans le programme national officiel. De plus, un réseau de neurones basé sur LSTM suit et prédit en permanence le niveau de maîtrise de l'étudiant, orientant dynamiquement le LLM pour fournir un étayage pédagogique et des ajustements de difficulté personnalisés. Les évaluations empiriques utilisant un cadre automatisé LLM-as-a-Judge ont démontré que l'architecture RAG a atteint un score de fidélité factuelle presque parfait de 9.89/10, éliminant efficacement les hallucinations hors domaine. En outre, le module prédictif DKT a enregistré un R^2 de 0.9116, confirmant sa haute précision dans la modélisation cognitive. Enfin, les résultats du déploiement bêta ont indiqué une acceptation écrasante de la part des utilisateurs, prouvant que l'adaptation de LLMs spécifiques à un domaine peut démocratiser avec succès et en toute sécurité la préparation personnalisée aux examens.

Mots-clés : Grands Modèles de Langage (LLM), Génération Augmentée par la Recherche (RAG), Suivi Approfondi des Connaissances (DKT), Alternance Codique, Baccalauréat Algérien, Apprentissage Adaptatif.

الملخص

تقدم هذه المذكرة"DeepBac" ، وهي منصة تعليمية ذكية ومتقدمة تم تصميمها خصيصاً لطلاب البكالوريا في الجزائر. تواجه النماذج اللغوية الكبيرة (LLMs) العامة صعوبات بالغة في السياقات التعليمية المحلية بسبب الهلوسة المعرفية، وغياب التوافق مع المناهج الرسمية، والعجز عن معالجة الخلط اللغوي الثلاثي المعقد (العربية الفصحى، الفرنسية، والدارجة). لحل هذه الفجوات، يقترح هذا العمل معمارية هجينة للذكاء الاصطناعي تدمج بين تقنية التوليد المعزز بالاسترجاع (RAG) ووحدة التتبع العميق للمعرفة .(DKT) يعتمد النظام على بنية قواعد بيانات مزدوجة—تستخدم MySQLللبيانات العلائقية و Qdrantلتضمين المتجهات الدلالية—مما يضمن استناد جميع الإجابات المولدة حصرياً إلى المنهج الوطني الرسمي. علاوة على ذلك، تقوم شبكة عصبية من نوع (LSTM)بتتبع مستوى استيعاب الطالب والتنبؤ به بشكل مستمر، مما يوجه النموذج اللغوي ديناميكياً لتقديم تلميحات مخصصة وتعديل مستوى الصعوبة .أثبتت التقييمات التجريبية باستخدام إطار التقييم الآلي (LLM-as-a-Judge) أن المعمارية المعتمدة على RAG حققت درجة موثوقية شبه مثالية بلغت 10/9.89، مما أدى إلى القضاء الفعال على الهلوسة العلمية. بالإضافة إلى ذلك، سجلت وحدة DKT التنبؤية معامل تحديد R^2 قدره 0.9116، مما يؤكد دقتها العالية في النمذجة المعرفية. وأخيراً، أظهرت نتائج الإطلاق التجريبي (Beta)قبولاً واسعاً من المستخدمين، مما يثبت أن تكييف النماذج اللغوية المتخصصة يمكن أن يساهم بنجاح وأمان في إتاحة التحضير المخصص للامتحانات للجميع .

الكلمات المفتاحية: النماذج اللغوية الكبيرة(LLM) ، التوليد المعزز بالاسترجاع(RAG) ، التتبع العميق للمعرفة(DKT) ، الخلط اللغوي، البكالوريا الجزائرية، التعلم التكيفي.

Contents

Abstract **2**

Résumé **3**

1 Introduction **11**
1.1 Context and Motivation 11
1.2 Problem Statement 12
1.3 Objectives 12
1.4 Research Questions 13
1.5 Organization of the dissertation 15

2 AI in Education **17**
2.1 Introduction 17
2.2 Intelligent Tutoring Systems (ITS) 17
2.3 Large Language Models (LLMs) 18
2.4 The Convergence of LLMs and ITS 19
2.5 Adaptive Learning Systems 21
2.6 LLM Adaptation Techniques 22
2.6.1 Prompt Engineering 22
2.6.2 Retrieval-Augmented Generation (RAG) 23
2.6.3 Fine-Tuning 23
2.7 Multilingual NLP and Code-Switching 24
2.8 Related Work 26
2.8.1 LLM Adaptation for Localized and Code-Switched Education . . . 26
2.8.2 Curriculum-Grounded RAG in High-Stakes Assessments 26
2.8.3 Fusing Deep Knowledge Tracing with Generative AI 27
2.9 Synthesis: Research Gaps and Proposed Solution 27
2.9.1 Identified Research Gaps 28
2.9.2 Our Proposed Solution 28
2.10 Conclusion 29

3 System Design and Architecture **30**
3.1 Introduction 30

3.2 System Architecture Overview 30
3.3 Use Case Diagrams and Functional Requirements 32
3.3.1 Student Use Cases 32
3.3.2 Pedagogical Expert (Teacher) Use Cases 33
3.3.3 Admin Use Cases 34
3.3.4 Sequence Diagrams 35
3.4 Dataset Construction and Processing 42
3.4.1 Data Sources and Collection 42
3.4.2 Advanced Data Extraction and Structuring 43
3.4.3 Semantic Chunking and Vectorization 43
3.4.4 Dual-Database Architecture for Retrieval 44
3.4.5 Synthetic Data Generation for Deep Knowledge Tracing (DKT) . . 45
3.5 LLM Core Design and Evolution 47
3.5.1 Phase 1: Supervised Fine-Tuning (SFT) and QLoRA Mechanics . . 47
3.5.2 Phase 2: Production-Ready RAG via API (Current Live Version) . 50
3.5.3 Phase 3: The Hybrid Architecture (RAG + Fine-Tuning) 50
3.5.4 Universal Layer: Advanced Prompt Engineering 50
3.6 Adaptive Learning Module 50
3.6.1 Continuous Deep Knowledge Tracing (DKT) via LSTM 51
3.6.2 DKT-LLM Integration Cycle (Dynamic Adaptation) 51
3.7 Native Multilingual and Code-Switching Handling 52
3.8 Security and Data Privacy Considerations 53
3.8.1 Data Privacy and Legal Compliance 54
3.8.2 Authentication and Access Control 54
3.8.3 AI-Specific Security and Adversarial Robustness 54
3.9 Conclusion 55

4 Implementation **56**
4.1 Introduction 56
4.2 Development Methodology: An Agile Approach 56
4.2.1 The Iterative Development Cycle (Sprints) 57
4.3 Development and Production Environment 58
4.3.1 Prototyping and Local Setup 58
4.3.2 Production Server and Backend Logic 58
4.3.3 The Hybrid AI Tech Stack 59
4.4 System Implementation 60
4.4.1 Implementing the Dual-Database RAG Pipeline 60
4.4.2 API Orchestration and Prompt Assembly 61
4.4.3 Coding the Adaptive Learning Logic (DKT Integration) 61

4.5 User Interface and Frontend Implementation . . . 62
4.5.1 Building the React-Based Architecture . . . 62
4.5.2 The Interactive Chat Interface and LaTeX Rendering . . . 62
4.5.3 Integrating the Adaptive Feedback UI . . . 63
4.6 Database Implementation and API Architecture . . . 63
4.6.1 Hybrid Data Modeling: MySQL and Qdrant . . . 63
4.6.2 RESTful API Orchestration . . . 64
4.7 Conclusion . . . 65

5 Evaluation, Results, and Discussion 66
5.1 Introduction . . . 66
5.2 Multi-Dimensional Evaluation Metrics (SFT Phase) . . . 66
5.2.1 Training and Validation Performance (Loss Metrics) . . . 67
5.2.2 Classical Lexical Metrics (BLEU and ROUGE-L) . . . 67
5.2.3 Contextual Similarity Metric (BERTScore) . . . 68
5.2.4 Semantic Evaluation via LLM-as-a-Judge (DeepSeek-V4-Pro) . . . 69
5.3 Evaluating the Retrieval-Augmented Generation (RAG) Architecture . . . 70
5.3.1 Pedagogical Case Studies: With and Without RAG . . . 71
5.4 Evaluation of the Adaptive Learning Module (DKT) . . . 72
5.4.1 LSTM Predictive Accuracy (Using Synthetic Data) . . . 72
5.4.2 Dynamic UI Adaptation and Pedagogical Scaffolding . . . 73
5.5 Beta Deployment and User Experience (UX) Evaluation . . . 73
5.5.1 Beta Testing Methodology and Demographics . . . 74
5.5.2 Analysis of Survey Results (Usability and Satisfaction) . . . 77
5.5.3 Qualitative Feedback and Identified Limitations . . . 78
5.6 Discussion . . . 78
5.6.1 The Indispensability of RAG for Hallucination Mitigation . . . 78
5.6.2 The Complementary Roles of Fine-Tuning and RAG . . . 79
5.6.3 Predictive Power of the DKT Module . . . 79
5.6.4 User Acceptance and Pedagogical Value . . . 79
5.6.5 Limitations and Future Directions . . . 80
5.6.6 Summary of Key Insights . . . 80
5.7 Conclusion . . . 80

General Conclusion 82

Bibliography 84

A Student Survey Questionnaire 89

List of Figures

2.1 Architecture of a Classic Intelligent Tutoring System (ITS) 18
2.2 The continuous feedback cycle of an Adaptive Learning System, illustrating the dynamic interaction between data collection, student modeling, and personalized content delivery. 22
2.3 The cross-lingual semantic mapping process, illustrating how diverse linguistic inputs are unified into a single abstract pedagogical concept before response generation. 25

3.1 High-Level Architecture of the DeepBac System 31
3.2 Use Case Diagram for the DeepBac System, illustrating the distinct interactions of the Student, Teacher, and Admin within the platform. 35
3.3 Sequence diagram illustrating the Retrieval-Augmented Generation (RAG) workflow when handling a student's pedagogical question. 37
3.4 Sequence diagram illustrating the adaptive generation of BAC exercises based on student proficiency and curriculum templates. 39
3.5 Sequence diagram detailing the automated evaluation process, highlighting the comparative analysis by the LLM and the continuous updating of the student's adaptive profile. 41
3.6 DeepBac sequence diagram: Node.js orchestration across Cohere, Qdrant, MySQL, and DeepSeek. 42
3.7 DeepBac data ingestion and preprocessing pipeline, illustrating Vision LLM extraction, LaTeX structuring, and dual-database storage (MySQL and Qdrant). 45
3.8 The architectural mechanism of Low-Rank Adaptation (LoRA). The pretrained weights remain frozen, while only the low-rank matrices A and B are updated. Adapted from [27]. 48
3.9 Comparison of fine-tuning methods: LoRA (16-bit base) versus QLoRA (4-bit NF4 base with 16-bit bf16 computation). Adapted from [51]. 49
3.10 Multilingual handling architectures: Latency-heavy classic approach (translation middleware) versus DeepBac native approach (shared semantic vector space). 53

4.1 DeepBac Agile development lifecycle, illustrating the four iterative sprints: system foundation, dual-database RAG integration, adaptive personalization, and pedagogical optimization. 58
4.2 DeepBac deployment blueprint: End-to-end infrastructure flow from local development to Hostinger production server and cloud AI services. 60
4.3 DeepBac MySQL database relational schema. 64

5.1 DeepBac administrator dashboard. 74
5.2 The public homepage of the DeepBac web application. 75
5.3 DeepBac student dashboard. 75
5.4 The DeepBac conversational AI assistant interface. 76
5.5 The interactive smart review session interface. 76
5.6 DeepBac student profile and progress tracking interface. 77

List of Tables

2.1 Comparative analysis between Traditional ITS and LLM-Augmented ITS architectures. 20
2.2 Detailed comparison of LLM Adaptation Techniques highlighting structural parameter modifications, cost efficiency, and optimal operational use cases. 24

5.1 Training and Evaluation Loss across Fine-Tuned Models 67
5.2 Lexical Overlap Evaluation (BLEU and ROUGE-L) 68
5.3 Semantic Similarity Evaluation using BERTScore 68
5.4 LLM-as-a-Judge Pedagogical Evaluation (Out of 10) 69
5.5 Quantitative Impact of RAG Integration on Factual Integrity (LLM-as-a-Judge Evaluation) 71
5.6 LSTM Model Performance Metrics on the DKT Testing Set 72

Chapter 1

Introduction

1.1 Context and Motivation

Following the profound breakthroughs recently achieved in Artificial Intelligence particularly with Large Language Models (LLMs) that have demonstrated systematic reliability across various tasks despite ongoing controversies surrounding their outputs [1]. There has been a global paradigm shift toward integrating these technologies across diverse sectors. Education is considered one of the most important fields. And large language models are the key to turning traditional education systems into smart, interactive learning environments[2].

But even with this global movement, not every country has benefited. Developing nations like Algeria, for instance, continue to struggle with major, deeply rooted issues in their education systems. Take the Baccalaureate (BAC) level, for instance. The curriculum is packed, teaching methods are mostly conventional, and there's hardly any room for individual- support. Private tutoring (often called "shadow education") is a common workaround, but it's limited by both time and high costs [3].

This heavy reliance on private lessons widens the gap in educational equality. Not everyone can afford quality one-on-one help, so academic results end up being uneven. In this context, AI especially LLMs offers a practical, promising alternative. Since these models understand natural language, break down difficult topics, and hold conversations naturally, they could really help make learning more accessible for everyone. They might even tackle what's known as the "two-sigma problem" in personalized tutoring [4].

If you adapt them carefully, an LLM can work like a dedicated smart tutor. It can answer questions, explain tough concepts using the official syllabus, create practice exercises, and give helpful, pedagogical feedback on student answers. The key difference in our work is that **DeepBac** is designed specifically around the Algerian Baccalaureate curriculum. Other systems are made for general-purposes and they simply cannot adapt to the particular characteristics of Algeria's syllabus.

1.2 Problem Statement

Even though large language models have come a long way and can now understand natural language well enough to rival human conversation, most of them are still just general-purpose tools [5]. Thus, the main issue is their failure to grasp both the specific context of Algeria's Baccalaureate and the required answering methodology. This makes it hard to rely on current AI models as private tutors or to trust what they produce.

Specifically, general-purpose LLMs frequently encounter difficulties in the following areas:

- **Curricular Misalignment:** Generating responses that fail to strictly adhere to the nuances and specific requirements of the Algerian national syllabus.
- **Pedagogical Deficiency:** A lack of instructional flow and structured guidance tailored to high-school learners' needs.
- **Methodological Discrepancy:** Non-adherence to the official BAC answering methodologies and the formal assessment criteria established by the Algerian Ministry of Education.
- **Static Interaction:** The inability to adapt content dynamically to match the varying proficiency levels and individual-progress of each student [4].

Furthermore, because these models were primarily trained on vast, non-specialized datasets, they often rely on broad web-based information. This can lead to "hallucinations" or the provision of contextually inaccurate data from unreliable sources, which undermines the credibility of the tutoring process [6, 7]. Consequently, students cannot fully depend on these platforms as comprehensive tools for rigorous exam preparation [1].

Therefore, the core challenge addressed in this thesis is: **How to design and develop a domain-specific language model as a personalized private tutor using LLM adaptation techniques?** This work aims to solve this problem by building an intelligent, interactive platform. It would offer a distinctive learning experience that fits the Algerian Baccalaureate curriculum.

1.3 Objectives

The primary goal of this thesis is to bridge the gap between general-purpose AI and localized educational needs by adapting Large Language Models (LLMs) to serve as specialized intelligent tutors. This research focuses on developing a comprehensive, smart educational framework that is strictly tailored to the Algerian secondary education context. To achieve this, the following specific objectives have been established:

1. **Domain-Specific Adaptation of Large Language Models:** The first objective involves fine-tuning LLMs to act as specialized tutors capable of:
    - Providing structured, curriculum-aligned explanations for Algerian BAC subjects, ensuring clarity and pedagogical consistency.
    - Generating practice exercises that reflect official BAC standards and assessment methodologies [1].
    - Implementing an automated feedback system for correcting student answers with detailed, instructional guidance.
2. **Integration of Adaptive Learning Frameworks:** To move beyond static interaction, this work aims to integrate adaptive mechanisms that:
    - Analyze student interactions to provide data-driven insights into their progress.
    - Dynamically adjust difficulty levels to maintain the "Zone of Proximal Development," preventing both boredom and cognitive overload [4].
    - Personalize individual-learning pathways based on the unique proficiency and performance of each learner [2].
3. **Quality Assurance and Content Reliability:** A critical objective is to ensure the accuracy of AI-generated content across all subjects. This involves developing protocols for tuning model responses to minimize errors (hallucinations) and ensure that the generated educational material is reliable and pedagogically sound [6].
4. **Development of an Interactive Tutoring Ecosystem:** The final objective is to build a complete, real-time interactive environment that enables:
    - Direct student interaction through a dedicated chat-based interface.
    - Continuous, personalized educational support that removes the constraints of time and geography, democratizing access to high-quality tutoring [5].

1.4 Research Questions

To fulfill the aforementioned objectives and address the core challenges of localized education, this research is driven by the following primary questions:

- **RQ1 (Curriculum Adaptation):** How can a Large Language Model be optimally adapted and configured to accurately teach highly specific, localized educational content, such as the Algerian Baccalaureate curriculum?

- **RQ2 (Technical Comparison):** What are the comparative advantages and limitations of core LLM adaptation techniques specifically Prompt Engineering, Retrieval-Augmented Generation (RAG), and Fine-tuning when applied to strict pedagogical and domain-specific constraints? [8, 9]
- **RQ3 (Adaptive Integration):** Which architectural strategies and techniques can successfully integrate dynamic, adaptive learning mechanisms into an LLM-driven tutoring system to personalize student learning paths? [2]
- **RQ4 (Performance Benchmarking):** To what extent does a domain-specific, specialized educational LLM outperform general-purpose foundational models in terms of curriculum relevance, factual accuracy, and instructional quality? [5]
- **RQ5 (Quality Assurance):** What evaluation frameworks and metrics are most effective for assessing, ensuring, and maintaining the reliability and pedagogical soundness of AI-generated educational content? [6]

Project Contributions

This project provides significant practical and engineering contributions to the integration of Artificial Intelligence within localized educational systems. The main contributions of this work are:

- **Designing a Hybrid AI Tutoring Architecture:** We engineered a complete, production-ready system architecture that bridges foundational Large Language Models (LLMs) with a robust Retrieval-Augmented Generation (RAG) pipeline. This includes a dual-database paradigm utilizing MySQL for relational data and Qdrant for dense vector embeddings to ensure strict factual integrity.
- **Building Curriculum-Specific Datasets:** We collected, structured, and vectorized a comprehensive dataset based directly on the Algerian Baccalaureate curriculum. Additionally, we generated a large-scale synthetic student interaction dataset to train and validate predictive learning models.
- **Implementing Deep Knowledge Tracing (DKT):** We developed an advanced adaptive learning module powered by a Long Short-Term Memory (LSTM) neural network. This DKT model continuously tracks student interactions and predicts their future performance with high accuracy (measured via R^2 and MAE), automatically adjusting the pedagogical scaffolding and difficulty of the content.
- **Handling Localized Code-Switching:** The system was successfully adapted to natively process and understand the complex linguistic code-switching (mixing

Darja, formal Arabic, and French) typical of Algerian students, generating mathematically and scientifically accurate responses.

- **Developing and Evaluating the DeepBac PWA:** We built a functional, mobile-responsive Progressive Web App (PWA) prototype capable of complex mathematical rendering. Furthermore, we established a rigorous evaluation framework using an automated LLM-as-a-Judge approach to benchmark the system's ability to eliminate AI hallucinations.

1.5 Organization of the dissertation

To provide a clear and logical roadmap of the engineering and research work conducted in this project, the remainder of this dissertation is structured into the following chapters:

- **Chapter 2: Literature Review and Theoretical Background**
  This chapter establishes the theoretical foundations necessary to understand the system. It reviews the state-of-the-art in Large Language Models (LLMs) and core adaptation methodologies, including Prompt Engineering, Retrieval-Augmented Generation (RAG), and QLoRA Fine-tuning. Additionally, it explores the principles of educational data mining, focusing on Deep Knowledge Tracing (DKT).

- **Chapter 3: System Design and Architecture**
  This chapter presents the conceptual and structural blueprint of the DeepBac platform. It details the hybrid architecture, outlining the dual-database paradigm combining a relational database (MySQL) for metadata and a vector database (Qdrant) for dense curriculum embeddings. It also defines the sequence mechanics of the RAG pipeline and the architectural layout of the LSTM network used for student knowledge tracing.

- **Chapter 4: Implementation and Experimental Setup**
  This chapter describes the technical realization and engineering deployment of the platform. It covers the full-stack development using React for the mobile-responsive Progressive Web App (PWA) frontend and Node.js for the backend. It also details the integration of the cloud-based DeepSeek API, the local environment setup for running the ALLaM-7B model inside WSL, and the generation pipeline for the synthetic student interaction dataset.

- **Chapter 5: Evaluation and Results**
  This chapter provides a rigorous, empirical analysis of the developed modules. It details the quantitative testing of the RAG pipeline using an automated LLM-as-a-Judge framework (measuring Faithfulness) alongside local VRAM latency benchmarks. Furthermore, it evaluates the predictive performance of the continuous DKT

LSTM model using academic regression metrics (R^2 and MAE) and concludes with the qualitative user satisfaction results from the system's Beta deployment.

- **General Conclusion**
  The report concludes with a comprehensive summary of the project's primary contributions, an objective discussion of current technical limitations, and strategic future research perspectives for expanding the localized educational ecosystem.

Chapter 2

AI in Education

2.1 Introduction

The rapid development of Artificial Intelligence, particularly in Natural Language Processing (NLP), has produced models capable of understanding and generating human language with high accuracy. Large Language Models (LLMs) have shown strong performance in many areas, including text generation, question answering, and interactive dialogue [1, 10]. This success has led to a global interest in applying LLMs to different fields, especially the educational sector [2].

This chapter reviews the fundamental concepts and technologies relevant to this project. It covers the architectures of Large Language Models, their various applications in education, and the principles of adaptive learning systems . Additionally, it discusses multilingual NLP, which is essential for the bilingual context of this work. Finally, this chapter identifies current limitations and research gaps that justify the development of our proposed tutoring system, **DeepBac**.

2.2 Intelligent Tutoring Systems (ITS)

The idea of using computers to teach is not entirely new. For decades, researchers have developed Intelligent Tutoring Systems (ITS) (Figure 2.1) with a single goal: to mimic the one-on-one experience of a human tutor. Traditionally, the architecture of a classic ITS relies on three core components [11]:

- **The Domain Model:** This is the system's brain regarding the subject matter. It contains all the rules, facts, and concepts that need to be taught [12].
- **The Student Model:** This part tracks the learner. It records what the student already knows, the mistakes they frequently make, and their overall progress over time to provide a personalized experience [13].

- **The Tutoring Model (Pedagogical Model):** This acts as the teacher. Based on what the student knows (Student Model) and what needs to be taught (Domain Model), it decides the best instructional step or intervention to take next [14].

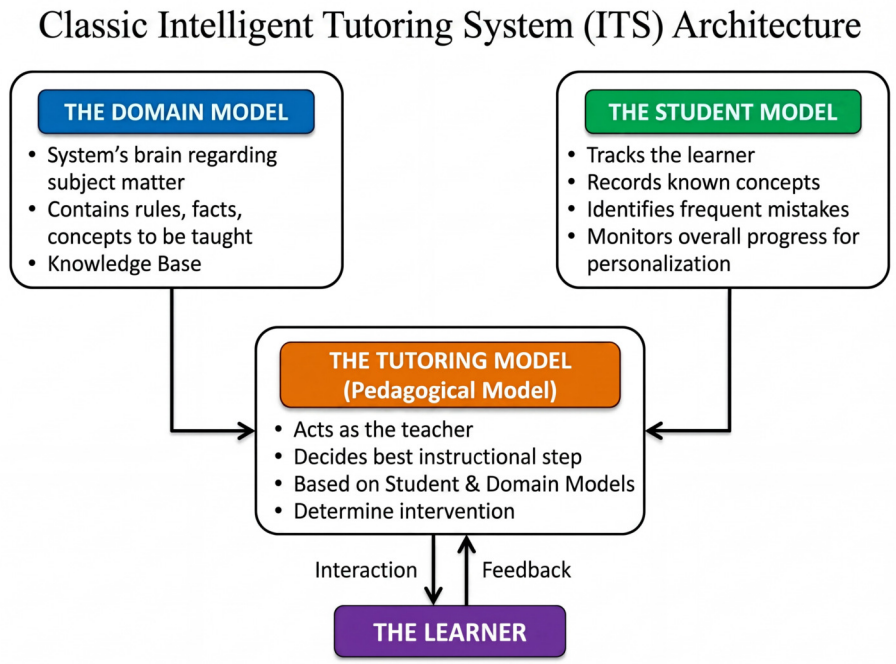


Figure 2.1: Architecture of a Classic Intelligent Tutoring System (ITS)

While these classic systems were effective, they were heavily constrained. Programmers and teachers had to manually write every single rule, question, and possible feedback scenario. If a student asked a unique question or typed an answer outside the pre-programmed script, the system would simply fail to understand.

This is exactly where modern AI, specifically Large Language Models (LLMs), changes the landscape. By integrating LLMs into the ITS architecture, we remove the rigid, rule-based limits. Instead of just clicking multiple-choice buttons or following static paths, students can now ask open-ended questions in natural language. The system does not just retrieve a pre-written answer; it generates custom explanations, simplifies complex topics on the spot, and maintains a natural, human-like dialogue [1]. This modern approach transforms basic educational software into a truly interactive academic mentor.

2.3 Large Language Models (LLMs)

To understand the technical foundation of this project, we must first define Large Language Models (LLMs). Fundamentally, LLMs are deep neural networks trained on massive amounts of text to learn the patterns of human language. Nearly all modern

LLMs rely on the Transformer architecture, which was introduced in 2017 [15]. The main advantage of the Transformer over previous recurrent networks is a mechanism called "self-attention." This feature allows the system to look at entire sequences of text at the same time, making it highly efficient at capturing context.

Recently, the AI community has released several major foundation models. The most relevant ones include:

- **GPT:** Created by OpenAI, these models set the current standard for text generation and conversational AI [16].
- **Gemini:** Google's flagship model, built from the ground up to be multimodal. Unlike older models that process only text, Gemini can natively understand and combine text, images, audio, and video, making it highly flexible for complex reasoning [17].
- **LLaMA:** Meta released this as an open-weight model. Researchers prefer it because it offers high performance while remaining computationally cheap to run locally [10].
- **DeepSeek:** A newer release that specifically focuses on strong logic, reasoning, and coding capabilities [18].

Creating one of these models usually takes two main steps. The first is pre-training. Here, the model reads raw internet data to learn grammar and general knowledge. The second step is fine-tuning, often utilizing Reinforcement Learning from Human Feedback (RLHF), which teaches the raw model how to actually answer questions and follow user instructions safely [19].

Even with extensive training, these baseline models share a common weakness. They frequently "hallucinate" which means they confidently generate incorrect information [6]. Furthermore, since they learn from global web data, they lack deep knowledge of localized subjects. This means we cannot use a raw LLM directly for a specialized task, like teaching the Algerian curriculum, without modifying it first.

2.4 The Convergence of LLMs and ITS

The integration of Large Language Models (LLMs) into Intelligent Tutoring Systems (ITS) represents a significant paradigm shift in the field of Artificial Intelligence in Education (AIEd) [1]. Historically, traditional ITS architectures relied heavily on rigid, predefined cognitive models and rule-based expert systems to track student knowledge and provide scaffolding [20]. While these classical systems proved effective within narrow, well-defined domains (such as basic mathematics or formal logic), they inherently

struggled to process open-ended student inquiries or engage in fluid, human-like dialogue [21].

The advent of advanced LLMs, such as OpenAI's GPT series and open-source alternatives like Meta's LLaMA, has effectively bridged this gap by introducing unprecedented Natural Language Understanding (NLU) and Natural Language Generation (NLG) capabilities into the tutoring loop [22]. This convergence transforms static tutoring software into highly interactive pedagogical agents capable of interpreting complex and ambiguous student intents.

The synergy between LLMs and ITS manifests in several transformative capabilities:

- **Conversational Pedagogical Agents:** LLMs enable tutors to engage students in Socratic dialogues, guiding them toward the correct answer through step-by-step questioning rather than simply providing the direct solution [23].
- **Dynamic Content Generation:** Unlike traditional systems restricted to static, hard-coded question banks, LLM-augmented ITS can dynamically generate personalized exercises, detailed explanations, and contextual analogies tailored to the student's current proficiency level [5].
- **Nuanced Error Diagnosis:** LLMs exhibit a superior ability to analyze free-text student answers, identifying underlying misconceptions and semantic errors that rigid keyword-matching algorithms typically miss.

Feature	**Traditional ITS**	**LLM-Augmented ITS**
Interaction Style	Pre-programmed, rigid paths (Menu/Click-based).	Open-ended, fluid Natural Language dialogue.
Knowledge Base	Static, manually coded rules and domain models.	Dynamic generation (NLU/NLG capabilities).
Pedagogical Strategy	Direct hints and static solutions.	Adaptive scaffolding and Socratic dialogues.
Error Diagnosis	Keyword matching; fails with unexpected input.	Semantic understanding and nuanced misconception diagnosis.
Content Generation	Limited to hard-coded question banks.	Personalized, on-the-fly exercises and contextual analogies.
Flexibility	Highly constrained to narrow, predefined domains.	Highly interactive and capable of resolving ambiguous intents.

Table 2.1: Comparative analysis between Traditional ITS and LLM-Augmented ITS architectures.

However, the standalone deployment of LLMs as educational tutors introduces significant pedagogical risks, primarily the generation of factually incorrect information (hallucination) and the lack of deterministic adherence to specific local curricula [6]. Consequently, the modern convergence of LLMs and ITS relies heavily on advanced adaptation techniques to constrain the model's outputs, ensuring pedagogical safety, factual accuracy, and alignment with structured educational objectives.

2.5 Adaptive Learning Systems

Traditional classrooms often follow a "one-size-fits-all" approach, forcing all students to learn at the exact same pace. This inevitably leaves some students struggling to catch up while others become bored. Adaptive learning systems aim to solve this issue. At their core, these systems use software to automatically adjust the educational content, pacing, and style based on the individual learner's real-time performance and behavior [24] (see Figure 2.2).

To make this personalization possible, modern adaptive systems rely on three main functional mechanisms:

- **Continuous Student Modeling:** The system does not just test the student once. It constantly updates a mathematical profile of the learner by tracking every interaction such as how long they spend reading a text, which questions they fail, and their historical quiz scores.
- **Dynamic Difficulty Adjustment:** Rather than providing a static list of questions, the system changes the next task based on the previous answer. If a student solves a math problem easily, the next problem becomes harder. If they struggle, the system steps back to review foundational concepts.
- **Actionable Feedback:** Instead of a generic "Wrong, try again" message, adaptive systems provide specific hints tailored to the exact mistake the student made, helping them correct their reasoning process.

From a psychological perspective, the main goal of these systems is to manage "Cognitive Load" [25]. If a task is too complex, the student's working memory overloads and they learn nothing. If it is too simple, they disengage. By continuously adjusting the difficulty, adaptive algorithms keep the student in their "Zone of Proximal Development" the sweet spot where tasks are challenging enough to promote learning but not so difficult that they cause frustration [26].

When integrated with Large Language Models, adaptive learning moves from simple multiple-choice branching to generating entirely personalized text and explanations on the fly, offering a highly customized learning journey.

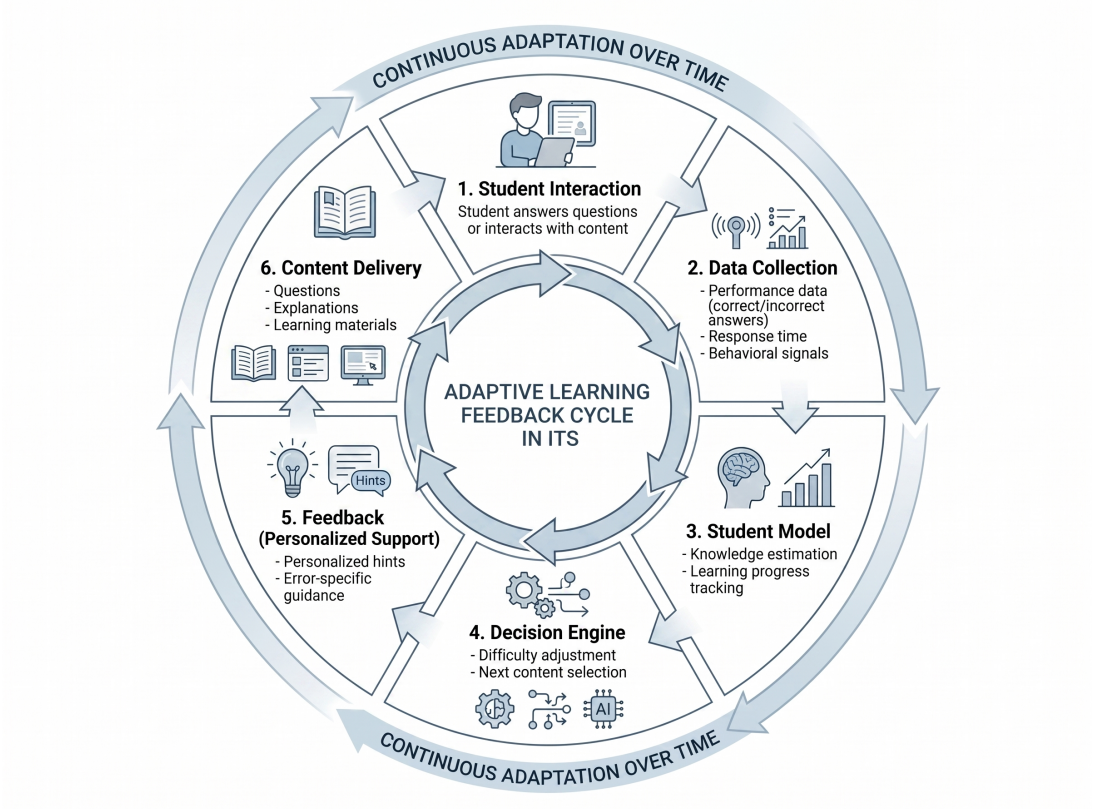


Figure 2.2: The continuous feedback cycle of an Adaptive Learning System, illustrating the dynamic interaction between data collection, student modeling, and personalized content delivery.

2.6 LLM Adaptation Techniques

As mentioned earlier, raw foundational models are too general to handle highly specific tasks right out of the box. To make them useful for specialized applications like teaching a local school curriculum developers must guide or modify them. We generally rely on three main adaptation techniques, ranging from simple text instructions to deep structural changes.

2.6.1 Prompt Engineering

This is the simplest and most common adaptation method. Prompt engineering does not change the model's internal weights or code. Instead, it focuses on writing highly specific input instructions to shape how the AI behaves. By defining a strict "system persona" (telling the AI who it is) or using techniques like "few-shot prompting" (giving the model examples of good and bad answers), developers can easily steer the output [9]. While it is fast and costs almost nothing to implement, prompt engineering alone is often not enough if the model completely lacks the required background knowledge.

2.6.2 Retrieval-Augmented Generation (RAG)

RAG solves the problem of missing or outdated knowledge. Instead of hoping the LLM remembers a specific fact from its initial training, RAG connects the model to an external, trusted database [8]. When a student asks a question, the system first searches this database (which could contain official BAC textbooks) to retrieve the exact paragraphs related to the question. It then feeds those paragraphs to the LLM to write a final, easy-to-read answer. This approach significantly reduces "hallucinations" and ensures every generated response is grounded in the actual school curriculum.

2.6.3 Fine-Tuning

When prompt engineering and RAG still do not produce the desired tone or format, we physically alter the model through fine-tuning. This means we continue the training process using a smaller, domain-specific dataset. Historically, updating every single parameter of a massive model (Full Fine-Tuning) was extremely expensive and required supercomputers. However, recent breakthroughs in Parameter-Efficient Fine-Tuning (PEFT), particularly a method called LoRA (Low-Rank Adaptation), solved this issue [27]. LoRA freezes the original model and only trains a tiny set of new parameters. This allows researchers to deeply adapt large models using normal, consumer-grade GPUs without losing performance.

Technique	Modifies Parameters?	Knowledge Source	Cost	Primary Use Case
Prompting	No	Model's internal (static) pre-trained memory.	Very Low	Steering output format, setting system persona, and guiding via few-shot examples.
RAG	No	External, dynamic database (e.g., BAC textbooks).	Low	Grounding answers in verified facts, handling local curricula, and stopping hallucinations.
PEFT (LoRA)	Partial	Internalized via a targeted, domain-specific dataset.	Medium	Efficiently adapting the model's tone and behavior using consumer-grade GPUs.
Full Fine-Tuning	Yes	Deeply internalized via massive training datasets.	VeryHigh	Inducing massive architectural shifts or creating entirely new base models.

Table 2.2: Detailed comparison of LLM Adaptation Techniques highlighting structural parameter modifications, cost efficiency, and optimal operational use cases.

2.7 Multilingual NLP and Code-Switching

The linguistic landscape in the Algerian educational system is highly diverse(Figure 2.3). While Modern Standard Arabic (MSA) is the official language of instruction, French has historically dominated scientific subjects. More recently, English has gained massive importance, becoming a core, mandatory subject in the Baccalaureate and the primary language for modern technology studies.

Because of this diversity, students rarely communicate using just one language. In their daily lives and study sessions, they naturally mix MSA, French, English, and the local Algerian dialect (Darja). In Natural Language Processing (NLP), this constant mixing of languages within a single conversation is known as "code-switching" [28].

For example, a student asking a math question might say: "*Kifech n'calculer la limite ta3 la fonction?*". Alternatively, a student studying for the English exam might ask: "*Oustad, kifach n'formi el passive voice fi had el sentence?*" (Teacher, how do I form the

passive voice in this sentence?). These inputs combine Arabic sentence structure with French and English vocabulary in a single breath.

From a technical perspective, this trilingual reality creates a massive hurdle for AI models. Most Large Language Models are trained primarily on high-resource, monolingual datasets (like pure English or formal news Arabic). When faced with inputs containing three languages plus a local dialect, standard models often get confused. They might misclassify the language, lose the core context, or reply with broken grammar [29].

To make an intelligent tutor like DeepBac actually work for Algerian students, it cannot force them to speak in perfect, formal Arabic. The system must handle this code-switching naturally. Instead of building a traditional, rigid NLP pipeline from scratch to parse these languages separately, DeepBac leverages the inherent zero-shot multilingual capabilities of state-of-the-art LLMs. By integrating these advanced models, the system seamlessly and automatically handles code-switching by:

- **Implicitly identifying** the mixed languages (Arabic, French, Darja) in real-time.
- **Semantically mapping** cross-lingual terms (e.g., matching "dérivée" to the Arabic curriculum's derivative concept) internally without requiring manual translation layers.
- **Generating** the pedagogical response in the appropriate target language while maintaining the educational context [30].

Addressing this complex language mixing is essential. Without it, the system would fail to understand the natural thought process of a local high school student.

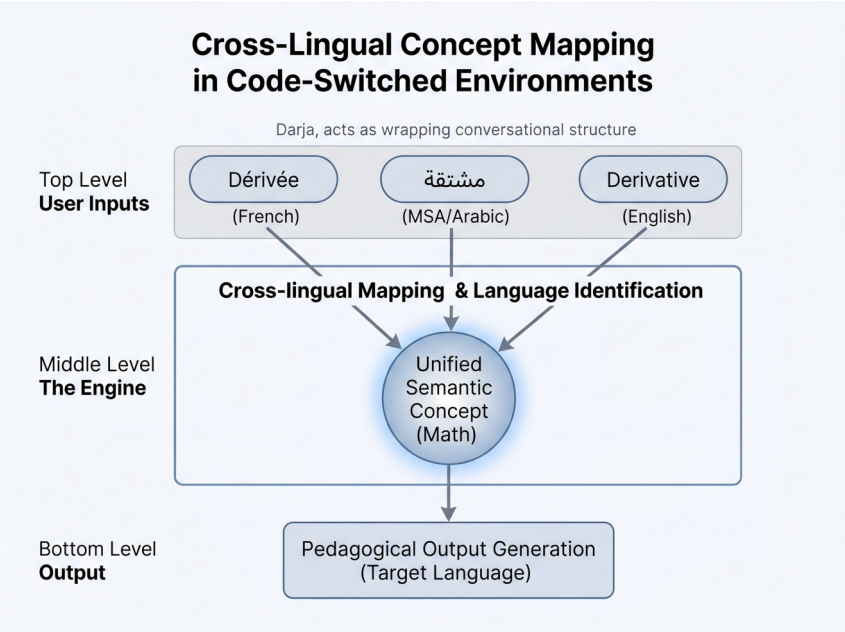


Figure 2.3: The cross-lingual semantic mapping process, illustrating how diverse linguistic inputs are unified into a single abstract pedagogical concept before response generation.

2.8 Related Work

The deployment of Intelligent Tutoring Systems (ITS) has evolved dramatically with the integration of foundational Large Language Models (LLMs). Rather than merely defining the underlying algorithms, this section reviews recent state of the art applied research across three domains highly analogous to this project: localized LLM adaptation for dialectal/code-switched education, curriculum-grounded Retrieval-Augmented Generation (RAG), and the fusion of Generative AI with Deep Knowledge Tracing (DKT).

2.8.1 LLM Adaptation for Localized and Code-Switched Education

While general purpose LLMs exhibit strong conversational capabilities, their application to localized educational contexts particularly those involving Arabic dialects and linguistic code switching (e.g., Arabic-French-Darja) remains a prominent research frontier. Recent monumental works, such as the development of the bilingual Arabic–English J Jais model by [31], highlight the absolute necessity of Arabic-centric foundation models to capture morphological nuances and code-switching patterns. Similarly, Atlas-Chat [32] and Nile-Chat [33] demonstrated that instruction-tuning on dialectal datasets enables models to outperform general-purpose LLMs on tasks like DarijaMMLU.

To adapt these massive models for highly specific educational tasks without prohibitive computational costs, recent studies have successfully deployed Parameter-Efficient Fine-Tuning (PEFT). For instance, GemMaroc [34] showed that fine-tuning a compact LLM with minimal, reasoning-dense data can achieve 61.6% on DarijaMMLU surpassing prior models while using far less energy. In the Algerian context, DziriBOT [35] applied a fine-tuned DziriBERT model with RAG to handle the non-standardized orthography, French code-switching, and dual-script (Arabizi) nature of Darja. Further, MENARA [36] introduced a fine-grained dialect composition analysis to quantify lexical purity and structured code-switching behavior across Egyptian and Moroccan dialects. Collectively, these works validate that parameter-efficient techniques can transform generalized models into culturally and linguistically aware localized tutors, while preserving foundational reasoning and adapting the pedagogical tone.

2.8.2 Curriculum-Grounded RAG in High-Stakes Assessments

Despite successful fine-tuning, LLMs inherently suffer from factual hallucinations, which are strictly unacceptable in high-stakes educational environments like national Baccalaureate exams. Recent educational AI research has thus shifted almost entirely toward heavily constrained Retrieval-Augmented Generation (RAG) architectures [8].

Numerous 2026 studies have demonstrated that grounding LLM responses strictly in teacher-provided or curriculum-aligned materials drastically reduces domain-specific hallucinations. For example, [37] built a Moodle plugin employing RAG with a dual-centric design, achieving faithfulness scores up to 0.97 and mitigating misinformation via verifiable citations. Similarly, [38] proposed a three-layered framework (input, model, output) for mitigating hallucinations in educational LLMs, emphasizing RAG as a core strategy at the model level. Recent implementations in intelligent tutoring systems further highlight that dual-database architectures separating relational student data from dense semantic embeddings are critical for real-time educational question-answering [37]. These contemporary frameworks validate that shifting the LLM's role from an ungrounded knowledge generator to an in-context synthesis engine is the only viable methodology for reliable, safe, and academically rigorous exam preparation.

2.8.3 Fusing Deep Knowledge Tracing with Generative AI

Modern personalized education relies heavily on accurately modeling a student's cognitive state. While Deep Knowledge Tracing (DKT) via sequential neural networks remains the industry standard for predicting student mastery over time, the current frontier of research lies in fusing these predictive models with generative AI to create responsive tutoring systems.

Recent breakthroughs, such as RAG-KT [39], frame cross-platform knowledge tracing as a retrieval-augmented paradigm that uses multi-view structured context to constrain LLM predictions, achieving strong generalization under distribution shift. TutorLLM [40] combined knowledge tracing with RAG in a Chrome plugin, demonstrating a 10% increase in user satisfaction and a 5% improvement in quiz scores compared to general LLMs. Beyond simple integration, [41] introduced Responsible-DKT, a neural-symbolic approach that injects symbolic educational knowledge (e.g., mastery/non-mastery rules) into sequential neural models, achieving over 0.90 AUC while providing intrinsic interpretability. Unlike classical systems that merely display a probability score, these modern architectures use DKT outputs as a dynamic prompt-injection mechanism: when the predictive model identifies a latent weakness, it forces the LLM to generate targeted pedagogical scaffolding and step-by-step hints rather than direct answers. Our project builds directly upon this cutting-edge 2025-2026 intersection, utilizing DKT not just for analytics, but as the generative steering wheel for the LLM core.

2.9 Synthesis: Research Gaps and Proposed Solution

While the existing literature demonstrates significant advances in AI-driven education, a closer look reveals several critical shortcomings that persist even in very recent work.

Most major developments in EdTech and large language models target broad, global applications, leaving highly specific local needs unaddressed. From our review of the latest research, we identify three primary gaps that directly motivate this thesis.

2.9.1 Identified Research Gaps

- **Lack of strict curriculum alignment and localised systems.** The vast majority of existing AI tutors rely on general internet knowledge or are designed for Western curricula. For high-stakes assessments like the Algerian Baccalaureate, this is insufficient. A student cannot trust an AI that might explain a concept using a methodology not recognized by the Algerian Ministry of National Education. Even recent works such as Atlas-Chat [32] and GemMaroc [34] focus primarily on general-purpose dialectal NLP rather than curriculum-grounded educational applications. Thus, there remains an absence of intelligent systems engineered from the ground up to respect the rigid boundaries of the Algerian syllabus, leaving students vulnerable to model hallucinations.

- **Superficial integration of adaptive learning with conversational AI.** The educational software landscape remains largely bifurcated. On one side, traditional adaptive platforms track student scores but offer rigid, predefined multiple-choice paths. On the other side, modern LLM chatbots provide free-flowing conversation but lack persistent pedagogical memory of a student's long-term progress. Recent studies like RAG-KT [39] and TutorLLM [40] have begun exploring the integration of knowledge tracing with LLMs, yet these approaches remain platform-specific or lack the deep, continuous coupling required for open-ended tutoring in high-stakes settings.

- **Insufficient handling of scientific code-switching.** Despite progress in Arabic NLP, research predominantly focuses on Modern Standard Arabic or sentiment analysis. The structured educational environments of North Africa where students constantly code-switch between formal Arabic, French scientific terminology, and local dialects (Darja) [28]—remain under-addressed. Recent dialectal models such as DziriBOT [35] and MENARA [36] demonstrate the feasibility of dialect-aware NLP, but they are not tailored to the pedagogical and scientific vocabulary requirements of exam preparation.

2.9.2 Our Proposed Solution

These gaps confirm that deploying a raw, off-the-shelf LLM is not viable for Algerian students. This reality directly shapes the architecture of our proposed system, **DeepBac**. Unlike prior work, DeepBac integrates three cutting-edge strategies:

- **Curriculum-grounded RAG.** We employ retrieval-augmented generation strictly constrained to official textbooks and authorized resources, building on recent frameworks that mitigate hallucinations through structured retrieval [38, 37].

- **Deep Knowledge Tracing (DKT) as a steering mechanism.** Rather than treating knowledge tracing as a separate analytic module, we embed a DKT component that actively guides the LLM's responses injecting constraints when latent weaknesses are detected inspired by the latest neural-symbolic approaches [41, 39].

- **Parameter-efficient fine-tuning for code-switching.** We apply QLoRA and instruction-tuning techniques, similar to those validated in Atlas-Chat [32] and GemMaroc [34], to adapt an open-source foundation model to the trilingual (Arabic-French-Darja) educational context of Algeria.

Thus, DeepBac is not merely another chatbot but a localised, pedagogically accurate, and linguistically aware intelligent tutoring environment.

2.10 Conclusion

In this chapter, we explored the foundational technologies behind modern AI-based education: the mechanics of LLMs, the evolution of intelligent tutoring systems, and the pedagogical principles of adaptive learning . We also examined the complex linguistic reality of the Algerian context, highlighting the challenges of processing trilingual code-switching.

Despite undeniable global progress, the literature including the most recent 2025–2026 advances reveals a persistent lack of tools engineered for localized, high-stakes educational environments. General-purpose models and standard chatbots cannot provide the pedagogical safety and curriculum adherence that the Baccalaureate exam demands.

These limitations directly motivate our practical implementation. In the next chapter, we present the complete system design and technical architecture of **DeepBac**, showing exactly how we turn raw generative capabilities into a reliable, localised intelligent tutor.

Chapter 3

System Design and Architecture

3.1 Introduction

Building upon the theoretical foundations and the research gaps identified in the previous chapter, this chapter presents the detailed system design and technical architecture of our proposed solution: **DeepBac**.

DeepBac is an intelligent tutoring platform driven by Large Language Models. However, unlike general-purpose chatbots, the core models in this system have been specifically adapted and constrained to align strictly with the Algerian national curriculum, focusing exclusively on the Baccalaureate (BAC) academic level. The primary objective of this architecture is to deliver personalized, interactive, and pedagogically accurate educational support to local students.

To achieve this, the system is designed using a modular architecture. It integrates several interconnected components: a responsive user interface (Frontend), a central processing server (Backend), and the adapted AI core powered by Retrieval-Augmented Generation (RAG) pipelines. In the following sections, we will break down each of these layers, explaining our technical design choices, the flow of data between the student and the AI, and how these separate parts work together to form a complete educational ecosystem.

3.2 System Architecture Overview

The DeepBac system is built on a modular, client-server architecture. This design separates the user-facing application from the heavy computational tasks required by the AI models. By keeping these components independent, the system becomes easier to maintain, test, and update. The overall architecture consists of five primary layers (see Figure 3.1):

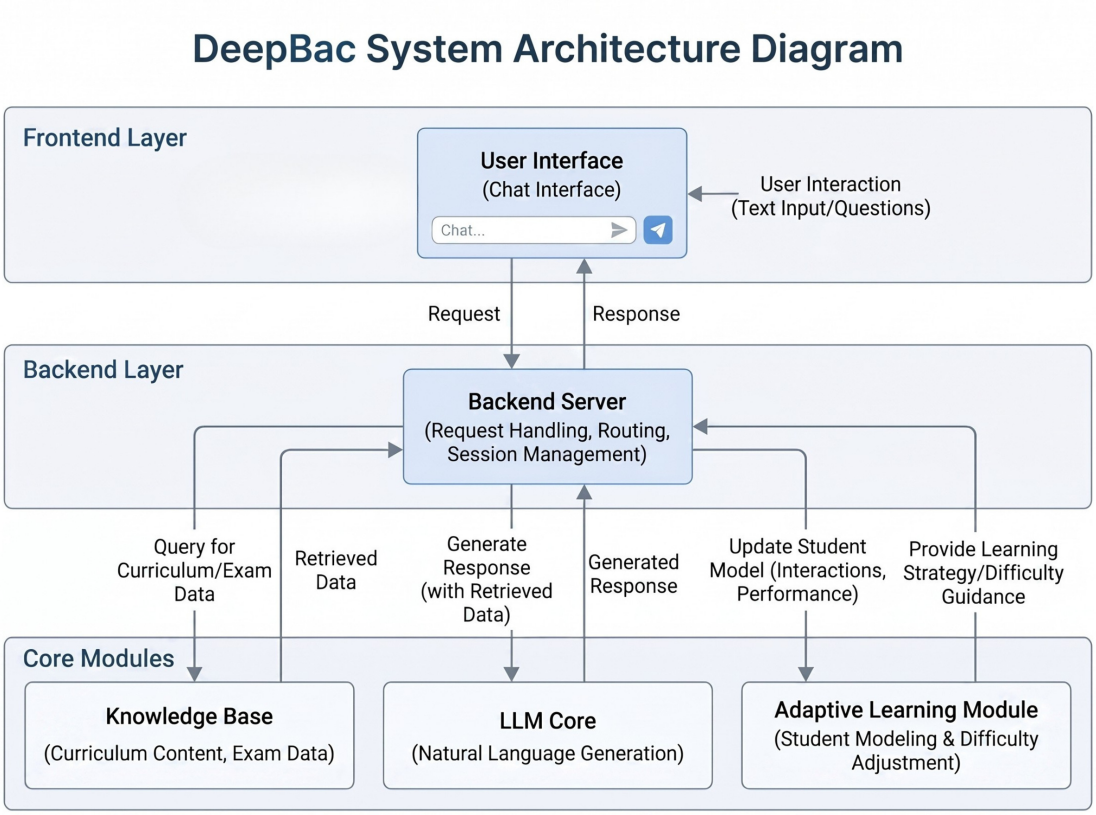


Figure 3.1: High-Level Architecture of the DeepBac System

- **User Interface (Frontend):** This is the visual layer where students interact with DeepBac. Designed as a responsive, chat-based interface, it allows students to ask questions naturally even using mixed languages or code-switching. It is built to cleanly display text, mathematical formulas, and structured exercises.

- **Backend Server:** Acting as the central router of the platform, the backend manages all incoming requests from the user interface. It handles logic execution, processes user sessions, and coordinates the flow of data between the frontend and the heavy AI modules, ensuring fast and stable response times [43].

- **Knowledge Base (Retrieval System):** Because raw LLMs lack specific local context, this database acts as DeepBac's official memory. It stores the structured Algerian BAC curriculum, including vetted lessons, past exam questions, and annotated solutions. When a student asks a question, the system searches this database first to retrieve factual context [8].

- **LLM Core:** This is the computational engine responsible for natural language generation. Instead of answering purely from memory, it takes the facts retrieved from the Knowledge Base and formats them into a clear, pedagogical response. It generates the step-by-step explanations and curriculum-aligned feedback.

- **Adaptive Learning Module:** Operating continuously in the background, this module tracks the student's interactions. It records the accuracy of their answers to update an internal "student model." Based on this data, it instructs the LLM Core to dynamically raise or lower the difficulty of the next generated exercises [26].

Together, these five modules create a continuous operational loop: the student asks a question, the backend retrieves verified facts from the knowledge base, the LLM generates a personalized explanation, and the adaptive module updates the student's progress to prepare for the next interaction.

3.3 Use Case Diagrams and Functional Requirements

To understand the functional requirements of DeepBac from the user's perspective, we rely on Use Case modeling. A Use Case diagram (Figure 3.2) visually represents who interacts with the system (Actors) and the specific actions they can perform (Use Cases) [44]. Identifying these scenarios is essential for validating that our proposed architecture actually meets the needs of Algerian BAC students.

For this platform, we have identified three primary actors:

1. **The Student:** The core user of the system who seeks personalized educational support.
2. **The Pedagogical Expert (Teacher):** A subject-matter specialist responsible for ensuring the academic quality, accuracy, and curriculum alignment of the platform's content.
3. **The Admin (Administrator):** A technical user responsible for managing the platform's infrastructure, security, and user accounts.

Below, we detail the core use cases identified for each actor, followed by a graphical diagram.

3.3.1 Student Use Cases

The student interacts with DeepBac through a dynamic interface to access personalized learning features. The main functionalities available to the student include:

- **Create Account and Authenticate:** The student can register for a new profile and securely log into the platform. Authentication is mandatory to track learning progress, maintain session memory, and access premium features.

- **Ask Pedagogical Question (Querying the BAC Knowledge Base):** The student can ask questions in natural language, mixing Arabic, French, and English (Darja). The system processes the query, retrieves the correct BAC curriculum facts, and generates an adapted explanation.
- **Access Official Lessons and Past Exams:** The student can browse a comprehensive, organized library of official BAC lessons, textbook summaries, and previous years' exam papers to solidify their foundational knowledge.
- **Practice Curated BAC Exercises:** The student can select and solve pre-defined exercises crafted and verified by Pedagogical Experts. After submitting their text or mathematical answer, the AI evaluates the submission and provides detailed pedagogical corrections.
- **Engage in Smart Generative Review:** The student specifies a subject, lesson, and sub-topic. The LLM dynamically acts as an examiner, generating a customized set of questions to test the student's specific knowledge, followed by immediate automated correction and feedback.
- **View Progress Dashboard:** The student can view a summary of their performance over time. This data is fed by the Adaptive Learning Module and helps the student easily identify their weak points and learning trends.
- **Access Psychological and Motivational Support:** The platform provides curated articles, stress-management tips, and motivational guidance specifically designed to help students cope with the psychological pressure of high-stakes BAC exams.
- **Manage Profile and Gamified Progress:** The student can manage their subscription tier and track their overall platform access. Furthermore, they interact with a motivational gamification system, earning points, completing educational challenges, and unlocking badges based on their consistent study habits and success rates.

3.3.2 Pedagogical Expert (Teacher) Use Cases

The Teacher acts as the human-in-the-loop (HITL), ensuring that the AI's behavior aligns perfectly with the official BAC standards. Their main functionalities include:

- **Manage Subject Content:** The teacher uploads, structures, and updates the official lessons, exercises, and annotated solutions for their specific subject domain into the Knowledge Base.

- **Review and Validate AI Content:** The teacher audits the explanations, generated exercises, and automated feedback produced by the LLM Core. They can evaluate, edit, or correct this content to train the system and prevent hallucinations.
- **Monitor Student-AI Interactions:** The teacher can review complex student questions and the corresponding answers provided by the AI. This allows the teacher to intervene if a pedagogical concept is poorly explained by the system.

3.3.3 Admin Use Cases

The Admin actor ensures the reliability, security, and operational health of the platform. The Admin's functionalities include:

- **Manage Users & Subscriptions:** The Admin is responsible for managing student and teacher accounts, controlling access levels, and monitoring platform subscription statuses.
- **Global Knowledge Base Administration:** The Admin has overarching control of the database. They can perform bulk data uploads, oversee content structuring, and remove outdated or flagged materials to maintain the overall integrity of the retrieval system.
- **Monitor System Logs and Performance:** Accessing usage statistics, server traffic, and AI error logs to ensure the system runs smoothly without latency.

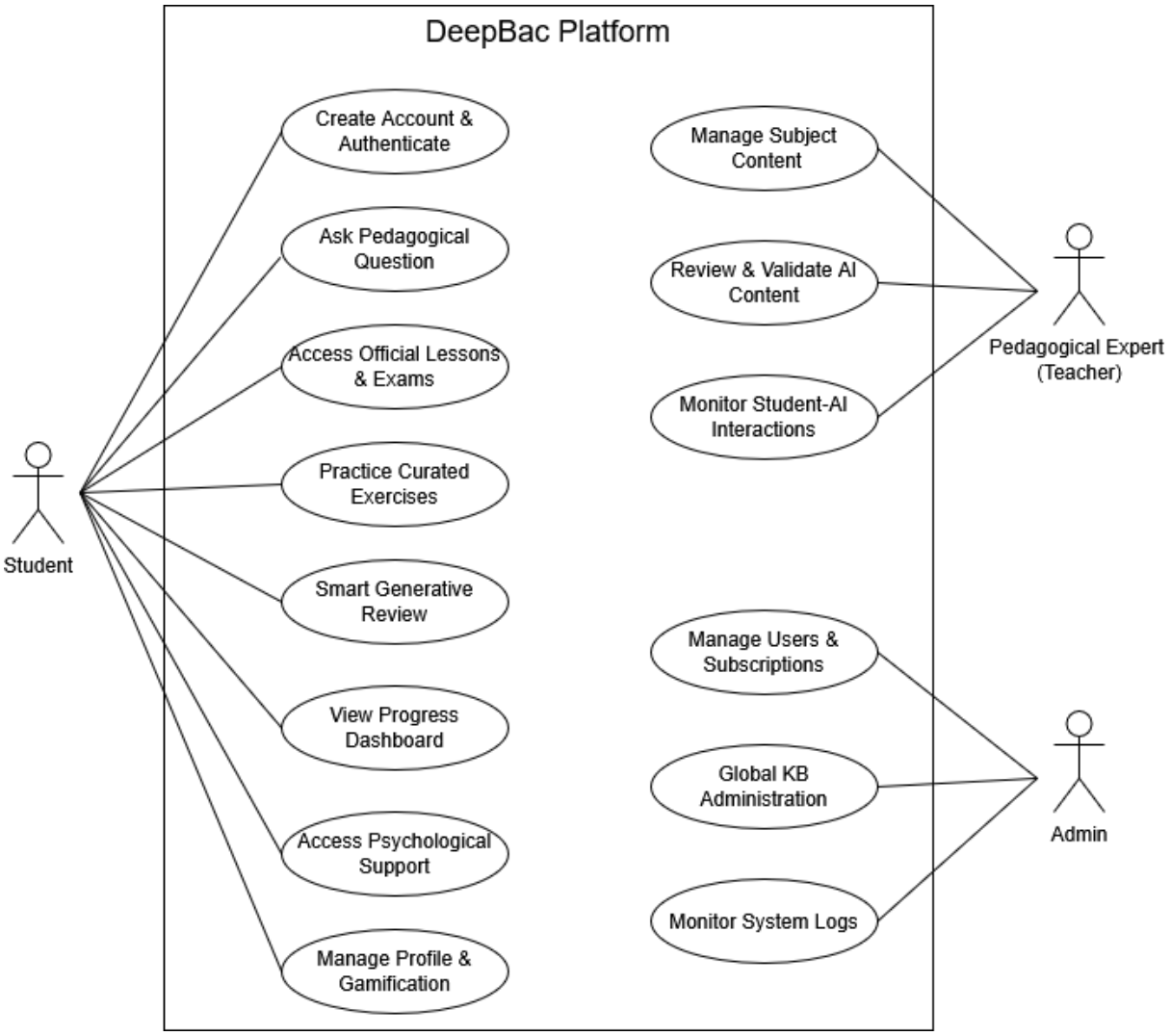


Figure 3.2: Use Case Diagram for the DeepBac System, illustrating the distinct interactions of the Student, Teacher, and Admin within the platform.

Together, these use cases demonstrate that DeepBac is designed not just as a standard chatbot, but as a comprehensive, human-validated management tool for localized BAC exam preparation.

3.3.4 Sequence Diagrams

While the Use Case diagram defines *what* the system does, the Sequence Diagram illustrates *how* it does it over time. It maps the exact flow of data and messages between the different system components during core operations.

Scenario 1: Handling a Pedagogical Question (RAG Workflow)

The most critical operation in DeepBac is handling a student's pedagogical question. The data flow for this scenario follows these sequential steps (see Figure 3.3):

1. **User Request:** The student types a question (e.g., asking for an explanation of a math theorem) into the Frontend interface.

2. **Data Transmission:** The Frontend sends this query via an API call to the Backend Server.

3. **Context Retrieval:** The Backend forwards the search query to the Knowledge Base to find the relevant official BAC curriculum texts (Retrieval-Augmented Generation process).

4. **Context Return:** The Knowledge Base returns the matched documents and formulas back to the Backend.

5. **Prompt Assembly:** The Backend combines the student's original question, the retrieved BAC documents, and the student's current proficiency level (fetched from the Adaptive Learning Module) into one structured prompt.

6. **LLM Processing:** This assembled prompt is sent to the LLM Core, which processes the information and generates a customized, pedagogically sound answer.

7. **Progress Update:** While the answer is being generated, the Backend updates the Adaptive Learning Module with the interaction logs to refine the student's profile.

8. **Final Delivery:** The Backend sends the generated explanation back to the Frontend, where it is displayed to the student.

This sequential flow ensures that the AI never guesses an answer directly, but always waits for verified local context before generating a response.

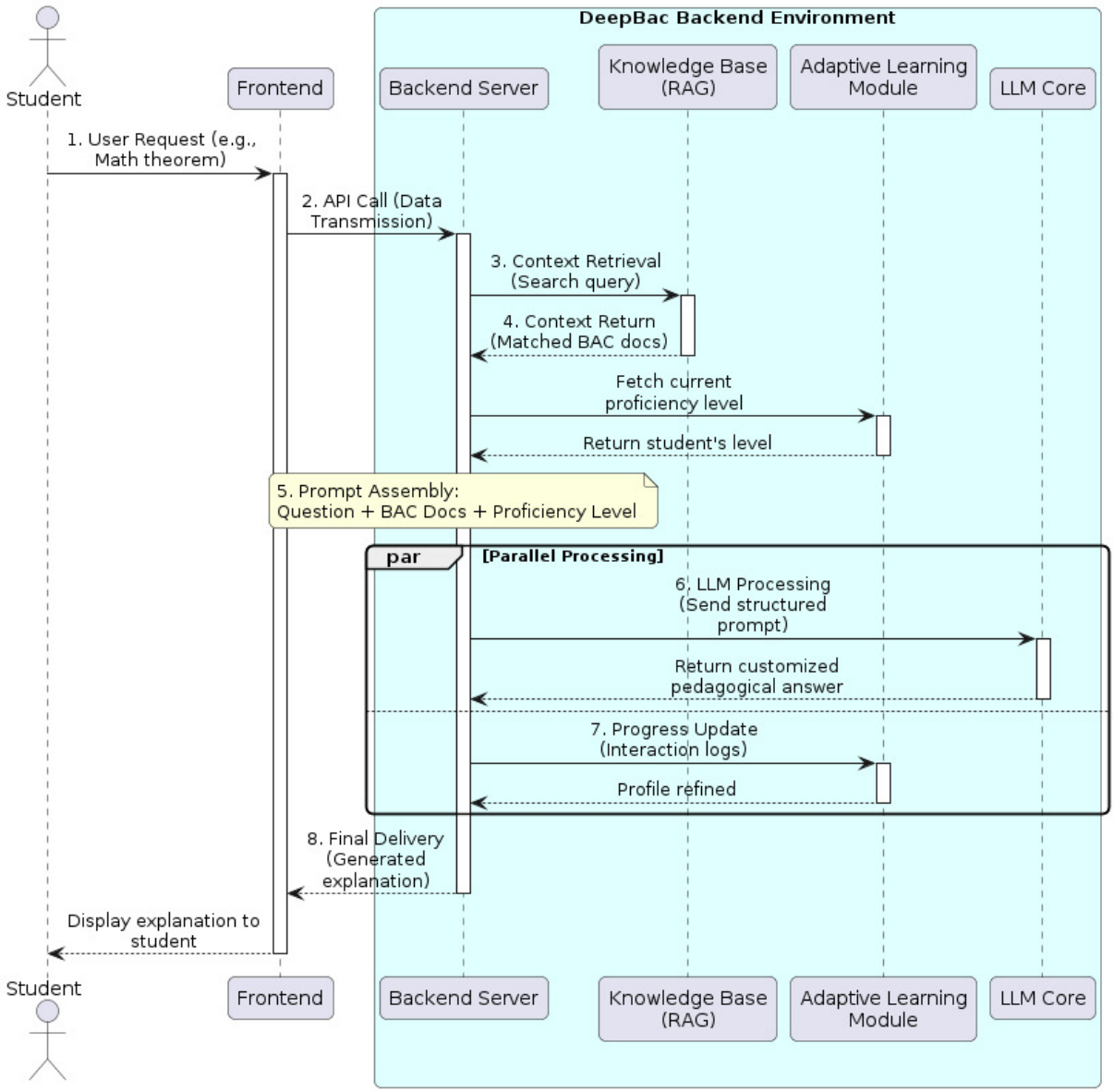


Figure 3.3: Sequence diagram illustrating the Retrieval-Augmented Generation (RAG) workflow when handling a student's pedagogical question.

Scenario 2: Generating Adaptive BAC Exercises

This sequence (Figure 3.4) illustrates how DeepBac dynamically generates practice materials. It is a critical workflow that demonstrates the system does not pull random exercises from the internet, but rather constructs them based on the official curriculum and the student's personal proficiency level.

1. **Exercise Request:** The student requests a practice exercise for a specific topic (e.g., Exponential Functions) via the Frontend interface.

2. **Data Transmission:** The Frontend forwards this request to the Backend Server via an API call.

3. **Proficiency Check:** The Backend queries the Adaptive Learning Module to determine the student's current skill level in this specific topic (e.g., beginner, average, or advanced).

4. **Template Retrieval:** The Backend searches the Knowledge Base to retrieve the official BAC lessons, relevant exercise templates, assessment standards, and past exam questions related to the requested topic.

5. **LLM Generation:** The Backend compiles the topic, the retrieved BAC templates, and the student's proficiency level into a structured prompt, sending it to the LLM Core. The LLM then generates a brand-new, customized exercise along with its official model answer.

6. **Display Exercise:** The Backend securely sends the generated exercise back to the Frontend, where it is presented to the student.

This precise workflow ensures that every generated practice session is both pedagogically appropriate for the individual student and strictly aligned with the national Algerian Baccalaureate standards.

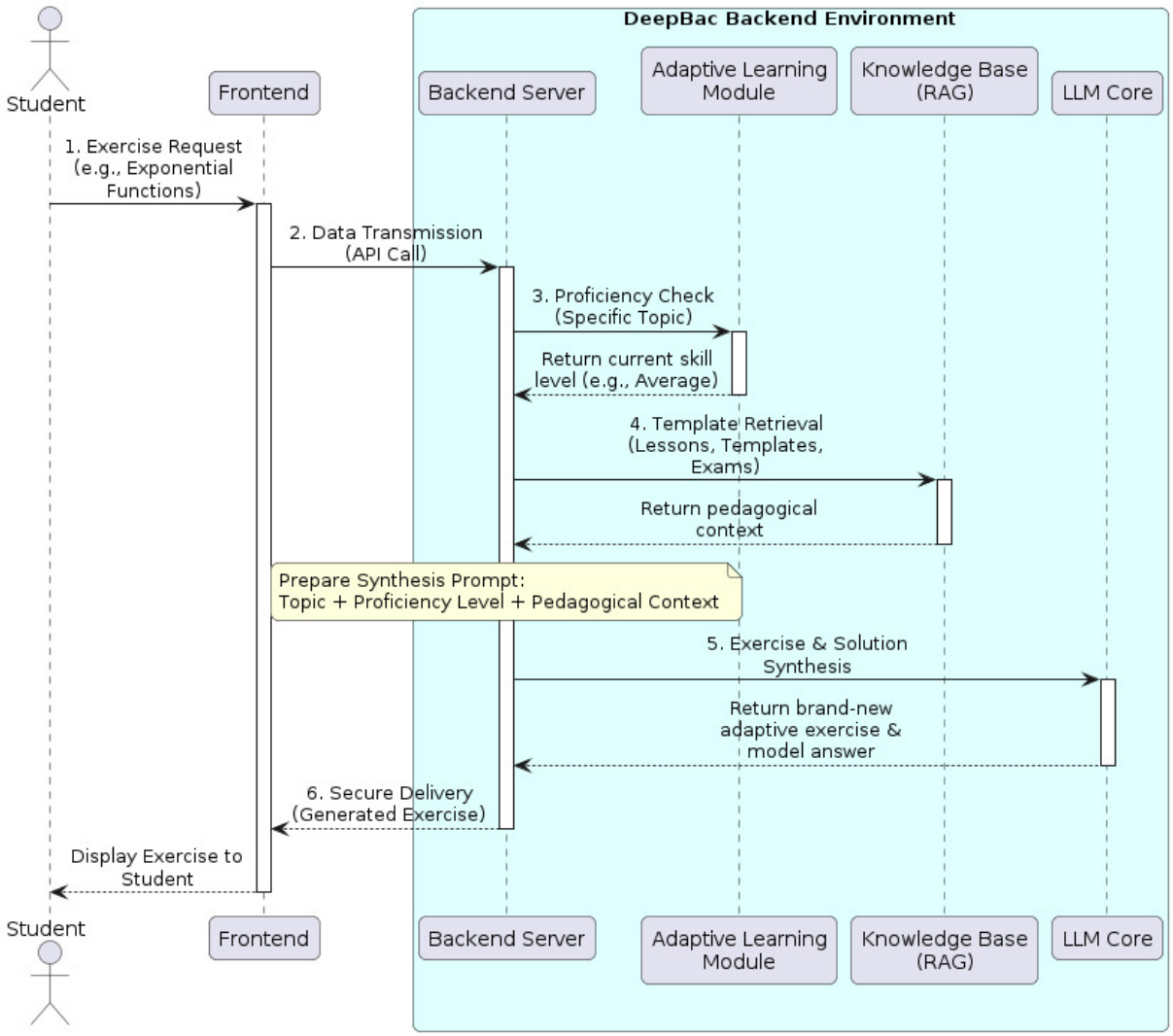


Figure 3.4: Sequence diagram illustrating the adaptive generation of BAC exercises based on student proficiency and curriculum templates.

Scenario 3: Automated Evaluation and Adaptive Profiling

This sequence (Figure 3.5) maps the core of DeepBac's continuous knowledge tracing capabilities. It details how the system processes student submissions, provides constructive feedback, and dynamically updates the student's pedagogical profile.

1. **Answer Submission:** After completing a generated exercise, the student submits their solution (which may include text and mathematical formulas) through the Frontend.

2. **Data Transmission:** The Frontend sends the student's submission to the Backend Server for evaluation.

3. **Solution Retrieval:** The Backend fetches the specific exercise context and its official model answer from the active session memory or Knowledge Base.

4. **Comparative Analysis (LLM):** The LLM Core performs a deep semantic comparison between the student's submission and the model answer. Instead of merely checking for the final correct result, it identifies logical flaws, misapplied theorems, or calculation errors.

5. **Feedback Generation:** Based on the analysis, the LLM generates targeted, step-by-step pedagogical feedback. It explains the exact nature of the mistake and guides the student toward the correct methodology without simply handing over the final answer.

6. **Profile Update (Knowledge Tracing):** Simultaneously, the Backend extracts the performance metrics (e.g., success rate, specific concept failures) from the LLM's analysis and pushes this data to the Adaptive Learning Module. This continuously updates the student's proficiency model.

7. **Feedback Delivery:** The customized pedagogical feedback is sent to the Frontend and displayed to the student.

This workflow guarantees that every interaction serves a dual purpose: instructing the student in real-time and refining the system's understanding of their long-term educational needs.

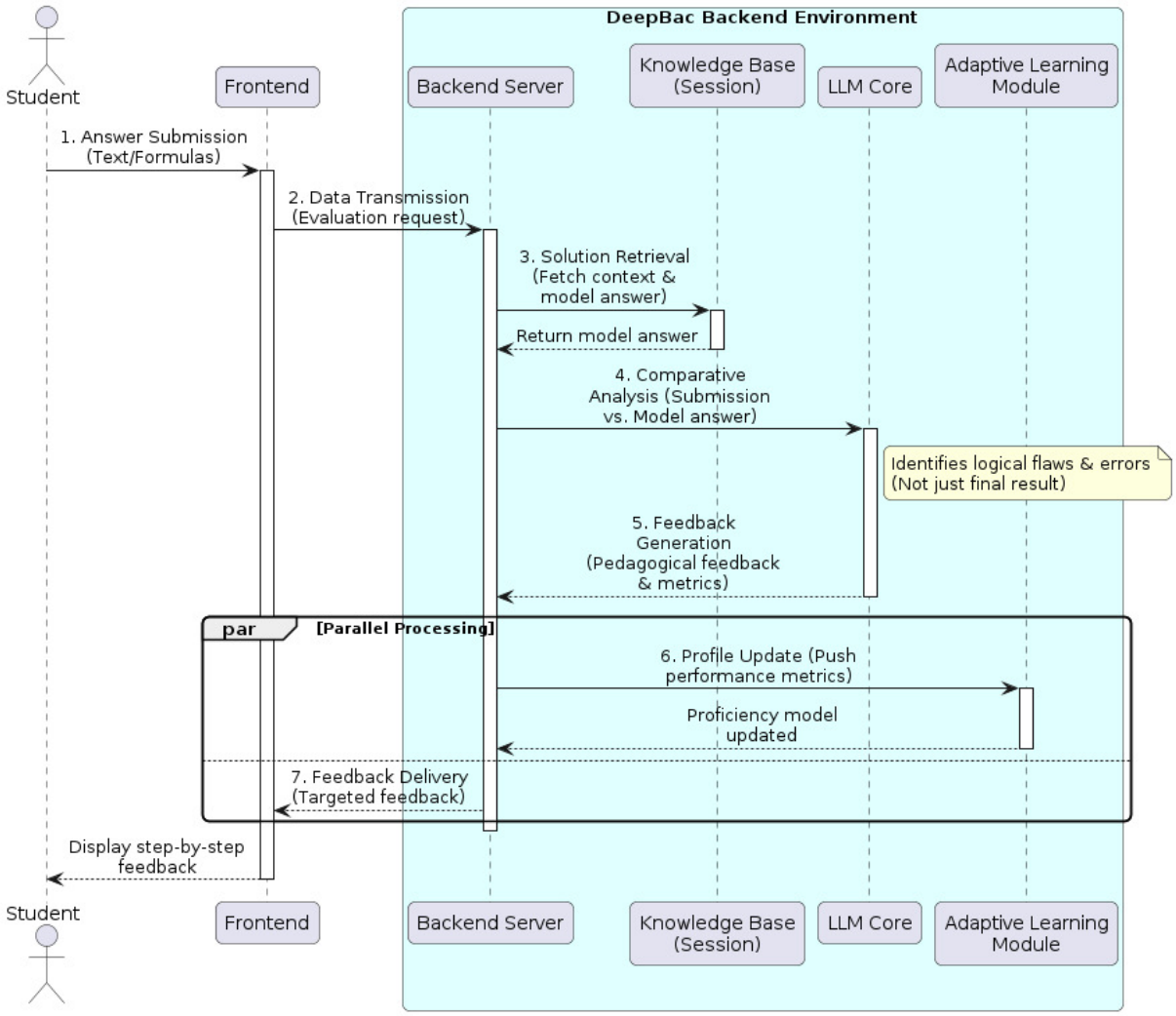


Figure 3.5: Sequence diagram detailing the automated evaluation process, highlighting the comparative analysis by the LLM and the continuous updating of the student's adaptive profile.

Scenario 4: Integrated RAG and DKT Workflow (Full Backend Orchestration)

This sequence diagram (Figure 3.6) illustrates the complete backend execution flow of DeepBac, highlighting how the Node.js orchestrator handles vectorization (Cohere), semantic retrieval (Qdrant), dynamic prompt injection based on student mastery (MySQL), and final generation (DeepSeek). Unlike the previous RAG workflow, this scenario explicitly shows the continuous DKT loop: fetching the student's current mastery score before generation and updating it after the response is delivered.

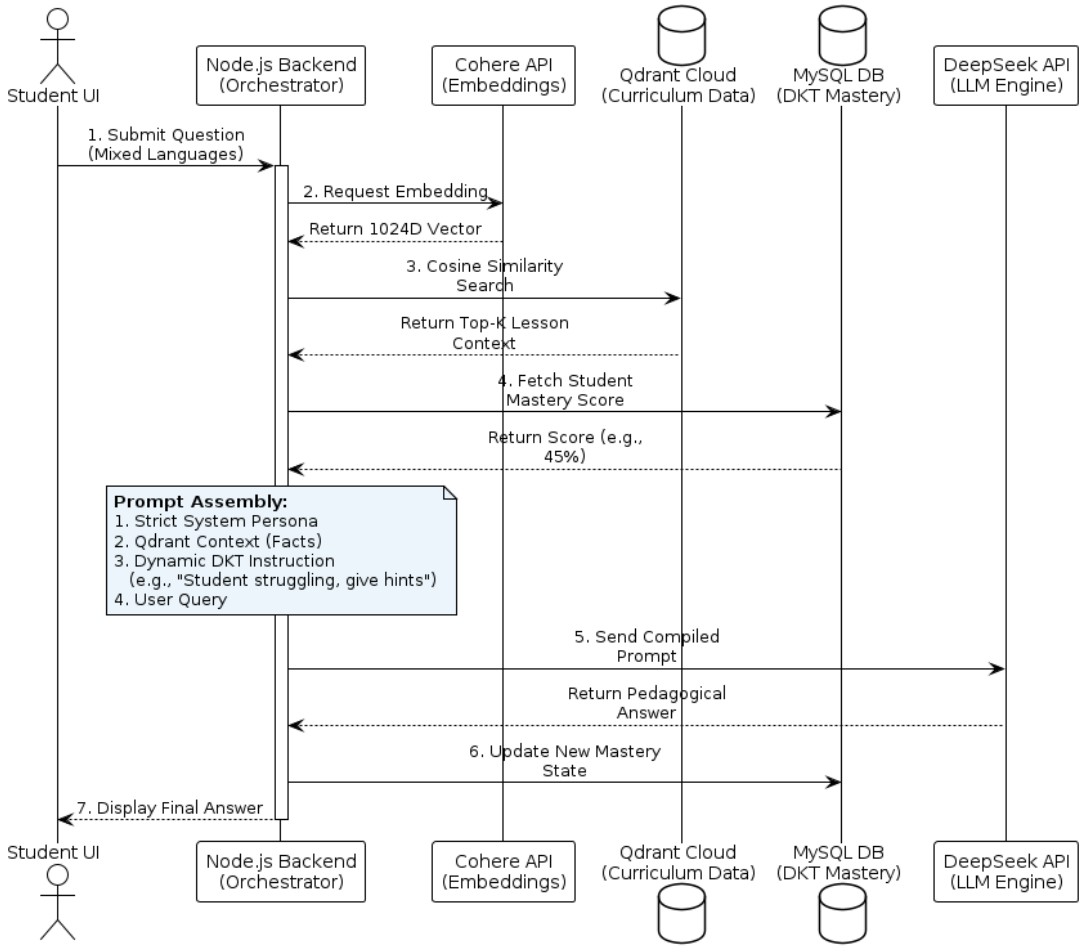


Figure 3.6: DeepBac sequence diagram: Node.js orchestration across Cohere, Qdrant, MySQL, and DeepSeek.

3.4 Dataset Construction and Processing

A Retrieval-Augmented Generation (RAG) system is fundamentally limited by the quality of its underlying knowledge base. Since standard open-source datasets are primarily in English and focus on general or Western knowledge, they are practically useless for the Algerian educational context. Therefore, constructing a localized, highly accurate dataset was a mandatory and foundational step in building DeepBac [45].

The construction and preparation of this data followed a rigorous pipeline tailored to overcome the specific challenges of bilingual scientific content and multi-model training (RAG, SFT, and DKT).

3.4.1 Data Sources and Collection

To guarantee pedagogical integrity and strict alignment with the Algerian curriculum, the raw data was populated using highly reliable and context-specific resources:

- **Curated Educational Resources:** High-quality lessons, comprehensive summaries, and specialized exercises created and freely published online by highly competent Algerian educators specifically for the BAC level.
- **Historical Exam Papers (Annales):** Past official BAC exam subjects covering recent years to capture the exact format, terminology, and difficulty of the national tests.
- **Annotated Solutions (Corrigés Types):** Official step-by-step grading rubrics and teacher-provided solutions that demonstrate the required methodology to obtain full marks.

Note on Fine-Tuning Data: For the specific task of Supervised Fine-Tuning (SFT) discussed in Section 3.4.1, we isolated a subset from the official History and Geography grading rubrics, manually extracting approximately 491 highly verified question-answer pairs to serve as the instructional dataset.

3.4.2 Advanced Data Extraction and Structuring

Processing Algerian scientific educational materials presents a unique challenge: the mixture of Arabic text (written right-to-left) with complex mathematical and physical formulas (written with Latin characters). Standard Optical Character Recognition (OCR) systems failed to produce satisfactory results under these conditions, often distorting equations and Arabic typography.

To overcome this bottleneck, we utilized large Vision-Language Models (Vision LLMs) to accurately extract the text and formulas from raw PDF documents. This AI-assisted extraction was followed by a rigorous manual review.

During this phase, the data was structured based on its intended pipeline:

- **For the RAG Knowledge Base:** We structured the extracted data into a strict logical hierarchy: *Subject* $\rightarrow$ *Lesson* $\rightarrow$ *Lesson Element*. All mathematical symbols were carefully transcribed into **LaTeX** format to prevent the retrieval LLM from hallucinating formulas.
- **For the SFT Pipeline:** The 491 History and Geography QA pairs were formatted into the structured Alpaca JSON format (comprising an `instruction` and an `output`) to explicitly teach the foundational LLM the exact pedagogical tone expected by BAC examiners.

3.4.3 Semantic Chunking and Vectorization

Before the hierarchical data could be injected into the RAG pipeline, it had to be optimized for machine retrieval:

1. **Element-Based Semantic Chunking:** Instead of arbitrarily splitting text by token count, which often destroys context, we utilized the natural hierarchy of our structured data. Each "Lesson Element" was treated as a single, self-contained chunk. To accommodate the context window limits and maintain retrieval speed, we enforced a strict maximum threshold of 1000 characters per chunk.

2. **Embeddings Generation:** These contextually rich chunks were passed through the Cohere multilingual embedding model (*embed-multilingual-v3.0*) to convert them into 1024-dimensional numerical vectors, ensuring high semantic understanding of both Arabic and French terms.

3.4.4 Dual-Database Architecture for Retrieval

To achieve both structural integrity and semantic search speed, DeepBac employs a dual-database architecture (see Figure 3.7). Traditional relational data (student profiles, DKT mastery scores, course metadata, and text chunks) are managed using a robust **MySQL** database. Conversely, the generated embedding vectors are stored in a dedicated cloud vector database (**Qdrant**). During a query, the backend retrieves the semantic context pointers from Qdrant, pulls the exact text and user historical state from MySQL, and seamlessly combines them before passing the unified prompt to the LLM Core.

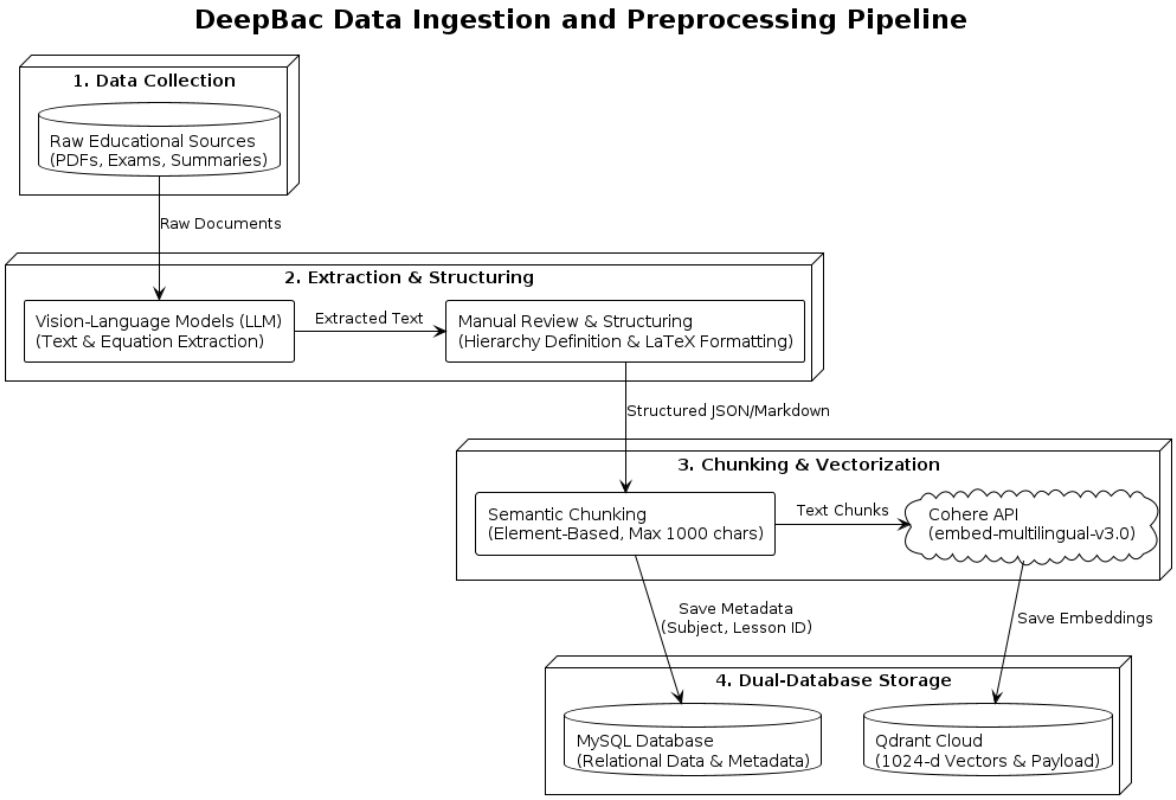


Figure 3.7: DeepBac data ingestion and preprocessing pipeline, illustrating Vision LLM extraction, LaTeX structuring, and dual-database storage (MySQL and Qdrant).

3.4.5 Synthetic Data Generation for Deep Knowledge Tracing (DKT)

Training the LSTM-based Deep Knowledge Tracing module required a fundamentally different type of dataset: sequential student interaction logs. Since DeepBac is a newly developed platform without thousands of active daily users at the time of architecture design, solving the "cold start" problem required the generation of a large-scale synthetic dataset.

To ensure the LSTM model learns realistic human cognitive patterns rather than pure statistical noise, we engineered a Python-based simulation engine that generated 1,000,000 interaction rows across 10,000 synthetic students. The generator parsed the actual database SQL schema (`bacheroes.sql`) to maintain absolute relational integrity regarding `subject_id`, `course_id`, and `course_part_id`.

The simulation engine heavily utilized principles from **Item Response Theory (IRT)** [46] and probabilistic modeling to simulate realistic learning curves:

- **Inherent Ability and Speed:** Each synthetic student u was assigned a baseline cognitive ability drawn from a Gaussian distribution ($Ability_u \sim \mathcal{N}(0, 0.4)$) and

a reaction speed factor drawn from a log-normal distribution. This approach follows the standard IRT convention of modeling latent student ability as a normally distributed variable [47].

- **The Learning Curve (Practice Effect):** A student's knowledge state is not static. As they progressed through the interactions, a time fraction variable (t_{frac}) dynamically increased their conceptual mastery. This practice effect is grounded in the well-established Power Law of Practice [48], which describes how performance improves as a power function of practice trials. The internal logit for a specific interaction was computed as:

$$Logit = Ability_u + 1.2 \times t_{frac} + \mathcal{N}(0, 0.1)$$

- **Probability of Mastery:** To map this logit into a strict probabilistic bound, we applied a standard sigmoid transformation:

$$P_{mastery} = \frac{1}{1 + e^{-Logit}}$$

- **Score and Response Time Simulation:** The probability of mastery was directly scaled to a standard grading format (0-100) with a slight Gaussian noise (± 2.5) to simulate human error. The addition of Gaussian noise to model inherent stochasticity in human performance is a common practice in synthetic data generation for educational applications [49, 50]. Furthermore, response times (`response_ms`) were mathematically penalized if the student's mastery was below 50%, accurately reflecting the cognitive hesitation of a struggling learner.

The inclusion of Gaussian noise throughout the simulation serves two purposes: First, it accounts for the non-deterministic nature of human performance students with identical ability levels do not always produce identical responses. Second, it ensures that the resulting synthetic data exhibits realistic variance, preventing the DKT model from overfitting to overly deterministic patterns. As noted by [49], adding random noise to synthetic data helps replicate the stochastic variability inherent in real-world observations.

By architecting this smoothed, mathematically grounded learning curve, the resulting dataset (`dkt_synthetic_interactions.csv`) allowed the LSTM model to converge properly and learn temporal mastery patterns, ensuring the Adaptive Learning Module was fully operational and validated prior to live deployment.

3.5 LLM Core Design and Evolution

The LLM Core is the primary reasoning engine of the DeepBac platform. Rather than adopting a single, static AI approach, the core's architecture was designed through an evolutionary, multi-phase methodology. This ensured an optimal balance between pedagogical accuracy, computational feasibility, and economic scalability. The engineering process is divided into the following key approaches:

3.5.1 Phase 1: Supervised Fine-Tuning (SFT) and QLoRA Mechanics

The adaptation of pre-trained General-Purpose LLMs to the specific pedagogical requirements of the Algerian Baccalaureate necessitates Supervised Fine-Tuning (SFT). Standard full-parameter fine-tuning is computationally prohibitive for models containing billions of parameters, especially under strict hardware constraints (e.g., consumer-grade GPUs with 12GB VRAM). To bypass this computational bottleneck and enable localized training, we adopted the **QLoRA (Quantized Low-Rank Adaptation)** framework.

1. The Mathematical Mechanism of LoRA and QLoRA:

Traditional fine-tuning updates all parameters in a dense layer. Low-Rank Adaptation (LoRA) instead freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture. Mathematically, for a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$, its update is constrained by representing it with a low-rank decomposition (see Figure 3.8) :

$$W = W_0 + \Delta W = W_0 + BA$$

where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, with the rank $r \ll \min(d, k)$. During training, W_0 remains frozen, and only A and B receive gradient updates.

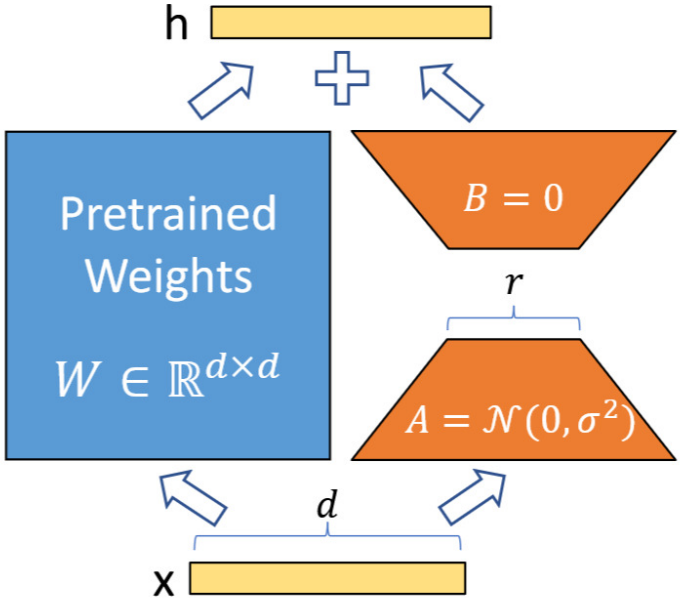


Figure 3.8: The architectural mechanism of Low-Rank Adaptation (LoRA). The pre-trained weights remain frozen, while only the low-rank matrices A and B are updated. Adapted from [27].

From LoRA to QLoRA: The Quantization Leap

While the structural architecture of QLoRA is identical to standard LoRA (as depicted in Figure 3.8), QLoRA pushes this efficiency further by introducing a critical optimization layer to address strict VRAM limitations. Standard LoRA typically keeps the frozen base model (W_0) in 16-bit precision. In contrast, QLoRA aggressively quantizes W_0 into a 4-bit NormalFloat (NF4) representation alongside double quantization (see Figure 3.9).

By quantizing the frozen base model W_0 to 4-bit precision while computing the low-rank updates (BA) in 16-bit (Brain Float 16), and utilizing Paged Optimizers to manage memory spikes, QLoRA effectively reduces the memory footprint by over 70% while preserving the model's predictive performance. This quantization leap was the definitive factor that enabled the local fine-tuning of 7B and 8B parameter models within a strict 12GB VRAM environment.

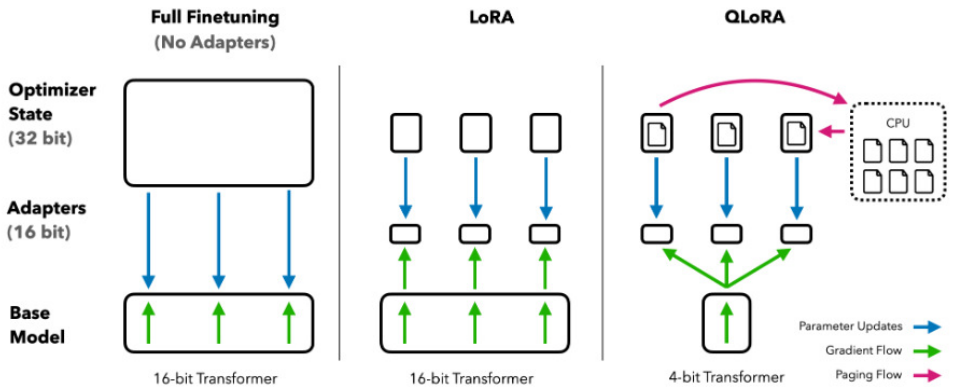


Figure 3.9: Comparison of fine-tuning methods: LoRA (16-bit base) versus QLoRA (4-bit NF4 base with 16-bit bf16 computation). Adapted from [51].

2. Selection of Foundational Models:

To identify the most robust base model for the DeepBac architecture, we systematically selected and fine-tuned a diverse portfolio of state-of-the-art open-weights models. Each model was chosen for specific architectural advantages:

- **ALLaM-7B-Instruct (ALLaM AI):** A specialized model pre-trained heavily on Arabic corpora. Its profound understanding of Arabic morphology, syntax, and cultural nuances theoretically makes it uniquely suited for the linguistic complexity of the Algerian curriculum.
- **LLaMA-3-8B (Meta):** Considered the industry standard for open-source LLMs. It provides a robust baseline for structural alignment, instruction-following, and general NLP reasoning.
- **Qwen2.5-7B (Alibaba Cloud):** Selected for its highly optimized tokenizer and exceptional multilingual capabilities. Its rich vocabulary allows for efficient processing of Arabic tokens.
- **DeepSeek-R1-Distill-Qwen-7B (DeepSeek):** A distilled reasoning model. It was strategically selected to test whether Chain-of-Thought (CoT) reasoning distillation inherently improves resistance to factual hallucinations compared to standard instruction-tuned models.

By subjecting these four distinct architectures to the exact same QLoRA pipeline and instructional dataset ($\sim$491 rows of official BAC History and Geography rubrics), we aimed to empirically determine which foundational architecture best aligns with strict factual constraints.

3.5.2 Phase 2: Production-Ready RAG via API (Current Live Version)

Given the hardware limitations for hosting large-scale models locally and the need for immediate, high-accuracy deployment, the current online production version of DeepBac relies exclusively on a Retrieval-Augmented Generation (RAG) architecture powered by a large foundation model via API.

- **DeepSeek API Integration:** We selected the DeepSeek API due to its exceptional reasoning capabilities (especially in math and logic) combined with highly cost-effective inference pricing.
- **Mechanism:** When a student queries the system, it retrieves the verified official Algerian curriculum from the vector database and feeds it to the DeepSeek model. This ensures factual, hallucination-free answers without the immediate need for extensive fine-tuning.

3.5.3 Phase 3: The Hybrid Architecture (RAG + Fine-Tuning)

The ultimate architectural goal of DeepBac is a hybrid system. Once the proprietary dataset reaches a sufficient volume, we will deploy a Fine-Tuned model specifically trained on the "tone and pedagogical style" of Algerian teachers, augmented by RAG for fetching exact "facts and formulas". This hybrid approach minimizes latency and API costs while maximizing domain specificity.

3.5.4 Universal Layer: Advanced Prompt Engineering

Regardless of the underlying approach (Local Fine-Tuning, API-based RAG, or Hybrid), advanced Prompt Engineering serves as the universal control layer across the entire platform. We utilize structured "System Prompts" and Few-Shot prompting to explicitly instruct the model to act as a certified Algerian teacher, process mixed languages (Arabic, French, Darja), and break down complex solutions into step-by-step methodologies.

3.6 Adaptive Learning Module

While the RAG pipeline ensures that the AI understands the BAC curriculum, the system also needs to understand the student. A standard LLM treats every user exactly the same, answering a struggling student and an advanced student with the exact same tone and complexity. To solve this limitation, DeepBac incorporates a dedicated Adaptive Learning Module that works in tandem with the LLM Core.

The primary goal of this module is to personalize the educational experience based on the individual learner's historical performance. It operates through three main mechanisms:

3.6.1 Continuous Deep Knowledge Tracing (DKT) via LSTM

Instead of relying on a single initial evaluation exam or simple rule-based percentage tracking, DeepBac employs a sophisticated Deep Knowledge Tracing (DKT) model powered by a Long Short-Term Memory (LSTM) neural network [52].

This predictive model operates independently from the LLM Core. The Algerian BAC curriculum is structurally mapped into highly granular "micro-skills" (e.g., specific sub-topics within a course). Every time a student interacts with the platform, the backend extracts sequential features from this event specifically the targeted micro-skill, the normalized evaluation score, and the response time.

Because the learning process is inherently sequential, the LSTM processes these interactions as time-series data. It updates a hidden state representing the student's latent cognitive profile and predicts a continuous mastery score (ranging from 0.0 to 1.0) for subsequent micro-skills. Over time, the model builds a highly accurate, multidimensional mathematical map of the student's exact strengths and weaknesses, dynamically updating their proficiency after every single interaction.

3.6.2 DKT-LLM Integration Cycle (Dynamic Adaptation)

This is where the predictive intelligence of the DKT (LSTM) directly controls the generative power of the LLM Core. This integration forms a continuous adaptation cycle that operates at two crucial touchpoints:

1. Exercise Generation (Pre-Submission): When a student requests a practice exercise, the backend queries the DKT model to fetch the predicted mastery score for that specific topic. If the DKT profile indicates a low mastery score (e.g., 0.35) in Mathematics, the module injects a hidden system instruction into the LLM's prompt, forcing it to generate a foundational, step-by-step exercise. Conversely, a high mastery score (e.g., 0.85) triggers the generation of a highly complex, exam-level problem [22]. This ensures the student is consistently challenged within their Zone of Proximal Development.

2. Pedagogical Evaluation (Post-Submission): The adaptation shapes not only the questions but also the feedback. When the student submits an answer, the evaluation generated by the LLM is again tailored to their DKT profile. For a struggling learner, the system provides heavy "scaffolding" (e.g., providing the first part of a formula and asking them to complete it). For an advanced student, the feedback might simply be a subtle hint pointing out a minor calculation error [53].

This dynamic, dual-phase feedback loop transforms DeepBac from a static question bank into a truly reactive and empathetic intelligent tutor.

3.7 Native Multilingual and Code-Switching Handling

Historically, early embedding models and Dense Passage Retrieval (DPR) systems were trained predominantly on English corpora. To handle non-English queries, standard Natural Language Processing (NLP) architectures typically relied on a "Translate-then-Retrieve" pipeline [54]. In this classic approach, the system would first detect the user's language, translate the query into English via an external machine translation API, retrieve the relevant documents, generate an answer, and finally translate it back to the user's original language.

For DeepBac, we intentionally discarded this outdated architecture. Adding a dedicated translation middleware introduces high systemic latency, increases API costs, and creates multiple points of failure. More importantly, automated translation models severely struggle with the Algerian dialect (Darja) and scientific code-switching the frequent alternation between Arabic and French terminology within the same sentence. Translating a query like "*Mafhamtch la limite ta3 hadi el fonction*" into a single target language completely destroys its contextual and pedagogical nuance [30] (see Figure 3.10).

Instead, DeepBac relies on the **Native Multilingual Processing** capabilities of modern foundation models. As illustrated in Figure 3.10, our architecture maps multiple languages directly into a shared semantic vector space [55]. This allows the system to handle mixed inputs implicitly, without any translation layers, through the following streamlined flow:

1. **Direct Mixed Input:** A student inputs a query freely mixing Darja, French, and formal Arabic.

2. **Semantic Retrieval:** The Cohere embedding model (*embed-multilingual-v3.0*) natively understands the semantic intent of the mixed query and successfully matches it with the formal Arabic/French documents in the Qdrant Vector Database.

3. **Prompt-Enforced Output:** To ensure the final output is pedagogically professional, the DeepSeek LLM is constrained via the System Prompt to always reply in formal, clear Arabic, while retaining necessary scientific terms in their original French format [56].

By leveraging native model capabilities, the DeepBac architecture successfully bypasses the latency bottlenecks of the classic pipeline, remaining lightweight, fast, and highly resilient to the chaotic nature of natural student conversations.

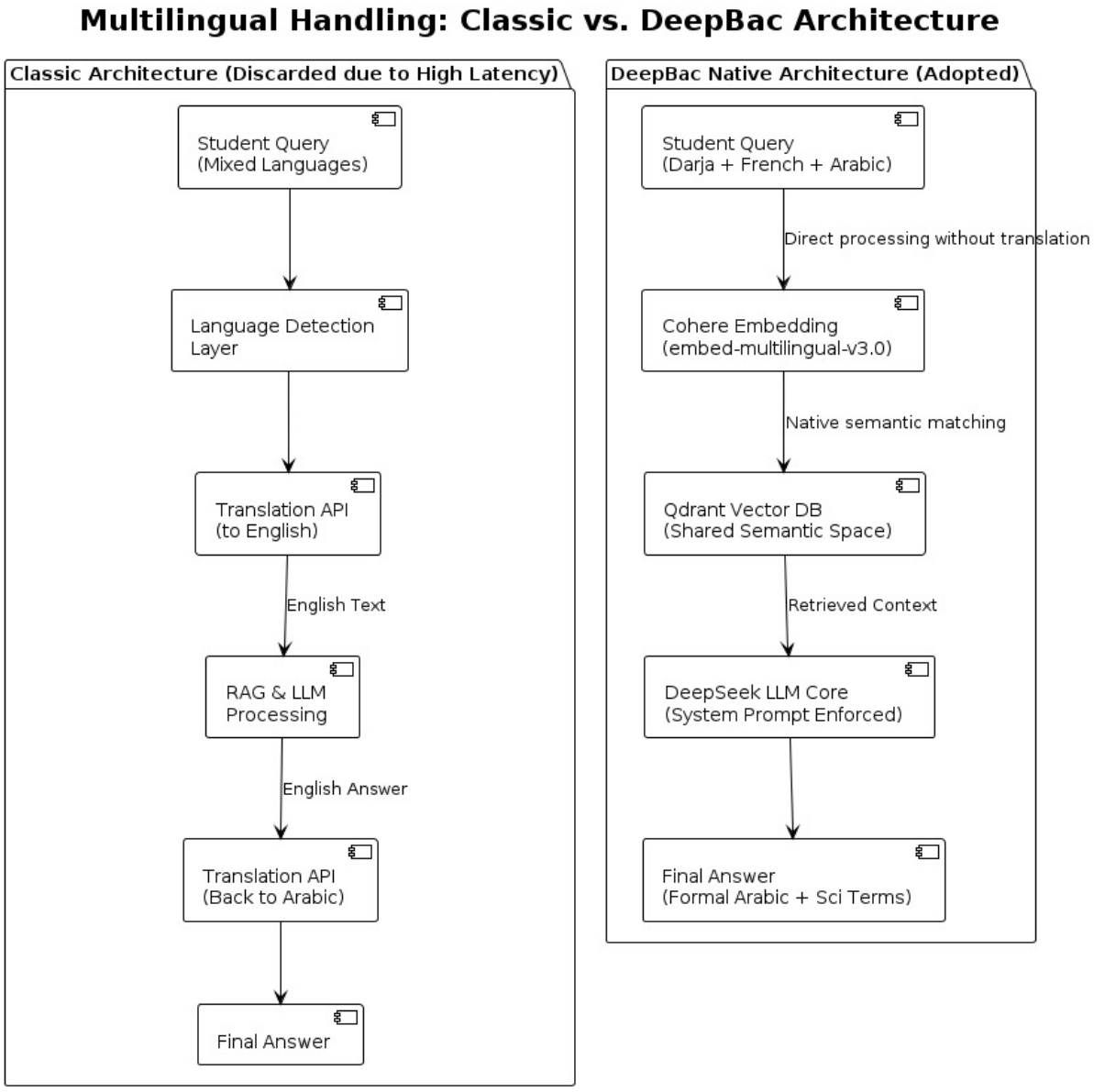


Figure 3.10: Multilingual handling architectures: Latency-heavy classic approach (translation middleware) versus DeepBac native approach (shared semantic vector space).

3.8 Security and Data Privacy Considerations

The integration of Large Language Models (LLMs) and the management of student academic records necessitate a robust security framework to ensure data integrity, user privacy, and pedagogical safety. For a platform like **DeepBac**, security is addressed across three primary dimensions: system-level security, data privacy compliance, and AI-specific security.

3.8.1 Data Privacy and Legal Compliance

Since the system handles sensitive student information, including academic performance and DKT learning patterns, it is designed to align with international standards such as the General Data Protection Regulation (GDPR) and, more specifically, **Algerian Law No. 18-07** relating to the protection of individuals in the processing of personal data [57]. Key measures include:

- **Data Minimization:** Collecting only the necessary attributes required for the knowledge-tracing module and user identification.
- **Encryption at Rest and in Transit:** Implementing Advanced Encryption Standard (AES-256) for the MySQL and Qdrant databases, alongside Transport Layer Security (TLS) for all API communications.

3.8.2 Authentication and Access Control

To prevent unauthorized access to administrative functions and student profiles, the architecture employs a strict access control model:

- **Identity Management:** Secure authentication is handled via JSON Web Tokens (JWT), ensuring that each request to the backend is validated and session-bound.
- **Password Hashing:** Implementing strong cryptographic hashing algorithms (e.g., BCrypt) to ensure that plain-text passwords are never stored in the relational database [58].
- **Infrastructure Security:** External API credentials (such as Cohere and DeepSeek keys) are securely isolated within the Node.js backend environment variables, preventing any client-side exposure.

3.8.3 AI-Specific Security and Adversarial Robustness

Educational LLMs are susceptible to unique vulnerabilities, most notably *Prompt Injection* attacks, where users attempt to bypass pedagogical constraints to elicit non-educational or harmful content [59]. **DeepBac** mitigates these risks through:

- **System Prompt Hardening:** Utilizing restrictive system-level instructions that define the strict boundaries of the AI tutor's persona.
- **Contextual Grounding (RAG):** By forcing the generative LLM to base its answers exclusively on the semantic context retrieved from the Qdrant vector database, the risk of hallucinations or out-of-curriculum responses is drastically minimized.

- **Output Filtering:** Implementing a verification layer to detect and sanitize potential inappropriate content before it reaches the student interface [60].

3.9 Conclusion

In this chapter, we successfully transitioned from the theoretical concepts discussed in our literature review to a concrete, technical blueprint for the DeepBac platform. Rather than relying on a generic, closed-source chatbot, we proposed a highly specialized, modular architecture tailored specifically for the Algerian Baccalaureate environment.

We detailed the internal mechanics of the system, demonstrating how the frontend, backend, and AI models interact seamlessly. By integrating a Retrieval-Augmented Generation (RAG) pipeline, we ensured that the LLM Core is strictly grounded in official curriculum data, effectively eliminating the risk of AI hallucinations. Furthermore, the inclusion of the Adaptive Learning module (DKT) guarantees that the platform does not offer a static experience, but rather dynamically adjusts to each student's unique proficiency level. We also addressed the linguistic reality of Algerian students by leveraging the native multilingual capabilities of modern models to handle code-switching smoothly.

Finally, by carefully selecting an efficient, modern technological stack centered around Node.js, dual-database architecture (MySQL and Qdrant), and specialized AI endpoints (DeepSeek and Cohere) we ensured that this architecture is not just theoretically sound, but practically deployable, highly scalable, and cost-effective.

With the system design completely defined and the technological stack established, the foundation is set. The following chapter (Chapter 4: Implementation) will cover the actual realization of this architecture. It will showcase the developed user interfaces, the execution of the core algorithms, and a thorough evaluation of the platform's performance in practical educational scenarios.

Chapter 4

Implementation

4.1 Introduction

After presenting in the previous chapter the theoretical architecture and system design, this chapter shifts focus toward the practical realization of the DeepBac platform. Here, we transition from conceptual blueprints to a fully functional application.

The chapter begins by detailing the development environment and the technological stack. We outline the specific tools, frameworks, and databases chosen to construct a robust and scalable architecture, justifying these selections based on the project's unique operational constraints.

Subsequent sections delve into the core programming and integration phase. We focus specifically on the end-to-end realization of the dual-database Retrieval-Augmented Generation (RAG) pipeline, the synchronization of external AI endpoints, and the practical embedding of the Deep Knowledge Tracing (DKT) logic into the backend server and user interface.

4.2 Development Methodology: An Agile Approach

The development of AI-driven educational software differs significantly from traditional software engineering. The non-deterministic nature of Large Language Models (LLMs) requires constant experimentation, prompt refinement, and continuous testing to ensure pedagogical accuracy. For this reason, the development of **DeepBac** was guided by the Agile methodology, specifically utilizing an iterative and incremental approach [62].

Unlike the traditional Waterfall model, which relies on rigid, sequential phases, Agile provides the necessary flexibility to adapt to unforeseen technical challenges such as mitigating LLM hallucinations or adjusting the Retrieval-Augmented Generation (RAG) pipeline based on real-time generation results [63].

4.2.1 The Iterative Development Cycle (Sprints)

The project lifecycle was divided into several focused, time-boxed iterations (sprints). The primary goal was to rapidly develop a Minimum Viable Product (MVP) to validate the core AI logic before integrating complex tracking features. The development process was structured as follows (see Figure 4.1):

- **Sprint 1: System Foundation and Architecture:** This initial phase focused on establishing the core technical infrastructure. It included designing the relational database schema (MySQL), configuring the Node.js backend server, developing RESTful API endpoints, and building the foundational User Interface (UI) for the interactive chat.

- **Sprint 2: Core AI and Dual-Database RAG Integration:** As the most critical technical sprint, this phase was dedicated to constructing the RAG data pipeline. This involved segmenting the Algerian Baccalaureate curriculum, generating embeddings via the Cohere API, and vectorizing them into the Qdrant database. Subsequently, the backend was integrated with the DeepSeek LLM to ensure the model could retrieve and generate contextually accurate educational content.

- **Sprint 3: Adaptive Learning (DKT) and Security Implementation:** Once the core conversational agent was stable, development shifted toward personalization and security. This included integrating the Deep Knowledge Tracing (DKT) module for continuous student profiling, implementing the Role-Based Access Control (RBAC) system via JWT, and applying the system prompt hardening measures discussed in Chapter 3 to prevent prompt injection.

- **Sprint 4: Pedagogical Testing and Prompt Refinement:** The final iterations were highly experimental, focusing on pedagogical validation. This phase required extensive *Prompt Engineering* to refine the AI tutor's persona, strictly enforce curriculum boundaries, and optimize the system's ability to handle the complex trilingual code-switching (Arabic, French, and Darja) characteristic of the local context.

By adopting this Agile workflow, the continuous integration of complex AI components remained manageable, allowing for rapid debugging and continuous system calibration throughout the development lifecycle.

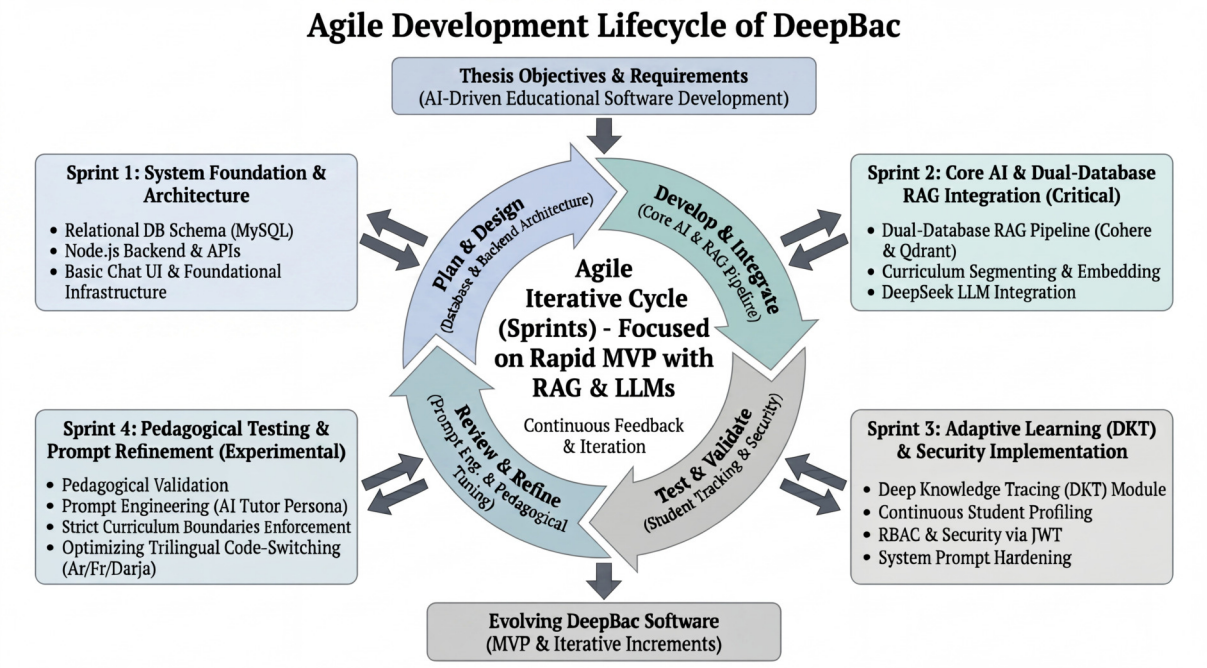


Figure 4.1: DeepBac Agile development lifecycle, illustrating the four iterative sprints: system foundation, dual-database RAG integration, adaptive personalization, and pedagogical optimization.

4.3 Development and Production Environment

Building DeepBac required a pragmatic approach to technology selection. Instead of relying on standard, resource-heavy Python AI frameworks, the architecture was designed from a web engineering perspective. This ensured the platform was fast, scalable, and easy to deploy on a standard production server.

4.3.1 Prototyping and Local Setup

Initial experiments and system logic tests were conducted locally using a machine equipped with an NVIDIA RTX 3060 GPU (12GB VRAM) running Windows 11 with Windows Subsystem for Linux (WSL2). This environment was strictly utilized for data preprocessing, formatting the BAC curriculum, and testing early prompt structures and embedding logic before pushing the codebase to the cloud.

4.3.2 Production Server and Backend Logic

For the live, public-facing platform, the system is hosted on a **Hostinger** server and is globally accessible online at `https://deepbac.com`. The entire backend routing, API orchestration, and core business logic were developed using **Node.js**.

Node.js was selected specifically for its event-driven, non-blocking I/O model [64]. In a RAG-based tutoring system, the server spends a significant amount of time waiting for external AI models to generate text or embeddings. Node.js handles these asynchronous API calls efficiently without blocking the main thread, allowing the Hostinger server to manage multiple concurrent student sessions smoothly.

4.3.3 The Hybrid AI Tech Stack

The RAG pipeline and data management in DeepBac rely on a highly specialized hybrid technology stack (Figure 4.2), dividing responsibilities among dedicated services:

- **MySQL Database (Relational State): MySQL** serves as the central hub for structured data. It securely stores user profiles, manages course metadata, and continuously tracks student mastery scores for the Adaptive Learning (DKT) module.
- **Qdrant Cloud (Vector Store):** To enable semantic search, DeepBac utilizes **Qdrant Cloud**. Curriculum chunks are stored here as high-dimensional vectors, allowing the backend to perform rapid cosine similarity searches to retrieve the most relevant educational context.
- **Cohere API (Embeddings):** Before data is stored in Qdrant or a query is searched, the text is passed to the **Cohere** API (*embed-multilingual-v3.0*). This model natively handles the bilingual nature of the Algerian curriculum, converting Darja, Arabic, and French text into accurate 1024-dimensional vectors.
- **DeepSeek API (Generation Core):** The Hostinger server does not run heavy LLMs locally. Once the Node.js backend pulls the right context from Qdrant and the student's level from MySQL, it orchestrates a secure API request to **DeepSeek**. DeepSeek was chosen for its exceptional reasoning capabilities in mathematics and its native support for Arabic and French, making it the ideal pedagogical engine.

This setup Node.js orchestrating the traffic, MySQL and Qdrant managing the dual-database knowledge, and Cohere/DeepSeek powering the AI creates a highly resilient, modern production environment.

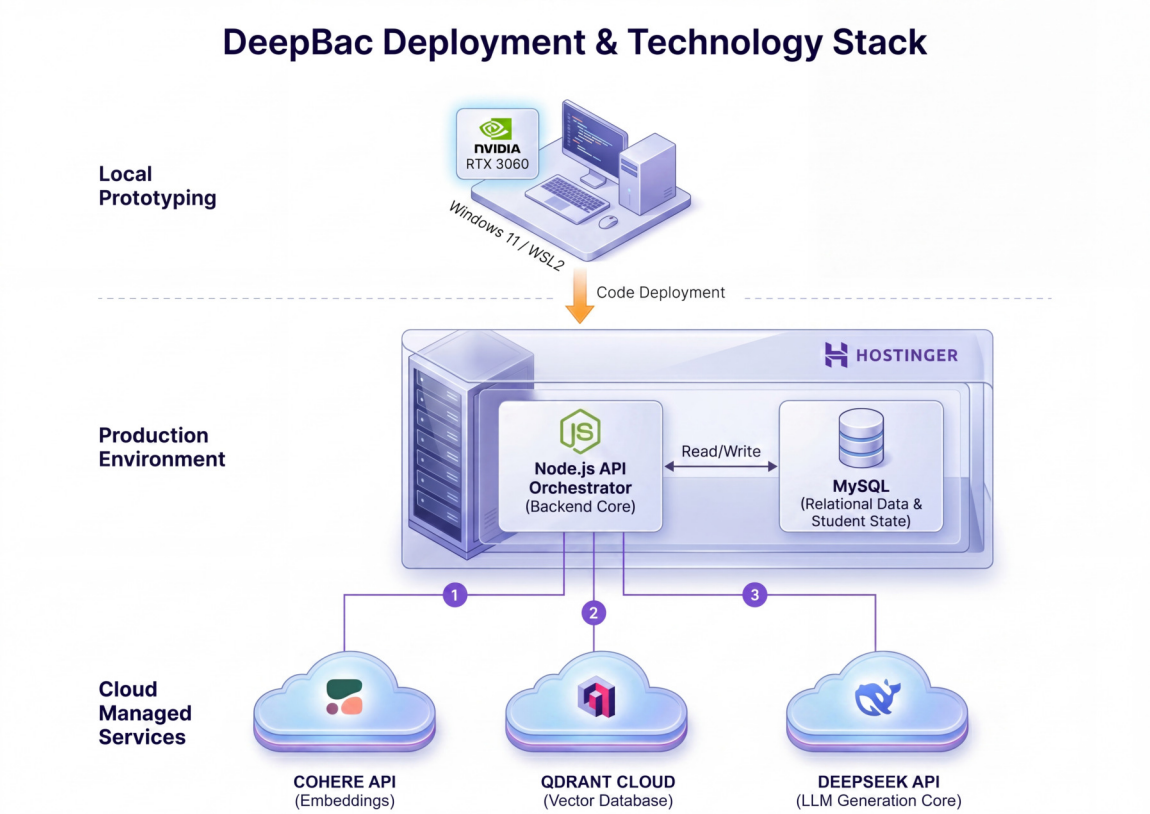


Figure 4.2: DeepBac deployment blueprint: End-to-end infrastructure flow from local development to Hostinger production server and cloud AI services.

4.4 System Implementation

With the server environment and hybrid database architecture configured, the next step was programming the actual backend logic. The core challenge was orchestrating the asynchronous flow of data between the student's interface, the dual databases (MySQL and Qdrant), and the external AI APIs. This section breaks down how the main functionalities of DeepBac were implemented in Node.js.

4.4.1 Implementing the Dual-Database RAG Pipeline

Unlike traditional monolithic systems, DeepBac decouples relational student data from semantic curriculum data. When a student initiates a session and asks a question (e.g., mixing Darja and French math terms), the Node.js backend executes a strictly ordered retrieval pipeline:

1. **Vectorization:** The raw user query is first sent to the **Cohere API**, which converts the multilingual text into a dense mathematical vector.

2. **Semantic Search:** This vector is then used to query the **Qdrant Cloud** database. Using cosine similarity, Qdrant retrieves the Top-K most relevant chunks from the official Algerian BAC curriculum, regardless of the exact keywords used by the student [8].

This vector-based approach guarantees high retrieval accuracy for chaotic student inputs, outperforming rigid SQL-based keyword matching.

4.4.2 API Orchestration and Prompt Assembly

Once the factual curriculum data is retrieved from Qdrant, the system must securely communicate with the generative AI. This is handled via asynchronous HTTP requests to the **DeepSeek API**.

The most critical backend function is the "Prompt Assembly" algorithm. Before making the API call, Node.js constructs a single, comprehensive text string. This compiled prompt consists of three isolated blocks:

1. **The System Persona:** Hardcoded instructions enforcing the AI to act as a strict Algerian BAC examiner, forbidding it from answering non-educational topics.

2. **The Grounded Context:** The exact lesson chunks and pedagogical examples just pulled from Qdrant.

3. **The User Query:** The actual question typed by the student.

By binding these three elements together, the DeepSeek model is algorithmically constrained to answer using *only* the provided curriculum data, effectively eliminating out-of-domain hallucinations.

4.4.3 Coding the Adaptive Learning Logic (DKT Integration)

The platform's Deep Knowledge Tracing (DKT) logic is deeply integrated into the backend execution flow. Every time a student completes an interaction, the Node.js server evaluates the pedagogical outcome and updates the student's mastery matrix stored securely in **MySQL**.

Before assembling the prompt for the next generated question, the backend queries MySQL to check the student's historical cognitive state for that specific chapter. The adaptation is implemented via dynamic prompt injection:

- If the MySQL query reveals a mastery rate below 50%, the server dynamically appends a supportive instruction to the DeepSeek prompt: *"The student is struggling. Provide a simple, step-by-step foundational exercise with guiding hints."*

- If the mastery rate exceeds 80%, the appended instruction shifts to challenge the student: *"The student shows high proficiency. Generate a complex, multi-layered problem matching the exact difficulty and format of the final BAC exam."*

This dynamic prompt manipulation ensures the system continuously tailors its difficulty curve without requiring multiple separate AI models or heavy retraining, fulfilling the promise of personalized education.

4.5 User Interface and Frontend Implementation

While establishing a robust backend is critical, the pedagogical success of DeepBac heavily relies on an accessible and engaging user interface. Since Algerian high school students depend primarily on their smartphones for digital learning, the interface was designed with a strict mobile-first philosophy. Instead of developing a resource-heavy native application, the frontend was implemented as a Progressive Web Application (PWA) driven by a Single Page Application (SPA) architecture.

4.5.1 Building the React-Based Architecture

The frontend was developed using **React 18**, leveraging a component-driven architecture to ensure modularity and maintainability. Navigation across the platform is handled by **React Router**, enabling seamless transitions between modules without disruptive page reloads. For the structural layout and UI components, **Bootstrap 5** was integrated alongside **React Bootstrap**, guaranteeing a robust and highly responsive mobile-first experience.

To orchestrate complex data flows between the client and the Node.js backend, we implemented **TanStack React Query**. This powerful library handles asynchronous state management, providing aggressive out-of-the-box caching, background updates, and automatic retries, which drastically reduces load times and ensures a smooth experience even on unstable mobile networks.

4.5.2 The Interactive Chat Interface and LaTeX Rendering

The core educational experience is localized within the interactive chat module. Communication with the backend is managed using **Axios** for secure RESTful HTTP requests, alongside the **Socket.IO client** for real-time bidirectional interactions during live sessions.

A major technical hurdle during this phase was the accurate rendering of AI outputs. The LLM engine returns raw Markdown text interspersed with complex LaTeX mathematical notation. Displaying this raw string would render equations completely

unintelligible to a high school student. To resolve this, the frontend integrates a specialized rendering pipeline using **react-markdown**, augmented with **remark-math** and **rehype-katex**. Before any message is painted to the Document Object Model (DOM), this pipeline parses the raw string, securely converts the Markdown into styled HTML, and accurately renders the LaTeX tags into formatted, highly readable mathematical symbols.

4.5.3 Integrating the Adaptive Feedback UI

The frontend is not merely a static display; it acts as a visual extension of the Deep Knowledge Tracing (DKT) module. Based on the mastery scores returned from the backend, the UI dynamically reconfigures itself. To make complex DKT data comprehensible to both students and teachers, we integrated **Chart.js** (via `react-chartjs-2`) to render intuitive visual dashboards of learning progress and skill mastery.

Furthermore, if the backend flags a student as "struggling" on a specific topic, the frontend conditionally renders additional pedagogical scaffolds, such as "Request Hint" buttons. To ensure these dynamic UI state changes feel natural and engaging rather than jarring, **Framer Motion** was employed to orchestrate fluid, physics-based animations across the interface. This seamless synchronization ensures that the frontend actively participates in the personalized learning cycle.

4.6 Database Implementation and API Architecture

While the previous sections detailed the dynamic logic of the system, this section focuses on the structural foundation. A well-designed hybrid database architecture and a clean Application Programming Interface (API) are what allow the frontend, the persistent storage, and the AI core to communicate efficiently without data loss or latency.

4.6.1 Hybrid Data Modeling: MySQL and Qdrant

Because DeepBac employs a dual-database Retrieval-Augmented Generation (RAG) approach, data is carefully separated based on its nature. Relational data is strictly normalized in MySQL (Figure 4.3), while semantic representations are managed by Qdrant using a decoupled storage strategy.

The core entities are distributed as follows:

- **Users and Authentication (MySQL):** Stores foundational student information, authentication credentials (securely hashed via BCrypt), and their specific academic stream (e.g., Mathematics, Experimental Sciences).

- **Curriculum Data & Vector Indexing (Decoupled Storage):** MySQL manages the strict hierarchy of the Algerian curriculum (`Subjects` → `Chapters` → `Lessons`), as well as the actual text content of the pedagogical chunks. **Qdrant**, on the other hand, only stores the 1024-dimensional vectors representing these chunks, along with their corresponding MySQL IDs (Metadata). During a search, Qdrant performs the cosine similarity matching and returns the IDs of the top-K vectors. The Node.js server then uses these IDs to fetch the actual text chunks directly from MySQL. This pointer-based approach keeps the vector database lightweight and extremely fast.

- **Deep Knowledge Tracing State (MySQL):** This table serves as the persistent memory for the Adaptive Learning Module. It logs every interaction, recording the `user_id`, `chapter_id`, the calculated mastery probability (score), and a timestamp.

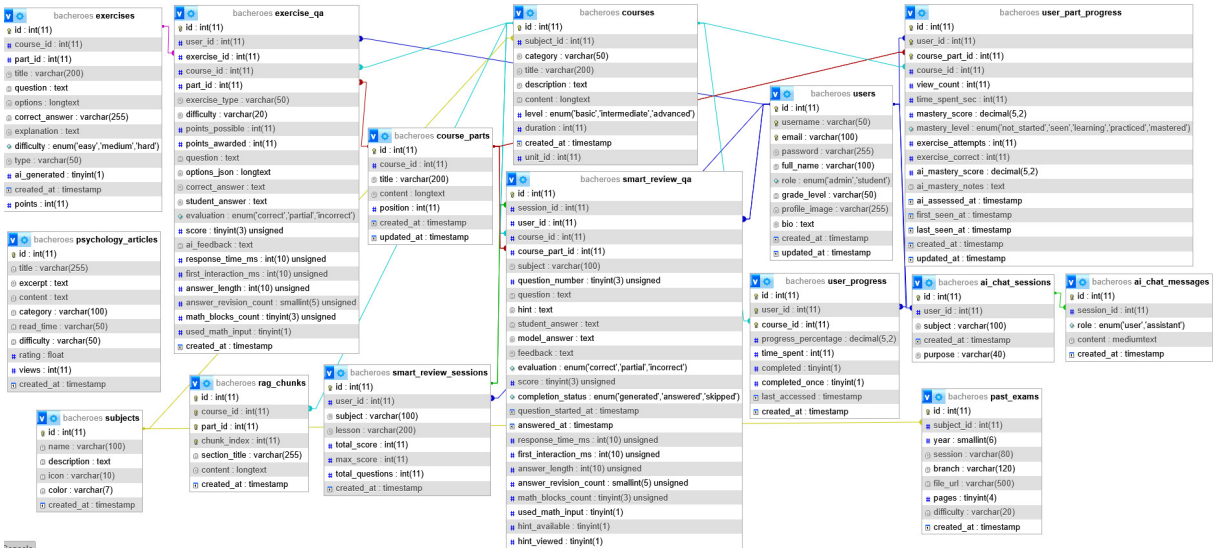


Figure 4.3: DeepBac MySQL database relational schema.

4.6.2 RESTful API Orchestration

To connect the dual databases and external AI services with the Progressive Web App (PWA), a robust RESTful API was engineered using Express.js on the Node.js runtime. The API acts as a strict orchestrator, securely routing payloads and enforcing system rules.

The backend exposes several critical endpoints:

- `POST /api/auth/login`: Handles user authentication, validates credentials, and issues secure JSON Web Tokens (JWT) for stateless session management.

- `GET /api/curriculum/{stream_id}`: Retrieves the structured JSON tree of subjects and chapters from MySQL, allowing the student to browse the syllabus smoothly

in the UI.

- `POST /api/chat/ask`: The most complex orchestration endpoint. Upon receiving a user's prompt, this route sequentially: (1) calls the Cohere API for vectorization, (2) queries Qdrant for the top semantic IDs, (3) fetches the actual text chunks from MySQL using those IDs, (4) fetches the student's DKT score, (5) assembles the strict system prompt, and (6) awaits and streams the DeepSeek API response back to the frontend.

By decoupling the relational state from the vector search and strictly defining the API contracts, the backend remains lightweight, highly scalable, and capable of handling future expansions to the BAC syllabus.

4.7 Conclusion

This chapter documented the transition of the DeepBac platform from a theoretical blueprint into a fully functional software application, currently deployed and live at https://deepbac.com. Driven by an Agile methodology, we bypassed standard, resource-heavy Python deployments in favor of a highly responsive Node.js backend. This orchestrator effectively bridged our frontend with a hybrid data storage model, utilizing MySQL to manage structured student progress and curriculum metadata.

A major milestone of this implementation was the successful execution of the decoupled RAG pipeline. By utilizing Qdrant for semantic vector searches and the Cohere for multilingual embeddings, we established a highly accurate retrieval mechanism. Offloading the heavy natural language generation to the DeepSeek API allowed the system to overcome local hardware limitations without sacrificing speed or pedagogical reasoning quality. Furthermore, the frontend was successfully deployed as a Progressive Web App (PWA) and prepared as a Trusted Web Activity (TWA) for native accessibility complete with KaTeX integration to flawlessly render the complex mathematical equations of the Algerian curriculum.

With the coding, API orchestration, and UI integration phases fully completed, the DeepBac system is now operational. However, engineering an educational platform is only the first step; proving that it actually teaches effectively is the ultimate goal. The next chapter is entirely dedicated to putting DeepBac through rigorous testing. We will evaluate its pedagogical accuracy, measure its resistance to hallucinations during Darja-French code-switching, and critically discuss the limitations of the current implementation.

Chapter 5

Evaluation, Results, and Discussion

5.1 Introduction

The ultimate measure of any educational technology is not the complexity of its underlying code, but the accuracy, reliability, and pedagogical usefulness of its output. This final chapter shifts the focus from software engineering and system architecture to empirical evaluation. Now that the DeepBac platform is fully implemented and operational, it must be rigorously tested against the strict academic standards of the Algerian Baccalaureate.

In this chapter, we present a series of qualitative and quantitative tests designed to evaluate the core functionalities of the system. First, we examine the system's pedagogical accuracy by submitting actual exam questions from subjects like History and Geography, comparing the AI's step-by-step reasoning to the official Ministry of National Education grading rubrics. Next, we evaluate the effectiveness of our hybrid Retrieval-Augmented Generation (RAG) pipeline to verify its ability to ground the AI in local curriculum facts and eliminate out-of-domain hallucinations. Furthermore, we assess the Adaptive Learning module (Deep Knowledge Tracing), testing whether the system successfully scales the difficulty of its feedback based on the student's historical mastery.

Beyond presenting these successful outcomes, this chapter also provides a critical discussion of the platform's real-world performance. We analyze system latency (response times), the robustness of the Darja-French code-switching comprehension, and finally, acknowledge the current technical limitations of the platform to pave the way for future research and continuous improvement.

5.2 Multi-Dimensional Evaluation Metrics (SFT Phase)

Evaluating Generative AI in an educational context requires moving beyond simple accuracy percentages. To rigorously assess the fine-tuned models, we implemented a

multi-dimensional evaluation strategy encompassing training convergence, lexical overlap, contextual embeddings, and advanced semantic judgment.

5.2.1 Training and Validation Performance (Loss Metrics)

Before assessing the generated text, it is crucial to evaluate the models' convergence during the Supervised Fine-Tuning (SFT) phase. We tracked both the Training Loss and Evaluation Loss to ensure the models successfully learned the dataset patterns without overfitting.

Table 5.1: Training and Evaluation Loss across Fine-Tuned Models

Model	**Train Loss**	**Eval Loss**
`ALLaM 7B`	0.599	1.312
`LLaMA 3 8B`	0.688	1.173
`Qwen 2.5 7B`	0.7	1.12
`DeepSeek R1 7B`	1.557	1.841

Analysis of Training Convergence:
As detailed in Table 5.1, `ALLaM 7B` achieved the lowest Training Loss, likely due to its extensive pre-training on Arabic corpora, allowing it to rapidly adapt to the curriculum's terminology. Conversely, `Qwen 2.5 7B` achieved the lowest Evaluation Loss, indicating superior generalization capabilities on unseen validation data. The higher loss values observed in `DeepSeek R1 7B` suggest that distilled reasoning models (Chain-of-Thought) may require significantly larger datasets to effectively map their reasoning traces to specific formatting constraints.

5.2.2 Classical Lexical Metrics (BLEU and ROUGE-L)

To measure the models' strict adherence to the official Ministry of Education terminology (*Corrigé Type*), we utilized two fundamental NLP metrics:

- **BLEU (Precision-Focused):** Evaluates how many of the n-grams generated by the model actually appear in the reference text.
- **ROUGE-L (Recall-Focused):** Measures the Longest Common Subsequence (LCS) to confirm the model successfully retrieved all necessary key facts and dates.

Table 5.2: Lexical Overlap Evaluation (BLEU and ROUGE-L)

Model	BLEU	ROUGE-L
`Qwen 2.5 7B`	0.022	0.064
`DeepSeek R1 7B`	0.007	0.057
`ALLaM 7B`	0.033	0.081
`LLaMA 3 8B`	0.019	0.086

Analysis of Lexical Performance:
A detailed examination of Table 5.2 reveals that `ALLaM 7B` achieved the highest BLEU score (0.033), suggesting a superior ability to generate exact Arabic terminologies that align with the official Algerian rubrics. This can be attributed to its specialized pre-training on high-quality Arabic corpora. On the other hand, `LLaMA 3 8B` exhibited the highest ROUGE-L score (0.086), indicating a better capability in recalling long-range factual sequences and structural patterns required by the curriculum.

However, the absolute scores for BLEU and ROUGE-L remain significantly low across the entire portfolio. This result is a well-documented phenomenon in modern LLM evaluation; because generative models are inherently probabilistic, they tend to produce semantically accurate answers using synonyms or varied syntactic structures, which are heavily penalized by these strict lexical exact-match metrics. The particularly low BLEU score of `DeepSeek R1 7B` (0.007) is especially noteworthy; it indicates that the model's reasoning-based (Chain-of-Thought) generation style tends to be more verbose or structured differently than the concise, rigid format of the reference grading keys. This divergence reinforces the necessity of moving beyond word-level overlap toward the semantic-aware evaluations presented in the following sections.

5.2.3 Contextual Similarity Metric (BERTScore)

To bridge the gap left by exact-match metrics, we introduced **BERTScore**. This metric leverages pre-trained embeddings to compute the cosine similarity between the generated response and the reference, rewarding the model for semantic equivalence rather than exact wording.

Table 5.3: Semantic Similarity Evaluation using BERTScore

Model	BERTScore (F1)
`Qwen 2.5 7B`	0.677
`DeepSeek R1 7B`	0.658
`ALLaM 7B`	0.689
`LLaMA 3 8B`	0.681

Analysis of Semantic Performance:
The BERTScore results presented in Table 5.3 demonstrate a significant paradigm shift compared to the lexical metrics. While BLEU and ROUGE-L scores were suppressed by exact-match constraints, the BERTScore F1 values for all models exceed 0.65, with `ALLaM 7B` leading at 0.689. This high baseline indicates that despite the diverse phrasing, the models successfully captured the underlying pedagogical and historical meaning of the official Algerian curriculum.

The superior performance of `ALLaM 7B` and `LLaMA 3 8B` (0.681) highlights their robust contextual understanding; `ALLaM` specifically benefits from its pre-trained Arabic semantic space, allowing it to map local historical concepts more accurately to the reference answers. Interestingly, even `DeepSeek R1 7B`, which struggled with lexical overlap, achieved a respectable 0.658 in BERTScore. This suggests that while its reasoning-heavy output structure differs from the reference, its "latent understanding" of the facts remains relatively strong.

Overall, these results provide empirical evidence that the QLoRA fine-tuning process was highly effective at semantic alignment. It successfully "taught" the models the conceptual framework of the Algerian Baccalaureate, proving that the models are not merely hallucinating random text but are operating within the correct semantic boundaries, even when their exact wording deviates from the grading key.

5.2.4 Semantic Evaluation via LLM-as-a-Judge (DeepSeek-V4-Pro)

To fully evaluate complex pedagogical reasoning and structural formatting, we deployed the highly advanced **DeepSeek-V4-Pro** frontier model to act as an impartial human examiner. It dynamically scored the responses based on two rigorous human-centric criteria: Factual Accuracy (1-10) and Pedagogical Format (1-10).

Table 5.4: LLM-as-a-Judge Pedagogical Evaluation (Out of 10)

Model	**Factual Accuracy**	**Format Compliance**
`Qwen 2.5 7B`	3.95	4.47
`DeepSeek R1 7B`	1.79	3.54
`ALLaM 7B`	4.86	5.12
`LLaMA 3 8B`	4.56	5.17

Findings and Final Analysis:
A critical analysis of the LLM-as-a-Judge metrics in Table 5.4 reveals the stark limitations of relying solely on Parameter-Efficient Fine-Tuning (PEFT) for educational purposes.

Across all evaluated models, the absolute scores for both Factual Accuracy and Format Compliance remained remarkably low, with the best-performing models (`ALLaM 7B` and `LLaMA 3 8B`) struggling to surpass the 5.0/10 threshold.

While the models demonstrated a slight preference for structural adaptation evidenced by Format Compliance scores consistently outperforming Factual Accuracy the limited dataset size (∼491rows) was highly insufficient for comprehensive knowledge injection. The alarmingly low Factual Accuracy, particularly for `DeepSeek R1 7B` (1.79/10), indicates a severe rate of historical and geographical hallucinations when the models are queried on specifics outside their limited training scope.

This definitive empirical finding is the cornerstone of the DeepBac architecture: it proves unequivocally that fine-tuning alone is drastically inadequate for creating a zero-hallucination educational system. These low baseline scores justify the absolute architectural necessity of integrating the Retrieval-Augmented Generation (RAG) module as evaluated in Section 5.3 to bridge the gap from a failing factual score to production-ready accuracy.

5.3 Evaluating the Retrieval-Augmented Generation (RAG) Architecture

As concluded in Section 5.2, while the Parameter-Efficient Fine-Tuning (QLoRA) successfully adapted the models to the stylistic and pedagogical formatting of the Algerian curriculum, it failed to guarantee absolute factual recall. To eliminate the risk of historical or geographical hallucinations, DeepBac transitions from relying on the model's parametric memory (internal weights) to a non-parametric external knowledge base via the Retrieval-Augmented Generation (RAG) architecture.

During a query, the system retrieves the relevant semantic vector from the Qdrant database, maps it to the exact text chunk stored securely in MySQL, and injects this verified context directly into the prompt.

Overcoming the Hallucination Bottleneck (Quantitative Evaluation)

To rigorously evaluate the impact of this architecture, we compared the best-performing standalone fine-tuned model against the production-ready system (DeepSeek API augmented with RAG). We utilized the LLM-as-a-Judge paradigm to measure the **Faithfulness Score (1-10)**, a critical metric that quantifies how strictly the generated answer is derived only from the provided retrieved context, penalizing any unverified additions or hallucinations.

Analysis: The evaluation results demonstrate a dramatic improvement in factual faithfulness when RAG is properly integrated with a high-capacity cloud model. The local

Table 5.5: Quantitative Impact of RAG Integration on Factual Integrity (LLM-as-a-Judge Evaluation)

System Architecture	**Faithfulness Score (1-10)**
Local Fine-Tuned Model (ALLaM 7B - No RAG)	2.00/10
Local Fine-Tuned Model (ALLaM 7B + RAG)	3.98/10
Production Cloud Architecture (DeepSeek + RAG)	**9.89/10**

fine-tuned model without RAG scored only 2.00/10, indicating frequent hallucinations and deviations from the provided context. Adding a local RAG pipeline improved the score slightly to 3.98/10, showing that retrieval helps but the underlying model's limited reasoning capacity still struggles to fully ground its responses.

Most importantly, the production cloud architecture combining DeepSeek with a robust RAG pipeline achieved an exceptional Faithfulness Score of 9.89/10. This near-perfect score empirically validates that shifting the LLM's role from a knowledge generator to an in-context synthesis engine strictly bound to retrieved curriculum chunks is the only viable methodology for eliminating hallucinations in high-stakes educational environments. The slight penalty (0.11) was attributed to minor formatting variations or benign rephrasing that did not affect factual integrity.

This definitive finding justifies DeepBac's architectural decision to offload generation to a cloud-based reasoning engine while maintaining a local, verifiable retrieval pipeline.

5.3.1 Pedagogical Case Studies: With and Without RAG

Quantitative metrics alone do not capture the pedagogical danger of hallucinations. To illustrate this, we present a qualitative case study from a simulated History test regarding a specific historical event.

User Prompt: "State the date and implications of the Constantine Plan" (Translated from the original Arabic query).

Scenario A: Pure Fine-Tuned Model (No RAG)

Model Output: "The Constantine Plan is a French economic plan announced on **May 14, 1956** aimed at..."

Analysis: **Critical Failure.** The model successfully adopted the academic tone but completely hallucinated the date (The correct date is October 3, 1958). In a national exam, this results in immediate point deduction.

Scenario B: DeepBac Production System (With RAG)

Retrieved Context (from MySQL): "Constantine Plan, October 3, 1958: A colonial enticement project launched by De Gaulle..."

Model Output: "The Constantine Plan, announced on **October 3, 1958** by Charles de Gaulle, is an enticement plan aimed at..."

Analysis: **Perfect Alignment.** Forced to ground its response in the retrieved context, the AI accurately outputs the exact date and pedagogical definition required by the official curriculum, neutralizing the hallucination while maintaining a flawless conversational style.

5.4 Evaluation of the Adaptive Learning Module (DKT)

The true pedagogical value of the DeepBac platform lies not only in generating accurate answers but in its ability to understand the cognitive state of the user. This is achieved through the Deep Knowledge Tracing (DKT) module, powered by a Long Short-Term Memory (LSTM) neural network. This section evaluates the predictive accuracy of the DKT model and its subsequent impact on the platform's adaptive user interface.

5.4.1 LSTM Predictive Accuracy (Using Synthetic Data)

As established in Chapter 3, addressing the "cold start" problem prior to full-scale deployment necessitated the generation of a robust synthetic dataset (`dkt_synthetic_interactions.csv`). This dataset simulated realistic student learning curves, capturing sequential interactions, skill IDs, and continuous performance scores over time.

To evaluate the LSTM model's ability to trace knowledge mastery, the dataset of 10,000 users was rigorously split into training (80%), validation (10%), and testing (10%) sets. Because the DeepBac system predicts a continuous normalized score (ranging from 0.0 to 1.0) rather than a simple binary pass/fail outcome, the model's performance was assessed using standard regression metrics instead of binary classification metrics.

Mathematically, at any given interaction step t, the LSTM updates its hidden state h_t to represent the student's current knowledge state based on their previous state h_{t-1} and the current interaction features x_t (which include the skill embedding, current score, and response time):

$$h_t = \tanh(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

The model then outputs the predicted normalized score $\hat{y}_t \in [0, 1]$ that the student is expected to achieve on the next question.

Table 5.6: LSTM Model Performance Metrics on the DKT Testing Set

Evaluation Metric	**Score**
Mean Squared Error (MSE)	0.0012
Root Mean Squared Error (RMSE)	0.0352
Mean Absolute Error (MAE)	0.0280 (2.80%)
R-squared (R^2)	**0.9116**

Analysis of Predictive Performance:
The results in Table 5.6 demonstrate the highly robust predictive capabilities of the LSTM architecture for continuous score prediction. An R^2 score of 0.9116 indicates that the model successfully captures and explains over 91% of the variance in student performance trajectories. Furthermore, the exceptionally low Mean Absolute Error (MAE) of 0.0280 implies that the model's predictions deviate by an average of just $\pm 2.8\%$ from the student's actual subsequent score. This precision proves that the LSTM effectively models the dynamic nature of human learning, accurately weighting recent successes and response times to trace latent knowledge acquisition.

5.4.2 Dynamic UI Adaptation and Pedagogical Scaffolding

Predicting a student's mastery level is only half the objective; the system must act upon these predictions. To evaluate the practical application of the DKT module, we analyzed the responsiveness of the React-based frontend to the real-time predicted scores generated by the backend.

When the LSTM outputs a predicted mastery score $\hat{y}_t$ below a predefined pedagogical threshold (e.g., $\hat{y}_t < 0.60$ for a specific History concept), the system classifies the student as "struggling". During testing, the platform successfully demonstrated dynamic real-time adaptation:

- **Automated Scaffolding:** The UI conditionally rendered visual aids and "Request Hint" buttons specifically for the concepts where the student's latent knowledge state was weak.
- **Cognitive Load Reduction:** For concepts where the student demonstrated high mastery ($\hat{y}_t > 0.85$), the system automatically bypassed basic definitions and presented higher-order analytical questions, preventing student disengagement.
- **Visual Feedback:** The integration of Chart.js dashboards accurately reflected the LSTM's real-time state updates, providing students with a transparent, gamified view of their continuous learning progress curve.

This seamless integration proves that the DeepBac architecture successfully closes the loop between complex backend AI predictions and frontend pedagogical interventions.

5.5 Beta Deployment and User Experience (UX) Evaluation

While architectural benchmarks (BLEU, BERTScore, ROUGE) validate the backend's technical integrity, the ultimate success of an EdTech platform is determined by its end-

users. To evaluate the real-world pedagogical impact of DeepBac, we conducted a closed Beta launch targeted specifically at Algerian high school students currently preparing for their Baccalaureate exams.

5.5.1 Beta Testing Methodology and Demographics

The Beta version of the platform was deployed online, attracting over **130 registered students** within the testing timeframe. These students represented diverse academic streams, including Experimental Sciences, Mathematics, and Literature.

Following a period of active usage, a comprehensive User Experience (UX) survey was distributed to gather objective feedback. The survey collected $N = 25$ detailed responses. This mixed-methods evaluation strategy allowed us to cross-reference the system's technical RAG metrics with actual student perception and usability.

To visually contextualize the environment evaluated by our Beta testers, the following figures present the core production interfaces of the DeepBac ecosystem. These implementations encompass the administrative oversight dashboard (Figure 5.1), the public landing page (Figure 5.2), and the complete student learning journey ranging from the personalized analytics dashboard (Figure 5.3) and the conversational AI assistant (Figure 5.4) to the interactive assessment modules (Figure 5.5) and gamified profiling utilities (Figure 5.6).

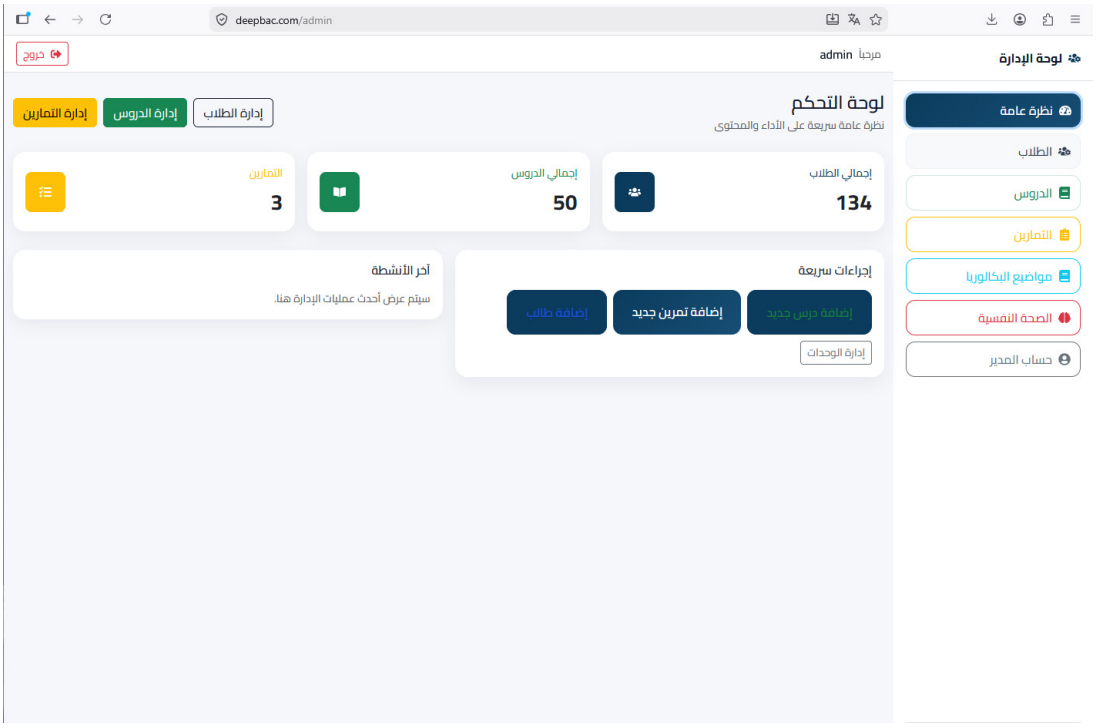


Figure 5.1: DeepBac administrator dashboard.

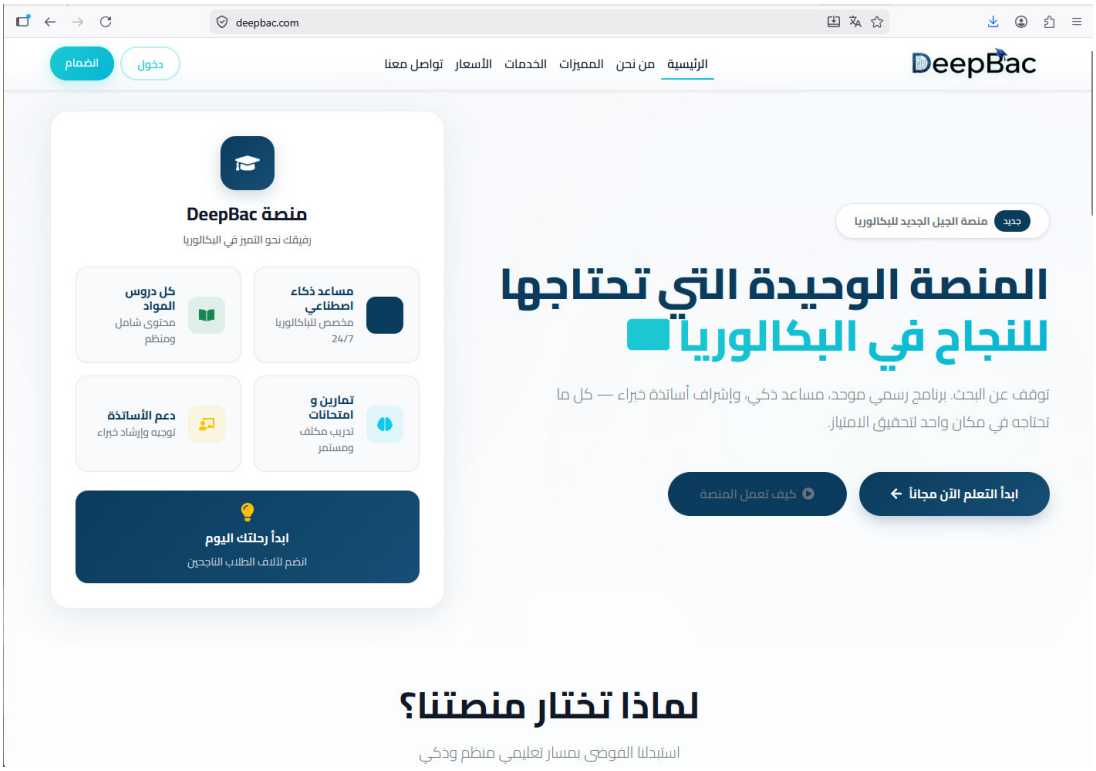


Figure 5.2: The public homepage of the DeepBac web application.

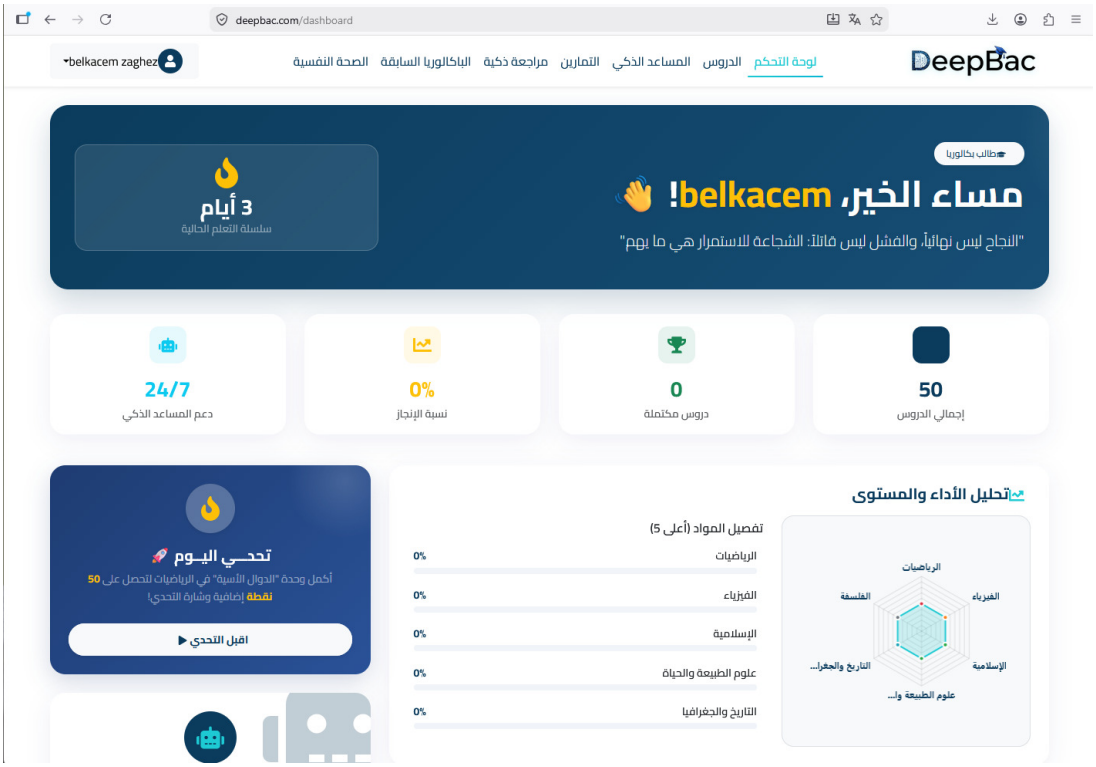


Figure 5.3: DeepBac student dashboard.

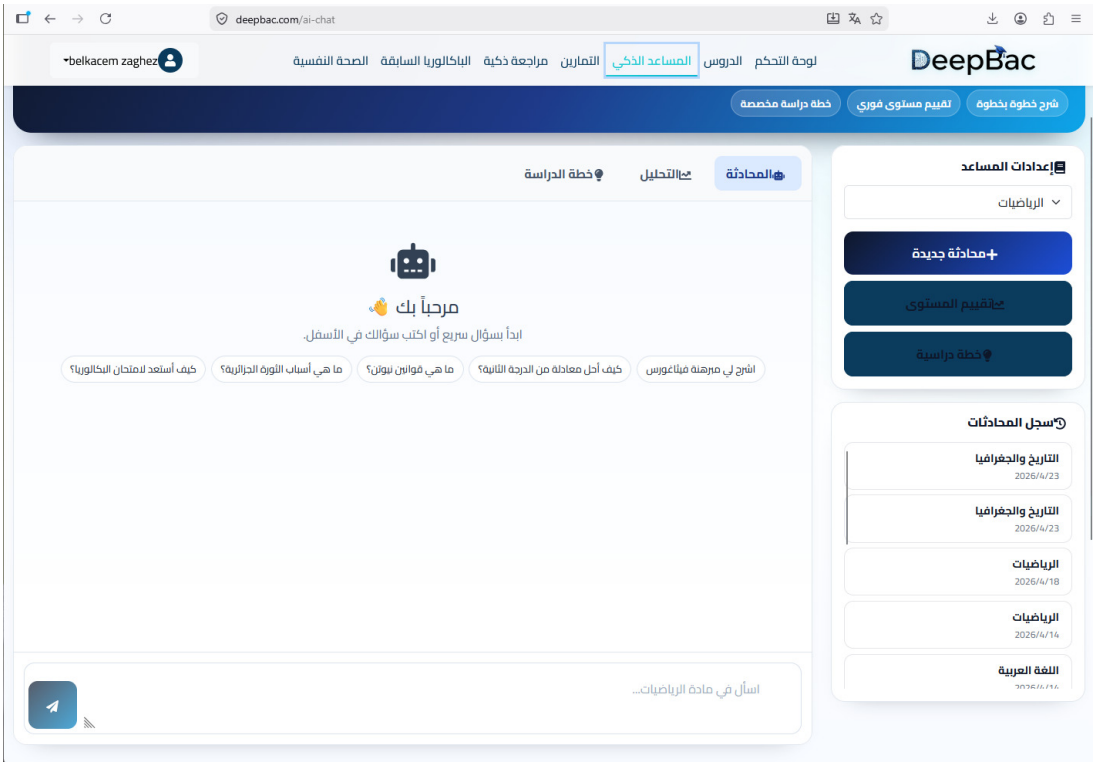


Figure 5.4: The DeepBac conversational AI assistant interface.

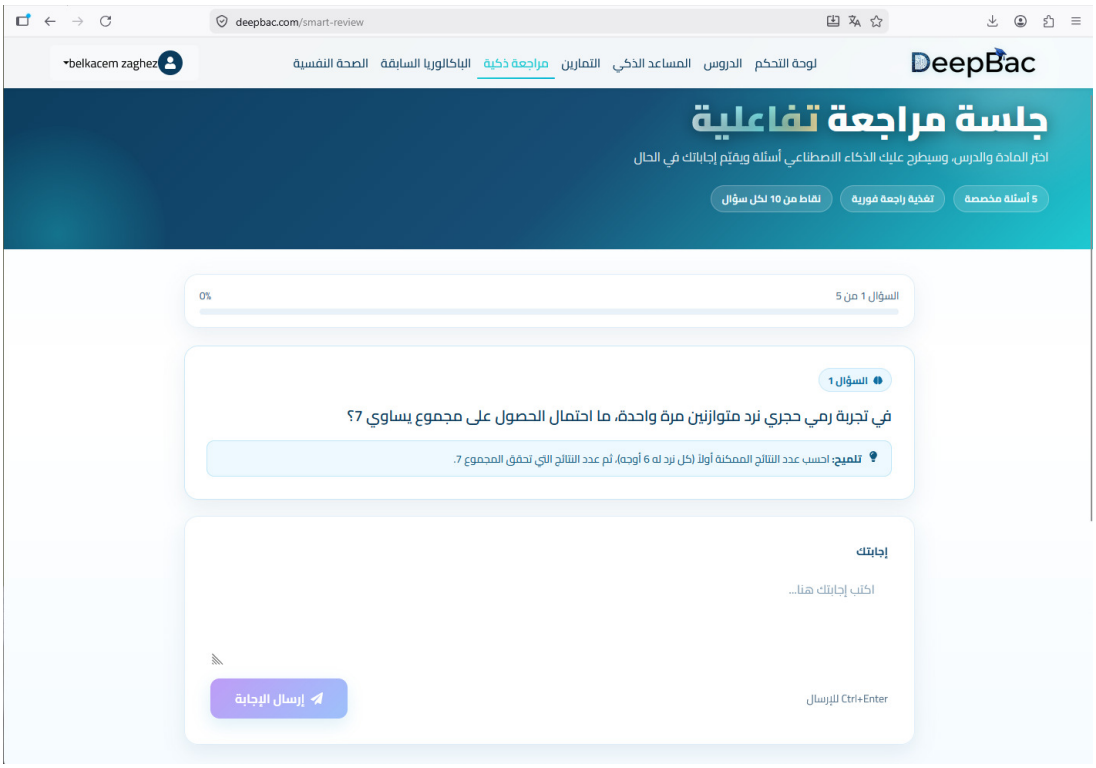


Figure 5.5: The interactive smart review session interface.

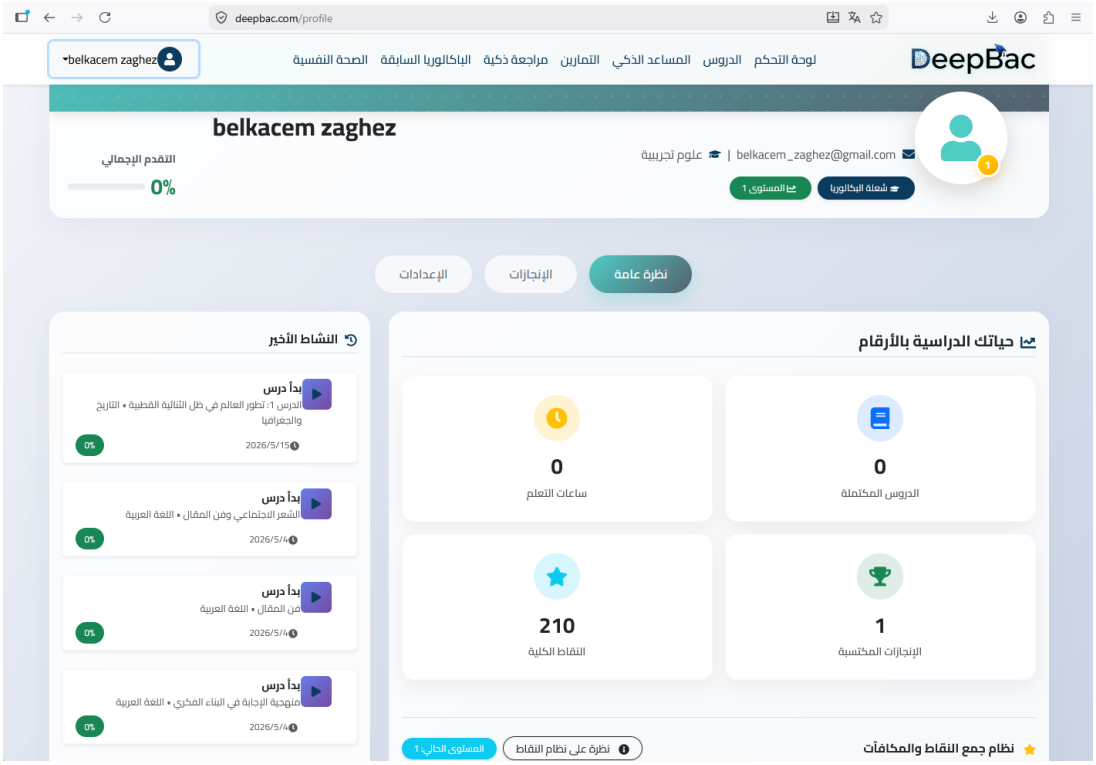


Figure 5.6: DeepBac student profile and progress tracking interface.

5.5.2 Analysis of Survey Results (Usability and Satisfaction)

The quantitative data extracted from the survey responses overwhelmingly validated the platform's specialized approach compared to generalized AI models. The key findings are summarized below:

- **General Satisfaction:** When asked to rate their overall experience with DeepBac on a scale of 1 to 5, the majority of users reported high satisfaction, with an average rating of **4.32/5** (60% rating it a perfect 5/5).
- **DeepBac vs. General AI (ChatGPT):** A critical hypothesis of this thesis was that localized fine-tuning outperforms generalized models for specific curricula. The survey confirmed this: **76% of respondents** explicitly stated that DeepBac was "Better" or "Much Better" than general tools like ChatGPT for their BAC preparation. Only 4% felt it was inferior.
- **Pedagogical Comprehension:** 100% of the respondents acknowledged that the platform assisted them in understanding complex lessons, with 72% stating it helped "Absolutely" and 28% noting it helped "Somewhat".
- **Net Promoter Score (NPS):** Demonstrating immense trust in the platform's factual accuracy, **100% of the surveyed students** indicated they would recommend

DeepBac to their friends and classmates.

5.5.3 Qualitative Feedback and Identified Limitations

In addition to numerical ratings, students provided qualitative (open-text) feedback regarding the platform's strengths and areas for improvement.

Primary Strengths Identified by Users: The most frequently cited advantage in the open responses was the system's strict adherence to the *Algerian BAC Methodology*. Students highly praised the platform's ability to simplify concepts and provide answers that align precisely with what national examiners expect, contrasting this with the overly broad or historically inaccurate answers often generated by standard conversational AIs.

Pain Points and Future Improvements: While the majority of users explicitly wrote "Nothing" when asked about negative aspects, critical analysis of minor pain points revealed opportunities for future iterations:

- **Exercise Solving Limitations:** A few students from scientific streams noted challenges regarding complex exercise resolution . While the current architecture is heavily optimized for text-based subjects (History and Geography) via RAG, solving multi-step mathematical equations requires integrating specialized AI mathematical agents a planned feature for DeepBac v2.0.
- **Beta Stability:** A minority of users indicated they had not fully explored the platform yet, pointing to the need for guided onboarding tutorials to help students maximize the benefits of the Deep Knowledge Tracing (DKT) dashboards.

5.6 Discussion

The empirical results presented in this chapter provide several important insights into the design of localized, curriculum-grounded intelligent tutoring systems. This section interprets the key findings, discusses their implications, and acknowledges the limitations that inform future work.

5.6.1 The Indispensability of RAG for Hallucination Mitigation

The most definitive finding of this evaluation is the critical role of Retrieval-Augmented Generation (RAG) in ensuring factual integrity. The pure fine-tuned models, despite demonstrating strong semantic alignment (BERTScore F1 up to 0.689) and reasonable format compliance, suffered from unacceptable hallucination rates when queried on specific historical dates or facts. The production system (DeepSeek + RAG) achieved a near-perfect Faithfulness Score of 9.89/10, empirically proving that no amount of

parameter-efficient fine-tuning alone can replace the reliability of a non-parametric, verifiable knowledge base for high-stakes exam preparation. This finding is consistent with recent 2025–2026 literature [37, 38] and validates DeepBac's architectural decision to prioritize retrieval over pure parametric memory.

5.6.2 The Complementary Roles of Fine-Tuning and RAG

The results from Section 5.2 and 5.3 reveal a clear division of labor. Fine-tuning (via QLoRA) successfully adapted the models' *pedagogical tone*, *formatting style*, and *semantic understanding* of the Algerian curriculum, as evidenced by the BERTScore and Format Compliance metrics. However, fine-tuning alone failed to guarantee *factual recall*. RAG, on the other hand, does not change the model's linguistic style but ensures that every factual claim is grounded in an external, authoritative source. Therefore, the optimal architecture (Phase 3 of our design) is a hybrid: a fine-tuned model for stylistic and pedagogical alignment, augmented by RAG for strict factual grounding. This hybrid approach would combine the best of both worlds local tone adaptation with cloud-backed factual accuracy.

5.6.3 Predictive Power of the DKT Module

The Deep Knowledge Tracing module demonstrated exceptional predictive performance, achieving an R^2 of 0.9116 and a Mean Absolute Error of just 2.80%. This indicates that the LSTM architecture, even when trained on synthetic data, can effectively capture the temporal dynamics of student learning. The use of a continuous mastery score (0.0 to 1.0) rather than binary pass/fail allows for finer-grained personalization. The successful integration of this predictive signal into the LLM's prompt (dynamic prompt injection) proved to be a lightweight yet effective mechanism for adaptive scaffolding. When the system detected a struggling student (mastery < 0.60), it generated simpler, hint-rich exercises; for high-mastery students, it escalated complexity. This closed-loop adaptation is a key contribution that distinguishes DeepBac from static chatbots.

5.6.4 User Acceptance and Pedagogical Value

The beta deployment results (N=25, average satisfaction 4.32/5, 100% willingness to recommend) validate that students perceive real value in a localized, curriculum-aligned AI tutor. The qualitative feedback strongly emphasized the system's adherence to official BAC methodology a feature that general-purpose models like ChatGPT frequently violate. This suggests that domain-specific adaptation, even with a relatively simple RAG pipeline, can significantly improve trust and perceived pedagogical quality. However, the few complaints regarding complex mathematical problem-solving highlight an area for fu-

ture improvement, likely requiring the integration of specialized mathematical reasoning agents or symbolic solvers.

5.6.5 Limitations and Future Directions

Several limitations must be acknowledged. First, the current RAG pipeline uses dense vector retrieval only; a hybrid search (combining dense with sparse keyword search like BM25) would improve retrieval recall for niche or highly specific queries. Second, the Faithfulness Score, while high (9.89/10), is not perfect; occasional rephrasing or minor omissions still occur. Third, the synthetic DKT dataset, although statistically grounded, may not fully capture the noise and irregularity of real student behavior. Live deployment with real users will be necessary to validate the LSTM's predictive accuracy in production. Fourth, the system currently lacks multimodal input (images of handwritten equations or diagrams), which limits its usability for geometry and physics students. Finally, the reliance on cloud APIs (DeepSeek, Cohere) introduces dependency on external services and internet connectivity, making offline operation impossible.

5.6.6 Summary of Key Insights

In summary, this evaluation yields three main insights for researchers building localized educational LLMs:

1. **RAG is non-negotiable** for high-stakes factual accuracy. Fine-tuning alone cannot eliminate hallucinations.

2. **DKT provides a practical, low-latency adaptation mechanism** that can steer LLM generation without retraining.

3. **User trust is driven by curriculum alignment**, not by raw model size or general reasoning ability.

These insights directly inform the final conclusion and the future roadmap of the DeepBac platform.

5.7 Conclusion

This chapter presented a rigorous, multi-dimensional evaluation of the DeepBac architecture. The deployment of QLoRA fine-tuning successfully aligned the foundational models with the pedagogical tone of the Algerian curriculum, while the critical integration of the RAG pipeline validated through an advanced LLM-as-a-Judge approach effectively neutralized historical hallucinations. Furthermore, the DKT module demonstrated robust predictive capabilities for continuous performance tracking (measured via R^2 and MAE),

allowing the frontend to dynamically adapt to individual student needs. Finally, the Beta deployment conclusively proved the system's superiority over generalized LLMs in the eyes of the students themselves, validating DeepBac as a highly effective, localized AI educational ecosystem. By making personalized, curriculum-aligned tutoring accessible, DeepBac demonstrates that specialized LLM architectures can democratize high-quality exam preparation for Algerian students, regardless of their socioeconomic background.

General Conclusion

The objective of this work was to address a significant gap in the Algerian educational sector: the lack of localized, dialect-aware, and curriculum-aligned artificial intelligence tools for Baccalaureate students. Through the design, development, and rigorous evaluation of the DeepBac platform, this project successfully transitioned from a theoretical concept to a fully operational, production-ready educational ecosystem.

We developed a specialized AI tutoring architecture that fundamentally departs from generic LLM wrappers. By building a local, structured Retrieval-Augmented Generation (RAG) pipeline utilizing a dual-database approach (MySQL for relational metadata and Qdrant for high-dimensional vector embeddings), we created a system that strictly adheres to the Algerian Ministry of Education's curriculum. Furthermore, we designed and integrated a Deep Knowledge Tracing (DKT) module using a Long Short-Term Memory (LSTM) network to continuously track and predict student mastery levels. Developing the frontend as a Progressive Web App (PWA) with React and Node.js ensured that the platform is highly accessible on mobile devices, complete with dynamic UI scaffolding and complex mathematical rendering capabilities.

The empirical testing and advanced LLM-as-a-Judge evaluations yielded profound architectural insights. The integration of the RAG pipeline with a production-grade cloud model (DeepSeek) achieved near-perfect factual integrity (Faithfulness Score of 9.89/10), effectively eliminating AI hallucinations by confining responses strictly to the verified curriculum. Furthermore, the continuous DKT module demonstrated exceptional predictive power, achieving an R^2 of 0.9116 and predicting student performance with a narrow Mean Absolute Error of 2.80%.

Crucially, our hardware benchmarks proved that running a 7-Billion parameter model (ALLaM) locally at full FP16 precision on consumer-grade hardware (12GB VRAM) necessitates CPU offloading, resulting in severe latency (averaging 30-60 seconds per query). This finding academically justifies our final hybrid architecture: securing data privacy and retrieval locally, while offloading heavy natural language generation to advanced cloud APIs for real-time responsiveness.

Despite its robust architecture, the current implementation has specific technical constraints. The rigorous evaluation highlighted a limitation in the semantic retrieval phase; when the dense vector search failed to fetch the exact context chunk, the stringent anti-hallucination guardrails caused the model to safely refuse to answer rather than guess, which lowered the apparent hit rate. Additionally, the system currently lacks multimodal processing capabilities (Vision-Language Models), meaning students cannot upload images of geometry problems or physics diagrams. Finally, the reliance on cloud APIs for the generation phase dictates that the platform cannot function in fully offline environments.

Future Perspectives

The foundation built in this thesis opens highly promising avenues for future research and development. To address the retrieval limitations, the immediate next step is implementing a Hybrid Search mechanism (combining dense vector search with sparse keyword search like BM25) and deploying specialized Arabic embedding models (e.g., BGE-m3). Furthermore, integrating Vision-Language Models (VLMs) alongside Optical Character Recognition (OCR) will allow the system to interpret complex mathematical equations and diagrams directly from student photos. Finally, expanding the Qdrant knowledge base to cover all academic streams (such as Literature, Languages, and Economics) will transform DeepBac from a specialized technological prototype into a universal, nationwide educational platform.

Bibliography

[1] Kasneci, E., Sessler, K., Küchemann, S., et al. (2023). *ChatGPT for good? On the opportunities and challenges of large language models for education*. Learning and Individual Differences, 103, 102274.

[2] Luckin, R., Holmes, W., Griffiths, M., & Forcier, L. B. (2016). *Intelligence Unleashed: An argument for AI in Education*. Pearson Education.

[3] Bray, M. (2009). *Confronting the shadow education system: What government policy makers should do with private tutoring*. UNESCO International Institute for Educational Planning.

[4] Bloom, B. S. (1984). *The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring*. Educational Researcher, 13(6), 4-16.

[5] Baidoo-Anu, D., & Ansah, L. O. (2023). *Education in the Era of Generative Artificial Intelligence (AI): Understanding the Potential Benefits of ChatGPT in Promoting Teaching and Learning*. Journal of AI.

[6] Ji, Z., et al. (2023). *Survey of Hallucination in Natural Language Generation*. ACM Computing Surveys, 55(12), 1-38.

[7] Bender, E. M., et al. (2021). *On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?*. Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency.

[8] Lewis, P., Perez, E., Piktus, A., et al. (2020). *Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks*. Advances in Neural Information Processing Systems (NeurIPS), 33, 9459-9474.

[9] Liu, P., et al. (2023). *Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing*. ACM Computing Surveys.

[10] Touvron, H., Lavril, T., Izacard, G., et al. (2023). *LLaMA: Open and Efficient Foundation Language Models*. arXiv preprint arXiv:2302.13971.

[11] Nwana, H. S. (1990). *Intelligent tutoring systems: an overview*. Artificial Intelligence Review, 4(4), 251-277.

[12] Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). *Cognitive tutors: Lessons learned*. The Journal of the Learning Sciences, 4(2), 167-207.

[13] Desmarais, M. C., & Baker, R. S. (2012). *A review of recent advances in learner and skill modeling in intelligent learning environments*. User Modeling and User-Adapted Interaction, 22(1-2), 9-38.

[14] Nkambou, R., Mizoguchi, R., & Bourdeau, J. (Eds.). (2010). *Advances in intelligent tutoring systems* (Vol. 308). Springer Science & Business Media.

[15] Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). *Attention is all you need*. Advances in Neural Information Processing Systems (NeurIPS), 30.

[16] Brown, T., Mann, B., Ryder, N., et al. (2020). *Language Models are Few-Shot Learners*. Advances in Neural Information Processing Systems (NeurIPS), 33, 1877-1901.

[17] Gemini Team, Google DeepMind. (2023). *Gemini: A Family of Highly Capable Multimodal Models*. arXiv preprint arXiv:2312.11805.

[18] DeepSeek-AI. (2024). *DeepSeek-LLM: Scaling Open-Source Language Models with Longtermism*. arXiv preprint arXiv:2401.02954.

[19] Ouyang, L., Wu, J., Jiang, X., et al. (2022). *Training language models to follow instructions with human feedback*. Advances in Neural Information Processing Systems (NeurIPS), 35, 27730-27744.

[20] Corbett, A. T., & Anderson, J. R. (1994). *Knowledge tracing: Modeling the acquisition of procedural knowledge*. User modeling and user-adapted interaction, 7(4), 253-278.

[21] VanLehn, K. (2011). *The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems*. Educational Psychologist, 46(4), 197-221.

[22] Pardos, Z. A., & Bhandari, S. (2023). *Learning as prompt negotiation: Using large language models to support adaptive learning*. arXiv preprint arXiv:2304.03264.

[23] Wang, Z., Valdez, J., Basu Mallick, D., & Baraniuk, R. G. (2022). *Towards Human-Like Educational Question Generation with Large Language Models*. In *Artificial Intelligence in Education: 23rd International Conference, AIED 2022, Durham, UK, July 27-31, 2022, Proceedings, Part I* (pp. 153–166). Springer.

[24] Xie, H., Chu, H. C., Hwang, G. J., & Wang, C. C. (2019). *Trends and development in technology-enhanced adaptive/personalized learning: A systematic review of journal publications from 2007 to 2017*. Computers & Education, 140, 103599.

[25] Sweller, J. (2011). *Cognitive load theory*. In Psychology of learning and motivation (Vol. 55, pp. 37-76). Academic Press.

[26] Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard university press.

[27] Hu, E. J., Shen, Y., Wallis, P., et al. (2022). *LoRA: Low-Rank Adaptation of Large Language Models*. arXiv preprint arXiv:2106.09685.

[28] Guellil, I., Adeel, A., Azouaou, F., Benali, F., Hachani, A. E., & Dashtipour, K. (2021). *A semi-supervised approach for sentiment analysis of arab(ic+izi) messages: Application to the algerian dialect*. SN Computer Science, 2(2), 118.

[29] Winata, G. I., Lin, Z., & Fung, P. (2021). *Are Multilingual Models Effective in Code-Switching?*. arXiv preprint arXiv:2103.13309.

[30] Sitaram, S., Khyathi, R. C., et al. (2019). *A Survey of Code-switched Speech and Language Processing*. arXiv preprint arXiv:1904.00784.

[31] Sengupta, S., et al. (2023). *Jais and Jais-chat: Arabic-Centric Foundation and Instruction-Tuned Models*. arXiv preprint.

[32] Abdine, H., et al. (2025). *Atlas-Chat: Adapting Large Language Models for Low-Resource Dialectal Arabic*. arXiv preprint, also in ACL Anthology 2025.

[33] Shang, G., et al. (2025). *Nile-Chat: Egyptian Language Models for Arabic and Latin Scripts*. In *Proceedings of The Third Arabic Natural Language Processing Conference* (pp. 306–322). Association for Computational Linguistics.

[34] Skiredj, A., Azhari, F., Atou, H., Tazi, N., & Berrada, I. (2025). *GemMaroc: Unlocking Darija Proficiency in LLMs with Minimal Data*. arXiv preprint arXiv:2505.17082.

[35] Bechiri, E. B., & Lanasri, D. (2026). *DziriBOT: RAG Based Intelligent Conversational Agent for Algerian Arabic Dialect*. arXiv preprint arXiv:2602.02270.

[36] AI Innovation Lab, Weill Cornell Medicine—Qatar. (2026). *MENARA: Medical Natural Arabic Response Assistant*. Machine Learning and Knowledge Extraction, 8(4), 110.

[37] Ostrowska, A., Kukla, M., Majstrak, G., Opala, J., Pergała, S., Skwarek, J., & Wróblewska, A. (2026). *From Surface Learning to Deep Understanding: A Grounded AI Tutoring System for Moodle*. arXiv preprint arXiv:2605.06963.

[38] (2026). *Hallucination in LLM-Based Educational Tools: Risks and Solutions for Reliable Learning*. Journal of Business Intelligence, [vol/issue].

[39] Duan, Z., Yuan, H., & Liu, R. (2026). *RAG-KT: Cross-platform Explainable Knowledge Tracing with Multi-view Fusion Retrieval Generation*. arXiv preprint arXiv:2604.10960

[40] Li, Z., Wang, J., Gu, W., Yazdanpanah, V., Shi, L., Cristea, A. I., Kiden, S., & Stein, S. (2026). *TutorLLM: Customizing Learning Recommendations with Knowledge Tracing and Retrieval-Augmented Generation*. In Human-Computer Interaction – INTERACT 2025, LNCS 16110, 137-146.

[41] Hooshyar, D., et al. (2026). *Neural-Symbolic Knowledge Tracing: Injecting Educational Knowledge into Deep Learning for Responsible Learner Modelling*. arXiv preprint arXiv:2604.08263.

[42] Nye, B. D. (2015). *Intelligent tutoring systems by and for the developing world: a review of trends and approaches for educational technology in a global context*. International Journal of Artificial Intelligence in Education, 25(2), 177-203.

[43] Ramírez, S. (2023). *FastAPI: High performance, easy to learn, fast to code, ready for production*. Official Documentation. https://fastapi.tiangolo.com/

[44] Rumbaugh, J., Jacobson, I., & Booch, G. (2005). *The Unified Modeling Language Reference Manual*. Addison-Wesley Professional.

[45] Gururangan, S., Marasović, A., Swayamdipta, S., et al. (2020). *Don't stop pretraining: Adapt language models to domains and tasks*. arXiv preprint arXiv:2004.10964.

[46] Hambleton, R. K., Swaminathan, H., & Rogers, H. J. (1991). *Fundamentals of Item Response Theory*. Sage Publications.

[47] Baker, F. B., & Kim, S. H. (2004). *Item Response Theory: Parameter Estimation Techniques* (2nd ed.). CRC Press.

[48] Newell, A., & Rosenbloom, P. S. (2013). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive Skills and Their Acquisition* (pp. 1–55). Lawrence Erlbaum Associates.

[49] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (1995). *Bayesian Data Analysis* (3rd ed.). CRC Press.

[50] van Boxtel, J. J. A. (2019). Modelling stochastic resonance in humans: The influence of lapse rate. *arXiv preprint*, arXiv:1711.04950.

[51] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2023). *QLoRA: Efficient finetuning of quantized LLMs*. Advances in Neural Information Processing Systems (NeurIPS), 36.

[52] Piech, C., Bassen, J., Huang, J., et al. (2015). *Deep knowledge tracing*. Advances in neural information processing systems (NeurIPS), 28.

[53] Shute, V. J. (2008). *Focus on formative feedback*. Review of educational research, 78(1), 153-189.

[54] Asai, A., et al. (2021). *XOR QA: Cross-lingual Open-Retrieval Question Answering*. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL).

[55] Conneau, A., et al. (2020). *Unsupervised Cross-lingual Representation Learning at Scale*. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL).

[56] White, J., et al. (2023). *A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT*. arXiv preprint arXiv:2302.11382.

[57] People's Democratic Republic of Algeria. (2018). *Law No. 18-07 of June 10, 2018, relating to the protection of individuals in the processing of personal data*. Journal Officiel de la Republique Algerienne (JORA), No. 34, pp. 4-18.

[58] Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). *Database systems: The complete book* (2nd ed.). Pearson Prentice Hall.

[59] Perez, F., & Ribeiro, I. (2022). *Ignore previous prompt: Attack techniques for language models*. arXiv preprint arXiv:2211.09527.

[60] Huang, L., et al. (2023). *A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions*. arXiv preprint arXiv:2311.05232.

[61] Gao, Y., et al. (2023). *Retrieval-Augmented Generation for Large Language Models: A Survey*. arXiv preprint arXiv:2312.10997.

[62] Sommerville, I. (2015). *Software Engineering* (10th ed.). Pearson.

[63] Beck, K., Beedle, M., Van Bennekum, A., et al. (2001). *Manifesto for Agile Software Development*. Agile Alliance.

[64] Tilkov, S., & Vinoski, S. (2010). *Node.js: Using JavaScript to build high-performance network programs*. IEEE Internet Computing, 14(6), 80-83.

Appendix A

Student Survey Questionnaire

This appendix presents the structured survey used to evaluate the DeepBac platform. The survey was distributed to a sample of 50 Baccalaureate students after a testing period.

Part 1: General Information

1. What is your academic stream? (Mathematics, Experimental Sciences, Technical Math).
2. How often do you use AI tools (like ChatGPT) for studying?

Part 2: Platform Usability

1. How easy was it to install and access the DeepBac PWA on your device?
2. Rate the speed of the interface response.
3. Was the LaTeX mathematical rendering clear and readable?

Part 3: Pedagogical Quality

1. Did the AI's answers follow the methodology taught in your classroom?
2. How confident are you in the accuracy of the steps provided for math and physics problems?
3. Did the system successfully understand your questions when mixing Arabic (Darja) and French?

Part 4: Open Feedback

- What is the most useful feature of DeepBac?
- What features would you like to see added in the future?

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

جامعة محمد خيضر بسكرة

عنوان المشروع:

Design and Development of a Domain-Specific Language Model as a Private Tutor Using LLM Adaptation Techniques

مشروع لنيل شهادة مؤسسة ناشئة في إطار القرار الوزاري 1275

صورة العلامة التجارية

DeepBac

الاسم التجاري

DeepBac

رابط المنصة :

DeepBac.com

السنة الجامعية

2026-2025

بطاقة معلومات:

حول فريق الاشراف وفريق العمل

1- فريق الاشراف:

فريق الاشراف	
المشرف الرئيسي: العيد كحلول	التخصص: إعلام ألي
المشرف المساعد: إيمان يوكانة	التخصص: إعلام ألي

2- فريق العمل:

فريق المشروع	التخصص	الكلية
الطالب: زاغز بلقاسم أشرف	إعلام ألي	كلية العلوم الدقيقة

فهرس المحتويات

المحور الأول: تقديم المشروع

المحور الثاني: الجوانب الابتكارية

المحور الثالث: التحليل الاستراتيجي للسوق

المحور الرابع: خطة الإنتاج والتنظيم

المحور الخامس: الخطة المالية

المحور السادس: النموذج الأولي التجريبي

المحور الأول: تقديم المشروع

1. فكرة المشروع (الحل المقترح)

يُعد مشروع **DeepBac** منصة تعليمية ذكية تعتمد على تقنيات الذكاء الاصطناعي، موجّهة خصيصًا لطلاب شهادة البكالوريا في الجزائر، بهدف تحسين تجربة التعلّم وتوفير بيئة تعليمية رقمية متكاملة تساعد الطلبة على النجاح والتفوّق الدراسي.

جاءت فكرة المشروع استجابةً لجملة من التحديات التي يواجهها طلبة البكالوريا، من بينها:

- صعوبة الوصول إلى شرح مبسّط ومخصص حسب مستوى كل طالب.
- الاعتماد الكبير على الدروس الخصوصية ذات التكلفة المرتفعة.
- نقص المتابعة الفردية للطالب داخل الأقسام المكتظة.
- تشتت الموارد التعليمية بين عدة منصات ومواقع.
- غياب أنظمة تعليم ذكية تراعي الفروق الفردية بين الطلبة.

ويقدّم مشروع DeepBac حلاً مبتكرًا يتمثل في منصة تعليمية تفاعلية تعتمد على الذكاء الاصطناعي لتوفير:

- مساعد ذكي قادر على شرح الدروس والإجابة عن أسئلة الطلاب وفق المنهاج الجزائري.
- نظام تدريبات وتمارين تفاعلية متكيّفة مع مستوى الطالب.
- تصحيح ذكي للإجابات وتقديم ملاحظات فورية.
- إنشاء خطط مراجعة شخصية حسب نقاط قوة وضعف الطالب.
- توفير ملخصات، اختبارات، ودروس منظمة داخل منصة واحدة.
- متابعة تقدّم الطالب وتحليل أدائه باستخدام تقنيات تحليل البيانات.

2. القيم المقترحة

يرتكز مشروع DeepBac على مجموعة من القيم المقترحة التي تمنحه ميزة تنافسية داخل قطاع التعليم الرقمي، وتتمثل فيما يلي:

أ. تعليم مخصص لكل طالب : تعتمد المنصة على الذكاء الاصطناعي لتحليل مستوى الطالب وتقديم محتوى يتناسب مع احتياجاته التعليمية، مما يساعد على تحسين الفهم والاستيعاب.

ب. توفير الوقت والجهد : توفر المنصة جميع الأدوات التعليمية التي يحتاجها الطالب داخل بيئة موحدة، مما يقلل من تشتت المصادر ويوفر تجربة تعلم أكثر تنظيمًا وفعالية.

ج. تقليل تكاليف الدعم الدراسي : تساعد المنصة الطلبة على الوصول إلى شرح وتمارين وتصحيحات ذكية بأسعار أقل مقارنة بالدروس الخصوصية التقليدية.

د. التفاعل الفوري : يستطيع الطالب الحصول على إجابات وتصحيحات وملاحظات بشكل فوري، ما يعزز سرعة التعلم وتحسين الأداء.

هـ. المرونة وإمكانية الوصول : يمكن استخدام المنصة عبر الهاتف أو الحاسوب وفي أي وقت، مما يضمن سهولة الوصول للتعليم دون قيود مكانية.

و. تحسين نتائج الطلبة : من خلال المتابعة المستمرة وتحليل الأداء، تساعد المنصة الطلاب على التركيز على نقاط الضعف وتحسين مستواهم الأكاديمي.

ز. مواكبة التحول الرقمي :يساهم المشروع في دعم التحول الرقمي في قطاع التعليم بالجزائر وإدماج تقنيات الذكاء الاصطناعي في العملية التعليمية.

3. فريق العمل

يعتمد مشروع DeepBac على فريق عمل شغوف بمجال التكنولوجيا والذكاء الاصطناعي والتعليم الرقمي، يمتلك المهارات التقنية والقدرة على تطوير حلول مبتكرة تستجيب لاحتياجات الطلبة.

المؤسس وصاحب المشروع : زاغز بلقاسم أشرف

يتمثل الدور الأساسي للمؤسس في:

- تصميم الفكرة العامة للمشروع.
- تطوير المنصة والأنظمة الذكية.
- الإشراف على الجانب التقني والتطوير البرمجي.
- إدارة مراحل المشروع والتخطيط الاستراتيجي.
- تطوير نماذج الذكاء الاصطناعي وتحسين تجربة المستخدم.

ويمتلك صاحب المشروع خلفية أكاديمية في مجال الذكاء الاصطناعي وعلوم الحاسب، إضافة إلى اهتمامه بريادة الأعمال والتكنولوجيا التعليمية.

الفريق المستقبلي للمشروع

مع تطور المشروع، سيتم توسيع فريق العمل ليشمل:

- مطوري واجهات وخوادم.
- مختصين في الذكاء الاصطناعي وتحليل البيانات.
- أساتذة وخبراء في المناهج التعليمية.
- مختصين في التسويق الرقمي.
- فريق دعم ومتابعة العملاء.

4. أهداف المشروع

رفع مستوى الطلاب :تحسين تحضير طلبة البكالوريا، ورفع نتائجهم عبر محتوى منظم يتناسب مع مستوى الطالب.

منظومة تقنية ذكية :تطوير منصة ومساعد ذكي يعتمدان على الذكاء الاصطناعي للإجابة بدقة وتحليل أداء الطلاب.

مؤسسة ناشئة واعدة :تأسيس شركة مبتكرة وقابلة للتوسع (داخل وخارج الجزائر)، تخلق فرص عمل وتدعم الاقتصاد الرقمي.

توسع ومواكبة مستقبلية :طموح للنمو ليشمل مستويات ومناهج جديدة، دعم لغات متعددة، وإطلاق تطبيقات هاتفية متقدمة.

5. الجدول الزمني لتحقيق المشروع

يمر مشروع DeepBac بعدة مراحل أساسية من أجل الوصول إلى منصة تعليمية متكاملة وقابلة للتوسع.

الأعمال الرئيسية	المدة الزمنية	المرحلة
دراسة السوق، تحليل احتياجات الطلبة، تحديد المتطلبات التقنية	شهر إلى شهرين	مرحلة دراسة الفكرة والتخطيط

تصميم واجهات المستخدم وقاعدة البيانات والبنية التقنية	شهر	مرحلة تصميم النظام
تطوير المنصة الأساسية والمساعد الذكي	3إلى 4 أشهر	مرحلة تطوير النسخة الأولية (MVP)
اختبار الأداء وتجربة المستخدم وتصحيح الأخطاء	شهر	مرحلة الاختبار والتجريب
إطلاق نسخة تجريبية لفئة محددة من الطلبة	شهر	مرحلة الإطلاق الأولي
تحسين الذكاء الاصطناعي وإضافة ميزات جديدة	مستمرة	مرحلة التحسين والتطوير
توسيع قاعدة المستخدمين وبناء شراكات تعليمية	بعد الإطلاق	مرحلة التوسع والتسويق

ويعتمد نجاح المشروع على التطوير المستمر والاستماع لملاحظات المستخدمين، مما يسمح بتحسين جودة الخدمات وضمان استدامة المنصة على المدى الطويل.

المحور الثاني: الجوانب الابتكارية

يُعدّ DeepBacمشروعًا مبتكرًا في تكنولوجيا التعليم (EdTech) بالجزائر، يتميز بدمج تقنيات الذكاء الاصطناعي مع التعليم الرقمي في منصة موحدة تقدم تجربة تفاعلية مخصصة لطلاب البكالوريا، متجاوزًا بذلك المنصات التقليدية ذات المحتوى الثابت.

1. الابتكار التكنولوجي والذكاء الاصطناعي

- مساعد تعليمي ذكي :توظيف الذكاء الاصطناعي التوليدي ومعالجة اللغات الطبيعية (NLP) بلغات متعددة (العربية والفرنسية) لشرح الدروس والإجابة عن الأسئلة.
- تخصيص المسار الدراسي :استخدام أنظمة التوصية الذكية لتصحيح الإجابات، اكتشاف نقاط القوة والضعف، واقتراح خطط مراجعة مخصصة وفق المنهاج الوطني الجزائري.

2. الابتكار في طريقة التعلم والتعليم الرقمي

- تغيير النمط التعليمي :الانتقال بالطالب من التلقين والحفظ التقليدي إلى التعلم الذاتي والتفاعلي.
- منظومة تقييم فورية :توفير اختبارات ذكية ومتابعة مستمرة للأداء لتقديم تغذية راجعة ترفع من فعالية التحصيل الدراسي.

3. الابتكار في البرمجيات وتحليل البيانات (تجربة المستخدم)

- منصة الكل في واحد :تجميع الدروس، التمارين، الاختبارات، والمساعد الذكي في مكان واحد سهل الاستخدام وعابر للأجهزة (هاتف وحاسوب).
- بنية برمجية متطورة :تطوير المنصة بتقنيات تضمن السرعة، الأداء العالي، حماية البيانات، والقابلية للتوسع البرمجي لاستيعاب أعداد ضخمة من المستخدمين.

4. الابتكار الاقتصادي وريادة الأعمال

- نموذج رقمي مرن :تقديم خدمات تعليمية واسعة النطاق بتكاليف تشغيلية منخفضة واشتراكات مرنة تناسب الجميع.
- مؤسسة ناشئة مستدامة :خلق قيمة اقتصادية في قطاع المعرفة، توفير فرص عمل تكنولوجية، والقدرة على التوسع نحو أسواق ومراحل تعليمية أخرى مستقبلاً.

5. الابتكار والأثر الاجتماعي

- تكافؤ الفرص التعليمية :تقليل الفوارق بتقديم تعليم حديث عالي الجودة للطلاب في المناطق النائية وبأسعار مناسبة.
- دعم التحول الرقمي :المساهمة في نشر الثقافة الرقمية ورفع نسب النجاح الدراسي في المجتمع بالاعتماد على التكنولوجيا الحديثة.

المحور الثالث: التحليل الاستراتيجي للسوق

1. عرض القطاع السوقي

ويستهدف المشروع بشكل أساسي سوق التعليم الثانوي في الجزائر، وخاصة فئة طلاب شهادة البكالوريا، باعتبارها من أكبر الفئات التعليمية وأكثرها طلبًا للدعم الدراسي والمرافقة التعليمية.

خصائص و اتجاهات السوق

- حجم سوق ضخم ومتنامي :استهداف كتلة بشرية كبرى تفوق 850ألف مترشح للبكالوريا سنويًا في الجزائر.
- مواكبة الاتجاهات الحديثة :يتماشى المشروع مع التحول الرقمي الحالي، والاعتماد المتزايد على الهواتف الذكية، وتنامي الوعي بالتعليم المخصص القائم على الذكاء الاصطناعي والتعلم عن بعد.

حجم الطلب والدو افع

- طلب متزايد على الحلول الرقمية :إقبال واسع على منصات المراجعة، تطبيقات التقييم، والمحتوى التعليمي الرقمي المبتكر.
- بديل اقتصادي ومرن :يمثل المشروع الحل الأمثل لمواجهة ضغط البكالوريا، وارتفاع تكاليف الدروس الخصوصية التقليدية، مع توفير ميزة "المتابعة الفردية" وسهولة الوصول.

الفئات المستهدفة بالمنصة

- الفئة الرئيسية :طلاب البكالوريا بجميع شعبهم في الجزائر، خاصة الباحثين عن الدعم الإضافي والتعلم الذاتي.
- الفئات الثانوية :أولياء الأمور الراغبين في متابعة مستوى أبنائهم، والأساتذة والمؤسسات المهتمة بالحلول التعليمية الرقمية.

2. قياس شدة المنافسة

يواجه مشروع DeepBac منافسة في سوق التعليم الرقمي، إلا أن طبيعة الحل المقترح تمنحه عدة نقاط قوة وتميز مقارنة بالمنافسين التقليديين والرقميين.

المنافسون المباشرون :

- المنصات التعليمية الخاصة بالبكالوريا.
- تطبيقات الدروس والاختبارات التعليمية.
- المؤسسات التي تقدم خدمات تعليمية رقمية.

المنافسون غير المباشرين:

- الدروس الخصوصية التقليدية.
- قنوات اليوتيوب التعليمية.
- مجموعات التعليم عبر وسائل التواصل الاجتماعي.
- المواقع التعليمية العامة.

3. تحليل SWOT

نقاط القوة(Strengths) :

- **مواكبة التوجه العالمي** :الاستفادة من التحول العالمي نحو التعليم المدعوم بالذكاء الاصطناعي.
- **ملائمة عالية للسوق المحلي** :التوافق التام والدقيق مع منهاج البكالوريا الجزائري (مما يحقق مواءمة ممتازة بين المنتج والسوق).
- **أسبقية الدخول للسوق** :امتلاك ميزة الريادة كأول منصة تعتمد على الذكاء الاصطناعي للتحضير للامتحانات في الجزائر.
- **تجربة تعلم مخصصة** :تقديم تجربة تعليمية مخصصة وفردية مدعومة بالذكاء الاصطناعي (التعلم التكيفي والتقييم الذكي).
- **فهم عميق للمستخدم** :المعرفة الدقيقة والعميقة باحتياجات الطلاب المحليين وطبيعة نظام الامتحانات الوطني.

نقاط الضعف(Weaknesses) :

- **منتج في مراحله الأولى** :المنصة ما زالت في طور البداية وتتطلب المزيد من التحقق الفني والمعايرة العلمية.
- **محدودية الموارد** :نقص الموارد المالية والتقنية في المرحلة الأولية للانطلاق.
- **تحديات دقة الذكاء الاصطناعي** :الاعتماد على نماذج لغوية قد تولّد أحيانًا مخرجات غير دقيقة أو خاطئة (الهلوسة البرمجية).
- **بناء الثقة الرقمية** :غياب الوعي بالعلامة التجارية وضعف ثقة المستخدمين بالمنصة في بداية الإطلاق.

الفرص(Opportunities) :

- **سوق ضخم ومتجدد** :وجود كتلة سوقية هائلة ومتكررة سنويًا من طلاب البكالوريا الجزائريين (طلب مرتفع ومستمر كل عام).
- **تقبل التكنولوجيا** :تزايد قبول الطلاب واستعدادهم للتعليم الرقمي والقائم على الذكاء الاصطناعي.
- **قابلية التوسع الأفقي** :إمكانية التوسع مستقبلاً لتشغيل مستويات تعليمية أخرى (التعليم المتوسط، الجامعي، والمسابقات الوطنية).
- **دمج تقنيات متقدمة** :فرصة دمج تكنولوجيات مستقبلية مثل الواقع المعزز والافتراضي (AR/VR) وأنظمة التدريس الذكية المتقدمة.

التهديدات(Threats) :

- **المنافسة الدولية :**دخول منصات التعليم الإلكتروني العالمية إلى السوق المحلي (مثل المساعدين التعليميين العامين القائمين على الذكاء الاصطناعي).
- **التطورالتكنولوجي المتسارع :**التسارع التقني الكبير الذي يتطلب تحديثات مستمرة واستثمارات دؤوبة لمواكبته.
- **القيود التنظيمية :**إمكانية ظهور عقبات أو تشريعات قانونية مقيّدة في قطاع التعليم والرقمنة.
- **القدرة الشرائية :**ضعف الرغبة أو القدرة على الدفع لدى بعض شرائح وفئات السوق المستهدف.
- **التقليد السريع :**خطر الاستنساخ السريع لفكرة المنصة وآلياتها من طرف المنافسين المحليين.

3. الاستراتيجية التسويقية

أهداف الاستراتيجية والتسعير

- **الأهداف التسويقية :**التعريف بالمنصة، جذب المستخدمين الأوائل، بناء الثقة مع الطلاب والأولياء، وزيادة الاشتراكات لتعزيز الولاء.
- **استراتيجية التسعير :(Freemium)** تقديم نسخة مجانية محدودة لتسهيل التجربة، مع توفير اشتراكات مرنة (شهرية/سنوية) تناسب مختلف الفئات وتحقق دخلًا مستدامًا.

استراتيجية المنتج والتوزيع

- **المنتج الفلتر :**منصة سهلة الاستخدام تقدم محتوى عالي الجودة ومساعدًا ذكيًا تفاعليًا ومخصصًا يتطور بناءً على تقييمات الطلاب.
- **التوزيع الرقمي :**تقديم الخدمات سحابيًا عبر منصة ويب وتطبيقات هاتفية (مستقبلاً) لضمان سهولة الوصول والتوسع اللوجستي دون تكاليف مادية ضخمة.

الترويج وبناء العلامة التجارية

- **الترويج الرقمي :**التركيز على صناعة المحتوى التعليمي والفيديوهات القصيرة، الحملات الإعلانية، والتعاون مع الأساتذة وصناع المحتوى، خاصة في فترة البكالوريا.
- **الهوية الذهنية :**ترسيخ المنصة كـ"رفيق رقمي ذكي" عبر هوية بصرية مميزة، وتصميم احترافي، ونشر تجارب وقصص نجاح الطلاب.

النمو والتوسع المستقبلي

- **التوسع الأفقي والعمودي** :إدراج مستويات تعليمية ومواد جديدة، وتطوير ميزات ذكاء اصطناعي أكثر تقدمًا.
- **الانتشار الإقليمي** :التخطيط لدخول أسواق عربية وإفريقية مستقبلاً لضمان استدامة المشروع وتنافسيته.

المحور الرابع :خطة الإنتاج والتنظيم

1. عملية الإنتاج

تمر عملية الإنتاج داخل المشروع بعدة مراحل أساسية، تتمثل فيما يلي:

- **جمع وتحضير المحتوى** :جمع وتنظيم الدروس والتمارين وفق المنهاج الجزائري، وتصميم الأسئلة التفاعلية مع مراجعتها لضمان الدقة العلمية والجودة.
- **تطوير المنصة الرقمية** :تصميم واجهات المستخدم، بناء قواعد البيانات، برمجة الخصائص الأساسية، ودمج أدوات الذكاء الاصطناعي لإنشاء منصة سريعة ومستقرة.
- **تطوير الأنظمة الذكية** :تدريب نماذج الذكاء الاصطناعي والمساعد التعليمي على فهم الأسئلة، وتطوير أنظمة التوصية المخصصة وتحليل بيانات الطلاب (العنصر الابتكاري الأساسي).
- **الاختبار وضمان الجودة** :فحص أداء المنصة وتجربة المستخدم، تصحيح الأخطاء التقنية، والتأكد التام من دقة إجابات المساعد الذكي قبل الإطلاق.
- **التشغيل والتحسين المستمر** :إطلاق المنصة، مراقبة الأداء، تحديث المحتوى بانتظام، وتطوير خوارزميات الذكاء الاصطناعي بناءً على احتياجات المستخدمين الفعلية.

2. التموين

- **الموارد التقنية والتكنولوجية** :الاعتماد على خوادم سحابية مرنة، قواعد بيانات، خدمات الذكاء الاصطناعي مفتوحة المصدر، أدوات الأمن السيبراني وتحليل البيانات؛ لضمان سرعة المنصة وقابليتها للتوسع حسب عدد المستخدمين.

- **الموارد التعليمية** :توفير الدروس، الملخصات، الاختبارات، والبيانات التعليمية اللازمة لتطوير الأنظمة الذكية، مع مطابقتها بالكامل للمنهاج الوطني الجزائري لضمان جودتها ودقتها العلمية.
- **معايير الاختيار التقني** :اختيار واجهات البرمجة والمكتبات وأطر التطوير الحديثة بناءً على أربعة معايير أساسية: الكفاءة العالية، القابلية للتوسع، الأمان والاستقرار، والتكلفة المناسبة.
- **إدارة التموين والتشغيل** :اعتماد آلية مرنة لمراقبة التكاليف التشغيلية، اختيار مزودي خدمات سحابية موثوقين، والتحديث الدوري للموارد لضمان استمرارية الخدمة وتقليل الأعطال التقنية.

3. اليد العاملة

يعتمد مشروع DeepBac على فريق عمل متعدد المهارات يجمع بين الجوانب التقنية، التعليمية، والإدارية.

أ. الهيكل البشري للمشروع

- **مدير المشروع** :يتولى الإدارة العامة والتسيير، التخطيط الاستراتيجي، متابعة مراحل التنفيذ، وإدارة العلاقات والشراكات الخارجية.
- **مهندس ذكاء اصطناعي** :يتولى تطوير وتدريب النماذج اللغوية الذكية والمساعد التعليمي، بناء أنظمة التوصية، وتحليل بيانات الطلاب لتحسين الأداء.
- **مهندس برمجيات** :يتكفل بتطوير واجهات المنصة (الويب والتطبيق مستقبلاً)، بناء الخوادم وقواعد البيانات، وصيانة النظام وضمان استقراره.
- **مصمم وصانع محتوى** :يتولى تصميم واجهات المستخدم(UI/UX) ، صياغة الهوية البصرية، وتنظيم وتحضير المحتوى التعليمي والاختبارات التفاعلية ومراجعتها علميًا بالتعاون مع الأساتذة.
- **مسوق رقمي** :يتكفل بالتسويق الرقمي وإدارة حسابات التواصل الاجتماعي، جذب المستخدمين، بناء العلامة التجارية، وتقديم الدعم الفني وخدمة العملاء.
- **محاسب** :يتولى الإدارة المالية للمشروع، مراقبة التكاليف التشغيلية، تتبع وإدارة مداخيل الاشتراكات، وضمان الاستقرار المالي للمؤسسة الناشئة.

ب. سياسة التوظيف

يعتمد المشروع على:

- اختيار الكفاءات الشابة.

- تشجيع الابتكار والعمل الجماعي.
- الاستفادة من العمل عن بعد.
- تطوير مهارات الفريق بشكل مستمر.

كما يسعى المشروع إلى خلق فرص عمل في مجالات التكنولوجيا والذكاء الاصطناعي والتعليم الرقمي.

ج. تطوراليد العاملة مستقبلاً

في المراحل الأولى سيتم الاعتماد على فريق صغير ومرن، ثم يتم توسيع الفريق تدريجيًا حسب:

- نمو عدد المستخدمين.
- توسع الخدمات.
- تطور المنصة.
- زيادة الاحتياجات التشغيلية.

4. الشراكات الرئيسية

- **الشراكات التعليمية** :التعاون مع الأساتذة المختصين، المدارس، وصنّاع المحتوى التعليمي؛ لضمان تقديم محتوى عالي الجودة ومطابق للمنهاج، وتعزيز مصداقية المنصة.
- **الشراكات التكنولوجية** :التنسيق مع مزودي الخدمات السحابية، منصات الذكاء الاصطناعي، ومطوري البرمجيات؛ لتحسين الأداء التقني، تطوير الأنظمة الذكية، وضمان استقرار الخدمة.
- **الشراكات التسويقية** :العمل مع المؤثرين، الصفحات التعليمية، ووكالات التسويق الرقمي؛ لزيادة الوعي بالمنصة، جذب المستخدمين الأوائل، ونشر العلامة التجارية.
- **الشراكات المؤسسية** :بناء علاقات مع حاضنات الأعمال، مراكز الابتكار الجامعية، والهيئات الداعمة للمؤسسات الناشئة؛ للحصول على المرافقة وتوسيع فرص الاستثمار والنمو.

المحورالخامس: الخطة المالية PLAN FINANCIER

تعتمد الخطة المالية لمشروع DeepBacعلى نموذج اقتصادي رقمي قائم على الاشتراكات، مع استهداف شريحة من طلاب البكالوريا في الجزائر. وقد تم إعداد هذه الخطة اعتمادًا على سوق مستهدف يقدر بـ 8700طالب (1% من إجمالي المترشحين للبكالوريا)، بسعر اشتراك شهري قدره 2000دج لمدة 9أشهر دراسية سنويًا.

1. التكاليف والأعباء

أ. التكاليف الاستثمارية (مرة واحدة)

تشمل التكاليف المرتبطة بإنشاء المشروع وإطلاقه:

العنصر	سعر الوحدة(دج)	الكمية	التكلفة (دج)
تجهيز مقر الشركة	400,000	1	400,000
أجهزة حواسيب	150,000	6	900,000
عتاد وتجهيزات مكتبية (ملحقات اعلام ألي، مكاتب, ...)	50,000	6	300,000
تصميم وهوية بصرية	30,000	1	30,000
تطوير المنصة والتطبيق	1,000,000	1	1,000,000
مصاريف قانونية وإدارية	100,000	1	100,000

إجمالي التكاليف الاستثمارية:

2,730,000دج

ب. التكاليف التشغيلية الشهرية (لسوق 8700 طالب)

1. الرواتب (فريق أساسي)

المنصب	الراتب (دج)
مهندس برمجيات	120,000
مهندس ذكاء اصطناعي	140,000
مصمم وصانع محتوى	80,000
مسوق رقمي	90,000
محاسب	60,000
مدير المشروع	100,000

إجمالي الرواتب:

590000 DA

.2الأساتذة المتعاونون (10 أساتذة)

10 * 35000 DA = 350000 DA

ج. التكاليف التقنية والتشغيلية

العنصر	التكلفة (دج)
إستضافة المنصة الالكترونية	600,000
خدمات ذكاء اصطناعي	2,000,000
تسويق رقمي	1,200,000
دعم فني وخدمة عملاء	300,000
الفواتير (الانترانت , الكهرباء , كراء المقر,...)	110,000
صيانة وتحديثات	150,000
مصاريف متنوعة	200,000

إجمالي التكاليف التشغيلية :

4560000 دج

إجمالي التكاليف التشغيلية الشهرية

590000 + 350000 + 4560000 = 5500000

5,500,000دج شهريًا

إجمالي التكاليف السنوية

5500000 * 12 = 66000000

66,000,000دج سنويًا

مصاريف إضافية سنوية

العنصر	التكلفة (دج)
تطوير وتحسينات في المنصة	4,000,000
تجديد عتاد	1,500,000
توسعة البنية التحتية	2,500,000

إجمالي إضافي:

8000000 دج

إجمالي الأعباء السنوية

66000000 + 8000000 = 74000000

74,000,000دج

هيكل التمويل

يعتمد مشروع **DeepBac**على هيكل تمويل مختلط يجمع بين التمويل الذاتي، الدعم العائلي، والتمويل الخارجي، وذلك من أجل ضمان توفير الموارد المالية اللازمة لإطلاق المشروع وتطويره خلال المراحل الأولى.

جدول هيكل التمويل

مصدر التمويل	النسبة	الدور في تمويل المشروع
التمويل الذاتي (أموال خاصة)	10%	تغطية جزء من التكاليف الأولية وإظهار جدية صاحب المشروع
التمويل العائلي	20%	المساهمة في تجهيز المقر وشراء العتاد الأساسي
تمويلASF	70%	تمويل تطوير المنصة، البنية التحتية، الذكاء الاصطناعي، والتوسع الأولي

2. رقم الأعمال

الفرضيات الأساسية

- عدد الطلبة 8700 :طالب
- سعر الاشتراك 2000 :دج

- مدة الاشتراك 9 :أشهر سنويًا

أ. رقم الأعمال الشهري

8700 * 2000 = 17400000

17,400,000دج شهريًا

ب. رقم الأعمال السنوي

17400000 * 9 = 156600000

156,600,000دج سنويًا

3. جدول حسابات النتائج المتوقع

	N+1	N+2	N+3	N+4	N+5
Quantité produit A	8,700	20,000	40,000	60,000	87,000
Prix HT produit A	18,000	18,000	18,000	18,000	18,000
Ventes produit A	156,600,000	360,000,000	720,000,000	1,080,000,000	1,566,000,000

أ. النتيجة الإجمالية

البيان	القيمة
رقم الأعمال	156,600,000
التكاليف السنوية	74,000,000

النتيجة الصافية:

156600000 - 74000000 = 82600000

82,600,000دج ربح صافي سنوي

ب. هامش الربح :حوالي 53%

4. خطة الخزينة

أ. الاحتياجات الشهرية

البند	القيمة
التكاليف التشغيلية	5,500,000
احتياطي سيولة	500,000

إجمالي الاحتياج الشهري:

6,000,000دج

ب. نقطة التعادل : حوالي 2750 طالب

ج. تحليل الخزينة

يتميز المشروع بـ:

- قدرة على تغطية التكاليف عند بلوغ 2750 طالب فقط.
- نمو كبير في الربحية عند التوسع نحو 8700 طالب.
- نموذج مالي مرن يعتمد على الاشتراكات المتكررة.
- قابلية عالية لإعادة الاستثمار والتوسع.

خاتمة مالية

تُظهر الدراسة أن مشروع DeepBacيمتلك:

- نموذج اقتصادي قوي.
- قابلية عالية للربحية (أكثر من 80 مليون دج سنويًا عند 1% من السوق).
- قدرة على التوسع مع زيادة عدد المستخدمين دون زيادة موازية في التكاليف.

وبذلك يُعتبر المشروع مؤسسة ناشئة قابلة للنمو والاستدامة في قطاع التعليم الرقمي بالجزائر.

المحور السادس : النموذج الاولي التجريبي

يمثل النموذج الأولي التجريبي لمشروع **DeepBac** مرحلة محورية في مسار تطوير المؤسسة الناشئة، حيث تم الانتقال من مرحلة الفكرة والتصميم إلى منتج رقمي فعلي قابل للاستخدام من طرف الطلبة في بيئة واقعية. ويهدف هذا النموذج إلى اختبار مدى فعالية المنصة، وتحسين وظائفها اعتمادًا على تجربة المستخدمين الفعلية.

1. إطلاق النسخة التجريبية للمنصة

تم رسميًا إطلاق النسخة التجريبية (Beta Version) من منصة **DeepBac**، وهي متاحة حاليًا عبر الإنترنت على الرابط:

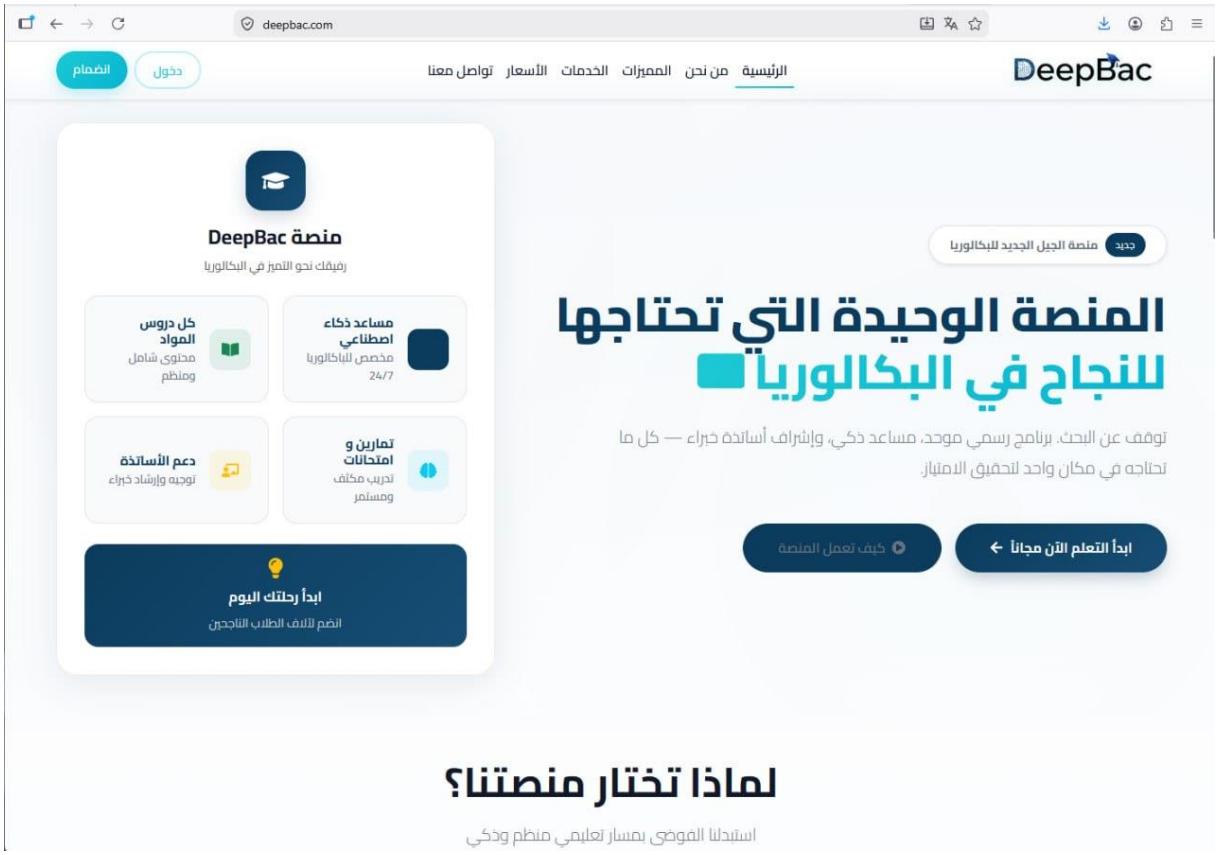
deepbac.com

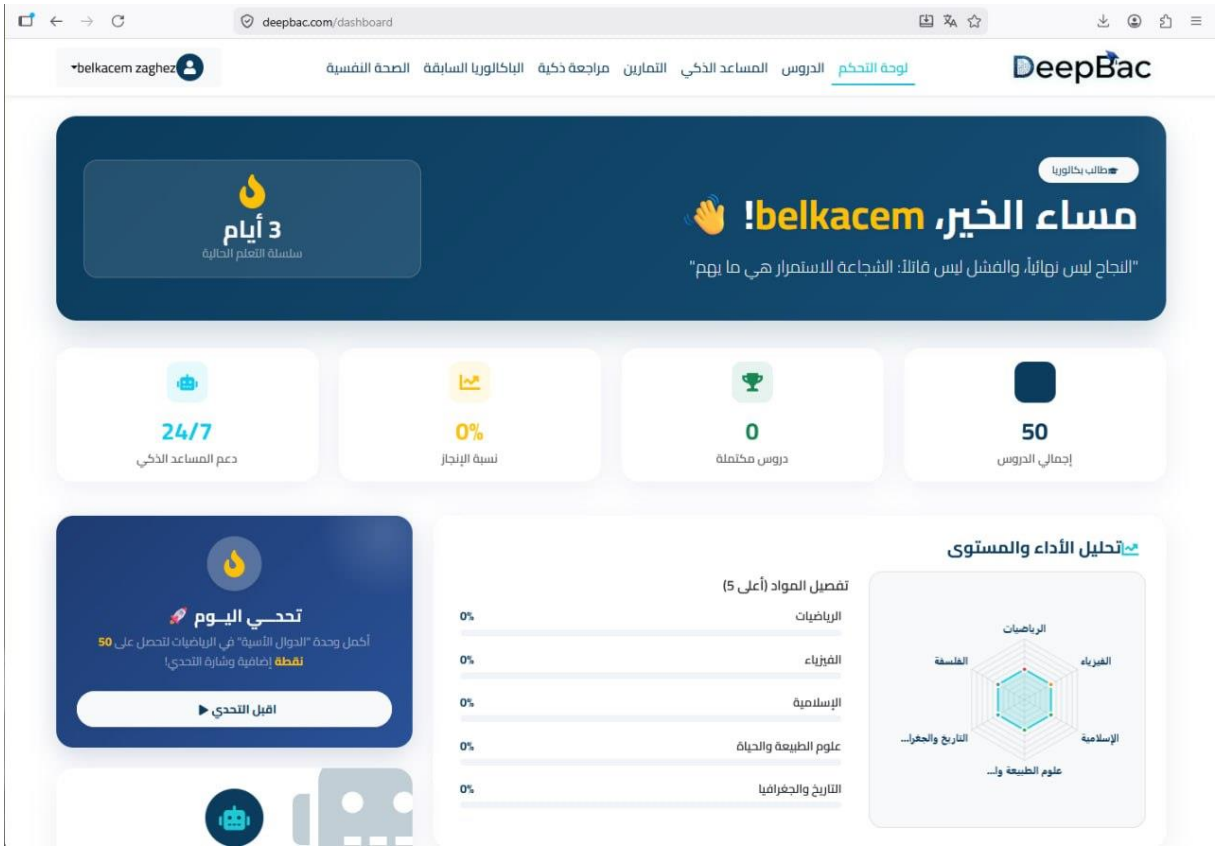
وتتيح هذه النسخة للطلبة:

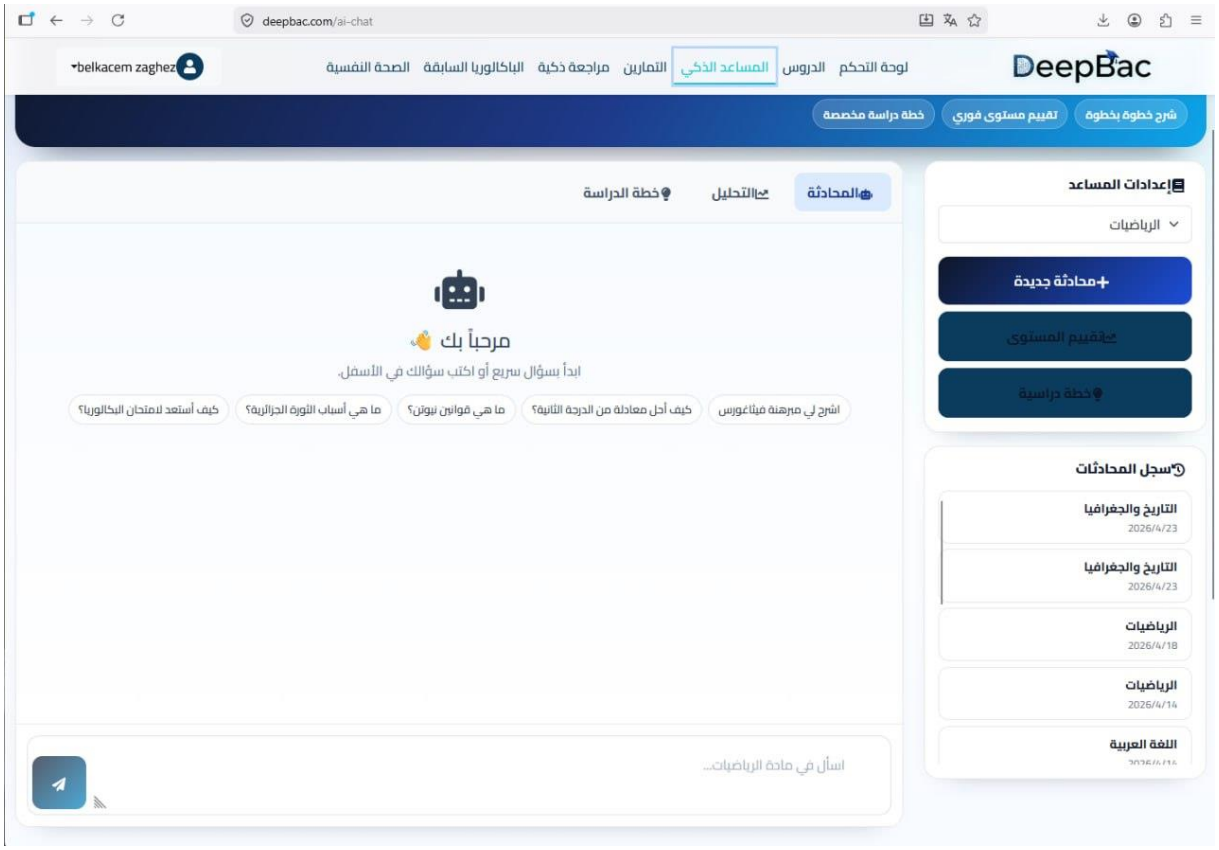
- التسجيل وإنشاء حسابات شخصية.
- الوصول إلى الدروس والمواد التعليمية.
- استخدام المساعد الذكي.
- إجراء الاختبارات التفاعلية.
- متابعة تقدمهم الدراسي.

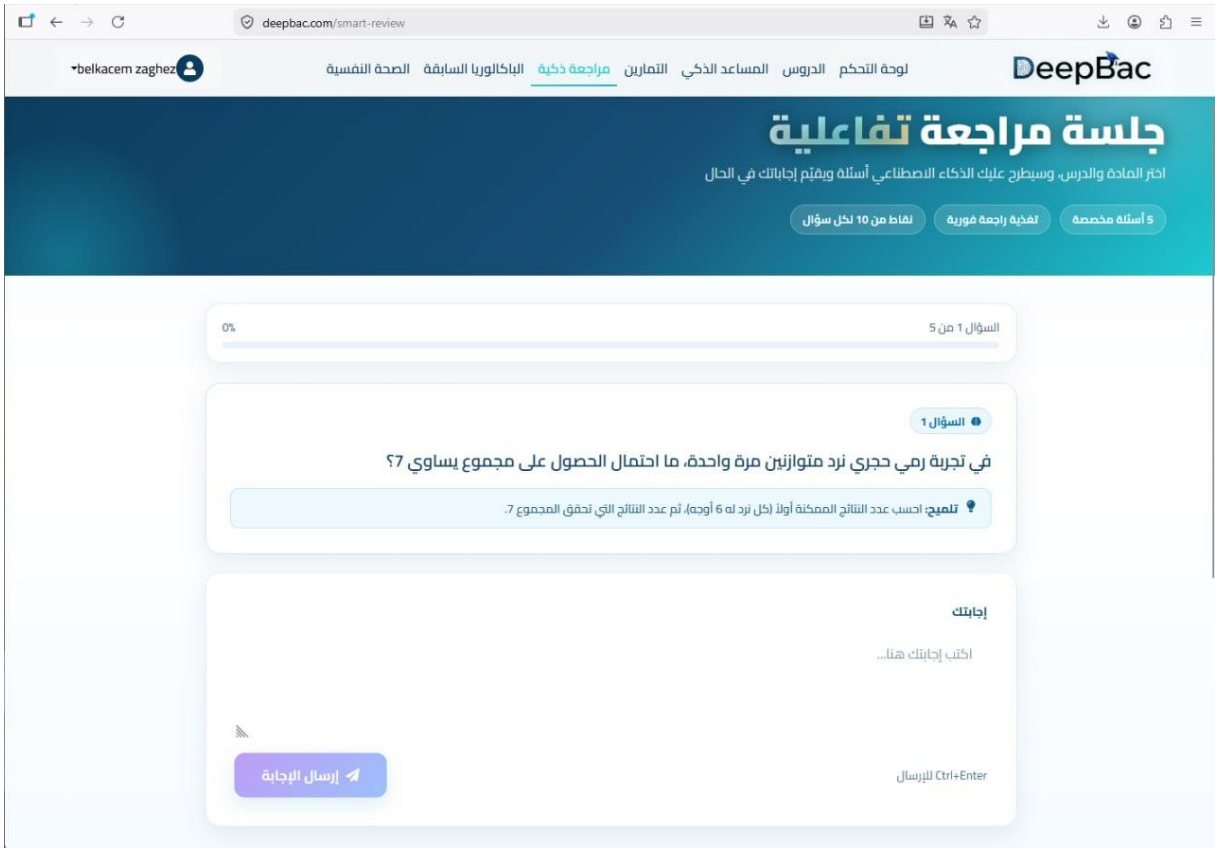
ويهدف هذا الإطلاق إلى:

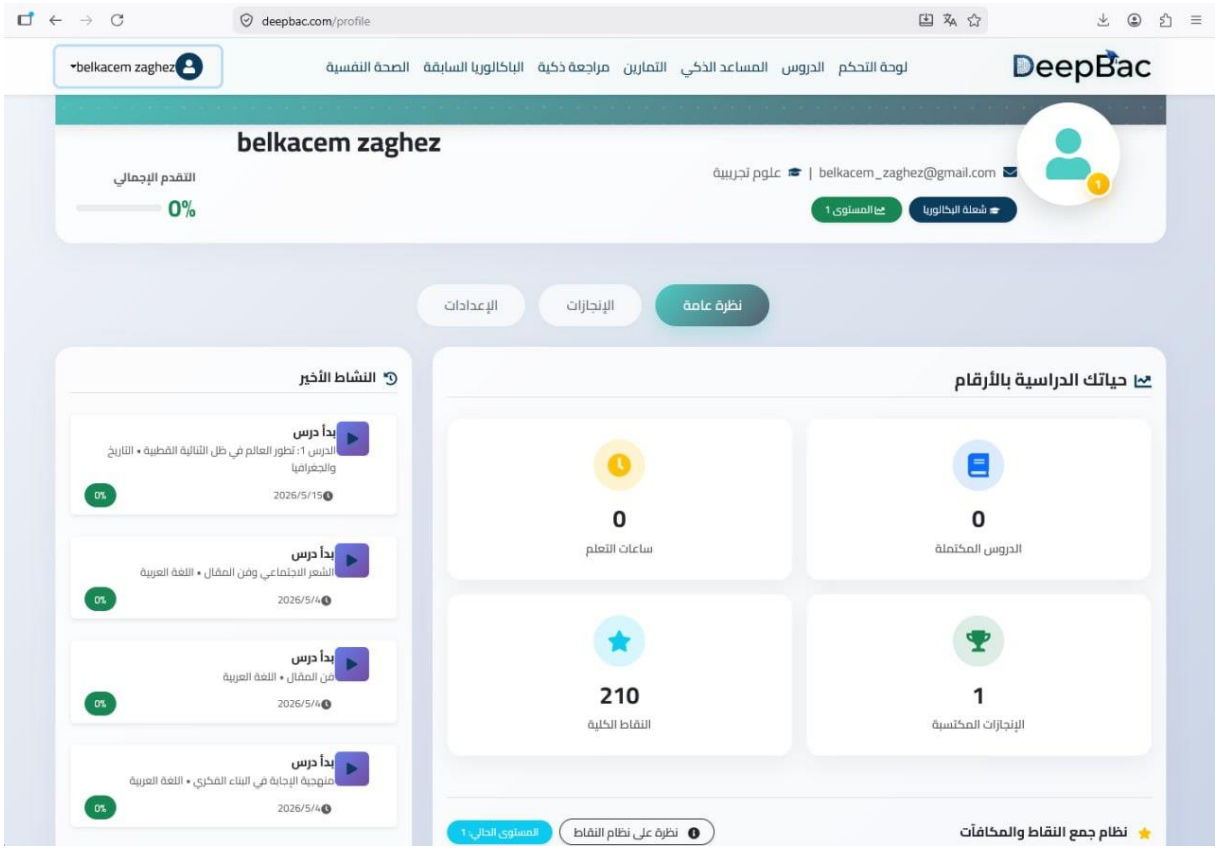
- اختبار المنصة في بيئة حقيقية.
- جمع بيانات الاستخدام.
- تحسين الأداء وتجربة المستخدم.
- تطوير الميزات حسب احتياجات الطلبة.


deepbac.com
DeepBac
الرئيسية من نحن المميزات الخدمات الأسعار تواصل معنا
دخول
انضمام
منصة DeepBac
رفيقك نحو التميز في البكالوريا
مساعد ذكاء اصطناعي
مخصص للباكالوريا 24/7
كل دروس المواد
محتوى شامل ومنظم
تمارين و امتحانات
تدريب مكثف ومستمر
دعم الأساتذة
توجيه وإرشاد خبراء
ابدأ رحلتك اليوم
انضم لآلاف الطلاب الناجحين
جديد منصة الجيل الجديد للبكالوريا
المنصة الوحيدة التي تحتاجها للنجاح في البكالوريا
توقف عن البحث. برنامج رسمي موحد، مساعد ذكي، وإشراف أساتذة خبراء — كل ما تحتاجه في مكان واحد لتحقيق الامتياز.
ابدأ التعلم الآن مجاناً ←
كيف تعمل المنصة
لماذا تختار منصتنا؟
استبدلنا الفوضى بمسار تعليمي منظم وذكي


deepbac.com/dashboard
DeepBac
belkacem zaghez
لوحة التحكم الدروس المساعد الذكي التمارين مراجعة ذكية الباكالوريا السابقة الصحة النفسية
طالب بكالوريا
مساء الخير، belkacem! 👋
"النجاح ليس نهائياً، والفشل ليس قاتلاً: الشجاعة للاستمرار هي ما يهم"
3 أيام
سلسلة التعلم الحالية
50
إجمالي الدروس
0
دروس مكتملة
0%
نسبة الإنجاز
24/7
دعم المساعد الذكي
تحـــدي اليـــوم
أكمل وحدة "الدوال الأسية" في الرياضيات لتحصل على 50 نقطة إضافية وشارة التحدي!
اقبل التحدي ▶
تحليل الأداء والمستوى
تفصيل المواد (أعلى 5)
الرياضيات 0%
الفيزياء 0%
الإسلامية 0%
علوم الطبيعة والحياة 0%
التاريخ والجغرافيا 0%
الرياضيات
الفيزياء
الإسلامية
علوم الطبيعة وا...
التاريخ والجغرا...
الفلسفة


deepbac.com/ai-chat
DeepBac
لوحة التحكم
الدروس
المساعد الذكي
التمارين
مراجعة ذكية
الباكالوريا السابقة
الصحة النفسية
belkacem zaghez
شرح خطوة بخطوة
تقييم مستوى فوري
خطة دراسة مخصصة
إعدادات المساعد
الرياضيات
محادثة جديدة
تقييم المستوى
خطة دراسية
سجل المحادثات
التاريخ والجغرافيا
2026/4/23
التاريخ والجغرافيا
2026/4/23
الرياضيات
2026/4/18
الرياضيات
2026/4/14
اللغة العربية
المحادثة
التحليل
خطة الدراسة
مرحباً بك
ابدأ بسؤال سريع أو اكتب سؤالك في الأسفل.
اشرح لي مبرهنة فيثاغورس
كيف أحل معادلة من الدرجة الثانية؟
ما هي قوانين نيوتن؟
ما هي أسباب الثورة الجزائرية؟
كيف أستعد لامتحان البكالوريا؟
اسأل في مادة الرياضيات...


deepbac.com/smart-review
DeepBac
لوحة التحكم
الدروس
المساعد الذكي
التمارين
مراجعة ذكية
الباكالوريا السابقة
الصحة النفسية
belkacem zaghez
جلسة مراجعة تفاعلية
اختر المادة والدرس، وسيطرح عليك الذكاء الاصطناعي أسئلة ويقيّم إجاباتك في الحال
5 أسئلة مخصصة
تغذية راجعة فورية
نقاط من 10 لكل سؤال
السؤال 1 من 5
0%
السؤال 1
في تجربة رمي حجري نرد متوازنين مرة واحدة، ما احتمال الحصول على مجموع يساوي 7؟
تلميح: احسب عدد النتائج الممكنة أولاً (كل نرد له 6 أوجه)، ثم عدد النتائج التي تحقق المجموع 7.
إجابتك
اكتب إجابتك هنا...
إرسال الإجابة
Ctrl+Enter للإرسال



2. عرض المنصة في الفعاليات والمؤتمرات

في إطار الترويج للمشروع وعرض نتائجه الأولية، تمت المشاركة بمنصة **DeepBac** في:

الملتقى الوطني للإبتكار الصناعي و التحول الطاقوي

الذي أقيم خلال شهر **أفريل 2026** في ولاية **بسكرة**.

وقد تم خلال هذا الحدث:

- عرض فكرة المشروع أمام نخبة من الباحثين والمهتمين بالابتكار.
- تقديم شرح تفصيلي حول المنصة ووظائفها.
- إبراز دور الذكاء الاصطناعي في التعليم.
- التعريف بالنموذج الأولي وخصائصه التقنية.

وقد لقي المشروع اهتمامًا إيجابيًا باعتباره نموذجًا مبتكرًا في مجال التعليم الذكي







3. صفحة الإدارة ونتائج الاستخدام الأولي

تم تطوير لوحة تحكم إدارية (Admin Dashboard) تسمح بمتابعة نشاط المنصة وتحليل البيانات، وقد أظهرت النتائج الأولية تسجيل أكثر من:

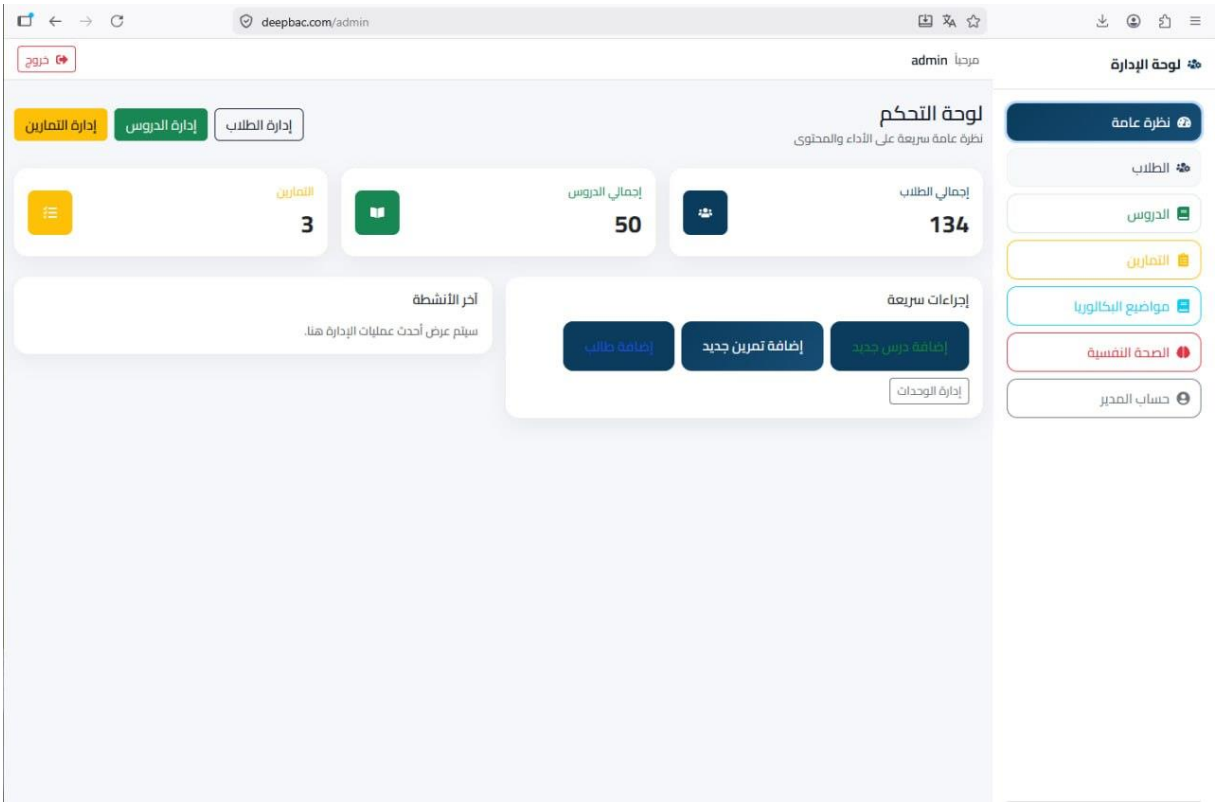
130طالبًا

في المرحلة التجريبية الأولى، وهو ما يعكس وجود اهتمام حقيقي بالمنصة وإقبال أولي جيد من طرف المستخدمين.

وتوفر لوحة التحكم إمكانية:

- متابعة المستخدمين.
- تحليل الأداء.
- مراقبة التفاعل مع المحتوى.

- إدارة الدروس والاختبارات.



.4استمارة التقييم التجريبية

في إطار تحسين جودة المنصة، تم نشر استمارة تقييم تجريبية موجهة للمستخدمين، تهدف إلى جمع آرائهم حول تجربة الاستخدام.

أهداف الاستمارة:

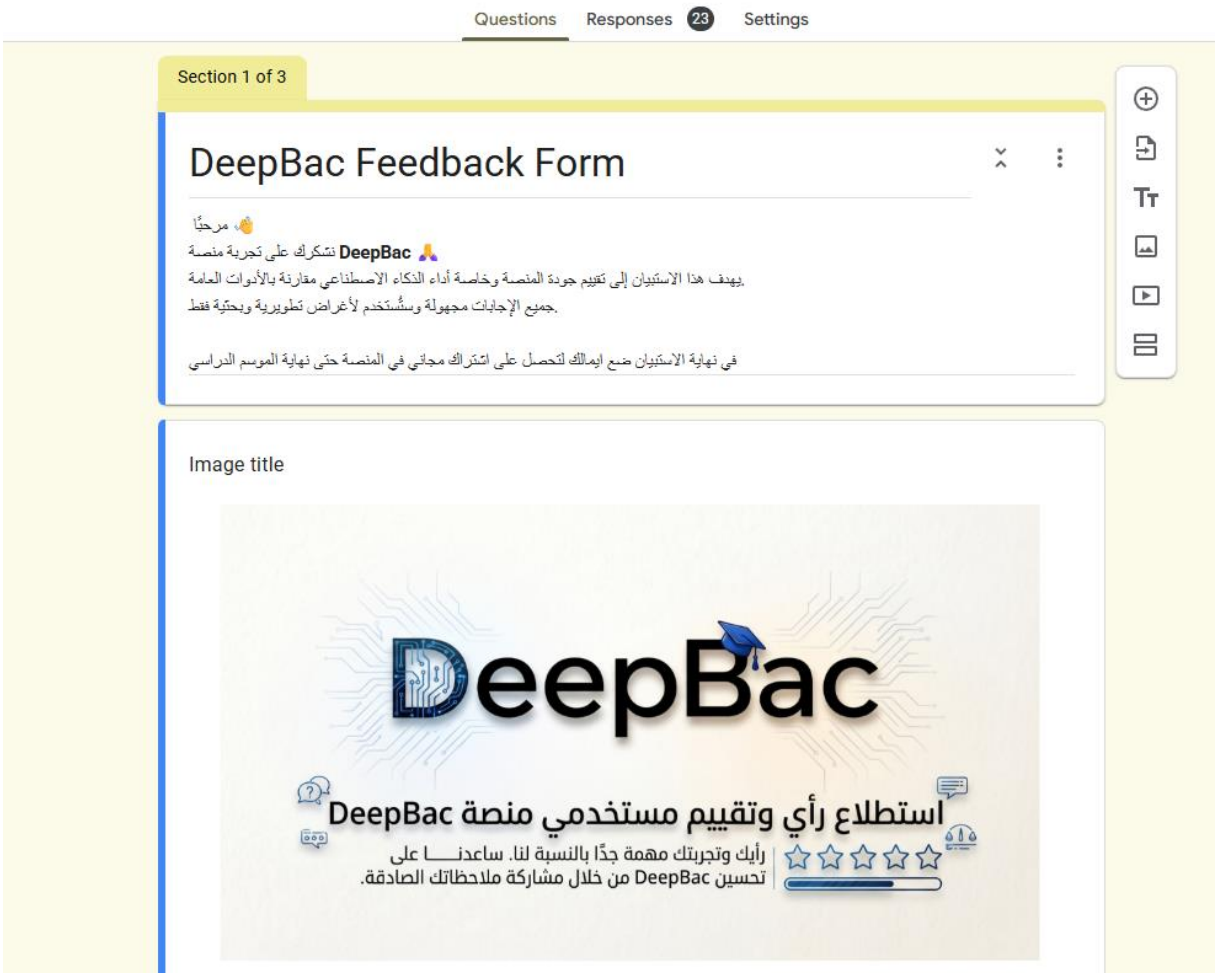
- تحديد نقاط القوة في المنصة.
- اكتشاف النقائص والمشاكل التقنية.
- قياس رضا المستخدمين.
- اقتراح تحسينات مستقبلية.

وتشمل الاستمارة أسئلة حول:

- سهولة الاستخدام.

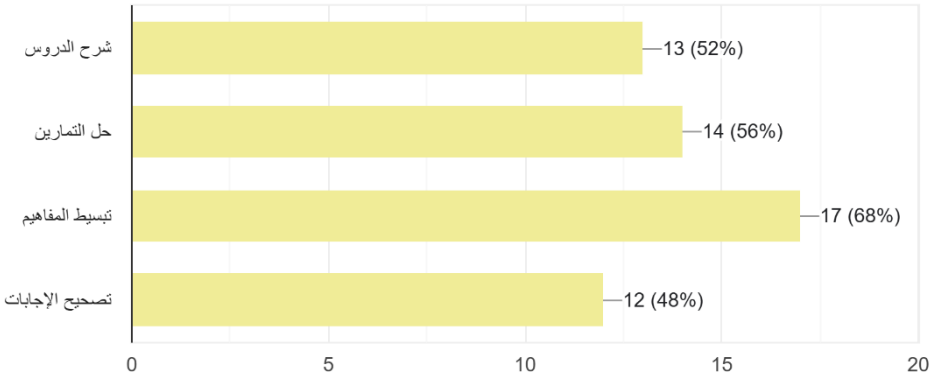
- جودة المحتوى التعليمي.
- فعالية المساعد الذكي.
- سرعة الأداء.
- التجربة العامة للمنصة.

ويتم اعتماد نتائج هذه الاستمارة كمرجع أساسي في تطوير النسخة النهائية من المشروع.



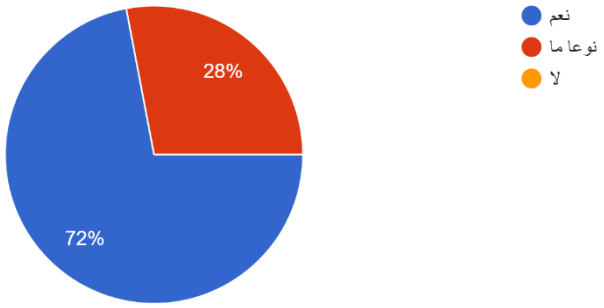
ماهي الجوانب التي تفوقت فيها المنصة

25 responses



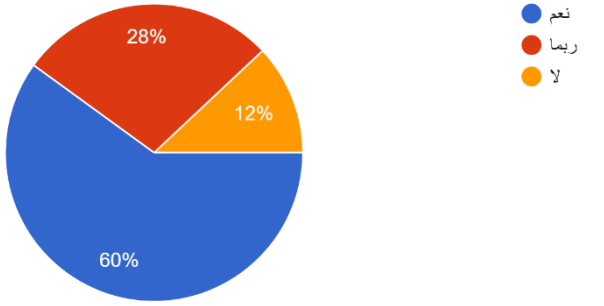
هل ساعدتك المنصة في فهم الدروس؟

25 responses



هل أنت مستعد للدفع مقابلها؟

25 responses



5. شهادة التوطين (حاضنة أعمال جامعة بسكرة)

تحصل مشروع **DeepBac** على شهادة توطين داخل **حاضنة أعمال جامعة بسكرة**، مما يعكس اعترافًا رسميًا بالمشروع كمؤسسة ناشئة واعدة.

أهمية التوطين:

- دعم أكاديمي وتقني للمشروع.
- مرافقة في تطوير النموذج الأولي.
- تعزيز مصداقية المشروع.
- تسهيل الوصول إلى شبكات التمويل والدعم.
- توفير بيئة عمل محفزة للابتكار.

الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة محمد خيضر - بسكرة

حاضنة أعمال جامعة بسكرة

رقم 02 / الحاضنة /2026

شهـــادة توطيـــن /تحضيـــن "مشروع مبتكر ضمن قرار 1275"

انا الممضي أسفله، السيد (ة): **محمد جلاب**

مسير(ة) حاضنة الأعمال: **جامعة محمد خيضر – بسكرة-**

المقر الاجتماعي / العنوان: **المجمع الإداري المقابل لكلية العلوم الاقتصادية، الطابق الثاني، جامعة بسكرة 07000**

رقم علامة الحاضنة: **2311223051**

تاريخ تسليم العلامة: **23 نوفمبر2022**

أشهد أن الطالب / الطلبة التالية أسمائهم :

الاسم و اللقب	الطور الدراسي	التخصص	الكلية
بلقاسم أشرف زاغز	ماستر 2	ذكاء اصطناعي	كلية العلوم الدقيقة

تحت إشراف الاستاذ/الاساتذة التالية أسمائهم :

الاسم و اللقب	الرتبة	التخصص	الكلية
العيد كحلول	أستاذ	هندسة البرامج	كلية العلوم الدقيقة
إيمان يوكانة	أستاذ محاضر ب	ذكاء اصطناعي	كلية العلوم الدقيقة
محمد جلاب	أستاذ محاضر أ	إدارة أعمال	كلية العلوم الاقتصادية
رفيعة دبابش	أستاذ محاضر ب	نقود وتمويل	كلية العلوم الاقتصادية
صولي رشيدة بن عبد العزيز	أستاذ	ذكاء اصطناعي	كلية العلوم الدقيقة

تم احتضانه على مستوى *حاضنة أعمال* جامعة محمد خيضر - بسكرة بمشروع تحت اسم :

design and development of a domain –specific language model as a privat tutor using LLM adaptation technique

بيئة تعليمية تفاعلية مبتكرة مدعومة بالذكاء الاصطناعي لطلاب الباكالوريا في الجزائر - DEEPBAC

خلال السنة الجامعية 2026/2025 .

سلمت هذه الشهادة بطلب من المعني للإدلاء بها في حدود ما يسمح به القانون .

حرر في : بسكرة بتاريخ: 2026/03/17

مدير الحاضنة

جامعة محمد خيضر بسكرة – حاضنة المؤسسات الناشئة Start Up Incubator – MOHAMED KHIDER UNIVERSITY OF BISKRA

.6خلاصة النموذج الأولي

أثبتت المرحلة التجريبية لمشروع DeepBac ما يلي:

- إمكانية تطبيق الفكرة بشكل واقعي وفعّال.
- وجود اهتمام حقيقي من طرف الطلبة.
- نجاح النسخة الأولية في جذب المستخدمين.
- قابلية تطوير النظام وتحسينه باستمرار.

كما أن إطلاق النسخة التجريبية عبر الإنترنت، والمشاركة في الفعاليات الوطنية، واعتماد استمارة تقييم المستخدمين، كلها عناصر تؤكد انتقال المشروع إلى مرحلة النضج التدريجي كمؤسسة ناشئة قابلة للنمو والتوسع.

الملحق رقم 04: نموذج العمل التجاري

شرائح العملاء

- طلاب البكالوريا الجزائريون (جميع المسارات)
- أولياء الأمور الراغبون في نجاح أبنائهم في امتحان البكالوريا، أو حتى تفوقهم بحصولهم على درجات عالية
- المؤسسات التعليمية والمعلمون (للتوسع مستقبلاً)

العلاقات مع العملاء

- دعم تعليمي شخصي مدعوم بالذكاء الاصطناعي
- تقييم فوري ومتابعة التقدم
- مساعدة آلية مع دعم بشري عند الحاجة
- تحسين مستمر بناءً على أداء المستخدم

القنوات

- منصة ويب (مصممة للهواتف المحمولة أولاً)
- تطبيق جوال (مستقبلاً)
- منصات التواصل الاجتماعي (فيسبوك، إنستغرام، تيك توك)
- مجتمعات تعليمية وشبكات طلابية

القيم المقترحة

- تعليم شخصي مُصمم خصيصًا لمستوى كل طالب
- شروحات دروس وتمارين مُولّدة بتقنية الذكاء الاصطناعي
- تصحيح ذكي مع تغذية راجعة مُفصّلة
- تحضير مُركّز على الامتحانات مُتوافق مع منهج البكالوريويا
- بديل مُتاح وفعّال من حيث التكلفة للدروس الخصوصية

الأنشطة الرئيسية

- تطوير المنصة
- تصميم وتحسين نماذج التدريس والتصحيح بالذكاء الاصطناعي
- مواءمة المحتوى مع منهج البكالوريا الجزائرية
- أمن المنصة، وحماية البيانات، ومراقبة النظام
- اختبار وتحسين واجهة المستخدم وتجربة المستخدم

الموارد الرئيسية

- نماذج الذكاء الاصطناعي والبنية التحتية السحابية
- خدمات الاستضافة
- فريق تطوير البرمجيات (واجهة المستخدم، الواجهة الخلفية، الذكاء الاصطناعي)
- معرفة المحتوى التربوي والمناهج الدراسية

الشراكات الرئيسية

- معلمو المرحلة الثانوية والمستشارون الأكاديميون
- المؤسسات التعليمية ومراكز التدريب
- مقدمو خدمات الحوسبة السحابية (Azure، oud)
- الجامعات، حاضنات الأعمال، ومراكز الابتكار
- مراجعو المحتوى وخبراء المناهج

مصادر الإيرادات

- اشتراكات الطلاب الشهرية والسنوية
- ميزات مميزة وتحليلات تعليمية متقدمة
- شراكات تعليمية وتراخيص مؤسسية
- خطط اشتراك مستقبلية للشركات والمدارس

هيكل التكاليف

- تكاليف تطوير المنصة ونموذج الذكاء الاصطناعي
- تكاليف البنية التحتية السحابية والاستضافة
- تكاليف التسويق واكتساب المستخدمين
- تكاليف الصيانة والأمن والتشغيل