People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
Mohamed Khider University – Biskra
Faculty of Exact Sciences and Sciences of Nature and Life
Computer Science Department

Order Number : RTIC07/M2/2023

Manuscript

Presented to obtain the academic master's degree in

COMPUTER SCIENCE

Option : RTIC

WorkersSpace Platform

Prepared By :

HARZALLAH MOHAMED ANES
SIFE EDDINE DAOUDI
HACENE MOHAMED
BENSAHBANE AYMEN
SALEM CHAKER

REZEG Khaled	**Prof**	President
MERIZIG Abdelhak	**MCB**	Supervisor
TERRISSA Sadek Labib	**Prof**	Co-Supervisor
BENDAHMANE Toufik	**MAA**	Co-Supervisor
YOUKANA Imane	**MCB**	Member

Academic year : 2022-2023

Acknowledgement

First and foremost, we express our gratitude to Allah, the Lord of the worlds, for His abundant blessings and guidance throughout this thesis journey.

To our parents, we extend our heartfelt thanks for instilling in us a love for knowledge and consistently encouraging us to strive for excellence. Your unwavering belief in our abilities has been the driving force behind our accomplishments. We are forever grateful for the sacrifices you have made to provide us with the best education and for your continuous support, including proofreading our drafts, offering valuable advice, and listening to our ideas.

We would also like to extend our sincere appreciation to our thesis advisors, Abdelhak Merizig, Terissa Sadek Labib, and Bendahmane Toufik. Your guidance, expertise, and patience have been instrumental in shaping this thesis. Your insightful feedback and dedicated support have helped us refine our ideas and go beyond our limits.

To our friends and classmates, we are grateful for the close friendship, intellectual discussions, and shared experiences. Your unwavering support and understanding during these challenging times have made this journey more enjoyable and meaningful.

Lastly, we acknowledge and appreciate the contributions of all the participants who generously dedicated their time and shared their insights for this study. Without their willingness to be part of our research, this thesis would not have been possible.

Thank you all for your invaluable contributions, guidance, and support.

Abstract

The field of digitalization has recently witnessed a radical shift in how to promote various businesses with the advent of modern technology. With the advancement of digital technology, digital platforms for craftsmen and self-employed workers have emerged in order to highlight their work and communicate with their customers smoothly. And in this thesis, we created a digital platform that contributes to promoting professionals and highlighting their services by providing an easy-to-use and easy-to-navigate platform that allows craftsmen to display their work and services to potential clients. The platform also includes features such as customer reviews, scheduling, payment processing, and a space for craftsman-customer chat. One of the main features of this platform is that it is compatible with all segments of society. On the other hand, our platform contains the latest technology, such as the recommendation system, which helps customers find craftsmen of interest to them, through their interactions with the platform, such as conducting searches for craftsmen within the platform, as well as the various evaluations that customers give to craftsmen. The platform also contains a chatbot that enables customers and craftsmen to answer their inquiries related to the platform. As for securing the data of customers and craftsmen, and the various transfers between them, we used blockchain technology.Regarding developing and testing the platform and its various services we created our own private cloud.

key words : *Digital platforms, Recommendation system, Blockchain, Chatbot, Private cloud* .

Résumé

Le domaine de la numérisation a récemment été témoin d'un changement radical dans la façon de promouvoir diverses entreprises avec l'avènement des technologies modernes. Avec l'avancée de la technologie numérique, des plateformes numériques destinées aux artisans et travailleurs indépendants ont émergé afin de mettre en valeur leur travail et de communiquer facilement avec leurs clients. Dans cette thèse, nous avons créé une plateforme numérique qui contribue à promouvoir les professionnels et à mettre en valeur leurs services en fournissant une plateforme facile à utiliser et à naviguer, permettant aux artisans de présenter leur travail et leurs services aux clients potentiels. La plateforme comprend également des fonctionnalités telles que les avis des clients, la planification, le traitement des paiements et un espace de discussion entre l'artisan et le client. L'une des principales caractéristiques de cette plateforme est sa compatibilité avec tous les segments de la société. D'autre part, notre plateforme intègre les dernières technologies, telles que le système de recommandation, qui aide les clients à trouver des artisans qui les intéressent, grâce à leurs interactions avec la plateforme, comme la recherche d'artisans sur la plateforme et les différentes évaluations que les clients donnent aux artisans. La plateforme comprend également un chatbot qui permet aux clients et aux artisans de répondre à leurs demandes liées à la plateforme. En ce qui concerne la sécurisation des données des clients et des artisans, ainsi que les différents transferts entre eux, nous avons utilisé la technologie de la blockchain. Pour le développement et le test de la plateforme et de ses différents services, nous avons créé notre propre cloud privé.

Mots clés : *Plateformes numériques, Système de recommandation, Blockchain, Chatbot, Cloud privé .*

الملخص

مجال الرقمنة شهد مؤخرًا تغيرًا جذريًا في كيفية الترويج للأعمال المختلفة مع ظهور التكنولوجيا الحديثة. مع تقدم التكنولوجيا الرقمية، ظهرت منصات رقمية للحرفيين والعاملين الذين يعملون بحسابهم الخاص من أجل إبراز عملهم والتواصل مع عملائهم بسلاسة. وفي هذه الرسالة، قمنا بإنشاء منصة رقمية تسهم في تعزيز المهنيين وتسليط الضوء على خدماتهم من خلال توفير منصة سهلة الاستخدام وسهلة الملاحة تتيح للحرفيين عرض أعمالهم وخدماتهم للعملاء المحتملين. تشمل المنصة أيضًا ميزات مثل آراء العملاء وجدولة المواعيد ومعالجة الدفع ومساحة للدردشة بين الحرفي والعميل. واحدة من الميزات الرئيسية لهذه المنصة هي أنها متوافقة مع جميع شرائح المجتمع. من ناحية أخرى، تحتوي منصتنا على أحدث التقنيات، مثل نظام التوصية، الذي يساعد العملاء في إيجاد الحرفيين الذين يهمهم، من خلال تفاعلاتهم مع المنصة، مثل إجراء البحث عن الحرفيين داخل المنصة، فضلاً عن التقييمات المختلفة التي يقدمها العملاء للحرفيين. تحتوي المنصة أيضًا على متحدث الي للدردشة يتيح للعملاء والحرفيين الإجابة على استفساراتهم المتعلقة بالمنصة. أما بالنسبة لتأمين بيانات العملاء والحرفيين والمعاملات المختلفة بينهما، فقد استخدمنا تكنولوجيا البلوكتشين. فيما يتعلق بتطوير واختبار المنصة وخدماتها المختلفة، قمنا بإنشاء سحابة خاصة تابعة لنا.

Contents

List of Figures

List of Tables

General Introduction

Today, web technologies continue to have an ever-increasing influence on our lives, becoming an integral part of them. The world today is characterized by its shift towards digital technology. Where we are witnessing rapid growth in the use of modern technology and Web technologies in all aspects of daily life. This major shift aims to improve services provided by companies to customers.

Digital platforms provide opportunities for communication between people as well as in various services and can also be used in electronic commerce and social networks.

For the world today, which is witnessing a fast development in the field of digital technology, various items and services of high quality can be found in the least possible time and with minimal effort, thanks to digital platforms, as for traditional methods, they are less effective and cost a lot of effort and time .

Nowadays, finding skilled and high-quality craftsmen (such as plumbers, masons, etc.) is very difficult using traditional means such as word-of-mouth referrals or online directories.

Based on this context, our manuscript aims at the possibility of creating a new digital platform that facilitates the link between customers and craftsmen, as the platform will provide an easy-to-use user interface and is in line with all segments of society . The platform also provides the latest technologies such as chatbot, which represents customer support in the platform. The platform also provides a recommendation system that recommends items that may be of interest to users. As for the data protection and security system, we have used blockchain technology in order to provide a high level of security and trust .

In order to deal with the growth of the platform and the joining of an increasing number of craftsmen and customers, we have used cloud-based solutions, and thus we can manage resources efficiently .

And by combining these technologies, we aim to create a comprehensive and flexible platform that enables craftsmen to highlight their work and enhance their presence on the Internet .
As for customers, it makes it easier for them to search for craftsmen of high quality and experience .

The manuscript is structured as follows :

Chapter One : In this chapter, we make a comprehensive study of the various types of digital platforms, as well as studying each technology separately and the work related to this technology, while highlighting the advantages and disadvantages of each technology .

Chapter Two : In this chapter, we will lay out the approach that we follow to build the platform, in addition to explaining the structure and design of the system, as well as highlighting the various algorithms used to build the various technologies of our platform .

Chapter Three : In this chapter, we explain the tools that we relied on in creating the platform, as well as we describe the main functions of the platform that we developed, in addition to evaluating the algorithms of the system, as well as highlighting the different interfaces of the platform.

Chapter 1

WorkersSpace platform : state of the art

1.1 Introduction

In the context of creating a platform to connect craftsmen and their clients, this first chapter plays an important role in creating a clear understanding of the digital platform and what its goals and characteristics are. The chapter's exploration of different types of platforms and traditional platforms provides a solid foundation for understanding the different ways platforms can be structured and how they work. This knowledge can inform platform design and development, as the dissertation can consider what type of platform will work best to connect craftsmen to their customers.

1.2 Definition of digital platforms

They are online and web-based virtual spaces based on unique business models for exchanging information, knowledge, goods, ideas, or services, etc. with financial or non-financial return, e.g., Google Search, Facebook, Spotify, Youtube, Apple Store, Amazon Marketplace, Uber, Apple/Google Pay, etc [9].

1.3 Types of platforms

As the digital platform approach continues to thrive, our engagement with these platforms has become increasingly frequent. These platforms vary widely in their structure and function, with their specific form and purpose being influenced by the business model employed. Examples of successful digital platforms are:

Type of platforms	**Definition of Platforms**	**Examples for platforms**
service marketplace	A service marketplace or platform for online services is a website that provides a space for individuals, professionals, or businesses to showcase their offered services [10].	Fiverr, Upwork, and Thumbtack
Product marketplace	A marketplace for products is a platform where suppliers or vendors sell physical, virtual, and downloadable goods to customers [11].	Esty, Ebay and Amazon
Payment platform	A payment platform is an online infrastructure that facilitates electronic money transfers and serves as a digital substitute for traditional paper-based methods, like checks and money orders [12].	Pypal, Alipay and Visa
Investment platform	A An Investment Platform is an account that is professionally managed and intended to combine multiple investments into a single account [13].	Prosper, CircleUp, LendingClub and AngelList
Social media platform	One can define a social media platform as a website or application that allows users to create, share and discover content, as well as connect with other users [14].	Facebook, Twitter, TikTok, Instagram
Communication platform	The term communication platform encompasses various methods of communication such as email, internet, telephone, and social media, among others [15].	Messenger, Zoom, Skype
Development platform	A development platform allows customers to create a personalized application that can run on platform services [16].	Windows, Linux, Mac, Ios, And Android

Table 1.1: Comparison of types of digital platforms

1.4 Classification of traditional platforms

The categorization of digital platforms based on numerous characteristics, such as their business strategy, interaction style, governance style, and ownership structure, offers insightful information about how they operate.

The integrator platform paradigm, exemplified by Apple's iOS and Google's Android, places an emphasis on creating a unified environment where multiple programs and services may seamlessly integrate. These platforms provide a tailored experience and seamlessly combine hardware and software, giving consumers and developers access to a controlled environment.

As demonstrated by Linux and cloud computing initiatives, the product platform strategy places a heavy focus on the development of underlying infrastructure or technology. Linux is a free and open-source operating system that offers a flexible and adaptable framework for many applications. Organizations may access scalable and flexible computing resources using cloud computing services like Amazon Web Services (AWS) or Microsoft Azure.

Platforms like Facebook, eBay, and Alibaba leverage the multisided platform concept to connect several user groups or markets. These platforms operate as middlemen, allowing interactions and transactions between diverse parties to facilitate communication, commerce, and social relationships.

Through collaborative platforms like Threadless.com and Wikipedia, collaboration and user-generated content are promoted. By depending on the contributions and work of their user communities to create and curate content, they foster a collaborative and participatory environment. Console video games and competitive websites like TopCoder thrive on competition. These platforms usually offer contests, challenges, or gaming environments where users compete with one another and showcase their skills and expertise.

Governance structures differ between open and closed platforms. Decentralized decision-making is supported by open systems that rely on user input and communal intelligence, like Wikipedia and Linux. Since they have more centralized administration of their ecosystems, closed platforms like Apple's iOS and Google's Android have tighter control over the production and distribution of apps.

Different ownership structures are also present. Certain businesses control and operate the platforms that rely on IP, such Sony's game consoles or Microsoft's platforms. Anyone may use, modify, and share the source code of community-driven, open-source systems like Linux and R.

If we have a deep grasp of these various facets of these platforms, we can appreciate their unique traits, business models, and governance systems. Additionally, it helps stakeholders decide whether to use, create, or participate in the complex world of digital platforms[6].

Categories of platforms	**Types of platforms**	**Examples for platforms**
Business model	Integrator platform model	Apple iOS; InnoCentive.com; Google Android
	Product platform model	Linux; Cloud computing initiatives
	Multisided platform model	Facebook; eBay; Alibaba
Interaction model	Collaborative platforms	Threadless.com; Wikipedia
	Competitive platforms	TopCoder; Video games on consoles
Governance mode	Opened platforms	Linux; Wikipedia
	Closed platforms	Apple iOS; Google Android
Ownership Structure	Property-based platform	Sony (game consoles); Microsoft platforms
	Open source-based platform	Linux; R

Table 1.2: Comparison of categories of digital platforms [6]

1.5 Platform objectives

The internet has revolutionized the way we do business, offering new opportunities for individuals and businesses to connect with each other. For traditional craftsmen, an online platform can provide a centralized location to showcase their work and services, offering benefits such as secure transactions, reputation building, and promotion of traditional crafts.

- Providing a centralized location for craftsmen to showcase their work and services: An online platform can bring together various craftsmen from different regions, specializations, and backgrounds. By creating a centralized location for their work and services, it becomes easier for clients to find and learn about them. This can include details such as their portfolios, reviews, ratings, contact information, and pricing. A well-designed platform can also help craftsmen differentiate themselves and their offerings from others, which can be particularly helpful for those who may not have a physical storefront or online presence.

- Making it easier for clients to find and hire the right craftsmen: For clients, an online platform can be a one-stop shop for finding and hiring the right craftsmen for their needs. By providing advanced search and filtering options, clients can easily narrow down their search by location, type of service, pricing, and other factors. They can also compare and contrast different craftsmen based on their reviews and ratings, making it easier to make an informed decision. This can save clients time and effort and provide a better overall experience.

- Creating a platform for craftsmen to build their reputation and establish themselves as experts: By showcasing their work and services on a reputable online platform, craftsmen can build their reputation and establish themselves as experts in their field. This can help them stand out from competitors and attract more business opportunities. Reviews and ratings from previous clients can also serve as social proof, which can help to further solidify their expertise and skill level. A well-designed platform can also provide resources and tools to help craftsmen market themselves and their services more effectively.

- Offering a secure and reliable platform for transactions between clients and craftsmen: One of the most important aspects of an online platform for craftsmen is ensuring that the platform is secure and reliable for transactions between clients and craftsmen. This can include providing secure payment options, clear terms and conditions, and dispute resolution mechanisms. A trustworthy platform can help build trust between clients and craftsmen and provide a better overall experience for all involved.

- Promoting traditional crafts and supporting the preservation of cultural heritage: Finally, an online platform for craftsmen can help promote traditional crafts and support the preservation of cultural heritage. Providing a platform for craftsmen to showcase their traditional skills and techniques, can help to ensure that these crafts continue to thrive and evolve in the modern world. This can be particularly important in areas where traditional crafts may risk disappearing due to a lack of demand or interest.

Overall, an online platform for craftsmen can provide many benefits for both craftsmen and clients. By bringing together a diverse range of skills and services in one place, it can make it easier for clients to find the right craftsmen for their needs and help craftsmen to establish themselves as experts in their field. It can also help to promote traditional crafts and support the preservation of cultural heritage.

1.6 Characteristics of good platform

A good platform is one that is able to effectively facilitate interactions between users , while simultaneously delivering benefits to all users. The following are some essential features of a good platform:

- **Scalability :** The capacity to scale up is a vital attribute of a quality platform, as it should be capable of accommodating a burgeoning user base without compromising its operational efficiency .

- **Flexibility :** Adequate flexibility is an essential characteristic of a superior platform, as it should have the ability to adapt to diverse users and their unique requirements .

- **User-centric :** An ideal platform ought to concentrate on delivering a user-centric experience , where the user's needs and preferences are given priority. This involves ensuring that the platform is user-friendly and intuitive, making it effortless for users to engage with and navigate through the platform .

- **Openness :** Transparency and openness in governance are crucial elements of a good platform. It should provide users with a platform that is governed in an open and transparent manner, where users can have a voice and contribute to the decision-making process. This promotes a sense of community and trust among users, and allows for a more democratic approach to platform management .

- **Security :** Ensuring the security of user data and transactions is of utmost importance for a good platform. It should prioritize safeguarding user information and transactions by implementing robust security measures. This helps to build user trust and confidence in the platform, and protects users from potential risks and vulnerabilities associated with online transactions .

- **Reliability :** is a key aspect of a good platform. It should be dependable and operate with minimal disruptions and downtime. This helps to ensure that users can access the platform when they need it, without any issues or delays. A reliable platform instills confidence in users, and enhances their overall experience and satisfaction with the platform.

- **Interoperability :** A good platform must possess the ability to integrate with other platforms and services seamlessly, thereby promoting interoperability. This integration allows users to experience a more comprehensive and streamlined process, where they can access different services and platforms through a single interface. The benefit of interoperability is that it enables users to capitalize on the strengths of multiple platforms and services, resulting in a more potent and efficient solution .

- **Innovation :** is a crucial element of a good platform. It should continuously evolve and introduce new features and capabilities to remain ahead of the competition and ful fill the evolving needs of users. By embracing innovation, the platform can deliver unique and innovative solutions that set it apart from competitors, and provide users with a fresh and exciting experience. This ensures that users stay engaged and satisfied with the platform and encourages them to remain loyal over time .

- **Accessibility :** is an essential aspect of a good platform. It should be designed to cater to a diverse range of users, including those with disabilities or limited access to technology. This involves ensuring that the platform is easy to use and navigate, with features and tools that enable all users to access and engage with the platform. By prioritizing accessibility, the platform can provide an inclusive experience for all users, irrespective of their background or technological abilities .

- **Responsiveness :** is a crucial aspect of a good platform. It should actively seek and respond to user feedback, continuously improving and evolving to meet changing user needs. By incorporating user feedback, the platform can refine its features and functionalities, providing a more tailored and satisfying user experience. Additionally, by demonstrating responsiveness, the platform can foster a strong relationship with its user base, instilling a sense of loyalty and trust in the platform .[17] .

1.7 Related works

In this section we present some similar works related to our web application with all the part of our work, so each part of our platform we present some related works to this task to make a comparisons and learn from various works. So, each part of our work discusses the problem definition, the solution, and some similar works.

1.7.1 Digital Platform

First of all, this task related to build a website which is the main part. In the next subsections we discuss the problem and the solution and some related works comparative study.

1.7.1.1 Problem Definition

- Finding reliable and skilled professionals for home repair and maintenance tasks can be challenging in today's fast-paced world.

- Traditional methods of finding and hiring craftsmen can be time-consuming and unreliable.

- Many people feel frustrated and unsure of whom to trust when searching for craftsmen.

- Urgent home repairs and maintenance tasks can lead to further damage and expense if delays occur in finding a suitable craftsman.

- The difficulty of finding craftsmen in big cities.

1.7.1.2 Synthesis comparative study

Digital platforms have gained significant attention over the past few years due to their ability to connect customers with service providers efficiently. Two of the most successful digital platforms in the home services sector are TaskRabbit and Thumbtack. Both platforms allow customers to find reliable and skilled professionals for home repair and maintenance tasks with ease. They offer different features and business models, which have resulted in various outcomes. To gain a better understanding of these platforms and their differences, a comparative study is necessary, and provide an analysis of their similarities and differences.

Name of Platform	Taskrabbit	Thumbtack
URL	https://www.taskrabbit.com	https://www.thumbtack.com
Headquarters	San Francisco	San Francisco
Definition	Focus more on one-off tasks and odd jobs. For example, installing a TV, putting together furniture, planting flowers, or cleaning a specific room in the house.	Thumbtack's primary focus is on services related to homes, such as installing floors, painting interiors and exteriors, and cleaning houses.
Services	+20	+2000
Pricing	25.00 dollar registration fee	Only pay for leads that you pursue and other pro features like targeted advertising
Review System	Users leave reviews and ratings on job completion	Users leave reviews and ratings on job completion
Customer Services	Very responsive. Users can connect with pros only through the app or website	Very responsive. Users can connect with pros only through the app or website.
Customer Services	No work experience required	Work experience required

Table 1.3: Comparison between Taskrabbit and Thumbtack

1.7.2 Chatbot for customer support

Sometimes customers have inquiries or require information regarding the services offered to them, because of that we made a chatbot to handle the queries of the client on the platform that we proposed.

1.7.2.1 Problem Defintion

Traditional customer service methods such as phone and email support can often be time-consuming, but traditional customer service channels such as phone or email can often lead to long wait times and delayed responses. This can lead to frustration and dissatisfaction among customers, which can ultimately impact a company's reputation and bottom line. Another challenge is the need to provide consistent and accurate information across multiple channels, such as phone, email, social media, and chat, which can be difficult to manage and can lead to confusion for both customers and support teams. One of the biggest problems in customer service is the challenge of providing fast and efficient customer service. Customers may have questions or problems that need to be resolved quickly. The primary problem for chatbots in customer service revolves around their ability to understand and interpret natural human language accurately. This issue, commonly known as the Natural Language Processing (NLP) problem, arises due to the inherent complexity, ambiguity, and variability of human communication.

When creating a website for craftsmen or any other website, there is always a request and concerns of interest to customers at any time, meaning that any inquiry from any customer may be at any time, and this must be taken care of by those in charge to develop and ensure customer satisfaction and service at any time , and at the same time this is a challenge on how to make a chatbot in terms of accuracy and customer service. Chatbots provide a range of benefits to customers by improving the customer experience and delivering faster, more personalized service. and it make easy ask and get the responses.

1.7.2.2 Natural Language Processing:

NLP is a sub-field of Artificial Intelligence (AI), it make the machine process and understand the human language, NLP provide methods and techniques to dealing with human language by process and understand this language[18].
There are many problems NLP is used to solve such as language translation, question answering problem and more.

1.7.2.3 Benefits of customer support chatbot

Chatbot very important and useful in case of customer services, chatbot offer these services :

- **Faster response times:** Chatbots can handle multiple queries at the same time, resulting in faster response times compared to human agents.
- **Cost-effective:** Without the need for additional staff, resulting in cost savings for businesses.
- **Availability all time:** Customers have access all times. This can improve customer satisfaction. Chatbots have the capability to gather customer information and utilize it to customize interactions, resulting in a more individualized experience for every customer.

1.7.2.4 History

One of the first chatbots was ELIZA, Eliza developed in the 1960s by Joseph Weizenbaum at MIT [19] ELIZA chatbot used a simple pattern-matching algorithm to simulate conversation and using scripted responses. and after the 1990s, chatbots were primarily used in customer service and support roles, providing automated responses to frequently asked questions. After that chat robots are becoming more and more advanced and more complicated. chatbots becoming more common in our life. Virtual assistants such as Apple Siri, Alexa are becoming more popular. In recent years, chatbots have become sophisticated with the advent of artificial intelligence, natural language processing and machine learning techniques. And with the advent of transformers, robots gained more sophisticated capabilities and greater computational power. And in recent months, chatgpt By 'OpenAI' [20] appeared and it revolutionized the field of AI, this is because chatgpt can perform difficult and complex tasks such as written code, solve problems, written paragraphs about anything, and more.

1.7.2.5 Types of chatbots

There are many approaches to build a chatbot for customer support like:

- **Rule-Based:** Based on predefined keywords that bots understand, this type of bots the system operates based on specific rules, this chatbot system cannot effectively answer questions if the input pattern does not match the predefined rules [21], the bot's response is static. While it's simple, it solves most repetitive tasks such as: Canceling an order or requesting a refund[22].

- **Retrieval based**: These models are designed to retrieve relevant information from a given dataset of textual information, based on the user's input. These models typically use deep learning algorithms[23].
- **Generative based**: This approach under the category of the models have the ability to generate new sentences as a response to user input [24].

1.7.2.6 Synthesis comparative study

In [25], the authors have proposed an online chatting system (chatbot) using a knowledgeable database for college inquiries using pattern matching, and this chatbot did not use advanced AI methods like machine learning and deep learning it is just a rule based chatbot. The application allows students to stay up-to-date with the latest cultural events at the faculty. this application is for College student, staff and parents. the first objective of this chatbot is to minimize the time required to solve the quries. and the second is To give response to the user based on queries. and finally to simplify communication between user and machine.

Another study in this paper [26], they develop a chatbot called FedBot it is a retrieval based chatbot using transformers, the main idea in this article is applying federated learning to share models parametres without sharing data. federated learning is using to protect user privacy. so FedBot is able to train a machine learning model to understand user conversations without having to access the data itself. by using federated learning all the clients participating data is protected so FL is about reducing privacy risks.

The authors in [27] made a conversational system for customer support , they worked with sequence to sequence with encoder and decoder as a LSTM neural networks , the LSTM is trained on a twitter conversation dataset, after feed the input request into the model the response is generated.

A pipeline optimization approach based on a deep learning-based open-source chatbot system called ArRASA proposed by [28] using the open source RASA framework to interact with users. the work is devided into four global parts which is tokenization , feature extraction, intent classification , entity extraction.

Type of chatbot	**Rule-based**	**Retrieval-based**	**Generative-based**	**Hybrid**
Methods apply	Knowledgeable database and pattern matching	Transformers and federated learning	Sequence to sequence model	Framwork based on transformers
Objectives	Online chatting system for college inquiries	Customer Support	Conversational system for customer service on social media	Arabic customer support
References	[25]	[26]	[27]	[28]

Table 1.4: Related work comparison studies for Chatbots

So, our work is to build a retrieval-based chatbot to handle customer support part of the website. We have used transformers and extractive question answering task to build this chatbot, so it consider as a retrieval-based chatbot because it extract the answer from a given document.

1.7.3 Blockchain for data protection

This task about build a network based blockchain to ensure security data of our customers.

1.7.3.1 Problem definition

The data collected through our platform always faces the problem of protecting it, because the biggest concern of electronic platforms is data protection, because in our platform there is a need to protect the job applicant's portfolio and the entries and data of both participants as well all other important concerns safely at all levels. Due to the constant demand for the data available on our electronic platform, ensuring security is of paramount importance and this is ensured by the blockchain application. Because of the decentralized system, data cannot be hidden, altered or destroyed, so improving security is paramount. Weak security will not work as registrants can be vulnerable to information and identity theft hackers. Sensitive information such as ID number and personal details are at risk, but thanks to blockchain technology, data becomes more secure

1.7.3.2 Synthesis comparative study

Name of Platform	**Everledger**	**IBM Food Trust**
Blockchain Type	public blockchain[29].	uses a permissioned blockchain [30].
Consensus algorithme	Proof-of work (Pow)[29].	Practical Byzantine Fault Tolerance (PBFT) [31].
Security Features	Data integrity :no manipulation or deletion of records[29] .	creating a tamper-proof record of transactions in the food supply chain [30].
Industry Adoption	tracking and authentication of high-value assets, particularly diamonds and other precious gemstones [29].	Track the path of food from producer to consumer[30].
Scalability	Difficult to expand[29] .	Easy to expand.[30].

Table 1.5: Comparison of two applications using blockchain

Blockchain Type: Both ibm food and everledger use public blockchains as all nodes have the same influence in the network, which has an impact on scalability [29][30].

Consent algorithme: Everledger uses a proof of work approval mechanism whereby all nodes are equal in proof of work, which is a relatively new and untested algorithm. . IBM Food Trust uses a consensus mechanism called Practical Byzantine Fault Tolerance (PBFT), which is a proven and secure algorithm to ensure the integrity of the blockchain [29][31] .

Security Features: Both Everledger and IBM Food Trust prioritize strong security measures. While Everledger emphasizes data safety, preventing tampering and deletion of records, IBM Food Trust focuses on creating tamper-proof transactional records in the food supply chain, ensuring the safety and authenticity of food products[29][30].

Industry Adoption: IBM Food Trust focuses on tracking the path of food from the producer to the consumer, ensuring transparency and traceability in the food supply chain. On the other hand, Everledger specializes in tracking and authenticating high-value assets, specifically diamonds and other precious gemstones[29][30].

Scalability: Everledger and IBM Food Trust are both designed for scalability but have different approaches. everledger uses a public blockchain, which can be more difficult to scale, while IBM Food Trust uses an approved blockchain, which is easier to scale[29][30].

1.7.4 Recommendation system

A Recommender System (RS) is a type of software that helps users find items they might be interested in from a large pool of choices. These items could be anything from movies and restaurants to books and craftsmen. RS uses various methods to determine a user's interest in an item. One way is through ratings, which are scores that users assign to items. Another approach is to look at the similarity between an item and the ones that the user has previously shown interest in, particularly in the case of textual domains. To provide personalized recommendations, RS uses complex algorithms and techniques to analyze user behavior, resulting in a better experience for the user.

Recommender Systems use user profiles to store their preferences about items that align with their interests. These profiles typically comprise a set of ratings and/or a set of keywords representing users' interests. Ratings range from 1 to 5, with higher numbers indicating higher levels of interest. Meanwhile, keywords are automatically extracted from the texts that users have consumed in the past. To recommend items of interest to users, ratings are first aggregated through a series of computations to measure users' similarity. Texts are then matched against the user profile, and the most similar texts are recommended. Additionally, the keywords in the user profile can have weights assigned to them, indicating how much the user values each keyword.

Recommendation techniques have become essential for big companies like Amazon, eBay, and Netflix as they assist in estimating the potential preferences of their customers and recommending relevant products or items to them. The accuracy of these recommendations has a significant impact on the commercial success of these companies, ultimately driving revenue and customer satisfaction.

Recommendation systems can be divided into three main types: Content-Based (CB), Collaborative Filtering (CF), and Hybrid Recommendation. (CB) systems recommend items that are similar to items that the user has liked in the past. Recommendation systems based on (CF) recommend items based on the preferences of similar users.

(CF) suffers from a "cold start" problem, which is the difficulty in making recommendations to new users or items with little or no data , and This problem is common in recommendation systems, where the algorithm must learn the user's preferences from scratch .

The hybrid approach in recommendation systems combines both (CB) and (CF) techniques to provide more personalized recommendations. The choice of approach depends on several factors, such as the type and size of the dataset, as well as the preferences of the user. By integrating both (CB) and (CF) techniques, hybrid systems can leverage the strengths of each approach to overcome their limitations and provide better recommendations to users.

For our project, we used the collaborative filtering technique and the content-based filtering technique, which is based in our work on the TF-IDF technique , TF-IDF is a valuable tool used in various areas such as content-based filtering, text mining techniques, and other contexts related to information retrieval. The acronym stands for "Term Frequency - Inverse Document Frequency". It is a statistical measure used to determine the relevance of a word to a document within a collection of documents. Typically, a weight is calculated for each word that indicates the significance of the word in the document and the corpus. This approach is a widely used technique in text mining and information retrieval [32] , [33].

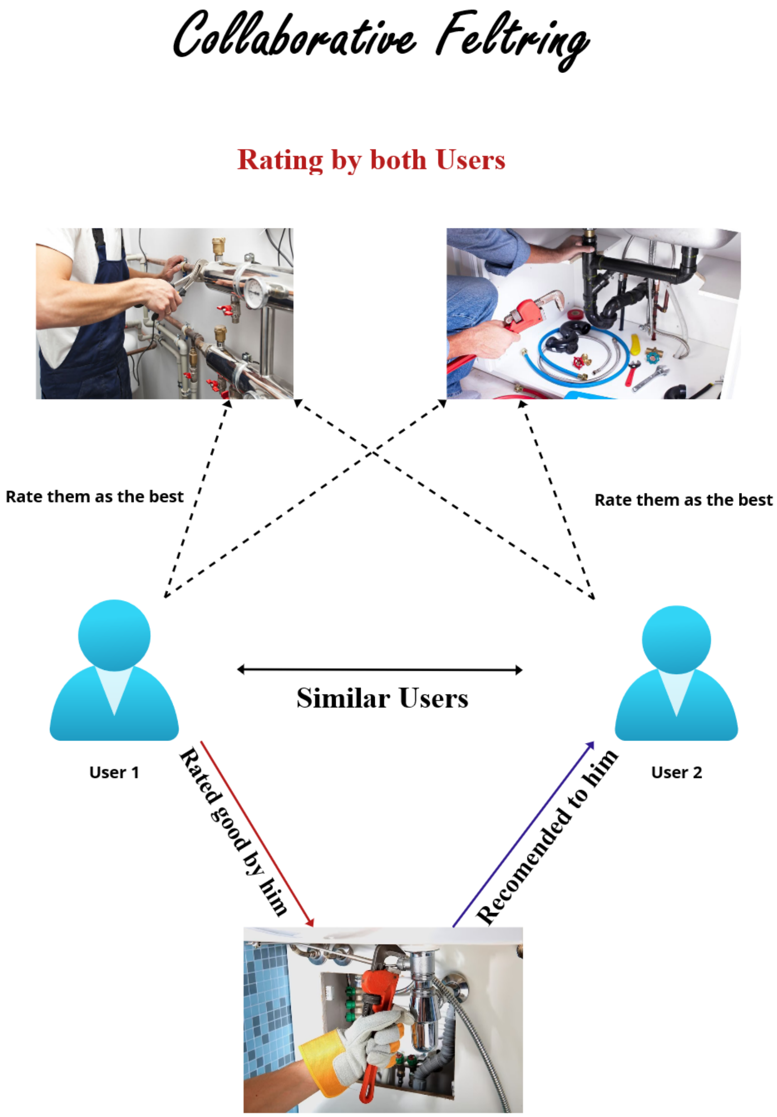


Figure 1.1: Collaborative Filtering .

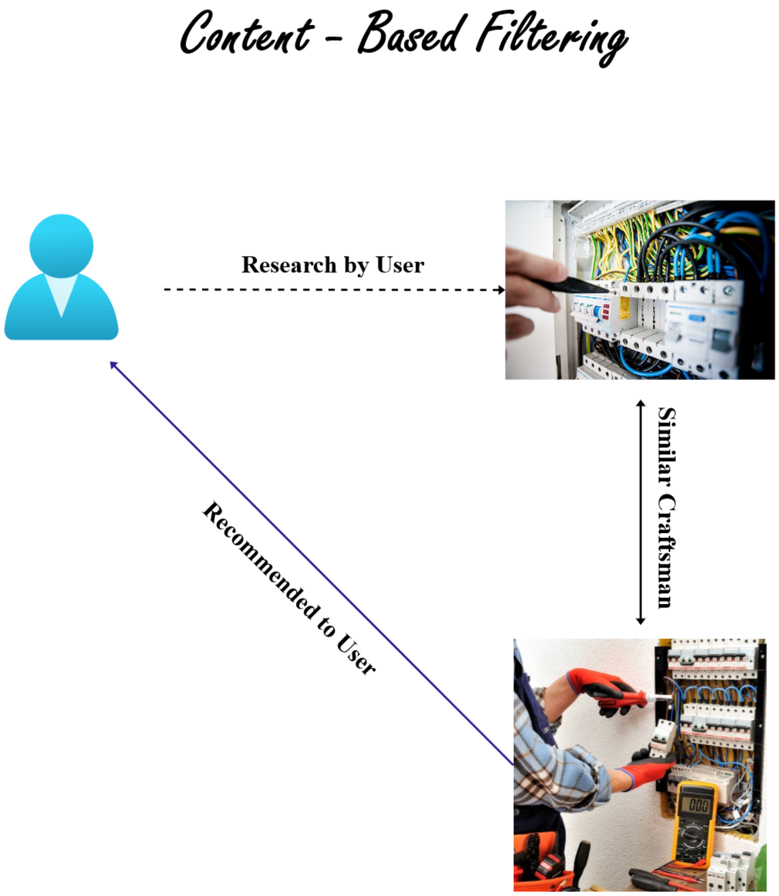


Figure 1.2: Content-based Filtering .

1.7.4.1 Problem definition

The advent of recommendation systems has revolutionized the way users interact with websites and online platforms. Before these systems, websites faced several challenges that negatively impacted user engagement and satisfaction, as well as business revenue. In this context, we will discuss some of the problems that websites encountered before the introduction of recommendation systems :

- Low user engagement.
- Poor user experience.

- Low conversion rates.
- Manual searching through large catalogs of products or content leading to frustration and potential abandonment.
- Lack of personalized recommendations resulting in users leaving the website without making a purchase or engaging with the content.
- Lost revenue for businesses and lower customer satisfaction.
- Limited means of gathering and analyzing user data to improve offerings and increase user engagement.

1.7.4.2 Synthesis comparative study

In recent years, digital platforms have become increasingly popular for their ability to connect customers with relevant products and services. Among these platforms, two of the most successful recommendation systems are Netflix and Amazon. Both platforms use recommendation algorithms to suggest content and products that are tailored to each individual user. Despite their similarities, these platforms have different features and business models, resulting in different outcomes. A comparative study of these platforms can provide insights into the effectiveness of their recommendation systems, and the strategies that they use to drive engagement and customer satisfaction. In this table , we will analyze the similarities and differences between Netflix and Amazon's recommendation systems.

Name of Platform	**Netflix**	**Amazon**
Definition of platforms	Netflix is a streaming service that operates on a subscription model, enabling its members to watch a variety of TV shows and movies on any internet-connected device.	It is an e-commerce and cloud computing website founded by Jeff Bezos and is headquartered in Seattle, Washington. It is the world's largest online retailer by total revenue and market capitalization.
The method used	- Collaborative Filtering - Content based filtering: Personalized Video Ranking (PVR) /Top-N Ranking /Ranking of interesting content for further viewing /Popular movies ranking [34]	- Content-based filtering - Item-to-item collaborative filtering - Bandit-based algorithm - Casual interference algorithm
Application domain	These types of algorithms can be applied in different platforms (such as YouTube, TikTok...etc).	These types of algorithms can be applied to different platforms such as electronic stores (such as AliExpress, Alibaba...) and these algorithms can also be applied to service platforms (such as Taskrabbit, Thumbtack...etc).
Problem treated	The lack of success of the Netflix recommendation system algorithms in some countries, such as India, despite the reduction in the subscription price. [35]	Amazon service is not available in some countries. producer to consumer
The limits	Limited ability to recommend niche or non-mainstream content	Users may find the recommendations too limited.

Table 1.6: Related work comparison

1.7.5 Private cloud

This task about building a private cloud to deploy the entire website on it. And make virtual machines to connect on the same network.

1.7.5.1 Problem definition

The task is to create a private cloud infrastructure for hosting a website and its database. Additionally, the goal is to deploy virtual machines (VMs) that are pre-configured and ready-to-use for website developers. Each developer should have access to their own VM according to their individual needs. The infrastructure should be designed with a focus on scalability, reliability, and security.The solution should consider the availability of open-source cloud platforms, such as Eucalyptus, OpenStack, and OpenNebula, as well as the specific requirements and constraints of the organization.

Private cloud Definition: The private cloud is a form of cloud computing which benefits from the same advantages offered by public clouds, such as scale and self service, but using proprietary architecture. In contrast to the public cloud, which provides services to multiple organizations, a private cloud, also known as an internal or corporate cloud, focuses on the needs and objectives of a single organization [36].

1.7.5.2 Synthesis comparative study

This comparative study explores the similarities and differences among three popular open-source cloud computing platforms: Eucalyptus, OpenStack, and OpenNebula. The study compares these platforms in various aspects such as their origin, community support, architecture, relation with Amazon, cloud implementation, programming/scripting language, hypervisors, operating system support, databases, image management, and VM migration. The study also notes that OpenStack has the largest active population among the three platforms.

Criteria	**Eucalyptus**	**OpenStack**	**OpenNebula**
Origin and Community Support	Originated from UCSB; Powerful community for development and fixing	Foundation-led project with broad industry support; Large global community	Funded by European infrastructure grants; Supported by large companies
Architecture	Fragmented, distributed architecture with five components	Fragmented, distributed architecture with three core software projects	Classical cluster-like architecture with front end and set of cluster nodes
Relation with Amazon	Embraces AWS API	Built-in management tools with native RESTful APIs; AWS EC2 compatibility API	Embraces AWS API
Cloud Implementation	Private cloud platform	Private and public cloud platform	Hybrid cloud platform with private and public cloud capabilities
Programming / Scripting Language	Primarily written in C and Java; Euca2ools written in Python	Written in Python and Unix Shell; Uses C++, Ruby, and Java	Uses multiple languages including C++, Ruby, and Java
Hypervisors	Compatible with Xen and KVM hypervisors	Compatible with many hypervisors including KVM, Xen, LXC, QEMU, and Hyper-V	Supports Xen, VMWare, and KVM hypervisors
Operating System Support	Supports Linux and images of Windows and Linux	Supports CentOS, Debian, Fedora, RHEL, open-SUSE, SLES, and Ubuntu	Supports CentOS, Debian, Fedora, RHEL, and Ubuntu
Databases	Uses PostgreSQL	Supports any database supported by SQL-Alchemy but commonly uses SQLite3, MySQL, and PostgreSQL	Supports SQLite backend in previous versions and now uses MySQL backend
Image Management	Uses Euca2ools for managing images	Uses OpenStack Image service (glance) for discovering, registering, and retrieving images	Uses image repository or datastores for managing images
VM Migration	Does not support VM migration	Supports VM migration from one resource to another	Supports VM migration from one resource to another
Miscellaneous	Has a powerful community; Second largest active population	Largest active population; Largest global community	Supported by large companies

Table 1.7: Comparison of Eucalyptus, OpenStack, and OpenNebula [7]

1.8 Conclusion

Through this chapter, we've seen some ideas about digital platforms and the difference between them. We have also identified problems and gaps in the market that our platform can fill. In the next chapter we will introduce and analyze the basic structure of the system.

Chapter 2

Design and Contribution

2.1 Introduction

This chapter focuses on the design and contribution of the proposed architecture of our platform. It starts by presenting an introduction to the proposed architecture, explain its components and how it works, and provide UML diagrams to envision the architecture. Additionally, this chapter discusses the algorithms and techniques used in the proposed architecture. Finally, we conclude by summarizing the proposed architecture and its contributions to the platform or system.

2.2 Proposed architecture

In this manuscript we decide to build a platform that act as intermediary between clients and craftsmen. In fact, this platform is a web application that provide services to ensure customer satisfaction, so we decide to develop systems inside this website which is an chatbot and recommendation system and use the blockchain technique to ensure the security data and all of this work hosts in a private cloud, so our architecure divided by five parts which is a digital platform and recommendation system and chatbot and a security data and a private cloud based, we have describe in details each part of this work in the next section which is architecture description.

Our architecture will be mainly focused on three layers as explains in figure 2.1 the general architecture of our proposed platform.

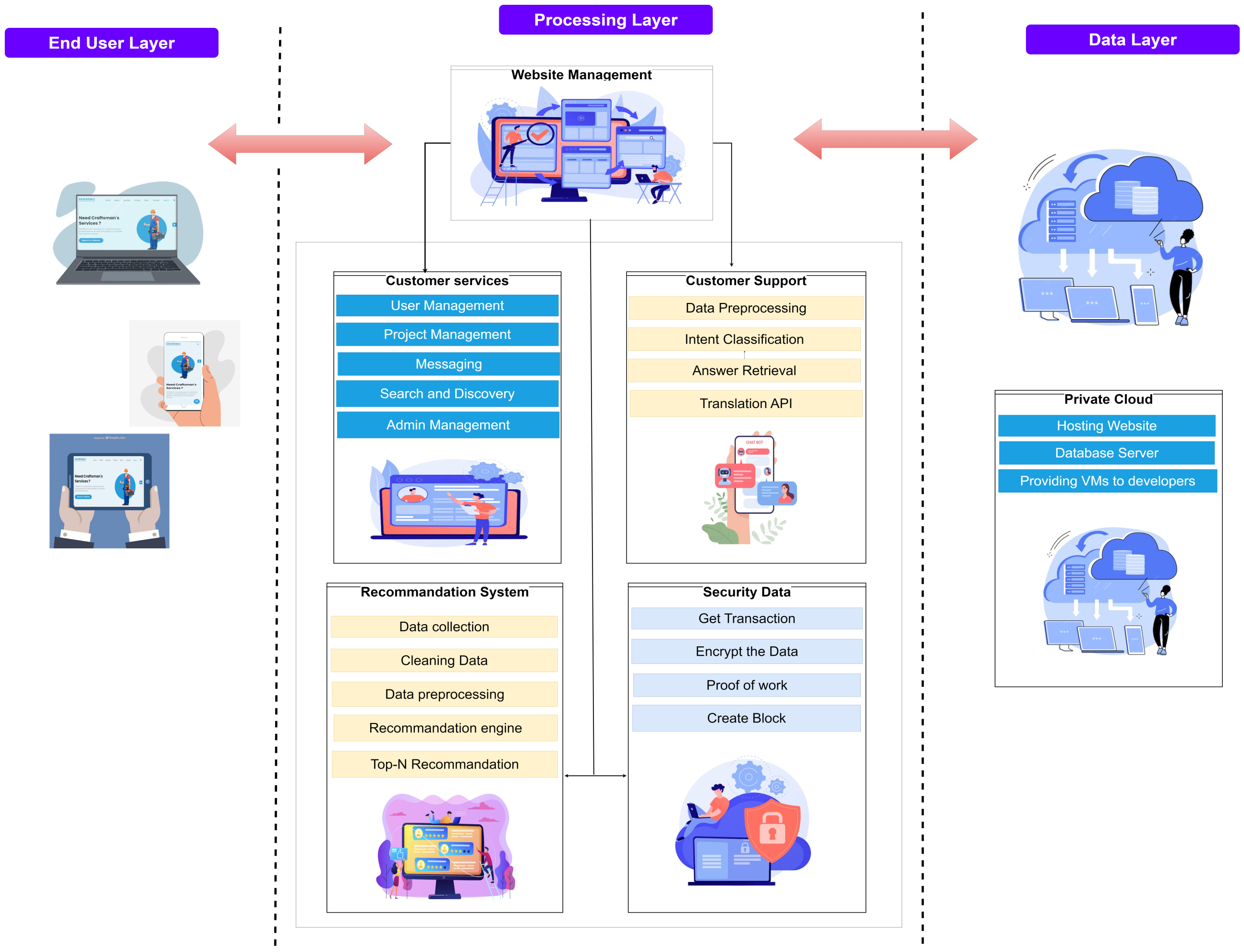


Figure 2.1: General architecture

End-user layer: Is responsible for providing the user interface components for user interaction. This layer contains various user interface elements such as buttons, menus, forms and other graphical elements that allow the user to interact with the platform. It mainly focuses on the user experience and usability of the platform.

Processing layer: Is responsible for providing the application logic for request processing and data processing. This layer contains the algorithms, procedures, and rules that allow the platform to serve user requests and process data accurately and efficiently.

Data layer: Is responsible for storing and managing platform data. In this case, the data plane is hosted in a private cloud that provides a secure and scalable environment for data storage and management. A private cloud is also used to host the website, providing a reliable and accessible platform for user interaction.

Typically, these layers work together to create an online platform that offers a seamless user experience while efficiently processing and managing data. The end user layer provides the user interaction interface, the process layer provides the application logic for handling requests and data processing, and the data layer provides a scalable and secure environment for storage and data management.

2.2.1 Architecture description

In this section we describe each layer that we proposed in the previous section, our proposed architecture consists of :

Private Cloud Infrastructure (Data Layer): where we will be hosting our website and database which will be a blockchain that secure our customer's data.

Processing Layer: This layer contains the platform application logic. It handles user requests and interacts with the underlying database. Our platform layer consists of several components, including:

Website services:

1. User Management: This component is responsible for managing user authentication, registration, and account information.

2. Project Management: The project management component is responsible for managing the project from its inception to completion.

3. Messaging: This component is responsible for managing the communication between users and craftsmen through message threads.

4. Search and Discovery: The search and discovery component is responsible for the search and discovery functions. Includes keyword research services, and category based search. This component also recommends services and service providers to users based on their search history and preferences.

5. Admin Management: Admin management involves overseeing user accounts, content moderation, and platform settings, ensuring smooth operation and security.

Customer support (chatbot):

1. Processing message
2. Understand the intention
3. Retrieve the response
4. Translation API

Security data (blockchain):

1. Get the transaction
2. encrypt the data
3. Proof of work
4. create a block

Recommendation system:

1. Data Collection.
2. Data Cleaning.
3. Data Preprocessing.
4. Recommendation engine.
5. Top-N Recommendation.

End User Layer: The final layer encompasses the user interface (UI) elements that engage with the user. Its purpose is to exhibit information and gather input from the user. This layer comprises a versatile website designed to adapt to various devices such as laptops, smartphones, and tablets.

2.2.2 Platform Process

Our proposed 'WorkersSpace' platform divided into five differents parts, as we can see in this figure 2.2. In this section we explain the process steps to build each one of these systems of the work in details.

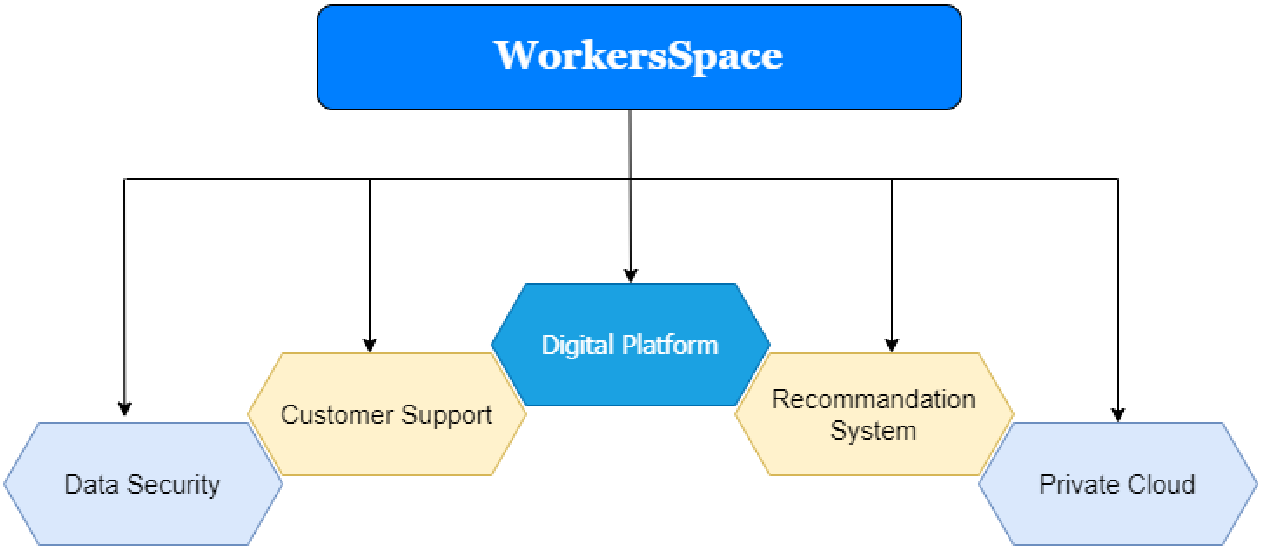


Figure 2.2: Global parts of the platform

2.2.2.1 Customer Services Platform

The platform architecture is designed to provide a comprehensive solution for users looking to connect with skilled craftsmen and complete projects in a reliable and efficient manner. The architecture consists of four major components that work together seamlessly to provide a seamless user experience.

1. **User Management**

   In this platform, user management refers to the processes involved in creating, maintaining, and updating user accounts. This includes the registration process, login mechanisms, authentication, and profile management.

   (a) **Registration**

   Users can create an account on the platform by providing basic information such as username, email address, password, and user type (craftsman or customer needing a craftsman). If the type of user is a craftsman, the system also requests additional information such as location, phone number, craft name, years of experience, diploma, a personal picture and his accounts on social media sites, if any. Once a user submits their registration information, the system sends an email to verify their identity.

(b) **Login and Authenticate**

The platform uses a secure login and authentication mechanism to ensure that only authorized users can access the platform. Users can log in with their username and password.

(c) **Profile Management**

Users can manage their profile information within the platform. This may include updating personal information such as name and contact details, as well as adding or removing services they provide.

2. **Project Management**

Project management is an important component of the platform. It involves managing the life cycle of a project, from its creation to its completion. Here are some more details about the project management component of the build:

(a) **Create a Project**

When a client places a craftsman hiring request, they provide details such as project description, location, and date for the project. This information is stored in the database and the request is sent to the craftsman.

(b) **Project Acceptance**

The craftsman has the option to accept or reject the project. If the craftsman accepts the project, he communicates with the craftsman.

(c) **Project Completion**

Once the project is completed, Craftsman marks the project as completed on the platform. The client is notified, and has the option to review and rate the craftsman.

3. **Messaging**

   Messaging is an important feature of the platform that facilitates communication between clients and craftsmen. It allows them to exchange messages, clarify assignment details, discuss schedules, and negotiate prices.

   The messaging feature of the platform is designed to be simple and easy to use. and Adding the feature of notifying users of new messages.

4. **Search and discovery**

   Search and discovery is an important component of the platform, as it helps users find artisans quickly and efficiently. The platform leverages several technologies to enable effective search and discovery, such as:

   (a) **Search by keyword**

   Users can search for craftsmen by entering keywords such as: craft name, state and municipality name. Making it easier to find and hire someone nearby.

   (b) **Recommendations**

   The platform uses machine learning algorithms to provide personalized recommendations for craftsmen based on users' past search history and behavior.

   (c) **Categories**

   The platform categorizes craftsmen into different categories, such as masonry, painter, electrician, plumber and so on. making it easier for users to find relevant craftsmen.

   The search and discovery component of the platform is critical to the success of the platform as it helps users find suitable craftsmen quickly and efficiently, resulting in increased satisfaction and frequency of use of the platform.

5. **Admin Management**

   Administrative management is an important aspect of a digital platform that involves managing administrative tasks and privileges within the system. Here are the main components of admin management in a digital platform:

(a) **Administrator Dashboard**

The Administrator Dashboard provides a centralized interface for managing and monitoring various aspects of the platform.

(b) **Reporting and Analytics**

The Administrator Dashboard provides reporting and analytics capabilities to track key platform metrics, user activity, and performance indicators. This helps administrators make data-driven decisions and identify areas for improvement.

(c) **Support and Communication**

Administrators are equipped with tools to communicate with users, handle support tickets, and provide assistance when needed. using messaging systems.

(d) **Security and data management**

Administrators play an important role in ensuring the security of the platform and user data. This includes implementing security measures, handling data backups, and addressing any security holes or breaches.

Effective administrative management is essential to maintaining the smooth operation of the Digital Platform, ensuring user compliance, resolving disputes, and providing timely support. Administrators serve as the backbone of the platform, overseeing its performance and maintaining its integrity.

2.2.2.2 Chatbot for customer support

To make an intelligent bot we must have a dataset and choose one of the architecture to make the model train on this data, then the model can be used for prediction, in this part we explain the process of training and using the model.

1. **Training** : To build any deep learning model must feed into this model a data to train on this data and make predictions, but in first we must clean and prepare this data to get ready for training.

   - **SQuAD dataset**

We choose squad the version 1.1 dataset it is the stanford question answering dataset, it consist of more than 100K pairs of question and answers, it came from wikipedia articles, the dataset contain contexts questions answers and the answer is somewhere in the context [2].

- **Data preparation and preprocessing:** We have two different model with two different architecture, we build a dataset with combination of [37] and some necessary intents from another customer support dataset [38] and [39] and we use squad1.1 [40] dataset to fine tune transformer model, the first dataset that we build is a patterns of questions and their tags, the preprocessing task is to remove stop words and use tokenization and lemmetazing and bag of words, first of all is read the data and and remove stop words and apply the tokenization and lemmetazation and convert the result into a bag of words to feed it into the model to train, and for the transformer model we get the data which is squad dataset and process this dataset by apply tokenizer of this pretrained model, we tokenize the context and question, this is by add special tokens [CLS] (classification taken) and [SEP] (separator token) , after that we encoding the tokenized input into numerical representation that BERT can understand, BERT requires two additional input tensors: segment IDs and attention masks, after that the data is going to be ready to fine tune transformer BERT model on it.

We use on this work two kind of deep learning models the first is a feed forward network for intent prediction and the second one is a transformers BERT model for questions answering task. here is the details of the training for each of these models:

(a) **Intent Classification**: Two reasons to choose this model for intent prediction, the first is to increase the accuracy of the question answering model, and the second reason is as we know the input of the pretrained model has a limit of 512 Token, and our database of course is bigger than that.

  - **Define the architecture of the model** : We have created an Multilayer perceptron (MLP) architecture with one input and output layer and two hidden layers.
  - **Train and save the model**: After we have preprocess the data and build the ar-

chitecture we must feed this results into the model to train, after the training is finish we saved the model for use it later for the inference.

(b) **Pretrained Model**: We have used more than one of the pretrained models on hugging face, we decide to fine tune these pratrained models with SQuAD dataset version 1.1 [40]. here is the steps to fine tune the model (all models passe by the same steps):

   i. **Load the pretrained model**: After the prerocessing of the dataset we use one of the base models of the hugging face by loading the model using the 'TFAutoModelForQuestionAnswering' class to get BERT ready to fine tune on question answering.

   ii. **Train the model and save it**: After loading the model and preprocess the dataset we feed the results into the model to train on this dataset. the model saved after each epoch automatically into my hugging face account by using 'PushToHubCallback' class. This for save the model in case the training stopped and we can load it directly from hugging face.

figure 2.3 explain the process of the customer support models training.

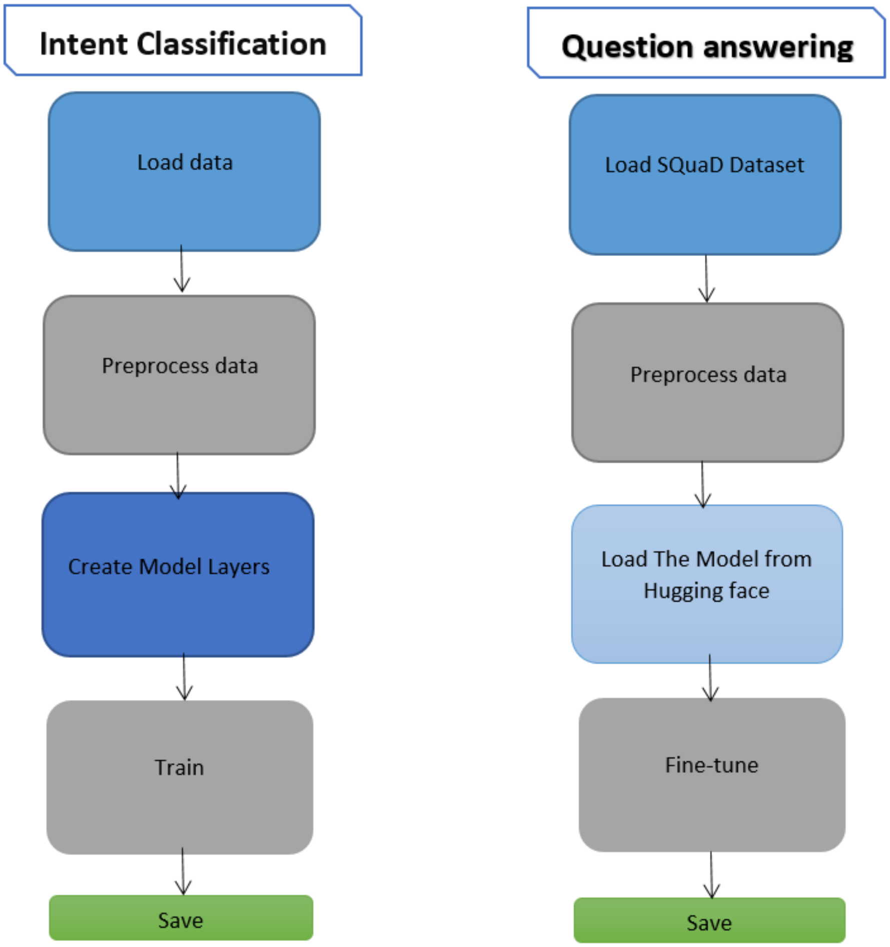


Figure 2.3: Training the models of chatbot

After the training ends, we can use the models to make predictions, the next part is how can we use this models for the inference:

2. **Using the Models**: After the training we save the models, and we can use the models to predict, here is the description of the prediction process.

   (a) **Get message (sentence)**: Getting request (message) from the user interface, In first it is important to detect the language that the user write the message with, to using this language in the response.

   (b) **Translation the message** : Detect the language of the message and translate the message into English.

(c) **Process the input**: Apply natural language processing (NLP) methods to this sentence, we use tokenization and lemmatizing and bag of words, all of this before the prediction.

(d) **Predict the intention**: After the preprocessing of the sentence that the user send, our model must understand the intention, here the intention is from what phrase the model can extract the answer. the intent is the number of phrase where the model can find the answer.

(e) **Question answering model**: This model take two inputs the context and the question of the user, the model is fine-tuned with extractive questions answering task which is must get two inputs, the first is the global context and the second is the question, the model can extract the answer from the context that we provide as a database in a json file.

(f) **Extract the answer**: Based on the two inputs the model can extract the answer from the database which is contain some phrases (contexts), after the model get the question predict the start and the end of the answer from the context.

(g) **Display the response**: After extracting the answer which is the response of the user input, The answer appears in the user interface.

Figure 2.4 describe the inference part by using the model.

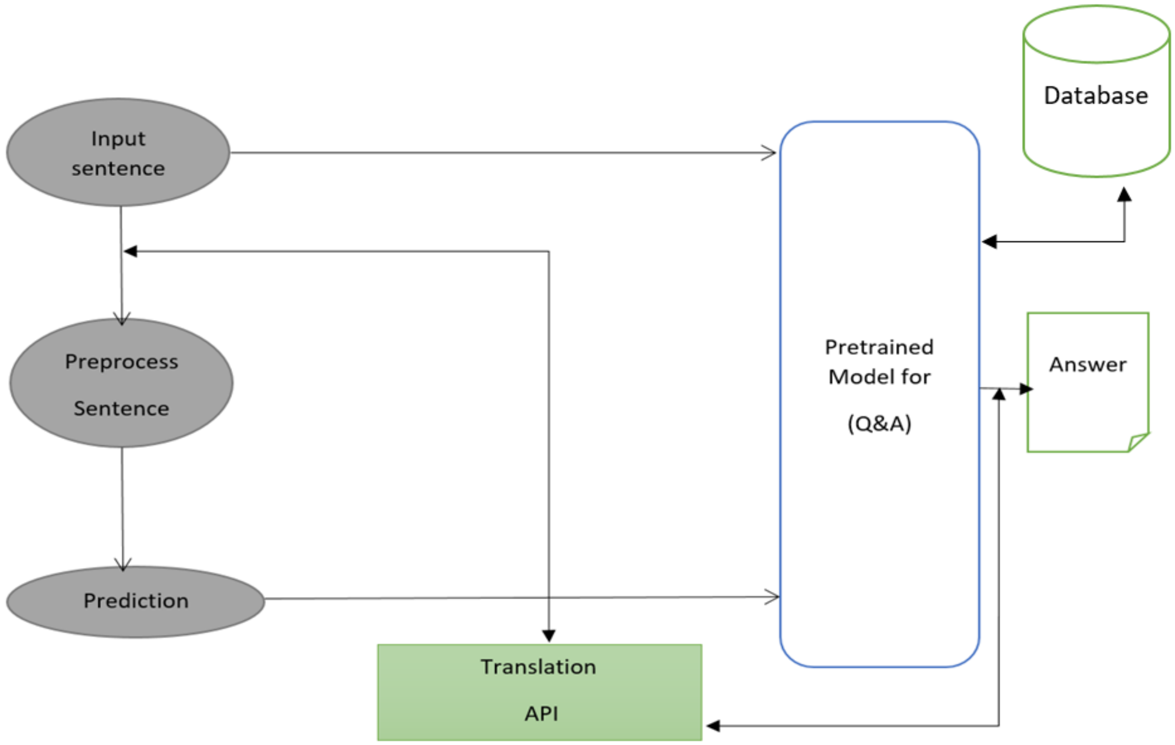


Figure 2.4: Prediction steps of chatbot

3. **Used Algorithms**

Here we describe the algorithms that we apply to build a retrieval-based chatbot using deep learning technique.

- **Trasformers**

  Deep learning is a subfield of machine learning, and both are subsets of artificial intelligence (AI), deep learning focuses on training artificial neural networks on datasets. In 2017 paper known as "attention is all you need"introduced a new type of neural network architecture, this architecture is designed to process the input data such as natural language processing in the tasks like translating, summarizing, text classification and others. The transformer architecture, introduced by the "attention is all you need" paper, consists of multiple layers of self-attention and feed-forward neural networks, the main idea of the transformers is self-attention, which allows the network to take an attention to all the different parts of the sequence, so the transformers can process the entire input sequence[1],[41],[42].

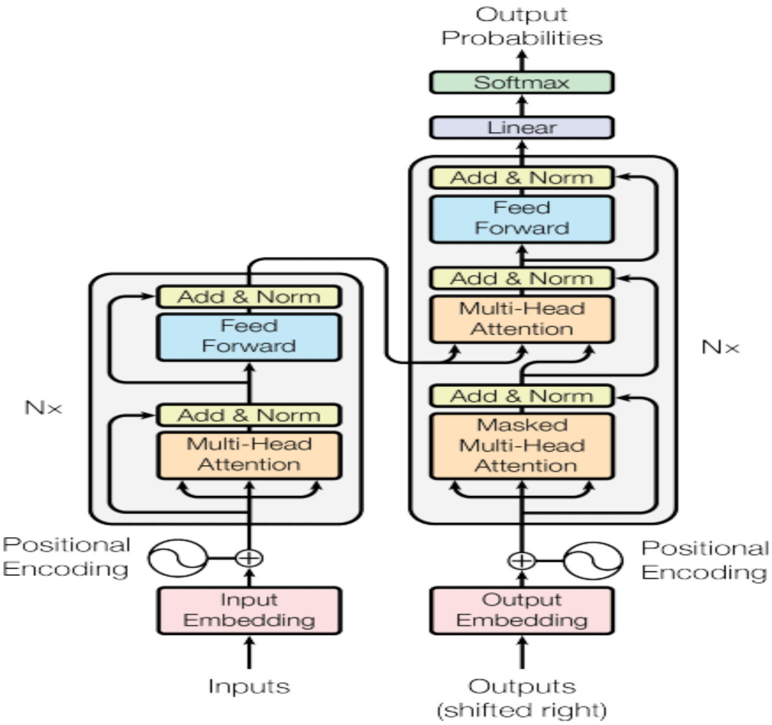


Figure 2.5: Transformers Architecture [1].

- **BERT (Bidirectional Encoder Representations from Transformers)**

  BERT is a pre-trained language model introduced by Google researchers in the paper "BERT: pre-training of deep bidirectional transformers for language understanding", BERT follow the transformers architecture which means BERT use a self-attention mechanism [1], the meaning of bidirectional that BERT using is considering left and right of the context during pre-training. the main idea behind BERT is pre-trained deep neural network with huge unlabeled text data. BERT was introduced with two different architecture a BERT-base and BERT-large with different number on parameters, the base version of BERT with 110 million of parameters and BERT-large with 340 million of parameters. The pretraining phase of BERT it divide into two tasks masked language modeling (MLM) and next sentence prediction (NSP) [2].
  We also can finetune BERT on different nlp tasks.

(a) **Pre-Training**

MLM: During pre-training maskst some words of the input and make the model predict the masked word.

NSP: The second phase of pre-training BERT the model to understand the relationship between two sentences in a given text.

(b) **Fine-tuning:** Is to take pre-trained model and adjusting its parameters on a task-specific, so exploiting the weights of the model and using it for another task such as question answering.

After BERT is released many derived models like 'ROBERTA' (Robustly Optimized BERT Approach) and 'DistilBERT' (Distilled BERT) appears. An example about 'ROBERTA' is a additional derived model of BERT with some modification on the pretraining process with large dataset and longer duration of training, and eliminate the NSP (Next Sentence Prediction) from 'RoBERTa'.

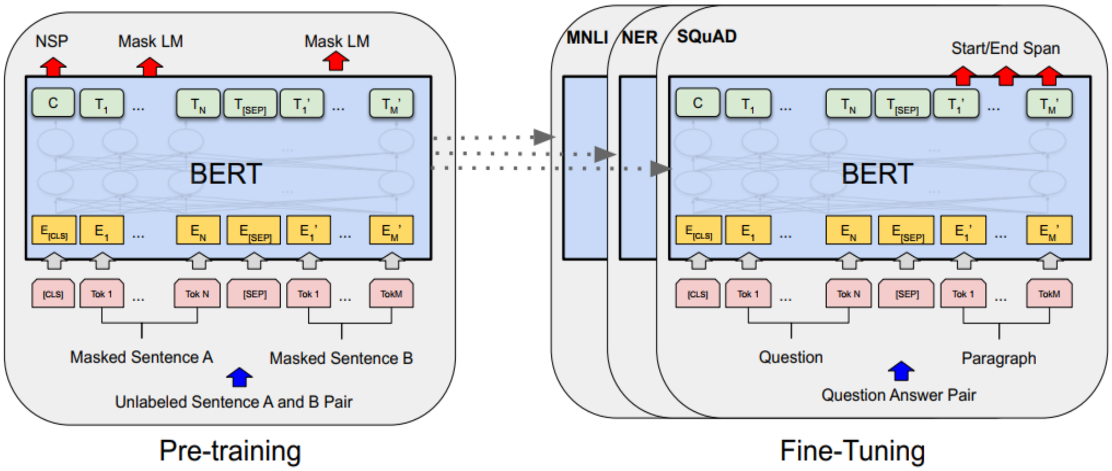


Figure 2.6: Fine tuning and pretraining BERT [2]

- **Transfer learning**

It is a popular technique in the field of AI by using a pre-trained model as a starting point. The main idea of this is that there is no need to train a model from scratch, but rather we can use the last pre-trained model to transfer learning to a specific task. So, instead of training a model from scratch, a pre-trained model can be used to direct it to a specific task [43].

In our case we did use on this manuscript a pretrained model called 'BERT' and fine-tune it with a specific task called 'extractive question answering'.

- **Feed forward**

  As we know neural network is a group of neurons connected with each other, it simulate the brain structure and function. A feed forward is a simple neural network. It is just pass the information forward. It is the simplest ANN model. In this artificial neural network, the network has multiple layers which is input and output layer and between them there are one or more hidden layers, this kind of networks called MLP (Multi-Layer Perceptron) [3].

  This figure show the architecture of MLP and SLP (single layer perceptron):

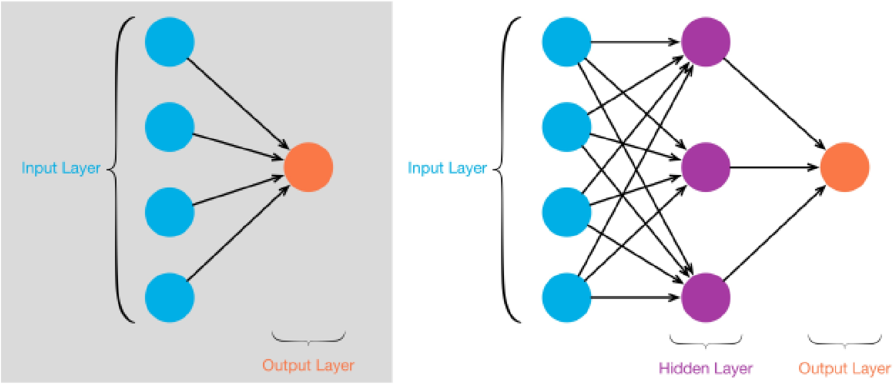


Figure 2.7: Feed Forward Architecture [3]

Pseudocode intent classification:

Algorithm 1 Intent classification model Training

```
Input: patterns, intents
Output: intention
inputs_train, intents_train ← data_preprocessing(inputs, intents)
inputs_train, intents_train, inputs_validation, intents_validation ← split_data(patterns,intents, 0.2)
model ← CreateSequentialModel()
model.add_Dense_layers(units,len(inputs))
model.add_Dense_layers(units, relu)
model.add_Dense_layers(units, relu)
model.add_Dense_layers(intents_length, softmax)
model.compile()
history ← model.fit(inputs_train, intents_train, validation_data=(inputs_validation, intents_validation), epochs=epochs)
results ← model.evaluate(test_data)
return results
```

- **Question Answering**

  Extractive question answering is the task that extract the answer from a given document or context, is a specific field of natural language processing (NLP), this kind of task focus on extracting an answer from a context, it is about searching an answer in a document[44]. Using the transformers library , we try to fine-tune BERT for question answering. There are a helpful hugging face courses to understand this task like [4]. As we can see on the figure, the answer is in the context, so when the model get the question it can predict the start and the end of the answer.

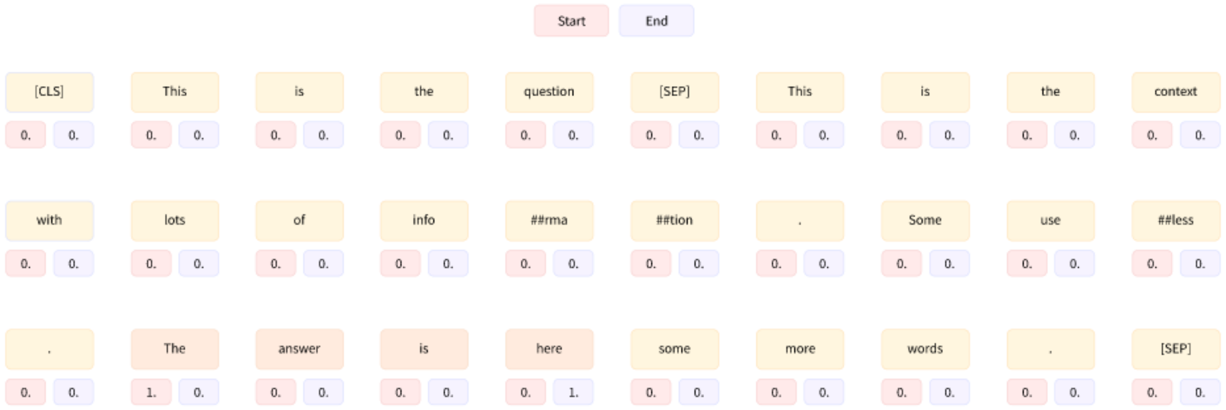


Figure 2.8: Extractive Question Answering [4]

The next listing present the used pseudocode to construct our question answering system.

Algorithm 2 Training question answering based Transformers model

```
Input: context, question
Output: start positions, end positions
squad ← load_datasets(squad)
data_train ← data_preprocessing(train_dataset)
data_validation ← data_preprocessing(validation_dataset)
model ← Load_BERT_Model()
model.compile()
model.fit(inputs_train, epochs=epochs)
metric ← load_metrics(squad)
results ← model.evaluate(metric)
return results
```

2.2.2.3 Data Security

During my programming for the blockchain, I created three classes, each class containing a set of functions, the first class for the administrator, the second for the nodes that are mining, and the third for the users.

(a) **Get transaction**

These functions are found at all nodes

i. **Read the data from the user who will perform a process :** If a person wants to register at the level of our application, I read his data to provide him with the following.

A. **Public key :** public key is created for the user.

B. **Private key :** We also create a special key that only the owner of the account should know because it is very dangerous if anyone has it.

C. **Pickle :** We also create a key named pickle, the latter of which is used to encrypt sensitive data, such as name, surname, etc., where no one can see it except the user.

ii. **Create transaction :** After that, a transaction is created that may be a registration or sending a currency or a request, for which we provide all the required inputs, since in the case of registration to create a new account, the data is encrypted except for the public key and the amount that should not be encrypted.

iii. **Create signature :** create a signature by a function called get signature whose inputs are transaction and private key where this function contains a function to create a signature called **ecdsa.sign()** .

iv. **Add transaction to pool :** The transaction signature has been verified. If true, add this operation to a temporary chain and send it to all mined nodes, both the transaction and the public key and signature to verify the transaction, if false, the transaction is rejected and not added to the temporary chain, when a certain size is reached in the temporary chain by nodes Metallurgical get into another process.

v. **Proof of work and Mining**

These functions are on mined nodes only.

A. **Verified transaction :** At this point, the following is executed: the mined node executes a function called valid transaction, this function eliminates all suspicious transactions, for example, that the same account performed the same operation at the same time, or that it sent a coin to itself, and so on.

B. **Creation block :** So that after verifying the mining nodes create a block in the form of a dictionary containing an index and a previous hash, i.e. the hash of the previous block plus transactions plus proof plus timestamp and hash.

C. **Mining :** At this stage, the nodes agree on the block and add it to the blockchain.

D. **Reach a solution :** The mining process is done by node mining, where in the mining process, each time the hash of the previous block, the current block, and the proof of the current block are taken, they are mounted in a chain, and then the hash of this chain is calculated. If this string's hash begins with 0009, the node will broadcast to the network. If not, 1 is added to the directory in the current block.

E. **Broadcast the solution in the network :** Once this block reaches the mined nodes, it performs a test if the block is valid and therefore does not complete mining and adds the extension block in a temporary chain and the same with all nodes and then it is agreed between the mined nodes on one block to add to the chain, but if it is not valid it will be rejected This block is from the side of the nodes.

(b) **Test attack :** After the previous process finishes, the mined nodes add the block to the **block.json** file, while the administrator adds the block to the **block.json** file and also to the cloud which is used as a backup in case his blockchain copy is hacked, where all the blocks are traversed and tested if the copy is have been hacked. If this happens, a copy is requested from the other null nodes, except for the administrator, who restores it from the cloud first. In the event of a hack, it restores the copy from other mined nodes.

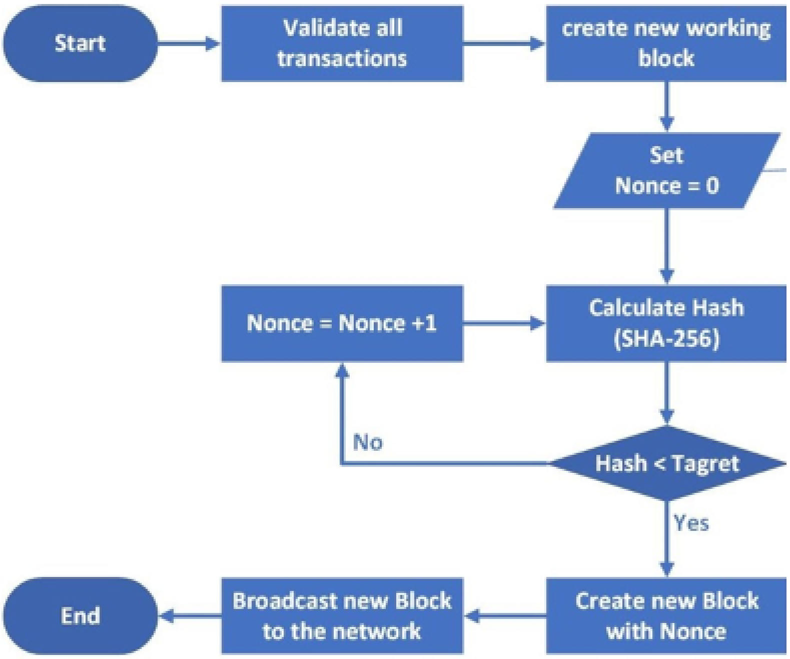


Figure 2.9: Diagram showing proof of work algorithm

(c) **Display information :** The mined nodes have the ability to reveal their account information in an encrypted way, but the administrator has a function in addition to the mined nodes, which is to reveal all non-encrypted accounts, while other users can reveal their accounts via a function that sends a request to the manager because he is the one who has all The information, and upon receiving the request, re-sends the account information.

used algorithme

- **Proof of work**

  It is an algorithm in which the network contract is unanimous in proving the addition of a block to the network. A fragmentation of the previous block and the current block is taken, and a hash is calculated for these two fragments. So by changing any byte of the two fragments, it changes the solution. Where the agreement is made in advance by the party to the network contract on a particular solution method, if the solution is not in the customary way, 1 is added to the fragmentation called proof at the end of reaching a solution that meets what is known by the network. If a solution is reached, the block is broadcast to the network; if it is true, it will be added to the chain; if it is not true, it will be blasted by consensus [45].

  - **Why is Proof of work ?** used because its use provides a high degree of security, as any change at the level of our blockchain chain requires change at the level of all nodes on the network level, and this is very difficult to happen [45].

- **pseudocode Proof of work**

```
$\text{Proof} \leftarrow 0$
$nodes \leftarrow address\_nodes$
$Block \leftarrow Createblock()$
$Stick \leftarrow Stick((hash(previous\_block), block, Proof(block))$
$Hash \leftarrow Calcul\_hash(stick)$
$While(debut(Hash)! = 00009)$
    $Proof \leftarrow Proof + 1$
    $Hash \leftarrow Calcul\_hash(stick)$
    $Ifdebut(Hash)! = 00009 :$
        $Send(block, nodes)$
$Agree\_on\_the\_block()$
$Save(block)$
```

2.2.2.4 Recommendation system

The overall structure of the recommendation system of our project is based on the collection of data related to the user evaluation of the craftsmen registered on our platform. This is done through collaborative filtering technique, artisans are recommended according to each user's history, on the other hand, craftsman are recommended through searches previously conducted by the user, through the use of content-based filtering technique, using TF-IDF algorithm and Cosine similarity, and building our recommendation system, we relied on this chart, which highlights the most important steps to building a good recommendation system :

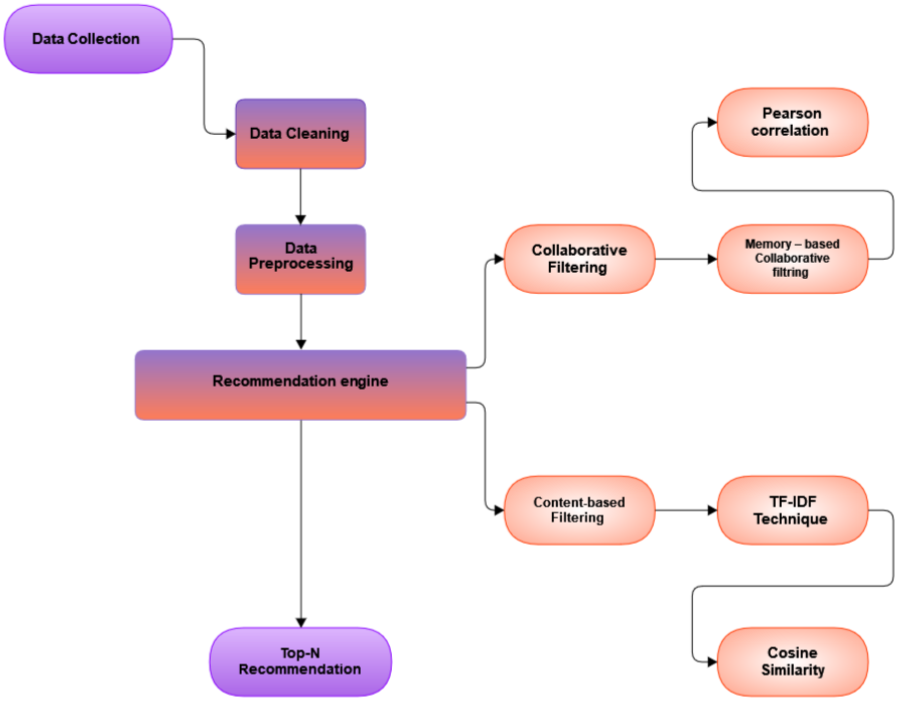


Figure 2.10: Recommendation System Process

1. **Data Collection**

   The first step is to collect data by importing it from sites that provide a free Dataset that corresponds to the data needed by our platform. For our Dataset, we have imported it from the following sites (Kaggle, data.world and BrianDunning.com), where the data has been modified to be compatible with our platform, and this data is organized in CSV format and then uploaded to the platform's database. And this DataSet contains (IdEmploi , NomEmploi , PrenomEmploi, NumTelephone, email, ville, ratingGlobal...), and in this table we do some description of the type of data in the DataSet.

Database	**Type**	**Description**
user_id	Integer	Is the identifier of each craftmen
firstname	String	Nickname of craftsman
lastname	String	Name of craftsman
phone	Integer	The phone number of the craftsman
address	String	Craftsman address
municipal	String	The municipal in which the craftsman lives
state	String	The state in which the craftsman lives
picturepersonal	String	Profile picture of the craftsman
craft	String	The craft of the craftsman
experience	String	The craftsman's experiences
facebook	String	The craftsman's account on Facebook
instagram	String	The craftsman's account on Instagram
youtube	String	Craftsman's YouTube channel
tiktok	String	The craftsman's account on Tiktok
ratingGlobal	Float	The overall rating of the craftsman in the platform

Table 2.1: Description of the database

2. **Data Cleaning**

   Data Cleaning is the process of preparing data for analysis by removing or altering invalid, incomplete, irrelevant, duplicate, or incorrectly formatted data. This data is generally not required or useful for data analysis as it may interfere with the process or lead to inaccurate results. There are different methods to clean data depending on how it is stored and what answers you are looking for. Data cleaning is considered a critical part of data science fundamentals because it plays an important role in the analysis process and in finding reliable answers [46].

   Here is a detailed description of the data set cleansing steps :

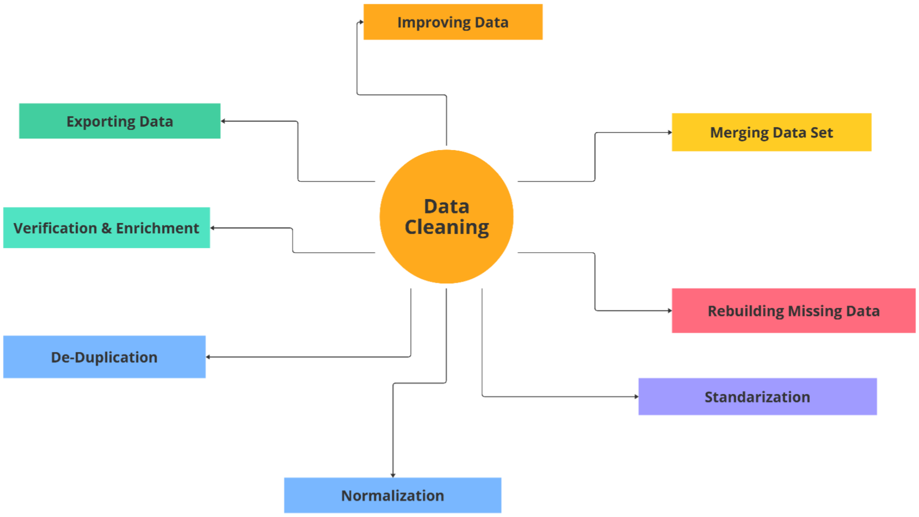


Figure 2.11: Steps for cleaning data [5]

3. **Data Pre-processing**

   Data pre-processing is an important step in the data mining process. It refers to the cleaning, transformation, and integration of data to prepare it for analysis. The purpose of data pre-processing is to improve the quality of the data and make it more suitable for a specific data mining task.

   These techniques are typically used in the early stages of machine learning and AI development to ensure accurate results [47].

   As for the data related to our platform, we removed it from all types of numbering and coding ... To prepare the data entry for processing step, consequently, to ensure good results using machine learning techniques.

4. **Recommendation engine**

   The reverse document frequency is used to make recommendations on user preferences that are based on their searches on the platform. Where each record is converted into a vector using the TF-IDF routing algorithm described earlier.

   The similarity measure for each craftsman is calculated with the content that was entered by the user, using the cosine similarity method. When the user requests a set of recommendations for specific craftsmen, this is done through the search engine in the user interface, by entering the name of the craft to be searched for and the name The city and the state in which he lives, and through these inputs, correlation coefficients are generated for the craftsmen related to the content of the user's inputs, and these recommendations are sorted from the highest similarity ratio to the lowest percentage and are presented to the user by the TF-IDF algorithm and cosine similarity method.

As for the recommendations made by the previously described collaborative filtering algorithm, it depends mainly on the rating that the user gives to the craftsman, and this rating is compared with the rating of users who rated the same craftsman with the same rating given by the previous user, and thus recommends to him other craftsmen who have By rating them highly by similar users and with this recommendation, there is a possibility for this user to like such recommendations. The results of these recommendations are due to the use of cosine similarity method .

As for the hybrid filter, the results of the previous two recommendation algorithms are combined.

5. **Top-N Recommendation**

   In this step, a file of recommendations is created for each user in several ways, according to his different behaviors on the platform.

Used Algorithms

In our project of creating a platform that connects craftsmen and customers, we have created a system of recommending craftsmen to customers according to their behavior on the platform, as our recommendation system relies on three technologies : Collaborative filtering, Content-based filtering and hybrid filtering.

- **Collaborative Filtering :**

  In terms of building intelligent recommendation systems, Collaborative filtering is the most common filter technique. that can make better recommendations by collecting a lot of information about users and about their preferences [48].

  In terms of calculating similarity between users, this is due to user ratings: if similar users rate items similarly. Collaborative filtering technology recommends targeted items to the user that have been rated highly by similar users [49].

As various global websites such as Amazon, Netflix and YouTube depend on their recommendation system on collaborative filtering as an important part of their recommendation systems. As for our platform , we used collaborative filtering to recommend craftsmen to the user through users similar to him in the ratings.

Where collaborative filtering depends on two types of techniques Memory – based Collaborative filtering and Model – based Collaborative filtering. In our project, we have worked using memory based collaborative filtering technique, specifically User based technique, which is one of the collaborative filtering techniques that seeks to improve recommendations for the user, based on the use of previous data to predict the unknown rating , This technique is based on its work on the following ratings User based.

As for calculating the similarity between elements or the similarity between users , we used the Pearson correlation method, which we explain as follows :

1. **Pearson Correlation :**

   It is a measure of the linear relationship between two variables, usually used to measure the degree of correlation between two variables. Where the linear correlation coefficient ranges between -1 and 1, where the value of -1 indicates a negative linear correlation, while 1 indicates a positive linear correlation, while the value of 0 means no correlation [50].

   The Pearson correlation formula is given as follows : [8]

$$\mathrm{sim}(a,u_i) = \mathrm{pearson}(a,u_i) = \frac{\sum_{j\in V}(r_{aj}-\overline{r_a})(r_{ij}-\overline{r_i})}{\sqrt{(r_{aj}-\overline{r_a})^2}\sqrt{(r_{ij}-\overline{r_i})^2}} \tag{2.1}$$

2. **User based :**

   This method relies on analyzing previous similar user ratings to recommend a specific item to a specific user. Where the principle of this method is to use elements of similarity between the active user and users similar to him in order to determine the closest neighbors, where the greater the similarity value, the closer the users are.

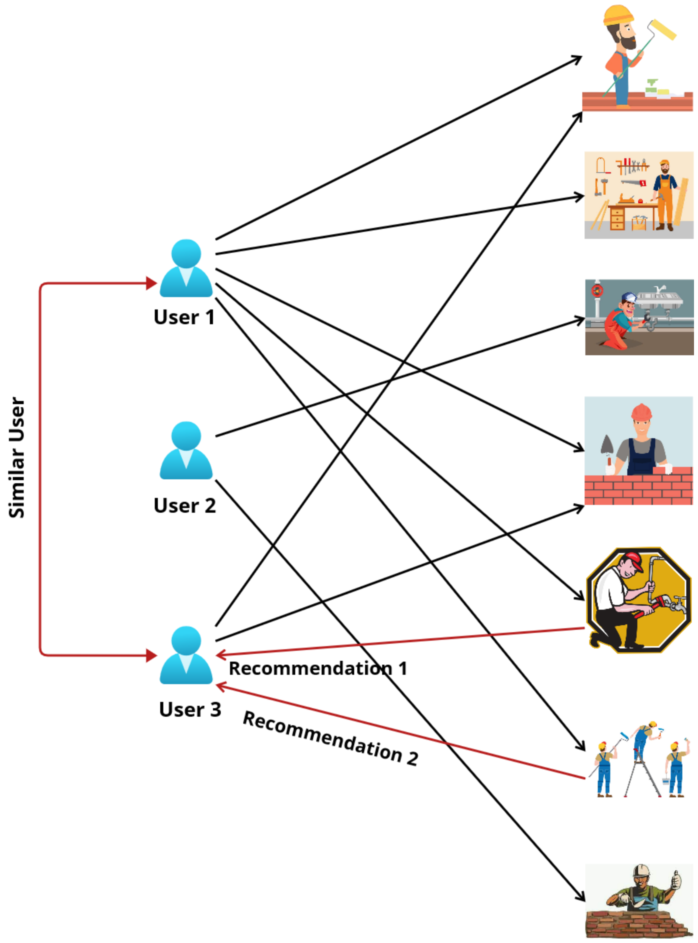


Figure 2.12: User based

To calculate similar neighbors, the following Pearson correlation equation can be used :

$$\mathrm{sim}(a,u_i) = \mathrm{pearson}(a,u_i) = \frac{\sum_{j\in V}(r_{aj}-\overline{r_a})(r_{ij}-\overline{r_i})}{\sqrt{(r_{aj}-\overline{r_a})^2}\sqrt{(r_{ij}-\overline{r_i})^2}} \tag{2.2}$$

To understand more about this technique and to know the similar neighbors of the active user, we offer this example applied to the Pearson correlation method :

- We suggest this table that represents 4 different items that are rated by 4 different users in order to explore similar users.

	Item 1	**Item 2**	**Item 3**	**Item 4**
User 1	$r_{11} = 1$	$r_{12} = 2$	$r_{13} = 1$	$r_{14} = 5$
User 2	$r_{21} = 2$	$r_{22} = 1$	$r_{23} = 2$	$r_{24} = 4$
User 3	$r_{31} = 4$	$r_{32} = 1$	$r_{33} = 5$	$r_{34} = 5$
User 4	$r_{41} = 1$	$r_{42} = 2$	$r_{43} = ?$	$r_{44} = ?$

Table 2.2: Rating matrix (user 4 is active user) [8].

$r_{i,j}$, i = user number, j = item number.

In our example, we want to calculate the similarity rate of users 1, 2 and 3 with user 4.

Firstly : We calculate the average ratings for both the active user and other users.

$$\overline{r_i} = \frac{1}{|I|} \sum_{j \in I} r_{i,j} \tag{2.3}$$

$$\overline{r_a} = \frac{1}{|A|} \sum_{j \in A} r_{a,j} \tag{2.4}$$

The first formula represents Average user ratings.

The second formula represents Average active user ratings.

$$\overline{r}_1 = \frac{1+2+1}{3} \simeq 1.33 \qquad \overline{r}_2 = \frac{2+1+2}{3} \simeq 1.66$$

$$\overline{r}_3 = \frac{4+1+5}{3} \simeq 3.33 \qquad \overline{r}_4 = \frac{1+2}{2} = 1.5$$

Secondly : Calculate the similarity between users.

We calculate the similarity between User 1 (U1) and User 4 (U4) :

$$
\begin{aligned}
\mathrm{sim}(u_4, u_1) &= \frac{(r_{41}-\overline{r}_4)(r_{11}-\overline{r}_1)+(r_{42}-\overline{r}_4)(r_{12}-\overline{r}_1)}{\sqrt{(r_{41}-\overline{r}_4)^2+(r_{42}-\overline{r}_4)^2}\sqrt{(r_{11}-\overline{r}_1)^2+(r_{12}-\overline{r}_1)^2}} \\
&= \frac{(1-1.5)(1-1.33)+(2-1.5)(2-1.33)}{\sqrt{(1-1.5)^2+(2-1.5)^2}\sqrt{(1-1.33)^2+(2-1.33)^2}} \simeq 0.95
\end{aligned}
$$

Calculating the similarity between User 2 (U2) and User 4 (U4) :

$$
\begin{aligned}
\mathrm{sim}(u_4, u_2) &= \frac{(r_{41}-\overline{r}_4)(r_{21}-\overline{r}_2)+(r_{42}-\overline{r}_4)(r_{22}-\overline{r}_2)}{\sqrt{(r_{41}-\overline{r}_4)^2+(r_{42}-\overline{r}_4)^2}\sqrt{(r_{21}-\overline{r}_2)^2+(r_{22}-\overline{r}_2)^2}} \\
&= \frac{(1-1.5)(2-1.66)+(2-1.5)(1-1.66)}{\sqrt{(1-1.5)^2+(2-1.5)^2}\sqrt{(2-1.66)^2+(1-1.66)^2}} \simeq -0.95
\end{aligned}
$$

Calculating the similarity between User 2 (U3) and User 4 (U4) :

$$
\begin{aligned}
\mathrm{sim}(u_4, u_3) &= \frac{(r_{41}-\overline{r}_4)(r_{31}-\overline{r}_3)+(r_{42}-\overline{r}_4)(r_{32}-\overline{r}_3)}{\sqrt{(r_{41}-\overline{r}_4)^2+(r_{42}-\overline{r}_4)^2}\sqrt{(r_{31}-\overline{r}_3)^2+(r_{32}-\overline{r}_3)^2}} \\
&= \frac{(1-1.5)(4-3.33)+(2-1.5)(1-3.33)}{\sqrt{(1-1.5)^2+(2-1.5)^2}\sqrt{(4-3.33)^2+(1-3.33)^2}} \simeq -0.87
\end{aligned}
$$

And from it we conclude that User 1 (U1) is the similar user to User 4 (U4) compared to Users 2 and 3, according to Pearson's correlation method.

Pseudo Code for Recommendation System (User-based) :

Algorithm 3 Recommendation System (User-based)

Input craftsman_df (craft), ratingtable (IdUser, IdCraftsman, rating), ratingUser (IdUser, IdCraftsman, rating)
Output Display recommendations

```
# Calculate similarity between users
shared_items ← {}
user_similarities ← {}
for all $user1$ in ratingtable.keys() do
    for all $user2$ in ratingtable.keys() do
        if $user1 \neq user2$ then
            items_user1 ← set(ratingtable[$user1$].keys()) # Set of items rated by user1
            items_user2 ← set(ratingtable[$user2$].keys()) # Set of items rated by user2
            #Shared items between user1 and user2
            shared_items[($user1, user2$)] ← items_user1 ∩ items_user2
            #List to store ratings of user1 for shared items
            ratings1 ← [ratingtable[$user1$][item] for item in shared_items[($user1, user2$)]]
            #List to store ratings of user2 for shared items
            ratings2 ← [ratingtable[$user2$][item] for item in shared_items[($user1, user2$)]]
            # Calculate the average ratings for a user1
            avg_rating1 ← sum(ratings1) / len(ratings1)
            # Calculate the average ratings for a user2
            avg_rating2 ← sum(ratings2) / len(ratings2)
            numerator ← $\sum_{i=0}^{\text{len}(ratings1)-1}(\text{ratings1}[i] - \text{avg\_rating1}) \cdot (\text{ratings2}[i] - \text{avg\_rating2})$
            denominator ← $\sqrt{\sum_{i=0}^{n-1}\left((\text{ratings1}[i] - \text{avg\_rating1})^2\right)} \cdot \sqrt{\sum_{i=0}^{n-1}\left((\text{ratings2}[i] - \text{avg\_rating2})^2\right)}$
            if denominator ≠ 0 then
                # Calculate similarity value between users
                user_similarities[(user1, user2)] ← $\frac{\text{numerator}}{\text{denominator}}$
            else
                # Users do not share any common item
                user_similarities[(user1, user2)] ← 0
            end if
        end if
    end for
end for
```

- **Content-Based Filtering :**

  Systems that apply the content-based recommendation approach analyze a set of documents or a set of items that the user has previously classified or searched for, and then a file of this user's interests is created [51].

  As for our project, we worked with the TF-IDF algorithm in order to analyze the content of the search for the user and create a file for his research in order to recommend to him items (ie the craftsmen in our project) similar to his interests.

  1. **Cosine similarity :**

     Cosine similarity is a measure utilized to assess the level of similarity between two vectors, commonly used in fields such as information retrieval, text classification, and machine learning. Cosine similarity is a conventional similarity measurement method incorporating the TF-IDF technique.

     The cosine similarity method is founded on the idea of measuring the angle between two vectors, where if the value of the angle is small between the two vectors here, the degree of similarity is high between the two vectors, and when the angle between the two vectors is 90°, that is, the degree of similarity is 0 This means that the two vectors are not similar if the angle is 180°, then the two vectors are opposite [52].

     The cosine similarity is calculated by the following formula :

     Cosine similarity:

     $$Sim_c(A,B) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \tag{2.3}$$

  2. **TF-IDF Technique :**

     TF-IDF (Term Frequency-Inverse Document Frequency) is a technique used in natural language processing and text analysis to determine the importance of a specific term within a collection of documents .

whereas :

(a) **TF (Term Frequency) :**

Calculates the number of times a term appears in a document. The higher the value, the greater the impact of the term in the document. TF is calculated as follows :

$$\text{TF(term)} = \frac{\text{Number of times term appears in a document}}{\text{Total number of terms in the document}}$$

Figure 2.13: The mathematical formula for TF

(2.4)

In order to improve and enhance the accuracy of the TF-IDF technology, the following rule has been developed :

$$TF = (1 + \log(\text{TF}_{t,d}))$$

(2.5)

(b) **IDF (Inverse Document Frequency) :**

Is a weight that indicates the popularity of a term. As a term is used more frequently across documents, its IDF value decreases, making it less significant [53], IDF (Inverse Document Frequency) is calculated as follows :

$$IDF_t = \log_{10}\left(\frac{N}{df_t}\right)$$

(2.6)

Or :

$$\text{DF(term)} = \frac{\textbf{d}\text{(Number of documents containing a given term)}}{\textbf{D}\text{(The size of the collection of documents)}} \tag{2.7}$$

$$\text{IDF(term)} = \log\left[\frac{\text{Total number of documents}}{\text{Number of documents with a given term in it}}\right] \tag{2.8}$$

(c) **TF * IDF :**

To calculate the TF-IDF value, we multiply the TF value by the IDF value. A high TF-IDF value indicates that a term appears frequently in a specific document while being rare in the overall document collection. Thus, the term is considered highly important in that particular document. TF-IDF is calculated as follows :

$$W_{t,d} = (1 + \log \text{TF}_{t,d}) \times \log_{10}\left(\frac{N}{\text{df}_t}\right) \tag{2.9}$$

In order to illustrate this technique, we suggest this example : [54]

- We have a number of sentences :

The sky is blue.

The sun is bright today.

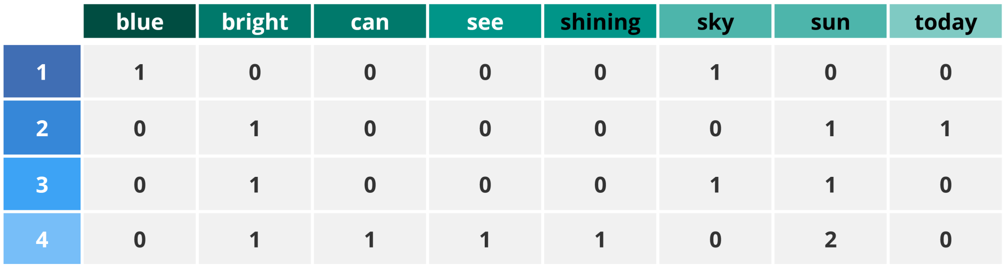
The sun in the sky is bright.

We can see the shining sun, the bright sun.

In order to calculate the importance of each term, we calculate the TF-IDF of each term in each sentence.

Firstly we calculate the TF for each term :

$$f_{t,d}$$

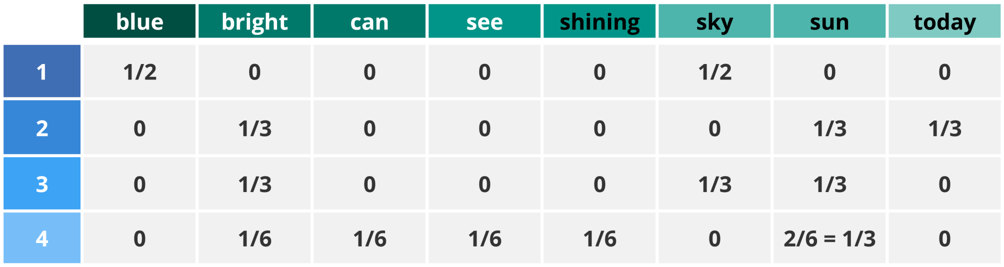


	blue	bright	can	see	shining	sky	sun	today
1	1	0	0	0	0	1	0	0
2	0	1	0	0	0	0	1	1
3	0	1	0	0	0	1	1	0
4	0	1	1	1	1	0	2	0

Figure 2.14: matrix to repeat the term in each sentence.

Calculating the importance of all terms in the sentence by neglecting the stop words.

$$\mathtt{tf}(t,d) = \frac{f_{t,d}}{\sum_{t'} f_{t',d}}$$

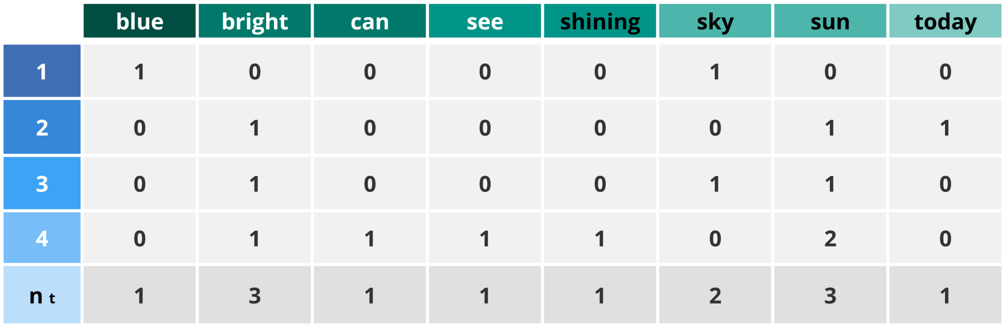


	blue	bright	can	see	shining	sky	sun	today
1	1/2	0	0	0	0	1/2	0	0
2	0	1/3	0	0	0	0	1/3	1/3
3	0	1/3	0	0	0	1/3	1/3	0
4	0	1/6	1/6	1/6	1/6	0	2/6 = 1/3	0

Figure 2.15: Matrix for the importance of all terms in the sentence .

Secondly We calculate the IDF :

$$f_{t,d}$$



	blue	bright	can	see	shining	sky	sun	today
1	1	0	0	0	0	1	0	0
2	0	1	0	0	0	0	1	1
3	0	1	0	0	0	1	1	0
4	0	1	1	1	1	0	2	0
n t	1	3	1	1	1	2	3	1

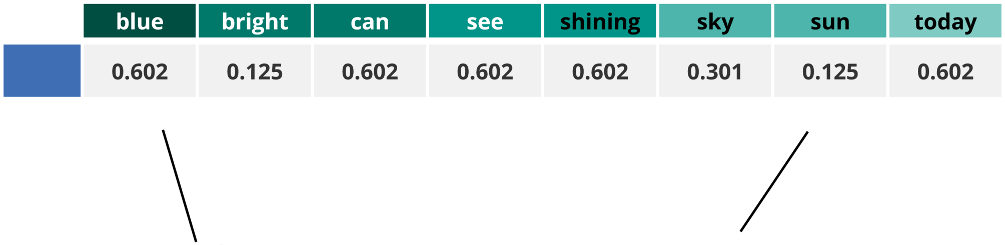
Figure 2.16: Matrix for repeating the term across all sentences DF .

Calculate the IDF value :

We have : N = 4

N : The total number of sentences .

$$\mathtt{idf}(t, D) = \log_{10} \frac{N}{n_t}$$

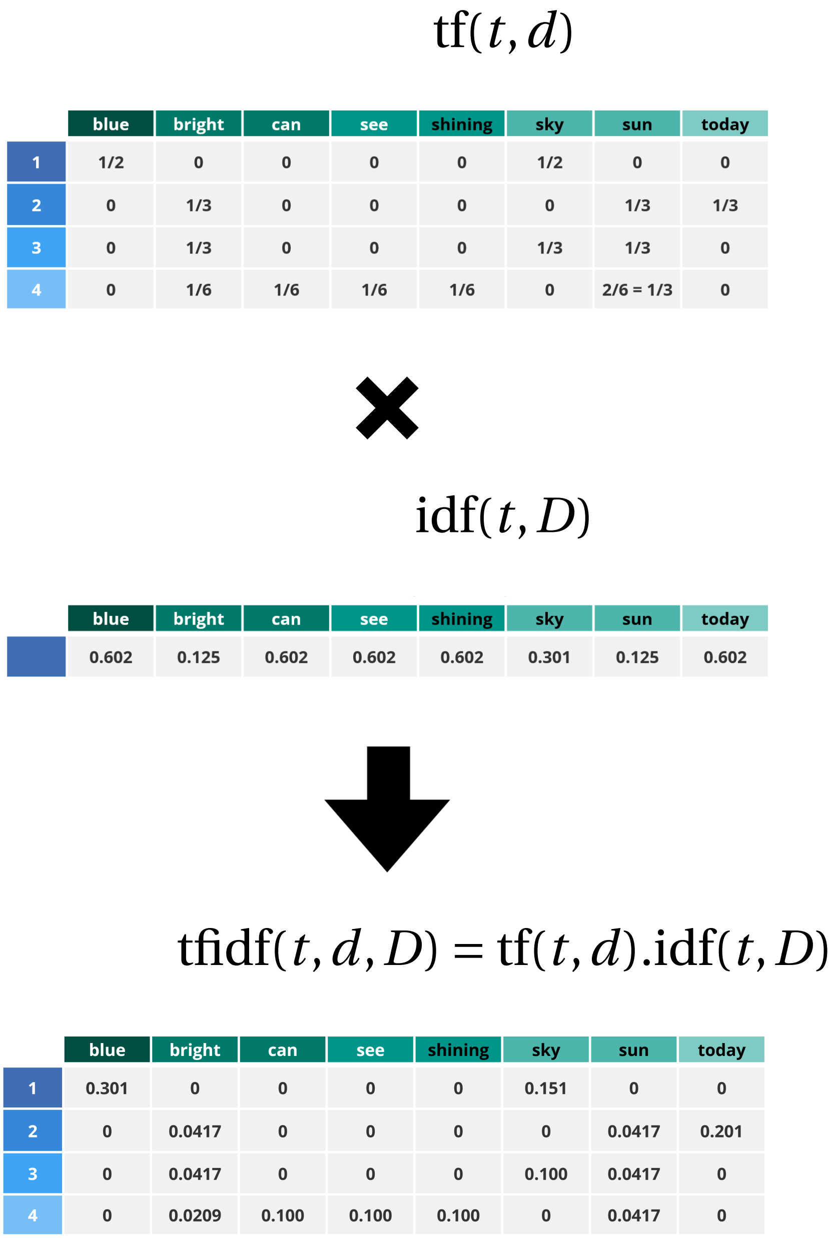


	blue	bright	can	see	shining	sky	sun	today
	0.602	0.125	0.602	0.602	0.602	0.301	0.125	0.602

$$\log_{10} \frac{4}{1} = 0.602 \qquad \log_{10} \frac{4}{3} = 0.125$$

Figure 2.17: Matrix containing IDF values .

Finally We multiply the two values TF with IDF in order to identify the important words .



$$\mathrm{tf}(t,d)$$

	blue	bright	can	see	shining	sky	sun	today
1	1/2	0	0	0	0	1/2	0	0
2	0	1/3	0	0	0	0	1/3	1/3
3	0	1/3	0	0	0	1/3	1/3	0
4	0	1/6	1/6	1/6	1/6	0	2/6 = 1/3	0

×

$$\mathrm{idf}(t,D)$$

	blue	bright	can	see	shining	sky	sun	today
	0.602	0.125	0.602	0.602	0.602	0.301	0.125	0.602

$$\mathrm{tfidf}(t,d,D) = \mathrm{tf}(t,d).\mathrm{idf}(t,D)$$

	blue	bright	can	see	shining	sky	sun	today
1	0.301	0	0	0	0	0.151	0	0
2	0	0.0417	0	0	0	0	0.0417	0.201
3	0	0.0417	0	0	0	0.100	0.0417	0
4	0	0.0209	0.100	0.100	0.100	0	0.0417	0

Figure 2.18: Important terms matrix .

Pseudo Code for Recommendation System (TF-IDF) :

Algorithm 4 Recommendation System (TF-IDF Model)

Input craftsman_df (craft , address , state) , search_df (craft , address , state)
Output Display recommendations

```
# Load craftsman data from the database
# Load search query data from the database
# Preprocess craftsman data
# Combine relevant fields into a single text field
craftsman_df(text) ← Combine_craft_Fields(craftsman_df)
vectorizer ← Create_Tfidf_Vectorizer()    # Create a TF-IDF vectorizer
text_data ← Extract text data and fill missing values
tfidf ← vectorizer.transform(text_data)    # Apply TF-IDF vectorizer
tfidf_matrix ← tfidf.toarray()    # Store TF-IDF matrix
for all index = 0 →length of search_df - 1 do
    cleaned_craft ← Clean_Title(craft)    # Clean the search query
    if User_Exists(IdUser) then
        # Transform the search query into a TF-IDF vector
        query_vec ← Transform_Query(cleaned_craft , address , state, vectorizer)
        # Calculate cosine similarity between the search query vector and craftsman data
        similarity_scores ← Calculate_Cosine_Similarity(query_vec, tfidf)
        # Get the top similar indices
        indices ← Get_Top_Similar_Indices(similarity_scores)
        # Calculate cosine similarity score based on text fields only
        similarity_score ← Calculate_Cosine_Similarity_Score(query_vec, craftsman_df(text))
        # Update the current results DataFrame with the relevant craftsman data
        current_results ← Update_Current_Results(indices, similarity_scores, similarity_score, craftsman_df)
    end if
    # Append the current results to the overall results DataFrame
    all_results ← Append_Current_Results(current_results)
end for
```

- **Hybrid recommendation :**

  Hybrid Recommendation is an approach that combines various different recommendation techniques, such as collaborative filtering and content-based filtering. This type of recommendation aims to improve the accuracy and effectiveness of the recommendation system by taking advantage of the strengths of each technique and compensating for the weaknesses of others. For our project, we worked on a hybrid recommendation system that combines collaborative filtering with content-based filtering in order to improve our recommendation system.

2.2.2.5 Private Cloud

The private cloud structure we worked with is a two- or dual-network architecture. It is an architecture where there are two networks one private (internal) and the other public (external) the two are connected by a router to ensure connectivity between them.

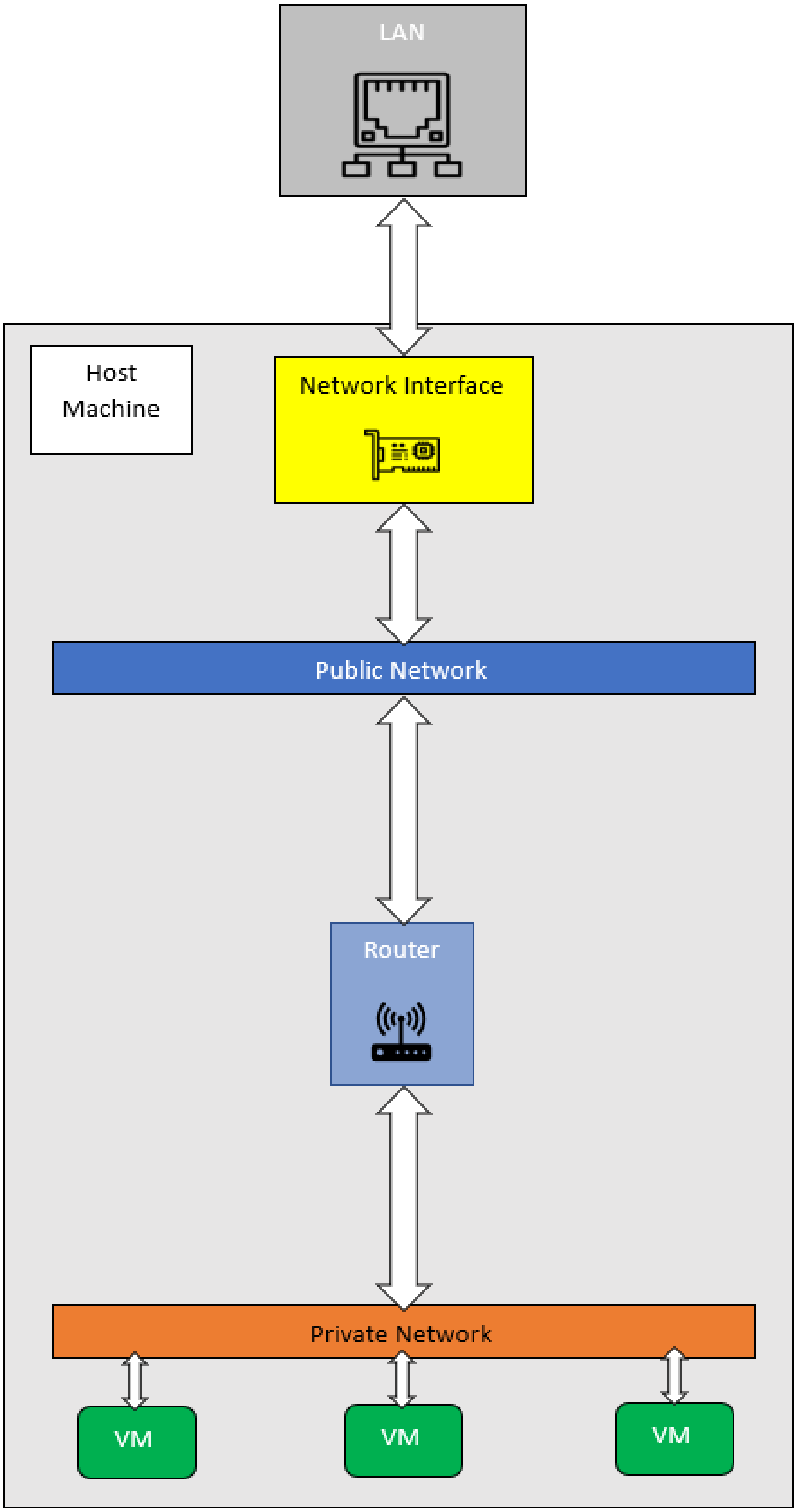


Figure 2.19: Dual-Network Architecture

1. **Private Network:**

   (a) **Isolation:** The private network in the private cloud provides a secure and isolated environment for the infrastructure's virtual machines (VMs). It permits internal communication among VMs whilst avoiding direct network exposure.

   (b) **Enhanced Security:** By placing the VMs on a private network, you may use security measures like firewall rules and access controls to govern and monitor network traffic.

2. **Public Network:**

   (a) **Internet Access:** The external network (named public) in the private cloud allows VMs with floating IPs to have external network connectivity and access to internet resources. It enables contact with external services, package downloads, program upgrades, and access to external APIs.

   (b) **Service Exposure:** By connecting the private cloud's public network to the internet, you can expose specific services or APIs operating on the VMs to other users or systems. This is especially beneficial when you want to make your applications or services available to users who are not on your private network.

3. **router:**

   (a) **Inter-Network Communication:** The router in the private cloud serves as a bridge between the private and public networks, allowing them to communicate with one another. It enables private network VMs to access public network resources such as external services or the internet, and vice versa.

   (b) **Network segmentation**: is provided by the router in the private cloud, which allows for logical separation of the private and public networks. It allows you to set routing rules, control traffic flow, and manage network connectivity.

   (c) **Flexibility and scalability:** With a private cloud router, you can quickly add and remove networks and sub nets from your architecture. It provides scalability and flexibility as your infrastructure and networking requirements shift over time.

CPU Mode, Model and Performance: To optimize the performance of the virtual machines (VMs) in our private cloud and meet the requirements of APIs and website hosting, we implemented CPU pass-through by setting the CPU mode to "host-passthrough." This configuration enables the VMs to fully leverage the capabilities, features, and specific CPU model of the host machine.

By adopting the host-passthrough mode, the VMs gain access to the complete set of CPU flags, instructions, and performance-enhancing features available on the host machine. This ensures that the VMs can utilize the same CPU model and take advantage of any specialized functionalities provided by the underlying hardware.

The CPU pass-through approach optimizes the performance of the VMs, enabling them to efficiently handle the demands of the APIs and website hosting tasks. By aligning the CPU capabilities of the VMs with those of the host machine, we can maximize performance and achieve a seamless and responsive computing experience within our private cloud environment.

Virtual machines Security: By associating a keypair to a virtual machine, you create a safe authentication method. Only users with the correct private key can authenticate and gain access to the VM. This helps to guarantee that only authorized individuals may access the VMs, which improves overall security.

1. **Keypairs:**

   (a) **Secure Remote Access:** SSH (Secure Shell) keypairs are used for secure remote access to VMs. When you generate a keypair, a public key is associated with the VM, while the user keeps the accompanying private key. This asymmetric key encryption ensures safe communication between the user and the virtual machine, preventing unauthorized access.

   (b) **Authentication and Authorization:** By associating a keypair to a virtual machine, you create a safe authentication method. Only users with the correct private key can authenticate and gain access to the VM. This helps to guarantee that only authorized individuals may access the VMs, which improves overall security.

2. **Security Groups:**

   (a) **Firewall Rules:** Security groups specify a collection of firewall rules that govern inbound and outbound traffic to virtual machines. You can allow or reject specific types of network traffic to and from the VM by establishing security group rules. This prevents unwanted access and malicious activity.

   (b) **Traffic Filtering:** Granular network traffic controls can be defined using security groups. Protocols, ports, and IP ranges can all be allowed or prohibited. This protects virtual machines against unwanted network access by, for example, restricting access to unnecessary ports or services.

   (c) **Isolation and Segmentation:** Security groups enable you to divide VMs into logical groups and restrict their communication. You can block communication across VMs within the same security group or allow communication across specified ports or protocols by establishing the proper rules. This adds an extra degree of security by blocking unauthorized access between VMs.

3. **Floating IPs:** Floating IPs provide VMs with publicly available IP addresses, allowing them to communicate with the outside world while retaining network isolation. They provide selective exposure, allowing you to provide floating IP addresses just to VMs that require public access. NAT is used to translate outward traffic, masking the VM's private IP address.

2.2.3 UML Diagrams

A UML diagram is a visualization tool that plays a vital role in software engineering and system design. It provides a unified language to capture and communicate the various aspects of a software system or application. A UML diagram allows developers, designers, and stakeholders to visually understand the structure, behavior, and relationships of system components. In this section, we will introduce the UML diagram and its importance in representing the architecture of a system. The UML diagram includes different types of diagrams, each serving a specific purpose. We will study three types: Sequence Diagrams, use case Diagrams, and Class Diagram. We will delve into its components and applications. We explore how each schema type contributes to a general understanding of system structure and behavior [55].

First we identify the interacting people in the system:

1. **Craftsman**

   These are professionals who offer their skills and services to clients in exchange for payment. Craftsmen can come from a wide variety of fields, such as plumber, masonry, electrician, and many more.

2. **Clients**

   These are individuals or organizations that seek the services of craftsmen to help with specific projects or tasks. Clients browse the craftsmans on the platform and connect with them to find the one that best suits their needs.

3. **Administrators**

   These are individuals or teams that manage the platform, enforce its rules and policies, and provide support to both craftsmen and clients. They are responsible for ensuring that the platform runs smoothly and that users have a positive experience.

2.2.3.1 Use Case Diagram

The objective of a use case diagram is to visually represent the interactions and relationships between actors and use cases within a system, providing a high-level overview of the system's functionality and the roles involved.

1. **Use Case Diagram - Client**

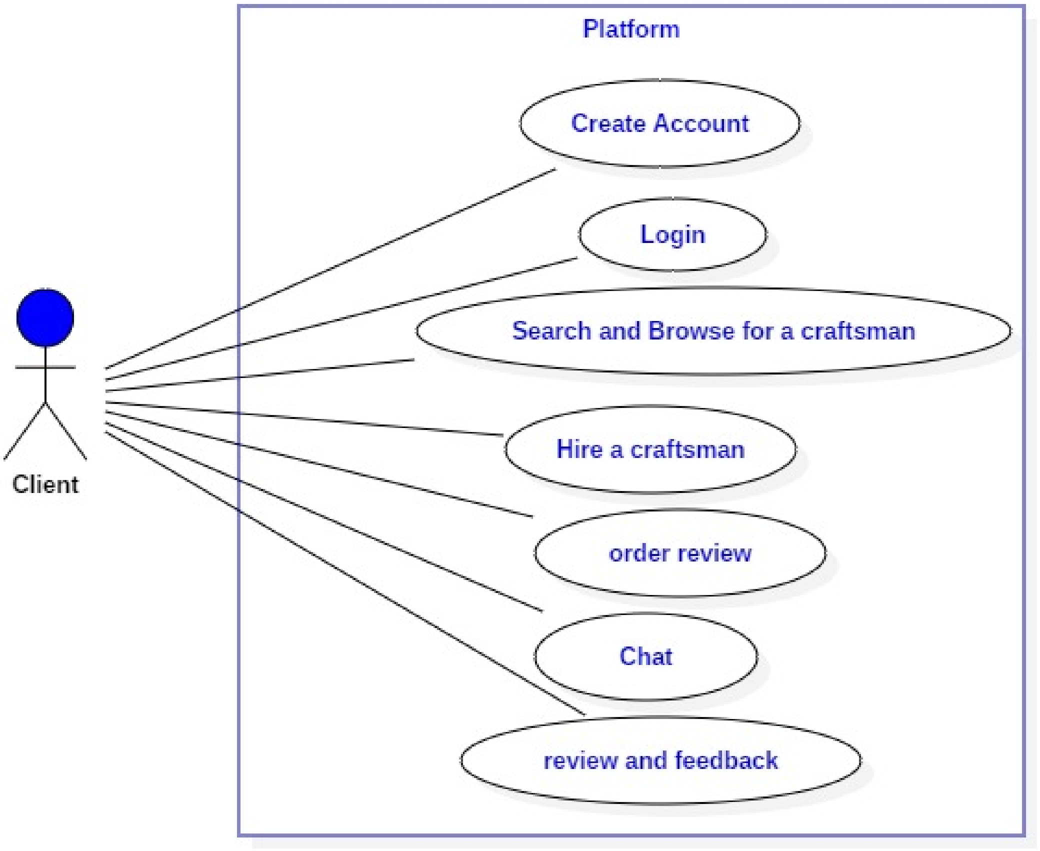


Figure 2.20: Use Case Diagram - Client

(a) **Create Account:** clients can create an account on the platform.

(b) **Search and Browsing craftsmen:** : Clients can search and browse the profiles of craftsmen on the platform to find someone who is a good fit for their project.

(c) **Hiring craftsmen:** Clients can hire craftsmen by placing an order for their services.

(d) **Review order:** clients can review their order.

(e) **Review and feedback:** Both the client and the craftsman can leave reviews and feedback for each other after the completion of a project.

These reviews are displayed on each user's profile and can help other users to make informed decisions when choosing a freelancer or client.

(f) **Chat:** The client and craftsman communicate through the platform's messaging system to exchange updates and complete the project.

2. **Use Case Diagram - Craftsman**

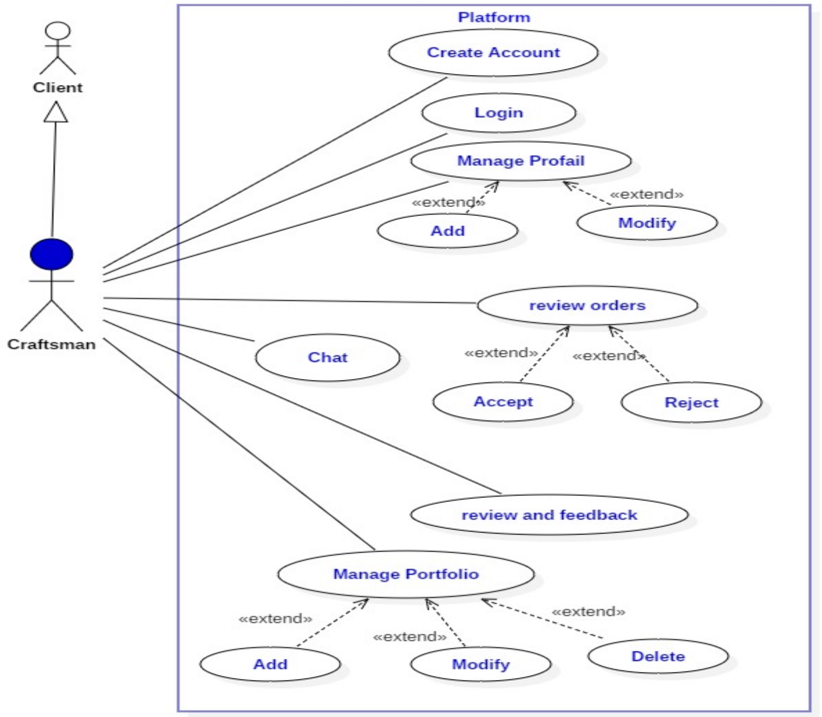


Figure 2.21: Use Case Diagram - Craftsman

(a) **Create Account:** Creating a profile and showcasing their portfolio.

(b) **Manage Account:** : This use case represents the ability of a user to manage their account information, including their profile, and order history.

(c) **Manage Portfolio:** This use case represents the user's ability to manage their portfolio information (add, edit, and delete).

(d) **Review order:** Reviewing order, the order can be accepted or rejected.

3. **Use Case Diagram - Admin**

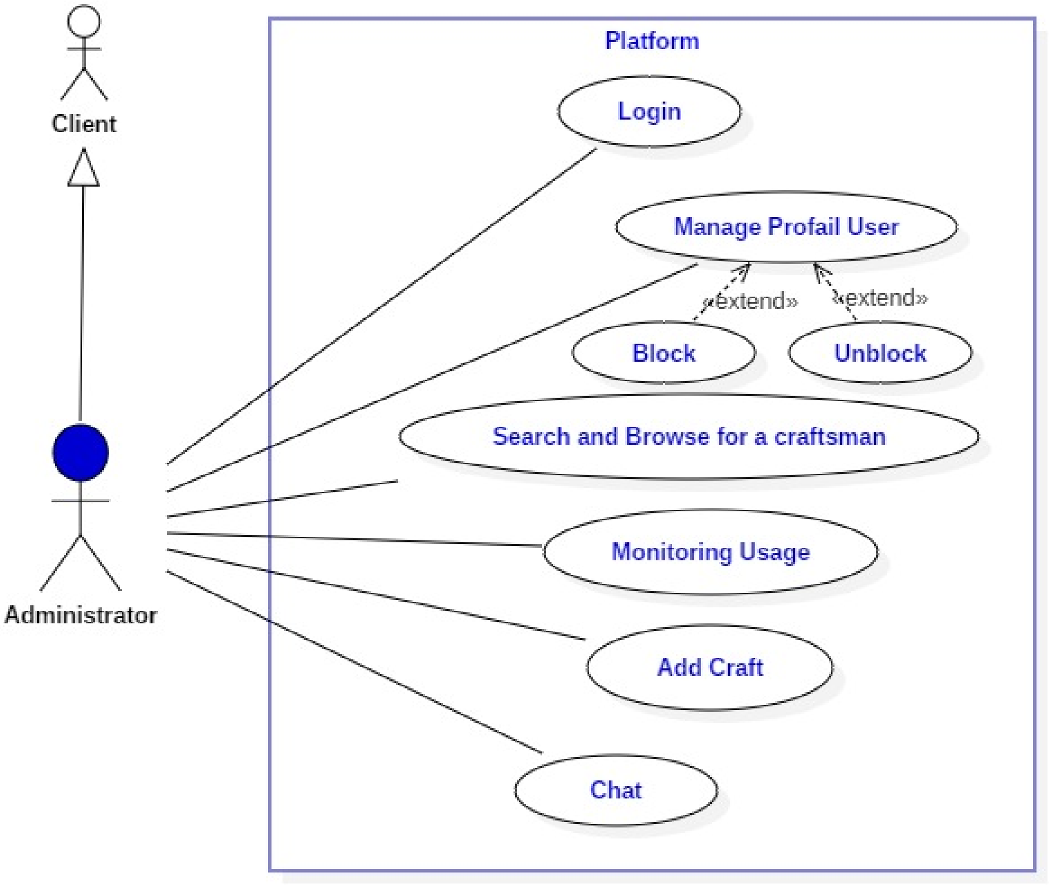


Figure 2.22: Use Case Diagram - Admin

(a) **Monitoring usage:** Administrators can monitor the usage of the platform, tracking key metrics such as the number of projects posted and the number of freelancers active on the platform.

(b) **Managing Profile users:** Managing user accounts and ensuring that all users are following the platform's policies and guidelines.

(c) **Add Craft:** The administrator can add a new crafts if it does not exist.

2.2.3.2 Sequence Diagram

The purpose of a sequence diagram is to visually represent the interactions and flow of messages between objects or components in a system, illustrating the dynamic behavior of the system. It helps in understanding the order of message exchanges and the collaboration between different elements [56].

1. **Sequence Diagram - Create Account**

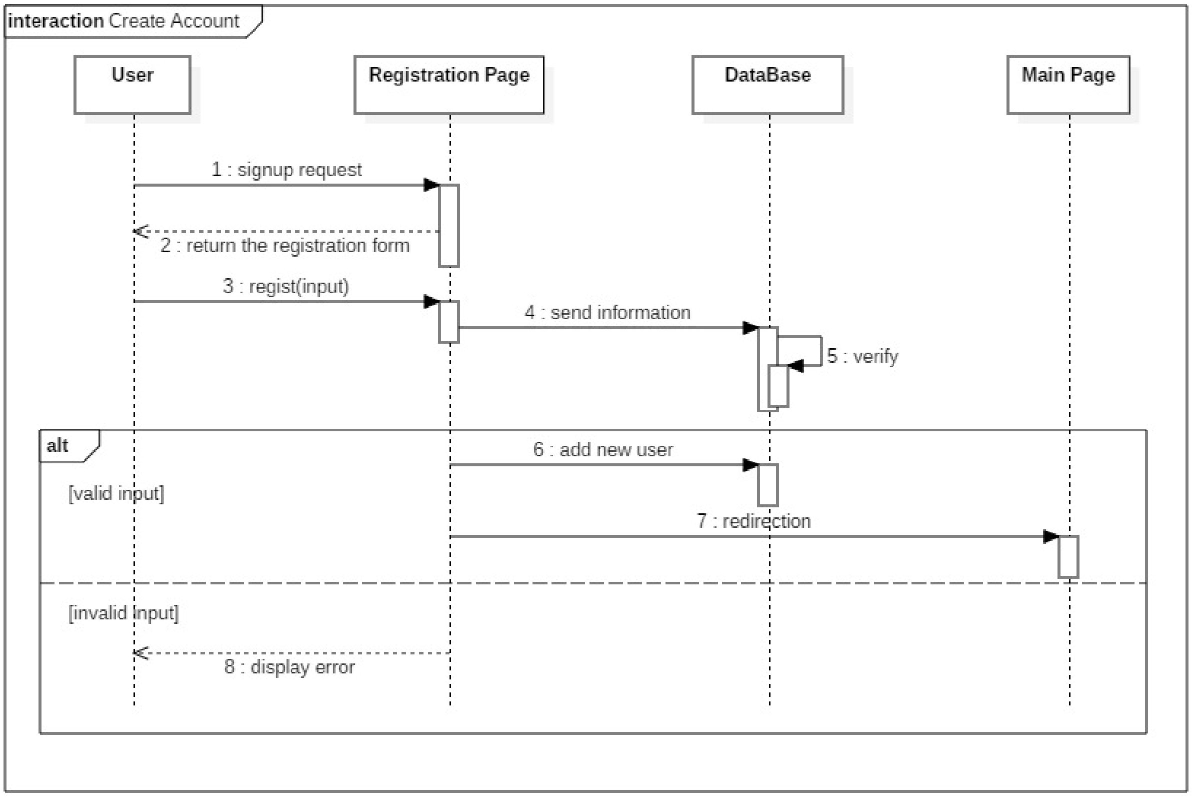


Figure 2.23: Sequence Diagram - Create account

In this sequence diagram, the client (i.e. the user) interacts with the website to create an account. The website displays the appropriate pages and communicates with the database to store the new user information and verify the login information. If there are any issues during the process, such as invalid registration information or failed login verification, the website displays an error message to the client.

2. **Sequence Diagram - Authentication**

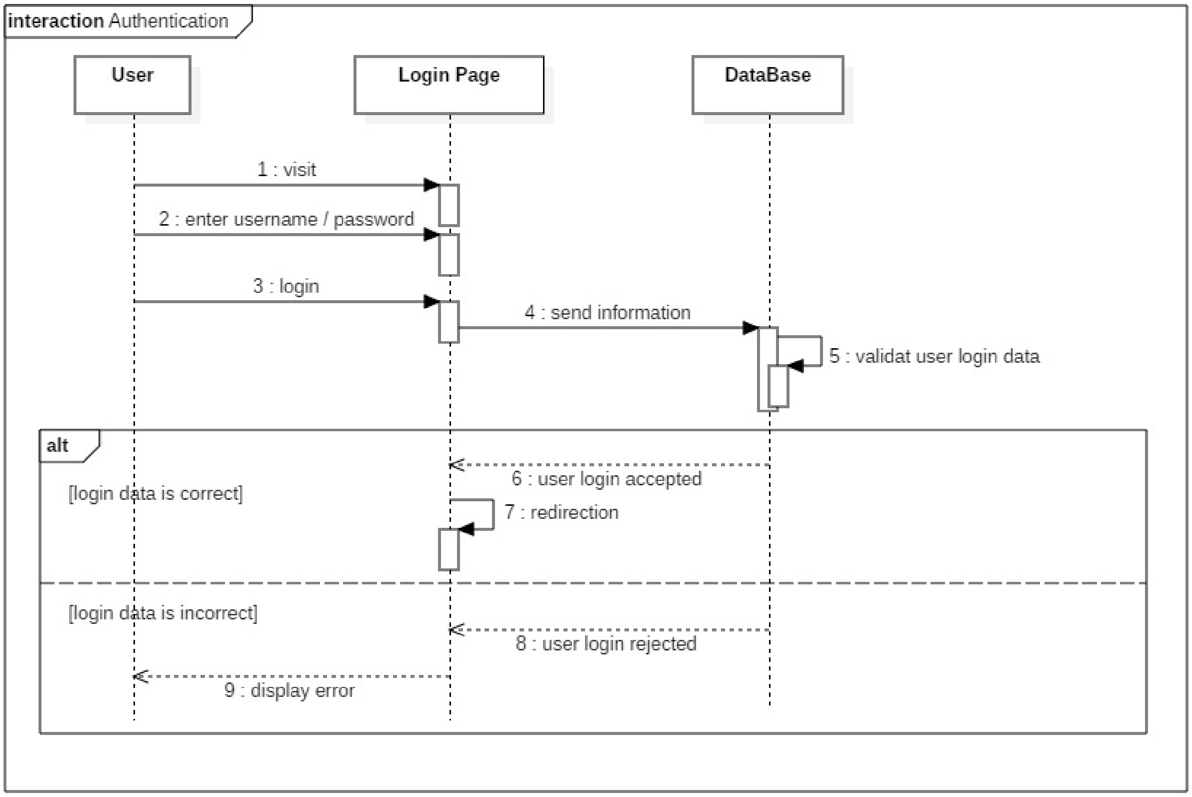


Figure 2.24: Sequence Diagram - Authentication

In this sequence diagram, the user enters their login credentials on the website. The website then sends these credentials to the authentication server, which queries the database to check if the credentials are valid. The database sends the user data back to the authentication server, which then sends the authentication result (either access granted or denied) back to the website. The website then displays the appropriate page to the user based on the authentication result.

3. **Sequence Diagram - Add Post**

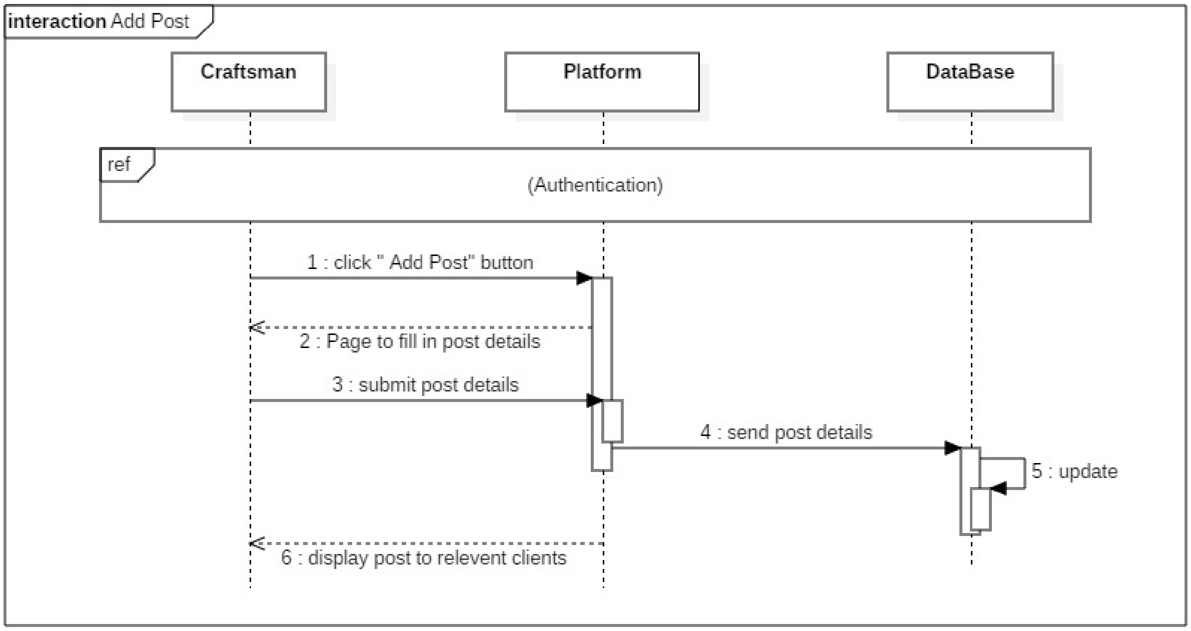


Figure 2.25: Sequence Diagram - Add Post

- The craftsman logs into the platform and clicks on the "add Post" button.
- The platform prompts the craftsman to fill in the project details, such as the project title, description, budget, and timeline.
- The craftsman submits the project details.
- The platform presents the project to interested freelancers who may be interested in hiring the craftsman.

4. **Sequence Diagram - Hire a Craftsman**

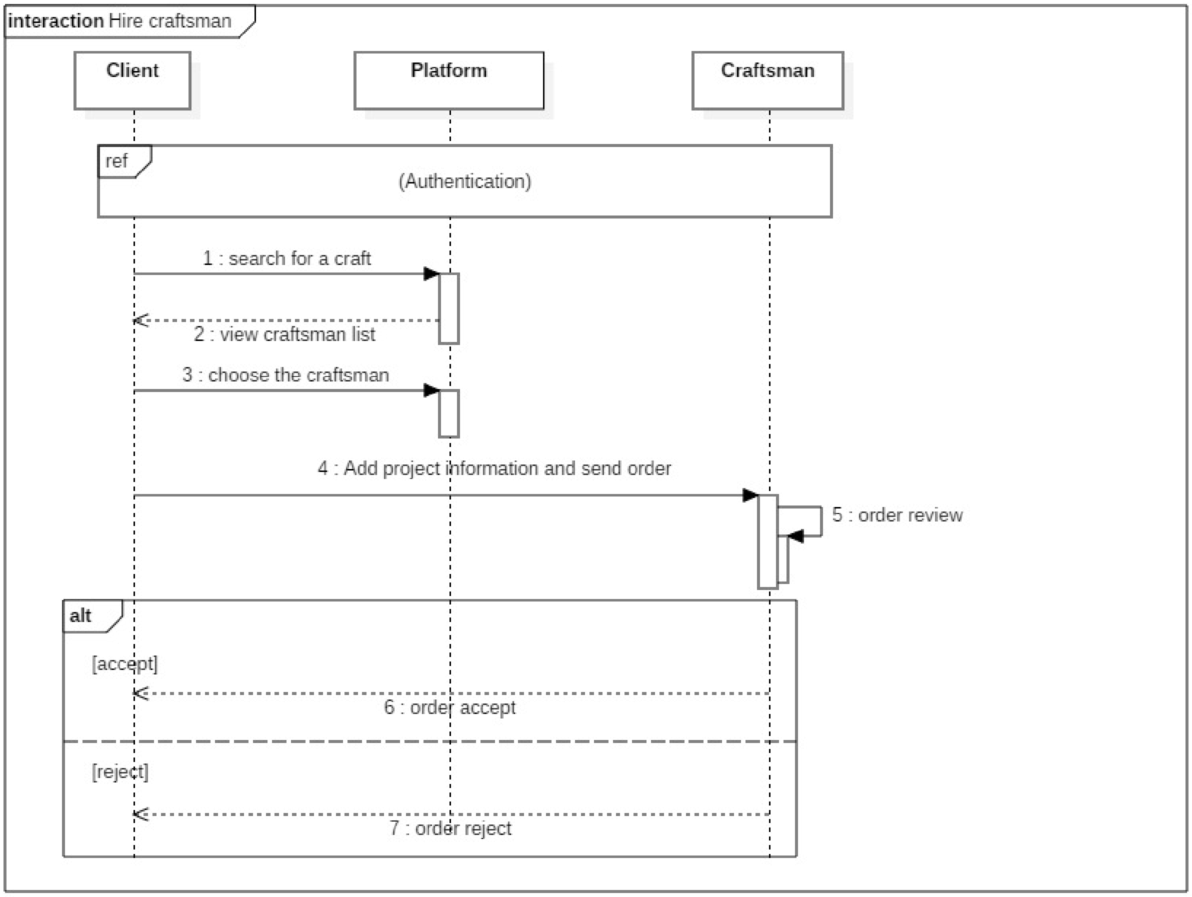


Figure 2.26: Sequence Diagram - Hire a Craftsman

The client logs into the platform, the client searches for a craft, browses the craftsmen, selects the suitable craftsman and hires him.

He sends him a request that contains information about the project that he wants to complete, and he awaits a response from the craftsman.

5. **Sequence Diagram - Order Accept**

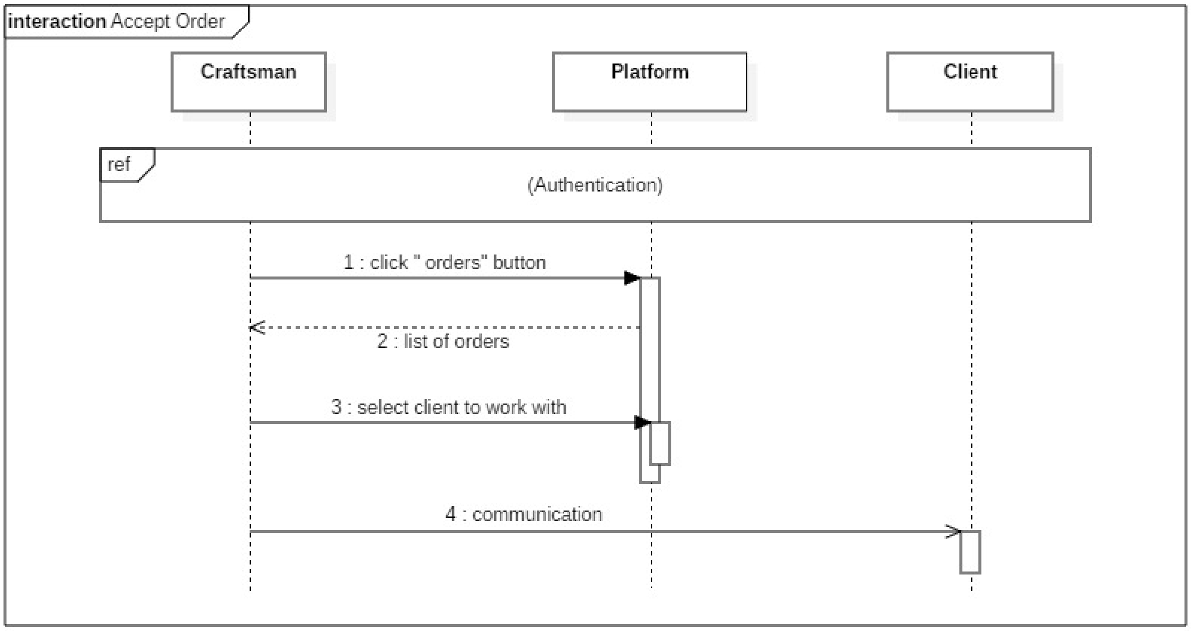


Figure 2.27: Sequence Diagram - Order Accept

The client logs into the platform and clicks on the "orders" button.
The craftsman reviews the proposals and selects the client they would like to work with.
The craftsman and client communicate to discuss the project details and agree on the terms and conditions.

6. **Sequence Diagram - Rating and Feedback**

- The craftsman complete the project.
- The client initiates the review and feedback process by accessing the project's review section.
- The client provides a rating and writes a review for the craftsman, highlighting their experience working together and the quality of the delivery.
- The platform notifies the craftsman about the new review and rating.

- The craftsman can also provide their own rating and review for the client, sharing their experience of working with the client.

7. **Sequence Diagram - Make Payments**

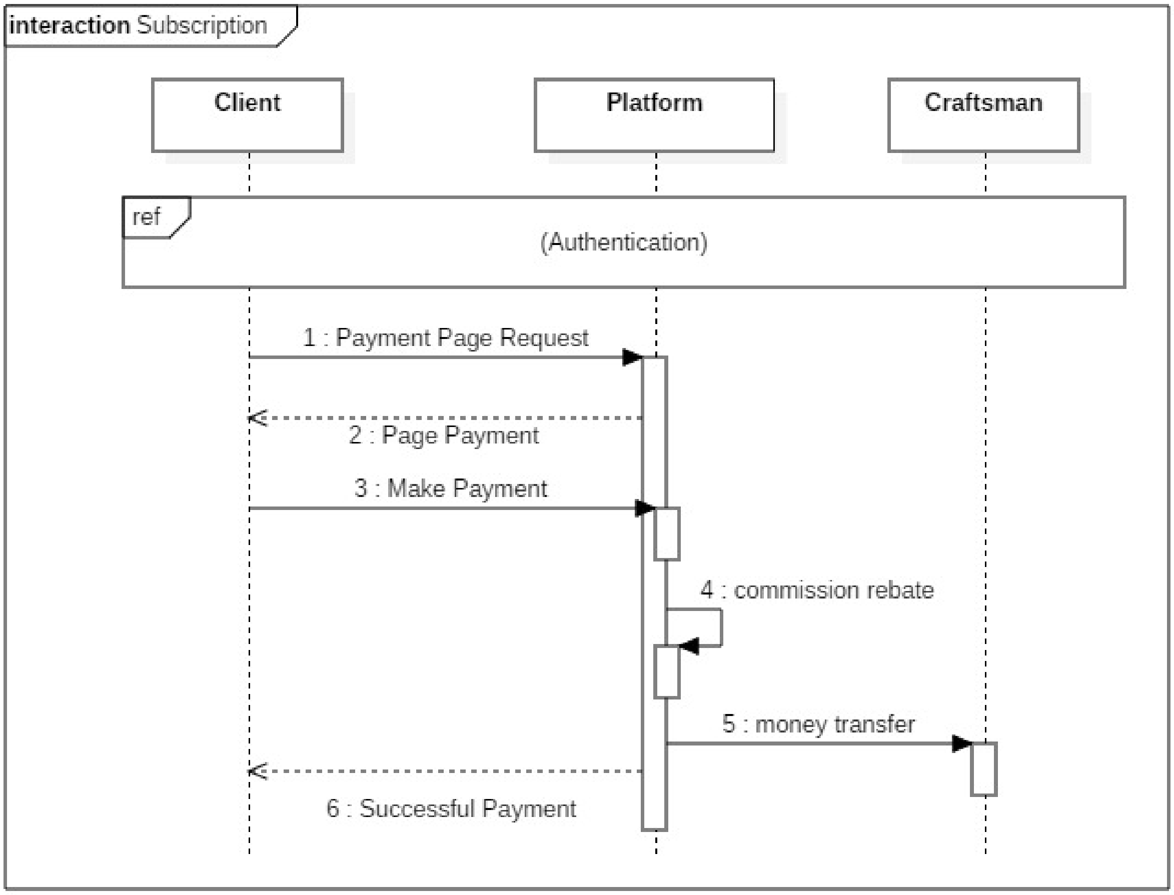


Figure 2.28: Sequence Diagram - Make Payments

The customer requests the payment page, the customer makes the payment through the platform's secure payment system. The platform deducts its own commission and sends the money to the craftsman.

8. **Sequence Diagram - Monitoring Usage**

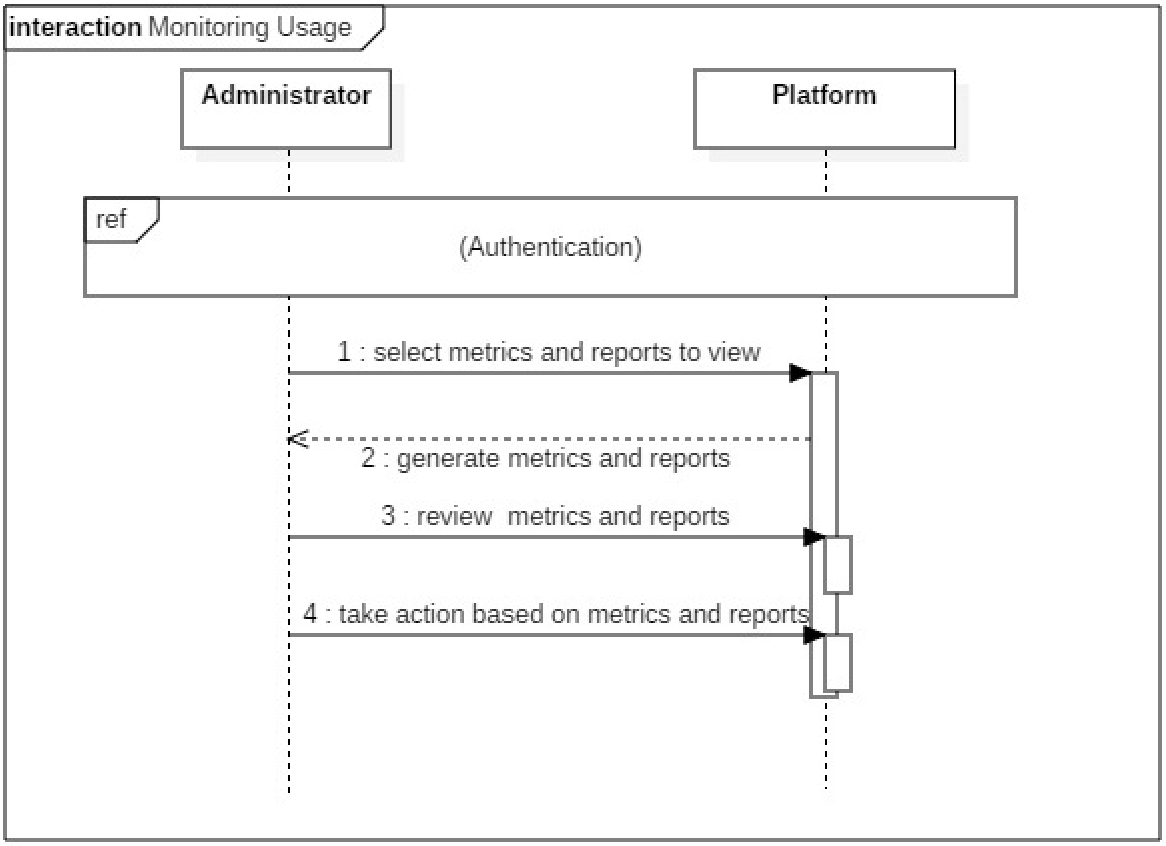

interaction Monitoring Usage
Administrator
Platform
ref
(Authentication)
1 : select metrics and reports to view
2 : generate metrics and reports
3 : review metrics and reports
4 : take action based on metrics and reports

Figure 2.29: Sequence Diagram - Monitoring Usage

- An administrator logs into the platform to monitor usage.
- The administrator selects the metrics and reports they want to view, such as the number of users, the number of craftsman of each craft, and overall platform usage.
- The platform generates the metrics and reports based on the data stored in the database.
- The administrator reviews the metrics and reports to assess platform usage.
- The administrator takes action based on the metrics and reports, such as identifying

and addressing issues, optimizing the platform to improve performance, or making decisions about future development.

2.3 Conclusion

In this chapter, we have seen the system's design and how it works. We have also identified the actors in the system and the processes they perform. We also explained the techniques and algorithms used to build the system. In the next chapter we will present and analyze the results.

Chapter 3

Implementation and results

3.1 Introduction

After presenting and discussing the theoretical part of our project, we now come to the practical part. In this chapter, we first introduce general concepts about the different languages, tools and software that we will use to develop our platform. Second, we introduce the interfaces of the system and discuss the given results provided by our model.

3.2 Development tools and used platforms

- **Extension Pem (Privacy Enhanced Mail)**

  Applications like FileZilla server and OpenSSL both employ the Privacy-enhanced mail (PEM) format. An RSA key pair, an SSL certificate, or both can be found in a PEM file. It could have a password. The PEM file's contents are base-64 encoded, unlike other keystore formats that only support binary data [57].

- **Extension Pickle**

  A pickle file—also called a "pkl file" —is a file that stores a serialized version of a Python object. Pickling is the procedure for turning a Python object into a byte stream that can be saved in a file or sent over the internet. The act of "unpickling" the produced pickle file allows it to be reloaded into a Python application and transformed back into the original object [58].

- **Extension Json (JavaScript Object Notation)**

  the end of the file.json to show that the file's data is saved in the JSON format [59].

3.2.1 Used Platform

- **Visual studio code (VS Code)**

Visual Studio Code, developed by Microsoft, is a free source code editor that provides developers with a flexible and expandable environment for coding in multiple programming languages. Its user-friendly interface and customize features empower developers to enhance their coding experience. It is compatible with various platforms and seamlessly integrates with other software development tools, streamlining the development workflow [60].

- **Jupyter notebook**

It is an open source web application that can be accessed via a browser. And it works as a local server on the computer, used for programming in more than 40 programming languages, including: Python, Julia, Ruby and R... Jupyter notebook allows combining Markdown text with Python code, as the user interface in Jupyter notebook works in the form of separate cells that are Inside it is writing the code and executing it, and its results are displayed immediately, in addition to the inclusion of pictures and websites. [61] , [62] .

- **XAMPP**

It is an open source program for developing web applications, as its advantages are in the ease and speed of installation, while Xampp works on various operating systems windows, linux and MAC. Xampp was developed by Apache Friends. The Xampp program converts the computer into a local server (Localhost) where it is used to develop and test web applications written in languages such as : HTML , CSS , PHP and Java Servlet ... It can also use a database for java applications, and it can also be used to connect to the database for storage and retrieve data by executing MySQL queries [63], [64] .

- **phpMyAdmin**

It is a free web application written in PHP. Its importance lies in the management of MySQL and MariaDB database systems, while it allows managing databases, tables, columns, relations, indexes and permissions... and all this is done through the user interface. It also provides dealing with MySQL queries. directly [65], [66] .

- **MySQL**

It is an open source relational database management system developed by Oracle [67].

MySQL is used to store and manage data, where MySQL is a widely popular database system in web applications and local applications that need an effective database. The reason for its spread and popularity among web applications is due to its flexibility, speed, and good performance. MySQL is based on the Query (SQL) provides powerful tools for creating and organizing database queries.

- **HTML**

(Hypertext Markup Language) is a standardized markup language used for creating and structuring web pages. It defines the elements and their layout within a web document, allowing for the display of text, images, multimedia, and interactive elements on the World Wide Web [68].

- **CSS**

short for Cascading Style Sheets, is a styling language used in web development to describe the presentation and layout of HTML documents. It allows developers to define how elements on a webpage should be displayed, including aspects such as fonts, colors, margins, and positioning. By separating the content and structure of a webpage from its visual design, CSS provides greater flexibility and consistency in creating visually appealing and responsive websites [69].

- **JAVASCRIPT**

JavaScript is a high-level programming language primarily used for web development. It enables interactive and dynamic functionality on web pages, allowing developers to create responsive and engaging user experiences. JavaScript is often integrated with HTML and CSS to enhance the interactivity and behavior of websites [70].

- **JQUERY**

is a concise and powerful JavaScript library that simplifies HTML document traversal, event handling, and animation for rapid web development. It provides a wide range of functions and utilities, allowing developers to write less code while achieving more functionality. jQuery is known for its cross-browser compatibility and user-friendly syntax, making it a popular choice for enhancing the interactivity and responsiveness of web pages [71].

- **Ajax**

Ajax, an acronym for Asynchronous JavaScript and XML, is a method in web development that facilitates real-time, non-blocking interaction between a web browser and a server, eliminating the need for refreshing the entire page. This technology empowers the development of web applications that are engaging and swiftly respond to user actions [72].

- **BOOTSTRAP**

Bootstrap is a popular open-source front-end framework used for developing responsive and mobile-first websites and web applications. It provides a comprehensive set of HTML, CSS, and JavaScript components and utilities that facilitate the creation of visually appealing and user-friendly interfaces. Bootstrap offers a grid system, responsive breakpoints, pre-designed components, and customizable styles, enabling developers to efficiently build responsive and consistent web designs [73].

- **PHP**

originally coined as "Hypertext Preprocessor," is a server-side scripting language extensively employed in web development. It operates by being embedded within HTML code and executed on the server side, resulting in the generation of dynamic web pages. Rasmus Lerdorf introduced PHP in 1994, and it has undergone significant evolution since then, becoming a widely embraced language for constructing interactive and dynamic websites [74].

- **PHP Mailer** PHP Mailer is a widely-used PHP library that enables the automated sending of emails. It is an open-source solution known for its convenience and dependability, offering web applications a reliable method for transmitting email messages [74].

- **Openstack**

OpenStack is an open-source platform for building and managing private and public clouds using pooled virtual resources. The OpenStack platform's tools, known as "projects," handle the essential cloud computing services of computation, networking, storage, identity, and image services.

3.2.2 Back-end Treatments

- **Google Colab**

Or (Colaboratory): is a cloud-based platform that allows users to write and execute codes entirely in the cloud, it provides access to powerful computing resources like GPU, TPU and RAM. We use it to train the models of the chatbot[75].

- **GoogleTrans**: A library provides a free online translation through Google.

- **Python**

It is a good and easy programming language, as it is object-oriented, easy to learn, and an ideal choice for entering the world of programming. It was created by a person named Guido van Rossum, and these are the most important libraries used in the project [76] :

- **TensorFlow and Keras**: One of the best AI library for numerical computution, developed by Google and uses for machine learning and deep learning, uses it to create a deep learning model, train it and predict [77].
- **Flask**: It is a web framework for building web applications and APIs using python, It can handle the server side functionality, we use it to make requests and responses from chatbot to the website [78]
- **Transformers**: It is an open-source library in Python. This library was developed by Hugging Face and offers a wide range of pre-trained models. We used this library to fine-tune some of these models, like bert and roberta and distilbert [79].
- **NLTK**: An open source library for natural language processing, provide a wide range of methods to work with texts like tokenization, stemming, and lemmetazing [80].
- **Matplotlib**: A library for generating plots to evaluate the performance of the model.
- **Datasets**: This library used for accessing to the dataset that we need to train the deep learning model on it in a single line of code, it uses to load the dataset very fast [81].
- **Datetime:** Timestamp formatting and interpretation are handled by the Python datetime package. This module provides a variety of functions, such as the ability to obtain the current time, determine the difference (or delta) between two timestamps, and transform popular timestamp formats into a date that can be read by humans [82].

- **Fernet:** Asymmetric encryption techniques like Fernet ensure that the message being protected cannot be altered or read without the key. Methods for best-practice cryptography are defined by Femet [83]

- **Threading:**Python has threading modules as well. The ability to generate and manage threads is provided by the threading module [84]

- **Socket:**This module helps us to create a socket, which leads to the creation of two-way communication between two devices and the transmission of data..[85]

- **Fastecdsa:** Python's fastecdsa function divides by 0. The multiprecision arithmetic library GNU MP, which is supported by the Python extension module fastecdsa, was written in C. The ECDSA signature technique is implemented, and it also offers adaptable EC arithmetic with affine coordinates. The module supports a number of standardized curves with set parameters, generic Ew curves, as well as Ew versions of contemporary EE and ET curves like Curve25519 and Curve448..[86]

- **Pyrebase:** All of the Firebase SDKs have an unauthorized Python wrapper called Pyrebase [87].

- **Hashlip:** The module hashlib The most popular cryptographic hash function is implemented as an interface by the module hashlib. Additionally, the Python hashlib includes properties and methods to assist programmers in creating and managing message digests..[88]

- **Pickle:** Without the need to convert Python objects into strings in order to send or save them, it is possible to keep complex and unreadable data as it is.[58]

- **Pandas** :

  It is an open source library licensed by BSD to process and analyze data in the Python language, such as tables, where Pandas provides a highly flexible data structure called "Dataframe", which is known to contain tables with rows and columns, where Pandas allows us to load data from various sources as CSV files, Excel, and databases and conduct operations on them, such as compilation, statistical analysis, etc. By converting unstructured data into usable data. [89], [90].

- **SQLAlchemy** :

  It is an open source Python and Object Relational Mapper toolkit that allows application developers to interact with relational databases through the Python language. SQLAlchemy allows interacting with databases by writing queries in Python. Instead of writing queries in SQL, SQLAlchemy allows defining tables, relationships, and operations using Python code that is simple and clear. In other words, the SQLAlchemy library is powerful and flexible in dealing with databases, using Python queries, [91].

- **Numpy** :

  It is the basic package for scientific computing in Python that facilitates working with numerical data. Numpy aims to manipulate matrices or multidimensional arrays in addition to the mathematical operations applied to these matrices. Where numpy is an essential tool for various libraries in the Python environment, such as scikit-learn, SciPy and Pandas, [92].

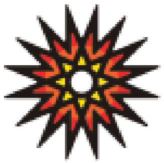
- **SciKit-learn** :

  It is an open source Python library that is used in the field of machine learning. The goal of Scikit-learn is to provide algorithms that facilitate the development and implementation of machine learning models. As this library is based on numpy and SciPy [93].

- **Openstack Services** :

  - **Nova (Compute)** :

Nova is an OpenStack project that allows you to provision compute instances (also known as virtual servers). Nova can create virtual machines, baremetal servers (through ironic), and has limited support for system containers. To deliver that service, Nova runs as a group of daemons on top of existing Linux servers [94].



- **Keystone (Identity)** :

Keystone is an OpenStack service that implements OpenStack's Identity API to offer API client authentication, service discovery, and distributed multi-tenant permission. It is compatible with LDAP, OAuth, OpenID Connect, SAML, and SQL [95].

- **Cinder (Block Strorage)** :

Cinder is an OpenStack Block Storage service. It virtualizes the management of block storage devices and enables end users with a self-service API to request and consume those resources without requiring any knowledge of where their storage is placed or what type of device is being used. This is accomplished by utilizing either a reference implementation (LVM) or plugin drivers for alternative storage [96].

- **Neutron (Networking)** :

OpenStack Neutron is a software-defined networking (SDN) networking project that focuses on providing networking-as-a-service (NaaS) in virtual compute environments [97].

- **Placement** :

Placement is an OpenStack service that provides an HTTP API for tracking cloud resource inventories and usages in order to enable other services manage and distribute resources more effectively [98].

- **Swift (Object store)** :

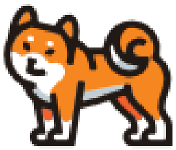
Swift is an object/blob storage that is extremely available, distributed, and finally consistent. Swift allows businesses to store large amounts of data in an efficient, secure, and cost-effective manner. It is designed for scalability and is optimized for durability, availability, and concurrency throughout the full data set. Swift is great for storing unstructured data that grows indefinitely [99].

– **Glance (Image)** :

Discovering, registering, and retrieving virtual machine images are all part of the Glance image services. Glance provides a RESTful API for accessing VM image metadata as well as retrieving the actual image. Glance-provided VM images can be stored in a variety of locations, ranging from simple filesystems to object-storage systems such as the OpenStack Swift project [100].

– **Horizon (Dashboard)** :

Horizon is the canonical OpenStack dashboard solution, which is extensible and provides a web-based user interface to OpenStack services [101].

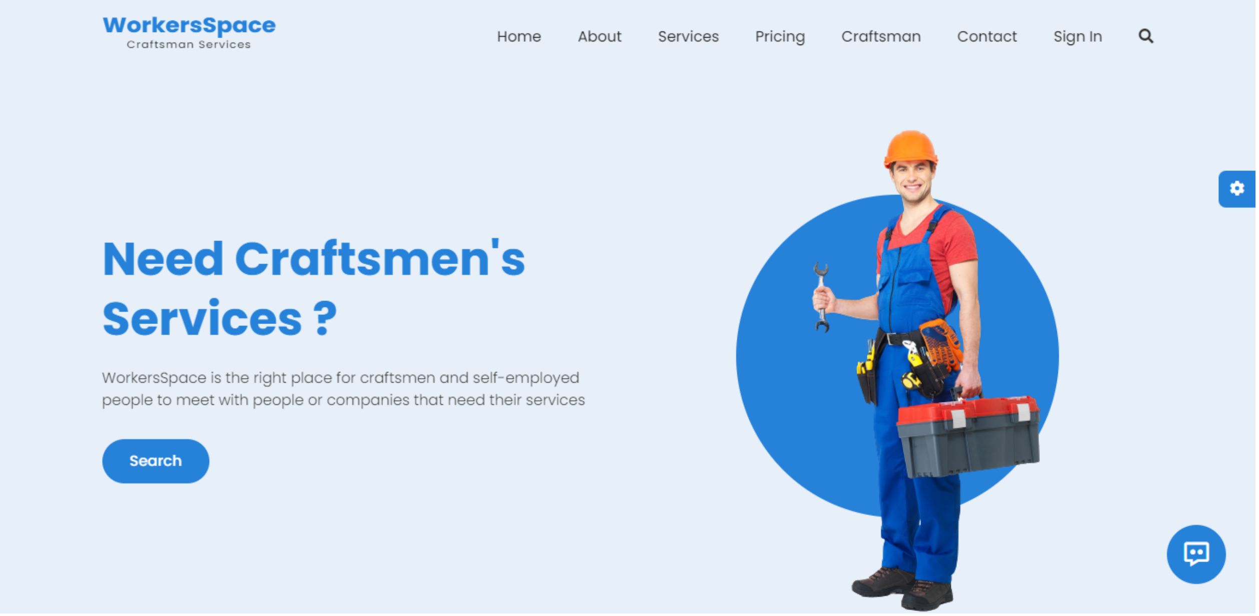


3.3 System Interfaces and examples

In this section we show the systems interfaces and examples of the all application and how it work.

3.3.1 Homepage

Our platform's main page showcases comprehensive information about our services, highlighting our diverse range of offerings. Additionally, we feature a curated list of highly recommended craftsmen, ensuring exceptional quality and expertise for our users.



WorkersSpace
Craftsman Services

Need Craftsmen's Services ?

WorkersSpace is the right place for craftsmen and self-employed people to meet with people or companies that need their services

Figure 3.1: Homepage - WorkersSpace

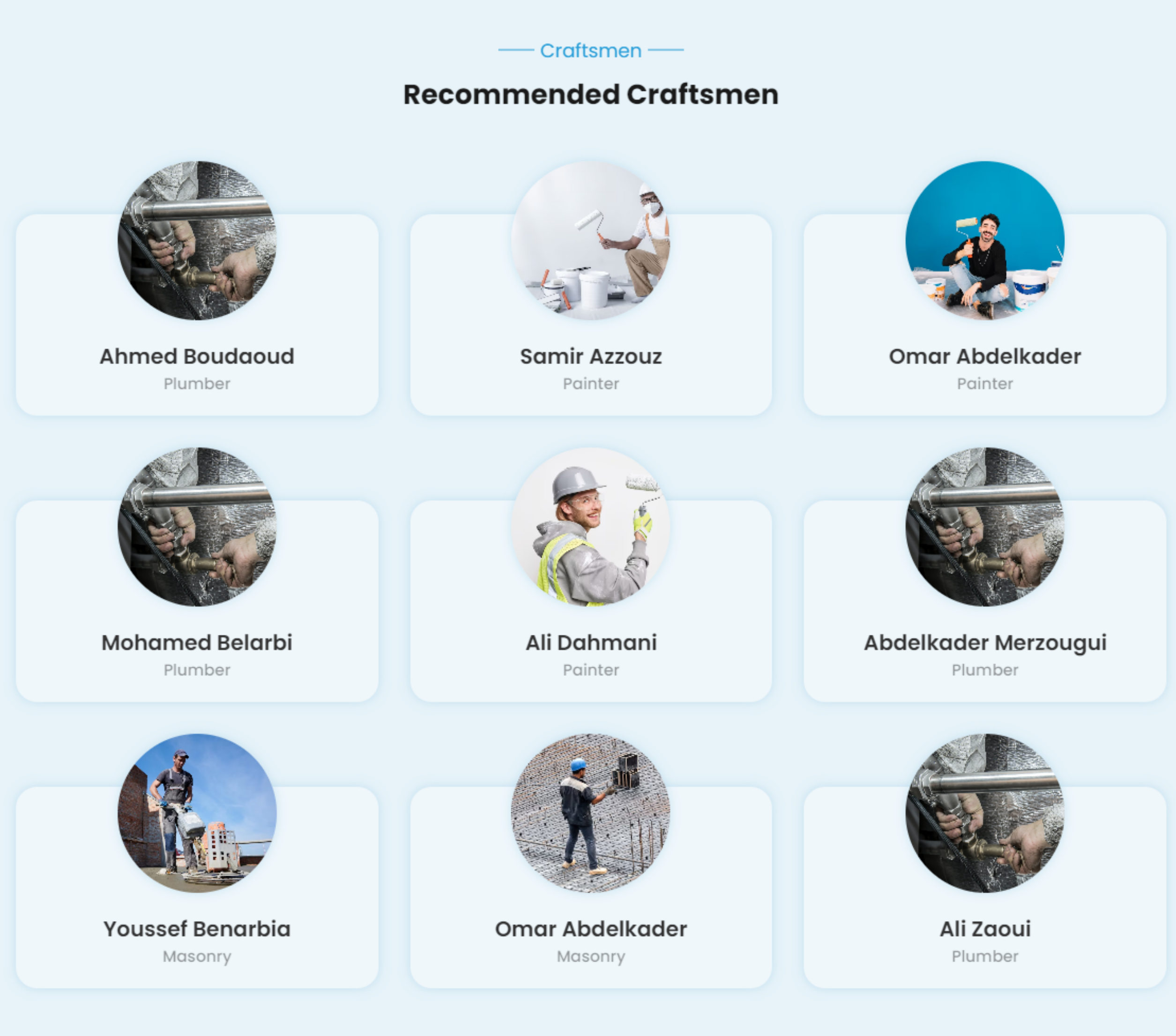
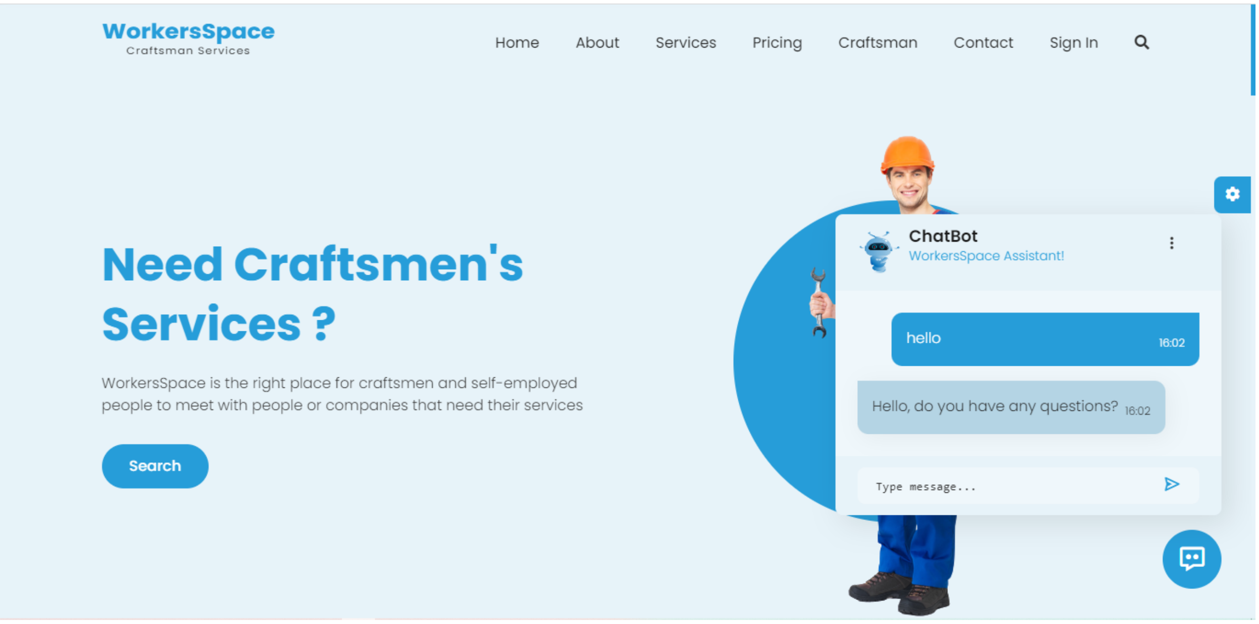


Figure 3.2: Recommended Craftsman - WorkersSpace

3.3.2 Chatbot

Our platform incorporates a chatbot feature that provides instant and personalized assistance to users. The chatbot utilizes artificial intelligence to understand user queries, offer relevant information, and guide users through various processes, ensuring a seamless and efficient user experience while addressing their needs promptly.



Need Craftsmen's Services ?

WorkersSpace is the right place for craftsmen and self-employed people to meet with people or companies that need their services

hello 16:02

Hello, do you have any questions? 16:02

Figure 3.3: Chatbot - WorkersSpace

3.3.3 Services

Our platform offers a wide range of services that cater to different needs and preferences. Such as plumbing, painting, etc. We provide a seamless experience, connecting users with skilled craftsmen to meet their specific requirements.

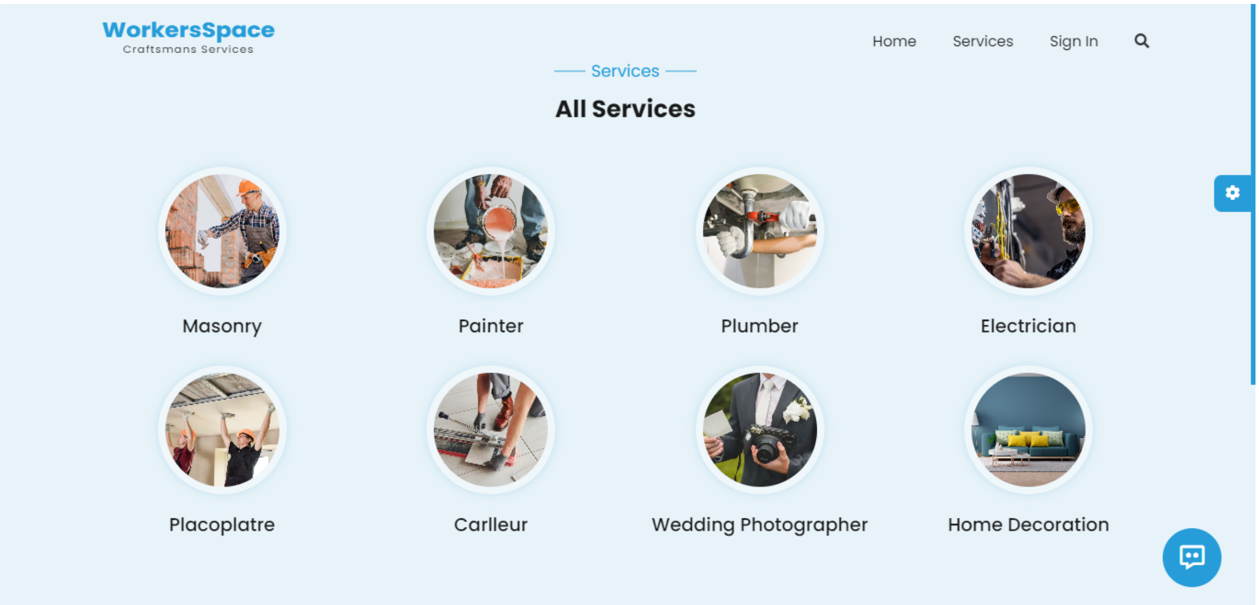
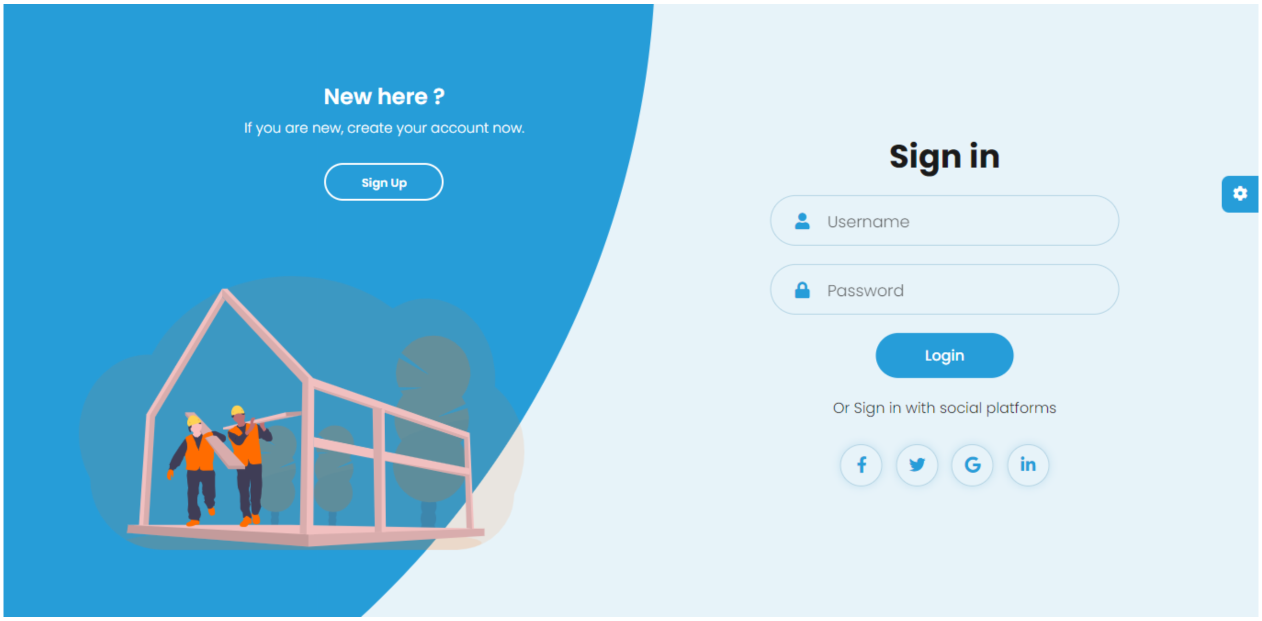


Figure 3.4: Services - WorkersSpace

3.3.4 Signin and Signup

Signing up on our platform is a breeze with a simple and user-friendly registration process. Once signed in, users gain access to personalized profiles, enabling them to easily manage their preferences, track orders, and engage with our vibrant community of craftsmen and customers.



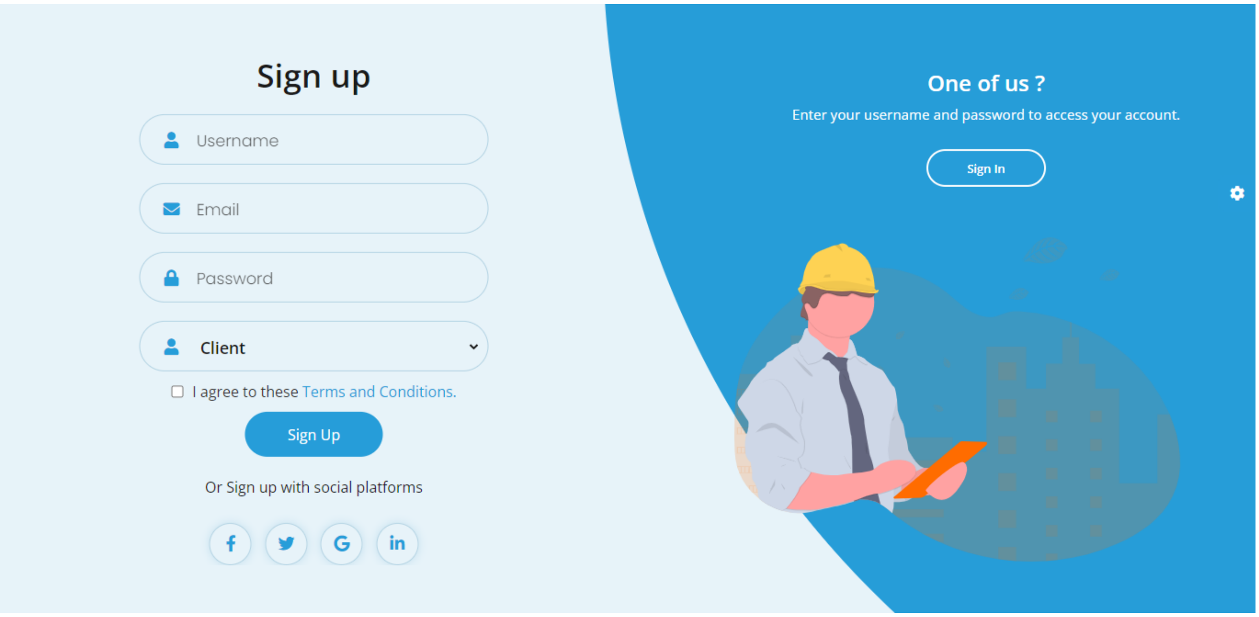
New here ?

If you are new, create your account now.

Sign in

Or Sign in with social platforms

Figure 3.5: Signin - WorkersSpace



Sign up

Client

□ I agree to these Terms and Conditions.

Or Sign up with social platforms

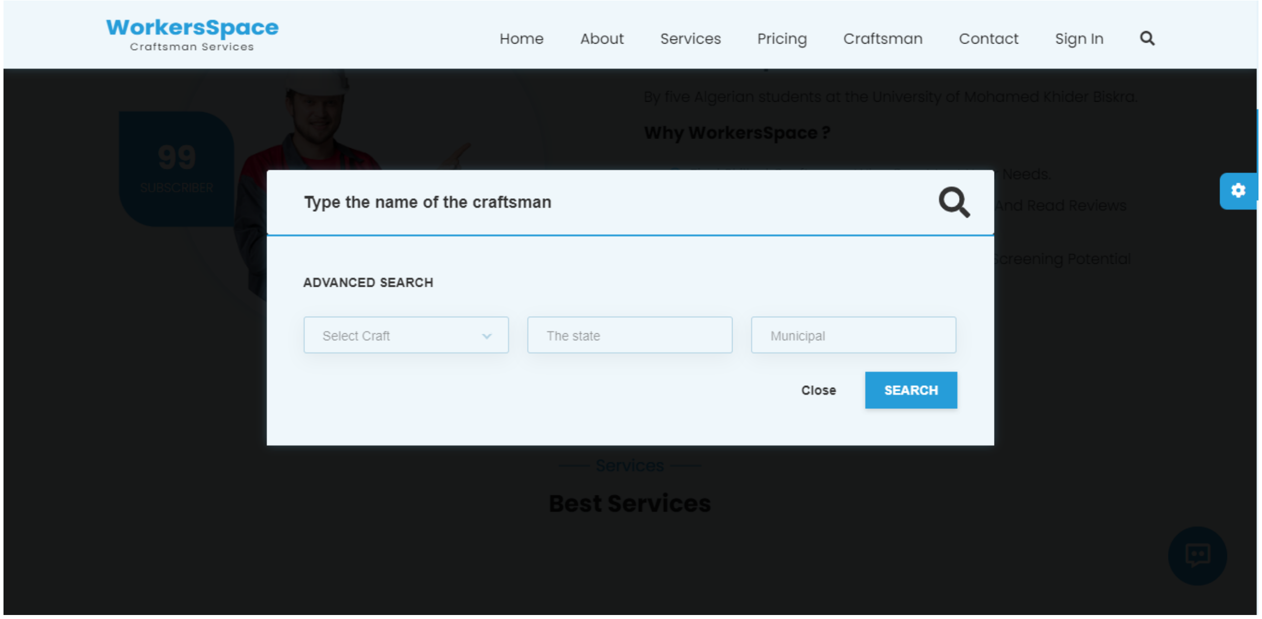
One of us ?

Enter your username and password to access your account.

Figure 3.6: Signup - WorkersSpace

3.3.5 Search

Our platform offers a robust search function that allows users to easily find craftsmen based on specific criteria such as craft type and location. Whether searching for a skilled woodworker in a particular state or exploring municipal artisans, our search feature ensures efficient and tailored results for every user's needs.

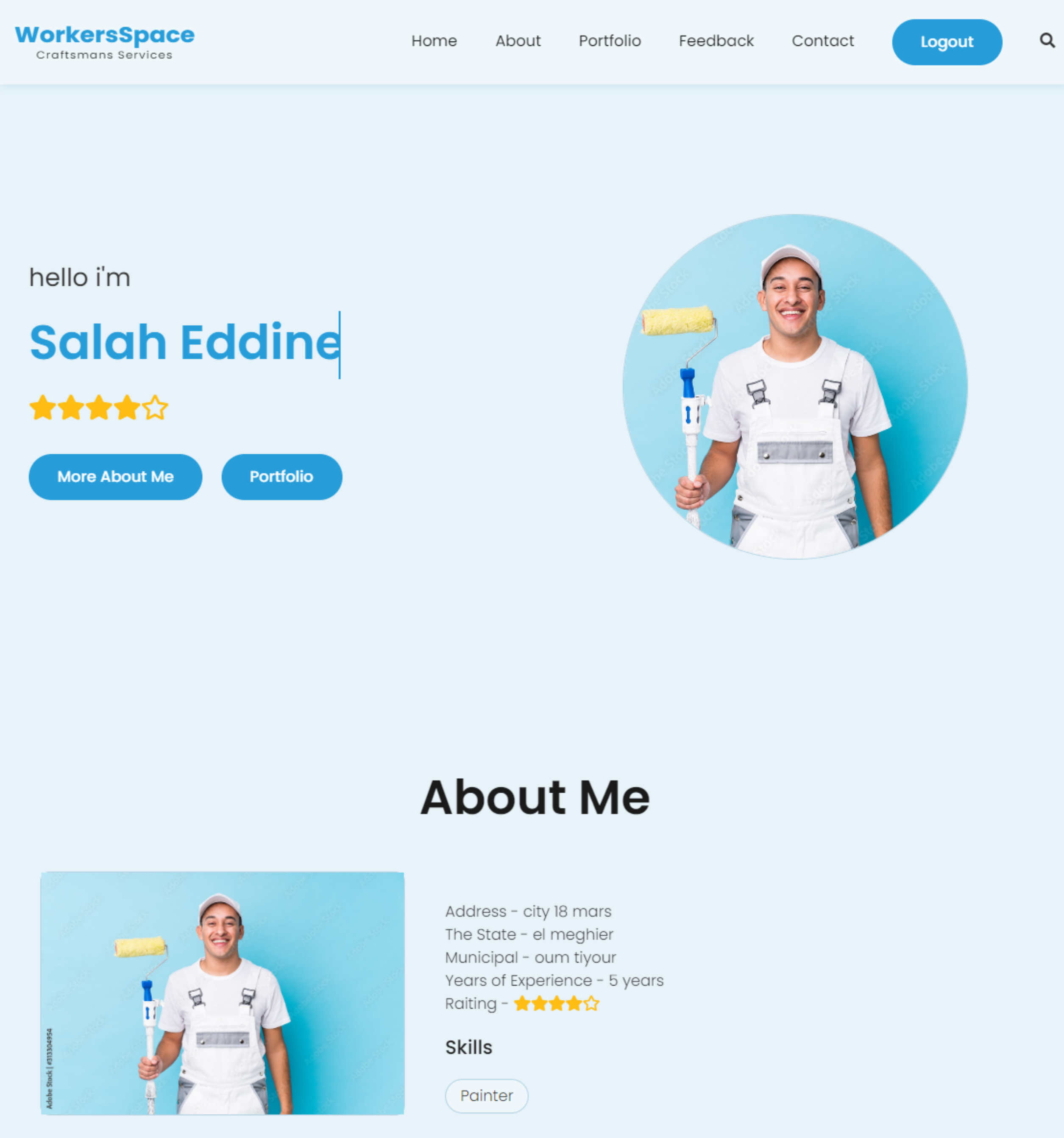


By five Algerian students at the University of Mohamed Khider Biskra.

Figure 3.7: Search - WorkersSpace

3.3.6 Profile Craftsman

The profile of each craftsman on our platform provides comprehensive details about their expertise, including ratings, reviews, and an extensive portfolio of their work. Users can make informed decisions by exploring these profiles and directly hiring the craftsmen that best match their requirements, ensuring a seamless and satisfactory experience.



hello i'm

Salah Eddine

★★★★☆

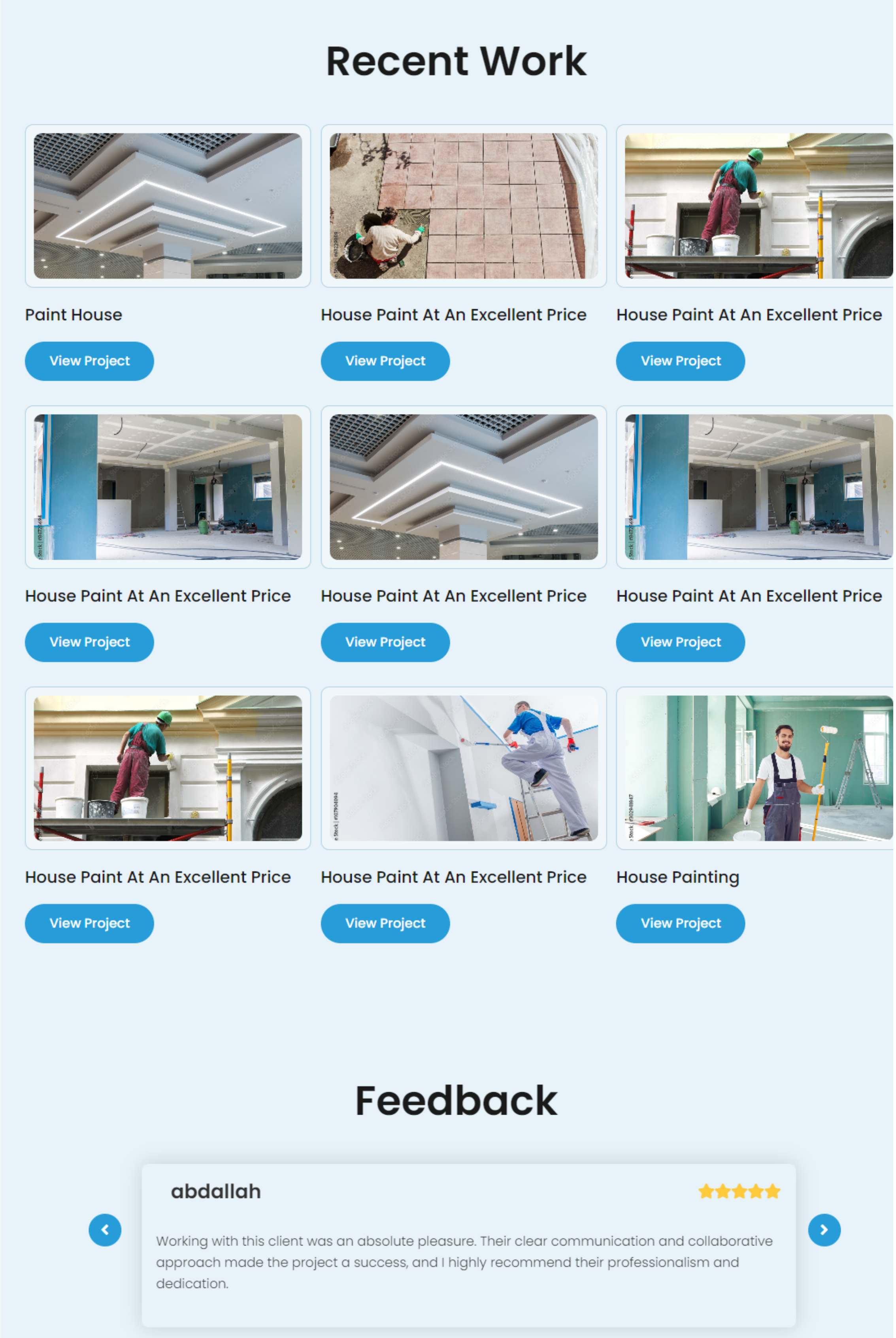
About Me

Address - city 18 mars
The State - el meghier
Municipal - oum tiyour
Years of Experience - 5 years
Raiting - ★★★★☆

Skills

Painter

Figure 3.8: Profile craftsman - WorkersSpace



Recent Work

Paint House

House Paint At An Excellent Price

House Paint At An Excellent Price

House Paint At An Excellent Price

House Paint At An Excellent Price

House Paint At An Excellent Price

House Paint At An Excellent Price

House Paint At An Excellent Price

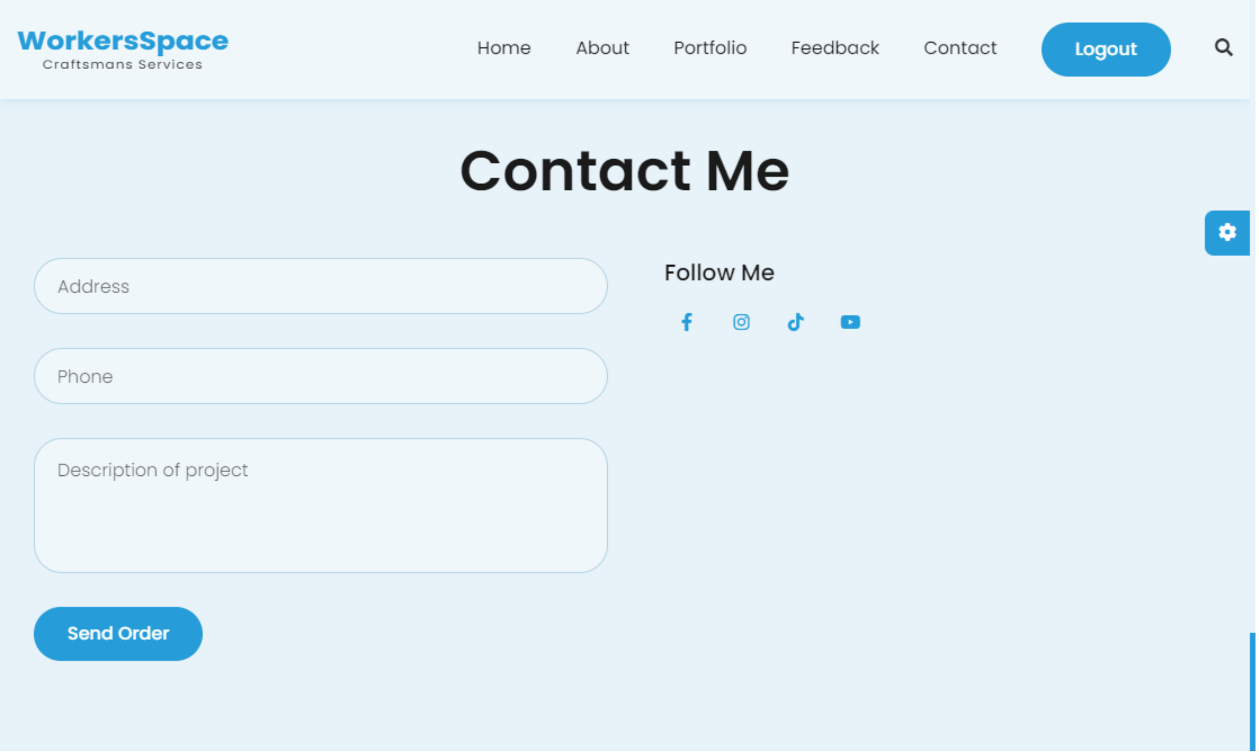
House Painting

Feedback

abdallah ★★★★★

Working with this client was an absolute pleasure. Their clear communication and collaborative approach made the project a success, and I highly recommend their professionalism and dedication.

Figure 3.9: Portfolio craftsman - WorkersSpace



Contact Me

Address

Phone

Description of project

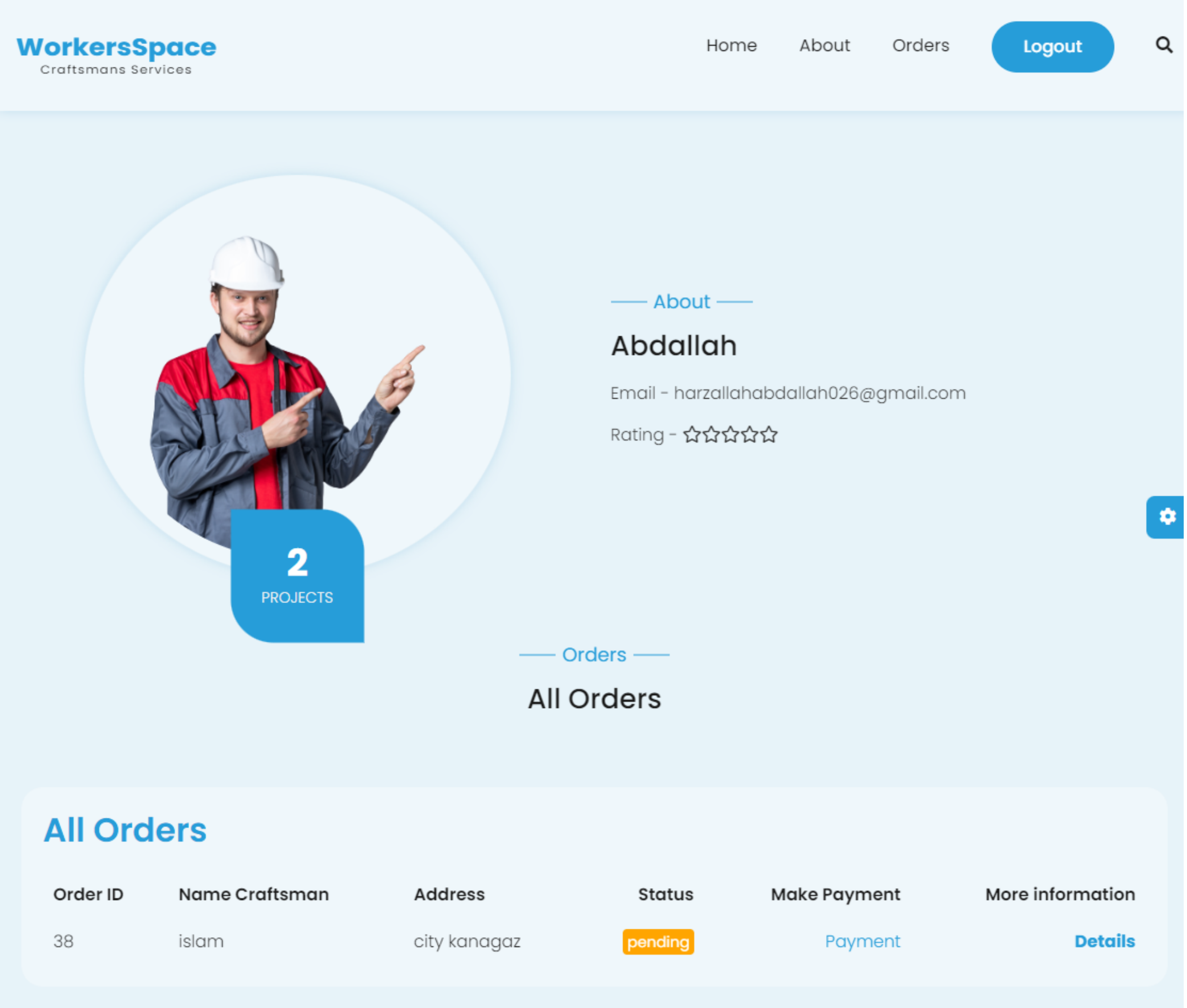
Send Order

Follow Me

Figure 3.10: Hire Craftsman - WorkersSpace

3.3.7 Client Profile

A customer profile on our platform provides a comprehensive overview of each user's interaction with craftsmen, including ratings and a detailed history of orders sent to specific craftsmen. This allows clients to track their involvement, make informed decisions based on past experiences, and ensure smooth and satisfying collaborations with skilled artisans.



About

Abdallah

Email - harzallahabdallah026@gmail.com

Rating - ☆☆☆☆☆

Orders

All Orders

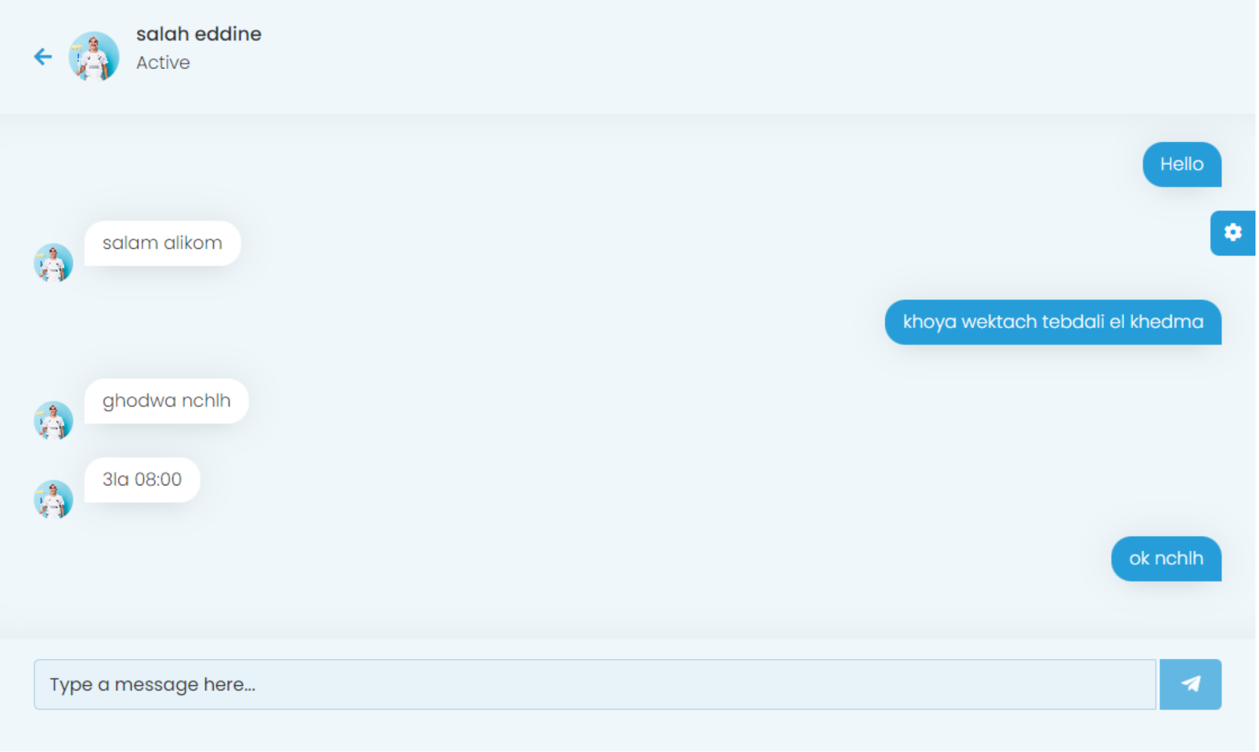
All Orders

Order ID	Name Craftsman	Address	Status	Make Payment	More information
38	islam	city kanagaz	pending	Payment	Details

Figure 3.11: Profile Client - WorkersSpace

3.3.8 Chats

Our platform features an integrated chat system that facilitates seamless communication between users and craftsmen. The chat function enables direct and real-time interactions, allowing users to discuss project details, clarify requirements, and ensure smooth collaboration, enhancing the overall user experience on our platform.



salah eddine
Active

Hello

salam alikom

khoya wektach tebdali el khedma

ghodwa nchlh

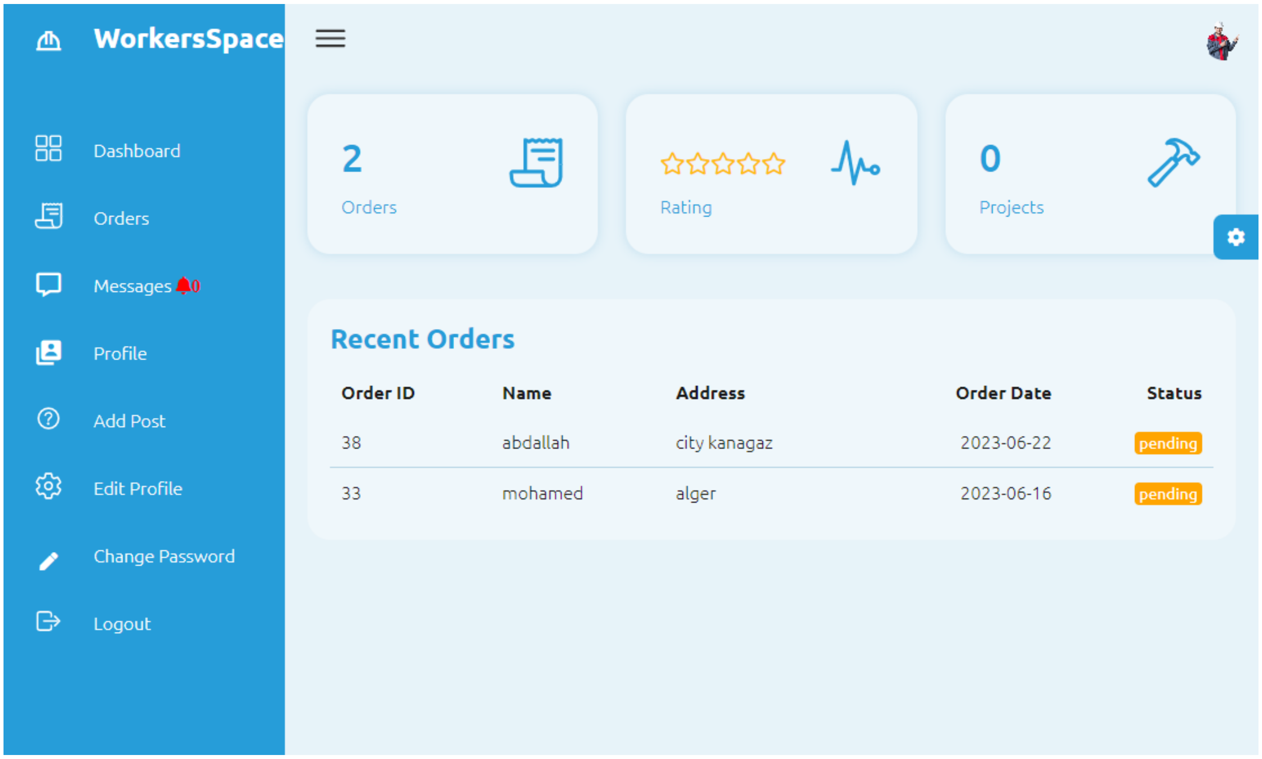
3la 08:00

ok nchlh

Figure 3.12: Chats - WorkersSpace

3.3.9 Craftsman Dashboard

The Craftsman Dashboard on our platform provides a comprehensive overview of a craftsman's performance, including ratings and a detailed record of orders received. Craftsmen can track their success, manage their workflow efficiently, and leverage client feedback to continually improve their services, all from a centralized and user-friendly dashboard interface.



2 Orders

Rating

0 Projects

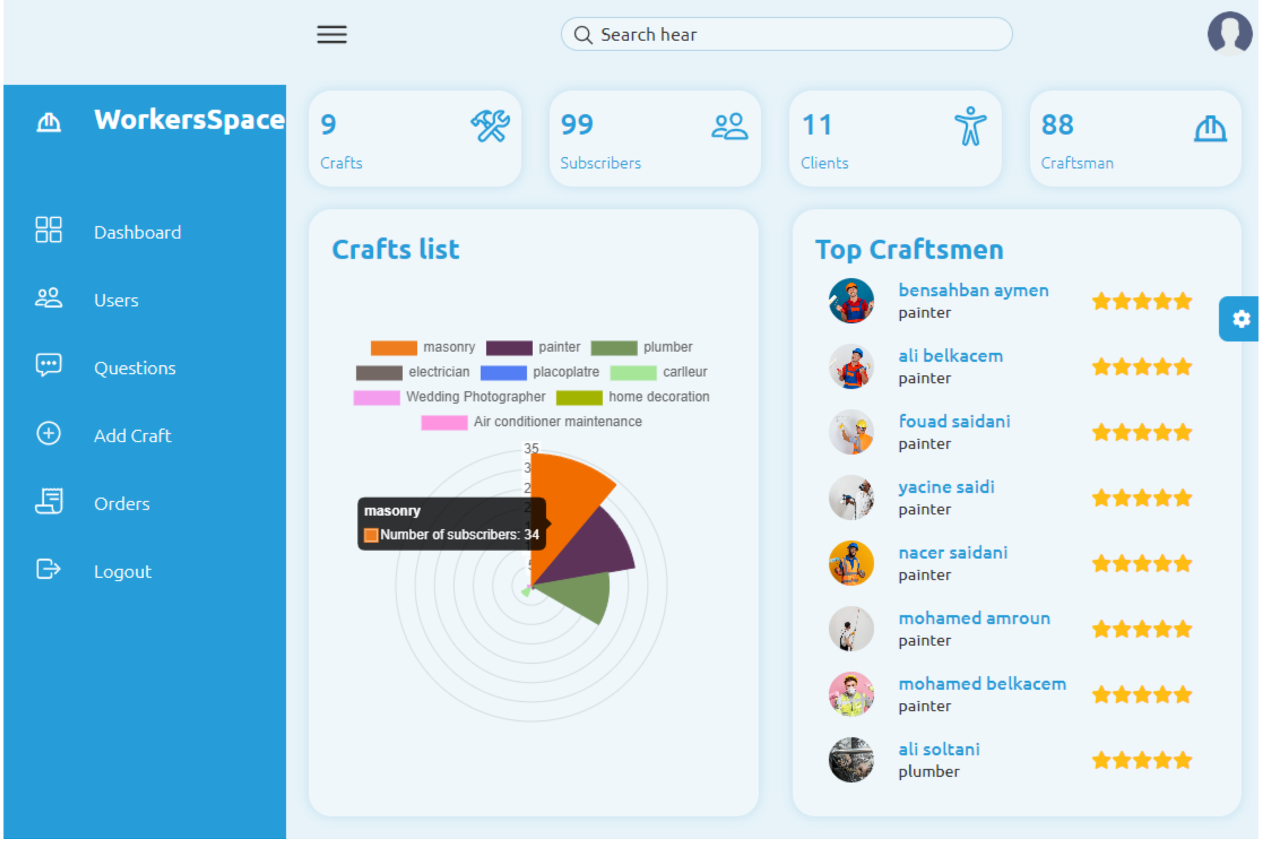
Recent Orders

Order ID	Name	Address	Order Date	Status
38	abdallah	city kanagaz	2023-06-22	pending
33	mohamed	alger	2023-06-16	pending

Figure 3.13: Craftsman Dashboard - WorkersSpace

3.3.10 Admin Dashboard

The Admin Dashboard on our platform offers a comprehensive view of all users, providing vital information such as ratings, transaction numbers, and numbers for subscribers, clients, craftsmen, and various crafts. Administrators can efficiently manage and analyze data, monitor platform performance, and make informed decisions to ensure a smooth and successful operation of the platform.



9 Crafts

99 Subscribers

11 Clients

88 Craftsman

Top Craftsmen

bensahban aymen painter	★★★★★
ali belkacem painter	★★★★★
fouad saidani painter	★★★★★
yacine saidi painter	★★★★★
nacer saidani painter	★★★★★
mohamed amroun painter	★★★★★
mohamed belkacem painter	★★★★★
ali soltani plumber	★★★★★

Figure 3.14: Admin Dashboard - WorkersSpace

3.4 Obtained results and discussion

In this section, we present the results and discuss them for the parts that have the necessary results.

3.4.1 Chatbot

Here we discuss and obtain the results of the chatbot.

- **Training phase**: We show in this part the training process for the two models that we build to make our chatbot.

  Intent classification model: This model is responsible to predict the intention, we need this intention for the question answering model. For the intent classification model based on mlp architecture, we set the training for 300 epochs and we define metrics of evaluation and loss function and the optimizer that we choose, as we can see in the training code 3.4.1

```
#build the architecture
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation=’relu’))
model.add(Dense(64, activation=’relu’))
model.add(Dense(32, activation=’relu’))
model.add(Dense(len(train_y[0]), activation=’softmax’))
# Compile model.
sgd = SGD(lr=0.001)
model.compile(loss=’categorical_crossentropy’, optimizer=sgd, metrics=[’accuracy’])
#fit model
history = model.fit(train_x, train_y,validation_data=(val_x,val_y), epochs=300, batch_size=8)
model.save(’model_intent.h5’, history)
```

And the values of the evaluation metrics saved every each epoch in history variable to plot the values using matplotlib as a graphs. After the training is finished we obtain a good results for this model as we can see in the figure of evaluation plots for the training and validation during the training 3.15.

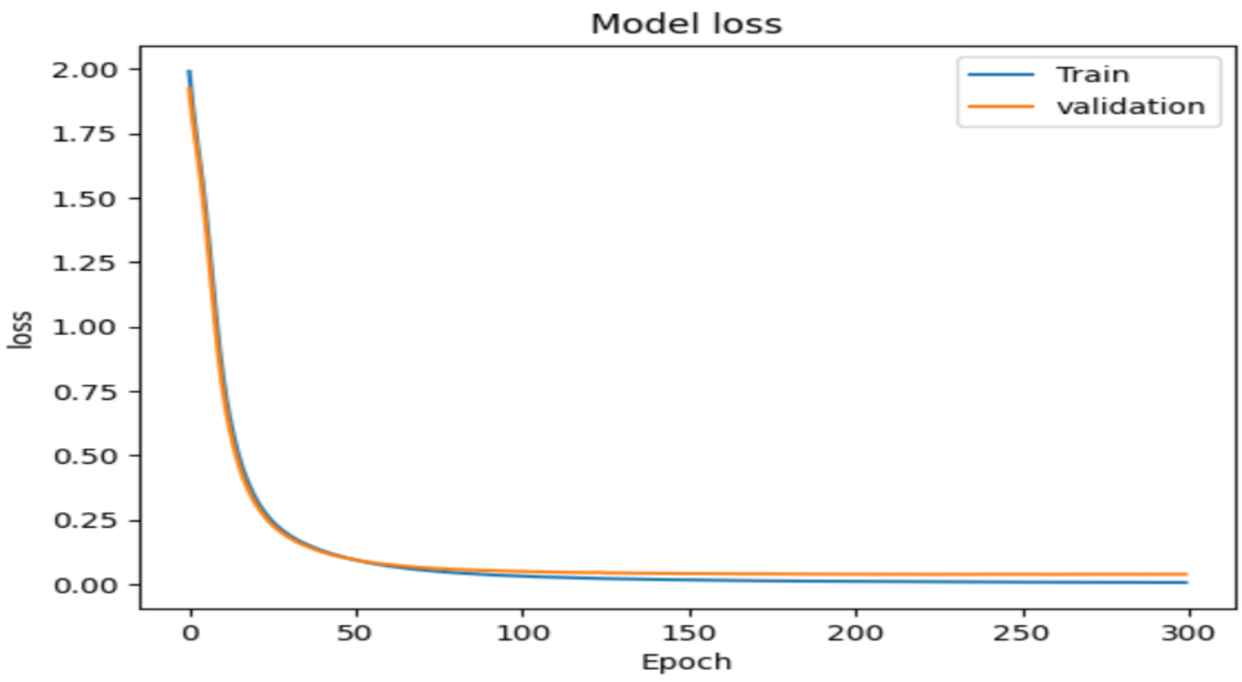


Figure 3.15: MLP Evaluation loss

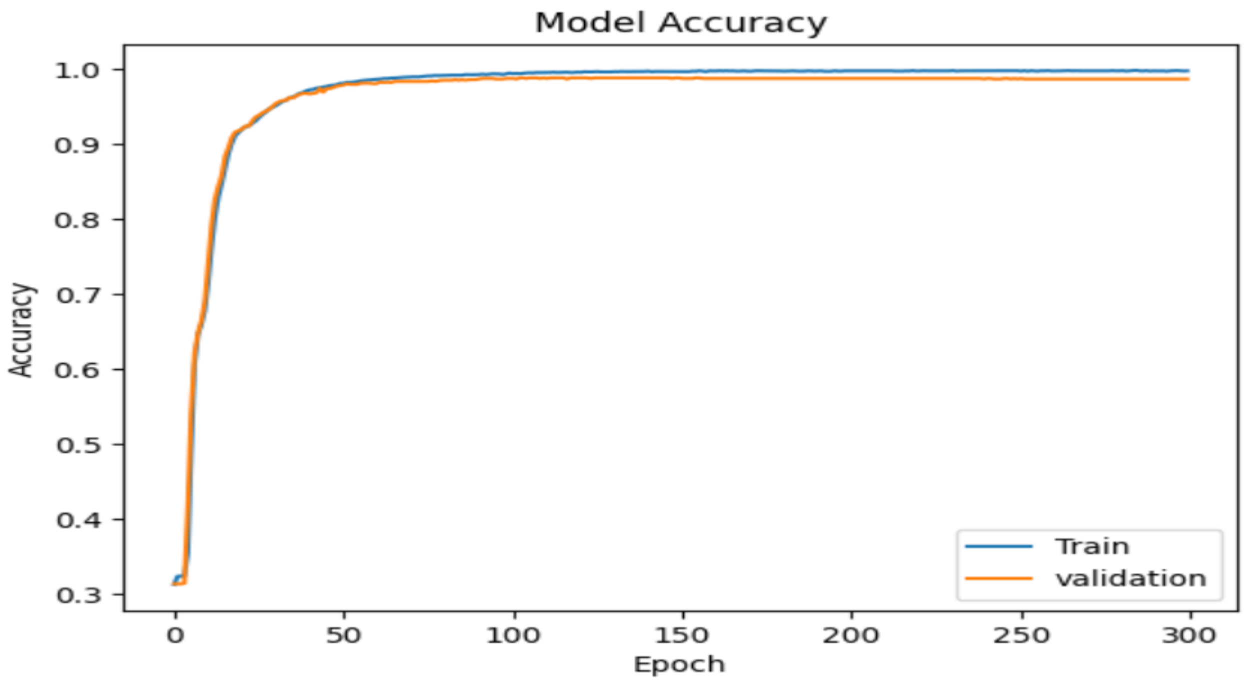


Figure 3.16: MLP Evaluation accuracy

Question answering: The second model which is based on transformer architecture this model take two inputs the intention and the question, it gives us a good results on dev part of squad dataset. We first finetune the pretrained model (finetune more than three derived models of bert) on a squad dataset. As we can see in training code3.4.1:

```
# must log into hugging face account
from huggingface_hub import notebook_login
notebook_login()
squad_set = load_dataset("squad")
#load the tokenizer
tokenizer = AutoTokenizer.from_pretrained('bert-base-uncased')
#After the preprocessing part
#load the pretrained model
model = TFAutoModelForQuestionAnswering.from_pretrained('bert-base-uncased')
# Compile model.
model.compile(optimizer=optimizer)
#using callback to store results after each epoch
callback = PushToHubCallback(output_dir="bert-for-squad", tokenizer=tokenizer)
#fit model
model.fit(train_set,callbacks=[callback], epochs=epochs)
model.save_pretrained('bert-for-squad')
```

And after the training is finished we get these results as we can see in the next table:

Evaluation metrics/ Derivative models	**Accuracy (exact match)**	**F1 Score**
Bert base (Uncased)	80.75	88.20
Bert base (Cased)	80.86	88.40
Roberta base	85.79	92.22
Distilbert base	76.66	85.25
Albert base	82.70	89.88

Table 3.1: Evaluation question answering model on squad dataset

RoBERTa: Is a derived model of bert trained on a larger corpus and for a longer time than the original bert.

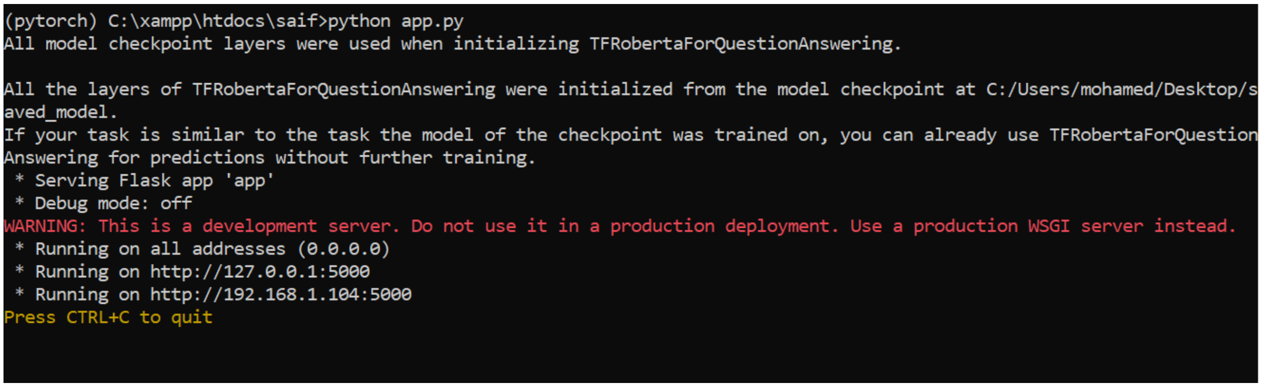
Distilbert: Is faster and smaller than original bert, has less layers.

Albert: Is another derived model of bert focuses on reducing the number of parameteres and model size.

- **Using the chatbot**: Here we describe how users can use the chatbot and the stages that the message of the users goes through in order to reach the answer and display it to the user.

1. **Activate the flask**

   We use flask to make requests and responses from the website into the server.

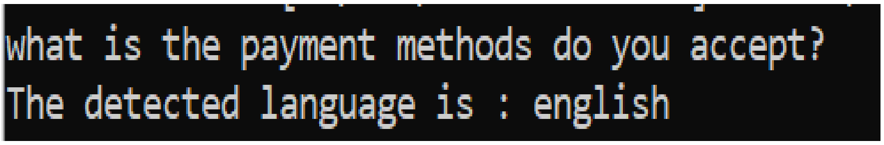


```
(pytorch) C:\xampp\htdocs\saif>python app.py
All model checkpoint layers were used when initializing TFRobertaForQuestionAnswering.

All the layers of TFRobertaForQuestionAnswering were initialized from the model checkpoint at C:/Users/mohamed/Desktop/s
aved_model.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFRobertaForQuestion
Answering for predictions without further training.
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.1.104:5000
Press CTRL+C to quit
```

Figure 3.17: Activate flask

2. **Submit the question and detect the language and translation**

   After the user send the message, we must detect the language to make the response with the same language.

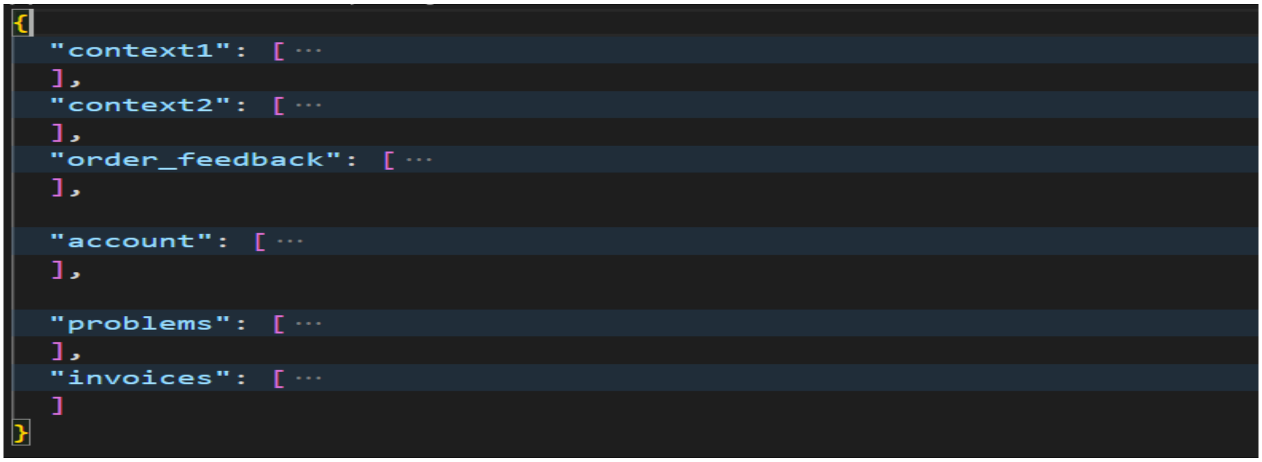


```
what is the payment methods do you accept?
The detected language is : english
```

Figure 3.18: Detect the language

3. **Predict the intention**

   Our database store in json file contain contexts(answers) each name of these contexts consider as an intention. By using the intent classification model we can predict the name of the context where the question answering model can find the answer because the bert based question answering model needs each context less than 512 token. This json file is our database in the form of intent (name of context) and answers (contexts).

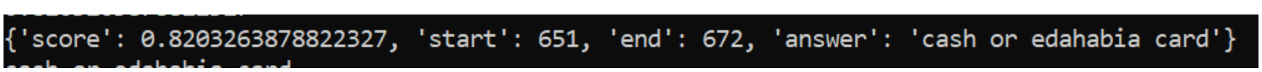


```
{
  "context1": [ ···
  ],
  "context2": [ ···
  ],
  "order_feedback": [ ···
  ],

  "account": [ ···
  ],

  "problems": [ ···
  ],
  "invoices": [ ···
  ]
}
```

Figure 3.19: json file

4. **Extract the answer**

   By using the question answering model pipline we can extract the answer as we can see the model predict the start and the end of the answer inside the context.

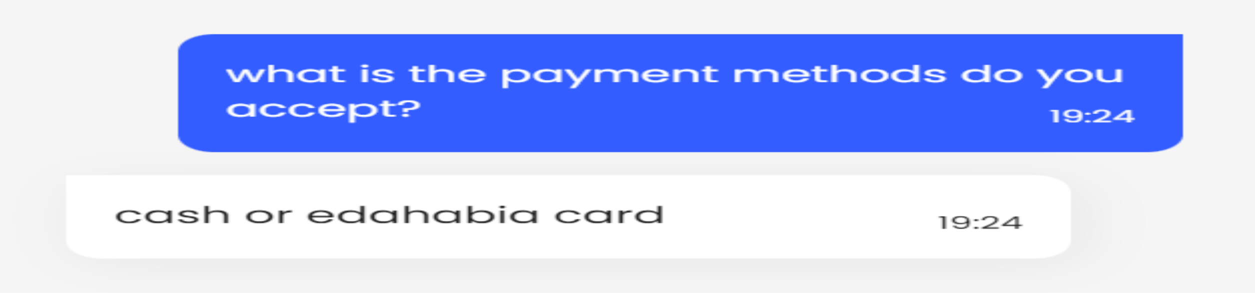


```
{'score': 0.8203263878822327, 'start': 651, 'end': 672, 'answer': 'cash or edahabia card'}
```

Figure 3.20: Predict start and end of the answer

5. **Display the response**

   Finally, display the response to the user in the chatbot interface in the website.



what is the payment methods do you accept? 19:24

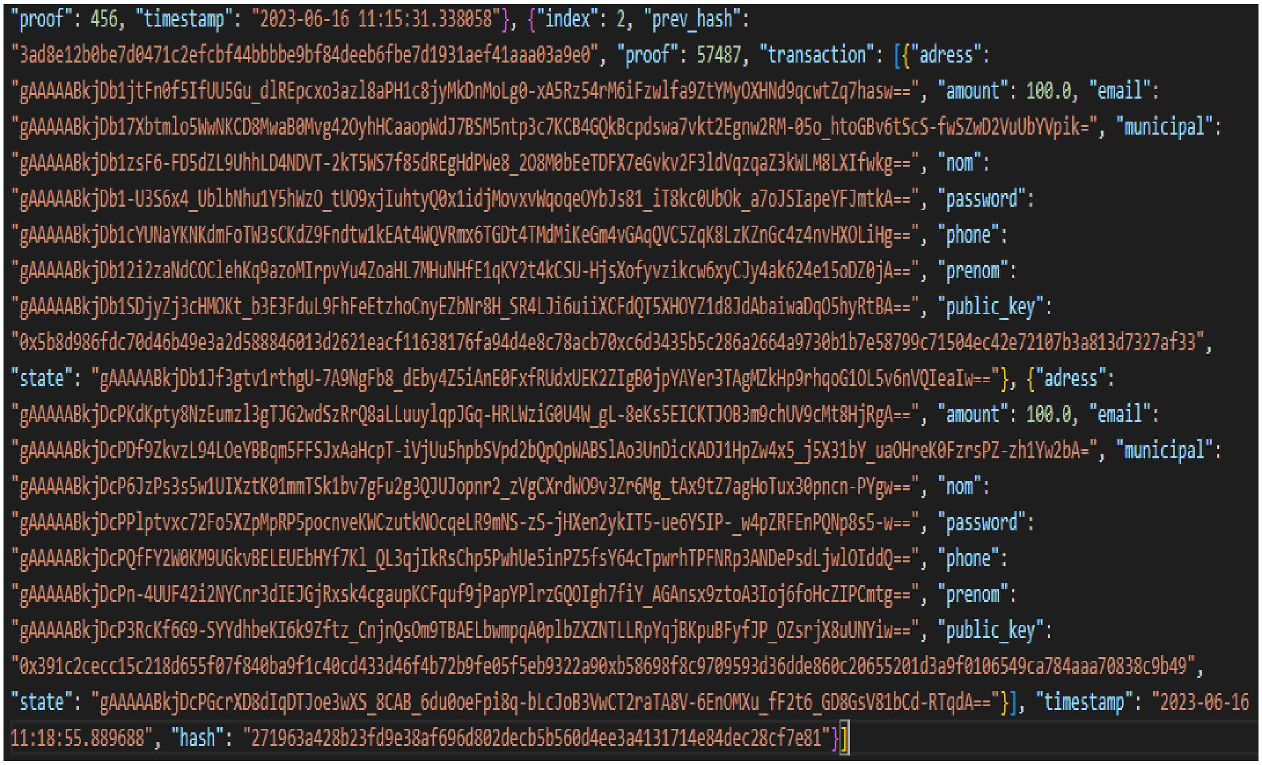
cash or edahabia card 19:24

Figure 3.21: Display the answer

3.4.2 Data security

At this stage, several prominent results have been achieved in the field of blockchain technology. Here are some notable advancements

- **Form the block**: The following figure represents how the block is stored at the level of the JSON block file, as it can contain information about the registered person, sent amount, or demand.

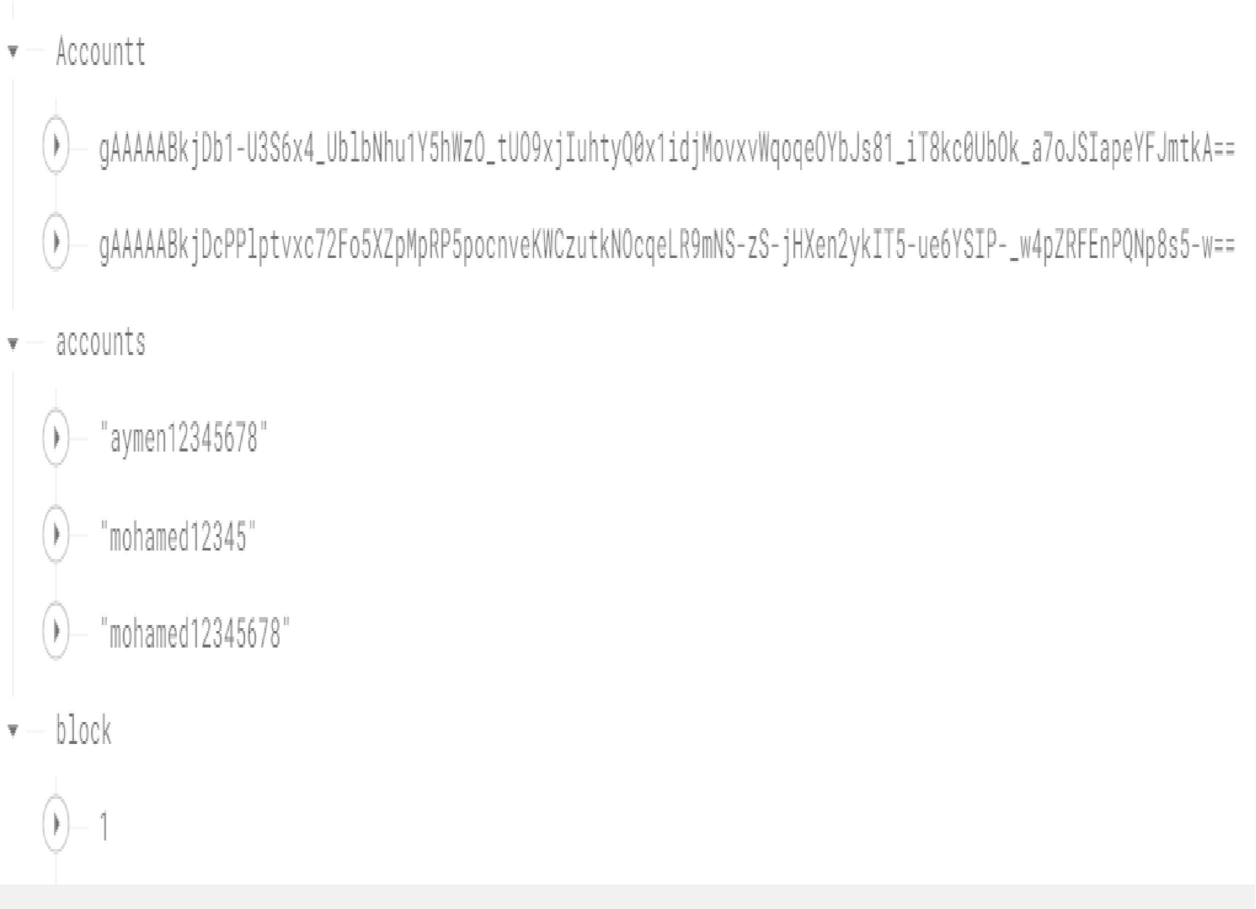


```
"proof": 456, "timestamp": "2023-06-16 11:15:31.338058"}, {"index": 2, "prev_hash":
"3ad8e12b0be7d0471c2efcbf44bbbbe9bf84deeb6fbe7d1931aef41aaa03a9e0", "proof": 57487, "transaction": [{"adress":
"gAAAAABkjDb1jtFn0f5IfUU5Gu_dlREpcxo3azl8aPH1c8jyMkDnMoLg0-xA5Rz54rM6iFzwlfa9ZtYMyOXHNd9qcwtZq7hasw==", "amount": 100.0, "email":
"gAAAAABkjDb17Xbtmlo5WwNKCD8MwaB0Mvg42OyhHCaaopWdJ7BSM5ntp3c7KCB4GQkBcpdswa7vkt2Egnw2RM-05o_htoGBv6tScS-fwSZwD2VuUbYVpik=", "municipal":
"gAAAAABkjDb1zsF6-FD5dZL9UhhLD4NDVT-2kT5WS7f85dREgHdPWe8_2O8M0bEeTDFX7eGvkv2F3ldVqzqaZ3kWLM8LXIfwkg==", "nom":
"gAAAAABkjDb1-U3S6x4_UblbNhu1Y5hWzO_tUO9xjIuhtyQ0x1idjMovxvWqoqeOYbJs81_iT8kc0UbOk_a7oJSIapeYFJmtkA==", "password":
"gAAAAABkjDb1cYUNaYKNKdmFoTW3sCKdZ9Fndtw1kEAt4WQVRmx6TGDt4TMdMiKeGm4vGAqQVC5ZqK8LzKZnGc4z4nvHXOLiHg==", "phone":
"gAAAAABkjDb12i2zaNdCOClehKq9azoMIrpvYu4ZoaHL7MHuNHfE1qKY2t4kCSU-HjsXofyvzikcw6xyCJy4ak624e15oDZ0jA==", "prenom":
"gAAAAABkjDb1SDjyZj3cHMOKt_b3E3FduL9FhFeEtzhoCnyEZbNr8H_SR4LJi6uiiXCFdQT5XHOYZ1d8JdAbaiwaDqO5hyRtBA==", "public_key":
"0x5b8d986fdc70d46b49e3a2d588846013d2621eacf11638176fa94d4e8c78acb70xc6d3435b5c286a2664a9730b1b7e58799c71504ec42e72107b3a813d7327af33",
"state": "gAAAAABkjDb1Jf3gtv1rthgU-7A9NgFb8_dEby4Z5iAnE0FxfRUdxUEK2ZIgB0jpYAYer3TAgMZkHp9rhqoG1OL5v6nVQIeaIw=="}, {"adress":
"gAAAAABkjDcPKdKpty8NzEumzl3gTJG2wdSzRrQ8aLLuuylqpJGq-HRLWziG0U4W_gL-8eKs5EICKTJOB3m9chUV9cMt8HjRgA==", "amount": 100.0, "email":
"gAAAAABkjDcPDf9ZkvzL94LOeYBBqm5FFSJxAaHcpT-iVjUu5hpbSVpd2bQpQpWABSlAo3UnDicKADJ1HpZw4x5_j5X31bY_uaOHreK0FzrsPZ-zh1Yw2bA=", "municipal":
"gAAAAABkjDcP6JzPs3s5w1UIXztK01mmTSk1bv7gFu2g3QJUJopnr2_zVgCXrdWO9v3Zr6Mg_tAx9tZ7agHoTux30pncn-PYgw==", "nom":
"gAAAAABkjDcPPlptvxc72Fo5XZpMpRP5pocnveKWCzutkNOcqeLR9mNS-zS-jHXen2ykIT5-ue6YSIP-_w4pZRFEnPQNp8s5-w==", "password":
"gAAAAABkjDcPQfFY2W0KM9UGkvBELEUEbHYf7Kl_QL3qjIkRsChp5PwhUe5inPZ5fsY64cTpwrhTPFNRp3ANDePsdLjwlOIddQ==", "phone":
"gAAAAABkjDcPn-4UUF42i2NYCnr3dIEJGjRxsk4cgaupKCFquf9jPapYPlrzGQOIgh7fiY_AGAnsx9ztoA3Ioj6foHcZIPCmtg==", "prenom":
"gAAAAABkjDcP3RcKf6G9-SYYdhbeKI6k9Zftz_CnjnQsOm9TBAELbwmpqA0plbZXZNTLLRpYqjBKpuBFyfJP_OZsrjX8uUNYiw==", "public_key":
"0x391c2cecc15c218d655f07f840ba9f1c40cd433d46f4b72b9fe05f5eb9322a90xb58698f8c9709593d36dde860c20655201d3a9f0106549ca784aaa70838c9b49",
"state": "gAAAAABkjDcPGcrXD8dIqDTJoe3wXS_8CAB_6du0oeFpi8q-bLcJoB3VwCT2raTA8V-6EnOMXu_fF2t6_GD8GsV81bCd-RTqdA=="}], "timestamp": "2023-06-16
11:18:55.889688", "hash": "271963a428b23fd9e38af696d802decb5b560d4ee3a4131714e84dec28cf7e81"}]
```

Figure 3.22: An figure representing the block

- **Store information in Firebase**:

  The figure represents how the administrator stores information on Firebase. At the Accountt level, it stores the information of each registrant on the blockchain, including the currency and operations associated with their account. Regarding the accounts, it includes the registered name and password on the website, as well as the public key and private key. It's important to note that I am not referring to the mined nodes by nature. All blocks on the network are stored at the block level.

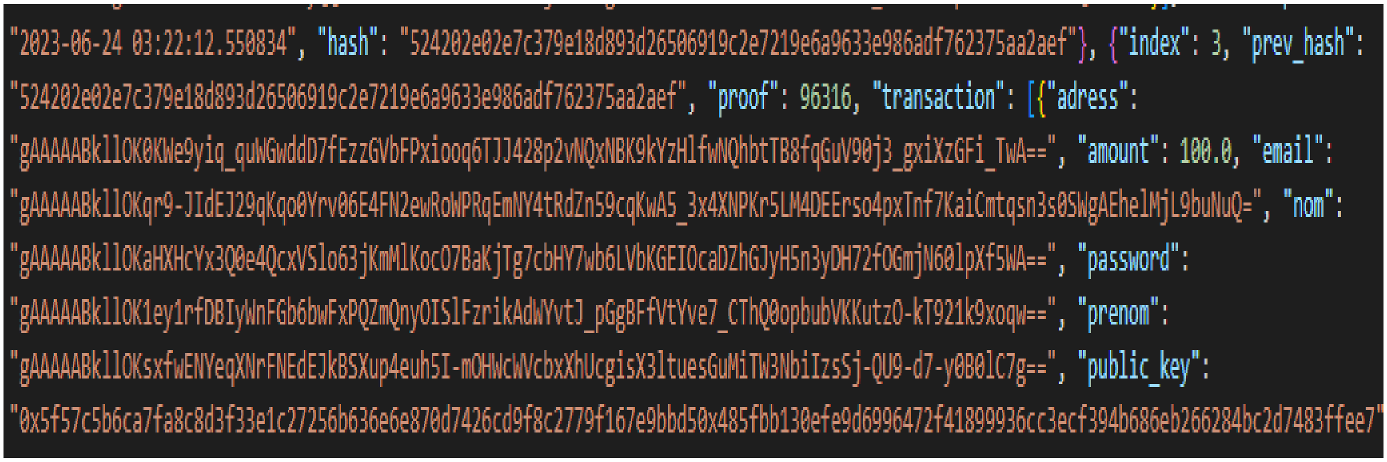


```
▾ Accountt
   ▸ gAAAAABkjDb1-U3S6x4_UblbNhu1Y5hWzO_tUO9xjIuhtyQ0x1idjMovxvWqoqeOYbJs81_iT8kc0UbOk_a7oJSIapeYFJmtkA==
   ▸ gAAAAABkjDcPPlptvxc72Fo5XZpMpRP5pocnveKWCzutkNOcqeLR9mNS-zS-jHXen2ykIT5-ue6YSIP-_w4pZRFEnPQNp8s5-w==
▾ accounts
   ▸ "aymen12345678"
   ▸ "mohamed12345"
   ▸ "mohamed12345678"
▾ block
   ▸ 1
```

Figure 3.23: figure representing information stored on Firebase

- **Inscription**:

  This option is for everyone in order to register, where private information must be entered, whether it is a private person, miner or moderator. the following image represents how information for each user is stored at the json file level

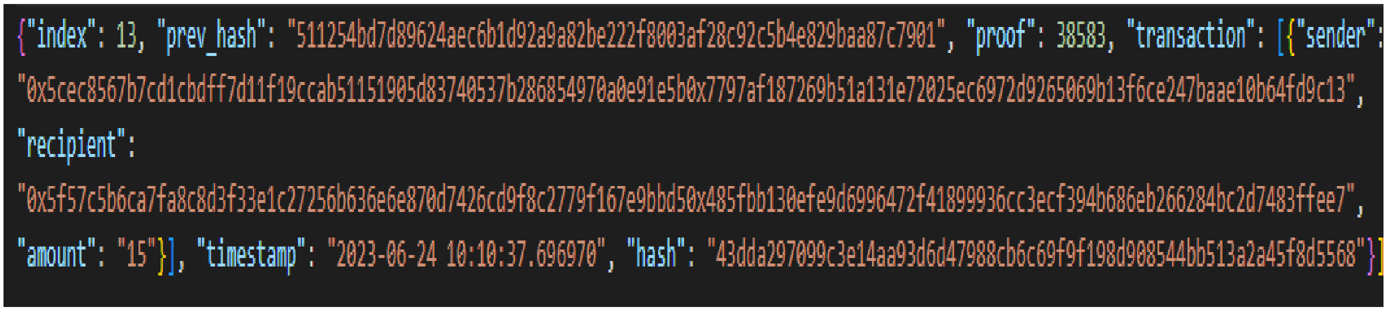


```
"2023-06-24 03:22:12.550834", "hash": "524202e02e7c379e18d893d26506919c2e7219e6a9633e986adf762375aa2aef"}, {"index": 3, "prev_hash":
"524202e02e7c379e18d893d26506919c2e7219e6a9633e986adf762375aa2aef", "proof": 96316, "transaction": [{"adress":
"gAAAAABkllOK0KWe9yiq_quWGwddD7fEzzGVbFPxiooq6TJJ428p2vNQxNBK9kYzHlfwNQhbtTB8fqGuV90j3_gxiXzGFi_TwA==", "amount": 100.0, "email":
"gAAAAABkllOKqr9-JIdEJ29qKqo0Yrv06E4FN2ewRoWPRqEmNY4tRdZn59cqKwA5_3x4XNPKr5LM4DEErso4pxTnf7KaiCmtqsn3s0SWgAEhelMjL9buNuQ=", "nom":
"gAAAAABkllOKaHXHcYx3Q0e4QcxVSlo63jKmMlKocO7BaKjTg7cbHY7wb6LVbKGEIOcaDZhGJyH5n3yDH72fOGmjN60lpXf5WA==", "password":
"gAAAAABkllOK1ey1rfDBIyWnFGb6bwFxPQZmQnyOISlFzrikAdWYvtJ_pGgBFfVtYve7_CThQ0opbubVKKutzO-kT921k9xoqw==", "prenom":
"gAAAAABkllOKsxfwENYeqXNrFNEdEJkBSXup4euh5I-mOHWcWVcbxXhUcgisX3ltuesGuMiTW3NbiIzsSj-QU9-d7-y0B0lC7g==", "public_key":
"0x5f57c5b6ca7fa8c8d3f33e1c27256b636e6e870d7426cd9f8c2779f167e9bbd50x485fbb130efe9d6996472f41899936cc3ecf394b686eb266284bc2d7483ffee7"
```

Figure 3.24: The figure shows the logging information at the json file level

- **Send amount**:

  Each person registered on the blockchain can send the currency by entering the value he wants to send, as well as the public key of the person he wants to send to. The following image represents the recording of currency sending information at the level of the json block file .

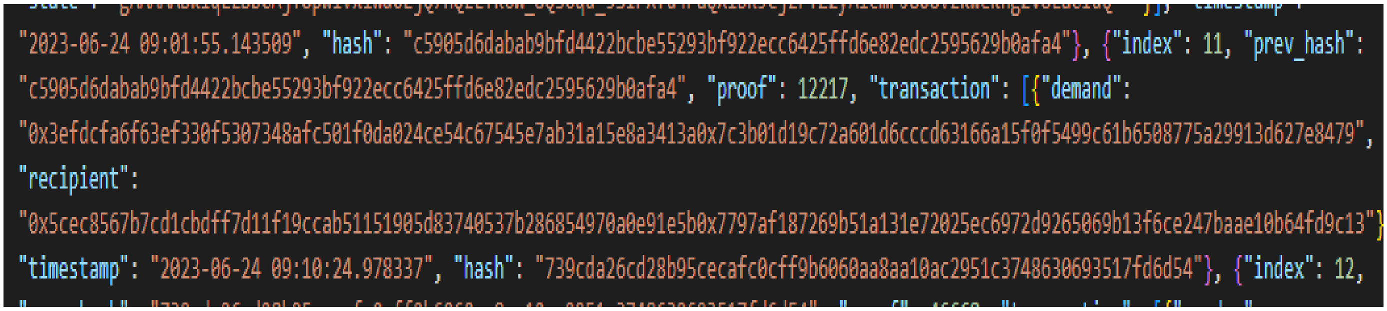


```
{"index": 13, "prev_hash": "511254bd7d89624aec6b1d92a9a82be222f8003af28c92c5b4e829baa87c7901", "proof": 38583, "transaction": [{"sender":
"0x5cec8567b7cd1cbdff7d11f19ccab51151905d83740537b286854970a0e91e5b0x7797af187269b51a131e72025ec6972d9265069b13f6ce247baae10b64fd9c13",
"recipient":
"0x5f57c5b6ca7fa8c8d3f33e1c27256b636e6e870d7426cd9f8c2779f167e9bbd50x485fbb130efe9d6996472f41899936cc3ecf394b686eb266284bc2d7483ffee7",
"amount": "15"}], "timestamp": "2023-06-24 10:10:37.696970", "hash": "43dda297099c3e14aa93d6d47988cb6c69f9f198d908544bb513a2a45f8d5568"}]
```

Figure 3.25: The figure represents a portion of the JSON block file, displaying the information for sending a coin

- **Demand**:

  Any individual registered at the level of the blockchain with its three categories can send a request to another individual by entering the public key of the recipient . The request information is stored in the block json file level as follows .

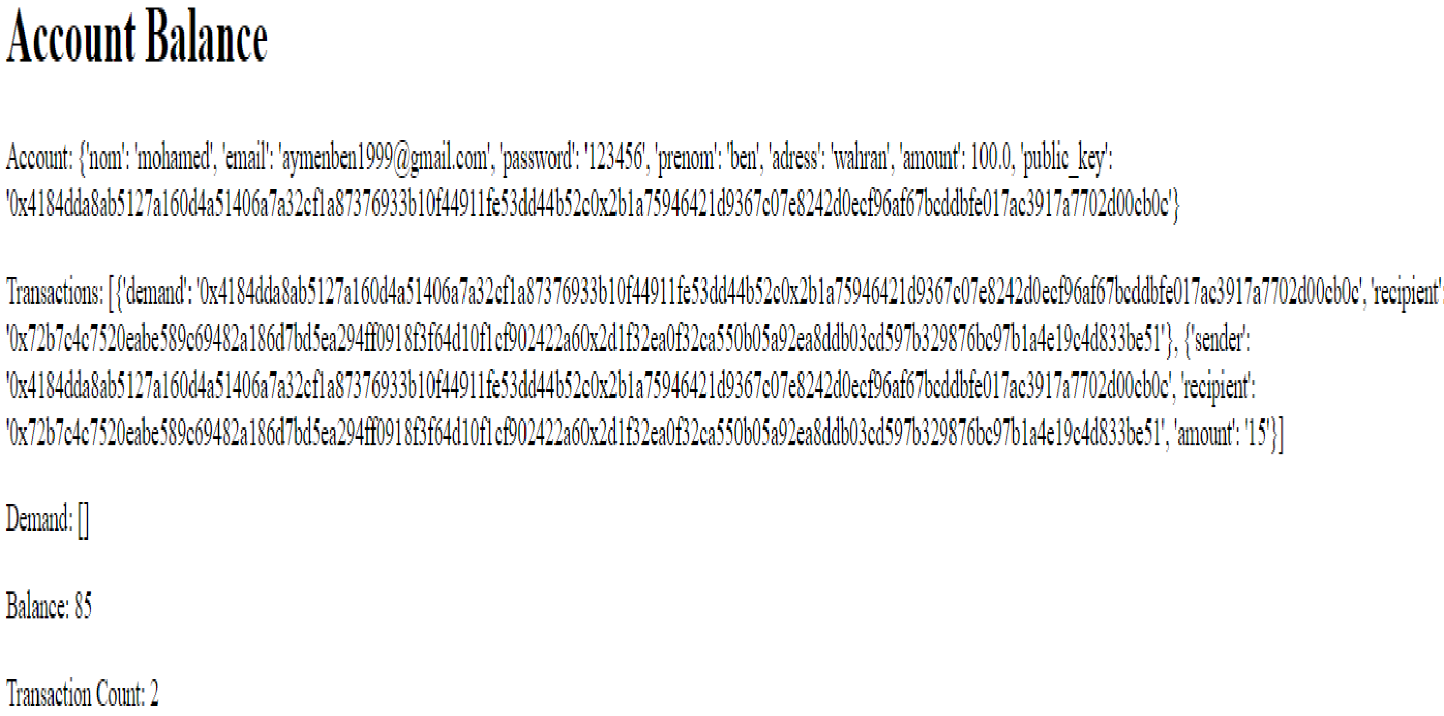


```
"2023-06-24 09:01:55.143509", "hash": "c5905d6dabab9bfd4422bcbe55293bf922ecc6425ffd6e82edc2595629b0afa4"}, {"index": 11, "prev_hash":
"c5905d6dabab9bfd4422bcbe55293bf922ecc6425ffd6e82edc2595629b0afa4", "proof": 12217, "transaction": [{"demand":
"0x3efdcfa6f63ef330f5307348afc501f0da024ce54c67545e7ab31a15e8a3413a0x7c3b01d19c72a601d6cccd63166a15f0f5499c61b6508775a29913d627e8479",
"recipient":
"0x5cec8567b7cd1cbdff7d11f19ccab51151905d83740537b286854970a0e91e5b0x7797af187269b51a131e72025ec6972d9265069b13f6ce247baae10b64fd9c13"}
"timestamp": "2023-06-24 09:10:24.978337", "hash": "739cda26cd28b95cecafc0cff9b6060aa8aa10ac2951c3748630693517fd6d54"}, {"index": 12,
```

Figure 3.26: The figure shows the request information stored at the level of the block json file

- **Show my account information**:

  Any registrant on the blockchain can view their information by selecting Get may account and pressing submit . Its information is displayed as follows .



Account Balance

Account: {'nom': 'mohamed', 'email': 'aymenben1999@gmail.com', 'password': '123456', 'prenom': 'ben', 'adress': 'wahran', 'amount': 100.0, 'public_key':
'0x4184dda8ab5127a160d4a51406a7a32cf1a87376933b10f44911fe53dd44b52c0x2b1a75946421d9367c07e8242d0ecf96af67bcddbfe017ac3917a7702d00cb0c'}

Transactions: [{'demand': '0x4184dda8ab5127a160d4a51406a7a32cf1a87376933b10f44911fe53dd44b52c0x2b1a75946421d9367c07e8242d0ecf96af67bcddbfe017ac3917a7702d00cb0c', 'recipient':
'0x72b7c4c7520eabe589c69482a186d7bd5ea294ff0918f3f64d10f1cf902422a60x2d1f32ea0f32ca550b05a92ea8ddb03cd597b329876bc97b1a4e19c4d833be51'}, {'sender':
'0x4184dda8ab5127a160d4a51406a7a32cf1a87376933b10f44911fe53dd44b52c0x2b1a75946421d9367c07e8242d0ecf96af67bcddbfe017ac3917a7702d00cb0c', 'recipient':
'0x72b7c4c7520eabe589c69482a186d7bd5ea294ff0918f3f64d10f1cf902422a60x2d1f32ea0f32ca550b05a92ea8ddb03cd597b329876bc97b1a4e19c4d833be51', 'amount': '15'}]

Demand: []

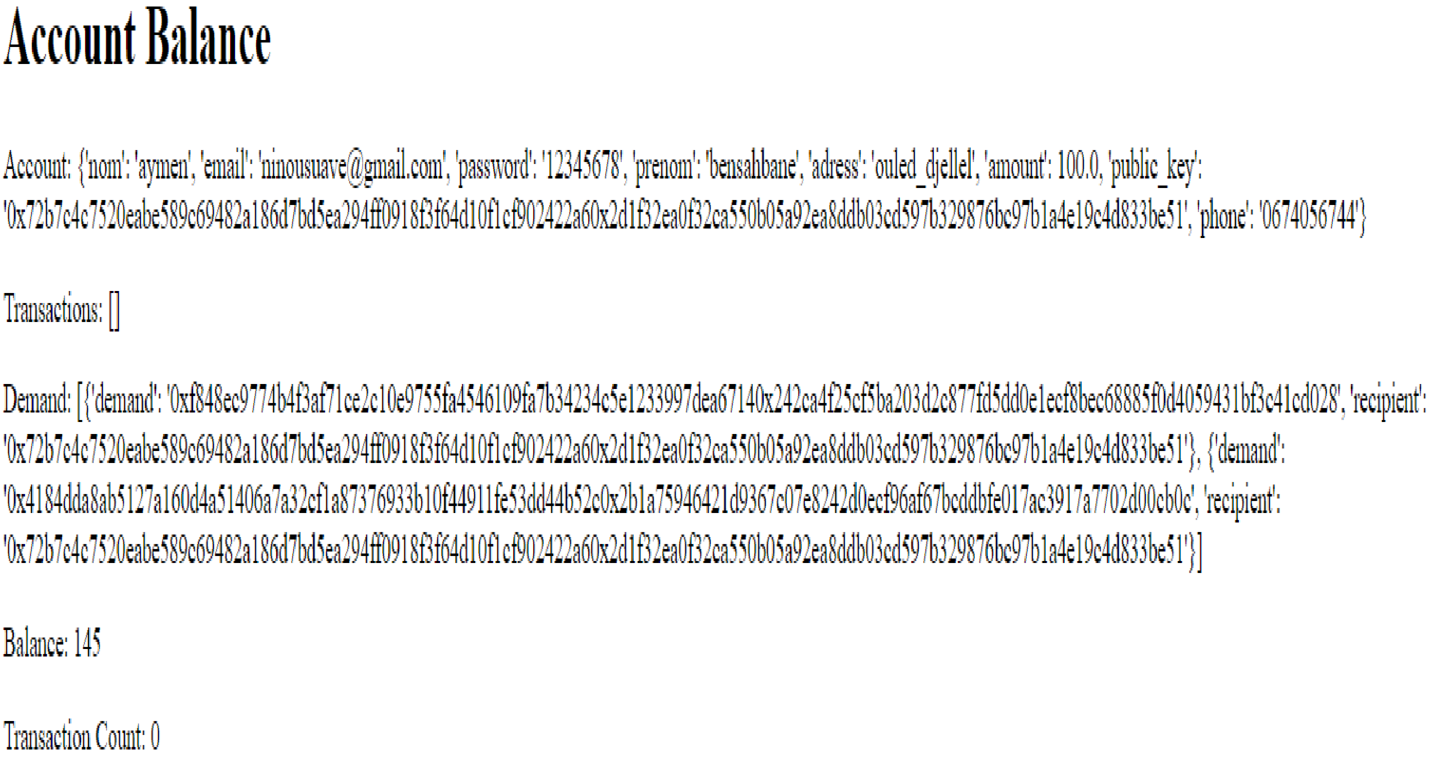
Balance: 85

Transaction Count: 2

Figure 3.27: The figure shows the account information display

- **Show specific account information**:

  This operation is available only to the administrator. By entering the public key of a person registered on the blockchain and clicking submit, information about that account will appear in unencrypted form. Its information is displayed as follows .



Account Balance

Account: {'nom': 'aymen', 'email': 'ninousuave@gmail.com', 'password': '12345678', 'prenom': 'bensahbane', 'adress': 'ouled_djellel', 'amount': 100.0, 'public_key': '0x72b7c4c7520eabe589c69482a186d7bd5ea294ff0918f3f64d10f1cf902422a60x2d1f32ea0f32ca550b05a92ea8ddb03cd597b329876bc97b1a4e19c4d833be51', 'phone': '0674056744'}

Transactions: []

Demand: [{'demand': '0xf848ec9774b4f3af71ce2c10e9755fa4546109fa7b34234c5e1233997dea67140x242ca4f25cf5ba203d2c877fd5dd0e1ecf8bec68885f0d4059431bf3c41cd028', 'recipient': '0x72b7c4c7520eabe589c69482a186d7bd5ea294ff0918f3f64d10f1cf902422a60x2d1f32ea0f32ca550b05a92ea8ddb03cd597b329876bc97b1a4e19c4d833be51'}, {'demand': '0x4184dda8ab5127a160d4a51406a7a32cf1a87376933b10f44911fe53dd44b52c0x2b1a75946421d9367c07e8242d0ecf96af67bcddbfe017ac3917a7702d00cb0c', 'recipient': '0x72b7c4c7520eabe589c69482a186d7bd5ea294ff0918f3f64d10f1cf902422a60x2d1f32ea0f32ca550b05a92ea8ddb03cd597b329876bc97b1a4e19c4d833be51'}]

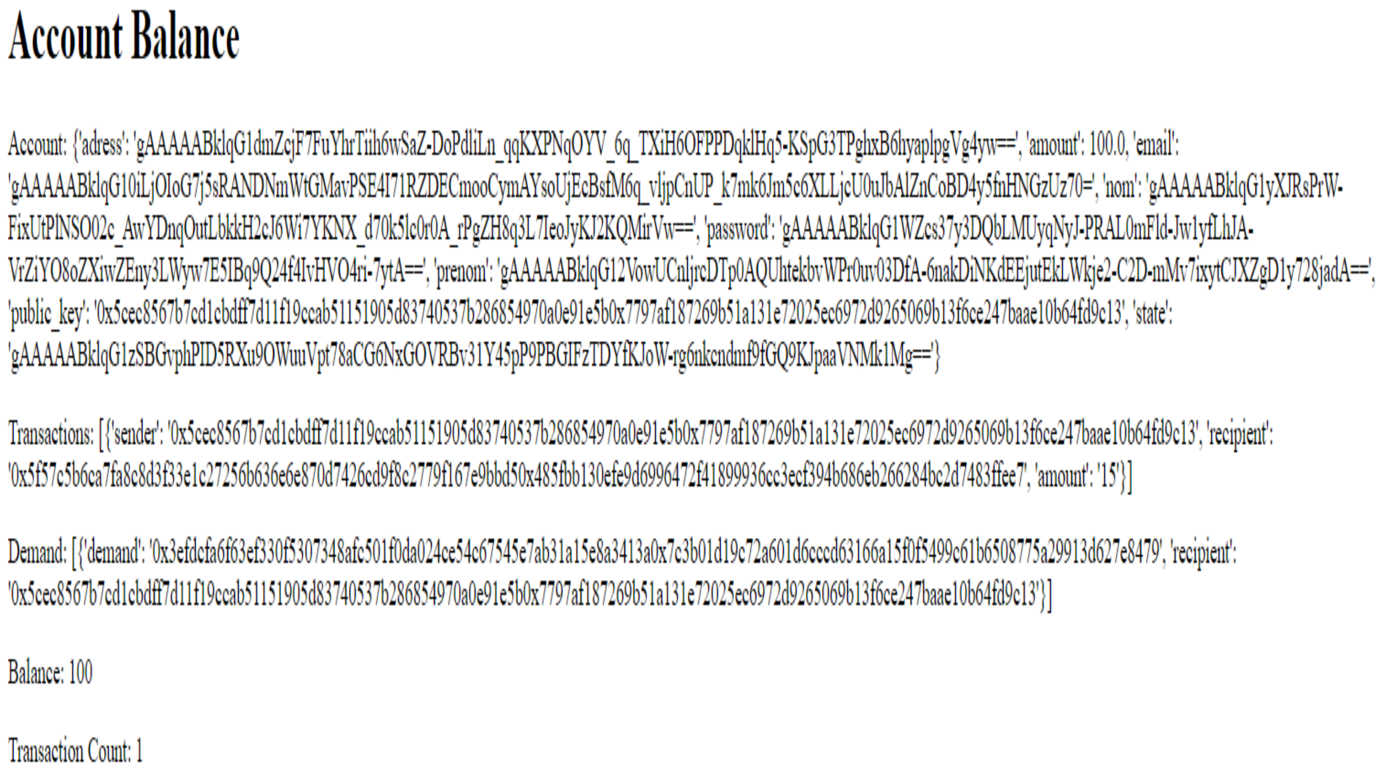
Balance: 145

Transaction Count: 0

Figure 3.28: The figure shows the account information display

- **Show the account encrypted**:

  The miner or administrator has the ability to display the information for a specific account, but it is presented in an encrypted form. This can be done by selecting "Get Account," entering the public key of the desired account, and then clicking on "Submit." It's important to note that this process is not available to regular registrants. Its information is displayed as follows .



Account Balance

```
Account: {'adress': 'gAAAAABklqG1dmZcjF7FuYhrTiih6wSaZ-DoPdliLn_qqKXPNqOYV_6q_TXiH6OFPPDqklHq5-KSpG3TPghxB6hyaplpgVg4yw==', 'amount': 100.0, 'email':
'gAAAAABklqG10iLjOIoG7j5sRANDNmWtGMavPSE4I71RZDECmooCymAYsoUjEcBsfM6q_vljpCnUP_k7mk6Jm5c6XLLjcU0uJbAlZnCoBD4y5fnHNGzUz70=', 'nom': 'gAAAAABklqG1yXJRsPrW-
FixUtPlNSO02c_AwYDnqOutLbkkH2cJ6Wi7YKNX_d70k5lc0r0A_rPgZH8q3L7IeoJyKJ2KQMirVw==', 'password': 'gAAAAABklqG1WZcs37y3DQbLMUyqNyJ-PRAL0mFld-Jw1yfLhJA-
VrZiYO8oZXiwZEny3LWyw7E5IBq9Q24f4IvHVO4ri-7ytA==', 'prenom': 'gAAAAABklqG12VowUCnljrcDTp0AQUhtekbvWPr0uv03DfA-6nakDiNKdEEjutEkLWkje2-C2D-mMv7ixytCJXZgD1y728jadA==',
'public_key': '0x5cec8567b7cd1cbdff7d11f19ccab51151905d83740537b286854970a0e91e5b0x7797af187269b51a131e72025ec6972d9265069b13f6ce247baae10b64fd9c13', 'state':
'gAAAAABklqG1zSBGvphPID5RXu9OWuuVpt78aCG6NxGOVRBv31Y45pP9PBGIFzTDYfKJoW-rg6nkcndmf9fGQ9KJpaaVNMk1Mg=='}

Transactions: [{'sender': '0x5cec8567b7cd1cbdff7d11f19ccab51151905d83740537b286854970a0e91e5b0x7797af187269b51a131e72025ec6972d9265069b13f6ce247baae10b64fd9c13', 'recipient':
'0x5f57c5b6ca7fa8c8d3f33e1c27256b636e6e870d7426cd9f8c2779f167e9bbd50x485fbb130efe9d6996472f41899936cc3ecf394b686eb266284bc2d7483ffee7', 'amount': '15'}]

Demand: [{'demand': '0x3efdcfa6f63ef330f5307348afc501f0da024ce54c67545e7ab31a15e8a3413a0x7c3b01d19c72a601d6cccd63166a15f0f5499c61b6508775a29913d627e8479', 'recipient':
'0x5cec8567b7cd1cbdff7d11f19ccab51151905d83740537b286854970a0e91e5b0x7797af187269b51a131e72025ec6972d9265069b13f6ce247baae10b64fd9c13'}]

Balance: 100

Transaction Count: 1
```

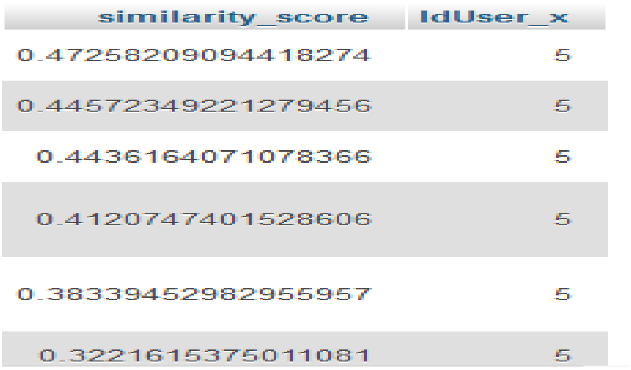
Figure 3.29: The figure shows the account information display

3.4.3 Recommendation system

Our recommendation system is based on three filtering technologies: content-based filtering, collaborative filtering, and hybrid filtering.

Our content-based filtering relies on the TF-IDF technology, on which we relied on to search for similar craftsman, and this is based on the search that the user performs. Calculating the similarity factor between the search made by the user with all the items in our database, and these results are retrieved in the form of recommendations that are classified from the highest degree of similarity to the lowest degree.

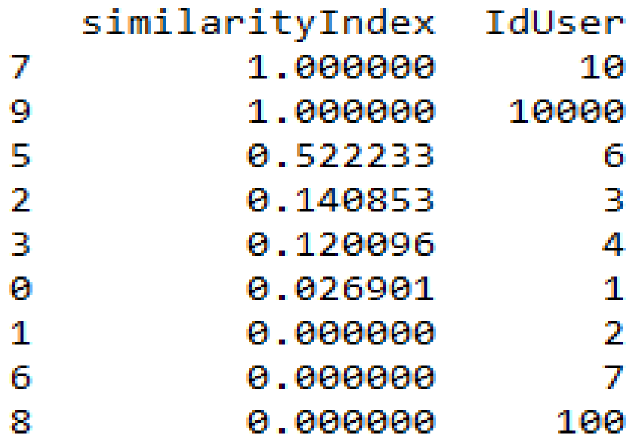
To show the results of this technique, we present this example for User 5:



similarity_score	IdUser_x
0.47258209094418274	5
0.44572349221279456	5
0.4436164071078366	5
0.4120747401528606	5
0.38339452982955957	5
0.3221615375011081	5

Figure 3.30: search similarity

As for the collaborative filtering technique, we relied on memory-based collaborative filtering. Specifically, we worked on User-based and Pearson correlation technology in order to calculate the similarity between users and identify similar users who have a high degree of similarity between them, through joint ratings between them.
In order to highlight the results of this technique, we present this example that calculates the similarity value of user 5 with all users:



```
   similarityIndex  IdUser
7         1.000000      10
9         1.000000   10000
5         0.522233       6
2         0.140853       3
3         0.120096       4
0         0.026901       1
1         0.000000       2
6         0.000000       7
8         0.000000     100
```

Figure 3.31: Similarity between users

To evaluate the performance of our models represented in the content-based filtering and collaborative filtering technologies, we subjected the system to the Precision test, the Recall test, and the F1-scor test, which are measures that give strong guidance to users to test the very high quality items from the set of available items. These metrics help determine the performance of the model in binary prediction, distinguishing good items from bad ones.

Precision test

It is a measure used to evaluate the performance of the model by measuring the amount of true positive results (True Positive) divided by true positive results (True Positives + False Positives).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3.1}$$

Recall test

Recall measures the ability of the model to detect all positive cases. If the retrieval value is high , it means that the model can detect many real positives and not miss them.

$$\text{Precision} = \frac{TP}{TP + FN} \tag{3.2}$$

F1-score test Combines precision and recall to provide a comprehensive assessment of model performance.

$$\text{F1-score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \tag{3.3}$$

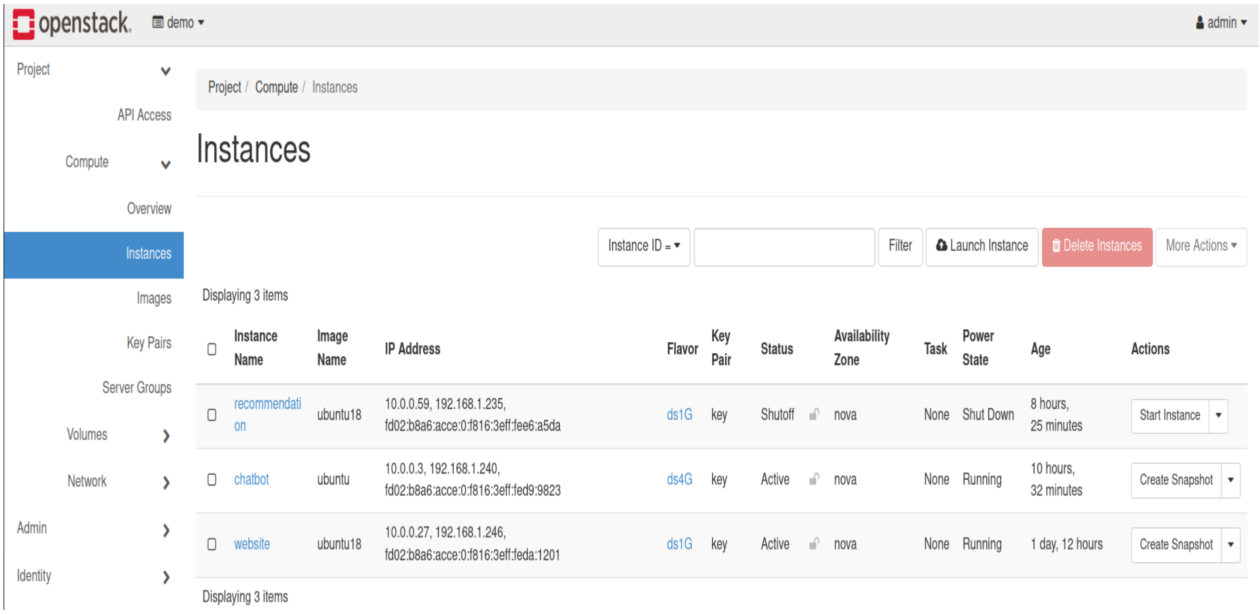
To highlight the obtained results, we took a sample of the results obtained from the two technologies and put them under the test Precision, Recall and F1-scor, where we got the following results:

	IdUser	**IdCraftsman actual**	**IdCraftsman Predicted**	**Recall**	**Precision**	**F1 Scor**
Collaborative Feltring (User-based)	5	[306 , 356 , 8 , 35 , 388 , 277 , 112 , 193 , 496 , 289 , 14 , 402 , 140 , 412 , 239 , 270 , 156 , 376 , 73 , 493 , 328 , 331 , 159 , 208]	[5 , 2 , 306 , 388 , 356 , 193 , 289 , 376 , 277 , 328 , 208 , 156 , 73 , 112 , 140 , 35 , 14 , 402 , 239 , 270 , 412 , 159 , 331 , 8]	91.66	91.66	91.66
Content based filtering (TF-IDF)	**5**	[7 , 24 , 95 , 107 , 130 , 141 , 231 , 233 , 239 , 249 , 251 , 259 , 265 , 276 , 306 , 321 , 328 , 350 , 390 ,15 ,68 ,119 , 221 , 250 , 272 , 300 , 319 , 335 , 348 , 406]	**[** 7 , 130 , 231 , 107 , 68 , 239 , 328 , 306 , 24 , 259 , 406 , 194 , 350 , 249 , 388 , 319 , 276 , 367 , 74 , 321 , 119 , 41 , 251 , 250 , 15 , 335 , 265 , 95 , 300 , 233 , 272 , 221 , 141]	**9**3.33	**8**4.84	**8**8.88

Table 3.2: Evaluation of content-based and collaborative filtering techniques

3.4.4 Private Cloud:

Dashbboard: The OpenStack dashboard was utilized to create VMs tailored to specific task requirements. This included the deployment of VMs dedicated to running the chatbot API, recommendation API, and hosting the website along with its associated database. By leveraging the flexibility of OpenStack, each VM was configured to meet the unique demands of its corresponding task, ensuring optimal performance and functionality.



Project / Compute / Instances

Instances

Displaying 3 items

□	Instance Name	Image Name	IP Address	Flavor	Key Pair	Status	Availability Zone	Task	Power State	Age	Actions
□	recommendation	ubuntu18	10.0.0.59, 192.168.1.235, fd02:b8a6:acce:0:f816:3eff:fee6:a5da	ds1G	key	Shutoff	nova	None	Shut Down	8 hours, 25 minutes	Start Instance ▾
□	chatbot	ubuntu	10.0.0.3, 192.168.1.240, fd02:b8a6:acce:0:f816:3eff:fed9:9823	ds4G	key	Active	nova	None	Running	10 hours, 32 minutes	Create Snapshot ▾
□	website	ubuntu18	10.0.0.27, 192.168.1.246, fd02:b8a6:acce:0:f816:3eff:feda:1201	ds1G	key	Active	nova	None	Running	1 day, 12 hours	Create Snapshot ▾

Displaying 3 items

Figure 3.32: Openstack Dashboard

Network Topology: As mentioned earlier, our architecture employs a Dual-Network setup, illustrated in the graph. This configuration consists of two distinct networks: a public network (blue) serving as the external network and a private network (green) functioning as the internal network. These networks are interconnected through a router, and the VMs are attached to the private network, enabling secure communication within this environment.

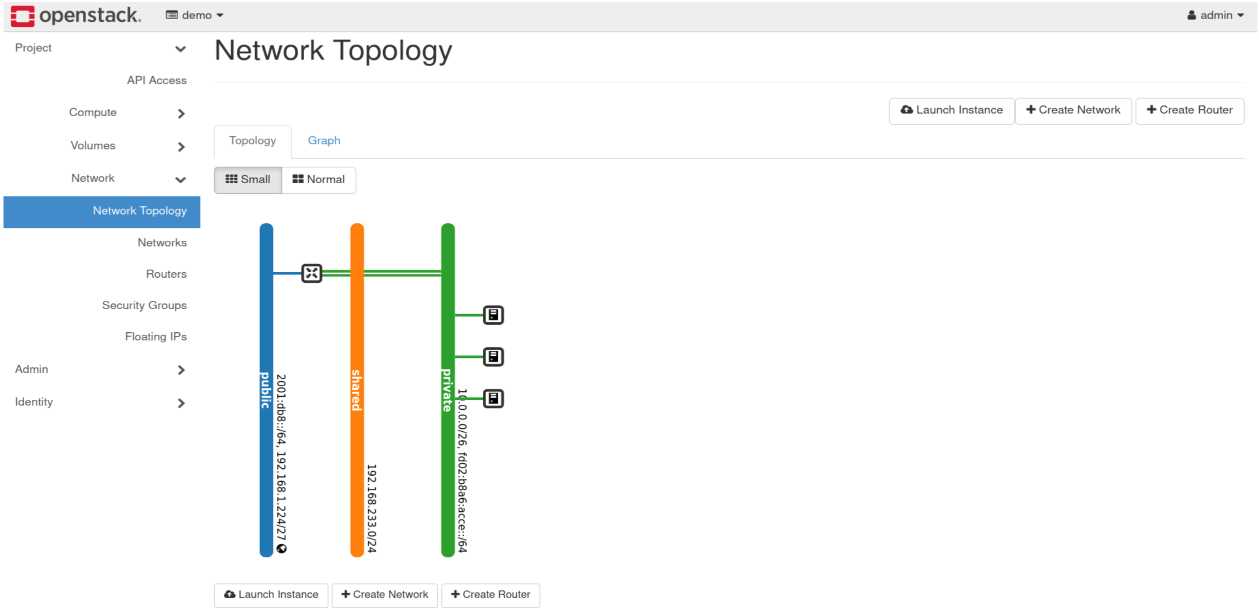
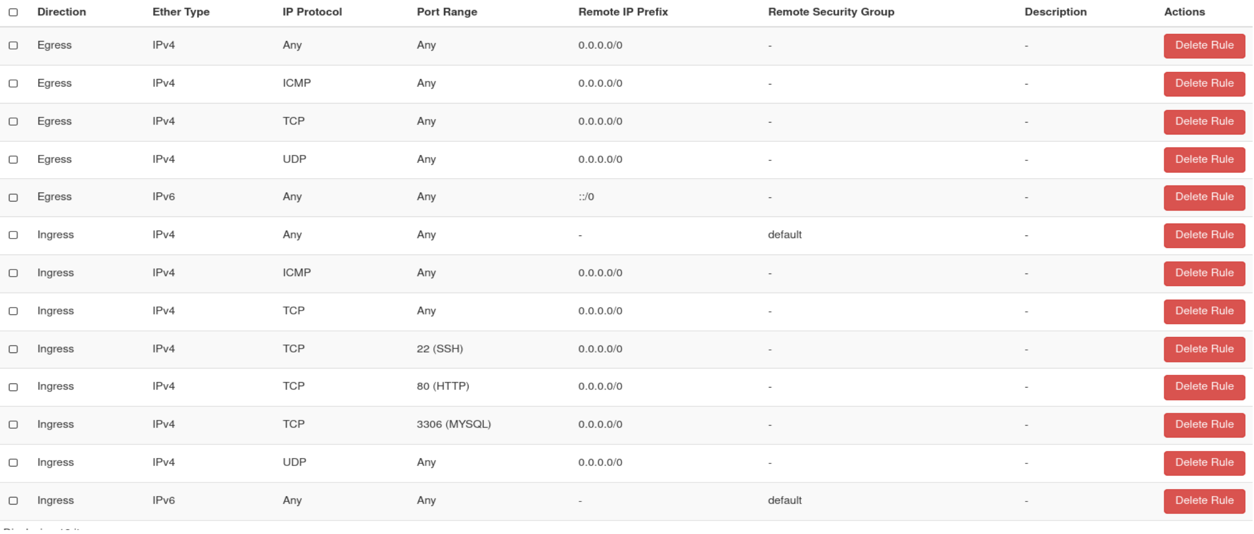


Figure 3.33: Network Topology

Security group: these rules enable seamless connections between VMs and grant them access to the required services within the OpenStack environment. By default, OpenStack follows a policy of rejecting all connections, making it essential to define specific rules to allow the necessary protocols and ports.

The figure outlines the protocols and ports that have been explicitly permitted, ensuring smooth communication between the VMs. These rules provide the necessary framework for secure and efficient interaction, enabling VMs to access the services they require.

By carefully configuring these rules, the desired protocols and ports are allowed, creating an environment where VMs can communicate and utilize the designated services effectively. This approach strikes a balance between accessibility and security, facilitating optimal performance within the OpenStack infrastructure.



□	Direction	Ether Type	IP Protocol	Port Range	Remote IP Prefix	Remote Security Group	Description	Actions
□	Egress	IPv4	Any	Any	0.0.0.0/0	-	-	Delete Rule
□	Egress	IPv4	ICMP	Any	0.0.0.0/0	-	-	Delete Rule
□	Egress	IPv4	TCP	Any	0.0.0.0/0	-	-	Delete Rule
□	Egress	IPv4	UDP	Any	0.0.0.0/0	-	-	Delete Rule
□	Egress	IPv6	Any	Any	::/0	-	-	Delete Rule
□	Ingress	IPv4	Any	Any	-	default	-	Delete Rule
□	Ingress	IPv4	ICMP	Any	0.0.0.0/0	-	-	Delete Rule
□	Ingress	IPv4	TCP	Any	0.0.0.0/0	-	-	Delete Rule
□	Ingress	IPv4	TCP	22 (SSH)	0.0.0.0/0	-	-	Delete Rule
□	Ingress	IPv4	TCP	80 (HTTP)	0.0.0.0/0	-	-	Delete Rule
□	Ingress	IPv4	TCP	3306 (MYSQL)	0.0.0.0/0	-	-	Delete Rule
□	Ingress	IPv4	UDP	Any	0.0.0.0/0	-	-	Delete Rule
□	Ingress	IPv6	Any	Any	-	default	-	Delete Rule

Figure 3.34: Security Group

Hosting website an APIs:

- **website**:

  The website is hosted on a virtual machine (VM) named "website," which serves as both the host for the website itself and the database. To ensure seamless access to the website from across the network, specific configurations were applied within the security groups. By specifying the appropriate settings in the security groups, HTTP connections have been authorized, allowing users to access the website's resources. This authorization extends throughout the network, enabling users from various locations to establish connections and interact with the website but since we're hsoting it in a local network it will allow only those who are connected to it.

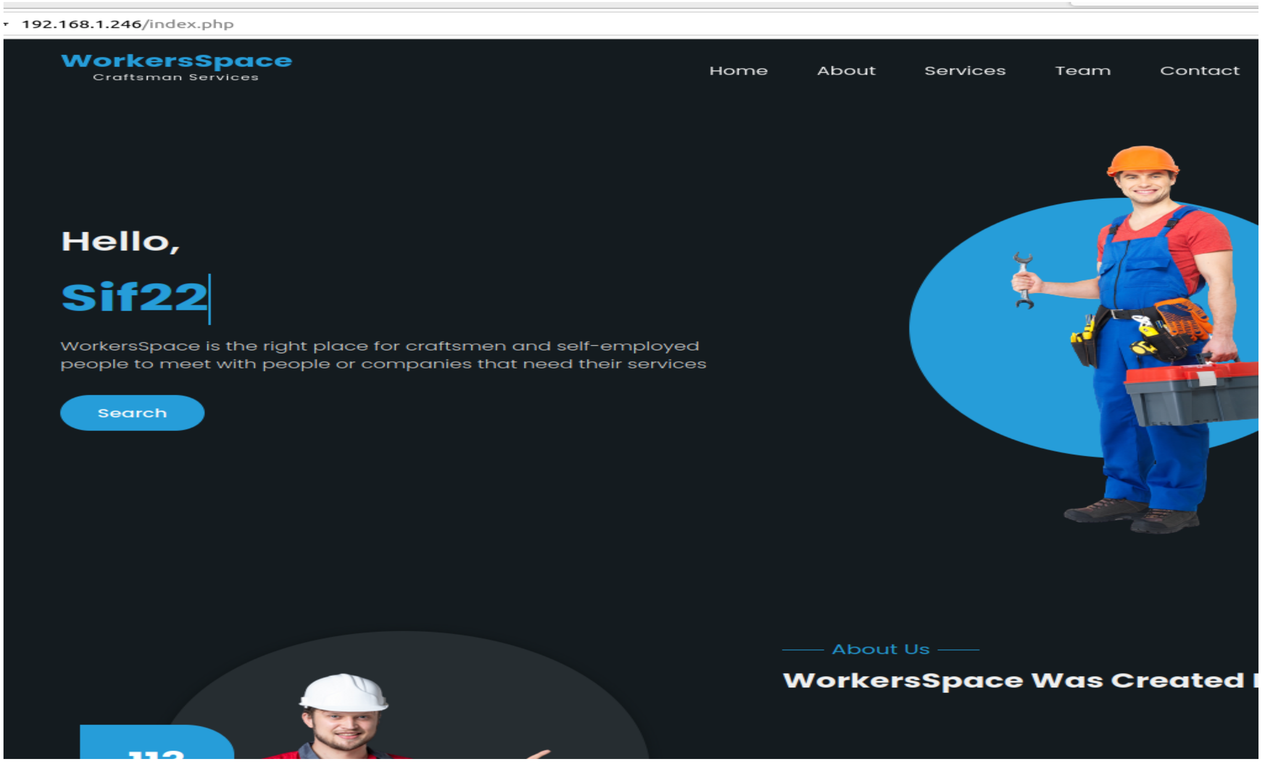
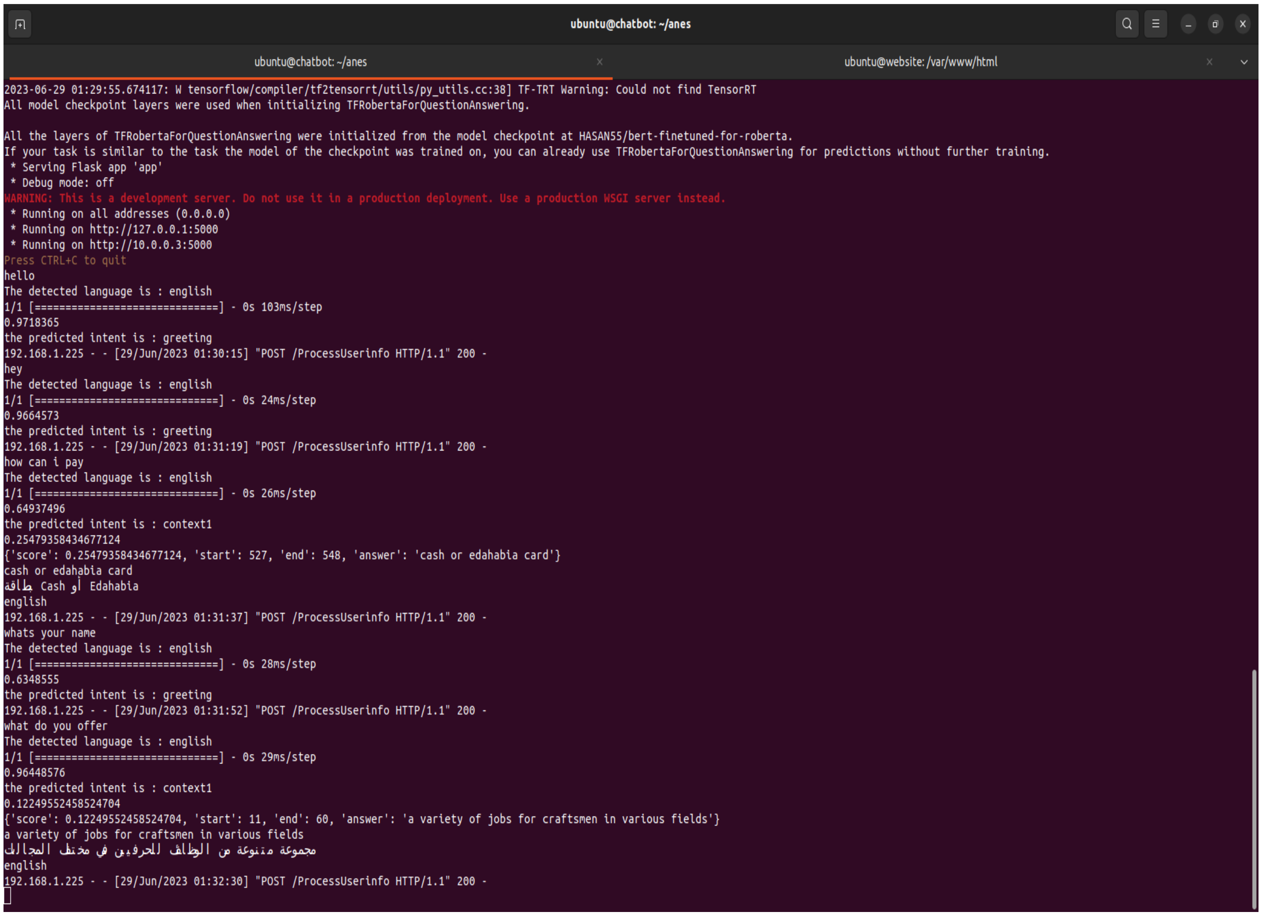


Figure 3.35: Website hosted in vm

- **Chatbot API**:

  Hosting the chatbot API necessitated configuring the virtual machine (VM) with AVX support, which was not initially available in the default CPU mode and model provided by OpenStack. To overcome this limitation, modifications were made to the installation script, enabling the VM's CPU mode to be set as "host-passthrough." This configuration ensures that all the CPU flags of the host machine are passed on to the VMs, including the necessary AVX support.

  The diagram below depicts the flow of messages between the Flask-based chatbot API and the messages sent by users of the website.



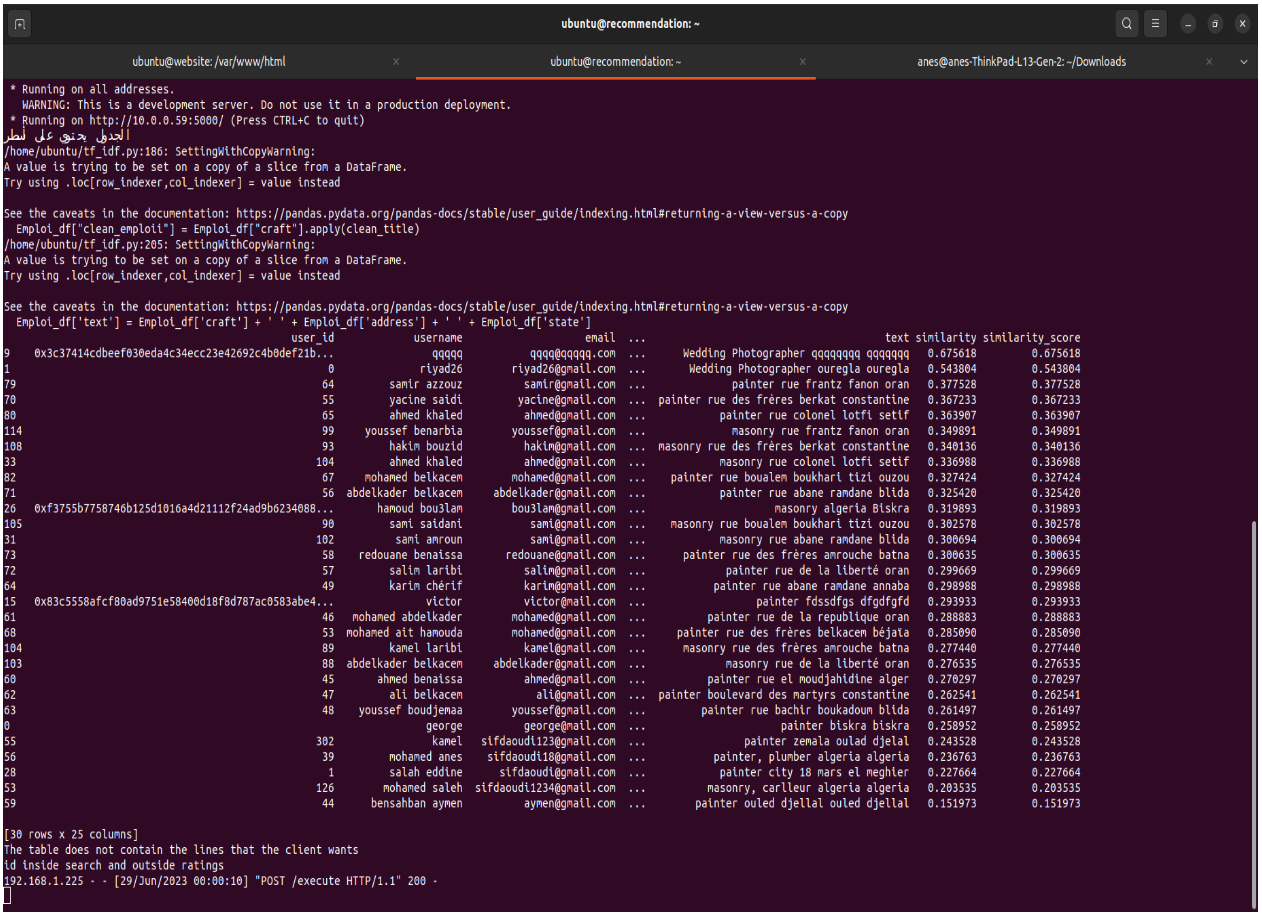
```
2023-06-29 01:29:55.674117: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
All model checkpoint layers were used when initializing TFRobertaForQuestionAnswering.

All the layers of TFRobertaForQuestionAnswering were initialized from the model checkpoint at HASAN55/bert-finetuned-for-roberta.
If your task is similar to the task the model of the checkpoint was trained on, you can already use TFRobertaForQuestionAnswering for predictions without further training.
 * Serving Flask app 'app'
 * Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://10.0.0.3:5000
Press CTRL+C to quit
hello
The detected language is : english
1/1 [==============================] - 0s 103ms/step
0.9718365
the predicted intent is : greeting
192.168.1.225 - - [29/Jun/2023 01:30:15] "POST /ProcessUserinfo HTTP/1.1" 200 -
hey
The detected language is : english
1/1 [==============================] - 0s 24ms/step
0.9664573
the predicted intent is : greeting
192.168.1.225 - - [29/Jun/2023 01:31:19] "POST /ProcessUserinfo HTTP/1.1" 200 -
how can i pay
The detected language is : english
1/1 [==============================] - 0s 26ms/step
0.64937496
the predicted intent is : context1
0.25479358434677124
{'score': 0.25479358434677124, 'start': 527, 'end': 548, 'answer': 'cash or edahabia card'}
cash or edahabia card
بطاقة Cash أو Edahabia
english
192.168.1.225 - - [29/Jun/2023 01:31:37] "POST /ProcessUserinfo HTTP/1.1" 200 -
whats your name
The detected language is : english
1/1 [==============================] - 0s 28ms/step
0.6348555
the predicted intent is : greeting
192.168.1.225 - - [29/Jun/2023 01:31:52] "POST /ProcessUserinfo HTTP/1.1" 200 -
what do you offer
The detected language is : english
1/1 [==============================] - 0s 29ms/step
0.96448576
the predicted intent is : context1
0.12249552458524704
{'score': 0.12249552458524704, 'start': 11, 'end': 60, 'answer': 'a variety of jobs for craftsmen in various fields'}
a variety of jobs for craftsmen in various fields
مجموعة متنوعة من الوظائف للحرفيين في مختلف المجالات
english
192.168.1.225 - - [29/Jun/2023 01:32:30] "POST /ProcessUserinfo HTTP/1.1" 200 -
```

Figure 3.36: Chatbot API

- **Recommendation API**:

  The recommendation API leverages the database hosted within the website's instance to deliver personalized recommendations of new craftsmen to users. This process occurs seamlessly following user interactions such as searching for or rating craftsmen.

  The diagram below illustrates the dynamic behavior of the recommendation API in response to these operations.


ubuntu@recommendation: ~
ubuntu@website: /var/www/html
ubuntu@recommendation: ~
anes@anes-ThinkPad-L13-Gen-2: ~/Downloads

```
 * Running on all addresses.
   WARNING: This is a development server. Do not use it in a production deployment.
 * Running on http://10.0.0.59:5000/ (Press CTRL+C to quit)
الجدول يحتوي على أسطر
/home/ubuntu/tf_idf.py:186: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  Emploi_df["clean_emploii"] = Emploi_df["craft"].apply(clean_title)
/home/ubuntu/tf_idf.py:205: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  Emploi_df['text'] = Emploi_df['craft'] + ' ' + Emploi_df['address'] + ' ' + Emploi_df['state']
                                               user_id            username                   email  ...                                       text similarity similarity_score
9    0x3c37414cdbeef030eda4c34ecc23e42692c4b0def21b...               qqqqq          qqqq@qqqqq.com  ...      Wedding Photographer qqqqqqqq qqqqqqq   0.675618         0.675618
1                                                    0             riyad26       riyad26@gmail.com  ...       Wedding Photographer ouregla ouregla   0.543804         0.543804
79                                                  64        samir azzouz         samir@gmail.com  ...              painter rue frantz fanon oran   0.377528         0.377528
70                                                  55        yacine saidi        yacine@gmail.com  ...  painter rue des frères berkat constantine   0.367233         0.367233
80                                                  65        ahmed khaled         ahmed@gmail.com  ...            painter rue colonel lotfi setif   0.363907         0.363907
114                                                 99    youssef benarbia       youssef@gmail.com  ...              masonry rue frantz fanon oran   0.349891         0.349891
108                                                 93        hakim bouzid         hakim@gmail.com  ...  masonry rue des frères berkat constantine   0.340136         0.340136
33                                                 104        ahmed khaled         ahmed@gmail.com  ...            masonry rue colonel lotfi setif   0.336988         0.336988
82                                                  67    mohamed belkacem       mohamed@gmail.com  ...    painter rue boualem boukhari tizi ouzou   0.327424         0.327424
71                                                  56 abdelkader belkacem    abdelkader@gmail.com  ...            painter rue abane ramdane blida   0.325420         0.325420
26   0xf3755b7758746b125d1016a4d21112f24ad9b6234088...      hamoud bou3lam       bou3lam@gmail.com  ...                     masonry algeria Biskra   0.319893         0.319893
105                                                 90        sami saidani          sami@gmail.com  ...    masonry rue boualem boukhari tizi ouzou   0.302578         0.302578
31                                                 102         sami amroun          sami@gmail.com  ...            masonry rue abane ramdane blida   0.300694         0.300694
73                                                  58   redouane benaissa      redouane@gmail.com  ...      painter rue des frères amrouche batna   0.300635         0.300635
72                                                  57        salim laribi         salim@gmail.com  ...             painter rue de la liberté oran   0.299669         0.299669
64                                                  49        karim chérif         karim@gmail.com  ...           painter rue abane ramdane annaba   0.298988         0.298988
15   0x83c5558afcf80ad9751e58400d18f8d787ac0583abe4...              victor         victor@mail.com  ...                  painter fdssdfgs dfgdfgfd   0.293933         0.293933
61                                                  46  mohamed abdelkader       mohamed@gmail.com  ...          painter rue de la republique oran   0.288883         0.288883
68                                                  53 mohamed ait hamouda       mohamed@gmail.com  ...     painter rue des frères belkacem béjaïa   0.285090         0.285090
104                                                 89        kamel laribi         kamel@gmail.com  ...      masonry rue des frères amrouche batna   0.277440         0.277440
103                                                 88 abdelkader belkacem    abdelkader@gmail.com  ...             masonry rue de la liberté oran   0.276535         0.276535
60                                                  45      ahmed benaissa         ahmed@gmail.com  ...          painter rue el moudjahidine alger   0.270297         0.270297
62                                                  47         ali belkacem           ali@gmail.com  ...  painter boulevard des martyrs constantine   0.262541         0.262541
63                                                  48   youssef boudjemaa       youssef@gmail.com  ...         painter rue bachir boukadoum blida   0.261497         0.261497
0                                                               george         george@mail.com  ...                      painter biskra biskra   0.258952         0.258952
55                                                 302               kamel  sifdaoudi123@gmail.com  ...                painter zemala oulad djelal   0.243528         0.243528
56                                                  39        mohamed anes   sifdaoudi18@gmail.com  ...           painter, plumber algeria algeria   0.236763         0.236763
28                                                   1        salah eddine     sifdaoudi@gmail.com  ...            painter city 18 mars el meghier   0.227664         0.227664
53                                                 126       mohamed saleh sifdaoudi1234@gmail.com  ...          masonry, carlleur algeria algeria   0.203535         0.203535
59                                                  44     bensahban aymen         aymen@gmail.com  ...        painter ouled djellal ouled djellal   0.151973         0.151973

[30 rows x 25 columns]
The table does not contain the lines that the client wants
id inside search and outside ratings
192.168.1.225 - - [29/Jun/2023 00:00:10] "POST /execute HTTP/1.1" 200 -
```

Figure 3.37: Recommendation API

3.5 Conclusion

In this chapter, we have showcased the implementation of our system and shared the outcomes achieved through our efforts. To ensure effective programming and the successful accomplishment of our objectives, we carefully selected the appropriate software and libraries. The results obtained were highly promising, demonstrating the effectiveness of our well-structured and organized system.

Conclusion

In this thesis, we have presented WorkersSpace, a comprehensive platform consisting of various components and tasks. The central focus of the platform is the website, which encompasses three distinct functionalities. Firstly, the chatbot utilizes deep learning transformers and MLP techniques to handle customer support efficiently. Secondly, blockchain technology ensures the security and integrity of user data. Lastly, the recommendation system incorporates content-based filtering, collaborative filtering, and hybrid recommendation techniques to enhance user satisfaction.

All these components are seamlessly integrated into the website, which is hosted within a private cloud environment using the open-source cloud computing infrastructure, OpenStack. Throughout the thesis, we have meticulously developed and integrated each component of the WorkersSpace platform, ensuring ease of use and user satisfaction.

The results obtained from our experiments have been promising, as demonstrated in the results section. Additionally, we have provided comprehensive system interfaces to showcase the user experience.

Our ultimate objective is to serve our customers by offering a user-friendly platform that facilitates efficient interactions between users and appropriate workers (craftsmen). Furthermore, the platform aims to assist craftsmen in finding suitable job opportunities, fostering collaboration and ultimately contributing to making the world a better place.

In conclusion, we have successfully developed and integrated the WorkersSpace platform, providing a valuable tool for users to connect with craftsmen effectively. Through this platform, we aim to create a positive impact by facilitating meaningful interactions and fostering mutually beneficial relationships between users and workers.

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[3] Fabio Tamburini, Philipp Cimiano, and Simone Preite. Deep question answering: A new teacher for distilbert.

[4] Hugging Face. The hugging face course, 2022. https://huggingface.co/course, 2022. [Online; accessed 02/05/2023].

[5] Kavita Ganesan. Data cleaning in 5 easy steps + examples, 2020. Accessed: April 30, 2023.

[6] Ahmad Asadullah, Isam Faik, and Atreyi Kankanhalli. Digital platforms: A review and future directions. *PACIS*, page 248, 2018.

[7] Sonali Yadav. Comparative study on open source software for cloud computing platform: Eucalyptus, openstack and opennebula. *Research Inventy :International Journal of Engineering Science*, 3:51, 10 2013.

[8] Minh-Phung Thi Do, DV Nguyen, and Loc Nguyen. Model-based approach for collaborative filtering. In *6th International Conference on Information Technology for Education*, pages 217–228, 2010.

[9] M.N. Almunawar, M. Anshari Ali, and S. Ariff Lim. *Handbook of Research on Innovation and Development of E-Commerce and E-Business in ASEAN*. Advances in Electronic Commerce. IGI Global, 2020.

[10] Juho Makkonen. How to build a service marketplace: Tips and tools, 2022. Accessed: April 18, 2023.

[11] Product marketplace solution by omnyfy. https://omnyfy.com/solution/product-marketplace/. Accessed: April 18, 2023.

[12] MercadoLibre. Mercadolibre reports third quarter 2021 financial results, 2021. Accessed: April 18, 2023.

[13] European Parliament. Amendments adopted by the european parliament on 13 november 2020 on the proposal for a regulation of the european parliament and of the council establishing the investeu programme (com(2020)0403 — c9-0158/2020 — 2020/0108(cod)) amendment 1. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:52020AP0306&from=EN, 2020. Accessed: April 17, 2023.

[14] Office of the Special Rapporteur for Freedom of Expression. OSR: CYBER PLUS, 2022. Accessed: April 17, 2023.

[15] Ministry of Transportation and Infrastructure. Maintenance specifications – see schedule 1, section 6: Snow avalanche response. https://www2.gov.bc.ca/assets/gov/driving-and-transportation/transportation-infrastructure/highway-bridge-maintenance/avalanche/maintenance-specifications–see-schedule-1-section-6-snow-avalanche-response-.pdf, 2019. Accessed: April 17, 2023.

[16] Bmc cloud services master agreement. https://bmc.pactsafe.io/versions/6132a482972a8766ee86acef.pdf. Accessed April 17, 2023.

[17] Liferay. Understanding the elements of digital experience platforms, 2022. march 15,2023.

[18] Navin Sabharwal and Amit Agrawal. *BERT Model Applications: Question Answering System*, pages 97–137. Apress, Berkeley, CA, 2021.

[19] Joseph Weizenbaum. Eliza—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, 26(1):23–28, 1983.

[20] OpenAi. OpenAi. https://openai.com/. 30 May 2023.

[21] Sandeep A Thorat and Vishakha Jadhav. A review on implementation issues of rule-based chatbot systems. In *Proceedings of the international conference on innovative computing & communications (ICICC)*, 2020.

[22] Akshay Kulkarni, Adarsha Shivananda, and Anoosh Kulkarni. *Natural language processing projects: Build next-generation NLP applications using AI techniques*. Springer, 2022.

[23] Rui Yan, Yiping Song, and Hua Wu. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 55–64, 2016.

[24] Guendalina Caldarini, Sardar Jaf, and Kenneth McGarry. A literature survey of recent advances in chatbots. *Information*, 13(1):41, 2022.

[25] Bathe Pooja Prashant, MS Anil, and KM Dilip. Online chatting system for college enquiry using knowledgeable database. 2017.

[26] Addi Ait-Mlouk, Sadi Alawadi, Salman Toor, and Andreas Hellander. Fedbot: Enhancing privacy in chatbots with federated learning. *arXiv preprint arXiv:2304.03228*, 2023.

[27] Anbang Xu, Zhe Liu, Yufan Guo, Vibha Sinha, and Rama Akkiraju. A new chatbot for customer service on social media. In *Proceedings of the 2017 CHI conference on human factors in computing systems*, pages 3506–3510, 2017.

[28] Meshrif Alruily. Arrasa: Channel optimization for deep learning-based arabic nlu chatbot framework. *Electronics*, 11(22):3745, 2022.

[29] A. Zwitter and O.J. Gstrein. *Identity and Privacy Governance*. Frontiers Research Topics. Frontiers Media SA, 2021.

[30] IBM. About ibm food trust. pages 3–17.

[31] Mohammad Hammoudeh Bamidele Adebisi Umar Raza. Sohail Jabbar, Huw Lloyd. Blockchain-enabled supply chain: analysis, challenges, and future directions. *Multimedia Systems*, 27(4):787–806, 2021.

[32] BENMAMAS Cylia BENLAKEHAL Kinza. Design and production of recommendation system for e-commerce. Master's thesis, University Abderrahmane Mira of Bejaia, September 2021. Thesis to obtain the Master of Computer Science Degree in Computer Network Administration and Security and Artificial Intelligence.

[33] Roberto Dias Torres Júnior. Combining collaborative and content-based filtering to recommend research papers.

[34] Arkadiusz Krysik . Netflix algorithm: Everything you need to know about the recommendation system of the most popular streaming portal, 2021. march 24, 2023.

[35] Ashish Gupta . Netflix recommender system - case study, 2022. march 24, 2023.

[36] Stephen J. Bigelow. What is a private cloud?, 2022. Accessed: May 4, 2023.

[37] Saad Makhdoom. Ecommerce faq chatbot dataset. https://www.kaggle.com/datasets/saadmakhdoom/ecommerce-faq-chatbot-dataset, 2023.

[38] Customer Support intent. Customer support intent dataset. https://huggingface.co/datasets/bitext/customer-support-intent-dataset, 2023.

[39] Nirali Vaghani. Chatbot dataset. https://www.kaggle.com/dsv/5024271, 2023.

[40] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250, 2016.

[41] Josh Patterson and Adam Gibson. *Deep learning: A practitioner's approach.* " O'Reilly Media, Inc.", 2017.

[42] Pariwat Ongsulee. Artificial intelligence, machine learning and deep learning. In *2017 15th international conference on ICT and knowledge engineering (ICT&KE)*, pages 1–6. IEEE, 2017.

[43] Asmaul Hosna, Ethel Merry, Jigmey Gyalmo, Zulf-ikar Alom, Zeyar Aung, and Mohammad Abdul Azim. Transfer learning: a friendly introduction. *Journal of Big Data*, 9(1):102, 2022.

[44] Tasmiah Tahsin Mayeesha, Abdullah Md Sarwar, and Rashedur M Rahman. Deep learning based question answering system in bengali. *Journal of Information and Telecommunication*, 5(2):145–178, 2021.

[45] R. Myers. *Proof of Work: Blockchain Provocations 20112021*. Urbanomic / Art Editions. MIT Press, 2023.

[46] Data cleaning? Accessed: April 30, 2023.

[47] George Lawton. data preprocessing. Accessed: May 03, 2023.

[48] Abhinav Ajitsaria. Build a recommendation engine with collaborative filtering, 2021. Accessed: May 13, 2023.

[49] Mehdi Elahi, Francesco Ricci, and Neil Rubens. A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20:29–50, 2016.

[50] David Nettleton. *Commercial data mining: processing, analysis and modeling for predictive analytics projects*. Elsevier, 2014.

[51] Dunja Mladenic. Text-learning and related intelligent agents: a survey. *IEEE intelligent systems and their applications*, 14(4):44–54, 1999.

[52] Alfirna Rizqi Lahitani, Adhistya Erna Permanasari, and Noor Akhmad Setiawan. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International Conference on Cyber and IT Service Management*, pages 1–6. IEEE, 2016.

[53] Kavita Ganesan. What is inverse document frequency (idf)?, 2023. Accessed: May 16, 2023.

[54] Hesham Asem. Nlp tf-idf, 2021. Accessed: May 20, 2023.

[55] Festim Halili and Avni Rustemi. Modeling of an e-business system. *INTERNATIONAL JOURNAL OF COMPUTERS TECHNOLOGY*, 15:7308–7315, 11 2016. Accessed: May 06, 2023.

[56] M. Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley object technology series. Addison-Wesley, 2004.

[57] J. Schmidbauer, J. Barnard, I. Franzki, K. Graul, D. Stoever, R. Trumpp, and IBM Redbooks. *Enhanced Networking on IBM z/VSE*. IBM Redbooks, 2014.

[58] Erin Manette Cartas Espinel Kim Gunnerson Joanne Liu Johnny Wei-Bing Lin, Hannah Aizenman. *An Introduction to Python Programming for Scientists and Engineers*. Cambridge University Press, 2022.

[59] Eric Matthes. *Python Crash Course, 3rd Edition: A Hands-On, Project-Based Introduction to Programming*. No Starch Press, 2019.

[60] Visual Studio Code. Visual Studio Code FAQ. https://code.visualstudio.com/docs/supporting/faq. 25 May 2023.

[61] Jupyter Notebooks . Jupyter notebooks in vs code, 2023. march 25, 2023.

[62] Jupyter . Jupyter, 2023. march 25, 2023.

[63] seddik Damiati . What is xampp and how to install and use it, 2021. march 25, 2023.

[64] xampp . Xampp apache + mariadb + php + perl. march 25, 2023.

[65] phpMyAdmin . Bringing mysql to the web. march 25, 2023.

[66] phpMyAdmin . phpmyadmin, 2022. march 25, 2023.

[67] mysql . Qu'est-ce que mysql ? une explication simple pour les débutants, 2022. march 25, 2023.

[68] J. Duckett. *HTML and CSS: Design and Build Websites*. Wiley, 2011.

[69] E. Meyer and E. Weyl. *CSS: the Definitive Guide: Web Layout and Presentation*. O'Reilly Media, 2023.

[70] D. Flanagan. *JavaScript: The Definitive Guide*. Definitive Guides. O'Reilly, 2002.

[71] D.S. McFarland. *JavaScript & JQuery: The Missing Manual*. Missing manual. O'Reilly Media, 2014.

[72] B. Bibeault and Y. Katz. *Jquery in Action*. CreateSpace Independent Publishing Platform, 2017.

[73] J. Spurlock. *Bootstrap: Responsive Web Development*. O'Reilly Media, 2013.

[74] L. Welling, L. Thomson, and an O'Reilly Media Company Safari. *PHP and MySQL® Web Development, Fifth Edition*. Developer's library. Addison-Wesley Professional, 2016.

[75] Ekaba Bisong and Ekaba Bisong. Google colaboratory. *Building machine learning and deep learning models on google cloud platform: a comprehensive guide for beginners*, pages 59–64, 2019.

[76] B. Mather. *Coding for Kids in Python: Python Programming Projects for Kids and Beginners to Get Started Programming Fun Games*. Abiprod Pty Ltd, 2022.

[77] Bo Pang, Erik Nijkamp, and Ying Nian Wu. Deep learning with tensorflow: A review. *Journal of Educational and Behavioral Statistics*, 45(2):227–248, 2020.

[78] PS Lokhande, Fankar Aslam, Nabeel Hawa, Jumal Munir, and Murade Gulamgaus. Efficient way of web development using python and flask. 2015.

[79] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[80] NLTK. NLTK: Natural language toolkit. https://www.nltk.org/. Accessed: 25-05-2023.

[81] Hugging Face. Datasets: The datasets library. https://huggingface.co/docs/datasets/index. Accessed: 25-05-2023.

[82] P. Miller and C. Bryce. *Learning Python for Forensics: Leverage the power of Python in forensic investigations, 2nd Edition*. Packt Publishing, 2019.

[83] A. Nath. *Python Cryptography*. INDEPENDENTLY PUBLISHED, 2018.

[84] W. Chun. *Core Python Programming*. Number v. 1 in Core Python Programming. Prentice Hall, 2001.

[85] K.S. Kaswan, J.S. Dhatterwal, and B. Balamurugan. *Python for Beginners*. CRC Press, 2023.

[86] M. Tibouchi and H. Wang. *Advances in Cryptology – ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part III*. Lecture Notes in Computer Science. Springer International Publishing, 2021.

[87] P.K. Singh, B.K. Panigrahi, N.K. Suryadevara, S.K. Sharma, and A.P. Singh. *Proceedings of ICETIT 2019: Emerging Trends in Information Technology*. Lecture Notes in Electrical Engineering. Springer International Publishing, 2019.

[88] A. Ayeva, K. Ayeva, and A. Saed. *Python In - Depth: Use Python Programming Features, Techniques, and Modules to Solve Everyday Problems*. BPB PUBN, 2020.

[89] pandas . Pandas : la bibliothèque python dédiée à la data science, 2023. march 26, 2023.

[90] Pandas Library . Panda, 2021. march 26, 2023.

[91] sqlalchemy . sqlalchemy, 2023. march 26, 2023.

[92] NumPy . Numpy, 2022. march 26, 2023.

[93] scikit-learn . scikit-learn-python. march 26, 2023.

[94] OpenStack Nova. https://docs.openstack.org/nova/latest/. Accessed: June 16, 2023.

[95] OpenStack Keystone. https://www.openstack.org/software/releases/ocata/components/keystone. Accessed: June 16, 2023.

[96] OpenStack Cinder. https://www.openstack.org/software/releases/zed/components/cinder. Accessed: June 16, 2023.

[97] OpenStack Neutron. https://www.openstack.org/software/releases/zed/components/neutron. Accessed: June 16, 2023.

[98] OpenStack Placement. https://www.openstack.org/software/releases/zed/components/placement. Accessed: June 16, 2023.

[99] OpenStack Swift. https://www.openstack.org/software/releases/zed/components/swift. Accessed: June 16, 2023.

[100] OpenStack Glance. https://www.openstack.org/software/releases/zed/components/glance. Accessed: June 16, 2023.

[101] OpenStack Horizon. https://www.openstack.org/software/releases/zed/components/horizon. Accessed: June 16, 2023.