People's Democratic Republic of Algeria

Ministry of Higher Education and Scientific Research

University of Science and Technology Houari Boumediene
Faculty of Computer Science

Department of Computer Science

Bachelor's Thesis

Field: Computer Science

Specialization: ISIL

Theme:

Development of a Collaborative Platform for Researchers and Developers in Arabic Natural Language Processing

Topic proposed by:

Pr. Guessoum Ahmed
Pr. Ferrat Kamel
As part of a research project by the Algerian Academy of the Arabic Language

Co-supervisor:
Dr. Lamia Berkani

Presented by:
BEN MAMMAR HAMZA
NEGGAZI MOHCEN
BENABDELMOUMENE ABDERRAOUF
TOUBACHE ABDELHAMID

In front of the jury composed of:

Dr. Lamia Berkani	President
Dr. Sebai Meriem	Member

Acknowledgements

I would like to express my sincere gratitude to everyone who contributed to the completion of this work.

My special thanks go to **Dr. Berkani**, **Prof. Ferrat**, and **Prof. Guessoum** for their close supervision, valuable guidance, and availability throughout this project. Their expertise and support have been essential at every stage of this work.

I also would like to thank the **Algerian Academy of the Arabic Language** for providing a stimulating and supportive environment, as well as for the resources made available during the research period.

Finally, I would like to extend my thanks to all those who, directly or indirectly, encouraged and supported me throughout this journey.

Summary:

This thesis presents the design and development of a collaborative platform dedicated to researchers and developers specializing in Arabic Natural Language Processing (Arabic NLP). Addressing the specific challenges of Arabic NLP, particularly the dispersion of linguistic resources, lack of coordination among researchers, and absence of centralized collaborative spaces, this research proposes an integrated technological solution.

The study begins with a comparative state-of-the-art analysis of existing platforms (ResearchGate, Hugging Face, CLARIN, Masader) and chatbots based on large language models (ChatGPT, Gemini, Claude), revealing a significant lack of specialized tools for the Arabic-speaking NLP community. The needs analysis identifies three user profiles (public, registered, administrator) and defines the functional and non-functional requirements of the platform.

The architectural design adopts a modular approach integrating user management, resources, collaborative projects, discussion forums, and scientific events. The technical implementation relies on a three-tier architecture using Django (backend), PostgreSQL (database), and a responsive multilingual web interface (Arabic/English). The integration of Elasticsearch optimizes Arabic content search, while an intelligent chatbot based on LLaMA 3 and LangChain provides contextual assistance to users.

This platform contributes to structuring the Arabic NLP scientific community by centralizing resources, facilitating interdisciplinary and geographical collaborations, and promoting knowledge sharing. Future perspectives include integrating an intelligent recommendation engine and developing an API for interoperability with other domain tools.

Keywords:

Arabic Natural Language Processing, collaborative platform, Django, PostgreSQL, Elasticsearch, intelligent chatbot, LLaMA 3, LangChain, WebSocket, discussion forum, project management, linguistic corpora, NLP, computational linguistics, multilingual interface, FastAPI, vector search, FAISS, HuggingFace Embeddings.

ملخص

تقدم هذه الأطروحة تصميم وتطوير منصة تعاونية مخصصة للباحثين والمطورين المتخصصين في معالجة اللغة العربية الآلية. في مواجهة التحديات المحددة لمعالجة اللغة العربية الآلية، وخاصة تشتت الموارد اللغوية، وعدم التنسيق بين الباحثين، وغياب المساحات التعاونية المركزية، يقترح هذا البحث حلاً تكنولوجياً متكاملاً.

تبدأ الدراسة بتحليل مقارن لأحدث المنصات الموجودة (Masader، CLARIN، Hugging Face، ResearchGate) ، مما يكشف عن نقص كبير في الأدوات (Claude، Gemini، ChatGPT) وروبوتات المحادثة القائمة على نماذج اللغة الكبيرة المتخصصة لمجتمع معالجة اللغة العربية الآلية.

يحدد تحليل الاحتياجات ثلاثة ملفات شخصية للمستخدمين (عام، مسجل، مسؤول) ويحدد المتطلبات الوظيفية وغير الوظيفية للمنصة. يعتمد التصميم المعماري نهجاً معيارياً يدمج إدارة المستخدمين والموارد والمشاريع التعاونية ومنتديات النقاش والأحداث العلمية.

يعتمد التنفيذ التقني على هندسة معمارية ثلاثية الطبقات باستخدام Django (الخلفية)، PostgreSQL (قاعدة البيانات)، وواجهة ويب سريعة الاستجابة متعددة اللغات (العربية/الإنجليزية). يحسن تكامل Elasticsearch من البحث في المحتوى العربي، بينما يوفر روبوت محادثة ذكي مبني على LangChain و LLaMA 3 مساعدة سياقية للمستخدمين.

تشمل الوظائف المطورة نظام مصادقة آمن مع التحقق عبر البريد الإلكتروني، ومحرك بحث متخصص للعربية، ومنتديات نقاش في الوقت الفعلي عبرWebSocket ، وإدارة المشاريع البحثية التعاونية، ولوحة تحكم إحصائية للمسؤولين.

تساهم هذه المنصة في هيكلة المجتمع العلمي لمعالجة اللغة العربية الآلية من خلال مركزة الموارد، وتسهيل التعاون متعدد التخصصات والجغرافي، وتعزيز تبادل المعرفة. تشمل آفاق التطوير المستقبلية دمج محرك توصيات ذكي وتطوير واجهة برمجة تطبيقات للتشغيل البيني مع أدوات المجال الأخرى

Table of content

1. Introduction: 7

2. State of the Art: 8

2.1. Existing Platforms: 8

2.2. Chatbots Based on Large Language Models: 9

3. Requirements Analysis and Architectural Design: 12

3.1. Introduction: 12

3.2. Identification des acteurs: 12

3.3. Needs analysis: 12

3.3.1. Functional Requirements: 12

3.3.2. Non-Functional Requirements: 13

3.4. Detailed Design 13

3.5. Use Case Diagram : 15

3.6. Class Diagram : 16

3.7. Sequence Diagram : 17

Chosen Scenario 17

4. **Presentation of the Practical Implementation** 18

4.1. Introduction 18

4.2. General Architecture and Technology Choices 18

4.2.1. General Technical Stack 18

4.2.2. Search Engine: 19

4.2.3. Chatbot "Chat AI" : 19

4.3. Development of the main features 21

4.3.1. Authentication system and role management 21

4.3.2. Resource management 21

4.3.3. Discussion forum and collaboration 22

4.3.4. Notification system 22

4.3.5. News management 22

4.3.6. Project Management 22

4.3.7. Event management 22

4.3.8. Management of Institutions 23

4.4. User Interface and User Experience 23

4.4.1. Responsive and multilingual design 23

4.4.2. Personalized Dashboard 23

4.5. Security and Data Protection 23

4.5.1. Implemented Security Mechanisms 23

4.5.2. Access Rights Management 23

4.6. Technical challenges and solutions 23

4.7. Scalability and perspectives 24

5. Conclusion: 25

List of Tables :

Table 1 : Comparison between existing platforms..9
Table 2 : Comparison between existing chatbots..11
Table 3 : Functional comparison between Elasticsearch and a traditional database..................19
Table 4 : The technological architecture of the intelligent chatbot system 20

List of Figures :

Figure 1 : Use Case Diagram 15
Figure 2 : Class Diagram...16
Figure 3 : Sequence Diagram...17
Figure 4 : Platform Homepage – Arabic Version 18
Figure 5 : Interface of the intelligent chatbot integrated into the collaborative platform..........21
Figure 6 : Resource Management...22

1. Introduction:

Arabic Natural Language Processing (Arabic NLP) is currently experiencing significant growth, notably due to recent advances in artificial intelligence, the massive increase in available textual data, and the growing interest in language-related technologies. Many applications such as voice assistants, machine translation, search engines, and educational tools now incorporate language processing modules. However, despite this enthusiasm, the Arabic language still lags behind other languages, particularly in terms of available corpora, suitable digital tools, and dedicated spaces for scientific collaboration. For example, although some recent initiatives—such as the Falcon Arabic model launched by Abu Dhabi's Advanced Technology Research Council—aim to better represent Arabic linguistic diversity by relying on high-quality datasets, the overall resources available for Arabic NLP remain limited [1][1].

In addition to this lack of resources, the Arabic language presents several major technical challenges. Its rich morphology, numerous dialects, and orthographic variations make the development of automatic processing tools even more complex [2][2]. One of the main obstacles to progress in Arabic NLP is also the lack of coordination among researchers, who are often scattered across different countries or disciplines, and the absence of a shared digital space to centralize knowledge. This situation hinders communication, slows down innovation, and makes it difficult to implement large-scale collaborative projects. In light of this, it becomes essential to create a digital environment that enables researchers, developers, and students to work together towards common goals.

The issue this project seeks to address is therefore the following: how can we design a digital platform capable of bringing together resources related to Arabic NLP, facilitating the exchange of ideas, overcoming geographical and disciplinary boundaries, and promoting effective collaboration between institutions and fields?

The proposed solution is the development of a collaborative web platform specifically dedicated to stakeholders in Arabic NLP in Algeria. This platform would offer two types of access: open access for consulting articles, courses, and open resources, and a restricted access for registered users, enabling them to participate in forum discussions, share tools, publish or enhance linguistic corpora, collaborate on research projects, and even interact with an integrated intelligent chatbot.

This project pursues several complementary objectives. First, it aims to centralize existing corpora, tools, and scientific publications to improve accessibility. It also seeks to strengthen exchanges among researchers from diverse backgrounds and to encourage new collaborations. Finally, it proposes a secure, user-friendly, and truly collaborative space that fosters the creation and sharing of high-quality resources.

Beyond the technical aspect, this project is part of a broader initiative: to help bring together members of the scientific community working on the Arabic language, and to contribute to the creation of an open, dynamic, and sustainable digital ecosystem for the advancement of Arabic NLP.

2. State of the Art:

2.1. Existing Platforms:

As part of the development of our collaborative platform aimed at researchers and developers in Arabic Natural Language Processing (NLP), it is important to examine a few existing platforms that share similar goals or offer comparable features. This comparison allows us to better position our project in relation to what already exists, to draw inspiration from best practices, and also to identify the gaps that our platform could fill.

- **ResearchGate** is presented as a widely used academic social network in the scientific community. It allows researchers to publish their work (articles, preprints, data), interact with peers, ask questions, and follow colleagues' activities. One of its major strengths is its vast international community, as well as the ease with which users can share publications and connect with other researchers. In 2024, ResearchGate had more than 20 million members [3][3]. However, this platform is rather generalist and remains poorly suited for specific languages like Arabic, especially in the field of Natural Language Processing (NLP).

- **Hugging Face**, is an open-source platform entirely dedicated to natural language processing (NLP). It offers a vast hub containing over 1.7 million machine learning models and more than 75,000 datasets [4][4]. The strength of Hugging Face lies in the diversity of its resources and the collaborative way they are organized. Users can find models, annotated datasets, powerful libraries like Transformers and Datasets, and even host applications. This makes access to NLP technologies easier, including for projects in the Arabic language. However, the platform is mainly aimed at users with a good level of programming skills and does not have a structure specifically designed for academic contexts or projects in the humanities.

- **CLARIN** (Common Language Resources and Technology Infrastructure) is a European infrastructure designed to facilitate access to linguistic resources in the humanities and social sciences. It offers tools such as the Virtual Language Observatory (VLO) and the Language Resource Switchboard, which allow researchers to locate and use datasets or tools according to their needs. CLARIN also has a network of certified centers spread across several European countries—over 30 centers in 2024 [5][5]. The focus is on interoperability and standardization. Despite its richness, CLARIN remains fairly complex to use, especially for researchers who are not familiar with the technical standards it imposes. Furthermore, it is mainly focused on European languages.

- **Masader** is a platform specifically developed for the Arabic language. It provides annotated datasets, lexicons, and various tools useful for Arabic NLP. It is one of the few platforms dedicated exclusively to this language. As of today, Masader hosts over 600 annotated datasets with more than 25 different attribute types, contributed by over 40 members of the community [6][6]. Its main strength lies in its specialization, which fills a real gap in the existing ecosystem. However, compared to platforms like Hugging Face or CLARIN, Masader is still limited in terms of community interaction, collaborative features, and international visibility.

Plateforme	Main features	Advantages	Disadvantages
ResearchGate	Academic social network: article publishing, exchanges between researchers, tracking citations and metrics	- Large scientific community - Easy networking - Publication tracking	- Focused only on scientific publications - No direct support for datasets or NLP tools
Hugging Face	Sharing NLP models, datasets, notebooks, inference APIs, model evaluation	- References in modern NLP - User-friendly interface for testing and sharing - Strong open-source community	- Less focused on the Arabic language - Difficult to use for non-coders
CLARIN	European infrastructure: access to linguistic resources, tools, certified centers	- Highly structured and academic - Supported by numerous institutions - Extensive multilingual resources	- Less accessible without university affiliation - Complexity in usage and navigation
Masader (ARBML)	Catalog of datasets for Arabic NLP, annotated with rich metadata	- Specialized in the Arabic language - Detailed metadata - Open source and rapidly evolving	- Does not offer a collaborative platform (exchange/discussion)

Table 1 : Comparison between existing platforms

2.2. Chatbots Based on Large Language Models:

In recent years, large language models (LLMs) have completely transformed the way chatbots are designed. An LLM is a deep learning model trained on vast amounts of text, capable of understanding and generating natural language. When given a prompt (a question or an instruction), it produces the most probable continuation of the text, allowing it to simulate a genuine human conversation. Chatbots based on this type of model are now highly popular, especially for their ability to understand context, accurately rephrase questions, and provide detailed answers—even on complex topics.

At the core of these models lies the Transformer architecture, introduced in 2017, which marked a true breakthrough in the field of natural language processing [7][7]. What makes it unique is its self-attention mechanism, which enables the model to identify the most important words in a sentence regardless of their position. Unlike sequential models (which process words one by one), Transformers analyze sequences in parallel, improving their efficiency and ability to understand long and complex sentences. This architecture has become the foundation for well-known models such as BERT and GPT.

Today, several solutions rely on these advancements, whether proprietary (cloud-based) or open source.

- **ChatGPT**, developed by OpenAI, is based on very large models from the GPT (Generative Pretrained Transformer) family. The ChatGPT-4 version, built on GPT-4, uses a multilingual and multimodal Transformer architecture. Although OpenAI has not officially disclosed the exact number of parameters for GPT-4, some estimates place this figure around 100 trillion parameters for the full version (GPT-4 with Mixture of Experts) [8][8]. This model was trained on a vast corpus sourced from the web (code, scientific articles, conversations, etc.), enabling it to maintain long dialogues, understand nuances, and respond contextually—even to complex questions.

  The ChatGPT API also allows integration of business plugins (code analysis, image generation, web search, etc.). However, this solution has limitations: it requires a constant internet connection, usage costs can be high (depending on API use), customization remains limited, and all data passes through OpenAI's servers, raising privacy concerns, especially in sensitive or institutional contexts.

- **Gemini**, developed by Google DeepMind, is Google's next-generation multimodal large language model, succeeding the Bard chatbot. Gemini 1.5 Pro has been announced as an enhanced version [9][9]. This model is based on an advanced Transformer architecture and is capable of processing multiple data modalities: text, image, audio, video, and code. One of Gemini's distinctive features is its native access to Google Search, allowing it to provide real-time, up-to-date answers. It also integrates with the Google Workspace ecosystem (Docs, Sheets, Gmail, etc.), making it particularly appealing for collaborative professional use. However, like ChatGPT, Gemini operates exclusively via a cloud API, which requires a constant internet connection, dependence on Google's servers, and limitations in advanced business customization. Furthermore, Google's data processing raises privacy and data sovereignty concerns, especially for institutional or European users.

- **Claude**, developed by the startup Anthropic, is a large language model designed with a strong focus on safety, ethics, and the robustness of responses. Claude 3 Opus, the most advanced model to date, is capable of competing with GPT-4 and Gemini 1.5 Pro in terms of performance [10][10]. Claude is trained to uphold principles of harmlessness, helpfulness, and honesty, relying on a strict policy of not retaining user data, making it an attractive choice for sensitive sectors such as healthcare, finance, and public administration. It operates exclusively via a cloud API with enhanced security controls but cannot be deployed locally (on-premise). This model is still paid and offers limited customization in its public version.

Plateforme	Main features	Advantages	Disadvantages
ChatGPT	- Long and nuanced conversations - Cloud REST API - Business plugins	- Very high dialogue quality - Rich ecosystem - Regular updates	- High recurring cost - Cloud dependency - Limited customization - Privacy concerns
Gemini	- Multimodal dialogue (text, images, code) - Real-time Search access - Cloud API	- Google Workspace integration - Document summarization - Updated information	- Limited customization - Cloud dependency - Data privacy and sovereignty concerns
Claude	- Enhanced security - Ethical alignment - Control over data non-retention	- Strict privacy policy - Reduction of problematic responses - Suitable for sensitive sectors	- Cloud dependency - Subscription cost - Limited customization - No on-premise deployment

Table 2 : Comparison between existing chatbots

3. Requirements Analysis and Architectural Design:

3.1. Introduction:

The development of our collaborative platform starts from a clear objective: to support the active community working on Arabic Natural Language Processing (NLP). This community, composed of researchers, educators, advanced students, and developers, faces several daily challenges. Among the most significant are the lack of access to specialized linguistic resources, insufficient communication between members working on related topics, and especially the absence of a centralized space to share corpora, tools, or to disseminate scientific events.

Our platform was designed to meet these needs. It aims to be a collaborative digital environment that is secure, intuitive, and feature-rich. The idea is to encourage exchanges, facilitate knowledge sharing, and stimulate research in this field by providing a common space where all stakeholders in Arabic NLP can come together, interact, and collaborate more easily.

3.2. Identification of the actors:

The use of the platform is based on three main user profiles, each with clearly defined roles and permissions.

- **The public user** is a person who accesses the platform without having an account. Their access is limited: they can only view certain sections, such as shared resources or scientific events, but cannot interact with content or participate in discussions.

- **The registered user** has validated their email address after signing up. They have full access to the platform. They can share resources, participate in forum discussions, register for events, edit their profile, and freely interact with other members.

- **The administrator** has the same rights as a registered user, but also holds administrative privileges: they can manage users (add, delete), moderate content, create or modify events, view overall statistics, and ensure the smooth operation of the entire platform.

3.3. Needs analysis:

3.3.1. Functional Requirements:

To meet the expectations of the scientific community in Arabic NLP, the platform was designed around several essential features:

- **Sharing of linguistic resources**: the platform allows the publication of corpora, articles, theses, dissertations, as well as tools and software libraries. Each resource is accompanied by metadata (title, author, language, type, format, etc.) to facilitate searching and filtering.
- **Advanced user management:** a CustomUser model is employed, including fields such as research specialty, institution, country, and profile photo. Registration requires email verification via a confirmation code, and administrators have special privileges. Institutions are also managed, with the ability to filter users by specialty or location.

- **Forum system:** a structured discussion space organized by topics and subtopics (chatrooms) is implemented. Conversations happen in real-time via WebSocket, featuring an interface similar to messaging apps like WhatsApp. Each user can edit their own messages.
- **Scientific events management:** users can publish or register for conferences, seminars, and workshops. This feature strengthens community engagement and encourages networking opportunities.
- **Statistical tracking**: an administrator dashboard enables monitoring of platform activity (new registrations, shared resources, forum participation, etc.). An additional module displays member statuses (online, offline, busy) to better understand activity periods.
- **Collaborative project management:** the platform allows each registered user to create and manage collaborative projects. Each project features a dedicated space including a descriptive profile (title, objectives, and dates), a shared file library, an internal discussion thread, and role management (coordinator, contributor, reader). This functionality promotes structured teamwork, enhances coordination among researchers, and enables clear tracking of project progress.

3.3.2. Non-Functional Requirements:

Beyond the functionalities, the platform must also meet technical and accessibility requirements:

- **Account security:** all users must verify their email address with a code sent during registration to ensure the authenticity of their profile.
- **Multilingual accessibility:** the platform is designed to be accessible in both Arabic and English. By default, the interface is in English, but each user can choose their preferred language in the settings.

3.4. Detailed Design

The platform is designed to be modular and scalable, with each major feature organized into independent modules. This facilitates code maintenance, makes the overall structure clearer, and allows new features to be easily added later.

- **The user management system** uses a custom model that includes additional information such as research specialty, affiliated institution, country, and a profile picture. Users must verify their email address via a unique code sent by email. Regular users and administrators are distinguished, with the latter having additional rights to manage the platform. Additionally, a dedicated module manages institutions with their specialties and locations, making it easier to search for members or partners based on various criteria.

- **The resource management system** allows browsing and searching linguistic resources with comprehensive metadata. Users can freely publish their documents within a structure designed to facilitate indexing. The interface has been designed to make access to resources simple and fast, even for those who are not registered.

- **The project management system** allows users to create, organize, and track their collaborative projects. Each project can contain tasks, deadlines, and associated members, which facilitates coordination and teamwork. Notifications and progress tracking help keep everyone informed and up to date.

- **The instant messaging system** is organized around thematic discussions called topics, which are further divided into more specific sub-discussion spaces called chatrooms. Exchanges are updated in real-time without page reloads. The interface resembles a classic messaging app, with a clear distinction between sent and received messages, the names and photos of participants, and the ability to edit one's own messages.

- **The scientific events management system** is a specific module that handles the creation, registration, and viewing of events. Each important action, such as creating an event or a new registration, triggers a notification visible to the concerned user, ensuring effective activity tracking. This module can manage different types of events, allowing for great flexibility.

- **Integrated intelligent chatbot:** The platform includes a conversational assistant capable of answering user questions in natural language. It can extract information from documents, query the database to provide precise answers, or perform web searches. The chatbot automatically understands the user's intent and selects the most appropriate action. It displays responses seamlessly in a real-time chat interface. This module facilitates quick access to knowledge and enhances the overall user experience.

- **A statistics and history system** automatically collects data on platform usage: number of registrations, volume of resources added, activity in discussions, event registrations, etc. This information is displayed in dynamic dashboards accessible only to administrators. Special monitoring of users' online statuses also helps better understand community engagement and tailor communication efforts.

Overall, each part of the platform is designed to evolve independently, which paves the way for adding new features in the future, such as a recommendation engine or an automatic annotation system.

3.5. Use Case Diagram :

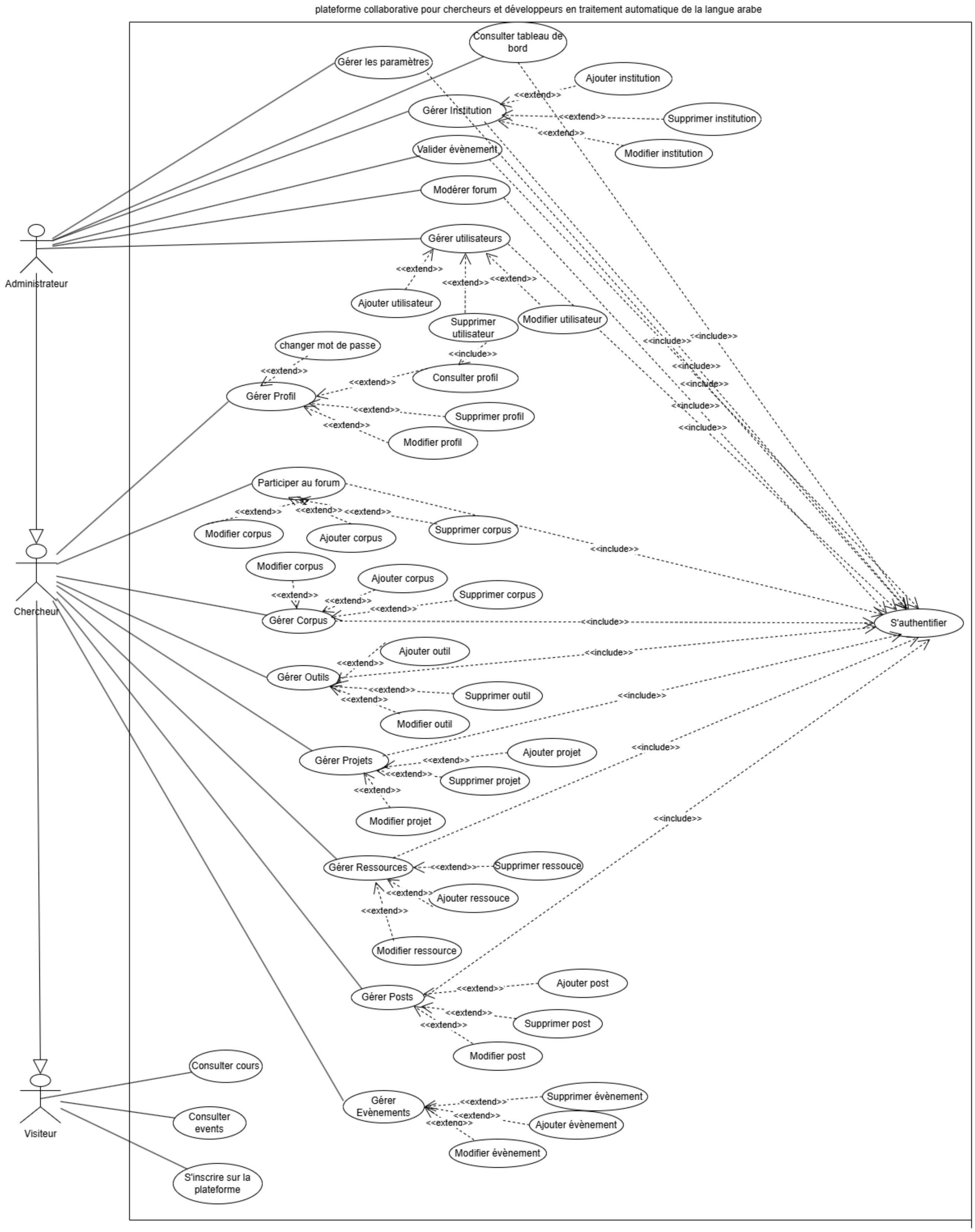


Figure 1 : Use Case Diagram

3.6. Class Diagram:

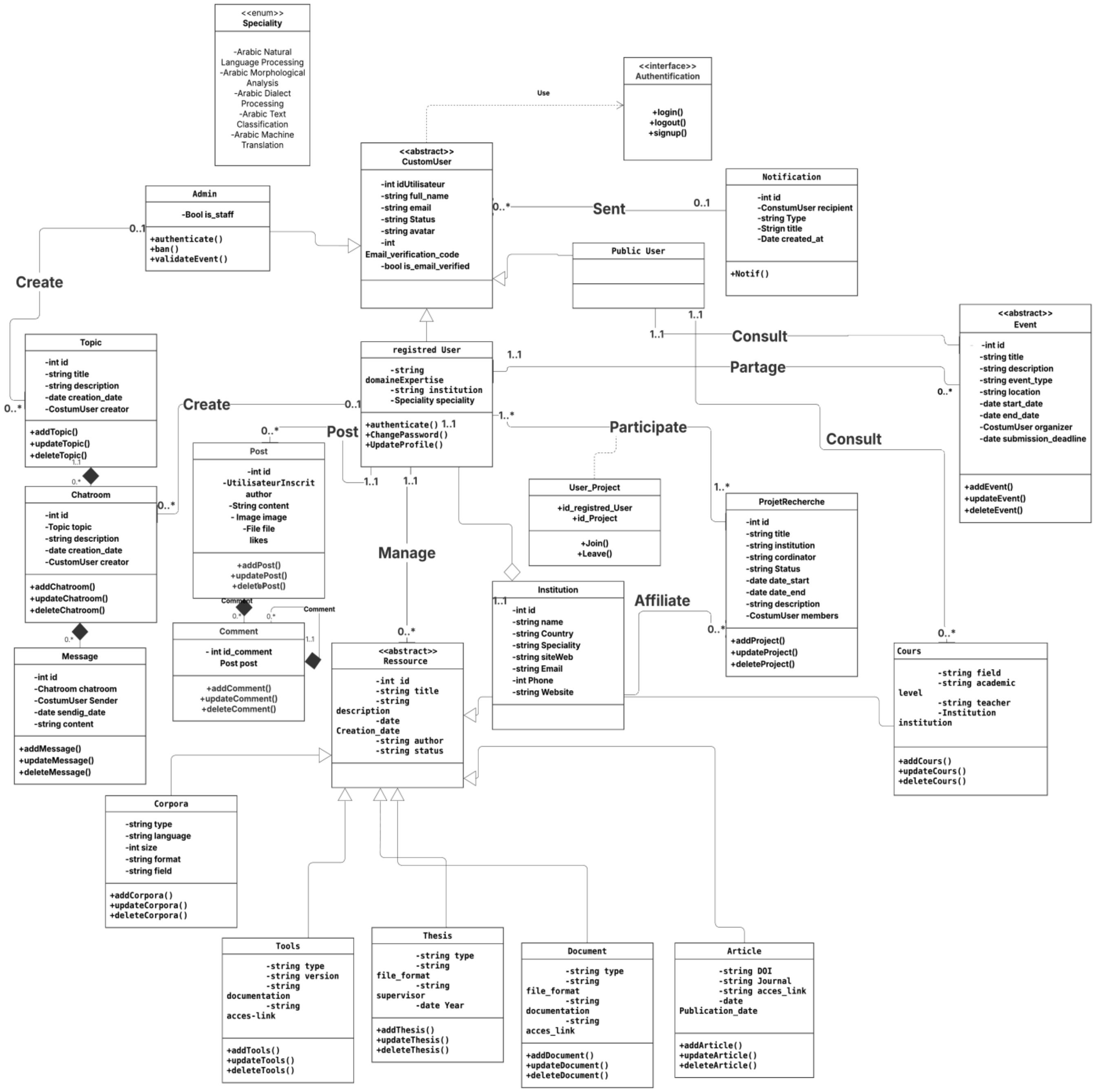


Figure 2 : Class Diagram

3.7. Sequence Diagram :

Chosen Scenario

As part of our collaborative platform dedicated to the automatic processing of the Arabic language, we have chosen to illustrate a representative and cross-functional user scenario. The selected user journey is as follows: A new user registers on the platform, confirms their email via a verification code, accesses a chatroom within a forum to interact with the community, and then creates a research project.

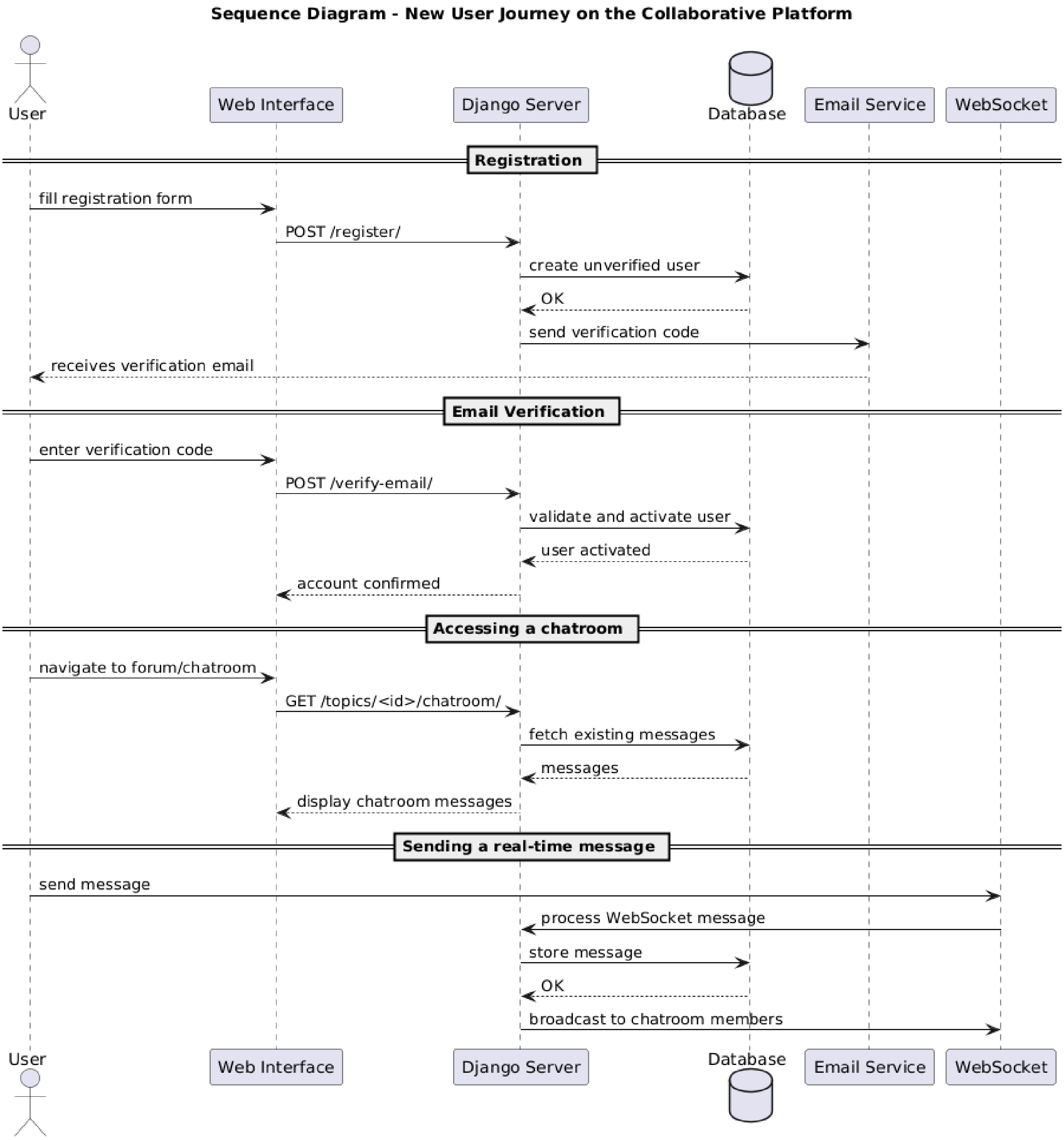
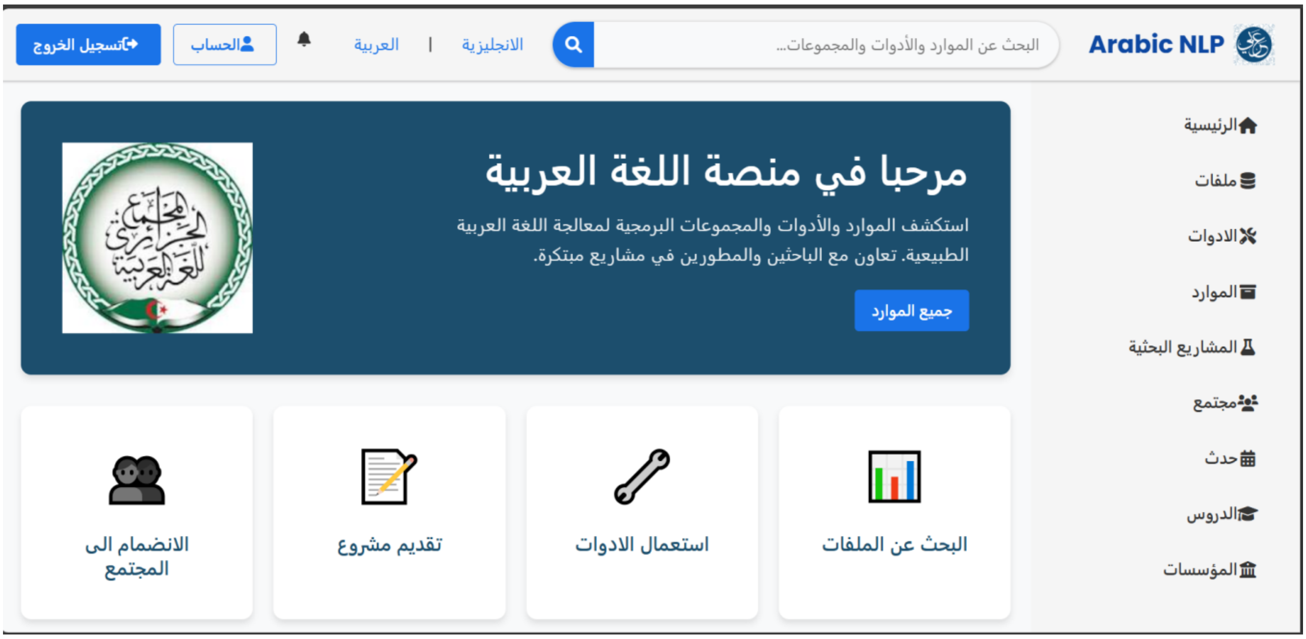


Figure 3 : Sequence Diagram

4. Presentation of the Practical Implementation

4.1. Introduction

To develop our collaborative platform dedicated to Arabic Natural Language Processing, a solid and adaptable architecture was required. In this chapter, I will explain the choices made during development, why they were made, and which features were implemented. The goal was to rely on modern and reliable tools to effectively meet the specific needs related to the Arabic language, while providing a smooth and enjoyable user experience.



مرحبا في منصة اللغة العربية

استكشف الموارد والأدوات والمجموعات البرمجية لمعالجة اللغة العربية الطبيعية. تعاون مع الباحثين والمطورين في مشاريع مبتكرة.

Figure 4 : Platform Homepage – Arabic Version

4.2. General Architecture and Technology Choices

4.2.1. General Technical Stack:

Our platform is based on a three-layer architecture that combines modern and reliable tools.

- **Backend:**
  For the server side, we chose a Python framework that is very popular and well-known for its robustness. This choice came quite naturally because it offers many built-in features, such as user management, administration, and forms [11][11]. Its clear structure helps write clean and maintainable code, which also speeds up development. Additionally, it simplifies database management through a system that prevents common errors and protects against certain security risks.
- **Frontend :**

  For the user interface, we opted for a classic yet effective solution: HTML, CSS, and JavaScript. These three technologies are quite flexible and allow the creation of an interface that is simple, aesthetically pleasing, and interactive. HTML structures the content, CSS handles the design and responsiveness for different screens, and JavaScript makes everything dynamic, for example for forms or filters. This foundation also makes it easy to add external libraries to enhance the visual rendering.

- **Database :**

  To store data, we use PostgreSQL, which is a very robust system well-suited to our needs. It handles Unicode very well, which is important for the Arabic language [12][12], and offers advanced features such as full-text search and storage of data in JSON format. It remains performant even with a large volume of data.

4.2.2. Search Engine:

For the search engine, we chose Elasticsearch, which truly became a key element in the project. This engine is very well suited for the Arabic language, notably thanks to built-in tools that properly analyze Arabic texts [13][13]. It handles word variations very well (such as derived forms or synonyms), which greatly improves search accuracy. Additionally, it is very fast, even when performing complex searches over large volumes of data. And since it can easily scale to higher workloads, it will be able to keep up with the platform's growth without any problem.

Functionality	Elasticsearch	Classic Database	Why is this important?
Arabic support	✓ Native support for Arabic management	✗ Limited Arabic support	Essential for our platform, which primarily handles texts in Arabic.
Search speed	✓ Very fast (< 1s)	✗ Slow for text search	Allows users to get results instantly.
Advanced search	✓ Search by synonyms and similarity	✗ Basic search only	Provides a better search experience for researchers
Performance with large volumes	✓ Excellent with millions of documents	✗ Slowdown with large volume	Ensures the future scalability of the platform.

Table 3 : Functional comparison between Elasticsearch and a traditional database

4.2.3. Chatbot "Chat AI" :

The chatbot on our platform, which we named Chat AI, plays a central role in interacting with users. It acts as a single entry point: it understands what the user is trying to do (analyze a PDF, query the database, perform a web search, etc.) and automatically triggers the appropriate tools to provide precise and tailored responses. Thanks to this, the user doesn't need to navigate through multiple menus or modules; everything is done through a single smooth interface.

To develop this chatbot, we relied on several technologies, each with a well-defined role, making the whole system modular, efficient, and easy to scale.

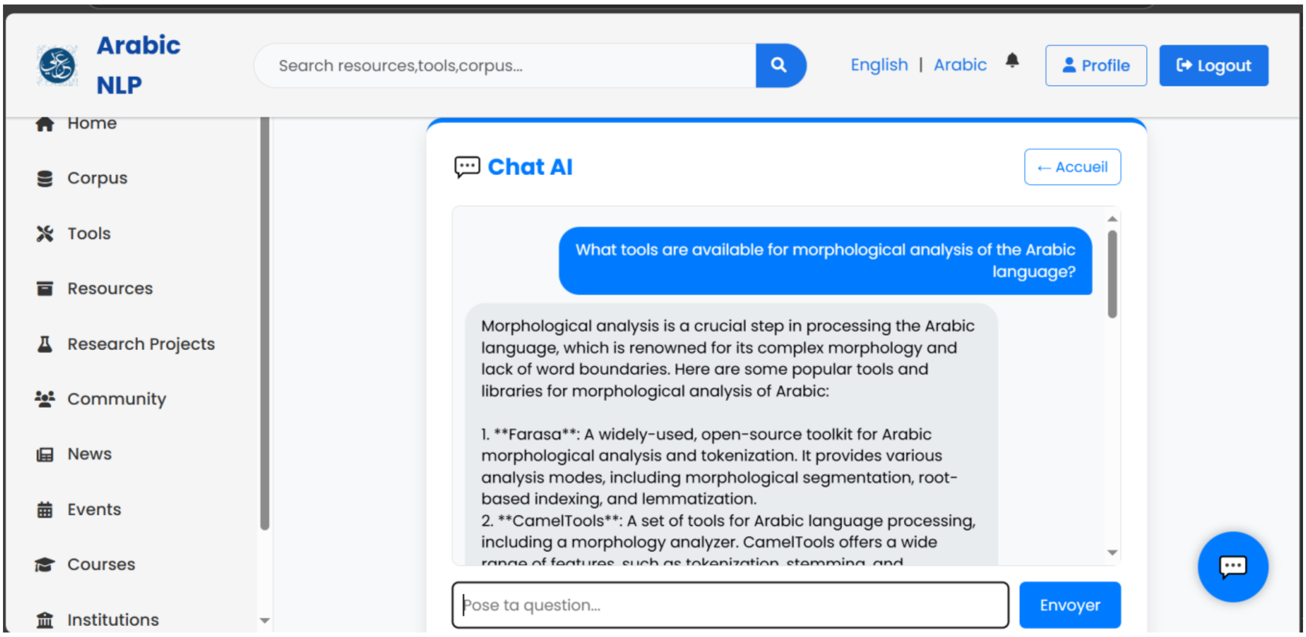
Here is an overview of the main technologies used and their purposes:

Technology	Role	Example of use
FastAPI	Asynchronous API framework	Handling HTTP calls /ask/, /ask_db/, etc.
Pydantic	Schema validation	Validation of JSON payloads
LangChain	Prompt orchestration and context management	Validation of JSON payloads
ChatGroq (LLaMA 3 8B)	Text generation model	Generating conversational responses
HuggingFaceEmbeddings + FAISS	Semantic encoding and vector search	Similarity search in PDFs, web, databases
SQLAlchemy + PostgreSQL	ORM and relational database	Executing dynamic SQL queries
Django + WebSocket	Real-time user interface	Live chat via WebSocket in the web interface
Requests + BeautifulSoup	Real-time user interface	Scraping snippets from DuckDuckGo

Table 4 : The technological architecture of the intelligent chatbot system

This system enables quite a few useful functions. For example, when a user asks a question based on a PDF document, the chatbot can automatically extract the important passages using vector search, then generate a clear summary with the language model. If the question concerns stored data, it converts the natural language sentence into an SQL query, executes that query, and then explains the result in simple terms. For web searches, it retrieves relevant content from the Internet, cleans it up, and creates a concise summary.

All of this is made possible thanks to an automatic classification system that detects what the user wants to do and directs the request to the appropriate "agent" [14][14], without it being visible to the user. And since all responses are returned in JSON, they integrate easily into our web interface, with the possibility to further enhance the display (charts, conversation history, etc.).



What tools are available for morphological analysis of the Arabic language?

Morphological analysis is a crucial step in processing the Arabic language, which is renowned for its complex morphology and lack of word boundaries. Here are some popular tools and libraries for morphological analysis of Arabic:

1. **Farasa**: A widely-used, open-source toolkit for Arabic morphological analysis and tokenization. It provides various analysis modes, including morphological segmentation, root-based indexing, and lemmatization.
2. **CamelTools**: A set of tools for Arabic language processing, including a morphology analyzer. CamelTools offers a wide range of features such as tokenization, stemming, and

Figure 5 : Interface of the intelligent chatbot integrated into the collaborative platform

4.3. Development of the main features

4.3.1. Authentication system and role management

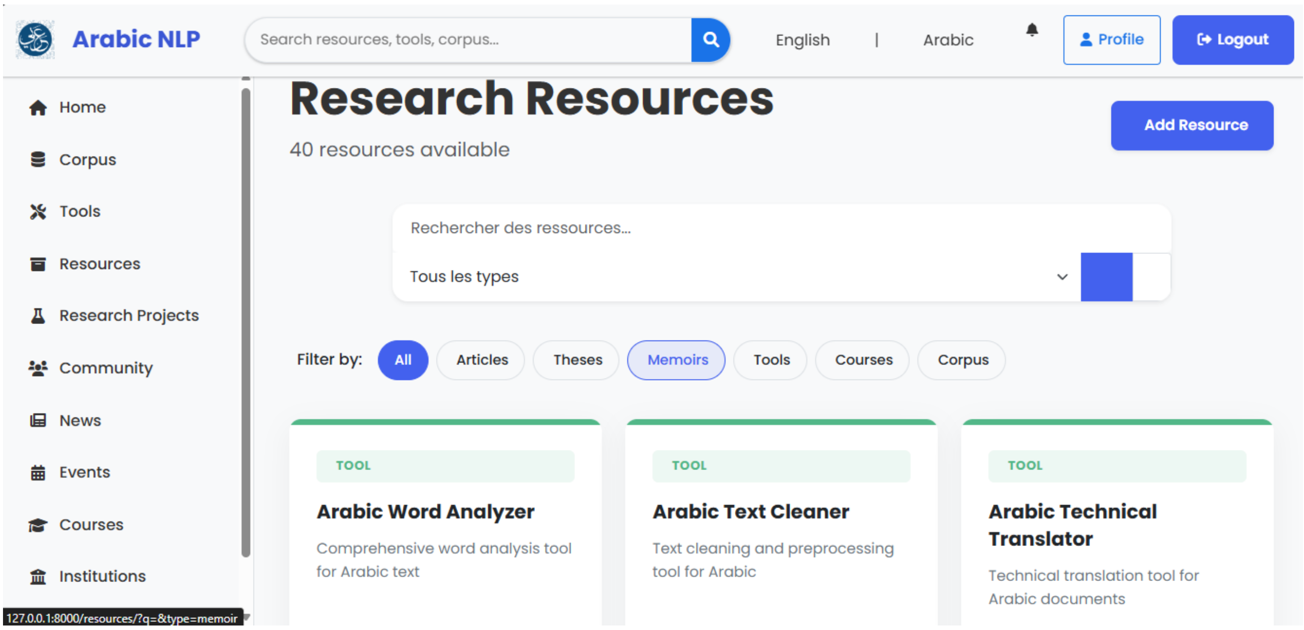
To manage access to the platform, we implemented an authentication system with three user levels. At the top of the hierarchy, the administrator has full access: they can manage users and moderate content. Next are the registered researchers, who can access the main features such as resources, the forum, and news. Finally, unregistered visitors have limited access to certain public information. The idea is to encourage them to register in order to fully benefit from the platform.

4.3.2. Resource management

The "Resources" section is somewhat the documentary core of the platform. It is organized into four main categories:

- **NLP Tools**: here, you will find tools for the analysis and automatic processing of the Arabic language, each with its name, a description, a link, and the contributor's name.
- **Courses**: this section gathers educational resources (course materials, tutorials, etc.) shared by teachers or researchers.
- **Linguistic corpora**: it gathers links to Arabic language datasets, with information such as the corpus size, its application domain, etc.
- **Academic documents**: this section references theses, articles, and dissertations, including essential metadata.

Users can add new resources, view details, directly access content, and manage their own contributions. The goal is to centralize access to resources without hosting them ourselves.



Research Resources

40 resources available

Arabic Word Analyzer

Comprehensive word analysis tool for Arabic text

Arabic Text Cleaner

Text cleaning and preprocessing tool for Arabic

Arabic Technical Translator

Technical translation tool for Arabic documents

Figure 6 : Resource Management

4.3.3. Discussion forum and collaboration

The forum is designed to encourage exchanges between researchers. It is organized by themes, which makes discussions clearer and more focused. Each topic can receive replies in a structured way, making it easier to follow the conversation. Real-time notifications are sent to keep users informed. It is also possible to share documents directly within discussions, which truly enriches the exchanges.

4.3.4. Notification system

The notification system allows users to stay informed about everything concerning them on the platform. Whenever there is a reply to a message, a new resource shared, or a mention in a discussion, a notification appears. Users can easily see what has been read or not and review the entire history. It is also possible to customize notifications to receive only what is relevant according to their preferences.

4.3.5. News management

The "News" section allows the dissemination of the latest information to the community. Administrators can publish articles categorized by theme. News items are displayed in chronological order, making it easy to follow updates. Each article can receive comments to encourage discussions. Users can also share these news posts on social media, and an archiving system enables retrieval of past publications through filters (date, category, etc.).

4.3.6. Project Management

In the "Projects" section, researchers can showcase their ongoing or completed work in the field of automatic processing of the Arabic language. It includes the objectives, team members, progress, etc. This is a great way to give visibility to projects, as well as to create connections between teams working on similar topics.

4.3.7. Event management

The "Events" section gathers information about conferences, workshops, seminars, or calls for papers related to the Arabic language. Each event is presented with its date, location, theme, and participation details. This is convenient for researchers who want to plan their attendance at these events.

4.3.8. Management of Institutions

Finally, the "Institutions" section functions as a directory of research or educational establishments, including universities, laboratories, and research centers. It is not limited to structures specialized in Arabic language processing but covers all disciplines. Each institution is described with its departments, laboratories, research areas, training programs, and contact details. This allows users to better understand the academic landscape and identify collaboration opportunities.

4.4. User Interface and User Experience

4.4.1. Responsive and multilingual design

To make the platform accessible to everyone, regardless of the device used, we designed a responsive interface: it automatically adapts to screens of all sizes, whether it’s a computer, tablet, or smartphone. On the language side, we integrated a multilingual system supporting Arabic and English, the platform’s two main languages. We also paid close attention to text direction: the display adjusts automatically so that the content appears correctly, whether in RTL mode (right-to-left for Arabic) or LTR mode (left-to-right for English).

4.4.2. Personalized Dashboard

The dashboard changes according to the type of logged-in user. For example, a researcher will see their ongoing projects, favorite resources, and notifications related to their activities directly. On the other hand, an administrator has a more global view: they can monitor users, moderate content, and access platform usage statistics. The idea is that everyone can quickly access the information that is truly useful to them.

4.5. Security and Data Protection

4.5.1. Implemented Security Mechanisms

To ensure the security of our platform, we have implemented several basic but essential protections. For example, we use CSRF tokens to prevent cross-site request forgery attacks, and all user inputs are sanitized to prevent XSS attacks (injection of malicious scripts).

4.5.2. Access Rights Management

On the access side, everything is tightly controlled. Only verified accounts can access certain features, and permissions are managed according to the user’s role. This ensures precise control over who can do what, while keeping the platform secure.

4.6. Technical challenges and solutions

During the development of the platform, we encountered several technical challenges that had to be overcome. One of the biggest was integrating **Elasticsearch** as the search engine, especially to properly handle the Arabic language. We had to configure specific analyzers and tokenizers so that searches would be relevant and adapted to the particularities of Arabic.

Multilingualism, especially the integration of Arabic, also required a lot of attention. It was necessary to ensure that characters display correctly and that the interface properly respects the RTL (Right-to-Left) reading direction. To make Elasticsearch handle Arabic well, we used analyzers like the "Arabic analyzer" and configured custom filters to normalize the characters.

Access management and data security were another major challenge. We implemented a robust authentication and authorization system that restricts access to resources based on whether the user has verified their account or not. This allows us to maintain control over who can access what.

4.7. Scalability and perspectives

The platform was designed to be scalable. Thanks to a modular architecture, it is easy to add new features or even adapt it to other languages in the future. The user interface was also built flexibly, allowing the integration of new elements without breaking what already exists.

5. Conclusion:

The development of this collaborative platform truly meets the needs of the scientific community working on the automatic processing of the Arabic language. By gathering all linguistic resources in one place, facilitating exchanges between researchers, and encouraging the sharing of knowledge and tools, we provide a real solution to a significant gap in the field of Arabic NLP.

The platform is built on a solid foundation with Django and PostgreSQL, and it offers several key features: a secure authentication system with email verification, real-time messaging via WebSocket, a dedicated resources section, a module to manage scientific events, and a dashboard to monitor community activity. All of this has been designed to provide a simple interface, adaptable to all screen sizes, and respectful of data privacy.

This project allowed me to put into practice the skills I acquired in software engineering, web architecture design, database management, and human-computer interaction. It also highlights the importance of developing technological solutions tailored to the specificities of a language and a research community, taking into account their linguistic, cultural, and scientific particularities.

Beyond the technical aspect, this platform creates a unifying digital space that can evolve and expand over time. Among the possible future improvements, we can envision integrating an intelligent recommendation engine based on user profiles, implementing a collaborative corpus annotation system, or even opening an API to connect the platform with other NLP tools.

In summary, this project is part of an effort to promote and modernize Arabic Natural Language Processing. It constitutes a first step towards creating a sustainable, inclusive, and collaborative digital environment that serves the linguistic and scientific development of the Arab world.

References Table:

[1] [1] Reuters. "UAE launches Arabic language AI model as Gulf race gathers pace." 21 mai 2025.

[2] [2] Habash, Nizar. "A Panoramic Survey of Natural Language Processing in the Arab World." Communications of the ACM, vol. 62, no. 3, 2019.

[3] [3] ResearchGate Statistics and User Count for 2024. DMR.

[4] [4] Every Hugging Face Statistics You Need to Know (2024) - Weam AI.

[5] [5] **Hinrichs, E., & Krauwer, S.** (2014). The CLARIN research infrastructure: Resources and tools for eHumanities scholars. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation* (pp. 1525-1531).

[6] [6] AlKhamissi, H., AlJallaf, C., & Elsayed, T. (2021). **Masader: Metadata Sourcing for Arabic Datasets**. *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, 215–223. Association for Computational Linguistics. https://aclanthology.org/2021.wanlp-1.25/. Accessed in April 2025

[7] [7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). **Attention is All You Need**. In *Advances in Neural Information Processing Systems* (NeurIPS 2017), 30. https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html. Accessed in April 2025

[8] [8] Exploding Topics. (n.d.). *Number of Parameters in GPT-4 (Latest Data)*. https://explodingtopics.com/blog/gpt-parameters. Accessed in April 2025

[9] [9] Google DeepMind. (2024). *Gemini 1.5 Technical Report*. https://deepmind.google/technologies/gemini/. Accessed in April 2025

[10] [10] Anthropic. (2024). *Introducing the Claude 3 model family*. https://www.anthropic.com/news/claude-3. Accessed in April 2025

[11] [11] Django Software Foundation. (2024). *Django Documentation - The Web framework for perfectionists with deadlines*. https://docs.djangoproject.com/. Accessed in April 2025

[12] [12] PostgreSQL Global Development Group. (2024). *PostgreSQL Documentation - The World's Most Advanced Open Source Relational Database*. https://www.postgresql.org/docs/. Accessed in April 2025

[13] [13] GeeksforGeeks. (n.d.). *Scaling Elasticsearch Horizontally: Understanding Index Sharding and Replication*. https://www.geeksforgeeks.org/scaling-elasticsearch-horizontally-understanding-index-sharding-and-replication/. Accessed in April 2025

[14] [14] IBM. (n.d.). *What is intent classification?*. https://www.ibm.com/topics/intent-classification. Accessed in April 2025