
Software Requirements Specification

for

Basmagly

Version 3.0 approved

Prepared by

Mohamed Hatem 202001235

Kareem Elozeiri 202000896

University of Science and Technology at Zewail City

15/1/2024

Table of Contents

1. Introduction	3
1.1 Purpose	3
1.2 Document Conventions	3
1.3 Intended Audience and Reading Suggestions	3
1.4 Project Scope	3
1.5 References	4
2. Overall Description	4
2.1 Product Perspective	4
2.2 Product Features (Major Features)	5
2.3 User Classes and Characteristics	5
Login FR:	5
Chatting FR:	5
2.4 Operating Environment	6
3. System Features	7
3.1 Document Management	7
3.1.1 Description and Priority:	7
3.1.2 Stimulus/Response Sequences:	7
3.1.3 Functional Requirements	7
3.2 User Authentication and Profile Management	8
3.2.1 Description and Priority	8
3.2.2 Stimulus/Response Sequences:	8
3.2.3 Functional Requirements	8
3.3 Chat-Based Document Querying and Q&A	9
3.3.1 Description and Priority	9
3.3.2 Stimulus/Response Sequences	9
3.3.3 Functional Requirements	9
3.4 Quiz and Summary Generation	9
3.4.1 Description and Priority	9
3.4.2 Stimulus/Response Sequences	10
3.4.3 Functional Requirements	10
4. External Interface Requirements	10
4.1 User Interfaces	10
4.2 Software Interfaces	14
5. Functional Requirements:	15
6. Other Nonfunctional Requirements	16
6.1 Performance Requirements	16
6.2 Software Quality Attributes	16

Revision History

Name	Date	Reason For Changes	Version
Initial Version	16/11	To start designing the project	1.0
Second Version	10/12	To develop 50% of the requirements	2.0
Last Version	15/12	To develop the whole requirements	3.0

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) outlines the requirements for the Student Study Assistant Website “Basmagly” (Release 1.0). The product provides a platform for students to enhance their study experience by uploading documents and leveraging features like question-answer generation, quizzes, summaries, and context-aware chats.

1.2 Document Conventions

This document follows standard conventions for formatting and prioritization:

- Fonts used are Times New Roman, size 12, for main content, with headings emphasized in bold.
- Priorities are explicitly assigned to each requirement statement where applicable, rather than assuming inheritance of priorities from higher-level requirements.
- Functional requirements are numbered sequentially (e.g., FR-1, FR-2) for clear reference, while non-functional requirements are labeled as NFR-1, NFR-2, etc.
- Any technical terms or abbreviations are defined in a glossary section to ensure clarity for all readers.

1.3 Intended Audience and Reading Suggestions

This document is intended for the following audiences:

- Developers: To understand detailed system requirements and implementation guidelines.
- Project Managers: To oversee the scope, timeline, and prioritization of requirements.
- Testers: To develop test cases based on functional and non-functional requirements.
- Marketing Staff: To understand the system’s features for promotional materials.
- End Users: To review the key functionality and features that the system will provide.

1.4 Project Scope

The product provides a platform for students to enhance their study experience by uploading documents and leveraging features like question-answer generation, quizzes, summaries, and

context-aware chats. the full scope of the system, including both frontend (React-based) and backend (Django-based) components, as well as the integration of key technologies like Hugging Face models and LangChain for natural language processing and orchestration. The document addresses all functional and non-functional requirements necessary for implementing and deploying the complete system.

1.5 References

The following documents and resources:

- **Hugging Face Documentation:**

Title: Hugging Face Transformers Documentation

Author: Hugging Face Team

Version: Latest

Date: N/A

Source: <https://huggingface.co/docs>

- **LangChain Documentation:**

Title: LangChain Official Documentation

Author: LangChain Developers

Version: Latest

Date: N/A

Source: <https://docs.langchain.com>

- **React and Django Documentation:**

Title: React and Django Frameworks Documentation

Authors: Meta Team (React), Django Developers

Version: Latest

Date: N/A

Source:

React: <https://reactjs.org/docs>

Django: <https://Django.palletsprojects.com>

2. Overall Description

2.1 Product Perspective

The **Basmagly** is a new, self-contained product designed to enhance students' study efficiency by offering AI-powered features like document-based Q&A, summarization, and quiz generation. It is not part of any existing product family or a replacement for a prior system. The product is entirely standalone and interacts with external systems (e.g., APIs from Hugging Face and LangChain) for advanced NLP functionalities.

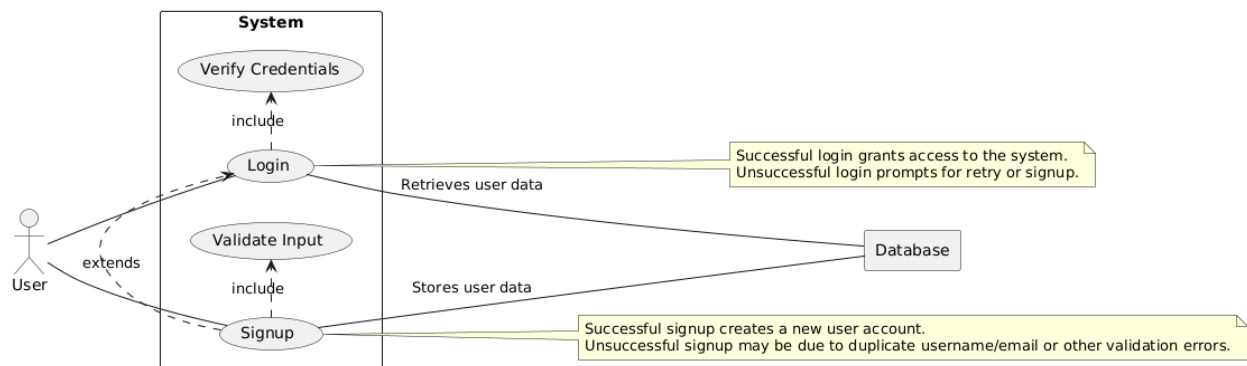
2.2 Product Features (Major Features)

The product offers the following high-level features:

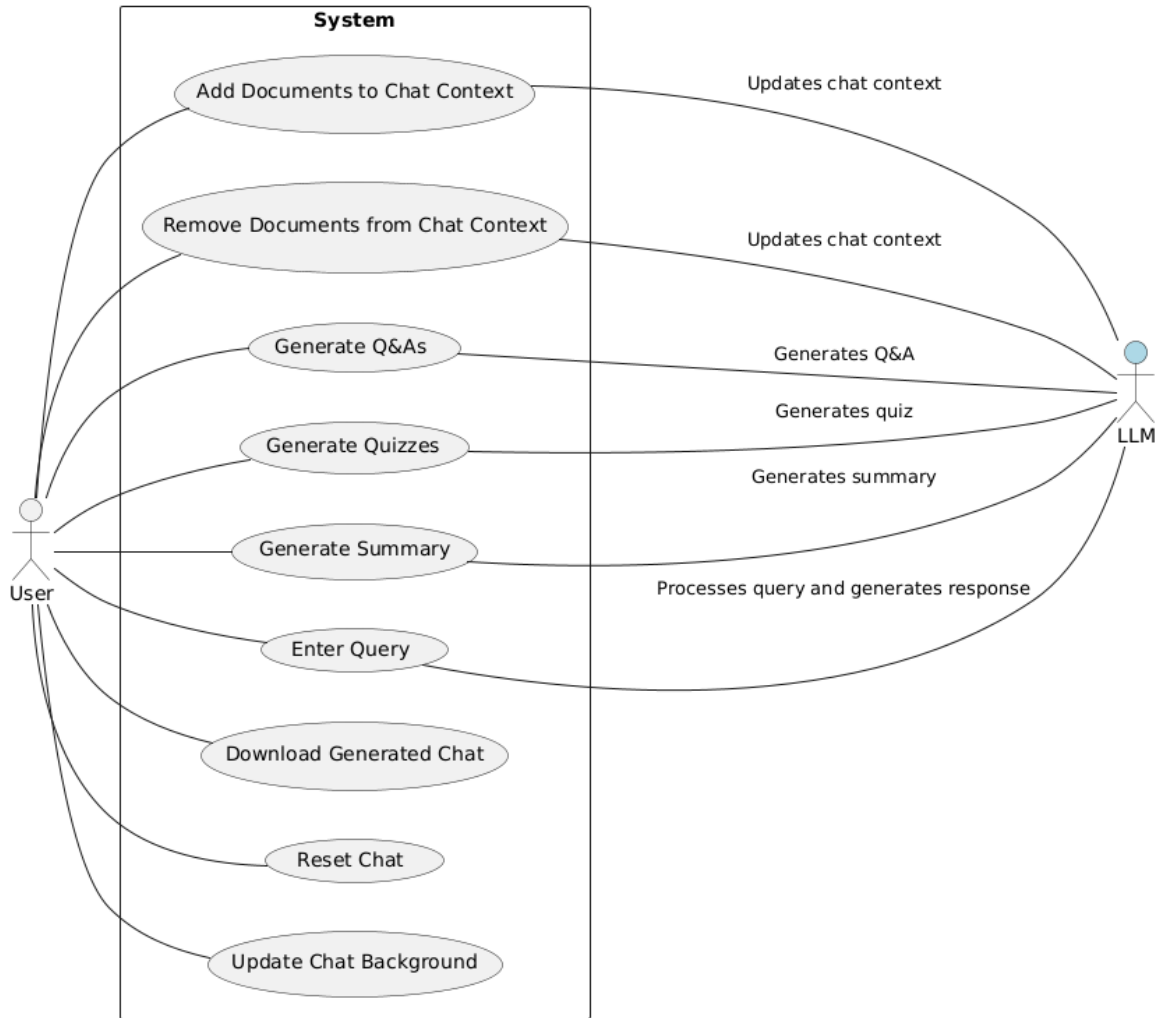
- **Document Management:** Upload, download, and delete documents.
- **AI-Powered Q&A:** Answer user queries based on uploaded document content.
- **Summarization:** Generate concise summaries of uploaded documents.
- **Quiz Creation and Grading:** Automatically generate quizzes and grade user responses.
- **Customizable Chat Interface:** Add or remove documents from the chat context, query based on context, and save chat history.
- **User Account Management:** Signup, login, update profile, and change account settings.
- **User Feedback:** Provide ratings and feedback for the platform's services.

2.3 User Classes and Characteristics

i. Login FR:



ii. Chatting FR:



2.4 Operating Environment

The software will operate in a web-based environment and is accessible through modern web browsers.

- Server Environment: Deployed on cloud platforms such as AWS or Azure
- Software Components:

Backend: Django

Frontend: React

Database: PostgreSQL

External APIs: Hugging Face

3. System Features

○ 3.1 Document Management

i. 3.1.1 Description and Priority:

This feature allows users to upload, view, download, and delete documents. It is a **High-priority** feature since it is fundamental to the system's core functionality, enabling users to provide the content required for subsequent AI-powered processing.

Priority Component Ratings:

- Benefit: 9
- Penalty: 8
- Cost: 3
- Risk: 2

ii. 3.1.2 Stimulus/Response Sequences:

1. Upload Document:
 - Stimulus: The user selects a file and clicks the "Upload" button.
 - Response: The system validates the file type and size, then stores it in the database. If successful, a confirmation message is displayed.
2. View Document:
 - Stimulus: The user selects a document from their list of uploaded files.
 - Response: The system retrieves the summary of the document and displays its content in a readable format.
3. Download Document:
 - Stimulus: The user clicks the "Download" button next to a document.
 - Response: The system provides the document for download in its original format.
4. Delete Document:
 - Stimulus: The user clicks the "Delete" button next to a document and confirms the action.
 - Response: The system removes the document from the database and updates the user's document list.

iii. 3.1.3 Functional Requirements

- **REQ-1:** The system shall allow users to upload documents in supported formats (e.g., PDF, DOCX) with a maximum size of 20 MB.
- **REQ-2:** The system shall validate uploaded files and reject unsupported formats or oversized files, displaying an appropriate error message.
- **REQ-3:** The system shall display a list of uploaded documents, including the file name and upload date.
- **REQ-4:** The system shall allow users to download their uploaded documents in their original format.

- **REQ-5:** The system shall allow users to delete uploaded documents, providing a confirmation prompt before deletion.
- **REQ-6:** The system shall handle errors gracefully, such as database connectivity issues, and display relevant error messages to the user.

○ 3.2 User Authentication and Profile Management

i. 3.2.1 Description and Priority

This feature includes user login, signup, and profile management functions such as updating username, password, and profile picture. It is **High-priority** as it ensures secure access to the system and allows users to manage their profiles.

Priority Component Ratings:

- Benefit: 9
- Penalty: 8
- Cost: 3
- Risk: 2

ii. 3.2.2 Stimulus/Response Sequences:

2. Signup:

- **Stimulus:** A new user provides an email, username, and password, then clicks "Sign Up."
- **Response:** The system validates the input, creates an account, and sends a confirmation email.

3. Login:

- **Stimulus:** The user enters valid credentials and clicks "Login."
- **Response:** The system verifies the credentials and grants access.

4. Update Username or Password:

- **Stimulus:** The user enters a new username or password and clicks "Update."
- **Response:** The system validates the input and updates the user's profile.

5. Update Profile Picture:

- **Stimulus:** The user uploads a new image file and clicks "Save."
- **Response:** The system processes the file and updates the profile picture.

i. 3.2.3 Functional Requirements

- **REQ-1:** The system shall allow new users to create accounts with a valid email, unique username, and secure password.
- **REQ-2:** The system shall validate login credentials and deny access for incorrect information.
- **REQ-3:** The system shall allow users to update their username and password.
- **REQ-4:** The system shall allow users to upload a profile picture in supported formats (e.g., PNG, JPEG).

- **REQ-5:** The system shall encrypt all user passwords.

- **3.3 Chat-Based Document Querying and Q&A**

- i. **3.3.1 Description and Priority**

This feature enables users to interact with their uploaded documents by entering queries in a chat interface. The system generates responses based on the context of the added documents. It is **High-priority** as it delivers the core functionality of the application.

Priority Component Ratings:

- Benefit: 9
- Penalty: 9
- Cost: 6
- Risk: 5

- ii. **3.3.2 Stimulus/Response Sequences**

- 1. **Add Document to Chat Context:**

- **Stimulus:** The user selects a document to include in the context.
 - **Response:** The system parses the document and prepares it for NLP processing.

- 2. **Enter Query:**

- **Stimulus:** The user types a question and submits it in the chat interface.
 - **Response:** The system processes the query and returns a response based on the context of the added documents.

- 3. **Reset Chat:**

- **Stimulus:** The user clicks "Reset Chat."
 - **Response:** The system clears the chat history and context.

- iii. **3.3.3 Functional Requirements**

- **REQ-1:** The system shall allow users to add or remove documents from the chat context.
 - **REQ-2:** The system shall process user queries using NLP models and retrieve relevant information from added documents.
 - **REQ-3:** The system shall provide options to reset the chat context.

- **3.4 Quiz and Summary Generation**

- i. **3.4.1 Description and Priority**

This feature generates quizzes and summaries based on user-uploaded documents. It is **Medium-priority** to enhance the learning process.

Priority Component Ratings:

- Benefit: 7

- Penalty: 5
- Cost: 4
- Risk: 3

ii. 3.4.2 Stimulus/Response Sequences

1. Generate Quiz:

- **Stimulus:** The user selects one or more documents and clicks "Generate Quiz."
- **Response:** The system creates a quiz with multiple-choice questions based on the content.

2. Grade Quiz:

- **Stimulus:** The user completes the quiz and clicks "Submit."
- **Response:** The system evaluates the answers and displays a score.

3. Generate Summary:

- **Stimulus:** The user selects a document and clicks "Generate Summary."
- **Response:** The system provides a concise summary of the document.

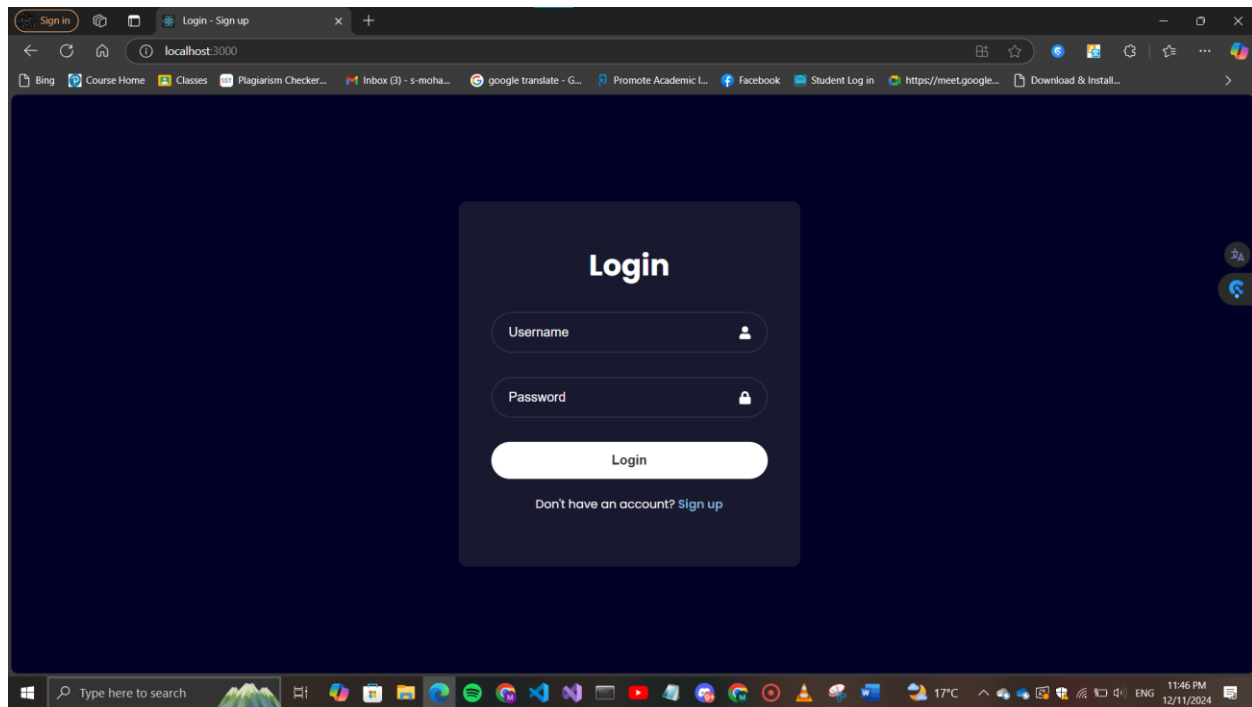
iii. 3.4.3 Functional Requirements

- **REQ-1:** The system shall allow users to generate quizzes based on the content of uploaded documents.
- **REQ-2:** The system shall evaluate user-submitted quizzes and display scores.
- **REQ-3:** The system shall generate summaries of user-uploaded documents using NLP models.

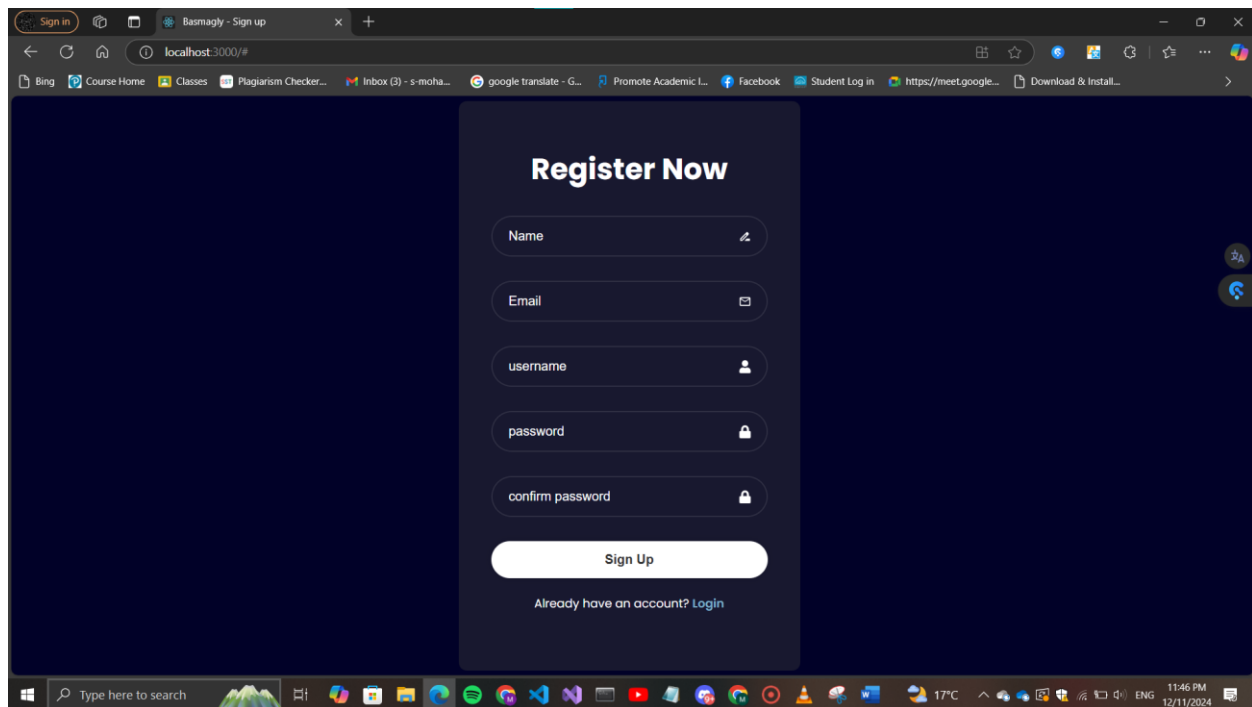
4. External Interface Requirements

4.1 User Interfaces

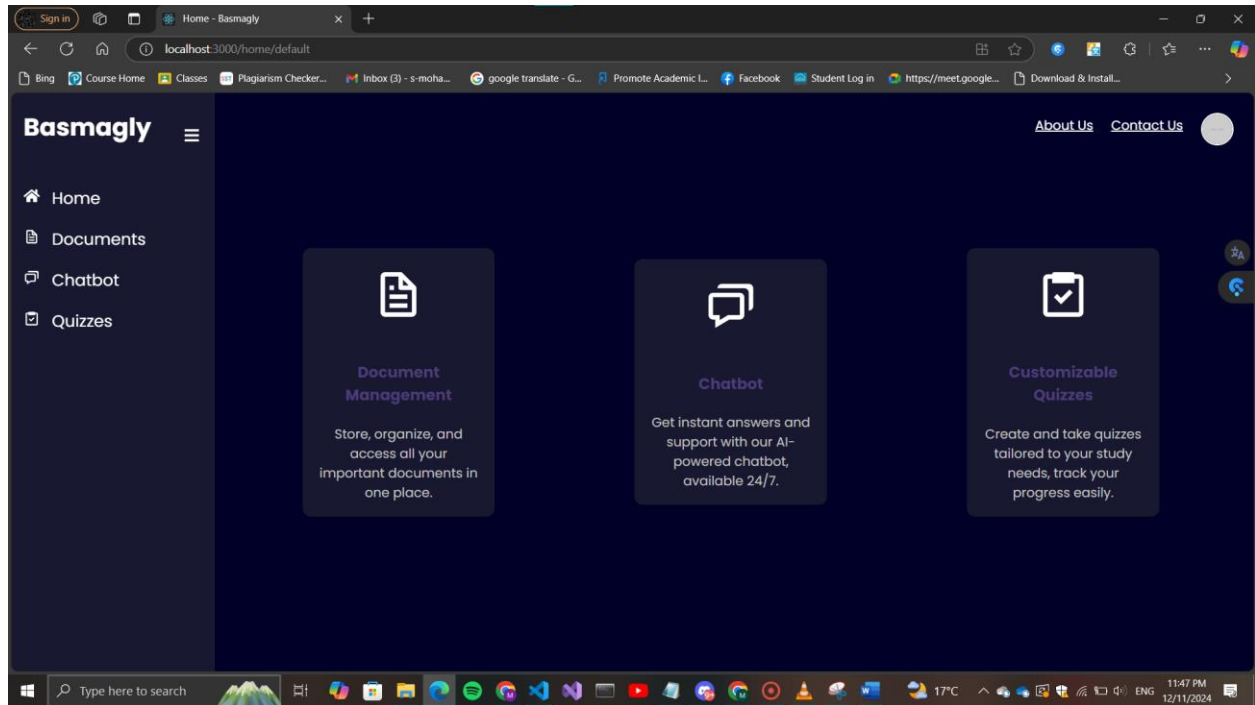
Sign in:



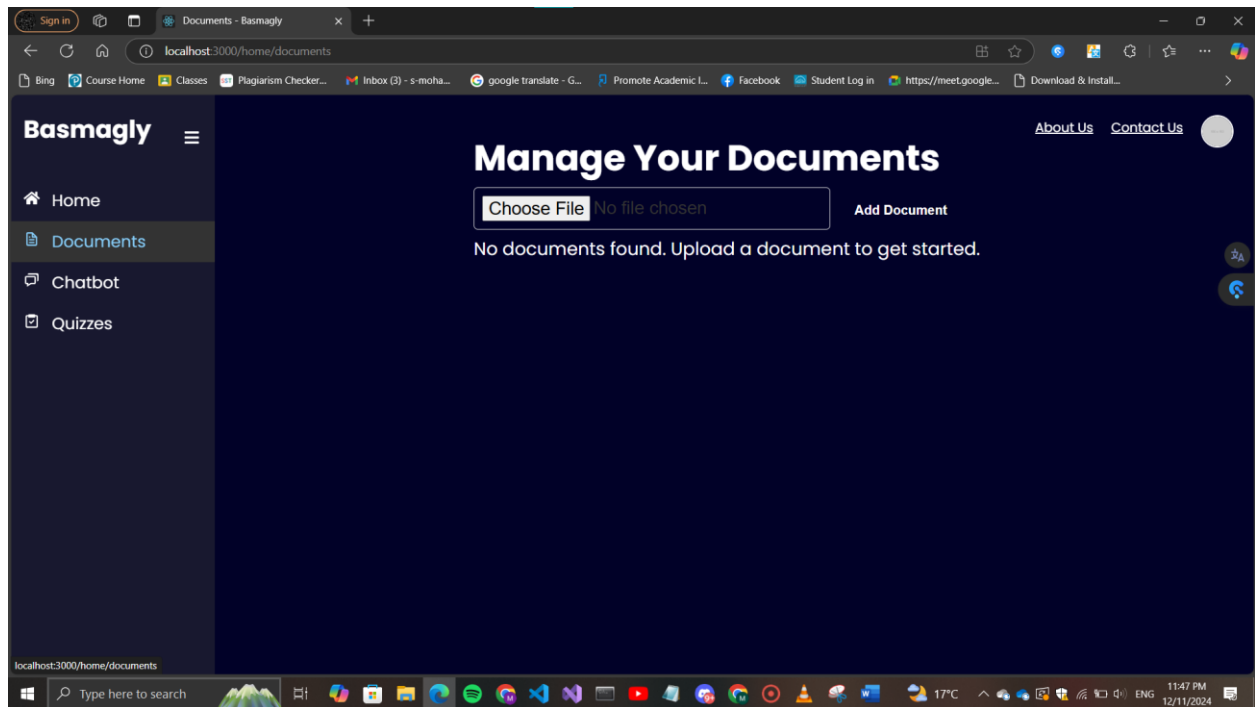
Sign up:



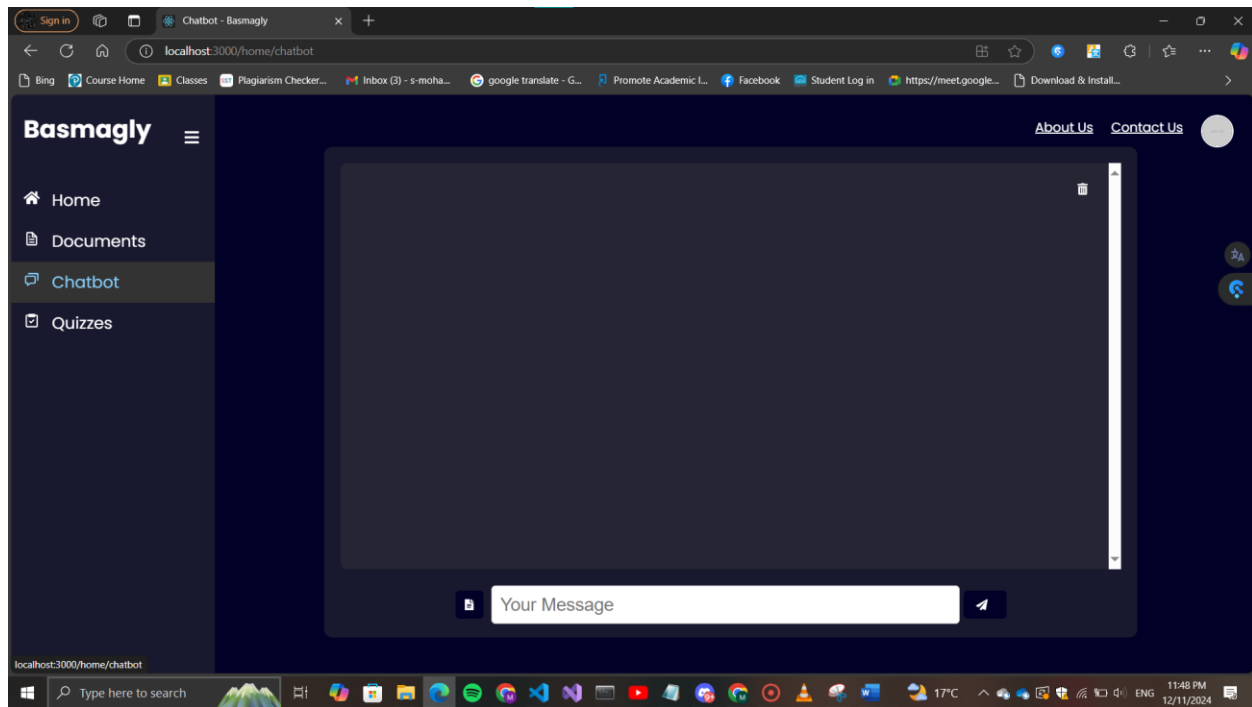
Home Page:



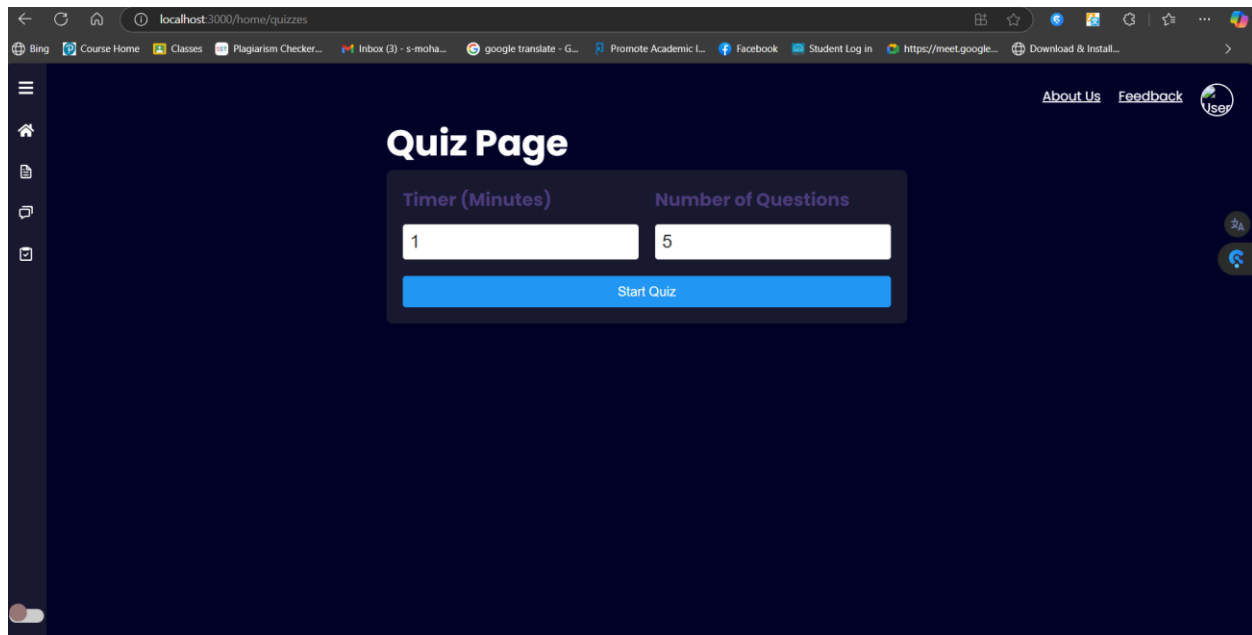
Document Management Page:



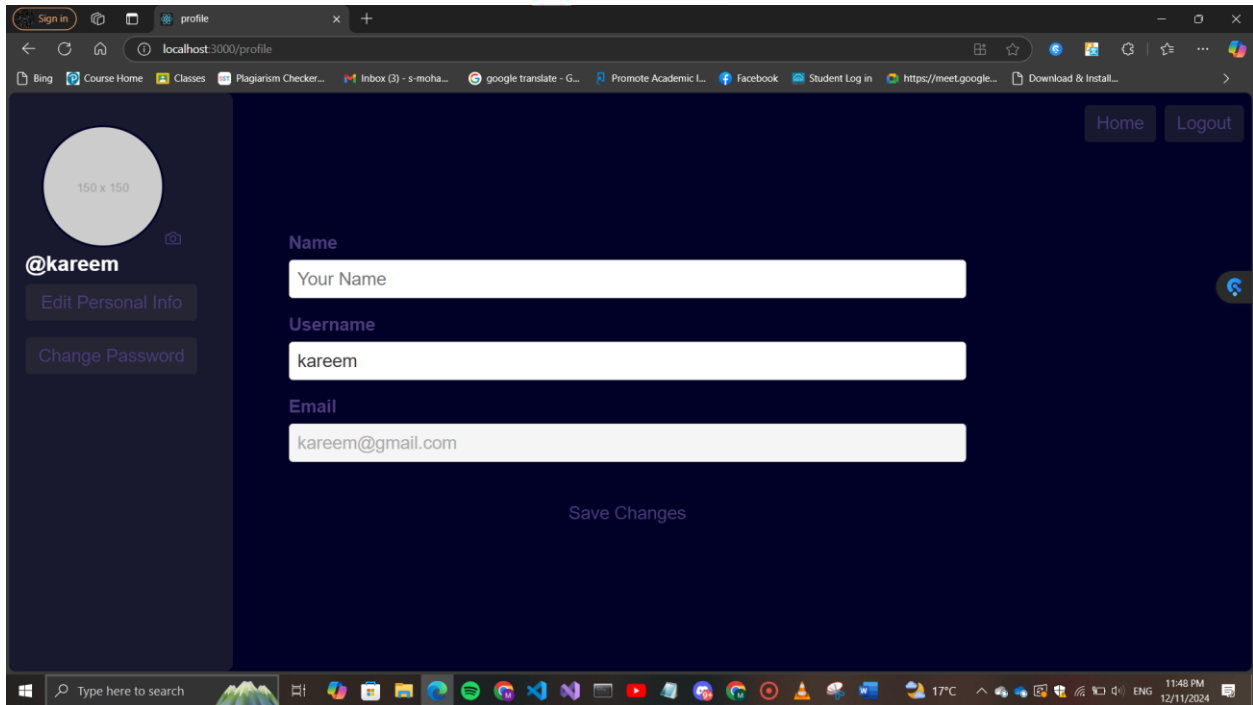
Chatbot Page:



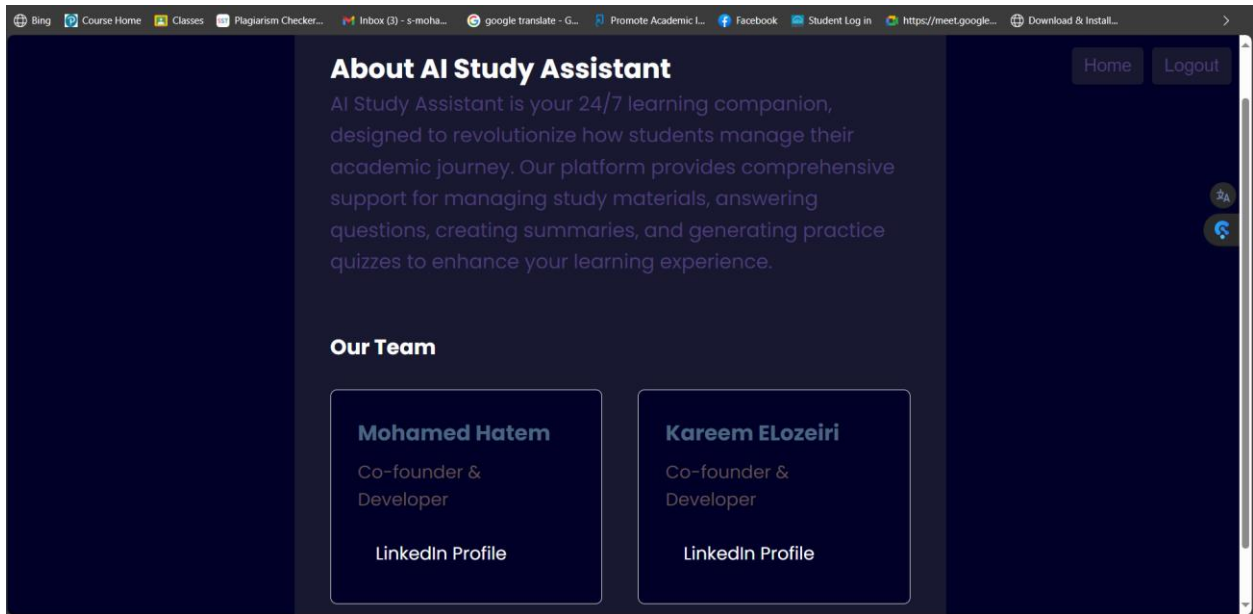
Quizzes Page:



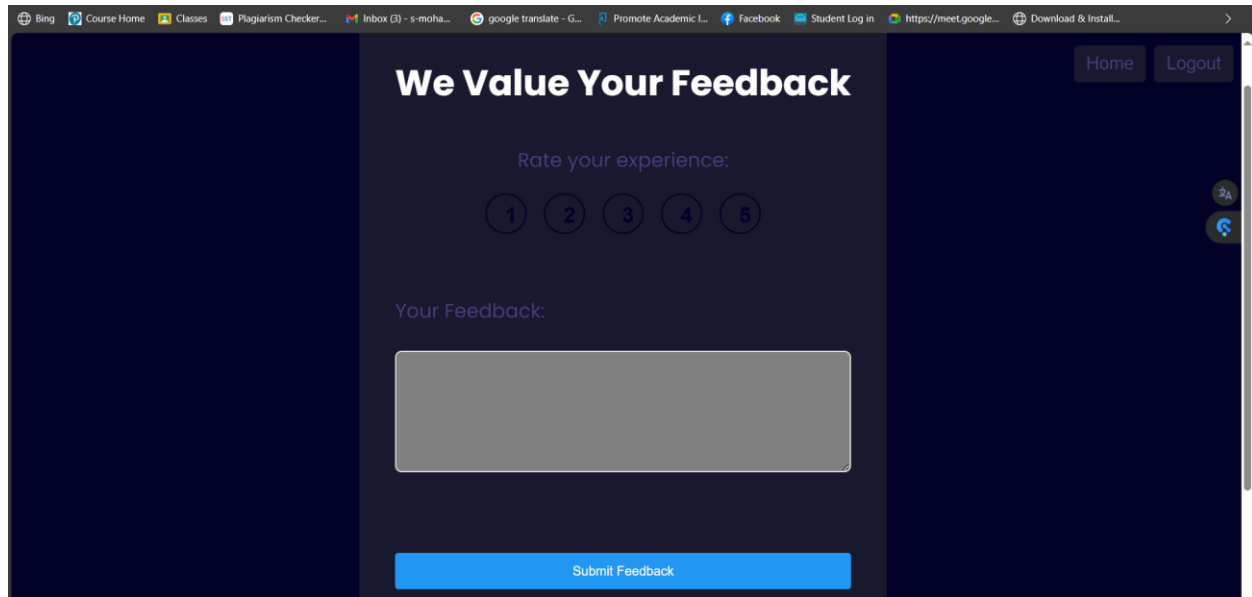
Profile Page:



About us:



Feedback:



4.2 Software Interfaces

1. 1. Databases

- **Component:** PostgreSQL (Version: Latest Stable Release)
- **Purpose:** Stores user information (e.g., credentials, profiles), uploaded documents, and chat history.
- **Data Inflow:**
 - o User credentials during signup and login.
 - o Uploaded documents metadata and file content.
 - o Generated chat history, summaries, quizzes, and ratings.
- **Data Outflow:**
 - o User profiles, documents, and chat history for retrieval.
 - o Query results or generated outputs for display in the user interface.

2. 2. Backend Framework

- **Component:** Django (Version: Latest Stable Release)
- **Purpose:** Acts as the server-side component, managing communication between the frontend, database, and external APIs.
- **Services Needed:**
 - HTTP requests handling for document upload, query processing, and user profile management.
 - Integration with Hugging Face models and LangChain for Natural Language Processing tasks.
- **Communication:** RESTful APIs facilitate secure data exchange between the server and client.

3. 3. Natural Language Processing Libraries

- **Component:** Hugging Face (Transformers Library)
- **Purpose:** Processes user queries, parses documents, and generates summaries or answers.
- **Data Inflow:** User queries and document context provided by Django.
- **Data Outflow:** Generated responses or relevant document excerpts for display in the frontend.

4. 4. Frontend Framework

- **Component:** React (Version: Latest Stable Release)
- **Purpose:** Provides a responsive and user-friendly interface for document uploads, querying, and interaction with generated content.
- **Data Inflow:** Processed data from Django (e.g., summaries, quiz questions, chat responses).
- **Data Outflow:** User input like uploaded files, queries, profile updates, and chat interactions.

5. Functional Requirements:

1. **Login:** Verifies user credentials (username and password) to grant access to their personalized dashboard and documents.
2. **Signup:** Collects user details like name, email, and password to create a new account in the system.
3. **Update Username:** Allows users to modify their current username to something new, with validation to ensure uniqueness.
4. **Update Password:** Provides a secure way for users to reset or change their password, requiring authentication.
5. **Update Profile Picture:** Enables users to personalize their account by uploading a new profile picture or replacing the existing one.
6. **Upload New Document:** Lets users upload documents (e.g., PDFs, Word files) to the platform for analysis and context generation.
7. **Retrieve Uploaded Documents:** Displays a list of all previously uploaded documents, allowing users to access and view them.
8. **Delete Uploaded Document:** Permanently removes a selected document from the user's account and the system.
9. **Download Uploaded Document:** Provides users with the option to download a copy of their uploaded documents back to their devices.
10. **Add Documents to Chat Context:** Allows users to select specific uploaded documents to include as a context for answering queries or generating summaries.
11. **About us:** Lets users know more information about our start-up.
12. **Generate Q&As:** Automatically creates relevant question-and-answer pairs based on the content of the selected documents.
13. **Generate Quizzes:** Uses the uploaded documents to create interactive quizzes with questions and options.

14. **Grade Quizzes:** Grades user responses to quizzes and provides feedback or scores.
15. **Timer:** Sets Timer to finish the quiz within it
16. **Number of Questions:** Sets the number of questions required in the quiz.
17. **Generate Summary:** Summarize the key points from the selected documents, presenting them in a concise format.
18. **Enter Query in Chat:** Users can type queries, and the system provides responses based on the context of the selected documents.
19. **Download Generated Chat:** Allows users to save the entire chat conversation as a downloadable file for future reference.
20. **Reset Chat:** Clears the current chat session, including its context and history, and starts a new one.
21. **Dark mode:** Enables users to customize the visual appearance of the website interface by selecting different modes.
22. **Feedback the Service:** Allows users to provide feedback or rate their experience with the platform, helping improve its quality.

6. Other Nonfunctional Requirements

6.1 Performance Requirements

- Login and Signup: Authentication must complete within 2 seconds.
- Document Handling: Uploads should finish within 5 seconds for files <10 MB and 15 seconds for files ≤50 MB; retrieval must be under 1 second.
- Content Generation: Summaries, Q&As, and quizzes must generate within 10-20 seconds, and chat responses within 2 seconds.
- Interactivity: Profile updates and UI customizations must apply instantly.

6.2 Software Quality Attributes

- Ease of UI: The interface must be simple and intuitive for users of all technical levels.
- Reliability: Ensure *99.9% uptime* for uninterrupted service.
- Maintainability: Codebase should support modular updates without downtime.
- Minimizing Hallucination: Leverage fine-tuning and retrieval-augmented generation to ensure 95% accuracy in model outputs.
- Adaptability: Support seamless scaling for increased document uploads or users.