Project Report

On

BUILDIQ AI Based Document Summarizer

Submitted to

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES RK VALLEY

In partial fulfillment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY In

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Submitted by

TALAMARLA FAIZA(R200367)

BATHALA RAMYA(R200210)

Under the Guidance of

A.Mahendra, Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES R K VALLEY

(catering the Educational Needs of Gifted Rural Youth of AP)

R.K Valley, Vempalli(M), Kadapa (Dist.)—516330

Accredited by ‘NAAC’ with ‘B+’ Grade

2024-2025

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(A.P.Government Act 18 of 2008) RGUKT-RK Valley

Vempalli,Kadapa,Andhrapradesh – 516330

Accredited by ‘NAAC’ with ‘B+’ Grade

CERTIFICATE OF PROJECT COMPLETION

This is to certify that the project work titled “**CSE Department Classroom Booking System** “ is a bonafied project work submitted by **T. Faiza(R200367)** and **B. Ramya(R200210)** in the department of COMPUTER SCIENCE AND ENGINEERING in partial fulfillment of requirements for the award of degree of Bachelor of Technology in **Computer Science and Engineering** for the year 2024-2025 carried out the work under the supervision

Project Guide	**Head of the Department**
A.Mahendra,	Dr. CH. Ratna Kumari,
M.Tech(JNTU A)	Mtech (UOH), Ph.D(JNTU H)
Asst.Prof. in Dept of CSE,	Asst. Prof. In Dept. Of CSE,
RGUKT-RKValley.	RGUKT-RKValley.

Signature of External Examiner

RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES

(A.P.Government Act 18 of 2008) RGUKT-RK Valley

Vempalli,Kadapa,Andhrapradesh – 516330

Accredited by 'NAAC' with 'B+' Grade

DECLARATION

We hereby declare that the project report entitled "BUILDIQ AI Based Document Summarizer" was done under the guidance of A. Mahendra is submitted in partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering during the academic session January 2024 – July 2024 at RGUKT-RK Valley. I also declare that this project is a result of our effort and has not been copied or imitated from any source. Citations from any websites are mentioned in the references. To the best of my knowledge, the results embodied in this dissertation work have not been submitted to any university or institute for the award of any degree or diploma.

With Sincere Regards,

B.Ramya(R2003210

T.Faiza(R200367)

ACKNOWLEDGEMENT

The satisfaction that accompaines the succesful completion of any task would be incomplete without the mention of the people who made it possible and whose guidance and encouragement crown all the efforts success

I am extremely grateful to our project guide **A.Mahendra** for her guidance, encouragement, co- operation,advices and support during the entire duration of the project work.

I would like to convey gratitude to our respected Director,**Prof.A V S S Kumar Swamy Gupta**, for fostering an excellent academic climate in our institution.

I also express my reverence to our respected Head of the Department of Computer Science and Engineering **Dr.Ch.Ratna Kumari** for the encouragement,overall guidance in viewing this project as a good asset and effort in bringing out this project.

With Sincere Regards,

T. Faiza-R200367

B. Ramya-R200210

Table of Contents

List Of Figures

1. INTRODUCTION

1.1 Project Overview

In an age where vast amounts of digital content are created and consumed daily, the ability to quickly comprehend large documents is increasingly critical. Whether it's for academic research, legal analysis, business reporting, or casual reading, users often find themselves overwhelmed by lengthy documents. Reading through dozens of pages to extract key information is not only time-consuming but also mentally taxing.

To address this growing need, our project introduces an **AI-based Document Summarizer Web Application** — a platform that allows users to upload or paste long-form documents and receive concise, high-quality summaries in return. The system leverages **Natural Language Processing (NLP)** techniques to extract the most important information from text files, PDFs, and Word documents. Additionally, the application supports **multilingual inputs**, translating non-English texts into English before summarization, and reverting the summary back to the original language if required.

This tool is particularly beneficial for students, educators, researchers, and professionals who deal with dense documents on a regular basis. Unlike traditional tools, this system provides a **user-friendly interface**, ensures **data privacy**, and offers **personalized history tracking** of past summaries.

The application is built using the **Flask** framework on the backend, with **HTML, CSS, and JavaScript** for the frontend. It integrates **SQLite** for secure user authentication and history management, and applies **NLTK** (Natural Language Toolkit) for text preprocessing and frequency-based extractive summarization. Additionally, the system incorporates language detection and translation support using the **langdetect** and **Googletrans** libraries.

This project not only demonstrates a working implementation of text summarization using AI but also showcases a complete software solution — from user login, file upload, and text analysis to result presentation and summary download. It provides an end-to-end experience for users while serving as a strong academic example of applied Artificial Intelligence in web applications.

1.2 Objectives

The primary objectives of this project are:

- **To provide a secure and user-friendly web platform** for document summarization, allowing users to register, log in, and manage their profiles with robust session and password management.
- **To enable users to upload documents (PDF, DOCX, TXT) or paste text directly** for summarization, supporting a wide range of real-world use cases.
- **To automate the extraction and summarization of text** using Natural Language Processing (NLP) techniques, specifically extractive summarization based on sentence ranking and frequency analysis.

- **To support multi-language documents** by detecting the input language and translating content to English for summarization, then translating the summary back to the original language if required.
- **To allow users to customize the summary length** (short, medium, long), making the platform adaptable to different needs and contexts.
- **To maintain a history of all user summaries**, enabling easy retrieval, review, and download of past results.
- **To ensure data privacy and security** through hashed passwords, session management, and secure file handling.
- **To deliver a clean, responsive, and intuitive user interface** for seamless navigation and interaction.

1.3 Scope of the Project

The scope of the project includes:

- **User Authentication:** Secure registration and login functionality with password hashing and session management, ensuring that only authorized users can access the platform and their data.
- **Document Upload and Text Input:** Support for uploading PDF, DOCX, and TXT files, as well as pasting text directly into the application for summarization.
- **Text Extraction and Language Handling:** Automatic extraction of text from uploaded files, detection of the document's language, and translation to and from English as needed for summarization.
- **Extractive Summarization:** Implementation of an extractive summarization algorithm using NLTK, ranking sentences by word frequency and relevance to produce concise summaries.
- **Customizable Summary Length:** Options for users to select short, medium, or long summaries, mapped to a specific number of sentences for flexibility.
- **Summary History:** Persistent storage of all user summaries, with a dashboard for viewing, downloading, and managing previous results.
- **Profile Management:** A dedicated profile page for users to view their information and manage their session.
- **Responsive UI:** A modern, responsive frontend built with HTML, CSS, and JavaScript, ensuring accessibility across devices.
- **Security:** Secure handling of user data, including input validation, file type restrictions, and session management to protect against unauthorized access and attacks.

Out of Scope:

- Collaborative document summarization or sharing between users.
- Advanced abstractive summarization using deep learning models.
- Editing or annotation of uploaded documents.
- Integration with third-party cloud storage or document management systems.

2. System Analysis

2.1 Existing Systems

Existing document summarization systems generally fall into two categories: standalone desktop applications and online web services. Many desktop tools require manual installation, limited file format support, and lack real-time accessibility. Online summarization services, while more accessible, often restrict the number of documents, lack user authentication, or provide only basic copy-paste functionality without persistent history or user profiles.

Common limitations in existing systems include:

- **Limited file type support:** Many only handle plain text or PDFs, neglecting DOCX or multi-language documents.
- **No user management:** Most do not offer secure registration, login, or personalized summary history.
- **Basic summarization:** Many use simple algorithms, often failing to capture key points or adapt summary length.
- **Lack of data privacy:** User data and uploaded files may not be securely handled or stored.

2.2 Proposed System

The proposed system addresses these shortcomings by providing a secure, full-featured web application for document summarization with the following enhancements:

- **Multi-format support:** Users can upload PDF, DOCX, and TXT files, or paste text directly.
- **User authentication:** Secure registration and login with hashed passwords and session management.
- **Personalized experience:** Each user has a private dashboard, summary history, and profile management.
- **Advanced summarization:** Utilizes NLP (NLTK) for extractive summarization, with automatic language detection and translation.
- **Custom summary length:** Users select short, medium, or long summaries to suit their needs.

- **Persistent storage:** Summaries and user data are stored in a SQLite database, allowing users to revisit and download their summaries at any time.
- **Responsive UI:** Clean, intuitive interface accessible across devices.

2.3 Feasibility Study

Technical Feasibility:
The project leverages well-established technologies: Flask (Python) for backend logic, SQLite for data storage, and HTML/CSS/JavaScript for the frontend. Libraries like NLTK, pdfminer, docx, and Googletrans are available and compatible, ensuring robust text extraction, language detection, and summarization.

Operational Feasibility:
Users interact with a simple, menu-driven interface. Secure authentication and session management ensure data privacy. The system is designed for easy navigation, with clear options for uploading, pasting, viewing history, and managing profiles.

Economic Feasibility:
The use of open-source technologies and lightweight local storage (SQLite) minimizes costs. The application can run on standard hardware and does not require expensive infrastructure.

2.4 Requirement Analysis

Functional Requirements:

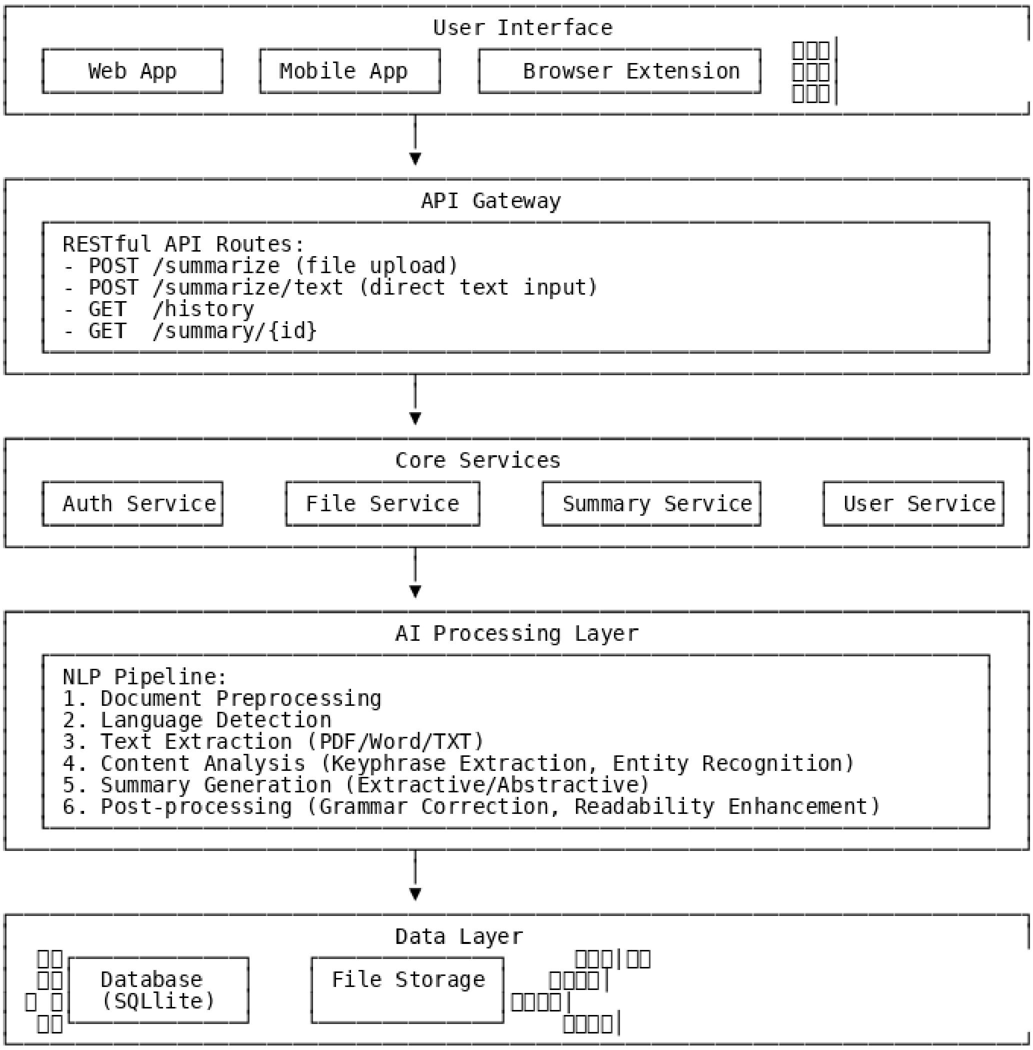
- User registration and login
- Secure password storage (hashing)
- File upload (PDF, DOCX, TXT) and text input
- Language detection and translation
- Extractive summarization with adjustable length
- Summary history with download option
- Profile management
- Session handling and logout

Non-Functional Requirements:

- Responsive and intuitive UI
- Secure handling of user data and files
- Efficient processing for quick summary generation
- Scalability for multiple users

3. System Design

3.1 Architecture Diagram



RESTful API Routes:
- POST /summarize (file upload)
- POST /summarize/text (direct text input)
- GET /history
- GET /summary/{id}

NLP Pipeline:
1. Document Preprocessing
2. Language Detection
3. Text Extraction (PDF/Word/TXT)
4. Content Analysis (Keyphrase Extraction, Entity Recognition)
5. Summary Generation (Extractive/Abstractive)
6. Post-processing (Grammar Correction, Readability Enhancement)

The architecture of the document summarization platform follows a classic three-tier model:

- **Presentation Layer (Frontend):**
  Built with HTML, CSS, and JavaScript, the frontend provides a responsive user interface for all user interactions. Key pages include Login, Register, Dashboard, Upload Document, Paste Text, History, Profile, and Result pages. The interface is styled for clarity and usability, featuring a sidebar menu for easy navigation between modules.
- **Application Layer (Backend):**
  The backend is developed using Flask (Python). It handles all business logic including user authentication, file uploads, text extraction, language detection, translation, summarization,

and session management. It communicates with the database for user and summary data, and renders templates for each page.

- **Data Layer (Database):**
  SQLite is used for persistent storage. Two primary tables are maintained: users (for authentication and profile management) and summaries (for storing summary history linked to each user).

Data Flow Overview:

1. User interacts with the frontend (e.g., uploads a document or pastes text).
2. The frontend sends data to the backend via HTTP POST requests.
3. The backend processes the input, performs summarization, and stores results in the database.
4. The backend renders the result page or updates the user’s history, which is displayed on the frontend.

3.2 Functional Modules

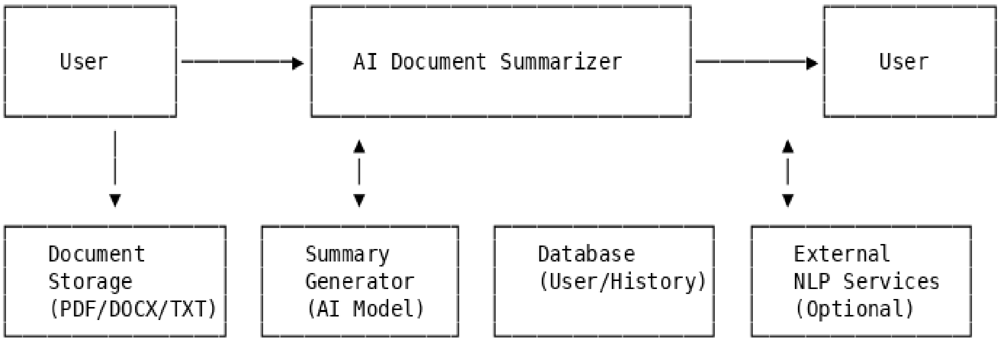
The application is divided into the following functional modules:

- **Authentication Module:**
  Handles user registration, login, session management, and secure password hashing using bcrypt.
- **Document Upload & Text Input Module:**
  Allows users to upload PDF, DOCX, or TXT files, or paste text directly for summarization.
- **Text Extraction & Language Handling Module:**
  Extracts text from uploaded files, detects the language, and translates non-English content to English and back as needed.
- **Summarization Module:**
  Performs extractive summarization using NLTK, with options for short, medium, or long summaries.
- **History Module:**
  Stores and displays all previous summaries for each user, with options to view or download results.
- **Profile Management Module:**
  Displays user profile information and manages session/logout.
- **Result Display Module:**
  Shows the generated summary, with options to copy or download the result.

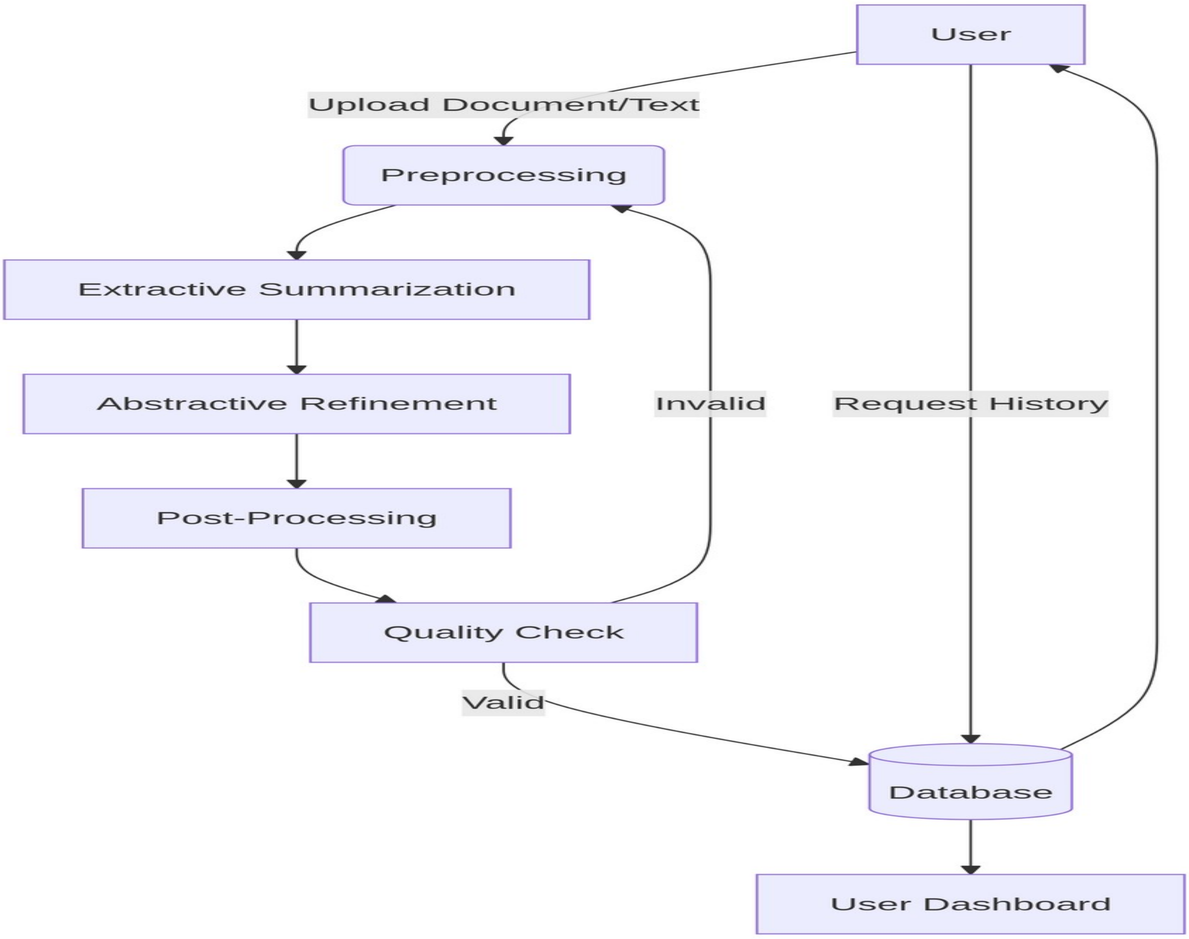
3.3 Data Flow Diagrams (DFD)

Level 0 (Context Diagram):

- User → [Web Application] ←→ [Database]


User
AI Document Summarizer
User
Document Storage (PDF/DOCX/TXT)
Summary Generator (AI Model)
Database (User/History)
External NLP Services (Optional)

Level 1 DFD (Major Processes):

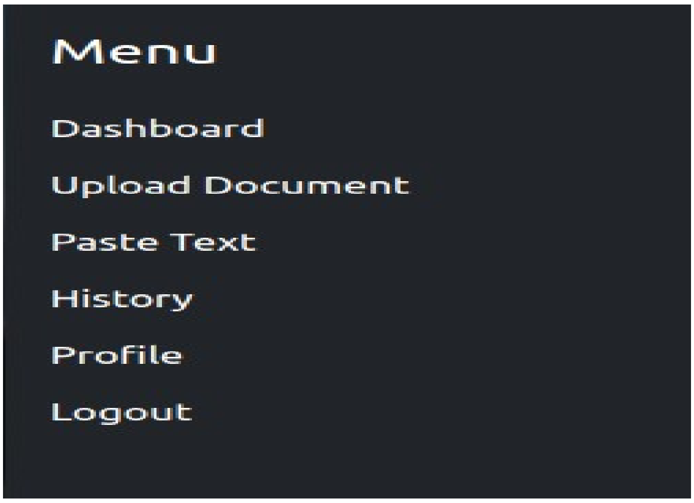


- User submits document/text → Authentication check → Text extraction & language detection → Summarization → Store summary → Display result/history

3.4 UI Layout and Interface Details

The user interface is clean, intuitive, and consistent across all modules:

- **Sidebar Menu:**



  Provides navigation links to Dashboard, Upload Document, Paste Text, History, Profile, and Logout.

- **Dashboard:**
  Welcome message and quick access to main features.
- **Upload Document:**
  File selector for PDF, DOCX, or TXT, summary length options, and submit button.
- **Paste Text:**
  Text area for input, summary length options, and submit button.
- **History:**
  Table listing previous summaries with filename, preview, date/time, and actions (view/download).
- **Profile:**
  Displays username and logout option.
- **Result Page:**
  Shows summary points, with options to copy or download as TXT.
- **Styling:**
  Consistent color scheme, card layouts, and responsive design for accessibility across devices.

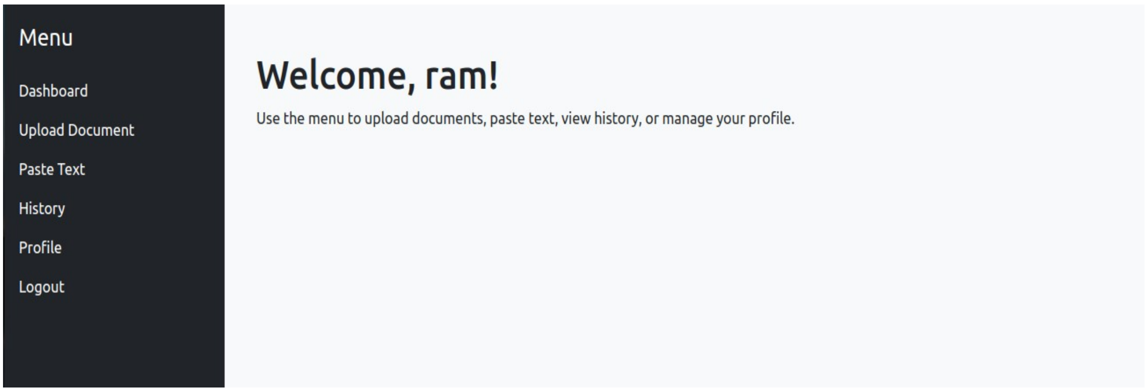
4. Implementation

4.1 Frontend (HTML/CSS/JS)

The frontend of the document summarization platform is developed using HTML, CSS, and JavaScript, providing a clean and responsive user experience. The interface is structured around a sidebar menu for easy navigation between core modules: Dashboard, Upload Document, Paste Text, History, Profile, and Logout.

Key Pages and Features:

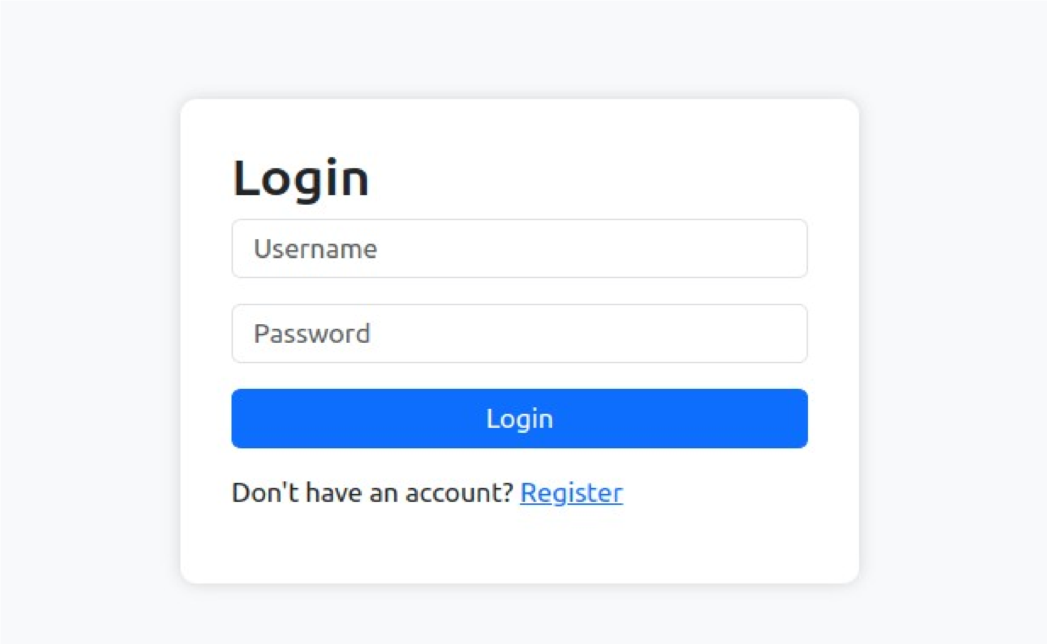
- **Dashboard:**
  Displays a welcome message with the logged-in username and quick access to main features, guiding users to upload documents, paste text, view history, or manage their profile.



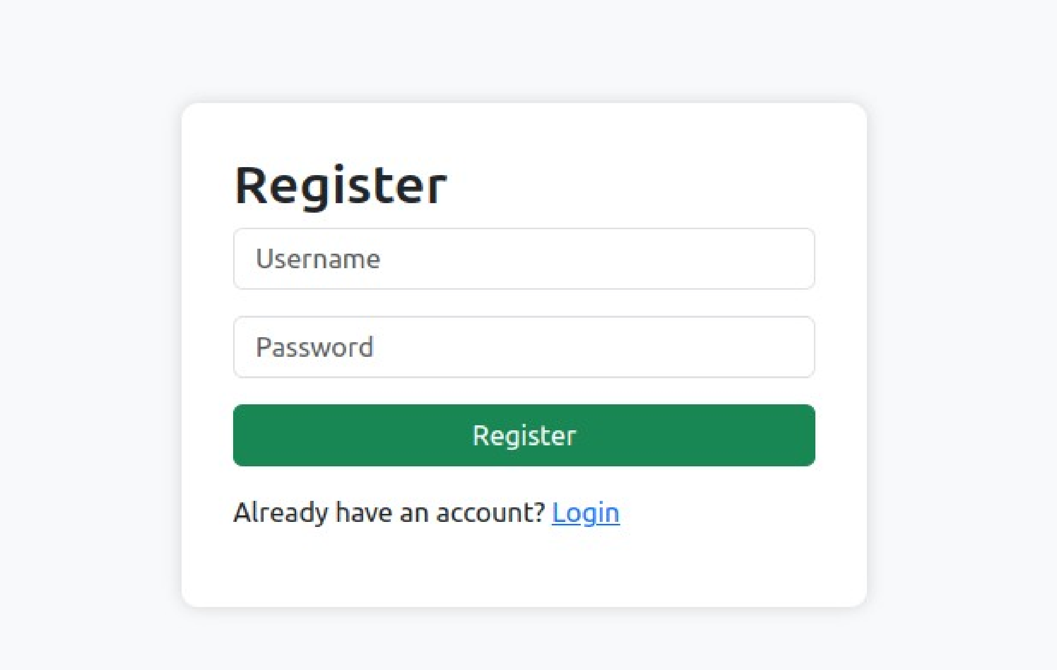
Welcome, ram!

Use the menu to upload documents, paste text, view history, or manage your profile.

- **Login/Register:**

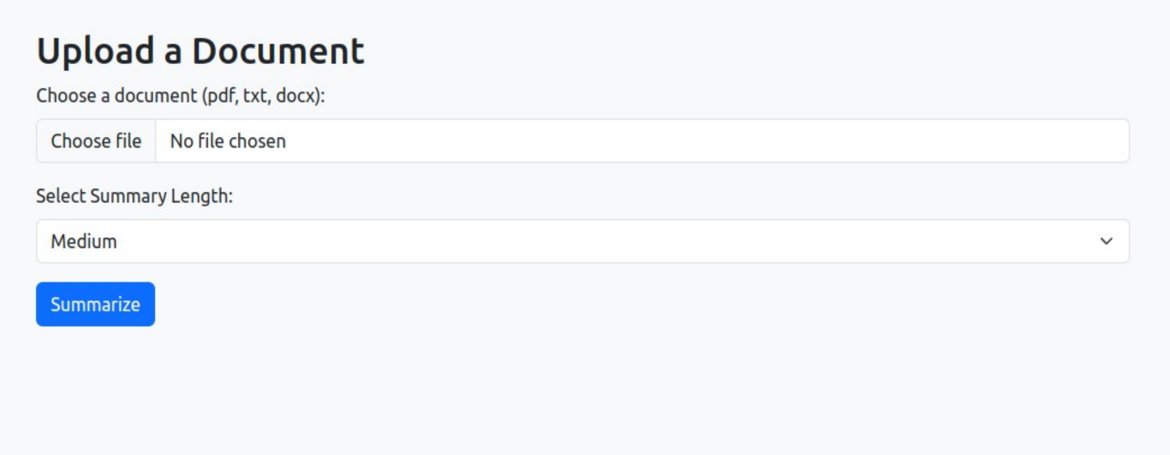


Don't have an account? Register



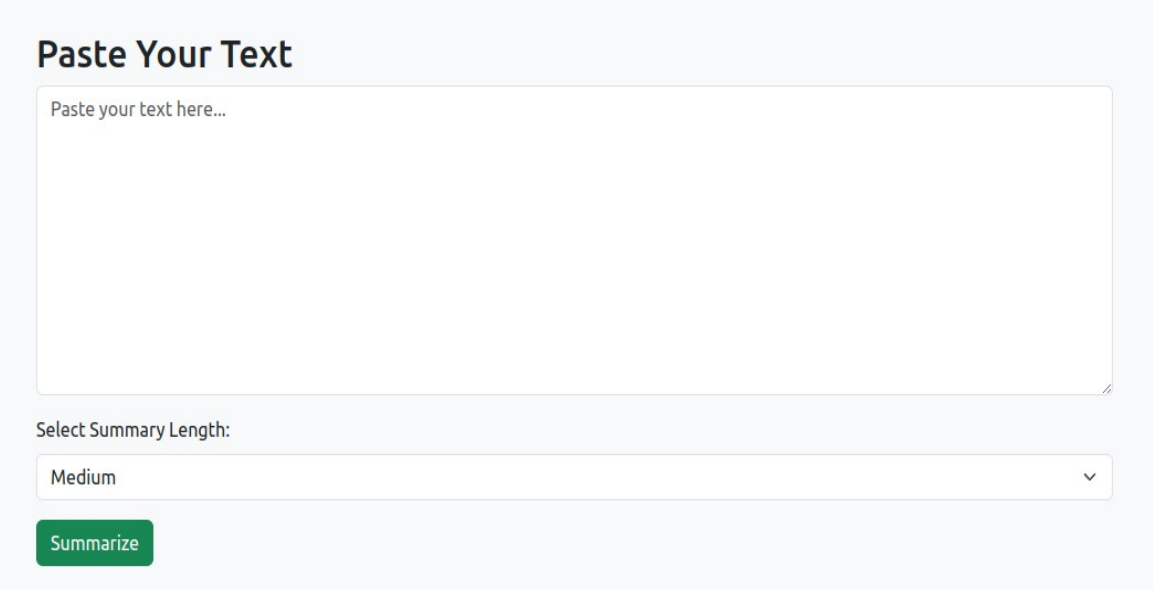
Secure login and registration forms with error feedback. Registration ensures unique usernames and secure password handling.

- **Upload Document:**
  Allows users to upload files in PDF, DOCX, or TXT format, select summary length (short, medium, long), and submit for processing. The UI provides clear instructions and feedback for file selection and summary options.



Select Summary Length: Medium

- **Paste Text:**
  Enables users to paste or type text directly, select summary length, and submit for summarization. The interface is simple and intuitive, with immediate access to the summary generation feature.



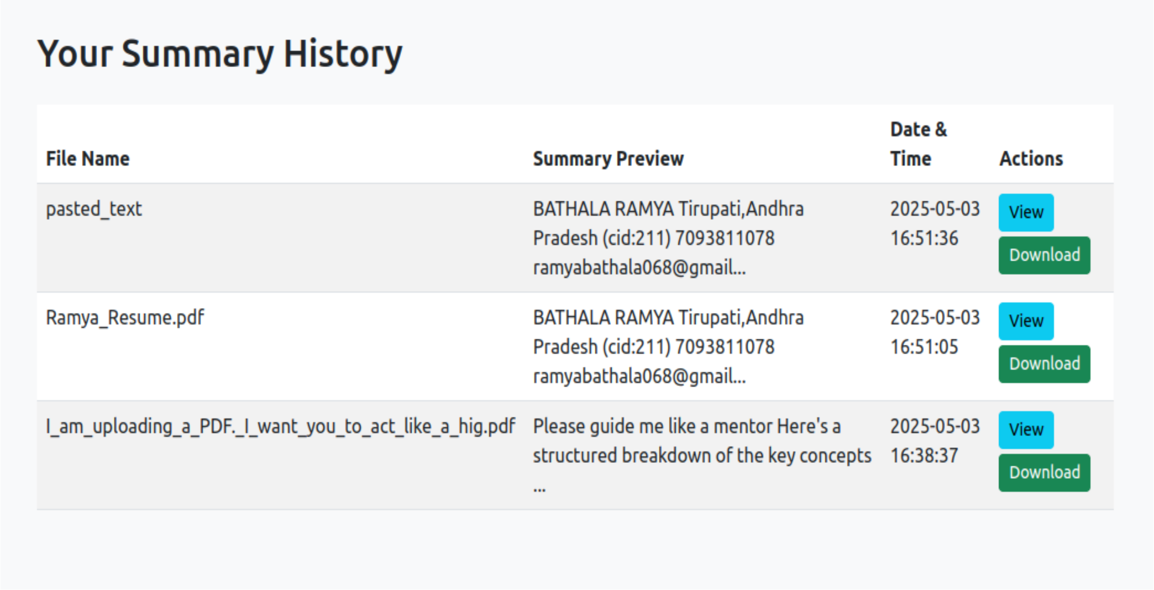
Paste Your Text

Paste your text here...

Select Summary Length:

Medium

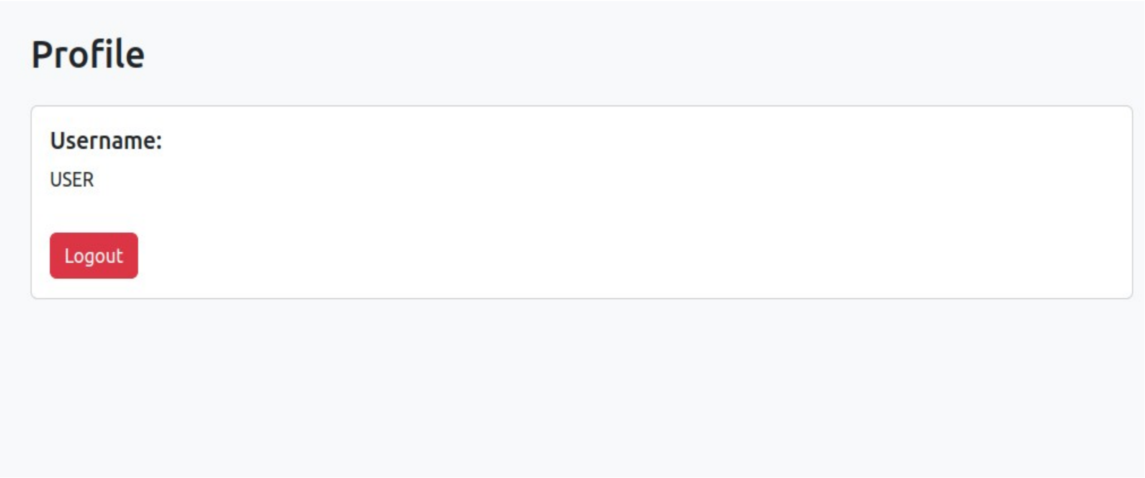
- **History:**
  Displays a table of all previous summaries, including file name, preview, date/time, and actions to view or download the full summary. If no summaries exist, a prompt encourages users to start uploading or pasting text.



Your Summary History

File Name	Summary Preview	Date & Time	Actions
pasted_text	BATHALA RAMYA Tirupati,Andhra Pradesh (cid:211) 7093811078 ramyabathala068@gmail...	2025-05-03 16:51:36	View Download
Ramya_Resume.pdf	BATHALA RAMYA Tirupati,Andhra Pradesh (cid:211) 7093811078 ramyabathala068@gmail...	2025-05-03 16:51:05	View Download
I_am_uploading_a_PDF._I_want_you_to_act_like_a_hig.pdf	Please guide me like a mentor Here's a structured breakdown of the key concepts ...	2025-05-03 16:38:37	View Download

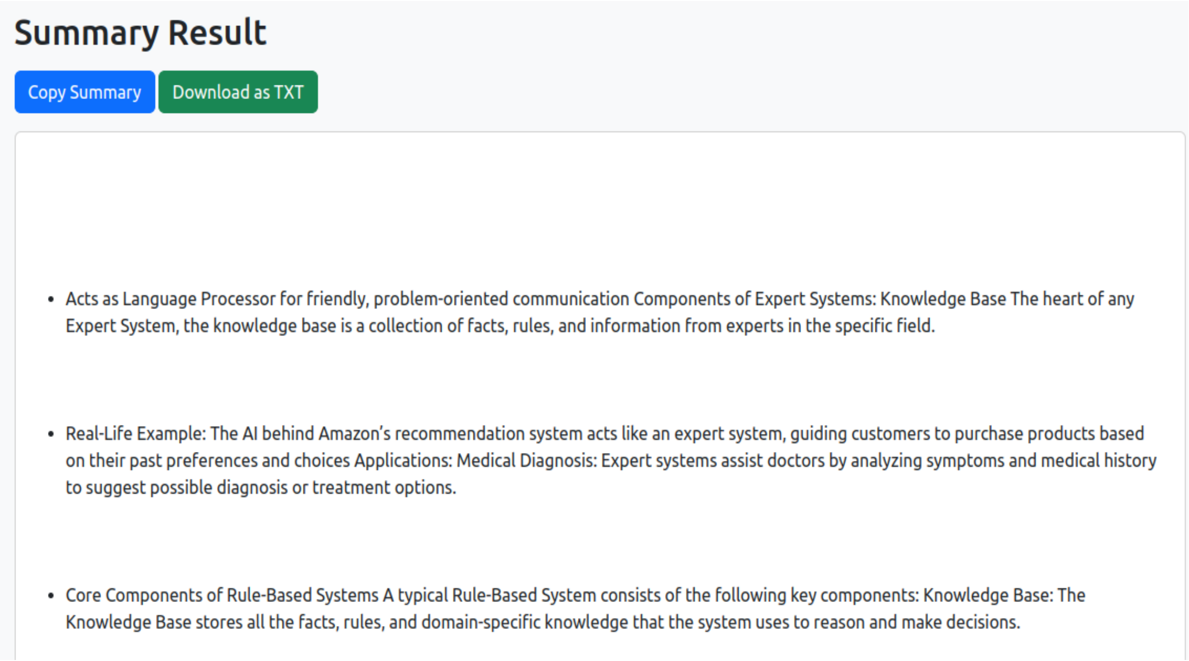
- **Profile:**
  Shows the logged-in username and provides a logout option, allowing users to manage their session securely.



Profile

Username:
USER

- **Result Page:**
  Presents the generated summary in a readable format, with options to copy the summary or download it as a TXT file. Users can return to the dashboard from this page.



Summary Result

- Acts as Language Processor for friendly, problem-oriented communication Components of Expert Systems: Knowledge Base The heart of any Expert System, the knowledge base is a collection of facts, rules, and information from experts in the specific field.
- Real-Life Example: The AI behind Amazon's recommendation system acts like an expert system, guiding customers to purchase products based on their past preferences and choices Applications: Medical Diagnosis: Expert systems assist doctors by analyzing symptoms and medical history to suggest possible diagnosis or treatment options.
- Core Components of Rule-Based Systems A typical Rule-Based System consists of the following key components: Knowledge Base: The Knowledge Base stores all the facts, rules, and domain-specific knowledge that the system uses to reason and make decisions.

Styling and Responsiveness:

- The application uses a consistent color scheme and card-based layouts for clarity.
- The sidebar and content areas are styled for accessibility, with hover effects and clear section headings.
- The layout adapts to different screen sizes, ensuring usability across devices.

4.2 Backend (Flask/Python)

The backend of the document summarization platform is built using Flask, a lightweight Python web framework. It handles user authentication, file uploads, text extraction, language detection, translation, summarization, session management, and database operations.

Key Backend Features:

- **User Authentication and Session Management:**
    - Registration and login are implemented with secure password hashing using bcrypt.
    - Flask sessions track logged-in users, storing the username and user ID securely.
    - Session management ensures only authenticated users can access dashboard, upload, paste, history, and profile pages.
- **Database Integration:**
    - SQLite is used for persistent data storage.
    - Two tables: users (stores username and hashed password) and summaries (stores user_id, filename, summary, and timestamp).
    - Helper functions manage database connections and queries, ensuring data integrity.
- **File Upload and Text Extraction:**
    - The backend accepts PDF, DOCX, and TXT files.
    - Uploaded files are saved securely, and text is extracted using appropriate libraries: pdfminer for PDFs, docx for Word documents, and standard file reading for TXT files.
    - After processing, uploaded files are deleted from the server for security and storage efficiency.
- **Language Detection and Translation:**

  - The langdetect library determines the language of the input text.
  - If the text is not in English, it is translated to English using Googletrans for summarization, and the summary is translated back to the original language if needed.

- **Summarization Logic:**
  - Extractive summarization is implemented using NLTK.
  - Sentences are tokenized, ranked by word frequency (excluding stopwords), and the top sentences are selected based on the user's chosen summary length (short, medium, long).
  - The summary is stored as a list of sentences and rendered for the user.
- **Summary History and Download:**
  - All summaries are saved in the database and displayed in the user's history.
  - Users can view, download, or revisit previous summaries.
  - Downloaded summaries are provided as plain text files.
- **Profile and Logout:**
  - Users can view their profile (username) and securely log out, which clears the session.

Security Measures:

- Passwords are never stored in plain text; bcrypt hashing is enforced for all credentials.
- File uploads are restricted by type and size, and filenames are sanitized.
- Sessions are securely managed, and all sensitive routes require authentication.
- Input validation and error handling are implemented throughout.

Backend Technologies Used:

- Flask (Python)
- SQLite
- NLTK, pdfminer, docx, langdetect, googletrans, bcrypt

4.4 Authentication

Authentication is a critical component of the document summarization platform, ensuring that user data and summaries remain private and secure. The project implements a robust authentication system using Flask sessions and bcrypt password hashing.

Key Features of Authentication:

- **User Registration:**
  Users register by providing a unique username and password. The password is hashed using bcrypt before being stored in the SQLite database, ensuring that plaintext passwords

- are never saved or transmitted. If a username already exists, the user is prompted to choose a different one.

Example from register.html:

text

Register

{% if error %}

{{ error }}

{% endif %}

Register

Already have an account?

Login

- **User Login:**
  During login, the entered password is checked against the stored bcrypt hash. If the credentials are valid, Flask's session mechanism stores the username and user ID, allowing access to authenticated routes. Incorrect credentials result in an error message.

Example from login.html:

text

Login {% if error %} {{ error }} {% endif %}

Login

Don't have an account? Register

- **Session Management:**
  Once logged in, the user’s session persists across pages, enabling access to the dashboard, upload, paste, history, and profile sections. Logging out clears the session, returning the user to the login page and protecting their data.
- **Access Control:**
  All sensitive routes (dashboard, upload, paste, history, profile) check for an active session. If a session is not present, the user is redirected to the login page, preventing unauthorized access.
- **Profile Management:**
  The profile page displays the logged-in username and provides a logout option, allowing users to end their session securely.

Example from profile.html:

text

Profile

Username:

{{ username }}

Logout

- **Security Best Practices:**
  - Passwords are always hashed using bcrypt before storage.
  - Sessions are managed securely using Flask’s built-in session system and a secret key.
  - Usernames are enforced to be unique at registration.
  - Error messages are provided for invalid login attempts or duplicate registrations.

This authentication system ensures that only authorized users can access their summaries and personal data, providing a foundation of trust and privacy for the platform.

5. Testing

5.1 Types of Testing

A comprehensive testing strategy was adopted to ensure the reliability, security, and usability of the document summarization platform. The following types of testing were performed:

- **Unit Testing:**
  Individual functions such as text extraction, language detection, translation, and summarization were tested with various input cases to verify correctness and robustness. For example, the summarize function was tested with short and long texts, different summary lengths, and edge cases (empty or single-sentence inputs).
- **Integration Testing:**
  The interaction between modules (e.g., file upload, text extraction, summary generation, and database storage) was tested to ensure seamless end-to-end workflow. This included uploading different file types and verifying that summaries were correctly stored and retrievable from the history.
- **System Testing:**
  The entire web application was tested as a whole, simulating real user scenarios such as registration, login, uploading documents, pasting text, viewing history, and downloading summaries. The system was checked for correct navigation, session management, and data persistence.
- **User Interface Testing:**
  The frontend was tested for responsiveness, accessibility, and usability. All UI elements, including forms, buttons, menus, and feedback messages, were verified for correct display and behavior across devices and screen sizes.
- **Security Testing:**
  Authentication mechanisms were tested for vulnerabilities such as SQL injection, session hijacking, and password brute-forcing. Passwords were validated to be securely hashed, and session management was checked for proper user isolation.
- **Error Handling and Validation Testing:**
  The application was tested for graceful handling of invalid inputs, unsupported file types, large files, and missing fields. Proper error messages and feedback were provided to guide the user.
- **Performance and Load Testing:**
  The system was tested with multiple users and large documents to evaluate response times and stability under load. File upload limits (16MB) and session timeouts were verified for effectiveness.

5.2 Test Cases

A range of test cases were designed to validate the core functionalities and edge cases of the application:

- **User Registration and Login**

  - *Valid registration*: Register with a unique username and strong password; expect successful account creation and redirect to login.
  - *Duplicate registration*: Attempt to register with an existing username; expect an error message (“Username already exists.”).
  - *Valid login*: Login with correct credentials; expect redirect to dashboard.
  - *Invalid login*: Login with incorrect username or password; expect error message (“Invalid credentials.”).

- **Document Upload and Summarization**
  - *Valid file upload*: Upload PDF, DOCX, or TXT; expect summary generation and display on result page.
  - *Unsupported file type*: Attempt to upload an unsupported file (e.g., .jpg); expect error or rejection.
  - *Large file*: Upload file near 16MB limit; expect successful summary or error if limit exceeded.
  - *Non-English document*: Upload a non-English file; expect language detection, translation, and summary in original language.

- **Paste Text Summarization**
  - *Valid text*: Paste a paragraph; expect summary generation.
  - *Empty text*: Submit with no text; expect validation error.
  - *Non-English text*: Paste non-English content; expect translation and summary in original language.

- **History and Download**
  - *History display*: After generating summaries, check history page for correct listing of summaries with file name, preview, timestamp, and actions (view/download).
  - *Download summary*: Download a summary; expect a .txt file with correct content.
  - *No history*: New user with no summaries; expect “No summaries found yet. Start uploading or pasting!”.

- **Profile and Logout**
  - *Profile page*: Access profile; expect display of username and logout option.
  - *Logout*: Click logout; expect session cleared and redirect to login page.

5.3 Error Handling and Validation

Robust error handling and validation are implemented throughout the application:

- **Form Validation:**
  All forms (login, register, upload, paste) check for required fields and valid input before processing. Missing or invalid inputs trigger user-friendly error messages.
- **File Handling:**
  Only allowed file types (PDF, DOCX, TXT) are accepted. Files are securely saved, processed, and deleted after use to prevent storage issues and security risks.
- **Session Management:**
  All sensitive routes check for active session; unauthenticated access attempts are redirected to login.
- **Database Integrity:**
  Unique constraint on usernames prevents duplicates. Database operations are wrapped in try-except blocks to catch and report errors10.
- **Graceful Degradation:**
  If language detection or translation fails, the system defaults to English and continues processing to avoid user disruption.

5.4 Load Testing and Performance Optimization

- **File Size Limits:**
  The application enforces a maximum upload size of 16MB to prevent excessive load and ensure timely processing.
- **Efficient Processing:**
  Text extraction, language detection, and summarization are optimized for speed. Uploaded files are deleted after processing to conserve disk space.
- **Database Queries:**
  Indexed queries and efficient data retrieval ensure fast loading of history and summaries, even as user data grows.
- **Scalability:**
  While the system is designed for single-server deployment, the use of lightweight technologies (Flask, SQLite) allows for easy migration to more robust solutions if needed.

6. Results and Discussion

6.1 Achievements

The document summarization platform successfully meets its core objectives and delivers a robust, secure, and user-friendly solution for automatic document summarization. The following achievements highlight the strengths of the system:

- **Multi-format Support:**
  Users can upload PDF, DOCX, and TXT files or paste text directly, making the platform versatile for various document sources.
- **Secure Authentication and Profile Management:**
  User registration and login are protected with bcrypt password hashing and session management, ensuring privacy and data security.
- **Language Detection and Translation:**
  The system detects the input language and, if necessary, translates it to English for summarization and back to the original language for output, supporting a global user base.
- **Customizable Summaries:**
  Users can select summary length (short, medium, long), with the backend dynamically adjusting the number of sentences in the summary8410.
- **Persistent Summary History:**
  All user summaries are stored in a history table, allowing users to view, download, and manage their previous summaries easily.
- **Responsive and Intuitive UI:**
  The interface is clean and accessible, with a sidebar for navigation and card-based layouts for content. The design is consistent across pages (dashboard, upload, paste, history, profile, result).
- **Efficient Processing and Security:**
  Uploaded files are processed and deleted after use, and all sensitive routes require authentication. The backend is optimized for quick summary generation and secure file handling.

6.2 Limitations

Despite its strengths, the platform has certain limitations:

- **Extractive Summarization Only:**
  The system uses extractive summarization (sentence ranking and selection) rather than abstractive approaches, which may not always capture the nuance or context of the original document.

- **Translation Quality:**
  Automated translation (Googletrans) may not always provide perfect results, especially for complex or technical documents.
- **File Size and Type Restrictions:**
  Maximum file upload size is limited to 16MB, and only PDF, DOCX, and TXT formats are supported.
- **No Collaborative Features:**
  The platform is designed for individual use; there is no sharing or collaboration on summaries.
- **Limited Customization:**
  Users cannot edit summaries within the platform or annotate documents.

6.3 Comparison: Computer vs Friends Mode

- **Computer Mode:**
  The platform automatically processes and summarizes documents using NLP algorithms, ensuring speed, consistency, and objectivity.
- **Friends Mode (Manual Summarization):**
  In a manual setting, users would need to read, interpret, and summarize documents themselves or rely on peers, which is time-consuming, subjective, and potentially inconsistent.

Feature	**Computer Mode (Platform)**	**Friends Mode (Manual)**
Speed	Instant	Slow
Consistency	High	Variable
Objectivity	Algorithmic	Subjective
Multi-language Support	Yes	Depends on user
History/Downloadable	Yes	Manual record-keeping

6.4 User Feedback and Improvements

Initial user feedback highlights the platform’s ease of use, clear navigation, and quick summary generation as major strengths. Users appreciate the ability to access summary history and download results. Suggestions for improvement include:

- Adding support for more file types (e.g., HTML, RTF)
- Providing options for summary editing or annotation
- Enhancing translation accuracy for non-English documents

- Integrating cloud storage for document management

6.5 Technical Evaluation

The platform demonstrates effective integration of Flask, SQLite, and NLP libraries (NLTK, pdfminer, docx, langdetect, googletrans, bcrypt)10. Security is robust, with hashed passwords and strict session management. The UI is responsive and intuitive, and backend processing is efficient for the target use case. The architecture allows for future scalability and feature enhancements.

7. Conclusion

The document summarization platform successfully delivers a secure, efficient, and user-friendly solution for automatic summarization of PDF, DOCX, and TXT files, as well as direct text input. By integrating modern Natural Language Processing (NLP) techniques, the system enables users to quickly extract key information from documents, supporting multiple languages and customizable summary lengths. Secure authentication, persistent summary history, and a responsive interface ensure a seamless user experience. The project demonstrates the effective integration of Flask, SQLite, and essential Python libraries to address the real-world need for rapid information extraction and document management.

8. Future Enhancement

While the platform meets its core objectives, several enhancements can further improve its utility and user experience:

- **Abstractive Summarization:**
  Incorporate advanced NLP models (such as transformer-based architectures) to provide more human-like, abstractive summaries.
- **Additional File Types:**
  Extend support to more document formats (e.g., HTML, RTF, ODT).
- **Collaborative Features:**
  Enable sharing and collaborative editing of summaries among users.
- **Cloud Integration:**
  Integrate with cloud storage platforms (Google Drive, Dropbox) for easier document management.

- **Summary Editing and Annotation:**
  Allow users to edit, annotate, or highlight summaries within the platform.
- **Enhanced Translation:**
  Improve translation quality and support more languages for global accessibility.
- **Mobile Optimization:**
  Develop a dedicated mobile app or further optimize the web interface for mobile devices.

9. Reference

- **Flask Documentation:** flask.palletsprojects.com
- **NLTK Documentation:** nltk.org
- **pdfminer Documentation:** github.com/pdfminer/pdfminer.six
- **python-docx Documentation:** python-docx.readthedocs.io
- **Googletrans Documentation:** py-googletrans.readthedocs.io
- **bcrypt Documentation:** pypi.org/project/bcrypt
- **langdetect Documentation:** pypi.org/project/langdetect
- **SQLite Documentation:** sqlite.org
- **Project Source Files:**
    - app.py (backend logic and routing)
    - HTML templates: dashboard.html, login.html, register.html, upload.html, paste.html, history.html, profile.html, result.html
    - CSS styling: style.css

Appendix: User Interface and Workflow Overview

Navigation and Layout

- **Sidebar Navigation:**
  All main pages (dashboard.html, upload.html, paste.html, history.html, profile.html) feature a sidebar with links to:
    - Dashboard
    - Upload Document
    - Paste Text
    - History
    - Profile
    - Logout
      The sidebar uses a dark background (#343a40) with white text and hover effects for usability.

- **Styling:**
  The interface employs a modern, card-based layout with a light background (#f8f9fa), rounded corners, and centered containers for forms. Pages are responsive and visually consistent across devices.

Main Pages and Features

Page	**Description**
Dashboard	Welcomes the user by name and provides quick links to all main features1.
Upload Document	Lets users upload PDF, DOCX, or TXT files, select summary length (short, medium, long), and submit for processing.
Paste Text	Allows direct text input, summary length selection, and instant summarization4.
History	Displays all previous summaries with filename, preview, timestamp, and actions to view/download. If no summaries exist, a prompt is shown.
Profile	Shows the logged-in username and provides a logout button.
Result	Presents the generated summary, with options to copy or download as a TXT file.
Login/Register	Secure forms for authentication, with error feedback for invalid credentials or duplicate usernames

Workflow Summary

1. **Authentication:**
   Users register or log in with secure password handling (bcrypt) and Flask session management. Passwords are hashed before storage, and sessions track logged-in users.
2. **Document Submission:**
   - **Upload:** Users select a document and summary length, then submit. The backend extracts text, detects and handles language, summarizes, and displays the result. Uploaded files are deleted after processing.
   - **Paste:** Users paste text, select summary length, and receive a summary instantly.
3. **Summary Management:**
   All summaries are stored and displayed in the user's history, with options to view or download results as TXT files. Each summary record includes filename, preview, timestamp, and actions.
4. **Profile and Logout:**
   Users can view their profile and securely log out at any time.

Security and Usability

- **Session-based access control:**
  All sensitive routes require authentication; unauthenticated users are redirected to login.
- **Input validation:**
  Only allowed file types and sizes are accepted; errors are clearly communicated on the frontend and backend.
- **Responsive design:**
  The layout adapts to different devices and screen sizes for accessibility, as defined in the CSS.