



BSCS-S24-017

03-134202-102

Rohaam Nadeem

03-134211-033

Noor Fatima

Extraction Of User Defined Information from PDF

In partial fulfilment of the requirements for the degree of
Bachelor of Science in Computer Science

Supervisor: Tahir Iqbal

Department of Computer Sciences
Bahria University, Lahore Campus

January 2025

Certificate



We accept the work contained in the report titled
Extraction Of User Defined Information from Pdf
written by

03-134202-102

Rohaam Nadeem

03-134211-033

Noor Fatima

as a confirmation to the required standard for the partial fulfilment of the degree of
Bachelor of Science in Computer Science.

Approved by:

Supervisor:

Tahir Iqbal

(Signature)

December 24, 2024

DECLARATION

We hereby declare that this project report is based on our original work except for citations and quotations which have been duly acknowledged. We also declare that it has not been previously and concurrently submitted for any other degree or award at Bahria University or other institutions.

Enrolment	Name	Signature
03-134202-102	Rohaam Nadeem	
03-134211-033	NoorFatima	

Date : December 24, 2024

Specially dedicated to

My beloved father, mother. All our teachers, and of course our supervisor
Tahir Iqbal who brought us up as the way we are today.

Rohaam Nadeem
Noor Fatima

ACKNOWLEDGEMENTS

We would like to thank everyone who has contributed to the successful completion of this project. We would like to express our gratitude to my research supervisor, Mr.Tahir Iqbal for his invaluable advice, guidance and his enormous patience throughout the development of the research.

Rohaam Nadeem
Noor Fatima

Extraction Of User Defined Information from PDF

ABSTRACT

Problem and Significance: Organizing and utilizing information contained in PDF files present different difficulties arising from the different organization of such documents. As shown in **Figure 3.3.1**, the methodology adopted highlights the systematic approach for solving these challenges.[\[Polak & Morgan, 2024\]](#) Due to the increasing need and availability of large documents in PDF format, lack of an efficient mechanism to upload these documents and query their contents automatically without supervised interventions with extracting user-specified information, this project comes as a response for these needs. The relevance of this project is in proposing an intelligent and efficient way of dealing with research materials in the PDF format and in gaining the desired piece of information from it thus enhancing research analysis and information process in general across numerous disciplines.

Method/Tool/Technology and Solution: The proposed system forms text data from uploaded PDFs using NLP techniques that are powered by LLMs to provide automation in the extraction of pertinent data from the pieces presented in PDF format. PDFs' data extracted using PyMuPDF, meta-analyses using Matplotlib, chunking large queries, and user query analysis is conducted through an LLM specifically a ChatGPT API. The front end of the application is created with help of React, while back-end part is implemented by Express and Python. It saves the file path, chat and user data; the entire system exists inside a safe and user-friendly space. This solution saves users' time and minimizes their efforts when searching for specific data in PDFs while ensuring increased efficiency of the data search process.

TABLE OF CONTENTS

DECLARATION	iii
ACKNOWLEDGEMENTS	vii
ABSTRACT	ix
TABLE OF CONTENTS	xi
LIST OF TABLES	xiii
LIST OF FIGURES	xv
LIST OF SYMBOLS / ABBREVIATIONS	xvii

CHAPTERS

1	1		
	1.1	Background	1
	1.2	Problem Statements	1
	1.3	Aims and Objectives	2
	1.4	Scope of Project	2
	1.5	This system excludes the following	3
2	4		
	2.1	Product Viewpoint	4
	2.2	Operating Environment	4
	2.3	Functional Requirements	5
	2.4	Non-Functional Requirements	6
	2.5	User Requirements	7
	2.6	Assumptions and Constraints	7
	2.7	Use case Diagram	8
	2.8	Product Perspective	8
3	10		
	3.1	Methodology	10
	3.2	Feature-Driven Development	10
	3.3	Tools and Techniques	11
4	12		
	4.1	Introduction	12
	4.2	Story Board	12
	4.3	System Design	13
	4.4	Sequence Diagram	13

5	14		
	5.1	Dashboard Page	14
		5.1.1 Methodology Diagram	14
		5.1.2 About project	15
		5.1.3 Project Workflow and Implementation Steps	15
	5.2	Auth Page:-	16
	5.3	Upload Page	17
		5.3.1 Chat Page	19
		5.3.2 Profile Page	22
6	24		
	6.1	Conclusion	24
	6.2	Recommendations	25
	6.3	Future Work	25
7	26		

LIST OF TABLES

TABLE	TITLE	PAGE
	Table 2.1: Functional Requirements	3
	Table 2.2: Non-Functional Requirements	4
	Table 2.3: Assumptions and Constraints	5
	Table 2.4: Product Perspective	6

LIST OF FIGURES

FIGURE	TITLE	PAGE
Figure 2.2.1:	Use case Diagram	5
Figure 3.3.1:	Methodology	7
Figure 4.1:	Sequence Diagram	10
Figure 5.1:	Project Title with Sidebar Featuring Logo and Links	11
Figure 5.2:	Project Overview	12
Figure 5.3:	Project Workflow and Steps	12
Figure 5.4:	Sign-up Flow with Google Login Option	13
Figure 5.5:	Login Form with Google Login Option	13
Figure 5.6:	Google Login Account Selection	14
Figure 5.7:	PDF Upload Page and google user in Sidebar	14
Figure 5.8:	User Icon and Email Displayed in Sidebar for Form Users	15
Figure 5.9:	PDF Upload with Filename, File size, and Remove Option	15
Figure 5.10:	Multiple PDFs Uploaded with Filename, File size, and Remove Option	16
Figure 5.11:	Redirect to Chat Page After File Upload	16
Figure 5.12:	Minimized Sidebar View	17
Figure 5.13:	Selected Previous Chat from Sidebar	17
Figure 5.14:	Option to Delete Chat from Sidebar	18
Figure 5.15:	Option to Edit Chat Name from Sidebar	18
Figure 5.16:	Google User Profile Page	19

Figure 5.17: Form User Profile Page with Edit Option for Username and Email	19
Figure 5.18: Option to Change Password in User Profile	20

CHAPTER 1

INTRODUCTION

1.1 Background

Documents in portable format and images can provide important data, however, to separate the required by the user information can take quite a lot of time. [\[Tiedemann, 2014\]](#). This challenge is further illustrated in the **Use Case Diagram (Figure 2.2.1)**, which depicts the interaction and requirements for the proposed system. With the increase in the rate at which people desire to get quick and accurate results in their work output ranging from research works, health diagnosis and even data analysis the need for automatic solutions cannot be overemphasized. This project will therefore seek to deal with these by proffering a web-based tool that can extract user-specified data from pdf documents. Moreover, through the use of LLM for NLP techniques, the tool will have a user-friendly interface where the users will insert variables and get simplified relevant data from complicated PDF files, also to save their record.

1.2 Problem Statements

One of the main issues is the possibility of extracting data from various PDF formats and images in case of working with big data sets. Many professionals who work with research materials, particularly those who deal with reports, brochures, magazines, encyclopaedias, journals, and other collections of papers, know the tremendous amount of time they spend on searching for particular pieces of information in out-of-order documents. As per this project, such a solution will be presented to primarily extract the user-centric content from PDFs.

1.3 Aims and Objectives

The goal of this project is thus to create an efficient, almost completely autonomous instrument, easily operated by the user, which is designed to extract desired information from the pdf files. The objectives include:

- i) To design an interactive computer system, which mainly consists of an application interface through which users can input key variables and get their corresponding answers in a PDF document.
- ii) To facilitate and add large language models for addressing Natural Language Processing (NLP).
- iii) To help cut down the time and energy that is taken in the sorting out of documents in an efficient way, hence increasing productivity.
- iv) ii) To offer the users concerns in research and analytical work and other fields dealing with large dataset. Icons for professionals in research, data analysis, and other fields dealing with large volumes of data.

1.4 Scope of Project

The project is centred on creating a system for text-based PDFs which contain text that can be easily extracted by a program. The tool will allow users to:

- 1. Just drop one or more PDFs for processing.
- 2. Access it like augmented reality chatbot where the user can directly ask specific information.
- 3. Save the results to check the information needed during the performance of the analysis and to compare them with other sources.
- 4. Provide meta-analysis of extracted data if required on request.

1.5 This system excludes the following

For example, text recognition from digital images of text, or scanned documents and PDFs (Optical Character Recognition or OCR tasks).

High levels of graphical or tabular data that cannot be converted into a plain text form.

There are pretty good potentialities for future extensions of the system: It would be helpful to include OCR, to extend beyond simple queries and more complex kinds of data visualizations.

CHAPTER 2

Software Requirements Specification (SRS)

2.1 Product Viewpoint

The purpose of this project is to offer an internet application, where the users could get information of interest from PDF files. The system will also use LLMs to deal with the content in a way that will allow for a swift and effective search of data defined by the users. The platform will be useful for such users as researchers, data analysts, and other employees who deal with the manipulation of large data sets to

2.2 Operating Environment

This application will perform in different web browsers, like:

- Google Chrome
- Mozilla Firefox
- Opera
- Safari

The platform will be optimized for both desktop and mobile usage.

2.3 Functional Requirements

Table 2.1: Functional Requirements

ID	Requirement	Priority.1
FR-1	User registration as well as user login with options for google sign-in.	High
FR-2	Submit a number of PDF files at once for conversion.	High
FR-3	Extract text from PDFs using PyMuPDF.	High
FR-4	Return extracted text based on queries in terms of natural language.	High
FR-5	Give analysis of the meta-extraction of information on online consultations.	Medium
FR-6	It's always wise to save the information from previous chats for future use.	High
FR-7	Introducing the ability to remove or rename a chat.	Medium
FR-8	Matplotlib for meta-analyses.	Medium

2.4 Non-Functional Requirements

Table 2.2: Non-Functional Requirements

Requirement	Description
Response Time	Process PDFs and queries in under 5 seconds for files whose size does not exceed 10 MB.
Scalability	Patients, other doctors and other clinicians can use this application at the same time without straining the systems capabilities.
Security	Secure user data and cryptographic control of program data, as well as the protection of API communication.
Accessibility	Ensure that it has a theme that would enable it's usage in both the large desktop and the small mobile screens.
Reliability	Make certain that the system needs a small duration of time whereby it is not in a position to perform its planned function.

2.5 User Requirements

The system must meet the following user requirements:

1. **Ease of Use:** User should be able to upload files, query data and organize their sessions in a friendly way.
2. **Customizable Queries:** The system must enable users to declare their information requirements by means of natural language inputs.
3. **Data Access:** Users gain the ability to view previous queries and chats together with the functionality of renaming the saved items and removing them permanently.
4. **Secure Authentication:** Ensure that there are multiple ways for logging in securely which should include Google OAuth.

2.6 Assumptions and Constraints

Table 2.3: Assumptions and Constraints

Type	Description
Assumptions	It assumes that the PDFs uploaded are text-based files. and not image based.
Constraints	However, natural language questioning calls for internet connection as it is the case with most other systems. and API operations.

2.7 Use case Diagram

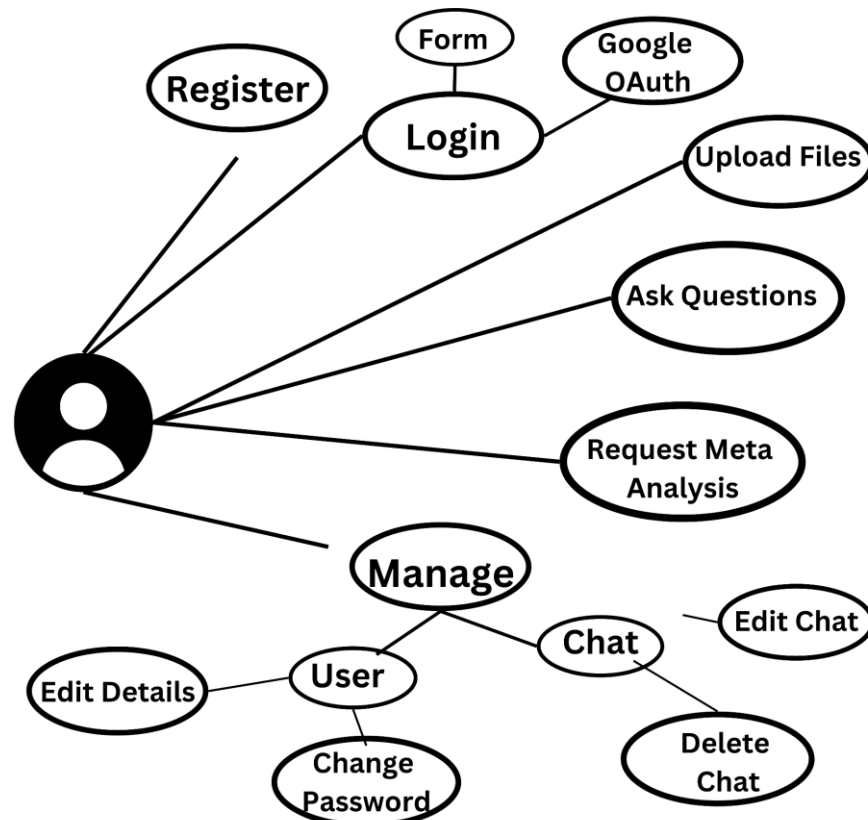


Figure 2.2.1: Use case Diagram

2.8 Product Perspective

It is an independent web based system for users who require extracting and analyzing certain data from PDF files. The software connects several state-of-art technologies to support product functionality, and it is intended for researchers, students, and professionals.

Table 2.4: Product Perspective

Aspect	Description
System Dependency	Requires a browser-based internet connected frontend server and backend server
Third-Party APIs	Integrated with LLM APIs of natural language query and analysis.
Integration	Based on Google OAuth for authentication and MongoDB for data persistence and storage.
User Role	The primary user has explicit interactions with the system for uploading files, for performing data queries,

|

CHAPTER 3

DESIGN AND METHODOLOGY

3.1 Methodology

This work is developed with an Agile-based approach with a constant feedback loop between iterations. This process is visualized in **Figure 3.3.1** (Methodology). It studies also include testing and subsequent updates of a development cycle [\[Polak & Morgan, 2024\]](#) and features it supports such as User Authentication, File uploading, and Data extraction services.

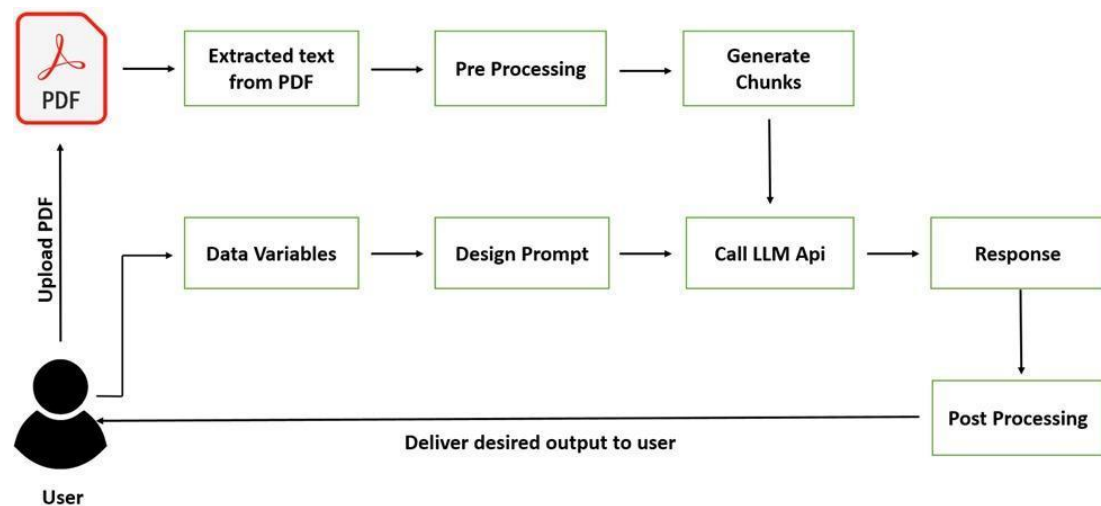


Figure 3.3.1: Methodology

3.2 Feature-Driven Development

Our chosen process model is FDD: feature based, which focuses on iteration and continuous delivery. Every one of the above features is developed and implemented as a module on its own but with the potential to interact with other modules in the LLM-Api.

3.3 Tools and Techniques

- Frontend: JS for interactive websites and react hence for user interface design.
- Backend: Express.js to manage requests, and Python to parse data from delivered requests.
- Database: MongoDB to store users data and files path, chats description and users' details.
- Data Extraction: PyMuPDF for the purpose of extracting text from PDFs.
- Graphical Response: Matplotlib for meta analyses.
- LLM API: For performing on user inputs and providing smart output in response.

CHAPTER 4

IMPLEMENTATION

4.1 Introduction

The subject of the project is construction of the system for file management and interaction at all levels. Video users can upload files, post questions related to the files and get answers with the help of artificial intelligence.

4.2 Story Board

- The dashboard layout, featuring the project title, sidebar, and links, is depicted in **Figure 5.1** (Project Title with Sidebar).
- Login/Signup: The user then gets to select registration or login.
- File Upload: User are lead to pause /upload PDF files.
- Chat Interface: Once the document is uploaded, a user is redirected to a chat interface in which he or she can ask a PDF content.
- Response Generation: An LLM delivers the text extraction as well as the processed output of the query.

4.3 System Design

The system consists of the following modules:

- **User Authentication:** employs a secure procedure to create accounts and to allow users, to sign in.
- **File Upload and Storage:** Files are stored on the server and when a user uploads something the information is placed in the MongoDB database information such as the files path, details on chats between users and more.
- Users can upload PDFs, as demonstrated in **Figure 5.7** [\[Tiedemann, 2014\]](#) (PDF Upload Page). Multiple PDFs can be uploaded with file details, as shown in **Figure 5.9** and **Figure 5.10**.
- **Text Extraction:** As it will be shown, PyMuPDF makes an extraction of text out of the uploaded files.
- **Graphical Response:** Matplotlib for meta analyses.
- **Query Processing:** The LLM API is utilized to recognize a user query which is identified from the scraped text.
- The interaction between different system components, including user authentication, file uploads, and text extraction, is represented in **Figure 4.1** (Sequence Diagram).

4.4 Sequence Diagram

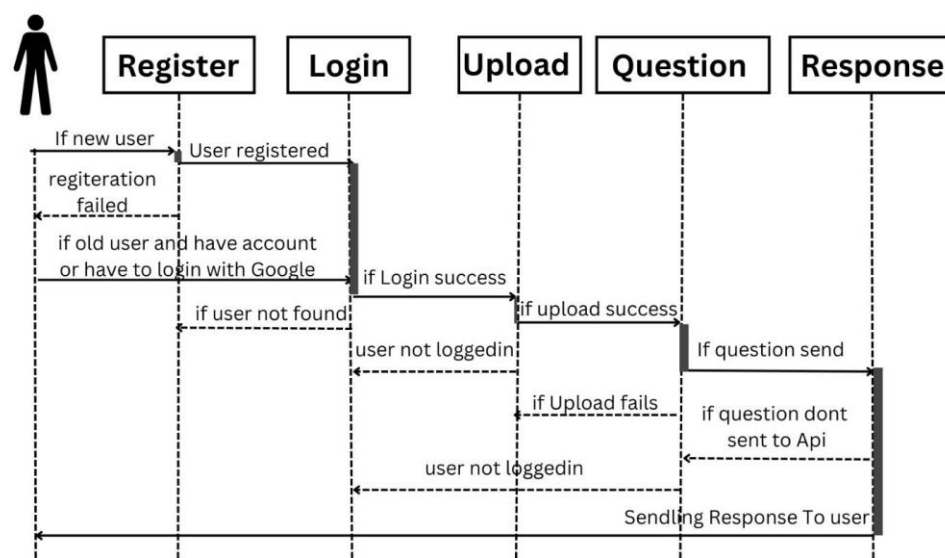


Figure 4.1: Sequence Diagram

CHAPTER 5

RESULTS

5.1 Dashboard Page

The sidebar, therefore, is not extendable as it is not scrollable – the only action one can take on it is to expand them or reduce them. On the Dashboard page the project information is proposed in the left panel whereas the personal information of the team members is available in the right panel of the page. The dashboard design, with the sidebar and personal information panel, is illustrated in **Figure 5.1**.[\[Goel et al., 2023\]](#)

5.1.1 Methodology Diagram

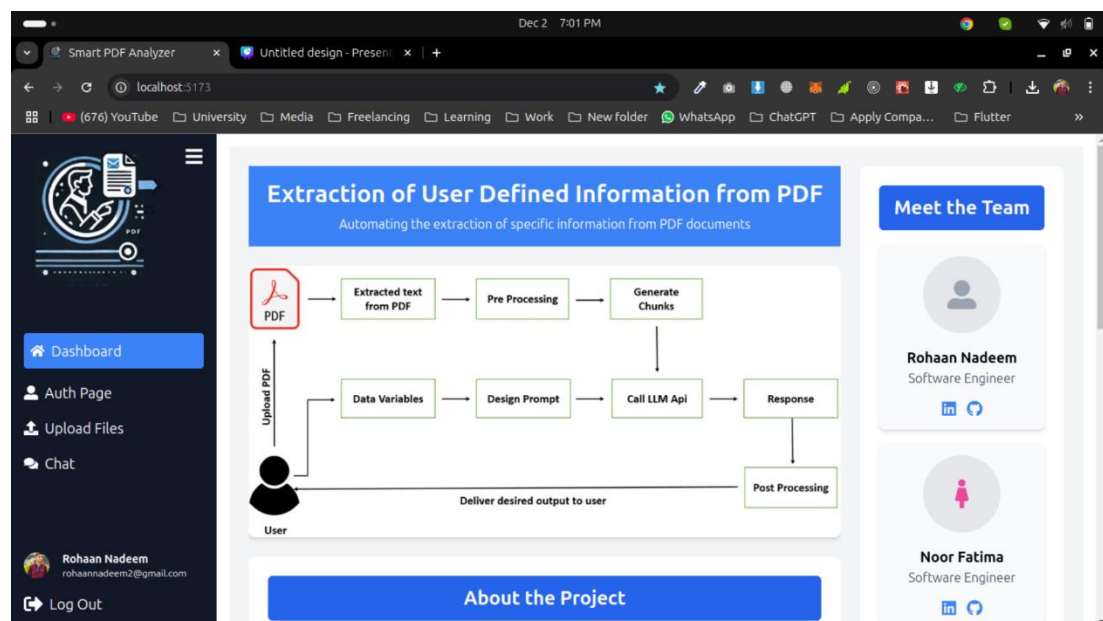


Figure 5.1: Project Title with Sidebar Featuring Logo and Links

5.1.2 About project

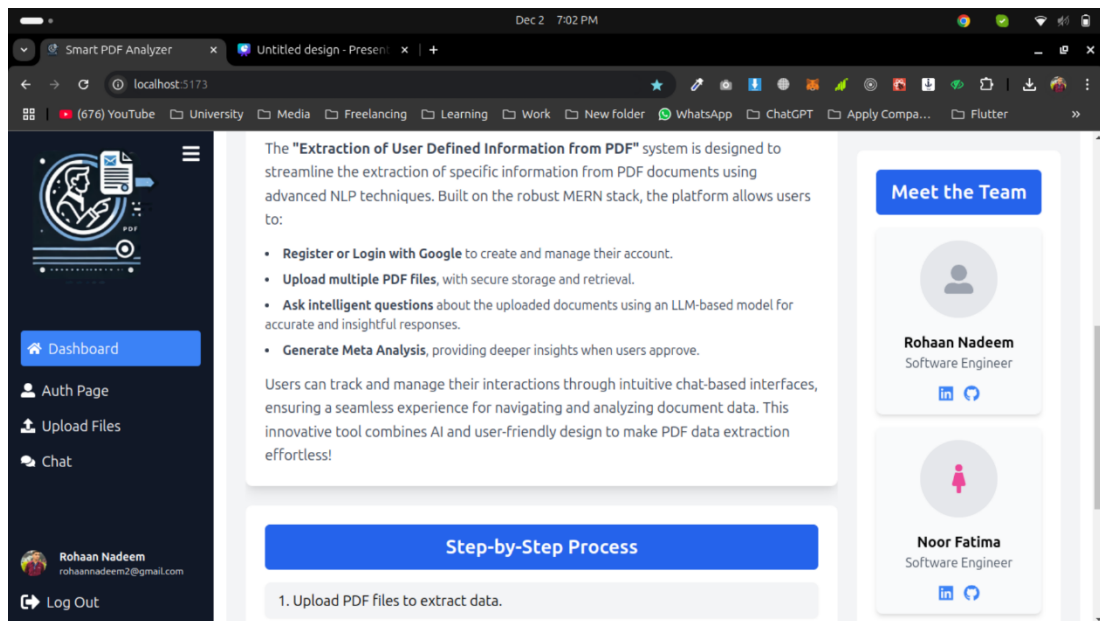


Figure 5.2: Project Overview

5.1.3 Project Workflow and Implementation Steps

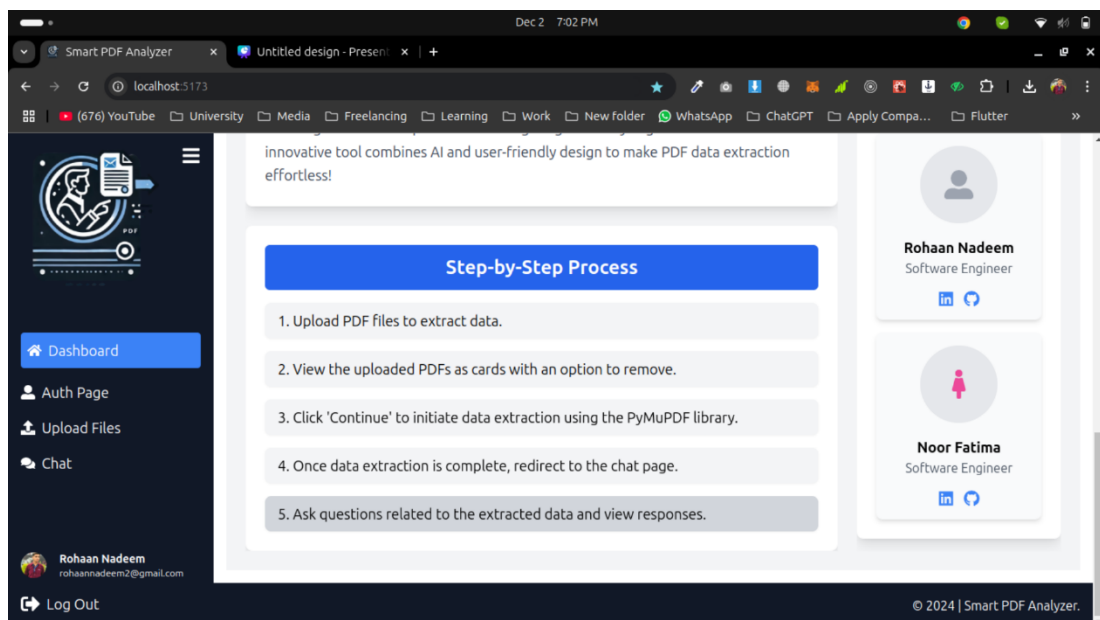
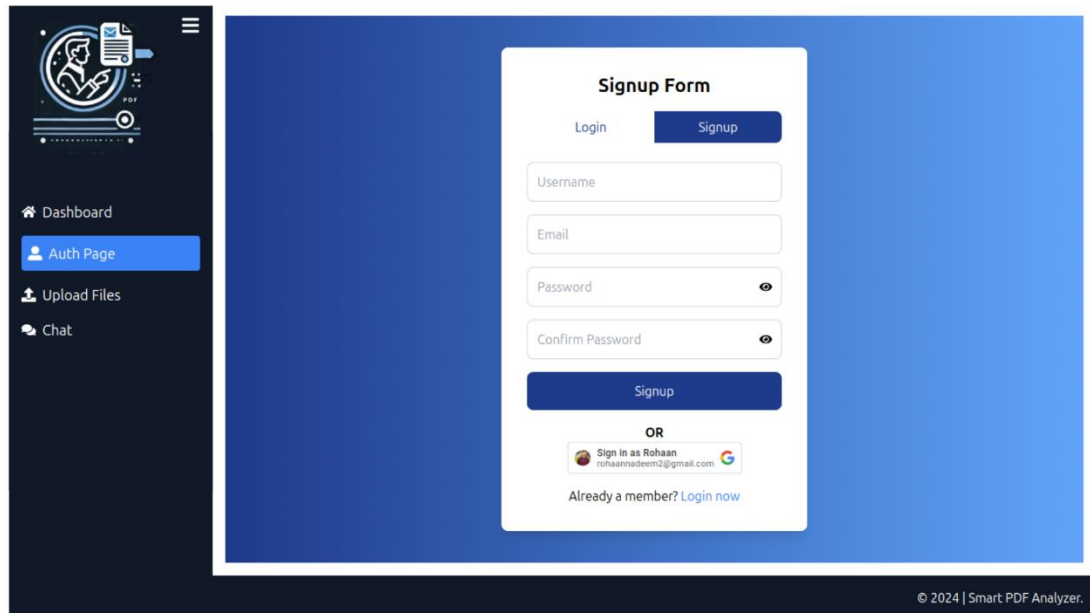


Figure 5.3: Project Workflow and Steps

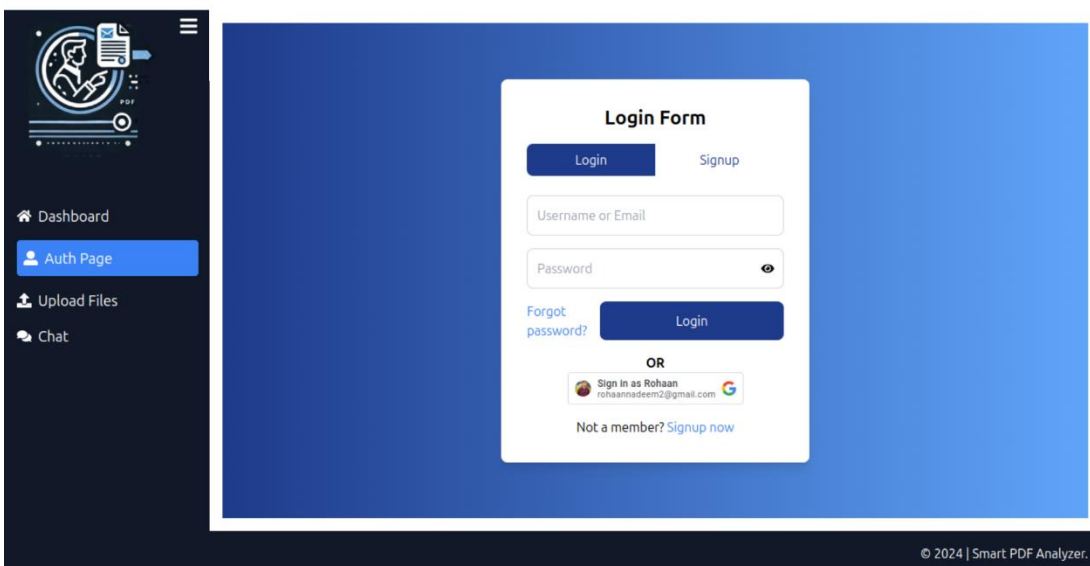
5.2 Auth Page:-

In case user is not logged in then, auth page can be open. The user authentication flow, including Google sign-in, is depicted in **Figure 5.4** (Sign-up Flow) and **Figure 5.5** (Login Form).



The screenshot displays the 'Sign-up Form' on a dark blue background. On the left, a sidebar menu contains icons and labels for 'Dashboard', 'Auth Page' (highlighted), 'Upload Files', and 'Chat'. The main form area has a white background and includes a 'Login' button and a 'Signup' button. Below these are input fields for 'Username', 'Email', 'Password', and 'Confirm Password', each with a toggle icon. A 'Signup' button is positioned below the input fields. Underneath, there is an 'OR' separator, a Google sign-in button labeled 'Sign in as Rohaan rohaannadeem2@gmail.com', and a link 'Already a member? Login now'. The footer of the page reads '© 2024 | Smart PDF Analyzer.'

Figure 5.4: Sign-up Flow with Google Login Option



The screenshot displays the 'Login Form' on a dark blue background. On the left, a sidebar menu contains icons and labels for 'Dashboard', 'Auth Page' (highlighted), 'Upload Files', and 'Chat'. The main form area has a white background and includes a 'Login' button and a 'Signup' button. Below these are input fields for 'Username or Email' and 'Password', each with a toggle icon. A 'Forgot password?' link is located below the 'Password' field. A 'Login' button is positioned below the input fields. Underneath, there is an 'OR' separator, a Google sign-in button labeled 'Sign in as Rohaan rohaannadeem2@gmail.com', and a link 'Not a member? Signup now'. The footer of the page reads '© 2024 | Smart PDF Analyzer.'

Figure 5.5: Login Form with Google Login Option

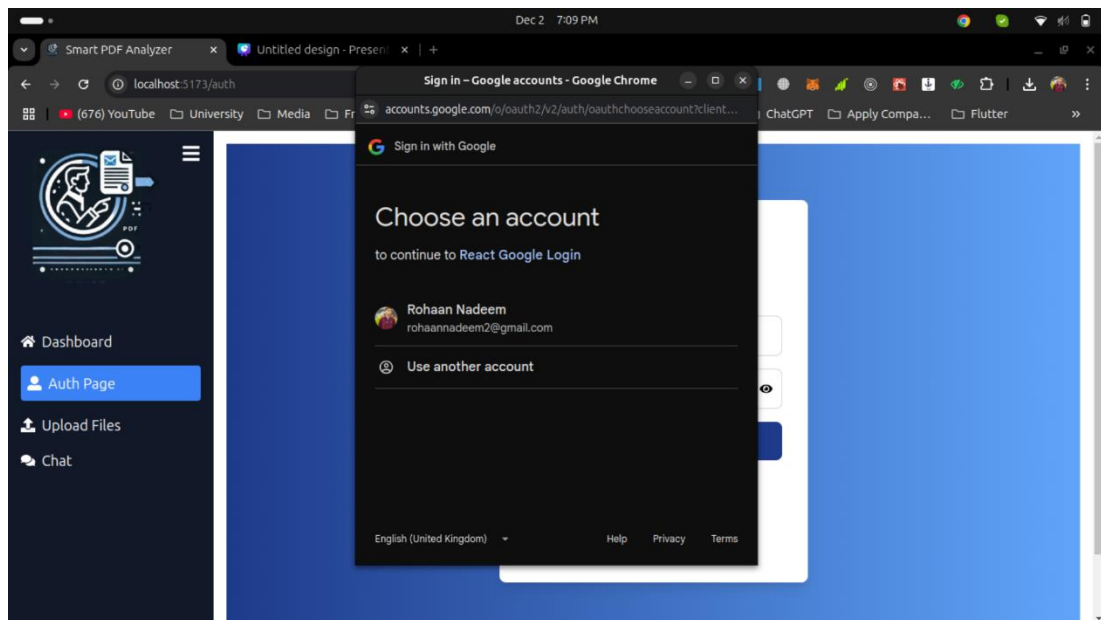


Figure 5.6: Google Login Account Selection

5.3 Upload Page

User can upload or drag pdf files over here if user is logged in with google than its google image will be placed in the side bar and its detail (name and email). The interface for uploading PDFs and managing uploaded files is detailed in **Figures 5.7, 5.9, and 5.10.**"

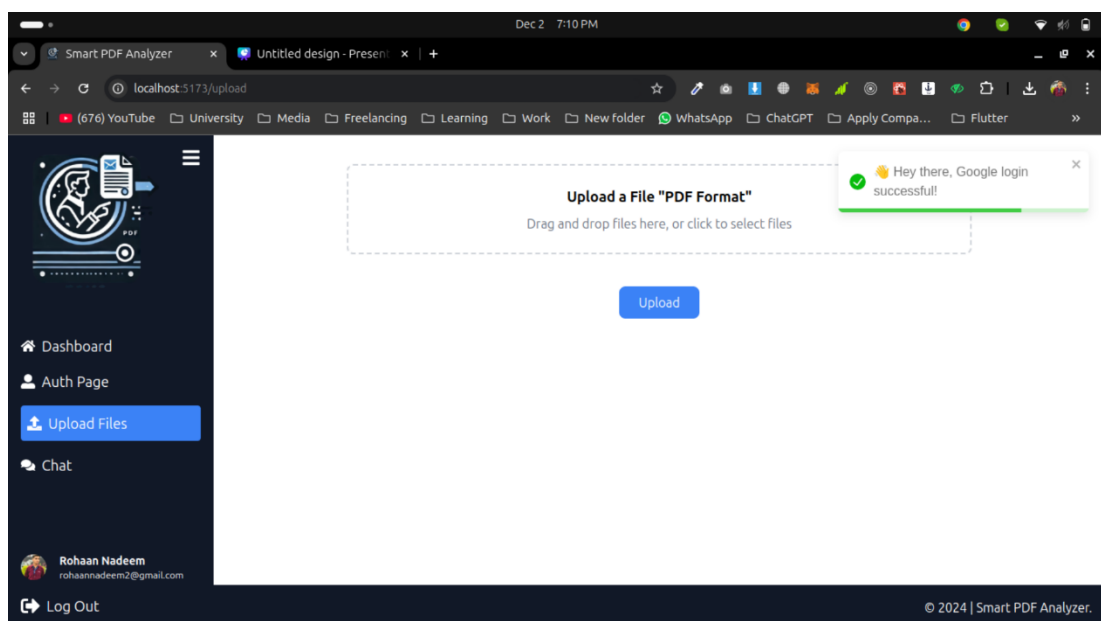


Figure 5.7: PDF Upload Page and google user in Sidebar

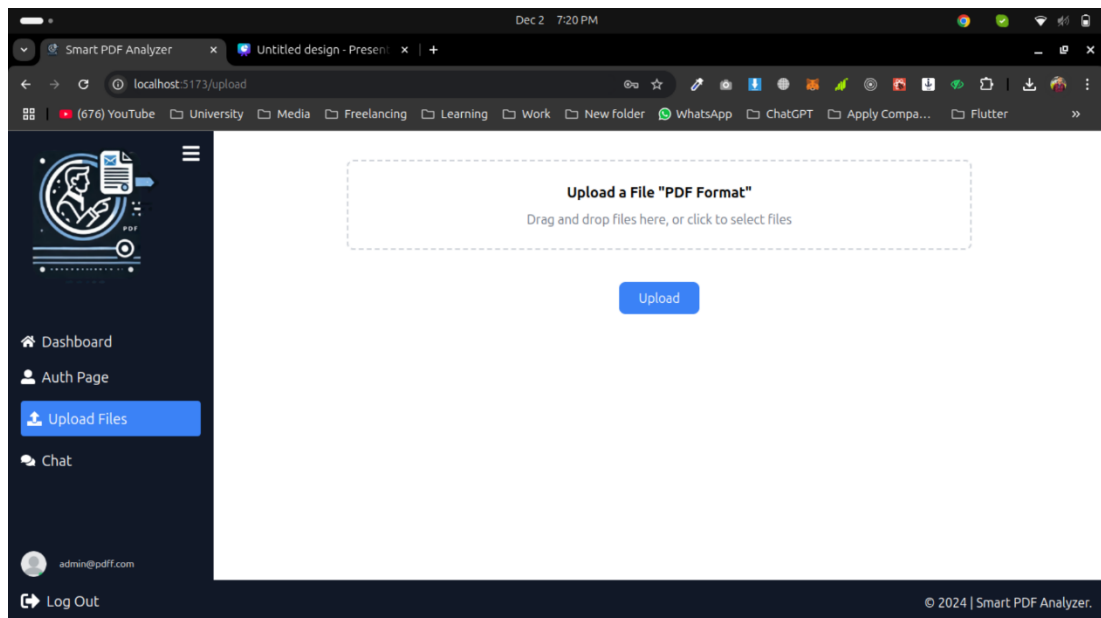


Figure 5.8: User Icon and Email Displayed in Sidebar for Form Users

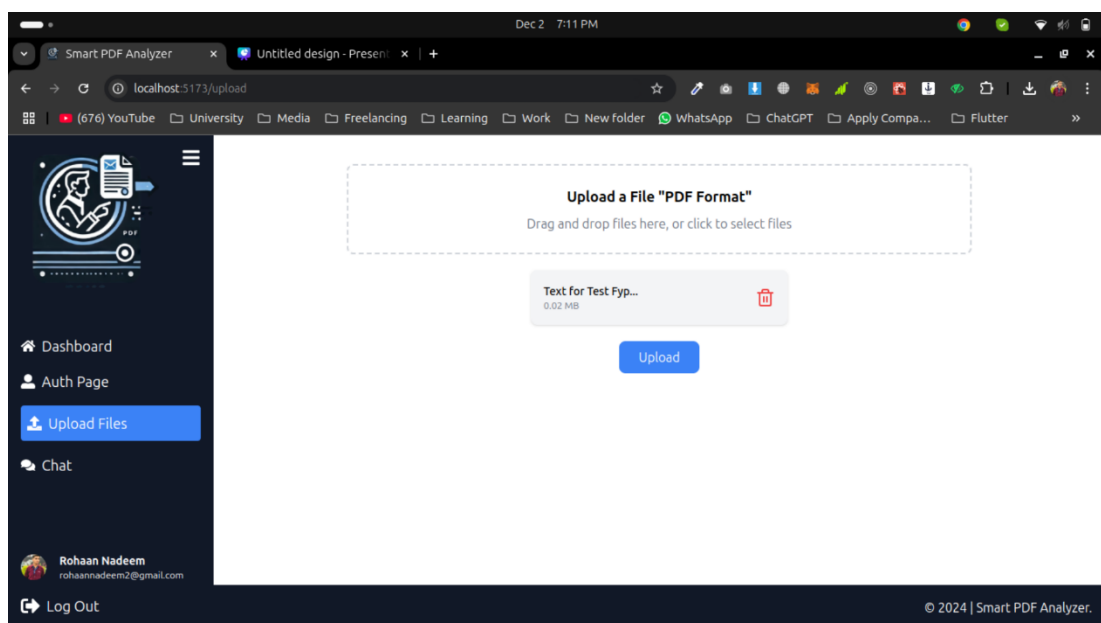


Figure 5.9: PDF Upload with Filename, File size, and Remove Option

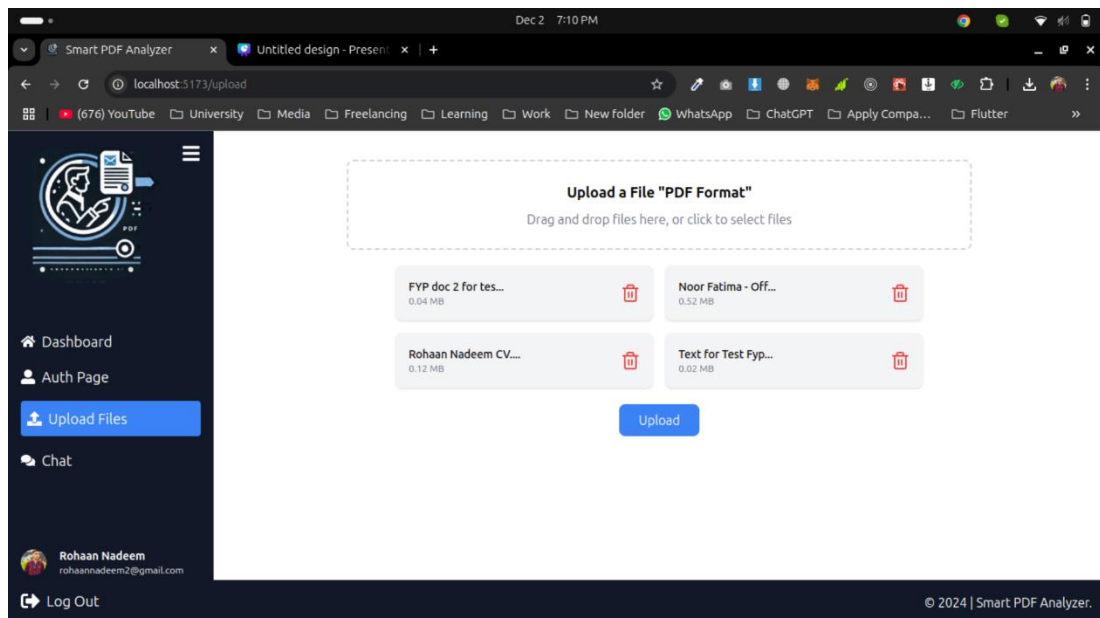


Figure 5.10: Multiple PDFs Uploaded with Filename, File size, and Remove Option

5.4 Chat Page

Chat Page contains to have a sidebar showing the previous chat records that include the options of editing/ deleting it. The right side is for the live chat window for communicating with the uploaded data from the PDF format. The chat interface, allowing users to query extracted PDF content, is shown in **Figure 5.11** [Polak & Morgan, 2024] (Redirect to Chat Page). Options for managing previous chats are illustrated in **Figures 5.13 and 5.15**.

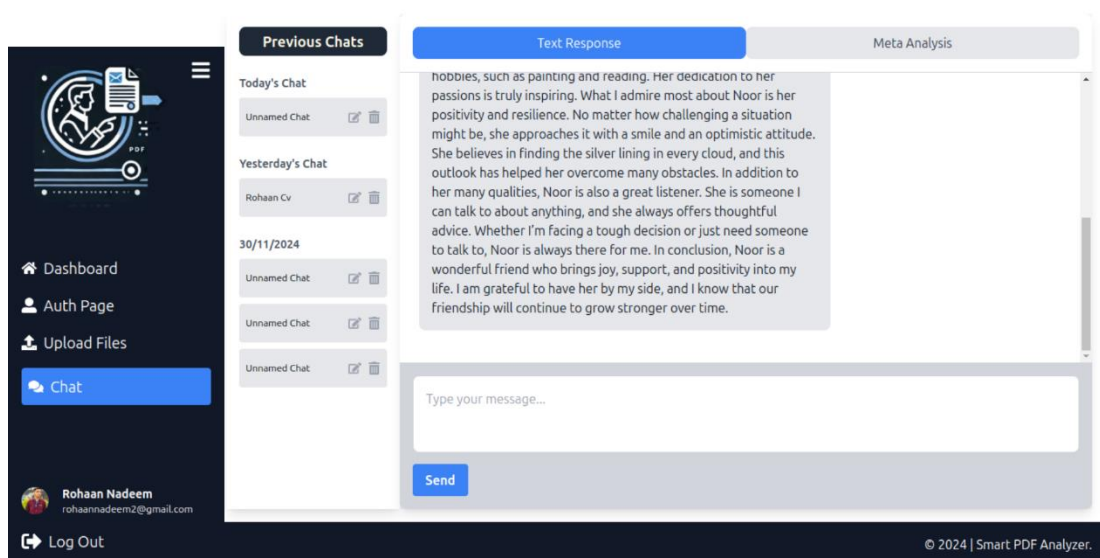
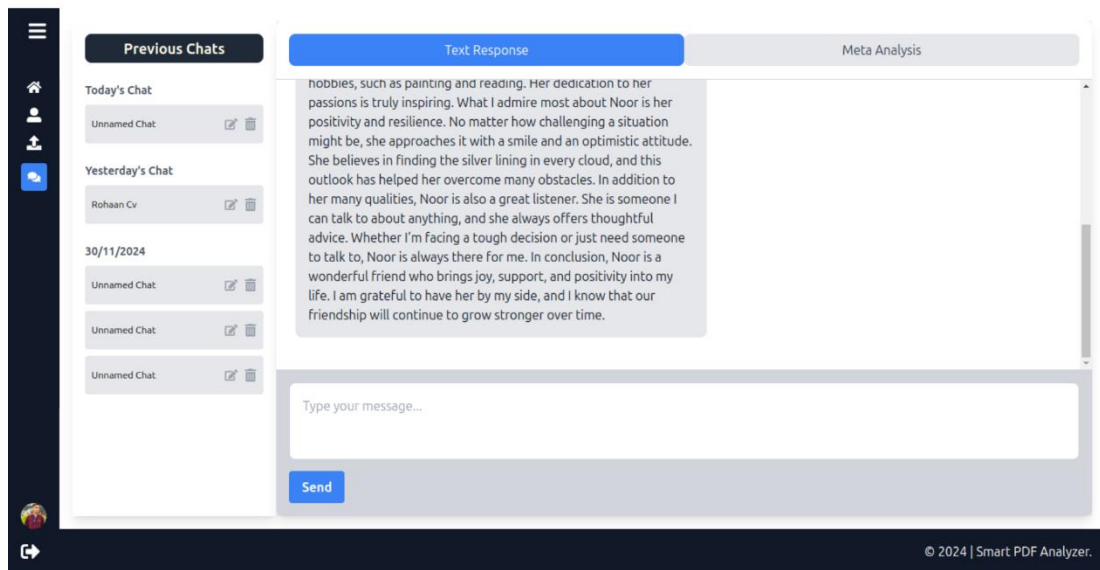
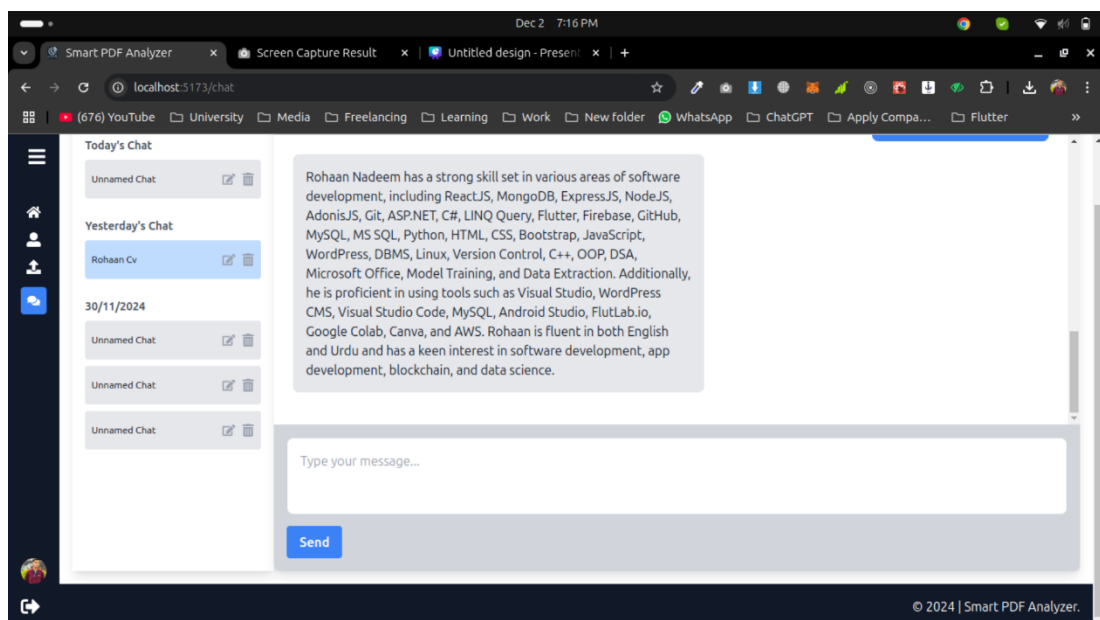


Figure 5.11: Redirect to Chat Page After File Upload**Figure 5.12: Minimized Sidebar View****Figure 5.13: Selected Previous Chat from Sidebar**

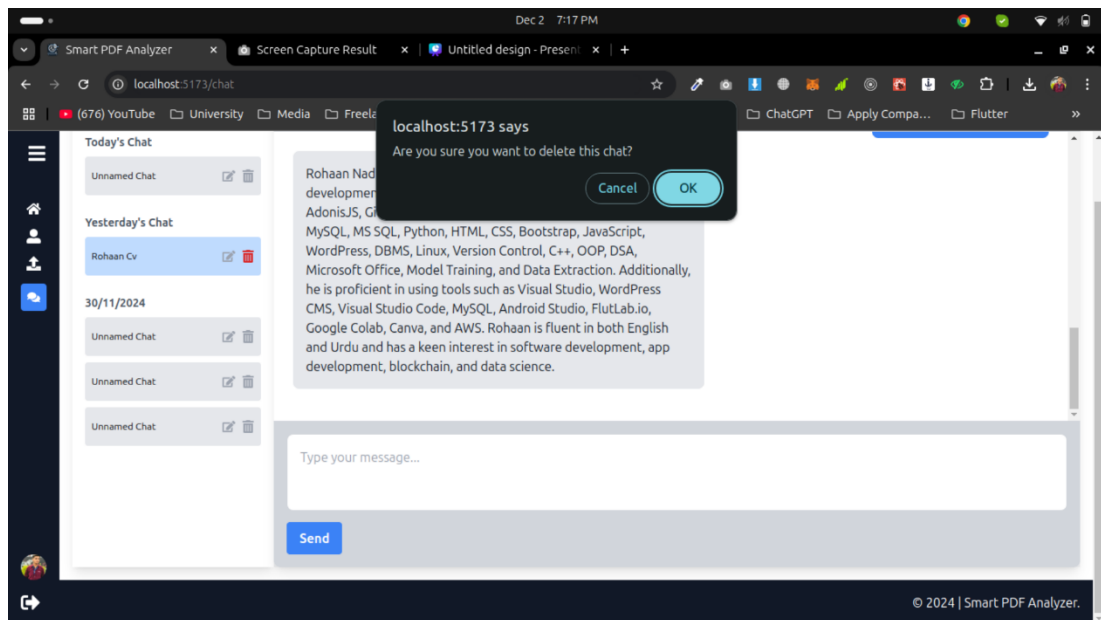


Figure 5.14: Option to Delete Chat from Sidebar

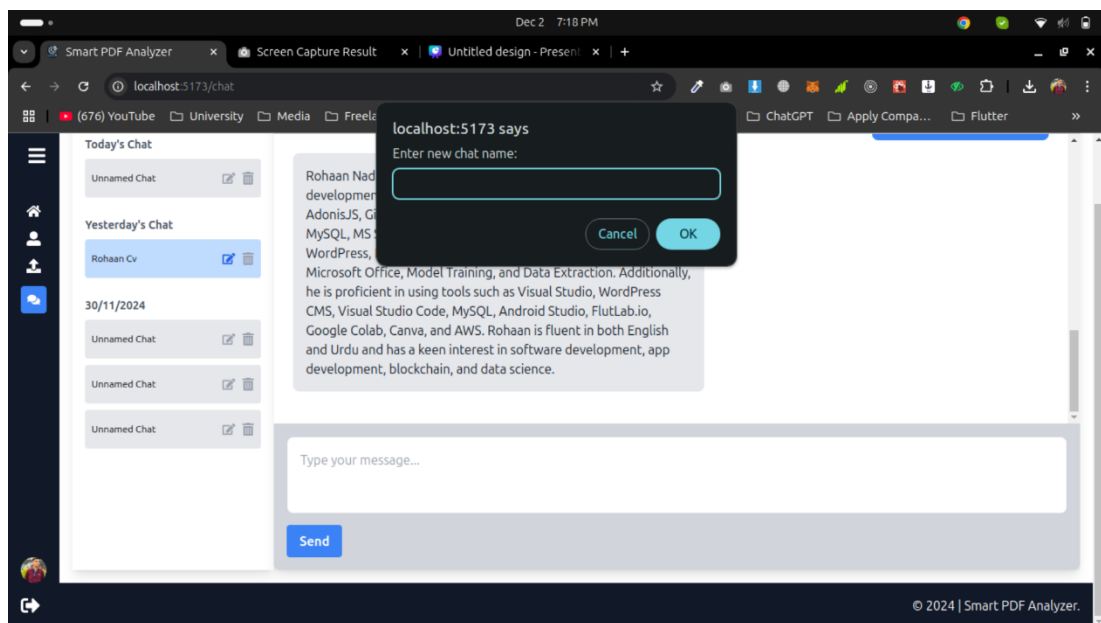


Figure 5.15: Option to Edit Chat Name from Sidebar

5.5 Profile Page

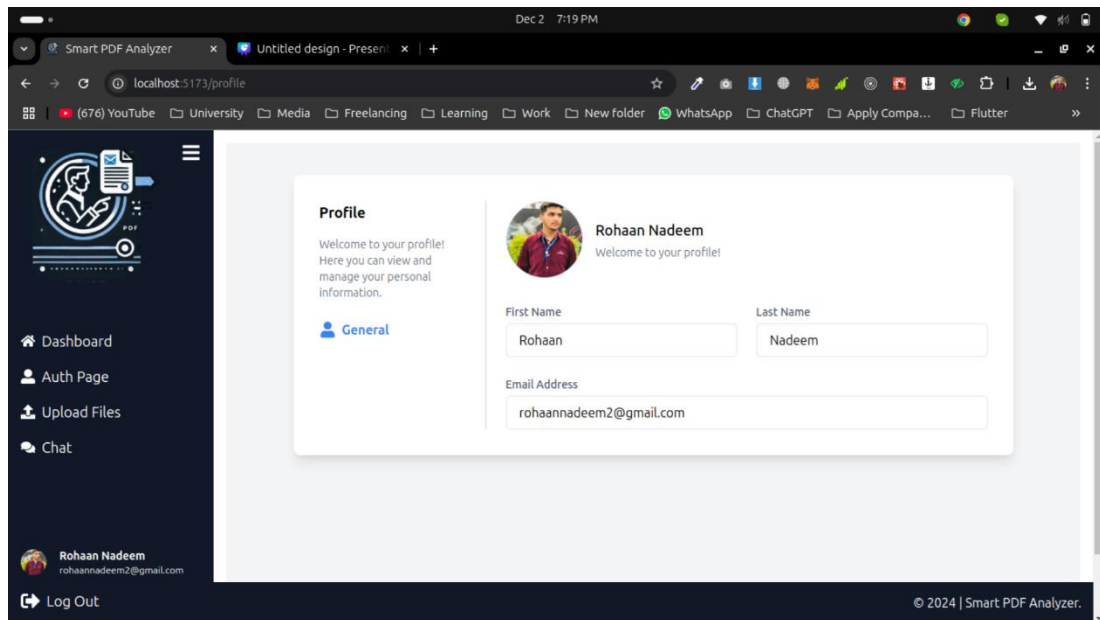


Figure 5.16: Google User Profile Page

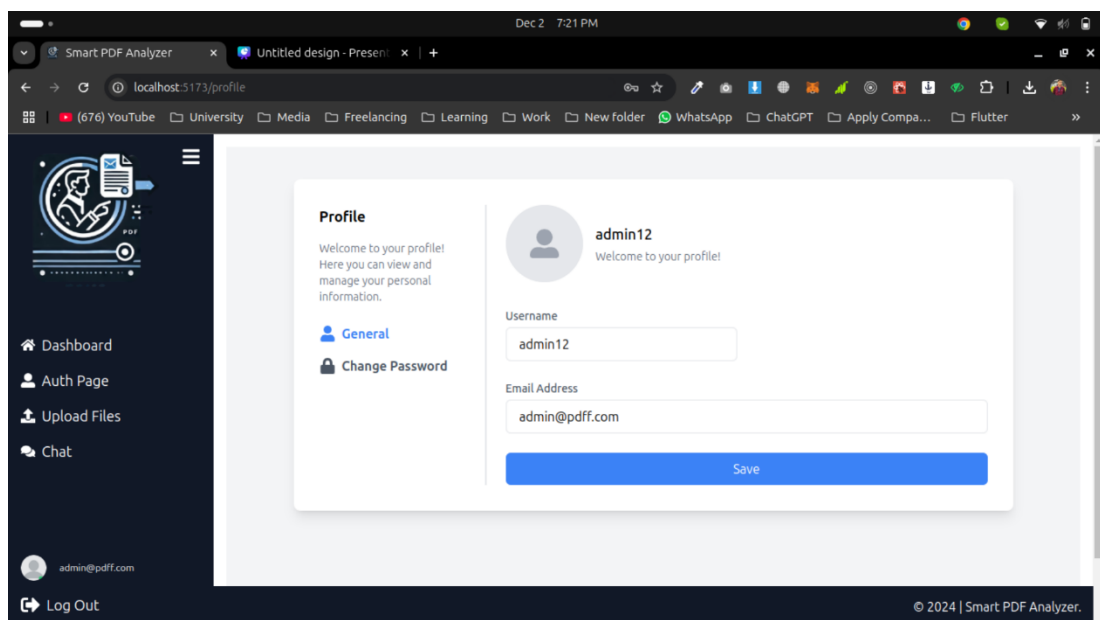


Figure 5.17: Form User Profile Page with Edit Option for Username and Email

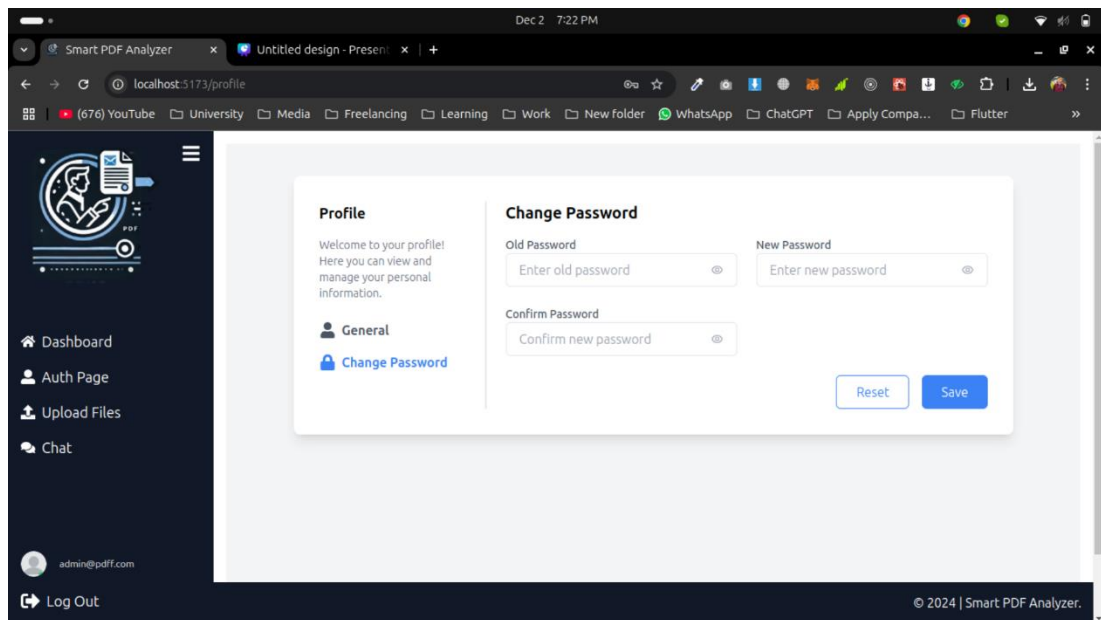


Figure 5.18: Option to Change Password in User Profile

CHAPTER 6

CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

The proposed application named as “Extraction of User Defined Information from PDF” has been successfully deployed with a complete full-stack web configuration wherein the users are able to upload PDF file, extract the important data by using Python pyMuPDF and Meta analysis having visualization of the extracted content through Python Matplotlib library and finally converse with the extracted content through an intelligent chat module. Using the MERN stack (MongoDB, Express.js, React), and incorporating LLM as NLP, users can input to search for specific information about a given, and receive precise responses in the form of PDF content extracts. The application also supports user’s authentication, file handling and chatting, the entire package that offers solution for efficient handling of PDF data extraction.

In this project, I have paid more attention to the user interface in order to provide a good user experience with a good back-end solution. Google login and the multiple file uploads added to the features make the system more accessible and the growing number of users would not pose any problem for the system.

6.2 Recommendations

- **Error Handling and Data Validation:** Another improvement could be made to the error handling procedures, were most necessary during the file uploading as well as during the interaction with the PDF extraction system. The extension limitation will increase the system reliability after having proper validation so that only the files of the supported types are uploaded.
- **User Interface Enhancements:** As it is available now, it still offers workable interface experience to users, but still can perform improvements related to usability and attraction. Even more, adding the more detailed UI pieces like the progress bar while uploading a file or the feedback messages that appear when something goes wrong, would add better experience worth.
- **Security Measures:** For strengthening the security concerns to the user data fundamentals of security best practices for custom applications should also apply additional layer of encryption for data which is stored in the database especially the login credentials of the users.

6.3 Future Work

In the future I will add another feature that will show the reference of the PDF content from which the response is derived. This will help the system users to know the source of the extracted information to improve the reliability of the system. Furthermore, I still want to investigate further possibilities of introducing greater complexity of certain types of inputs and introducing certain level of more accurate answers. To enhance the existing approaches, broader context support is also another area of development, as well as the extension of the system's ability to read and handle different file formats and the integration of machine learning models to help the system to gain a better understanding of the context of the files. In future iterations, the methodology outlined in **Figure 3.3.1** could be extended to include OCR capabilities and support for more complex queries [\[Polak & Morgan, 2024\]](#), [\[Tiedemann, 2014\]](#).

REFERENCES

- [1] M. P. Polak and D. Morgan, "Extracting accurate materials data from research papers with conversational language models and prompt engineering," *Nature Communications*, 2024. [Online]. Available: <https://www.nature.com/articles/s41467-024-45914-8>
- [2] J. Tiedemann, "Improved text extraction from PDF documents for large-scale natural language processing," in *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, 2014. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-54906-9_9.
- [3] N. Goel et al., "LLMs Accelerate Annotation for Medical Information Extraction," *Proc. of MLR*, 2023. [Online]. Available: <https://proceedings.mlr.press/v225/goel23a>
- [4] M. P. Polak and D. Morgan, "Extracting accurate materials data from research papers with conversational language models and prompt engineering," *Nature Communications*, 2024. [Online]. Available: <https://www.nature.com/articles/s41467-024-45914-8>
- [5] J. Tiedemann, "Improved text extraction from PDF documents for large-scale natural language processing," in *International Conference on Intelligent Text Processing and Computational Linguistics*, Springer, 2014. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-54906-9_9.
- [6] N. Goel et al., "LLMs Accelerate Annotation for Medical Information Extraction," *Proc. of MLR*, 2023. [Online]. Available:
- [7] <https://proceedings.mlr.press/v225/goel23a>

PDF Extraction

ORIGINALITY REPORT

2%

SIMILARITY INDEX

1%

INTERNET SOURCES

1%

PUBLICATIONS

1%

STUDENT PAPERS

PRIMARY SOURCES

1

123dok.com

Internet Source

1%

2

V. Sharmila, S. Kannadhasan, A. Rajiv Kannan,
P. Sivakumar, V. Vennila. "Challenges in
Information, Communication and Computing
Technology", CRC Press, 2024

Publication

<1%

3

Submitted to Universiti Tunku Abdul Rahman

Student Paper

<1%

4

constitution.org

Internet Source

<1%

5

Submitted to University of Greenwich

Student Paper

<1%

Exclude quotes

Off

Exclude matches

Off

Exclude bibliography

Off

0% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-generated text that was also likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

1. **AI-generated only** 0%
Likely AI-generated text from a large-language model.
2. **AI-generated text that was AI-paraphrased** 0%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer

Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (it may occasionally writing that is likely AI-generated as AI-generated and AI-paraphrased or likely AI-generated and AI-paraphrased writing as only AI-generated) so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?

The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?

Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.

