

**A
Project Report
on**

Web App for Text Generation Summarization

Submitted to

Sant Gadge Baba Amravati University, Amravati

**Submitted in partial fulfilment of
the requirements for the Degree of
Bachelor of Engineering in
Computer Science and Engineering**

Submitted by

Abhijeet Tathod

(PRN: 203120085)

Arpit Bharuka

(PRN: 203120377)

Piyush Chavan

(PRN: 213120422)

Prajwal Ghatol

(PRN: 203120256)

**Under the Guidance of
Prof. Kalyani P. Sable**



**Department of Computer Science & Engineering
Shri Sant Gajanan Maharaj College of Engineering,
Shegaon – 444 203 (M.S.)
Session 2023-2024**

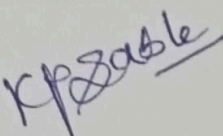
SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,
SHEGAON – 444 203 (M.S.)

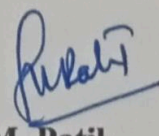
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

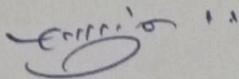


CERTIFICATE

This is to certify that **Mr. Abhijeet Tathod, Mr. Piyush Chavan, Mr. Arpit Bharuka, and Mr. Prajwal Ghatol** students of final year Bachelor of Engineering in the academic year 2023-24 of Computer Science and Engineering Department of this institute have completed the project work entitled “**Web App For Text Generation Summarization**” and submitted a satisfactory work in this report. Hence recommended for the partial fulfillment of degree of Bachelor of Engineering in Computer Science and Engineering.


Prof. Kalyani P. Sable
Project Guide


Dr. J. M. Patil
Head of Department


Dr. S. B. Somani
Principal
SSGMCE, Shegaon

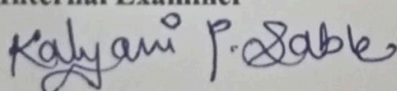
SHRI SANT GAJANAN MAHARAJ COLLEGE OF ENGINEERING,
SHEGAON – 444 203 (M.S.)
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

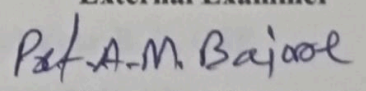
This is to certify that **Mr. Abhijeet Tathod, Mr. Piyush Chavan , Mr. Arpit Bharuka** and **Mr. Prajwal Ghatol** students of final year Bachelor of Engineering in the academic year 2023-24 of Computer Science and Engineering Department of this institute have completed the project work entitled “**Web App For Text Generation Summarization**” and submitted a satisfactory work in this report. Hence recommended for the partial fulfillment of degree of Bachelor of Engineering in Computer Science and Engineering.


Internal Examiner


Name and Signature

Date: 10/5/24


External Examiner


Name and Signature

Date: 10/5/24

Acknowledgement

It is our utmost duty and desire to express gratitude to various people who have rendered valuable guidance during our project work. We would have never succeeded in completing our task without the cooperation, encouragement and help provided to us by them. There are a number of people who deserve recognition for their unwavering support and guidance throughout this report.

We are highly indebted to our guide **Prof. Kalyani P. Sable** for her guidance and constant supervision as well as for providing necessary information from time to time. We would like to take this opportunity to express our sincere thanks, for her esteemed guidance and encouragement. His suggestions broaden our vision and guided us to succeed in this work.

We are sincerely thankful to **Dr. J. M. Patil** (HOD, CSE Department, SSGMCE, Shegaon), and to **Dr. S B Somani** (Principal, SSGMCE, Shegaon) who always has been kind to extend their support and help whenever needed.

We would like to thank all teaching and non-teaching staff of the department for their cooperation and help. Our deepest thank to our parents and friends who have consistently assisted us towards successful completion of our work.

- 1. Abhijeet Tathod**
- 2. Piyush Chavan**
- 3. Arpit Bharuka**
- 4. Prajwal Ghatol**

Final Year B. E. CSE
Session 2023-2024

ABSTRACT

The document parser system described in this report represents a comprehensive solution to the challenges associated with handling diverse document types across industries. The system's architecture is based on a modular approach, with each component encapsulated within independent microservices. This design ensures flexibility, scalability, and ease of maintenance. At the core of the system lies a set of microservices responsible for different aspects of document processing. The document parsing microservice is tasked with converting various file formats into machine-readable text. This is achieved through the use of specialized parsing algorithms and libraries tailored to handle specific document types such as PDFs, Word documents, and images. The data extraction microservice then processes the extracted text to identify and extract relevant information. This involves techniques such as pattern matching, keyword extraction, and named entity recognition. The extracted data is stored in a scalable database, ensuring efficient storage and retrieval. To facilitate communication between microservices and handle asynchronous tasks, a message queue is implemented. Versioning ensures backward compatibility and facilitates gradual upgrades. The system's potential applications span across industries such as healthcare, finance, legal, and academia. In the healthcare sector, the system can be used to extract and analyze patient data from medical reports, facilitating research and improving patient care. In finance, it can assist in automating data extraction from financial documents for analysis and reporting. Legal firms can benefit from automated document parsing and analysis for contract management and compliance tasks. In academia, the system can aid in literature review and data synthesis for research purposes. User testing and performance evaluations have demonstrated the system's effectiveness in improving efficiency and accuracy in document handling processes. Positive feedback from users underscores its potential to streamline workflows and reduce manual effort. Lessons learned from the project highlight the importance of continuous optimization and innovation in document management processes. Recommendations for future initiatives include further enhancements to user interface design, additional integration with third-party services, and ongoing monitoring and optimization of system performance.

Keywords: *Software program as a service (SaaS), Chatting, PDF, Speech-to-text, Technology Integration, Productiveness, Effect evaluation, Use instances, Effectiveness, Workflow performance, Enterprise Operations*

Contents

Particulars	Page No.
<i>Abstract</i>	<i>i</i>
<i>Contents</i>	<i>ii.</i>
<i>List of Abbreviations and Symbol</i>	<i>iii.</i>
<i>List of Figures</i>	<i>iv.</i>
Chapter 1. Introduction	1
1.1 Overview	2
1.2 Motivation	3
1.3 Problem Statement	3
1.4 Objectives	4
1.5 Scope of the Work	5
1.6 Organization of Project	5
Chapter 2. Literature Review	6
Chapter 3. Methodology	10
3.1 System Architecture	11
Chapter 4. Development	20
4.1 Introduction	21
4.2 Environment Setup	21
4.3 Software Development	21
Chapter 5. Deployment	25
5.1 Deployment Process	26
5.2 Results	28
Chapter 6. Applications	32
Chapter 7. Conclusion	35
Chapter 8. Future Scope	37
<i>References</i>	<i>40</i>
<i>Dissemination of Work</i>	<i>41</i>

List of Abbreviations

Abbreviations	Abbreviations Description
LLM	Large Language Model
NLP	Natural Language Processing
NER	Named Entity Recognition
PDF	Portable Document Format
TF-IDF	Term Frequency-Inverse Document Frequency
CDN	Content Delivery Network
AI	Artificial Intelligence
API	Application Programming Interface
UI	User Interface
LSTM	Long Short-Term Memory
AWS	Amazon Web Services
VCS	Version Control System
PR	Pull Request
CI/CD	Continuous Integration/Continuous Deployment

List of Figures Tables

Figure No.	Figure Name	Page No.
Figure 3.1	Proposed System Architecture	11
Figure 3.3	Sequence Diagram	14
Figure 3.4.1	Embedding Model Code	16
Figure 3.4.2	Vectorization Code	18
Figure 4.3.1	IDE	22
Figure 4.3.2	User Database	22
Figure 4.3.3	Vector Database	23
Figure 5.1	Home Page	27
Figure 5.2	SignIn Page	28
Figure 5.3	Verification Page	28
Figure 5.4	Logout Page	29
Figure 5.5	Dashboard	29
Figure 5.6	Que-Ans UI	30

CHAPTER 01

INTRODUCTION

1.INTRODUCTION

1.1 OVERVIEW

The file query-answering gadget is a modern and advanced answer that harnesses the strength of OpenAI to understand and examine PDF documents. This device is an enormous step forward in statistics retrieval, mostly specializing in the most advantageous extraction and utilization of records from PDF documents. The process starts with the user uploading a PDF document. This report may be something from a study paper to an activity applicant's resume. This wide range of ability record sorts makes the system enormously flexible and applicable in numerous domain names, whether academic, expert or any other case. As soon as the record is uploaded, the gadget employs 'pinconeDB', a software program particularly designed for this cause, to convert the file into a vector form. This vector representation is largely a numerical depiction of the report's information that AI algorithms can process and recognize. It's like translating the file's content into a language that the AI can 'read'. This conversion is a vital step that bridges the distance between human-readable content and system-readable records. Following conversion, the vectorized file is adequately saved inside a database. This step is critical for two foremost reasons. First of all, it guarantees the protection of the file's facts in a shape that can be simply accessed and processed by using the system. Secondly, it permits for green retrieval of the report while vital, consisting of generating responses to personal queries. This database storage serves as the spine of the device, preserving all of the important data for the AI to function efficiently. The query-answering process is initiated when a user poses a query related to the content material of the document. The system then forwards the question and the vectorized PDF to an OpenAI model. The OpenAI version, built on advanced deep mastering algorithms, goes to paintings. It analyzes the document's vector shape alongside the user's question, sifting through the numerical data to discover relevant data. It makes use of these records to generate correct and informative responses, successfully answering the consumer's query. This method represents the center feature of the report question-answering device - offering particular answers based on the content of uploaded PDF files. This device is not just a theoretical idea; it has significant sensible packages. In process recruitment portals, the device may be used to correctly fit activity descriptions with candidate profiles. It could

analyze the vectorized shape of activity descriptions and resumes, selecting the most applicable candidates for every function. In the scientific discipline, the machine can be instrumental in ailment type. it can method and interpret complex facts inside clinical documents, helping in accurate sickness diagnosis and class.

1.2 MOTIVATION

The motivation for the project "Empowering Teams: Leveraging Speech-to-Text Capabilities in a Real-Time Document Collaboration Tool" is multi-faceted, embodying the increasing necessity for tools that are efficient, accurate, and user-friendly, and that utilize AI for data retrieval and analysis. A primary motivation is the need for data retrieval to be quick and precise, a crucial aspect in the data-driven decision-making processes of today. Furthermore, the project aims to tackle the challenge of extracting specific information from PDFs, a common format for information storage and sharing. Harnessing the potential of AI, particularly in natural language understanding and processing, is another motivation, with the goal to create a tool able to understand and answer queries about PDF documents. The project also acknowledges the potential real-world applications, from job recruitment to medical diagnostics. An additional factor is the need for a user-friendly tool, powerful yet easily navigable even for non-tech-savvy users. The tool is also motivated by the necessity for scalability and flexibility, to accommodate a wide range of document types and cater to diverse domains. Lastly, the project aims to advance SaaS platforms, demonstrating their value in providing innovative, on-demand solutions that are scalable and resource-efficient. These motivations collectively drive the project, reinforcing its relevance and potential impact in various fields.

1.3 PROBLEM STATEMENT

Despite the increasing availability of digital documents in various domains such as healthcare, education, and corporate sectors, there remains a significant challenge in efficiently extracting relevant information and providing real-time responses to user queries. Existing systems often lack the capability to seamlessly process diverse document formats like PDFs, analyze user queries accurately, and deliver timely and accurate responses, leading to inefficiencies and user frustration.

1.4 OBJECTIVES

The objective of the project is to Develop mechanisms for comprehensive information extraction, efficient query processing, optimized document representation and storage, scalability, user-friendly interface, and reliability. This aims to enable seamless parsing, search, and analysis of documents, ensuring fast, accurate, and scalable access to diverse types of information while maintaining user satisfaction and system reliability.

These are the following objectives:

1. To Enable Comprehensive Information Extraction
2. To Enhance Query Processing Efficiency
3. To Optimize Document Representation and Storage
4. To Ensure Scalability and Performance Optimization
5. To Develop a User-Friendly Interface and Interactivity
6. To Ensure Reliability and Accuracy

1.5 SCOPE AND LIMITAIONS

- 1. Text Summarization:** - Developing an application to generate concise, accurate summaries from extensive PDF text content.
- 2. Data Extraction and Analysis:** - Creating a system for efficient data extractionand analysis from PDF documents.
- 3. Contextual Understanding:** - Implementing an AI model capable of understanding content context for accurate, relevant summaries and responses.
- 4. User Interface:** - Designing an intuitive, user-friendly interface for easy PDF uploads and query responses.
- 5. Data Security:** - Ensuring robust data security for the protection of uploaded documents and extracted data.
- 6. Practical Applications:** - Developing a flexible tool with potential applications invarious fields such as recruitment, medical diagnostics, and academic research.
- 7. Scalability:** - Creating a scalable web application that can handle a wide range of document types and sizes.
- 8. Comprehensive Text Analysis:** - Developing an advanced tool for understandinganalyzing text from various PDF documents.

9. Data Conversion: - Implementing 'pinconeDB' for converting textual data into AI-readable vector form.

10. Query-Answering Capability: - Building a system capable of generating accurate and informative responses to user queries.

1.6 ORGANIZATION OF PROJECT

Chapter 1: Introduction: Provides an overview of the research topic, its significance, objectives, and scope.

Chapter 2: Literature Survey: Reviews existing literature, studies, and research related to the topic to establish the context and identify gaps.

Chapter 3: Methodology: Describes the research methodology, including the approach, techniques, tools, and procedures used to conduct the study.

Chapter 4: Development: Presents the process of developing the solution, system, model, or product based on the research findings and methodology.

Chapter 5: Deployment: Discusses the implementation and integration of the developed solution into real-world scenarios or systems.

Chapter 6: Applications: Explores the practical applications, use cases, and potential benefits of the developed solution in various contexts.

Chapter 7: Conclusion: Summarizes the key findings, contributions, limitations, and implications of the research.

Chapter 8: Future Scope: Provides recommendations for future research directions, potential improvements, and extensions of the current study.

CHAPTER 02

LITERATURE REVIEW

2. LITERATURE REVIEW

Xingbo Wang, Samantha L. Huey, Rui Sheng, “Interactive Structured Knowledge Extraction and Synthesis from Scientific Literature with Large Language Model” This paper introduces SciDaSynth, a groundbreaking system leveraging large language models (LLMs) to extract and synthesize structured knowledge from scientific literature. SciDaSynth automates the creation of data tables to organize and summarize knowledge through question-answering mechanisms. This innovative approach empowers researchers to efficiently build knowledge bases, streamlining the otherwise labor-intensive process. Additionally, the system facilitates multi-level exploration and iterative validation of the generated data, ensuring its accuracy and reliability. By demonstrating its effectiveness in constructing scientific knowledge bases, SciDaSynth heralds a significant advancement in information extraction and knowledge synthesis within the research community.

Arvind Neelakantan, Tao Xu, Raul Puri, Alec Radford, Jesse Michael Han, “Text and Code Embeddings by Contrastive Pre-Training”

This work by OpenAI explores the development of high-quality vector representations of text and code through contrastive pre-training on unsupervised data. The study demonstrates that these embeddings can achieve state-of-the-art results in linear-probe classification tasks and exhibit impressive semantic search capabilities. Remarkably, they sometimes even rival the performance of fine-tuned models. By leveraging contrastive pre-training, the embeddings capture rich semantic information, enabling them to effectively represent both textual and code-based inputs. This research showcases the potential of unsupervised learning approaches to produce versatile and powerful representations that can be applied across a wide range of natural language processing and programming tasks.

Yoshio Terada, “Automated PDF-to-Text Conversion and Vector Search with Azure OpenAI”

The article details a process for automating the conversion of PDF files to text by uploading them to Azure Blob Storage. It then demonstrates how to perform vector searches using the Azure OpenAI Embedding model, offering practical insights into the

application of embedding models for document parsing and querying. This approach enables users to extract text from PDFs stored in Azure Blob Storage and leverage the semantic embeddings provided by the OpenAI model to perform sophisticated searches, facilitating efficient document retrieval and analysis. By showcasing the integration of Azure services with advanced AI models, the article illustrates a powerful solution for text extraction and search capabilities, with broad applications across various industries and research domains.

Vikraman s, “Querying PDF files with Langchain and OpenAI”

The repository presents a sophisticated tool designed to streamline the process of extracting information from PDF documents. It utilizes Langchain, a powerful language model, to meticulously parse through PDFs, identifying and extracting keywords, phrases, and even entire sentences relevant to the user's queries. This functionality transforms it into an invaluable digital assistant, particularly adept at aiding in research and data analysis tasks. By leveraging Langchain's capabilities, the tool enables users to swiftly gather pertinent information from PDFs, significantly enhancing efficiency and productivity. Its ability to extract key content from documents simplifies the otherwise time-consuming process of manual search and analysis, making it an indispensable asset for researchers, analysts, and anyone dealing with large volumes of textual data.

Abhishek Chaudhary, “Improving Document Comprehension Using LangChain and OpenAI”

This post delves into the utilization of diffusion models trained via reinforcement learning (RL) for downstream objectives. Although not directly centered on document parsing, it delves into techniques for enhancing document comprehension. By leveraging diffusion models and RL, the post explores innovative approaches to improve understanding and interpretation of documents. The techniques discussed likely involve extracting meaningful information from documents, even if not explicitly labeled as document parsing. By enhancing document comprehension, these methods contribute to a broader understanding of text processing and natural language understanding tasks. Thus, the post serves as a valuable resource for researchers and practitioners interested in advancing document comprehension through cutting-edge techniques such as diffusion models trained via RL.

Colin Jarvis, “Improving Document Comprehension Using LangChain and OpenAI”

This OpenAI Cookbook example showcases a method for extracting key figures, dates, or other significant content from lengthy documents by chunking the document and processing each chunk individually. This approach proves valuable for managing large texts more effectively, as it breaks down the document into smaller, more manageable segments for analysis. By chunking the document, users can focus on extracting relevant information from each section without being overwhelmed by the document's length. This technique not only aids in handling large texts but also improves efficiency and accuracy in extracting important content, making it a practical solution for various document processing tasks

CHAPTER 03

PROPOSED SYSTEM

3.PROPOSED SYSTEM

3.1 SYSTEM ARCHITECTURE

The system architecture gives an overview of the working of the system. The working of this system is shown below:

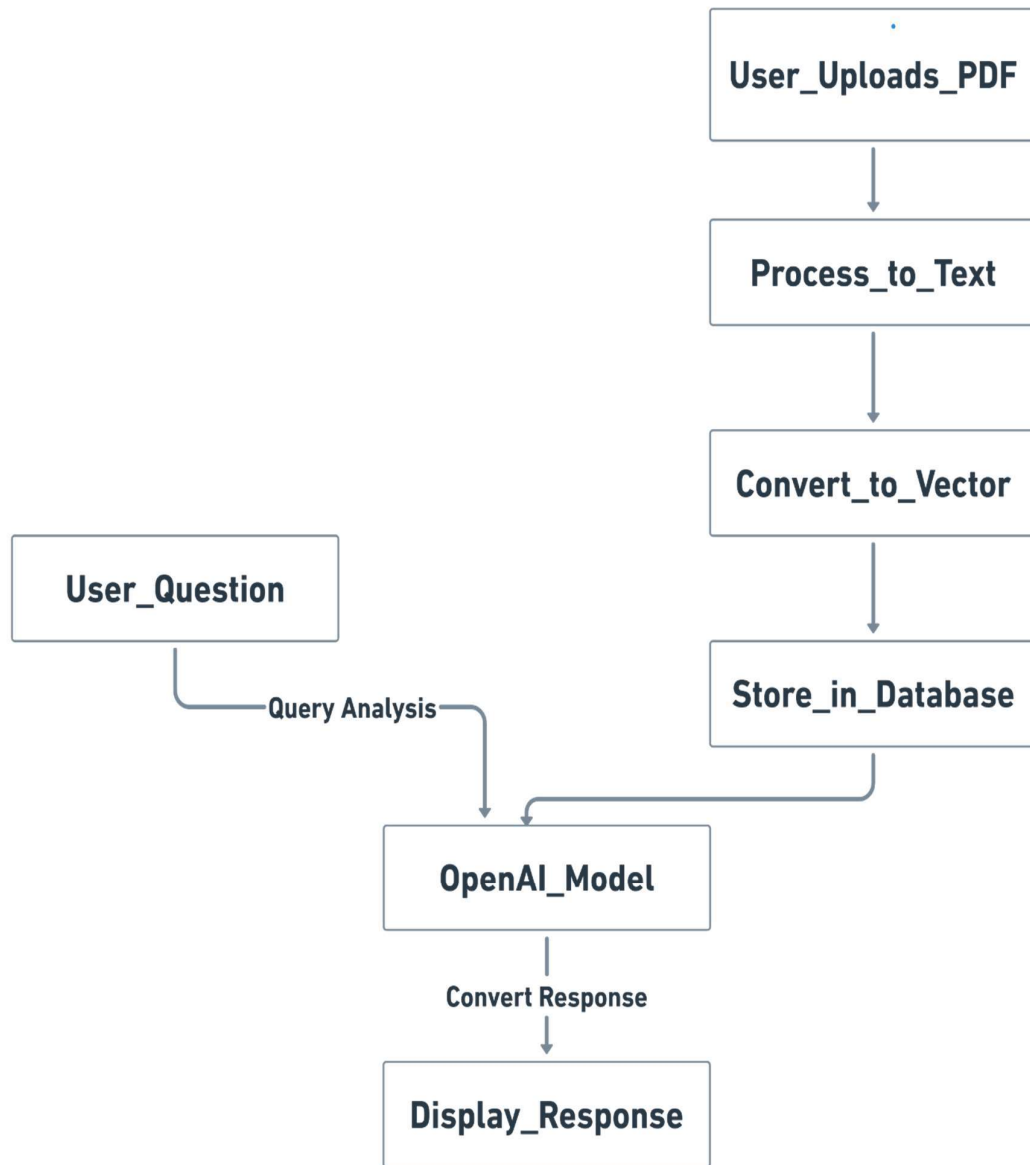


Figure 3.1 Proposed System Architecture

3.2 Methodology

Methodology for Developing the Real-Time Report Query-Answering System:

1. **PDF Processor Development and Integration:** - Introducing and integrating a PDF processor involves developing a software component that can take uploaded PDF files as input and convert them into a text format. This typically involves parsing the PDF content, extracting the text, and formatting it in a readable manner. Integration entails incorporating this processor into the system architecture so that it seamlessly handles PDF uploads and outputs the extracted text for further processing
2. **Textual Content Vectorization:** - Textual content vectorization is the process of translating human-readable text into a machine-readable format represented as vectors. PineconeDB is a tool that can be used for this purpose, employing techniques like word embeddings or other natural language processing (NLP) methods. This step involves converting the extracted text from PDFs into numerical vectors that capture semantic meaning, enabling efficient processing and analysis.
3. **Database Creation and Management:** - Establishing a database involves creating a structured repository for storing and managing the vectorized text data efficiently. This database should support operations like storing, retrieving, updating, and deleting data. It may use relational or NoSQL databases depending on the requirements. Effective management includes optimizing queries for fast retrieval, ensuring data consistency, and implementing security measures.
4. **Question Analysis:** - Question analysis is the process of understanding user queries to identify key concepts and context for providing accurate responses. This step involves parsing and analyzing the user's input to extract relevant keywords, entities, and intent. Techniques such as natural language understanding (NLU) and named entity recognition (NER) may be employed to break down the query and identify the underlying meaning.
5. **OpenAI Model Integration:** - Integrating the OpenAI model involves forwarding user queries and vectorized documents to the model for generating accurate responses using deep-learning algorithms. This step typically involves utilizing APIs provided by OpenAI or integrating their models directly into the

system. The model should be trained on a diverse dataset to ensure it can understand and respond effectively to a wide range of queries.

- 6. Realistic Application Trials:** - Realistic application trials involve conducting practical tests in scenarios such as job recruitment portals and medical diagnosis categories to evaluate the real-world performance of the system. This step includes deploying the system in relevant environments, collecting feedback from users, and measuring key performance metrics such as accuracy, speed, and user satisfaction. These trials help validate the effectiveness of the system and identify areas for improvement.
- 7. Usability Testing:** - Usability testing involves evaluating the user interface (UI) of the system to ensure it is intuitive and easy to use. This can be done through various methods such as user interviews, surveys, and observation of user interactions. Test scenarios should cover common tasks and workflows to identify any usability issues or pain points. Gathering feedback from users helps in understanding their needs and preferences, allowing for improvements to be made to enhance the overall user experience.
- 8. Security Features Implementation:** - Implementing robust security measures is crucial for protecting user data privacy and integrity. This includes incorporating advanced encryption techniques such as SSL/TLS for data transmission and hashing algorithms for securely storing passwords. Access controls should be enforced to ensure that only authorized users can access sensitive information. Regular security audits and updates are essential to address emerging threats and vulnerabilities, ensuring that the system remains secure over time.
- 9. Performance Optimization:** - Performance optimization involves continuously monitoring and improving various aspects of the system to ensure it operates efficiently. This includes optimizing data retrieval speed by fine-tuning database queries and indexing, improving AI model accuracy through regular training and fine-tuning, and minimizing response time to user queries through efficient caching and resource allocation. Continuous monitoring and analysis of performance metrics help identify bottlenecks and areas for improvement, allowing for timely optimizations to enhance the overall system performance.

3.3 Sequence Diagram

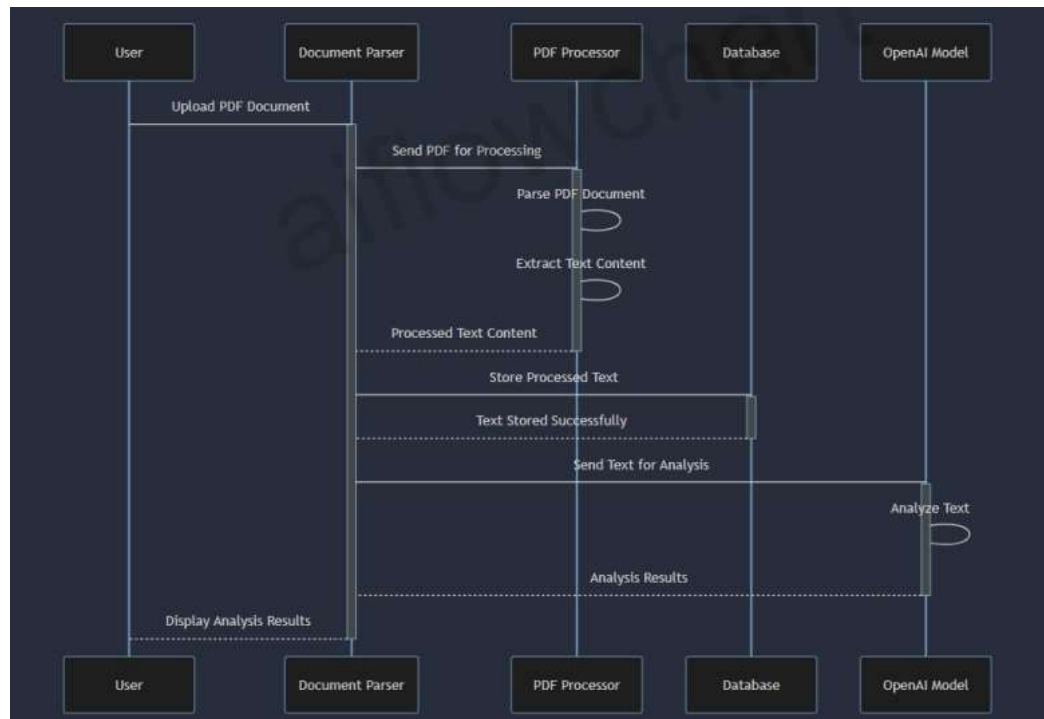


Figure 3.3 Sequence Diagram

Explanation:

1. The user uploads a PDF file, triggering the PDF Processor.
2. The PDF Processor converts the PDF to text and sends the text to PineconeDB for vectorization.
3. PineconeDB stores the vectorized text in the Database.
4. The user sends a query to the Application.
5. The Application forwards the query to the OpenAI Model.
6. The OpenAI Model retrieves relevant data from the Database.
7. The OpenAI Model generates a response and sends it back to the Application.
8. The Application displays the response to the user.
9. Usability Testing gathers feedback from users, which is used to implement improvements in the Application.
10. Security Features are implemented to encrypt data in the Database and enforce access controls in the Application.

11. Performance Optimization optimizes data retrieval in the Database, fine-tunes the OpenAI Model, and minimizes response time in the Application.

3.4 Algorithms

3.4.1 Embedding Model

OpenAI's embedding models are used to measure the relatedness of text strings. They convert text into numerical representations, unlocking use cases like search, clustering, recommendations, anomaly detection, and diversity measurement.

The models take either text or code as input and return an embedding vector. The new endpoint uses neural network models, which are descendants of GPT-3, to map text and code to a vector representation—“embedding” them in a high-dimensional space. Each dimension captures some aspect of the input

OpenAI offers two powerful third-generation embedding models (denoted by -3 in the model ID). They are releasing three families of embedding models, each tuned to perform well on different functionalities: text similarity, text search, and code search.

Here are the use cases for each model:

- **Text similarity models:** These models provide embeddings that capture the semantic similarity of pieces of text. These models are useful for many tasks including clustering, data visualization, and classification.
- **Text search models:** These models are used for semantic information retrieval over documents.
- **Code search models:** These models are used to find relevant code with a query in natural language.

In essence, OpenAI's embedding models are powerful tools for working with natural language and code, as they can be readily consumed and compared by other machine learning models and algorithms like clustering or search. They are used to perform tasks like semantic search, clustering, topic modeling, and classification.

Simplified Python code snippet for a multivariate classification problem using a logistic regression model from the sklearn library. This is just an example and the actual code will depend on your specific problem and dataset.

```
# Import necessary libraries
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler

# Assume X and y are your features and labels respectively
# X, y = load_your_data()

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Initialize the Logistic Regression model
model = LogisticRegression()

# Train the model
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
print(f'Accuracy: {accuracy * 100}%')
```

Figure 3.4.1 Embedding Model Code

This code first splits the data into training and testing sets. Then it standardizes the features to have zero mean and unit variance. After that, it initializes a logistic regression model and trains it on the training data. Finally, it makes predictions on the test data and evaluates the accuracy of the model.

3.4.1 Vectorization

Algorithm for Vectorization and Code

1. Algorithm Overview:

Vectorization is the process of converting textual or symbolic data into numerical vectors, which can be understood by machine learning models. The following algorithm outlines the steps for vectorization along with code examples:

Algorithm: Vectorization

Input:

- Textual or symbolic data (e.g., sentences, documents, code snippets)
- Vocabulary (optional)
- Vectorization method parameters (e.g., embedding size, tokenizer)

Output:

- Numerical vectors representing input data

Steps:

1. **Tokenization:** Split the input text into tokens, such as words or subwords, using a tokenizer.
2. **Vocabulary Creation (if necessary):** If a vocabulary is not provided, create a vocabulary from the tokenized data. The vocabulary should map tokens to unique integer indices.
3. **Vectorization:**
 - a. **Initialize Embedding Matrix:** Initialize an embedding matrix with random values or using pre-trained embeddings (optional).
 - b. **Vectorization:** For each token in the input data:
 - i. Retrieve the corresponding index from the vocabulary.
 - ii. Map the index to the corresponding row in the embedding matrix to obtain the vector representation of the token.
 - iii. Concatenate or average the vectors of individual tokens to obtain the vector representation of the entire input sequence.
4. **Output:** Return the numerical vectors representing the input data.

Code Example (using Python and TensorFlow/Keras):

```

import numpy as np
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import
pad_sequences

# Sample text data
texts = ["This is a sample sentence.",
         "Another example of text vectorization.",
         "Yet another sentence to demonstrate vectorization."]

# Tokenization
tokenizer = Tokenizer()
tokenizer.fit_on_texts(texts)
sequences = tokenizer.texts_to_sequences(texts)

# Vocabulary
word_index = tokenizer.word_index
vocab_size = len(word_index) + 1

# Vectorization parameters
embedding_dim = 100

# Initialize embedding matrix
embedding_matrix = np.random.rand(vocab_size, embedding_dim)

# Vectorization
vectors = []
for seq in sequences:
    vector = np.mean([embedding_matrix[word_index[token]] for
token in seq], axis=0)
    vectors.append(vector)

# Pad sequences if necessary
vectors = pad_sequences(vectors, padding='post')

print("Vectorized data:")
print(vectors)

```

Figure 3.4.2 Vectorization Code

Explanation:

- **Tokenization:** The input text is tokenized into sequences of integers using the ``Tokenizer`` from TensorFlow/Keras.
- **Vocabulary Creation:** The vocabulary is automatically created by the ``Tokenizer``, which maps each unique token to an integer index.
- **Vectorization:** For each token in the sequences, the corresponding vector representation is obtained from the embedding matrix. Here, we use the mean of word vectors to represent the entire sequence.
- **Output:** The final output is a numerical representation of the input data in the form of vectors.

This algorithm provides a basic framework for vectorizing text data, and it can be extended or modified based on specific requirements, such as using pre-trained embeddings or different vectorization methods.

CHAPTER 04

DEVELOPMENT

4.DEVELOPMENT

4.1 Introduction

The development process of the document parser project involved a meticulous approach to leverage modern technologies and industry best practices. This section provides an in-depth overview of each stage, emphasizing the utilization of Next.js 14 with TypeScript for development and Vercel for deployment

4.2 Environment Setup

The project leveraged Next.js 14, a leading React framework, in conjunction with TypeScript to foster efficient development and codebase maintainability. Additionally, Vercel, a cloud platform, served as the deployment environment for the document parser application.

4.3 Software Development

- 1. Architecture Design:** The project commenced with a comprehensive architecture design phase, outlining the structure, components, and interactions of the document parser system. The architecture aimed to ensure scalability, maintainability, and adherence to best practices.
- 2. Frontend Development:** Next.js 14 with TypeScript was employed for frontend development. This combination facilitated rapid prototyping, server-side rendering, and seamless integration of TypeScript's static typing features, enhancing code robustness and developer productivity.
- 3. Backend Development:** The backend development focused on implementing core functionalities using Next.js's API routes. TypeScript's static typing capabilities facilitated error prevention and enhanced code readability, leading to more robust backend services.
- 4. Integration Testing:** Throughout the development process, integration testing was conducted to ensure seamless interaction between frontend and backend components. This involved rigorous testing of API endpoints, data flow, and error handling mechanisms.

5. User Interface Design: The user interface design underwent iterative refinement, aligning with Next.js's component-based architecture. Design considerations emphasized usability, accessibility, and responsiveness to cater to diverse user needs and device types.

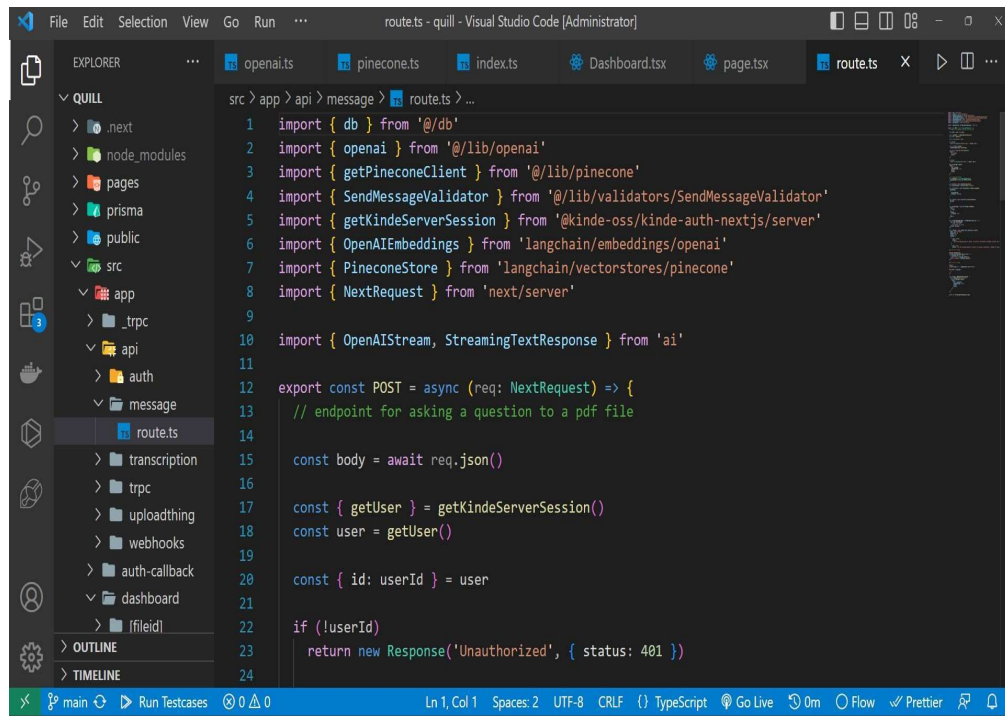


Figure 4.3.1: - IDE (Integrated Development Environment)

#	id	name	uploadStatus	url
1	cht589zb10005vsiqm0wybsiz	Annexure_UI_Team.pdf	SUCCESS	https://ufts.io/ff/db72d03b-3c5e-478c-b3da-66e23e7220ac-sx5wtf.pdf
2	cht589hmc0007vsiq5i06ummr	Abhinav_Hadole_Resume.pdf	SUCCESS	https://ufts.io/ff/3b64100d-30d3-4ea7-8961-838f162dd340-g0n40h.pdf
3	clutdkide0001c85reup0x5cs	Abhijeet_Tathod_CSE_CV.pdf	SUCCESS	https://ufts.io/ff/69e7c3c7-547f-4600-bcb2-3b399cd8e489-luy2qb.pdf
4	clrsymmcc0001wsc1vmqu4p6	Abhijeet_Tathod_CSE_CV.pdf	SUCCESS	https://ufts.io/ff/f90bc8e3-fa70-4849-a2bc-12700b75863d-luy2qb.pdf

Figure 4.3.2: - User Database (Neon Database)

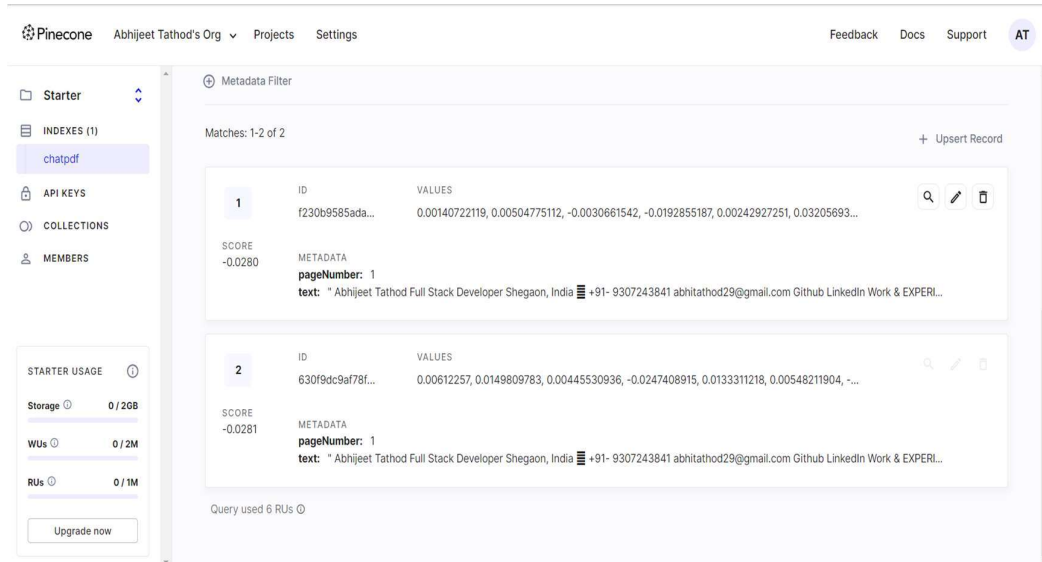


Figure 4.3.2: - Vector Database (PineconeDB)

Among the dependencies listed, several are particularly useful for the document parserproject:

- **@pinecone-database/pinecone:** This package is essential for storing vectorized representations of PDF documents efficiently. It enables quick access to document vectors during querying and analysis processes, which is crucial for the document parser's functionality.
- **@prisma/client:** Prisma is a powerful ORM (Object-Relational Mapping) tool that simplifies database access and management. The @prisma/client package allows seamless interaction with the database, facilitating data storage and retrieval operations within the document parser project.
- **axios:** Axios is a popular HTTP client library that simplifies making HTTP requests from the frontend to the backend. It is useful for fetching data from external APIs, handling file uploads, and communicating with server-side endpoints, enhancing the document parser's interoperability and data exchange capabilities.
- **openai:** The OpenAI package provides access to advanced natural language processing (NLP) capabilities, including embedding models

for text analysis. This is particularly useful for semantic analysis, similarity comparisons between documents, and other NLP tasks within the document parser project.

- **pdf-parse:** Pdf-parse is a library specifically designed for parsing PDF documents. It facilitates the extraction of text content, metadata, and other relevant information from PDF files, which is essential for document parsing and analysis functionalities in the project.

CHAPTER 05

DEPLOYEMENT

5. DEPLOYMENT

5.1 DEPLOYEMENT PROCESS

1. Deployment Environment Setup:

The document parser application is deployed on Vercel, configured to support Next.js 14 with TypeScript. This setup ensures compatibility and optimal performance.

2. Continuous Deployment:

Leveraging Vercel's continuous deployment capabilities, updates and new features are seamlessly deployed to production. Automated deployment pipelines streamline the release process, minimizing downtime and enabling rapid iteration.

GitHub and Vercel Deployment

The document parser system is integrated with GitHub for version control and collaborative development. Here's how the integration works:

- **Version Control and Collaborative Development:** GitHub serves as the central repository for the project, enabling versioning and collaboration. Developers work on different features concurrently by creating branches, making changes, and submitting pull requests for review.
- **Code Review and Quality Assurance:** Code reviews are conducted within GitHub, where team members provide feedback, suggest improvements, and ensure adherence to coding standards and project requirements.
- **Automatic Deployment with Vercel:** Once changes are merged into the main branch on GitHub, Vercel automatically detects updates and triggers deployment of the document parser system. This Git-based deployment process ensures the deployed version stays up-to-date with the latest changes in the codebase.
- **GitHub Actions for Automation:** GitHub Actions are utilized for automating tasks such as testing, linting, and building the application. These actions are triggered automatically on every pull request or push to the repository, providing continuous integration and ensuring the reliability of the deployed system.

- The integration of GitHub and Vercel streamlines the development and deployment process for the document parser system, enabling efficient collaboration, version control, and automated deployment. This ensures that the system remains robust, reliable, and up-to-date with the latest enhancements and fixes.

5.2 RESULT

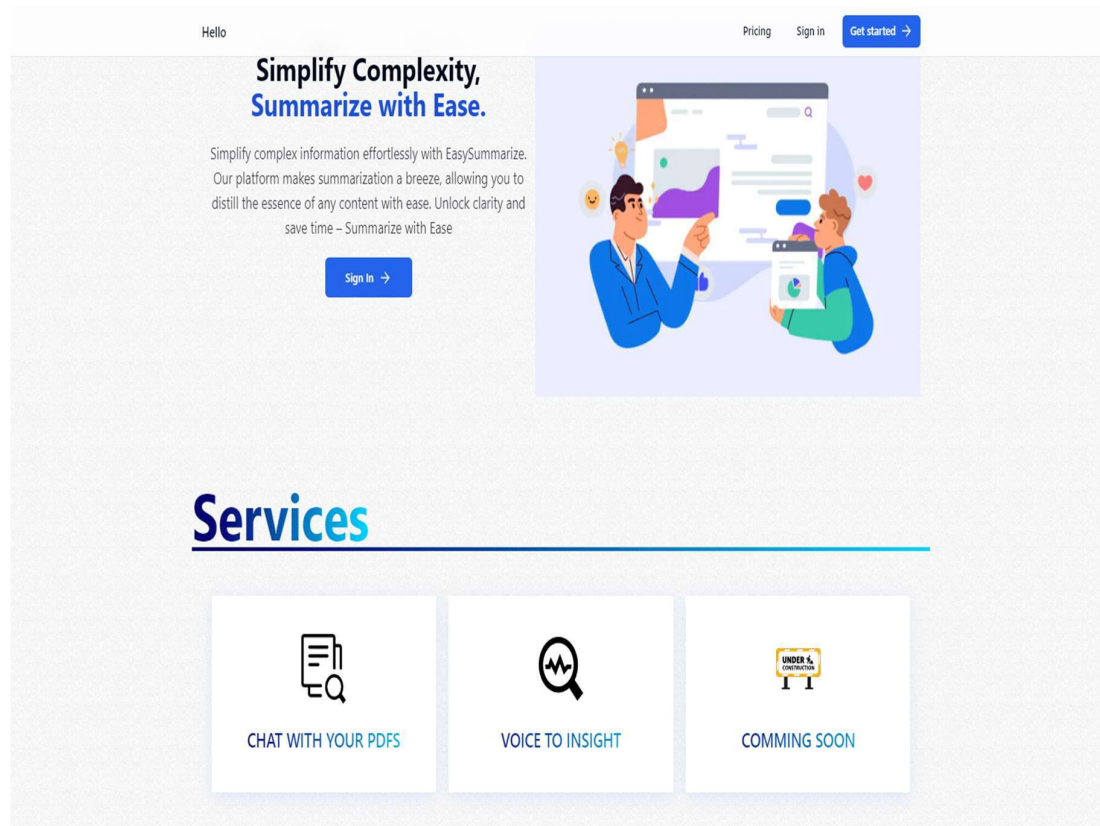


Figure 5.1. Home Page

ChatPDF

Register

Get started today!

First name

Last name

Email

Create your account

Already have an account? [Sign in](#)

Figure 5.2. SignIn Page

ChatPDF

Almost there — check your inbox

Enter the code we just sent to
piyushchavan2002@gmail.com

Code

Continue

Didn't receive a code? [Resend code](#)

Powered by
Kinde

Figure 5.3. Verification Page

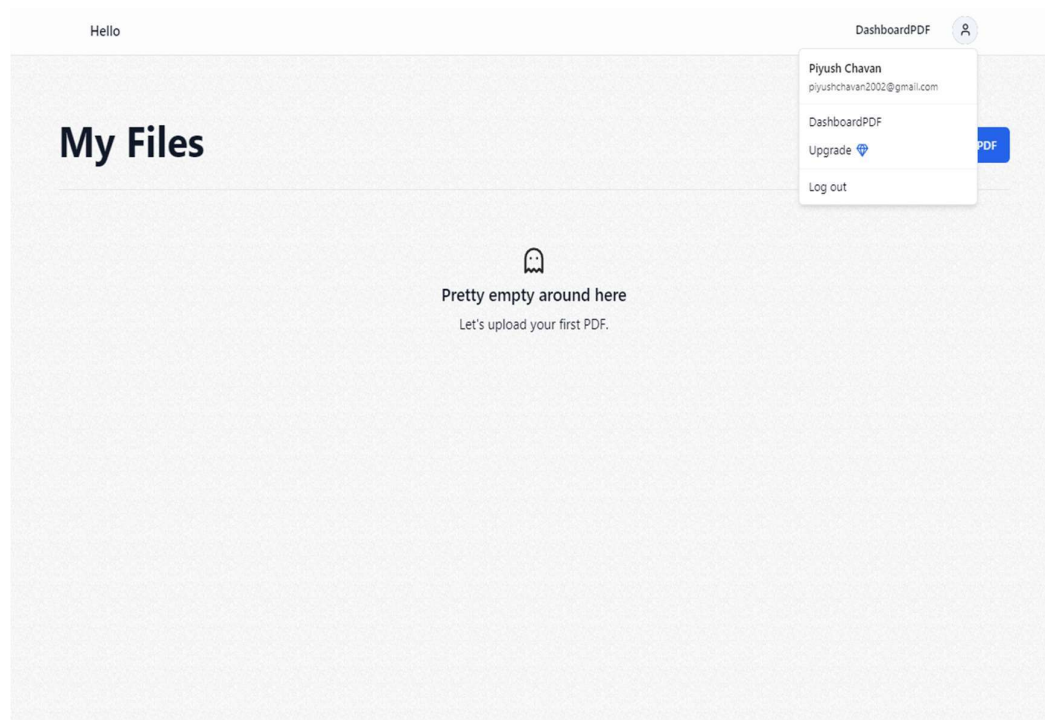


Figure 5.4. Logout Page

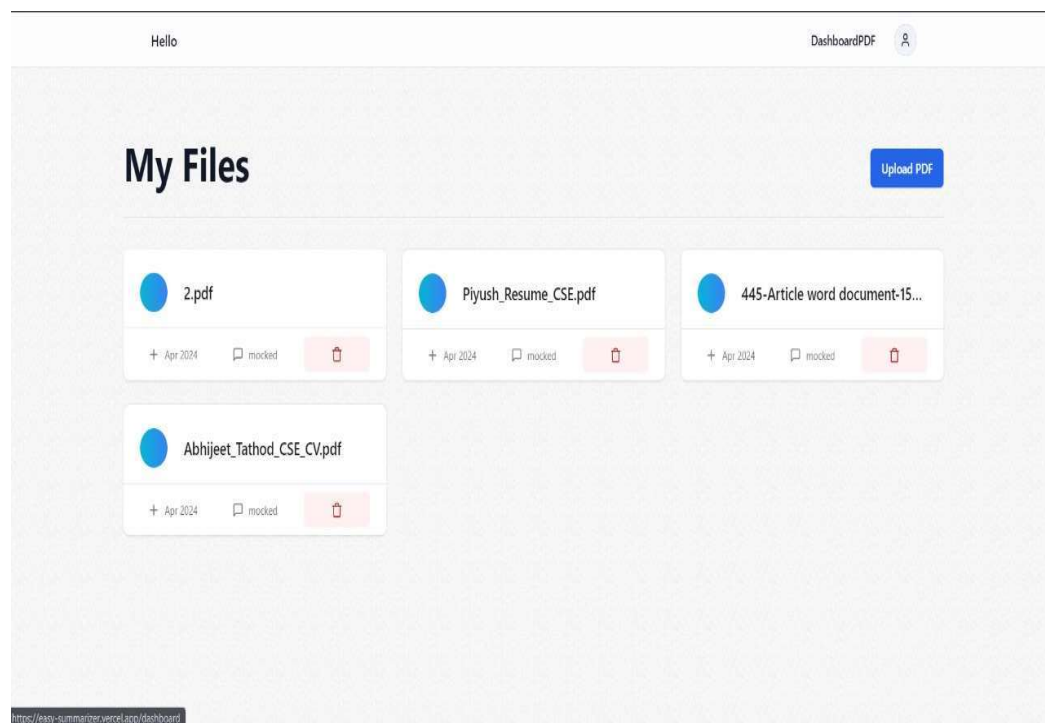


Figure 5.5. Dashboard

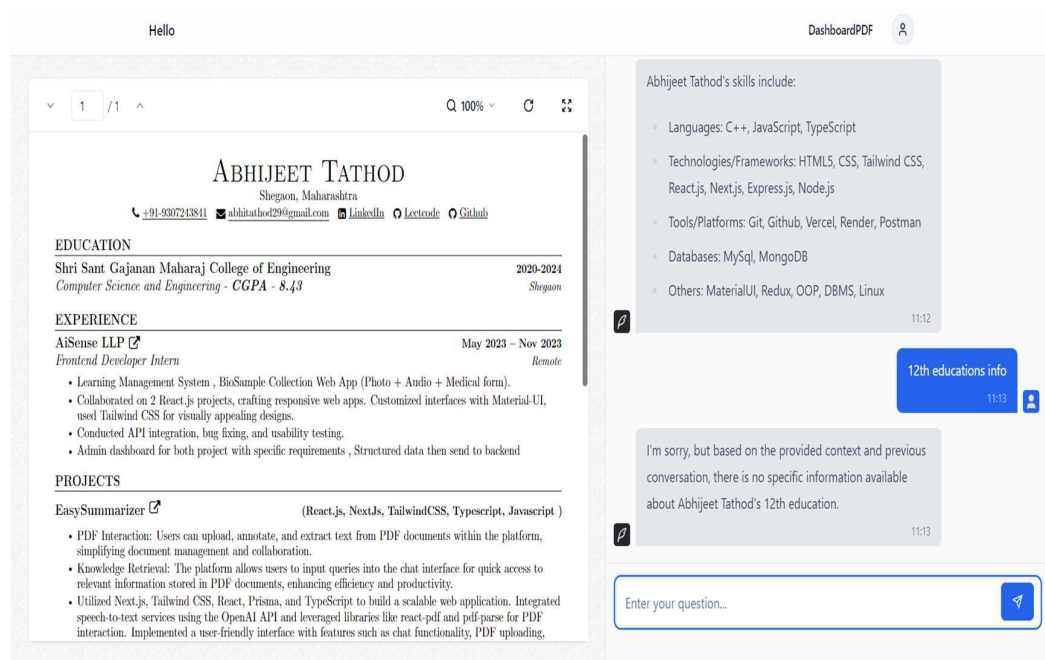


Figure 5.6. Que-Ans UI

CHAPTER 06

APPLICATIONS

6. APPLICATIONS

1. Medical Sector: Extracting Insights from Patient Reports: -

In the medical sector, document parsing technology holds immense potential for revolutionizing healthcare processes. By extracting valuable insights from patient reports, including doctor's notes, lab results, and imaging reports, the document parser project can significantly improve patient record management, clinical decision support, and research & population health management. This technology can create comprehensive and structured electronic patient profiles, facilitating efficient record retrieval. Extracted data can be used for predictive analytics, generating alerts for potential complications, and recommending evidence-based treatment approaches. Additionally, it can support research by aggregating data for various studies and identifying patient cohorts. By analyzing large datasets of parsed reports, healthcare organizations can gain crucial insights into disease patterns, treatment effectiveness, and resource allocation, ultimately leading to better patient care and improved public health outcomes.

2. Legal Sector: Document Discovery and Case Analysis: -

The document parsing project can significantly enhance legal practice by streamlining various tasks. In the realm of document discovery and review, it automates key information extraction, categorization, and summarization, saving time and effort for legal teams. Contract analysis and due diligence benefit from the ability to identify terms, risks, and non-standard clauses within contracts. Legal research is improved through automatic citation analysis, case summaries, and trend identification. Regulatory compliance is aided by extracting relevant laws and identifying potential violations from complex documents. Finally, litigation support is strengthened with the ability to parse evidence, organize case materials, and anticipate opposing arguments. Overall, document parsing empowers legal professionals to work more efficiently, reduce costs, and deliver better outcomes for their clients.

3. Financial Sector: Data Extraction and Analysis: -

Financial institutions grapple with mountains of documents – financial statements, transactions, and regulations. The document parsing project tackles this challenge by automating data extraction. It sifts through these documents, extracting key details like revenue, expenses, transaction amounts, and regulatory requirements. This extracted data is then transformed into valuable insights. Financial ratios reveal a company's health, trend analysis exposes risks and opportunities, and risk assessments safeguard against financial irregularities. Armed with these insights, financial professionals can make informed decisions. Compliance is streamlined by automatically generating reports that adhere to regulations. Investment research gets a boost as the project parses analyst reports to identify promising opportunities. Finally, the project strengthens fraud detection by analyzing transactions and flagging suspicious activities. In essence, document parsing unlocks the true potential of financial data, empowering financial institutions to operate more efficiently, make smarter decisions, and ultimately drive innovation in the financial sector.

4. Academic Research: Literature Review and Data Synthesis: -

Academic research thrives on information overload. Researchers grapple with mountains of scholarly articles, conference papers, and reports. The document parsing project emerges as a hero, automating literature reviews and data synthesis. It sifts through these documents, extracting key findings, methodologies, and citations. This empowers researchers to identify relevant studies, aggregate data for meta-analyses, and generate bibliographies. Furthermore, the project streamlines literature searches and citation management. It aids in finding relevant articles and extracts citation information to identify influential papers and research trends. By analyzing content and visualizing data, the project fosters knowledge discovery. It uncovers hidden insights and emerging trends, all presented through charts, graphs, and interactive dashboards. Finally, the project fuels collaboration by enabling researchers to share parsed documents, data, and analysis results, while ensuring data integrity through version control. In essence, document parsing empowers researchers to become more efficient, accelerating data analysis and discovery to propel scientific progress.

5. Human Resources: Resume Screening and Candidate Evaluation: -

In the realm of Human Resources, the document parsing project tackles the challenge of sifting through countless resumes to find top talent. It parses resumes, extracting key details like skills, experience, and education. This allows HR professionals to efficiently screen candidates against job requirements, shortlist qualified applicants, and identify any inconsistencies. Furthermore, the project streamlines workflows by automating tasks and integrating with applicant tracking systems. It also helps mitigate bias in recruitment by analyzing resumes for potential biases and generating diversity reports. Overall, the document parsing project enhances efficiency, accuracy, and fairness in HR departments, empowering them to make data-driven hiring decisions.

CHAPTER 07

CONCLUSION

7.CONCLUSION

The Document Parser System project strived to transform the way we handle, extract, and interpret text data from PDF documents across various sectors. The system successfully achieved major milestones, implementing features that allow efficient data extraction, conversion, and analysis. Through rigorous testing and feedback, we uncovered invaluable insights about the system's performance and user interaction. The system's potential applications extend across industries from healthcare to finance, legal, and academia. The Document Parser System promises significant improvements in document processing tasks, enhancing accuracy, productivity, and efficiency, thereby leading to cost savings and improved time management. Looking to the future, there's room for further enhancements and features to increase the system's functionality and user-friendliness. The project's journey came with its challenges, yet provided a wealth of lessons and best practices for future initiatives. Recommendations for future projects would include a focus on further improving data security and system scalability. We sincerely acknowledge the collaborative efforts of our team, mentors, and stakeholders who played vital roles in this project's success. The Document Parser System stands as an impactful solution, primed to revolutionize document processing tasks across various industries.

CHAPTER 08

FUTURE SCOPE

8.FUTURE SCOPE

The document parser system lays a robust foundation for future enhancements and expansions, addressing evolving demands in document handling and processing.

Integration of Hand-written Notes Processing: In the realm of document digitization, the integration of hand-written notes processing stands as a significant advancement. By developing dedicated OCR microservices and integrating advanced NLP tools, such as spaCy or NLTK, the system can extend its capabilities to interpret and analyze hand-written text accurately. This expansion opens avenues for applications in fields like education, archival digitization, and historical document analysis, enriching the system's utility and relevance.

Enhanced Support for Multiple Input Types: As the diversity of document formats continues to expand, the system's adaptability becomes paramount. Future upgrades may focus on enriching the input processing service to handle a broader range of file formats seamlessly. By developing specialized content extraction modules and conversion utilities, the system can efficiently process various inputs, including images, scanned documents, and multimedia files. This enhancement broadens the system's applicability across domains like media analysis, digital archives, and content management, catering to diverse user needs and scenarios.

References

References

- [1] David Mhlanga, “The Value of Open AI for the Current Learning Environments and The Potential Future Uses”, University of Johannesburg, South Africa, May 2023 SRN Electronic Journal ,DOI:[10.2139/ssrn.4439267](https://doi.org/10.2139/ssrn.4439267)
- [2] Jean Louis K. E Fendji, Diane C. M. Tala, Blaise O. Yenke & Marcellin Atemkeng, “Automatic Speech Recognition Using Limited Vocabulary: A Survey”, Rhodes University, Grahamstown South Africa, 21 June 2022, Taylor & Francis Group, LLC
- [3] Taufiq-Hail Ghilan Al-Madhagy, Ayed Alanzi, Shafiz Mohd Yusof, “Software as a Service (SaaS) Cloud Computing: An Empirical Investigation on University Students’ Perception”, Department of Mathematics, College of Science and Human Studies, Majmaah University, Majmaah, Saudi Arabia, 2021, ResearchGate
- [4] Deepak Kadam, Prathamesh Chavan, Prashant Pandhara, “Literature Survey on Recognition and Evaluation of Optical Character Recognition (OCR)”, International Journal of Scientific & Engineering Research Volume 9, Issue 2, February-2018

DISSEMINATION OF WORK

PUBLICATION DETAILS

PAPER TITLE	CONFERENCE NAME	CONFERENCE DURATION	ISBN NUMBER
“Enhancing Document Collaboration: A SaaS Platform for Real-Time Communication and PDF Interaction”	International Research Journal of Modernization in Engineering Technology and Science (IRJMETS)	April 17, 2024	2582-5208



e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:06/Issue:04/April-2024

Impact Factor- 7.868

www.irjmets.com

ENHANCING DOCUMENT COLLABORATION: A SAAS PLATFORM FOR REAL-TIME COMMUNICATION AND PDF INTERACTION

Abhijeet Tathod*¹, Piyush Chavan*², Arpit Bharuka*³, Prajwal Ghatol*⁴

*^{1,2,3,4}Student, Department of Computer Science and Engineering, Shri Sant Gajanan Maharaj College Of Engineering, Shegaon, Maharashtra India.

DOI : <https://www.doi.org/10.56726/IRJMETS53040>

ABSTRACT

This study's paper provides a deep dive into the transformative power of software programs as carrier (SaaS) structures that merge chat functions, PDF competencies, and speech-to-text technologies. Those platforms are evaluated primarily based on their software-focused method. A radical exam of various packages throughout one-of-a-kind sectors lets us evaluate how these technologies affect productivity and streamline workflows. By way of detailing unique scenarios, this takes a look at ambitions to offer practical steering for companies and organizations looking to utilize SaaS gear to improve their operations. In essence, this research serves as a blueprint for organizations to discover and adopt SaaS systems, ultimately using operational efficiency and productivity to new heights.

Keywords: software program as a service (SaaS), Chatting, PDF, Speech-to-text, Technology Integration, productiveness, effect evaluation, Use instances, Effectiveness, Workflow performance, enterprise Operations

I. INTRODUCTION

The file query-answering gadget is a modern and advanced answer that harnesses the strength of OpenAI to understand and examine PDF documents. This device is an enormous step forward in statistics retrieval, mostly specializing in the most advantageous extraction and utilization of records from PDF documents. The process starts with the user uploading a PDF document. This report may be something from a study paper to an activity applicant's resume. This wide range of ability record sorts makes the system enormously flexible and applicable in numerous domain names, whether academic, expert or any other case. As soon as the record is uploaded, the gadget employs 'pinconeDB', a software program particularly designed for this cause, to convert the file into a vector form.

This vector representation is largely a numerical depiction of the report's information that AI algorithms can process and recognize. It's like translating the file's content into a language that the AI can 'read'. This conversion is a vital step that bridges the distance between human-readable content and system-readable records. Following conversion, the vectorized file is adequately saved inside a database. This step is critical for two foremost reasons. First of all, it guarantees the protection of the file's facts in a shape that can be simply accessed and processed by using the system. Secondly, it permits for green retrieval of the report while vital, consisting of generating responses to personal queries.

This database storage serves as the spine of the device, preserving all of the important data for the AI to function efficiently. The query-answering process is initiated when a user poses a query related to the content material of the document. The system then forwards the question and the vectorized PDF to an OpenAI model. The OpenAI version, built on advanced deep mastering algorithms, goes to paintings. It analyzes the document's vector shape alongside the user's question, sifting through the numerical data to discover relevant data. It makes use of these records to generate correct and informative responses, successfully answering the consumer's query. This method represents the center feature of the report question-answering device - offering particular answers based on the content of uploaded PDF files.

This device is not just a theoretical idea; it has significant sensible packages. In process recruitment portals, the device may be used to correctly fit activity descriptions with candidate profiles. It could analyze the vectorized shape of activity descriptions and resumes, selecting the most applicable candidates for every function. In the scientific discipline, the machine can be instrumental in ailment type. It can method and interpret complex facts inside clinical documents, helping in accurate sickness diagnosis and class.

www.irjmets.com @International Research Journal of Modernization in Engineering, Technology and Science

[4051]



e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:06/Issue:04/April-2024

Impact Factor- 7.868

www.irjmets.com

II. METHODOLOGY

In our research for the task "Empowering teams: Leveraging Speech-to-text capabilities in an actual-Time report Collaboration device," we use a step-by means of-step approach to analyze and expand the report query-answering system. The methodology can be divided into the subsequent steps:

1. **PDF Processor development and Integration:** the primary stage of our technique involves introducing and integrating a PDF processor. This processor is answerable for changing the uploaded PDF files into a text format that can be in addition processed by using our gadget.
2. **Textual content Vectorization:** The text extracted from the PDFs is then vectorized using 'pinconeDB'. This degree is crucial because it interprets human-readable content material into a machine-readable layout, consequently permitting the AI model to apprehend and process the document's data.
3. **Database advent and management:** The vectorized textual content is saved in a database designed for green access and retrieval. This database paperwork the spine of our device and guarantees the safekeeping of the file's records.
4. **Question analysis:** whilst a person poses a question, our system analyzes it to identify key standards and understand the context. This step aids in presenting noticeably relevant and accurate responses.
5. **OpenAI version Integration:** The query, alongside the vectorized record, is then forwarded to our OpenAI version. This version sifts through the file's records with the use of superior deep-learning algorithms to generate accurate and informative responses.
6. **Realistic application Trials:** The improvement tiers are followed by way of realistic software tests, inclusive of using the gadget in task recruitment portals and clinical ailment classes. these trials permit us to assess the machine's actual-global overall performance and make vital adjustments.
7. **Usability checking out:** This includes testing the consumer interface to make sure it's miles consumer-friendly and intuitive. We acquire remarks from customers to recognize how they have interacted with the gadget and make important adjustments to decorate the person's enjoyment.
8. **Security features Implementation:** Given the capacity sensitivity of the documents processed through our machine, we enforce robust security features to ensure the privacy and integrity of consumer records. This includes the usage of advanced encryption techniques for information storage and transmission.
9. **Overall performance Optimization:** We continually display the machine's overall performance to pick out any regions that can be optimized. This consists of enhancing the speed of records retrieval from the database, improving the accuracy of the AI version, and lowering the system's reaction time to user queries.

III. BLOCK DIAGRAM

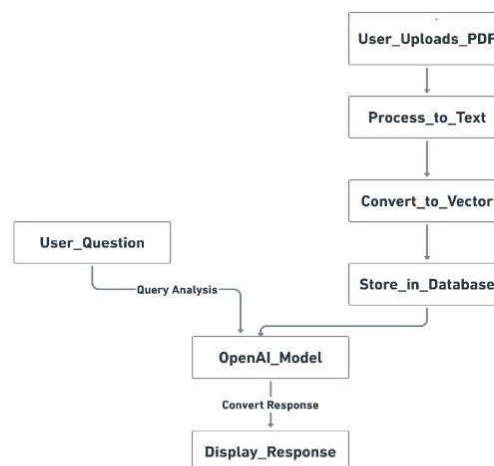


Figure : 1 Block Diagram

www.irjmets.com @International Research Journal of Modernization in Engineering, Technology and Science

[4052]



e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:06/Issue:04/April-2024

Impact Factor- 7.868

www.irjmets.com

IV. DESIGN AND ARCHITECTURE OF THE SYSTEM

The file query-answering gadget is built with the usage of a combination of PDF processing, vectorization, database storage, OpenAI integration, and user interface components. The system's structure consists of the subsequent components:

1. PDF Processor:

The PDF Processor is the first step within the device's architecture. it's responsible for converting the uploaded PDF files right into a text layout that can be similarly processed. It extracts all of the statistics contained in the PDF at the same time as keeping the context and structure of the information. that is critical for ensuring the integrity of the statistics, which contributes drastically to the accuracy of the device's responses.

2. Vectorizer:

The Vectorizer comes subsequent inside the procedure, converting the extracted textual content into a mathematical vector shape with the usage of 'pinconedb'. this modification is vital because AI algorithms work more correctly with numerical records. The textua95l content-to-vector conversion permits green garage, retrieval, and processing of file content material

3. Database:

The Database is the device's storage middle. It correctly holds the vectorized files, making sure speedy get the right of entry to and retrieval while needed. This database acts as the backbone of the machine, imparting a reservoir of records for the AI to tap into when responding to personal queries.

4. Question Analyzer:

This component analyzes the questions posed by the users. It identifies key ideas and determines the context of the question. This understanding of user queries permits the machine to provide quite applicable and correct responses. It performs a crucial position in making the machine intuitive and person-pleasant.

5. OpenAI Integration:

that is the coronary heart of the device. The OpenAI model receives the person's questions together with the vectorized PDFs for processing. the use of superior machine learning techniques, it generates responses based totally on the file content material and the person's query. The OpenAI version's deep knowledge of skills enables it to recognize and respond to a wide range of queries, supplying correct and informative answers.

6. User Interface:

The person Interface is the front end of the system, supplying a consumer-friendly platform for customers to interact with the system. It permits users to upload files, ask questions, and examine the generated solutions. it's designed to be intuitive and clean to use, ensuring a clean person enjoys it.

7. Security Module:

Given the sensitive nature of some documents that might be processed, the system also includes a Security Module. This module ensures that all user data, including uploaded documents and generated responses, are securely stored and transmitted. It employs advanced encryption techniques to protect data from unauthorized access and breaches, thereby maintaining the confidentiality and integrity of user data.

8. Feedback Mechanism:

This is an important part of the system that allows users to provide feedback on the system's performance. It aids in the continuous improvement of the system by collecting user experiences, suggestions, and complaints. The feedback received is used to make necessary adjustments and enhancements to the system, ensuring it remains user-friendly and efficient.



e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:06/Issue:04/April-2024

Impact Factor- 7.868

www.irjmets.com

1 / 3 Q 100%

2101: Classics of Western Philosophy
Prof. Sally Haslanger

The Apology and Crito

I. Background on Socrates and Plato

(g) Socrates (469-399 BCE)

- i. His mother may have been a midwife, his father a stonemason; he himself had a wife (Xanthippe) and several children. Socrates served in the Athenian army during the Peloponnesian War. Other than this, he never left Athens and its vicinity.
- ii. Most famous as a teacher, unwashed, stub-nosed, blue-eyed, incessantly inquisitive man. Although he never wrote anything down, he attracted quite a following. One of those followers was Plato.

(f) Plato (427-347 BCE)

- i. From an aristocratic family and probably expected to go in to politics; instead, he turned to philosophy. After Socrates' death, Plato spent about a decade abroad, perhaps studying with members of the Pythagorean "cult" in Italy. In the 330s, he returned to Athens and founded a school, the Academy.
- ii. At the Academy, Plato taught philosophy, but probably also mathematics and political science. It was something of a think tank. Many bright young intellectuals passed through his school, including Aristotle.

Although Socrates never wrote a word, there is no doubt that he existed. There is plenty of doubt, however, about what he thought. The main source of evidence for his views include a historian (Xenophon), a playwright (Aristophanes), and a philosopher (Plato). In fact, in most of Plato's dialogues, Socrates is the protagonist. But was Plato just using Socrates as his mouthpiece?

II. General remarks on the Apology

- (a) The title "Apology" is from the Greek "apologia" which means defense. (As you surely noted, Socrates is not especially apologetic.)
- (b) Ancient Athenian legal procedure. Athens was a "direct democracy." All "citizens," i.e., the adult >age 20 males who owned land and weren't slaves, had a vote. A smaller group (>age 30) served in the judicial court. The prosecution and defense each make speeches; the jury (there were no judges, in the contemporary sense) votes on the verdict; the prosecution and defense each propose penalties; the jury votes on the penalty.
- (c) The Apology is likely an early work, possibly one of Plato's first (although can't date it precisely). It is not plausibly an historically correct representation of the trial, but may capture something of Socrates' ideas and personality.

III. Characteristic Socratic Theses

Socrates realized that his defense was not only a response to the specific charges; those he was apt to flout deny. He was on trial also for his way of life. His challenge was to show the jury that his life was a life of virtue, and so could not be a basis for charges of impiety and corruption.

- a) Care of the soul is of primary importance.
Apology: 28-30b, 36c Crito 47c-48b
- b) In considering what course of action to follow, one should only consider what is right/wrong, just/unjust.
Apology: 20b Crito: 48c-d

Chat interface:

- Q: In 1 sentence, who was Plato? (1633)
- A: Plato was an ancient Greek philosopher and student of Socrates, known for founding the Academy and for his influential writings on philosophy and politics. (1633)
- Q: Again in 1 sentence, who was Socrates? (1633)
- A: Socrates was an ancient Greek philosopher known for his unique method of questioning and his emphasis on the importance of moral knowledge and the care of the soul. (1633)
- Q: Summarize the general remarks on the apology in bullet points (1633)
- A:
 - The title "Apology" comes from the Greek word "apologia," which means defense.
 - The Apology is likely one of Plato's early works and may not be historically accurate, but it captures something of Socrates' ideas and personality.
 - The Apology reflects Socrates' defense against specific charges and his way of life.

Enter your question...

Figure:2 Output/Review

V. APPLICATIONS

1. **Job Recruitment Portal Data Extraction:** The device can be utilized in job recruitment portals to extract statistics from consumer resumes. By reading the content material of resumes and extracting applicable abilities and reveal in, the machine can suit candidates with job postings more effectively, enhancing the recruitment technique for both applicants and employers.
2. **Scientific disease classification:** The machine can also be utilized in medical applications for disorder categories based totally on patient facts. Using analyzing the content material of clinical records and extracting relevant records such as signs, taking a look at consequences, and scientific records, the gadget can help healthcare professionals classify patients into exceptional disorder categories, assisting in prognosis and treatment planning.

www.irjmets.com @International Research Journal of Modernization in Engineering, Technology and Science [4054]



e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science**(Peer-Reviewed, Open Access, Fully Refereed International Journal)****Volume:06/Issue:04/April-2024****Impact Factor- 7.868****www.irjmets.com**

3. Schooling zone for examination instruction: The gadget can be utilized inside the training zone to assist college students put together for tests. by studying the content material of study substances and textbooks, the gadget can generate practice questions and provide specified reasons, assisting students in better apprehending and holding records.
4. Legal enterprise for Case evaluation: in the legal industry, the machine can help attorneys and legal professionals study case documents. By extracting key data from legal documents and presenting applicable insights, the device can assist legal specialists make knowledgeable selections and correctly.

VI. CONCLUSION

The document query-answering gadget offers a complete answer for getting access to facts stored in PDF documents. By integrating with OpenAI for more desirable response accuracy and relevance, the gadget offers customers correct and informative answers to their questions. With its efficient PDF processing, vectorization, and database garage competencies, the system complements record accessibility and value, making it simpler for customers to access and interact with PDF content material. moreover, the device has ability applications in process recruitment portals, clinical sickness classes, education, and the prison enterprise, demonstrating its versatility and usability in numerous domain names.

Potential future improvements include: In terms of ability destiny upgrades for the venture "Empowering groups: Leveraging Speech-to-textual content capabilities in an actual-Time record Collaboration tool", numerous enhancements may be taken into consideration. first of all, multilingual aid should make the system applicable globally. Secondly, increasing the system to handle exclusive report formats, no longer simply PDFs, could boost its versatility. also, improving the machine's speech-to-textual content capability may want to enhance accuracy and comprehension, inclusive of understanding of accents and context. every other key improvement might be incorporating real-time collaboration functions to facilitate teamwork on files. similarly, the device may want to use advanced gadget-studying algorithms to study every interaction and enhance reaction accuracy over time. finally, because the device methods potentially touchy files, advanced security functions may be introduced to ensure the fact's privacy and integrity.

VII. REFERENCES

- [1] David Mhlanga, "The Value of Open AI for the Current Learning Environments and The Potential Future Uses", University of Johannesburg, South Africa, May 2023 SRN Electronic Journal, DOI:10.2139/ssrn.4439267
- [2] Jean Louis K. E Fendji, Diane C. M. Tala, Blaise O. Yenke & Marcellin Atemkeng, "Automatic Speech Recognition Using Limited Vocabulary: A Survey", Rhodes University, Grahamstown South Africa, 21 June 2022, Taylor & Francis Group, LLC
- [3] Taufiq-Hail Ghilan Al-Madhagy, Ayed Alanzi, Shafiz Mohd Yusof, "Software as a Service (SaaS) Cloud Computing: An Empirical Investigation on University Students' Perception", Department of Mathematics, College of Science and Human Studies, Majmaah University, Majmaah, Saudi Arabia, 2021, ResearchGate
- [4] Deepak Kadam, Prathamesh Chavan, Prashant Pandhara, "Literature Survey on Recognition and Evaluation of Optical Character Recognition (OCR)", International Journal of Scientific & Engineering Research Volume 9, Issue 2, February-2018



International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 04 /60400130637

Date: 17/04/2024

Certificate of Publication

This is to certify that author "**Piyush Chavan**" with paper ID "**IRJMETS60400130637**" has published a paper entitled "**ENHANCING DOCUMENT COLLABORATION: A SAAS PLATFORM FOR REAL-TIME COMMUNICATION AND PDF INTERACTION**" in *International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS)*, Volume 06, Issue 04, April 2024

A. Desai

Editor in Chief



We Wish For Your Better Future
www.irjmets.com







***International Research Journal Of Modernization
in Engineering Technology and Science***

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 04 /60400130637

Date: 17/04/2024

Certificate of Publication

This is to certify that author "Arpit Bharuka" with paper ID "IRJMETS60400130637" has published a paper entitled "ENHANCING DOCUMENT COLLABORATION: A SAAS PLATFORM FOR REAL-TIME COMMUNICATION AND PDF INTERACTION" in International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS), Volume 06, Issue 04, April 2024

A. Deyshi

Editor in Chief



We Wish For Your Better Future
www.irjmets.com





International Research Journal Of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

e-ISSN: 2582-5208

Ref: IRJMETS/Certificate/Volume 06/Issue 04 /60400130637

Date: 17/04/2024

Certificate of Publication

This is to certify that author "**Prajwal Ghatol**" with paper ID "**IRJMETS60400130637**" has published a paper entitled "**ENHANCING DOCUMENT COLLABORATION: A SAAS PLATFORM FOR REAL-TIME COMMUNICATION AND PDF INTERACTION**" in *International Research Journal Of Modernization In Engineering Technology And Science (IRJMETS)*, Volume 06, Issue 04, April 2024

A. Doush

Editor in Chief



We Wish For Your Better Future

www.irjmets.com



PLAGIARISM REPORT

Final Report_merged.pdf

ORIGINALITY REPORT

16%

SIMILARITY INDEX

13%

INTERNET SOURCES

8%

PUBLICATIONS

9%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Higher Education Commission
Pakistan

Student Paper

2%

2

openai.com

Internet Source

2%

3

www.coursehero.com

Internet Source

1%

4

deepnote.com

Internet Source

1%

5

Submitted to The University of Manchester

Student Paper

1%

6

vdocuments.mx

Internet Source

1%

7

sist.sathyabama.ac.in

Internet Source

<1%

8

analyticsindiamag.com

Internet Source

<1%

9

dev.to

Internet Source

<1%

PROJECT GROUP MEMBERS

Name: Piyush Chavan
Address: Aadhi Shakti Nagar, Talav Road, Khamgaon
Email id: piyushchavan2002@gmail.com
Mobile no: 9096217636



Name: Abhijeet Tathod
Address: Bondri, Shegaon
Email id: abhitathod29@gmail.com
Mobile no: 9307243841



Name: Arpit Bharuka
Address: Khandsari Parisar
Balaji Nagar, Kannad
Email id: bharukaarpit@gmail.com
Mobile no: 8180878644



Name: Prajwal Ghatol
Address: Near Old Mahadeo Temple, Shegaon
Email id: prajwalghatol100@gmail.com
Mobile no: 8999443530

