Bahria University,

Karachi Campus



COURSE: Artificial Intelligence

TERM: SPRING 2025, CLASS: BSE- 6B

PROJECT NAME:

Medical Report Analyzer

|  |  |
| --- | --- |
| Members | Enrollment |
| Rehab Khaleel | 02-131222-108 |
| Khadija Muskan | 02-131222-110 |
| Fatima Majeed | 02-131222--017 |

Engr.Faiz ul Haq / Engr. Hamza

Signed Remarks: Score:

Table of Contents

[1. Introduction & Problem 3](#_Toc199730369)

[2. Problem Statement: 3](#_Toc199730370)

[3. Technology Stack 3](#_Toc199730371)

[4. Functionalities 3](#_Toc199730372)

[5. Module Distribution 4](#_Toc199730373)

[6. Code Overview 4](#_Toc199730374)

[7. Interfaces 4](#_Toc199730375)

[a) Homepage (index.html) 4](#_Toc199730376)

[b) Analyze Page (analyze.html) 4](#_Toc199730377)

[c) Chat Interface 4](#_Toc199730378)

[8. Conclusion 5](#_Toc199730379)

# **Introduction & Problem**

The **Medical Report Analyzer** is an AI-powered web application that processes medical documents in TXT, PDF, or DOCX format. Its goal is to assist users—especially patients and medical professionals—in automatically extracting, summarizing, and understanding key health information from medical reports.

1. Problem Statement:  
   Patients and even professionals often struggle to understand lengthy and jargon-filled medical reports. There is a need for an intelligent assistant that can highlight key insights, extract medical terms, provide summaries, and answer relevant questions—all while preserving data privacy.

# **Technology Stack**

|  |  |
| --- | --- |
| Technology | Purpose |
| Flask (Python) | Backend framework |
| HTML/CSS/JS | Frontend interface |
| NLP (spaCy, NLTK) | Medical term extraction |
| Gensim | Word embedding generation |
| Google Gemini API | AI summary, insights, chatbot |
| PyPDF2, python-docx | Text extraction from documents |
| dotenv | Environment variable management |
| JSON, Regex | Data handling and parsing |

# **Functionalities**

* File Upload: Upload medical reports in .pdf, .docx, or .txt format.
* Text Extraction: Extracts raw text using appropriate libraries.
* NLP Preprocessing: Tokenizes and cleans data using NLTK.
* Entity Extraction: Detects both named entities and medical-specific terms.
* Key Term Identification: Uses Word2Vec to find important terms.
* AI Analysis: Generates a summary, insights, and recommendations using Google Gemini.
* Chatbot: Users can ask questions about the uploaded report, answered contextually by Gemini or fallback logic.
* Session Handling: Keeps report data in session for cross-endpoint usage.

# **Module Distribution**

|  |  |  |
| --- | --- | --- |
| Module | Responsibility | Team Member |
| File Upload & Extraction | File handling and text parse | Rehab Khaleel |
| NLP + Entity Extraction | NLTK, spaCy, RegEx | Khadija Muskan |
| AI Summary & Chatbot | Gemini API, Chat logic | Fatima Majeed |
| Frontend Templates | HTML Integration | Shared |
| Testing & Documentation | End-to-end checks, Report | Shared |

# **Code Overview**

**Key Functions in app.py:**

* extract\_text\_from\_file: Detects file type and extracts readable text.
* preprocess\_text: Tokenizes, lowers, removes stopwords.
* extract\_medical\_entities: Extracts medical keywords using spaCy + custom RegEx patterns.
* analyze\_with\_gemini: Sends structured prompts to Gemini for a summary, insights, and recommendations.
* generate\_word\_embeddings: Creates embeddings using Gensim Word2Vec.
* chat\_response: Handles medical question-answering from the user.
* chat\_response\_rag: Placeholder for future RAG-based implementation.

**Key Routes:**

* /: Landing page for upload.
* /upload: Backend logic to process the uploaded report.
* /analyze: Displays processed insights on a new page.
* /chat: Receives and returns chatbot responses.

# **Interfaces**

## a) Homepage (index.html)

* Upload section for reports
* Informative layout on supported formats

## **b) Analyze Page (analyze.html)**

* Report preview
* Key medical entities
* Extracted terms
* AI summary + insights + recommendations

## **c) Chat Interface**

* Input for questions
* Real-time AI-generated answers based on uploaded report context

# **Conclusion**

The *Medical Report Analyzer* successfully demonstrates the integration of AI and NLP in simplifying complex medical documentation. By leveraging spaCy, NLTK, Gensim, and Google Gemini, the system provides end-to-end intelligence—from raw file input to contextual understanding and interaction. The application can be extended with more robust RAG integration, better medical ontologies, and multi-language support.

**Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  
**Remarks:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**  
**Score:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**