Health-Matrix

PROJECT SYNOPSIS

OF MAJOR PROJECT

BACHELOR OF TECHNOLOGY

Computer Science and Engineering

SUBMITTED BY

GUNDEEP SINGH	SUKHMAN KAUR
2104100	2104199

August 2024

UNDER THE GUIDANCE OF

ER. JASWANT SINGH TAUR

GURU NANAK DEV ENGINEERING COLLEGE,
LUDHIANA

INDEX

INTRODUCTION

Efficiently managing and analysing medical reports is essential for maintaining optimal health in today’s fast-paced environment. The Health Matrix project is designed to develop a cutting-edge web application that allows users to upload their medical reports in PDF format and compare them with previous records. By tracking changes in health data over time, users can gain valuable insights into their health trends and take proactive measures.

Health Matrix is built on the robust MERN stack, comprising MongoDB, Express.js, React.js, and Node.js. This technology stack offers a reliable and scalable solution that aligns seamlessly with modern web standards, ensuring both performance and flexibility. MongoDB serves as the database backbone, providing a flexible and scalable environment for storing user data and medical reports. Express.js and Node.js power the server-side logic, handling requests and efficiently managing user sessions. React.js delivers a dynamic and responsive user interface, enabling an intuitive and seamless user experience.

A significant feature of Health Matrix is its advanced PDF extraction functionality, which converts the text within PDF medical reports into a format suitable for analysis. The application employs sophisticated data comparison algorithms to identify and highlight changes between different reports, providing users with clear and actionable information about their health progress. Additionally, secure user management is a core aspect of the project, ensuring that all user data is protected through encryption and secure authentication methods.

Overall, Health Matrix aims to empower users by giving them the tools to monitor and understand their health data more effectively.

RATIONALE

Comparing medical reports manually can be a tedious and error-prone task, often leading to the misinterpretation of vital health information. Health Matrix is developed to address the need for an automated, intuitive solution that simplifies this process. The platform allows users to upload their medical reports and seamlessly track their health data over time.

By automating the comparison and analysis of these reports, Health Matrix not only reduces the risk of errors but also provides users with clear, actionable insights into their health trends. This enables users to make informed decisions about their health, fostering proactive health management. The project ultimately aims to enhance personal healthcare by offering a reliable, user-friendly tool that supports better understanding and management of health data.

OBJECTIVES

- To create a streamlined system for data extraction using NLP.
- To deliver comparison reports and actionable health insights through engaging visualisations.
- To provide easy access to previous reports’ entities using LLM.

LITERATURE REVIEW

Ref No.	Title	Key Findings
[1]	Automating Medical Report Analysis: A Review	Explores techniques for automating the extraction and analysis of medical report data using ML and NLP.
[2]	Tracking CBC Parameters: A Comparative Analysis	Examines the significance of monitoring CBC parameters and reviews methods for comparing these metrics.
[3]	Web-Based Health Management Systems: Design and Implementation	Reviews the design and implementation of web-based health management systems, focusing on architecture, security, and UI.
[4]	The MERN Stack: Advantages and Applications	Discusses the benefits of using the MERN stack for developing scalable web applications with JavaScript.
[5]	Techniques for Extracting Data from PDFs	Provides insights into methods for parsing and extracting data from PDFs, crucial for processing medical reports efficiently.

FEASIBILITY STUDY

1. **Operational Feasibility**: Health Matrix effectively meets the rising need for digital health management by offering a streamlined and intuitive platform for users. The application's design ensures that individuals can easily track and analyse their health data, aligning with modern trends and user expectations for convenient, digital health monitoring solutions.
2. **Economic Feasibility**: Health Matrix is financially sustainable, which help keep development and maintenance costs low. This cost-effective approach makes the platform accessible and affordable for users, while still providing advanced features and capabilities that enhance personal health management.
3. **Technical Feasibility**: The project leverages the MERN stack to create a scalable and reliable application. With JWT tokens ensuring secure authentication and advanced algorithms supporting accurate data comparison, Health Matrix is built on a solid technical foundation that supports its core functionalities and user needs.
4. **Integrational Feasibility**: Health Matrix seamlessly incorporates NLP for efficient data extraction and LLM for easy access to previous report entities. This integration ensures that the application functions smoothly, providing users with a cohesive and user-friendly experience that enhances their ability to manage and understand their health data.

METHODOLOGY

Step 1: Research and Design: Project Requirements: Document functional and non-functional requirements, including the need for LLM integration to search entities and NLP for parsing and comparison. Identify user stories related to entity search, PDF parsing, and comparison features.

System Architecture: Define the architecture to include LLM integration for entity search and NLP for data extraction and comparison. Consider how these components will interact with the MERN stack, Firebase, and MongoDB.

Data Models: Design data models that accommodate the extracted data and comparison results. Ensure the models support storing and querying LLM outputs and parsed data.

Step 2: Implementation-Build the Application:

MERN Stack: Develop the frontend using React and the backend with Node.js/Express.

Authentication: Implement JWT-based authentication to secure access.

PDF Parsing: Use NLP libraries (e.g., spaCy, NLTK) to extract data from PDFs.

LLM Integration: Integrate an LLM (e.g., GPT-based model) to search for specific entities in the parsed data and generate insights.

Comparison Algorithms: Develop algorithms to compare extracted data from new PDFs with previously stored data.

Data Storage: Utilize MongoDB for storing PDF metadata, parsed data, and comparison results. Ensure efficient querying and indexing.

Step 3: Testing

Unit Tests: Write unit tests for individual components, including LLM integration, NLP parsing, and comparison algorithms.

Integration Tests: Test the interactions between different components, such as the frontend with the backend and the LLM with the NLP parser.

End-to-End Tests: Ensure the entire workflow, from PDF upload to entity search and comparison, functions correctly. Use frameworks like Jest and Mocha for comprehensive testing.

Step 4: Deployment

Backend Deployment: Deploy the backend server on cloud providers like Heroku or AWS.

Frontend Deployment: Use Vercel or Netlify for deploying the frontend application.

CI/CD Pipelines: Set up CI/CD pipelines to automate testing, building, and deployment processes.

Step 5: Monitoring and Maintenance

Monitoring Tools: Implement tools like New Relic or Sentry to monitor application performance and track errors.

Regular Updates: Periodically update the application to fix bugs, improve performance, and ensure security. Maintain the NLP and LLM components to handle evolving data and queries.

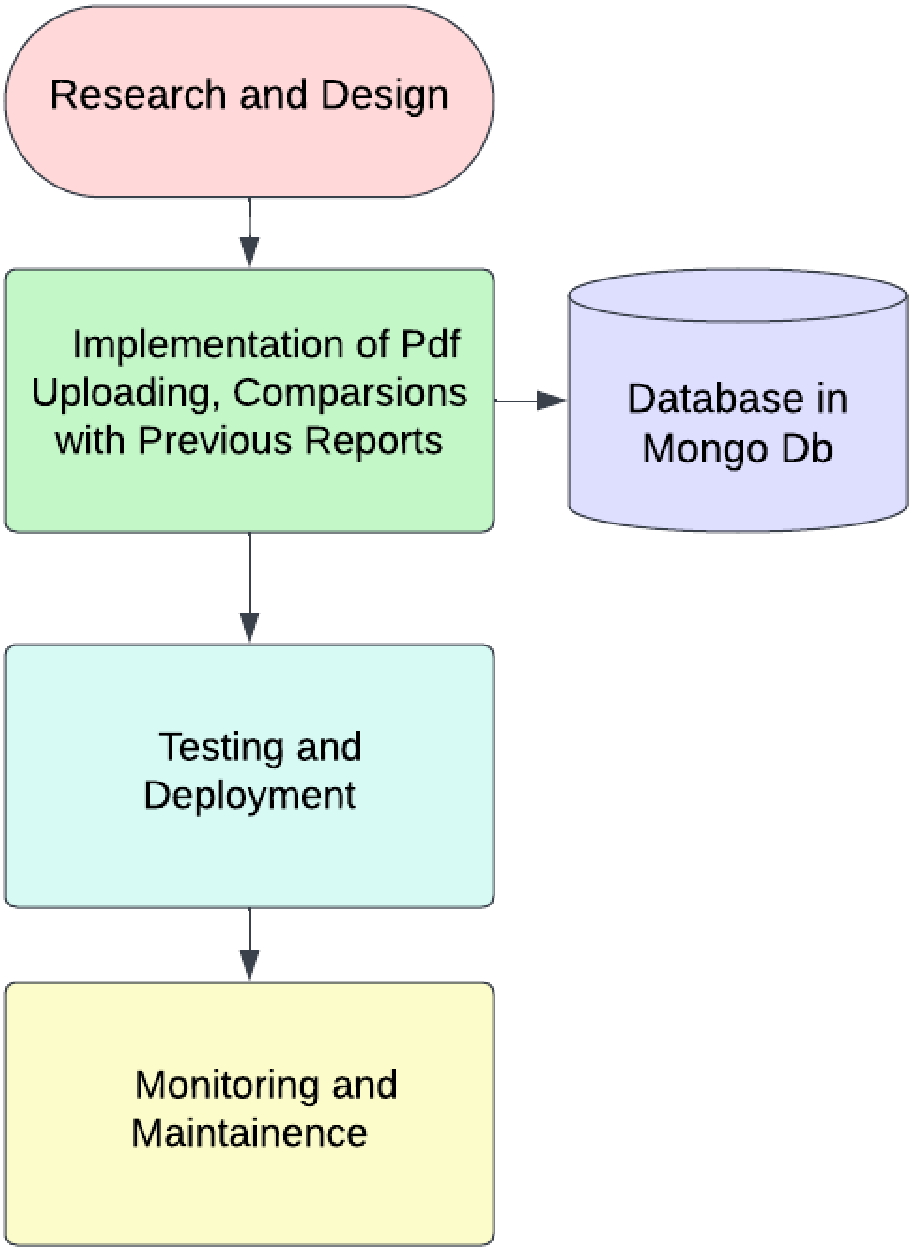
Research and Design
Implementation of Pdf Uploading, Comparsions with Previous Reports
Database in Mongo Db
Testing and Deployment
Monitoring and Maintainence

Figure 1: Workflow of Health-Matrix

FACILITIES REQUIRED FOR PROPOSED WORK

Hardware Requirements:

- Render and Vercel for Deployment
- Processor: Quad-core or higher
- RAM: 8 GB
- Storage: Solid State Drive
- Network: High speed internet connection

Frameworks to be Used:

- Node.js
- Express.js
- MongoDB
- React.js
- PDF Parsing Libraries
- JWT Libraries
- LLM (Transformers Model)

EXPECTED OUTCOMES

The expected outcomes of the project are:

- Users will be able to:
  1. Create and manage their accounts securely.
  2. Upload and store medical reports in PDF format.
  3. Compare newly uploaded PDFs with previous test reports.
  4. Search for specific results within their reports using an advanced language model (LLM).
  5. Visualize their medical data through intuitive and interactive graphs
- The project will facilitate the seamless management and organization of medical records.
- Users will receive actionable health insights based on the analysis of their medical data.

REFERENCES

- GeeksforGeeks, "5 Simple Steps for Authentication and Authorization in MERN Stack." [Online]. Available: https://www.geeksforgeeks.org/5-simple-steps-for-authentication-and-authorization-in-mern-stack/. [Accessed: 23-Aug-2024].
- GetStream.io, "Building an LLM Chatbot: A Beginner's Guide." [Online]. Available: https://getstream.io/blog/llm-chatbot-docs/. [Accessed: 23-Aug-2024].
- P. Yadav, "How to Extract Text from a PDF – NLP," Medium, 28-Jun-2023. [Online]. Available: https://ydv-poonam.medium.com/how-to-extract-text-from-a-pdf-nlp-b64409422cfd2. [Accessed: 23-Aug-2024].