International Journal of Creative and Open Research in Engineering and Management
ISSN: **3108-1754 (Online)**
Volume 02 Issue 04 April-2026 | Impact Factor: 3.5

GenAI Assistant Analysis for Health Record Management

Dr. Sheryl Radley [1], Hemanath Kumar R [2], Muthu Murugan Harish S[3], Syed Abdhul Kadhar S [4]
[1]Department of Electronics and Communication Engineering
[2,3,4]*Department of Information Technology*
Meenakshi College of Engineering, West K.K. Nagar, Chennai
sherylradley@gmail.com[1], heamrvsh@gmail.com [2], muthumuruganharish426@gmail.com [3]
syedabdhul21@gmail.com [4]

How to Cite this Article:
S, S. A. K., S, M. M. H. & R, H. K. (2026). GenAI Assistant Analysis for Health Record Management. International Journal of Creative and Open Research in Engineering and Management, 02(04). https://doi.org/10.55041/ijcope.v2i4.952

https://doi.org/10.55041/ijcope.v2i4.952

***Abstract*—**Access to timely and accurate analysis of medical data remains a major challenge due to the increasing volume of health records, lack of automation, and limited availability of intelligent processing systems. This This paper is about a system that uses GenAI to analyze health reports. The GenAI-based Health Report Analysis and Management System is designed to help doctors and patients understand documents.This thing uses Optical Character Recognition to change reports into text that computers can understand. It takes these reports. Turns them into something that computers can read. Then it uses Large Language Models to figure out what the text actually means. After that it uses Machine Learning to try to predict what might happen to a patients health. It is really using Optical Character Recognition and Large Language Models and Machine Learning to do all of this for the patients health. The system also has a feature that lets users ask questions about their medical data and get answers that are relevant to them.This works because of a framework called Retrieval-Augmented Generation.

The system is like a team of workers that do different jobs and work together. These workers are built using something called microservices architecture. * The part that users see called the frontend is built with React. The backend services, which do the lifting and the authentication part, which checks who is who are built with Django. The AI processing, which is, like the brain of the system is done with FastAPI. All the data is stored in a MySQL database, which's like a big filing cabinet, for computers. The system also uses a vector database, which helps the system find the information quickly in the MySQL database and the vector database. The MySQL database and the vector database work together to make it easy to find the information.. The people who made the system tested it. Found that it works well. It can process reports extract the important information and even predict what might happen to a patients health. The GenAI-based Health Report Analysis and Management System is an improvement, over the old way of doing things. It saves time. Makes it easier for people to get the information they need. The system uses different technologies, including GenAI, Health Record Management, OCR, Large Language Models, RAG, Machine Learning, Django, FastAPI and Disease Prediction. hese technologies all work together to make the system work. The system is an example of how Healthcare Analytics can be used to improve peoples lives. Healthcare Analytics is really making a difference here. We use Generative AI, Health Record Management and OCR to get things done. Large Language Models and RAG help us understand and connect the dots. Machine Learning is also key, well as Django and FastAPI. All these come together for Disease Prediction and more. It's all, about using Healthcare Analytics to help people. Healthcare Analytics helps us make peoples lives better. We are using Healthcare Analytics for peoples lives improvement.

Keywords: *Generative AI, Health Record Management, OCR, Large Language Models, RAG, Machine Learning, Django, FastAPI, Disease Prediction, Healthcare Analytics.*

I. INTRODUCTION

Healthcare systems worldwide are facing challenges. The population is growing more people are getting sick and managing medical data is tough. We normally wait until people are really sick before we give them help. This usually means that we find out what is wrong with them late and they do not get the care they need. We should be treating problems like this when people are not as sick so people can get better faster. This is a problem, with the way we do things now. It affects the medical care that people get.This issue is particularly severe for those living in resource- areas. People in these areas struggle to access quality healthcare. They have limited access to facilities and staff. As a result they often receive treatment. The way we do healthcare now is not working. We need to change the healthcare approach. The healthcare approach needs to change so we can get results. We need to change the healthcare approach so we can get results with the healthcare approach. We should try to stop people from getting sick at home and in the community and help them soon as they get sick. The healthcare approach should help people soon as they need it with the healthcare approach. This means the healthcare approach should focus on keeping people from getting sick in the place and helping them right away if they do get sick with the healthcare approach. The healthcare approach needs to change to include prevention and helping people early to get results, with the healthcare approach.

The healthcare approach can be improved by using technology and data to make things better. This can be achieved by using technology and data to help with the healthcare approach. By changing the way we do things with the healthcare approach we can make people healthier. By adopting an approach, to the healthcare approach we can improve healthcare outcomes and make the healthcare approach better. This is especially crucial for people in areas with resources. They deserve access to quality healthcare. We must address these challenges to create a healthcare system. This requires an effort from governments, healthcare providers and individuals. Together we can make an impact, on peoples lives. We also have a lot of trouble with records. We have to deal with things like laboratory reports and prescriptions by hand. This takes a time and it is easy to make mistakes. It is also hard to keep these records safe. Healthcare professionals have to do the tasks over and over. They have to look at reports and explain what they mean to patients. This means they have time to actually take care of patients. At the time patients often have a hard time understanding what their healthcare professionals are telling them. They do not know what the medical words mean so they have a time making good choices, about their health. Healthcare systems also have to worry about keeping data safe. This is a problem when we try to use digital healthcare systems.

1.1 Aim

The goal of this project is to create a GenAI-based helper for looking at and managing health records. This helper will fix problems we have in healthcare now like people having to enter data by hand not being able to get to the information and having trouble keeping the data safe. The system uses a tool to take information from medical papers and big language models to turn messy data into useful information. It also uses computer programs to guess who might get very sick in the future. The GenAI-based helper uses these tools to make health records easier to work with and understand and to help doctors and patients make decisions about health care. The GenAI-based helper is very good at looking at health records and finding information and it can even help doctors find people who are at risk of getting very sick. The project is, about using GenAI to make health records better and to help people get the care they need. A RAG framework enables users to query their medical data and receive context-aware responses in a secure, scalable environment.

1.2 Objectives

The primary objectives of this work are:

- We want to use computers to get information from medical papers like lab results and prescriptions. We will use something called OCR to do this.
- We need to take the text from medical papers and turn it into something that makes sense. We will use computer programs called Generative AI and LLMs to do this.
- We will set up a system called RAG that lets people talk to their information and get answers that are just for them.
- We will use computers to try to guess if someone is likely to get a long-term sickness based on how healthy they're. This can really help doctors find problems, on.
- We will make sure that peoples medical information is safe and private. We will use a kind of lock called AES-GCM encryption to keep peoples medical information safe and private. using Django and FastAPI that can handle a lot of users.
- We want to make a website that's easy for people to use even if they are not good with computers or live in areas that are hard to reach. This website will be, for patients who need to see their information.

II. LITERATURE SURVEY

The use of Artificial Intelligence in healthcare is one of the changes we have seen in a long time. Artificial Intelligence technologies, like Natural Language Processing and Machine Learning and Generative Artificial Intelligence are really helping us with data. We can process it. Understand it and use it better now. This part of the work looks at what other people have found out about the parts of the Artificial Intelligence system we are proposing..

El-Sofany [1] proposed a system for prediction of heart diseases using machine learning. This system focused on improving the prediction accuracy of heart diseases by using classification algorithms. These algorithms include Random Forest and Support Vector Machine and K-Nearest Neighbours. The main goal of this study was to show how important it is to diagnose heart diseases on. This is because heart diseases can be very serious.

Vasudevan et al. [2] developed a system for early disease prediction using supervised machine learning algorithms. The study emphasized the use of structured medical data to predict diseases at an early stage. Their approach helps in reducing delays in diagnosis and supports healthcare professionals in making better decisions.

Sanap et al. [3] introduced **MediLens AI**, an intelligent platform designed for medical report summarization and clinical workflow automation. The system uses artificial intelligence to extract key insights from medical reports and present them in a simplified manner. The main objective was to reduce the workload of healthcare professionals and improve efficiency in clinical processes.

Brown et al. [4] presented the concept of large language models as few-shot learners. Their work demonstrated that models trained on large datasets can perform various tasks with minimal examples. This study showed the potential of language models in understanding and generating human-like text, making them useful in applications like medical report analysis.

Lewis et al. [5] proposed Retrieval-Augmented Generation (RAG), a method that combines information retrieval with text generation. The system retrieves relevant data from external sources and uses it to generate more accurate and context-aware outputs. This approach improves the reliability of language models, especially in knowledge-intensive tasks.

Smith [6] provided an overview of the Tesseract OCR engine, a widely used tool for extracting text from images. The system enables the conversion of scanned documents into machine-readable text. This technology plays an important role in processing medical reports that are available in image or PDF formats.

Vaswani et al. [7] introduced the Transformer architecture, which is based on the attention mechanism. This model significantly improved the performance of natural language processing tasks by enabling parallel processing and better context understanding. It forms the foundation for many modern AI models used today.

Qwen Team [8] presented the Qwen language model, a large-scale AI model developed for advanced text understanding and generation. The model is capable of handling complex queries and generating meaningful responses, making it suitable for applications like healthcare data analysis and report interpretation.

LangChain [9] provides a framework for building applications using large language models. It helps in integrating LLMs with external data sources, APIs, and workflows. This framework is useful in developing intelligent systems that require dynamic data processing and interaction.

ChromaDB [10] is an open-source embedding database designed for storing and retrieving vector data efficiently. It supports similarity search, which is essential for applications like Retrieval-Augmented Generation. This tool helps in managing large amounts of data and improving the performance of AI-based systems.

III. SYSTEM ANALYSIS

3.1 Existing System

The healthcare system we have today is mostly done by hand and it reacts to things after they happen especially when it comes to handling data and managing records. Patients usually have to rely on doctors and nurses to explain things like lab results and prescriptions which can slow down the process of understanding what is going on and making decisions. Doctors and nurses have to look over analyze and explain these records, which takes a lot of time and can lead to mistakes when they are very busy. The Health Report Analysis Platform is made up of parts that work well together. This system has four parts: the Presentation Layer, the Application Layer, the AI Processing Layer and the Data Layer. The Presentation Layer is the part that users actually see and interact with when they use the system. The Presentation Layer is, like the face of the system it is what users deal with directly. It is built with React. Gives users a simple way to upload Health Reports look at results and use the Health Report Analysis Platform. The Application Layer is made with Django. It takes care of making sure users are who they say they are handling requests, from the Health Report Analysis Platform and figuring out what the Health Report Analysis Platform should do. It uses OTP and JWT for secure login and AES-GCM encryption to protect user data.

The AI Processing Layer uses FastAPI to perform tasks such as extracting text from reports using OCR, converting data using LLM inference, and generating predictions using machine learning models. The Data Layer stores all information, where MySQL is used for structured data and a vector database supports fast semantic searching in the RAG module.In places where

computers are used many hospitals and clinics use something called Electronic Health Record systems to store information. However these systems are mostly a place to store data and they need people to put information in and interpret it. They do not have the ability to automatically look at documents process the information and come up with useful insights on their own. Also most of these systems do not have ways to protect sensitive health information. The fact that they do not use intelligence to predict what might happen next means that doctors and nurses cannot diagnose problems early on or take care of patients in a proactive way. Healthcare systems like these need to be improved so that healthcare professionals can focus on taking care of patients of just dealing with paperwork and records. The healthcare system needs to use things, like Electronic Health Record systems in a way so that patients can get the care they need quickly and easily.

3.1.1 Drawbacks of Existing Systems

- Manual processing of records takes a lot of time and people make mistakes.
- We do not have systems to read and understand medical information like lab reports and prescriptions that are not organized.
- It is hard to get information from medical records for people who are not doctors.
- We have different systems for healthcare and people have to use many platforms to get what they need.
- We do not have systems that can look at information in real time and predict when someone might get sick.
- Medical records are not safe and private like they should be.
- Doctors have to do a lot of work to look at information and this takes a lot of time.
- People who live away from cities or in areas with limited healthcare do not have easy access, to medical records and services.

3.2 Proposed System

To fix the problems with healthcare systems we are suggesting a new GenAI-based Health Record Analysis and Management System. This system brings together a smart platform that uses artificial intelligence to automatically handle, understand and safely manage medical data. The system uses OCR to pull out text from reports. It uses LLMs to turn text into neat and easy-to-understand information. It also uses machine learning algorithms to predict the risk of getting diseases, like diabetes or heart disease. The GenAI-based Health Record Analysis and Management System helps make sense of data. The system helps doctors and patients make decisions. The GenAI-based Health Record Analysis and Management System makes healthcare better. We are using the GenAI-based Health Record Analysis and Management System to improve healthcare. The GenAI-based Health Record Analysis and Management System is a platform. The unified platform uses intelligence techniques. The artificial intelligence techniques are advanced. The system helps patients and doctors. The system makes healthcare easy. The system is a GenAI-based Health Record Analysis and Management System.

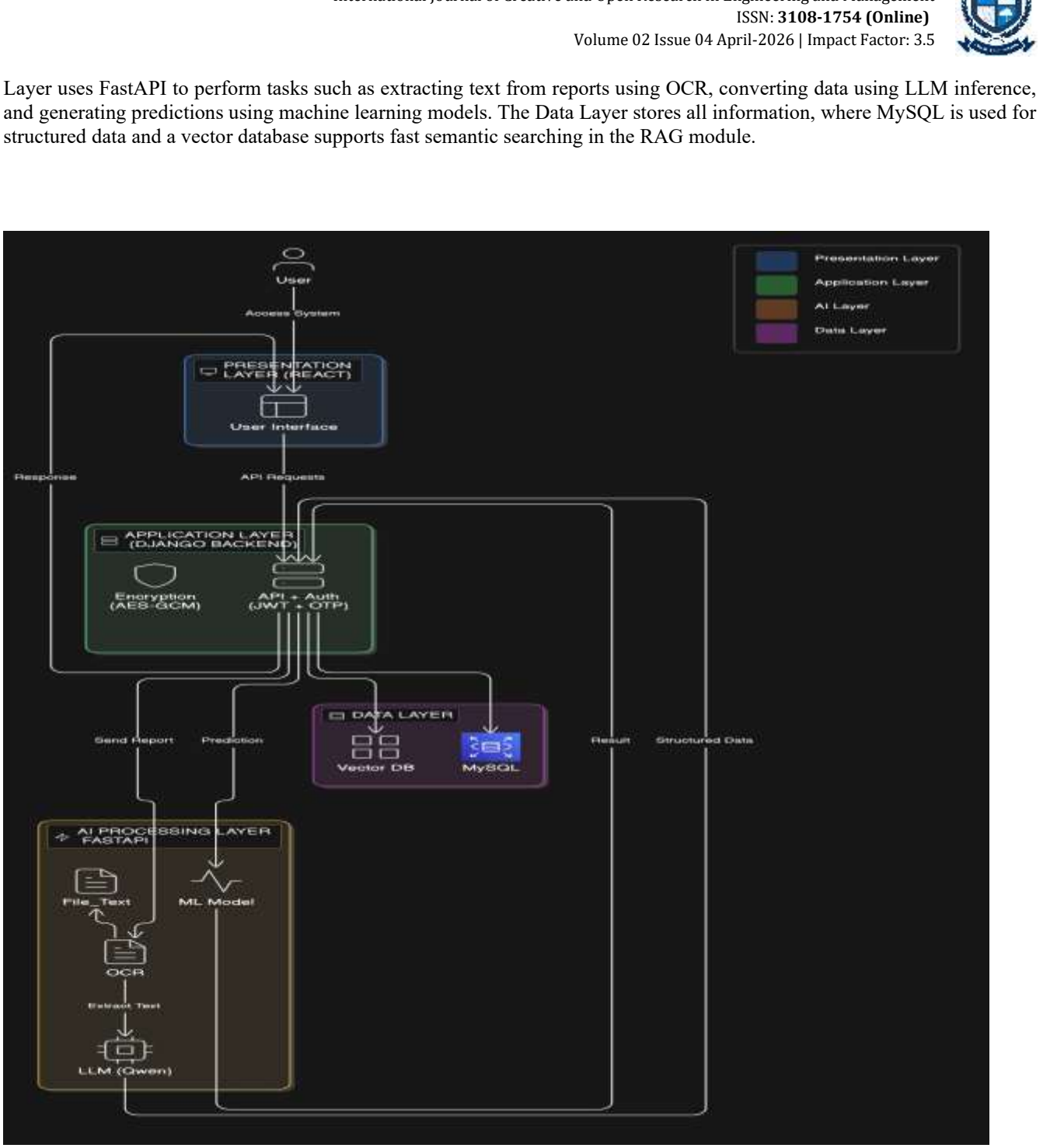
Additionally, a RAG framework is incorporated to provide context-aware responses, allowing users to interact with their medical data through intelligent query-based insights. A key aspect of the proposed system is its microservices-based architecture, where the backend is divided into a Django-based gateway for data management and a FastAPI-based service for AI processing. The system is cloud-deployable and implements strong data security measures using AES-GCM encryption to protect sensitive medical information during storage and transmission.

3.2.1 Advantages of the Proposed System

- Automated medical data processing using AI significantly reduces manual effort in analyzing lab reports and prescriptions.
- We have a system that helps manage health records in one place so we do not need to use different systems.
- This system uses something called LLMs to give us simple and easy to understand summaries of medical reports.
- It also uses things like OCR and RAG to get the information from reports and answer our questions in a way that makes sense.
- We want to keep information safe so we use a strong kind of encryption called AES-GCM.
- Our system is made up of small parts that work together using things like Django and FastAPI which makes it easy to use and update.
- People can use this system from anywhere which is really helpful, for people who live in areas where it's hard to get medical help.

IV. SYSTEM ARCHITECTURE AND DESIGN

The Health Report Analysis Platform is made up of parts that work well together. This system has four parts: the Presentation Layer, the Application Layer, the AI Processing Layer and the Data Layer.The Presentation Layer is the part that users actually see and interact with when they use the system. The Presentation Layer is, like the face of the system it is what users deal with directly. It is built with React. Gives users a simple way to upload Health Reports look at results and use the Health Report Analysis Platform. The Application Layer is made with Django. It takes care of making sure users are who they say they are handling requests, from the Health Report Analysis Platform and figuring out what the Health Report Analysis Platform should do. It uses OTP and JWT for secure login and AES-GCM encryption to protect user data. The AI Processing



Layer uses FastAPI to perform tasks such as extracting text from reports using OCR, converting data using LLM inference, and generating predictions using machine learning models. The Data Layer stores all information, where MySQL is used for structured data and a vector database supports fast semantic searching in the RAG module.

Fig. 1. System Architecture Diagram

4.1 UML Diagrams

The system is modelled using standard UML notations. The Use Case Diagram represents interactions between the User actor and the system's core functions: Register/Login with OTP+JWT, Upload Medical Report, View Analysis Results, Query Insights via RAG, and Disease Prediction. All functions follow a sequential dependency beginning with authentication. The database structure is shown in the Entity Relationship Diagram. This diagram is about the User. The User has some information like id, name, email and date_of_birth. The User is connected to some things. The User is connected to OTP. This means one User can have OTPs. Each OTP has a code and an expiry time. The User is also connected to PersonalDetail. This is one to one. So one User has one PersonalDetail. PersonalDetail has things like phone number, blood type, gender and allergies. The User is connected to MedicalReport. This is one to many. So one User can have MedicalReports. Each MedicalReport has a report UUID, a summary and a created date.

The MedicalReport is connected to OCR Data. OCR Data has the text that was extracted from the MedicalReport. The User is connected to Prediction. This is one to many. So one User can have Predictions. Each Prediction has a result, a probability

and a creation datetime. The Data Flow Diagram shows how all the information moves around. The User logs in. Then the User uploads reports. The OCR module looks at these reports. Extracts the text. The LLM takes this text. Makes sense of it. The prediction module looks at this information. Tries to figure out if the User has any health risks. The RAG module answers any questions the User has. All of this information is then given back, to the User as insights and notifications.

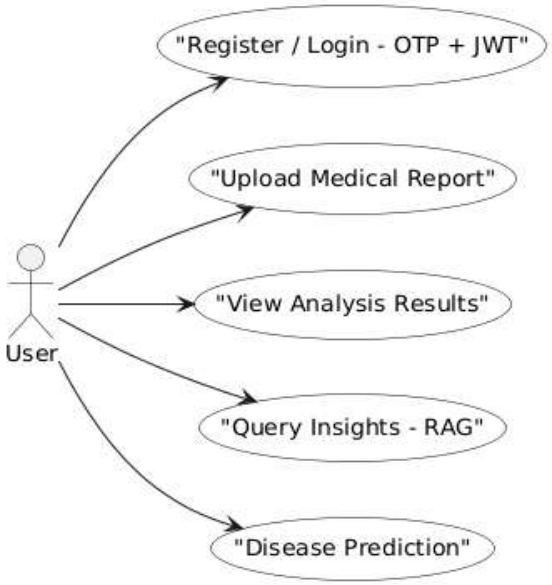


Fig. 2. Use Case Diagram

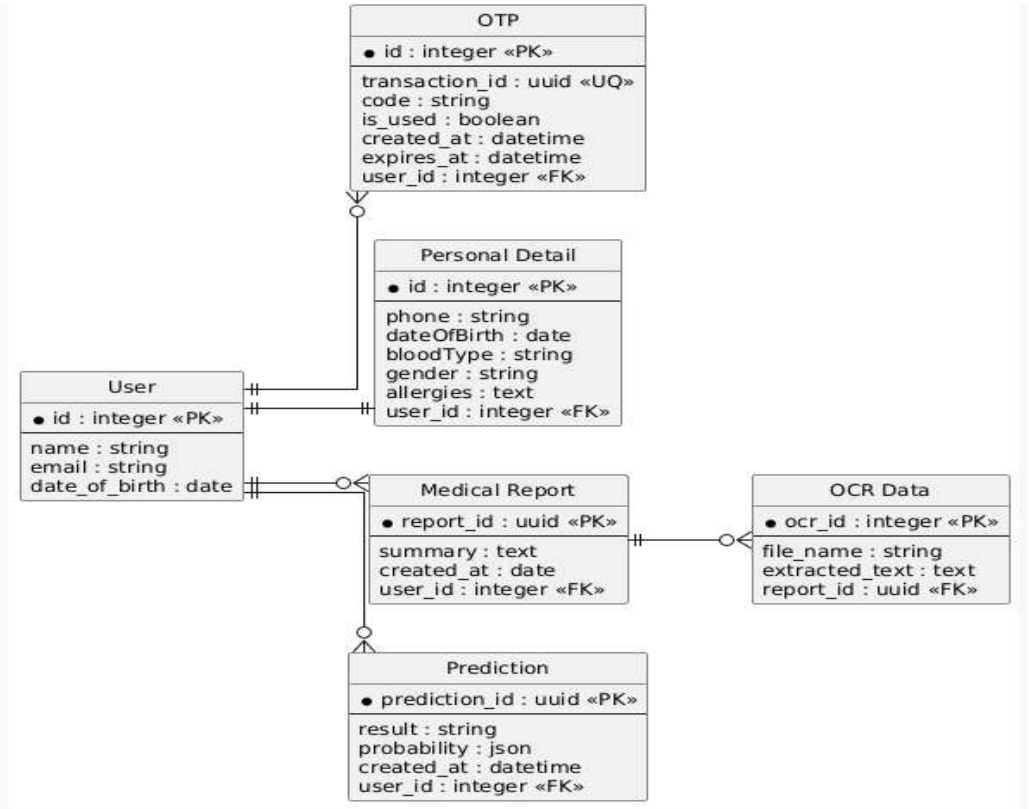


Fig. 3. Entity Relationship (ER) Diagram

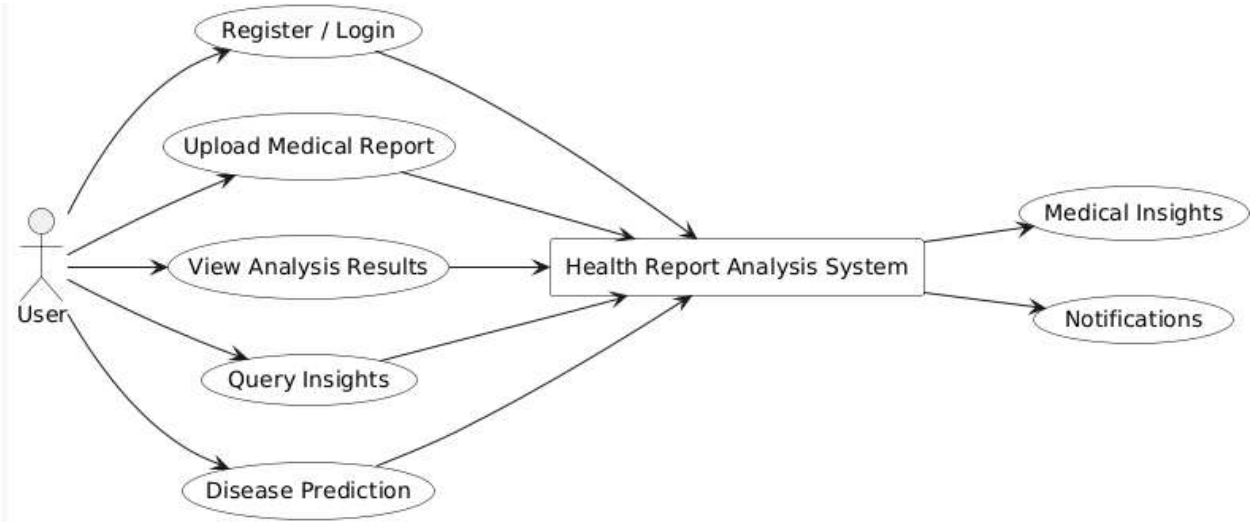


Fig. 4. Data Flow Diagram

V. SYSTEM DESCRIPTION AND MODULES

The GenAI Health Report Analysis System is composed of seven functional modules, each responsible for a distinct operation within the overall pipeline.

5.1 Authentication Module

The Authentication Module is really important for keeping the system safe. It uses a code called an OTP to verify who the user is. The Authentication Module also uses something called JWT authentication. When you want to register or log in the system generates this OTP. Checks it to make sure it is correct. After that you get a token that you need to use when you make requests to the system. The Authentication Module does a job of keeping your information private and making sure that communications, with the system are secure. The Authentication Module is very important for the security of the system and the Authentication Module helps to keep your data safe.

5.2 Medical Report Upload Module

This module allows users to upload medical documents such as lab reports or prescriptions in image or PDF format (up to 10 MB per file). The uploaded files are stored securely, metadata is recorded in MySQL, and the module acts as the entry point for all downstream AI processing.

5.3 OCR Processing Module

The medical documents that you upload are used by the OCR module to get the text from them. It uses a tool called the zai-org/GLM-OCR vision-language model. This tool is loaded with something called PyTorch. It can work with CUDA and CPU inference.

The OCR module changes the things that're in pictures into text that the computer can understand. This helps get the information from the documents. The medical information includes things like the results of tests and the details of reports. It also includes the data, from the medical documents. The OCR module does this so that the medical information can be used by the computer.

5.4 LLM Data Structuring Module

The LLM module takes the text that we get from OCR. It makes sense of it. It uses the Qwen model to turn this text into something that's easy to understand. The LLM module figures out what the medical words mean it puts the information into a format that's easy to read and it makes a simple summary of the main points using the Qwen model. The LLM module is really good, at dealing with terminology and it helps to organize the data in a way that is easy to read. This module simplifies complex medical content using generative AI techniques, enabling non-specialist users to understand their reports.

5.5 Prediction Module

The Prediction Module uses a logistic regression classifier (serialized via pickle) to analyze structured health data and generate risk predictions. The information we look at includes the persons gender, how old they're how many cigarettes they smoke

each day if they take medicine for blood pressure if they have high blood pressure if they have diabetes their total cholesterol level, their systolic blood pressure, their body mass index and their glucose level.

5.6 RAG Query Module

The Retrieval-Augmented Generation module allows users to query their medical data using natural language. It uses HuggingFace's all-MiniLM-L6-v2 sentence transformer for embedding generation, stores vectors in a ChromaDB Cloud instance, and uses a LangChain ChatPromptTemplate to compose prompts that are passed to the Qwen LLM. Responses are strictly grounded in the user's own medical data, preventing hallucinated generic advice.

5.7 Deployment Module

The system is containerized using Docker for consistent environments across development and production. The React frontend is deployed on Vercel for fast global delivery. The Django backend is deployed on Render, handling authentication, API routing, and business logic. The FastAPI AI microservice is deployed separately on Render to manage computationally intensive OCR, LLM inference, and prediction tasks. MySQL is used for structured storage, and ChromaDB Cloud serves vector embeddings for the RAG pipeline.

VI. SYSTEM IMPLEMENTATION

The system implementation follows a modular, iterative development process covering all core components.

6.1 Hardware and Software Specifications

Component	Specification
Processor	Intel Core i3/i5/i7 (Multi-core recommended)
RAM	32 GB (minimum 8 GB)
Storage	500 GB SSD
GPU	NVIDIA CUDA-enabled GPU (recommended for LLM inference)
Network	High-speed internet, low latency
OS	Windows 10/11 or Linux-based systems
Frontend	React.js (JavaScript)
Backend	Django (Python), FastAPI (Python)
Database	MySQL, ChromaDB Cloud (vector DB)
AI Libraries	Transformers (HuggingFace), LangChain, PyTorch, OpenCV, NumPy, Pandas, scikit-learn
Security	AES-GCM Encryption, OTP, JWT
Deployment	Docker, Vercel (frontend), Render (backend/FastAPI)

Table 1. Hardware and Software Requirements

6.2 Frontend Implementation

The React-based frontend is really easy to use. It has five main parts. The Health Dashboard is one of them. It shows how your hemoglobin is doing over time and some other important information from the reports you upload. The React-based frontend also has a part called Upload Medical Reports where you can add your reports. It works with PDF, JPG and PNG files that are up to 10 MB. You can just drag and drop them.Then there is the Report Analysis part of the React-based frontend which gives you a summary made by a computer the values it found and how they compare to what's normal. The Disease Detection part of the React-based frontend tells you what it thinks might be wrong with you how sure it is and what you should do about it.

6.3 Backend Implementation

The Django backend manages user authentication (OTP generation, JWT issuance, AES-GCM data encryption), REST API routing, MySQL operations, and asynchronous task dispatching to the FastAPI microservice via HTTPX. The FastAPI service exposes endpoints for OCR processing, LLM structuring, prediction inference, and RAG query handling. An async OCR job queue was implemented to handle Render's 30-second timeout constraint, ensuring reliable report processing even for large documents.

6.4 ML Model Implementation

The heart disease risk classifier uses logistic regression trained on the Framingham Heart Study (TenYearCHD) dataset. Features are scaled using a StandardScaler, and the model along with scaler and threshold are serialized in a single ML_model_bundle.pkl file. The FastAPI endpoint (POST /MLmodel/predict) loads this bundle, constructs a pandas DataFrame from the request payload in exact feature column order, applies the scaler, and returns both a binary result and probability score. Threshold optimization via ROC-AUC analysis was used to balance precision and recall for clinical relevance.

6.5 RAG Pipeline Implementation

The RAG pipeline class (rag_pipeline_class) uses LangChain's ChatPromptTemplate to construct structured prompts. The system prompt instructs the LLM to generate strictly report-grounded insights, prohibiting generic advice. Embeddings are generated using all-MiniLM-L6-v2 and stored in ChromaDB Cloud. For prediction insight summaries, the system retrieves the relevant prediction record from the vector database, constructs a prompt with the report context and a question type identifier, invokes the Qwen LLM chain, and stores the resulting insight back into ChromaDB with metadata (content type, user ID, prediction ID, and question ID) for future retrieval.

VII. RESULTS AND DISCUSSION

The GenAI Health Report Analysis System was fully implemented, deployed, and evaluated. All modules functioned correctly, demonstrating stable and scalable performance across the React frontend, Django backend, FastAPI AI microservice, and cloud databases.

7.1 Health Dashboard

The Health Dashboard provides a centralized overview of patient health metrics. It displays hemoglobin trend graphs plotted from extracted lab report values, backend server status indicators (Django Online / FastAPI Online), and additional automatically extracted metrics. The clean, dark-themed design allows users to monitor health changes over time without requiring clinical expertise.

Fig. 5. Health Dashboard

7.2 Medical Report Upload

The medical report upload interface is really easy to use. You can choose report files, like PDF or JPG or PNG files and add them to the website. The files that you are going to use can be as big as 10 MB each. You also get to choose the date, for the report. Then you can send it in for processing. The upload interface has buttons that say upload and cancel so it is easy for anyone to use even if they are not good, with computers.When you send in the file it gets added to a line of files that need to be looked at by the FastAPI microservice. This microservice uses something called OCR to look at the report files and it does this one by one.

Fig. 6. Medical Report Upload Interface

7.3 Report Analysis

The Report Analysis page is where you can see the parts of your medical reports. It shows you what your medical reports say, like what your hemoglobin level's which is 15.8 g/dL and that is, within the normal range of 12 to 17. It also tells you about your hematocrit levels. The Report Analysis page gives you a summary of your medical report. This summary explains what your medical report says and tells you what you can do, like change what you eat or go see a doctor. The Report Analysis page helps people understand their reports easily.

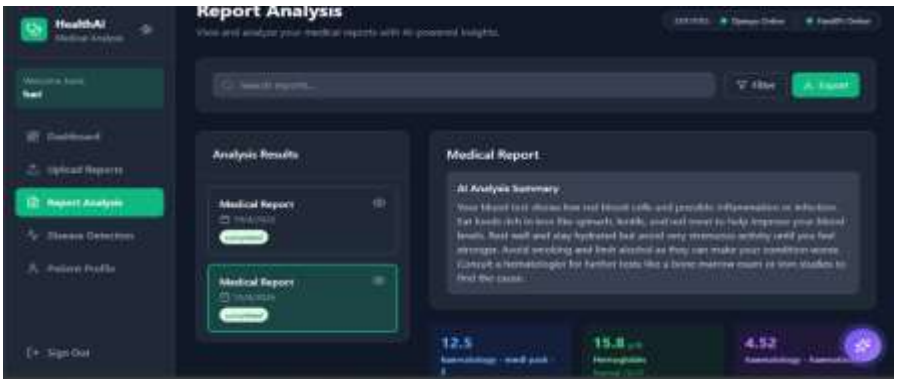


Fig. 7. Report Analysis Page with AI-Generated Summary

7.4 Disease Detection

The Disease Detection tool takes in some health information from the user, such as age, sex, how many cigarettes they smoke per day if they take blood pressure medication if they have hypertension, diabetes their total cholesterol level, systolic blood pressure, BMI and glucose level. It then gives a simple yes or no prediction of the risk of heart disease along with a percentage score showing how confident it is in this prediction.

When we tested it out with some sample information it said there was a risk of heart disease, with 87 percent confidence. It also gave a report that said the risk level was high based on the data it analyzed. The report recommended some actions to take including going to see a cardiologist, making some changes to diet, starting to do some exercise, and keeping an eye on blood pressure.

Fig. 9. Disease Detection Page showing Heart Disease Prediction at 87% Confidence

7.5 Patient Profile

The Patient Profile page is where you can find all the information about a patient. You can see the Patient Profile. Make changes to it. The Patient Profile has things like the patients name, email, phone number and date of birth. It also has room for information like what type of blood the patient has. The Patient Profile shows if the patient is a man or a woman. It lists what the patient is allergic.The page is easy to use so people can keep all the information correct and this helps the Patient Profile and the computer system that looks at all the information. The Patient Profile page is used to keep track of the patients information and the computer system uses this information to make decisions. The Patient Profile is important because it has all the information, about the patient.

VIII. SYSTEM TESTING

Comprehensive testing was conducted across all modules at unit, integration, and system levels.

8.1 Testing Methodologies

Black-box testing was applied to all user-facing features: registration/login with OTP+JWT, medical report upload, structured output display, prediction results, and RAG query responses. White-box testing examined internal API logic (Django views and FastAPI endpoints), OCR and LLM processing pipelines, error handling, and database operations. Performance testing assessed OCR processing time for large reports, LLM response latency, prediction model execution time, and API response time under concurrent requests.

8.2 Test Cases

ID	Test Case	Input	Expected Result	Status
TC01	User Registration with OTP	Valid email, OTP code	Account created, JWT token issued	Pass
TC02	Invalid OTP login	Wrong OTP	400 error: Invalid OTP	Pass
TC03	Upload PDF report	Valid JWT + PDF file	Report saved, OCR queued	Pass
TC04	Upload oversized file	File > 10 MB	400 error: File too large	Pass
TC05	OCR text extraction	Clear lab report image	Accurate text extracted	Pass
TC06	LLM structured summary	OCR extracted text	Structured summary generated	Pass
TC07	ML prediction – high risk	High-risk health parameters	High risk, probability > 0.5	Pass
TC08	ML prediction – low risk	Normal health parameters	Low risk, probability ≤ 0.5	Pass
TC09	RAG query – relevant	Question about uploaded report	Accurate, grounded answer	Pass
TC10	RAG query – generic advice	Generic health question	Report-grounded response only	Pass
TC11	Unauthorized API access	Request without JWT token	401 Unauthorized returned	Pass
TC12	AES-GCM encryption check	Stored medical record	Data encrypted at rest	Pass

Table 2. System Test Cases and Results

IX. CONCLUSION AND FUTURE ENHANCEMENT

9.1 Conclusion

This paper presented a GenAI-based Health Report Analysis and Management System that successfully addresses the challenges associated with manual interpretation of medical reports. The system uses a few things like OCR and Large Language Models and Machine Learning all together to make something new. It also uses a framework that helps it generate information. This all happens in an scalable system that is made up of many small parts. The system is built with things, like React and Django and FastAPI. This means that the system is made in a way that's easy to change and update and it can handle a lot of work. The use of MySQL for structured data storage and ChromaDB Cloud for vector embeddings supports efficient real-time processing. The system keeps your information private and safe, with a code sent to your phone, known as OTP and a way of scrambling data called AES-GCM encryption. This is really crucial when it comes to healthcare data. The use of

OTP and AES-GCM encryption helps to protect healthcare data. So the system uses OTP and AES-GCM to keep healthcare data safe.

The system was. It works very well. It can read reports and give simple summaries that people who are not doctors can understand. The system can also look at the information. Tell people if they might have health problems. People can ask the system questions about their health information. The system shows that Generative AI can really help with healthcare data. The system makes it easier to manage records and get information from the records. The system is good, for the users because the system is easy to use and the system works well. Generative AI is used in the system to make it work. The system is good, at managing records and helping people understand their health information. The system uses Generative AI to make healthcare data analysis.

9.2 Future Enhancements

Several directions are available for extending the system in subsequent work:

- The system will be more useful if we add voice-based interaction so people can talk to the system without having to use their hands to type or click on things.. This will be really helpful for people or people who are differently-abled.
- We can also make the system work in languages so people from all over India can use it even if they speak different languages at home.
- The system can be connected to devices like smartwatches and fitness bands so it can get health information, from these devices all the time.
- We can also connect the system to Electronic Health Record systems so doctors can see a patients history easily.
- The system can use Artificial Intelligence to suggest treatments based on the medical information and the patients health.
- We can make the system work when the internet is not working well so people can still use it when they need to.

REFERENCES

[1] El-Sofany, H. F., "Predicting Heart Diseases Using Machine Learning and Different Data Classification Techniques," International Journal of Advanced Computer Science and Applications, 2023.

[2] I. Vasudevan, G. Saravanan, M. Senthil Kumar, M. Mohamed Nasurudeen, R. Sadaieswaran, and R. M. Dilip Charaan, "Early Disease Prediction using Supervised Machine Learning," in Proc. 3rd International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), 2025.

[3] M. Sanap, S. Bhandary, S. Mhasde, S. Pande, S. Dedgaonkar, and A. Suryawanshi, "MediLens AI: An Intelligent Platform for Medical Report Summarization and Clinical Workflow Automation," in Proc. IEEE 5th International Conference on ICT in Business Industry & Government (ICTBIG), 2025.

[4] T. Brown et al., "Language Models are Few-Shot Learners," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, pp. 1877–1901, 2020.

[5] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020.

[6] R. Smith, "An Overview of the Tesseract OCR Engine," in Proc. 9th International Conference on Document Analysis and Recognition (ICDAR), pp. 629–633, 2007.

[7] A. Vaswani et al., "Attention Is All You Need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017.

[8] Qwen Team, Alibaba Cloud, "Qwen Technical Report," arXiv preprint arXiv:2309.16609, 2023.

[9] LangChain, "LangChain Documentation," [Online]. Available: https://python.langchain.com/docs/

[10] ChromaDB, "Chroma: The AI-native open-source embedding database," [Online]. Available: https://www.trychroma.com/