FINAL YEAR (BE) - MAJOR PROJECT PROPOSAL

A.Y. 2025-26

Title of Project :

MediVault: Intelligent Healthcare Documentation System with Encrypted Access and AI Support

---

Submitted by

Group No.	Name of Students	Roll No.
6	Swed Lengare	35
	Hrishikesh Patil	47
	Sujay Thasale	63
	Sidhik Thorat	64

Name of Project Guide: Mr. Abhijit Shete

DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

V.E.S. INSTITUTE OF TECHNOLOGY, CHEMBUR, MUMBAI-74

(An Autonomous College Affiliated to University of Mumbai, Approved by AICTE & Recognized by Govt. of Maharashtra)

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background Information: Context and Relevance of the Project

In today's healthcare ecosystem, the secure management of sensitive medical records, authorized access, and the ability to derive meaningful insights from a growing volume of healthcare documents are critical needs. Traditional Electronic Health Record (EHR) systems often lack key capabilities such as tamper-proof auditability, fine-grained access control, and efficient information retrieval, especially when maintaining patient privacy is paramount. Emerging technologies such as blockchain, AI, and large language models (LLMs) offer promising solutions to these challenges. This project leverages these technologies to develop a modern, scalable platform for secure document storage and AI-assisted interaction tailored to healthcare applications.

1.2 Problem Statement

Despite advancements in EHR systems, several persistent challenges hinder their effectiveness:

- **Security and Integrity**: Ensuring that medical records are tamper-proof and traceable over time.
- **Privacy and Access Control**: Enforcing role-based access to sensitive data in compliance with privacy laws like HIPAA and India's NDHM.
- **Scalability**: Managing large and diverse medical documents (e.g., PDFs, DICOMs) efficiently.
- **Intelligent Information Retrieval**: Enabling fast, context-aware responses to natural language queries for clinicians, students, and patients.

  Traditional systems struggle to meet all these demands simultaneously. There is a need for a hybrid architecture that guarantees data security, provides intelligent assistance, and supports compliance with healthcare standards.

1.3 Objectives

The project aims to address the above challenges by developing a secure, AI-integrated healthcare data platform with the following specific objectives:

1. **Integrate blockchain technology** to ensure verifiable tracking of medical history through immutable metadata and audit logs.
2. **Implement encrypted off-chain storage** (e.g., IPFS or cloud) for handling large medical files while maintaining data integrity.
3. **Deploy a cloud-based AI chatbot** using external LLM APIs to:
    - Summarize patient histories.
    - Respond to natural language queries regarding treatments, medications, or reports.
    - Search and retrieve relevant documents based on user input.
4. **Design and enforce a smart contract-based Role-Based Access Control (RBAC)** model to manage permissions for doctors, nurses, admins, and patients.
5. **Ensure compliance** with data protection standards such as HIPAA and NDHM.
6. **Improve efficiency and trust** in clinical workflows through intelligent document handling and secure data architecture.

2. Literature Review

2.1 Existing Research: Summary of relevant studies and technologies.

Bangera, 2025 – Proposed a **multi-tiered summarization framework** for medical evidence using **LLaMA 3.2 models**. The system operates at three levels:

- **Extractive summaries** identifying key sentences (Intervention, Outcome, Observation) with direct source hyperlinks.
- **Generative summaries** using prompt-engineered LLMs to create coherent document-level summaries.
- **Collective summaries** synthesizing insights across multiple documents for meta-analysis.

  Additionally, the framework introduces **visualization tools** for mapping evidence relationships across studies and **date filtering** for temporal refinement. Evaluations using **ROUGE and BLEU metrics** showed significant improvements over baseline summarization methods, enhancing comprehension and enabling efficient multi-document analysis for medical researchers.[1]

Yang et al., 2025 – Introduced a **multi-agent architecture for medical question answering (MedQA)** leveraging the **LLaMA3.1:70B model**. The system deploys specialized agents for:

- **Question-specific analysis** (breaking down complex queries).
- **Option analysis** (evaluating possible answers).
- **Case generation modules** producing **contextually rich, supportive clinical cases** for better interpretability.

  The model uses **zero-shot learning** on the MedQA dataset, eliminating the need for fine-tuned data while improving **accuracy and F1 scores by 7%** over existing models. The multi-agent design also improves **explainability**, making the responses more trustworthy for critical medical applications. [2]

Adams et al., 2025 – Introduced **LongHealth**, a benchmark specifically designed to evaluate **LLMs' ability to process long, real-world clinical documents**. The dataset includes **20 fictional patient cases** (ranging **5,090–6,754 words per case**), covering various diseases with discharge summaries, diagnostic reports, and lab results 666.

- **Evaluation Framework:** The benchmark features **400 multiple-choice questions** across three categories:
    - **Information retrieval** (single-hop extraction from lengthy documents).
    - **Negation handling** (identifying treatments or findings explicitly *not* present in the text).
    - **Temporal sorting** (ordering diagnostic procedures chronologically despite shuffled input).
- **Model Assessment:** Evaluated **11 open-source LLMs** with ≥16k-token context windows (e.g., **Mistral-Small-24B-Instruct-2501, LLaMA-4-Scout-17B-16E-Instruct, Mixtral-8×7B**) plus **GPT-3.5-Turbo** for comparison.
- **Results:**
    - **Task 1 (Single-patient retrieval):** Top model accuracy: **Mistral-Small-24B-Instruct-2501 (84%)** and **LLaMA-4-Scout-17B (81%)**.
    - **Task 2 (Multi-patient with distractors):** Accuracy dropped for most models (e.g., GPT-3.5-Turbo fell to **27%**), while **Mistral-Small-24B** improved to **93%**.
    - **Task 3 (Identifying missing information):** All models struggled significantly (**Mistral-Small-24B: 76%, Mixtral-8×7B: 50%**), showing a tendency toward **hallucination instead of refusal** when data was absent. [3]

Wang et al., 2024 – Developed **Joint Medical LLM and Retrieval Training (JMLR)**, a paradigm that **jointly trains retrievers and LLMs** during fine-tuning. Unlike traditional Retrieval-Augmented Generation (RAG) where retrievers and LLMs are trained separately, JMLR synchronizes them, optimizing the retriever based on how well retrieved documents improve LLM responses. Key contributions include:

- Reduction of **hallucinations** through contextually grounded retrieval.
- Enhanced **reasoning and factual accuracy** on multiple QA benchmarks (MMLU-Medical, MedQA, Amboss).
- **Improved efficiency**, with **13B JMLR models outperforming 70B Meditron models** at a fraction of computational cost (148 GPU hours vs. 42,630).

  This joint optimization provides a blueprint for building **low-cost yet high-performance** medical QA systems.[4]

Zhou & Ngo, 2024 – Designed a **two-stage biomedical information retrieval and QA pipeline** for BioASQ 2024. The system:

- Uses **LLM-driven keyword extraction and query expansion** (with few-shot prompts) to build advanced PubMed search queries.
- Performs **document reranking** using **sentence embeddings** for precise snippet extraction.
- Integrates **prompt-engineered responses** to generate structured answers for yes/no, factoid, list, and summary questions.

  Their approach achieved strong scores (e.g., **0.96 F1 for yes/no questions**) and demonstrated that **prompt-engineered retrieval combined with few-shot in-context learning** can enhance biomedical QA without extensive fine-tuning. [5]

Gaurav et al., 2023 – Showed that **LLMs can effectively summarize unstructured healthcare data** such as physician notes, diagnostic findings, and treatment histories. Their framework generated **concise, contextually accurate summaries** to support clinical decision-making. They also highlighted key challenges in adopting LLMs for healthcare:

- Maintaining **data privacy** and ensuring compliance with medical data-sharing regulations.
- Handling **domain-specific terminology** and abbreviations.
- Ensuring **faithfulness** of generated summaries to prevent factual inaccuracies.

  These findings underscore the need for robust validation, privacy-preserving mechanisms, and medical-domain adaptation when using LLMs in clinical settings. [6]

3. Methodology

3.1 Project Design:

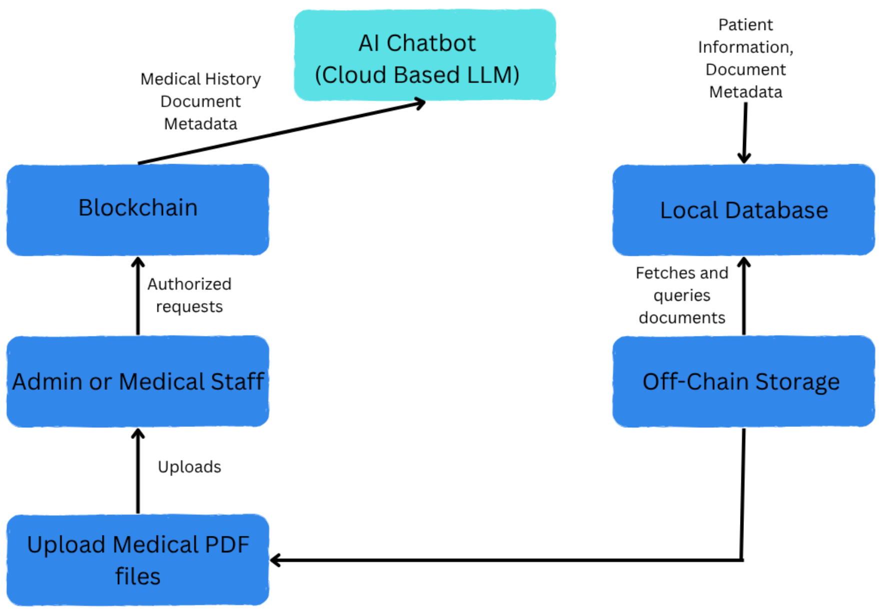

Fig 3.1. Block Diagram of the System

3.2 Method of Analysis:

This project adopts a hybrid system architecture that combines the security of blockchain (via Ganache and Truffle), scalable off-chain document storage, a local metadata database, and intelligent document interaction through external large language models (LLMs) accessed via platforms such as **OpenRouter** or **Groq**. The methodology is structured into five core phases:

1. Document Upload and Metadata Registration

- Medical staff (Admins) upload PDFs, DICOMs and other documents containing healthcare information (e.g., prescriptions, diagnoses, lab reports).
- The system extracts metadata: patient ID, document type, upload date, uploader ID.
- A unique cryptographic hash (e.g., SHA-256) is generated from the PDF content to ensure file integrity.

- The file is encrypted and stored in off-chain storage (e.g., IPFS, encrypted cloud, or local directory).
- Using **Ganache (local blockchain network)** and **Truffle (smart contract framework)**, metadata and file hash are committed to the blockchain, enabling immutability and verifiability.

2. Role-Based Access Control via Smart Contracts

- Smart contracts, developed using **Truffle**, are deployed on the **Ganache network** to define and enforce roles such as:
  - **Admin**: Can upload documents, approve/reject access requests.
  - **Doctor/Nurse**: Can request access to patient documents.
  - **Patient**: Can request their own records.
  - **Medical Students** : Can request access to patient documents.
- Wallet-based authentication is used for identity management (e.g., MetaMask).
- When a user requests document access, the smart contract:
  - Validates their role and permissions.
  - Records approval or rejection by Admin.
- Backend middleware ensures access enforcement by querying the blockchain before serving any files or data.

3. Local Metadata Storage and Indexing

- A local database (e.g., PostgreSQL(structured data) and MongoDB(unstructured data)) is used to:
  - Store metadata indexed by patient ID, document type, and upload date.
  - Track access logs, request status, and document associations.
- This DB supports rapid querying and forms the retrieval layer for the AI chatbot and dashboard.

4. AI Chatbot Integration via External LLM APIs (OpenRouter/Groq)

- Authorized users interact with an AI chatbot embedded in the UI.
- The chatbot backend:
  - Verifies user permissions via the blockchain.

    - Fetches relevant documents from secure storage and extracts their textual content (PDF parsing).
    - Prepares prompt data and securely sends it to **external LLMs via OpenRouter or Groq APIs** for processing.
- The chatbot can:
    - The doctor summarizes any types of scans then related documents should be shown there from the patient's history.
    - Answer domain-specific questions.
    - Retrieve relevant files and insights.
- Document content is never sent to the model unless the user is explicitly authorized to access it.

5. Response Delivery and System Logging

- Users receive real-time responses such as:
    - Summarized medical history
    - Exact quotes or highlights
    - Lists of documents matching criteria
- Every transaction (upload, access, query) is logged for auditing and traceability.
- Blockchain validation routines periodically confirm the integrity of stored documents by comparing current hashes to those on-chain.

3.3 List of Software: -

Software:

Table 3.1: List of Software

Sr. No.	Name of Software	Version
1	Ganache CLI or Ganache UI	v2.7.1.
2	Truffle Suite	5.11.5
3	MetaMask	12.23.1
4	Web3.js	4.16.0
5	IPFS	v1.1.2

6	Node.js	24.4.1
7	Express.js	5.1.0
8	PostgreSQL	17
9	MongoDB	8.0
10	React.js	19.1.1

4. Schedule Plan

Table 4.1: Plan of Action

Activity	July	Aug	Sept	Oct	Nov	Dec	Jan	Feb	Mar
Study of Literature									
Analysis and preparation of Proposed Scheme									
Implementation of Schemes/Model									
Result Formulation									
Report Write-up, Technical paper, Submission etc.									

5. Expected Project Outcomes

5.1 Results: Anticipated Results and Their Significance

The **MediVault** system will result in a decentralized, AI-supported healthcare documentation platform that ensures **data security, accessibility, and intelligence-driven documentation**. The key anticipated results include:

- **Decentralized and Encrypted Health Records**: Using **IPFS** for distributed storage and **Node.js crypto / libsodium** for strong encryption ensures tamper-proof, secure health records accessible only via user-controlled access (using MetaMask and Web3Modal).
- **Smart Contract Integration**: With **Ganache, Truffle Suite, and Web3.js**, we will implement Ethereum-based smart contracts to manage access rights, transaction logging, and audit trails for accountability.
- **AI-Assisted Document Understanding**: By integrating **Groq API** with **LangChain / custom Retrieval-Augmented Generation (RAG)** logic, the system will assist healthcare providers with summarizing medical documents, retrieving key insights, and improving documentation efficiency.
- **Semantic Search for Medical Records**: Using **Chroma / Pinecone / Weaviate**, users will experience vector-based semantic search, allowing context-aware document retrieval, even from large unstructured datasets.
- **Multi-Database Architecture**: With **PostgreSQL** for transactional data (e.g., access logs, metadata) and **MongoDB** for document storage, the system ensures performance, scalability, and flexibility.
- **Real-time Web Application**: Built using **React.js** (frontend), **Express.js**, and **Node.js** (backend), the system will provide a responsive, user-friendly interface with real-time interactions.
- **Integrated PDF Handling**: Using **PDF.js**, users will be able to view and interact with medical documents directly in-browser, with added annotation or AI interaction capabilities.

Significance:

This system will significantly enhance the digital healthcare experience by combining blockchain transparency, AI-powered assistance, and decentralized storage. It addresses pressing issues such as data breaches, inefficient documentation, and lack of interoperability in medical record systems.

5.2 Applications: Practical Applications of the Project

The **MediVault** system has several real-world and scalable applications:

- **Hospitals and Clinics**: To maintain encrypted, decentralized health records with smart contract-based access management and AI-powered clinical documentation assistance.
- **Decentralized Health Platforms**: Provides patient-controlled access to health records via wallet-based login (MetaMask + Web3Modal), eliminating reliance on centralized data custodians.
- **Telemedicine & Virtual Consultations**: Doctors can access and summarize patient records using AI tools, improving consultation quality and speed.
- **HealthTech Startups & Insurance Firms**: Allows secure access to patient histories, while enabling document verification and fraud prevention using blockchain logs.
- **Pharmaceutical & Research Institutions**: With vector-based search and AI summarization, researchers can find relevant cases/data points quickly from anonymized datasets.
- **Public Health Monitoring Systems**: Scalable backend and smart search help governments maintain secure, searchable EHR repositories for better decision-making.

6. References

[1] H. V. Bangera, “LLM-Powered Summarization and Hierarchical Visualization of Medical Evidence Across Documents,” M.S. thesis, Univ. of North Carolina at Greensboro, Greensboro, NC, USA, 2025.

[2] H. Yang, H. Chen, H. Guo, et al., “LLM-MedQA: Enhancing Medical Question Answering through Case Studies in Large Language Models,” unpublished, 2025.

[3] L. Adams, F. Busch, T. Han, et al., “LongHealth: A Question Answering Benchmark with Long Clinical Documents,” *J. Healthcare Informatics Research*, 2025.

[4] J. Wang, Z. Yang, Z. Yao, and H. Yu, “JMLR: Joint Medical LLM and Retrieval Training for Enhancing Reasoning and Professional Question Answering Capability,” Univ. of Massachusetts, Amherst, MA, USA, 2024.

[5] W. Zhou and T. H. Ngo, “Using Pretrained Large Language Models with Prompt Engineering to Answer Biomedical Questions,” in *CLEF BioASQ 2024 Workshop*, Georgia Inst. of Technology, Atlanta, GA, USA, 2024.

[6] G. Malode, P. Mahajan, P. Shelar, A. Sardar, and N. Dhamale, “Automated Summarization of Healthcare Records using LLM,” MET Bhujbal Knowledge City, Nashik, India, 2023.

Undertaking by Students

I have adhered to all departmental guidelines and instructions in preparing and submitting this project proposal. Under the guidance of Mr. Abhijit Shete**,** I have incorporated all feedback and suggestions received during the drafting process. This proposal complies with all ethical standards, with all sources of information properly cited. I take full responsibility for the content of this proposal and acknowledge that any errors or omissions are my own, committing to corrective action if necessary. I am dedicated to executing the proposed project as outlined and will strive to achieve its objectives to the best of my ability.

Roll No.	Name of Students	Sign
35	Swed Lengare	
47	Hrishikesh Patil	
64	Sujay Thasale	
65	Sidhik Thorat	

Project Guide Comments

Signature of Supervisor (with date)
Mr. Abhijit Shete