VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi-590 014

A Mini - Project Report

On

"MEDISCAN AI – INSTANT PATIENT HISTORY RETRIEVAL & EMERGENCY AID"

Submitted in partial fulfilment of the requirements for the **MINI PROJECT (BCD586)** course of the 5th semester

Bachelor of Engineering

In

Computer Science & Engineering (DATA SCIENCE)

Submitted by

Mr. Tarun G
(4AI23CD056)

Mr. Likeeth G Urs
(4AI23CD025)

Mr. Sai Gokul P V
(4AI23CD045)

Mr. Nikhil L U
(4AI23CD031)

Under the guidance of

Ms. HARSHITHA H D B.E., M.Tech.
Assistant Professor

Department of CS&E (DATA SCIENCE)
Adichunchanagiri Institute of Technology
CHIKKAMAGALURU - 577102

2025-26

ADHICHUNCHANAGIRI INSTITUTE OF TECHNOLOGY

Jyothinagar, Chikkamagaluru-577102

DEPARTMENT OF CS&E (DATA SCIENCE)

CERTIFICATE

This is to certify that the Mini project work entitled "**MEDISCAN AI - INSTANT PATIENT HISTORY RETRIEVAL & EMERGENCY AID**" is a Bonafide work carried out by **Mr. Tarun G (4AI23CD056), Mr. Likeeth G Urs (4AI23CD025), Mr. Sai Gokul P V (4AI23CD045), Mr. Nikhil L U (4AI23CD031)** in partial fulfillment for the **Mini Project (BCS586)** course of 5th semester Bachelor of Engineering in **Computer Science and Engineering (Data Science)** of the Visvesvaraya Technological University, Belagavi during the academic year **2025-2026**. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The Mini project report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said Degree.

Signature of the Guide
Ms. Harshitha H D B.E., M.Tech
Assistant Professor

Signature of Coordinator
Mrs. Shilpa K V. B.E., M.Tech
Assistant Professor

Signature of the HOD
Dr. Adarsh M J B.E., M.Tech., Ph.D
Associate Professor and Head

ABSTRACT

Accurate patient medical history is essential for quality healthcare delivery, timely diagnosis, and effective clinical decision-making. However, traditional methods of accessing patient data—such as manual record searches, paper-based files, and disconnected digital systems—are often time-consuming, error-prone, and poorly integrated across hospitals. To address these challenges, MediScan AI introduces an intelligent and automated solution that enables instant retrieval of patient medical history using secure digital identification.

MediScan AI is a web and mobile-based hospital information system designed for doctors, patients, and hospital authorities. The platform allows quick access to medical records using patient ID, name, or phone number, and provides complete treatment details, diagnostic information, and downloadable reports with strong data security. Doctors can add clinical notes, hospitals can upload verified records with supporting media, and patients can manage their profiles and view their health history. Built using modern technologies such as Node.js, MongoDB, React/React Native, and MedCAT for AI-based summarization, MediScan AI streamlines healthcare data management and supports improved clinical outcomes.

ACKNOWLEDGEMENTS

We express our humble pranamas to his holiness **Divine Soul Parama Poojya Jagadguru Padmabushana Sri Sri Sri Dr.Balagangadharanatha Maha Swamiji** and **Parama Poojya Jagadguru Sri Sri Sri Dr. Nirmalanandanatha Maha Swamiji Pontiff, Sri Adichunchanagiri Maha Samsthana Matt and Sri Sri Gunanatha Swamiji,** Chikkamagaluru branch, Sringeri who have showered their blessings on us.

The completion of any project involves the efforts of many people. We have been lucky enough to have received a lot of help and support from all quarters during the making of this project, so with gratitude, we take this opportunity to acknowledge all those whose guidance and encouragement helped us emerge successful.

We express our gratitude to **Dr. C K Subbaraya**, Director, Adichunchanagiri Institute of Technology.

We express our sincere thanks to our beloved principal**, Dr. C T Jayadeva** for having supported us in our academic endeavors.

We are thankful to the resourceful guidance, timely assistance and graceful gesture of our guide **Ms. Harshitha H D,** Asst. Professor, Department of CS&E (DATA SCIENCE), who has helped us in every aspect of our project work.

We thank our project coordinator **Mrs. Shilpa K V,** Asst. Professor, Department of CS&E (DATA SCIENCE), for her lively correspondence and assistance in carrying on with this project

We are also indebted to **Dr. Adarsh M J,** HOD of CS&E (DATA SCIENCE) Department, for the facilities and support extended towards us.

We would be very pleased to express our heart full thanks to all the teaching and non-teaching staff of CS&E (DATA SCIENCE) Department and our friends who have rendered their help, motivation and support.

Tarun G

Likeeth G Urs

Sai Gokul P V

Nikhil L U

CONTENTS

List of Figures

List of Tables

List of Snapshots

Chapter 1

INTRODUCTION

1.1 Background

Having efficient access to a patient's medical history is a must if you want to make a correct diagnosis and provide a safe treatment. Still, many hospitals continue to use paper-based files or incompatible digital systems, which makes it very hard for doctors to get the information they need quickly, especially if there is a crisis.

- **Context:** The patient records are sometimes left behind in the racks of the departments or are in different hospitals, saved in different ways and without standardization. The increasing number of patients makes obtaining the right history now of diagnosis increasingly difficult, which has a negative influence on the whole care quality.
- **Problem:** Present systems have trouble with slow and non-uniform record fetching. Besides that, it is hard to recognize those patients who do not have any documentation, and the issue of clinicians missing important details while manually going through long medical histories arises. Patients on their part also have very limited rights to their own records, which results in incomplete or lost medical information.
- **Opportunity:** Today AI technologies, cloud databases, and mobile platforms can be used to build a single and smart patient-history system. The usage of simple face recognition, automatic summarization, and digital record uploads will bring accessibility and accuracy to a new level. MediScan AI is doing this by combining these developments to enable instant patient history retrieval, AI-assisted insights, and secure record management for doctors, patients, and hospital authorities.

1.2 Problem Statement

- **Overview of the Problem:**

  In many healthcare settings, retrieving a patient's medical history is still a slow and fragmented process. Doctors often rely on manual searches, paper files, or disconnected digital systems, which leads to delays during diagnosis and emergency treatment. Patients struggle to maintain complete records, and hospitals lack a unified platform to store, update, and verify medical information efficiently.
- **Specific Issues:**
  - Difficulty accessing complete patient history across different hospitals and formats.
  - Time-consuming manual processes for checking previous treatments, medications, and diagnoses.
  - Inaccuracies or missing data due to inconsistent record-keeping practices.
  - Lack of real-time retrieval during critical situations such as emergencies.

- Inability to identify patients quickly when they lack documentation or are unconscious.
- Limited patient control over personal medical records and updates.
- Fragmented systems that do not support AI-based summarization or automated insights.

1.3 Objective of the System

- The primary objective of the MediScan AI System is to provide an automated, intelligent, and secure platform for instant retrieval and management of patient medical history, ensuring faster diagnosis, improved emergency response, and enhanced healthcare efficiency.

Key Goals:

- **Instant Access:** Enable doctors to quickly retrieve complete patient history using ID, phone number, name, or face recognition.
- **AI Assistance:** Provide AI-generated summaries, risk alerts, and critical insights to support accurate diagnosis.
- **Reliable Record Management:** Allow hospitals to upload verified medical records, ensuring consistent and accurate documentation.
- **Patient Empowerment:** Enable patients to view their own health history and update personal details securely.
- **Security and Privacy:** Ensure encrypted storage, secure authentication, and strict role-based access to sensitive medical data.
- **Scalability:** Support large volumes of medical records, multiple hospitals, and growing user bases without performance issues.
- **Interoperability:** Maintain a flexible system structure that can integrate with external healthcare platforms and government health systems in the future.

1.4 Significance of the System

- **Faster Decisions:** In a flash, a look at the patient's history is what makes the doctors can take quicker, more informed clinical decisions.
- **Higher Accuracy:** Digital records are free from errors, missing pieces of information, and inconsistencies that are common in manual systems.
- **Real-Time Access:** Any updated medical data can be viewed by doctors, patients, and hospitals at once, and thus, communication and coordination are enhanced.
- **AI Support:** Briefs and risk alerts that give the doctor an easy understanding of the most important points without the need for manual checking of long histories.
- **Better Patient Safety:** The cases in which the patient's allergies, medications, and past treatments are immediately visible, hence, the chances of medical errors are lowered.

- **Efficient Record Management:** Hospitals can keep and share digitally standardized records thus, the dependence on paper files is less.

1.5 Scope of the Project

- **In Scope:**
  - Web and mobile applications that can be used by doctors, patients, and hospital authorities.
  - Finding records through patient ID, name, or phone number.
  - AI module for summarization.
  - Record upload with media from the hospital side.
  - Secure authentication and role-based access.
  - Reports that can be downloaded by both doctors and hospitals.

- **Out of Scope:**
  - Fully functional Electronic medical report(EMR) integration with government systems.
  - Real-time hospital-to-hospital record transfer protocols which are beyond the scope of the project.

1.6 Methodology

- **Backend Development**
  The backend is implemented using Node.js with REST APIs and MongoDB, ensuring secure and scalable handling of medical records, authentication, and file uploads.
- **AI Module Integration**
  The system uses
  - MedCAT for medical history summarization,
  - Tesseract OCR for extracting text from scanned documents.
    These modules work with the backend to generate real-time insights.
- **Frontend Development**
  The user interfaces are built using Typescript(React.js) for web and React Native with Capacitor for mobile deployment, providing responsive and role-based dashboards for doctors, patients, and hospitals.
- **Development Model**
  An Agile approach is followed, enabling iterative development, continuous improvement, and integration of user feedback.
- **Testing**
  The system undergoes unit, integration, and functional testing to validate API performance, AI module accuracy, and overall user workflows.

1.7 Target Audience

- **Doctors:** Retrieve history, view summaries, note observations and download reports.
- **Hospital Authorities:** Upload verified medical records and view patient history.
- **Patients:** View personal history, update basic profile.

1.8 Overview of the Report

This report is organized into several chapters that explain the design, development, and evaluation of the **MediScan AI – Instant Patient History Retrieval System**. The chapters are structured as follows:

- **Chapter 1: Introduction** – Describes the background of the system, problem statement, objective, scope, methodology and overview of the system.
- **Chapter 2: System Design** – Describes the overall architecture of the system, including functional modules, technology stack, and data flow.
- **Chapter 3: Implementation** – Explains the development process, backend and frontend implementation, AI module integration, and database structure.
- **Chapter 4: Testing and Validation** – Presents the testing methods used, test cases executed, and validation of system performance.
- **Chapter 5: Results and Discussion** – Summarizes the system's output, discusses performance observations, and highlights current limitations.
- **Chapter 6: Conclusion and Future Enhancements** – Concludes the project and outlines potential improvements and expansion opportunities.

Chapter 2

SYSTEM DESIGN

This chapter describes the technical design of MediScan AI, explaining its architecture, core components, and how they interact to retrieve, manage, and process patient medical history. The design emphasizes speed, accuracy, security, and ease of use across both web and mobile platforms.

2.1 System Architecture

- **High-Level Overview:** MediScan AI follows a **client-server architecture** where users access the system through web or mobile interfaces. The backend handles authentication, record retrieval, AI summarizing, and communication with the database. The system ensures secure, real-time data flow between doctors, patients, hospitals, and AI modules.
- **Architecture Diagram:**
  As Figure 2.1 should illustrate the major components of the system:
  - Frontend (Web & Mobile UI).
  - Backend Server (Node.js with Express.js & APIs).
  - AI Processing Module (Summarization, OCR).
- **Components:**
  - **Frontend:** A user-friendly web and mobile interface built with Typescript(React.js and React Native). Doctors, patients, and hospital authorities use this interface to search records, view summaries, upload documents, and manage profiles.
  - **Backend Server:** Handles all API requests, implements business logic, manages role-based authentication, processes file uploads, and communicates with both the AI modules and the MongoDB database.
  - **AI Module:** Generates medical summaries, extracts text from medical documents, and highlights risk indicators. Works alongside backend services to provide instant insights.
  - **Database:** A MongoDB database stores patient profiles, medical records, uploaded files, AI-generated summaries. It supports scalable, flexible data storage needed for large healthcare datasets.

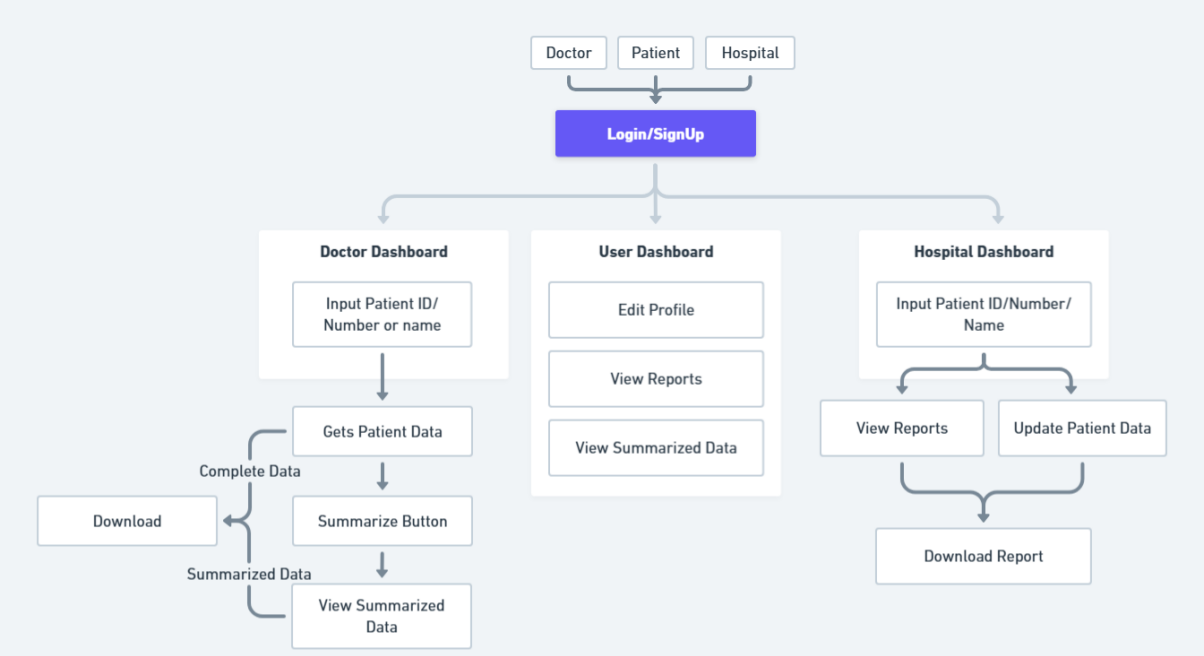

Fig 2.1: System Workflow

2.2 Module Design

The system is divided into functional modules, each responsible for a specific set of operations. These modules work together to ensure efficient, secure, and accurate retrieval and management of patient medical history.

2.2.1 User Authentication Module

Secure login via:

- Doctor ID / Govt ID for doctors
- Patient ID / phone number / name for patients
- Hospital ID for authorities

Role-based dashboards.

2.2.2 Patient Data Retrieval Module

Search by:

- Phone number
- Name
- Patient ID
- Face recognition image

Displays full medical history.

2.2.3 AI Summarization Module

AI generates:

- Summary of history
- Emergency warnings

2.2.4 Hospital Upload Module

Hospitals can upload:

- Visit reports
- Upload reports
- Upload images & PDFs
- Edit reports

Editable only for 1 hour.

2.2.5 Patient Profile Module

Patients can:

- Update personal data
- View and download history

2.3 Database Design

The database for MediScan AI is built using MongoDB, which provides a flexible and scalable structure suitable for storing diverse medical records, user profiles, and AI-generated insights. The database is organized into multiple collections, each responsible for managing a specific type of information within the system.

Collection overview:

- **Users:** Stores login information and role details for doctors, patients, and hospital authorities.
- **Medical History:** Contains patient medical records, diagnoses, treatments, and visit details.
- **Documents:** Stores uploaded medical files such as prescriptions, scans, and reports.
- **AI Summaries:** Holds AI-generated summaries of patient medical history for quick review.
- **Risk Flags:** Stores AI-identified risk alerts and emergency indicators linked to patient records.

2.4 User Interface (UI) Design

The user interface of MediScan AI is designed to be simple, intuitive, and role-specific to ensure smooth interaction for doctors, patients, and hospital authorities. Each interface is optimized for clarity, quick access to critical information, and ease of navigation across both web and mobile platforms.

Key Screens:

- **Login/Registration Screen**
  Provides secure authentication for all users, allowing access based on their designated roles.
- **Doctor Dashboard**
  Allows doctors to search for patient records using ID, phone number, or name. Displays complete medical history, AI-generated summaries, and risk indicators.
- **Patient Dashboard**
  Enables patients to view their personal medical history, and update basic information.
- **Hospital Upload Panel**
  Allows hospital authorities to upload new medical records, prescriptions, scanned documents, and treatment details, along with associated doctor and hospital information.
- **AI Summary View**
  Shows condensed medical summaries generated by the AI module, highlighting diagnoses, treatments, and important medical notes.
- **Risk Indicator Section**
  Displays color-coded alerts that represent critical health risks or emergency warnings extracted from medical history.

2.5 Technology Stack

- **Frontend:** Typescript(React.js) for the web interface and React Native with Capacitor for mobile application development.
- **Backend:** Node.js (Express), APIs, and secure server-side processing.
- **Database:** MongoDB for scalable and flexible storage of patient records, documents, and AI outputs.
- **AI Modules:** MedCAT for summarization and Tesseract OCR for document text extraction.

Chapter 3
IMPLEMENTATION

3.1 Backend Implementation

The backend is designed as a RESTful API to support scalable, role-based access.

3.1.1 Authentication

- **POST /register** for doctors, patients, hospital authorities
- **POST /login** returning JWT token and role
- Role validation: doctor, patient, hospital

3.1.2 Medical Record Retrieval

- **GET /patient/:id** fetches complete history
- **POST /search** reconstructs patient data using name, phone, or face
- **GET /records/:patientId** returns all treatment entries

3.1.3 Hospital Record Upload

- **POST /upload-record**
  - Uploads PDFs, images, prescriptions
  - Metadata includes doctor name, hospital name, date/time, treatment details
  - Editable within 1 hour only

3.1.4 AI Services

- **POST /summarize** → returns condensed medical history
- **POST /face-match** → matches patient face with stored dataset
- **POST /risk-analysis** → returns color-coded warning indicators

3.1.5 Security

- Password hashing using bcrypt
- Encrypted token-based authentication
- Protected routes using middleware

3.2 Frontend Implementation

3.2.1 Web Interface

Developed with Typescript(React.js):

- Component-based UI
- Patient search screen
- AI summary viewer
- Hospital upload panel
- Doctor dashboard

3.2.2 Mobile App

- Built using React Native + Capacitor
- Responsive UI suitable for mobile screens
- Fetches backend data using Axios
- Installed on devices as APK

3.2.3 Role-Specific Dashboards

- Doctors: search patients, view summaries, add notes and download reports
- Patients: view history, update profile
- Hospitals: view and upload medical reports of patients

3.3 AI Module Implementation

3.3.1 Medical History Summarization

- Uses MedCAT model
- Extracts keywords, diagnoses, medications
- Formats summary into:
  - Chief Complaints
  - Past Diagnoses
  - Treatments
  - Recommended Follow-up

3.3.2 OCR Processing

- Tesseract reads text from scanned PDFs/images
- Cleaned and converted into structured medical data

3.4 Database Implementation

- **MongoDB Collections**
  - **Users** → patient, doctor, hospital
  - **Medical History** → treatment records
  - **Documents** → uploaded scans/images
  - **AI Summary** → generated summaries

- **Schema Features**
  - Flexible NoSQL structure
  - References using ObjectId
  - File URLs stored as strings
  - Time-stamped entries for each medical update

Chapter 4

TESTING

This chapter describes the testing processes and methodologies applied to the MediScan AI system. Testing ensures that all modules function correctly, the system meets both functional and non-functional requirements, and it performs reliably under real-world healthcare scenarios.

4.1 Testing Objectives

- Verify that all system features operate as intended across doctors, patients, and hospital authorities.
- Ensure accurate retrieval, upload, and display of patient medical history and AI-generated outputs.
- Validate the security of user authentication, role-based access, and protection against unauthorized access.
- Test the performance of AI modules such as summarization, and OCR. Assess overall system stability, response time, and reliability during continuous usage.

4.2 Testing Environment

- **Hardware:**
  Laptop/PC with minimum 8GB RAM and a multi-core processor.

- **Software:**
  Backend: Node.js running on a local server
  Frontend: React.js / React Native
  Database: MongoDB
  Testing Tools: Postman (API testing)

- **Operating System:**
  Windows 10

- **Browsers:**
  Google Chrome, Microsoft Edge for frontend compatibility testing.

- **Mobile Testing:**
  Android Studio emulator and physical Android devices (for React Native/Capacitor app).

4.3 Types of Testing

4.3.1 Unit Testing

- **Objective:**
  To test individual backend functions, AI services, and frontend components for correctness.

- **Tools:**
  Jest/Mocha for backend logic, React Testing Library for frontend UI components.

- **Example Test Cases:**

  - **User Authentication:** Ensures login and registration logic works correctly for each role.
  - **Medical Record Retrieval:** Tests if patient records are fetched accurately using ID, name, phone number, or face recognition.
  - **AI Summary Generation:** Verifies that the AI module returns valid summaries for input medical text.
  - **File Upload Handling:** Confirms that documents are uploaded, validated, and stored correctly.

4.3.2 Integration Testing

- **Objective:**
  To validate communication between different parts of the system such as the frontend, backend, database, and AI modules.

- **Example Test Cases:**

  - **Patient Record Search:** Checks the interaction between frontend input, backend API processing, and database retrieval.
  - **Hospital Record Upload:** Ensures uploaded documents reach the database correctly and appear under the patient's history.
  - **AI Module Interaction:** Verifies backend–AI pipeline for summarization, OCR extraction, and face recognition.
  - **Role-Based Access:** Ensures correct permissions for doctors, patients, and hospital authorities.

4.3.3 Functional Testing

- **Objective:**
  To ensure the system meets all user requirements and functions correctly from an end-user perspective.

- **Test Scenarios:**

  - **Login and Registration:** Validates access for all user types and checks correct redirection to dashboards.
  - **Medical History Retrieval:** Ensures doctors can retrieve complete patient history instantly using any search method.
  - **Patient Profile Management:** Confirms that patients can view history and update personal information.
  - **Record Upload:** Tests hospital authorities' ability to upload and modify medical records within the allowed time window.
  - **Report Viewing & Download:** Ensures that users can view and download medical history or summaries without errors.

4.4 Test Cases

Below Table 4.1 shows sample test cases for different components of the MediScan AI system:

Table 4.1: Test Cases

Test Case ID	Description	Test Steps	Expected Result	Status
TC-001	Login with valid credentials	Enter valid ID/phone and password; click "Login"	User is redirected to their role-specific dashboard	Pass
TC-002	Login with invalid credentials	Enter invalid credentials; click "Login"	Error message displays, login fails	Pass
TC-003	Retrieve Patient History	Doctor enters patient ID/phone/name and searches	Complete medical history is displayed correctly	Pass
TC-004	Hospital Record Upload	Hospital uploads PDF/image with treatment details	Hospital uploads PDF/image with treatment details	Pass
TC-005	Generate AI Summary	Doctor clicks "Generate Summary" on patient history	Summary is generated and displayed accurately	Pass
TC-006	Download Medical Report	User clicks "Download" on history/summary	PDF/download file is generated successfully	Pass

Chapter 5

RESULTS AND DISCUSSION

This chapter summarizes the results of the **MediScan AI - Instant Patient History Retrieval System**, discussing its effectiveness, reliability, and alignment with the project objectives. It also highlights the challenges encountered during development and the limitations of the current system.

5.1 Results

The MediScan AI system was successfully designed and implemented with all core functionalities working as intended. The system enables doctors to retrieve patient medical history instantly using multiple identifiers such as patient ID, phone number, name, and face recognition. Hospitals can upload verified medical records along with supporting documents, while patients can securely view their own medical history and update personal details.

Project Snapshots:

- **Login and Registration Screen:** Users can securely log in as doctors, patients, or hospital authorities using role-based authentication, as shown in **Figure 5.1**.

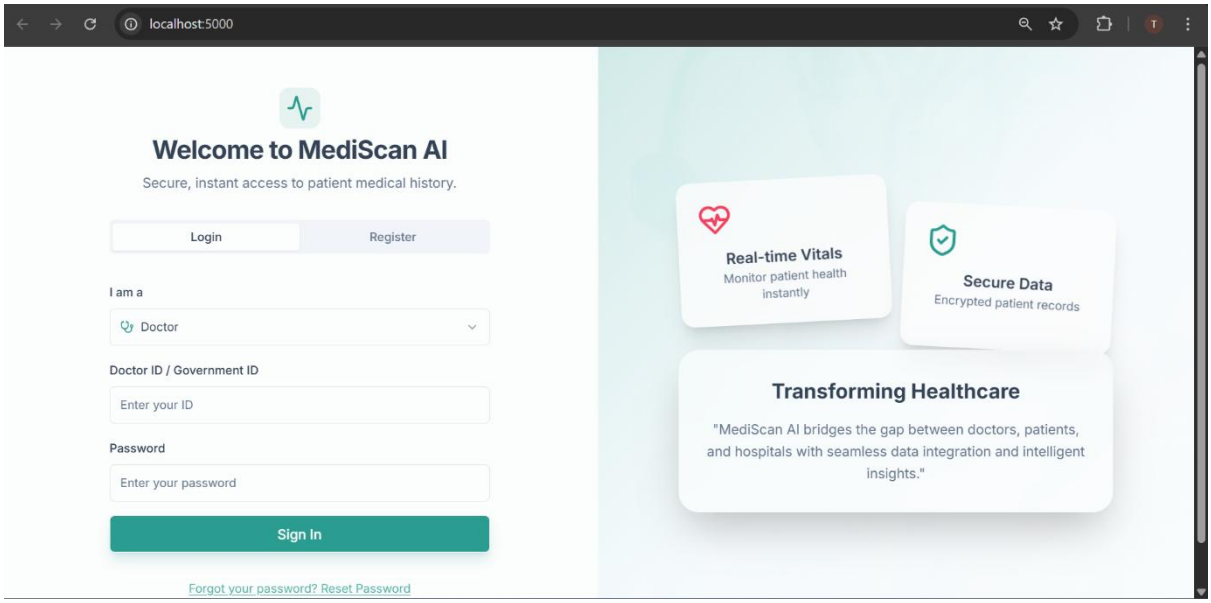

Fig 5.1: Login and Registration Screen

- **Doctor Dashboard:** Doctors can search patients, view complete medical history, access AI-generated summaries, and view emergency risk indicators, as shown in **Figure 5.2**.

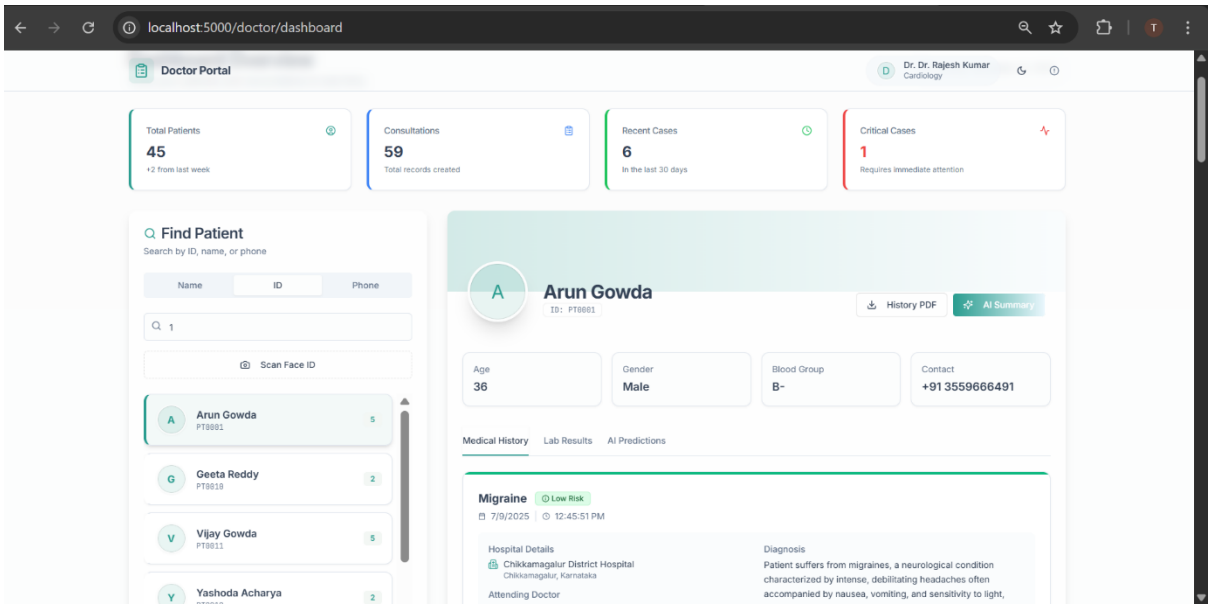

Fig 5.2: Doctor Dashboard

- **Patient Dashboard:** Patients can view treatment history, and manage personal profile information, as shown in **Figure 5.3**.

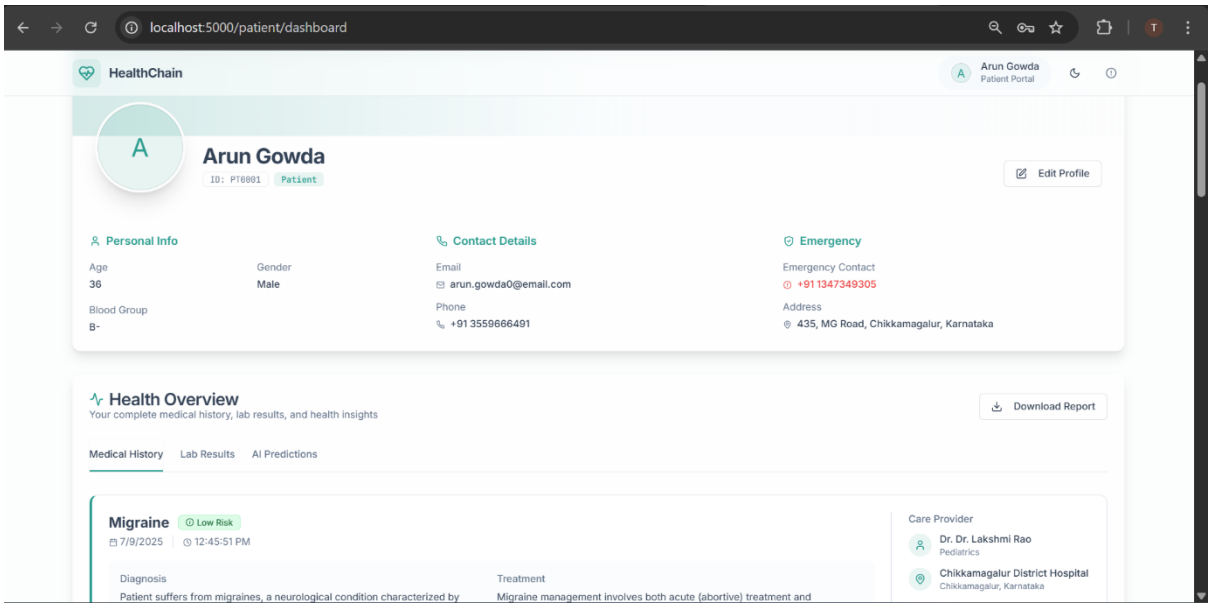

Fig 5.3: Patient Dashboard

- **Hospital Upload Panel:** Hospitals can upload prescriptions, reports, and scanned documents, with time-bound editing control, as shown in **Figure 5.4**.

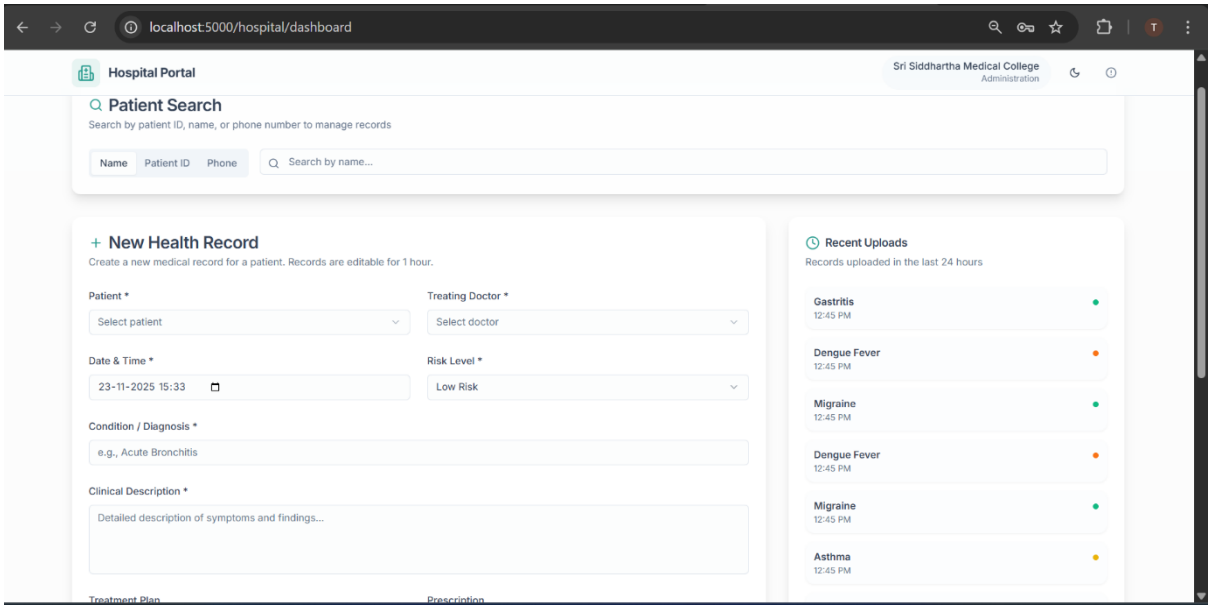

Fig 5.4: Hospital Dashboard

- **AI Summary:** The system generates concise medical summaries for quick decision-making, as shown in **Figure 5.5**.

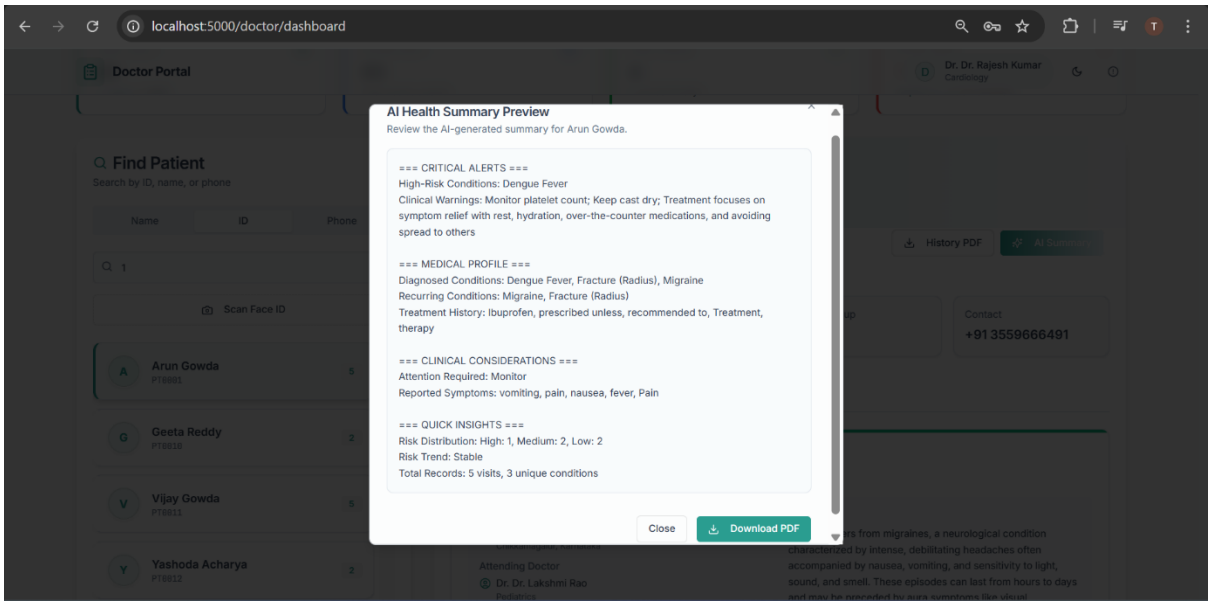

Fig 5.5: AI Summary Page

The system demonstrated fast response time in retrieving records, accurate display of uploaded data, and consistent AI output for summarization and document extraction.

5.2 Discussion

5.2.1 Effectiveness of the System:

- The MediScan AI system was successful in very little time in providing patient medical history records which was the primary objective of the system, thus it diagnosis and emergency response time was reduced drastically.
- Multi-mode patient identification (ID, phone number, and name) increased trustworthiness in patient verification.
- The use of AI summarization enabled doctors to get a quick grasp of complex medical histories without the need for manual reviewing.
- Role-based access control allowed doctors, patients, and hospitals to access only those features which were relevant to their responsibilities.
- The system exhibited efficient performance, secure data handling, and was user-friendly on both web and mobile platforms.

5.2.2 Challenges Encountered

- Integrating multiple AI modules (OCR, summarization) into a single backend pipeline required careful optimization.
- Ensuring secure handling of sensitive medical data while maintaining fast access posed implementation challenges.
- Managing uploads of large medical files such as high-resolution scans and PDFs affected performance during early testing stages.
- Network dependency affected real-time performance when backend servers were accessed over mobile networks.
- Training and fine-tuning AI models for consistent medical data interpretation required significant preprocessing effort.

5.2.3 Limitations of the Current System

- As of now, the system functions as a central platform only, meaning that it has not been connected or integrated with national or government EMR systems.
- Presently, the app only works with an active internet connection if the user wants to access the data immediately.
- Suggestions of AI-based and risk evaluation can only be made from the given medical parameters.
- Hospital-to-hospital information exchange is only possible between those hospitals that have been registered within the system.

Chapter 6

CONCLUSION AND FUTURE ENHANCEMENTS

6.1 Conclusion

- The MediScan AI - Instant Patient History Retrieval System was able to carry out its main function of giving fast, secure, and intelligent access to patient medical records. The system thus becomes a very efficient and reliable healthcare information management solution for all user roles by enabling doctors to retrieve complete medical history instantly, allowing hospitals to upload verified records, and empowering patients to view their own health data.
- In general, MediScan AI is a good example of how traditional medical record systems can benefit from digitization through AI integration, real-time access, and secure data handling. It is said to be very useful in diagnostics, thereby saving the doctors' time, in emergency situations, and in reducing the manual workload of the healthcare staff. The system, thus, carries a strong possibility to become a very instrumental healthcare institution's tool in the future.

6.2 Future Enhancements

- **Image-Based Patient Search (Face Recognition)**
    - The system can be enhanced with advanced face recognition to enable instant patient identification using live camera input or uploaded images.
    - This feature will be highly useful in emergency cases where patients are unconscious or unable to provide personal details.

- **Live Integration with Government Health ID (ABHA)**
    - Integration with the **Ayushman Bharat Health Account (ABHA)** will allow automatic retrieval of verified national health records.
    - This ensures nationwide interoperability of patient data and reduces duplication of medical tests and reports.

- **Real-Time Health Monitoring using IoT**
    - Integration with IoT-based health devices such as heart rate monitors, glucose meters, and BP sensors will enable real-time health monitoring.

- Live patient vitals can be directly linked to medical records for continuous tracking and early risk detection.

- **Multi-Lingual Medical Summarization**

  - The AI summarization module can be extended to support multiple regional and international languages.
  - This will improve accessibility for doctors and patients from diverse linguistic backgrounds.

- **AI-Based Prescription Generator and Medical Chatbot**

  - An AI-powered prescription generation module can assist doctors in creating accurate prescriptions based on diagnosis and patient history.
  - A medical chatbot can provide patients with basic health guidance, medication reminders, and post-treatment care instructions.

REFERENCES

- Government of India. (2024). ***Ayushman Bharat Digital Mission (ABDM) & ABHA*** *documentation*. Retrieved from https://abdm.gov.in
- MedCAT Researchers. (2024). ***Medical Concept Annotation Toolkit (MedCAT)***. Retrieved from https://github.com/CogStack/MedCAT
- MongoDB Inc. (2024). ***MongoDB*** *official documentation*. Retrieved from https://www.mongodb.com/docs
- Meta Platforms Inc. (2024). ***React Native*** *documentation*. Retrieved from https://reactnative.dev
- Node.js Foundation. (2024). ***Node.js*** *official documentation*. Retrieved from https://nodejs.org/en/docs
- React Team. (2024). ***React.js*** *official documentation*. Retrieved from https://react.dev
- **World Health Organization**. (2020). *Electronic health records: Manual for developing countries*. WHO Press.
- Smith, R. (2007). *An overview of the* ***Tesseract OCR*** *engine*. Proceedings of the Ninth International Conference on Document Analysis and Recognition, IEEE.