

HealthFlow Connect

A Project Report

Submitted by

Ayush Ambhorkar 112103016

Vipul Chaudhari 112103028

Aditya Choudhary 112103030

in fulfilment

for

**Third Year B.Tech (Computer Engineering)/
(Software Engineering Mini Project)**

Under the guidance of

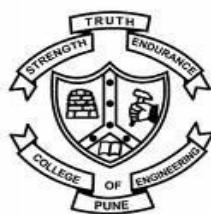
Asst. Prof. Tanuja Pattanshetti

College of Engineering, Pune

AND

Dr. Sanjay Bhalchandra Chaudhari

Navi Mumbai Municipal Corporation's Vashi hospital



**DEPARTMENT OF COMPUTER ENGINEERING
COEP Technological University**

April, 2024

DEPARTMENT OF COMPUTER ENGINEERING
COEP Technological University

CERTIFICATE

Certified that this project, titled “HealthFlow Connect” has been successfully completed by

Ayush Ambhorkar 112103016

Vipul Chaudhari 112103028

Aditya Choudhary 112103030

and is approved for the fulfilment of the requirements for the degree of “Third Year
B.Tech (Computer Engineering) Software Engineering Mini Project
”.

SIGNATURE

Tanuja Pattanshetti

Project Guide

Department of Computer Engineering

COEP Technological University,

Shivajinagar, Pune - 5.

SIGNATURE

Dr P. K. Deshmukh

Head

Department of Computer Engineering

COEP Technological University,

Shivajinagar, Pune - 5.

Abstract

The HealthFlow Connect project aims to digitize the case paper system used in government hospitals in India. This project leverages the MERN (MongoDB, Express.js, React.js, Node.js) stack to develop a comprehensive web-based platform for managing patient information and medical records efficiently. The system caters to various user roles including admin, doctor, counter staff, lab technician, pharmacist, and patients, each with specific functionalities tailored to their roles.

The admin, typically a doctor overseeing hospital administration, has the responsibility of managing user profiles, including doctors, pharmacists, counter staff, and lab technicians. They have overall access to user information and system settings. Doctors utilize the platform to diagnose patients, record patient complaints, conduct examinations, prescribe medications, and maintain comprehensive patient histories for effective diagnosis and treatment planning. Counter staff handle patient registration, record updates, case paper creation, and printing, ensuring seamless patient management.

Lab technicians manage assigned lab tests and upload test reports, while pharmacists oversee medication requirements and dispensation, all of which are seamlessly integrated into the patient's medical history. Patients have access to their medical records, allowing them to view case papers, diagnoses, prescribed medications, and lab test reports.

The HealthFlow Connect project not only streamlines the patient management process but also enhances accessibility, accuracy, and efficiency in healthcare delivery. This abstract provides an overview of the project's objectives, functionalities, and technologies employed, highlighting its significance in modernizing healthcare systems.

Contents

List of Tables	ii
List of Figures	v
1 Synopsis	1
1.1 Project Title	1
1.2 Internal Guide	1
1.3 Problem Statement	1
1.4 Goals and Objectives	2
1.4.1 Goals:	2
1.4.2 Objectives:	3
1.5 Plan of Project Execution	3
2 Problem Definition and scope	6
2.1 Problem Definition	6
2.2 Statement of Scope	7
2.3 Software Context	8
2.3.1 Technologies Used	8
2.3.2 Development Methodology	9
2.3.3 Software Architecture	10

2.3.4	Conclusion	10
2.4	Major Constraints	10
2.5	Outcomes	11
2.6	Application	12
3	Project Plan	14
4	Software Requirement Specification	15
4.1	Use-cases	15
4.1.1	Admin Management	15
4.1.2	Doctor Interaction	15
4.1.3	Counter Operations	16
4.1.4	Lab Technician Tasks	16
4.1.5	Pharmacist Functions	16
4.2	Use Case View	18
4.2.1	Admin Staff UCD	18
4.2.2	Patient Counter UCD	19
4.2.3	Patient Doctor UCD	20
4.2.4	Patient Lab Assistant UCD	21
4.3	Data Model and Description	22
4.3.1	Data Description	22
4.3.2	Entity relationship diagram	24
4.4	Data Flow Diagrams	25
4.4.1	Level 0 DFD	25
4.4.2	Level 1 DFD	26
4.4.3	Level 2 DFD	27

4.5	Description of functions	28
4.5.1	Non Functional Requirements	28
5	Detailed Design Document	30
5.1	Diagrams	30
5.1.1	Class Diagram	30
5.1.2	Sequence Diagram	31
5.1.3	Componenet Diagram	32
5.1.4	Deployment Diagram	32
5.1.5	Traceability Matrix	33
5.2	Project Pages	35
5.2.1	Login Page(Landing Page)	35
5.2.2	Admin Home Page	35
5.2.3	Admin Add Counter	36
5.2.4	Admin Add Doctor	36
5.2.5	Admin Add Lab Technician	37
5.2.6	Admin Add Pharmacist	37
5.2.7	Admin Add Lab Technician	38
5.2.8	Admin Add Pharmacist	38
5.2.9	Admin Update Staff	39
5.2.10	Admin Delete Staff	39
5.2.11	Admin Update Staff	40
5.2.12	Admin Delete Staff	40
5.2.13	Admin All Staff Records	41
5.2.14	OTP Verification	41
5.2.15	Doctor Home Page	42

5.2.16	Doctor Serve Page	42
5.2.17	Pharmacist Home Page	43
5.2.18	Pharmacist Serve Page	43
5.2.19	Lab Technician Home Page	44
5.2.20	Lab Technician Serve Page	44
5.2.21	Lab Technician Home Page	45
5.2.22	Lab Technician Serve Page	45
5.2.23	Counter Home Page	46
5.2.24	Counter Add Patient Page	46
5.2.25	Counter Update Patient Page	47
5.2.26	Counter Redirect Patient Page	47
5.2.27	Counter Print Case Paper Page	48
5.2.28	Counter View All Patient Page	49
5.3	A few Code Snippets	50
5.3.1	Database Connect	50
5.3.2	Server.js	51
5.3.3	App.jsx	52
5.4	Final Case Paper	53
6	Summary and Conclusion	54
6.1	Summary	54
6.2	Conclusion	54

List of Tables

1.1	Project Task Schedule	5
4.1	Database Entities	23

List of Figures

3.1	Project Schedule (Gantt Chart)	14
4.1	Admin Staff UCD	18
4.2	Patient Counter UCD	19
4.3	Patient Doctor UCD	20
4.4	Patient Lab Assistant UCD	21
4.5	Entity relationship diagram	24
4.6	Level 0 DFD	25
4.7	Level 1 DFD	26
4.8	Level 2 DFD	27
4.9	Activity Diagram	28
5.1	Class Diagram	30
5.2	Sequence Diagram	31
5.3	Component Diagram	32
5.4	Deployment Diagram	32
5.5	Traceability Matrix	34
5.6	Login Page(Landing Page)	35
5.7	Admin Home Page	35
5.8	Admin Add Counter	36

5.9 Admin Add Doctor	36
5.10 Admin Add Lab Technician	37
5.11 Admin Add Pharmacist	37
5.12 Admin Add Lab Technician	38
5.13 Admin Add Pharmacist	38
5.14 Admin Update Staff	39
5.15 Admin Delete Staff	39
5.16 Admin Update Staff	40
5.17 Admin Delete Staff	40
5.18 Admin All Staff Records	41
5.19 OTP Verification	41
5.20 Doctor Home Page	42
5.21 Doctor Serve Page	42
5.22 Pharmacist Home Page	43
5.23 Pharmacist Serve Page	43
5.24 Lab Technician Home Page	44
5.25 Lab Technician Serve Page	44
5.26 Lab Technician Home Page	45
5.27 Lab Technician Serve Page	45
5.28 Counter Home Page	46
5.29 Counter Add Patient Page	46
5.30 Counter Update Patient Page	47
5.31 Counter Redirect Patient Page	47
5.32 Counter Print Case Paper Page	48
5.33 Counter View All Patient Page	49

5.34 Database Connect	50
5.35 Server.js	51
5.36 App.jsx	52
5.37 Case paper	53

Chapter 1

Synopsis

1.1 Project Title

HealthFlow Connect

1.2 Internal Guide

Asst. Prof. Tanuja Pattanshetti

1.3 Problem Statement

The current paper-based case paper system used in government hospitals in India poses significant challenges in terms of accessibility, efficiency, and accuracy. Healthcare providers struggle with manual data entry processes, limited accessibility to patient records, and fragmented systems for managing patient information. These challenges hinder effective patient management and healthcare delivery, leading to delays, errors, and inefficiencies.

To address these issues, the HealthFlow Connect project aims to digitize the case paper system, leveraging modern technologies to create a comprehensive digital platform for managing patient information and medical records. The project seeks to streamline

patient management processes, enhance accessibility to medical records, improve data accuracy, and ensure compliance with data security and privacy regulations.

Addressing these challenges requires the development of a robust, scalable, and user-friendly digital platform that can effectively digitize the case paper system used in government hospitals. The HealthFlow Connect project seeks to tackle these issues by leveraging modern technologies and best practices in software development to create a comprehensive solution that streamlines patient management, enhances accessibility to medical records, improves data accuracy, and ensures compliance with data security and privacy regulations.

1.4 Goals and Objectives

The goals and objectives of the HealthFlow Connect project are as follows:

1.4.1 Goals:

1. **Digitize Patient Records:** Develop a digital platform to replace the paper-based case paper system, enabling the storage and management of patient records in electronic format.
2. **Streamline Patient Management:** Create a centralized system for patient registration, record updates, case paper generation, and retrieval, to improve efficiency and accuracy in healthcare delivery.
3. **Enhance Accessibility:** Provide healthcare providers with remote access to patient records, facilitating timely decision-making and improving overall patient care.
4. **Ensure Data Security and Privacy:** Implement robust security measures to safeguard patient data, ensuring compliance with relevant regulations and standards.

5. Integrate Healthcare Services: Integrate functionalities for doctors, counter staff, lab technicians, pharmacists, and patients into a cohesive platform, promoting collaboration and coordination in healthcare delivery.

1.4.2 Objectives:

1. Develop a user-friendly web-based interface for accessing and managing patient records.
2. Create role-based access control mechanisms to ensure appropriate access to patient records based on user roles.
3. Implement features for patient registration, record updates, case paper generation, and printing.
4. Integrate modules for doctors to input diagnoses, prescriptions, and medical histories seamlessly.
5. Provide functionalities for lab technicians to upload test reports and pharmacists to manage medication records.

1.5 Plan of Project Execution

Task	Category	Assigned to	Completed By	Start	Days
Initiation					
Characterizing the Issue Domain	High Risk	ALL	ALL	20-01-2024	3
Articulating the Problem Description	Med Risk	Vipul Chaudhari	Vipul Chaudhari	23-01-2024	2
Pinpointing Dependencies	Milestone	Aditya Choudhary	Aditya Choudhary	25-01-2024	3
Project Boundaries	Milestone	Ayush Ambhorkar	Ayush Ambhorkar	26-01-2024	3
Specifying Operational Requirements	High Risk	Aditya Choudhary	Aditya Choudhary	29-01-2024	2
Recognizing Limiting Factors		Vipul Chaudhari	Vipul Chaudhari	29-01-2024	2
Planning and Design					
Gantt Chart Composition	Med Risk	Ayush Ambhorkar	Ayush Ambhorkar	31-01-2024	2
Process Model Recognition	High Risk	Aditya Choudhary, Ayush Ambhorkar	Aditya Choudhary, Ayush Ambhorkar	02-02-2024	1
Tech Infrastructure Identification	High Risk	Aditya Choudhary, Vipul Chaudhari	Aditya Choudhary, Vipul Chaudhari	03-02-2024	3
Software Requirement Specification	High Risk	Vipul Chaudhari, Ayush Ambhorkar	Vipul Chaudhari, Ayush Ambhorkar	05-02-2024	5
Data Flow Diagram	Low Risk	Aditya Choudhary, Vipul Chaudhari	Aditya Choudhary, Vipul Chaudhari	10-02-2024	2
ER Diagram	Low Risk	Ayush Ambhorkar	Ayush Ambhorkar	10-02-2024	2
UML	Med Risk			12-02-2024	5

Backend Development					
Preliminary Configuration	Low Risk		17-02-2024	2	
Crafting Routing Structures and Controllers	High Risk		19-02-2024	15	
Debugging Tool for Evaluation (Postman)	Milestone		05-03-2024	3	
Frontend Development					
Preliminary Arrangements	Low Risk		08-03-2024	2	
Establishing Communication between Backend and Frontend	High Risk		09-03-2024	3	
Designing the Foundation of the Web Page	Med Risk		11-03-2024	10	
Upgrading Design Elements Using Bootstrap	Low Risk		21-03-2024	5	
Integration of Complete Web Structure	Goal		26-03-2024	7	
Testing					
On site real time testing	Med Risk		02-04-2024	3	
Feedback analysis	On Track		03-04-2024	2	
Doing necessary changes	On Track		05-04-2024	6	

Table 1.1: Project Task Schedule

Chapter 2

Problem Definition and scope

2.1 Problem Definition

The traditional paper-based case paper system employed in government hospitals across India presents several challenges in terms of accessibility, efficiency, and accuracy. These challenges include:

1. **Manual Data Entry:** The process of recording patient information, medical histories, diagnoses, prescriptions, and lab test results is labor-intensive and prone to errors, leading to inaccuracies in patient records.
2. **Limited Accessibility:** Patient records stored in physical files are often inaccessible remotely, hindering healthcare providers' ability to access and update patient information efficiently.
3. **Inefficient Patient Management:** The absence of a centralized system for managing patient records results in inefficiencies in patient registration, record updates, and case paper generation, leading to delays and confusion in healthcare delivery.
4. **Data Security and Privacy Concerns:** Paper-based records are susceptible to loss, theft, or unauthorized access, compromising patient data security and privacy

compliance.

5. **Lack of Integration:** Fragmented systems for managing patient information, such as separate records for diagnosis, prescriptions, and lab test results, lead to disjointed workflows and hinder comprehensive patient care.

To address these challenges, the HealthFlow Connect project aims to digitize the case paper system used in government hospitals in India. By leveraging modern technologies and best practices in software development, the project seeks to create a comprehensive digital platform for managing patient information and medical records effectively. The digital platform will streamline patient management processes, enhance accessibility to medical records, improve data accuracy, and ensure compliance with data security and privacy regulations.

2.2 Statement of Scope

The scope of the HealthFlow Connect project encompasses the following key aspects:

1. **Digitization of Patient Records:** The project aims to digitize the existing paper-based case paper system used in government hospitals in India. This involves transitioning patient records, medical histories, diagnoses, prescriptions, and lab test results from paper format to a digital platform.
2. **User Roles and Functionalities:** The digital platform will support various user roles, including administrators, doctors, counter staff, lab technicians, pharmacists, and patients. Each user role will have specific functionalities tailored to their responsibilities in the healthcare process.
3. **Centralized Management System:** The project seeks to create a centralized system for managing patient information and medical records. This includes function-

alities for patient registration, record updates, case paper generation, prescription management, lab test result tracking, and medication dispensation.

4. **Integration and Accessibility:** The digital platform will integrate seamlessly with existing healthcare systems and promote accessibility to medical records. Healthcare providers will be able to access patient information securely from any location, facilitating timely decision-making and improving patient care.
5. **Data Security and Privacy:** Ensuring the security and privacy of patient data is a priority for the project. Robust security measures will be implemented to protect patient information from unauthorized access, data breaches, and other security threats.
6. **Scalability and Flexibility:** The digital platform will be designed to accommodate future enhancements and scalability. It will be flexible enough to adapt to evolving healthcare needs and regulatory requirements.

The Statement of Scope outlines the objectives and boundaries of the HealthFlow Connect project, providing a clear understanding of its aims and functionalities.

2.3 Software Context

The software context section provides an overview of the technologies, frameworks, and methodologies used in the development of the HealthFlow Connect project.

2.3.1 Technologies Used

- **Backend Development:**

- Node.js: JavaScript runtime environment for server-side development.

- Express.js: Minimalist web framework for Node.js, facilitating server-side development.
- MongoDB: NoSQL database for storing patient data in a JSON-like format.
- GridFS: GridFS is a specification for storing and retrieving large files in MongoDB. It works by splitting large files into smaller chunks, which are then stored as separate documents in two collections: files and chunks. The GridFSBucket class in Mongoose provides an interface for working with GridFS in MongoDB.
- Amazon SNS: Fully managed Pub/Sub service for A2A and A2P messaging.
Used in Project for OTP message and verification.

- **Frontend Development:**

- HTML: Markup language for creating web pages and applications.
- CSS: Stylesheet language for styling HTML elements.
- JavaScript: Programming language for adding interactivity and dynamic behavior to web pages.
- React.js: JavaScript library for building reusable UI components.

- **Communication Between Frontend and Backend:**

- REST APIs: Standardized way for communication between client and server using HTTP methods.
- Postman: Tool for testing and debugging APIs.

2.3.2 Development Methodology

The project follows an iterative development approach, allowing for incremental enhancements and regular feedback gathering from stakeholders. Agile principles are incorporated

to promote flexibility, collaboration, and continuous improvement throughout the development lifecycle.

2.3.3 Software Architecture

HealthFlow Connect is designed using a modular architecture, separating concerns between frontend and backend components. The backend is responsible for data storage, business logic, and API endpoints, while the frontend handles user interfaces and interactions. This architecture promotes scalability, maintainability, and reusability of components.

2.3.4 Conclusion

In summary, the software context of HealthFlow Connect outlines the technologies, methodologies, and architecture employed in its development. By leveraging modern web technologies and following agile principles, the project aims to deliver a robust and user-friendly solution for digitizing healthcare records in government hospitals.

2.4 Major Constraints

- 1. Regulatory Compliance:** Healthcare software systems must comply with various regulations and standards related to patient data privacy and security, such as HIPAA in the United States or similar regulations in India.
- 2. Data Security:** Patient data stored in the system must be securely managed to prevent unauthorized access, data breaches, and data loss.
- 3. Interoperability:** The system may need to integrate with existing healthcare systems and standards to ensure interoperability with other healthcare providers, laboratories, pharmacies, and government agencies.

4. Scalability: The system should be able to handle a large volume of patient data and user traffic, especially in government hospitals where patient flow can be high.

5. Usability and Accessibility: The system should be user-friendly and accessible to healthcare professionals with varying levels of technical expertise.

6. Infrastructure Requirements: The availability and reliability of infrastructure components such as servers, network infrastructure, and cloud services can impact the system's performance and uptime.

By addressing these constraints effectively, we aim to mitigate risks and ensure the successful development and implementation of the HealthFlow Connect system.

2.5 Outcomes

The outcomes of the HealthFlow Connect project are summarized as follows:

1. Digitization of Case Papers: Successfully digitized the case papers used in government hospitals in India, eliminating the need for manual paper-based records.

2. Improved Efficiency: Streamlined the process of managing patient records, diagnosis, prescriptions, lab tests, and medication dispensation, leading to improved efficiency in healthcare delivery.

3. Enhanced Accessibility: Made healthcare information more accessible to healthcare professionals, patients, and other stakeholders, facilitating better decision-making and coordination of care.

4. Data Security and Compliance: Implemented robust security measures to protect patient data and ensure compliance with regulatory requirements such as HIPAA in the United States or similar regulations in India.

5. Interoperability: Designed the system to integrate with existing healthcare systems and standards, enabling interoperability with other healthcare providers, laboratories, pharmacies, and government agencies.

6. Scalability and Performance: Developed a scalable and high-performance system capable of handling a large volume of patient data and user traffic, especially in government hospitals with high patient flow.

7. Cost Savings: Helped reduce costs associated with paper-based record-keeping, manual data entry, and administrative overhead, resulting in cost savings for government hospitals.

Overall, the outcomes of the HealthFlow Connect project have significantly contributed to improving healthcare delivery, enhancing patient care, and modernizing healthcare operations in government hospitals in India.

2.6 Application

The HealthFlow Connect project has various applications in the healthcare sector, benefiting healthcare professionals, patients, and healthcare organizations:

1. Improved Patient Care: Enables healthcare professionals to access comprehensive patient records, streamline diagnosis, and provide personalized treatment plans, leading to improved patient care outcomes.

2. Efficient Healthcare Operations: Automates administrative tasks, reduces paperwork, and minimizes manual data entry errors, resulting in more efficient healthcare operations and resource utilization.

3. Enhanced Collaboration: Facilitates communication and collaboration among

healthcare professionals, enabling seamless coordination of care and interdisciplinary teamwork.

- 4. Patient Empowerment:** Empowers patients to actively participate in their healthcare journey by providing access to their medical records, treatment plans, and educational resources.
- 5. Data-Driven Insights:** Generates valuable insights from aggregated healthcare data, enabling healthcare organizations to identify trends, patterns, and areas for improvement in patient care delivery.
- 6. Remote Healthcare Delivery:** Supports remote healthcare delivery models by enabling telemedicine consultations, remote monitoring of patient health metrics, and electronic prescription management.
- 7. Public Health Management:** Contributes to public health management efforts by facilitating disease surveillance, outbreak detection, and monitoring of population health indicators.
- 8. Research and Education:** Provides a rich source of anonymized healthcare data for research purposes and supports medical education and training through case-based learning and virtual patient simulations.

Overall, the application of the HealthFlow Connect project extends across various aspects of healthcare delivery, administration, research, and education, ultimately leading to improved health outcomes and patient experiences.

Chapter 3

Project Plan

PROJECT: HealthFlow Connect						
T1 Batch						
112103016, 112103028, 112103030						
Task	Category	Assigned to	Completed By	Progress	Start	Days
Initiation						
Characterizing the Issue Domain	High Risk	ALL	ALL	100%	1/20/2024	3
Articulating the Problem Description	Med Risk	Vipul Chaudhari	Vipul Chaudhari	100%	1/23/2024	2
Pinpointing Dependencies	Milestone	Aditya Choudhary	Aditya Choudhary	100%	1/25/2024	3
Project Boundaries	Milestone	Ayush Amborkar	Ayush Amborkar	100%	1/26/2024	3
Specifying Operational Requirements	High Risk	Aditya Choudhary	Aditya Choudhary	100%	1/29/2024	2
Recognizing Limiting Factors		Vipul Chaudhari	Vipul Chaudhari	100%	1/29/2024	2
Planning and Design						
Gantt Chart Composition	Med Risk	Ayush Amborkar	Ayush Amborkar	100%	1/31/2024	2
Process Model Recognition	High Risk	Aditya Choudhary, Ayush Amborkar	Aditya Choudhary, Ayush Amborkar	100%	2/2/2024	1
Tech Infrastructure Identification	High Risk	Aditya Choudhary, Vipul Chaudhari	Aditya Choudhary, Vipul Chaudhari	100%	2/3/2024	3
Software Requirement Specification	High Risk	Vipul Chaudhari, Ayush Amborkar	Vipul Chaudhari, Ayush Amborkar	100%	2/5/2024	5
Data Flow Diagram	Low Risk	Aditya Choudhary, Vipul Chaudhari	Aditya Choudhary, Vipul Chaudhari	100%	2/10/2024	2
ER Diagram	Low Risk	Ayush Amborkar	Ayush Amborkar	100%	2/10/2024	2
UML	Med Risk			100%	2/12/2024	5
Backend Development				100%		
Preliminary Configuration	Low Risk			100%	2/17/2024	2
Crafting Routing Structures and Controllers	High Risk			100%	2/19/2024	15
Debugging Tool for Evaluation (Postman)	Milestone			100%	3/5/2024	3
Frontend Development				100%		
Preliminary Arrangements	Low Risk			100%	3/8/2024	2
Establishing Communication between Backend and Frontend	High Risk			100%	3/9/2024	3
Designing the Foundation of the Web Page	Med Risk			100%	3/11/2024	10
Upgrading Design Elements Using Bootstrap	Low Risk			100%	3/21/2024	5
Integration of Complete Web Structure	Goal			100%	3/26/2024	7
Testing				100%		
On site real time testing	Med Risk			100%	4/2/2024	3
Feedback analysis	On Track			100%	4/3/2024	2
Doing necessary changes	On Track			100%	4/5/2024	6

Figure 3.1: Project Schedule (Gantt Chart)

Chapter 4

Software Requirement Specification

4.1 Use-cases

The following use cases illustrate the functionality and interactions of the HealthFlow Connect system:

4.1.1 Admin Management

UC1. Manage Users: The admin can create, update, and delete user accounts for doctors, pharmacists, counters, and lab technicians.

UC2. Generate Reports: The admin can generate reports on user activity, patient data, and system performance for analysis and decision-making.

4.1.2 Doctor Interaction

UC1. Diagnose Patient: The doctor can input patient complaints, conduct general examinations, prescribe medications, and order lab tests for diagnosis.

UC2. Access Patient History: The doctor can access the entire medical history of a patient, including past diagnoses, treatments, and medications prescribed.

UC3. Update Case Paper: The doctor updates the patient's case paper with new information during each visit or consultation.

4.1.3 Counter Operations

UC1. Create New Patient Record: The counter staff creates a new patient record, including demographic information and initial symptoms.

UC2. Update Patient Record: The counter staff updates existing patient records with any changes in personal information or medical history.

UC3. Print Case Paper: The counter staff prints the patient's case paper for reference during consultations or referrals.

4.1.4 Lab Technician Tasks

UC1. View Lab Tests: The lab technician views the list of lab tests to be performed for a patient.

UC2. Enter Test Results: The lab technician enters and uploads test results in the form of documents or images.

UC3. Update Patient Records: The lab technician updates the patient's case paper with the completed lab test results.

4.1.5 Pharmacist Functions

UC1. View Medications: The pharmacist views the list of medications prescribed to a patient.

UC2. Dispense Medications: The pharmacist dispenses medications to patients and updates their records accordingly.

These use cases demonstrate the various interactions and functionalities of the Health-Flow Connect system, catering to the needs of different user roles and facilitating seamless healthcare management and delivery.

4.2 Use Case View

4.2.1 Admin Staff UCD

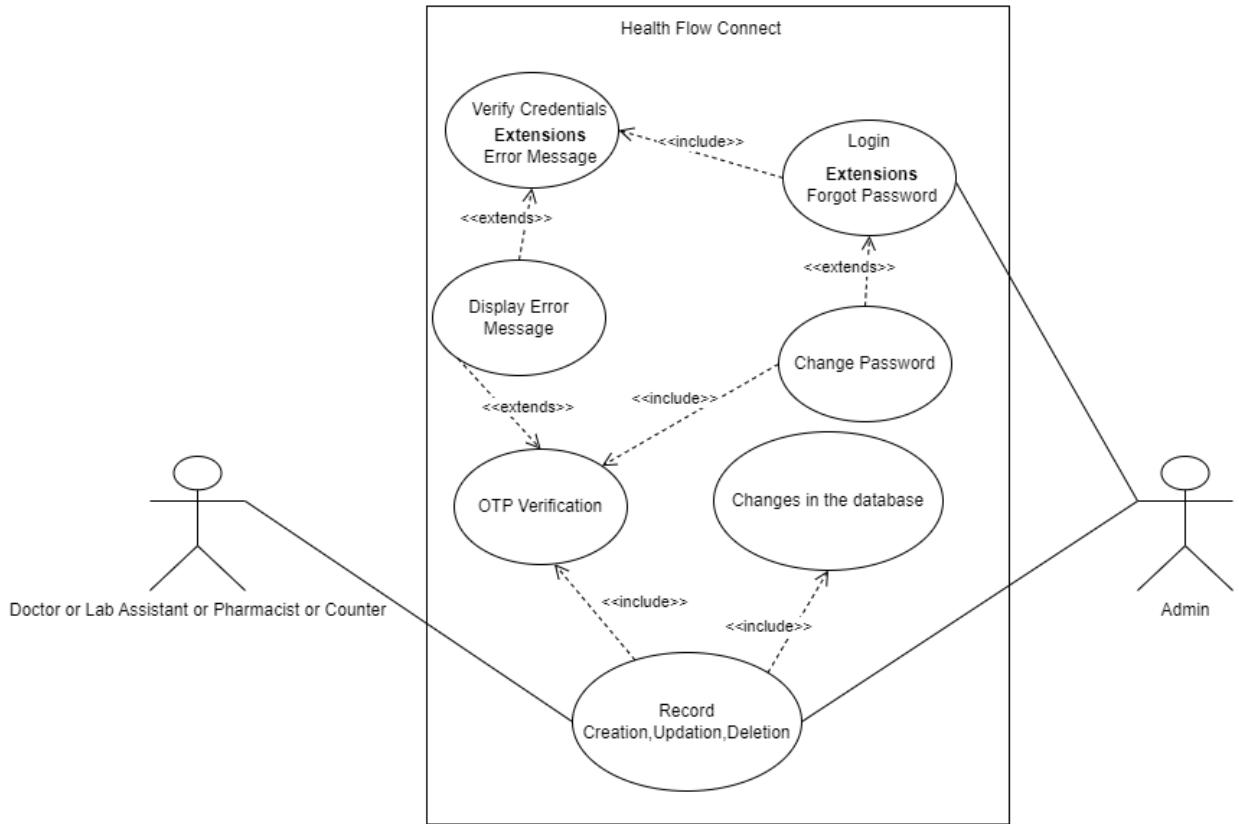


Figure 4.1: Admin Staff UCD

4.2.2 Patient Counter UCD

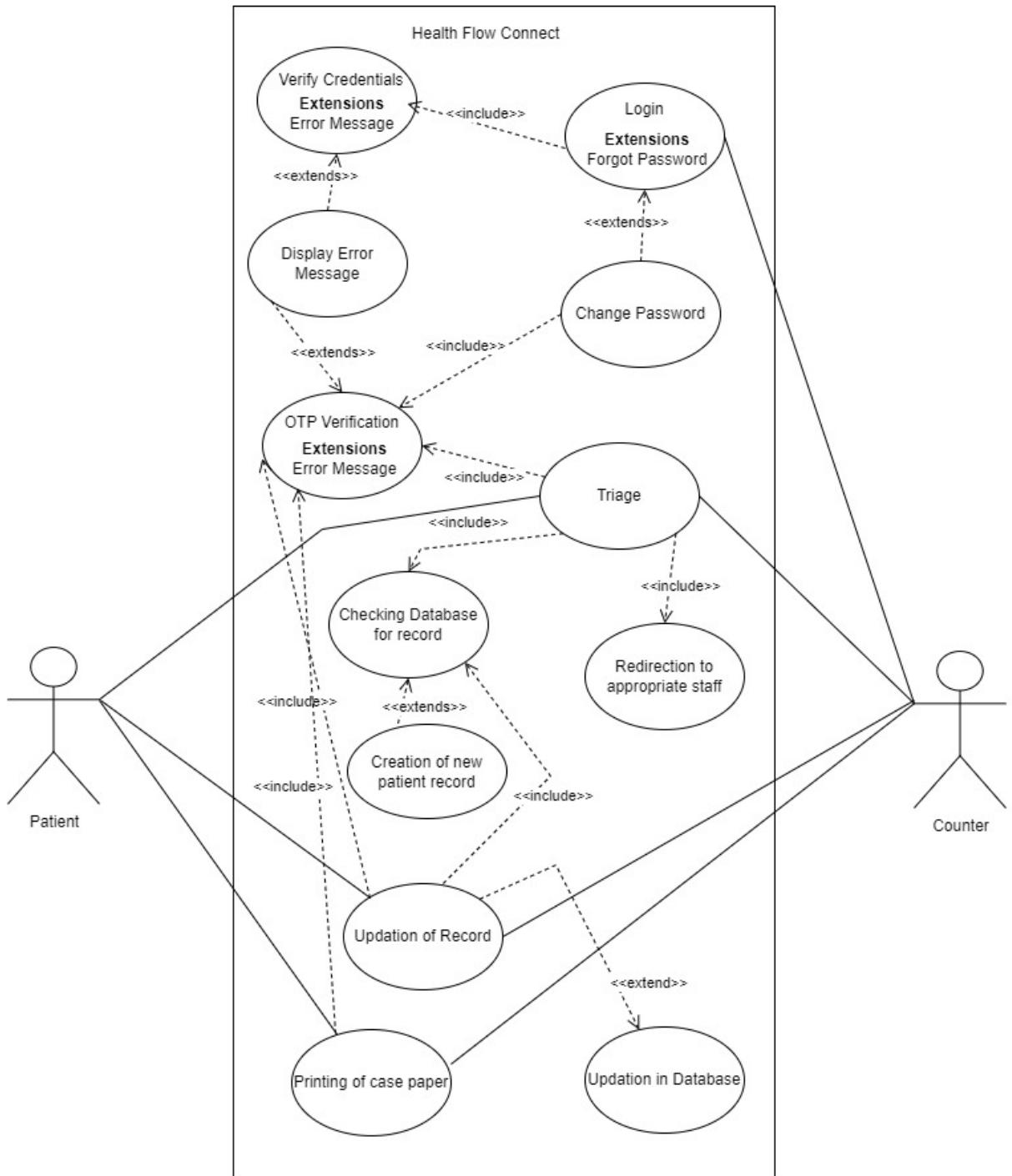


Figure 4.2: Patient Counter UCD

4.2.3 Patient Doctor UCD

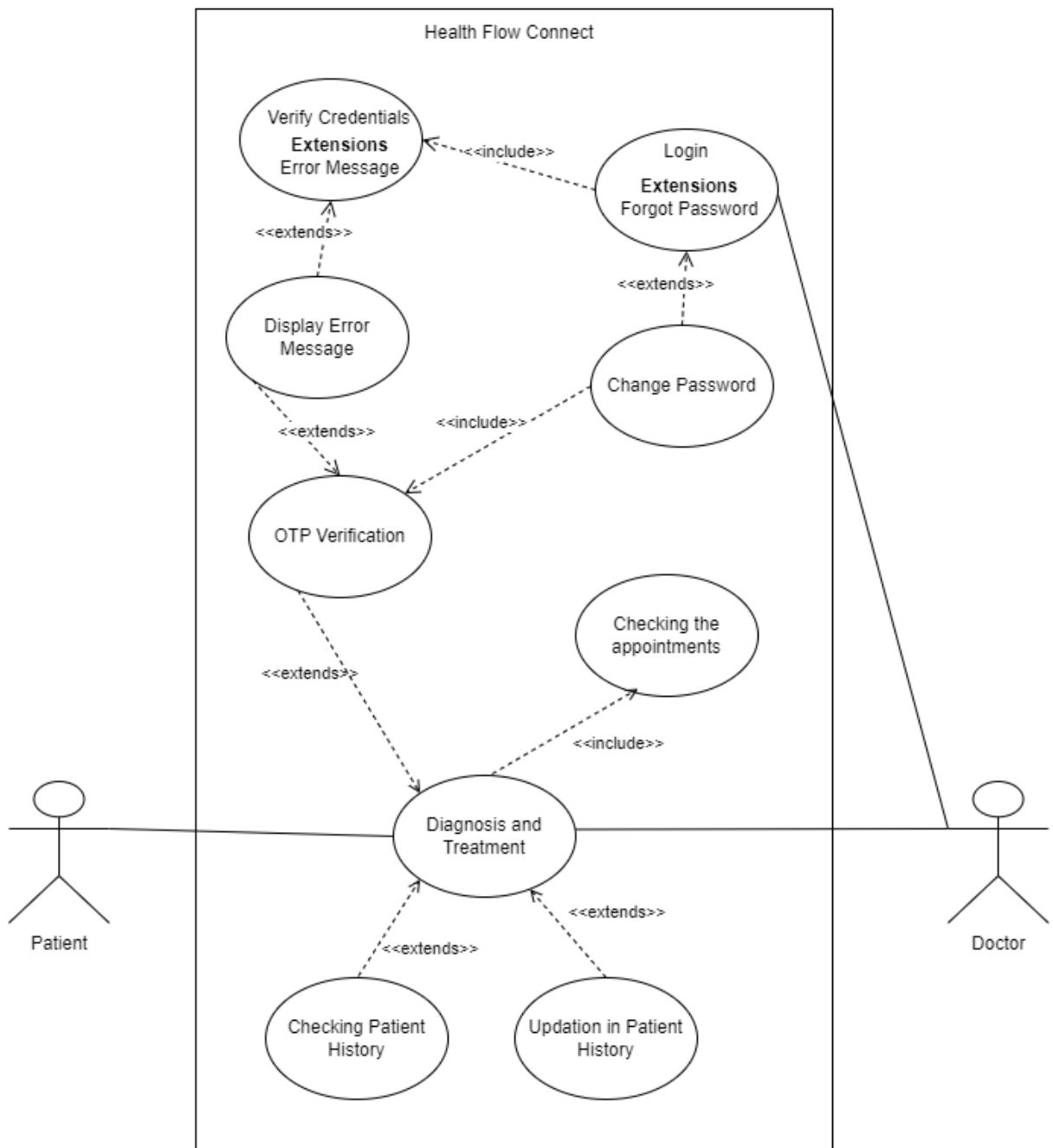


Figure 4.3: Patient Doctor UCD

4.2.4 Patient Lab Assistant UCD

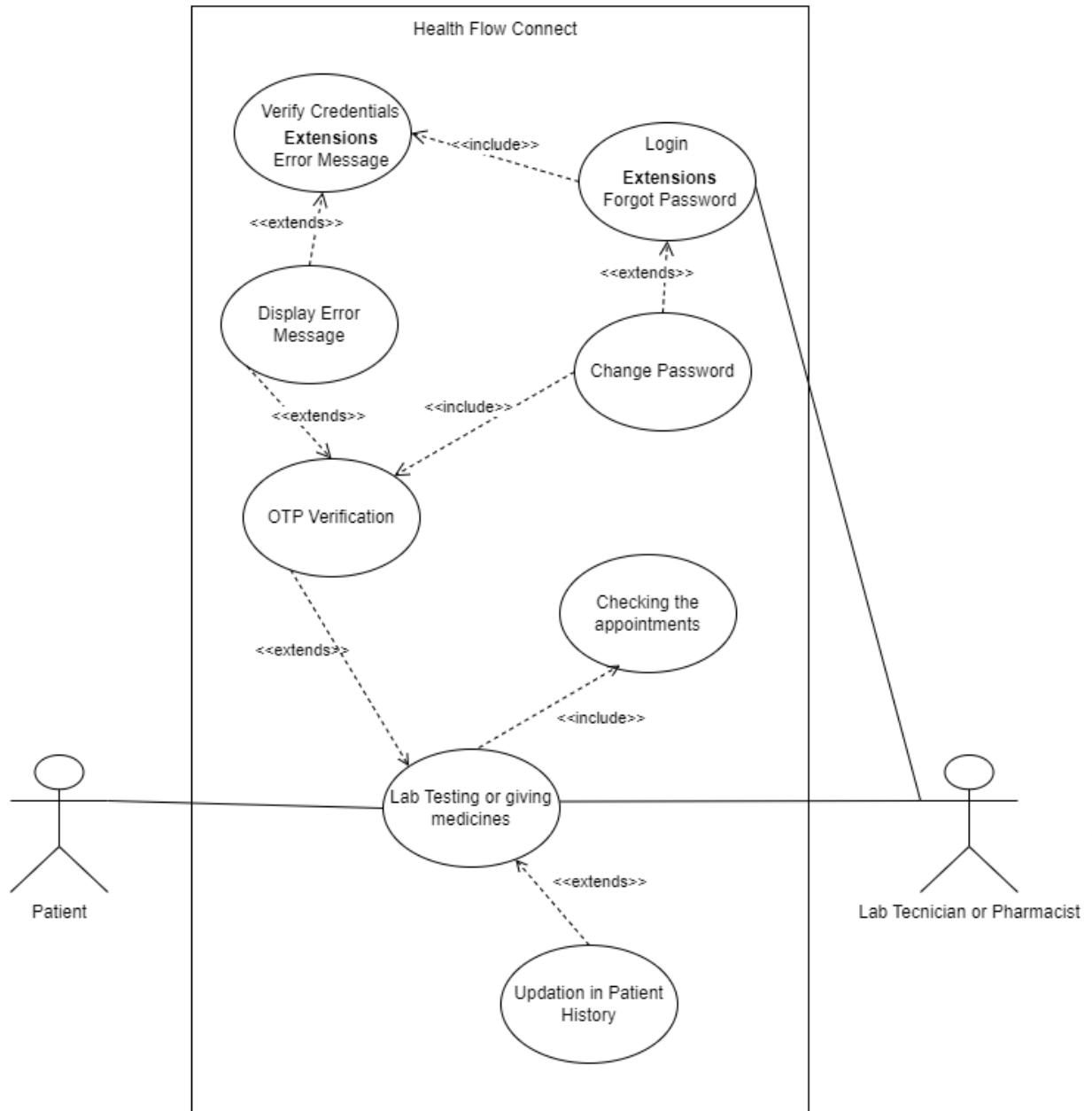


Figure 4.4: Patient Lab Assistant UCD

4.3 Data Model and Description

4.3.1 Data Description

Entity	Attributes	Description
Users	username, password, designation	Individuals with access to the system
Doctors	person ID, OPD, degrees, specializations, admin status	Medical practitioners
Patients	UID, name, age, gender, address, contact number, medical history	Individuals receiving medical treatment
Case Papers	patient UID, history ID array, start date/time, end date/time, active status	Medical records for patients
History Records	patient UID, user UID, user role, date/time, complaints, examination details, lab tests, medications, extra notes	Detailed medical history for patients

Lab Technicians	person ID, degrees, specializations, lab type	Staff responsible for laboratory testing
Pharmacists	person ID, degrees, specializations	Staff responsible for medication dispensing
Redirection	patient UID, user UID,	Tracks patient redirections between staff members
Records	user role, redirection creation date/time, redirection served date/time, redirection status, respective history ID, lab testings to be done, medicines to be given	
Persons	UID, first name, middle name, last name, gender, date of birth, phone number, address, role	Individuals involved in the healthcare process

Table 4.1: Database Entities

4.3.2 Entity relationship diagram

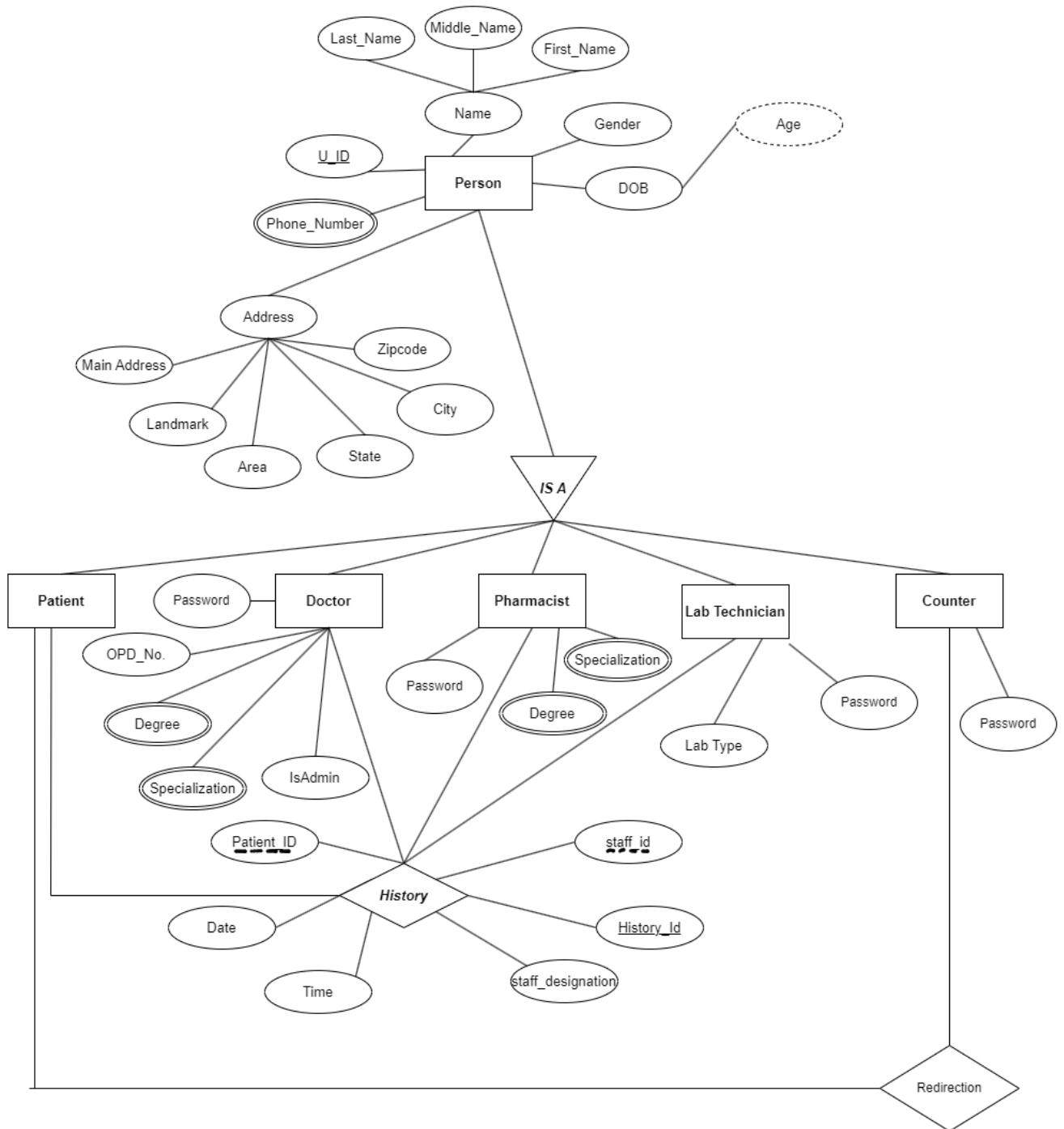


Figure 4.5: Entity relationship diagram

4.4 Data Flow Diagrams

4.4.1 Level 0 DFD

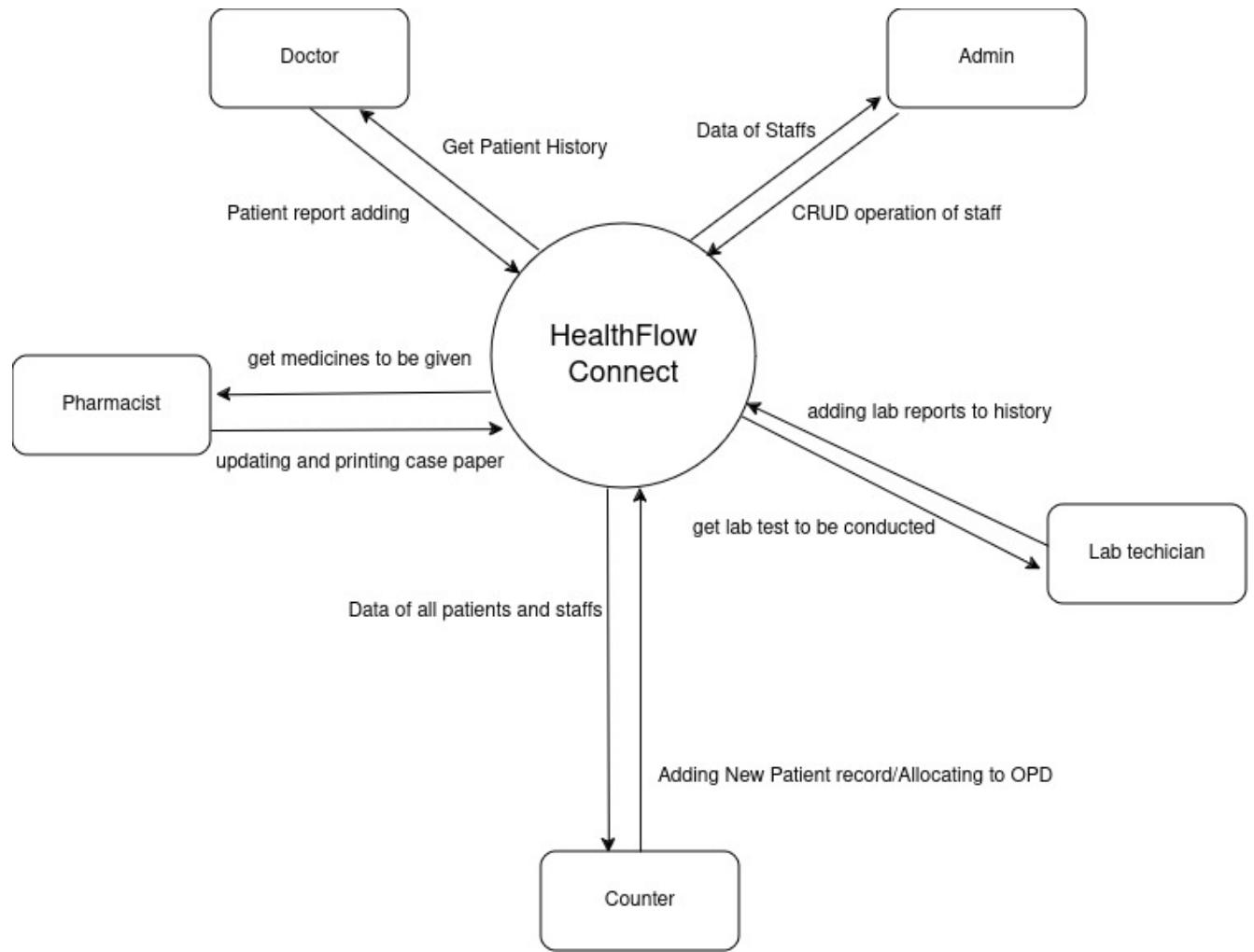


Figure 4.6: Level 0 DFD

4.4.2 Level 1 DFD

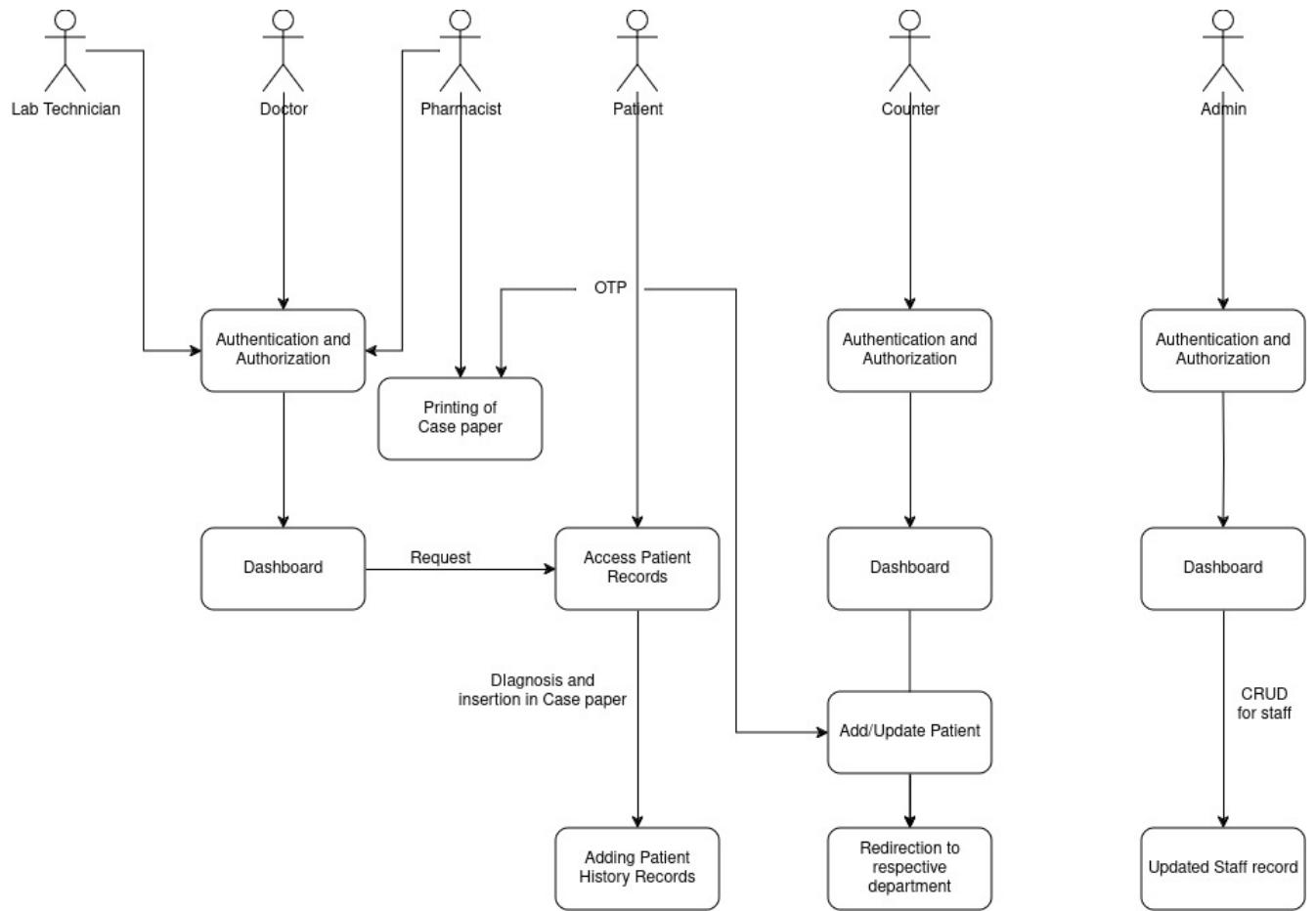


Figure 4.7: Level 1 DFD

4.4.3 Level 2 DFD

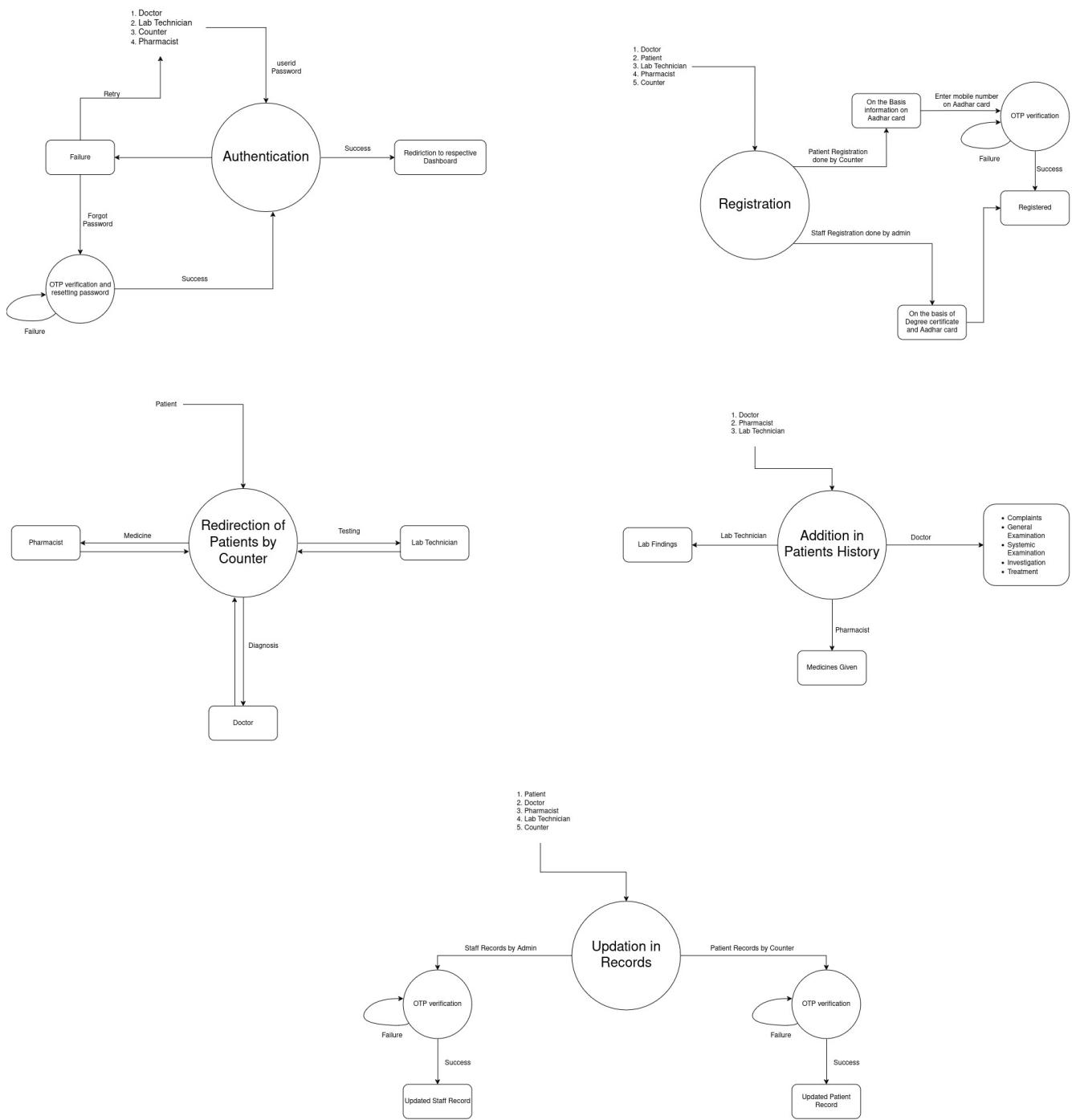


Figure 4.8: Level 2 DFD

4.5 Description of functions

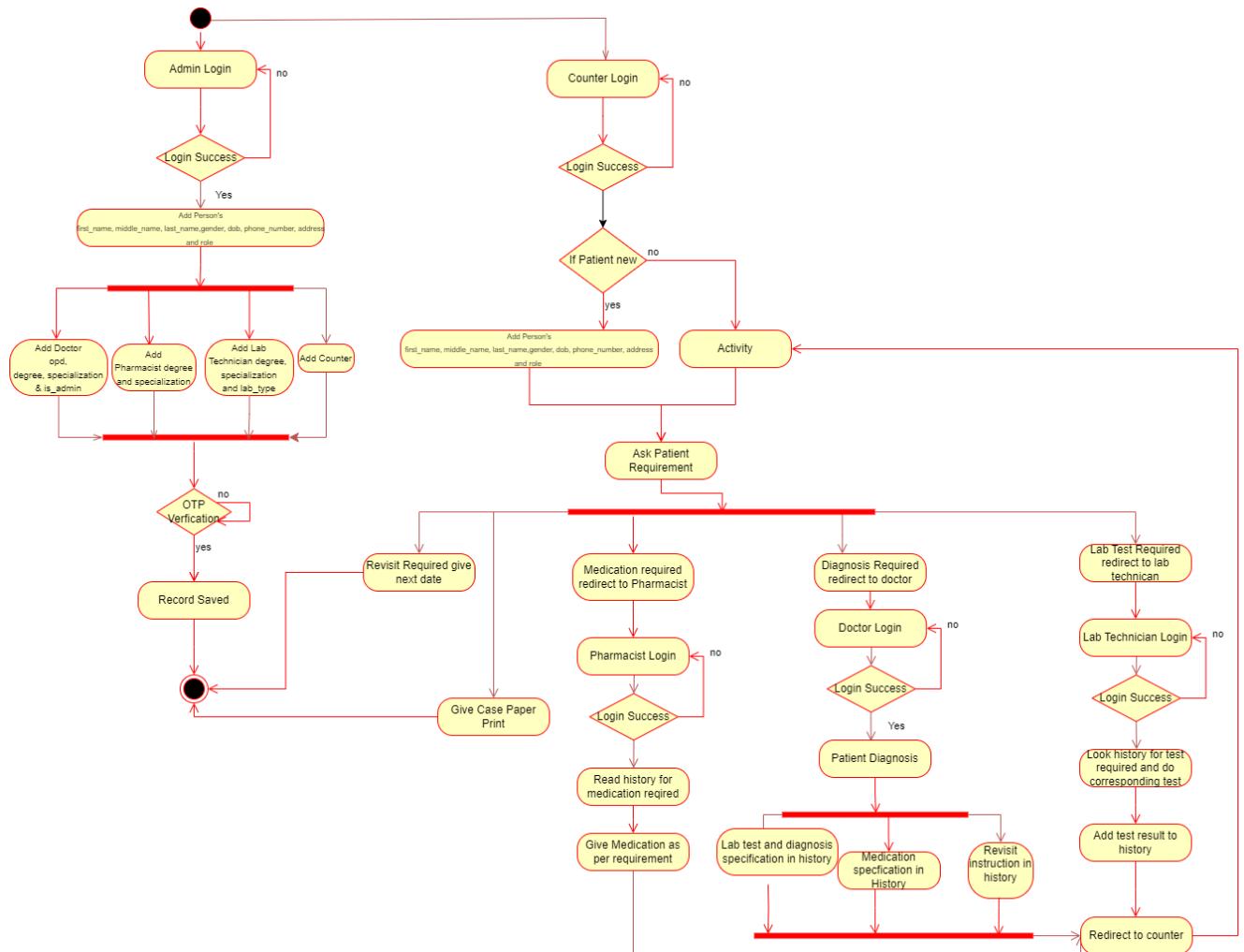


Figure 4.9: Activity Diagram

4.5.1 Non Functional Requirements

NF1. Performance:

- The system should respond to user interactions within a few seconds.
 - Reports generation should be completed within a few minutes for datasets of size huge amount of records.

NF2. Reliability:

- The system should have an uptime of high as possible.
- Reports should be accurate and reliable, with less error rate.

NF3. Scalability:

- The system should be able to handle a growing user base without significant degradation in performance.
- It should support adding additional hardware resources to accommodate increased load.

NF4. Security:

- Access to sensitive information should be restricted based on user roles and permissions.
- The system should implement measures to prevent unauthorized access, such as authentication and authorization mechanisms.

NF5. Maintainability

- The codebase should be well-documented and follow coding standards.
- It should be easy to modify or extend the system to accommodate future requirements.

Chapter 5

Detailed Design Document

5.1 Diagrams

5.1.1 Class Diagram

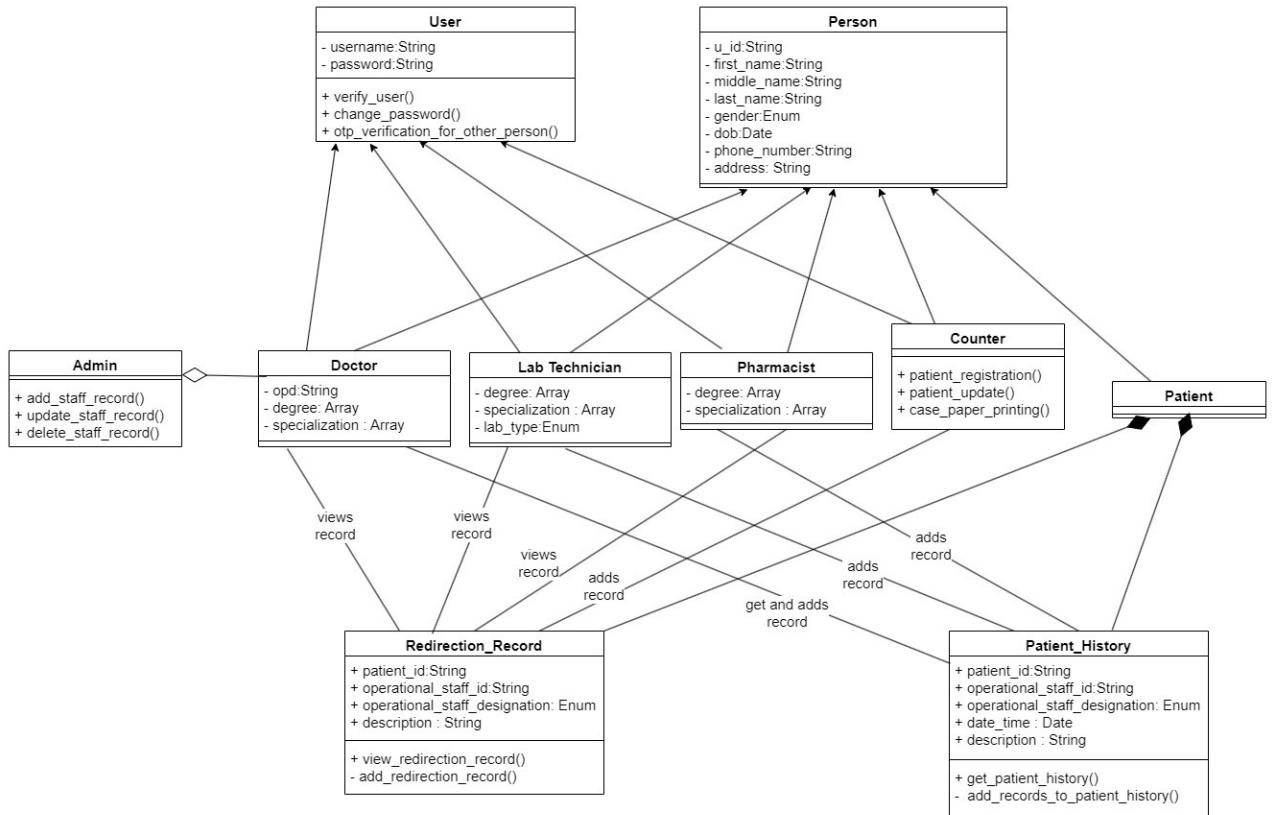


Figure 5.1: Class Diagram

5.1.2 Sequence Diagram

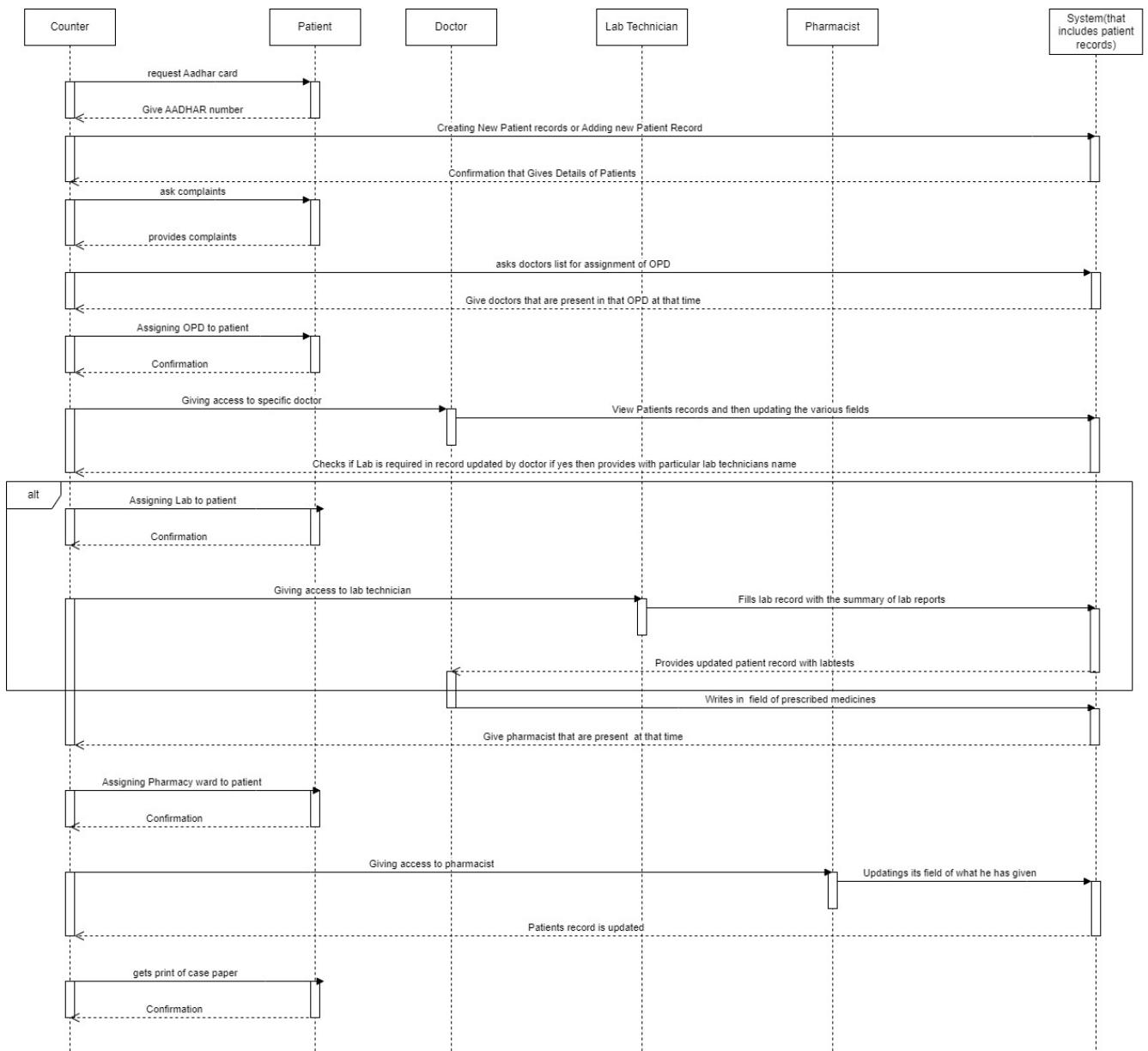


Figure 5.2: Sequence Diagram

5.1.3 Componenet Diagram

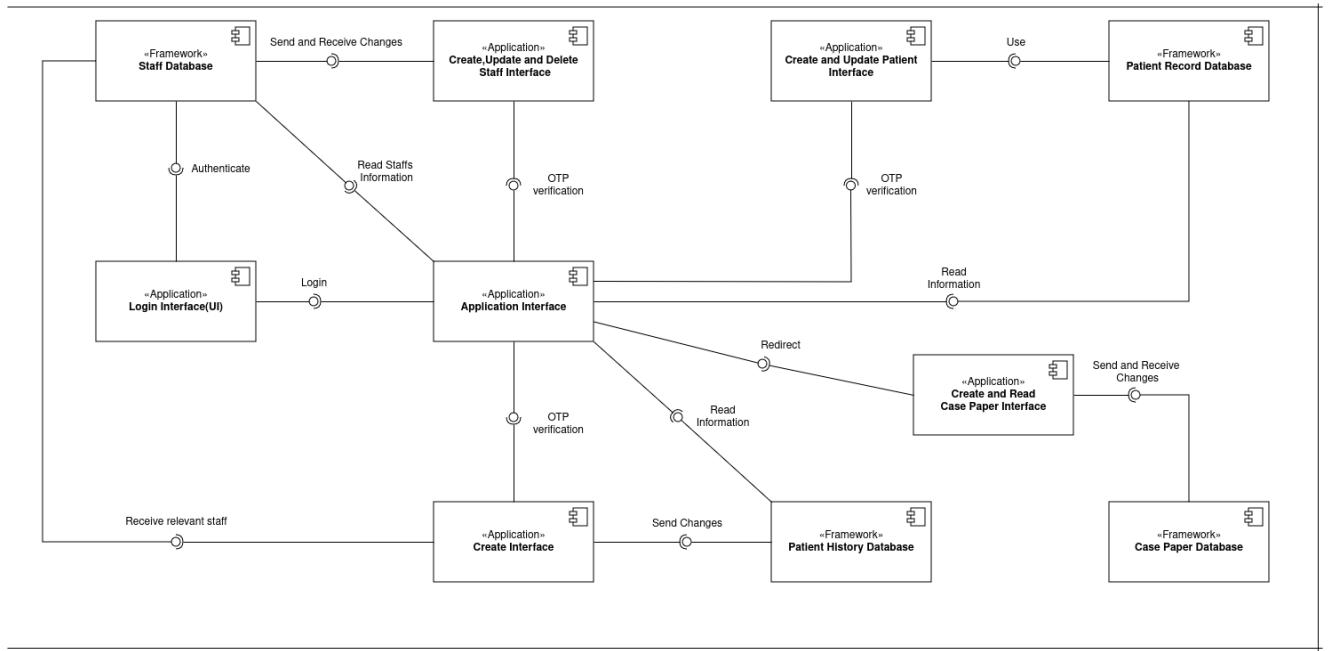


Figure 5.3: Component Diagram

5.1.4 Deployment Diagram

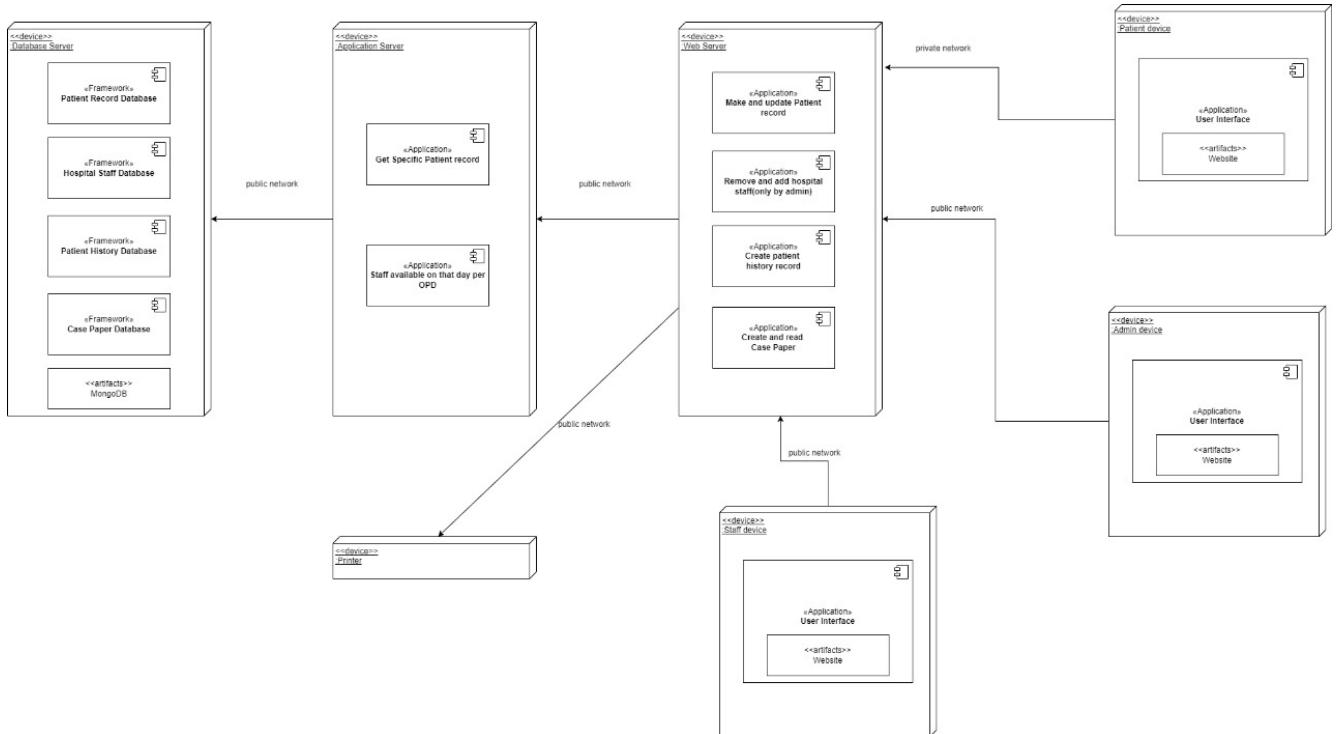


Figure 5.4: Deployment Diagram

5.1.5 Traceability Matrix

Project Name – HealthFlow Connect					
Business Requirements Document		Functional Requirements Document			Test Case Document
Business Requirement ID	Business Requirements/Business Use Case	Functional Requirement ID	Functional Requirement/Use Case	Priority	Test Case ID
BR_1	Patient Registration and Record Management	FR_1	The system should allow for efficient registration of new patients.	High	TC_1
		FR_2	The system should provide easy access to existing patient records.	High	TC_2
BR_2	Assessment and OPD Assignment	FR_3	Counter personnel should have access to a system for assessing patients' conditions.	High	TC_3
		FR_4	The counter should assist in assigning patients to appropriate OPDs based on their conditions.	Med	TC_4
BR_3	OPD Management	FR_5	The system should facilitate communication between counter personnel and OPD staff.	High	TC_5
		FR_6	OPD staff should be able to access patient records and update them as necessary.	High	TC_6
BR_4	Doctor's Examination and Diagnosis	FR_7	The system should ensure patients are directed to the correct OPD based on their medical needs.	High	TC_7
		FR_8	Doctors should have access to a system for conducting thorough examinations.	High	TC_8
BR_5	Lab Testing and Reporting	FR_9	The system should support doctors in making accurate diagnoses and updating patient records.	High	TC_9
		FR_10	The system should facilitate communication between doctors and other healthcare professionals.	High	TC_10
BR_6	Treatment Planning and Medication Management	FR_11	There should be a system for requesting lab tests and tracking their status.	High	TC_11
		FR_12	Lab personnel should be able to conduct tests efficiently and report results accurately.	High	TC_12
BR_7	Admission Process and Inpatient Care	FR_13	The system should integrate lab findings into patient records automatically.	High	TC_13
		FR_14	Pharmacy staff should have access to a system for managing medication dispensing.	High	TC_14
BR_8	Comprehensive Patient Records	FR_15	There should be a system for managing the admission process efficiently.	High	TC_15
		FR_16	The system should facilitate the monitoring of inpatient care and treatment progress.	High	TC_16
BR_9	Discharge and Follow-up	FR_17	Comprehensive records should be maintained for admitted patients.	High	TC_17

BR_8	Pharmacy Management	FR_18	The system should support pharmacy staff in dispensing medications accurately.	High	TC_18
		FR_19	Prescription records should be maintained securely.	High	TC_19
BR_9	Case Paper and Admit Counter Paper Generation	FR_20	There should be a system for generating printed case papers for patients.	High	TC_20
		FR_21	The system should ensure accuracy and completeness of generated papers.	High	TC_21

Figure 5.5: Traceability Matrix

5.2 Project Pages

5.2.1 Login Page(Landing Page)

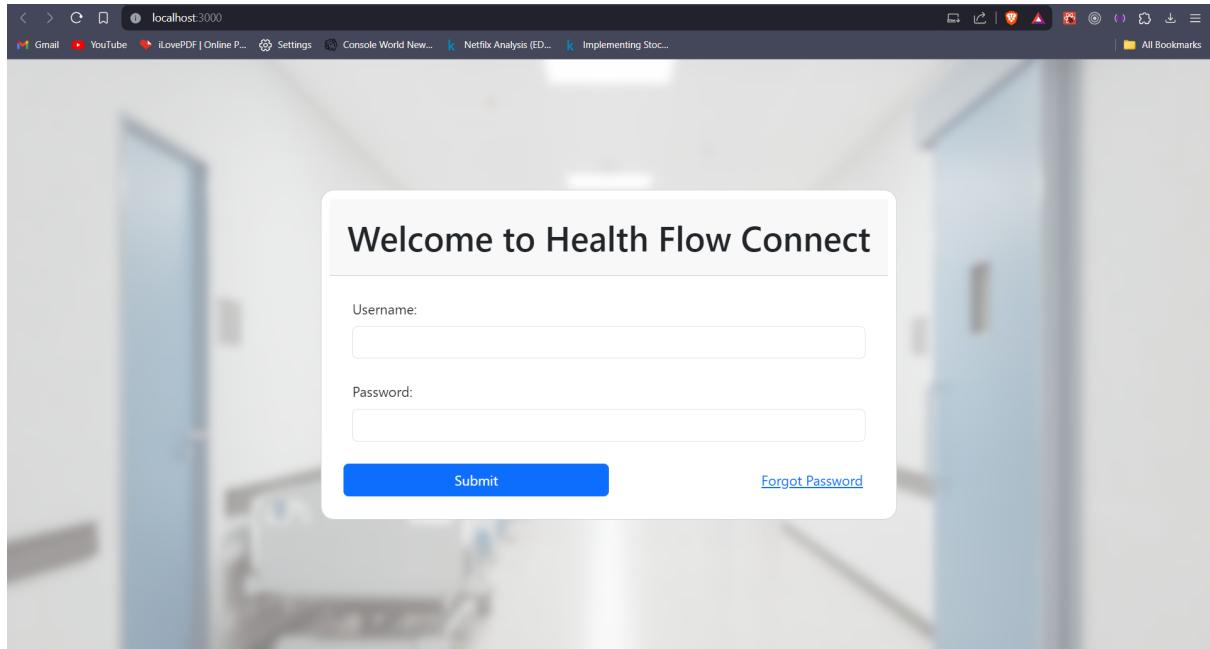


Figure 5.6: Login Page(Landing Page)

5.2.2 Admin Home Page

A screenshot of a web browser showing the "Admin Home Page" for "Health Flow Connect". The page has a light gray background with a blurred image of a hospital or medical facility in the background. It features two main sections:

- Admin Facilities** (left sidebar):
 - Create Doctor Record
 - Create Pharmacist Record
 - Create Lab Technician Record
 - Create Counter Record
 - Update Staff Record
 - Delete Staff Record
 - Get all staff Records
- Admin Information** (right panel):

UID	000000000000
First Name	Aditya
Middle Name	Prashant
Last Name	Choudhury
Gender	Male
Date of Birth	26-08-2003
Phone Number	7276333039
Address	Pune
Role	Admin
OPD	Surgery
Degree	MBBSMD

The browser's address bar shows "localhost:3000/admin/home_page" and the tab title is "Health Flow Connect". The top navigation bar includes links like WhatsApp, Gmail, YouTube, iLovePDF, Settings, Netflix Analysis, and Implementing Stoc... .

Figure 5.7: Admin Home Page

5.2.3 Admin Add Counter

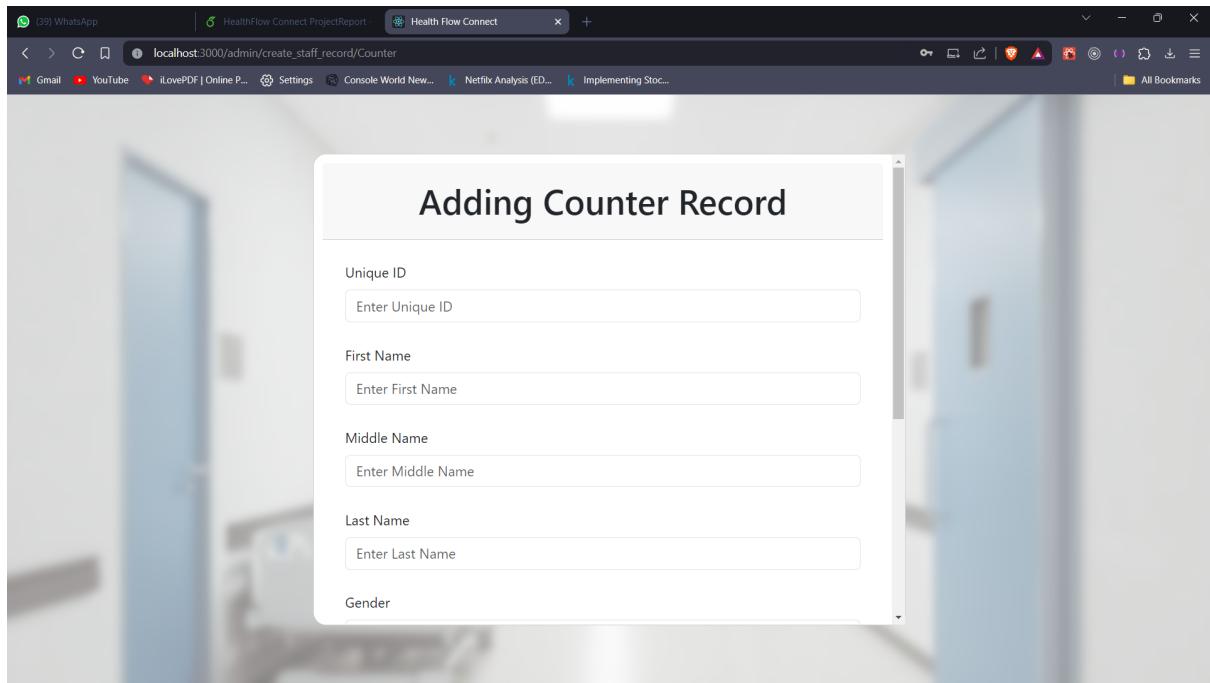


Figure 5.8: Admin Add Counter

5.2.4 Admin Add Doctor

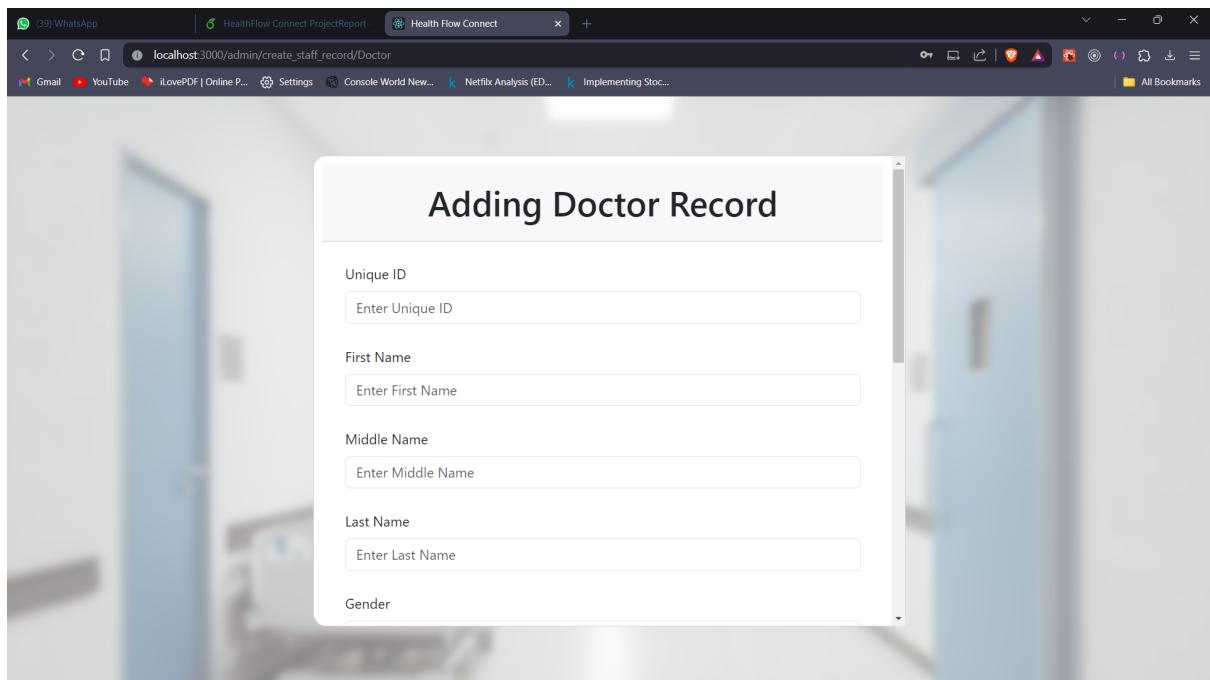


Figure 5.9: Admin Add Doctor

5.2.5 Admin Add Lab Technician

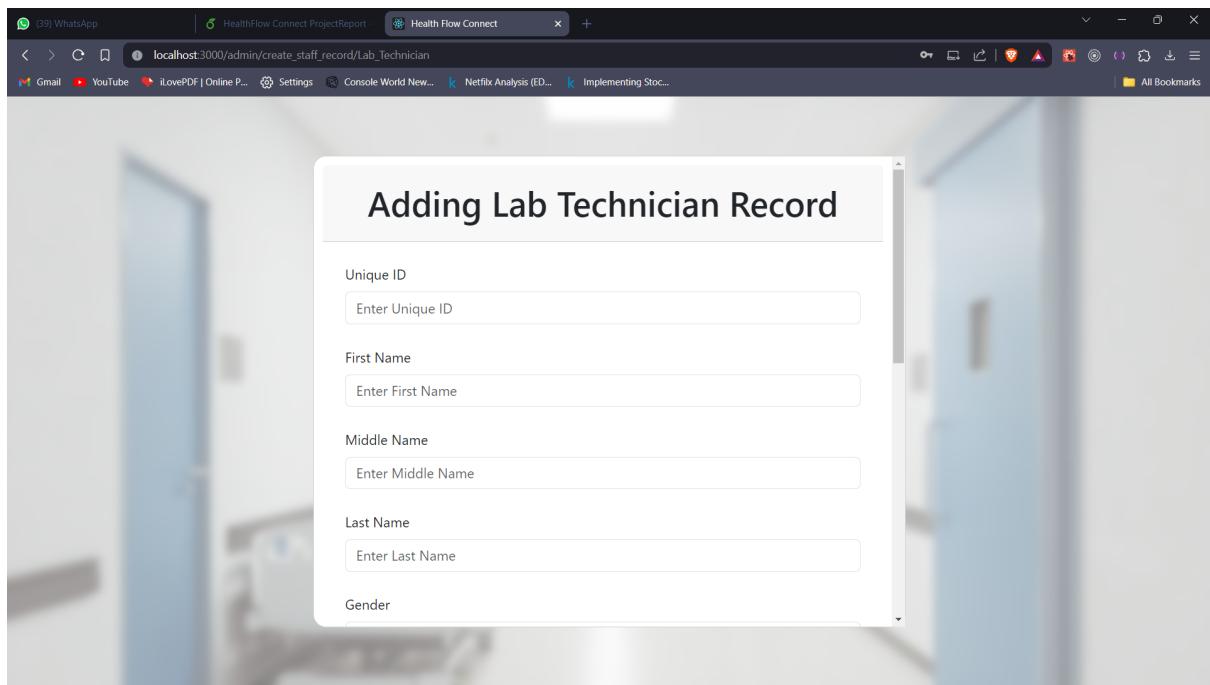


Figure 5.10: Admin Add Lab Technician

5.2.6 Admin Add Pharmacist

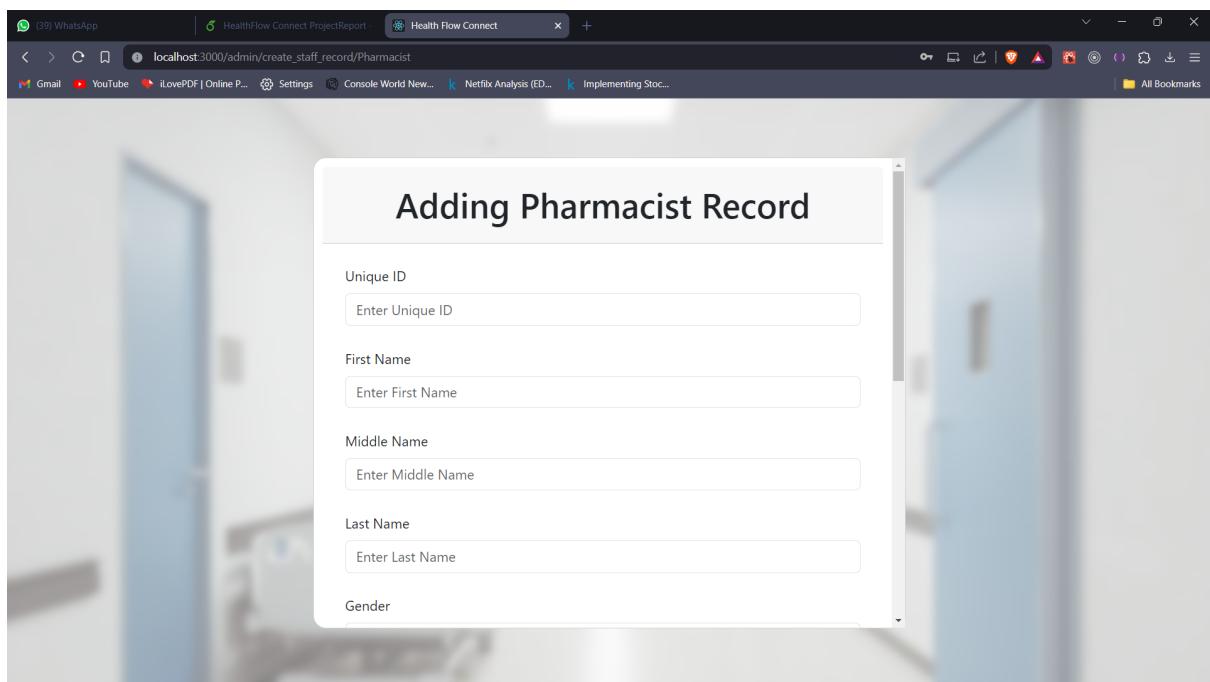


Figure 5.11: Admin Add Pharmacist

5.2.7 Admin Add Lab Technician

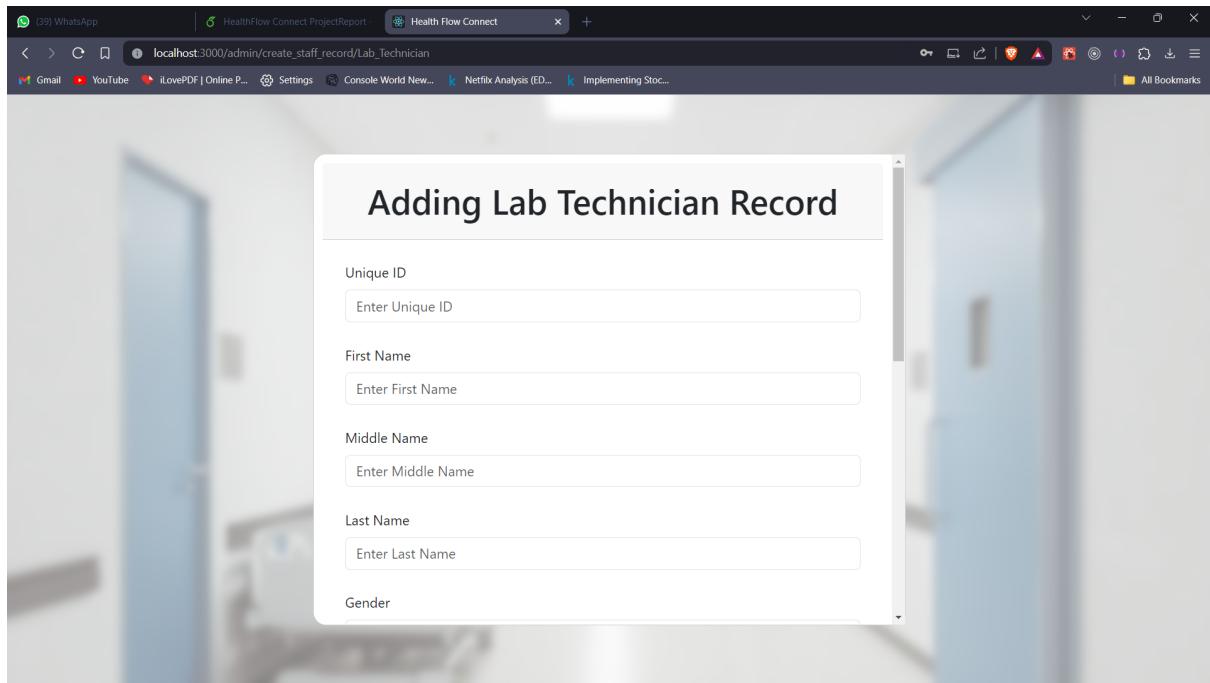


Figure 5.12: Admin Add Lab Technician

5.2.8 Admin Add Pharmacist

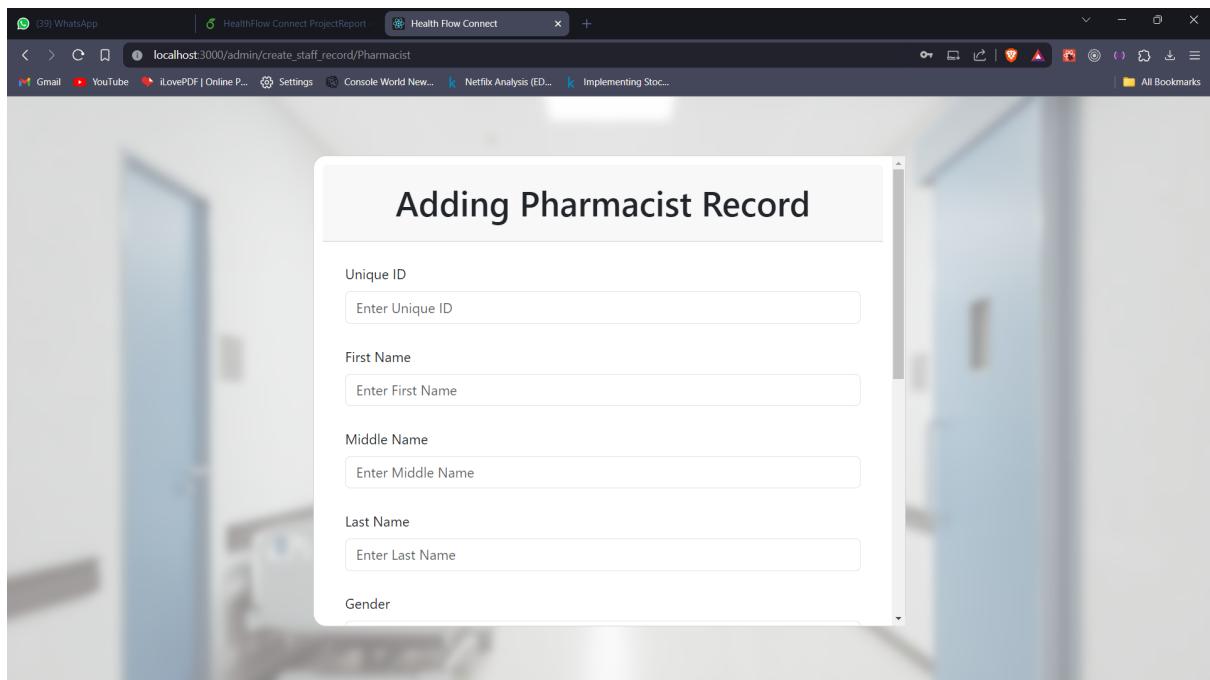


Figure 5.13: Admin Add Pharmacist

5.2.9 Admin Update Staff

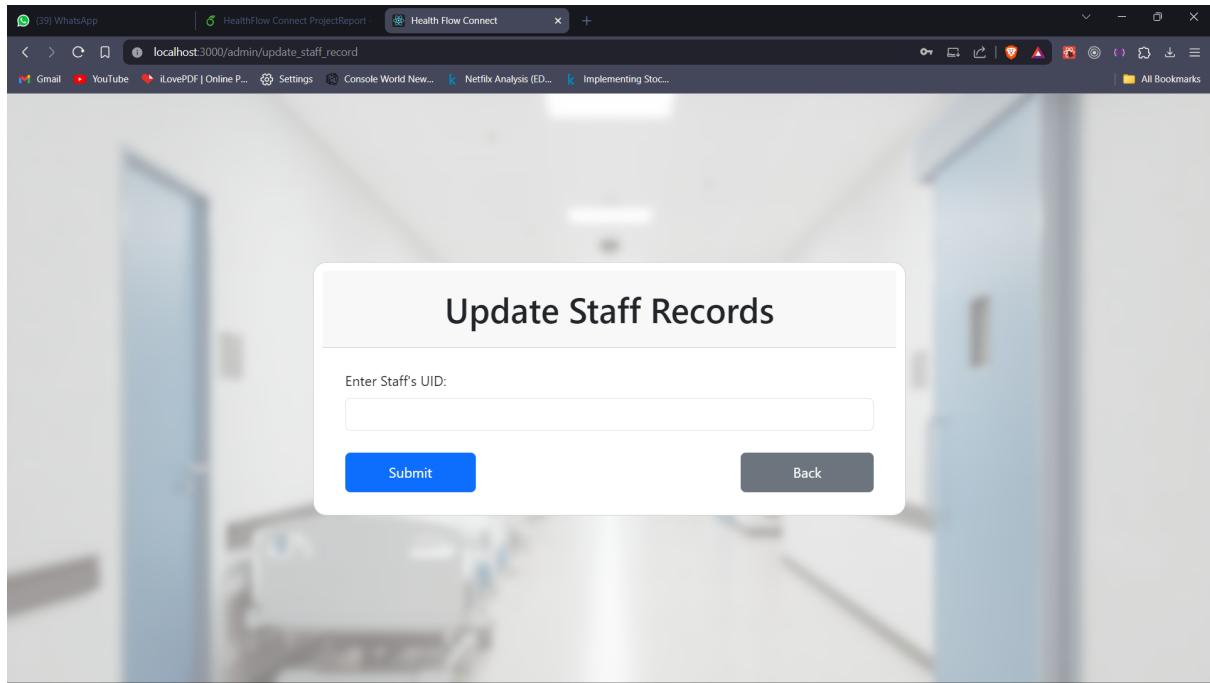


Figure 5.14: Admin Update Staff

5.2.10 Admin Delete Staff

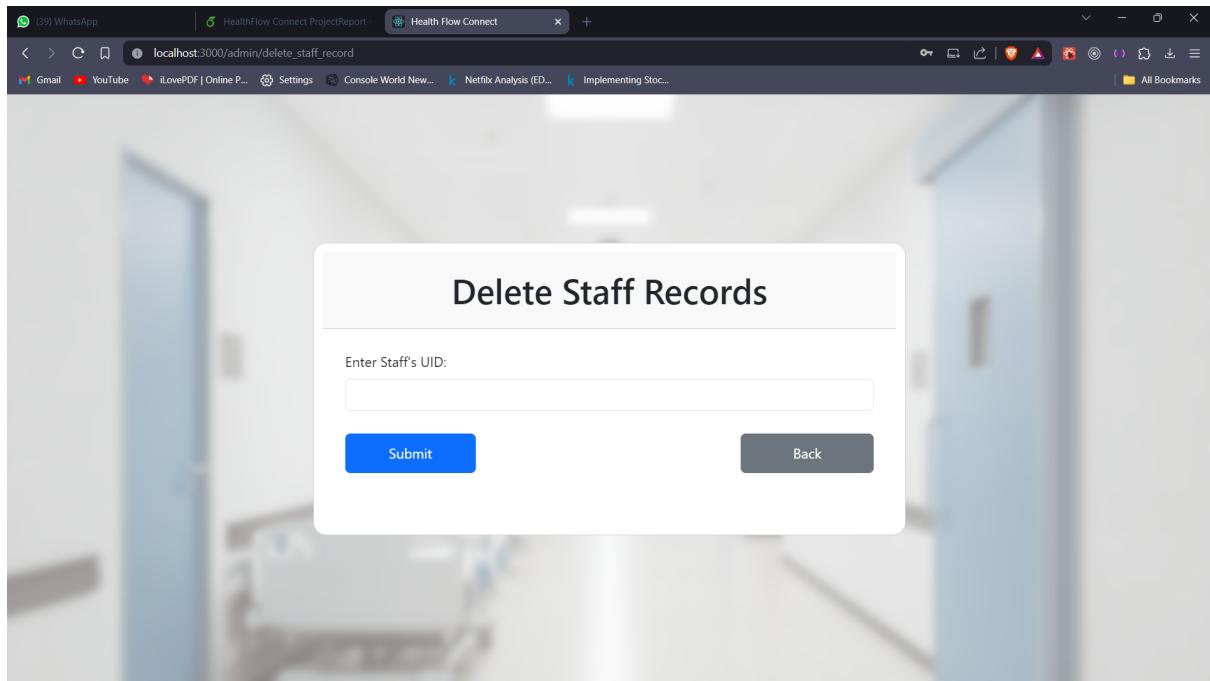


Figure 5.15: Admin Delete Staff

5.2.11 Admin Update Staff

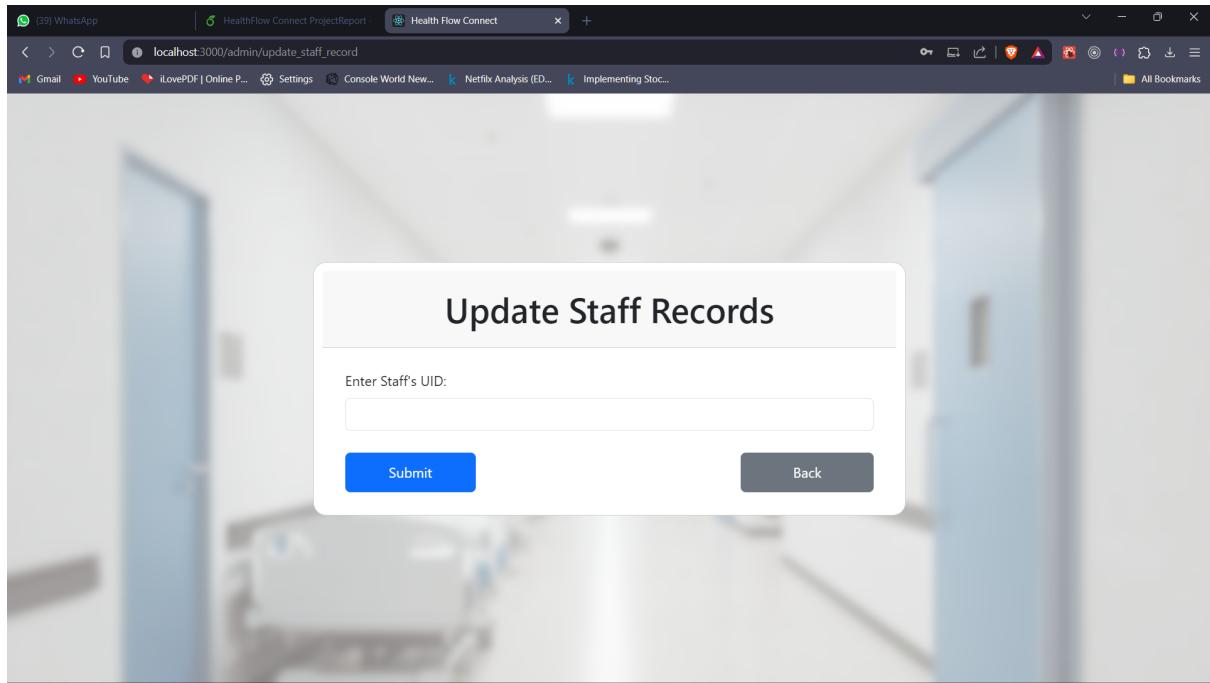


Figure 5.16: Admin Update Staff

5.2.12 Admin Delete Staff

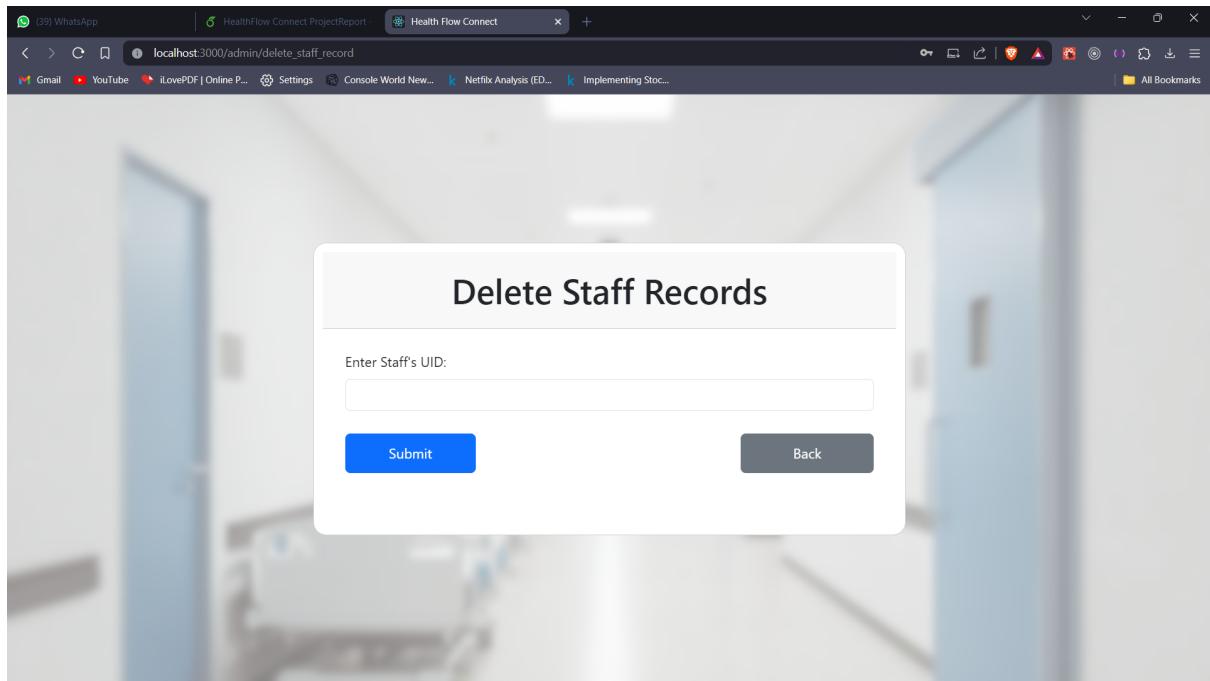


Figure 5.17: Admin Delete Staff

5.2.13 Admin All Staff Records

The screenshot shows a web browser window titled "Health Flow Connect" with the URL "localhost:3000/admin/all_staff_records". The page has a header "STAFF INFORMATION" and a "Back" button. A dropdown menu "Select Staff Designation:" is set to "Doctor". Below it is a "Submit" button. A search bar "Search..." and a "Print PDF" button are also present. The main content is a table with the following data:

UID	First Name	Middle Name	Last Name	Gender	Date of Birth	Phone Number	Address	OPD	Degree	Specialization
000000000000	Aditya	Prashant	Choudhary	Male	26-08-2003	7276333039	Pune	Surgery	MBBSMD	Heart Surgery
111100000000	Ayush	Sagar	Amborkar	Male	2024-04-05	7276333039	Pune	Surgery	MBBS	Surgeon

Figure 5.18: Admin All Staff Records

5.2.14 OTP Verification

The screenshot shows a web browser window titled "Health Flow Connect" with the URL "localhost:3000/otp_verify/forgot_password". The page has a central modal titled "OTP Verification". It contains fields for "U ID:" (with an input box) and "Send OTP" (a blue button). Below that is a field for "OTP:" (with an input box) and two buttons: "Submit" (blue) and "Resend OTP" (grey).

Figure 5.19: OTP Verification

5.2.15 Doctor Home Page

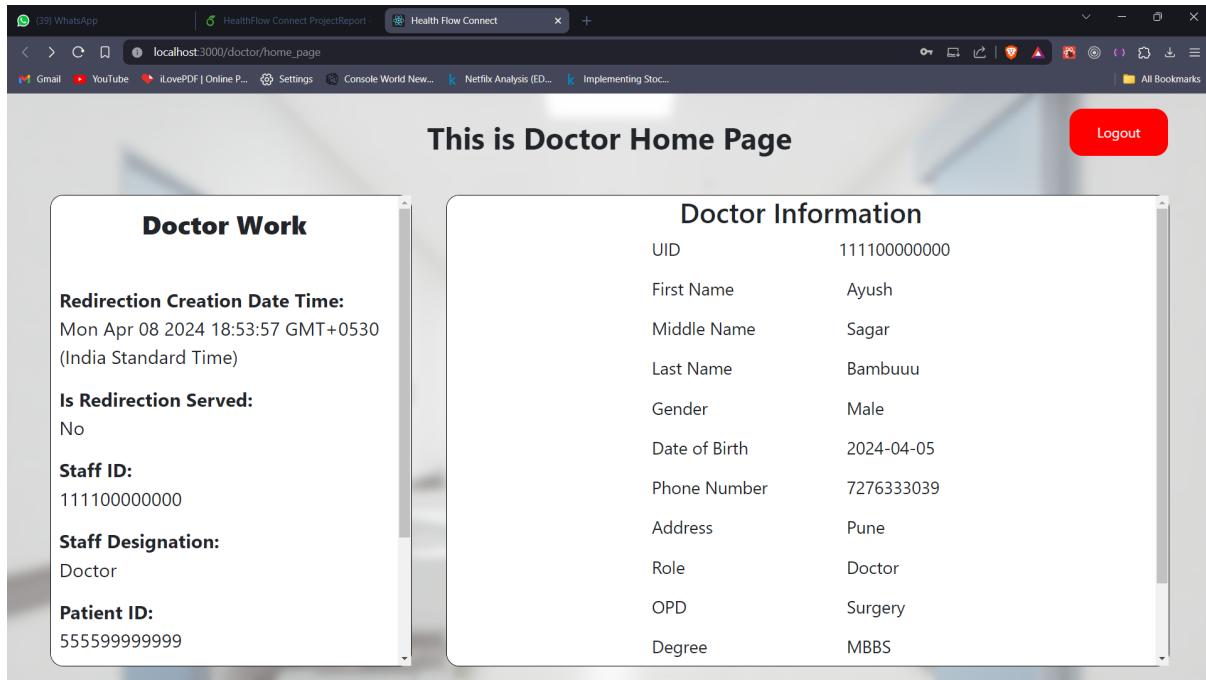


Figure 5.20: Doctor Home Page

5.2.16 Doctor Serve Page

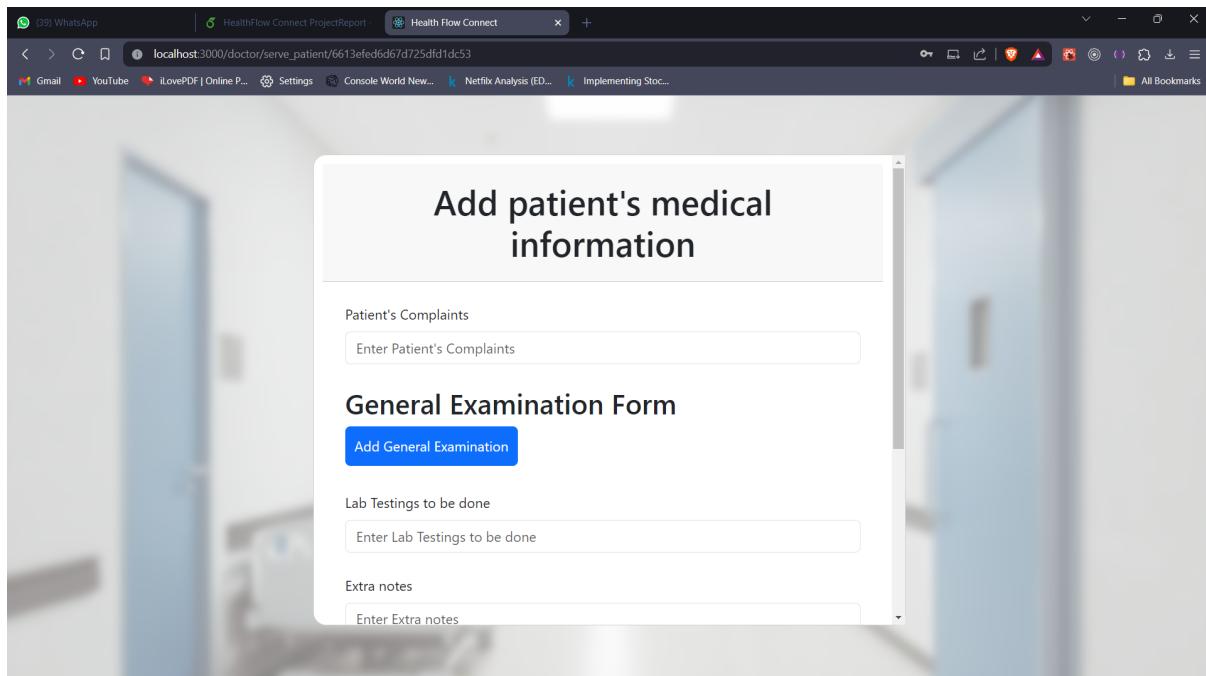


Figure 5.21: Doctor Serve Page

5.2.17 Pharmacist Home Page

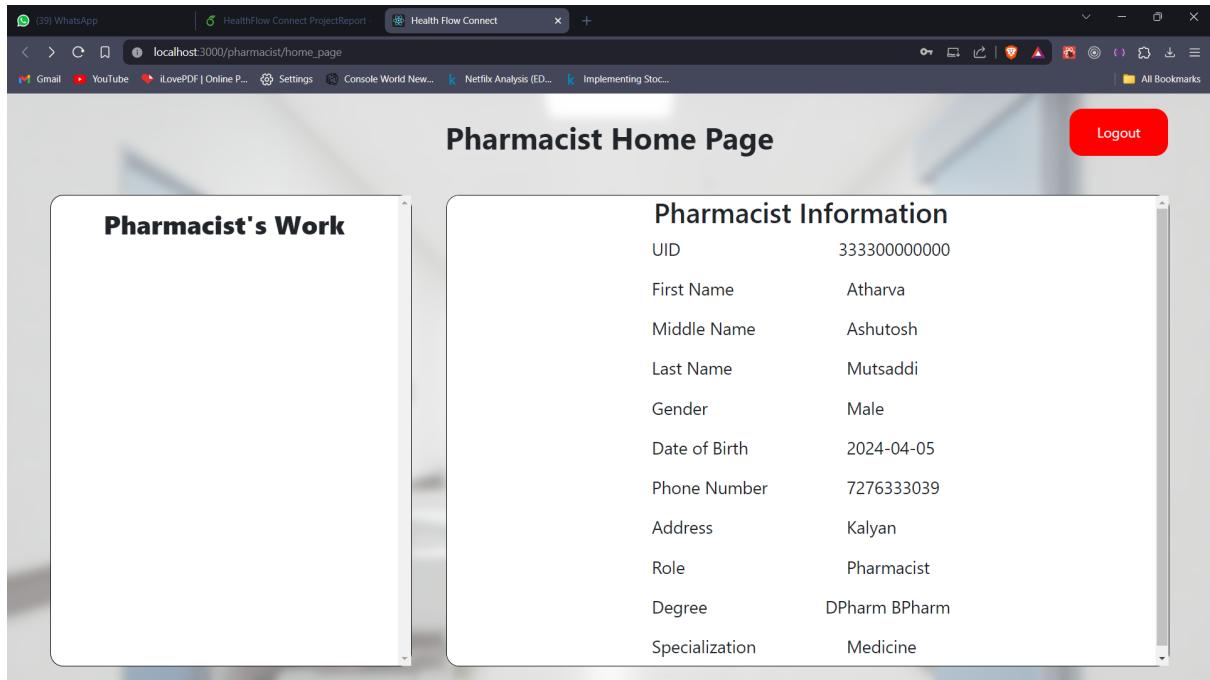


Figure 5.22: Pharmacist Home Page

5.2.18 Pharmacist Serve Page

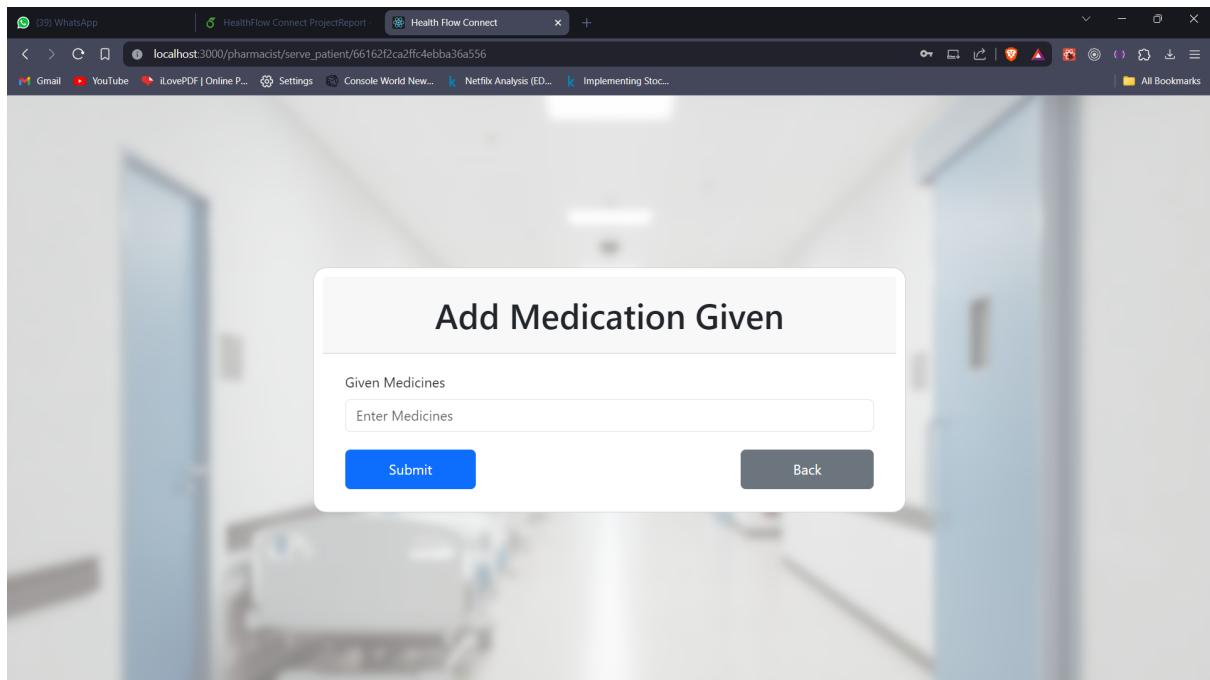


Figure 5.23: Pharmacist Serve Page

5.2.19 Lab Technician Home Page

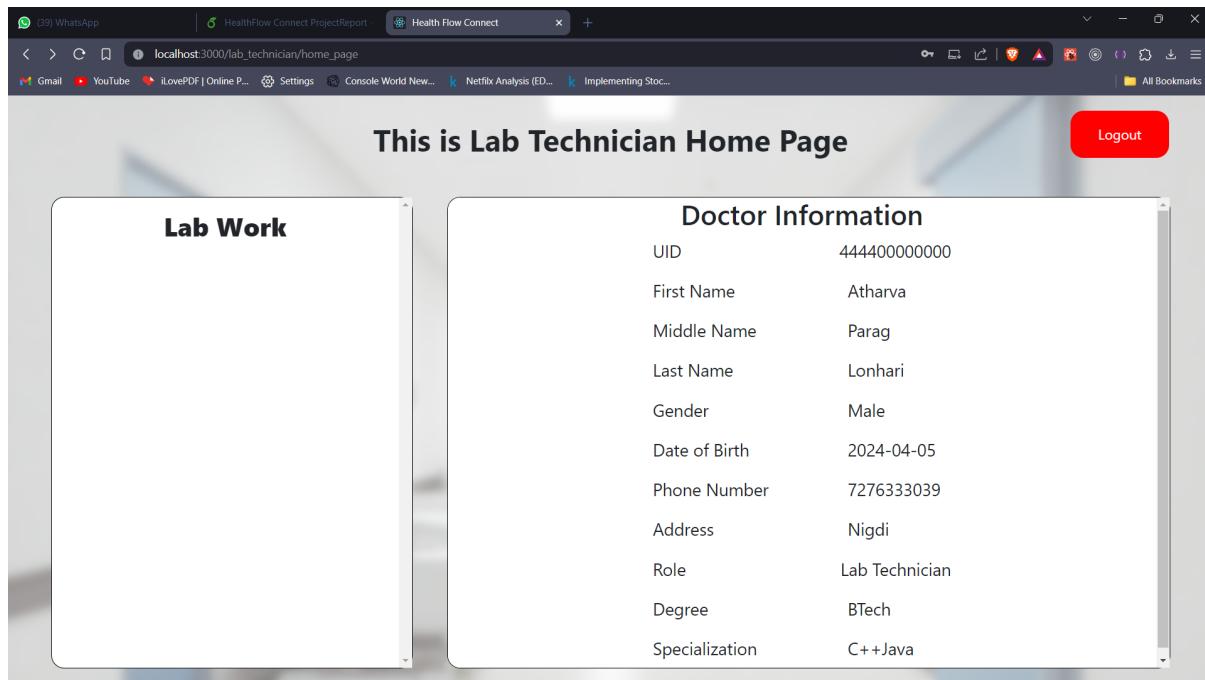


Figure 5.24: Lab Technician Home Page

5.2.20 Lab Technician Serve Page

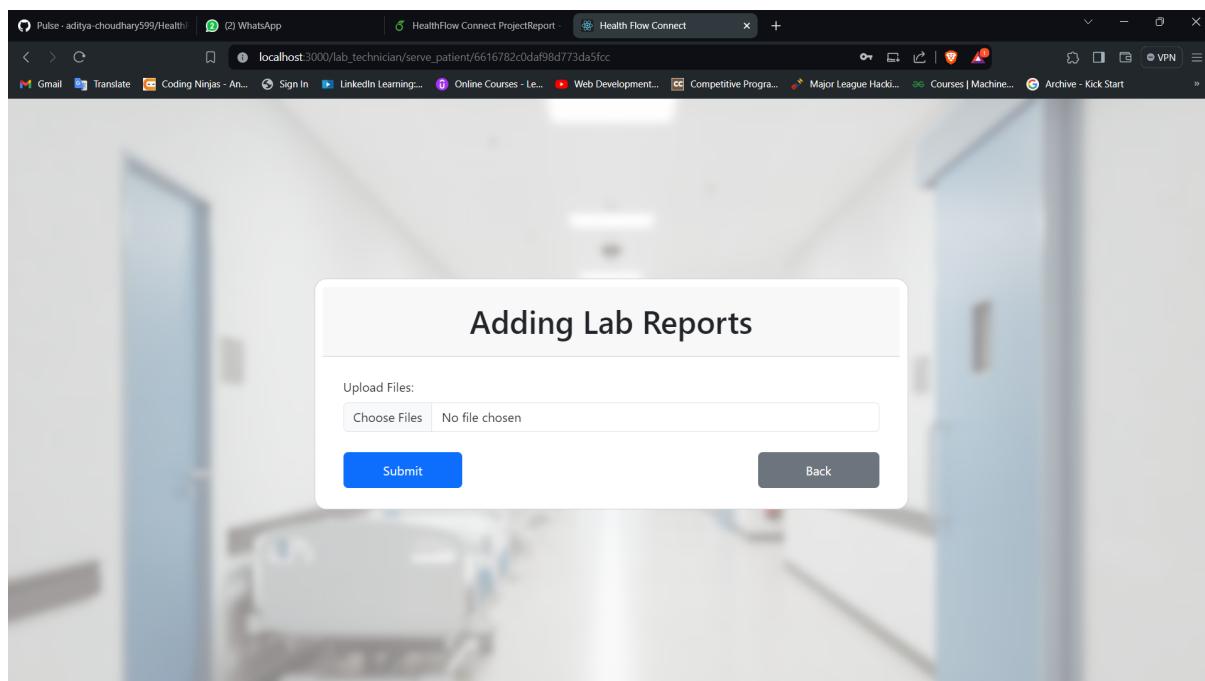


Figure 5.25: Lab Technician Serve Page

5.2.21 Lab Technician Home Page

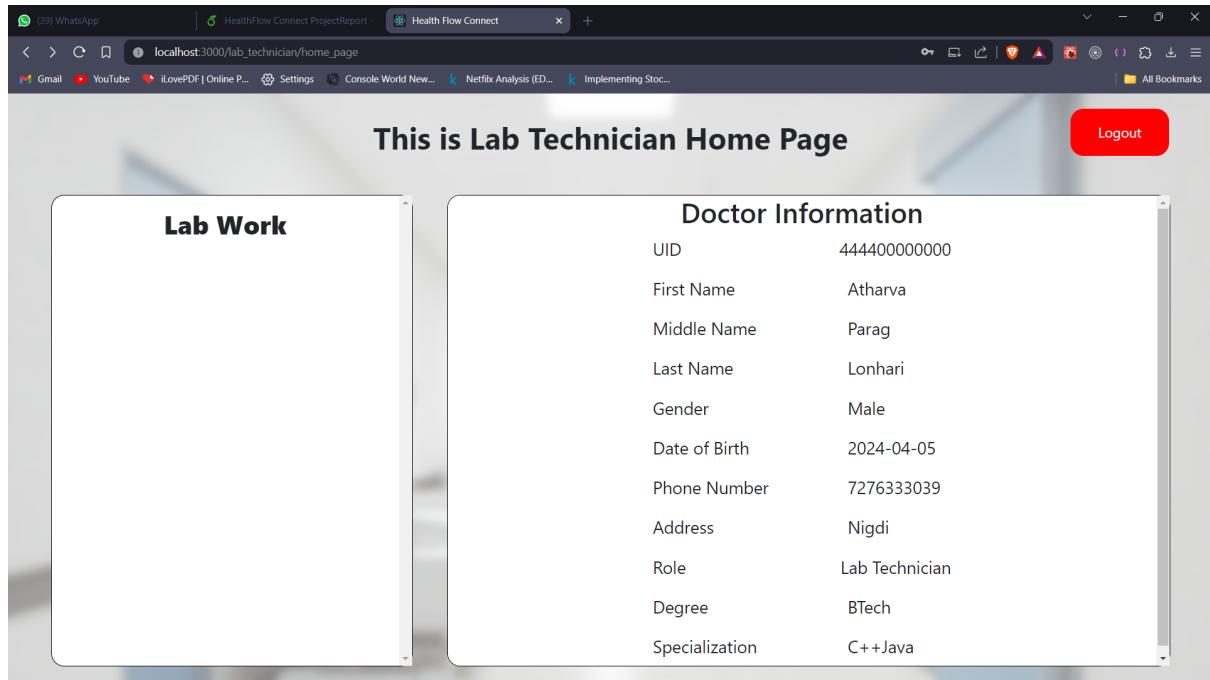


Figure 5.26: Lab Technician Home Page

5.2.22 Lab Technician Serve Page

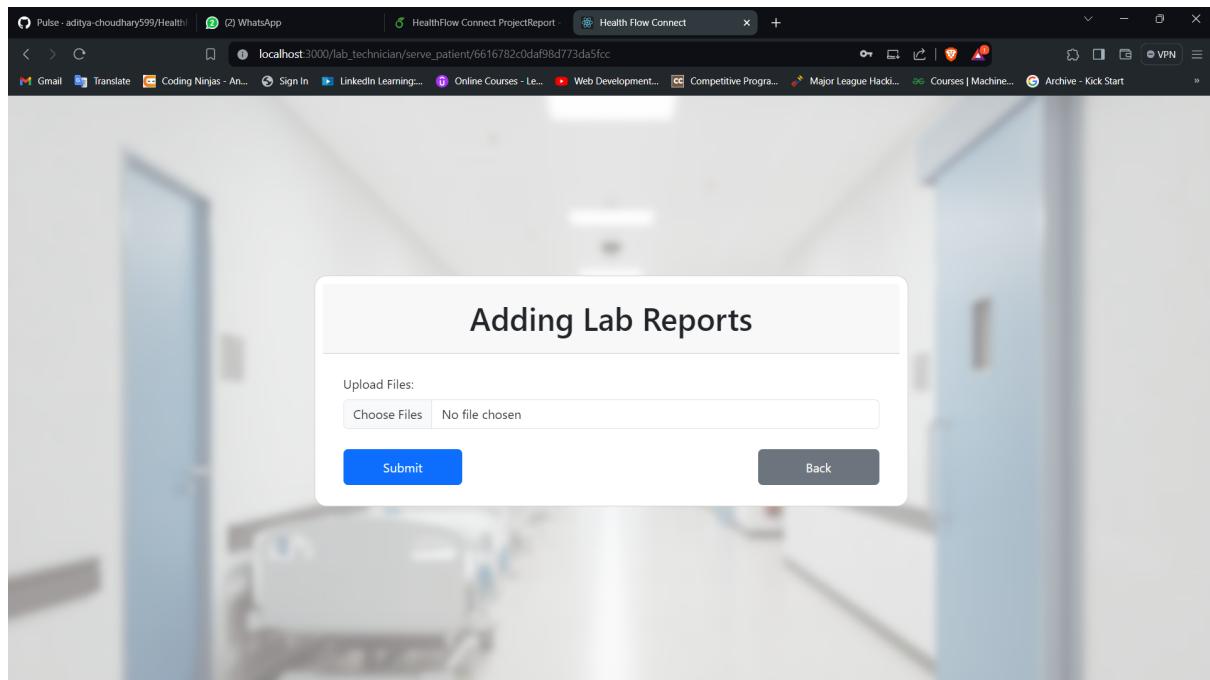


Figure 5.27: Lab Technician Serve Page

5.2.23 Counter Home Page

The screenshot shows a web browser window titled "Health Flow Connect" with the URL "localhost:3000/counter/home_page". On the left, there is a sidebar titled "Counter Facilities" containing links for "Create Patient Record", "Update Patient Record", "Redirect Patient", "Show Patient Redirection Records", "Print Case Paper", "Show all Patient Records", and "Logout". The main content area is titled "Counter Information" and displays the following data:

UID	222200000000
First Name	Vipul
Middle Name	Sanjay
Last Name	Chaud
Gender	Male
Date of Birth	2024-04-05
Phone Number	7276333039
Address	Kalyan
Role	Counter

Figure 5.28: Counter Home Page

5.2.24 Counter Add Patient Page

The screenshot shows a web browser window titled "Health Flow Connect" with the URL "localhost:3000/counter/create_patient_record". The main content area is titled "Adding Patient Record" and contains five input fields:

- Unique ID: A text input field with placeholder text "Enter Unique ID".
- First Name: A text input field with placeholder text "Enter First Name".
- Middle Name: A text input field with placeholder text "Enter Middle Name".
- Last Name: A text input field with placeholder text "Enter Last Name".
- Gender: A text input field with placeholder text "Select Gender".

Figure 5.29: Counter Add Patient Page

5.2.25 Counter Update Patient Page

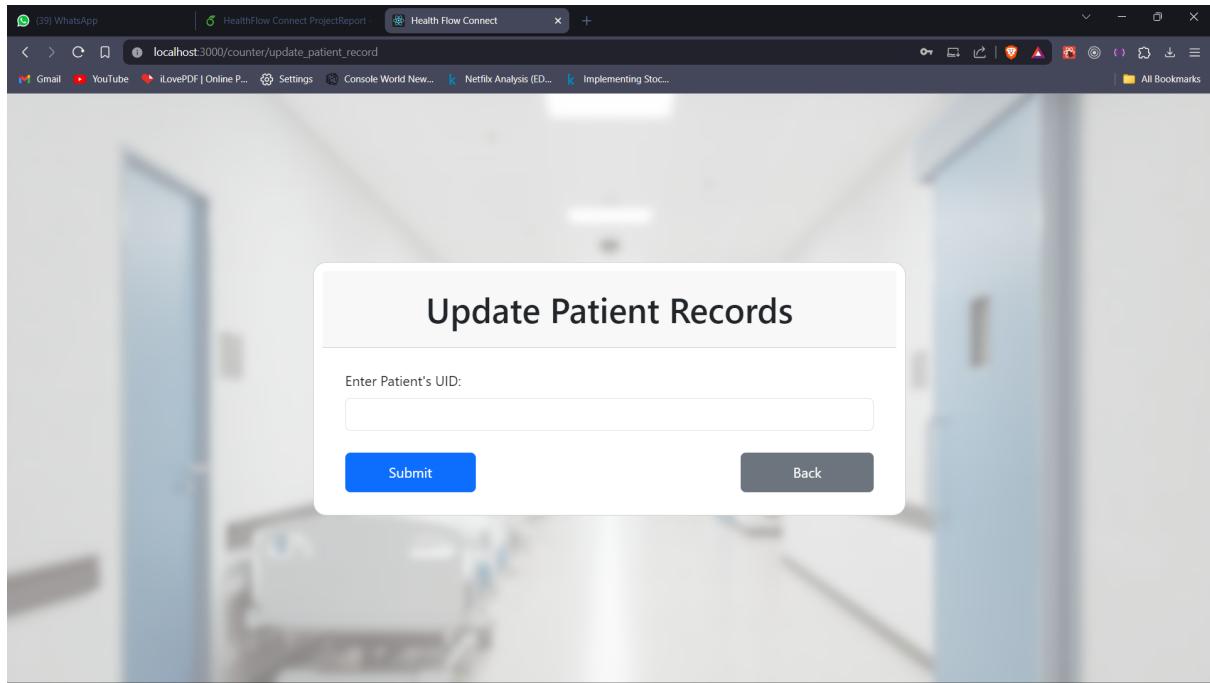


Figure 5.30: Counter Update Patient Page

5.2.26 Counter Redirect Patient Page

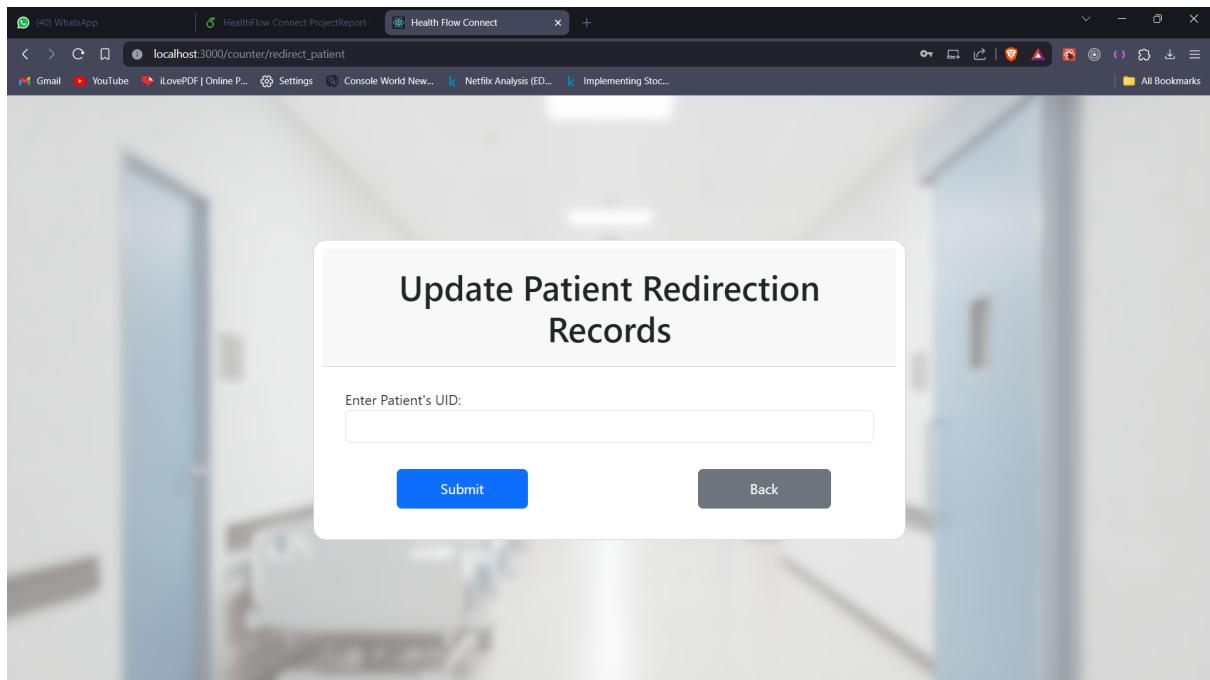


Figure 5.31: Counter Redirect Patient Page

5.2.27 Counter Print Case Paper Page

The image displays two screenshots of a web application interface for printing case papers. Both screenshots are titled "Print Case Paper".

Screenshot 1 (Top): This version shows a simple form where a patient's UID is entered. The input field contains "555511111111". Below the input field are two buttons: "Submit" (blue) and "Back" (grey). The main content area is titled "Patient Case Paper" and displays the following information:

- Start Date: Wed Apr 10 2024 16:59:05 GMT+0530 (India Standard Time)
- Harsh Dinesh Marke**
- Staff UID: 444400000000
- Staff Name: Atharva Parag Lonhari
- Staff Designation: Lab Technician
- Date Time: Wed Apr 10 2024 17:01:53 GMT+0530 (India Standard Time)
- Lab Report Ids:
 - 1. "661678a90daf98d773da5fd7"
 - 2. "661678a90daf98d773da5fd6"

Screenshot 2 (Bottom): This version shows a more detailed and structured case paper. It includes fields for:

- Sr no. : 1**
- Date & Time: Sat Apr 13 2024 13:01:34 GMT+0530 (India Standard Time)**
- Staff UID: 111100000000
- Staff Name: Ayush Sagar Amborkar
- Staff Designation: Doctor
- Complaint:
 - 1. Intense Pain in Shoulder
 - 2. High Fever
- General Examination:
 - 1. **Soulder:** Bruise
 - 2. **Temperature:** 101F
- Medicines Prescribed:
- Extra Notes:
 - 1. Chance of Dislocation
 - 2. Trauma due to Pain

At the bottom of this screenshot are three buttons: "Back" (grey), "Close Case Paper" (blue), and "Print" (grey).

Figure 5.32: Counter Print Case Paper Page

5.2.28 Counter View All Patient Page

The screenshot shows a web browser window titled "Health Flow Connect" with the URL "localhost:3000/counter/show_all_patient_records". The page has a header "All Patient Information" with a "Back" button. Below the header is a search bar labeled "Search..." and a blue "Print as PDF" button. A table displays eight patient records with columns: UID, First Name, Middle Name, Last Name, Gender, Date of Birth, Phone Number, and Address. The data is as follows:

UID	First Name	Middle Name	Last Name	Gender	Date of Birth	Phone Number	Address
555500000000	Manthan	Sanjay	Kshetrapal	Male	2024-04-05	7276333039	Nagpur
555511111111	Harsh	Dinesh	Marke	Male	2024-04-05	7276333039	Shirgaon
555512341234	Snehashish		Bose	Male	2024-04-06	7276333039	Pune
555511112222	Soham	Sunil	Kumthekar	Male	2024-05-03	7276333039	Wai(Sangli)
555522222222	Aditya	Vijaykumar	Agre	Male	2024-04-01	7276333039	Udgir
555599999999	Kaustubh	Dattatray	Virkar	Male	2024-04-08	7276333039	Seawood
555522223333	bgb	bgfb	bhg	Male	2024-04-08	7276333039	ffbfgb

Figure 5.33: Counter View All Patient Page

5.3 A few Code Snippets

5.3.1 Database Connect

```
backend > db_scripts > js db_connect.js > ...
1  import mongoose from "mongoose";
2
3  export const connect_db = async (DATABASE_URL) => {
4      try {
5          const encodedPassword = encodeURIComponent('Anushree@2011');
6          const uri = DATABASE_URL.replace('Anushree@2011', encodedPassword);
7
8          await mongoose.connect(uri, {
9              useNewUrlParser: true,
10             useUnifiedTopology: true,
11             serverApi: { version: '1', strict: true, deprecationErrors: true }
12         });
13
14          console.log("Connected to Database Successfully !");
15      } catch (err) {
16          console.log(err);
17      }
18  };
19
20 let gfs;
21 mongoose.connection.once('open', () => {
22     gfs = new mongoose.mongo.GridFSBucket(mongoose.connection.db, {
23         bucketName: 'uploaded_lab_reports',
24     });
25 });
26
27 export { gfs };
```

Figure 5.34: Database Connect

5.3.2 Server.js

```
backend > JS server.js > ...
1  // Configuration for accessing the env file data
2  import dotenv from "dotenv";
3  dotenv.config();
4
5  // Express is required for creating the server
6  import express from "express";
7
8  // Cors is required for cross-site origin access
9  import cors from "cors";
10
11 // Importing the function for connection to database
12 import { connect_db } from "./db_scripts/db_connect.js";
13
14 // Importing the routes
15 import { general_router } from "./api_routes/general_routes.js";
16 import { admin_router } from "./api_routes/admin_routes.js";
17 import { counter_router } from "./api_routes/counter_routes.js";
18 import { lab_technician_router } from "./api_routes/lab_technician_routes.js";
19 import { doctor_router } from "./api_routes/doctor_routes.js";
20 import { pharmacist_router } from "./api_routes/pharmacist_routes.js";
21 import { otp_router } from "./api_routes/otp_routes.js"
22
23 // This will be used for connection with local mongo database
24 await connect_db(process.env.DATABASE_URL);
25
26 // Configuration for backend express server
27 const app = express();
28 const server_port = process.env.SERVER_PORT;
29
30 // Configuration for api interaction
31 app.use(express.json());
32 app.use(cors());
33 app.use(express.urlencoded({ extended: false }));
34
35 // Configuration of routes
36 app.use('', general_router);
37 app.use('/admin', admin_router);
38 app.use('/counter', counter_router);
39 app.use('/lab_technician', lab_technician_router);
40 app.use('/doctor', doctor_router);
41 app.use('/pharmacist', pharmacist_router);
42 app.use('/otp', otp_router);
43
44 // Setting up the server
45 app.listen(server_port, () => {
46   |   return console.log(`Server is running at http://localhost:\${server\_port}`);
47});|
```

Figure 5.35: Server.js

5.3.3 App.jsx

```
frontend > src > App.jsx > [App] App
1  import React, { Fragment } from "react";
2  import { BrowserRouter, Routes, Route } from "react-router-dom";
3
4  import { Logout_Component } from "./components/Logout_Component.jsx";
5
6  import { Website_Landing_Page } from "./pages/Website_Landing_Page.jsx";
7  import { OTP_Verification_Page } from "./pages/OTP_Verification_Page.jsx";
8  import { Forgot_Password_Page } from "./pages/Forgot_Password_Page.jsx";
9
10 import { Admin_Home_Page } from "./pages/Admin/Admin_Home_Page.jsx";
11 import { Create_Staff_Records_Page } from "./pages/Admin/Create_Staff_Records_Page.jsx";
12 import { Delete_Staff_Records_Page } from "./pages/Admin/Delete_Staff_Records_Page.jsx";
13 import { Update_Staff_Records_Page } from "./pages/Admin/Update_Staff_Records_Page.jsx";
14 import { See_All_Staff_Records_Page } from "./pages/Admin/See_All_Staff_Records_Page.jsx";
15
16 import { Counter_Home_Page } from "./pages/Counter/Counter_Home_Page.jsx";
17 import { Create_Patient_Records_Page } from "./pages/Counter/Create_Patient_Records_Page.jsx";
18 import { Update_Patient_Records_Page } from "./pages/Counter/Update_Patient_Records_Page.jsx";
19 import { Redirect_Patient_Page } from "./pages/Counter/Redirect_Patient_Page.jsx";
20 import { Show_Patient_Redirection_Record } from "./pages/Counter>Show_Patient_Redirection_Record.jsx";
21 import { Print_Case_Paper } from "./pages/Counter/Print_Case_Paper.jsx";
22 import { Show_All_Patients_Page } from "./pages/Counter/show_All_Patients_Page.jsx";
23
24 import { Doctor_Home_Page } from "./pages/Doctor/Doctor_Home_Page.jsx";
25 import { Doctor_Serve_Patient_Page } from "./pages/Doctor/Doctor_Serve_Patient_Page.jsx"
26
27 import { Lab_Technician_Home_Page } from "./pages/Lab_Technician/Lab_Technician_Home_Page.jsx";
28 import { Lab_Technician_Serve_Patient_Page } from "./pages/Lab_Technician/Lab_Technician_Serve_Patient_Page.jsx";
29
30 import { Pharmacist_Home_Page } from "./pages/Pharmacist/Pharmacist_Home_Page.jsx";
31
32 import { Show_Patient_History } from "./pages/Doctor>Show_Patient_History.jsx";
33
34 import { Pharmacist_Serve_Patient_Page } from "./pages/Pharmacist/Pharmacist_Serve_Patient_Page.jsx";
35
36 export const App = () => {
37   return (
38     <Fragment>
39       <BrowserRouter>
40         <Routes>
41           <Route path="/" element={<Website_Landing_Page />} />
42           <Route path="/logout/" element={<Logout_Component />} />
43           <Route path="/forgot_password/" element={<Forgot_Password_Page />} />
44           <Route path="/otp_verify/:success_url" element={<OTP_Verification_Page />} />
45
46           <Route path="/admin/home_page" element={<Admin_Home_Page />} />
47           <Route path="/admin/create_staff_record/:entity" element={<Create_Staff_Records_Page />} />
48           <Route path="/admin/delete_staff_record" element={<Delete_Staff_Records_Page />} />
49           <Route path="/admin/update_staff_record" element={<Update_Staff_Records_Page />} />
50           <Route path="/admin/all_staff_records" element={<See_All_Staff_Records_Page />} />
51
52           <Route path="/counter/home_page" element={<Counter_Home_Page />} />
53           <Route path="/counter/create_patient_record" element={<Create_Patient_Records_Page />} />
54           <Route path="/counter/update_patient_record" element={<Update_Patient_Records_Page />} />
55           <Route path="/counter/redirect_patient" element={<Redirect_Patient_Page />} />
56           <Route path="/counter/show_patient_redirection_record" element={<Show_Patient_Redirection_Record />} />
57           <Route path="/counter/print_case_paper" element={<Print_Case_Paper />} />
58           <Route path="/counter/show_all_patient_records" element={<Show_All_Patients_Page />} />
59
60           <Route path="/doctor/home_page" element={<Doctor_Home_Page />} />
61           <Route path="/doctor/serve_patient/:redirection_id" element={<Doctor_Serve_Patient_Page />} />
62           <Route path="/doctor/show_patient_history" element={<Show_Patient_History />} />
63
64           <Route path="/lab_technician/home_page" element={<Lab_Technician_Home_Page />} />
65           <Route path="/lab_technician/serve_patient/:redirection_id" element={<Lab_Technician_Serve_Patient_Page />} />
```

Figure 5.36: App.jsx

5.4 Final Case Paper

Patient Case Paper

Start Date: Fri Apr 12 2024 10:46:45 GMT+0530 (India Standard Time)

Harsh Dinesh Marke

Sr no. : 1

Date & Time: Fri Apr 12 2024 10:47:56 GMT+0530 (India Standard Time)

Staff UID: 333300000000

Staff Name: Atharva Ashutosh Mutsaddi

Staff Designation: Pharmacist

General Examination:

Medicines Prescribed:

Medicines Given:

1. Paracetamol
 2. Sinarist
-

Figure 5.37: Case paper

Chapter 6

Summary and Conclusion

6.1 Summary

The HealthFlow Connect project aimed to digitize the case papers used in government hospitals in India, streamlining the process of managing patient records and improving healthcare delivery. Leveraging modern technologies such as the MERN stack and RESTful APIs, the project developed a comprehensive system that caters to the needs of various healthcare professionals and stakeholders. Key features include user management, patient record management, diagnosis and treatment tracking, lab test management, medication dispensation, and reporting capabilities. Through iterative development, stakeholder collaboration, and adherence to regulatory standards, the project successfully delivers a robust and user-friendly solution for digitizing healthcare records in government hospitals.

6.2 Conclusion

In conclusion, the HealthFlow Connect project can make significant strides in modernizing healthcare operations and improving patient care in government hospitals in India. By digitizing case papers and implementing a centralized system for managing patient

records, the project can enable healthcare professionals to make informed decisions, collaborate more effectively, and deliver better outcomes for patients. Moving forward, continued enhancements, user training, and stakeholder engagement will be crucial to maximizing the impact of the HealthFlow Connect system and ensuring its long-term sustainability. With a commitment to innovation and excellence in healthcare delivery, the project aims to set a new standard for digitized healthcare systems in India, driving positive change and improving health outcomes for all.