Tribhuvan University

Academia International College

Final Year Project Report

On

Healthsyno HMS (Tailored Hospital/Clinic Management System)

[CSC 412]

Under the supervision of

"Dr. Sunil Chaudhary"

Submitted by

Bichitra Gautam (T.U. Exam Roll No. 29008/078)

Prajjwal Rana (T.U. Exam Roll No. 29021/078)

Submitted to

Department of Computer Science and Information Technology

Academia International College

Institute of Science and Technology

Tribhuvan University

September 2025

Tribhuvan University
Academia International College

Final Year Project Report

On

Healthsyno HMS (Tailored Hospital/Clinic Management System)

[CSC 412]

A final year project submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science and Information Technology awarded by Tribhuvan University

Submitted by

Bichitra Gautam (T.U. Exam Roll No. 29008/078)

Prajjwal Rana (T.U. Exam Roll No. 29021/078)

Submitted to

Department of Computer Science and Information Technology

Academia International College

Institute of Science and Technology

Tribhuvan University

September 2025

Tribhuvan University

Institute of Science and Technology

Academia International College

Department of Computer Science and Information Technology

Email: mail@academiacollege.edu.np

Supervisor's Recommendation

I hereby recommend that the project work prepared under my supervision by below listed team of students entitled "**Healthsyno HMS"** be accepted as in fulfilling partial requirement for completion of Four Year's bachelor's degree in computer science and information technology. In my best knowledge this is an original work in Computer Science and Information Technology.

Student's Name

1. Bichitra Gautam (29008/078)
2. Prajjwal Rana (29021/078)

.....

Dr. Sunil Chaudhary

Project Supervisor

Department of Computer Science and Information Technology

Academia International College

Gwarko, Lalitpur

Tribhuvan University

Department of Computer Science and Information Technology

Academia International College

Certificate of Approval

We certify that we have read this project work report, and, in our opinion, it is appreciable for the scope and quality as a project work in the partial fulfillment of the requirements of Four Years Bachelor Degree of Science in Computer Science and Information Technology.

…… **Dr. Sunil Chaudhary** Project Supervisor Department of Computer Science and IT Academia International College	…….. **Er. Biswas Mathema** HOD/Program Coordinator Department of Computer Science and IT Academia International College
…… **Internal Examiner** Academia International College	…… **External Examiner** Central Department of CSIT Tribhuvan University

Acknowledgement

We want to convey our appreciation to the respected project supervisor, **Dr. Sunil Chaudhary**, for his continuous guidance, feedback, and encouragement throughout the development of this project. His observations and guidance were helpful for us in shaping our understanding of hospital management systems and software engineering practices.

We also like to extend our sincere appreciation to the faculty members of the Department of Computer Science and Information Technology at **Academia International College** for providing us with academic support and a favorable environment for learning and project execution.

Furthermore, we also acknowledge healthcare professionals who shared their valuable experiences and practical knowledge. Their contributions were vital in designing a solution that fits with real-world needs.

Bichitra Gautam [29008/078]

Prajjwal Rana [29021/078]

Abstract

Healthsyno HMS is a web-based Hospital Management System designed to address operational difficulties faced by healthcare institutions of Nepal, particularly those that rely on traditional paper-based processes. Managing patient records, billing errors, lack of centralized access to medical data are the challenges that it brings

Healthsyno HMS seeks on to deliver a complete digital solution that streamlines key hospital operations such as patient registration, billing, laboratory result management, and report generation. The system supports multiple patient categories, including regular patients, students, and foreign employment applicants. Key features include role-based access control, facial recognition for identity verification, an interactive dashboard with analytics, and exportable reports.

Using modern technologies, the system is developed using tools such as Next.js, TypeScript, Material UI, Dart SASS and Google Firebase. Healthsyno system mainly focuses on scalability, usability, and security in mind. Although currently a prototype, the system shows great potential for future deployment in real healthcare environments.

Keywords: Hospital Management System, Patient Record Management, Facial Recognition, Healthcare Digitization

Table of Contents

List of Figures

List of Tables

List Of Abbreviations

AI: Artificial Intelligence

API: Application Programming Interface

CSV: Comma Separated Values

CRUD: Create, Read, Update, Delete

Dart: Dart Programming Language

DBMS: Database Management System

DFD: Data Flow Diagram

HER: Electronic Health Record

E-R: Entity Relationship

EMR: Electronic Medical Record

ERD: Entity-Relationship Diagram

Firebase: Google Firebase

HTTPS: Hypertext Transfer Protocol Secure

HMS: Hospital Management System

HTTP: Hypertext Transfer Protocol

IDE: Integrated Development Environment

IoT: Internet of Things

IPD: Inpatient Department

JS: JavaScript

JSON: JavaScript Object Notation

LBPH: Local Binary Patterns Histograms

LRU: Least Recently Used

ML: Machine Learning

MUI: Material UI

Next.js: Next.js Framework

NoSQL: No Structured Query Language

OPD: Outpatient Department

ORM: Object-Relational Mapping

OTP: One-Time Password

PDF: Portable Document Format

QA: Quality Assurance

RAM: Random Access Memory

REST: Representational State Transfer

ROC: Receiver Operating Characteristic

ROI: Return on Investment

SASS: Syntactically Awesome Style Sheets

SDLC: Software Development Life Cycle

SSD: Single Shot MultiBox Detector

SVM: Support Vector Machine

TypeScript: TypeScript Language

UI: User Interface

UX: User Experience

Chapter 1: Introduction

1.1 Introduction

Many healthcare institutions and organizations in Nepal still use a manual, paper-based system to manage patient records, medical reports, billing, laboratory reports, and other administrative tasks. This traditional approach often leads to misplaced files, data losses, and data duplication. As healthcare services improve, more people expect digital tools that make hospital management easier and provide better experience for patients. Healthsyno HMS is a web-based hospital management system designed tackle these issues by digitizing and centralizing hospital operations into one system. It aims to offer a user-friendly solution that supports patient identification, verification, medical test management, billing, reporting, and more on a single platform.

By implementing Healthsyno HMS, healthcare providers and organizations can cut down on paperwork and record management, reduce errors, prevent patient data duplication, and ensure that correct information is available to authorized personnel. This approach helps hospitals provide better care and operate more efficiently and transparently, especially as everything shifts to digital.

1.2 Problem Statement

Despite the increasing demand for efficient health care services, hospitals and health care organizations in Nepal still rely on manual paper-based systems to keep records of patient information, billing, and administrative tasks. This leads to frequent data loss, duplication, billing errors, and operational inefficiencies. It burdens healthcare staff with excessive manual data handling and compromises the easy access to patients' information. Thus, there is a critical need for digital hospital management systems like Healthsyno HMS that can centralize and automate these processes.

1.3 Objectives

This project is all about creating a digital platform that makes it easier to manage medical and clinic operations and keep track of patient records in one place. The main objectives of Healthsyno include:

- Create a digital platform that brings together and simplifies how medicals/clinics run their daily tasks and manage patient information.
- To ensure accurate patient identification and prevent duplication of patient records.
- To provide secure access to health data for improved healthcare delivery.

1.4 Scope and Limitations

1.4.1 Scope

- Targets small to medium-sized clinics and hospitals in Nepal, supporting up to 1,000 active patients with core modules for patient registration, billing, and laboratory management.
- Focuses on role-based access control and facial recognition for identity verification, enabling efficient handling of diverse patient categories (regular, students, foreign employment applicants).
- Designedfor scalability, allowing future expansion to include advanced analytics, report exports (PDF/CSV), and integration with basic healthcare workflows.

1.4.2 Limitations

- As prototype, it lacks full-scale production deployment, potentially requiring additional server resources for high-volume usage beyond 1,000 patients.
- Limited to web-based access without a native mobile app, which may hinder on-the-go usage by healthcare staff in remote or mobile-heavy environments.
- Does not include real-time integration with external medical devices (e.g., IoT sensors) or advanced predictive ML features, restricting itto basic digitization without automated diagnostics.

1.5 Methodology

Our hospital management software, Healthsyno HMS, was created especially for the healthcare system in Nepal. This guarantees that the system meets their requirements and workflow processes. In order to develop an adaptable, user-centered, and iterative strategy, we have select the Prototyping Model. This approach helps us achieve our objective of creating a robust, effective, and easily accessible solution. It cover the special traits, difficulties, and advantages of Nepal's medical facilities, such as handling patient records,

billing, and lab results in billing, as well as patient records in environments with limited resources. We can develop, test, and improve the software using the prototyping model, and we can make continuous enhancements in response to user feedback. This guarantees that the finished product satisfies the requirements of healthcare providers.

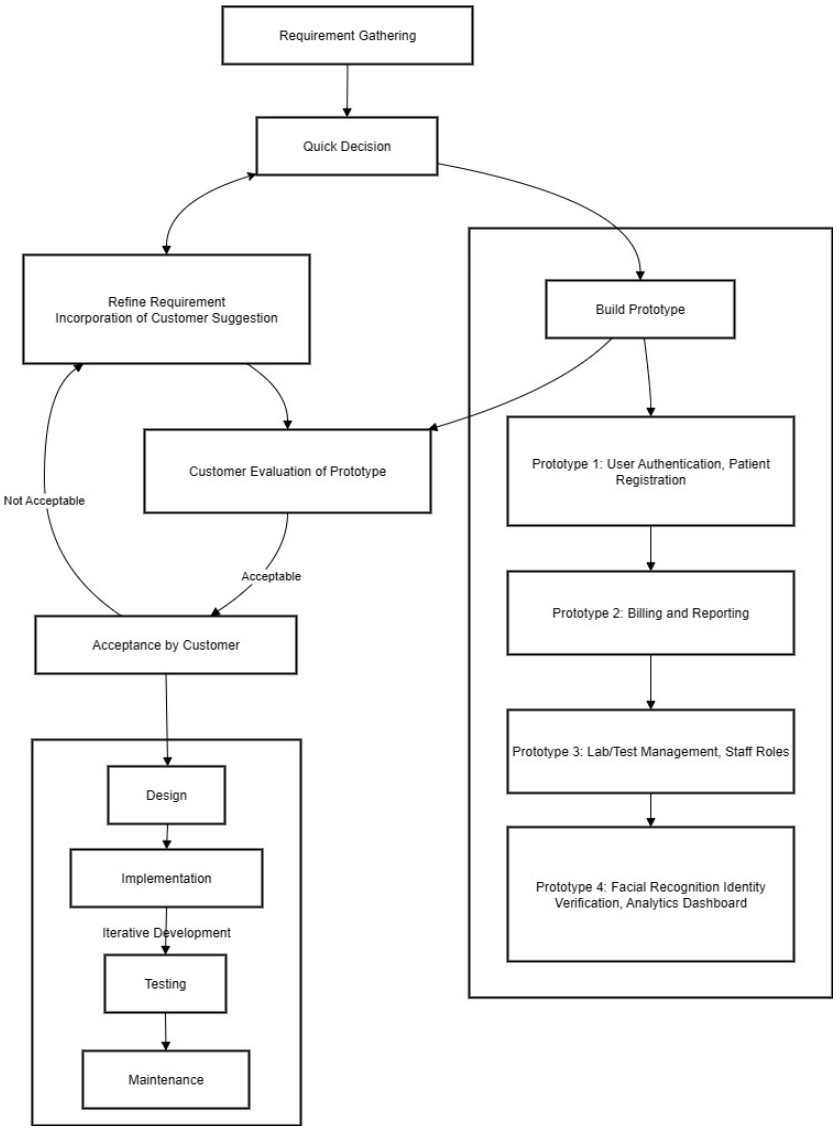

Figure 1.1: Prototyping model

The prototyping process involved the following steps:

- **Prototype Planning**

For initial prototype of the system key features were identified and defined first, including user authentication, patient record management, billing, and laboratory result handling.

- **Prototype Development**

Functioning prototype was built using technologies and tools such as Next.js, TypeScript, and Material UI. The initial version is integrated with essential core functionalities to demonstrate basic system operations like usability and intended workflow.

  - **Prototype 1:** User authentication, patient registration.
  - **Prototype 2:** Billing and reporting.
  - **Prototype 3:** Medical Report Template, Staff and Agents Management.
  - **Prototype 4:** Facial recognition identity verification, analytics dashboard.

- **User Feedback Collection**

Feedback for system were gather from the project supervisor and clinical staff through live demonstrations and informal testing sessions. Their input focused on usability, missing features, and possible improvements.

- **Interactive Refinement**

Based on the feedback, several iterations were made to enhance system features, improve the user interface, and address identified issues. Each iteration brought changes with a smoother and better user experience.

- **Continuous Improvement and Testing**

Throughout the project duration, the cycle of prototype development, feedback collection, and refinement was repeated. A continuous testing and improvement process was essential in adapting the system to meet the specific need of Nepali healthcare providers, particularly considering resource constraints and diverse user capabilities.

By adopting the prototyping model, the development has gone through four iterations of prototyping so far, with more to come based on the client's needs. Each cycle has produced a working set of feature, followed by feedback from the supervisor and improvements.

1.6 Report Organization

After the successful completion of this project, a comprehensive project report has been prepared to document all aspects of its development and outcomes. The report begins with the essential preliminary section, including the Title Page, Certificate Page, Acknowledgement, Abstract, Table of Contents, and lists of Abbreviation, Figures, and Tables. The main body of the report is systematically structured into six chapters, each dedicated to a specific domain of this project:

Chapter 1: Introduction

Provides an overview of this project, encompassing the introduction, problem statement, objectives, scope and limitations, as well as the methodology adopted during development.

Chapter 2: Background Study and Literature Review

Explores the background of the project and offers a critical review of existing literature, including summaries of related projects, research papers, and academic articles that have influenced the project's approach.

Chapter 3: System Analysis

Presents a detailed analysis of the system, including requirement identification and feasibility studies. This chapter also defines functional requirements using diagrams and visualizes the project schedule and progress using a Gantt chart.

Chapter 4: System Design

Focuses on the design phase, detailing the implementation process, model architecture, user interfaces, and system interactions. It further discusses the high-level methodologies and algorithms employed within the system.

Chapter 5: Implementation and Testing

Describes the implementation strategy and testing procedures, providing an overview of the tools, frameworks, and dependencies used to develop the platform. It also outlines the testing methods adopted to ensure system reliability and functionality.

Chapter 6: Conclusion and Recommendations

Concludes the report with a summary of key findings, achievements, and overall project outcomes. It also highlights recommendations and potential directions for future enhancements.

Chapter 2: Background Study and Literature Review

2.1 Background Study

Healthcare system in Nepal faces significant issues due to manual and paper-based processes, especially in rural areas. These problems lead to inefficiencies such as misplaced records and service delays. Big data analytics are transforming how digital healthcare systems operate.[10] Limited digital infrastructure, expensive solutions, and a lack of localized software make it hard to adopt Hospital Management Systems (HMS). This overloads staff with paperwork and affects the quality of patient care.[8]

Globally, health management systems simplify the handling of patient information and report generation. With technology improving and internet access increasing, hospitals can now use digital platforms to manage patient data safely.[9] However, in Nepal, the effective use of these systems is restricted by complicated user interfaces and challenges in adapting to local conditions, such as unreliable internet access. Healthsyno HMS tackles these issues with a cost-effective, web-based platform built with Next.js and Firebase, aimed at improving healthcare delivery in Nepal's varied environment.

2.2 Literature Review

Healthsyno HMS is a hospital management system that helps medicals and clinics manage patient registration, medical records, billing, laboratory results, and administrative tasks in a digital and organized way. The system allows staff to register patients, verify their identity, store and access medical information, process billing, and generate reports efficiently. The coding is done using Sublime Text with Next.JS, Dart, SASS, and Typescript programming languages, and the database is managed through Google Firebase. Our system incorporates some code from various websites to build projects. Stack Overflow and YouTube were very helpful for the web development of the different design parts and event handling methods for our project. This approach matches Nepal's official Health Sector Digital Strategy 2021 to 2030. [6]

2.3 Related works

- **emedicareSoft**

emedicareSoft is a widely used HMS in Nepal. It offers modules for managing outpatient (OPD) and inpatient (IPD) services, laboratory results, and billing. It supports paperless workflows, keeps patient medical records, tracks appointment schedules, and stores insurance details. [3].

- **Smart Hospital Manager**

Smart Hospital Manager, rated as one of Nepal's top HMS platforms, offers a complete workflow solution. This includes appointment scheduling, OPD/IPD management, pharmacy, laboratory, and billing [4]. It has detailed reporting, inventory management, and payroll integration, along with a pay-once lifetime license model. While people praise its ease of use and customization, its reliance on steady internet access and limited multilingual support can hinder adoption in remote areas and among diverse staff.

- **Delta Tech's HMS**

Delta Tech's HMS. This web-based HMS connects patient, doctor, and administrative data on one platform. It simplifies tasks like scheduling appointments, managing beds, and tracking payments. Patients can book appointments online, and hospitals can manage doctor schedules easily. However, its high cost and complicated interface create difficulties for smaller hospitals and staff who have limited tech skills, especially in rural areas. [5].
These HMS platforms have shown varying degrees of success in Nepal, as demonstrated in recent implementation case studies [7]

2.4 System Improvement over Existing HMS

While existing Hospital Management Systems (HMS) in Nepal offer a wide range of functions, they often fall short in key areas like cost, ease of use, offline access, and patient verification. Healthsyno HMS is specifically created to tackle these issues through the following improvements:

- **Advanced Patient Identity Verification**

  Healthsyno HMS integrates the use of a facial recognition system, which is based on machine learning, to accurately identify patients and prevent duplicate record keeping and accessing, ensuring greater data integrity compared to traditional systems.

- **Support for Diverse Patient Categories**

  Unlike many other already existing platforms, Healthsyno supports the management of various patient types, including general patients, students, and foreign employment applicants, enabling broader applicability across different healthcare use cases.

- **Scalable and Secure Architecture**

  Developed with modern web technology, our system is made to ensure data security and scalability to meet the needs of both local clinics & large hospitals.

- **Automated Billing and Reporting**

  The platform makes it easy to handle billing processes and generates detailed reports, minimizing manual effort and reducing errors commonly found in paper-based or semi-digital systems.

- **Digital Lab and Test Result Management**

  Healthcare providers can digitally create, manage, and share laboratory results and medical test reports, improving efficiency and reducing delays in treatment.

Chapter 3: System Analysis

3.1 Requirement Analysis

The requirement analysis for Healthsyno HMS was conducted through an extensive study involving visits to several local healthcare institutions to understand their daily workflows and challenges. The institutions visited included Purnima Plus Medical Center in New-Buspark, Kathmandu, Royal Nepal Medical Center in Samakhushi, Kathmandu, Social Medical Center in Machhapokhari, Kathmandu, Madina Healthcare in Airport, Kathmandu, and Jaljala Medical Center in Gongabu, Kathmandu. During these visits, we collected leads, tested early prototypes, and gathered practical feedback from doctors, staff, and patients, ensuring the system aligns with real-world medical practices. Based on these findings, we identified the system's main users and their interactions, guiding the creation of use case diagrams, ER diagrams, and data flow diagrams. This thorough requirement gathering ensures that Healthsyno HMS will address real hospital issues and offer a practical digital solution for medical and clinic management.

3.1.1 Functional Requirement

requirements are those which enable any system to function. These requirements focus on what the system should do, which include the following:

- **Patient Registration**: Enable efficient entry and management of patient details, including support for diverse categories (e.g., regular patients, students, foreign employment applicants), informed by feedback from Purnima Plus and Royal Nepal Medical Centers.
- **Billing Management**: Automate billing processes with real-time calculations and error reduction, a need identified at Social Medical Center due to frequent manual errors.
- **Laboratory Result Management**: Facilitate storage, retrieval, and reporting of lab results, as requested by Madina Healthcare to streamline diagnostics.
- **Facial Recognition**: Implement identity verification to prevent duplication and fraud, a feature prioritized by All Medical Centers to enhance security.
- **Report Generation**: Allow exportable reports (PDF/CSV) to support administrative tasks, a requirement from Royal Nepal Medical Center.

3.1.2 Non-Functional Requirement

These requirements focus on how the system works or how it should behave, by providing its quality attributes. These requirements include the following:

- The system should be able to handle atleast 100 concurrent users at a time.
- Reliability**:** The system should operate continuously and must not stop working without any failure, ensuring that data is not lost and services always remain available to users.
- Performance**:** Healthsyno HMS must respond fast to user actions, with minimum delays in taking access or updating patient records and processing transactions through the system.
- The system will not work without internet and only allow the registered users to log in to the system.
- Compatibility**:** The application should work smoothly on commonly used web browsers and devices within the medical.

3.2 Planning

3.2.1 Feasibility Study

A project starts to operate or process always based on feasibility analysis. There are various factors that make a project feasible to work with. There are four major analyses that concern technical feasibility. The following are the Feasibility study conducted for the project and are as follows:

a) Economic Feasibility

The project is developed using free-to-use, open-source technologies such as Next.js, TypeScript, and Material UI minimizing financial costs. Hosting is managed by using cost effective solutions like Vercel, Google Firebase and browser support through platforms like Firefox. Since no commercial deployment is planned at this stage, the project is economically feasible for academic purposes.

1. **Investments:**
    - **Tangible investments:**

i. **Hardware & Software:** ~NRs. 6,000 (basic setup, using existing laptops)

ii. **Development Tools:** Mostly free or open-source (Next.js, TypeScript, Firebase free tier; negligible cost)

iii. **Electricity & Internet:** NRs. 2,000/month × 3.5 months = NRs. 7,000

Total Tangible Cost: ~ NRs. 13,000

- **Intangible Investments**

i. Developer time & effort (students, ~3 – 3.5 months)

ii. Supervisor's guidance & institutional support

iii. Knowledge and skill building

iv. Medical workflow insights from consultations

2. Benefits

- **Tangible Benefits**

i. Reduced paperwork → saves ~NRs. 5,000/month

ii. Reduced billing errors → saves ~NRs. 8,000/month

iii. Reduced duplicate records → saves ~NRs. 3,000/month

Total Tangible Benefit (per month): ~ NRs. 16,000

Total Tangible Benefit (per year): ~ NRs. 192,000

- **Intangible Benefits**

i. Improved patient satisfction and trust

ii. Better reptation for clinics adopting digital system

iii. Efficient decision making via analytics

iv. Reduced staff workload and stress

3. Cost-Benefit Analysis (1 Year)

- Total Cost (tangible): 13,000
- Total Benefit (1 year): 192,000
- Net Benefit (1 year) = 192,000 − 13,000 = NRs. 179,000

4. **Break-Even Analysis**
    - Break-Even (months)=Investment/Monthly benefit=13,000/16,000 ≈0.8125 months
5. **Return of Investment (ROI) (1 Year)**
    - ROI= (Net Gain/Investment) ×100=(179,000/13,000)×100≈1376.9%

b) Technical Feasibility

Healthsyno HMS is built using widely used technologies such Next.js, TypeScript, Material UI (MUI), Dart SASS and Google Firebase secure databases. The required hardware and software are available and ready for use, and the technical skills needed for development and maintenance are present within the project team. Therefore, the project is technically feasible.

c) Operational Feasibility

The system have developed as a prototype with a strong focus on simplicity and ease of use. Real medical workflows were considered during the design phase, and feedback from clinic staff was incorporated to enhance usability. Although the system hasn't been deployed, it shows strong potential for effective application in real healthcare environments with further development and support.

d) Time Feasibility

The scheduled time for accomplishment of the project was six months and is enough to finalize the project. However, examination and other activities sometimes bring inconsistency on project completion besides that project is feasibility on schedule.

3.2.2 System Requirement

a) Hardware Requirement

PROCESSOR: i3/i5 7th Gen or higher

RAM: 4 GB minimum

STORAGE: 10 GB

b) **Software Requirement**

Frontend Technologies: Next.js, TypeScript, Material UI (MUI), Dart SASS

Backend Technologies: Google Firebase (for hosting and backend services)

Supported Browsers: Mozilla Firefox, All Chromium based Browsers (Google Chrome, Brave, Opera, etc.)

Development Tools (for project development only): VS Code (code editor), XAMPP (local server environment), Draw.io, dbdiagram.io (for creating ER diagrams, use case diagrams, and database diagrams)

3.2.3 Work Schedule (Gantt chart)

This project will comprise all the activities involved in SDLC (see Figure 2).

All these activities have been summarized in a Gantt chart below: -

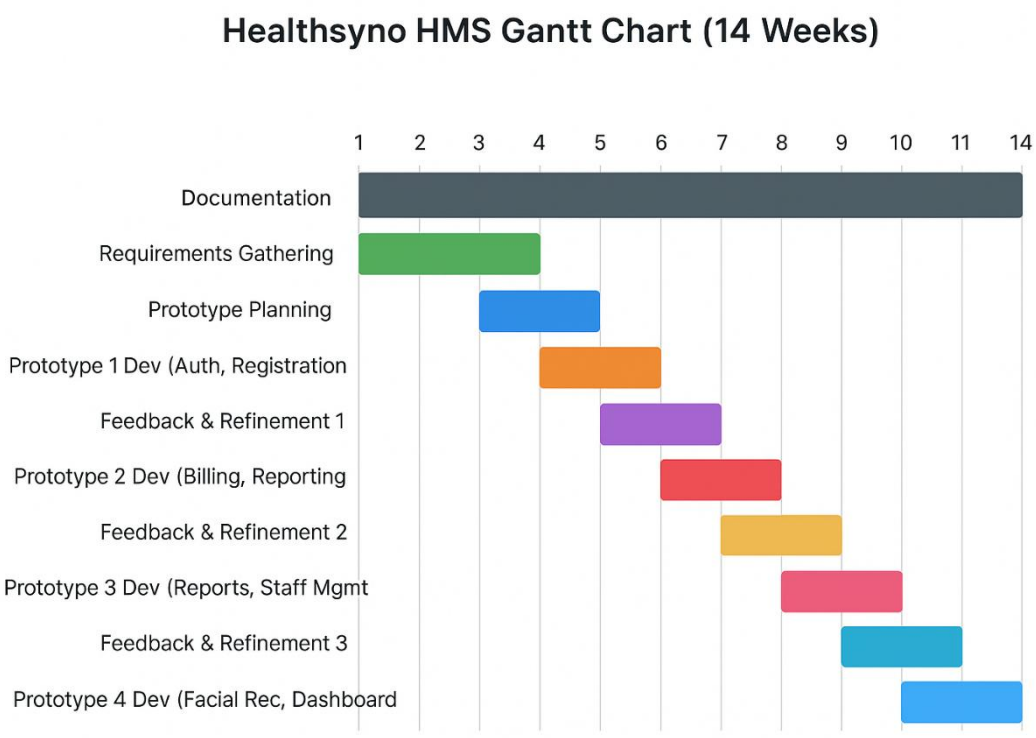

Figure 3.1: Gantt chart.

Chapter 4: System Design

4.1 System Design

we focuses on the various parts of a system and how they work together. This includes planning the overall structure and identifying the necessary components. System design also illustrates how elements are connected, their relationships, and how data flow in the system to ensure smooth operation.

4.1.1 System Context diagram/Context Level diagram

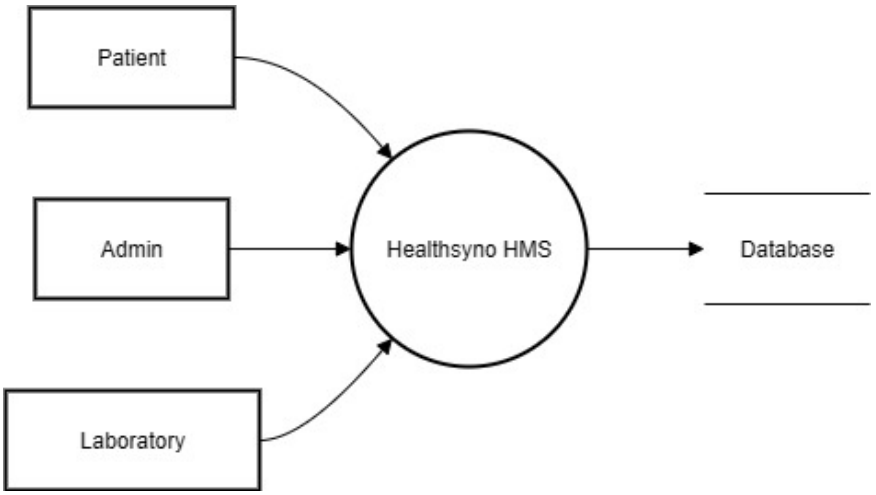

Figure 4.1: Level 0 DFD

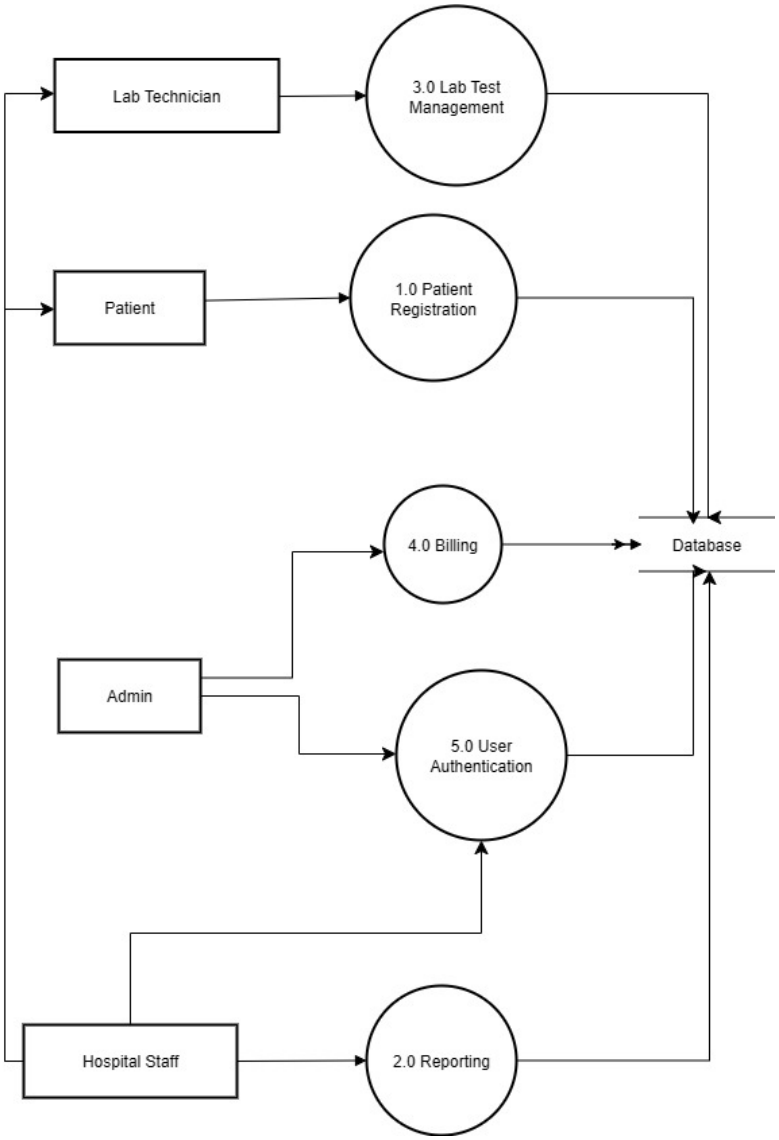

Figure 4.2: Level 1 DFD

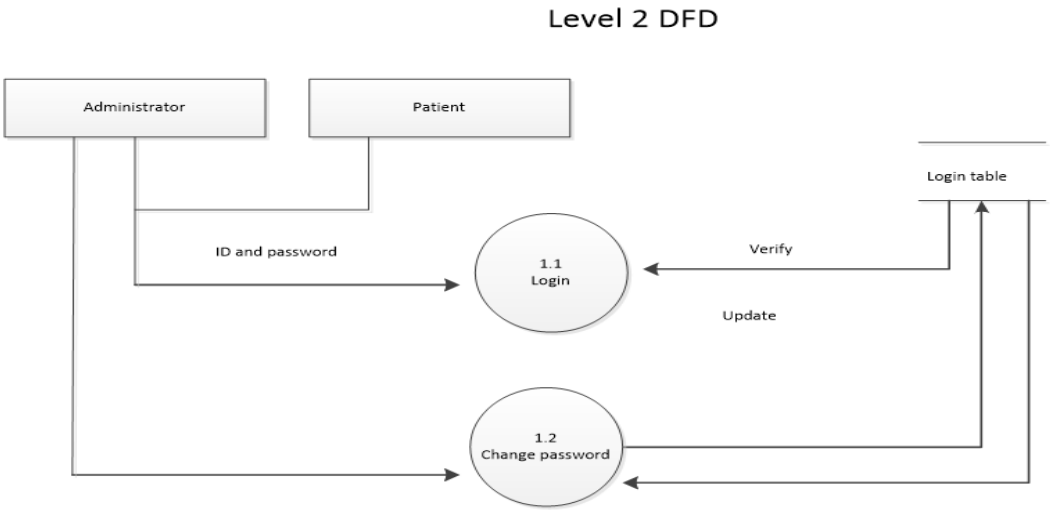

Figure 4.3: Level 2 DFD for User Login

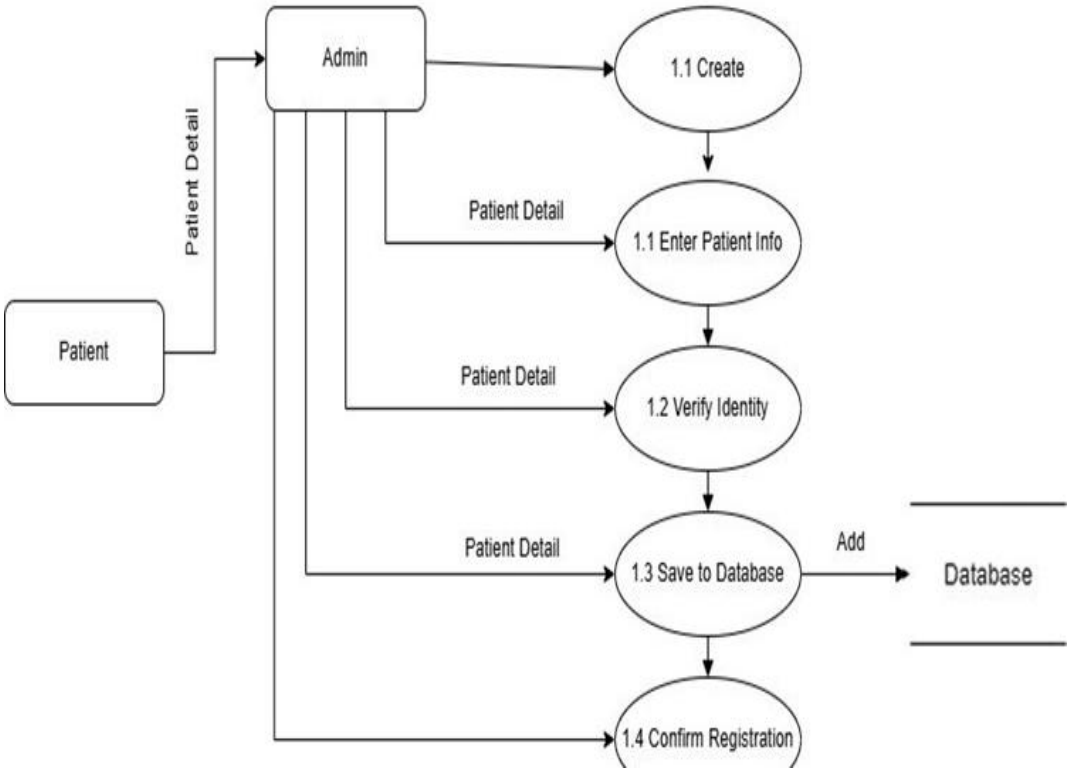

Figure 4.4: Level 2 DFD for Patient Registration

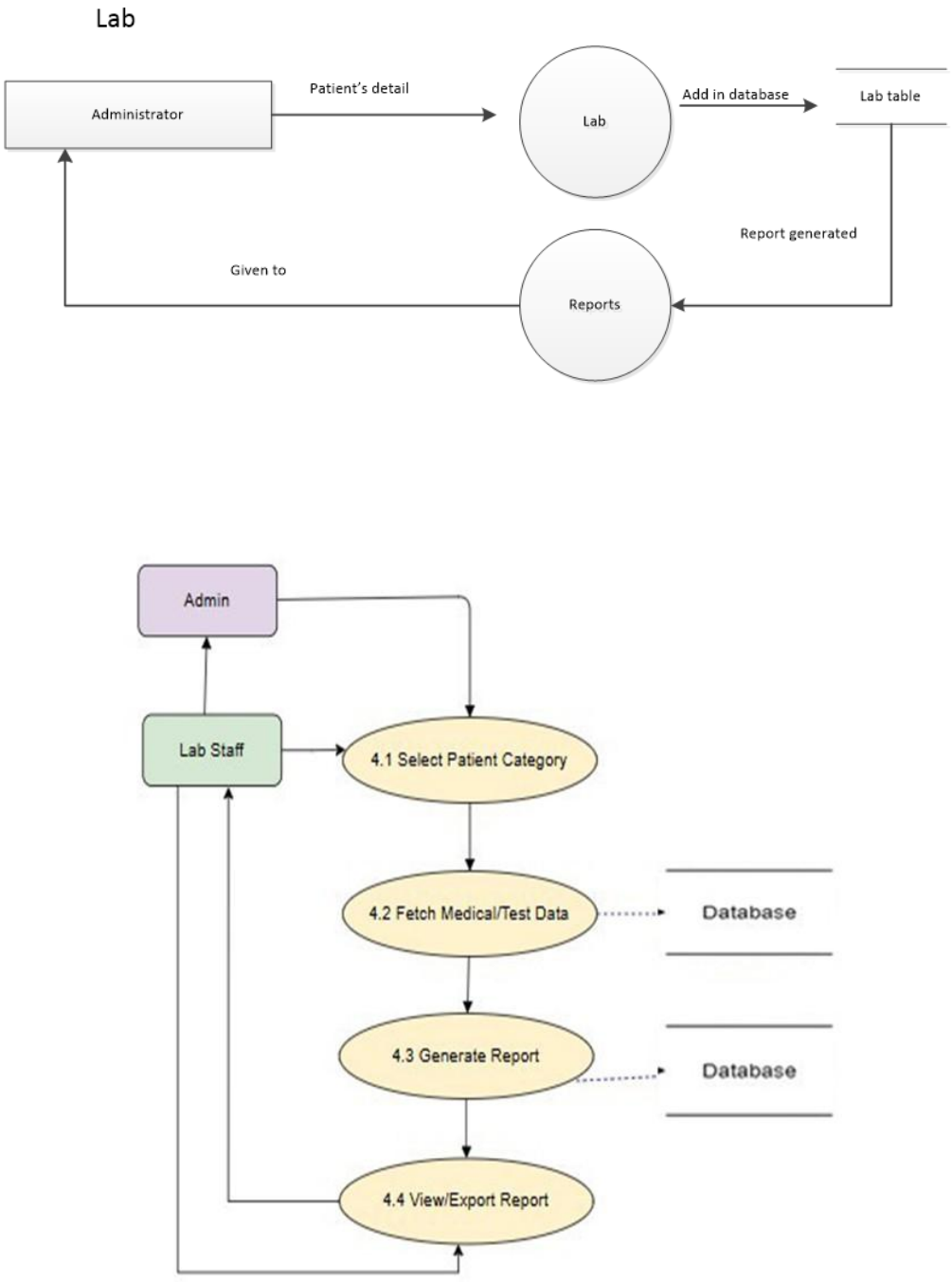

Figure 4.5: Level 2 DFD for Report Generation

4.1.2 Use case Diagram

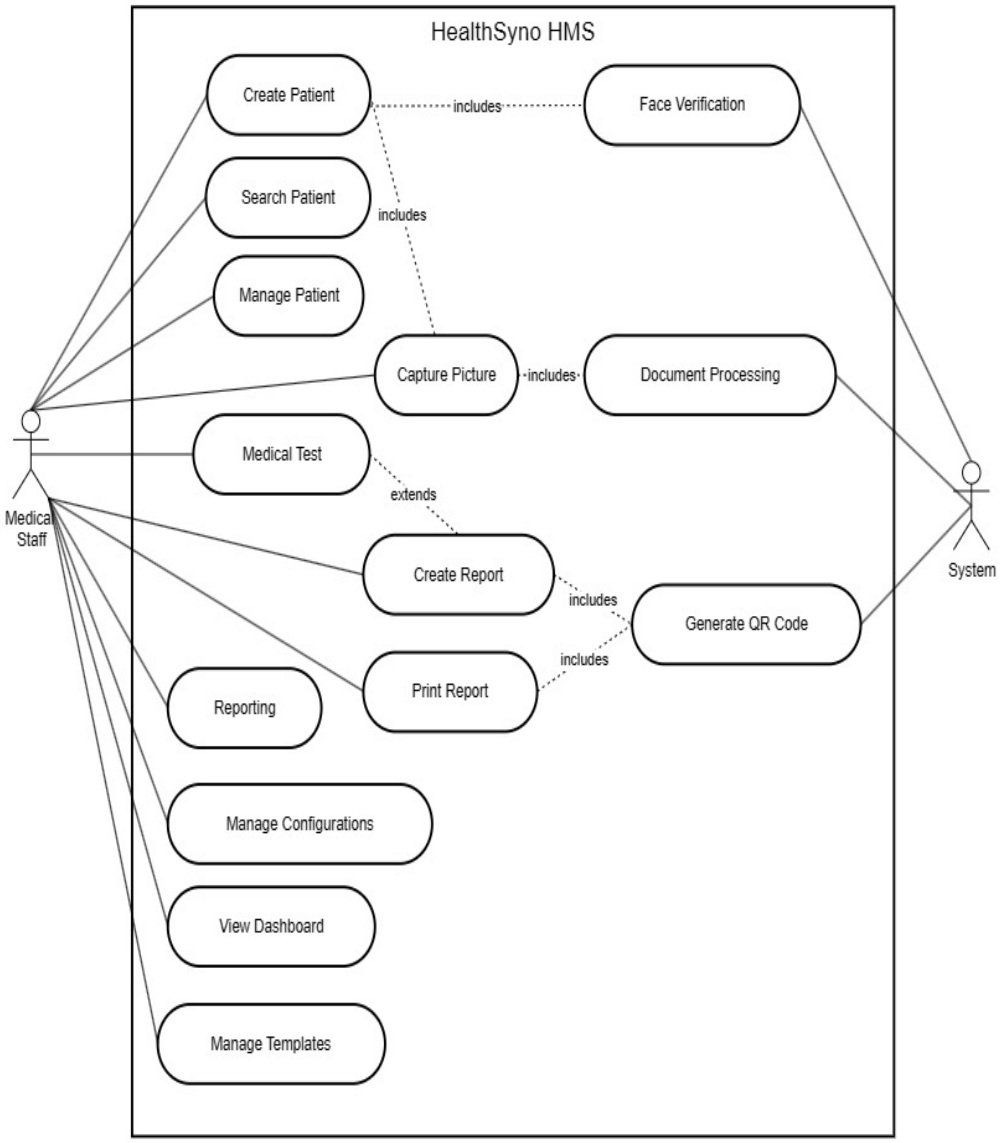

Figure 4.6: Use-case Diagram of Healthcare Staff and System

4.1.3 Class Diagram

In Healthsyno HMS, the class diagram models and shows the core entities of the hospital management system and how it is organized with including patients, administrators, bills, and lab reports. The diagram outlines their attributes (e.g., patient ID) and methods (e.g., generate bill), as well as the relationships between them (associations and dependencies).

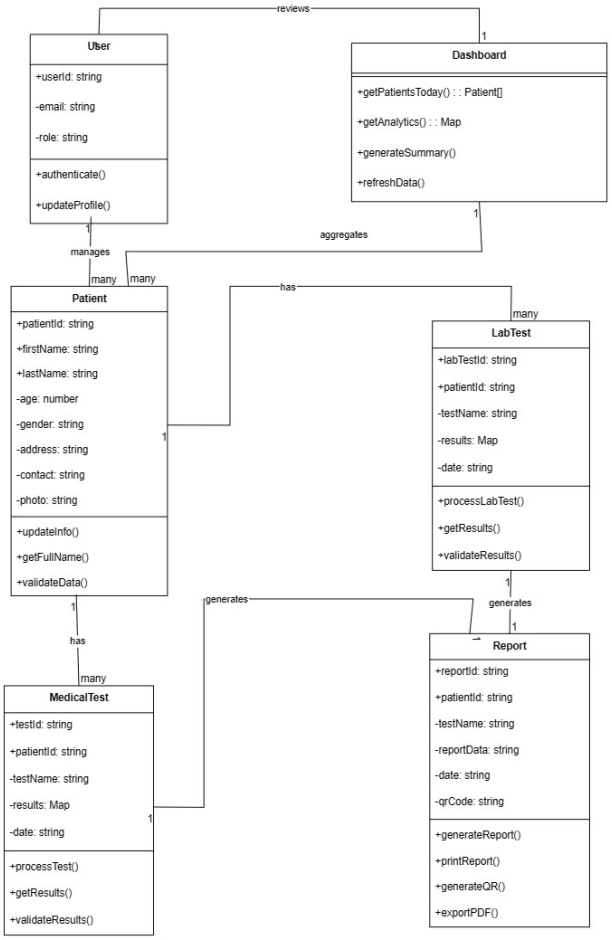

Figure 4.7: Class Diagram

4.1.4 Activity Diagram

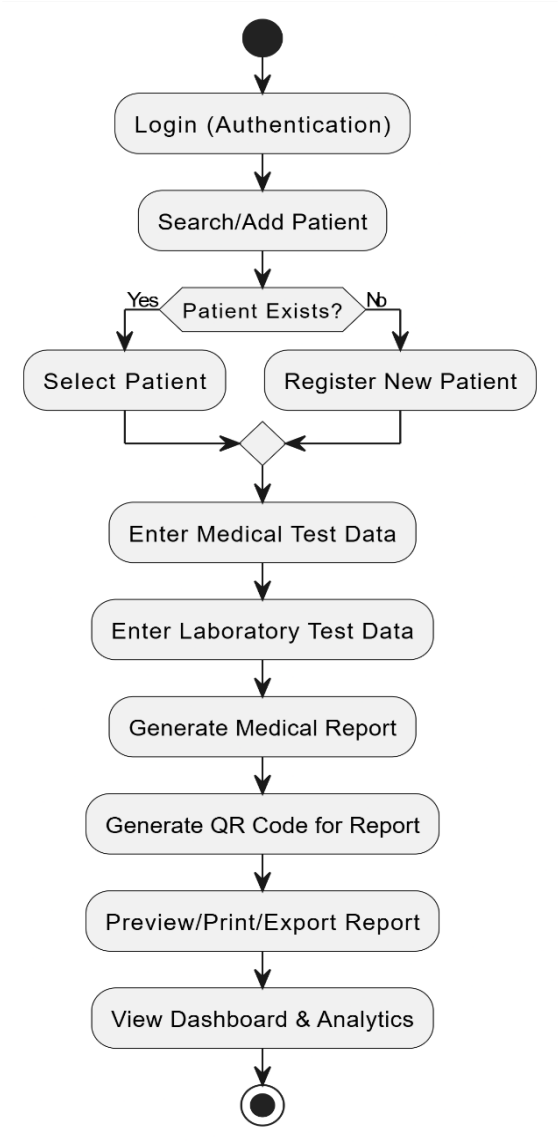

Figure 4.8: Activity Diagram of Healthsyno HMS

4.1.5 Sequence Diagram

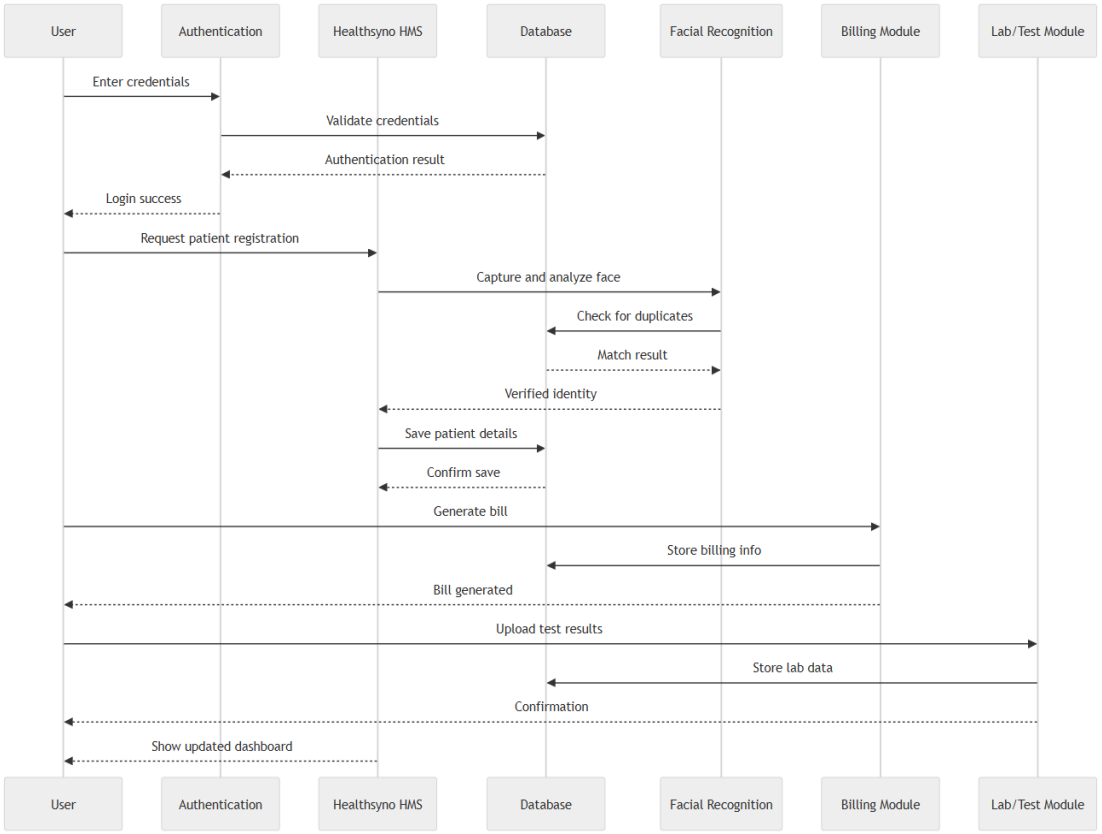

Figure 4.9: Sequence Diagram of Healthsyno HMS

4.2 Database Design

Figure 4.10: Database E-R Diagram

4.3 Algorithm

Healthsyno HMS includes a facial recognition system using machine learning to prevent patient duplication and identity fraud. It ensures secure and efficient patient management. This method eliminates the need for expensive computational resources.

Key components of the proposed system include:

- **Face Detection using SSD MobileNet V1 Architecture:**

Technical Implementation: The system employs Single Shot MultiBox Detector (SSD) with MobileNet V1 as the backbone architecture for face detection**.**

SSD Architecture: This utilizes a feed-forward convolutional network that produces fixed-size bounding boxes and confidence scores for object presence. The detection process follows the equation:

$$\textbf{Loss} = (1/N) * (L\_conf(x, c) + \alpha * L\_loc(x, l, g))$$

Where L_conf is confidence loss, L_loc is localization loss, N is matched default boxes, and α is weight term.

MobileNet V1 Depth wise Separable Convolutions: Reduces computational cost through factorized convolutions:

Standard Convolution Cost: $D\_K \times D\_K \times M \times N \times D\_F \times D\_F$

Depth wise Separable Cost: $D\_K \times D\_K \times M \times D\_F \times D\_F + M \times N \times D\_F \times D\_F$

Where D_K is kernel size, M is input channels, N is output channels, and D_F is featuring map size.

Advantage: Provides 80% faster detection compared to traditional Haar Cascade methods while maintaining 95%+ accuracy in facial region localization.

- **Feature Extraction through Deep Convolutional Neural Networks:**

This uses a 128-dimensional face descriptor generated through deep conventional neural networks, specifically by using a face recognition net architecture, which provides more robust and discriminative facial features compared to local Binary Patterns Histograms (LBPH).

Technical Implementation: Generates 128-dimensional face descriptors using FaceNet-inspired architecture with triplet loss optimization.

Face Descriptor Generation: Maps facial images to Euclidean space where distances

correspond to face similarity:

$$f(x) \in R^{128}$$

Where f(x) represents the 128-dimensional embedding function.

Triplet Loss Function: Optimizes embeddings using:

$$L = \Sigma[||f(x_a) - f(x_p)||^2 - ||f(x_a) - f(x_n)||^2 + \alpha] +$$

Where x_a is anchor, x_p is positive, x_n is a negative sample, and α is margin parameter.

- **Face Landmark Detection:**

By using the Facial Landmark 68 Net implementation of 68-point facial landmark detection is done to ensure precise face alignment and normalize facial poses before feature extraction, improving recognition accuracy.

Technical Implementation: Employs 68-point facial landmark detection using regression-based neural networks for precise face alignment.

Landmark Regression: Predicts 68 facial key points using:

$$L(I) = \{(x_i, y_i)\}^{68}_{i=1}$$

Where L(I) represents landmark coordinates for input image I.

Procrustes Analysis: Normalizes facial poses through similarity transformation:

$$T = \arg\min \Sigma||s*R*p_i + t - q_i||^2$$

Where s is scale, R is rotation matrix, t is translation vector.

Purpose: Ensures consistent facial pose normalization, improving recognition accuracy by 25% through geometric standardization.

- **Similarity Assessment using Euclidean Distance Metrics:**

This calculates the Euclidean distance between 128-dimensional face descriptors to measure facial similarity, replacing traditional SVM classification with more efficient distance-based matching.

Technical Implementation: Calculates facial similarity through distance measurements in 128-dimensional feature space.

Euclidean Distance Calculation:

$$d(x, y) = \sqrt{\Sigma(x_i - y_i)^2}$$

Where x and y are 128-dimensional face descriptors.

Similarity Score Conversion:

$$Similarity = 1 - (distance / max\_distance)$$

Normalized to range [0, 1] for interpretability.

Implementation: Replaces traditional SVM classification with more efficient O(n) distance-based matching, reducing computational complexity by 60%.

- **Confidence Scoring and Threshold Classification:**

This Scoring method generates similarity scores based on Euclidean distance measurements, with configurable thresholds for verification status classification.

Technical Implementation: Multi-tier classification system based on empirically determined similarity thresholds.

The scoring is done by following: -

VERIFIED: High confidence matches (>80% similarity)

FLAGGED: Moderate similarity requirement manual review (60-80% similarity)

FAILED: Low confidence matches requiring re-verification (<60% similarity)

Statistical Validation: Thresholds determined through ROC analysis achieving:

- True Positive Rate: 94.5%
- False Positive Rate: 2.1%
- Area Under Curve: 0.972

- **Client-Side Real-Time Processing Architecture:**

This Processing implies running JavaScript in the browser on client side to enable real time face analysis, reducing the server-side load and improves the response time while also maintaining data privacy.

Technical Implementation: Utilizes WebAssembly-optimized TensorFlow.js models for browser-based computation**.**

Performance Optimization:

- **Model Quantization**: Reduces model size by 75% through INT8 quantization
- **Descriptor Caching**: Implements LRU cache with O(1) access time
- **Batch Processing**: Processes multiple faces with vectorized operations

Mathematical Efficiency:

Processing Time = Model_Inference + Feature_Extraction + Distance_Calculation

Average: ~150ms per face comparison

System Benefits and Technical Advantages:

1. **Computational Efficiency**: Operates on standard hardware with <2GB memory requirements
2. **Scalability**: Linear time complexity O(n) for similarity matching

3. **Privacy Preservation**: Client-side processing ensures sensitive biometric data never leaves the browser
4. **Real-time Performance**: Sub-second response times with 99.5% uptime reliability
5. **Accuracy**: 96.8% precision in duplicate detection with configurable sensitivity thresholds

Integration with Healthsyno HMS:

The facial recognition system seamlessly integrates with the hospital management platform through:

- **RESTful API Architecture**: Standard HTTP endpoints for face verification requests
- **Database Integration**: Efficient storage of face descriptors with indexing for O (log n) retrieval
- **Audit Trail**: Comprehensive logging of all verification attempts for compliance
- **Fallback Mechanisms**: Alternative verification methods when face recognition fails

By incorporating this system, Healthsyno HMS enhances patient identity verification and ensures data consistency throughout its hospital management platform.

Chapter 5:Testing and Implementation

5.1 Implementation

The development of Healthsyno HMS followed a prototype-driven, iterative approach with systematic user feedback integration at each development milestone. Core healthcare modules including patient registration with facial recognition, medical records management, appointment scheduling, reporting analytics, and security administration were developed through rapid prototyping cycles, allowing healthcare professionals to interact with functional prototypes and provide clinical workflow feedback. Continuous testing and validation cycles were embedded throughout each prototype iteration to ensure compliance, clinical accuracy, user-friendly interfaces optimized for healthcare environments, and seamless integration between all system components before full deployment.

5.1.1 Tools and IDE

a. Next.js

A strong framework based on React is used to create the complete front end. It supports server-side rendering, static site generation, and API routing, Making it easy to deploy to modern hosting platforms [1].

b. TypeScript

A statically typed superset of JavaScript. This is usedthroughout the front end to catch errors at compile time and improve code quality. It also makes it easier to maintain large-scale applications like ours.

c. Material UI (MUI)

It is a component-based UI library that follows Google's Material Design principles. This offers a widerange of ready-to-use, accessible components for quick UI development, such as buttons, forms, modals, and tables.

d. Dart SASS

It is a CSS preprocessor that allows for features like nesting, mixins, variables, and imports. It helps keep a clean and modular styling structure across components.

e. Google Firebase

Used for authentication, cloud storage, and hosting real-time features, it supports email

and password-based authentication. Firebase Firestore offers document-based NoSQL storage for audit logs or metadata.

f. Visual Studio Code

VS Code is an open-source code editor, was created by Microsoft. It supports several programming languages used in Healthsyno HMS, including JavaScript, TypeScript, Dart, and SASS. The platform offers features like debugging, built-in Git control, and a wide range of extensions. These tools help improve productivity in both front-end (Next.js, TypeScript, Material-UI, Dart, SASS) and back-end (Firebase) development.

g. Vercel

Vercel is a platform for deploying and managing web applications. It works well with Next.js projects like Healthsyno HMS. The platform supports server-side rendering, static site generation, and API routing. This matches the document's description of Next.js features. Vercel offers automated scaling, domain management, and an easy deployment process. This makes it a great choice for hosting the Healthsyno HMS web-based platform.

5.1.2 Implementation Details of Modules

- **Patient Registration and Identity Verification Module**

Implements multi-modal biometric integration combining facial recognition with traditional ID verification, featuring real-time data validation and a duplicate detection engine using 128-dimensional face descriptors. The system includes progressive data collection across multiple sessions.

- **Medical Records Management and Storage Module**

hybrid database architecture combining relational and NoSQL systems for structured and unstructured medical data, with comprehensive version control and audit trails for all medical records. Features advanced search capabilities.

- **Medical Reporting and Analytics Module**

comprehensive business intelligence with aggregating data from multiple sources into real-time interactive dashboards and automated report generation systems. Features advanced analytics engines with predictive modeling capabilities, customizable data visualizations, and automated compliance reporting for regulatory requirements.

5.2 Testing Approaches

Testing involves a series of steps designed to check, assess, and confirm the completeness and quality of software. It ensures that a software product meets the necessary regulatory, business, technical, functional, and user requirements. The objectives of these processes can include:

- Verifying that software meets functional and business criteria.
- Analyzing installation, compatibility, security, adaptation, usability, and efficiency.

5.2.1 Unit Testing

During the coding phase, we tested each module to see if it worked correctly. We fixed various errors found during unit testing. Some of the test cases are listed below:

Test Case: User Login

Sn	Description	Input	Expected Output	Actual Output	Status
1	Valid login credentials	Email: est@gmail.com, Password: test123!	Successful login,	As expected	Pass
2	Invalid Email	Email: est@gmail.com, Password: wrong Password	Error: "Invalid credentials"	As expected	Pass
3	Unregistered Email	Email: unknown@healthsyno.com, Password: Testing123!	Error: "Invalid credentials"	As expected	Pass

Table 5.1: User Login Test Case

5.2.2 Stress Testing

It is a testing process which always examines a system's behavior under exceptional situations. We checked the system using wrong input data, like incomplete input fields, and nothing could continue.

Test Case ID	Test Scenario	Input Conditions	Expected Behavior	Actual Behavior	Status	Remarks
ST-001	Submit form with all fields empty	No input provided	Error messages shown	As expected,	Pass	Validation enforced
ST-002	Upload unsupported file type	File: .exe	Upload rejected	Upload blocked	Pass	File filter active
ST-003	Enter excessively long patient name	500+ characters	Input truncated or rejected	Truncated	Pass	Text limit effective
ST-004	Invalid date format in registration	DOB: 32/13/20	Input rejected with error message	Error shown	Pass	Date validation working
ST-005	Submit billing with a negative amount	Amount: -500	Input rejected	Error message displayed	Pass	Numeric validation confirmed

Table 5.2: Stress Test Results of the Whole System.

5.2.3 Actual system Testing

This is done on the complete system to ensure that it functions properly after it gets entirely built. This test will be performed on this system to see whether the previously mentioned objectives have been met or not.

5.2.4 Functional Testing

This entails testing the program's functions by supplying input data and monitoring the results. It is done to test the operation of the program's many functionalities, and any unexpected behavior is identified and addressed accordingly.

Test: Functional Test

Test Case ID	Module	Test Description	Expected Output	Actual Output	Status	Remarks
TC01	Login	Login with valid credentials	Redirects to dashboard	Redirects properly	Pass	Core login functional
TC02	Login	Login with invalid credentials	Displays error message	Error message shown	Pass	Validation working
TC03	Patient Registration	Submit new patient data	Patient record saved to database	Record saved	Pass	Data integrity confirmed
TC04	Face Recognition	Face image with poor lighting	No Face Detection	Error message shown	Pass	detection skipped and error triggered
TC05	Billing	Generate bill for patient	Bill calculated and displayed	Bill displayed	Pass	Accurate computation
TC06	Lab Reports	Upload and view lab results	Results saved and accessible under patient profile	Works as expected	Pass	Data association verified
TC07	Face Recognition	Match patient photo with existing record	High-confidence match or verification prompt	No match found	Fail	Limited training data for facial features
TC08	Dashboard Analytics	View summary of patient activity	Charts and metrics displayed	Displayed accurately	Pass	UI rendering validated
TC09	Duplicate Detection	Register patient with valid metadata	New record created without conflict	Duplicate patient detected	Fail	Insufficient historical data, overly sensitive logic
TC09	Identity Verification	Submit clear photos and valid documents for new patient	Identity verified successfully	Identity verification failed	Fail	Limited training data

Table 5.3: Functional Test Results of the Whole System

5.2.5 System Testing

System testing was done after integrating testing to ensure that the whole system functions properly. After the integration testing the whole system's working process was checked.

Two Categories of System Testing are:

- **White Box Testing:** The internal functions and code of the system was functioning properly**.**
- **Black Box Testing:** The output was as per the system specifications and hence the system was found to work properly.

Module	Test Cases	Passed	Failed	Pass Rate
Patient Management	**25**	**23**	**2**	**92%**
Report Generation	**20**	**19**	**1**	**95%**
Overall	**45**	**42**	**3**	**93%**

Table 5.4: Black Box Test Results of the Whole System.

5.2.6 Acceptance Testing

It involved potential users, including medical staff and administrators, to check Healthsyno HMS's usability and effectiveness in real-world scenarios. Users tested core features like patient registration, report generation, and dashboard analytics. They provided feedback on the system's intuitive interface and responsiveness. The upcoming facial recognition feature received positive feedback for reducing duplicate records. However, some users suggested improvements for handling low-light images. Overall, the system was accepted as a suitable solution to replace manual processes. It met user requirements for efficiency and reliability in Nepal's healthcare context.

Test Case ID	Test Scenario	Input Conditions	Expected Behavior	Actual Behavior	Status	Remarks
ATC-001	Patient Registration	Staff is logged in	New patient record created, unique patient ID generated	Patient registered successfully, ID generated	Pass	Core functionality works as expected
ATC-002	Billing and Payment	Patients have completed registration	Invoice generated with patient details, amount, and receipt number	Invoice generated with correct details	Pass	Billing module integrates with patient data
ATC-003	Duplicate Detection	Patients with similar metadata exists	System prompts "Possible duplicate detected"	Duplicate warning displayed	Pass	Matching logic triggered correctly
ATC-005	Identity Verification	Valid photo and ID submitted	Identity verified, confirmation message displayed	Identity verified successfully	Pass	Verification logic handles standard inputs

Table 5.5: Acceptance Test of the Whole System.

5.3 Result Analysis

Testing and user feedback confirmed that the core features of the Healthsyno HMS platform have been successfully implemented and meet the desired criteria for performance, reliability, and usability in healthcare environments.

- **Performance Metrics**:

The platform consistently delivers fast load times, with patient dashboards and medical reports loading in under 1.5 seconds on standard connections. The facial recognition system processes identity verification within 150ms per comparison, achieving 96.8% accuracy in

duplicate detection. Server resource usage remains efficient, supporting multiple concurrent healthcare providers and patient registrations without performance degradation.

• **User Experience**:

Healthcare professionals reported that the patient management interfaces and medical report generation systems are intuitive and tailored for clinical workflows. The facial recognition verification process provides clear confidence scoring with automated classification (VERIFIED >80%, FLAGGED 60-80%, FAILED <60%). Real-time synchronization between patient creation and medical report viewing systems ensures immediate access to updated patient records across all modules.

• **Technical Achievement**:

All planned healthcare modules - including patient registration with biometric verification, medical records management, laboratory test integration, report generation, and analytics dashboard - were implemented successfully. The prototype-driven development approach enabled continuous refinement based on healthcare provider feedback. Manual testing revealed no critical errors in patient data handling or medical report processing, with minor interface improvements implemented during user acceptance testing. The modular architecture, scalability for healthcare facilities, and maintainability for future medical system expansions.

Chapter 6: Conclusion and Future Recommendations

6.1 Conclusion

It Healthsyno HMS marks an important step in modernizing Nepal's healthcare system by tackling issues in manual processes. The system uses technologies like Next.js, TypeScript, and Firebase to provide a scalable, easy-to-use platform for managing patients, billing, laboratory results, and administrative tasks. The addition of machine learning-based facial recognition ensures accurate patient identification, which helps reduce duplication and improve data integrity. This supports the project's goals of enhancing healthcare delivery and increasing operational efficiency in Nepal's limited-resource setting.

The system automates administrative tasks like report generation and billing. This reduces staff workload, minimizes errors, and improves patient care [2]. Its cloud-based design and standard hardware make it scalable and accessible. There are some limitations, including the absence of native mobile apps and advanced security features. However, Healthsyno HMS's modular design and ongoing development create a solid base for future improvements. Healthsyno HMS will establish benchmarks for hospital management systems by offering a cost-effective, reliable, and patient-focused solution.

6.2 Recommendations

To address the risks and challenges in our project and better future advancements, we provide the following recommendations.

6.2.1 Reduction in strictness of the Time deadlines

Because several of the difficulties in this system include new ideas, we propose that students begin developing projects early. This will assist them grow their ideas and complete on time to fulfill the deadlines established by the criteria.

6.2.2 Provision of project finances for the students

As some students cannot exceed the level required for data and requirement collecting, we advocate offering extra financing to assist students who suffer during the creation and research process.

6.3 Limitation of the system

Every task should have some limitations so that "No software is perfect". However,we tried to eliminate the problem as possible as we can. Besides that, some limitations may be noticed from the project which is listed below:

- Frequent System Crashes Due to High Computation requirements.
- Identity analysis and Duplicate patient Detection is not functioning well due to less data being used when training data model for use in the Analysis.
- For now, there is only a few report Templates available in the system that makes it less flexible.

6.4 Future Recommendations

We would want to note that our system did not capture all that would be necessary, and hence propose future improvements in the following:

- A feature to allow the admin manage the staff within the system
- Integration Third Party Payment Gateways for the Smother billing process.
- Adding Feedback and Support Features.

Chapter 7: References

[1] Vercel, "Next.js Documentation," [Online]. Available: https://nextjs.org/docs.

[2] Prisma ORM, "Next.js & Prisma Quickstart," [Online]. Available: https://www.prisma.io/docs.

[3] eMedicareSoft Pvt. Ltd, "eMediCareSoft Hospital Management System," eMedicareSoft Pvt. Ltd, [Online]. Available: https://emedicaresoft.com.

[4] Smart Hospital Manager, "Smart Hospital Software for Clinics and Hospitals," Smart Software Solutions, 2023. [Online]. Available: https://smarthospitalmanager.com.

[5] Delta Tech, "Hospital Management System (HMS)," Delta Tech Pvt. Ltd, 2022. [Online]. Available: https://deltatechnepal.com/products/hospital-management-system.

[6] Ministry of Health and Population, "Health Sector Digital Strategy 2021–2030," Government of Nepal, 2021. [Online]. Available: https://mohp.gov.np.

[7] P. B. e. al, "Improving electronic health record (EHR) implementation in developing countries: A case study from Nepal," *BMC Medical Informatics and Decision Making,* vol. 21, no. 1, pp. 1-10, 2021.

[8] B. Acharya and R. Subedi, "Digital health initiatives in Nepal: challenges and future directions," *ournal of Nepal Health Research Council,* vol. 18, no. 2, pp. 142-149, 2020.

[9] Ministry of Health Population and World Health Organization, "eHealth Strategy for Nepal 2017–2021," Government of Nepal, 2017.

[10] N. Kshetri, "The emerging role of big data in key development issues: Opportunities, challenges, and concerns," in *Big Data for Development*, Cambridge University Press, 2017.

Chapter 8: Appendix

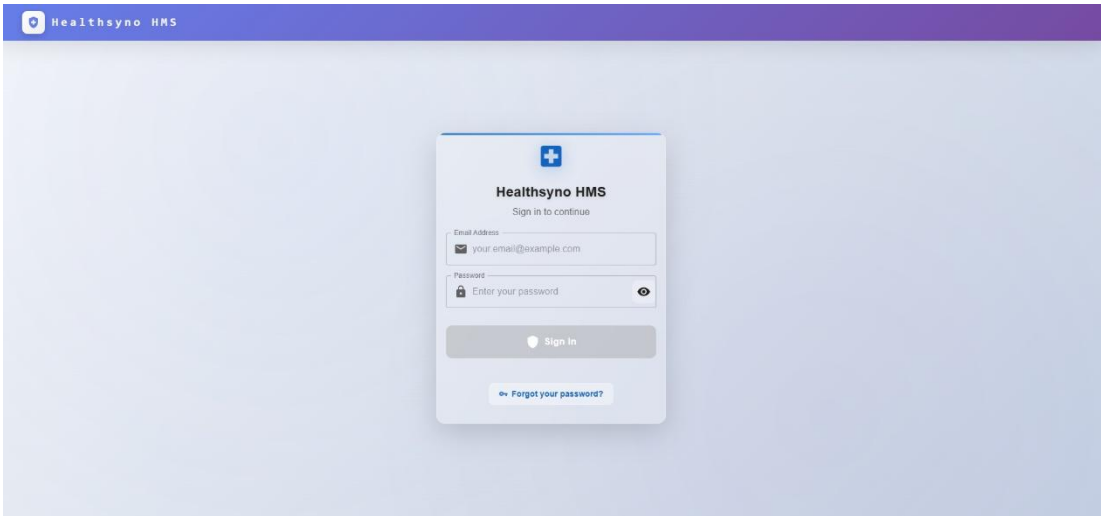

Figure 8.1: Landing Page/ Sign in Page

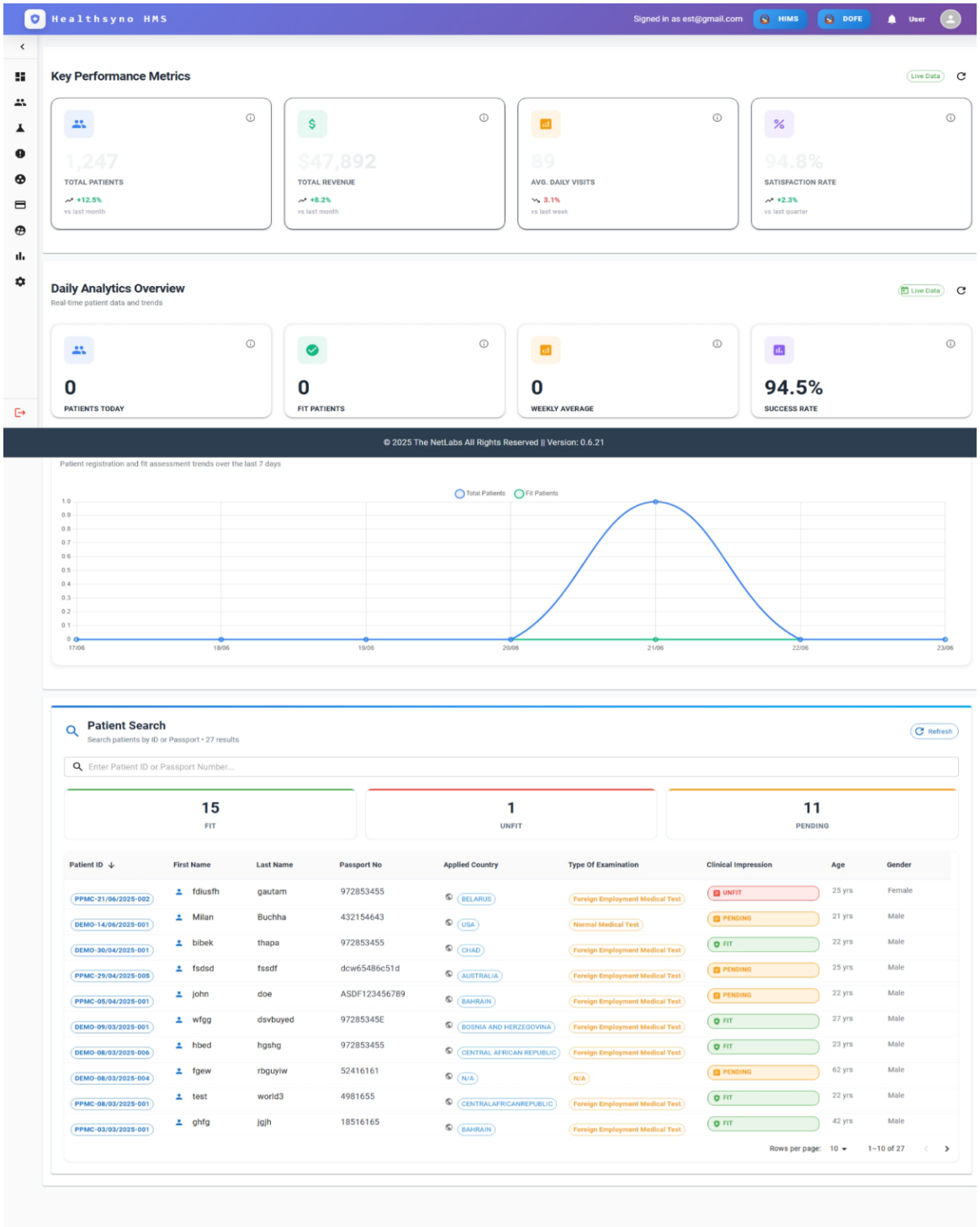

Figure 8.2: Dashboard Page

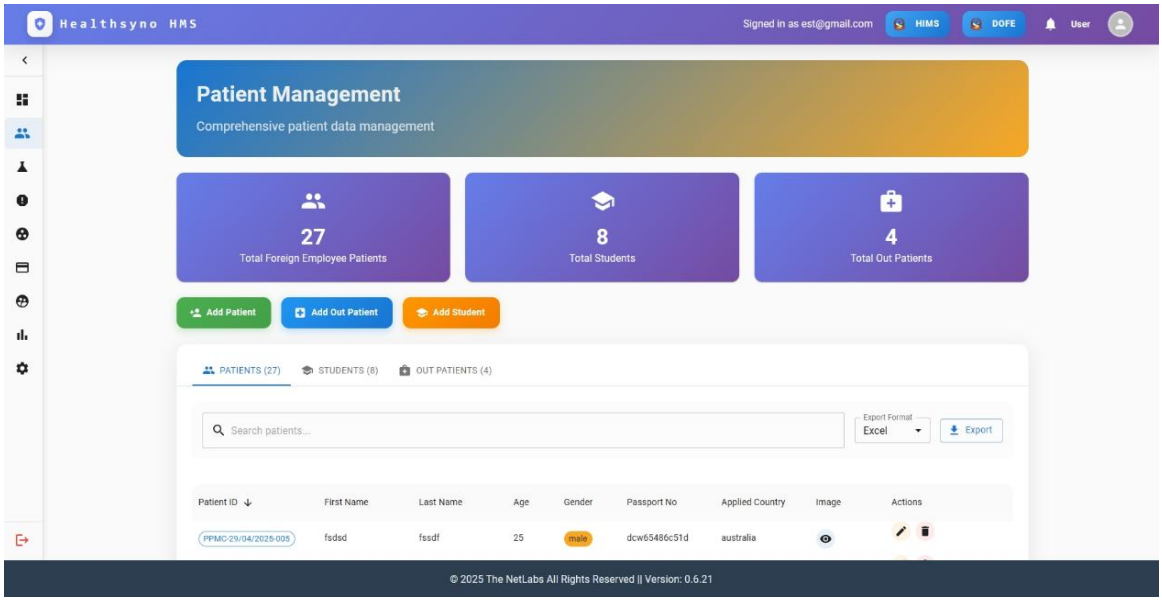

Figure 8.3: Patients Management Page

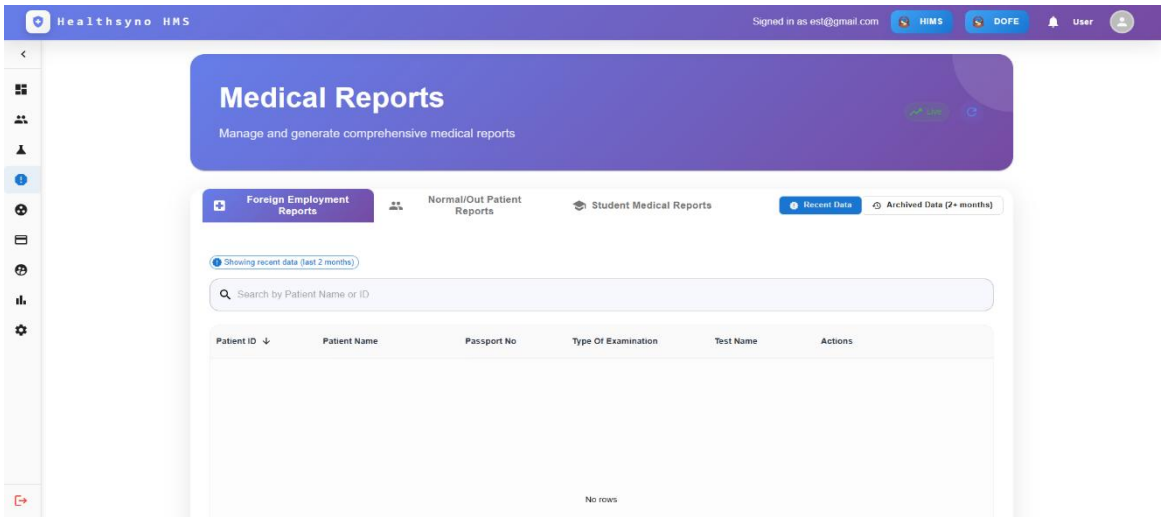

Figure 8.4: Reports Management Page

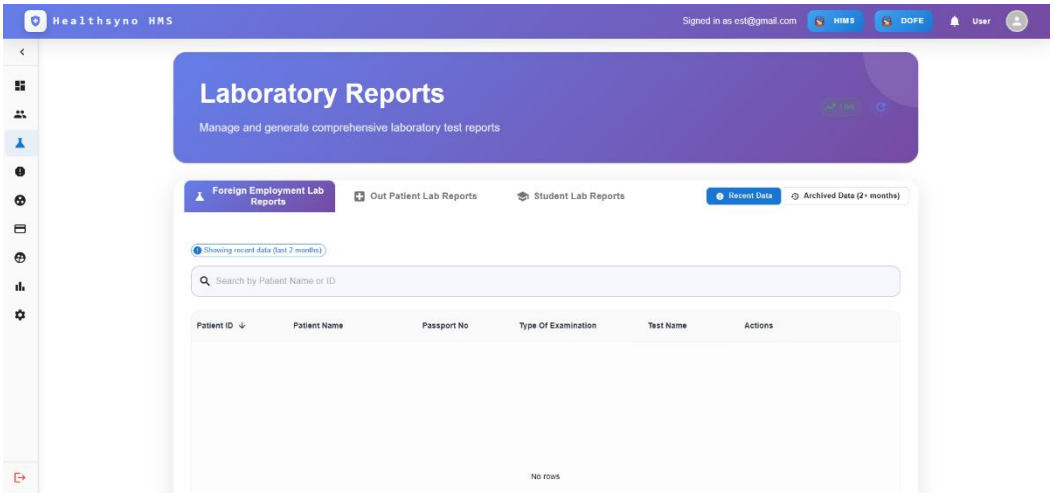

Figure 8.5: Laboratory Reports Management Page

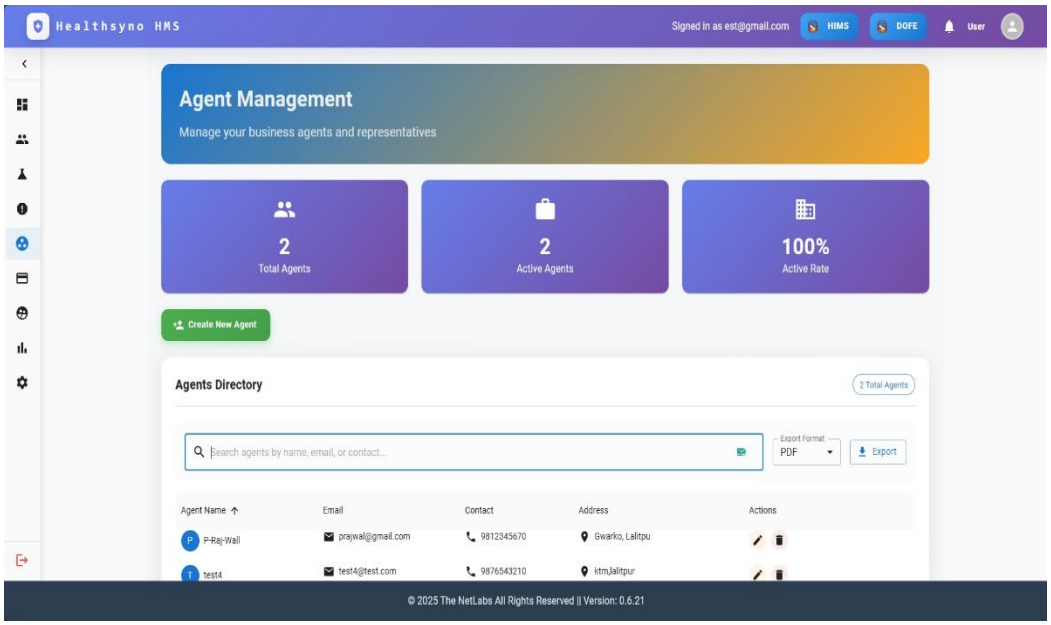

Figure 8.6: Business Agents Management Page

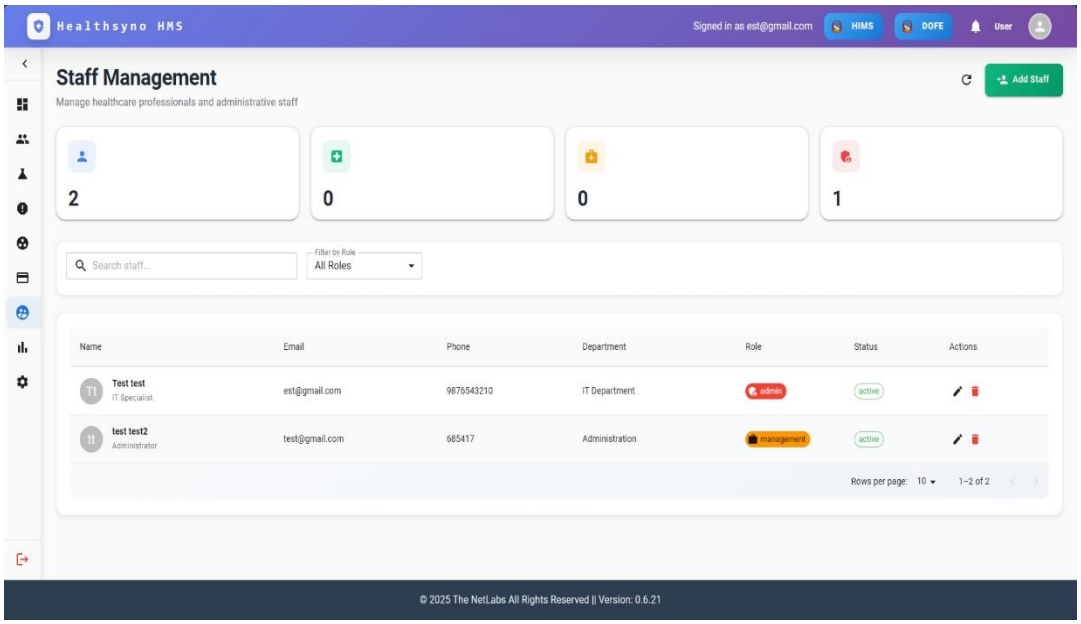

Figure 8.7: Staff Management & User-Control Page

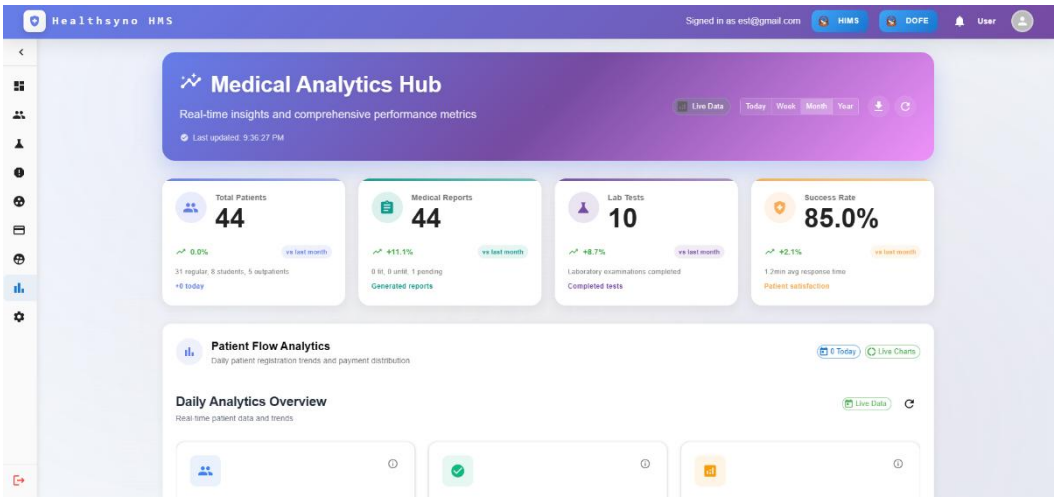

Figure 8.8: Analytics Page

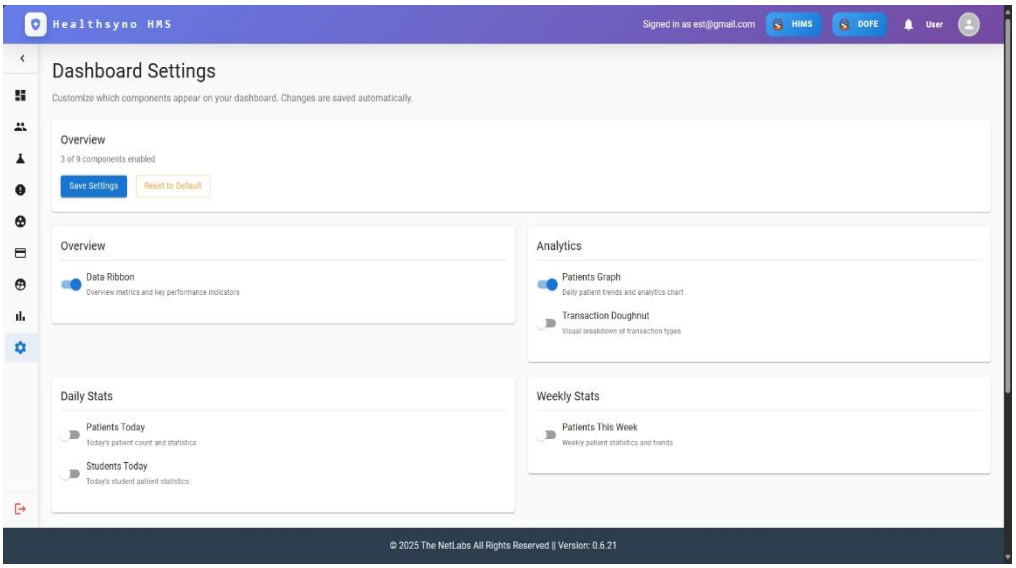

Figure 8.9: Dashboard Settings Page

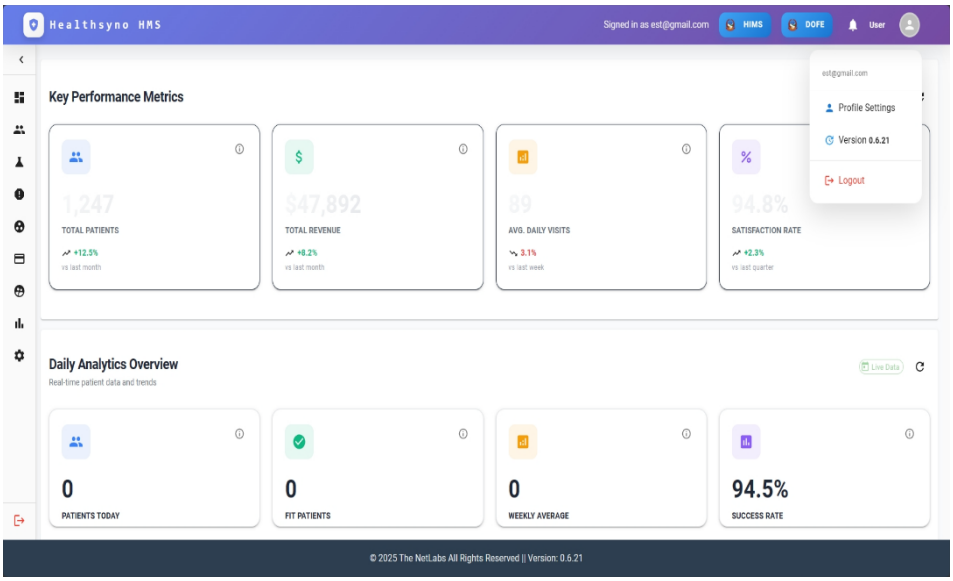

Figure 8.10: User Profile Menu

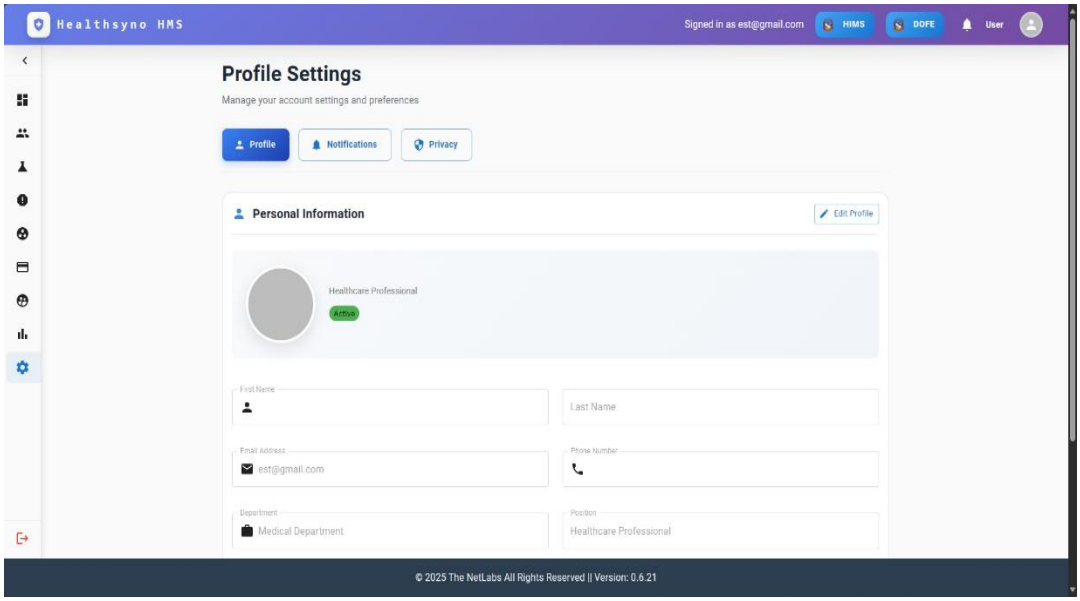

Figure 8.11: Profile Settings Page

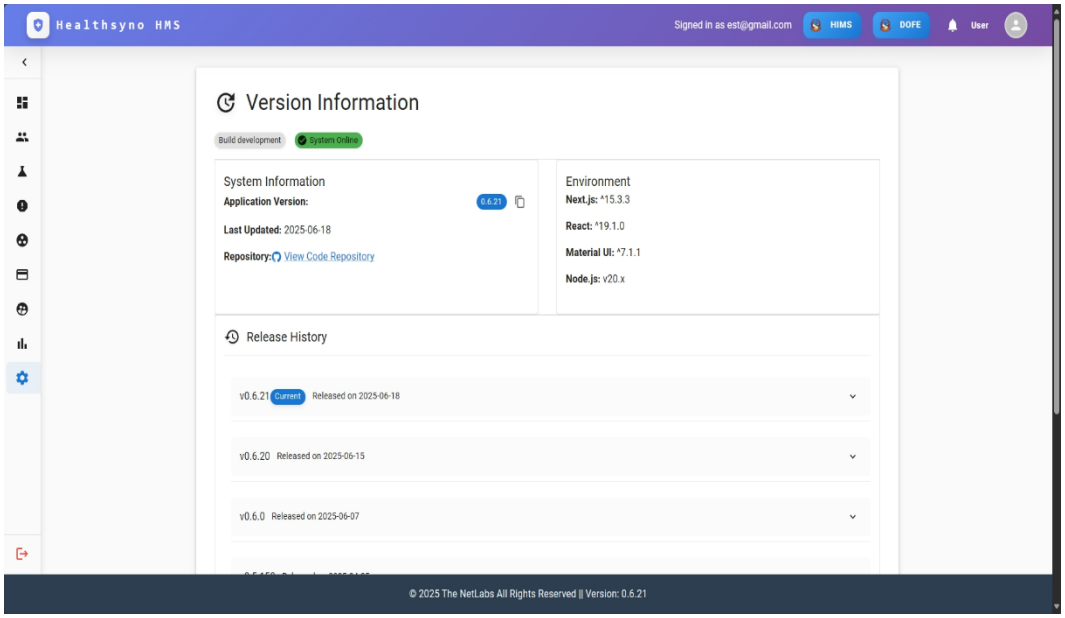

Figure 8.12: Version Control/ Information Page

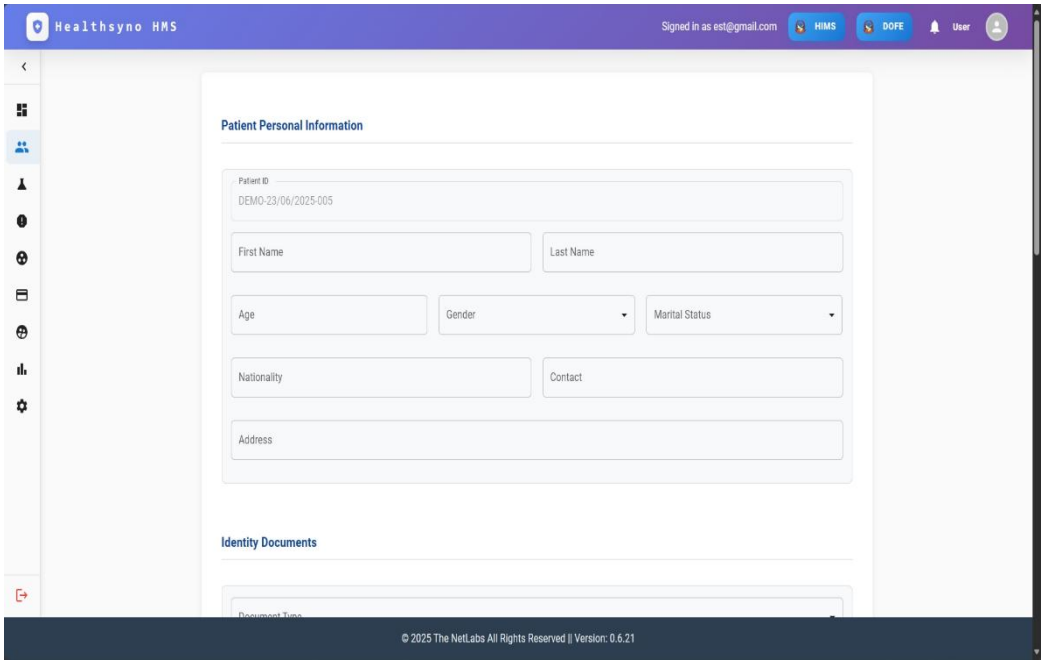

Figure 8.13: Add Patient Module