# A PROJECT ON

# "HOSPITAL MANAGEMENT SYSTEM"

SUBMITTED IN

PARTIAL FULFILMENT OF THE REQUIREMENT

FOR THE COURSE OF DIPLOMA IN ADVANCED COMPUTING FROM CDAC

## SUNBEAM INSTITUTE OF INFORMATION TECHNOLOGY,

## HINJEWADI

**SUBMITTED BY:**

Patil Mayuresh Sanjay (80558)

Sandesh Balu Tribhuvan (80734)

Mrunali Ramesh Barde (80555)

Sainath Govind Ibitwar (80478)

Kate Mayur Rajendra (80713)

**UNDER THE GUIDANCE OF:**

Mr. Shubham Borle

Faculty Member

Sunbeam Institute of Information Technology, Pune

# CERTIFICATE

This is to certify that the project work under the title ‘Hospital Management System’ is done by **Patil Mayuresh Sanjay (80558), Sandesh Balu Tribhuvan (80734), Mrunali Ramesh Barde (80555), Sainath Govind Ibitwar (80478), Kate Mayur Rajendra (80713)** in partial fulfillment of the requirement for award of Diploma in Advanced Computing Course.

**Mr. Shubham Borle**
**Project Guide**

**Mr. Yogesh Kolhe**
**Course Coordinator**

Date:

# ACKNOWLEDGEMENT

A project usually falls short of its expectation unless aided and guided by the right persons at the right time. We avail this opportunity to express our deep sense of gratitude towards **Mr. Nitin Kudale** (Center Coordinator, SIIT, Pune) and **Mr. Yogesh Kolhe** (Course Coordinator, SIIT, Pune).

We are deeply indebted and grateful to them for their guidance, encouragement and deep concern for our project. Without their critical evaluation and suggestions at every stage of the project, this project could never have reached its present form.

Last but not the least we thank the entire faculty and the staff members of Sunbeam Institute of Information Technology, Pune for their support.

**Patil Mayuresh Sanjay (80558)**

**Sandesh Balu Tribhuvan (80734)**

**Mrunali Ramesh Barde (80555)**

**Sainath Govind Ibitwar (80478)**

**Kate Mayur Rajendra (80713)**

PG-DAC, September 2023 Batch

SIIT, Pune

## CONTENTS: Page No.:

# ABSTRACT

The Hospital Management System (HMS) project endeavors to modernize and optimize healthcare operations by introducing a robust digital platform. This comprehensive system is designed to address the challenges inherent in traditional hospital management, offering innovative solutions to streamline processes, improve patient experiences, and empower healthcare professionals.

**Objectives:**

The primary objectives of the HMS project encompass a spectrum of improvements in healthcare administration, including the digitization of operations, enhancement of patient access, facilitation of seamless doctor-patient interactions, efficient bed allocation, data-driven decision-making through reporting, and the implementation of an accurate billing system.

**Methodology:**

Following the Agile methodology, the project embraces collaboration, adaptability, and iterative development. Utilizing two-week sprints, user stories, and continuous feedback loops, the methodology is implemented through Jira, ensuring a dynamic and responsive development process. Continuous integration and deployment practices further enhance efficiency.

**Technologies Used:**

The project employs a contemporary technology stack, featuring React.js and Redux for the frontend, Node.js and Spring Boot for the backend, and MySQL for secure data storage. The implementation of JWT-based authentication adds a robust layer of security, while Maven streamlines project build and dependency management.

**User Modules:**

Tailoring the system to diverse end-users, including patients, doctors, administrators, receptionists, and accountants, each module is designed to cater to specific roles and responsibilities. This approach ensures a personalized and efficient experience for all stakeholders.

**Development Tools:**

Primary integrated development environments include IntelliJ IDEA and VS Code, with MySQL Workbench and command line facilitating database management. Google Chrome is utilized for testing purposes, while Postman aids in API testing during development.

**Conclusion:**

The Hospital Management System project marks a significant stride towards the modernization of healthcare administration. By embracing Agile methodologies and leveraging cutting-edge technologies, the system promises to address challenges in hospital management comprehensively, making it a transformative force in the realm of healthcare administration.

# 1. INTRODUCTION

This web-based "Hospital Management System" project aims to bring the fundamental concepts of healthcare management into a digital framework. This system empowers users to efficiently manage patient, doctor, and administrative information, while also offering appointment booking, billing, and ward bed management capabilities.

The Hospital Management System simplifies the daily operations of healthcare institutions by providing a user-friendly interface for patients, doctors, and administrators. Patients can easily access and navigate the system to make appointments with their preferred doctors, streamline the billing process, and manage their hospital stays, all from the comfort of their homes.

The system offers a comprehensive suite of features, allowing patients to search for available doctors, view their profiles, and schedule appointments on convenient dates and times. Furthermore, it maintains a patient's medical history, making it readily accessible for doctors during consultations and treatment.

Hospital administrators can utilize the system to efficiently manage administrative tasks. Accounts department can utilize the system for generation of bill and daily reports. Additionally, the system provides real-time updates on bed availability, enabling efficient management of ward beds and ensuring that patients are promptly accommodated.

The Hospital Management System is designed to enhance the healthcare experience for patients, doctors, administrators and accountants. It simplifies the processes involved in managing a hospital, from appointment booking and billing to patient care and bed management. By implementing this digital solution, healthcare institutions can achieve greater efficiency and an enhanced patient experience.

### 1.1 Problem Statement

One of the biggest challenges faced by healthcare institutes is managing their resources efficiently. Handling patient information, appointments, billing, bed allocation manually can be error-prone. With unified and digital system these inefficiencies can be overcome and quality of care provided can be improved.

### 1.2 Scope

The Hospital Management System is designed to cover a wide range of healthcare management processes, including patient management, appointment scheduling, billing, and bed allocation. It caters to patients, doctors, administrators, and accounts departments.

### 1.3 End User

- Patient
- Doctor
- Administrator
- Receptionist
- Accountant

### 1.4 Objectives

- Streamline and digitize hospital operations.
- Enhance patient access and experience.
- Provide doctors with easy access to patient information.
- Efficiently manage bed allocation.
- Generate daily reports for decision-making.
- Accurately calculate and generate bills for patients.
- Enhance the overall efficiency and effectiveness of healthcare management.

# 2. METHODOLOGY

SDLC (Software Development Life Cycle) is also called as Software Development Process. It is a well-defined, structured sequence of stages in software engineering to develop the intended software product.

Agile is one of the models of SDLC.

Overview of Agile principles and values:

- Emphasis on collaboration, adaptability, and customer satisfaction.
- Iterative development and continuous improvement.

**Agile methodology used: Scrum**

Application of Agile in the project:

- Use of one-week sprints for incremental development.
- User stories and product backlog management in Jira.
- Daily Scrum for communication and issue resolution, lasting for about 15 minutes
- Sprint Review – at the end of every sprint, an event to demo what has been achieved.
- Sprint Retrospective – after sprint review, to plan what to take into next sprint session.

# 3. SYSTEM ARCHITECTURE:

**Technologies used:**

- **Frontend:** React.js with Redux for state management.
- **Backend:** Node.js for the login and sign-up modules, Spring Boot with Spring Data JPA for all the remaining modules.
- **Database:** MySQL for secure storage of structured data.
- **Authentication:** JWT-based authentication for secure user access.
- **Image Storage:** Storing images as blobs in MySQL for seamless integration.
- **Version Control System:** Git.
- **Build and Dependency Management:** Maven for project build and dependency management.
- **Java Development Kit (JDK):** Oracle Open JDK 21.0.2 for Java development.

# 4. DEVELOPMENT TOOLS:

**Integrated Development Environments (IDEs):**

- IntelliJ IDEA: Primary IDE for Java-based development.
- VS Code: Used for frontend development with React.js and also for backend development with Node.js.

**Database Management:**

- MySQL Workbench: Visual tool for database design and administration, provided ease for SELECTING from tables and for generating ER (Entity Relationship) diagram.
- Command Line: Used for direct interaction with the database.

**API Testing:**

- Postman: Used for testing and validating APIs during development.

**Build and Dependency Management:**

- Maven: Utilized for project build automation and managing project dependencies.

**Browser:**

- Google Chrome: Testing and optimizing the web application.

# 5. REQUIREMENTS

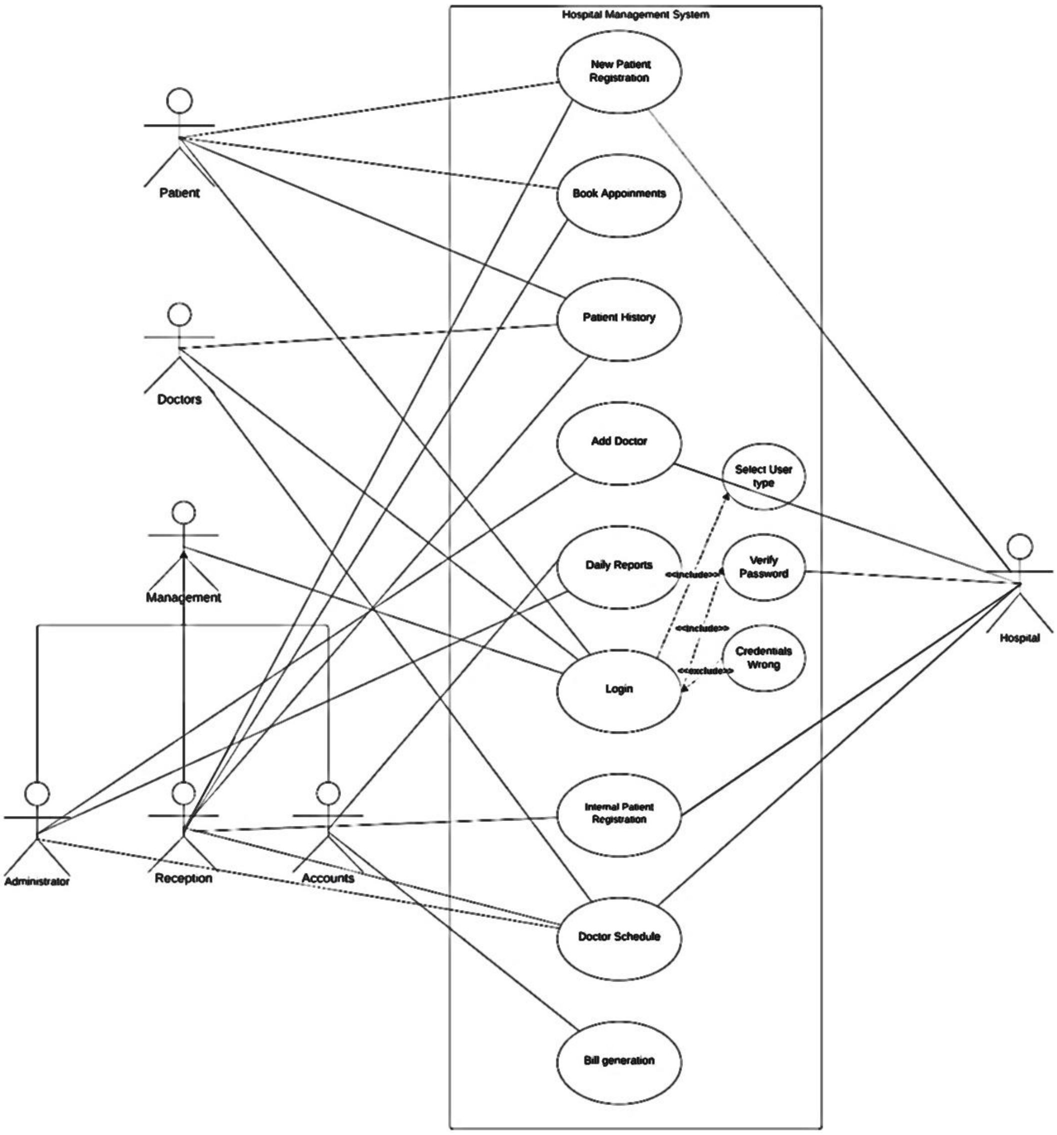


Fig 5.1: Use-Case Diagram.

### 5.1 Actor – Patient

New Patient Registration – The website allows a patient to register on hospital's system by entering personal information, contact details. This will allow patient to check availability for required doctor.

Book Appointments - The patient can later update or edit his/her profile to book appointments at preferred date and time with the doctor. This feature will enhance their access to medical care.

Patient History – Patients can view their Electronic Health Record (EHR) from ease of their home. They can review past treatments and diagnoses.

Login – Patient can login into their profile securely using their username and password. In case the patient forgets his/her password, then it can be reset by answering security questions set by patient during new patient registration.

### 5.2 Actor – Doctor

Patient History – Doctor can view Electronic Health Record (EHR) of the patient seeking treatment from that doctor. This will help doctor to make informed decisions regarding treatments to follow.

Doctor Schedule – Doctors can view their schedule on the website for efficient time management. This will aid doctor in providing quality medical care.

Login - Doctor can login into their profile securely using their username and password.

### 5.3 Actor – Administrator

Add Doctor – Administrator can add doctor into the hospital’s system after background verification. Administrators can efficiently manage doctors affiliated to their hospital.

Daily Reports – Administrator can view daily reports on the website to efficiently manage resources of the hospital. These reports will aid in identifying the areas for improvement.

Doctor Schedule – Administrator can view schedule of doctor.

Login - Administrator can login into their profile securely using their username and password.

### 5.4 Actor – Reception

New Patient Registration – Reception can assist patients to register on the website by collecting their personal information and contact details.

Book Appointments – Reception can interact with patient, check availability of doctor for patient's preferred date and time and fix an appointment.

Patient History – Reception can quickly fetch details of patient's Electronic Health Record (EHR) who are registered on the website. Upon request this information can be made available to patient or doctor from whom the patient is seeking medical treatment.

Doctor Schedule – Reception can check doctor schedule and inform patients regarding availability of doctor in their preferred time slot.

Login - Receptionist can login into their profile securely using their username and password.

The facility of reception will be very useful for the patients who are not well versed in registering and using hospital's website.

### 5.5 Actor – Accounts

Daily Reports – The accounts department can generate daily reports consisting of billing, payments and other financial and non-financial aspects. These reports will help to maintain transparency in management of hospital.

Bill Generation – Accounts department can generate bill based on medical facilities availed by patients. It also helps keep financial records up to date.

Login - Accountant can login into their profile securely using their username and password.

# 6. FUNCTIONALITIES

| Patient registration | | |
|---|---|---|
| 1 | Demographic Information | Collect basic patient details, including name, date of birth, gender, address, phone number, and email address. This information is essential for identifying and contacting the patient. |
| 2 | Identification | Record government-issued identification details, such as a driver's license or passport. |
| 3 | Medical History | Capture the patient's medical history, including any pre-existing conditions, allergies, and a brief family medical history. This information is crucial for providing personalized care. |
| 4 | Emergency Contacts | Collect contact information for one or more emergency contacts in case of critical situations. |
| 5 | Unique Patient ID | Assign a unique patient identifier, such as a Medical Record Number (MRN), to each patient for easy tracking and reference. |
| 8 | Appointment Scheduling | Integrate appointment scheduling, so patients can immediately schedule an appointment after registration. |

Table 6.1: Patient functionality.

| Appointment Scheduling | | |
|---|---|---|
| 1 | User Access Control | Patient can create account or log in to the system to schedule appointments. This allows for a personalized experience and helps in tracking appointments. |
| 2 | Online Booking | Patients can search for doctors by specialty, location, or name and select a convenient time slot. This information should be presented in an easy-to-navigate interface. |
| 3 | Appointment Confirmation | After selecting a time slot, patients receive an immediate confirmation. |
| 4 | Doctor's Dashboard | Doctors should have access to a dashboard that displays their appointment schedule. |
| 5 | Hospital Reception Assistance | For patients who are not comfortable using online booking, hospital reception should be able to schedule appointments on their behalf. |
| 6 | Feedback and Ratings | After an appointment, patients can provide feedback and ratings for the doctor, which can help improve the quality of healthcare services. |

Table 6.2: Appointment functionality.

| Billing and Invoicing | | |
|---|---|---|
| 1 | Patient Billing Information | Collect and maintain patient billing information, including name, address, contact details. |
| 2 | Service Documentation | Record the services provided to patients. This information is essential for generating accurate bills. |
| 3 | Bill Generation | Automatically generate itemized bills for patients based on the services received. |
| 4 | Reports | The reports can help in financial planning and decision-making. |

Table 6.3: Billing functionality.

| Ward and Bed Management | | |
|---|---|---|
| 1 | User Access Control | Access to the Ward and Bed Management module is limited to authorized staff administrators, and doctors. |
| 2 | Real-Time Bed Availability | Display the real-time status of available beds, including information on occupancy. |
| 3 | Patient Admission | Register and admit patients, including capturing patient demographics, medical history, and reason for admission. |
| 4 | Bed Allocation | Assign suitable beds to admitted patients based on factors such as medical condition, special requirements. |
| 6 | Ward Classification | Categorize wards or rooms based on factors like medical specialty, level of care (e.g., intensive care, general care). |

Table 6.4: Ward and Bed functionality.

| Patient Portal | | |
|---|---|---|
| 1 | User Registration | Patients can create accounts or register for the portal securely, ensuring the privacy and security of their health information. |
| 3 | Visit History | Show a history of appointments and visits. |
| 4 | Appointment Scheduling | Patients can schedule, reschedule, or cancel appointments through the portal. |
| 5 | Billing and Payment | Provide access to billing information, including itemized bills, payment history. |
| 6 | Feedback and Satisfaction Surveys | Collect patient feedback and satisfaction surveys through the portal to improve the quality of healthcare services. |

Table 6.5: Patient portal functionality.

| Doctor and Staff Management | | |
|---|---|---|
| 1 | User Access Control | Access to the system is restricted to authorized personnel. Doctors have secure login credentials to access their profiles and schedules. |
| 2 | Profile Information | Each staff member has a profile that includes personal details, contact information. |
| 3 | Role-Based Access | Implement role-based access control to restrict or grant access to specific features and patient data based on the staff member's role (e.g., doctor, receptionist, administrator). |
| 6 | Specialties and Departments | Assign doctors to specific specialties and departments within the hospital to facilitate patient referrals and department-based scheduling. |
| 7 | Staff Search | Search feature is available to quickly find staff members based on their names, roles, or departments. |
| 9 | Emergency Contact Details | Maintain emergency contact information for each staff member in case of unforeseen events. |

Table 6.6: Doctor functionality.

# 7. SYSTEM DESIGN

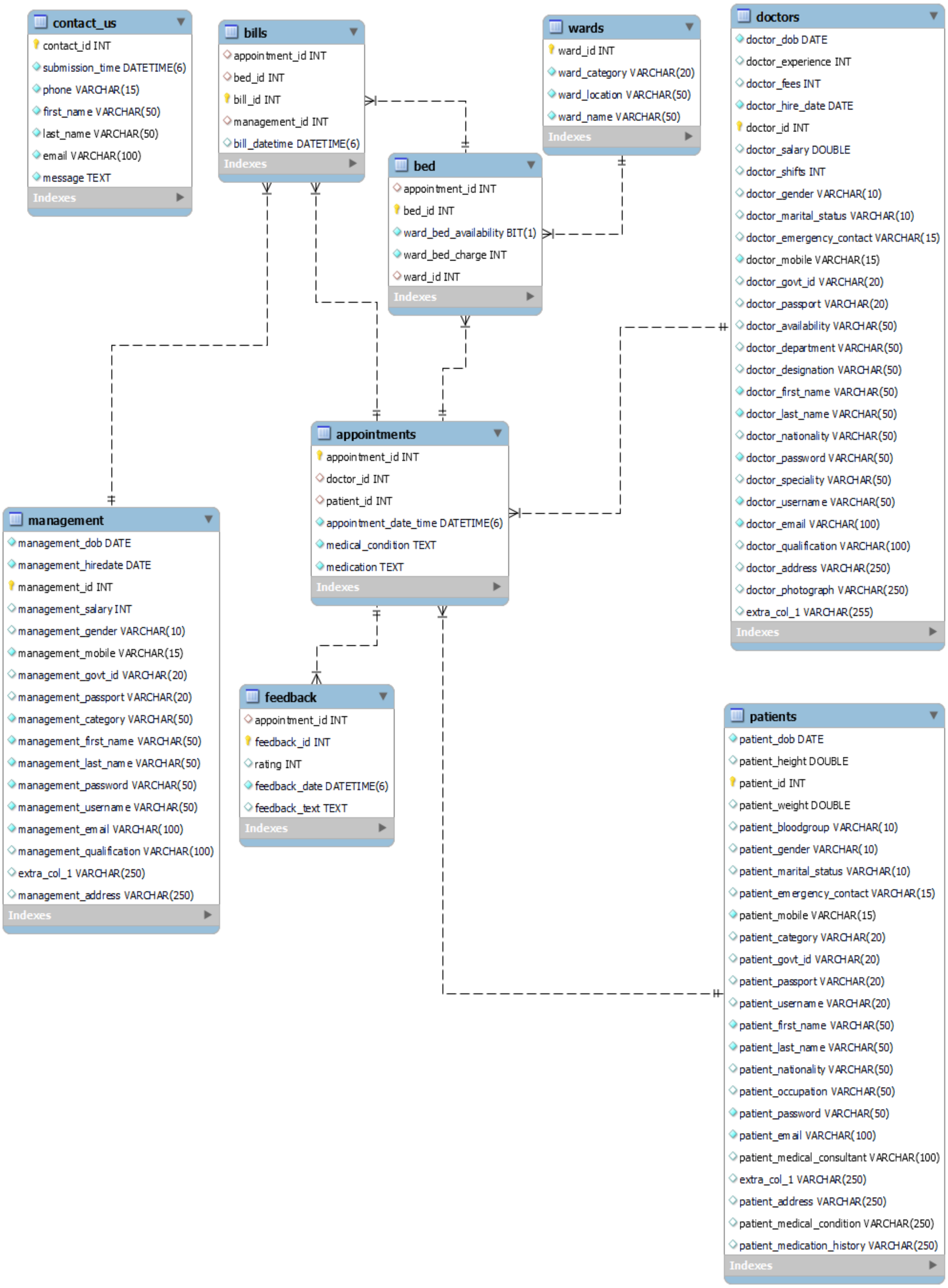
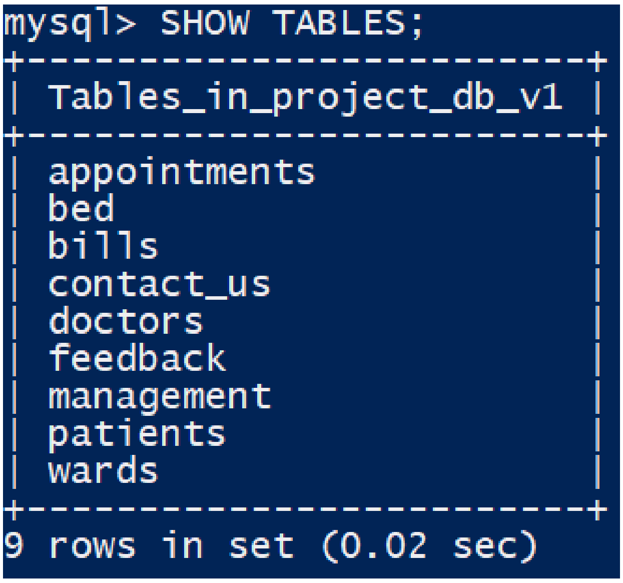

contact_us
contact_id INT
submission_time DATETIME(6)
phone VARCHAR(15)
first_name VARCHAR(50)
last_name VARCHAR(50)
email VARCHAR(100)
message TEXT
Indexes
bills
appointment_id INT
bed_id INT
bill_id INT
management_id INT
bill_datetime DATETIME(6)
Indexes
wards
ward_id INT
ward_category VARCHAR(20)
ward_location VARCHAR(50)
ward_name VARCHAR(50)
Indexes
doctors
doctor_dob DATE
doctor_experience INT
doctor_fees INT
doctor_hire_date DATE
doctor_id INT
doctor_salary DOUBLE
doctor_shifts INT
doctor_gender VARCHAR(10)
doctor_marital_status VARCHAR(10)
doctor_emergency_contact VARCHAR(15)
doctor_mobile VARCHAR(15)
doctor_govt_id VARCHAR(20)
doctor_passport VARCHAR(20)
doctor_availability VARCHAR(50)
doctor_department VARCHAR(50)
doctor_designation VARCHAR(50)
doctor_first_name VARCHAR(50)
doctor_last_name VARCHAR(50)
doctor_nationality VARCHAR(50)
doctor_password VARCHAR(50)
doctor_speciality VARCHAR(50)
doctor_username VARCHAR(50)
doctor_email VARCHAR(100)
doctor_qualification VARCHAR(100)
doctor_address VARCHAR(250)
doctor_photograph VARCHAR(250)
extra_col_1 VARCHAR(255)
Indexes
bed
appointment_id INT
bed_id INT
ward_bed_availability BIT(1)
ward_bed_charge INT
ward_id INT
Indexes
appointments
appointment_id INT
doctor_id INT
patient_id INT
appointment_date_time DATETIME(6)
medical_condition TEXT
medication TEXT
Indexes
management
management_dob DATE
management_hiredate DATE
management_id INT
management_salary INT
management_gender VARCHAR(10)
management_mobile VARCHAR(15)
management_govt_id VARCHAR(20)
management_passport VARCHAR(20)
management_category VARCHAR(50)
management_first_name VARCHAR(50)
management_last_name VARCHAR(50)
management_password VARCHAR(50)
management_username VARCHAR(50)
management_email VARCHAR(100)
management_qualification VARCHAR(100)
extra_col_1 VARCHAR(250)
management_address VARCHAR(250)
Indexes
feedback
appointment_id INT
feedback_id INT
rating INT
feedback_date DATETIME(6)
feedback_text TEXT
Indexes
patients
patient_dob DATE
patient_height DOUBLE
patient_id INT
patient_weight DOUBLE
patient_bloodgroup VARCHAR(10)
patient_gender VARCHAR(10)
patient_marital_status VARCHAR(10)
patient_emergency_contact VARCHAR(15)
patient_mobile VARCHAR(15)
patient_category VARCHAR(20)
patient_govt_id VARCHAR(20)
patient_passport VARCHAR(20)
patient_username VARCHAR(20)
patient_first_name VARCHAR(50)
patient_last_name VARCHAR(50)
patient_nationality VARCHAR(50)
patient_occupation VARCHAR(50)
patient_password VARCHAR(50)
patient_email VARCHAR(100)
patient_medical_consultant VARCHAR(100)
extra_col_1 VARCHAR(250)
patient_address VARCHAR(250)
patient_medical_condition VARCHAR(250)
patient_medication_history VARCHAR(250)
Indexes

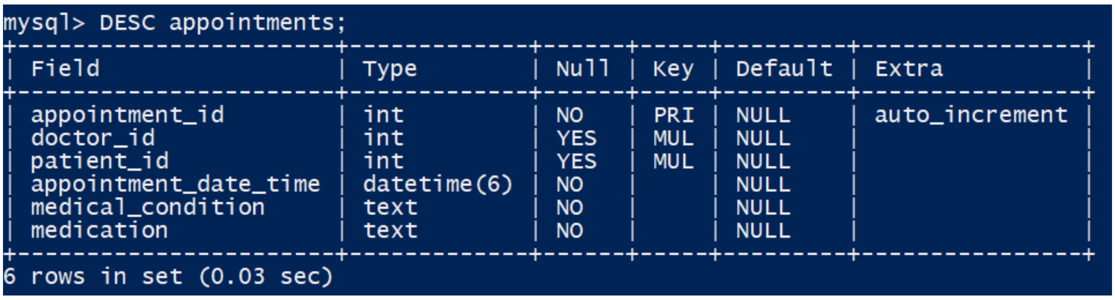
Fig 7.1: ER (Entity Relationship) Diagram.

### 7.1 Tables from MySQL database.



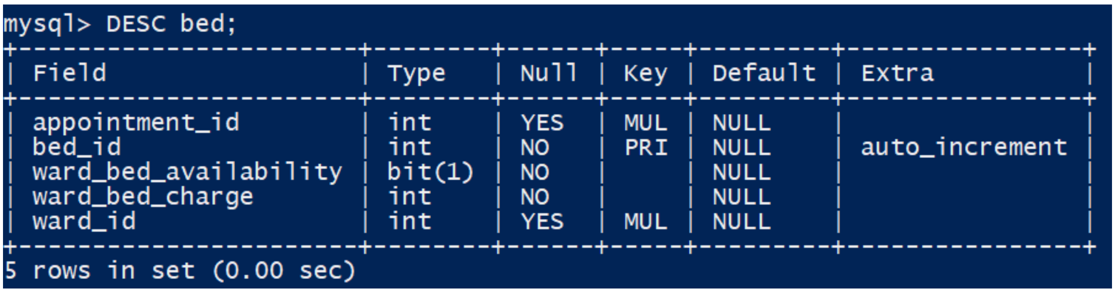
```
mysql> SHOW TABLES;
+-------------------------+
| Tables_in_project_db_v1 |
+-------------------------+
| appointments            |
| bed                     |
| bills                   |
| contact_us              |
| doctors                 |
| feedback                |
| management              |
| patients                |
| wards                   |
+-------------------------+
9 rows in set (0.02 sec)
```

Fig 7.1: List of tables from the database.



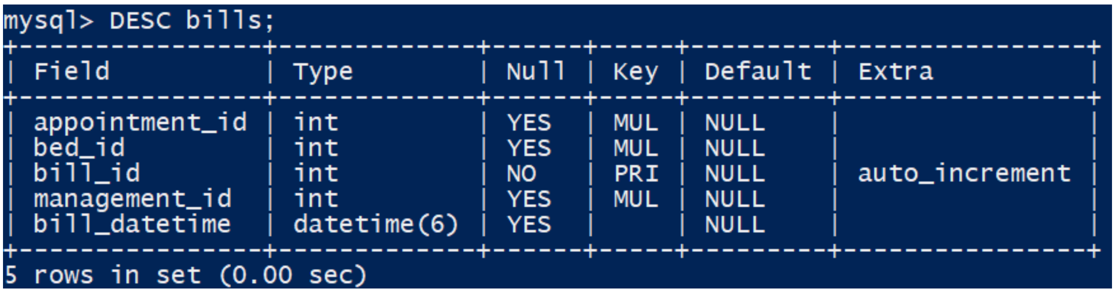
```
mysql> DESC appointments;
+-----------------------+-------------+------+-----+---------+----------------+
| Field                 | Type        | Null | Key | Default | Extra          |
+-----------------------+-------------+------+-----+---------+----------------+
| appointment_id        | int         | NO   | PRI | NULL    | auto_increment |
| doctor_id             | int         | YES  | MUL | NULL    |                |
| patient_id            | int         | YES  | MUL | NULL    |                |
| appointment_date_time | datetime(6) | NO   |     | NULL    |                |
| medical_condition     | text        | NO   |     | NULL    |                |
| medication            | text        | NO   |     | NULL    |                |
+-----------------------+-------------+------+-----+---------+----------------+
6 rows in set (0.03 sec)
```

Fig 7.2: Description of appointments table.



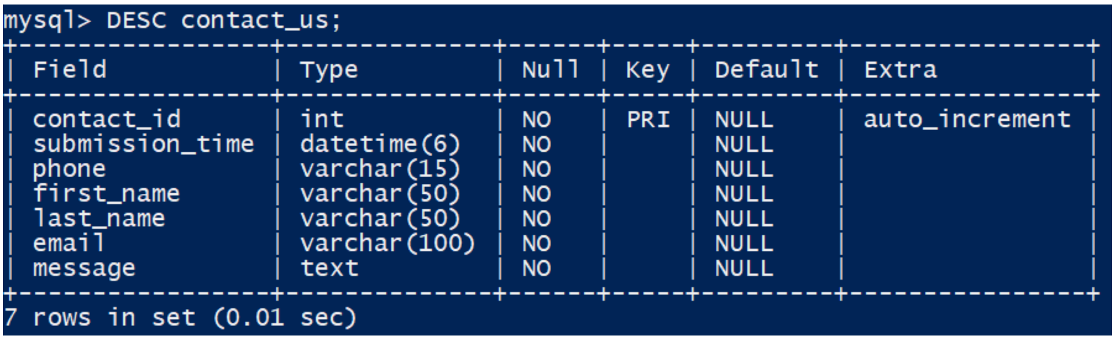
```
mysql> DESC bed;
+-----------------------+--------+------+-----+---------+----------------+
| Field                 | Type   | Null | Key | Default | Extra          |
+-----------------------+--------+------+-----+---------+----------------+
| appointment_id        | int    | YES  | MUL | NULL    |                |
| bed_id                | int    | NO   | PRI | NULL    | auto_increment |
| ward_bed_availability | bit(1) | NO   |     | NULL    |                |
| ward_bed_charge       | int    | NO   |     | NULL    |                |
| ward_id               | int    | YES  | MUL | NULL    |                |
+-----------------------+--------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Fig 7.3: Description of bed table.



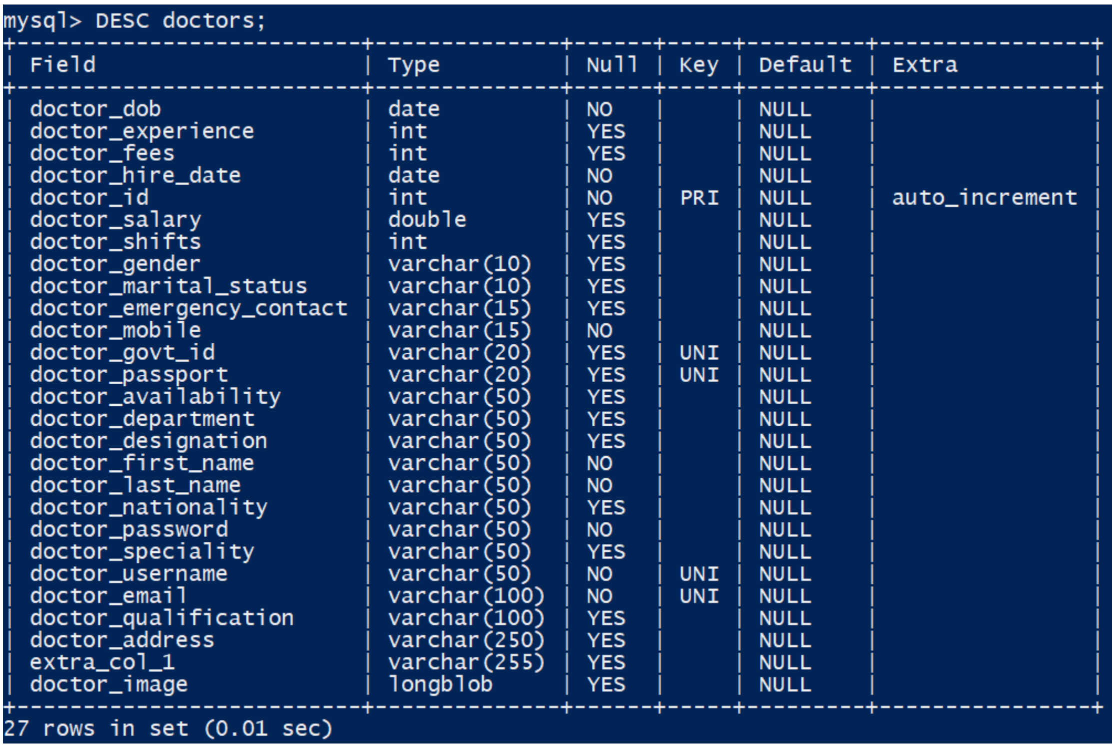
```
mysql> DESC bills;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| appointment_id | int         | YES  | MUL | NULL    |                |
| bed_id         | int         | YES  | MUL | NULL    |                |
| bill_id        | int         | NO   | PRI | NULL    | auto_increment |
| management_id  | int         | YES  | MUL | NULL    |                |
| bill_datetime  | datetime(6) | YES  |     | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Fig 7.4: Description of bills table.



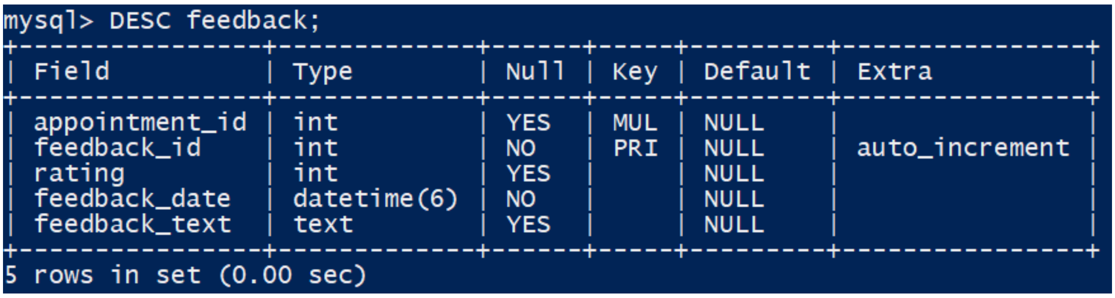
```
mysql> DESC contact_us;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| contact_id      | int          | NO   | PRI | NULL    | auto_increment |
| submission_time | datetime(6)  | NO   |     | NULL    |                |
| phone           | varchar(15)  | NO   |     | NULL    |                |
| first_name      | varchar(50)  | NO   |     | NULL    |                |
| last_name       | varchar(50)  | NO   |     | NULL    |                |
| email           | varchar(100) | NO   |     | NULL    |                |
| message         | text         | NO   |     | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
7 rows in set (0.01 sec)
```

Fig 7.5: Description of contact_us table.



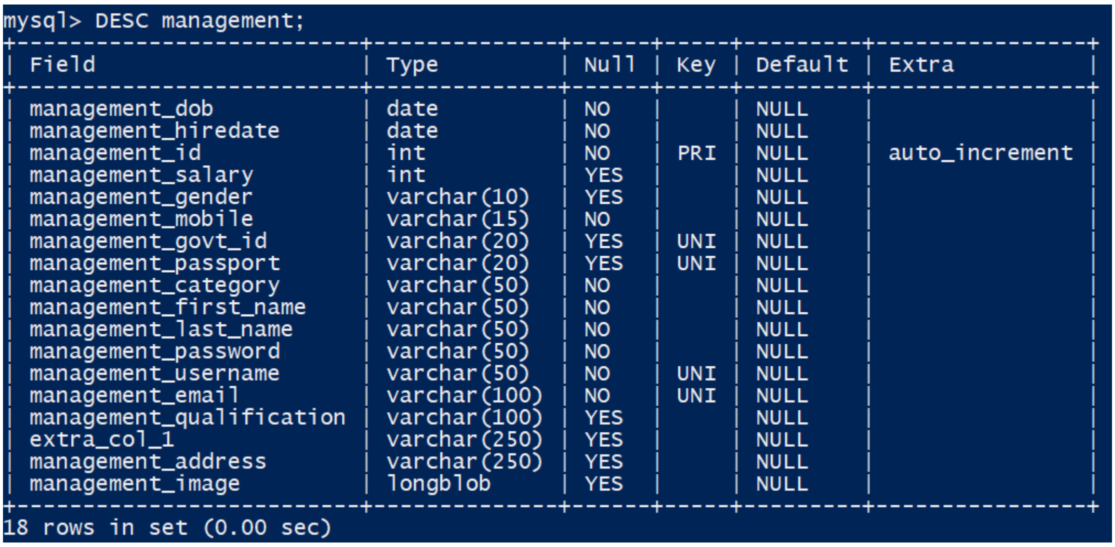
```
mysql> DESC doctors;
+--------------------------+--------------+------+-----+---------+----------------+
| Field                    | Type         | Null | Key | Default | Extra          |
+--------------------------+--------------+------+-----+---------+----------------+
| doctor_dob               | date         | NO   |     | NULL    |                |
| doctor_experience        | int          | YES  |     | NULL    |                |
| doctor_fees              | int          | YES  |     | NULL    |                |
| doctor_hire_date         | date         | NO   |     | NULL    |                |
| doctor_id                | int          | NO   | PRI | NULL    | auto_increment |
| doctor_salary            | double       | YES  |     | NULL    |                |
| doctor_shifts            | int          | YES  |     | NULL    |                |
| doctor_gender            | varchar(10)  | YES  |     | NULL    |                |
| doctor_marital_status    | varchar(10)  | YES  |     | NULL    |                |
| doctor_emergency_contact | varchar(15)  | YES  |     | NULL    |                |
| doctor_mobile            | varchar(15)  | NO   |     | NULL    |                |
| doctor_govt_id           | varchar(20)  | YES  | UNI | NULL    |                |
| doctor_passport          | varchar(20)  | YES  | UNI | NULL    |                |
| doctor_availability      | varchar(50)  | YES  |     | NULL    |                |
| doctor_department        | varchar(50)  | YES  |     | NULL    |                |
| doctor_designation       | varchar(50)  | YES  |     | NULL    |                |
| doctor_first_name        | varchar(50)  | NO   |     | NULL    |                |
| doctor_last_name         | varchar(50)  | NO   |     | NULL    |                |
| doctor_nationality       | varchar(50)  | YES  |     | NULL    |                |
| doctor_password          | varchar(50)  | NO   |     | NULL    |                |
| doctor_speciality        | varchar(50)  | YES  |     | NULL    |                |
| doctor_username          | varchar(50)  | NO   | UNI | NULL    |                |
| doctor_email             | varchar(100) | NO   | UNI | NULL    |                |
| doctor_qualification     | varchar(100) | YES  |     | NULL    |                |
| doctor_address           | varchar(250) | YES  |     | NULL    |                |
| extra_col_1              | varchar(255) | YES  |     | NULL    |                |
| doctor_image             | longblob     | YES  |     | NULL    |                |
+--------------------------+--------------+------+-----+---------+----------------+
27 rows in set (0.01 sec)
```

Fig 7.6: Description of doctors table.



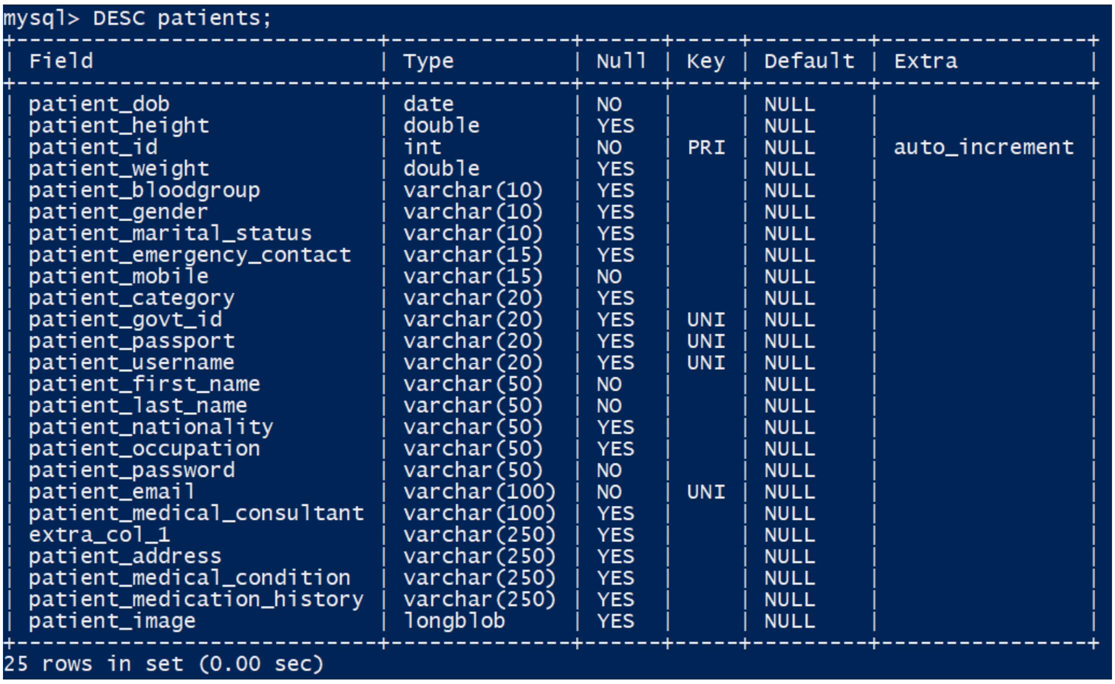
```
mysql> DESC feedback;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| appointment_id | int         | YES  | MUL | NULL    |                |
| feedback_id    | int         | NO   | PRI | NULL    | auto_increment |
| rating         | int         | YES  |     | NULL    |                |
| feedback_date  | datetime(6) | NO   |     | NULL    |                |
| feedback_text  | text        | YES  |     | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
5 rows in set (0.00 sec)
```

Fig 7.7: Description of feedback table.



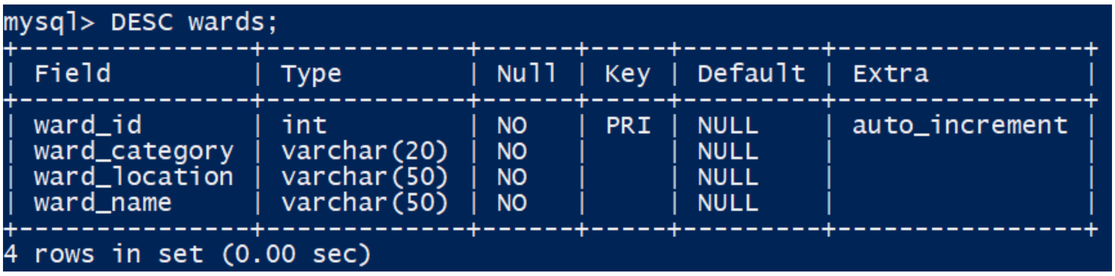
```
mysql> DESC management;
+--------------------------+--------------+------+-----+---------+----------------+
| Field                    | Type         | Null | Key | Default | Extra          |
+--------------------------+--------------+------+-----+---------+----------------+
| management_dob           | date         | NO   |     | NULL    |                |
| management_hiredate      | date         | NO   |     | NULL    |                |
| management_id            | int          | NO   | PRI | NULL    | auto_increment |
| management_salary        | int          | YES  |     | NULL    |                |
| management_gender        | varchar(10)  | YES  |     | NULL    |                |
| management_mobile        | varchar(15)  | NO   |     | NULL    |                |
| management_govt_id       | varchar(20)  | YES  | UNI | NULL    |                |
| management_passport      | varchar(20)  | YES  | UNI | NULL    |                |
| management_category      | varchar(50)  | NO   |     | NULL    |                |
| management_first_name    | varchar(50)  | NO   |     | NULL    |                |
| management_last_name     | varchar(50)  | NO   |     | NULL    |                |
| management_password      | varchar(50)  | NO   |     | NULL    |                |
| management_username      | varchar(50)  | NO   | UNI | NULL    |                |
| management_email         | varchar(100) | NO   | UNI | NULL    |                |
| management_qualification | varchar(100) | YES  |     | NULL    |                |
| extra_col_1              | varchar(250) | YES  |     | NULL    |                |
| management_address       | varchar(250) | YES  |     | NULL    |                |
| management_image         | longblob     | YES  |     | NULL    |                |
+--------------------------+--------------+------+-----+---------+----------------+
18 rows in set (0.00 sec)
```

Fig 7.8: Description of management table.



```
mysql> DESC patients;
+----------------------------+--------------+------+-----+---------+----------------+
| Field                      | Type         | Null | Key | Default | Extra          |
+----------------------------+--------------+------+-----+---------+----------------+
| patient_dob                | date         | NO   |     | NULL    |                |
| patient_height             | double       | YES  |     | NULL    |                |
| patient_id                 | int          | NO   | PRI | NULL    | auto_increment |
| patient_weight             | double       | YES  |     | NULL    |                |
| patient_bloodgroup         | varchar(10)  | YES  |     | NULL    |                |
| patient_gender             | varchar(10)  | YES  |     | NULL    |                |
| patient_marital_status     | varchar(10)  | YES  |     | NULL    |                |
| patient_emergency_contact  | varchar(15)  | YES  |     | NULL    |                |
| patient_mobile             | varchar(15)  | NO   |     | NULL    |                |
| patient_category           | varchar(20)  | YES  |     | NULL    |                |
| patient_govt_id            | varchar(20)  | YES  | UNI | NULL    |                |
| patient_passport           | varchar(20)  | YES  | UNI | NULL    |                |
| patient_username           | varchar(20)  | YES  | UNI | NULL    |                |
| patient_first_name         | varchar(50)  | NO   |     | NULL    |                |
| patient_last_name          | varchar(50)  | NO   |     | NULL    |                |
| patient_nationality        | varchar(50)  | YES  |     | NULL    |                |
| patient_occupation         | varchar(50)  | YES  |     | NULL    |                |
| patient_password           | varchar(50)  | NO   |     | NULL    |                |
| patient_email              | varchar(100) | NO   | UNI | NULL    |                |
| patient_medical_consultant | varchar(100) | YES  |     | NULL    |                |
| extra_col_1                | varchar(250) | YES  |     | NULL    |                |
| patient_address            | varchar(250) | YES  |     | NULL    |                |
| patient_medical_condition  | varchar(250) | YES  |     | NULL    |                |
| patient_medication_history | varchar(250) | YES  |     | NULL    |                |
| patient_image              | longblob     | YES  |     | NULL    |                |
+----------------------------+--------------+------+-----+---------+----------------+
25 rows in set (0.00 sec)
```

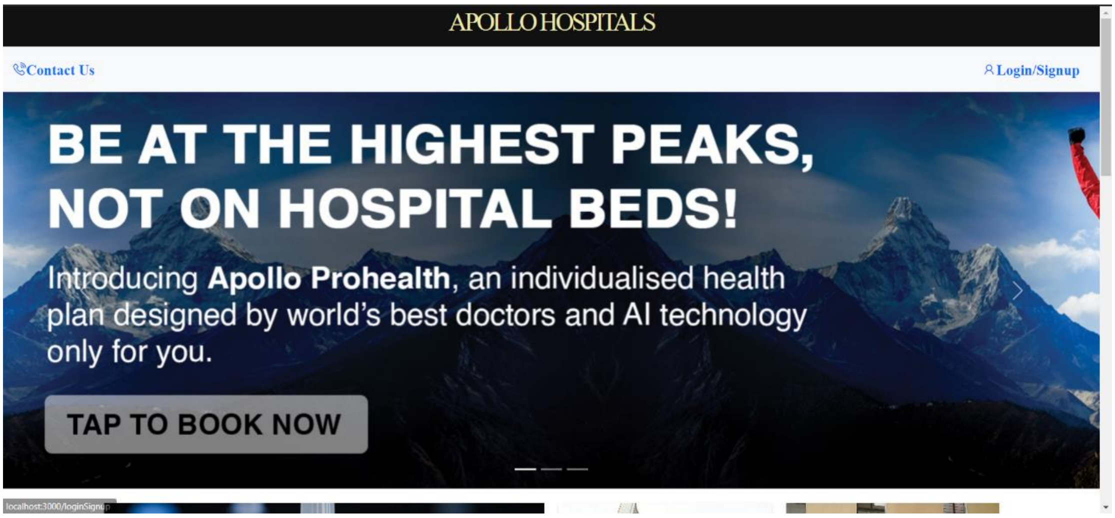
Fig 7.8: Description of patients table.



```
mysql> DESC wards;
+---------------+-------------+------+-----+---------+----------------+
| Field         | Type        | Null | Key | Default | Extra          |
+---------------+-------------+------+-----+---------+----------------+
| ward_id       | int         | NO   | PRI | NULL    | auto_increment |
| ward_category | varchar(20) | NO   |     | NULL    |                |
| ward_location | varchar(50) | NO   |     | NULL    |                |
| ward_name     | varchar(50) | NO   |     | NULL    |                |
+---------------+-------------+------+-----+---------+----------------+
4 rows in set (0.00 sec)
```

Fig 7.10: Description of wards table.

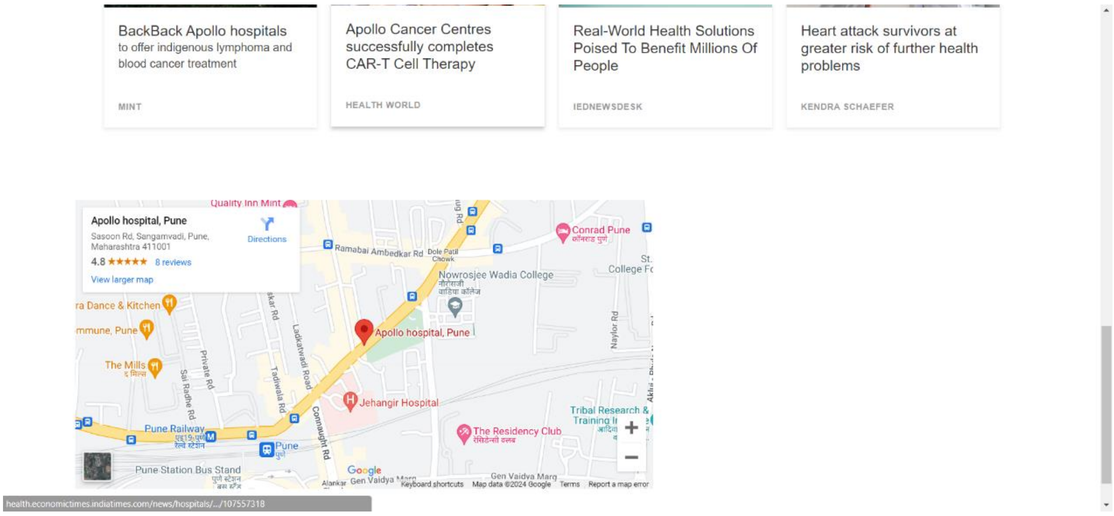
# 8. SYSTEM IMPLEMENTATION



BE AT THE HIGHEST PEAKS, NOT ON HOSPITAL BEDS!

Introducing **Apollo Prohealth**, an individualised health plan designed by world's best doctors and AI technology only for you.

Fig 8.1: Home screen.



BackBack Apollo hospitals to offer indigenous lymphoma and blood cancer treatment

MINT

Apollo Cancer Centres successfully completes CAR-T Cell Therapy

HEALTH WORLD

Real-World Health Solutions Poised To Benefit Millions Of People

IEDNEWSDESK

Heart attack survivors at greater risk of further health problems

KENDRA SCHAEFER

Apollo hospital, Pune

Sasoon Rd, Sangamvadi, Pune, Maharashtra 411001

4.8 ★★★★★ 8 reviews

Fig 8.2: Home screen with map API.

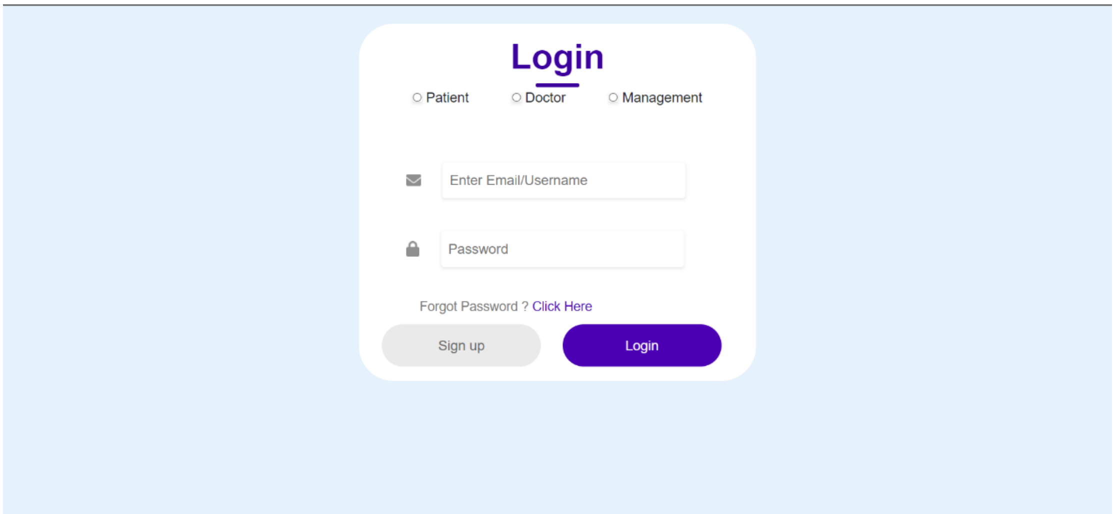


Fig 8.3: Login screen.

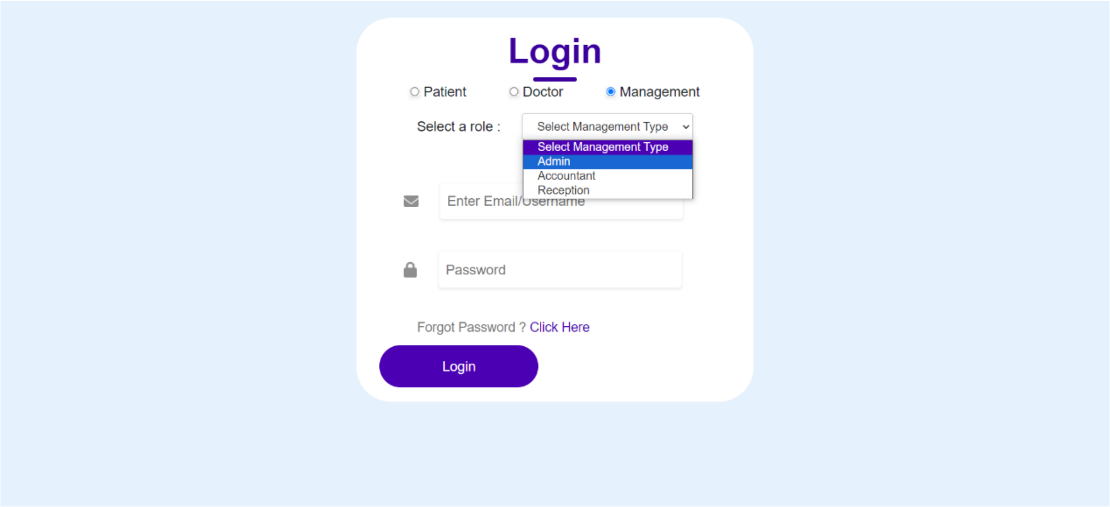
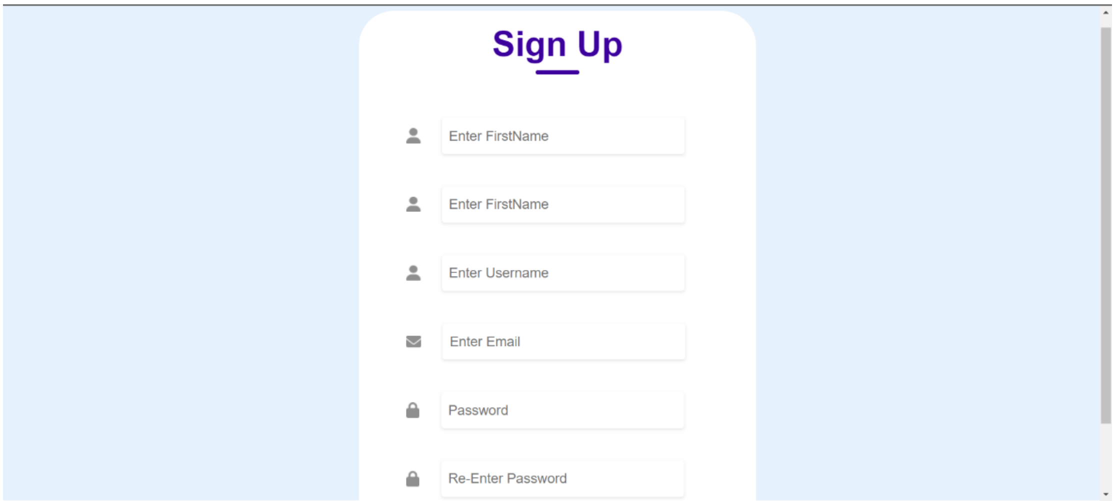


Fig 8.4: Login screen with management role selected.



Sign Up

Enter FirstName

Enter FirstName

Enter Username

Enter Email

Password

Re-Enter Password

Fig 8.5: Sign-up screen.

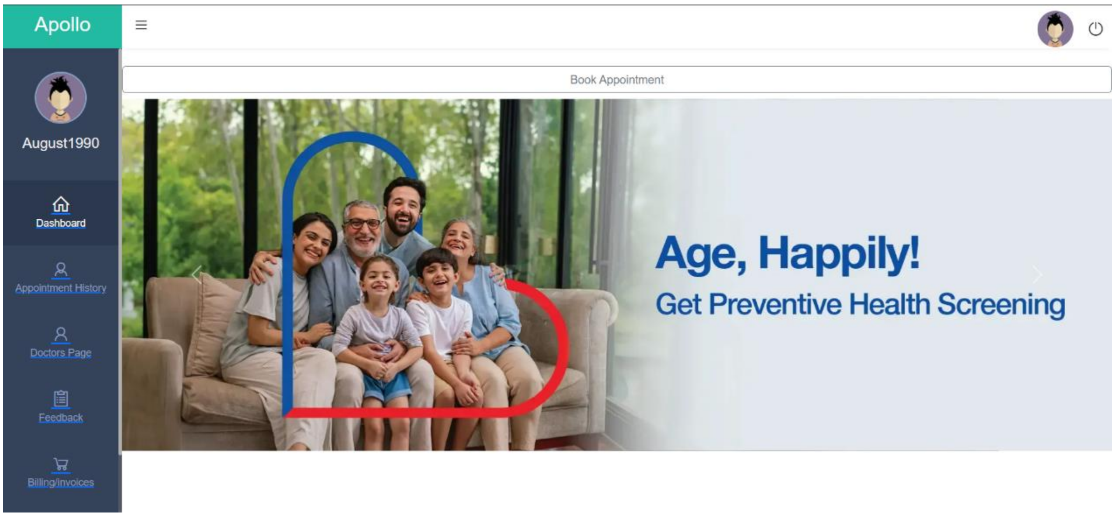
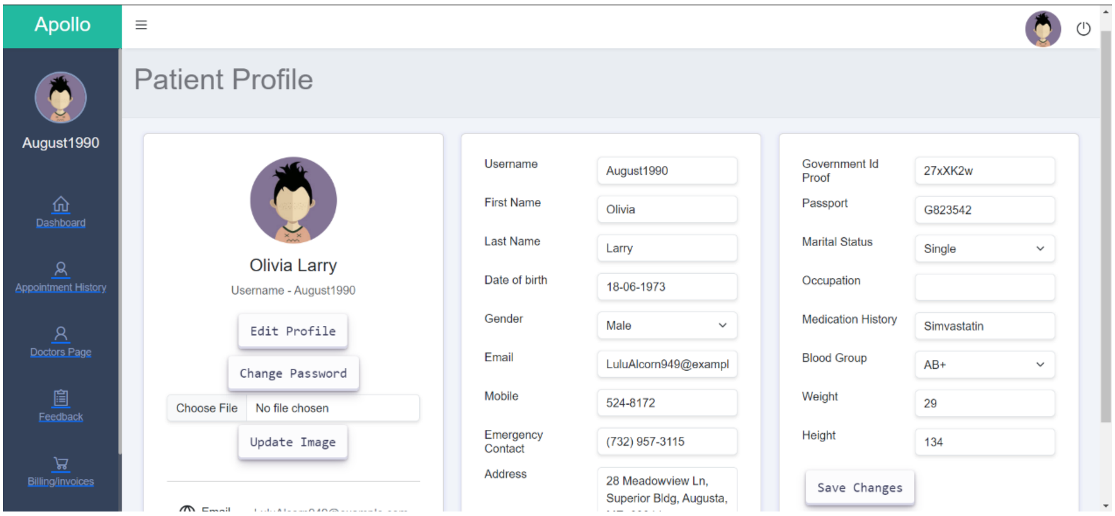


Fig 8.6: Patient dashboard.



Patient Profile

Olivia Larry

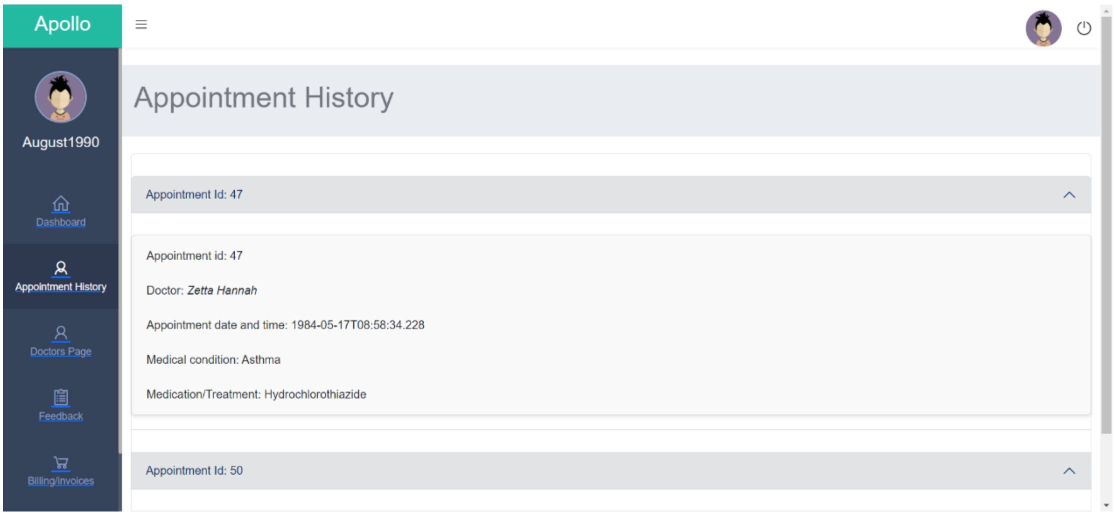
Username - August1990

No file chosen

| | |
|---|---|
| Username | August1990 |
| First Name | Olivia |
| Last Name | Larry |
| Date of birth | 18-06-1973 |
| Gender | Male |
| Email | LuluAlcorn949@exampl |
| Mobile | 524-8172 |
| Emergency Contact | (732) 957-3115 |
| Address | 28 Meadowview Ln, Superior Bldg, Augusta, |

| | |
|---|---|
| Government Id Proof | 27xXK2w |
| Passport | G823542 |
| Marital Status | Single |
| Occupation | |
| Medication History | Simvastatin |
| Blood Group | AB+ |
| Weight | 29 |
| Height | 134 |

Fig 8.7: Patient profile.



Appointment History

Appointment Id: 47

Appointment id: 47

Doctor: *Zetta Hannah*

Appointment date and time: 1984-05-17T08:58:34.228

Medical condition: Asthma

Medication/Treatment: Hydrochlorothiazide

Appointment Id: 50

Fig 8.8: Patient appointment history.

## 9. CONCLUSION

In wrapping up our Hospital Management System journey, it's clear that we've created something pretty special. Our goal was to make hospital processes smoother, give patients more control, and help healthcare providers do their jobs better. Let's take a quick look at what we've achieved.

**Making Things Easier:**

We successfully brought the hospital into the digital age. Imagine checking into the hospital just as easily as ordering food online. That's the kind of simplicity we aimed for, and we made it happen.

**Better for Patients:**

We designed a patient portal that's like a personal health assistant. Patients can book appointments, check medical records, and even handle bills online. It's all about putting people in charge of their health journey.

**Helping Doctors Help You:**

Doctors now have quick access to your medical history. This isn't just about making their jobs easier; it's about providing better and faster care to patients like you.

**No More Bed Hassles:**

Our bed management system ensures that there's always a bed when you need one. It's like having a smart system that knows where beds are available and makes sure you get one without any fuss.

**Smart Decision-Making:**

Hospital leaders can now make smarter decisions with the daily reports we generate. It's like having a dashboard that helps them see what's working well and what needs improvement.

**Clear and Fair Bills:**

The billing system is now crystal clear. No more surprises. Patients can see exactly what they're paying for, making the financial side of healthcare more transparent.

**What Worked Well:**

Our success story is a team effort. The Agile method we used was like taking small steps and adjusting our path along the way. This way, we could respond quickly to what everyone needed.

**Keeping Things Safe:**

We also made sure your data is safe and secure. Think of it like a lock and key – only you and your healthcare team can access your information.

In a nutshell, our Hospital Management System isn't just about technology; it's about making healthcare better for everyone involved. We're excited about the positive changes it will bring to hospitals and the way people experience healthcare.