PROJECT REPORT FILE

Topic:- Hospital Management System

Subject: Database Management Systems LAB

Course Code: 20CP208P

Submitted By:

Name: Akbari Kartik Rajnikantbhai
Roll No.: 22BCP255

Name: Barasara Om Hemantbhai
Roll No.: 22BCP254

Submitted To:

Dr. Yogesh Kumar
Department of CSE

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

SCHOOL OF TECHNOLOGY
PANDIT DEENDAYAL ENERGY UNIVERSITY, GANDHINAGAR

ACKNOWLEDGEMENT

I would like to extend my heartfelt gratitude to Prof. Yogesh Kumar for his invaluable guidance and unwavering support throughout the duration of our Database Management System project. His profound knowledge, insightful suggestions, and constructive feedback were instrumental in shaping our understanding and refining our work. Prof. Yogesh Kumar dedication to our academic growth has been truly inspiring, and we are immensely grateful for the opportunity to learn under his mentorship.

Thank you for providing us with this opportunity to explore and learn about Database Management System in depth. We have put in our best efforts to ensure the quality and comprehensiveness of the report.

INDEX

OVERVIEW:

This project focused on implementing Hospital Management System using web development. The main topics covered were:

- A Hospital Management System (HMS) is a comprehensive software solution that facilitates the efficient management of various aspects of hospital operations, including patient information, doctor schedules, medical records, inventory management, billing, and more. When developing a database management system (DBMS) project for a hospital management system, you'll need to design a robust database schema that can effectively store and manage all the necessary data.

1. **Patient Management**:
- Capture and store patient demographics (e.g., name, age, gender, contact details).
- Maintain a unique identifier (e.g., patient ID) for each patient.
- Track patient Prescriptive history, appointment, room no. etc.
- Record patient visits, including admission dates.

2. **Doctor Management**:
- Store doctor details such as name, specialty, contact information, and schedule.
- Maintain a unique identifier (e.g., doctor ID) for each doctor.
- Link doctors to their patients and appointments.

3. **Appointment Scheduling**:
- Enable patients to schedule appointments with doctors.
- Manage appointment slots, availability, and scheduling conflicts.
- Provide reminders and notifications for upcoming appointments.

4. **Room Management:**
- Show that room is allocated or not for the patient.
- Show the room availability.

5. **Billing and Payments**:
- Generate bills for services rendered, including consultations, procedures, and medications.
- Calculate charges based on treatment plans, insurance coverage, and discounts.
- Record payments, issue receipts, and manage billing disputes.

Overall, the project showcased the application of Hospital Management System concepts using the Web development and SQL database. Through practical implementations, it provided insights into data of patient, doctor and all in the backend database.

APPLICATION

A Hospital Management System (HMS) finds application across various aspects of healthcare delivery, administration, and management. Here are some key areas where an HMS is commonly applied

1) **Patient Management**:

- Capturing and maintaining patient demographics, medical history, and treatment plans.
- Facilitating patient registration, admission, and discharge processes.
- Managing patient appointments, consultations, and follow-up visits.
- Providing patients with access to their health records and appointment scheduling through online portals or mobile apps.

2) **Clinical Operations:**

- Streamlining workflows for doctors, nurses, and other healthcare professionals.
- Automating prescription management, including medication ordering, dispensing, and administration.
- Integrating with medical devices and laboratory systems for test ordering and results reporting.
- Facilitating communication and collaboration among healthcare team members for coordinated patient care.

3) **Inventory and Resource Management:**

- Tracking and managing hospital inventory, including medicines, medical supplies.
- Optimizing inventory levels to prevent stockouts and minimize wastage.
- Managing hospital assets, including equipment maintenance schedules and depreciation tracking.
- Allocating resources such as hospital beds, operating rooms, and staff based on patient needs and demand.

4) **Billing and Financial Management**:

- Generating bills and invoices for services rendered, including consultations, procedures, and medications.
- Integrating with insurance systems to verify coverage, submit claims, and process reimbursements.
- Managing accounts receivable, tracking payments, and handling billing disputes.
- Analysing financial data and performance metrics to optimize revenue and reduce costs.

5) **Patient Engagement and Experience**:

- Enhancing patient satisfaction through improved access to healthcare services and information.
- Providing patients with personalized health education resources, treatment reminders, and wellness programs.

OBJECTIVE

The objectives of a Hospital Management System (HMS) project built on a database management system (DBMS) encompass various aspects aimed at enhancing the efficiency, accuracy, and effectiveness of hospital operations. Here are some key objectives:

A. **Efficient Patient Management**:

- Capture and maintain comprehensive patient records, including demographics, medical history, and treatment plans, in a centralized database.
- Facilitate easy access to patient information for healthcare providers, improving the quality and continuity of care.

B. **Streamlined Clinical Processes**:

- Automate clinical workflows, such as appointment scheduling, prescription management, and diagnostic test ordering, to reduce errors and delays.
- Enable healthcare providers to access patient data and collaborate seamlessly, enhancing communication and care coordination.

C. **Optimized Resource Utilization**:

- Manage hospital resources effectively, including beds, equipment, and staff, to meet patient demand while minimizing waste and inefficiency.
- Monitor inventory levels and automate supply chain processes to ensure timely availability of medicines and medical supplies.

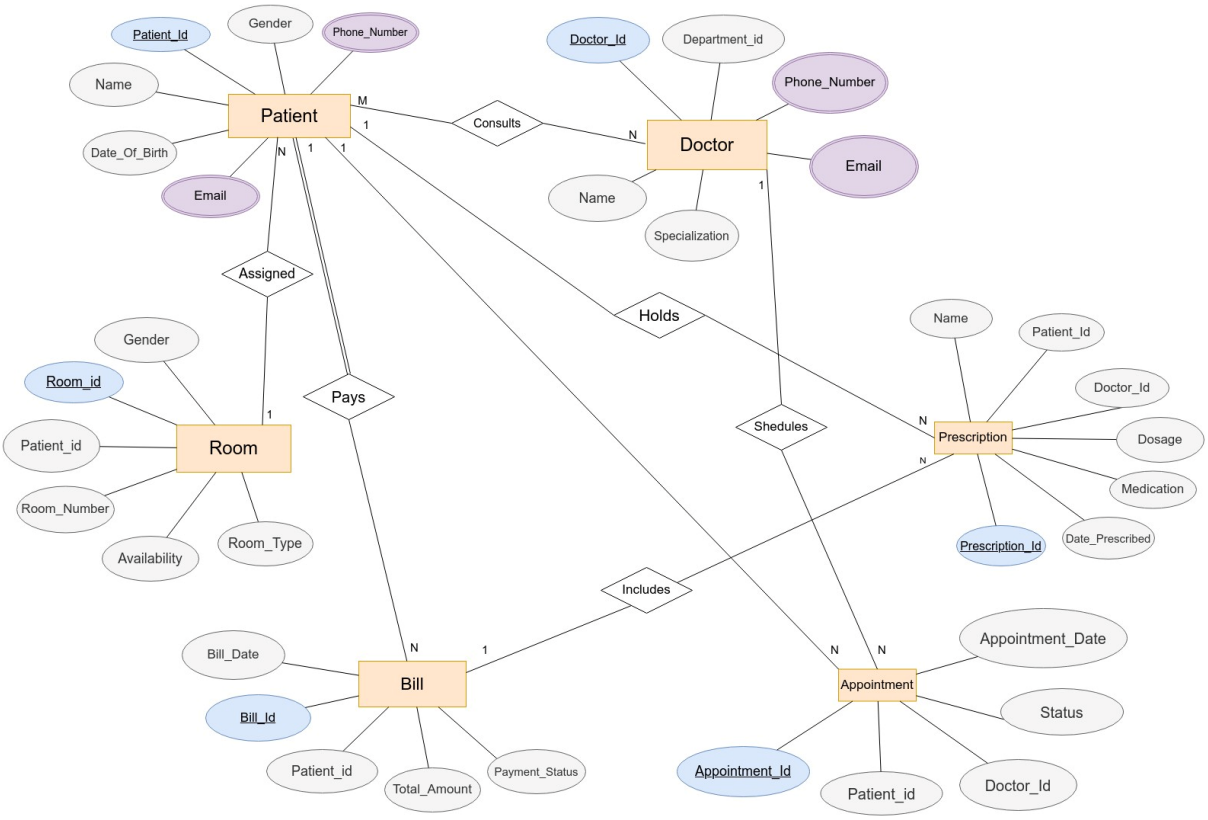
D. **Enhanced Billing and Revenue Management**:

- Simplify billing processes and improve revenue cycle management through accurate documentation of services rendered, insurance claims processing, and timely invoicing.
- Reduce billing errors and identify opportunities for revenue optimization, such as coding compliance and charge capture improvement.

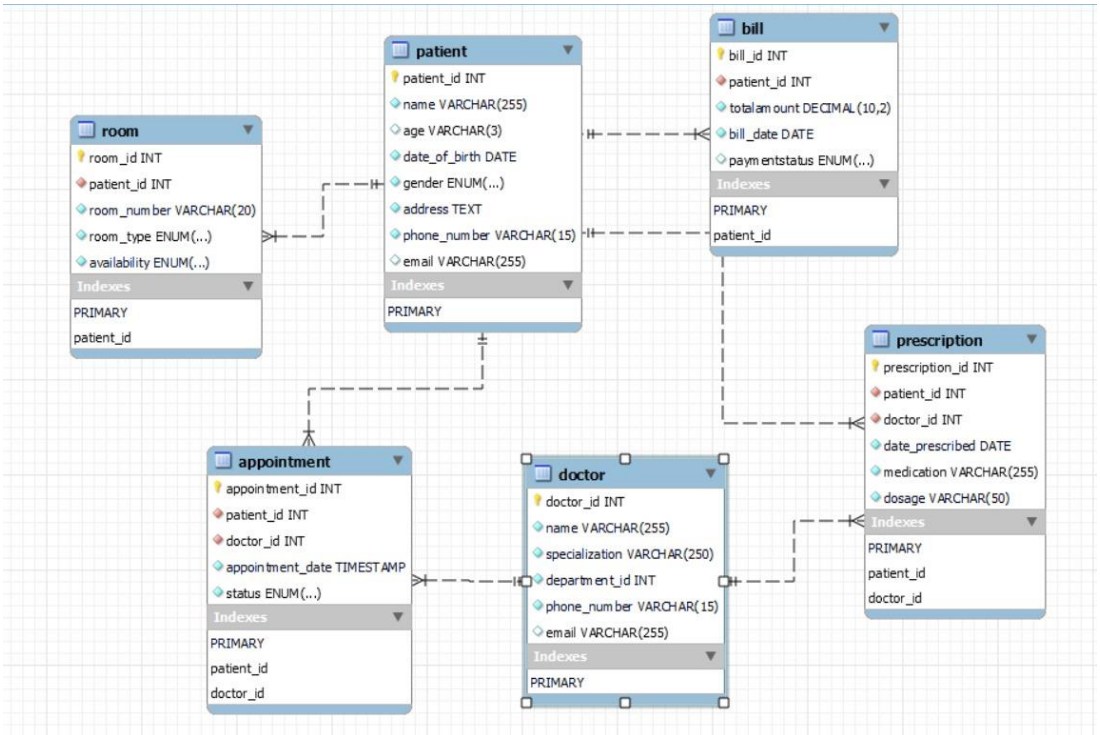
E. **Scalability and Flexibility**:

- Design a modular and scalable system architecture that can accommodate future growth and changes in healthcare practices and technology.
- Adapt the HMS to evolving healthcare needs and emerging technologies, such as interoperability standards and digital health innovations.

ER DIAGRAM:



RELATIONAL TABLE:



NORMALISATION:

Normalization involves breaking down tables into smaller, less redundant structures to minimize data anomalies and maintain data integrity.

First Normal Form (1NF):

- All attributes are atomic.
- All tables are already in 1NF.

Second Normal Form (2NF):

- All attributes fully depend on the primary key.
- All tables are in 2NF as there are no partial dependencies.

Third Normal Form (3NF):

- **All tables are in 2NF.**
- There are no transitive dependencies. Every non-prime attribute is non-transitively dependent on the primary key.

➢ Patient Table:

- All columns are atomic and directly related to the primary key (patient_id).
- There are no repeating groups.
- Therefore, the Patient table is in 1NF, 2NF, and 3NF.

➢ Doctor Table:

- All columns are atomic and directly related to the primary key (doctor_id).
- There are no repeating groups.
- Therefore, the Doctor table is in 1NF, 2NF, and 3NF.

➢ Appointment Table:

- All columns are atomic and directly related to the primary key (appointment_id).
- The appointment_date attribute depends only on the appointment_id.
- The status attribute depends only on the appointment_id.
- Therefore, the Appointment table is in 1NF and 2NF but not fully in 3NF due to the status attribute, which is transitively dependent on the appointment_id. To achieve 3NF, we could move the status attribute to a separate table where the primary key is appointment_id.

- **Prescription Table**:
- All columns are atomic and directly related to the primary key (prescription_id).
- There are no repeating groups.
- Therefore, the Prescription table is in 1NF, 2NF, and 3NF.

- **Room Table**:
- All columns are atomic and directly related to the primary key (room_id).
- The room_type attribute depends only on the room_id.
- The availability attribute depends only on the room_id.
- Therefore, the Room table is in 1NF and 2NF but not fully in 3NF due to the room_type and availability attributes, which are transitively dependent on the room_id. To achieve 3NF, we could move these attributes to separate tables where the primary key is room_id.

- **Bill Table**:
- All columns are atomic and directly related to the primary key (bill_id).
- There are no repeating groups.
- Therefore, the Bill table is in 1NF, 2NF, and 3NF.

The provided schema mostly conforms to the principles of normalization up to the second normal form (2NF). However, there are some dependencies in the Appointment and Room tables that violate the third normal form (3NF) and may require further normalization to achieve full compliance.

This normalized relational model ensures data integrity and minimizes redundancy, facilitating efficient data management within the Hospital Management System.

SQL IMPLEMENTATION:

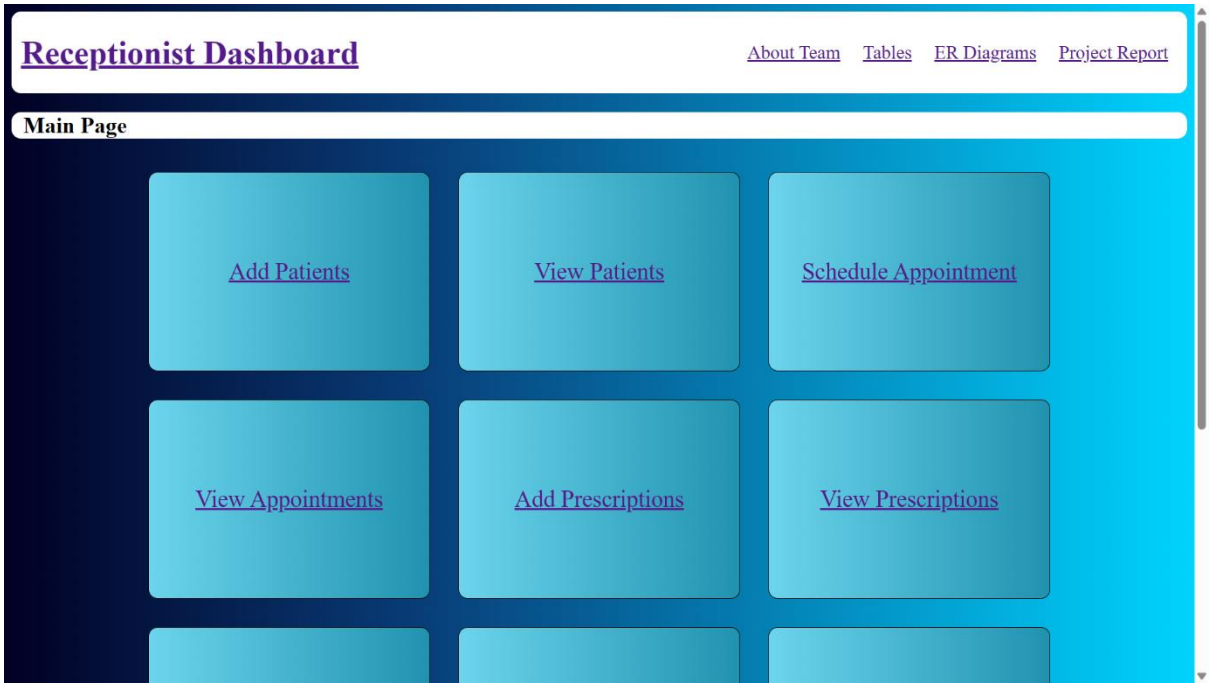
```
create database project;
USE project;

-- Patient Tabel
CREATE TABLE Patient (
    patient_id INT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    date_of_birth DATE NOT NULL,
    gender ENUM('Male', 'Female', 'Other') NOT NULL,
    address TEXT NOT NULL,
    phone_number VARCHAR(15) NOT NULL,
    email VARCHAR(255)
);
-- Doctor table
CREATE TABLE Doctor (
    doctor_id INT PRIMARY KEY,
    name VARCHAR(255) NOT NULL,
    specialization VARCHAR(250) NOT NULL,
    department_id INT NOT NULL,
    phone_number VARCHAR(15) NOT NULL,
    email VARCHAR(255)
);
-- Appointment table
CREATE TABLE Appointment (
    appointment_id INT PRIMARY KEY,
    patient_id INT NOT NULL,
    doctor_id INT NOT NULL,
    appointment_date TIMESTAMP NOT NULL DEFAULT NOW(),
    status ENUM('Scheduled', 'Cancelled', 'Completed') NOT NULL,
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctor(doctor_id)
);
-- Prescription table
CREATE TABLE Prescription (
    prescription_id INT PRIMARY KEY,
    patient_id INT NOT NULL,
    doctor_id INT NOT NULL,
    date_prescribed DATE NOT NULL,
    medication VARCHAR(255) NOT NULL,
    dosage VARCHAR(50) NOT NULL,
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id),
    FOREIGN KEY (doctor_id) REFERENCES Doctor(doctor_id)
);
-- Room table
CREATE TABLE Room (
```

```
    room_id INT PRIMARY KEY,
    patient_id INT NOT NULL,
    room_number VARCHAR(20) NOT NULL,
    room_type ENUM('Standard', 'Deluxe', 'ICU') NOT NULL,
    availability ENUM('Available', 'Occupied', 'Maintenance') NOT NULL,
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id)
);
-- Bill table
CREATE TABLE Bill (
    bill_id INT PRIMARY KEY,
    patient_id INT NOT NULL,
    totalamount DECIMAL(10, 2) NOT NULL,
    bill_date DATE NOT NULL,
    paymentstatus ENUM('DONE','PENDING'),
    FOREIGN KEY (patient_id) REFERENCES Patient(patient_id)
);
```

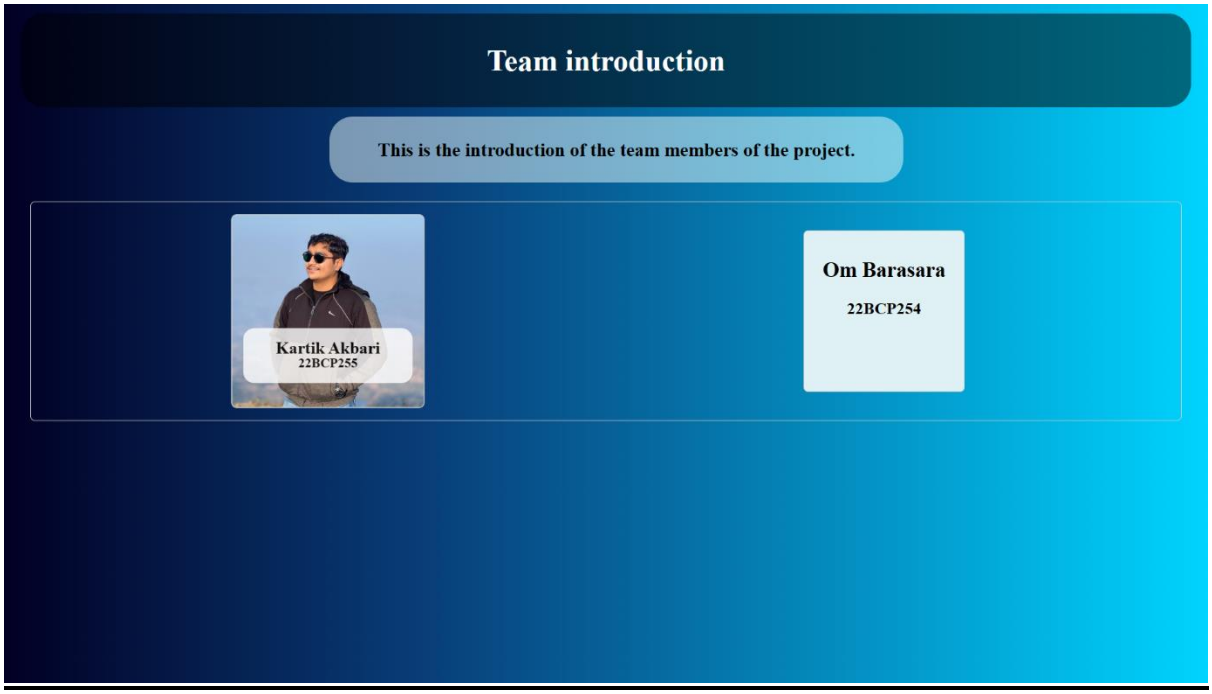
UI IMAGE OF THE PROJECT:

Front page of the project

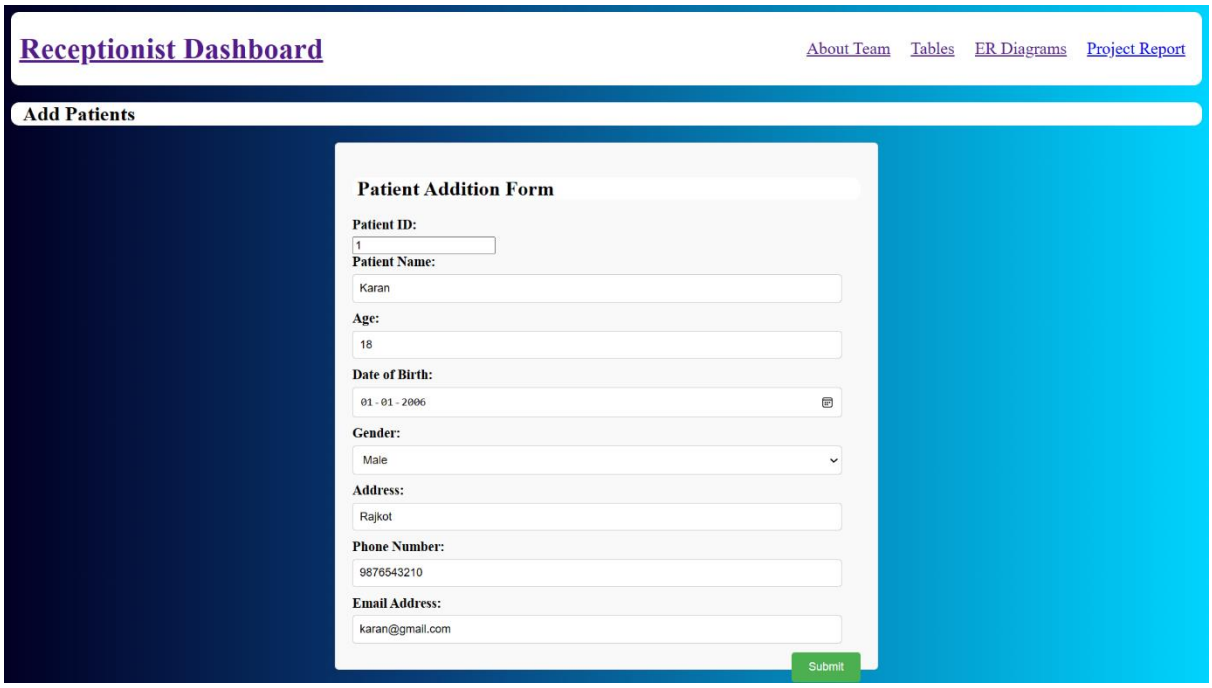


- In front page all the table are shown.
- Here Functionality of Team Introduction, Table, ER Diagram and Project Report are also available.

Team Introduction Page:



Patient Addition Page:

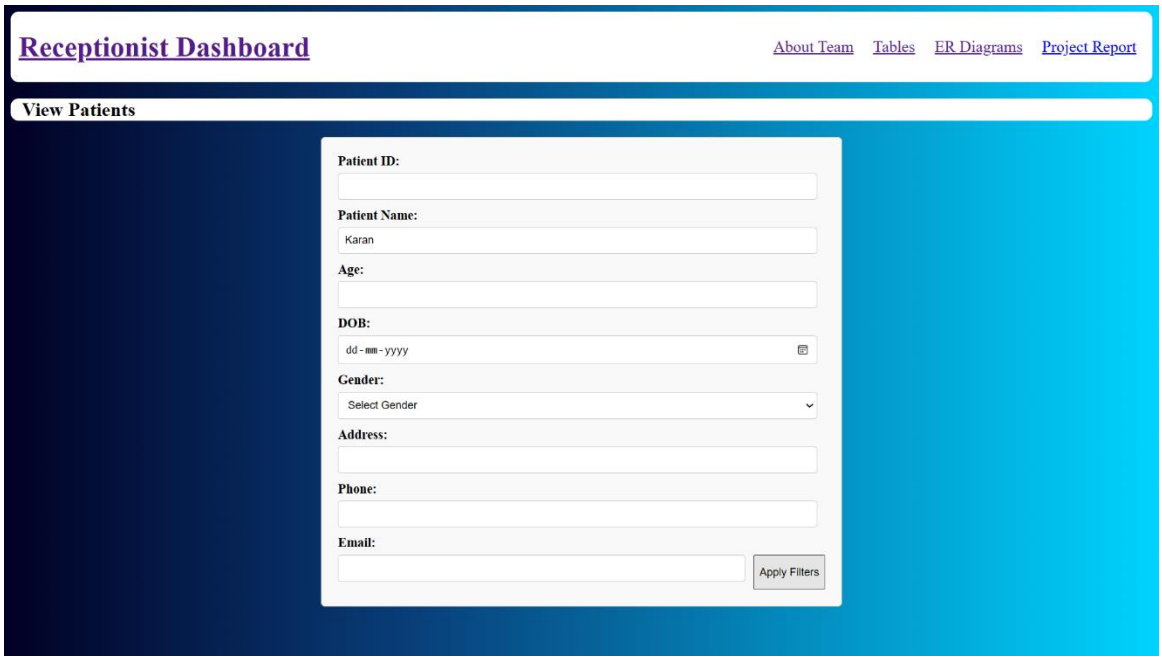


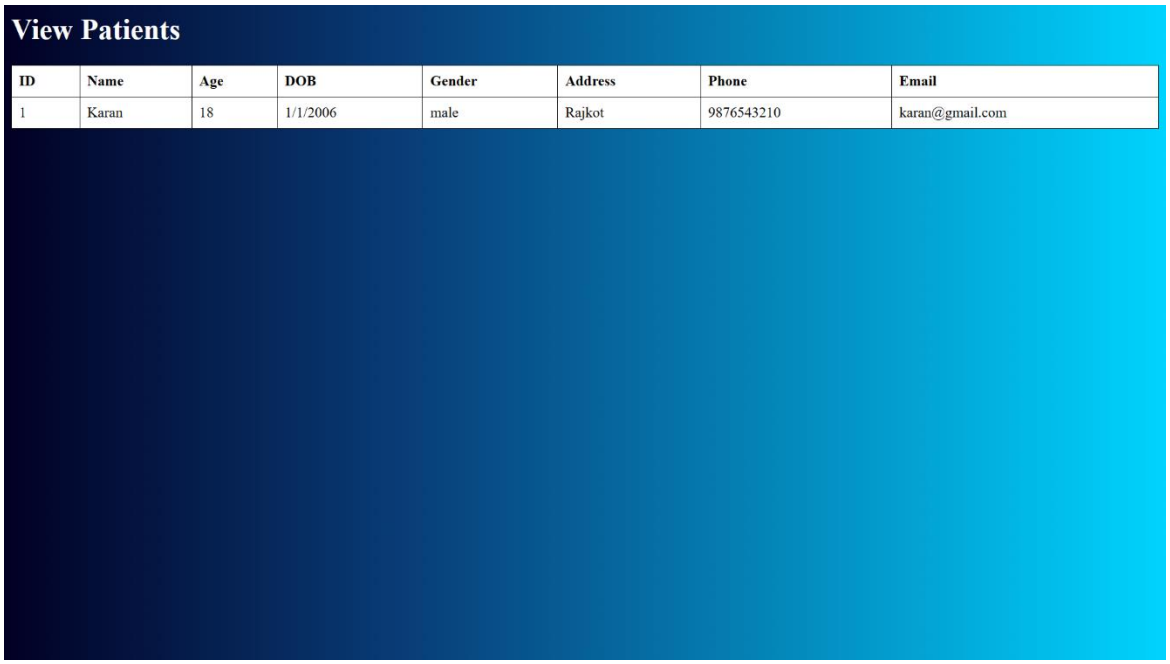
- In this page all the details of the patient will add

Page After Adding the Patient Details:

Patient added successfully

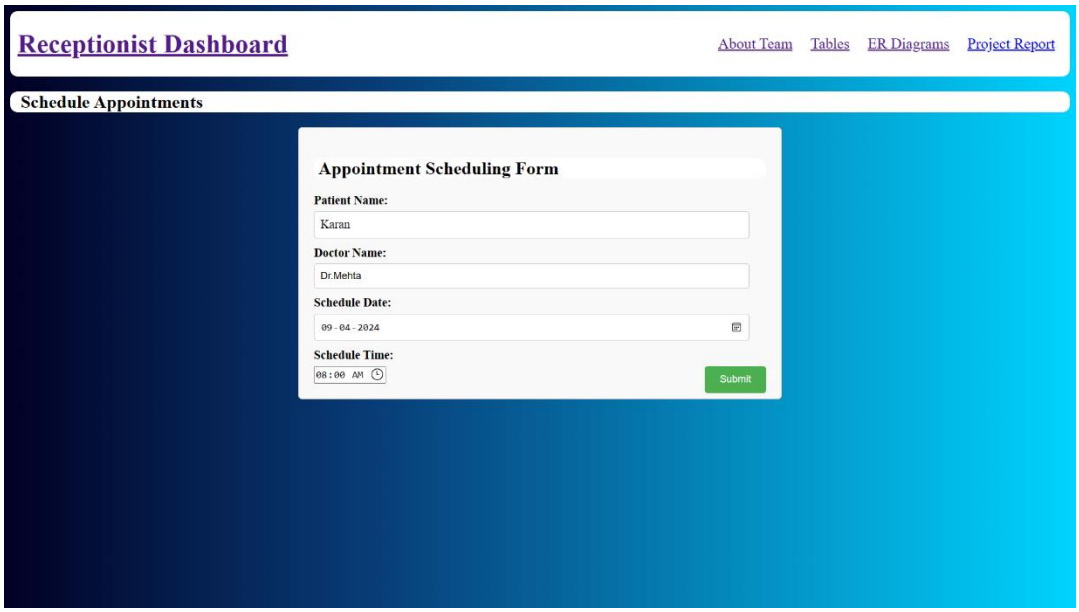
Patient View Page:



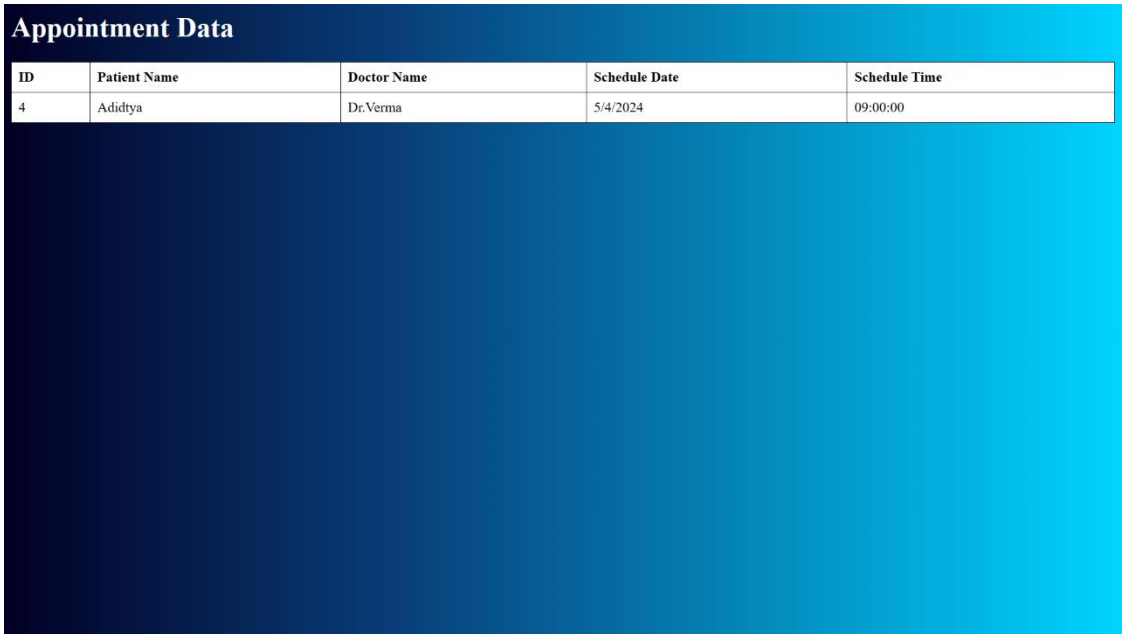


- Here all the patient details are stored in the backend.

Appointment Schedule Page:

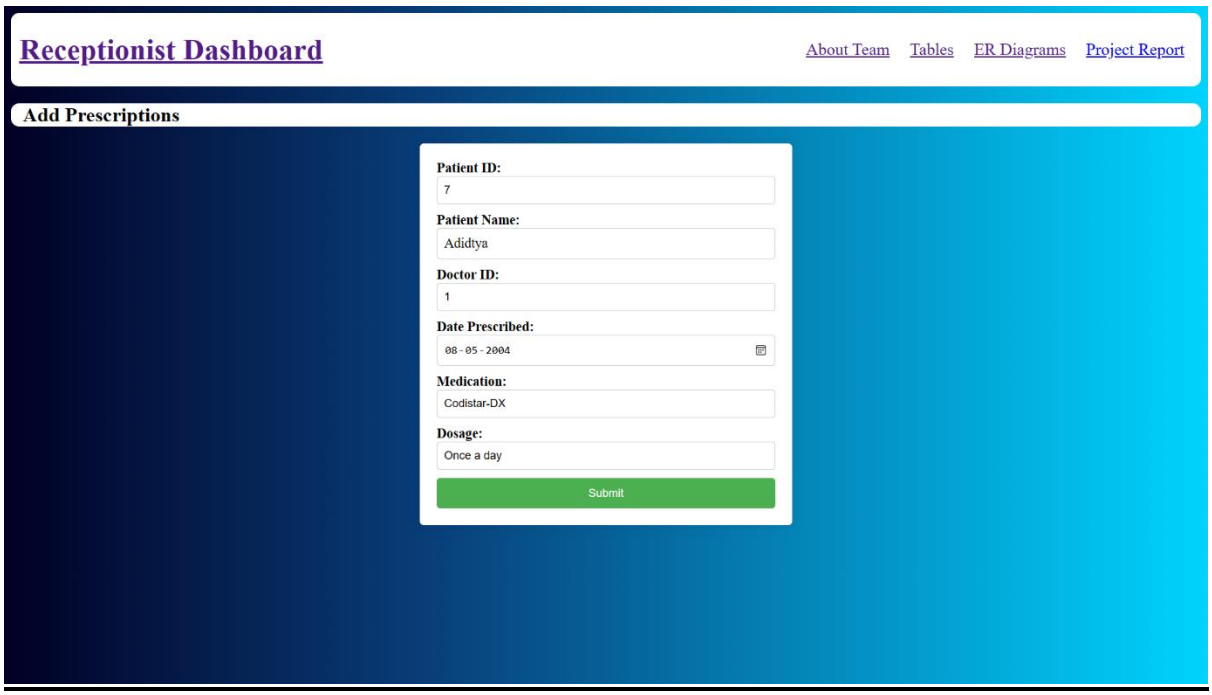


Appointment Data Page:



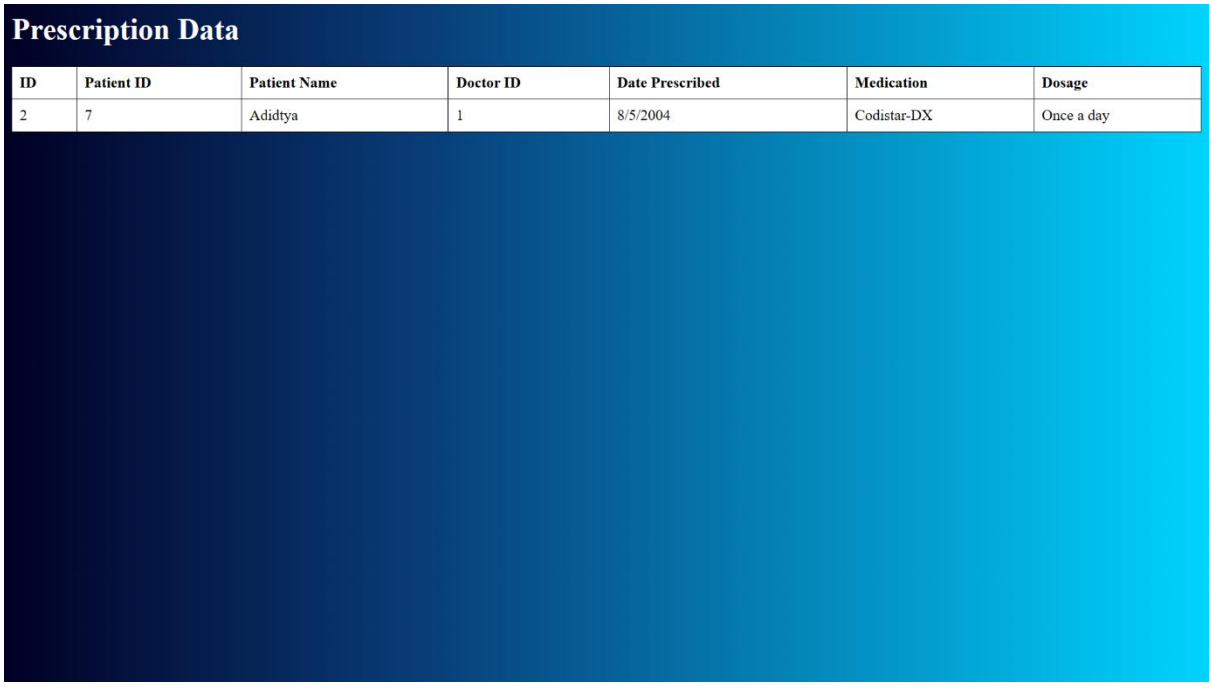
- In this page all the appointment data are stored.

Prescriptions Add Page:



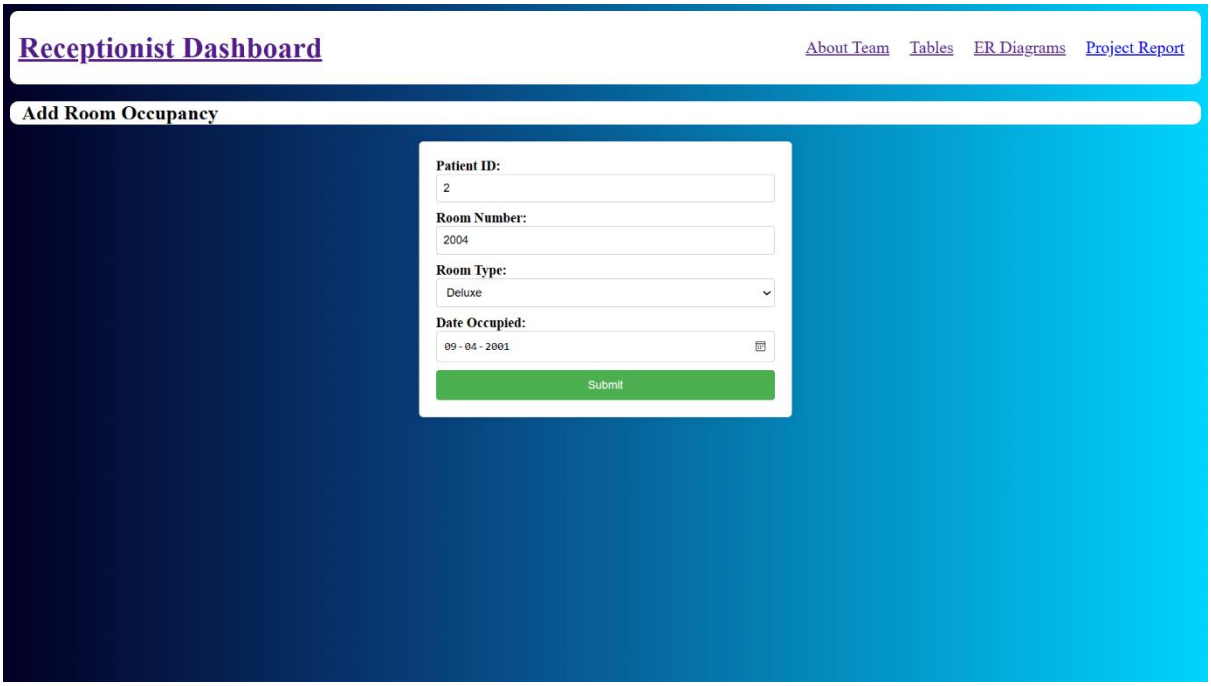
- In this page prescriptions details are add.

Prescriptions Data Page:

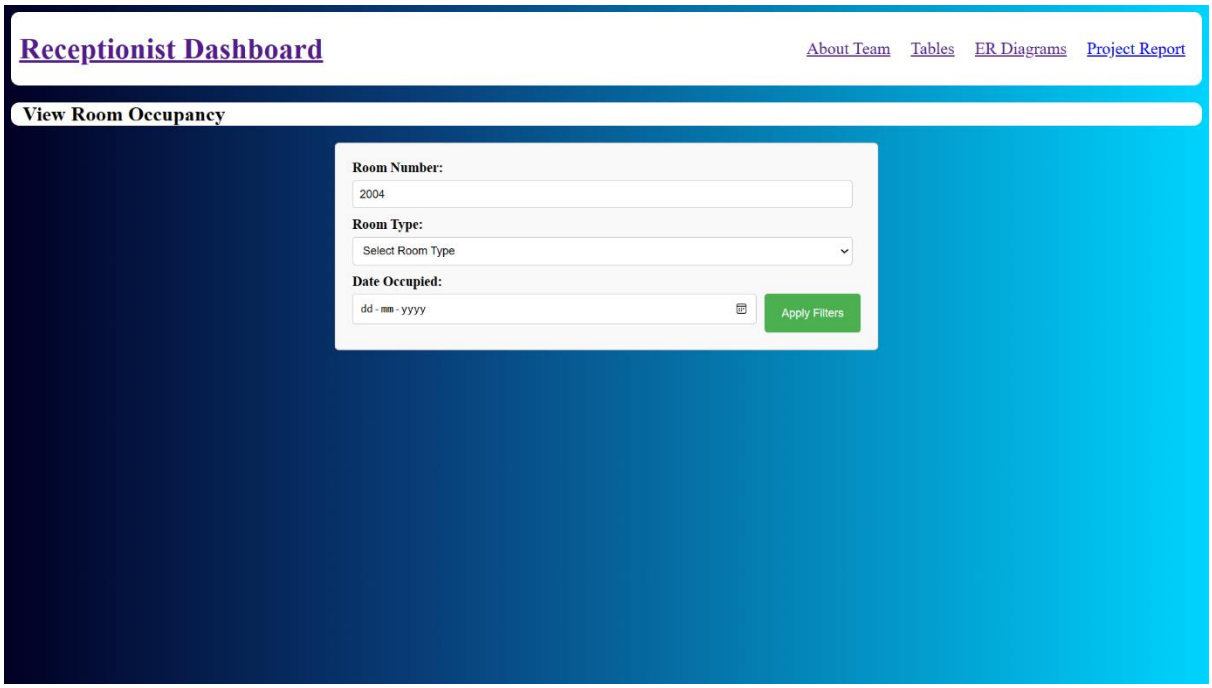


- All the prescription data stored in this page

Room Add Page:

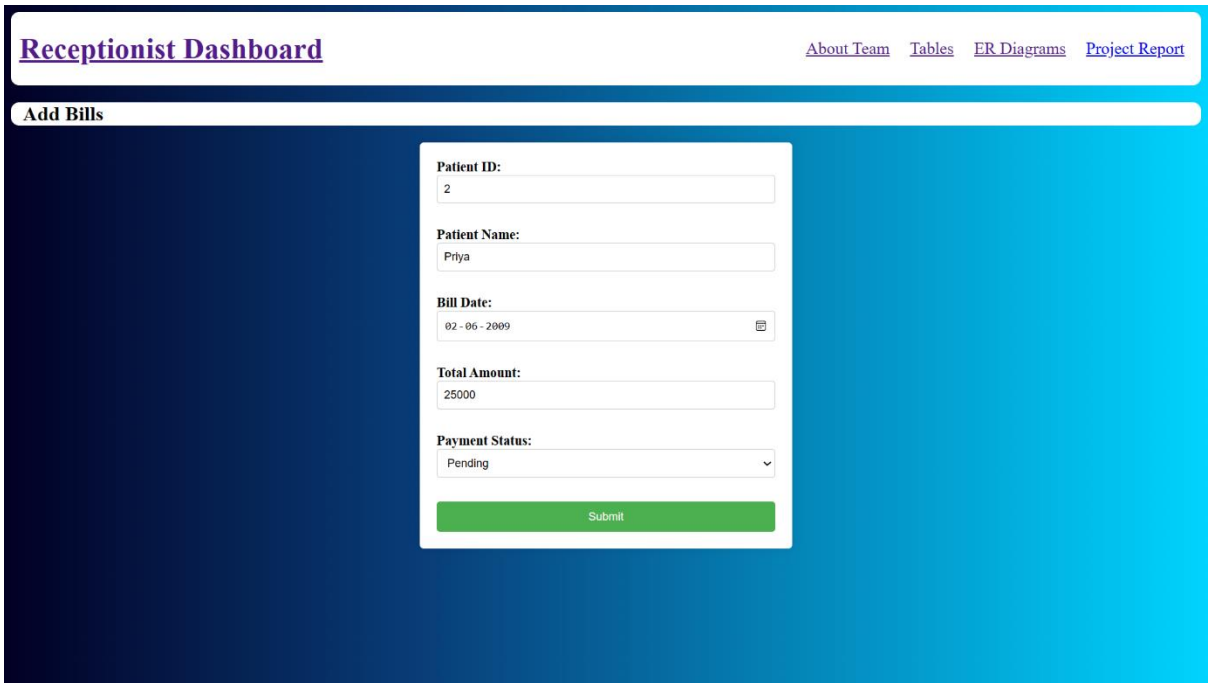


Room View Page:

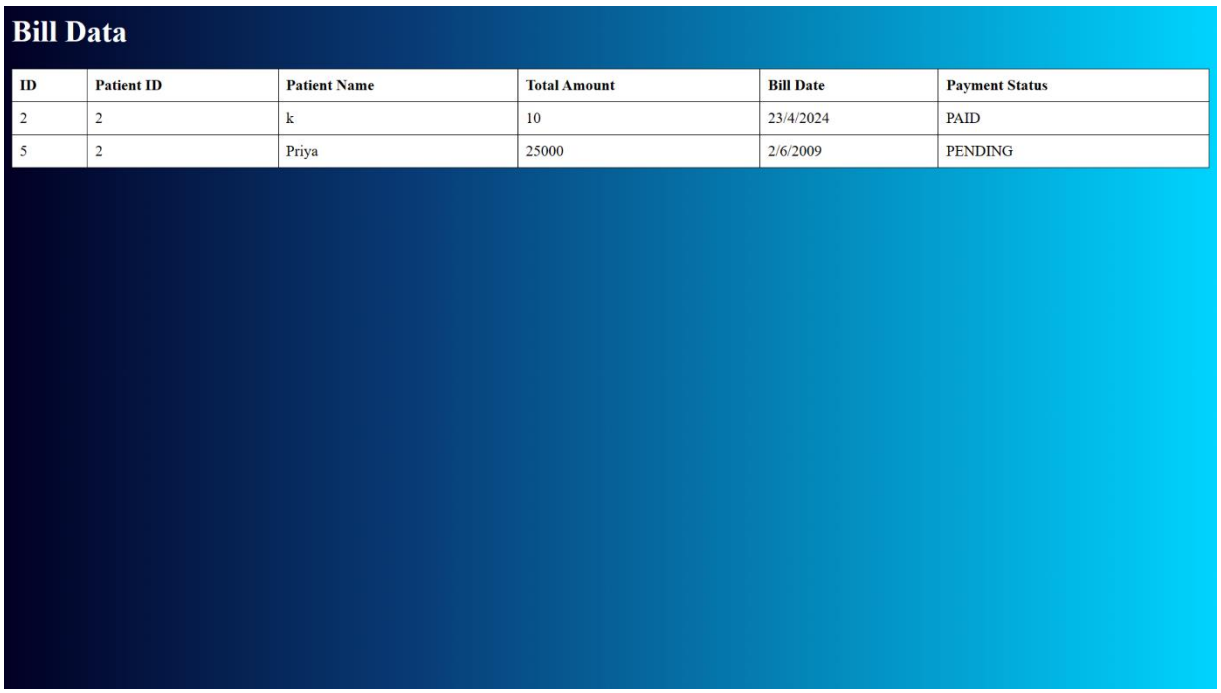


Room Data Page :

Bill Add Page:



Bill Data Page:



CONCLUSION:

Hospital Management System (HMS) project built on a reliable Database Management System (DBMS) serves as a cornerstone for enhancing healthcare delivery and management. Through meticulous planning, design, and implementation, such a system can address the multifaceted needs of modern healthcare institutions.