

ANUDIP FOUNDATION

A Project Report on

HOSPITAL MANAGEMENT SYSTEM

By

Batch: ANP-D0453

Student ID: AF0477089

Name: Yogita Pardeshi.

Under the Guidance of

Mrs. Rajshri Chandrabhan Thete

HOSPITAL MANAGEMENT SYSTEM

Introducing our Java-based Hospital Management System (HMS):

Efficiently manage all aspects of hospital operations with our robust and user-friendly system. From patient registration and doctor appointments to billing and medical records, our HMS streamlines workflows, enhances efficiency, and improves patient care. With features like staff management, test report tracking, and real-time room availability updates, our system ensures seamless hospital administration, allowing healthcare providers to focus on delivering exceptional medical services.

Entities:

- Patient Management:
- Doctor Management:
- Room Management:
- Nurse Management:
- Test Report Management:
- Record Management:
- Billing:
- Receptionist Management:

ATTRIBUTES OF ENTITIES:

1. Patient

❖ Attributes:

- P-ID
- Name
- DOBGender
- Mob-No
- Age

2. Employee

❖ Attributes:

- E-ID
- Name
- Salary
- Sex
- Mob-No
- Address:
- State
- City:
- Pin-no

3. Doctor

- ❖ Attributes:
 - E-ID (Foreign Key referencing Employee)
 - Department
 - Qualification

4. Nurse

- ❖ Attributes:
 - E-ID: E-ID is a foreign key linking a table to the Employee table through the Employee ID.

5. Room

- ❖ Attributes:
 - R-ID:
 - Type
 - Capacity
 - Availability

6. Receptionist

- ❖ Attributes:
 - E-ID (Foreign Key referencing Employee)

7. Test Report

- ❖ Attributes:
 - R-ID (Primary Key)
 - P-ID (Foreign Key referencing Patient)
 - Test Type
 - Result:

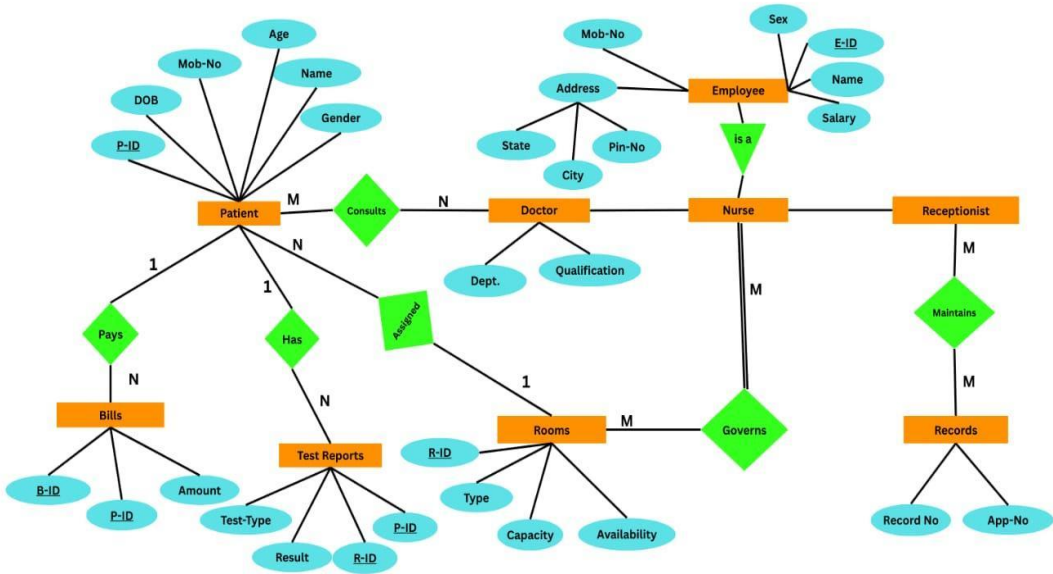
8. Bill

- ❖ Attributes:
 - B-ID
 - P-ID (Foreign Key referencing Patient)
 - Amount

9. Records

- ❖ Attributes:
 - Record-no
 - App-no:

ENTITY RELATIONSHIP DIAGRAM - HOSPITAL MANAGEMENT SYSTEM



CONCLUSION:

In conclusion, a Hospital Management System (HMS) serves as a robust and integrated solution for efficiently managing hospital operations, ensuring seamless coordination between patients, medical staff, and administrative functions. By digitizing critical processes such as patient registration, doctor assignments, room allocation, test report management, and billing, the system enhances accuracy, reduces manual workload, and minimizes errors.

Furthermore, the system streamlines record-keeping and resource allocation, allowing for better decision-making based on real-time data. The centralized database facilitates quick access to patient history, treatment plans, and financial transactions, leading to improved patient care and operational efficiency. By automating routine tasks, the HMS significantly enhances hospital workflow, staff productivity, and service quality, ultimately resulting in a more organized and patient-friendly healthcare environment.

Overall, the Hospital Management System modernizes healthcare administration, ensuring better patient management, optimized hospital resources.

DATABASE CREATION QUERY:

```
mysql> use hospital_management_system;  
Database changed
```

```
mysql> show tables;  
Empty set (0.14 sec)
```

```
mysql> CREATE TABLE Patient (  
-> P_ID INT AUTO_INCREMENT PRIMARY KEY,  
-> Name VARCHAR(100) NOT NULL,  
-> DOB DATE NOT NULL,  
-> Gender VARCHAR(10) NOT NULL,  
-> Mob_No VARCHAR(15) NOT NULL,  
-> Age INT NOT NULL  
-> );
```

Query OK, 0 rows affected (4.59 sec)

```
mysql> CREATE TABLE Employee (  
-> E_ID INT AUTO_INCREMENT PRIMARY KEY,  
-> Name VARCHAR(100) NOT NULL,  
-> Salary DECIMAL(10,2) NOT NULL,  
-> Sex VARCHAR(10) NOT NULL,  
-> Mob_No VARCHAR(15) NOT NULL,  
-> Address VARCHAR(255) NOT NULL,  
-> State VARCHAR(50) NOT NULL,  
-> City VARCHAR(50) NOT NULL,  
-> Pin_No VARCHAR(10) NOT NULL  
-> );
```

Query OK, 0 rows affected (3.73 sec)

```
mysql> CREATE TABLE Doctor (  
-> E_ID INT PRIMARY KEY,  
-> Department VARCHAR(100) NOT NULL,  
-> Qualification VARCHAR(100) NOT NULL,  
-> FOREIGN KEY (E_ID) REFERENCES Employee(E_ID) ON DELETE CASCADE  
-> );
```

Query OK, 0 rows affected (1.37 sec)

```
mysql> CREATE TABLE Nurse (  
-> E_ID INT PRIMARY KEY,  
-> FOREIGN KEY (E_ID) REFERENCES Employee(E_ID) ON DELETE CASCADE  
-> );
```

Query OK, 0 rows affected (0.80 sec)

```
mysql> CREATE TABLE Receptionist (  
-> E_ID INT PRIMARY KEY,  
-> FOREIGN KEY (E_ID) REFERENCES Employee(E_ID) ON DELETE CASCADE  
-> );
```

Query OK, 0 rows affected (0.78 sec)

```
mysql> CREATE TABLE Bill (  
-> B_ID INT AUTO_INCREMENT PRIMARY KEY,  
-> P_ID INT NOT NULL,  
-> Amount DECIMAL(10,2) NOT NULL,  
-> FOREIGN KEY (P_ID) REFERENCES Patient(P_ID) ON DELETE CASCADE  
-> );
```

Query OK, 0 rows affected (1.28 sec)

```
mysql> CREATE TABLE Records (  
-> Record_No INT AUTO_INCREMENT PRIMARY KEY,  
-> App_No INT NOT NULL  
-> );
```

Query OK, 0 rows affected (1.47 sec)

```
mysql> CREATE TABLE Room (  
-> R_ID INT AUTO_INCREMENT PRIMARY KEY,  
-> Type VARCHAR(50) NOT NULL,  
-> Capacity INT NOT NULL,  
-> Availability BOOLEAN NOT NULL DEFAULT TRUE  
-> );
```

Query OK, 0 rows affected (0.80 sec)

```
mysql> CREATE TABLE Test_Report (  
-> R_ID INT AUTO_INCREMENT PRIMARY KEY,  
-> P_ID INT NOT NULL,  
-> Test_Type VARCHAR(100) NOT NULL,  
-> Result VARCHAR(255) NOT NULL,  
-> FOREIGN KEY (P_ID) REFERENCES Patient(P_ID) ON DELETE CASCADE  
-> );
```

Query OK, 0 rows affected (0.63 sec)

```
mysql> show tables;
```

```
+-----+  
| Tables_in_hospital_management_system |  
+-----+  
| bill                                  |  
| doctor                              |  
| employee                             |  
| nurse                                |  
| patient                              |  
| receptionist                         |  
| records                              |  
| room                                 |  
| test_report                          |  
+-----+
```