DATABASE MANAGEMENT SYSTEMS LAB

Assignment: 4

Web Application Development for Hospital Management System

Group PHOENIX

Members:

Deepiha .S   20CS30015

Devendra Palod   20CS10024

Rohit Kumar Prajapati   20CS30041

Suhas A M   20CS10066

Tanmay Mohanty   20CS10089

INTRODUCTION

The healthcare industry is constantly evolving, and technological advancements have become integral to improving the quality of patient care. A Hospital Management System (HMS) is an essential tool for healthcare providers, enabling them to streamline their operations and improve patient outcomes. Web application development has revolutionized the way healthcare providers manage patient care and operational workflows.

This report provides complete information of the system including the schemas, technologies used, requirements and web interface.

DATABASE SCHEMA / CLASSES

Class Doctor:

```python
class Doctor(models.Model):
    user=models.OneToOneField(User,on_delete=models.CASCADE)
    profile_pic=
models.ImageField(upload_to='profile_pic/DoctorProfilePic/',null=True,blank=True)
    address = models.CharField(max_length=40)
    mobile = models.CharField(max_length=20,null=True)
    department=
models.CharField(max_length=50,choices=departments,default='Cardiologist')
```

Each instance will have a unique ID (user) and the features include profile pic, address (char of max length 40), mobile (char of max length 20 and could be null/empty) , department (char of max length 50 and default set to 'Cardiologist'). Here the departments refers to list of all possible departments:

```
departments=[('Cardiologist','Cardiologist'),
('Dermatologists','Dermatologists'),
('Emergency Medicine Specialists','Emergency Medicine Specialists'),
('Allergists/Immunologists','Allergists/Immunologists'),
('Anesthesiologists','Anesthesiologists'),
('Colon and Rectal Surgeons','Colon and Rectal Surgeons')
]
```

Class Patient:

```
user=models.OneToOneField(User,on_delete=models.CASCADE)
    profile_pic=
models.ImageField(upload_to='profile_pic/PatientProfilePic/',null=True,blank=True)
    address = models.CharField(max_length=40)
    mobile = models.CharField(max_length=20,null=False)
    symptoms = models.CharField(max_length=100,null=False)
    assignedDoctorId = models.PositiveIntegerField(null=True)
    admitDate=models.DateField(auto_now=True)
    status=models.BooleanField(default=False)
```

Each patient is assigned a doctor and is linked by the doctor ID. Contains patient describing fields like symptoms, admitDate.

Class Appointment:

```
class Appointment(models.Model):
    Regular=1
    Emergency=2

    PRIOR=[
        (Regular, 'Regular'), (Emergency, 'Emergency'),
```

```
    ]

    patientId=models.PositiveIntegerField(null=True)
    doctorId=models.PositiveIntegerField(null=True)
    patientName=models.CharField(max_length=40,null=True)
    doctorName=models.CharField(max_length=40,null=True)
    appointmentDate=models.DateTimeField(null=True)
    description=models.TextField(max_length=500)
    status=models.BooleanField(default=False)
    priority=models.IntegerField(choices=PRIOR, null=True)
    room = models.CharField(max_length=15, null=True)
```

Every instance of appointment has fields patientID, doctorID which links the appointment to a patient and a doctor.

Also has some self-describing fields like appointmentDate, description, etc.

Class Test:

```
class Test(models.Model):
    patientId=models.PositiveIntegerField(null=True)
    doctorId=models.PositiveIntegerField(null=True)
    patientName=models.CharField(max_length=40,null=True)
    doctorName=models.CharField(max_length=40,null=True)
    testDate=models.DateField(auto_now=True)
    testName=models.CharField(max_length=40,null=True)
    testResult=models.CharField(max_length=40,null=True)
```

The Test class refers to the tests prescribed by doctor to a patient and hence patientID and DoctorID in the fields. And other important fields like testDate, testName (test Names are unique), etc.

Class Treatment:

```
class Treatment(models.Model):
    patientId=models.PositiveIntegerField(null=True)
    doctorId=models.PositiveIntegerField(null=True)
    patientName=models.CharField(max_length=40,null=True)
    doctorName=models.CharField(max_length=40,null=True)
```

```
    treatmentDate=models.DateField(auto_now=True)
    treatmentName=models.CharField(max_length=40,null=True)
    description=models.TextField(max_length=500)
```

The Treatment class refers to the treatment a patient is undergoing as prescribed by the doctor and hence has patientID and doctorID fields.

Class PatientDischargeDetails:

```
class PatientDischargeDetails(models.Model):
    patientId=models.PositiveIntegerField(null=True)
    patientName=models.CharField(max_length=40)
    assignedDoctorName=models.CharField(max_length=40)
    address = models.CharField(max_length=40)
    mobile = models.CharField(max_length=20,null=True)
    symptoms = models.CharField(max_length=100,null=True)

    admitDate=models.DateField(null=False)
    releaseDate=models.DateField(null=False)
    daySpent=models.PositiveIntegerField(null=False)

    roomCharge=models.PositiveIntegerField(null=False)
    medicineCost=models.PositiveIntegerField(null=False)
    doctorFee=models.PositiveIntegerField(null=False)
    OtherCharge=models.PositiveIntegerField(null=False)
    total=models.PositiveIntegerField(null=False)
```

When a patient is getting discharged, the room charges, medicinal expenses, doctor fees are all calculated for the payment. Also, the details of the stay like date of admission, date of release, days spent are all recorded.

Class Room:

```
class Room(models.Model):
    room_no=models.CharField(max_length=15)
    curr_availability=models.PositiveIntegerField()
    max_capacity=models.PositiveIntegerField()
```

Each room has a unique room_no. Each room has some maximum capacity max_capacity and current slots left in the room is the curr_availability.

TECHNOLOGY

Python language is used for the overall implementation of the project. And Django (Python based) is used as the high-level web framework.

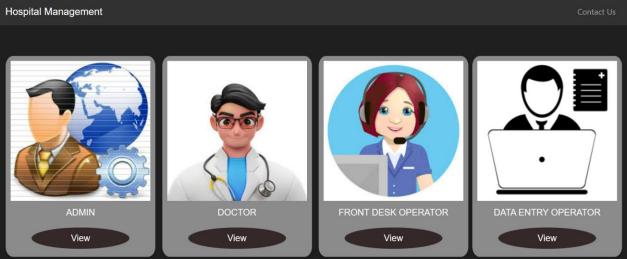For the front end of the web application, HTML, CSS and Javascript are used.

Libraries used are mentioned in the requirements.txt file.
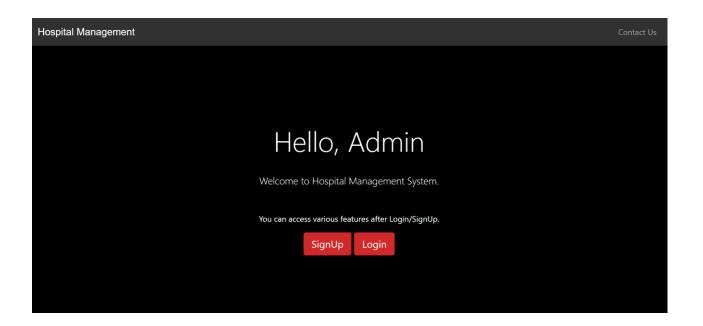
Can be run in both Windows and Linux operating system.

WEB INTERFACE:

- **Home page**

• Admin

The administrator has the most control and can register, update, or delete users and accommodate rooms by clicking appropriate links on the sidebar.

Hello, Admin

Welcome to Hospital Management System.

You can access various features after Login/SignUp.

SignUp   Login

> ## Signup:

Admin can signup here by filling the required details and setting a password.

Add New Admin To Hospital

First Name	Last Name
Username	Password

Submit

Already have an account? Login here

> ## Login:

Admin can login from here by filling the username and password.

> ➤ **Dashboard:**

This is the dashboard of an administrator containing information like the total users, recent records and the required links on the sidebar.

➢ Doctor Records/Registration:

Doctor records can be viewed, updated, deleted and registered from here.

Homepage > Admin > login > Doctor

Homepage > Admin > login > Doctor > Doctor Record

Doctor records can be updated and deleted from here

Below is the record updating form:

Homepage > Admin > Login > Doctor > Doctor Record > Update

Below is the Doctor Registration form:

Homepage > Admin > Login > Doctor > Register Doctor

➢ **Front Desk Operator Records/Registration:**

Front Desk Operator records can be viewed, deleted and registered from here as done in the Doctor.

➢ **Data Entry Operator Records/Registration:**

Data Entry Operator records can be viewed, deleted, and registered from here as done in the Doctor.

➢ **Rooms:**

Records of rooms can be accessed from here and also new rooms can be added giving their details.

Homepage > Admin > Login > Rooms

Homepage > Admin > Login > Doctor > Rooms > Current Status of Room

This gives a list of all rooms and their current Availability.

New Room form:

Homepage > Admin > Login > Doctor > Rooms > Add new room

Unique room number, its availability, maximum capacity can be filled and registered.

• Doctor
➢ Login:

Homepage > Doctor > Login

➢Dashboard:

Homepage > Doctor > Login > Dashboard

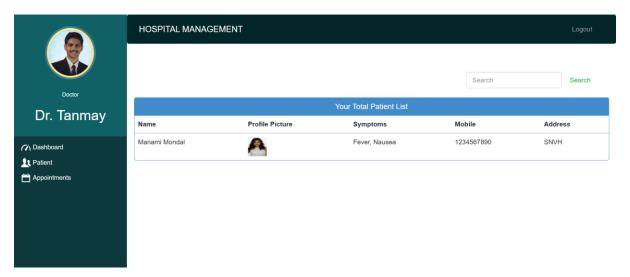The above dashboard contains some quick information and the required links.
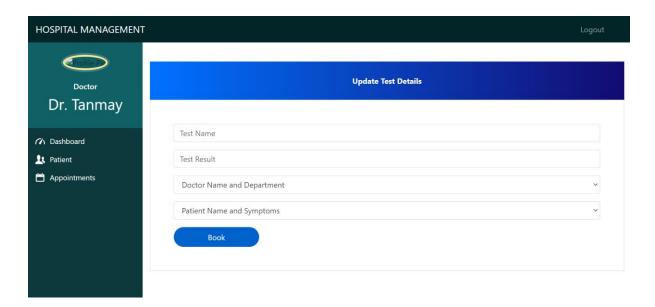
➢Patient:

Homepage > Doctor > Login > Patient

Features related to patient records, discharged patients, prescribing a test and treatment can be accessed from here.

Homepage > Doctor > Login > Patient > Your Patient Record

Homepage > Doctor > Login > Patient > Prescribe a test

Here the doctor can prescribe a test and update its result.

Homepage > Doctor > Login > Patient > Treat Patient

Here the Doctor inputs the Treatment steps taken and its results.

Your Discharged Patients gives the list of all discharged patients with all its details.

➤ Appointments:

Here the Doctor can view his/her appointments and delete them.

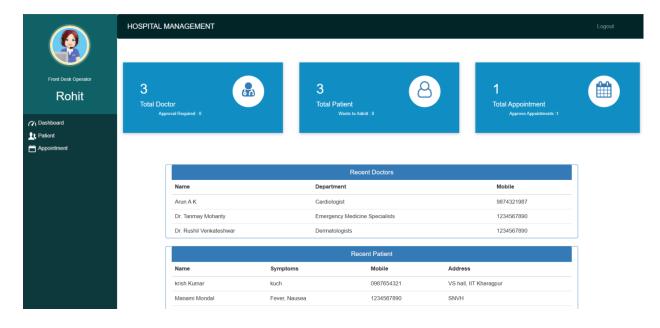Homepage > Doctor > Login > Appointments

• Front Desk operator

A front desk operator books and manages appointments. Also, admission and discharging of patients are managed by the front desk operator.

➤ Login:

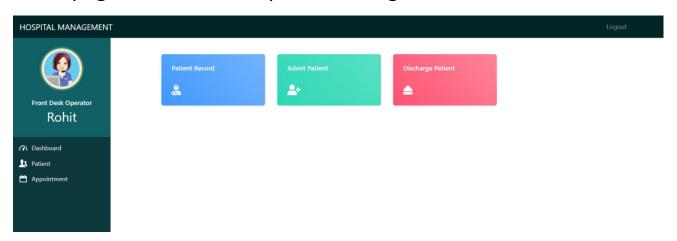Login similar to previously shown users using the username and the password.

➤ Dashboard:

Homepage > Front Desk Operator > Login > Dashboard

As other dashboards, it gives quick information and required links on the side bar.
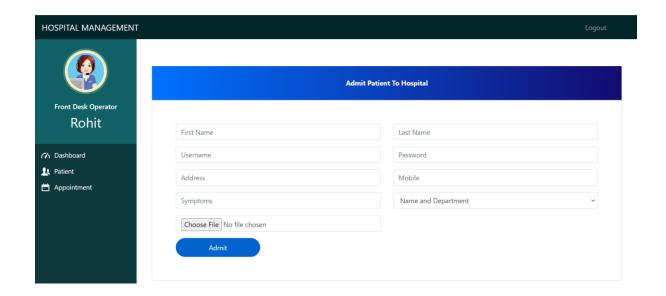
➢**Patient:**
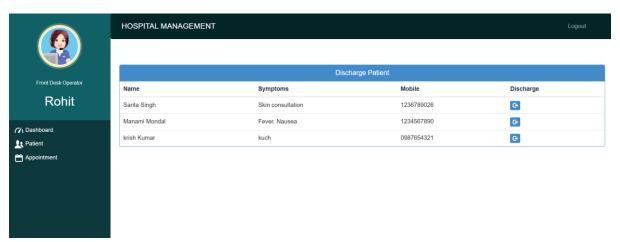
Homepage > Front Desk Operator > Login > Patient

Admit Patient form:

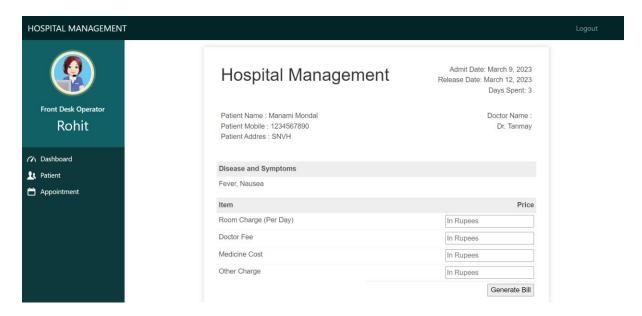Homepage > Front Desk Operator > Login > Patient

Homepage > Front Desk Operator > Login > Patient > Discharge Patient
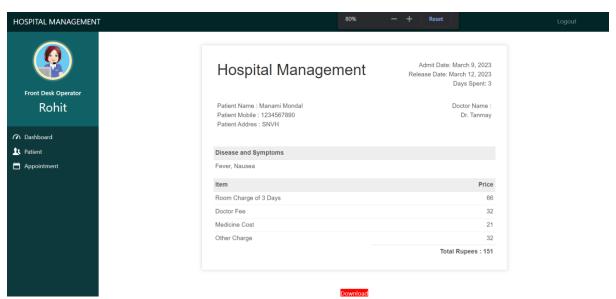

Patient Discharge form:

Homepage > Front Desk Operator > Login > Patient > Discharge Patient > Discharge
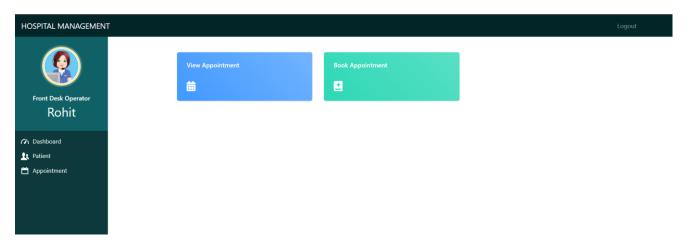
All the details of individual expenses are filled and the bill can be generated.

Homepage > Front Desk Operator > Login > Patient > Discharge Patient > Discharge > Generate Bill

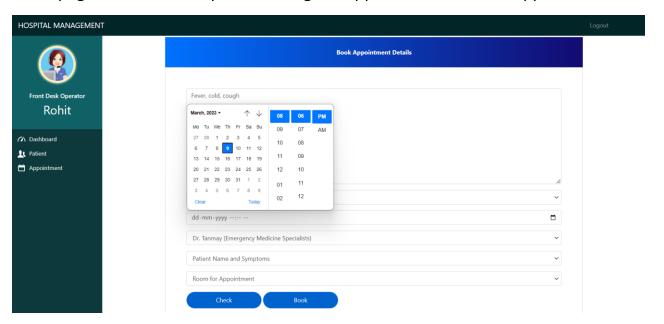The total generated bill including all the details of the patient's discharge can be downloaded from here.

➢Appointments:

Appointments can be viewed and booked from here.

Book appointment form:

Homepage > Front Desk Operator > Login > Appointments > Book Appointment

Here an appointment can be booked by filling in the details like symptoms, doctor specialization, etc and can select the appointment date by the calendar.

Finally, the availability can be checked and the appointment can be booked successfully.
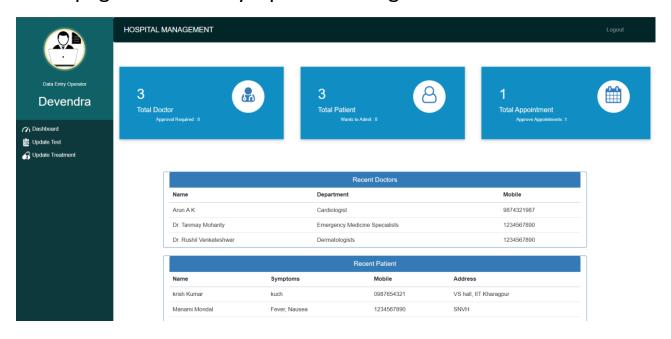
- **Data Entry Operator**

A Data Entry Operator can update data related to tests and treatments

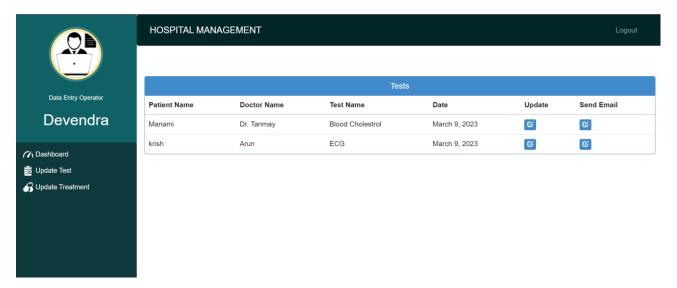➢**Login:**

Login by username and password.

➢**Dashboard:**

Homepage > Data Entry Operator > Login > Dashboard

➢**Updating Tests:**

The test results can be updated from here from the list of all tests currently prescribed. Also the

test results could be mailed to the patient's registered email id.

Homepage > Data Entry Operator > Login > Update Test

➤**Update Treatments:**
The data entry operator can also update treatments the patients are undergoing, from here.

There can be many other features seen, like search bar for searching a particular record from a list, contact us feature, etc.