



North South University

Title: Patient and Treatment Management in a Hospital

Project ID : 01

Group ID : 01

Course Code : CSE 311

Section : 02

Date : 04-12-2024

Name	ID
Md Abu Bokar	2222977642
Natasha Monir Shawon	2121744642

Submitted To

Faculty : Dr. Abu Sayed Mohammad Latiful Hoque [SLf]

Instructor : Sadia Afrin

→ Project Title :

- ◆ Patient and treatment management in a hospital

→ Objective :

- ◆ To design and develop a database management system for patients and treatments in a hospital.

→ Project scope :

- ◆ This project focuses on building a robust database management system that manages a hospital. Especially focused on patients and their treatments. This project store's patient's details including name, address, phone number etc. This project helps to manage a patient's medical history, make appointments with a doctor, maintain a patient's treatment etc in an easier way. It supports essential data management functionalities using crud operations. A better way to manage patients in a hospital.

→ Features :

- ◆ **Patient Registration :**
 - Record patient details like name, contact info, address, and personal information
 - Maintain medical history, including prescriptions, services, tests etc.
 - CRUD operations to add, view, update, or delete patient information.
- ◆ **Treatment Plan Management :**
 - Create a treatment plan for patients.
 - Create and update detailed treatment plans for patients based on doctor recommendations.
- ◆ **Doctor and Appointments :**
 - Schedule appointments between doctors and patients.
 - Allows doctors to see their appointment list.
 - CRUD operations to add, view, update, or delete doctor information.
- ◆ **Medical Records Management :**
 - Keeping a record of patient prescriptions, tests, and diagnoses.
 - Keep track of each patient's medical history , treatment plans.
- ◆ **Billing System :**
 - Calculate costs based on provided services, treatments, and tests.
 - Generate detailed bills for each patient.

→ Relation Schema :

- ◆ admin_details (admin_id (PK), name, email, password, role)
- ◆ department (dept_id (PK), dept_name)
- ◆ patient_details (patient_id (PK), name, email, phone_no, address, password, gender, blood_group, dob, height, weight, occupation, role)
- ◆ doctor_details (doctor_id (PK), name, email, phone_no, address, password, gender, speciality, dept_id, role)
- ◆ treatment_plan (treatment_id (PK), patient_id (FK), diagnosis, medications, plan_details)
- ◆ appointment (appointment_id (PK), doctor_id (FK), patient_id (FK), appointment_date, appointment_time)
- ◆ prescription (prescription_id (PK), patient_id (FK), doctor_id (FK))
- ◆ medicine (medicine_id (PK), medicine_name, medicine_quantity, medicine_price)
- ◆ service (service_id (PK), patient_id (FK), service_name (PK), service_cost (PK))
- ◆ test (test_id (PK), treatment_id (FK), test_name (PK), test_cost (PK))
- ◆ bill (bill_id (PK), treatment_id (FK), amount)
- ◆ medical_record (record_id (PK), patient_id (FK), doctor_id (FK))
- ◆ dept_head(head_id, doctor_id)
- ◆ regular (patient_id (FK), insurance_provider, policy_number)
- ◆ emergency (patient_id (FK), allergies, condition)
- ◆ admin_request (request_id (PK), name, email, password, role)
- ◆ prescription_medicines (prescription_id (FK), medicine_name)

→ SQL DDL for Relation Schema :

-- 1. Create admin table

```
CREATE TABLE admin_details(  
    admin_id INT AUTO_INCREMENT,  
    name VARCHAR(50) NOT NULL,  
    email VARCHAR(80) NOT NULL,  
    password VARCHAR(150) NOT NULL,  
    role VARCHAR(10) NOT NULL,  
    PRIMARY KEY(admin_id)  
);  
  
INSERT INTO admin_details (name, email, password, role)  
VALUES  
    ('Md Abu Bokar', 'abu@gmail.com', 'testpass1', 'admin'),  
    ('Natsha Monir Shawon', 'nat@gmail.com', 'testpass2', 'admin');
```

-- 2. Create department table

```
CREATE TABLE department(  
    dept_id INT AUTO_INCREMENT NOT NULL,  
    dept_name VARCHAR(55) NOT NULL,  
    PRIMARY KEY(dept_id)  
);  
  
INSERT INTO department (dept_name)  
VALUES  
    ('Ophthalmology'),  
    ('Cardiology');
```

-- 3. Create doctor table

```
CREATE TABLE doctor_details(  
    doctor_id INT AUTO_INCREMENT,  
    name VARCHAR(50) NOT NULL,  
    email VARCHAR(80) UNIQUE NOT NULL,  
    phone_no VARCHAR(15) UNIQUE NOT NULL,  
    address VARCHAR(150),  
    password VARCHAR(150) NOT NULL,  
    gender VARCHAR(10),  
    speciality VARCHAR(55) NOT NULL,  
    dept_id INT NOT NULL,  
    role VARCHAR(10) NOT NULL,  
    PRIMARY KEY(doctor_id),  
    FOREIGN KEY(dept_id) REFERENCES department(dept_id)  
);  
  
INSERT INTO doctor_details (name, email, phone_no, address, password, gender,  
speciality, dept_id, role)  
  
VALUES  
  
('Nafis Anzum', 'nafis@gmail.com', '31234567890', 'Bangladesh', 'docpass1',  
'Male', 'Ophthalmologist', 1, 'doctor'),  
  
('Ava Brown', 'ava@gmail.com', '41234567890', 'Australia', 'docpass2', 'Female',  
'Cardiologist', 2, 'doctor');
```

-- 4. Create patient table

```
CREATE TABLE patient_details(  
    patient_id INT AUTO_INCREMENT,  
    name VARCHAR(50) NOT NULL,  
    email VARCHAR(80) UNIQUE NOT NULL,  
    phone_no VARCHAR(15) UNIQUE NOT NULL,  
    address VARCHAR(150),  
    password VARCHAR(150) NOT NULL,  
    gender VARCHAR(10),  
    blood_group VARCHAR(5),  
    dob VARCHAR(10),  
    height DOUBLE(5,2),  
    weight DOUBLE(5,2),  
    occupation VARCHAR(50),  
    role VARCHAR(10) NOT NULL,  
    PRIMARY KEY(patient_id)  
);  
  
INSERT INTO patient_details  
(name, email, phone_no, address, password, gender, blood_group, dob, height, weight,  
occupation, role)  
  
VALUES  
  
('Emily Parker', 'emily@gmail.com', '91234567890', 'Usa', 'patpass1', 'Female',  
'O+', '1990-05-12', 1.65, 55.50, 'Teacher', 'patient'),  
  
('Sophia Johnson', 'sophia@gmail.com', '92345678901', 'Russia', 'patpass2!',  
'Female', 'A+', '1985-08-20', 1.70, 60.00, 'Software Engineer', 'patient'),
```

```
('Md Akib', 'akib@gmail.com', '95678901234', 'Bangladesh', 'patpass3', 'Male',  
'B+', '1995-03-07', 1.68, 62.35, 'Architect', 'patient');
```

-- 5. Create treatment table

```
CREATE TABLE treatment_plan(  
  
    treatment_id INT AUTO_INCREMENT,  
  
    patient_id INT,  
  
    doctor_id INT,  
  
    diagnosis VARCHAR(50),  
  
    medications VARCHAR(100),  
  
    plan_details VARCHAR(100),  
  
    PRIMARY KEY (treatment_id ),  
  
    FOREIGN KEY (patient_id) REFERENCES patient_details(patient_id),  
  
    FOREIGN KEY (doctor_id) REFERENCES doctor_details(doctor_id)  
  
);  
  
INSERT INTO treatment_plan (patient_id, doctor_id, diagnosis, medications,  
plan_details)  
  
VALUES  
  
    (1, 1, 'Fracture', 'Paracetamol, Ibuprofen', 'Patient to take rest, apply ice, and visit  
after a week for X-ray'),  
  
    (2, 1, 'Viral Infection', 'Cough Syrup, Vitamin C', 'Patient to rest and take  
prescribed medications for 7 days'),  
  
    (3, 2, 'Heart Condition', 'Aspirin, Beta-blockers', 'Patient to undergo ECG and  
heart monitoring, follow-up after 1 week');
```

-- 6. Create appointment table

```
CREATE TABLE appointment (  
    appointment_id INT AUTO_INCREMENT,  
    doctor_id INT,  
    patient_id INT,  
    appointment_date varchar(15),  
    appointment_time varchar(10),  
    PRIMARY KEY (appointment_id),  
    FOREIGN KEY (doctor_id) REFERENCES doctor_details(doctor_id),  
    FOREIGN KEY (patient_id) REFERENCES patient_details(patient_id)  
);  
  
INSERT INTO appointment(doctor_id, patient_id, appointment_date,  
appointment_time)  
VALUES  
    (1, 1, '2024-11-07', '10:00 AM'),  
    (1, 2, '2024-11-07', '09:30 AM'),  
    (2, 1, '2024-11-07', '10:00 AM');
```

-- 7. Create medicine table

```
CREATE TABLE medicine (  
    medicine_id INT AUTO_INCREMENT,  
    medicine_name VARCHAR(255),  
    medicine_quantity INT,  
    medicine_price INT,  
    PRIMARY KEY (medicine_id)  
);
```



```
INSERT INTO medicine (medicine_name, medicine_quantity, medicine_price)
VALUES
```

```
    ('Paracetamol', 30, 50),
```

```
    ('Ibuprofen', 20, 40),
```

```
    ('Aspirin', 25, 60);
```

-- 8. Create service table

```
CREATE TABLE service (
    service_id INT AUTO_INCREMENT,
    treatment_id INT,
    service_name VARCHAR(255),
    service_cost INT,
    PRIMARY KEY (service_id),
    FOREIGN KEY (treatment_id) REFERENCES treatment_plan(treatment_id)
);
```

```
INSERT INTO service (treatment_id, service_name, service_cost)
```

```
VALUES
```

```
    (1, 'X-ray', 1500),
```

```
    (2, 'MRI', 3000),
```

```
    (3, 'CT Scan', 4000);
```

-- 9. Create test table

```
CREATE TABLE test (
    test_id INT AUTO_INCREMENT,
    treatment_id INT,
    test_name VARCHAR(255),
```

```

        test_cost INT,

        PRIMARY KEY (test_id),

        FOREIGN KEY (treatment_id) REFERENCES treatment_plan(treatment_id)

    );

INSERT INTO test (treatment_id, test_name, test_cost)

VALUES

    (1, 'Blood Test', 1000),

    (2, 'Urine Test', 800),

    (3, 'ECG', 1200);

```

-- 10. Create prescription table

```

CREATE TABLE prescription (

    prescription_id INT AUTO_INCREMENT,

    patient_id INT,

    doctor_id INT,

    medicine_id INT,

    PRIMARY KEY (prescription_id),

    FOREIGN KEY (patient_id) REFERENCES patient_details(patient_id),

    FOREIGN KEY (doctor_id) REFERENCES doctor_details(doctor_id),

    FOREIGN KEY (medicine_id) REFERENCES medicine(medicine_id)

);

INSERT INTO prescription (patient_id, doctor_id, medicine_id)

VALUES

    (1, 1, 1),

    (2, 2, 2),

```

```
(3, 1, 3);
```

-- 11. Create bill table

```
CREATE TABLE bill (  
    bill_id INT AUTO_INCREMENT,  
    treatment_id INT,  
    PRIMARY KEY (bill_id),  
    FOREIGN KEY (treatment_id) REFERENCES treatment_plan(treatment_id)  
);  
  
INSERT INTO bill (treatment_id)  
VALUES  
    (1),  
    (2),  
    (3);
```

-- 12. Create medical_record table

```
CREATE TABLE medical_record (  
    record_id INT AUTO_INCREMENT,  
    patient_id INT,  
    doctor_id INT,  
    PRIMARY KEY (record_id),  
    FOREIGN KEY (patient_id) REFERENCES patient_details(patient_id),  
    FOREIGN KEY (doctor_id) REFERENCES doctor_details(doctor_id)  
);  
  
INSERT INTO medical_record (patient_id, doctor_id)  
VALUES
```

(1, 1),

(2, 2),

(3, 1);

-- 13. Create dept_head table

```
CREATE TABLE dept_head (  
    head_id INT AUTO_INCREMENT,  
    doctor_id INT,  
    PRIMARY KEY (head_id),  
    FOREIGN KEY (doctor_id) REFERENCES doctor_details(doctor_id)  
);
```

```
INSERT INTO dept_head (doctor_id)
```

VALUES

(1),

(2);

-- 14. Create regular table

```
CREATE TABLE regular (  
    patient_id INT AUTO_INCREMENT,  
    insurance_provider varchar(100),  
    policy_number INT,  
    PRIMARY KEY (patient_id),  
    FOREIGN KEY (patient_id) REFERENCES patient_details(patient_id)  
);
```

```
INSERT INTO regular (patient_id, insurance_provider, policy_number)
```

VALUES

```
(1, 'HealthCare Inc.', '12345'),  
(2, 'LifeCare Insurance', '98765'),  
(3, 'MediSecure', '56789');
```

-- 15. Create emergency table

```
CREATE TABLE emergency (  
    patient_id INT AUTO_INCREMENT,  
    allergies varchar(100),  
    conditions varchar(100),  
    PRIMARY KEY(patient_id),  
    FOREIGN KEY (patient_id) REFERENCES patient_details(patient_id)  
);  
  
INSERT INTO emergency (patient_id, allergies, conditions)  
VALUES  
    (1, 'Peanuts', 'Severe Headache'),  
    (2, 'Dust, Pollen', 'Severe Asthma Attack'),  
    (3, 'Bee Sting', 'Anaphylactic Shock');
```

-- 16. Create admin request

```
CREATE TABLE admin_request(  
    request_id INT AUTO_INCREMENT,  
    name VARCHAR(50) NOT NULL,  
    email VARCHAR(80) NOT NULL,  
    password VARCHAR(150) NOT NULL,  
    role VARCHAR(10) NOT NULL,  
    PRIMARY KEY(request_id)
```


```
);
```

-- 17. Create prescription medicines

```
CREATE TABLE prescription_medicines (  
    prescription_id INT,  
    medicine_name varchar(50),  
    FOREIGN KEY (prescription_id) REFERENCES prescription(prescription_id)  
);
```

→ UI Screenshots :

1. Login Page : Anyone can login from here (Admin, doctor, patient)



Welcome back

Login


Email

Password

LOGIN

Don't have an account? [Sign Up](#)

2. Registration Page : Only admin can register



Welcome

Register

Name

Email

Password

REGISTER

Already have an account? [Login](#)

3. Request Page : When someone registers as admin, a request will come here. An existing admin must accept the request, then the registration will be successful.

Serial No.	Name	Email	Accept	Reject
1	admin2	admin2@gmail.com	✓	X

4. Edit Profile : From here any user can change their password.

User Details
ID : 1
Name : Md Abu Bokar
Email : abu@gmail.com

Change Password

Enter Your Old Password Enter Old Password

Enter Your New Password Enter New Password

UPDATE

5. Read and Delete : Here all the doctors lists are retrieved from the database, also can delete any doctor from here

The screenshot shows a web application interface for managing doctors. On the left is a teal sidebar with a menu containing: Dashboard, Doctors (highlighted), Patients, Appointments, Departments, Tests, Treatments, Services, Medicines, Bills, Medical Records, and Requests. The main content area is titled 'Doctors' and features a blue 'Add New' button at the top left. Below the button is a table with the following columns: Doctor ID, Name, Email, Phone No., Gender, Address, View, and Delete. The table contains two entries: Doctor ID 1, Name Nafis Anzum, Email nafis@gmail.com, Phone No. 31234567890, Gender Male, Address Bangladesh; and Doctor ID 2, Name Ava Brown, Email ava@gmail.com, Phone No. 41234567890, Gender Female, Address Australia. Each row has a 'View' icon (eye) and a 'Delete' icon (trash can). A user profile icon with the letter 'M' is in the top right corner.

Doctor ID	Name	Email	Phone No.	Gender	Address	View	Delete
1	Nafis Anzum	nafis@gmail.com	31234567890	Male	Bangladesh		
2	Ava Brown	ava@gmail.com	41234567890	Female	Australia		

6. Create : From here can do the create operation for doctors

The screenshot shows the 'Create Doctor' form in the same application. The sidebar is identical. The main content area is titled 'Create Doctor' and contains a form with the following fields: Name (text input), Email (text input), Phone No. (text input), Address (text input), Password (text input), Gender (dropdown menu with '-Select Gender-'), Speciality (dropdown menu with '-Select Speciality-'), and Department Name (dropdown menu with '-Select Department-'). A large teal 'CREATE' button is at the bottom of the form. A user profile icon with the letter 'M' is in the top right corner.

7. Update : From here can do the update operation, also after creating a profile a id card is automatically generated and admins can download it from here as pdf

Dashboard

Doctors

Patients

Appointments

Departments

Tests

Treatments

Services

Medicines

Bills

Medical Records

Requests

Edit Doctor

Name

Nafis Anzum

Email

nafis@gmail.com

Phone No.

31234567890

Address

Bangladesh

Password

**** *

Gender

Male

Speciality

Opthalmologist

Department Name

Opthalmology

UPDATE

N

Nafis Anzum

Doctor

1

Opthalmologist

This card is the property of Database Management System

Plot : 15, Block : B, Bashundhara, Dhaka-1229, Bangladesh, Please return to the above address or call : 16667

Download

8. Bill : A detailed bill based on treatment, test and services

Dashboard

Doctors

Patients

Appointments

Departments

Tests

Treatments

Services

Medicines

Bills

Medical Records

Requests

View Bill

Patient and Treatment Management

Plot : 15, Block : B, Bashundhara, Dhaka-1229, Bangladesh

Helpline : 16667

Date : 3/12/2024

Patient : Emily Parker

Doctor : Nafis Anzum

No.	Names	Cost
1	X-ray (Service)	1500
2	Blood Test, CBC (Test)	2000
		Total : 3500

Seal & Signature

DOWNLOAD

9. Record : A detailed medical record for patients

Dashboard

Doctors

Patients

Appointments

Departments

Tests

Treatments

Services

Medicines

Bills

Medical Records

Requests

View Record

M

Patient and Treatment Management

Plot : 15, Block : B, Bashundhara, Dhaka-1229, Bangladesh
Helpline : 16667

Doctor:
Patient: 1

1. PLANS

Treatment ID	Doctor ID	Diagnosis	Medications	Plan Details
1	1	Fracture	Paracetamol, Ibuprofen	Patient to take rest, apply ice, and visit after a week for X-ray
2	1	Viral Infection	Cough Syrup, Vitamin C	Patient to rest and take prescribed medications for 7 days

2. SERVICES

Service ID	Service Name	Service Cost
1	X-ray	1500

3. TESTS

10. Appointment : All the appointment lists

Appointments


Prescriptions

Appointments

N

Appointment ID	Doctor Name	Patient Name	Date	Time
1	Nafis Anzum	Emily Parker	2024-11-07	10:00 AM
2	Nafis Anzum	Sophia Johnson	2024-11-07	09:30 AM

11. Prescription : Create a prescription for a patient


Appointments
Prescriptions

Create Prescription

Your ID
Enter Your ID

Patient ID
Enter Patient ID

Medicine Name
Enter Medicine Name


Medicines :
Monas

Remove


+

CREATE

12. View Prescription : Read the prescription from here, both doctor and patient. Patients may forget to bring their prescription, so they can just login and view it or show it to their doctors.


Appointments
Prescriptions

View Prescription


Patient and Treatment Management
Plot : 15, Block : B, Bashundhara, Dhaka-1229, Bangladesh
Helpline : 16667

Date : 3/12/2024
Doctor ID : 1
Patient ID : 2
Patient Name : Sophia Johnson

Medicines:
Sergel
Civic
Napa

Breakfast	Lunch	Dinner
Breakfast	Lunch	Dinner
Breakfast	Lunch	Dinner

Seal & Signature

DOWNLOAD

→ Technologies :

◆ Frontend :

- React JS
- TailwindCSS
- Material UI

◆ Backend :

- Node JS
- Express JS

◆ Database :

- MySQL

→ Contribution :

◆ Md Abu Bokar

- Backend
- Database
- Api Handling
- Frontend
 - Login and Registration
 - Dashboard
 - Admin
 - Doctor
 - Patient
 - Department
 - Appointment
 - Prescription
 - Medical Record

◆ Natasha Monir Shawon

- Database
- Frontend
 - Medicine
 - Service
 - Test
 - Treatment
 - Bill

→ Conclusion :

- ◆ This project aims to provide an organized database management system for patients. In this project modern technologies and frameworks were used to cope up with the modern world, also focused on the UI/UX design to make it more attractive. As for the security, in this project JWT(Json Web Token) was used to make it more secure. JWT allowed token based authentication in this project, and made it more reliable. Also the crud operations with axios makes it a more advanced project.

→ References :

- ◆ *Lama Dev. (n.d.). Home [YouTube Channel]. YouTube.*
<https://www.youtube.com/@LamaDev>
- ◆ *Web Dev Simplified. (n.d.). Home [YouTube Channel]. YouTube.*
<https://www.youtube.com/c/WebDevSimplified>
- ◆ *Bro Code. (n.d.). Home [YouTube Channel]. YouTube.*
<https://www.youtube.com/c/BroCodez>
- ◆ *JavaScript Mastery. (n.d.). Home [YouTube Channel]. YouTube.*
<https://www.youtube.com/c/JSMAstery>
- ◆ *W3Schools. (n.d.). W3Schools.*
<https://www.w3schools.com>