COLLEGE CODE	:9623
COLLEGE NAME	:Amrita College of Engineering and Technology
DEPARTMENT	:Computer Science Engineering
STUDENT NM -ID	: AB0F2B3EBEDF8BDB2EC44A024D951EA7
ROLL NO	:96232310492
DATE	:25-10-2025

Completed the project named as Phase 5

TECHNOLOGY PROJECT NAME :AngularJS with SQL Integration

SUBMITTED BY,

NAME: K.J.SHAHAMBHARI

MOBILE NO: 8072123291

Phase 5 – Project Demonstration & Documentation

1. Demo Walkthrough

1.1 Landing Page / UI Overview

- Show the main dashboard with options for:
  - o Patient Registration and Login.
  - o Book Appointment form.
  - o Doctor List with available timings.
  - o My Appointments section.
- Highlight key UI sections:
  - o Login / Register Forms: To manage patient access.
  - o Doctor Cards: Display each doctor's name, specialty, and available timings.
  - o Appointment Form: Allows selecting doctor, date, and time.
  - o Appointment Table: Shows booked appointments with options to cancel.

1.2 Live Data Fetching

- Demonstrate booking an appointment:
  1. Login as a patient.
  2. Choose a doctor, select date/time, and submit.
- Explain how AngularJS sends AJAX requests to the backend API.
- Show that the data is fetched from the SQL database and dynamically displayed without refreshing the page, thanks to AngularJS two-way data binding.

1.3 Appointment & Doctor Details Display

- Walk through displayed appointment and doctor details:
  - o Doctor name, specialty, and available hours.
  - o Appointment date, time, and status.
  - o Notes or reason for appointment.
- When “View Schedule” is clicked, show doctor’s booked schedule with patient names and timings.

1.4 Appointment List & Management

- My Appointments :
  - o Fetched dynamically from SQL database.
  - o Displayed in a clean, paginated table.
  - o Each row shows: Doctor, Date, Time, Status.
- Demonstrate:
  - o Booking a new appointment
  - o Cancelling an existing appointment
  - o Automatic UI refresh after changes (no reload).

1.5 Technical Explanation (short)

- Backend / API: RESTful API built with Node.js + Express, connected to MySQL database.
- Frontend: Built using AngularJS for dynamic UI rendering and user interactions.
- Data Handling: Backend fetches or modifies appointment/doctor/patient records in SQL and returns data in JSON format. AngularJS updates the UI instantly.

1.6 Demo Scenarios

- Register a new patient and log in.

- View doctor list and availability.
- Book an appointment with a chosen doctor.
- View the appointment in My Appointments.
- Cancel an appointment.
- View doctor’s full schedule.
- Show error message if booking is incomplete or slot unavailable.

1.7 Closing

- Highlight usefulness:
  - o Centralized, digital appointment management for hospitals.
  - o Simplifies patient booking and doctor scheduling.
  - o Works on any device via web browser.
- Mention future improvements:
  - o Add doctor/admin login for schedule management.
  - o Integrate email/SMS reminders.
  - o Add analytics or health report tracking.

2. Project Report

2.1 Objectives

- Enable easy and quick appointment booking between patients and doctors.
- Provide real-time access to doctor availability and patient appointments.
- Store and manage data securely in a SQL database.
- Deliver a responsive, easy-to-use interface for both patients and staff.

2.2 System Design

Architecture:

- Frontend: AngularJS handles user interaction and data binding.

- Backend: Node.js + Express REST API to handle requests.
- Database: MySQL stores patient, doctor, and appointment data.

Modules:

1. Patient Authentication – Register and login.
2. Doctor Management – List and availability.
3. Appointment Booking – Create, view, and cancel appointments.
4. Doctor Schedule Viewer – View all booked appointments.
5. Error Handling – For invalid input or unavailable slots.

2.3 Features

- Patient Registration & Login system.
- Book, View, and Cancel Appointments.
- Doctor availability display and schedule viewing.
- Responsive UI using Bootstrap.
- Dynamic data loading via AngularJS without page reloads.
- Real-time feedback for booking success or errors.

2.4 Implementation

Tech Stack:

- Frontend: AngularJS, Bootstrap 4.
- Backend: Node.js + Express REST API.
- Database: MySQL (SQL integration).

Data Handling:

- Backend API executes SQL queries for patients, doctors, and appointments.
- JSON data returned to AngularJS, which updates the view instantly using data binding.

2.5 Results

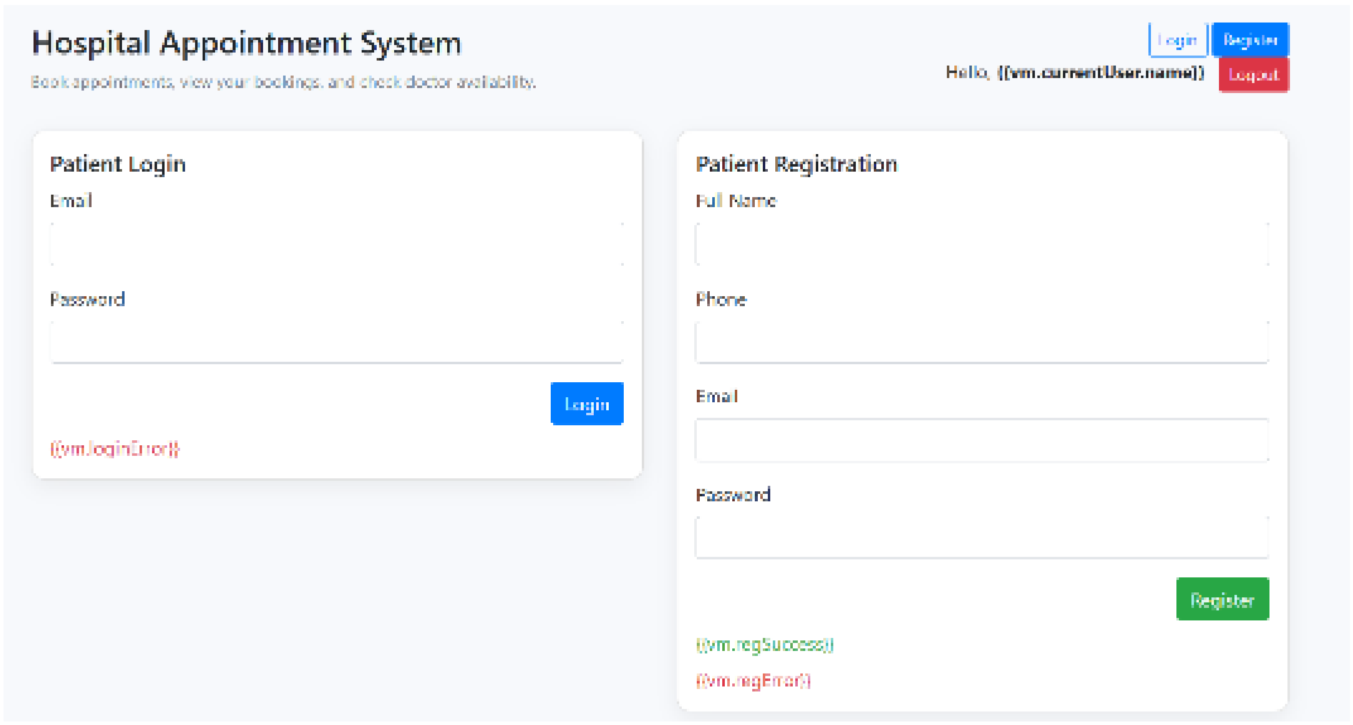
- Fast, user-friendly booking experience.
- Smooth dynamic updates without full-page reloads.
- Reliable data storage and retrieval using SQL.
- Responsive layout that works on both desktop and mobile.

2.6 Future Enhancements

- Add doctor/admin roles for managing appointments.
- Add notifications (SMS/Email) for reminders.
- Implement report generation for daily bookings.
- Integrate with hospital EMR (Electronic Medical Records).
- Include payment integration for online consultation.

x

3. Screenshot / API Documentation



Hospital Appointment System

Book appointments, view your bookings, and check doctor availability.

Hello, {{vm.currentUser.name}}

Patient Login

Email

Password

{{vm.loginError}}

Patient Registration

Full Name

Phone

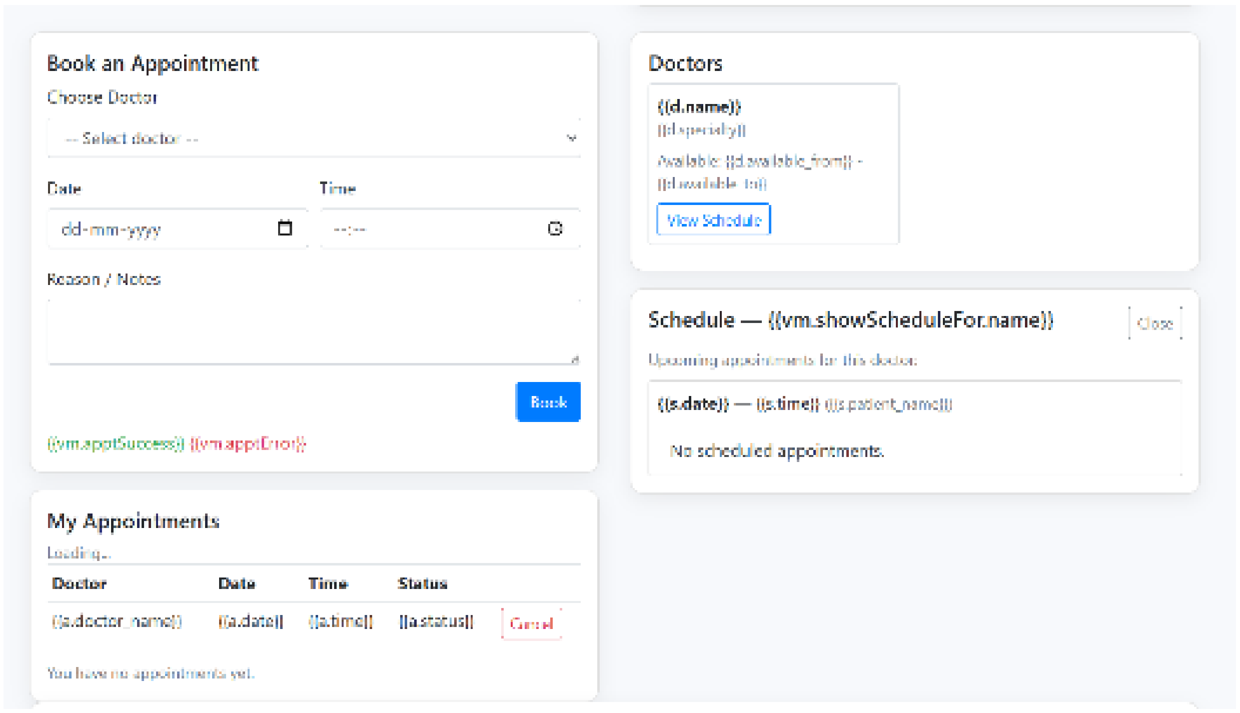
Email

Password

{{vm.regSuccess}}

{{vm.regError}}

a)Home page before login


Book an Appointment

Choose Doctor

-- Select doctor --

Date dd-mm-yyyy

Time --:--

Reason / Notes

Book

{{vm.apptSuccess}} {{vm.apptError}}

My Appointments

Loading...

Doctor	Date	Time	Status	
{{a.doctor_name}}	{{a.date}}	{{a.time}}	{{a.status}}	Cancel

You have no appointments yet.

Doctors

{{d.name}}

{{d.specialty}}

Available: {{d.available_from}} - {{d.available_to}}

View Schedule

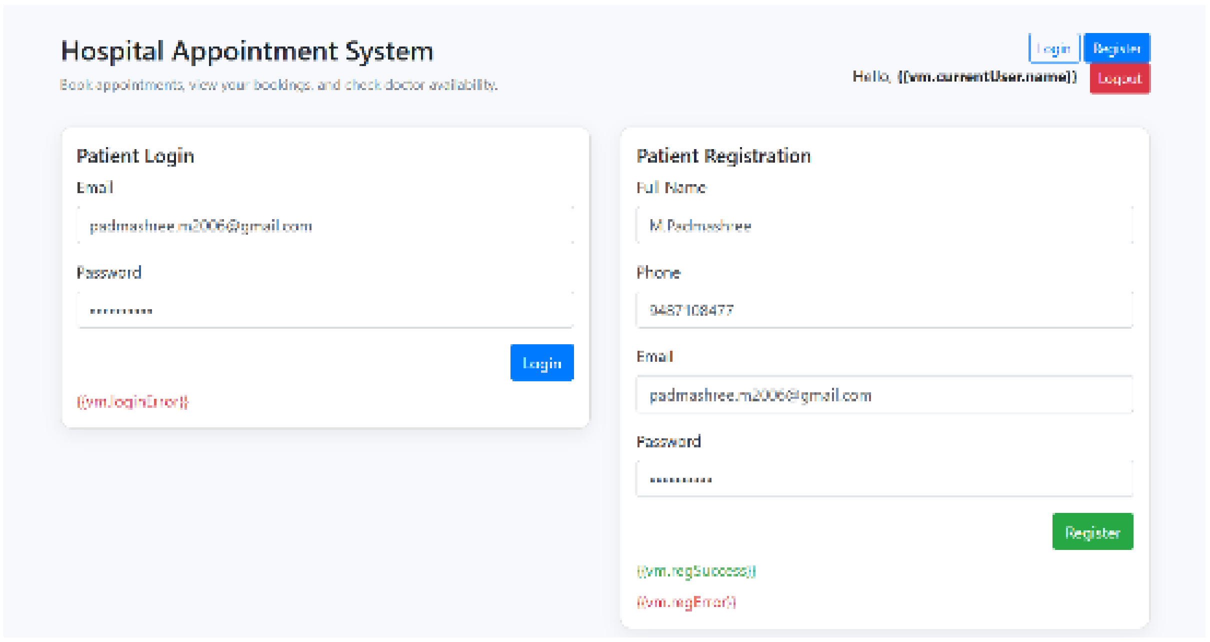
Schedule — {{vm.showScheduleFor.name}} Close

Upcoming appointments for this doctor:

{{s.date}} — {{s.time}} ({{s.patient_name}})

No scheduled appointments.

b)Dashboard page before login


Hospital Appointment System

Book appointments, view your bookings, and check doctor availability.

Login Register

Hello, {{vm.currentUser.name}} Logout

Patient Login

Email: padmashree.m2006@gmail.com

Password: ••••••••••

Login

{{vm.loginError}}

Patient Registration

Full Name: M Padmashree

Phone: 9487108477

Email: padmashree.m2006@gmail.com

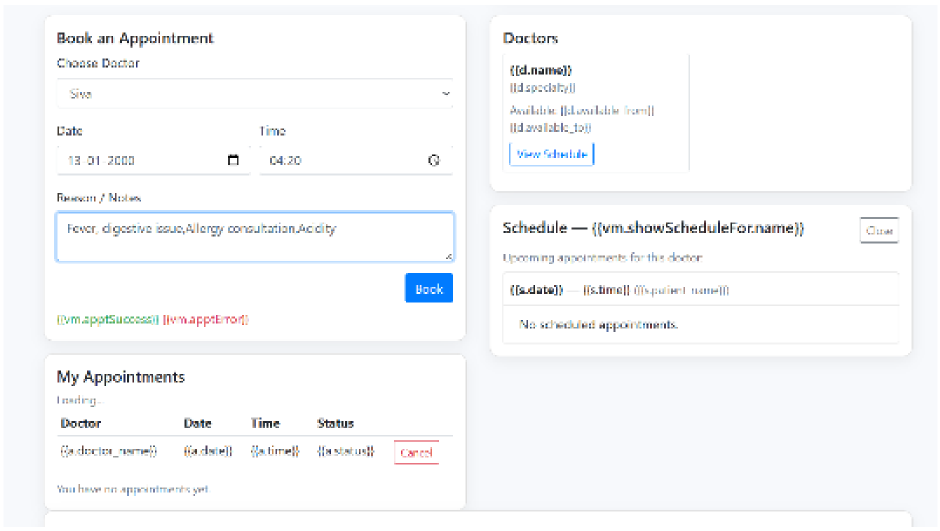
Password: ••••••••••

Register

{{vm.regSuccess}}

{{vm.regError}}

c)Home page after login


Book an Appointment

Choose Doctor

Siva

Date 13-01-2000

Time 04:20

Reason / Notes

Fever, digestive issue,Allergy consultation,Acidity

Book

{{vm.apptSuccess}} {{vm.apptError}}

Doctors

{{d.name}}
{{d.specialty}}
Available: {{d.available_from}}
{{d.available_to}}

View Schedule

Schedule — {{vm.showScheduleFor.name}}

Close

Upcoming appointments for this doctor

{{s.date}} — {{s.time}} {{s.patient_name}}

No scheduled appointments.

My Appointments

Loading...

Doctor	Date	Time	Status	
{{a.doctor_name}}	{{a.date}}	{{a.time}}	{{a.status}}	Cancel

You have no appointments yet.

d) Dashboard page after login

API Endpoints

Method	Endpoint	Description
GET	/api/doctors	Fetch all doctor records
POST	/api/patients	Register a new patient
POST	/api/login	Patient login
POST	/api/appointments	Book a new appointment
GET	/api/patients/:id/appointments	Get appointments for a patient
GET	/api/doctors/:id/appointments	Get all appointments for a doctor
DELETE	/api/appointments/:id	Cancel an appointment

Sample JSON Response (Appointments List)

```
[
 {
  "id": 201,
  "doctor_name": "Dr. Meera Nair",
  "date": "2025-11-05",
  "time": "11:00",
  "status": "Scheduled"
 },
 {
  "id": 202,
  "doctor_name": "Dr. Arun Kumar",
  "date": "2025-11-10",
  "time": "09:30",
  "status": "Cancelled"
 }
]
```

Sample JSON Response (Doctor List)

```
[
 {
  "id": 1,
  "name": "Dr. Priya Sharma",
  "specialty": "Cardiologist",
  "available_from": "09:00",
  "available_to": "15:00"
 },
 {
  "id": 2,
  "name": "Dr. Arun Kumar",
  "specialty": "General Physician",
  "available_from": "10:00",
  "available_to": "17:00"
 }
]
```

4. Challenges & Solutions

Challenge	Solution
Connecting AngularJS to SQL	Implemented REST API via Node.js + Express

backend	with MySQL driver
Handling unavailable backend	Added fallback demo data and error messages
Ensuring responsive design	Used Bootstrap grid and card components
Avoiding page reloads	Used AngularJS AJAX and two-way data binding for live updates

6. GitHub Link

https://github.com/shahambhari/Angular-JS-with-SQL-integration