Università
Ca'Foscari
Venezia

Software Architecture

Project: Hospital Management System

Group Member Details:

Muhammad Umair – 908758

Hospital Management System - Project Documentation

1. Introduction

This Hospital Management System is a comprehensive web-based application designed to manage the administrative and clinical functions of a hospital. The system is divided into three main user roles: Admin, Patient, and Staff. Each role has its own dedicated modules to streamline tasks such as appointments, user management, patient records, and more.

2. Admin Panel Modules

The Admin panel allows the management of the entire hospital system. Key modules include:

- **User Management:** Add, edit, and remove users including patients and staff.
- **Dashboard:** View system overview, statistics, and activity log.
- **Appointment Management:** Oversee all appointments, assign staff, and manage schedules.
- **Emergency Intake:** Handle emergency cases with rapid intake and assignment features.
- **Examination Management:** Manage patient examinations.
- **Ward Management:** See assigned patients and managed ward assignment.

3. Patient Panel Modules

The Patient panel is designed for individuals receiving treatment. Modules include:

- **Profile Management:** View and update personal and medical details.
- **Appointment Management:** View appointments list with updated status.
- **Medical Records:** Access personal treatment history and prescriptions.

4. Staff Panel Modules

The Staff panel is for doctors and nurses. Modules include:

- **Patient Management:** View and manage patient treatment plans.
- **Appointment Handling:** Accept or reschedule appointments.
- **Ward Management:** See assigned patients and managed ward assignment.

5. Technologies Used

- **Frontend:** HTML, CSS, JavaScript, Bootstrap
- **Backend:** PHP (CodeIgniter Framework)
- **Database:** MySQL
- **Containerization:** Docker

6. Microservices Architecture

6.1 Appointment Service (appointment-service)

Port: 8084
Path: /services/appointment-service
Responsibilities:
- Manage appointment creation, updates, deletions.
- Serve appointment data to all modules (admin, patient, staff).
- Centralize all business logic for appointment scheduling and status handling.

– **APIs:**
– GET /appointments/fetch_all – Fetch all appointments
– POST /appointments/save – Add or edit an appointment
– GET /appointments/delete/{id} – Cancel appointment
– GET /appointments/patient/{id} – Appointments for a specific patient
– GET /appointments/staff/{id} – Appointments for a specific doctor/staff

6.2 Ward Management Service (ward-service)

Port: 8085
Path: /services/ward-service
Responsibilities:
- Manage hospital ward creation, updates, and deletions.
- Handle bed availability, assignments, and discharges.
- Provide patient–ward mapping data to patient, staff, and admin modules.

– **APIs:**
– GET /ward/fetch_all – List all wards
– POST /ward/save – Create or update a ward
– GET /ward/delete/{id} – Delete a ward
– POST /ward/assign_patient – Assign a patient to a bed
– GET /ward/get_ward_assignments – View all active assignments
– GET /ward/get_available_beds/{ward_id} – Fetch available beds
– POST /ward/discharge_patient – Discharge a patient
– GET /ward/get_patients_by_ward/{ward_id} – Patients in a specific ward
– GET /ward/get_patient_assignment/{patient_id} – Assignment for one patient
– GET /ward/get_ward_status – View bed occupancy summary

6.3 Docker Integration

Both microservices are registered in docker-compose.yml as services. Each runs in its own container:

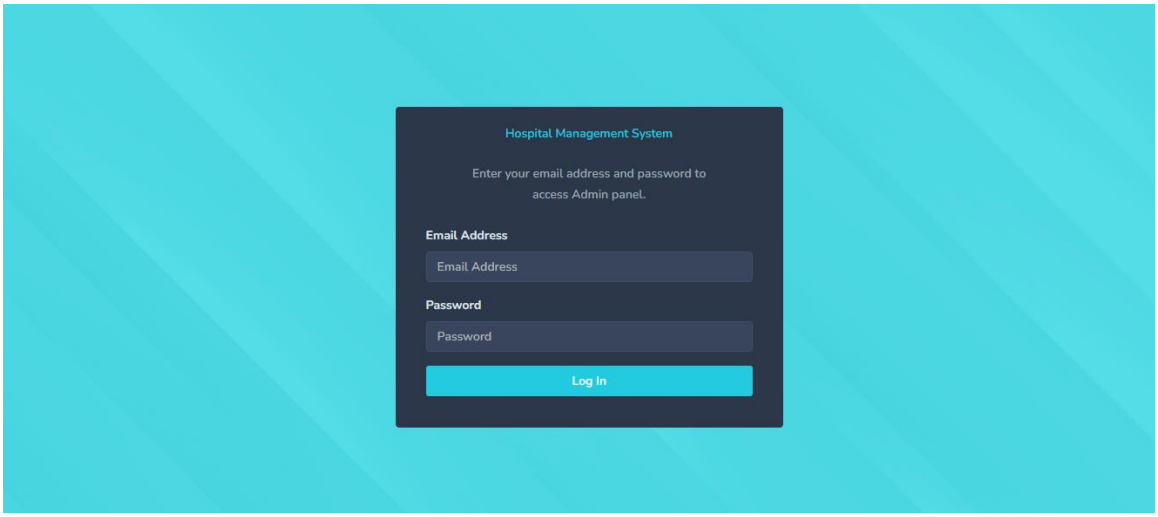
– **appointment-service:**
  - build: ./services/appointment-service
  - ports: 8084
– **ward-service:**
  - build: ./services/ward-service
  - ports: 8085

They share the MySQL container and are accessible independently.

7. Accessing the Project

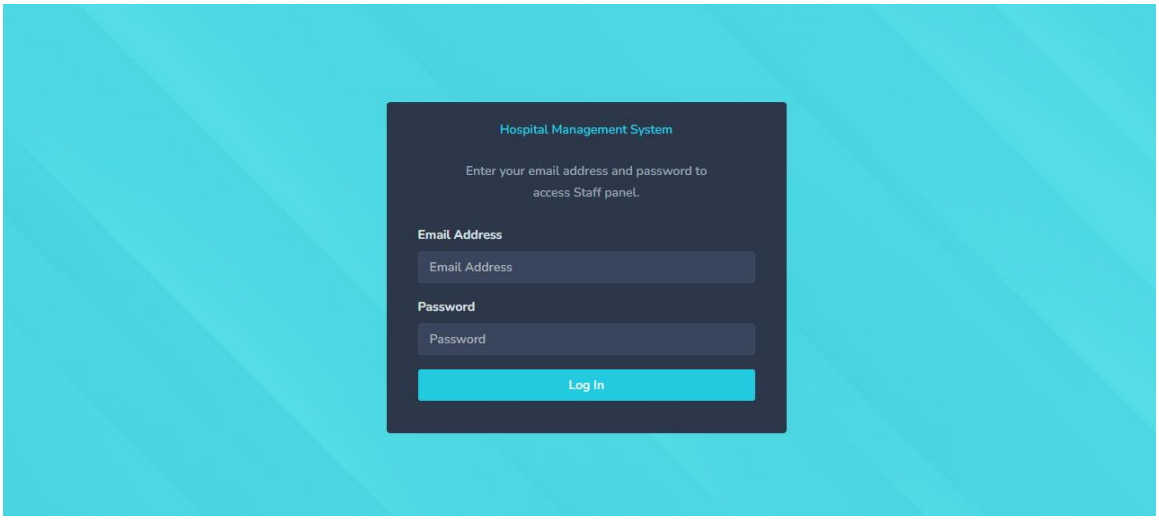
Follow the steps below to clone and run the project locally using Docker:

- Download Docker: Download
- Clone the repository: https://github.com/Umairi54321/908758-umair-muhammad.git
- Navigate to the project directory: **cd project-directory**
- Start Docker containers: **docker-compose up –d**
- Start a MySQL database container
- Start a phpMyAdmin interface
- Access via phpMyAdmin:
- **Open** [http://localhost:9090]
- **Use:**
- - **Username**: `root`
- - **Password**: `root`
- Import your SQL schema manually fromdb folder of your reporsitory within phpMyAdmin.
- **Log in to**:
- **appointment:**
- **Server:** mysql-appointment
- **Username:** appointment_user
- **Password**: appointment_pass
- **ward:**
- **Server:** mysql-ward
- **Username:** ward_user
- **Password:** ward_pass
- Choose the respective database (e.g., appointment_db)
- Go to Import tab and upload the corresponding .sql file
- **Admin Panel** -> http://localhost:8081/. By accessing this you will see interface like this below:

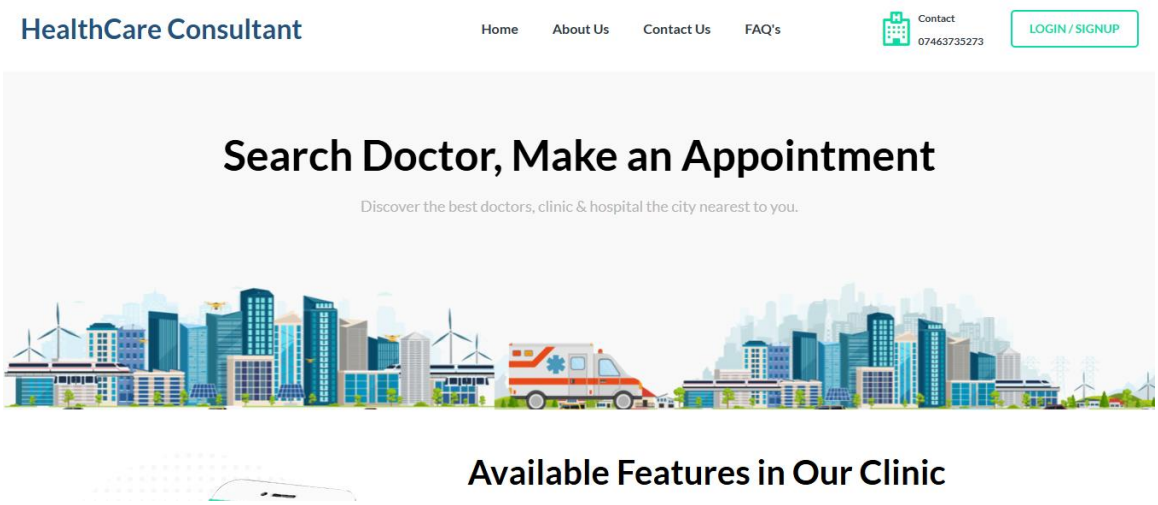


For Login, use below credentials:
Email: admin@gmail.com
Password: admin123

– **Staff Panel** -> http://localhost:8083/. By Accessing this you will see:



– **Patient Panel** -> http://localhost:8082/. By accessing this you will see:



8. Git Commit History

Below are notable commits as observed from GitHub history:

- Apr 30, 2025: Admin panel complete

- Apr 29, 2025: Folder structure, admin appointment, emergency intake, patient modules

- Apr 25, 2025: Admin part started, login, user management, dashboard

- Apr 23, 2025: Folder structure and patient portal

- Apr 19, 2025: Home page, About Us, Contact, FAQ frontend

- May 17, 2025: Fixes & Completed

- May 16, 2025: Completed

- May 15, 2025: Completed

- May 9, 2025: Staff panel setup and patient management