CS355 – Database Project Proposal Form

Section 1: Metadata

to be filled by the student

1.1. Project Information to be filled by the student

Title: Hospital Management System	
Section: L2	Instructor: Mohsin Nagaria

1.2. Student(s) Information

Name: Muhammad Quddusi Kashaf	ID: mk08457
Section: L2	Batch: 2026

Name: Basil Khowaja	ID: bk08432
Section: L2	Batch: 2026

Name: Muhammad Aqeel Mehdi	ID: mm08442
Section: L2	Batch: 2026

Submission guideline: Save your project proposal as a pdf file and rename as Project Proposal_L1_ProposedTitle where L1 is to be replaced with your section

Section 2: The Project

to be filled by the student

2.1. Project Description: *Please provide an introduction of the project including its scope.*

The **Hospital Management System (HMS)** is designed to streamline the management and operations of hospitals by efficiently organizing, storing, and retrieving critical hospital data. This system provides a comprehensive solution to manage patient information, doctor appointments, medical records, treatments, billing, and hospital resources like staff, rooms, and equipment.

The system caters to multiple departments within the hospital, including specialized units such as Cardiology, Neurology, and Pediatrics. It enables seamless coordination between departments, doctors, nurses, and administrative staff to provide better patient care and management.

Key Features:

1. **Patient Management**:
    - Register patients and maintain detailed medical histories, including past visits, treatments, and medications.
    - Manage outpatient appointments and inpatient admissions, including room assignment and discharge procedures.
2. **Doctor Management**:
    - Track doctor schedules, including their assigned departments and specialties.
    - Manage appointments and link doctors to specific patients for consultation or treatment.
3. **Appointment Scheduling**:
    - Facilitate booking of appointments between patients and doctors, with reminders for upcoming visits.
    - Manage scheduling for routine checkups, surgeries, and emergency consultations.
4. **Medical Records**:
    - Maintain detailed patient records, including diagnosis, tests, treatments, and medications.
    - Update medical records with doctor notes, prescriptions, and test results for easy retrieval.
5. **Billing and Payment System**:
    - Generate bills based on treatments, tests, medications, and hospital stays.
    - Manage payments, track outstanding balances, and provide patients with detailed billing reports.
6. **Staff and Resource Management**:
    - Maintain records of doctors, nurses, and administrative staff, including their roles and schedules.
    - Manage room availability, assigning rooms to admitted patients and tracking occupancy.

7. **Test and Medication Tracking**:
    - Record tests ordered for patients and track results.
    - Manage medications prescribed by doctors, including dosage and frequency

The **Hospital Management System** is designed to enhance the quality of patient care, streamline hospital operations, and optimize resource utilization. It serves as a central platform to manage all hospital-related activities, ensuring that information is readily available, accurate, and up-to-date.

2.2 Functional Requirements

This section describes each function/feature provided by your system. These functions are logically grouped into modules based on their purposes. The users in your system must be categorized such as client, customer or administrator etc. These users will be accessing the database with the level of access that they are authorized with.

Sample functional Requirements:

Module 1: Patient Registration & Management

- **Function 1: Register a new patient**
  - The system allows the **Receptionist** to register a new patient. The form prompts for the patient's details (e.g., Name, Age, Gender, Address, Contact Information).
  - The system assigns a unique patient ID and stores the patient's medical history, allergies, and emergency contacts.
- **Function 2: Edit patient information**
  - The **Receptionist** or **Admin** can update patient details. The system prompts for the changes, and the updated information is saved to the patient's profile.
- **Function 3: Search for a patient**
  - **Doctors**, **Nurses**, or **Receptionists** can search for patients by their name, ID, or contact information.
  - The system retrieves and displays the patient's information, including their medical history and ongoing treatments.

Module 2: Appointment Scheduling

- **Function 1: Schedule an appointment**
  - The system allows the **Patient** or **Receptionist** to book an appointment with a **Doctor**. It prompts for the preferred date, time, and department (e.g., Cardiology, Pediatrics).
  - The system checks the doctor's availability and confirms the appointment, sending a notification to the patient.
- **Function 2: Reschedule or cancel an appointment**
  - The **Patient** or **Receptionist** can reschedule or cancel an appointment. The system updates the schedule and notifies the doctor.
- **Function 3: View appointment history**
  - The **Patient** and **Doctor** can view previous appointment details, including dates, times, and the purpose of the visit.

Module 3: Medical Records Management

- **Function 1: Update medical records**
  - **Doctors** can add diagnoses, treatments, prescriptions, and test results to a patient's medical record after each consultation.
  - **Nurses** can update patient vitals and daily health observations for admitted patients.
- **Function 2: View medical history**

- **Doctors** and **Patients** can view the complete medical history, including past diagnoses, treatments, and surgeries.

- **Function 3: Generate medical reports**
  - The system allows **Doctors** or **Admin** to generate detailed medical reports for a patient, including test results, diagnoses, and prescriptions.

Module 4: Billing and Payments

- **Function 1: Generate a bill**
  - The **Receptionist** or **Admin** can generate bills based on treatments, medications, room charges, and diagnostic tests.
  - The system itemizes each charge and calculates the total amount due.
- **Function 2: Process payments**
  - The **Receptionist** processes payments made by patients. The system records the payment details (e.g., amount, date, payment method).
  - **Patients** can also pay through an online portal.
- **Function 3: View billing history**
  - **Patients** can view their past bills and payment history through their accounts.

Module 5: Room and Bed Management

- **Function 1: Assign rooms to patients**
  - The **Receptionist** or **Admin** assigns rooms to admitted patients based on room availability and type (e.g., single, shared).
  - The system tracks room occupancy and availability in real time.
- **Function 2: Update room status**
  - **Nurses** or **Admin** can update room status (e.g., occupied, available, under maintenance) based on patient admission or discharge.

Module 6: Staff Management

- **Function 1: Add or remove staff members**
  - The **Admin** can add or remove staff members such as **Doctors**, **Nurses**, and **Receptionists** from the system.
  - The system prompts for staff details (e.g., Name, Role, Department, Contact Information) and assigns a unique staff ID.
- **Function 2: Assign staff to shifts**
  - **Admin** can assign staff to shifts (morning, evening, night). The system tracks shift schedules and sends notifications to staff members.

Module 7: Prescription and Medication Management

- **Function 1: Prescribe medication**
  - **Doctors** can prescribe medications to patients. The system prompts for medication details (e.g., drug name, dosage, frequency).

- The prescription is added to the patient's medical record and sent to the pharmacy.

- **Function 2: View medication history**
  - **Doctors**, **Patients**, and **Pharmacists** can view a patient's medication history, including past and current prescriptions**.**

2.3. Planned Schedule: *Kindly list the start/end dates and the timeline for the achievement of any intermediate milestones and the expected contribution to be made by the participant(s).*

Week 1: Project Setup & Database Design

- **Day 1-2:** Set up project repository, define project structure, and tools (e.g., frameworks, database, etc.)
- **Day 3-5:** Design the ERD (Entity-Relationship Diagram) based on the entities and relationships.
- **Day 6-7:** Implement the SQL database schema for tables (patients, doctors, appointments, medical records, billing, etc.).

Week 2: Implement SQL & Basic CRUD Operations

- **Day 1-2:** Define SQL queries for creating, updating, reading, and deleting (CRUD) records in the database for patients, doctors, appointments.
- **Day 3-4:** Test SQL queries for individual modules (e.g., patient management, appointment scheduling).
- **Day 5-7:** Set up and configure the database connection in the backend, integrate SQL with the application.

Week 3: Front-end Development

- **Day 1-2:** Design and implement the login screen and user-specific dashboards (admin, doctor, patient).
- **Day 3-4:** Implement patient registration, patient details, and appointment form screens (using HTML, CSS, JavaScript, or framework like React).
- **Day 5-7:** Create responsive design and ensure smooth navigation between different screens.

Week 4: Back-end Development

- **Day 1-2:** Implement user authentication (login) and role-based access control (RBAC) to differentiate between admin, doctor, nurse, and patient.
- **Day 3-4:** Set up APIs or server-side logic for CRUD operations (patient management, appointments, billing).
- **Day 5-7:** Implement backend logic for appointment scheduling, rescheduling, and cancellations.

Week 5: Integration & Feature Completion

- **Day 1-2:** Integrate the front-end with the back-end for the appointment form, patient details screen, and admin dashboard.
- **Day 3-4:** Complete the integration of billing and payment features.
- **Day 5-7:** Implement notifications (email or in-app) for appointment confirmations, updates, and reminders.

Week 6: Testing, Debugging, & Final Touches

- **Day 1-2:** Conduct unit testing for each module (patient registration, appointment scheduling, billing, etc.).
- **Day 3-5:** Perform system and integration testing to ensure all components work seamlessly.

- **Day 6-7:** Final debugging, polish UI, and ensure smooth user experience. Prepare project documentation and presentation.

2.4. Technology Stack: *If you are utilizing any language or database besides PyQt and SQL Server, please complete this section; otherwise, leave it blank. Specify the programming language and database management system intended for constructing this application, as well as the application type (Desktop, Web, or Mobile).*

Softwares:

QTdesigner

DBdesigner

Github

Visual Studio Code

Programming Languages:

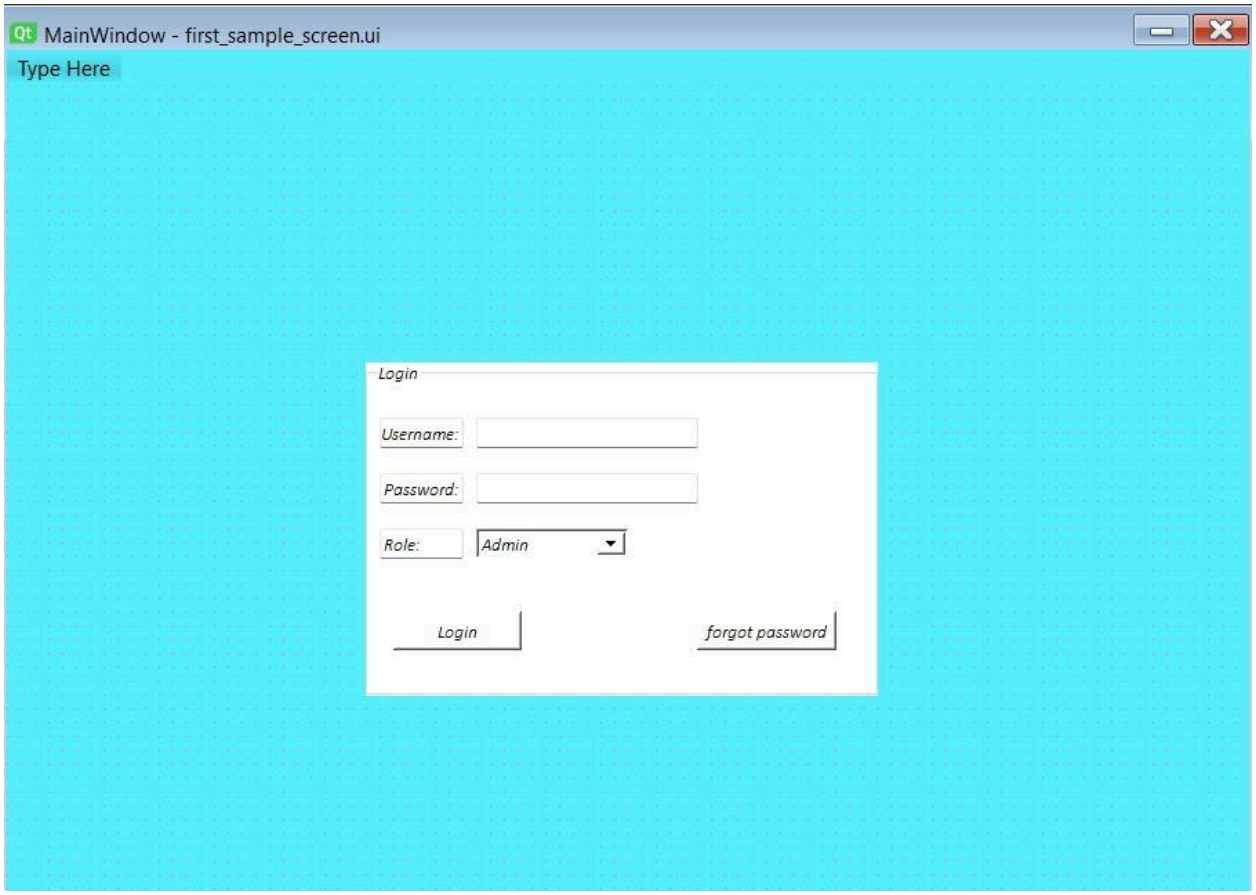
Python

SQL

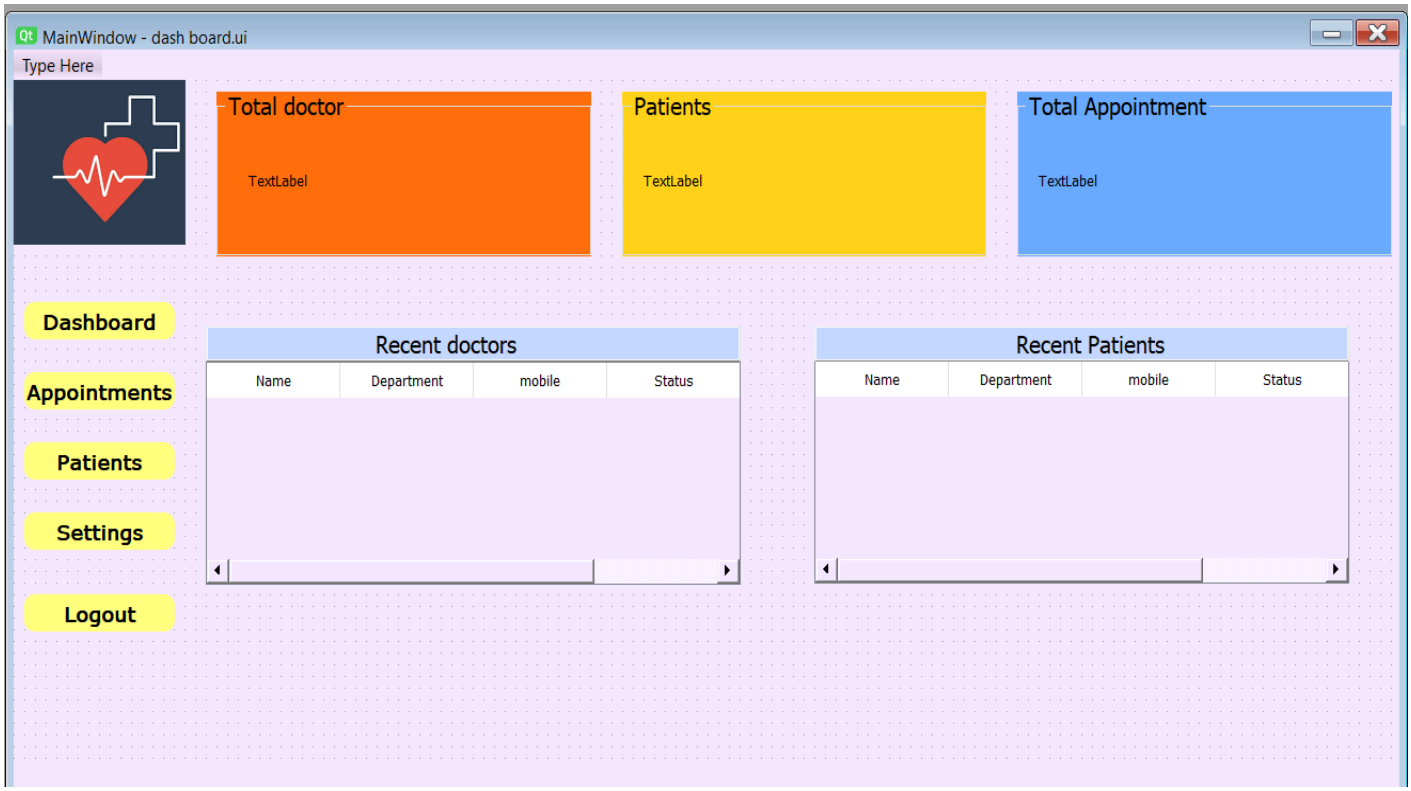
2.5. Screens: *Provide images of all application screens, showcasing clear input and corresponding outputs. Ensure each image includes a concise caption explaining user action and expected/observed output. You can create these screens using Qt Designer.*

Screen 1

The login page is the entry point for all users, including admins, doctors, nurses, and patients. Each user is required to input their credentials, and upon successful login, they are redirected to their respective dashboards. The system recognizes user roles and grants access to customized features, allowing for role-specific data visibility and functionality.

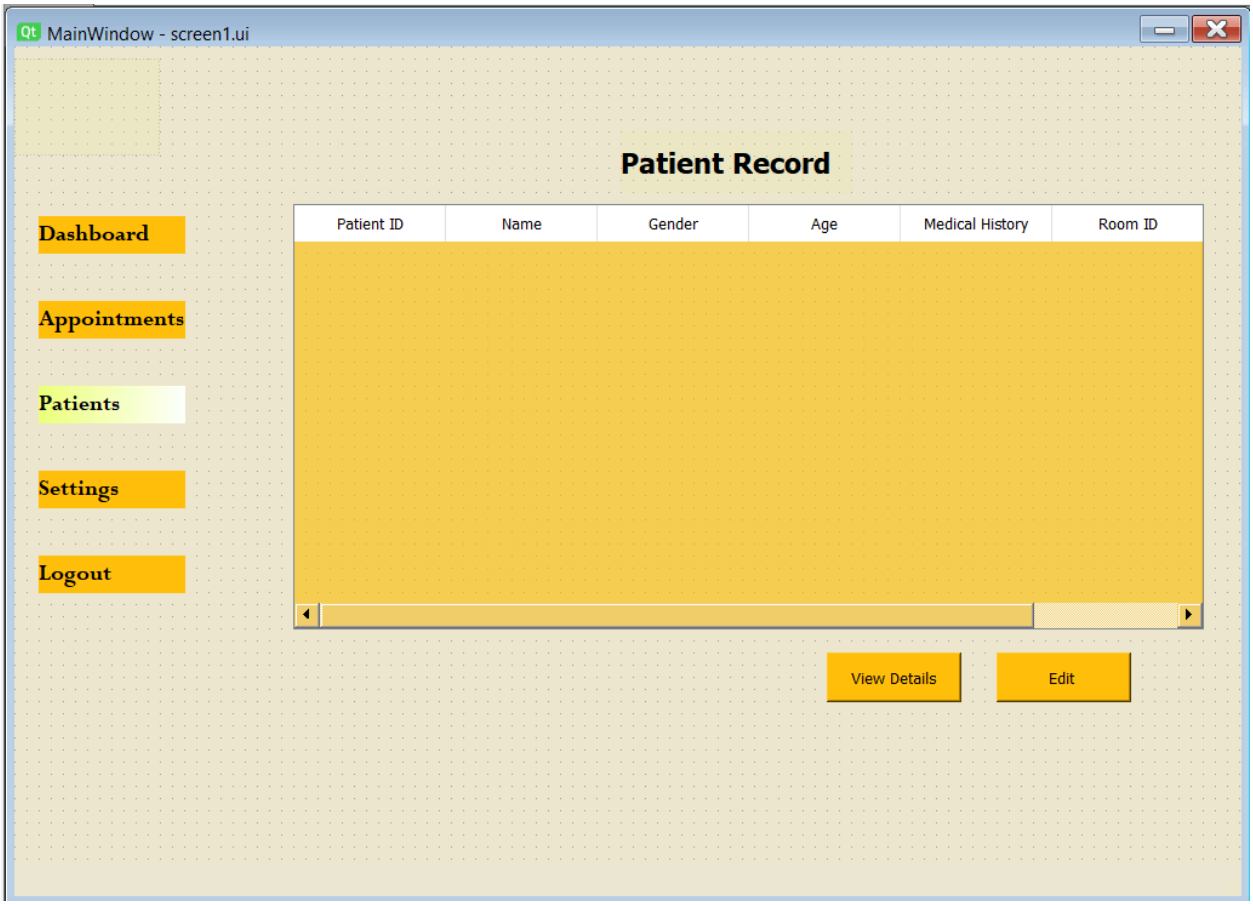


Screen 2:
This is the initial dashboard we made when an admin logs into the database. The admin dashboard provides a centralized control panel for managing hospital operations. Admins can view and manipulate data for appointments, patients, doctors, billing, and staff. This dashboard allows the admin to update, add, or remove records and track overall hospital activities, ensuring efficient management and oversight of all departments.



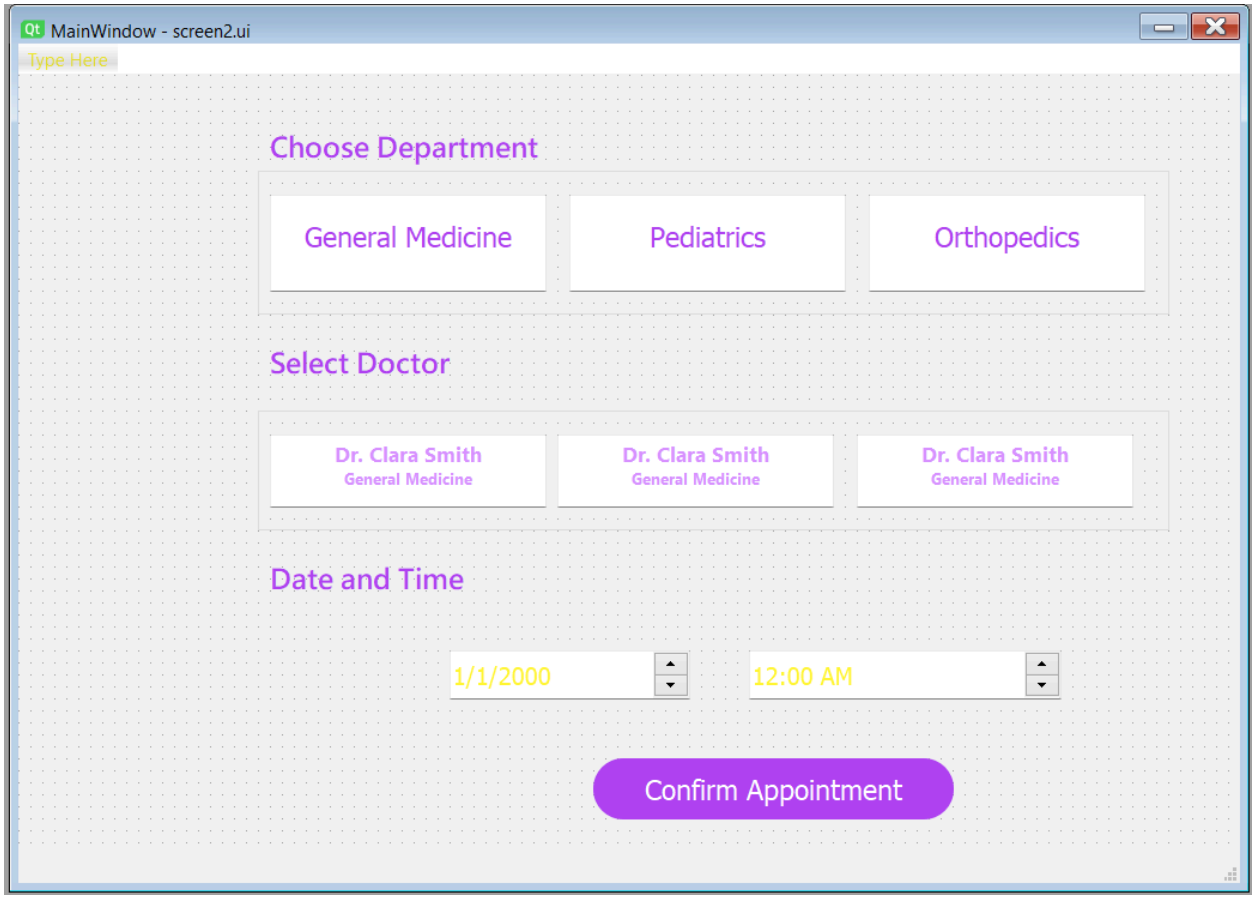
Screen 3

This screen displays comprehensive information about a patient, such as personal details, medical history, and ongoing treatments. It allows authorized users, including admin, doctors, and nurses, to view and edit patient data in real time.



Screen 4

This screen allows patients to book appointments by selecting their preferred doctor, department, and time slot. It shows available dates and times, checks doctor availability, and allows the patient to confirm the appointment.



2.6.initial ERD Diagram: