



UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Malaysia-Japan
International
Institute of Technology
(MJIT)

SECD2523-DATABASE

20242025 - SEMESTER 1

PHASE 5

FACULTY OF MJIT

NAME	MATRIC ID
Liu Ruoyang	A23MJ4022
Pranto Anik Islam	A23MJ4024
Kahlan Sultan	A23MJ4021
Bu Guoshun	A23MJ4019

Relational Database Schemas

1. Patient Table

This table stores the basic information of patients.

- **Attributes:**
 - PatientID (Primary Key): Unique identifier for each patient.
 - FullName: Name of the patient.
 - DateOfBirth: Date of birth of the patient.
 - EmailAddress: Email address of the patient.
- **Relationships:**
 - Referenced by PatientContact, Visitor, Bill, LabTestInstances, Appointment, and Treatment tables.

2. PatientContact Table

This table manages contact information for patients.

- **Attributes:**
 - PatientID (Primary Key, Foreign Key): Links to Patient(PatientID).
 - ContactNumber: Contact number of the patient.
- **Relationships:**
 - One-to-One with the Patient table.

3. Visitor Table

This table records visitor information associated with patients.

- **Attributes:**

- VisitorID: Unique identifier for the visitor.
- PatientID (Foreign Key): Links to Patient(PatientID).
- Relationship: Relationship of the visitor to the patient.
- **Relationships:**
 - Many-to-Many relationship between visitors and patients.

4. Bill Table

This table handles billing information for patients.

- **Attributes:**
 - BillID (Primary Key): Unique identifier for each bill.
 - Amount: The bill amount.
 - PaymentStatus: Status of the payment (e.g., Paid, Pending).
 - PatientID (Foreign Key): Links to Patient(PatientID).
- **Relationships:**
 - One-to-Many with the Patient table.

5. LabTest Table

This table contains information about the types of lab tests available.

- **Attributes:**
 - TestID (Primary Key): Unique identifier for the test.
 - TestName: Name of the test.
- **Relationships:**
 - Referenced by LabTestInfo and LabTestInstances.

6. LabTestInfo Table

This table provides additional information about lab tests.

- **Attributes:**
 - LabID (Primary Key): Unique identifier for the lab test info.
 - TestID (Foreign Key): Links to LabTest(TestID).
- **Relationships:**
 - One-to-Many relationship with the LabTest table.

7. LabTestInstances Table

This table records individual lab tests conducted for patients.

- **Attributes:**
 - TestID (Primary Key, Foreign Key): Links to LabTest(TestID).
 - PatientID (Primary Key, Foreign Key): Links to Patient(PatientID).
 - DateConducted: The date the lab test was conducted.
- **Relationships:**
 - Many-to-Many relationship between LabTest and Patient.

8. Department Table

This table manages information about hospital departments.

- **Attributes:**
 - DepartmentID (Primary Key): Unique identifier for the department.
 - DepartmentName: Name of the department.
 - Location: Location of the department.
- **Relationships:**
 - Referenced by the Staff table.

9. Staff Table

This table stores information about hospital staff.

- **Attributes:**
 - StaffID (Primary Key): Unique identifier for each staff member.
 - FullName: Name of the staff member.
 - ContactNumber: Contact number of the staff member.
 - DepartmentID (Foreign Key): Links to Department(DepartmentID).
- **Relationships:**
 - One-to-Many relationship with the Department table.
 - Referenced by Doctor and Nurse tables.

10. Doctor Table

This table records information about doctors.

- **Attributes:**
 - DoctorID (Primary Key): Unique identifier for each doctor.
 - Specialty: Specialization of the doctor.
 - StaffID (Unique, Foreign Key): Links to Staff(StaffID).
- **Relationships:**
 - One-to-One with the Staff table.
 - Referenced by Appointment and Treatment tables.

11. Nurse Table

This table records information about nurses.

- **Attributes:**
 - NurseID (Primary Key): Unique identifier for each nurse.
 - StaffID (Unique, Foreign Key): Links to Staff(StaffID).
- **Relationships:**
 - One-to-One with the Staff table.

12. Appointment Table

This table handles patient appointments with doctors.

- **Attributes:**
 - AppointmentID (Primary Key): Unique identifier for each appointment.
 - DateTime: Date and time of the appointment.
 - Status: Status of the appointment (e.g., Scheduled, Completed).
 - PatientID (Foreign Key): Links to Patient(PatientID).
 - DoctorID (Foreign Key): Links to Doctor(DoctorID).
- **Relationships:**
 - Many-to-Many relationship between Patient and Doctor.

13. Treatment Table

This table records treatments provided to patients.

- **Attributes:**
 - TreatmentID (Primary Key): Unique identifier for each treatment.
 - TreatmentDescription: Description of the treatment.
 - DateOfTreatment: Date the treatment was provided.
 - PatientID (Foreign Key): Links to Patient(PatientID).
 - DoctorID (Foreign Key): Links to Doctor(DoctorID).
- **Relationships:**
 - One-to-Many with Patient and Doctor tables.
 - Referenced by the TreatMedication table.

14. Medication Table

This table manages information about medications.

- **Attributes:**

- MedicationID (Primary Key): Unique identifier for each medication.
- Name: Name of the medication.
- Dosage: Dosage details for the medication.
- **Relationships:**
 - Referenced by the TreatMedication table.

15. TreatMedication Table

This table represents the relationship between treatments and medications.

- **Attributes:**
 - TreatmentID (Primary Key, Foreign Key): Links to Treatment(TreatmentID).
 - MedicationID (Primary Key, Foreign Key): Links to Medication(MedicationID).
- **Relationships:**
 - Many-to-Many relationship between Treatment and Medication.

2. Queries with Detailed Explanations

Basic Queries

1. Retrieve all patients' basic information.

```
-- 1. Retrieve all patients' basic information.
SELECT PatientID, FullName, DateOfBirth, EmailAddress
FROM Patient;
```

	PatientID	FullName	DateOfBirth	EmailAddress
1	1	John Smith	2001-01-15	john.smith@gmail.com
2	2	Mary Johnson	2003-03-22	mary.johnson@gmail.com
3	3	Robert Brown	2000-07-08	robert.brown@gmail.com
4	4	Jennifer Davis	2005-12-02	jennifer.davis@gmail.com
5	5	Michael Wilson	1995-05-10	michael.wilson@gmail.com
6	6	Emily Moore	2002-11-23	emily.moore@gmail.com
7	7	Daniel Taylor	1975-08-19	daniel.taylor@gmail.com
8	8	Sarah Anderson	2008-02-15	sarah.anderson@gmail.com
9	9	David Thomas	2007-06-11	david.thomas@gmail.com
10	10	Laura Jackson	2010-09-25	laura.jackson@gmail.com
11	11	James White	1977-04-30	james.white@gmail.com
12	12	Linda Harris	2004-10-09	linda.harris@gmail.com
13	13	Andrew Martin	1981-01-20	andrew.martin@gmail.com
14	14	Elizabeth Thompson	2006-07-14	elizabeth.thompson@gmail.com
15	15	Christopher Garcia	2009-03-07	christopher.garcia@gmail.com
16	16	Patricia Martinez	1988-05-18	patricia.martinez@gmail.com
17	17	Matthew Robinson	2003-11-01	matthew.robinson@gmail.com
18	18	Barbara Clark	1991-02-25	barbara.clark@gmail.com
19	19	Joshua Lewis	2000-06-30	joshua.lewis@gmail.com
20	20	Karen Hall	2012-08-08	karen.hall@gmail.com

- **Function:** Displays basic details for all patients, including ID, name, date of birth, and email address.
- **Purpose:** Provides a quick overview of patient records.

2. Retrieve all doctors' names and their specialties.

```
-- 2. Retrieve all doctors' names and their specialties.
SELECT DoctorID, Specialty
FROM Doctor;
```

	DoctorID	Specialty
1	1	Cardiology
2	2	Neurology
3	3	Orthopedics
4	4	Radiology
5	5	Pediatrics
6	6	Oncology
7	7	Emergency
8	8	Dermatology
9	9	Gastroenterology
10	10	Endocrinology
11	11	Urology
12	12	Ophthalmology
13	13	Psychiatry
14	14	Surgery
15	15	Anesthesiology
16	16	Nephrology
17	17	Pulmonology
18	18	Hematology
19	19	Rheumatology
20	20	Rehabilitation

- **Function:** Shows the IDs and specialties of all doctors.
- **Purpose:** Understands the distribution of medical specialties in the hospital.

3. Retrieve all scheduled appointments after a specific date.

```
-- 3. Retrieve all scheduled appointments after a specific date.
SELECT AppointmentID, DateTime, Status, PatientID, DoctorID
FROM Appointment
WHERE DateTime > '2024-01-01' AND Status = 'Scheduled';
```

	AppointmentID	DateTime	Status	PatientID	DoctorID
1	13	2024-01-05 12:30:00.000	Scheduled	13	13
2	14	2024-02-10 13:30:00.000	Scheduled	14	14
3	16	2024-04-20 15:30:00.000	Scheduled	16	16
4	17	2024-05-25 16:30:00.000	Scheduled	17	17
5	19	2024-07-15 18:30:00.000	Scheduled	19	19
6	20	2024-08-20 09:30:00.000	Scheduled	20	20

- **Function:** Lists appointments scheduled after a specific date, including appointment ID, time, status, patient, and doctor details.
- **Purpose:** Tracks upcoming appointments.

4. Retrieve all medications with their dosages.

```
-- 4. Retrieve all medications with their dosages.
SELECT MedicationID, Name, Dosage
FROM Medication;
```

	MedicationID	Name	Dosage
1	1	Paracetamol	500mg
2	2	Amoxicillin	250mg
3	3	Ibuprofen	400mg
4	4	Ciprofloxacin	500mg
5	5	Metformin	850mg
6	6	Atorvastatin	10mg
7	7	Omeprazole	20mg
8	8	Losartan	50mg
9	9	Aspirin	81mg
10	10	Clopidogrel	75mg
11	11	Levothyroxine	100mcg
12	12	Cetirizine	10mg
13	13	Loratadine	10mg
14	14	Prednisone	20mg
15	15	Albuterol	90mcg
16	16	Insulin	10 units
17	17	Metoprolol	25mg
18	18	Gabapentin	300mg
19	19	Hydrochlorothiazide	25mg
20	20	Fluoxetine	20mg

- **Function:** Lists all medications and their dosages.
- **Purpose:** Displays a catalog of available medications.

5. Retrieve all treatments provided by a specific doctor.

```
-- 5. Retrieve all treatments provided by a specific doctor.  
SELECT TreatmentID, TreatmentDescription, DateOfTreatment  
FROM Treatment  
WHERE DoctorID = 1;
```

TreatmentID	TreatmentDescription	DateOfTreatment
1	Routine Checkup	2023-01-16

- **Function:** Lists treatments conducted by a specific doctor (DoctorID = 1), including treatment description and date.
- **Purpose:** Tracks the treatment history of an individual doctor.

Aggregate and Statistical Queries

6. Count the total number of patients in the system.

```
-- 6. Count the total number of patients in the system.  
SELECT COUNT(*) AS TotalPatients  
FROM Patient;
```

TotalPatients
20

- **Function:** Counts the total number of patients in the hospital system.
- **Purpose:** Provides an overall patient count.

7. Count the number of appointments completed by each doctor.

```
-- 7. Count the number of appointments completed by each doctor.  
SELECT DoctorID, COUNT(AppointmentID) AS CompletedAppointments  
FROM Appointment  
WHERE Status = 'Completed'  
GROUP BY DoctorID;
```

DoctorID	CompletedAppointments
2	1
5	1
7	1
10	1
12	1
15	1
18	1

- **Function:** Counts the number of appointments completed by each doctor.
- **Purpose:** Analyzes doctor workload and completion rates.

8. Retrieve the total amount of bills paid by all patients.

```
-- 8. Retrieve the total amount of bills paid by all patients.  
SELECT SUM(Amount) AS TotalPaidBills  
FROM Bill  
WHERE PaymentStatus = 'Paid';
```

177 %

Results Messages

	TotalPaidBills
1	3500.00

- **Function:** Calculates the total amount of all paid bills.
- **Purpose:** Tracks total hospital revenue from paid bills.

9. Count the number of treatments provided for each specialty.

```
-- 9. Count the number of treatments provided for each specialty.  
SELECT Doctor.Specialty, COUNT(Treatment.TreatmentID) AS TreatmentCount  
FROM Treatment  
JOIN Doctor ON Treatment.DoctorID = Doctor.DoctorID  
GROUP BY Doctor.Specialty;
```

177 %

Results Messages

	Specialty	TreatmentCount
1	Anesthesiology	1
2	Cardiology	1
3	Dermatology	1
4	Emergency	1
5	Endocrinology	1
6	Gastroenterology	1
7	Hematology	1
8	Nephrology	1
9	Neurology	1
10	Oncology	1
11	Ophthalmology	1
12	Orthopedics	1
13	Pediatrics	1
14	Psychiatry	1
15	Pulmonology	1
16	Radiology	1
17	Rehabilitation	1
18	Rheumatology	1
19	Surgery	1
20	Urology	1

- **Function:** Counts the treatments provided, grouped by doctor specialty.
- **Purpose:** Evaluates the workload of each department or specialty.

Join Queries

10. Retrieve all patients along with their contact numbers.

```
-- 10. Retrieve all patients along with their contact numbers.
SELECT Patient.FullName, PatientContact.ContactNumber
FROM Patient
JOIN PatientContact ON Patient.PatientID = PatientContact.PatientID;
```

	FullName	ContactNumber
1	John Smith	196183079
2	Mary Johnson	183828492
3	Robert Brown	145827931
4	Jennifer Davis	193782640
5	Michael Wilson	134576981
6	Emily Moore	168297430
7	Daniel Taylor	192874930
8	Sarah Anderson	148273092
9	David Thomas	172839400
10	Laura Jackson	193284762
11	James White	182763492
12	Linda Harris	147293860
13	Andrew Martin	192873620
14	Elizabeth Thompson	183920740
15	Christopher Garcia	195847201
16	Patricia Martinez	132846970
17	Matthew Robinson	174839205
18	Barbara Clark	183729405
19	Joshua Lewis	172839560
20	Karen Hall	198273640

- **Function:** Lists all patients with their contact numbers.
- **Purpose:** Quickly accesses patient contact information.

11. Retrieve all appointments along with the doctors and patients involved.

```
-- 11. Retrieve all appointments along with the doctors and patients involved.
SELECT Appointment.DateTime, Patient.FullName AS PatientName, Doctor.Specialty AS DoctorSpecialty
FROM Appointment
JOIN Patient ON Appointment.PatientID = Patient.PatientID
JOIN Doctor ON Appointment.DoctorID = Doctor.DoctorID;
```

	DateTime	PatientName	DoctorSpecialty
1	2023-01-15 10:00:00.000	John Smith	Cardiology
2	2023-02-20 11:00:00.000	Mary Johnson	Neurology
3	2023-03-25 12:00:00.000	Robert Brown	Orthopedics
4	2023-04-10 13:00:00.000	Jennifer Davis	Radiology
5	2023-05-05 14:00:00.000	Michael Wilson	Pediatrics
6	2023-06-12 15:00:00.000	Emily Moore	Oncology
7	2023-07-19 16:00:00.000	Daniel Taylor	Emergency
8	2023-08-15 17:00:00.000	Sarah Anderson	Dermatology
9	2023-09-20 18:00:00.000	David Thomas	Gastroenterology
10	2023-10-25 09:00:00.000	Laura Jackson	Endocrinology
11	2023-11-30 10:30:00.000	James White	Urology
12	2023-12-15 11:30:00.000	Linda Harris	Ophthalmology
13	2024-01-05 12:30:00.000	Andrew Martin	Psychiatry
14	2024-02-10 13:30:00.000	Elizabeth Thompson	Surgery
15	2024-03-15 14:30:00.000	Christopher Garcia	Anesthesiology
16	2024-04-20 15:30:00.000	Patricia Martinez	Nephrology
17	2024-05-25 16:30:00.000	Matthew Robinson	Pulmonology
18	2024-06-30 17:30:00.000	Barbara Clark	Hematology
19	2024-07-15 18:30:00.000	Joshua Lewis	Rheumatology
20	2024-08-20 09:30:00.000	Karen Hall	Rehabilitation

- **Function:** Shows the time of appointments, patient names, and doctor specialties.
- **Purpose:** Visualizes appointment participants and schedules.

12. Retrieve all treatments along with the medications used.

```
-- 12. Retrieve all treatments along with the medications used.
SELECT Treatment.TreatmentDescription, Medication.Name AS MedicationName
FROM TreatMedication
JOIN Treatment ON TreatMedication.TreatmentID = Treatment.TreatmentID
JOIN Medication ON TreatMedication.MedicationID = Medication.MedicationID;
```

	TreatmentDescription	MedicationName
1	Routine Checkup	Paracetamol
2	MRI Analysis	Amoxicillin
3	Fracture Fixation	Ibuprofen
4	Radiology Review	Ciprofloxacin
5	Vaccination	Metformin
6	Oncology Screening	Atorvastatin
7	Emergency Response	Omeprazole
8	Skin Treatment	Losartan
9	Cardiology Consultation	Aspirin
10	Neurology Test	Clopidogrel
11	Joint Repair	Levothyroxine
12	Radiology Scan	Cetirizine
13	Pediatrics Follow-up	Loratadine
14	Cancer Diagnosis	Prednisone
15	Surgery Preparation	Albuterol
16	Dermatology Consultation	Insulin
17	Heart Monitoring	Metoprolol
18	Brain Scan	Gabapentin
19	Orthopedics Evaluation	Hydrochlorothiazide
20	Radiology Test	Fluoxetine

- **Function:** Lists treatments with the medications used for each treatment.
- **Purpose:** Tracks the relationship between treatments and medications.

13. Retrieve the list of patients who have undergone lab tests.

```
-- 13. Retrieve the list of patients who have undergone lab tests.
SELECT DISTINCT Patient.FullName AS PatientName
FROM LabTestInstances
JOIN Patient ON LabTestInstances.PatientID = Patient.PatientID;
```

	PatientName
1	Andrew Martin
2	Barbara Clark
3	Christopher Garcia
4	Daniel Taylor
5	David Thomas
6	Elizabeth Thompson
7	Emily Moore
8	James White
9	Jennifer Davis
10	John Smith
11	Joshua Lewis
12	Karen Hall
13	Laura Jackson
14	Linda Harris
15	Mary Johnson
16	Matthew Robinson
17	Michael Wilson
18	Patricia Martinez
19	Robert Brown
20	Sarah Anderson

- **Function:** Displays a unique list of patients who have completed lab tests.
- **Purpose:** Understands the coverage of lab testing among patients.

Conditional and Advanced Queries

14. Retrieve patients who have unpaid bills.

```
-- 14. Retrieve patients who have unpaid bills.
SELECT Patient.FullName, Bill.Amount, Bill.PaymentStatus
FROM Bill
JOIN Patient ON Bill.PatientID = Patient.PatientID
WHERE Bill.PaymentStatus = 'Pending';
```

	FullName	Amount	PaymentStatus
1	Mary Johnson	300.00	Pending
2	Jennifer Davis	450.00	Pending
3	Emily Moore	350.00	Pending
4	Sarah Anderson	400.00	Pending
5	Laura Jackson	300.00	Pending
6	Linda Harris	150.00	Pending
7	Elizabeth Thompson	330.00	Pending
8	Patricia Martinez	390.00	Pending
9	Barbara Clark	410.00	Pending
10	Karen Hall	230.00	Pending

- **Function:** Lists patients with unpaid bills, including the bill amount and status.
- **Purpose:** Identifies overdue payments for follow-up.

15. Retrieve all patients who visited a specific department.

```
-- 15. Retrieve all patients who visited a specific department.
SELECT DISTINCT Patient.FullName AS PatientName, Department.DepartmentName
FROM Appointment
JOIN Patient ON Appointment.PatientID = Patient.PatientID
JOIN Doctor ON Appointment.DoctorID = Doctor.DoctorID
JOIN Staff ON Doctor.StaffID = Staff.StaffID
JOIN Department ON Staff.DepartmentID = Department.DepartmentID
WHERE Department.DepartmentName = 'Cardiology';
```

	PatientName	DepartmentName
1	David Thomas	Cardiology
2	John Smith	Cardiology
3	Matthew Robinson	Cardiology

- **Function:** Lists patients who visited a specific department (e.g., Cardiology).
- **Purpose:** Tracks patient volume for a specific department.

16. Retrieve the most recent appointment for each patient.

```
-- 16. Retrieve the most recent appointment for each patient.
SELECT PatientID, MAX(DateTime) AS MostRecentAppointment
FROM Appointment
GROUP BY PatientID;
```

	PatientID	MostRecentAppointment
1	1	2023-01-15 10:00:00.000
2	2	2023-02-20 11:00:00.000
3	3	2023-03-25 12:00:00.000
4	4	2023-04-10 13:00:00.000
5	5	2023-05-05 14:00:00.000
6	6	2023-06-12 15:00:00.000
7	7	2023-07-19 16:00:00.000
8	8	2023-08-15 17:00:00.000
9	9	2023-09-20 18:00:00.000
10	10	2023-10-25 09:00:00.000
11	11	2023-11-30 10:30:00.000
12	12	2023-12-15 11:30:00.000
13	13	2024-01-05 12:30:00.000
14	14	2024-02-10 13:30:00.000
15	15	2024-03-15 14:30:00.000
16	16	2024-04-20 15:30:00.000
17	17	2024-05-25 16:30:00.000
18	18	2024-06-30 17:30:00.000
19	19	2024-07-15 18:30:00.000
20	20	2024-08-20 09:30:00.000

- **Function:** Displays the most recent appointment time for each patient.
- **Purpose:** Tracks the latest interactions for each patient.

17. Retrieve the total revenue generated by each department.

```
-- 17. Retrieve the total revenue generated by each department.
SELECT Department.DepartmentName, SUM(Bill.Amount) AS TotalRevenue
FROM Bill
JOIN Patient ON Bill.PatientID = Patient.PatientID
JOIN Appointment ON Patient.PatientID = Appointment.PatientID
JOIN Doctor ON Appointment.DoctorID = Doctor.DoctorID
JOIN Staff ON Doctor.StaffID = Staff.StaffID
JOIN Department ON Staff.DepartmentID = Department.DepartmentID
WHERE Bill.PaymentStatus = 'Paid'
GROUP BY Department.DepartmentName;
```

	DepartmentName	TotalRevenue
1	Cardiology	1100.00
2	Emergency	480.00
3	Orthopedics	1100.00
4	Pediatrics	820.00

- **Function:** Calculates the total revenue for each department based on paid bills.
- **Purpose:** Evaluates the financial contribution of each department.

18. Retrieve the number of appointments scheduled for each month.

```
-- 18. Retrieve the number of appointments scheduled for each month.  
SELECT MONTH(DateTime) AS Month, COUNT(*) AS AppointmentCount  
FROM Appointment  
GROUP BY MONTH(DateTime);
```

	Month	AppointmentCount
1	1	2
2	2	2
3	3	2
4	4	2
5	5	2
6	6	2
7	7	2
8	8	2
9	9	1
10	10	1
11	11	1
12	12	1

- **Function:** Counts the number of appointments grouped by month.
- **Purpose:** Analyzes monthly trends in appointment scheduling.

3. User Manual for Hospital Management System

Introduction

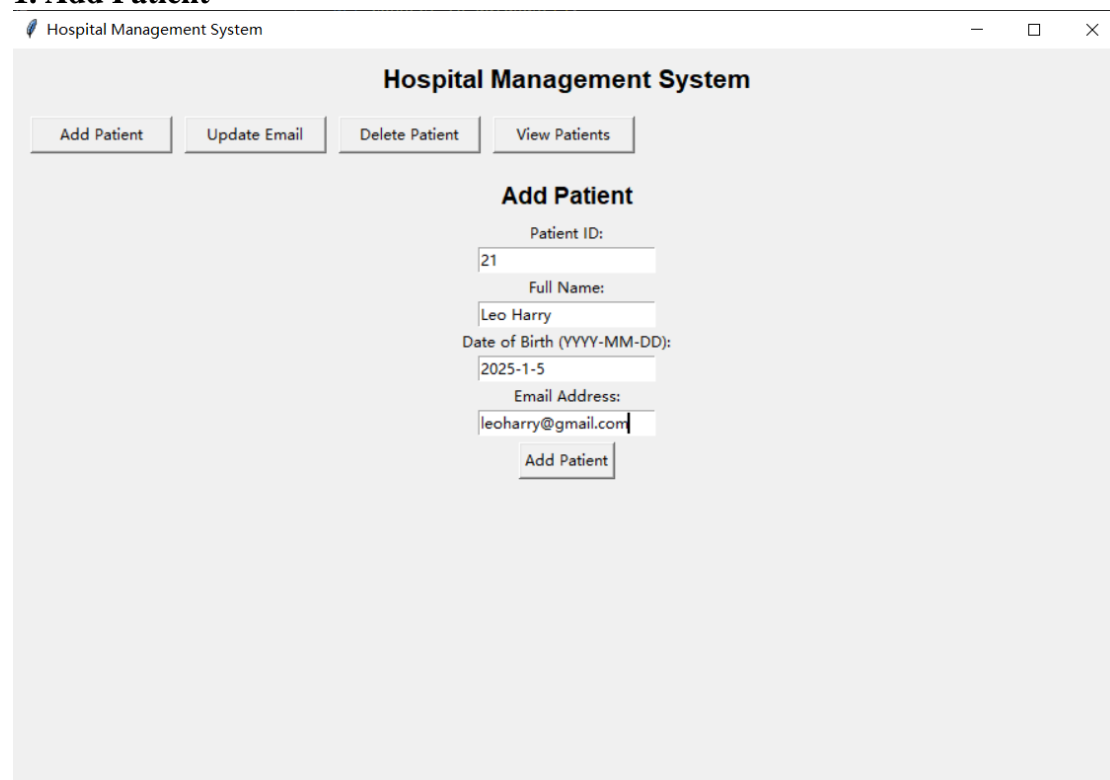
This application is a simple Hospital Management System designed for managing patient data efficiently. It provides a graphical user interface (GUI) for performing basic database operations such as adding, updating, deleting, and viewing patient records.

Main Features

1. **Add Patient:** Add a new patient record to the database.
2. **Update Email:** Update the Email Address of an existing patient.
3. **Delete Patient:** Delete a patient record from the database.
4. **View Patients:** View all patient records or search for a specific patient by ID.

How to Use the Application

1. Add Patient

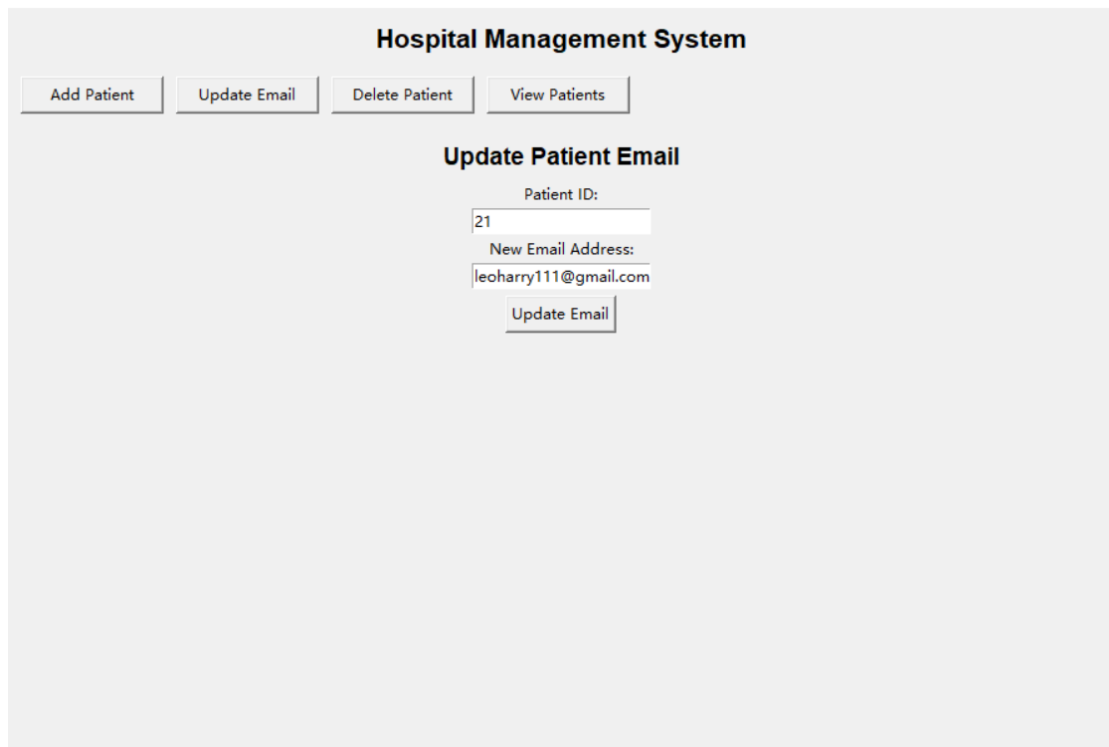


The screenshot displays the 'Hospital Management System' application window. At the top, there is a title bar with the text 'Hospital Management System' and standard window controls (minimize, maximize, close). Below the title bar, the main window has a header 'Hospital Management System' and four buttons: 'Add Patient', 'Update Email', 'Delete Patient', and 'View Patients'. The 'Add Patient' button is highlighted. Below these buttons, the 'Add Patient' form is visible. It contains the following fields: 'Patient ID:' with the value '21', 'Full Name:' with the value 'Leo Harry', 'Date of Birth (YYYY-MM-DD):' with the value '2025-1-5', and 'Email Address:' with the value 'leoharry@gmail.com'. An 'Add Patient' button is located at the bottom of the form.

- **Steps to Add a Patient:**
 1. Click on the "Add Patient" button at the top of the screen.
 2. Fill in the required details in the form:
 - **Patient ID:** A unique identifier for the patient.
 - **Full Name:** The full name of the patient.
 - **Date of Birth:** Enter the date in YYYY-MM-DD format.
 - **Email Address:** Enter the patient's email address.
 3. Click the "Add Patient" button below the textboxes to save the record.
 4. If successful, a confirmation message will appear.

2. Update Email

Hospital Management System



The screenshot shows a web application window titled "Hospital Management System". At the top, there are four buttons: "Add Patient", "Update Email", "Delete Patient", and "View Patients". The "Update Email" button is highlighted. Below these buttons, the section is titled "Update Patient Email". It contains two text input fields. The first field is labeled "Patient ID:" and contains the value "21". The second field is labeled "New Email Address:" and contains the value "leoharry111@gmail.com". Below these fields is a button labeled "Update Email".

- **Steps to Update a Patient's Email:**
 1. Click on the "**Update Email**" button at the top of the screen.
 2. Enter the following details:
 - **Patient ID:** The ID of the patient whose contact details need to be updated.
 - **New Email Address:** Enter the updated email address.
 3. Click the "**Update Email**" button below the textboxes.
 4. A confirmation message will appear if the update is successful.

3. Delete Patient

Hospital Management System

Hospital Management System

Add PatientUpdate EmailDelete PatientView Patients

Delete Patient

Patient ID:

Delete Patient

- **Steps to Delete a Patient:**

1. Click on the **"Delete Patient"** button at the top of the screen.
2. Enter the **Patient ID** of the record to be deleted.
3. Click the **"Delete Patient"** button below the textbox.
4. A confirmation message will appear if the deletion is successful.

4. View Patients

Hospital Management System

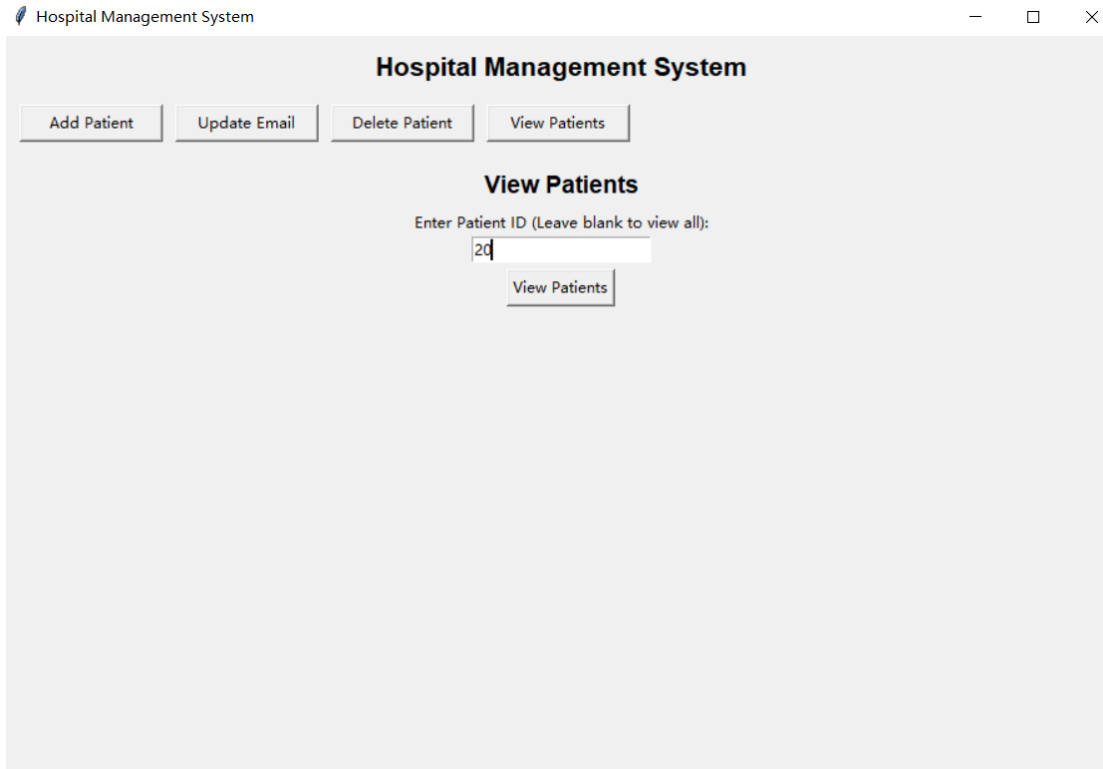
Hospital Management System

Add PatientUpdate EmailDelete PatientView Patients

Patient List

ID	Name	Date of Birth	Email
1	John Smith	2001-01-15	john.smith@gmail.com
2	Mary Johnson	2003-03-22	mary.johnson@gmail.com
3	Robert Brown	2000-07-08	robert.brown@gmail.com
4	Jennifer Davis	2005-12-02	jennifer.davis@gmail.com
5	Michael Wilson	1995-05-10	michael.wilson@gmail.com
6	Emily Moore	2002-11-23	emily.moore@gmail.com
7	Daniel Taylor	1975-08-19	daniel.taylor@gmail.com
8	Sarah Anderson	2008-02-15	sarah.anderson@gmail.com
9	David Thomas	2007-06-11	david.thomas@gmail.com
10	Laura Jackson	2010-09-25	laura.jackson@gmail.com
11	James White	1977-04-30	james.white@gmail.com
12	Linda Harris	2004-10-09	linda.harris@gmail.com
13	Andrew Martin	1981-01-20	andrew.martin@gmail.com
14	Elizabeth Thompson	2006-07-14	elizabeth.thompson@gmail.com
15	Christopher Garcia	2009-03-07	christopher.garcia@gmail.com
16	Patricia Martinez	1988-05-18	patricia.martinez@gmail.com
17	Matthew Robinson	2003-11-01	matthew.robinson@gmail.com
18	Barbara Clark	1991-02-25	barbara.clark@gmail.com

- **Steps to View All Patients:**
 1. Leave the textbox blank and click on the "**View Patients**" button below the textbox.
 2. All patient records will be displayed in a table format.
 3. Each record includes:
 - **ID:** Patient ID
 - **Name:** Patient's Name
 - **Date of Birth:** Patient's Date of Birth
 - **Email:** Patient's Email Address
- **Steps to Search for a Specific Patient:**



The screenshot displays the 'Hospital Management System' window. At the top, there is a title bar with the text 'Hospital Management System' and standard window controls (minimize, maximize, close). Below the title bar, the main content area has a header 'Hospital Management System'. Underneath the header, there are four buttons: 'Add Patient', 'Update Email', 'Delete Patient', and 'View Patients'. The 'View Patients' button is highlighted. Below these buttons, the section is titled 'View Patients'. Under this title, there is a text prompt 'Enter Patient ID (Leave blank to view all):' followed by a text input field containing the number '20'. Below the input field is a button labeled 'View Patients'.

1. Enter the **Patient ID** in the search field.
2. Click the "**View Patients**" button.
3. The record of the specific patient will be displayed.

Hospital Management System

Add PatientUpdate EmailDelete PatientView Patients

Patient List

ID	Name	Date of Birth	Email
20	Karen Hall	2012-08-08	karen.hall@gmail.com

Error Handling

1. Missing Required Fields:

- If any required field is left blank, an error message will be displayed.
- Ensure all fields are filled before submitting the form.

2. Invalid Input:

- Ensure the **Patient ID** is unique and matches the database record when updating or deleting.
- Use the correct format for the **Date of Birth** (YYYY-MM-DD) and valid phone numbers/email addresses.

3. Database Connection Error:

- If the application cannot connect to the database, check the server connection and retry.