Punjab University College of Information Technology

BS (SE) Fall 2022 Morning

Web Engineering

Lab 03 - Introduction to SQL Databases

Time Duration: 2 hr, Total Marks: 40

Objective:

- Introduction to the Flight Booking System
- Setting Up the Database
- Creating the Flight Class
- Adding New Flights (Create)
- Retrieving Flights (Read)
- Updating Flight Information (Update)
- Deleting Flights (Delete)
- Storing Flight Data in JSON Format
- Reading Flight Data from JSON

Introduction to the Flight Booking System

In this lab, you will create a flight booking system that allows you to perform CRUD (Create, Read, Update, Delete) operations using ADO.NET. Additionally, you will store and retrieve flight data using JSON format, providing insight into data serialization.

Task 1: Setting Up the Database

Objective: Create a SQL Server database and a table for storing flight information.
Instructions:

- Create a new database named `FlightBooking`.
- Create a table named `Flights` with the following fields:
  - Id (int, primary key, auto-increment)
  - FlightNumber
  - Destination
  - Departure
  - Arrival
  - Price

Field	Data Type	Description
Id	int (Primary Key)	Auto-incrementing identifier
FlightNumber	nvarchar(50)	Unique flight number
Destination	nvarchar(100)	Flight destination
Departure	datetime	Departure date and time
Arrival	datetime	Arrival date and time
Price	decimal	Ticket price

Task 2: Creating the Flight Class

Objective: Create a C# class to represent the flight entity.
Instructions:

- Create a `Flight` class with properties for `Id`, `FlightNumber`, `Destination`, `Departure`, `Arrival`, and `Price`.

Task 3: Adding New Flights

Objective: Implement a method to add new flights to the database.
Instructions:

- Create a method `AddFlight` that takes a `Flight` object as a parameter and inserts it into the `Flights` table using parameterized queries.

Task 4: Retrieving Flights

Objective: Implement a method to retrieve all flights from the database.
Instructions:

- Create a method `GetAllFlights` that retrieves all records from the `Flights` table and returns a list of `Flight` objects.

Task 5: Updating Flight Information

Objective: Implement a method to update flight details.
Instructions:

- Create a method `UpdateFlight` that takes a `Flight` object as a parameter and updates the corresponding record in the database.

Task 6: Deleting Flights

Objective: Implement a method to delete flights from the database.
Instructions:

- Create a method `DeleteFlight` that takes an `Id` as a parameter and removes the corresponding flight from the `Flights` table.

Task 7: Storing Flight Data in JSON Format

Objective: Serialize the list of flights to a JSON file.
Instructions:

- Create a method `SaveFlightsToJson` that retrieves all flights and saves them to a JSON file using serialization.

Task 8: Reading Flight Data from JSON

Objective: Deserialize flight data from a JSON file and insert it into the database.
Instructions:

- Create a method `LoadFlightsFromJson` that reads flight data from a JSON file and adds each flight to the database.