C# Project Report

**1. Project Overview**

**1.1 Key Features Implemented**

1. User registration and login system with role-based password protection for users.

2. Create, Read, Update, Delete for managing records.

3. Data validation and error handling.

4. GUI with responsive form design using Windows Forms

5. Database integration using SQLite.

6.Admin only add , update ,Delete other user’s data

7.If a student logs into the app, they can’t add, update, and delete records, and also user management system will not be accessible to them.

8. If a staff or lecture logs into the app they can’t add, update and delete records, but they can access student’s management system

**1.2 Technologies Used**

1. Programming Language: C#

2. Framework: .NET Framework

3. IDE: Visual Studio

4. Database: SQLite

5. UI Framework: Windows Forms

6. "Additional Tools: GitHub, ChatGPT (used for guidance and reference)

**1.3 Challenges Faced and How They Were Solved**

1. Challenge: Difficulty with database connectivity

Solution: I learned from YouTube and used some references from ChatGPT to understand database connections."

2. Challenge: hard to handle Click events and panel organizing and panels changing while click buttons

Solution: I asked ChatGPT to explain it step by step and also learned about it through YouTube videos."

3. Challenge: UI responsiveness and layout issues

Solution: Used layout panels to make UI adaptive and cleaner.

4.Challenge I struggled to push and pull the project to GitHub; sometimes it would completely crashed.

Solution: I saved a backup before pushing the project to GitHub and got help from seniors to use GitHub correctly.

**2. Code Samples (Screenshots)**

**A screen shot of a computer program

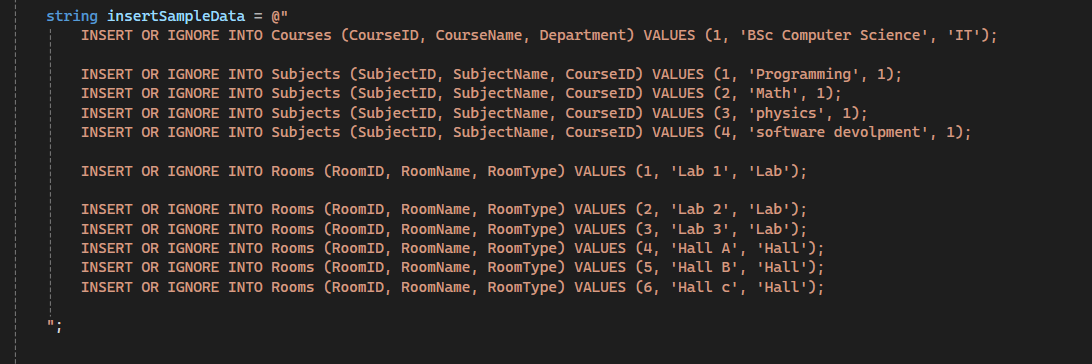
AI-generated content may be incorrect.2.1 . This method (SetButtonAccessBasedOnRole) enables or disables specific buttons on a WinForms UI based on the logged-in user's role (Admin, Staff, Lecture, or Student) to control access to different features.**

**2.2 . This method handles the login button click by validating inputs, authenticating the user with username, password, and role, and then opening the dashboard if successful or showing an error if not.**

**A computer screen with many colorful text

AI-generated content may be incorrect.**

**2.3 . This SQL script inserts sample data into the Courses, Subjects, and Rooms tables using INSERT OR IGNORE to prevent duplicate entries if they already exist.**

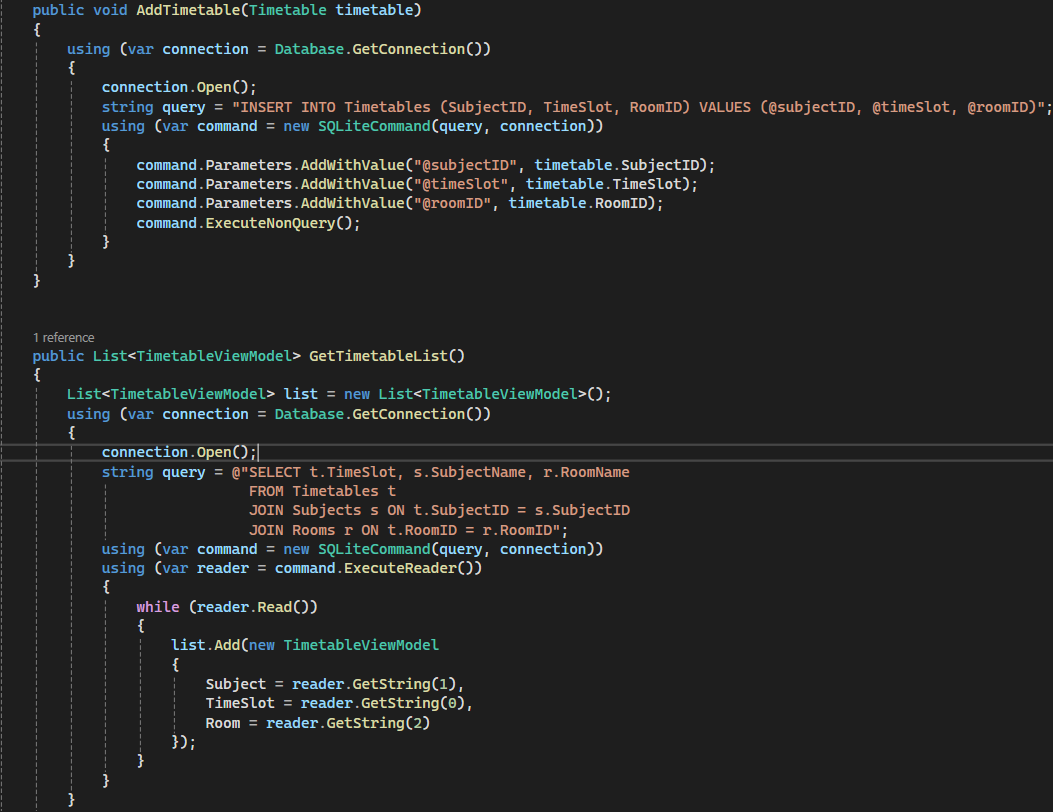


**2.4 . This Database class provides a static method GetConnection() that returns a new SQLite connection using the specified connection string to access the unicomticDB\_new.db database.**

A computer screen shot of a program

AI-generated content may be incorrect.

**2.5 .This code provides two methods: Add Timetable to insert a new timetable entry into the database, and GetTimetableList to retrieve a list of timetable records joined with subject and room names as a list of TimetableViewModel objects.**



**2.6 . This StudentController class provides methods to add, retrieve, update, and delete student records in the Students table using SQLite in a C# WinForms application.**

