**Project Report – Unicom TIC Management System**

Name: Nitharshini Jesuthasan

UT no Number: UT010430

Institution: UNICOM TIC

Group: A

Date: 24-June 202

**1. Project Title**

Unicom TIC Management System

# 2. Introduction

This project is a simple C# WinForms desktop application designed to manage school operations like Courses, Subjects, Students, Exams, Marks, and Timetables. It uses an SQLite database and includes a login system with four user roles: Admin, Staff, Lecturer, and Student.

# 3. Objectives

**Software Requirements:**

* **Windows Operating System**  
  This project is developed as a Windows Forms Application, which runs exclusively on the Windows operating system. WinForms is a UI framework that supports building desktop applications specifically for Windows.
* **Visual Studio**  
  Visual Studio is an Integrated Development Environment (IDE) from Microsoft. It provides tools such as Form Designer, Code Editor, and Debugging utilities, making it ideal for developing Windows Forms applications.
* **.NET Framework 4.x**  
  The project uses libraries like Windows Forms and System.Data.SQLite that are part of the .NET Framework 4.x. The C# programming language used here runs on this framework.
* **SQLite**  
  SQLite is a lightweight database engine that stores data in a single .db. file. In this project, tables such as Students, Courses, Exams, Marks, and Users are created within an SQLite database. The C# code accesses the SQLite database using the System.Data.SQLite library.

**Hardware Requirements:**

* **RAM: Minimum 4GB**  
  Visual Studio and Windows Forms applications require significant memory to build, debug, and run efficiently. Therefore, at least 4GB of RAM is recommended.
* **Processor: Intel i3 or higher**  
  To ensure smooth and fast building, debugging, and running of the application, a processor equivalent to Intel i3 or better is required.

# Module Description

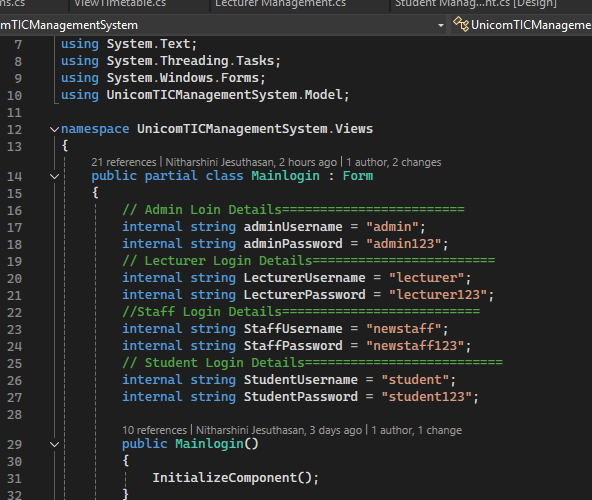
# Folder Structure

UnicomTICManagementSystem  
├── Models  
├── Views  
├── Controllers  
├── Repositories

## **Login Module**

Allows Admin and Student to log in. Role-based access is provided.

* The login system is designed using **Username and Password** authentication.
* It supports multiple **user roles**, including:
  + **Admin**
  + **Staff**
  + **Student**
  + **Lecturer**
* After a successful login, the application dynamically loads a **dashboard** based on the user's role.
* Each role will have **specific buttons and functionalities** displayed on their dashboard, allowing users to perform only the actions permitted for their role.

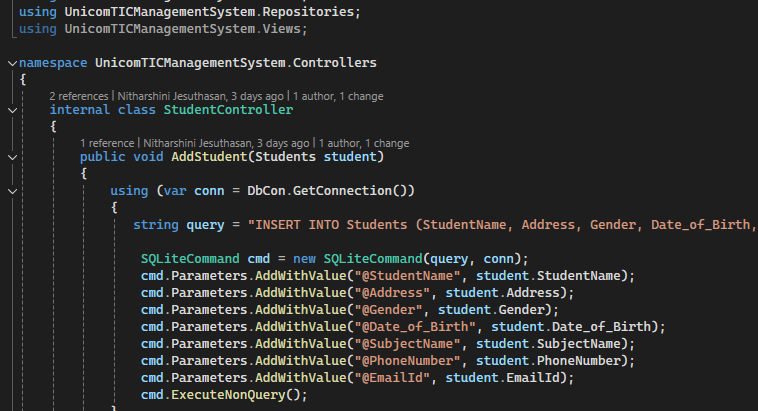


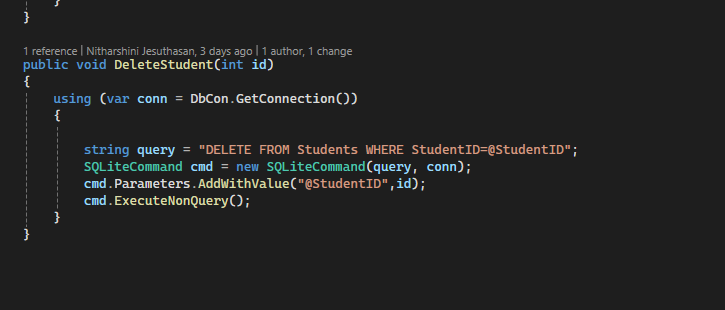
## **Student Module**

Handles student record operations such as Add, Edit, Delete, and View.

* Each student record includes the following details:
  + **StudentID**
  + **Name**
  + **CourseID**
* The **Admin** has full control over student data and can:
  + **Add** new students
  + **Update** existing student information
  + **Delete** student records
* A **Student** can only **view their own details** after logging in. They do not have permission to modify or view other students' data.

**Code Snippets (Optional but Useful)**

****

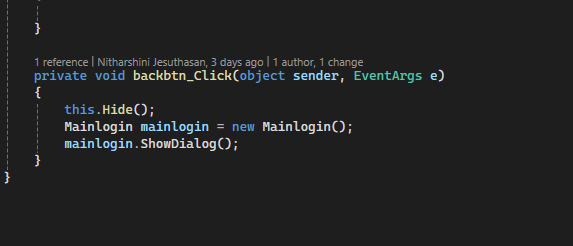


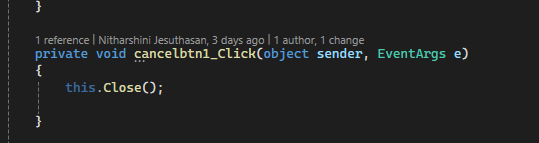
#### 🔹 **Back Button Functionality**

This button allows the user to navigate back to the previous form. For example, from the Student form to the Dashboard form.

#### 🔹 **Cancel Button Functionality**

This button clears all the inputs on the form or closes the current form without saving data.





# Conclusion

This project helped me to understand how to develop a desktop application using C#. It is useful for educational institutions to maintain student information in an efficient way.

# Future Enhancement

* • Add mobile application support
* • Integrate cloud storage
* • Add email notification system

# References

* Microsoft Docs
* Stack Overflow
* ChatGPT
* YouTube tutorials