Unicom Tic

Management System

**Module Title: - Windows Form**

**Student Name: - S.Lokaraj**

**Student ID: -UT010443**

**Course Title: - Windows Form application**

# Acknowledgement

I would like to express my sincere gratitude to all those who helped me throughout the development of my Student Management System project.

First, I would like to thank my instructor[ kathir sir], for their guidance, support, and valuable suggestions during the project.

I would also like to thank my friends and classmates for their encouragement and feedback during the various phases of development.

Lastly, I am thankful to my family for their continuous support and encouragement that motivated me to complete this project successfully.

This project helped me gain practical knowledge in C# Windows Forms, the MVC architecture, and database integration using SQLite.

By-Lokaraj

# Introduction

In today’s educational institutions, managing student data manually can be time-consuming and error-prone. To address this, a computerized Student Management System is essential for efficient administration.

This project, titled \*\*"Student Management System"\*\*, was developed using \*\*C# Windows Forms\*\*, following the \*\*Model-View-Controller (MVC)\*\* design pattern, and uses \*\*SQLite\*\* as the backend database.

The system allows administrators and staff to perform essential tasks such as adding, updating, and deleting records related to students, courses, subjects, exams, marks, rooms, and timetables.

The goal of this project is to automate and simplify the student data management process, reduce paperwork, and improve accuracy and accessibility of academic information.

Through this project, I have learned how to design and implement a real-world application using industry-standard technologies and architecture, ensuring modularity and maintainability of code.

Contents

**Objectives**

* To design and develop a comprehensive system for managing students, courses, subjects, exams, and marks.
* To reduce manual paperwork and data inconsistencies by automating administrative tasks.
* To provide secure and role-based access to different users.
* To enable easy data retrieval and reporting through a user-friendly interface.

**Software and Tools Used**

* **Programming Language:** C# (C Sharp)
* **Framework:** .NET Framework with Windows Forms for GUI
* **Database:** SQLite (lightweight, file-based database)
* **Design Pattern:** Model-View-Controller (MVC)
* **IDE:** Visual Studio
* **Other Tools:** Microsoft Word (for project report)

**System Architecture (MVC Model)**

The system follows the **MVC architecture** which separates the application into three main components:

* **Model:** Represents the data and business logic (e.g., Student, Course, Lecturer models).
* **View:** The User Interface (Windows Forms) that users interact with.
* **Controller:** Handles user input, communicates between Model and View, and updates the view.

# Module Descriptions

### Student

This module allows adding, updating, deleting, and viewing student records. Each student is linked to a course and has fields like Student ID, Name, Address, and Course.

### Course

Enables managing courses offered by the institution. Users can add new courses, update course names, and remove courses if needed.

### Subject

Manages subjects linked to courses. Users can add subjects, assign them to courses, update subject names, and remove subjects.

### Room and Timetable

This module handles classrooms (labs/halls) and the timetable schedule. Rooms are categorized by type. Timetables link subjects, lecturers, rooms, and timeslots.

### Exam

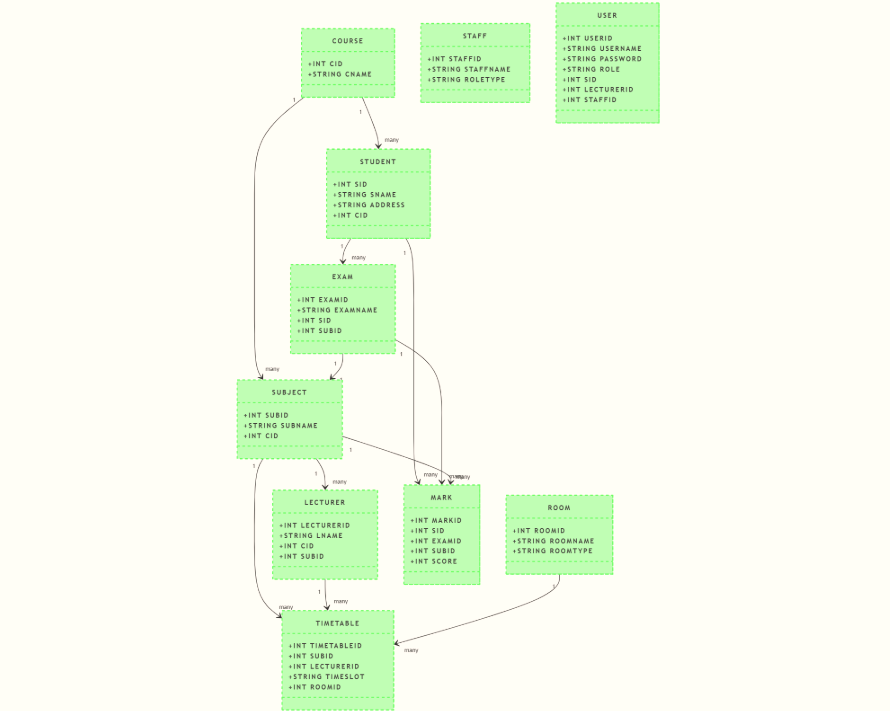
Allows the creation and management of exams associated with subjects and students. Users can schedule exams and maintain exam details.

### Marks

Manages marks scored by students in exams. It includes functionality for adding, updating, and deleting marks, and viewing detailed reports.

**Database Design**

**Entity Relationship Diagram (ERD)**

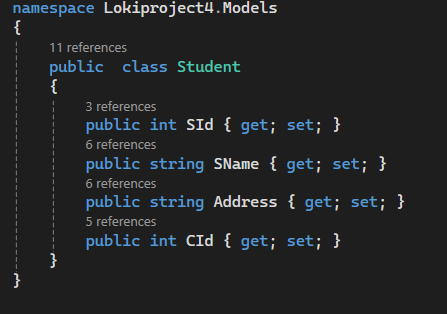


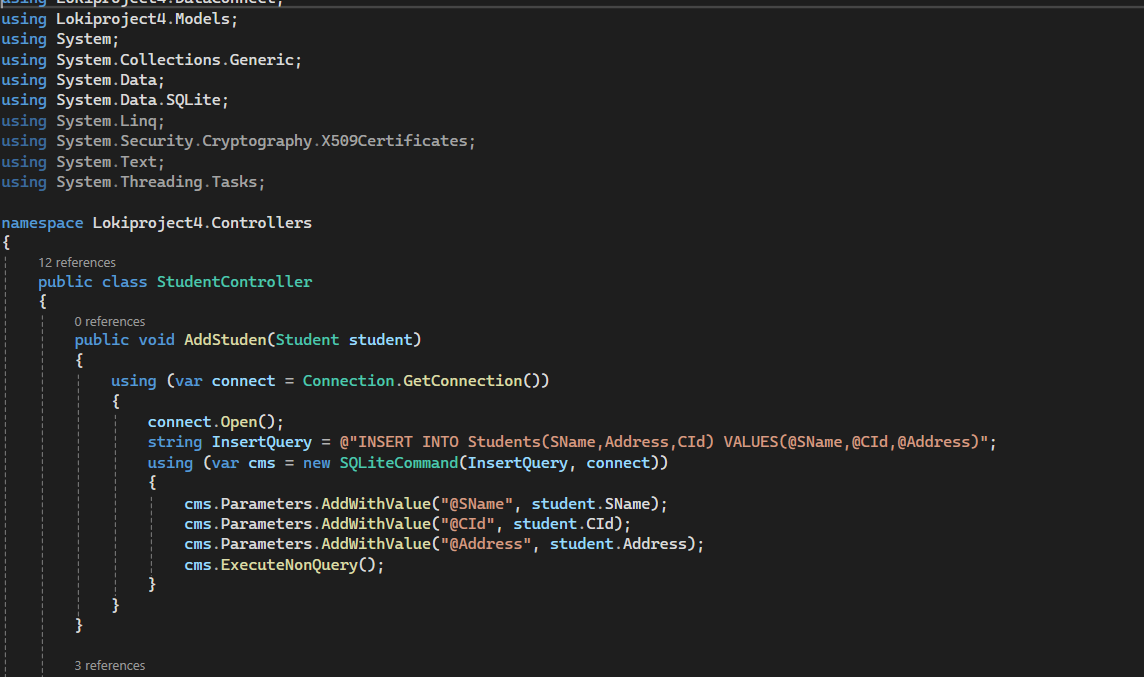
**Database Tables and Fields**

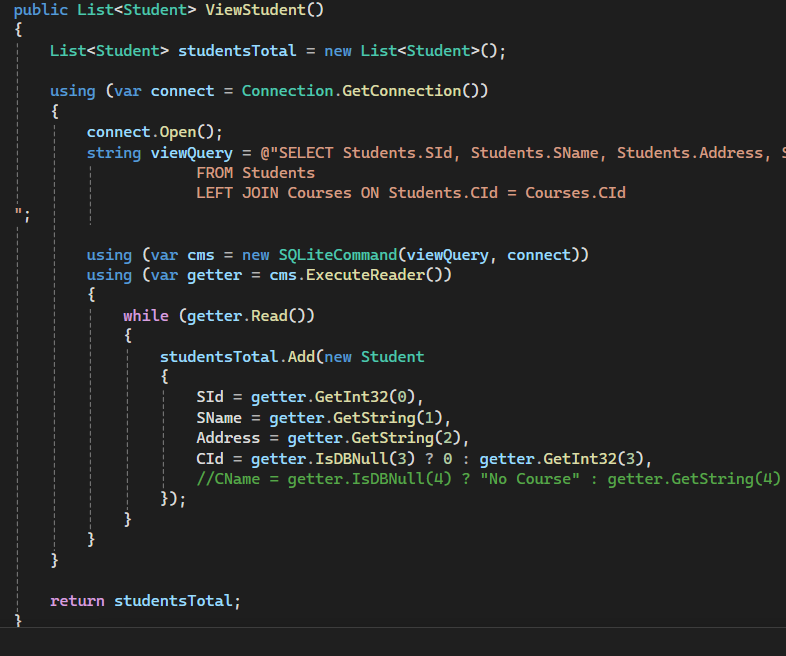
* **Students:** SId (PK), SName, Address, CId (FK)
* **Courses:** CId (PK), CName
* **Subjects:** SubId (PK), SubName, CId (FK)
* **Rooms:** RoomId (PK), RoomName, RoomType
* **Timetables:** TimetableId (PK), SubId (FK), LecturerId (FK), TimeSlot, RoomId (FK)
* **Exams:** ExamId (PK), ExamName, SId (FK), SubId (FK)
* **Marks:** MarkId (PK), SId (FK), ExamId (FK), SubId (FK), Score
* **Lecturers:** LecturerId (PK), LName, CId, SubId
* **Users:** UserId (PK), Username, Password, Role, SId (FK), LecturerId (FK), StaffId (FK)

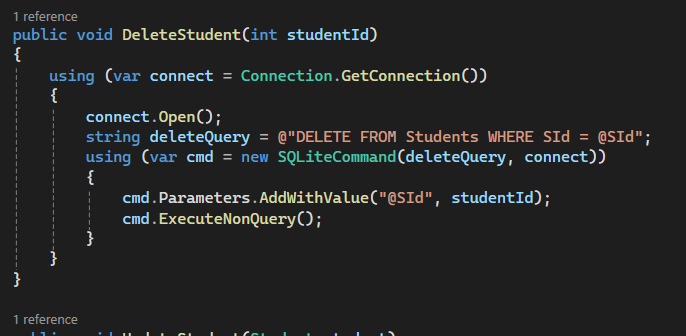
Screenshots of the Application

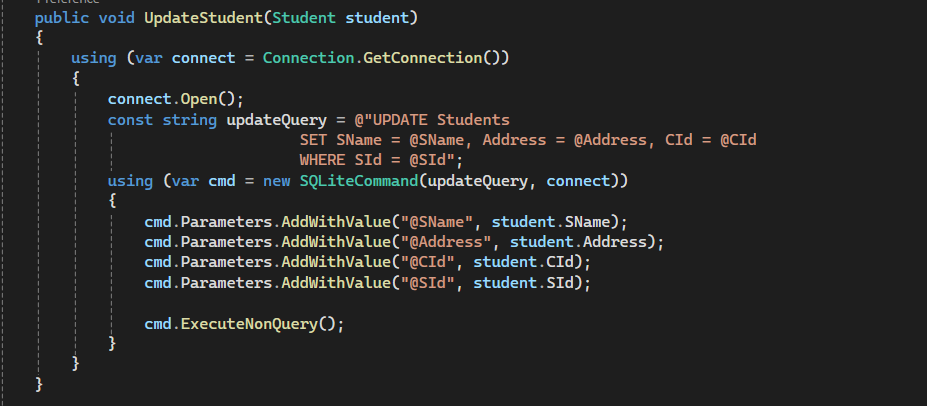
Screenshot 1: Student Management

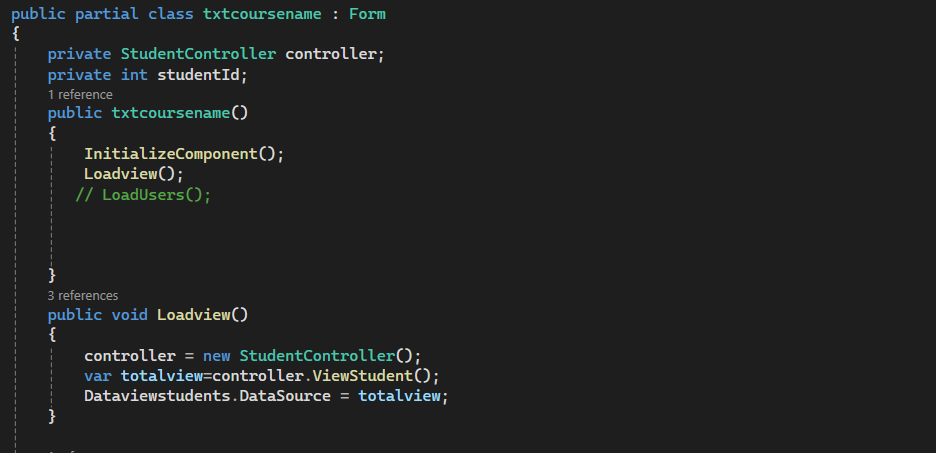


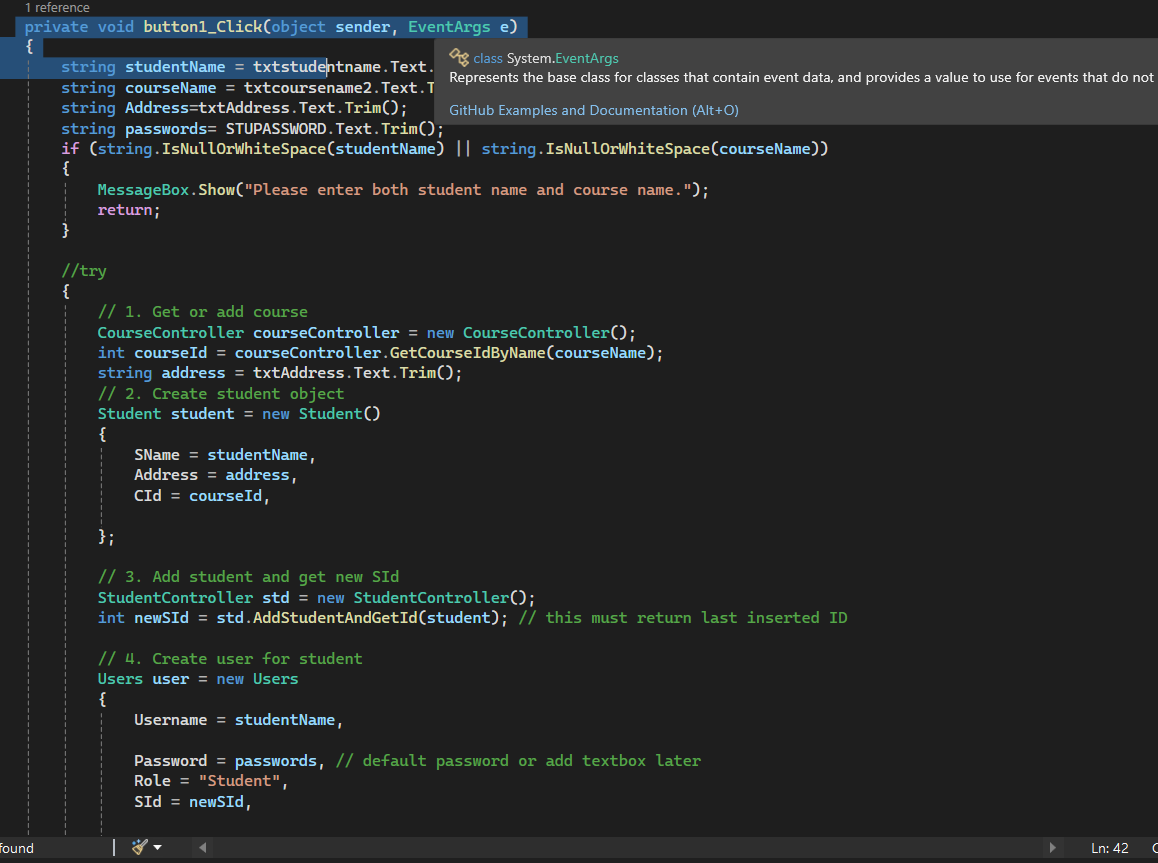


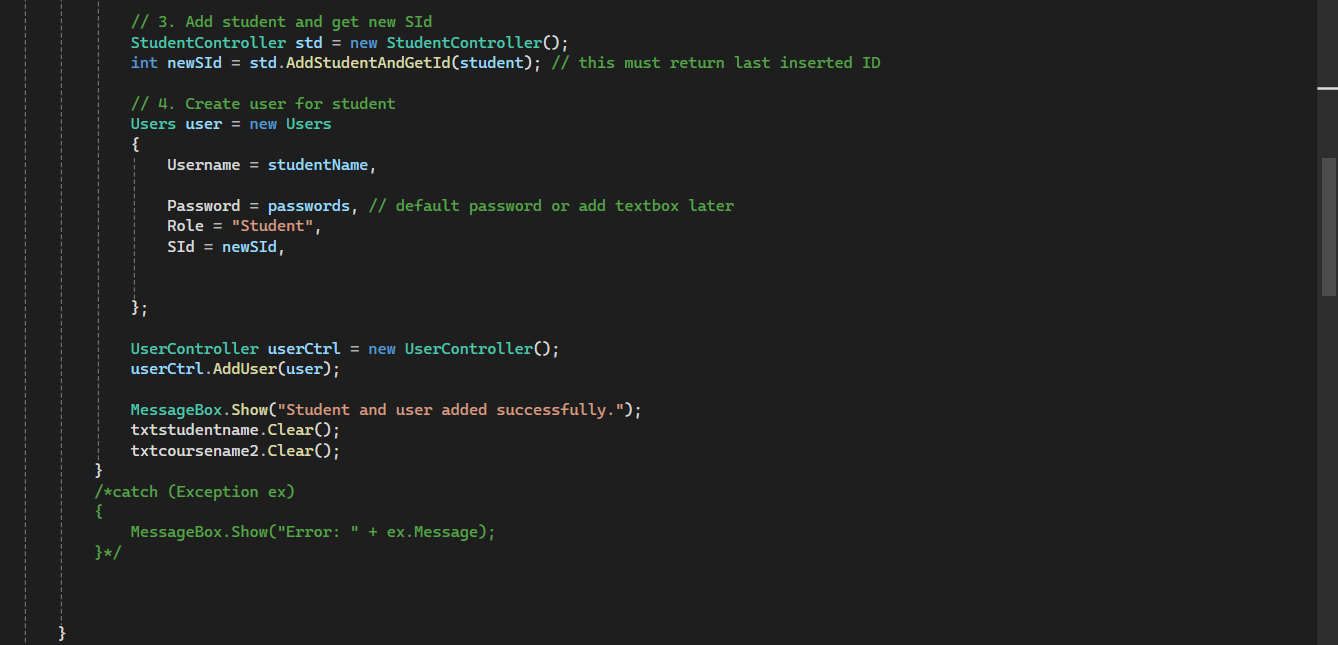


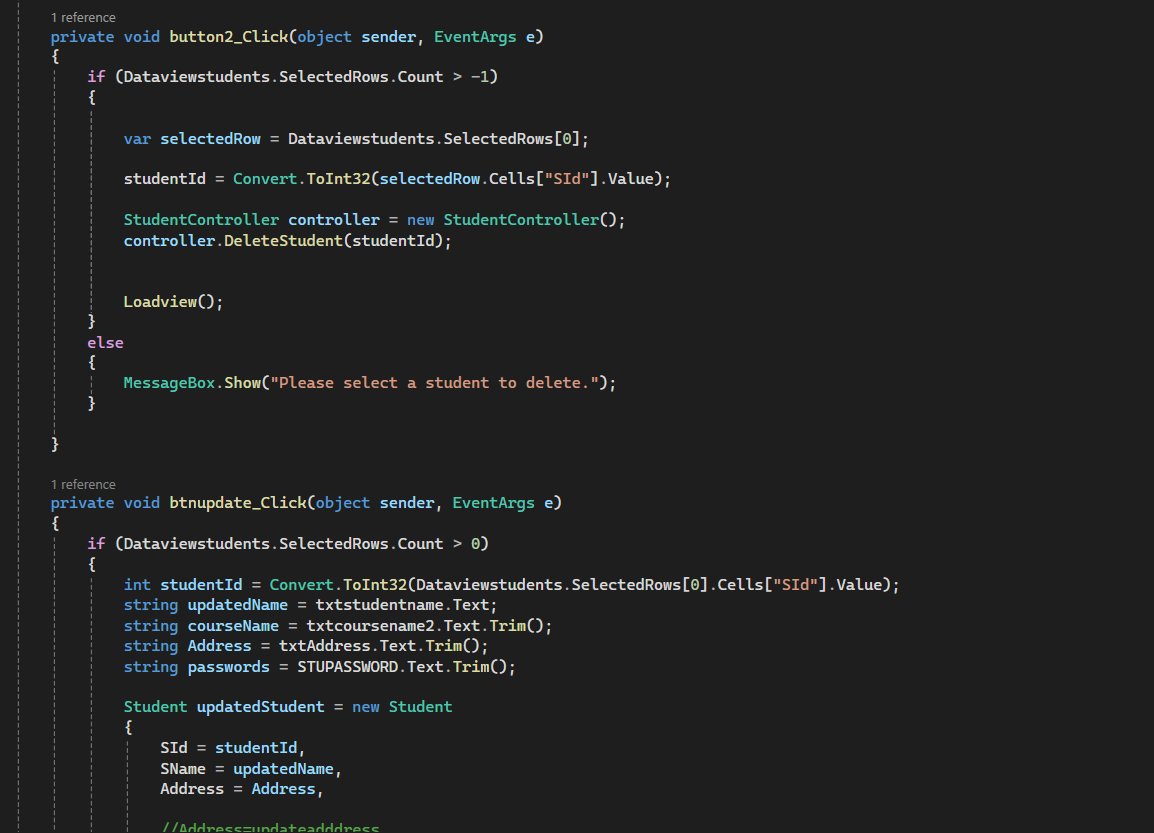


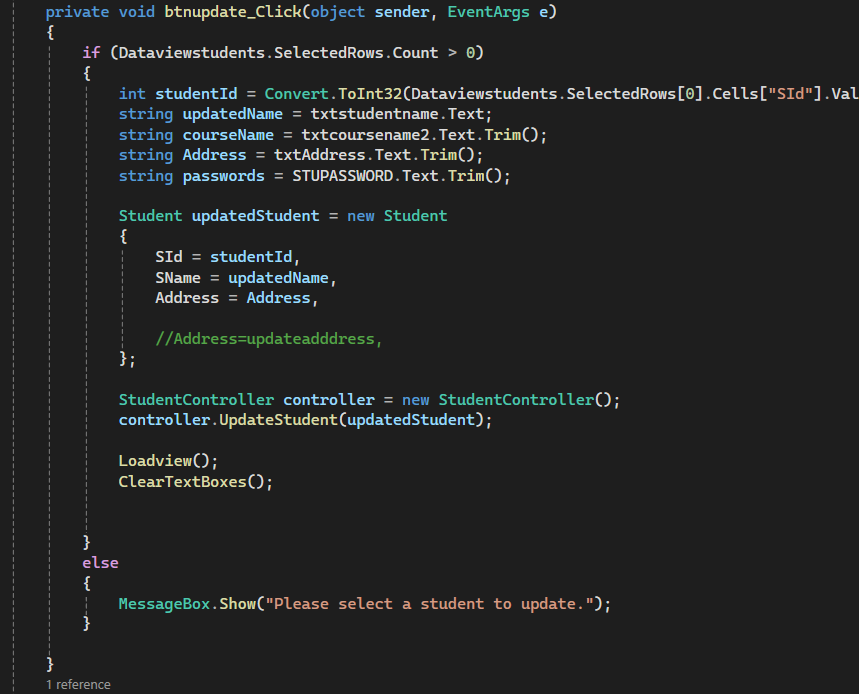


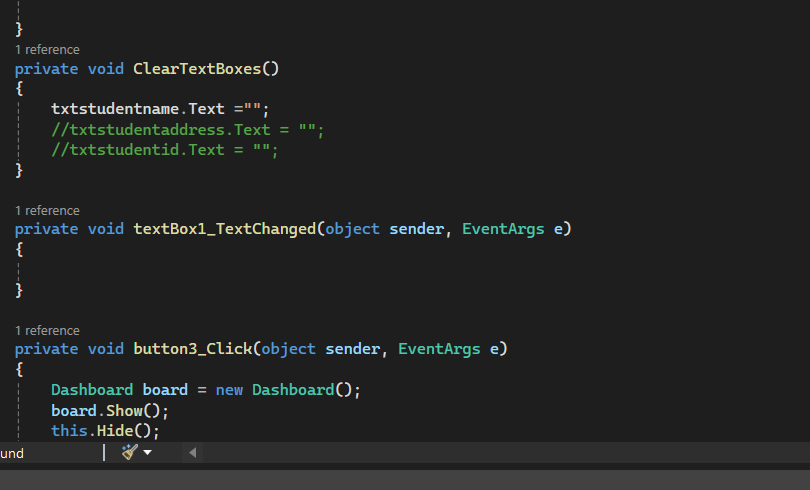












Other models same as its .

Screenshot 2: Course Management

Screenshot 3: Subject Management

Screenshot 4: Room Management

Screenshot 5: Timetable Management

Screenshot 6: Exam Management

Screenshot 7: Marks Management

Screenshot 8: Login Form

Screenshot 9: Dashboard Form

**9. Advantages of the System**

* Simplifies student and academic management processes.
* Reduces errors and data redundancy.
* Provides real-time updates and access to data.
* Supports multiple user roles for better security and management.
* User-friendly interface with clear navigation.

**10. Conclusion**

The Student Management System developed using C# and MVC pattern effectively automates the management of student information, courses, subjects, exams, and marks. It provides an efficient and reliable solution for educational institutions to maintain their academic data. This project enhances data integrity, reduces manual work, and offers a scalable platform for future improvements.