Unicom TIC Management System -

Project Submission Report

# Project Overview (Point-Wise)

## Key Features Implemented:

* Login System with role-based access for Admin, Staff, Students, and Lecturers.
* Course and Subject Management module.
* Student Management with course association.
* Exam and Marks Management with entry and viewing privileges by role.
* Timetable Management including Computer Lab and Lecture Hall allocation.
* Role-based dashboards that restrict access to appropriate features.
* SQLite database with relationships between tables.
* Error messages and input validation included.
* Simple and intuitive WinForms user interface.

## Technologies Used:

* Programming Language: C#
* Framework: WinForms (.NET Framework)
* Database: SQLite using System.Data.SQLite
* Architecture: MVC (Model–View–Controller)
* IDE: Visual Studio
* Design Patterns: MVC
* UI Elements: Buttons, ComboBoxes, DataGridViews, TextBoxes, Labels

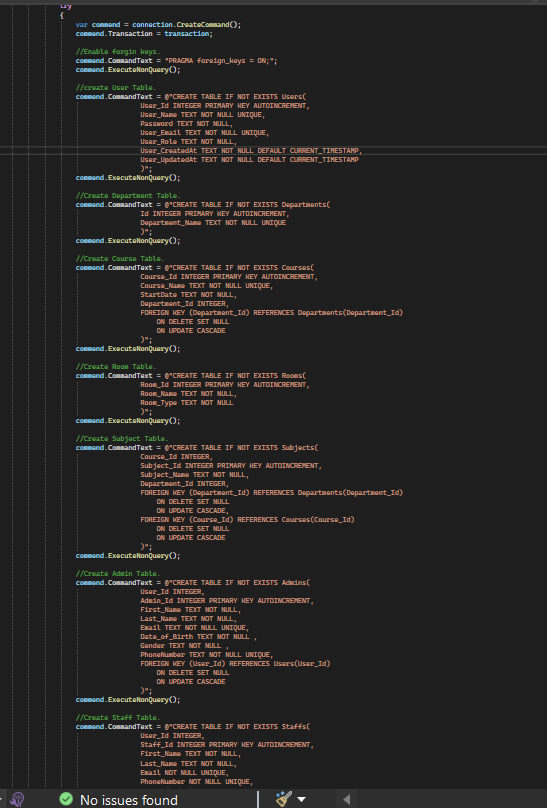
## Challenges Faced and Solutions:

* Role-Based Dashboard Visibility: Solved by checking user role after login and showing/hiding buttons accordingly.

* Room Allocation with Combo Box: Created a separate table for Rooms with room type filtering, and populated Combo Box dynamically.
* Login Validation: Added a simple user validation system querying the Users table and managing sessions based on roles.
* Database Table Creation: Implemented DatabaseManager.cs to auto-create tables if they don’t exist.
* Navigation Between Forms: Used controller logic to open/close forms based on user role without error.

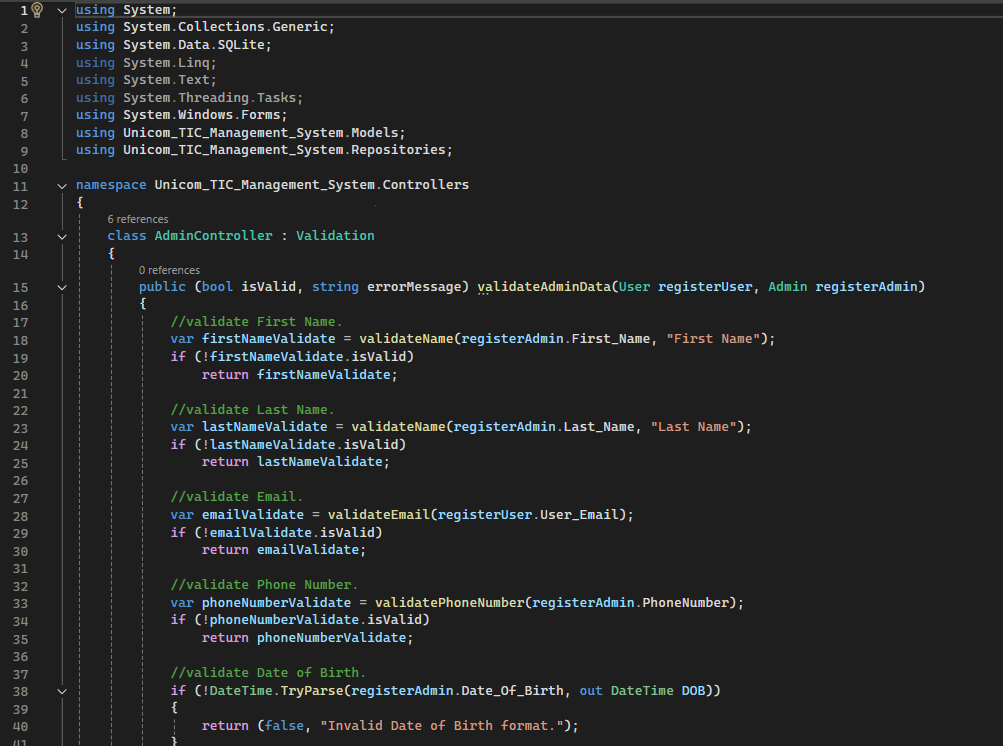
# Code Samples (Screenshots)

* Create table

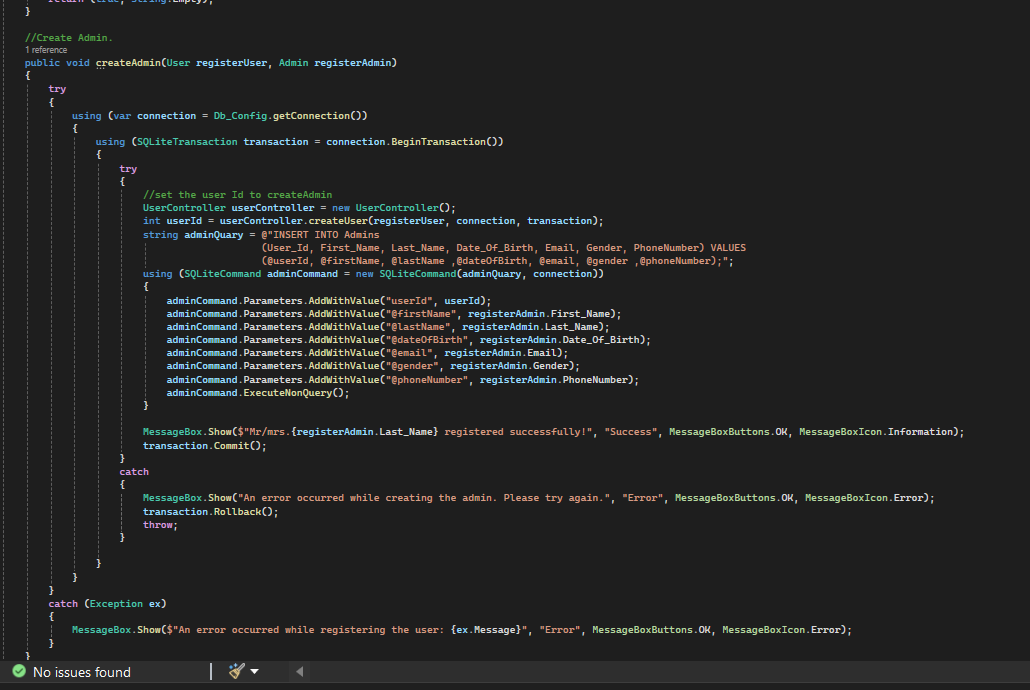


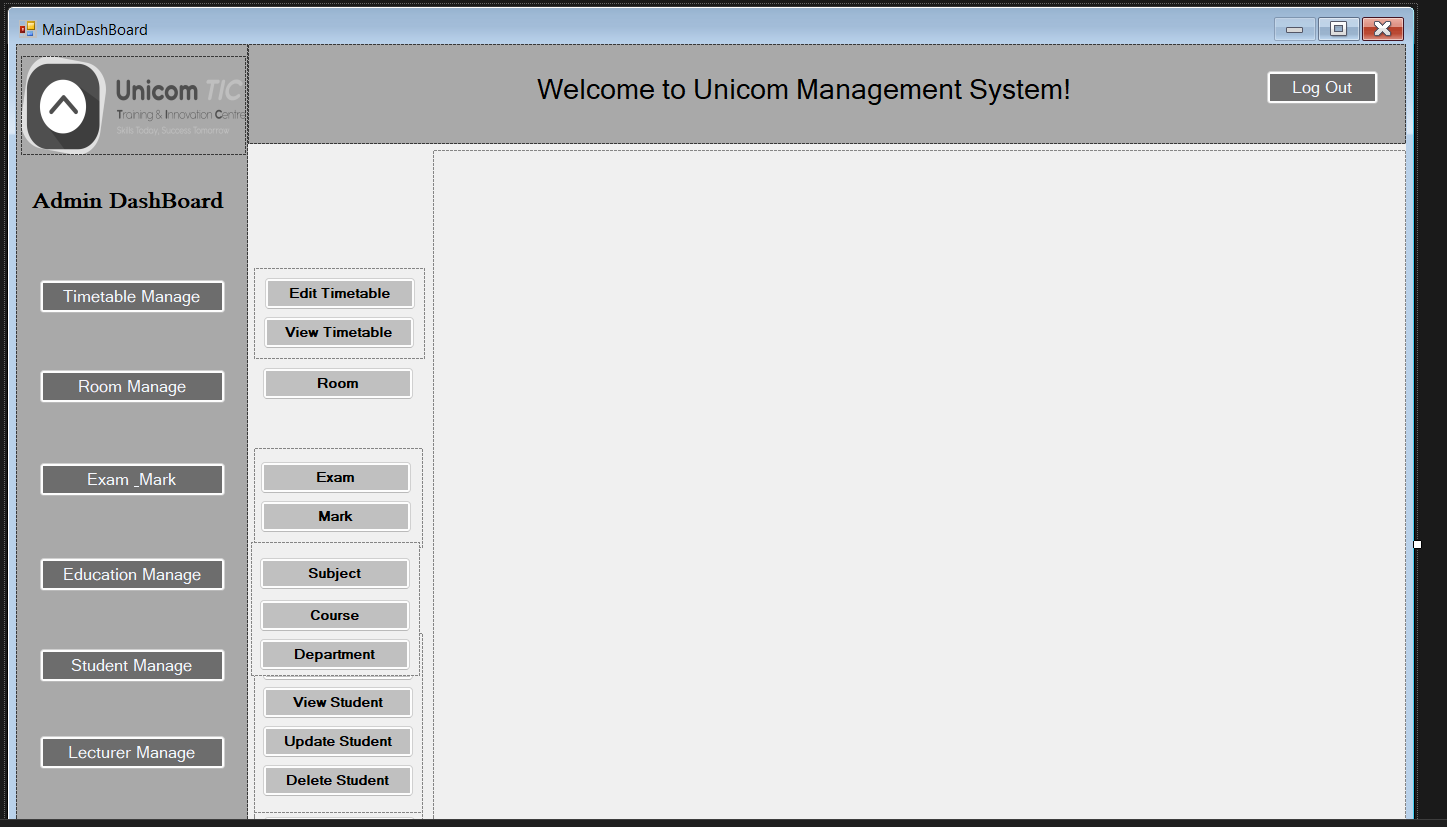


* Admin Controller Using inheritance



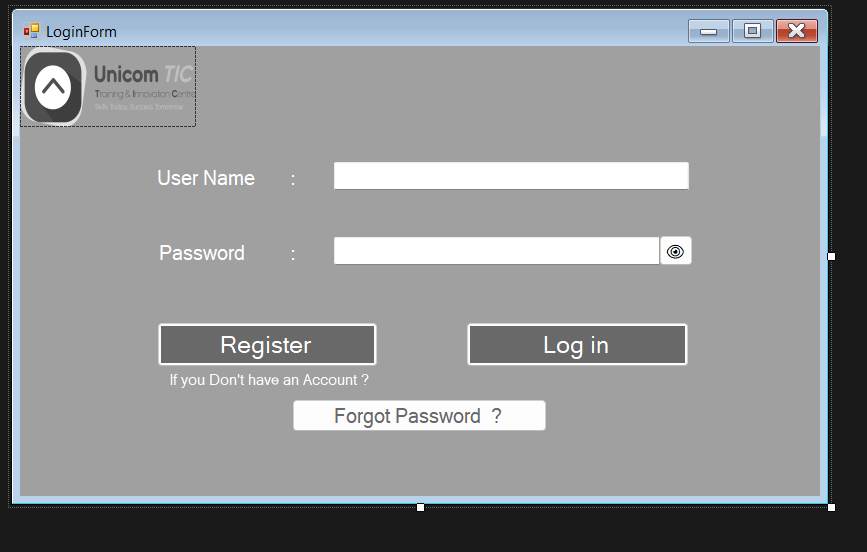
Using Error Handelling





Above main dash board

Login Form



Features

Filter (Courses, Subjects, Students, Exams, Rooms in dropdown list)

Error Messages (e.g., “Please select a room”, “Login failed”)

Input Validation (e.g., Score must be 0–100)

Descriptive Mappings for Viewing Data:

o Student name | Course name

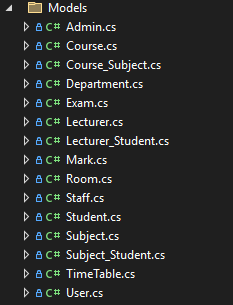
o Exam name | Subject name

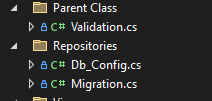
o Timetable – Subject name | Room name | Time slot

o Room name | Room type

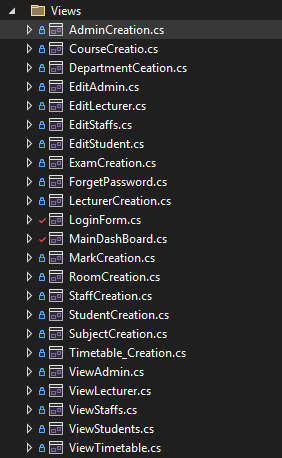
o Subject name | Course name

* Folder Structure

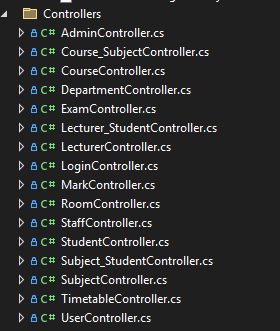




* Views



* Controllers



* Conclusion

This project helped in learning:

• How to implement an MVC architecture in a desktop application.

• SQLite CRUD operations in C#.

• Role-based access control with a simple UI.

• Structuring a school management system from scratch.