Unicom TIC Management System (UMS)

Developed Using: C#, WinForms, SQLite **Submitted by:** A. Annet **UT Number:** UT010188

1. Introduction

The Unicom TIC Management System (UMS) is a comprehensive desktop application designed to streamline daily academic operations within an educational institution. This project serves as a foundational learning experience, aiming to provide a practical understanding of essential concepts in C#, Windows Forms, the Model-View-Controller (MVC) architectural pattern, SQLite database management, and role-based access control systems.

UMS offers an efficient solution for managing core school operations, including the handling of courses, subjects, students, exams, marks, and timetables.

2. Acknowledgements

I would like to extend my sincere gratitude to Unicom TIC, my lecturers, mentors, and friends for their invaluable guidance and unwavering encouragement throughout the development of this project. I also appreciate the steadfast support from my peers and the rich resources provided by Microsoft and SQLite documentation, which were instrumental in aiding the development process.

3. Project Objectives

The primary objectives of the Unicom TIC Management System project were:

- To develop a **user-friendly and responsive desktop application** using C# WinForms.
- To comprehend and effectively apply the **MVC design pattern** for enhanced organization and maintainability of application logic.
- To implement robust **Create, Read, Update, and Delete (CRUD) operations** for data management using SQLite.
- To establish a **role-based login access system** catering to Admins, Staff, Lecturers, and Students.
- To facilitate basic scheduling functionalities through **timetable management,** including room (lab/hall) allocation.
- To demonstrate effective **form design and data-binding techniques**, incorporating proper validation and error handling mechanisms.

---

4. Key Features

4.1 Role-Based Login System

A secure login system with distinct roles ensures tailored access to the application's functionalities:

- **Admin:** Possesses full access, enabling the addition, editing, and deletion of all data, including courses, students, exams, marks, timetables, and attendance.
- **Staff:** Can manage exams and marks, and view timetables.
- **Lecturer:** Can view timetables and manage (mark/edit) student attendance and marks.
- **Student:** Can view their personal profile, timetable, marks, and attendance records.

Role-based dashboards dynamically adjust, displaying or hiding features based on the logged-in user's access level.

4.2 Room Allocation in Timetable

The system allows for efficient room management:

- **Admins** can assign a specific room (lab or hall) when creating or modifying a timetable entry.
- Rooms are meticulously stored in a dedicated database table, categorized by type (Lab, Hall).
- A **ComboBox** is utilized for convenient room selection.

4.3 Form Navigation and UI Management

The application employs dynamic form handling for a seamless user experience:

- Upon successful login, the LoginForm is hidden, and the MainForm is displayed.
- Forms are shown or hidden dynamically using the Show() and Hide() methods, ensuring smooth transitions.
- Upon logout, the current form closes, and the LoginForm is re-opened.

4.4 Pop-up Messages

Important alerts and confirmations are delivered via MessageBox.Show(), providing clear feedback to the user:

- Login success/failure notifications.

- Data saving or update confirmations (e.g., "Timetable saved!").
- Validation errors (e.g., "Please select a room").

4.5 Error Logging

To aid in debugging and system maintenance, errors are systematically logged:

- Failures such as unsuccessful logins, database issues, or form validation problems are recorded in a text file named **errorlog.txt**.

5. Technical Details

Feature	Technology Used
GUI	Windows Forms (WinForms)
Language	C# (.NET Framework)
Database	SQLite (Local database)
Database Driver	System.Data.SQLite via NuGet
Architecture	MVC (Model-View-Controller)
UI Elements	DataGridView, ComboBox, Buttons, TextBoxes
Data Storage	.db file for SQLite

Export to Sheets

6. Budget Plan

The Unicom TIC Management System was developed with a zero-cost budget, leveraging open-source and free community tools:

Item	Estimated Cost (USD)	Notes
Visual Studio Community	$0	Free version used
SQLite	$0	Open-source
System.Data.SQLite Library	$0	Installed via NuGet

Hardware (Laptop/PC)	$0	Personal use, no additional cost
Total	**$0**	No direct costs incurred for development

Export to Sheets

7. Design

(This section would typically include visual representations of the Use Case Diagram and ER Diagram. Since I cannot generate images, please ensure these diagrams are included in the final report.)

- **Use Case Diagram:** Illustrates the interactions between users and the system's functionalities.
- **ER Diagram:** Depicts the relationships between different entities within the database.

---

8. References

The development of UMS was guided by the following resources:

- Microsoft Docs
- SQLite Official Site
- C# Tutorials: Tamil Programmer C#
- System.Data.SQLite: https://system.data.sqlite.org/
- YouTube Tutorials on WinForms & MVC
- Stack Overflow (for debugging and best practices)
- ChatGPT (for assistance with problem-solving)

The project's folder structure adheres to best practices, separating concerns into controllers, services, repositories, DTOs (Data Transfer Objects), Enums, and Mappers.

9. Challenges and Solutions

Throughout the development process, several challenges were encountered and successfully overcome:

Challenge	Solution
Incorrect table queries	Each form was run individually, and errors or missing parts were meticulously noted.
Runtime errors	Forms were corrected systematically based on identified errors, followed by verification of data persistence in the database.
Role-based login implementation	Focused debugging and careful implementation of authorization logic.
Exam Marks management complexity	Addressed through iterative development and testing of CRUD operations for marks.
Timetable management (time slot logic)	Refined the logic for time slot allocation and room assignment to ensure accuracy and prevent conflicts.
Data not saving in DataGridView	Investigated data binding issues and ensured proper data saving mechanisms were in place.
Repository (Database Manager) design	Optimized the repository pattern to ensure efficient and consistent database interactions, abstracting data access logic.

Export to Sheets

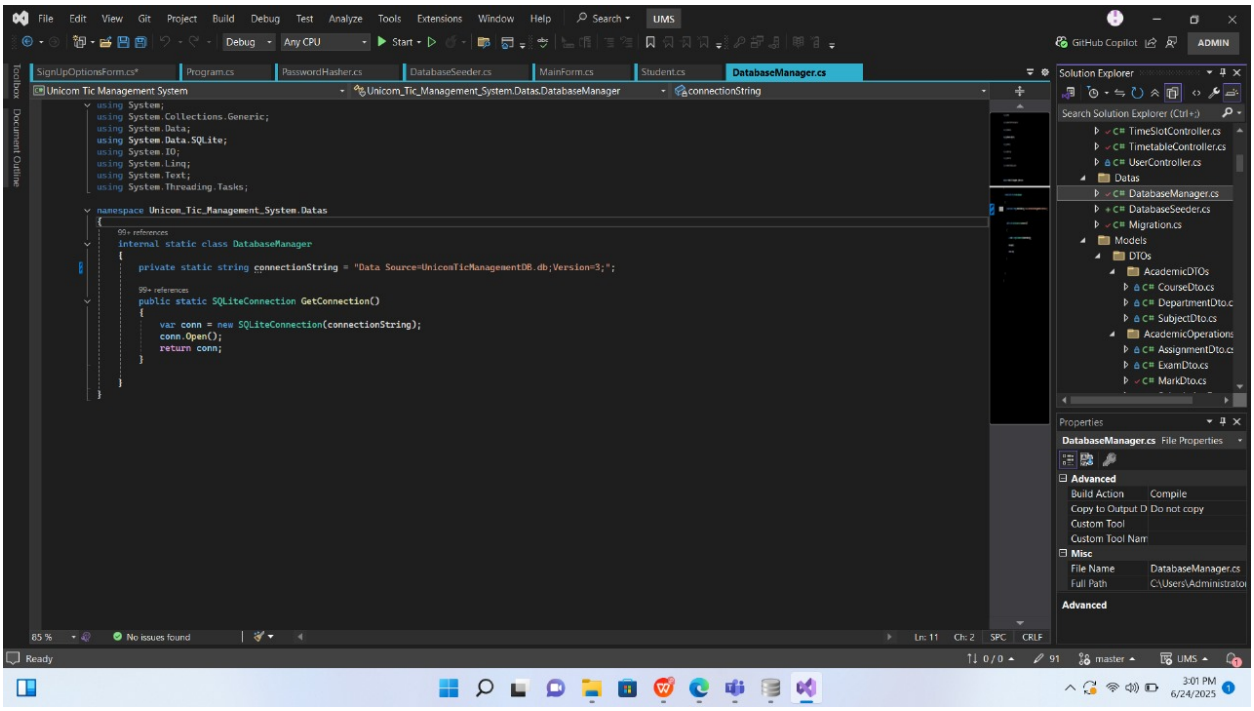
Assistance from online resources like ChatGPT also proved valuable in resolving complex issues.

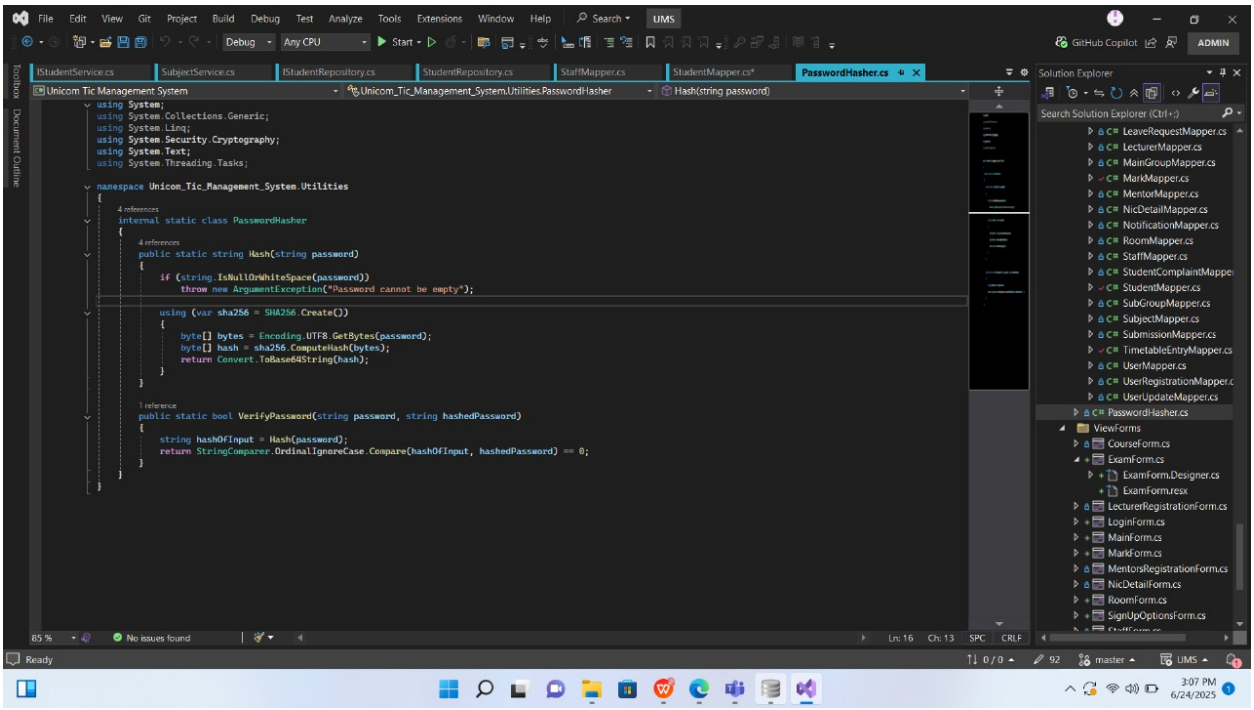
---

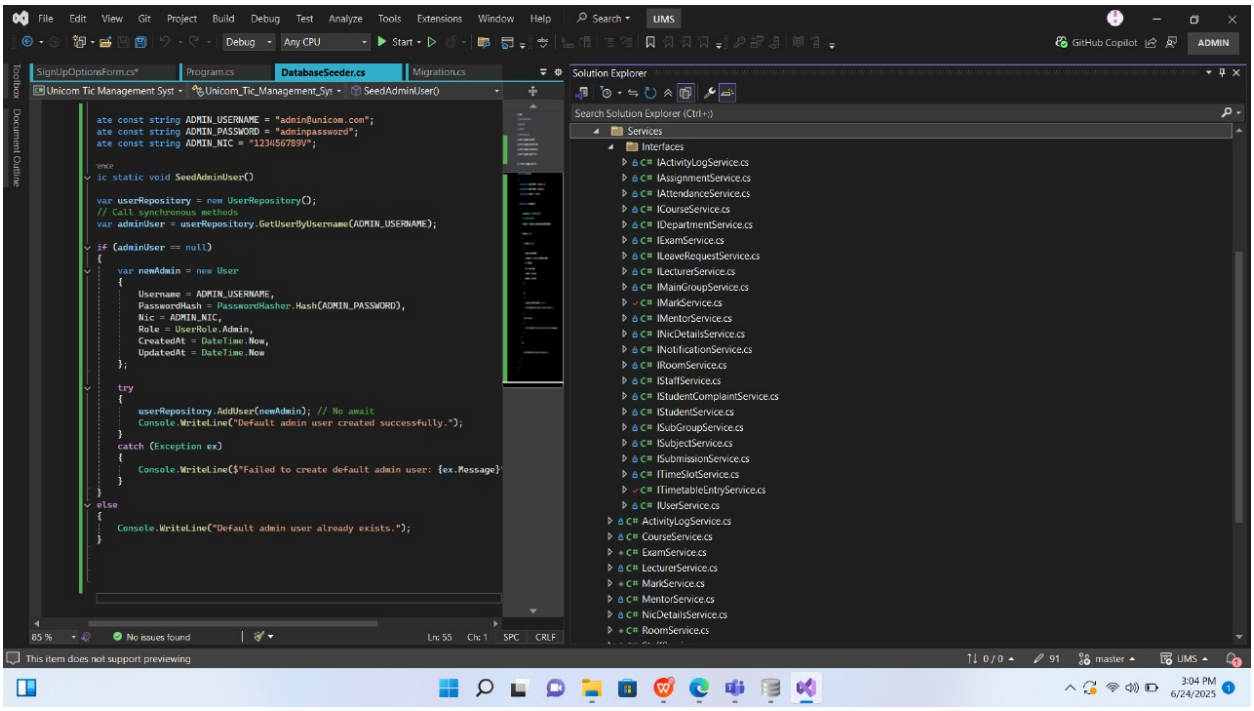
10. Code Samples (Screenshots)

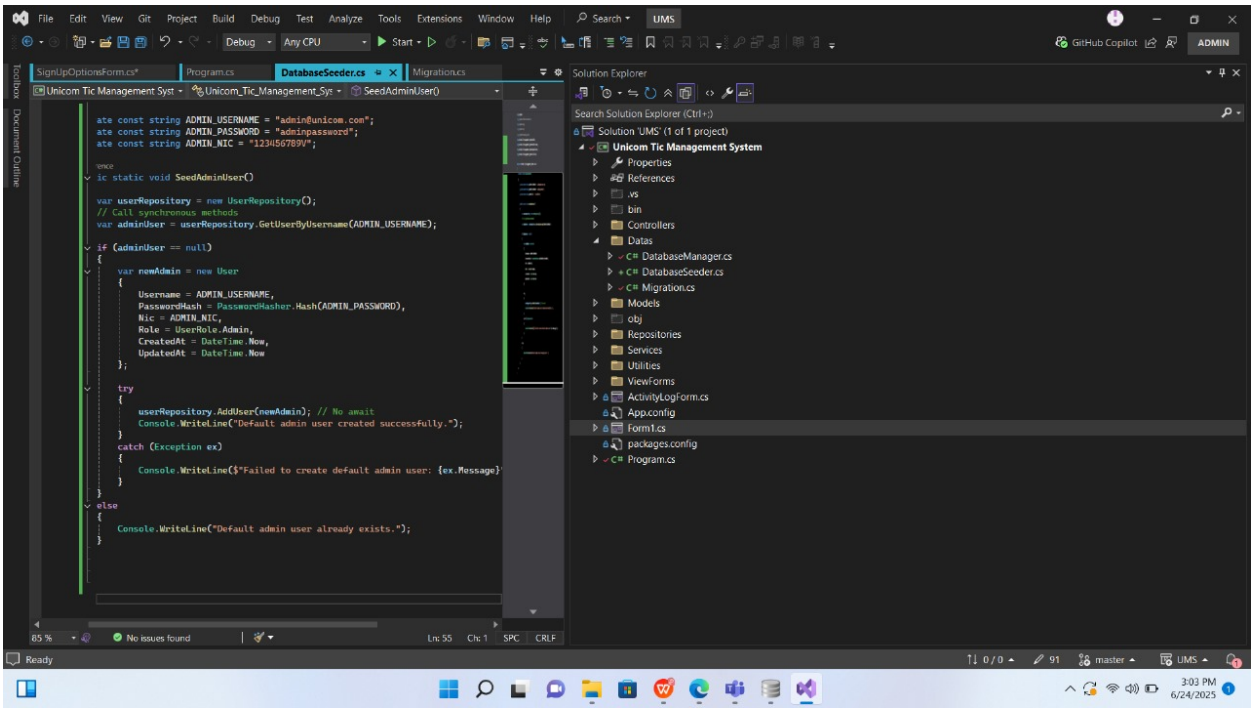
(This section would include screenshots of key code implementations. Please ensure these images are included in the final report.)

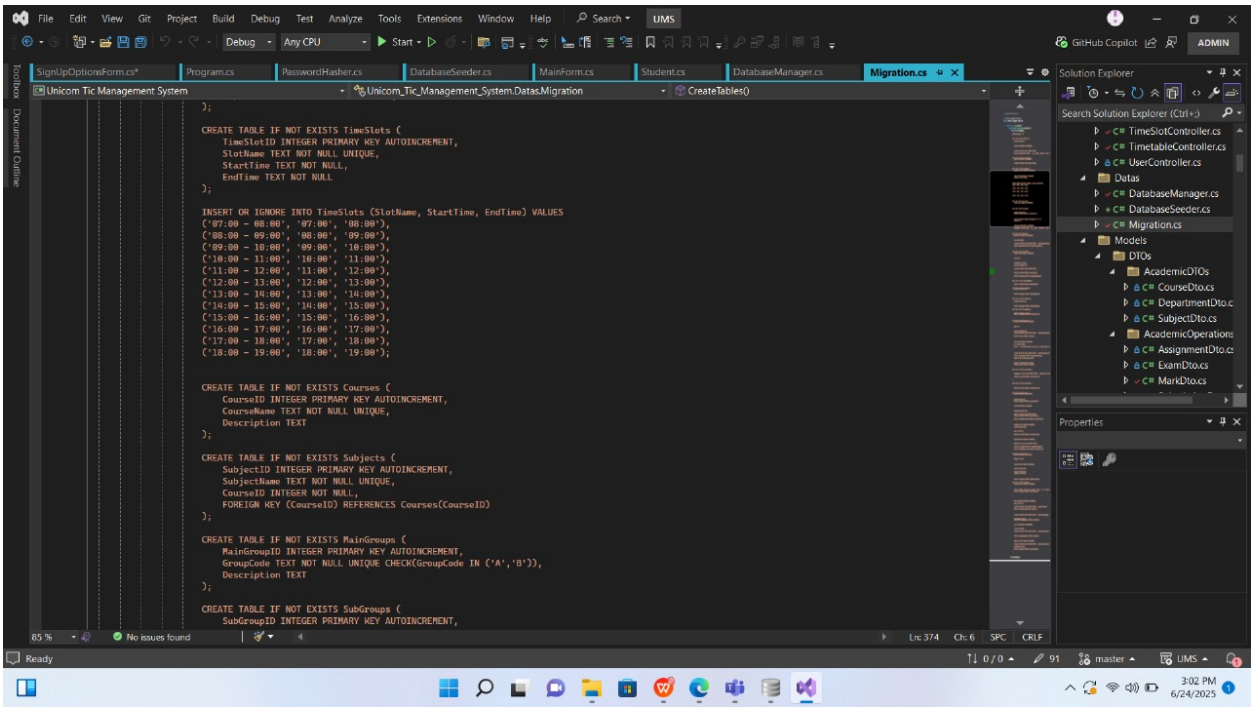
- Loading enum items into ComboBoxes and dynamically inserting forms into panels.
- Screenshot of the Login and Main Forms during runtime.
- Examples of CRUD (Create, Read, Update, Delete) function implementations for various entities (staff, courses, students, lecturers, subjects, etc.).
- Code demonstrating the conversion between entities and DTOs (Data Transfer Objects).
- Illustrations of proper interface usage, including the declaration of method signatures.

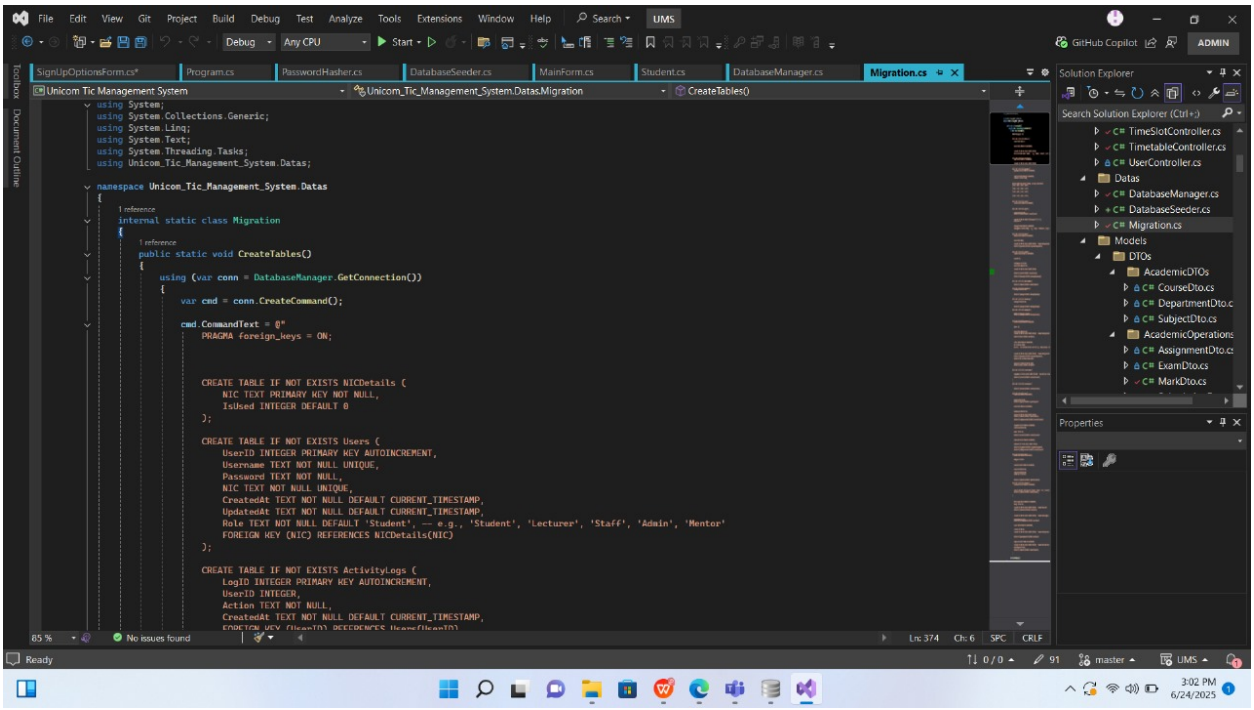

SignUpOptionsForm.cs*
Program.cs
PasswordHasher.cs
DatabaseSeeder.cs
MainForm.cs
Student.cs
DatabaseManager.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Datas.DatabaseManager
connectionString
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SQLite;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Unicom_Tic_Management_System.Datas
{
99+ references
internal static class DatabaseManager
{
private static string connectionString = "Data Source=UnicomTicManagementDB.db;Version=3;";
99+ references
public static SQLiteConnection GetConnection()
{
var conn = new SQLiteConnection(connectionString);
conn.Open();
return conn;
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
TimeSlotController.cs
TimetableController.cs
UserController.cs
Datas
DatabaseManager.cs
DatabaseSeeder.cs
Migration.cs
Models
DTOs
AcademicDTOs
CourseDto.cs
DepartmentDto.c
SubjectDto.cs
AcademicOperations
AssignmentDto.cs
ExamDto.cs
MarkDto.cs
Properties
DatabaseManager.cs File Properties
Advanced
Build Action Compile
Copy to Output D Do not copy
Custom Tool
Custom Tool Nam
Misc
File Name DatabaseManager.cs
Full Path C:\Users\Administrator
Advanced
85 %
No issues found
Ln: 11 Ch: 2 SPC CRLF
Ready
master
UMS
3:01 PM
6/24/2025

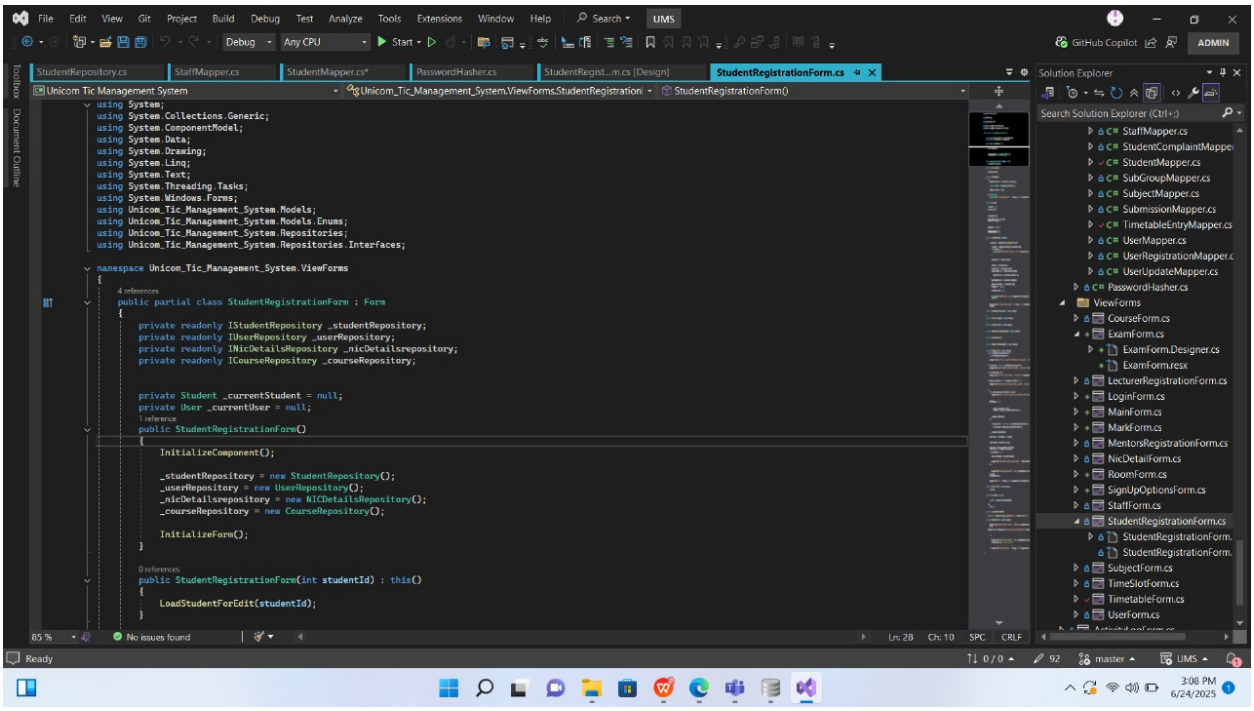

IStudentService.cs
SubjectService.cs
IStudentRepository.cs
StudentRepository.cs
StaffMapper.cs
StudentMapper.cs*
PasswordHasher.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Utilities.PasswordHasher
Hash(string password)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Threading.Tasks;
namespace Unicom_Tic_Management_System.Utilities
{
4 references
internal static class PasswordHasher
{
4 references
public static string Hash(string password)
{
if (string.IsNullOrWhiteSpace(password))
throw new ArgumentException("Password cannot be empty");
using (var sha256 = SHA256.Create())
{
byte[] bytes = Encoding.UTF8.GetBytes(password);
byte[] hash = sha256.ComputeHash(bytes);
return Convert.ToBase64String(hash);
}
}
1 reference
public static bool VerifyPassword(string password, string hashedPassword)
{
string hashOfInput = Hash(password);
return StringComparer.OrdinalIgnoreCase.Compare(hashOfInput, hashedPassword) == 0;
}
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
LeaveRequestMapper.cs
LecturerMapper.cs
MainGroupMapper.cs
MarkMapper.cs
MentorMapper.cs
NicDetailMapper.cs
NotificationMapper.cs
RoomMapper.cs
StaffMapper.cs
StudentComplaintMapper
StudentMapper.cs
SubGroupMapper.cs
SubjectMapper.cs
SubmissionMapper.cs
TimetableEntryMapper.cs
UserMapper.cs
UserRegistrationMapper.c
UserUpdateMapper.cs
PasswordHasher.cs
ViewForms
CourseForm.cs
ExamForm.cs
ExamForm.Designer.cs
ExamForm.resx
LecturerRegistrationForm.cs
LoginForm.cs
MainForm.cs
MarkForm.cs
MentorsRegistrationForm.cs
NicDetailForm.cs
RoomForm.cs
SignUpOptionsForm.cs
85 %
No issues found
Ln: 16 Ch: 13 SPC CRLF
Ready
master
UMS
3:07 PM
6/24/2025

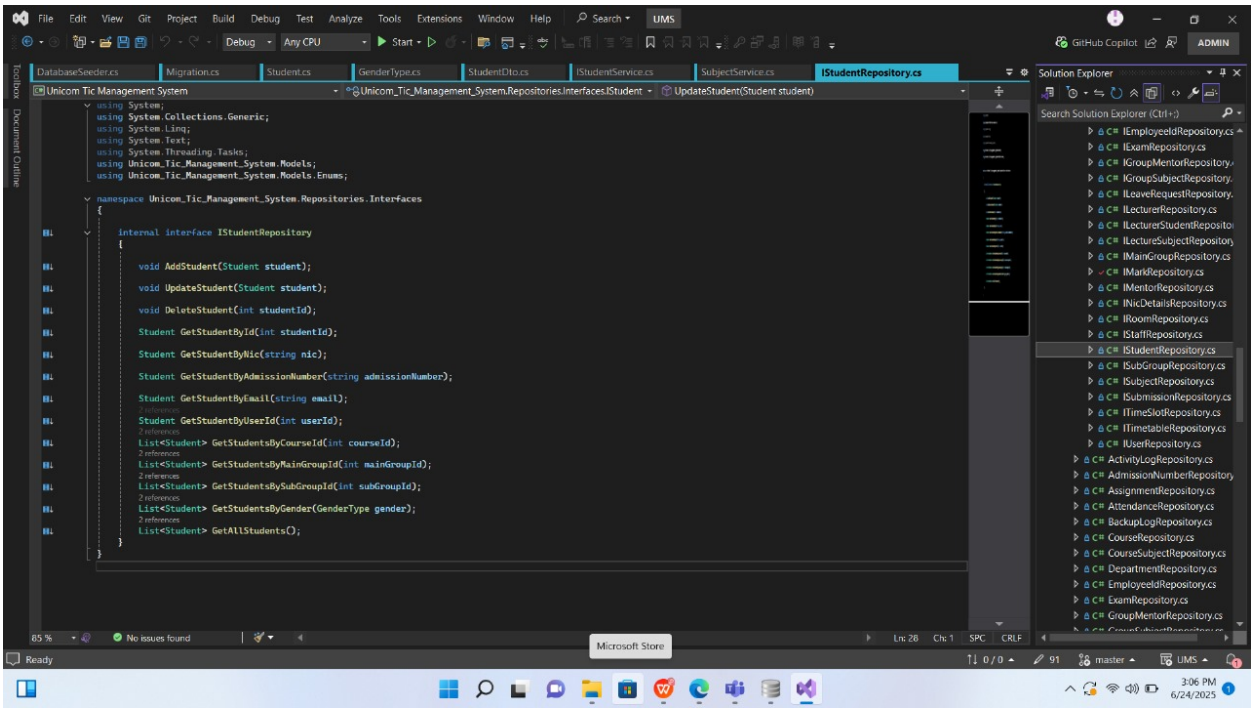

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
SignUpOptionsForm.cs*
Program.cs
DatabaseSeeder.cs
Migration.cs
Unicom Tic Management Syst
Unicom_Tic_Management_Syst
SeedAdminUser()
ate const string ADMIN_USERNAME = "admin@unicom.com";
ate const string ADMIN_PASSWORD = "adminpassword";
ate const string ADMIN_NIC = "1234567890V";
ence
ic static void SeedAdminUser()
var userRepository = new UserRepository();
// Call synchronous methods
var adminUser = userRepository.GetUserByUsername(ADMIN_USERNAME);
if (adminUser == null)
{
var newAdmin = new User
{
Username = ADMIN_USERNAME,
PasswordHash = PasswordHasher.Hash(ADMIN_PASSWORD),
Nic = ADMIN_NIC,
Role = UserRole.Admin,
CreatedAt = DateTime.Now,
UpdatedAt = DateTime.Now
};
try
{
userRepository.AddUser(newAdmin); // No await
Console.WriteLine("Default admin user created successfully.");
}
catch (Exception ex)
{
Console.WriteLine($"Failed to create default admin user: {ex.Message}
}
}
else
{
Console.WriteLine("Default admin user already exists.");
}
85 %
No issues found
Ln: 55
Ch: 1
SPC
CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Services
Interfaces
IActivityLogService.cs
IAssignmentService.cs
IAttendanceService.cs
ICourseService.cs
IDepartmentService.cs
IExamService.cs
ILeaveRequestService.cs
ILecturerService.cs
IMainGroupService.cs
IMarkService.cs
IMentorService.cs
INicDetailsService.cs
INotificationService.cs
IRoomService.cs
IStaffService.cs
IStudentComplaintService.cs
IStudentService.cs
ISubGroupService.cs
ISubjectService.cs
ISubmissionService.cs
ITimeSlotService.cs
ITimetableEntryService.cs
IUserService.cs
ActivityLogService.cs
CourseService.cs
ExamService.cs
LecturerService.cs
MarkService.cs
MentorService.cs
NicDetailsService.cs
RoomService.cs
This item does not support previewing
master
UMS
3:04 PM
6/24/2025

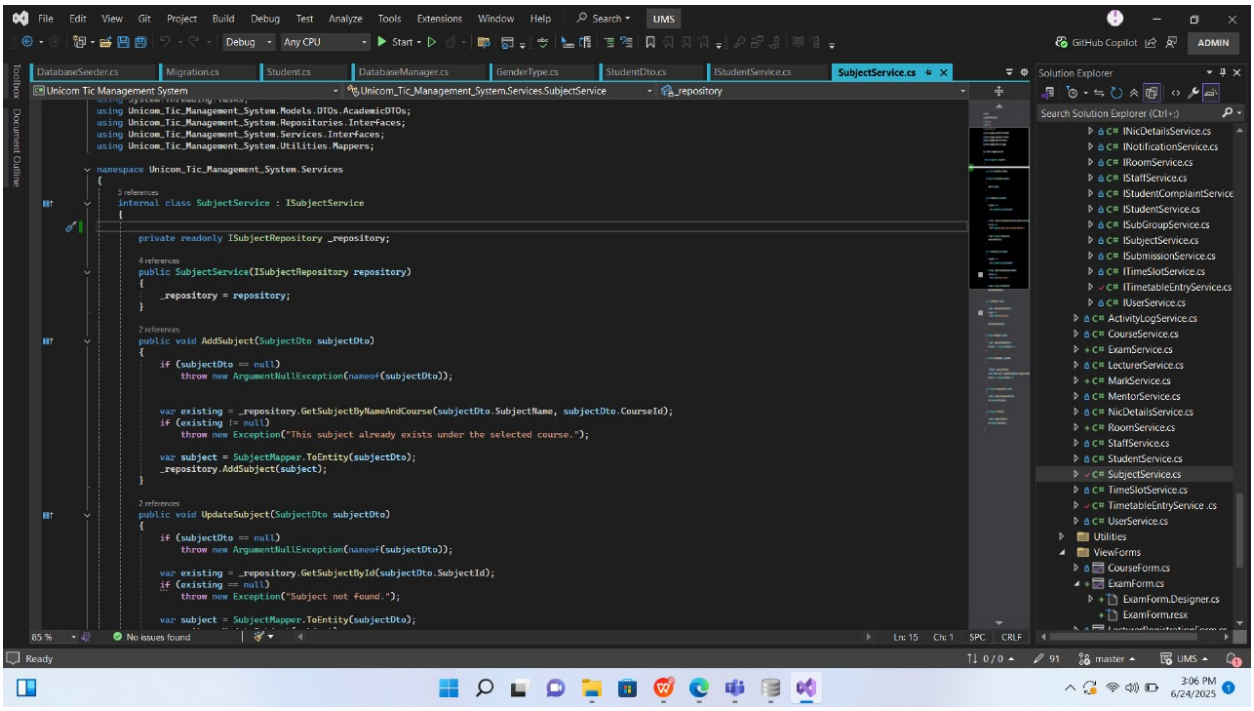

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
SignUpOptionsForm.cs*
Program.cs
DatabaseSeeder.cs
Migration.cs
Unicom Tic Management Syst
Unicom_Tic_Management_Syst
SeedAdminUser()
ate const string ADMIN_USERNAME = "admin@unicom.com";
ate const string ADMIN_PASSWORD = "adminpassword";
ate const string ADMIN_NIC = "1234567890V";
ence
ic static void SeedAdminUser()
var userRepository = new UserRepository();
// Call synchronous methods
var adminUser = userRepository.GetUserByUsername(ADMIN_USERNAME);
if (adminUser == null)
{
var newAdmin = new User
{
Username = ADMIN_USERNAME,
PasswordHash = PasswordHasher.Hash(ADMIN_PASSWORD),
Nic = ADMIN_NIC,
Role = UserRole.Admin,
CreatedAt = DateTime.Now,
UpdatedAt = DateTime.Now
};
try
{
userRepository.AddUser(newAdmin); // No await
Console.WriteLine("Default admin user created successfully.");
}
catch (Exception ex)
{
Console.WriteLine($"Failed to create default admin user: {ex.Message}
}
}
else
{
Console.WriteLine("Default admin user already exists.");
}
85 %
No issues found
Ln: 55
Ch: 1
SPC
CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'UMS' (1 of 1 project)
Unicom Tic Management System
Properties
References
.vs
bin
Controllers
Datas
DatabaseManager.cs
DatabaseSeeder.cs
Migration.cs
Models
obj
Repositories
Services
Utilities
ViewForms
ActivityLogForm.cs
App.config
Form1.cs
packages.config
Program.cs
Ready
master
UMS
3:03 PM
6/24/2025


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
SignUpOptionsForm.cs*
Program.cs
PasswordHasher.cs
DatabaseSeeder.cs
MainForm.cs
Student.cs
DatabaseManager.cs
Migration.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Datas.Migration
CreateTables()
);
CREATE TABLE IF NOT EXISTS TimeSlots (
TimeSlotID INTEGER PRIMARY KEY AUTOINCREMENT,
SlotName TEXT NOT NULL UNIQUE,
StartTime TEXT NOT NULL,
EndTime TEXT NOT NULL
);
INSERT OR IGNORE INTO TimeSlots (SlotName, StartTime, EndTime) VALUES
('07:00 - 08:00', '07:00', '08:00'),
('08:00 - 09:00', '08:00', '09:00'),
('09:00 - 10:00', '09:00', '10:00'),
('10:00 - 11:00', '10:00', '11:00'),
('11:00 - 12:00', '11:00', '12:00'),
('12:00 - 13:00', '12:00', '13:00'),
('13:00 - 14:00', '13:00', '14:00'),
('14:00 - 15:00', '14:00', '15:00'),
('15:00 - 16:00', '15:00', '16:00'),
('16:00 - 17:00', '16:00', '17:00'),
('17:00 - 18:00', '17:00', '18:00'),
('18:00 - 19:00', '18:00', '19:00');
CREATE TABLE IF NOT EXISTS Courses (
CourseID INTEGER PRIMARY KEY AUTOINCREMENT,
CourseName TEXT NOT NULL UNIQUE,
Description TEXT
);
CREATE TABLE IF NOT EXISTS Subjects (
SubjectID INTEGER PRIMARY KEY AUTOINCREMENT,
SubjectName TEXT NOT NULL UNIQUE,
CourseID INTEGER NOT NULL,
FOREIGN KEY (CourseID) REFERENCES Courses(CourseID)
);
CREATE TABLE IF NOT EXISTS MainGroups (
MainGroupID INTEGER PRIMARY KEY AUTOINCREMENT,
GroupCode TEXT NOT NULL UNIQUE CHECK(GroupCode IN ('A','B')),
Description TEXT
);
CREATE TABLE IF NOT EXISTS SubGroups (
SubGroupID INTEGER PRIMARY KEY AUTOINCREMENT,
Solution Explorer
Search Solution Explorer (Ctrl+;)
TimeSlotController.cs
TimetableController.cs
UserController.cs
Datas
DatabaseManager.cs
DatabaseSeeder.cs
Migration.cs
Models
DTOs
AcademicDTOs
CourseDto.cs
DepartmentDto.c
SubjectDto.cs
AcademicOperations
AssignmentDto.cs
ExamDto.cs
MarkDto.cs
Properties
85 %
No issues found
Ln: 374
Ch: 6
SPC
CRLF
Ready
master
UMS
3:02 PM
6/24/2025

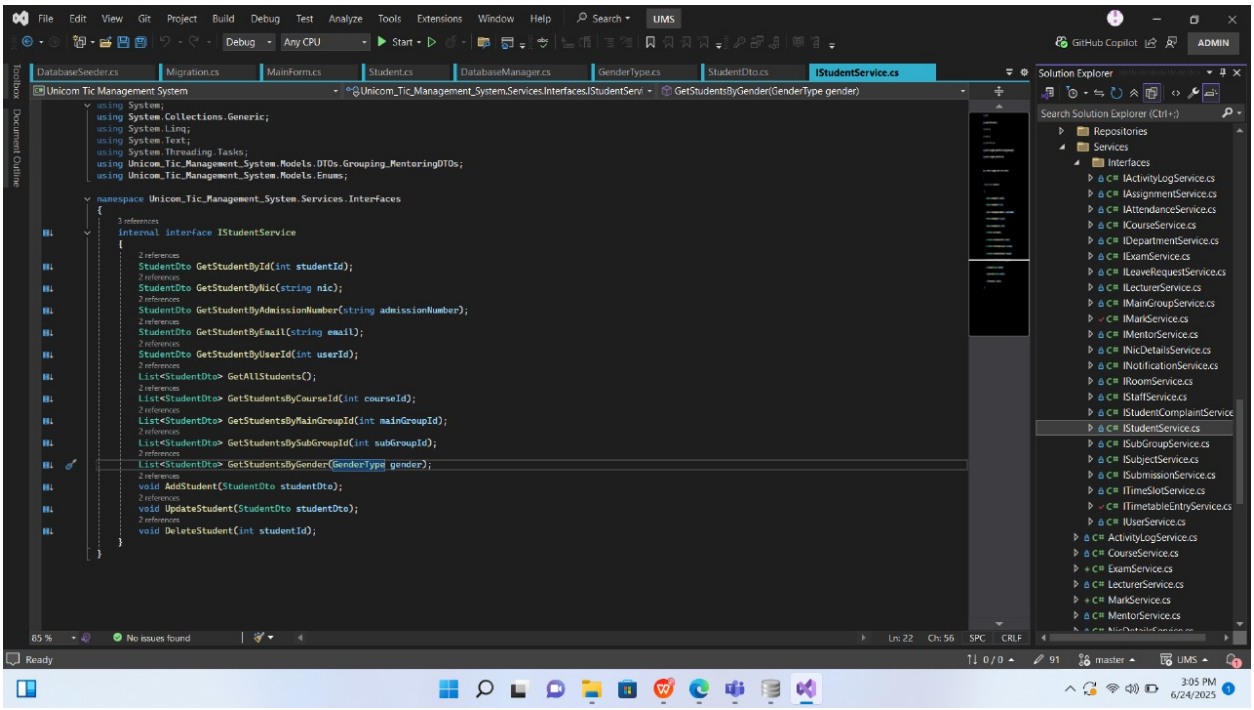

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
SignUpOptionsForm.cs*
Program.cs
PasswordHasher.cs
DatabaseSeeder.cs
MainForm.cs
Student.cs
DatabaseManager.cs
Migration.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Datas.Migration
CreateTables()
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Unicom_Tic_Management_System.Datas;
namespace Unicom_Tic_Management_System.Datas
{
1 reference
internal static class Migration
{
1 reference
public static void CreateTables()
{
using (var conn = DatabaseManager.GetConnection())
{
var cmd = conn.CreateCommand();
cmd.CommandText = @"
PRAGMA foreign_keys = ON;
CREATE TABLE IF NOT EXISTS NICDetails (
NIC TEXT PRIMARY KEY NOT NULL,
IsUsed INTEGER DEFAULT 0
);
CREATE TABLE IF NOT EXISTS Users (
UserID INTEGER PRIMARY KEY AUTOINCREMENT,
Username TEXT NOT NULL UNIQUE,
Password TEXT NOT NULL,
NIC TEXT NOT NULL UNIQUE,
CreatedAt TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP,
UpdatedAt TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP,
Role TEXT NOT NULL DEFAULT 'Student', -- e.g., 'Student', 'Lecturer', 'Staff', 'Admin', 'Mentor'
FOREIGN KEY (NIC) REFERENCES NICDetails(NIC)
);
CREATE TABLE IF NOT EXISTS ActivityLogs (
LogID INTEGER PRIMARY KEY AUTOINCREMENT,
UserID INTEGER,
Action TEXT NOT NULL,
CreatedAt TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP,
Solution Explorer
Search Solution Explorer (Ctrl+;)
TimeSlotController.cs
TimetableController.cs
UserController.cs
Datas
DatabaseManager.cs
DatabaseSeeder.cs
Migration.cs
Models
DTOs
AcademicDTOs
CourseDto.cs
DepartmentDto.c
SubjectDto.cs
AcademicOperations
AssignmentDto.cs
ExamDto.cs
MarkDto.cs
Properties
85 %
No issues found
Ln: 374
Ch: 6
SPC
CRLF
Ready
master
UMS
3:02 PM
6/24/2025


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
StudentRepository.cs
StaffMapper.cs
StudentMapper.cs*
PasswordHasher.cs
StudentRegist...m.cs [Design]
StudentRegistrationForm.cs
Unicom Tic Management System
Unicom_Tic_Management_System.ViewForms.StudentRegistration
StudentRegistrationForm()
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Unicom_Tic_Management_System.Models;
using Unicom_Tic_Management_System.Models.Enums;
using Unicom_Tic_Management_System.Repositories;
using Unicom_Tic_Management_System.Repositories.Interfaces;
namespace Unicom_Tic_Management_System.ViewForms
{
4 references
public partial class StudentRegistrationForm : Form
{
private readonly IStudentRepository _studentRepository;
private readonly IUserRepository _userRepository;
private readonly INicDetailsRepository _nicDetailsrepository;
private readonly ICourseRepository _courseRepository;
private Student _currentStudent = null;
private User _currentUser = null;
1 reference
public StudentRegistrationForm()
{
InitializeComponent();
_studentRepository = new StudentRepository();
_userRepository = new UserRepository();
_nicDetailsrepository = new NICDetailsRepository();
_courseRepository = new CourseRepository();
InitializeForm();
}
0 references
public StudentRegistrationForm(int studentId) : this()
{
LoadStudentForEdit(studentId);
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
StaffMapper.cs
StudentComplaintMapper
StudentMapper.cs
SubGroupMapper.cs
SubjectMapper.cs
SubmissionMapper.cs
TimetableEntryMapper.cs
UserMapper.cs
UserRegistrationMapper.c
UserUpdateMapper.cs
PasswordHasher.cs
ViewForms
CourseForm.cs
ExamForm.cs
ExamForm.Designer.cs
ExamForm.resx
LecturerRegistrationForm.cs
LoginForm.cs
MainForm.cs
MarkForm.cs
MentorsRegistrationForm.cs
NicDetailForm.cs
RoomForm.cs
SignUpOptionsForm.cs
StaffForm.cs
StudentRegistrationForm.cs
StudentRegistrationForm.
StudentRegistrationForm.
SubjectForm.cs
TimeSlotForm.cs
TimetableForm.cs
UserForm.cs
85 %
No issues found
Ln: 28
Ch: 10
SPC
CRLF
Ready
master
UMS
3:08 PM
6/24/2025

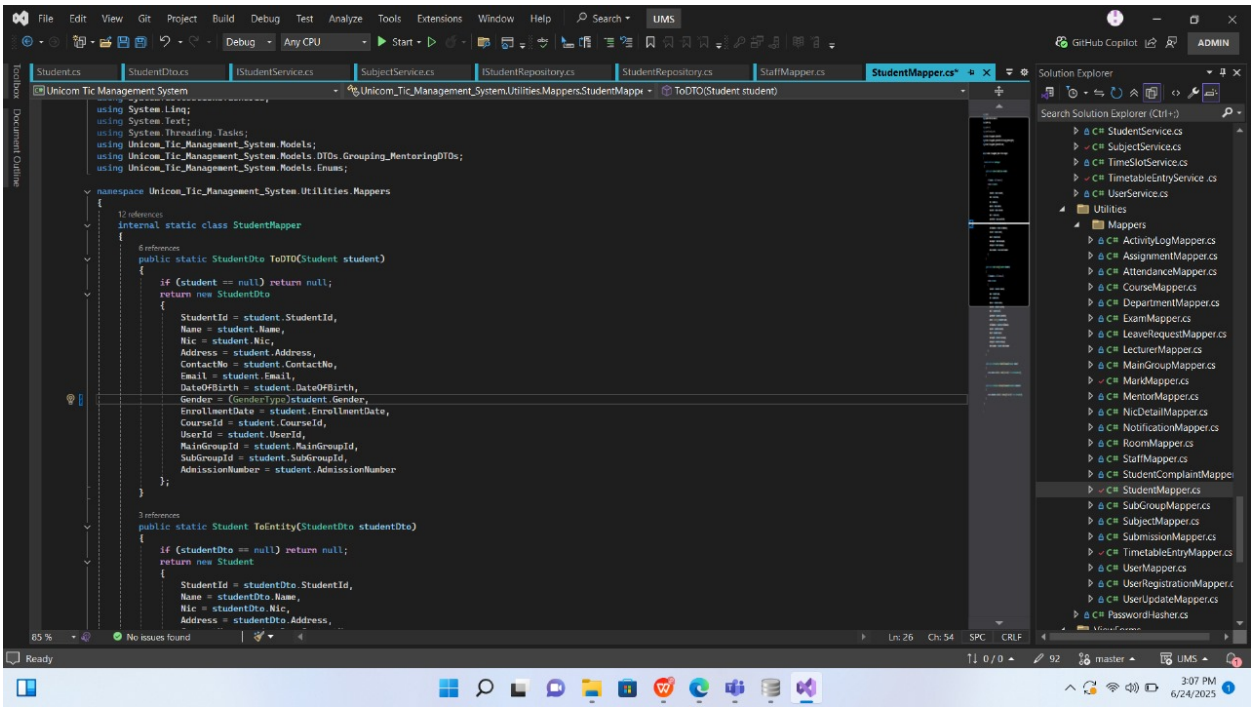

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
DatabaseSeeder.cs
Migration.cs
Student.cs
GenderType.cs
StudentDto.cs
IStudentService.cs
SubjectService.cs
IStudentRepository.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Repositories.Interfaces.IStudent
UpdateStudent(Student student)
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Unicom_Tic_Management_System.Models;
using Unicom_Tic_Management_System.Models.Enums;
namespace Unicom_Tic_Management_System.Repositories.Interfaces
{
internal interface IStudentRepository
{
void AddStudent(Student student);
void UpdateStudent(Student student);
void DeleteStudent(int studentId);
Student GetStudentById(int studentId);
Student GetStudentByNic(string nic);
Student GetStudentByAdmissionNumber(string admissionNumber);
Student GetStudentByEmail(string email);
2 references
Student GetStudentByUserId(int userId);
2 references
List<Student> GetStudentsByCourseId(int courseId);
2 references
List<Student> GetStudentsByMainGroupId(int mainGroupId);
2 references
List<Student> GetStudentsBySubGroupId(int subGroupId);
2 references
List<Student> GetStudentsByGender(GenderType gender);
2 references
List<Student> GetAllStudents();
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
IEmployeeIdRepository.cs
IExamRepository.cs
IGroupMentorRepository.
IGroupSubjectRepository.
ILeaveRequestRepository.
ILecturerRepository.cs
ILecturerStudentRepositor
ILectureSubjectRepository
IMainGroupRepository.cs
IMarkRepository.cs
IMentorRepository.cs
INicDetailsRepository.cs
IRoomRepository.cs
IStaffRepository.cs
IStudentRepository.cs
ISubGroupRepository.cs
ISubjectRepository.cs
ISubmissionRepository.cs
ITimeSlotRepository.cs
ITimetableRepository.cs
IUserRepository.cs
ActivityLogRepository.cs
AdmissionNumberReposito
AssignmentRepository.cs
AttendanceRepository.cs
BackupLogRepository.cs
CourseRepository.cs
CourseSubjectRepository.cs
DepartmentRepository.cs
EmployeeIdRepository.cs
ExamRepository.cs
GroupMentorRepository.cs
85 %
No issues found
Ln: 28
Ch: 1
SPC
CRLF
Microsoft Store
Ready
master
UMS
3:06 PM
6/24/2025



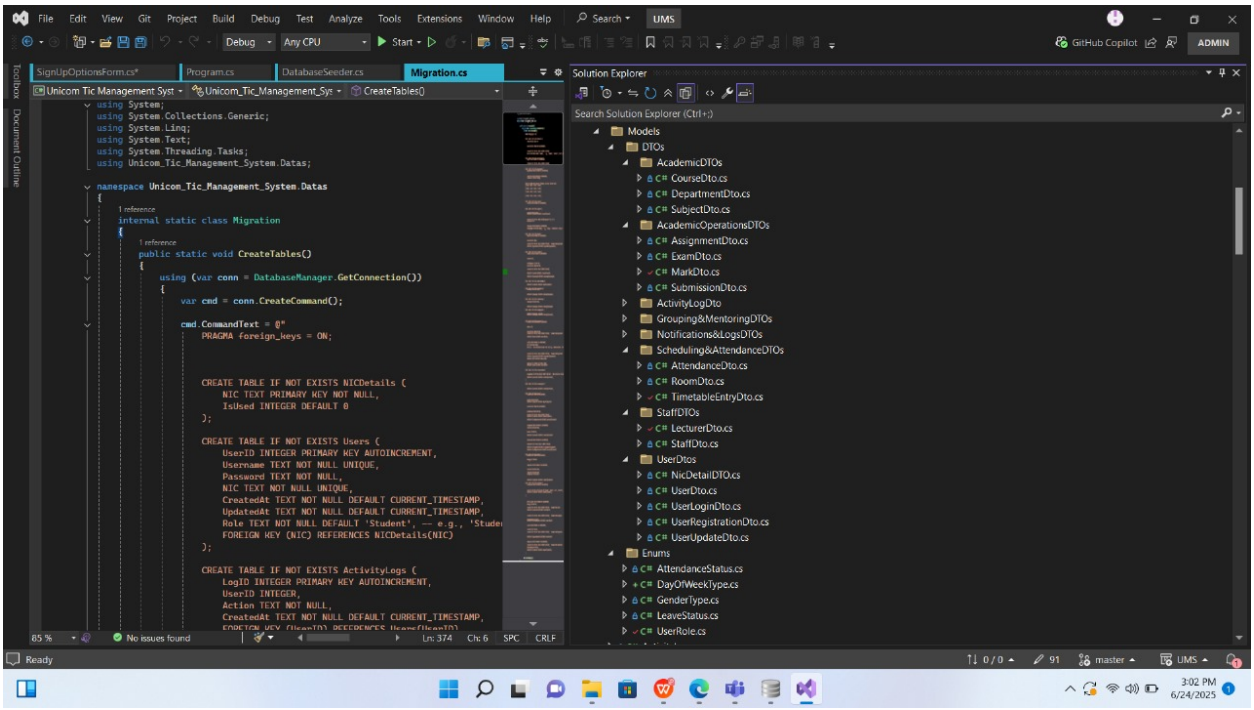
```
using Unicom_Tic_Management_System.Models.DTOs.AcademicDTOs;
using Unicom_Tic_Management_System.Repositories.Interfaces;
using Unicom_Tic_Management_System.Services.Interfaces;
using Unicom_Tic_Management_System.Utilities.Mappers;

namespace Unicom_Tic_Management_System.Services
{
    5 references
    internal class SubjectService : ISubjectService
    {
        private readonly ISubjectRepository _repository;

        4 references
        public SubjectService(ISubjectRepository repository)
        {
            _repository = repository;
        }

        2 references
        public void AddSubject(SubjectDto subjectDto)
        {
            if (subjectDto == null)
                throw new ArgumentNullException(nameof(subjectDto));


            var existing = _repository.GetSubjectByNameAndCourse(subjectDto.SubjectName, subjectDto.CourseId);
            if (existing != null)
                throw new Exception("This subject already exists under the selected course.");

            var subject = SubjectMapper.ToEntity(subjectDto);
            _repository.AddSubject(subject);
        }

        2 references
        public void UpdateSubject(SubjectDto subjectDto)
        {
            if (subjectDto == null)
                throw new ArgumentNullException(nameof(subjectDto));

            var existing = _repository.GetSubjectById(subjectDto.SubjectId);
            if (existing == null)
                throw new Exception("Subject not found.");

            var subject = SubjectMapper.ToEntity(subjectDto);
```



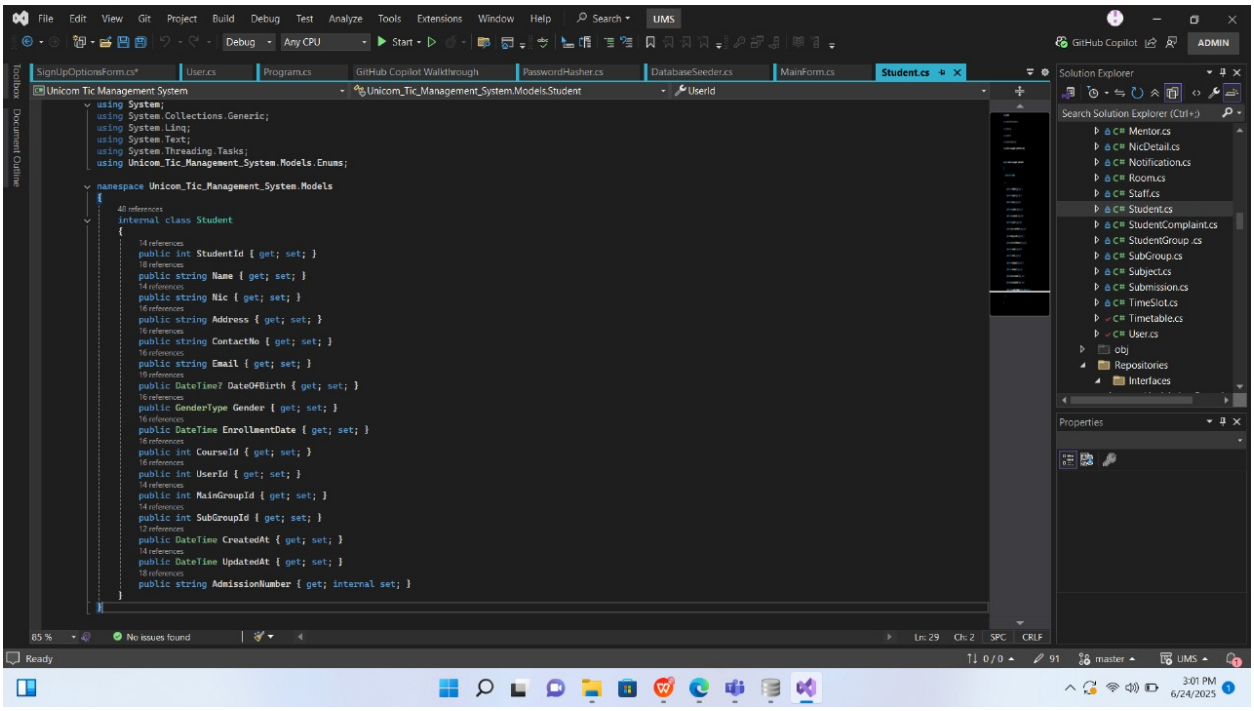
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Unicom_Tic_Management_System.Models.DTOs.Grouping_MentoringDTOs;
using Unicom_Tic_Management_System.Models.Enums;

namespace Unicom_Tic_Management_System.Services.Interfaces
{
    3 references
    internal interface IStudentService
    {
        2 references
        StudentDto GetStudentById(int studentId);
        2 references
        StudentDto GetStudentByNic(string nic);
        2 references
        StudentDto GetStudentByAdmissionNumber(string admissionNumber);
        2 references
        StudentDto GetStudentByEmail(string email);
        2 references
        StudentDto GetStudentByUserId(int userId);
        2 references
        List<StudentDto> GetAllStudents();
        2 references
        List<StudentDto> GetStudentsByCourseId(int courseId);
        2 references
        List<StudentDto> GetStudentsByMainGroupId(int mainGroupId);
        2 references
        List<StudentDto> GetStudentsBySubGroupId(int subGroupId);
        2 references
        List<StudentDto> GetStudentsByGender(GenderType gender);
        2 references
        void AddStudent(StudentDto studentDto);
        2 references
        void UpdateStudent(StudentDto studentDto);
        2 references
        void DeleteStudent(int studentId);
    }
}
```



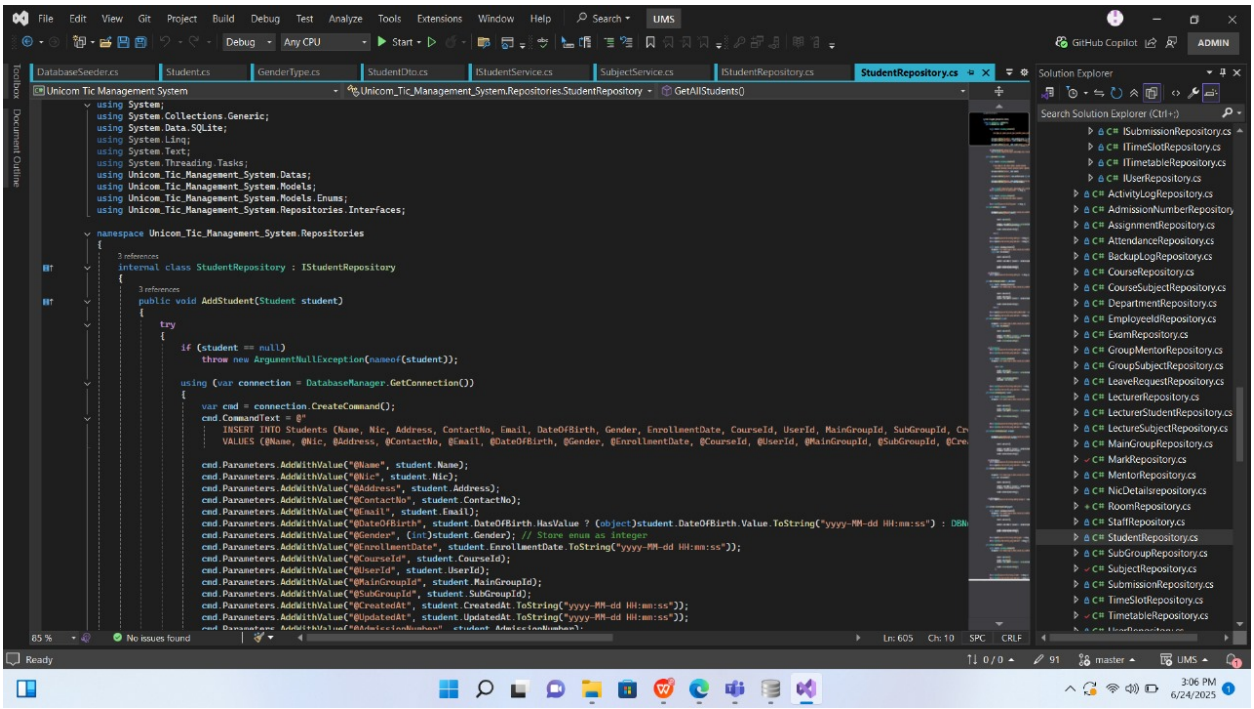
```
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Unicom_Tic_Management_System.Models;
using Unicom_Tic_Management_System.Models.DTOs.Grouping_MentoringDTOs;
using Unicom_Tic_Management_System.Models.Enums;

namespace Unicom_Tic_Management_System.Utilities.Mappers
{
    12 references
    internal static class StudentMapper
    {
        6 references
        public static StudentDto ToDTO(Student student)
        {
            if (student == null) return null;
            return new StudentDto
            {
                StudentId = student.StudentId,
                Name = student.Name,
                Nic = student.Nic,
                Address = student.Address,
                ContactNo = student.ContactNo,
                Email = student.Email,
                DateOfBirth = student.DateOfBirth,
                Gender = (GenderType)student.Gender,
                EnrollmentDate = student.EnrollmentDate,
                CourseId = student.CourseId,
                UserId = student.UserId,
                MainGroupId = student.MainGroupId,
                SubGroupId = student.SubGroupId,
                AdmissionNumber = student.AdmissionNumber
            };
        }

        3 references
        public static Student ToEntity(StudentDto studentDto)
        {
            if (studentDto == null) return null;
            return new Student
            {
                StudentId = studentDto.StudentId,
                Name = studentDto.Name,
                Nic = studentDto.Nic,
                Address = studentDto.Address,
```



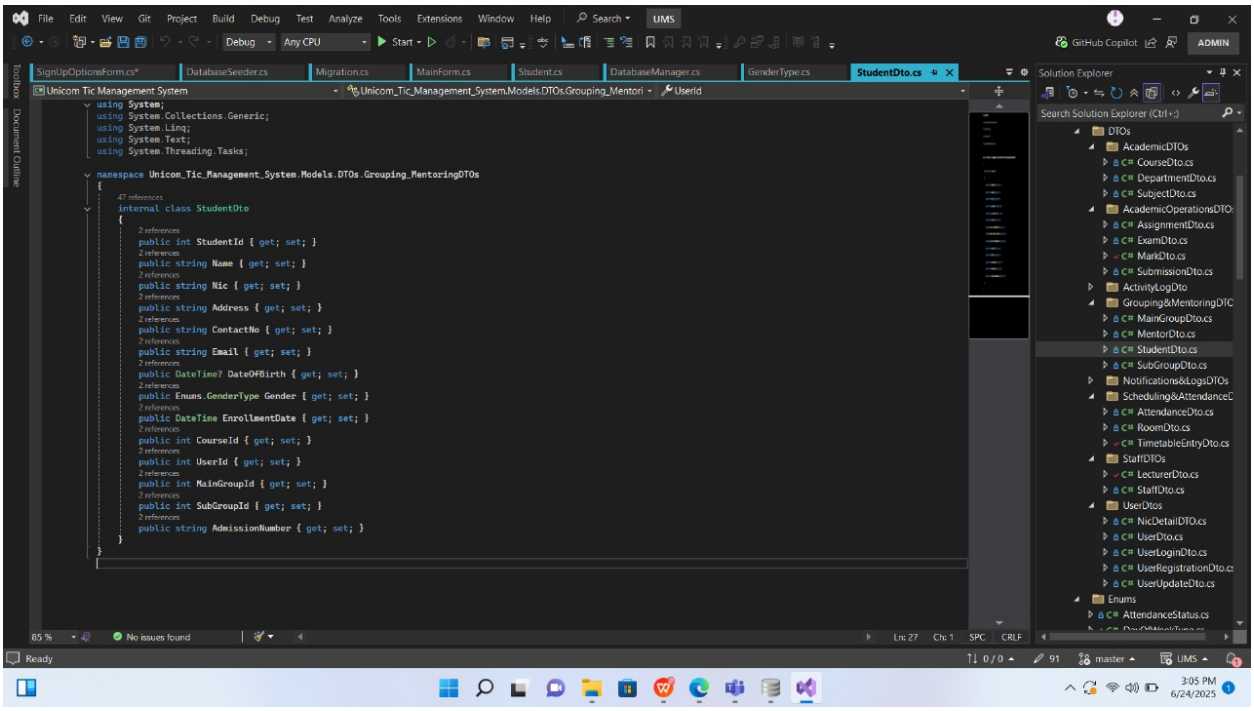
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Unicom_Tic_Management_System.Datas;

namespace Unicom_Tic_Management_System.Datas
{
    1 reference
    internal static class Migration
    {
        1 reference
        public static void CreateTables()
        {
            using (var conn = DatabaseManager.GetConnection())
            {
                var cmd = conn.CreateCommand();

                cmd.CommandText = @"
                    PRAGMA foreign_keys = ON;


                    CREATE TABLE IF NOT EXISTS NICDetails (
                        NIC TEXT PRIMARY KEY NOT NULL,
                        IsUsed INTEGER DEFAULT 0
                    );

                    CREATE TABLE IF NOT EXISTS Users (
                        UserID INTEGER PRIMARY KEY AUTOINCREMENT,
                        Username TEXT NOT NULL UNIQUE,
                        Password TEXT NOT NULL,
                        NIC TEXT NOT NULL UNIQUE,
                        CreatedAt TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP,
                        UpdatedAt TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP,
                        Role TEXT NOT NULL DEFAULT 'Student', -- e.g., 'Stude
                        FOREIGN KEY (NIC) REFERENCES NICDetails(NIC)
                    );

                    CREATE TABLE IF NOT EXISTS ActivityLogs (
                        LogID INTEGER PRIMARY KEY AUTOINCREMENT,
                        UserID INTEGER,
                        Action TEXT NOT NULL,
                        CreatedAt TEXT NOT NULL DEFAULT CURRENT_TIMESTAMP,
```

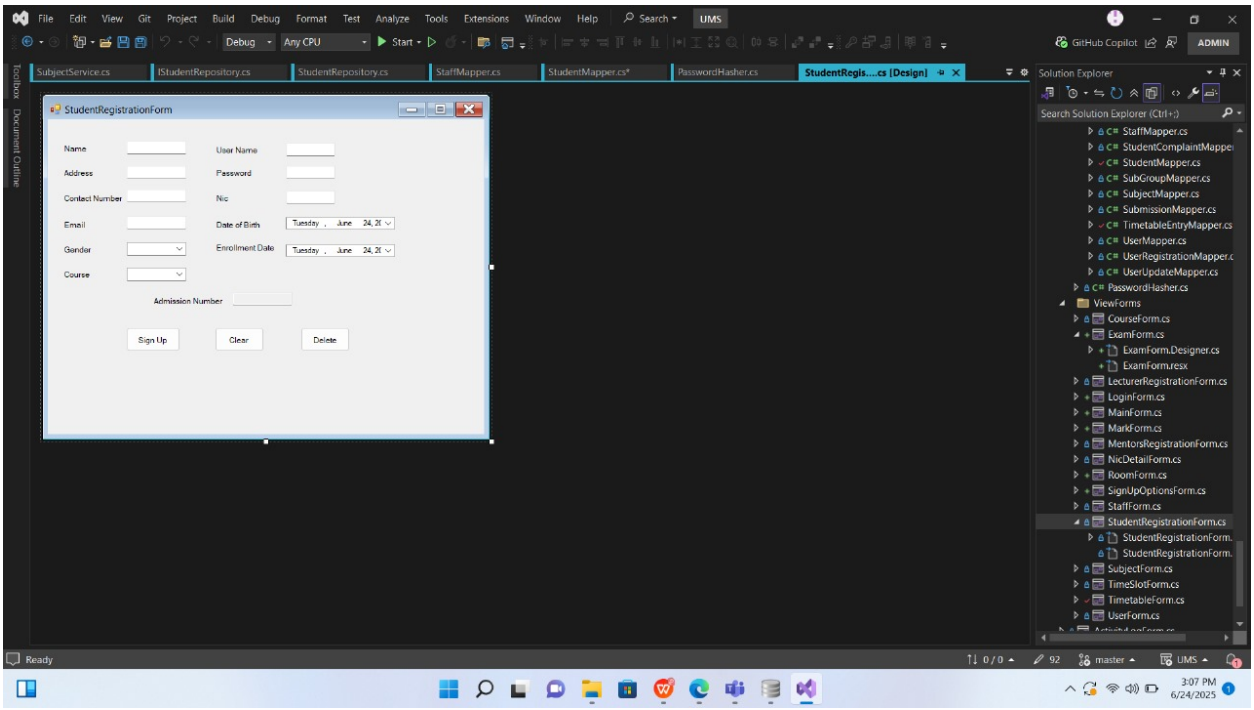


```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Unicom_Tic_Management_System.Models.Enums;

namespace Unicom_Tic_Management_System.Models
{
    internal class Student
    {
        public int StudentId { get; set; }
        public string Name { get; set; }
        public string Nic { get; set; }
        public string Address { get; set; }
        public string ContactNo { get; set; }
        public string Email { get; set; }
        public DateTime? DateOfBirth { get; set; }
        public GenderType Gender { get; set; }
        public DateTime EnrollmentDate { get; set; }
        public int CourseId { get; set; }
        public int UserId { get; set; }
        public int MainGroupId { get; set; }
        public int SubGroupId { get; set; }
        public DateTime CreatedAt { get; set; }
        public DateTime UpdatedAt { get; set; }
        public string AdmissionNumber { get; set; }
    }
}
```



```
using System;
using System.Collections.Generic;
using System.Data.SQLite;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Unicom_Tic_Management_System.Datas;
using Unicom_Tic_Management_System.Models;
using Unicom_Tic_Management_System.Models.Enums;
using Unicom_Tic_Management_System.Repositories.Interfaces;

namespace Unicom_Tic_Management_System.Repositories
{
    internal class StudentRepository : IStudentRepository
    {
        public void AddStudent(Student student)
        {
            try
            {
                if (student == null)
                    throw new ArgumentNullException(nameof(student));

                using (var connection = DatabaseManager.GetConnection())
                {
                    var cmd = connection.CreateCommand();
                    cmd.CommandText = @"
                        INSERT INTO Students (Name, Nic, Address, ContactNo, Email, DateOfBirth, Gender, EnrollmentDate, CourseId, UserId, MainGroupId, SubGroupId, Cr
                        VALUES (@Name, @Nic, @Address, @ContactNo, @Email, @DateOfBirth, @Gender, @EnrollmentDate, @CourseId, @UserId, @MainGroupId, @SubGroupId, @Cre

                    cmd.Parameters.AddWithValue("@Name", student.Name);
                    cmd.Parameters.AddWithValue("@Nic", student.Nic);
                    cmd.Parameters.AddWithValue("@Address", student.Address);
                    cmd.Parameters.AddWithValue("@ContactNo", student.ContactNo);
                    cmd.Parameters.AddWithValue("@Email", student.Email);
                    cmd.Parameters.AddWithValue("@DateOfBirth", student.DateOfBirth.HasValue ? (object)student.DateOfBirth.Value.ToString("yyyy-MM-dd HH:mm:ss") : DBN
                    cmd.Parameters.AddWithValue("@Gender", (int)student.Gender); // Store enum as integer
                    cmd.Parameters.AddWithValue("@EnrollmentDate", student.EnrollmentDate.ToString("yyyy-MM-dd HH:mm:ss"));
                    cmd.Parameters.AddWithValue("@CourseId", student.CourseId);
                    cmd.Parameters.AddWithValue("@UserId", student.UserId);
                    cmd.Parameters.AddWithValue("@MainGroupId", student.MainGroupId);
                    cmd.Parameters.AddWithValue("@SubGroupId", student.SubGroupId);
                    cmd.Parameters.AddWithValue("@CreatedAt", student.CreatedAt.ToString("yyyy-MM-dd HH:mm:ss"));
                    cmd.Parameters.AddWithValue("@UpdatedAt", student.UpdatedAt.ToString("yyyy-MM-dd HH:mm:ss"));
```


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
UMS
ADMIN
SignUpOptionsForm.cs*
DatabaseSeeder.cs
Migration.cs
MainForm.cs
Student.cs
DatabaseManager.cs
GenderType.cs
StudentDto.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Models.DTOs.Grouping_Mentori
UserId
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace Unicom_Tic_Management_System.Models.DTOs.Grouping_MentoringDTOs
{
47 references
internal class StudentDto
{
2 references
public int StudentId { get; set; }
2 references
public string Name { get; set; }
2 references
public string Nic { get; set; }
2 references
public string Address { get; set; }
2 references
public string ContactNo { get; set; }
2 references
public string Email { get; set; }
2 references
public DateTime? DateOfBirth { get; set; }
2 references
public Enums.GenderType Gender { get; set; }
2 references
public DateTime EnrollmentDate { get; set; }
2 references
public int CourseId { get; set; }
2 references
public int UserId { get; set; }
2 references
public int MainGroupId { get; set; }
2 references
public int SubGroupId { get; set; }
2 references
public string AdmissionNumber { get; set; }
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
DTOs
AcademicDTOs
CourseDto.cs
DepartmentDto.cs
SubjectDto.cs
AcademicOperationsDTO
AssignmentDto.cs
ExamDto.cs
MarkDto.cs
SubmissionDto.cs
ActivityLogDto
Grouping&MentoringDTC
MainGroupDto.cs
MentorDto.cs
StudentDto.cs
SubGroupDto.cs
Notifications&LogsDTOs
Scheduling&AttendanceD
AttendanceDto.cs
RoomDto.cs
TimetableEntryDto.cs
StaffDTOs
LecturerDto.cs
StaffDto.cs
UserDtos
NicDetailDTO.cs
UserDto.cs
UserLoginDto.cs
UserRegistrationDto.c
UserUpdateDto.cs
Enums
AttendanceStatus.cs
85 %
No issues found
Ln: 27 Ch: 1 SPC CRLF
Ready
master
3:05 PM
6/24/2025


File Edit View Git Project Build Debug Format Test Analyze Tools Extensions Window Help
UMS
ADMIN
SubjectService.cs
IStudentRepository.cs
StudentRepository.cs
StaffMapper.cs
StudentMapper.cs*
PasswordHasher.cs
StudentRegis....cs [Design]
StudentRegistrationForm
Name
User Name
Address
Password
Contact Number
Nic
Email
Date of Birth
Gender
Enrollment Date
Course
Admission Number
Sign Up
Clear
Delete
Solution Explorer
Search Solution Explorer (Ctrl+;)
StaffMapper.cs
StudentComplaintMappe
StudentMapper.cs
SubGroupMapper.cs
SubjectMapper.cs
SubmissionMapper.cs
TimetableEntryMapper.cs
UserMapper.cs
UserRegistrationMapper.c
UserUpdateMapper.cs
PasswordHasher.cs
ViewForms
CourseForm.cs
ExamForm.cs
ExamForm.Designer.cs
ExamForm.resx
LecturerRegistrationForm.cs
LoginForm.cs
MainForm.cs
MarkForm.cs
MentorsRegistrationForm.cs
NicDetailForm.cs
RoomForm.cs
SignUpOptionsForm.cs
StaffForm.cs
StudentRegistrationForm.cs
StudentRegistrationForm.
StudentRegistrationForm.
SubjectForm.cs
TimeSlotForm.cs
TimetableForm.cs
UserForm.cs
Ready
master
3:07 PM
6/24/2025