Unicom TIC Management System - Project Report
Developed Using: C#, WinForms, SQLite

Submitted by: Sathusha Santhiravarnan

UT Number: UT010651

1. Introduction

The Unicom TIC Management System (UMS) is a robust and user-friendly Windows desktop application designed to streamline the academic and administrative operations of an educational institution. Building upon the foundational requirements of the C# Assignment, this project meticulously implements a comprehensive suite of features leveraging C# WinForms, a layered architecture incorporating Model-View-Controller (MVC) principles, and a persistent SQLite database.

This report details the system's architecture, implemented modules, advanced features, and the successful resolution of various development challenges, demonstrating a strong understanding of modern software development practices.

2. Project Objectives

The primary objectives achieved through this project include:

User-Friendly Desktop Application: Developed a highly intuitive and responsive Windows desktop application using C# and WinForms.

Layered Architecture Implementation: Successfully implemented a sophisticated layered architecture, including Models, DTOs, Enums, Mappers, Interfaces, Repositories, and Services, along with a Controller layer for strict separation of concerns and maintainability.

Comprehensive CRUD Operations: Enabled efficient Create, Read, Update, and Delete (CRUD) operations across all key entities using SQLite as the local database.

Robust Role-Based Access Control: Implemented a secure login system with granular role-based access for Admins, Staff, Lecturers, and Students, ensuring appropriate permissions for each user type.

Timetable and Room Scheduling: Developed functionalities for managing timetables, including the allocation of computer labs and lecture halls.

Advanced Validation & Error Handling: Integrated robust form validation and comprehensive error handling mechanisms, including try-catch blocks across all database and service operations for enhanced reliability.

Automated ID Generation: Implemented automatic generation of Admission Numbers for

Students and Employee IDs for Staff and Lecturers upon registration.

3. System Architecture

The UMS adheres to a multi-layered architectural pattern, significantly enhancing maintainability, scalability, and testability.

Core Layers:

Models: Defines the data structures (e.g., User, Student, Course). Includes specialized DTO (Data Transfer Objects) for efficient data transfer between layers and Enums for consistent status and role definitions.

Mappers: Dedicated classes (UserMapper, StudentMapper, etc.) responsible for converting data between Model entities and DTOs, ensuring clean data segregation.

Repositories: Provides an abstraction layer for data access. Interfaces define contracts (IUserRepository, IStudentRepository), while concrete Repository implementations (UserRepository, StudentRepository) handle direct SQLite database interactions (CRUD operations).

Services: Contains the business logic of the application. Interfaces (IUserService, IStudentService) define the service contracts, and concrete Service implementations (UserService, StudentService) orchestrate interactions between repositories and apply business rules.

Controllers: Acts as the intermediary between the Views (UI) and the Services layer. Controllers handle user input, invoke appropriate service methods, and update the view.

Views: The Windows Forms (UI) that users interact with.

Utilities & Datas:

Utilities (e.g., PasswordHasher): Contains common utility functions, such as password hashing, promoting reusability and security best practices.

Datas (e.g., DatabaseManager, Migration, DatabaseSeeder): Manages database connection, schema creation, and initial data seeding.

This layered approach ensures a strong separation of concerns, making the application modular and easy to manage.

4. Key Features & Modules Implemented

The system incorporates several core modules, each designed to manage specific aspects of the educational institution's operations:

4.1. User Management & Authentication

Secure Login System: Implements a robust login mechanism. Instead of a hardcoded admin, all users, including the Admin, register and authenticate against the Users table in the SQLite database.

Role-Based Access Control: Each user is assigned a UserRole (Admin, Student, Lecturer, Staff, Mentor) upon registration. The system dynamically adjusts UI elements and functionalities based on the logged-in user's role (e.g., only Admin can perform Add/Update/Delete operations on core data entities).

Unique NIC Validation: Ensures that only "UnicomTic users" (identified by a unique NIC) can register, preventing unauthorized sign-ups and maintaining data integrity. Each NIC used for registration is tracked in the NICDetails table.

Automated ID Generation: Upon registration:

Students are automatically assigned an AdmissionNumber.

Staff and Lecturers are automatically assigned an EmployeeID.

Password Hashing: Passwords are securely hashed using the dedicated PasswordHasher utility class before storage, and verified during login, enhancing security.

4.2. Core Academic Modules

Course Management: Allows admins to add, edit, and delete course information.

Subject Management: Enables the creation and linking of subjects to specific courses.

Student Management: Comprehensive module for student entry, approval, and management of student details.

Lecturer Management: Records and manages lecturer details.

Staff Management: Handles staff records and details.

Mentor Module: Facilitates the registration and listing of mentors, allowing for assignment to student groups.

Main & Sub Groups: Implemented a hierarchical grouping system (MainGroups and SubGroups) to organize students and classes effectively, supporting various academic structures.

4.3. Timetable & Attendance

Timetable Management: Enables scheduling courses, subjects, and lecturers for specific time slots and rooms (computer labs/lecture halls).

Attendance Tracking: Facilitates the tracking of student attendance per group and time slot.

4.4. Exams & Marks
Exam Management: Allows linking subjects to specific exams.
Marks Management: Enables entering, viewing, and modifying student marks for exams. Top performers functionality can be extended from this data.
4.5. Logging & Notifications
Activity Logs: Tracks system logs of user actions (primarily for Admin review), providing an audit trail for important operations.
Notifications: Supports a notification system for conveying important messages to specific users or roles.

4.6. Other Features
Leave Requests: Manages leave requests submitted by users, with approval workflows.
Student Complaints: Provides a mechanism for students to submit complaints and for administrators to resolve them.
Backup Logs: Tracks database backup activities.
Comprehensive Error Handling: Extensive use of try-catch blocks throughout the application, especially within Repository and Service layers, ensures graceful handling of exceptions and provides informative error messages to the user and console.

5. Database Design (SQLite)
The application utilizes a SQLite database (UnicomTicManagementDB.db) for persistent data storage. The database schema is designed with proper relationships (Foreign Keys) to ensure data integrity and consistency. Key tables include:
NICDetails

Users

ActivityLogs

Departments

TimeSlots

Courses

Subjects

MainGroups

SubGroups

Mentors

Students

AdmissionNumbers

GroupMentors

StudentGroups

GroupSubjects

Lecturers

Staff

EmployeeIDs

LecturerStudents

LectureSubjects

Exams

Marks

Assignments

Submissions

Rooms

Timetables

Attendance

Notifications

BackupLogs

LeaveRequests

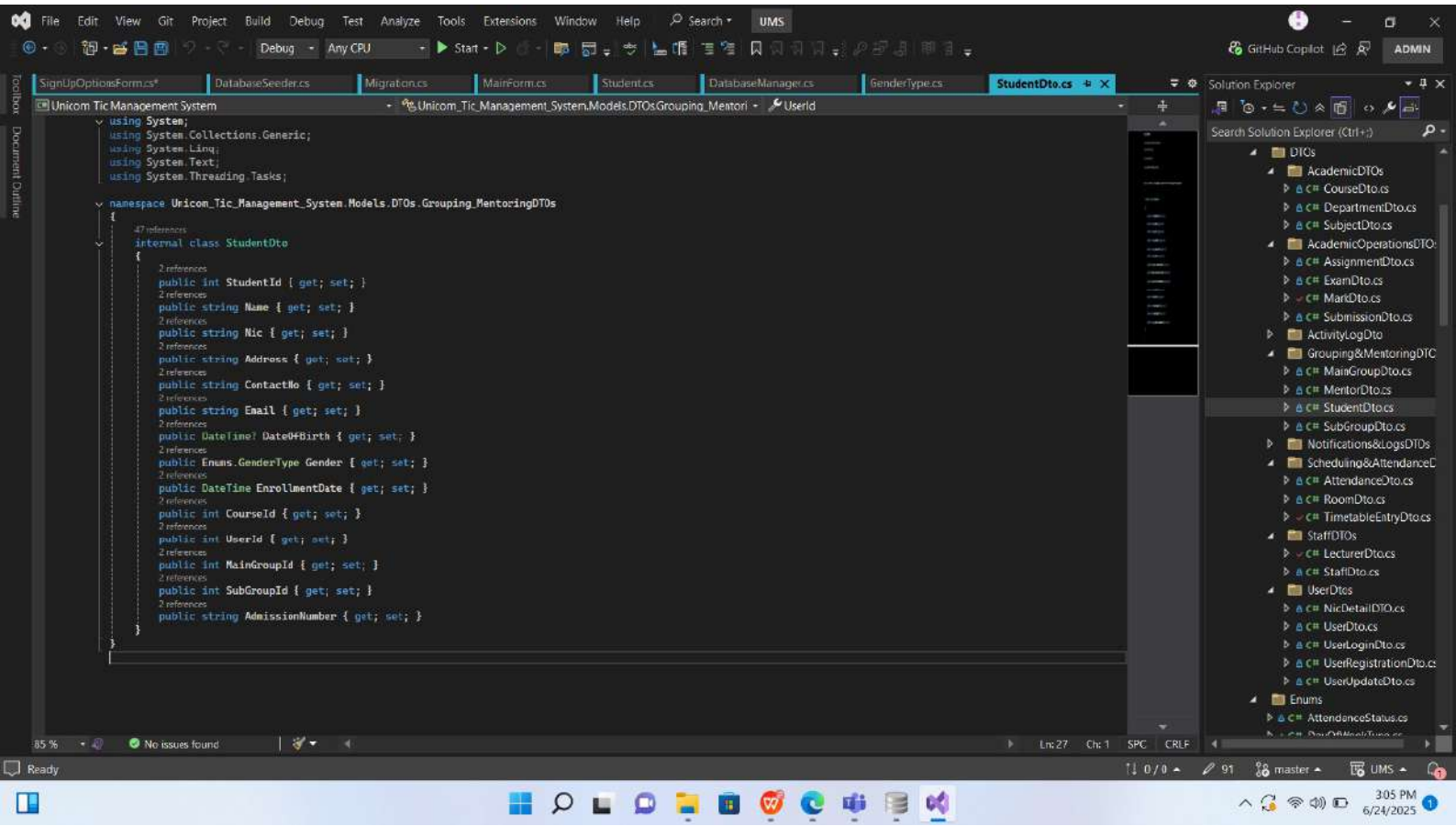
StudentComplaints

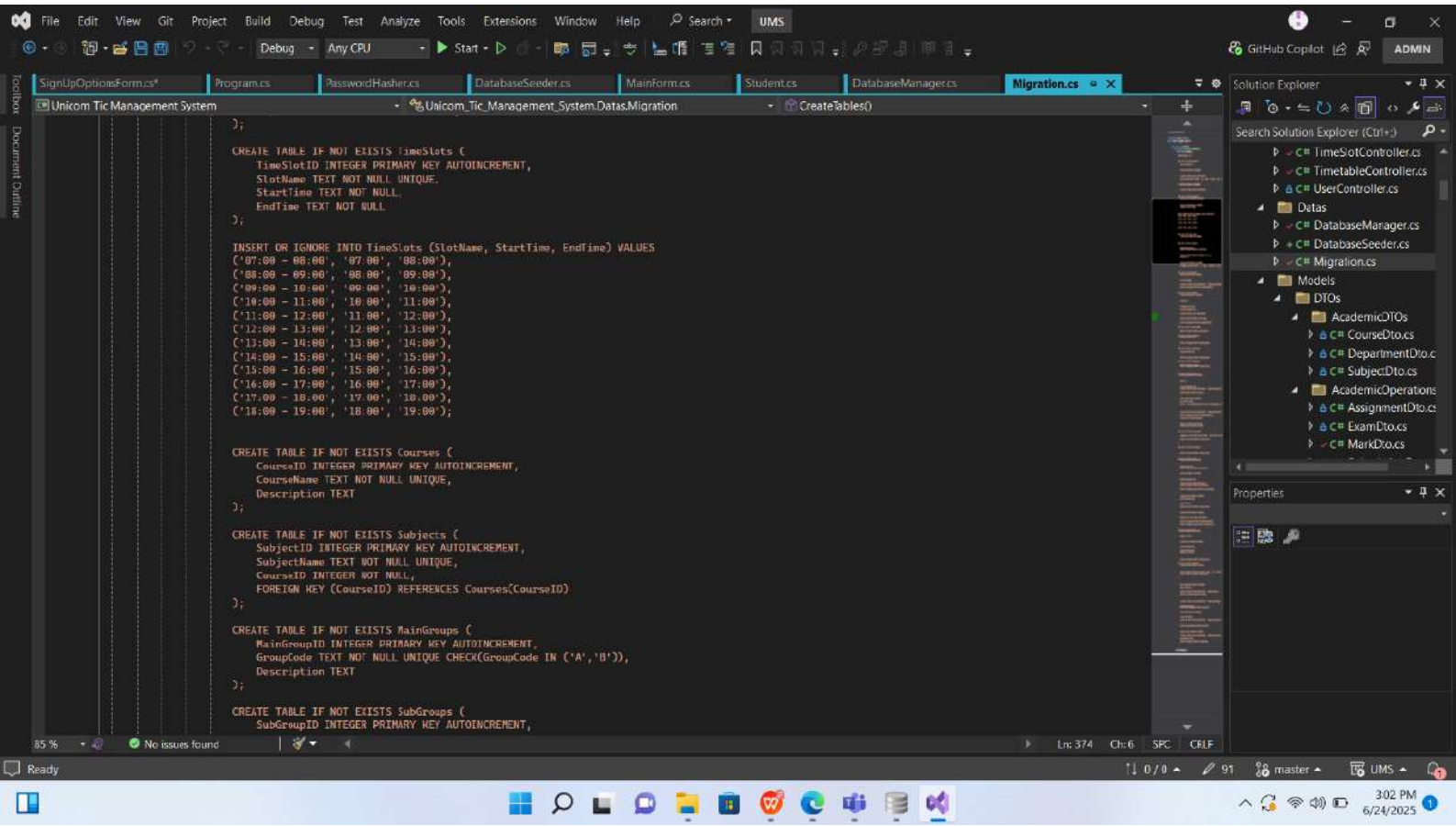
The database tables are auto-created on application startup via SQL scripts within Migration.cs.

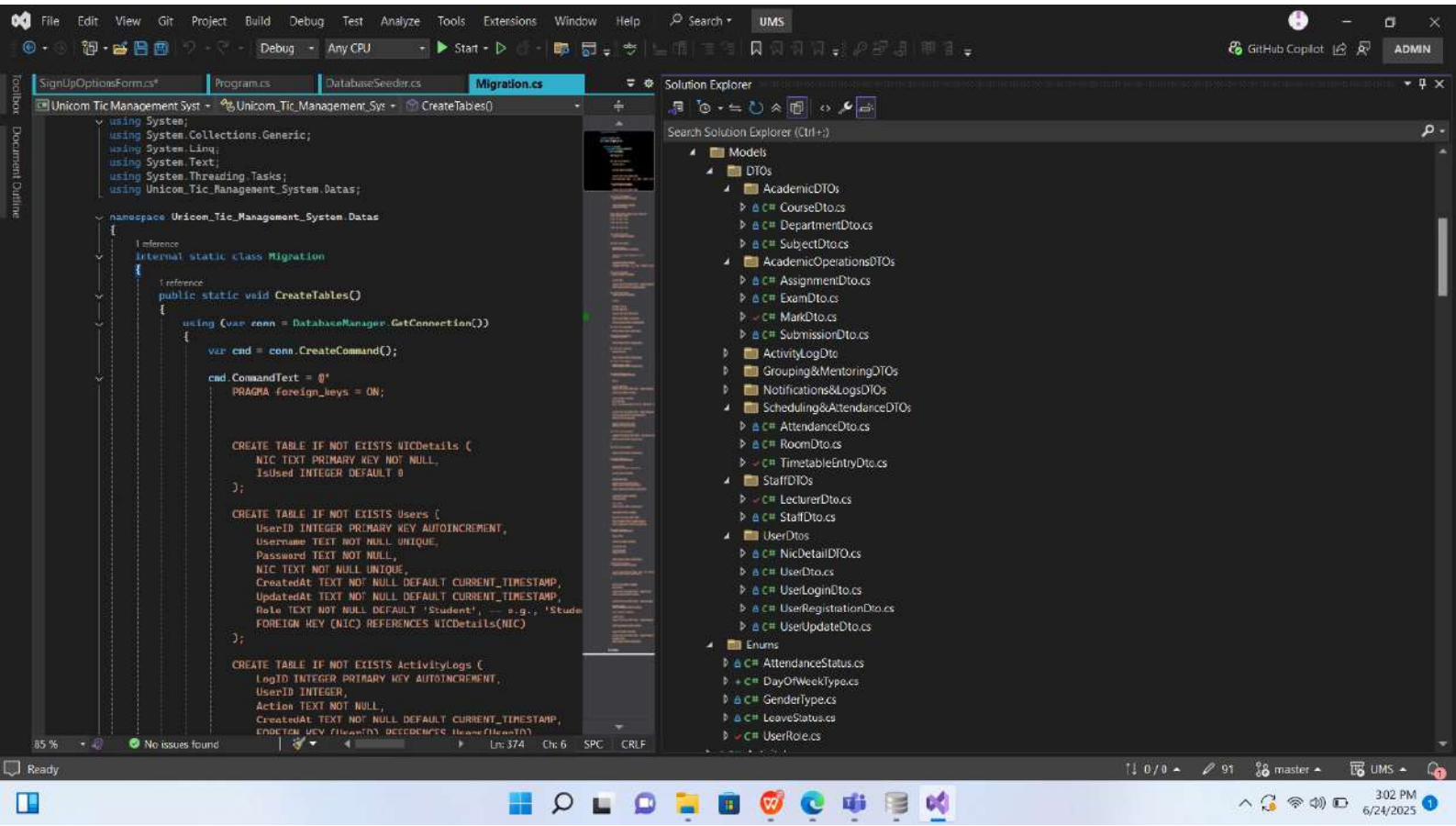
6. Development Methodologies & Best Practices
SOLID Principles: The layered architecture and interface-driven design (IUserRepository, IUserService, etc.) adhere to SOLID principles, promoting maintainability, flexibility, and testability.
Separation of Concerns: Each layer and module has a distinct responsibility, ensuring that changes in one area do not adversely affect others.
Code Reusability: Utility classes like PasswordHasher centralize common functionalities, preventing code duplication.
Robust Error Handling: Consistent try-catch blocks with specific exception handling provide a resilient application.
Clean UI/UX: The WinForms interface is designed to be intuitive and user-friendly, providing clear navigation and feedback.

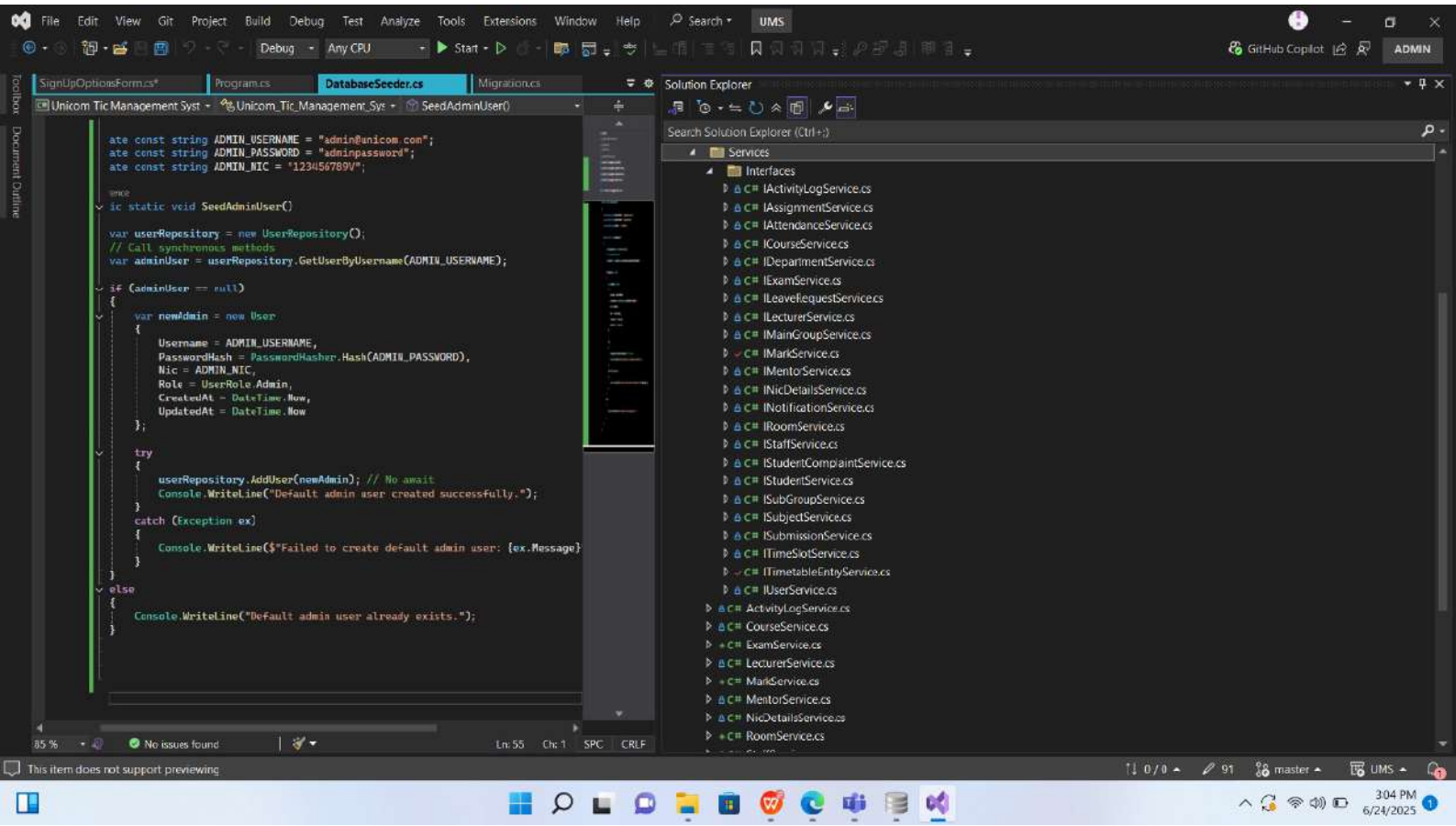
7. Challenges Faced & Solutions
Database Schema Evolution: Adapting the initial database design to incorporate complex relationships (e.g., Main/Sub Groups, specific user IDs like NIC, Admission Numbers, Employee IDs) while maintaining foreign key integrity.
Solution: Iterative refinement of Migration.cs with careful attention to table creation order and foreign key constraints.
Role-Based Access Implementation: Dynamically enabling/disabling UI elements and controlling

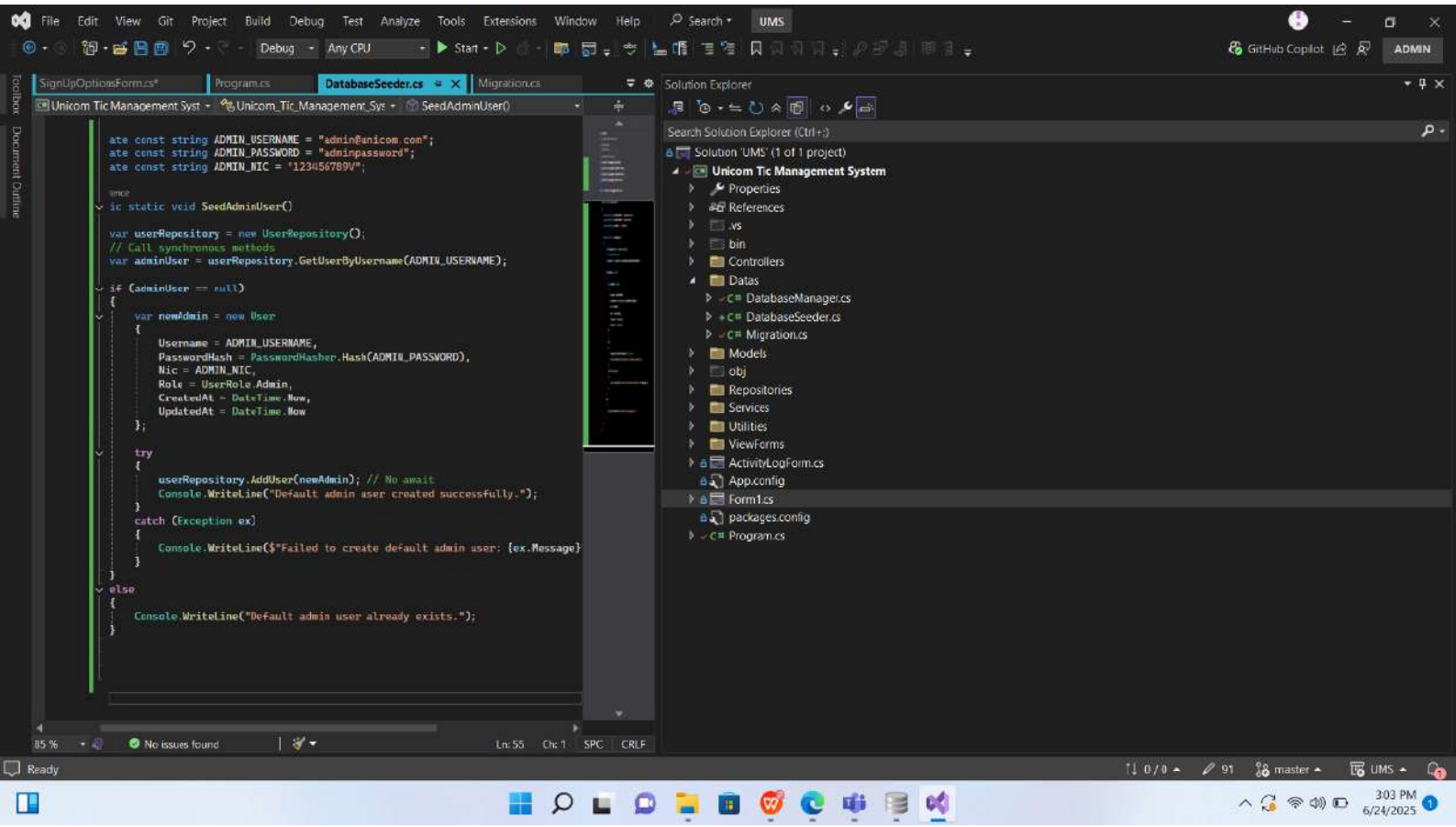
access to specific operations based on user roles.
Solution: Implementing role checks in Controllers and UI event handlers, ensuring only authorized users can perform sensitive actions.
Password Security: Storing and verifying passwords securely.
Solution: Implemented a dedicated PasswordHasher class using SHA256 hashing and Base64 encoding for storage, with a VerifyPassword method for secure authentication.
Enum Conversion: Handling the conversion between C# enum types (like UserRole) and string representations stored in SQLite.
Solution: Explicitly parsing strings to enums (Enum.Parse) when reading from the database and converting enums to strings (.ToString()) when writing to the database.
Automated ID Generation Logic: Ensuring unique AdmissionNumber and EmployeeID generation for different user types.
Solution: Integrated logic within the registration process to assign these unique identifiers based on the user's role.

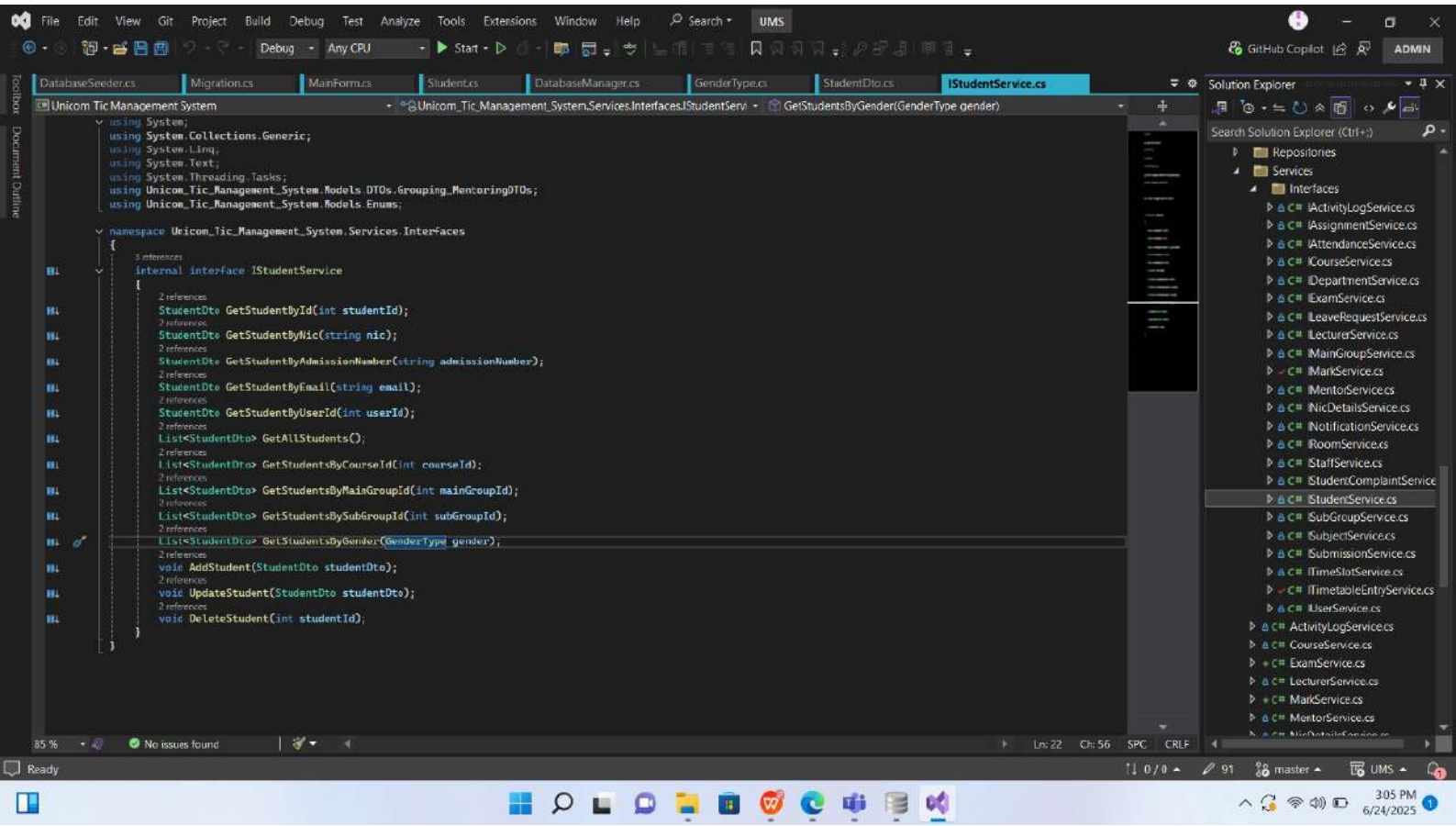

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
SignUpOptionsForm.cs*
DatabaseSeeder.cs
Migration.cs
MainForm.cs
Student.cs
DatabaseManager.cs
GenderType.cs
StudentDto.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Models.DTOs.Grouping_MentoringDTOs.StudentDto
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
AcademicOperationsDTOs
AssignmentDto.cs
ExamDto.cs
MarkDto.cs
SubmissionDto.cs
ActivityLogDto
Grouping&MentoringDTOs
MainGroupDto.cs
MentorDto.cs
StudentDto.cs
SubGroupDto.cs
Notifications&LogsDTOs
Scheduling&AttendanceDTOs
AttendanceDto.cs
RoomDto.cs
TimetableEntryDto.cs
StaffDTOs
LecturerDto.cs
StaffDto.cs
UserDtos
NicDetailDTO.cs
UserDto.cs
UserLoginDto.cs
UserRegistrationDto.cs
UserUpdateDto.cs
Enums
AttendanceStatus.cs
85 %
No issues found
Ln: 27
Ch: 1
SPC
CRLF
Ready
0 / 0
91
master
UMS
305 PM
6/24/2025

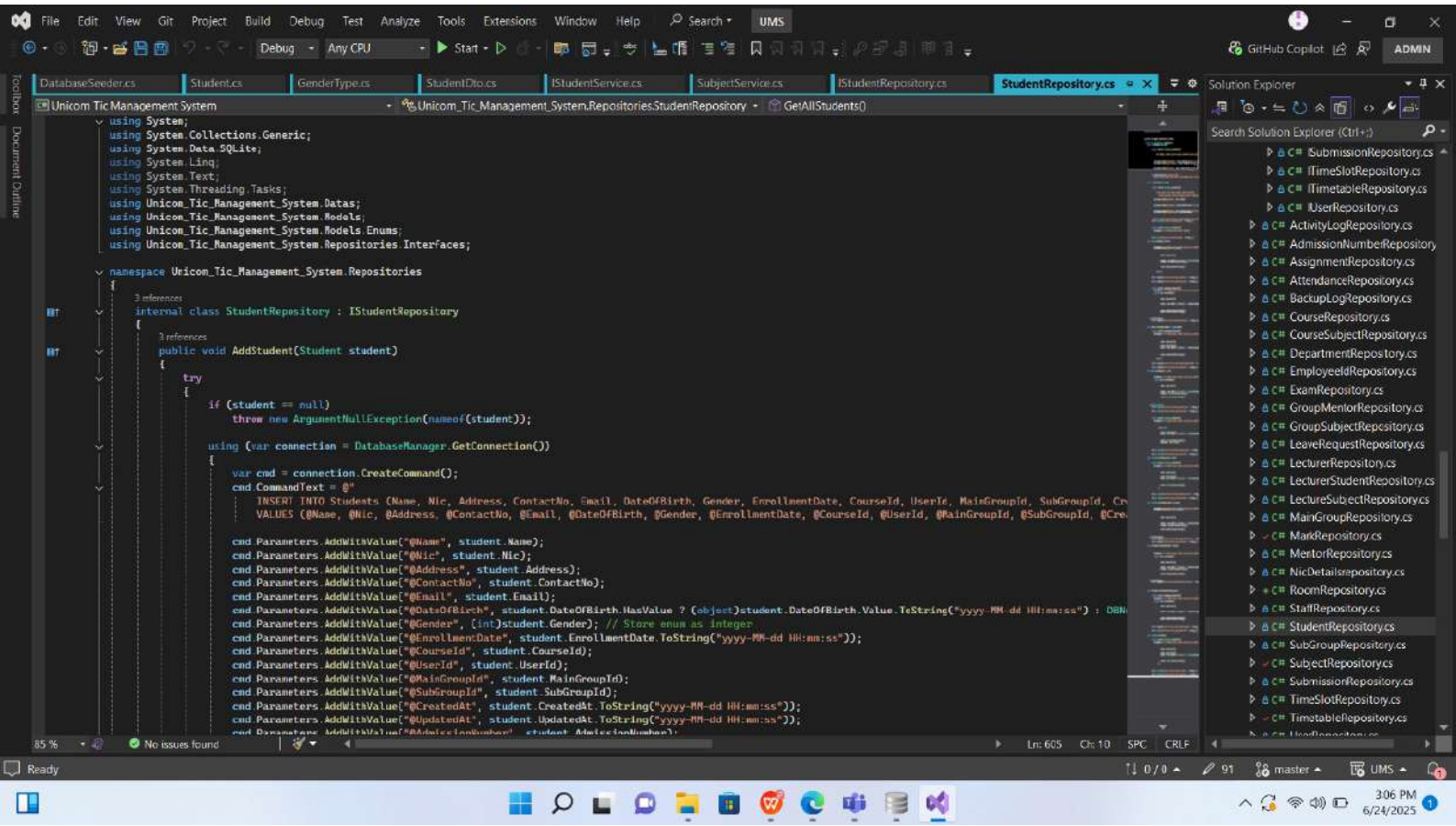

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
SignUpOptionsForm.cs*
Program.cs
PasswordHasher.cs
DatabaseSeeder.cs
MainForm.cs
Student.cs
DatabaseManager.cs
Migration.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Datas.Migration
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
0 / 0
91
master
UMS
302 PM
6/24/2025

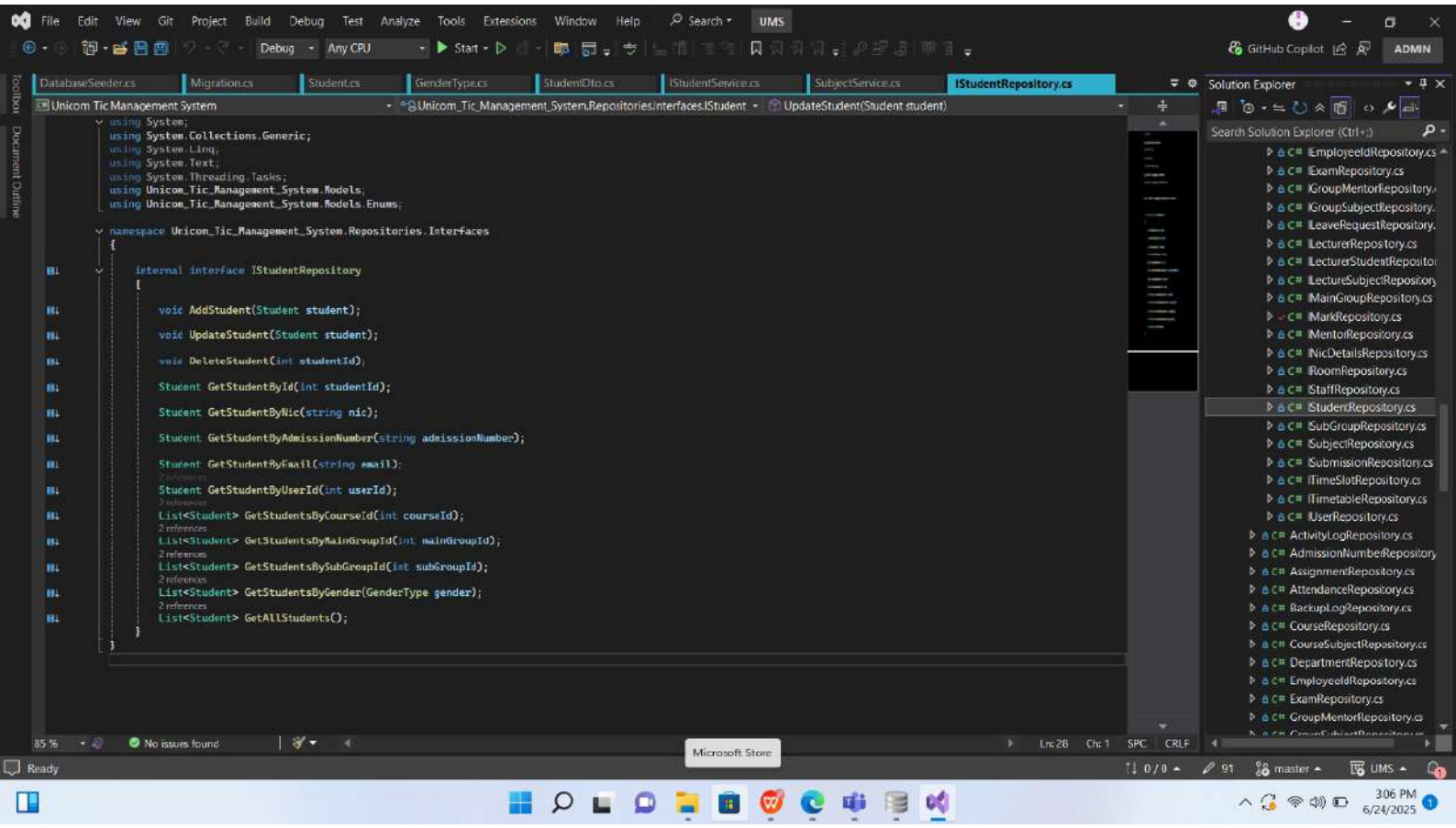

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
SignUpOptionsForm.cs*
Program.cs
DatabaseSeeder.cs
Migration.cs
Unicom Tic Management Syst
Unicom_Tic_Management_Sys
CreateTables()
using System;
using System.Collections.Generic;
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., '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,
85 %
No issues found
Ln: 374
Ch: 6
SPC
CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Models
DTOs
AcademicDTOs
CourseDto.cs
DepartmentDto.cs
SubjectDto.cs
AcademicOperationsDTOs
AssignmentDto.cs
ExamDto.cs
MarkDto.cs
SubmissionDto.cs
ActivityLogDto
Grouping&MentoringDTOs
Notifications&LogsDTOs
Scheduling&AttendanceDTOs
AttendanceDto.cs
RoomDto.cs
TimetableEntryDto.cs
StaffDTOs
LecturerDto.cs
StaffDto.cs
UserDtos
NicDetailDTO.cs
UserDto.cs
UserLoginDto.cs
UserRegistrationDto.cs
UserUpdateDto.cs
Enums
AttendanceStatus.cs
DayOfWeekTypo.cs
GenderType.cs
LeaveStatus.cs
UserRole.cs
Ready
0 / 0
91
master
UMS
302 PM
6/24/2025

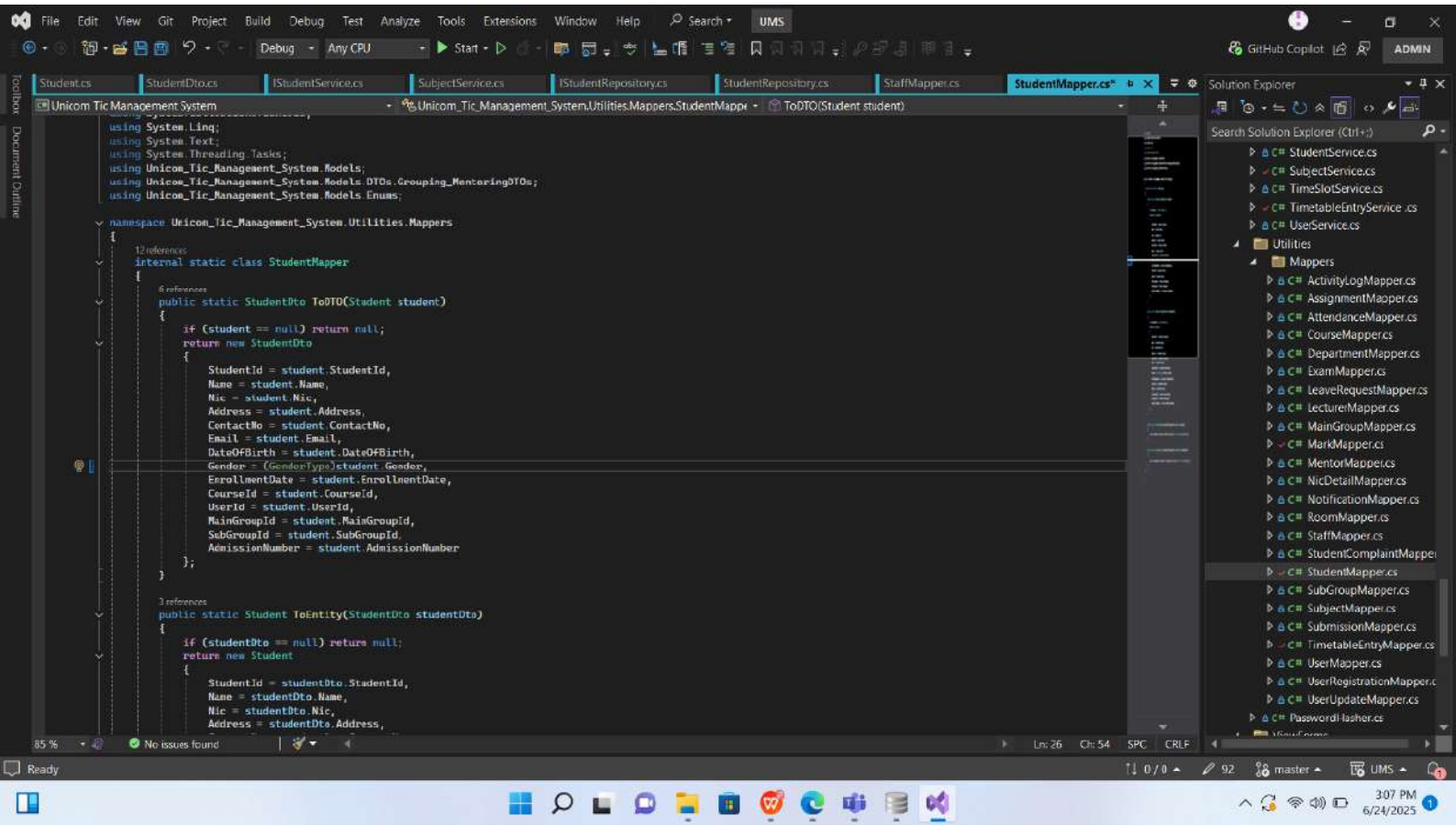

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
SignUpOptionsForm.cs*
Program.cs
DatabaseSeeder.cs
Migration.cs
Unicom Tic Management Syst
Unicom_Tic_Management_Sys
SeedAdminUser()
ate const string ADMIN_USERNAME = "admin@unicom.com";
ate const string ADMIN_PASSWORD = "adminpassword";
ate const string ADMIN_NIC = "123456789V";
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.");
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
85 %
No issues found
Ln: 55
Ch: 1
SPC
CRLF
This item does not support previewing
master
UMS
304 PM
6/24/2025

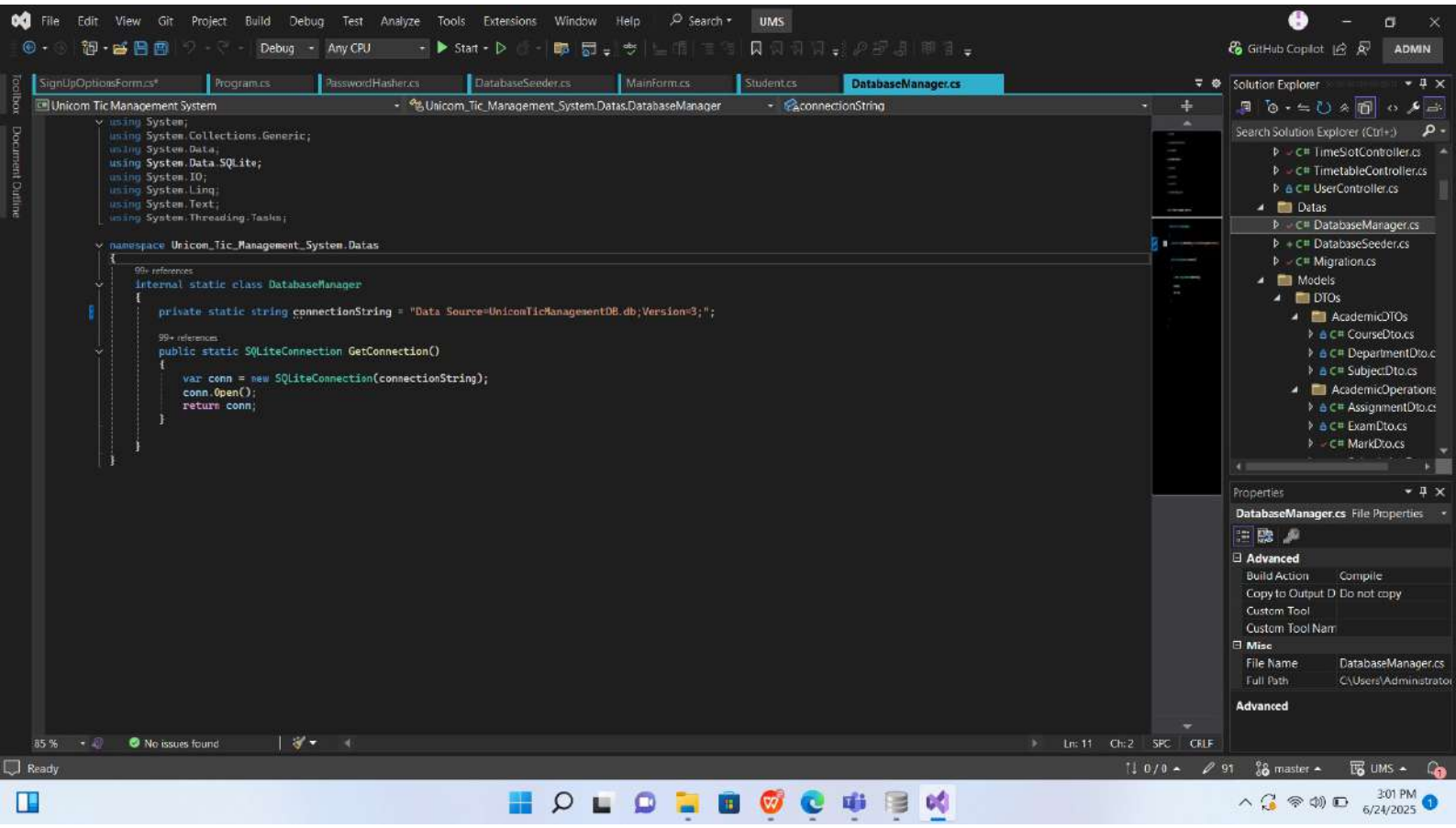

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
SignUpOptionsForm.cs*
Program.cs
DatabaseSeeder.cs
Migration.cs
Unicom Tic Management Syst
Unicom_Tic_Management_Sys
SeedAdminUser()
ate const string ADMIN_USERNAME = "admin@unicom.com";
ate const string ADMIN_PASSWORD = "adminpassword";
ate const string ADMIN_NIC = "123456789V";
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.");
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
85 %
No issues found
Ln: 55
Ch: 1
SPC
CRLF
Ready
master
UMS
303 PM
6/24/2025

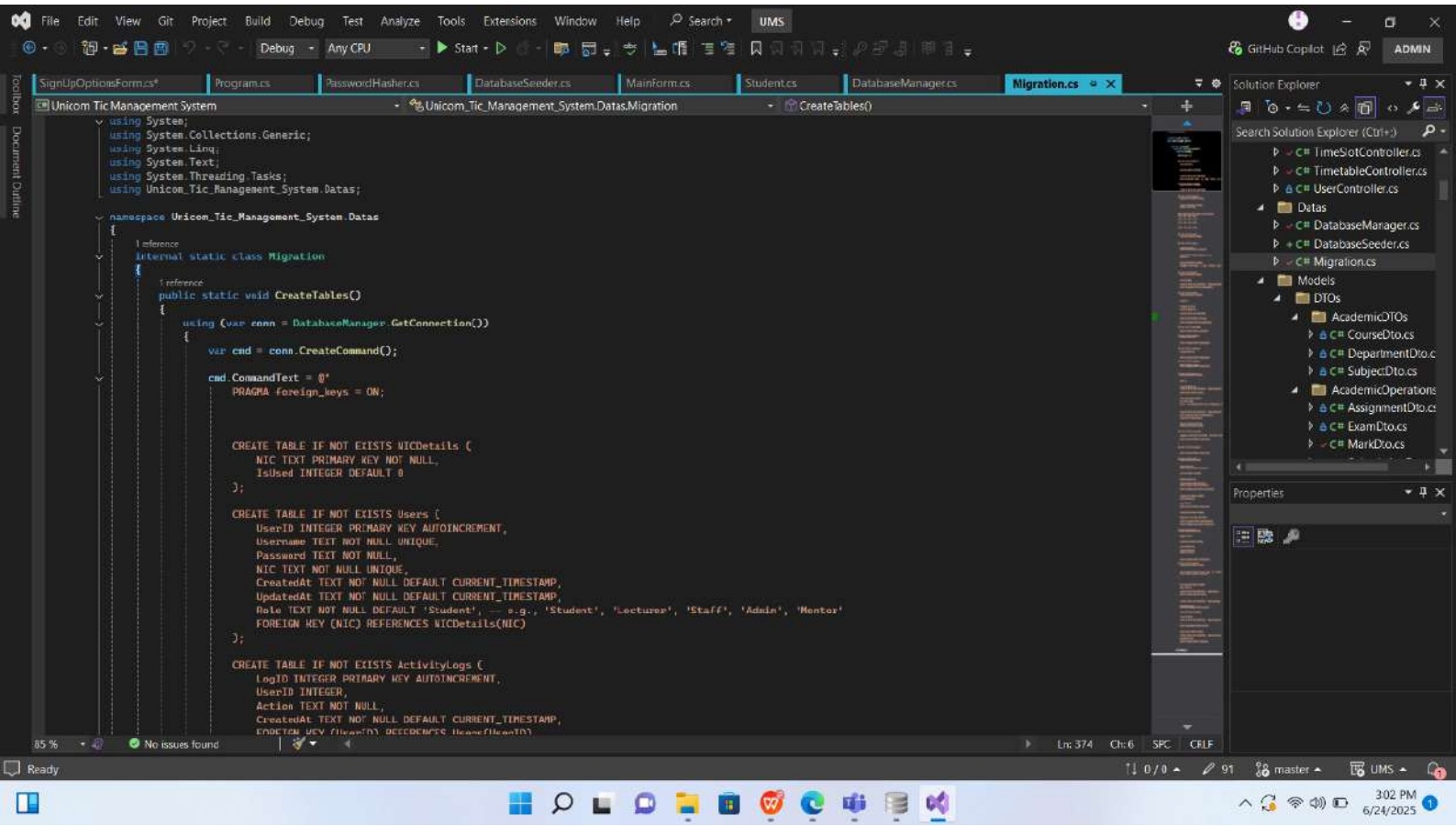

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
DatabaseSeeder.cs
Migration.cs
MainForm.cs
Student.cs
DatabaseManager.cs
GenderType.cs
StudentDto.cs
IStudentService.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Services.Interfaces.IStudentServi
GetStudentsByGender(GenderType gender)
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);
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Repositories
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
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
85 %
No issues found
Ln: 22
Ch: 56
SPC
CRLF
Ready
0 / 0
91
master
UMS
305 PM
6/24/2025

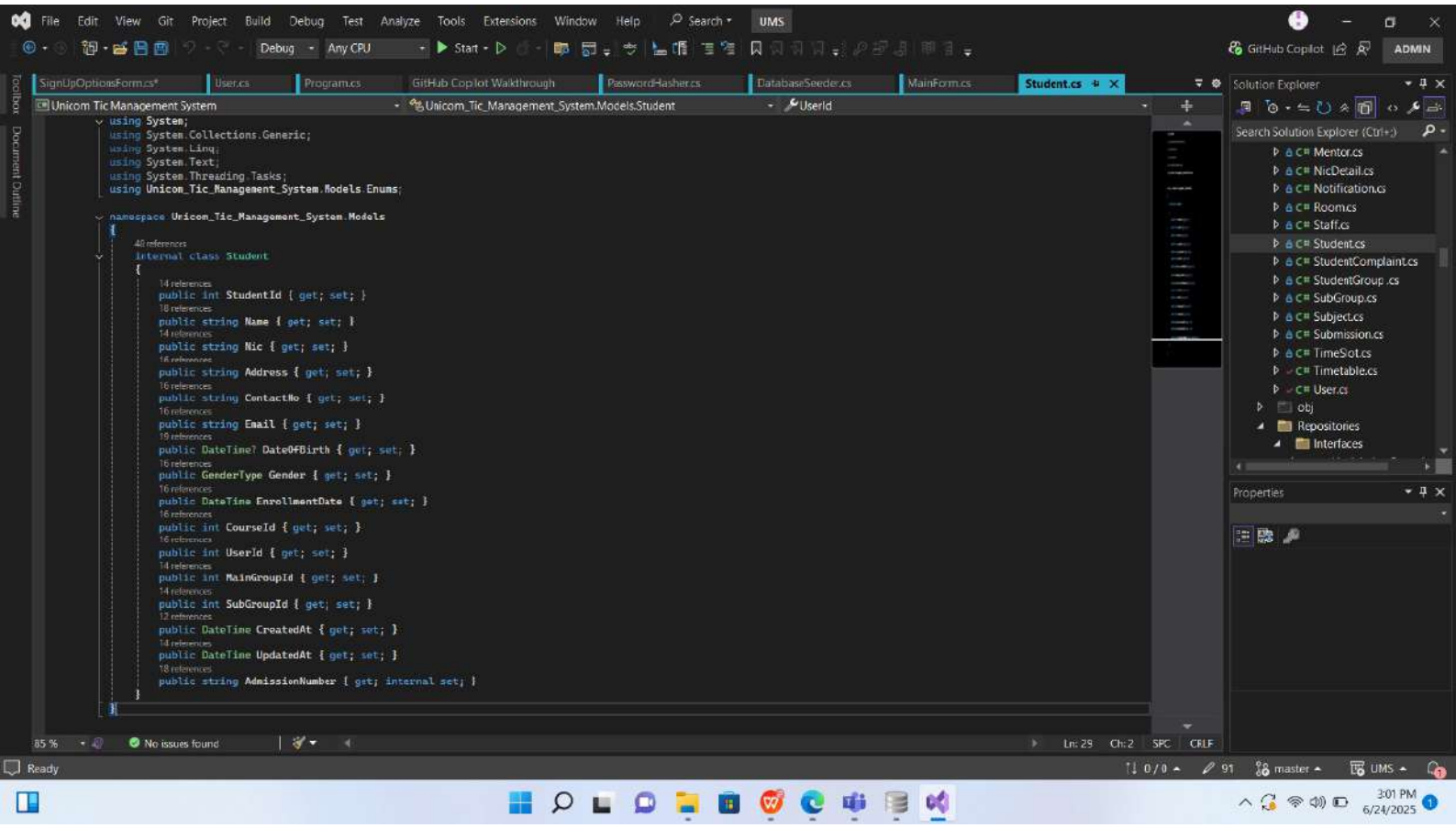

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
DatabaseSeeder.cs
Student.cs
GenderType.cs
StudentDto.cs
IStudentService.cs
SubjectService.cs
IStudentRepository.cs
StudentRepository.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Repositories.StudentRepository
GetAllStudents()
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
{
3 references
internal class StudentRepository : IStudentRepository
{
3 references
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"));
Solution Explorer
Search Solution Explorer (Ctrl+;)
ISubmissionRepository.cs
ITimeSlotRepository.cs
ITimetableRepository.cs
IUserRepository.cs
ActivityLogRepository.cs
AdmissionNumberRepository
AssignmentRepository.cs
AttendanceRepository.cs
BackupLogRepository.cs
CourseRepository.cs
CourseSubjectRepository.cs
DepartmentRepository.cs
EmployeeIdRepository.cs
ExamRepository.cs
GroupMentorRepository.cs
GroupSubjectRepository.cs
LeaveRequestRepository.cs
LecturerRepository.cs
LecturerStudentRepository.cs
LectureSubjectRepository.cs
MainGroupRepository.cs
MarkRepository.cs
MentorRepository.cs
NicDetailsRepository.cs
RoomRepository.cs
StaffRepository.cs
StudentRepository.cs
SubGroupRepository.cs
SubjectRepository.cs
SubmissionRepository.cs
TimeSlotRepository.cs
TimetableRepository.cs
85 %
No issues found
Ln: 605
Ch: 10
SPC
CRLF
Ready
0 / 0
91
master
UMS
306 PM
6/24/2025

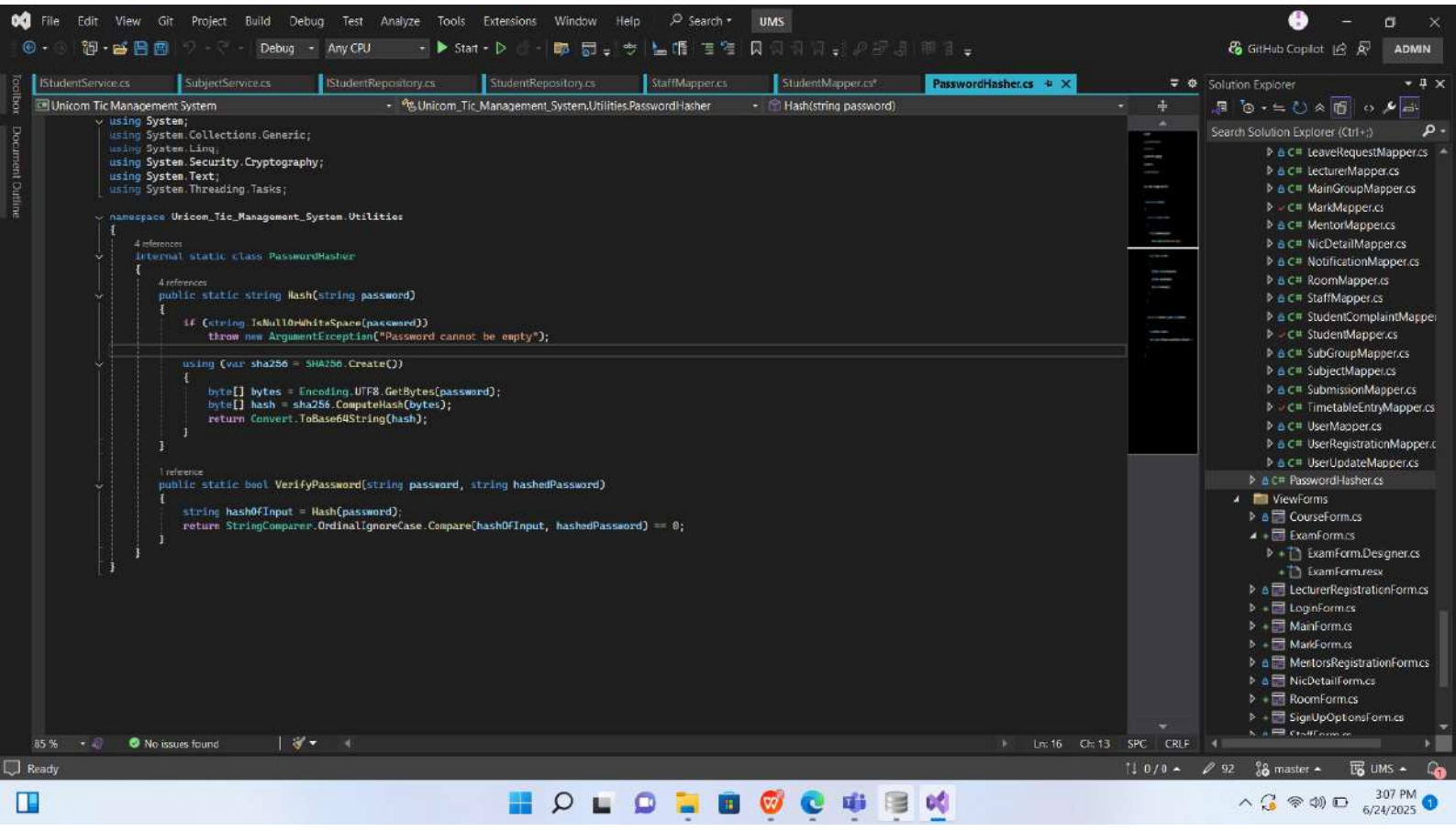

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
DatabaseSeeder.cs
Migration.cs
Student.cs
GenderType.cs
StudentDto.cs
IStudentService.cs
SubjectService.cs
IStudentRepository.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Repositories.Interfaces.IStudent
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);
Student GetStudentByUserId(int userId);
List<Student> GetStudentsByCourseId(int courseId);
List<Student> GetStudentsByMainGroupId(int mainGroupId);
List<Student> GetStudentsBySubGroupId(int subGroupId);
List<Student> GetStudentsByGender(GenderType gender);
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
AdmissionNumberRepository
AssignmentRepository.cs
AttendanceRepository.cs
BackupLogRepository.cs
CourseRepository.cs
CourseSubjectRepository.cs
DepartmentRepository.cs
EmployeeIdRepository.cs
ExamRepository.cs
GroupMentorRepository.cs
85 %
No issues found
Ln: 26
Ch: 1
SPC
CRLF
Ready
Microsoft Store
0 / 0
91
master
UMS
306 PM
6/24/2025

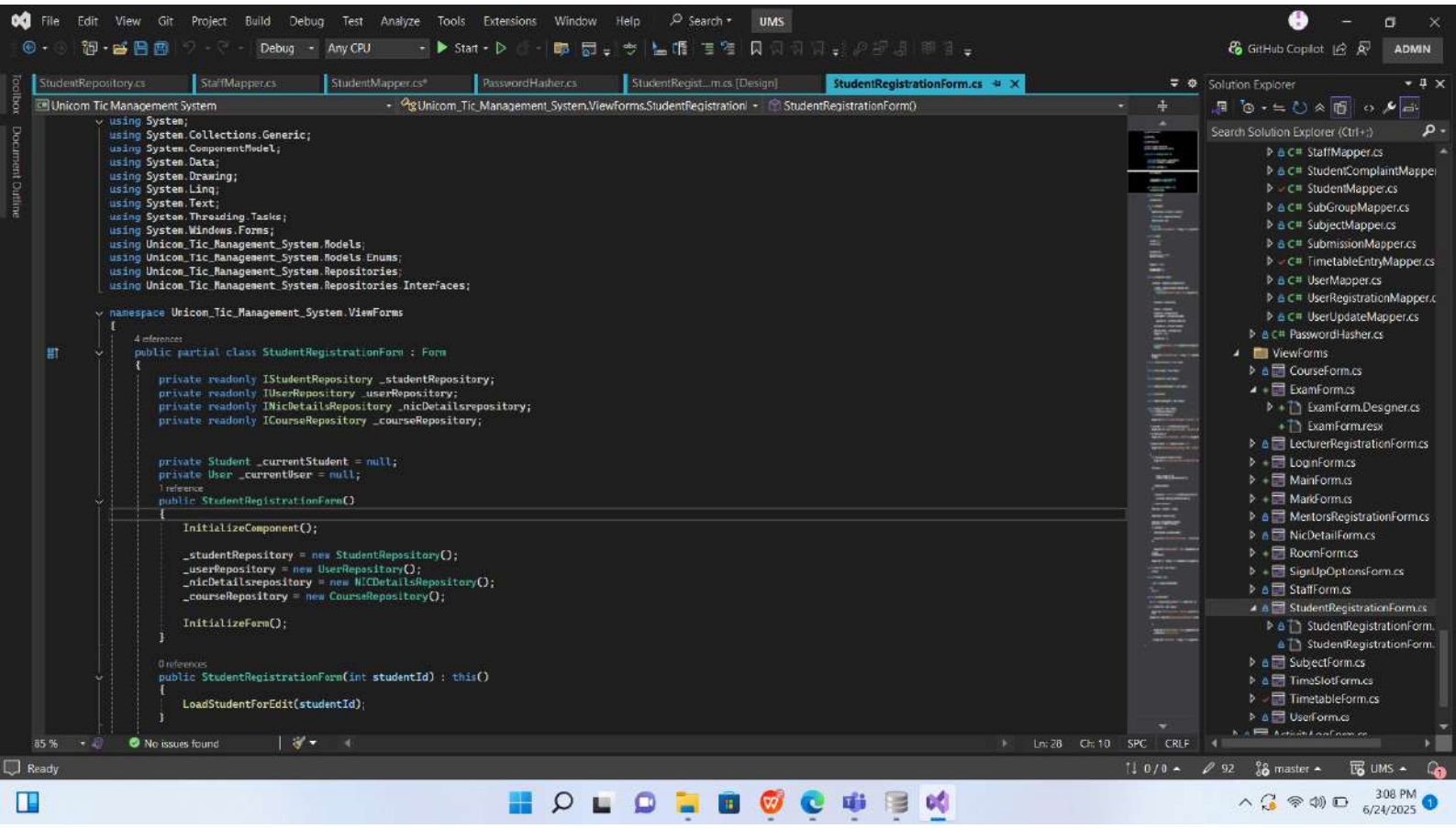

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
Student.cs
StudentDto.cs
IStudentService.cs
SubjectService.cs
IStudentRepository.cs
StudentRepository.cs
StaffMapper.cs
StudentMapper.cs*
Unicom Tic Management System
Unicom_Tic_Management_System.Utilities.Mappers.StudentMappe
ToDTO(Student student)
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Unicom_Tic_Management_System.Models;
using Unicom_Tic_Management_System.Models.DTOs.Grouping_MenteringDTOs;
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,
Solution Explorer
Search Solution Explorer (Ctrl+;)
StudentService.cs
SubjectService.cs
TimeSlotService.cs
TimetableEntryService .cs
UserService.cs
Utilities
Mappers
ActivityLogMapper.cs
AssignmentMapper.cs
AttendanceMapper.cs
CourseMapper.cs
DepartmentMapper.cs
ExamMapper.cs
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
85 %
No issues found
Ln: 26
Ch: 54
SPC
CRLF
Ready
0 / 0
92
master
UMS
3:07 PM
6/24/2025

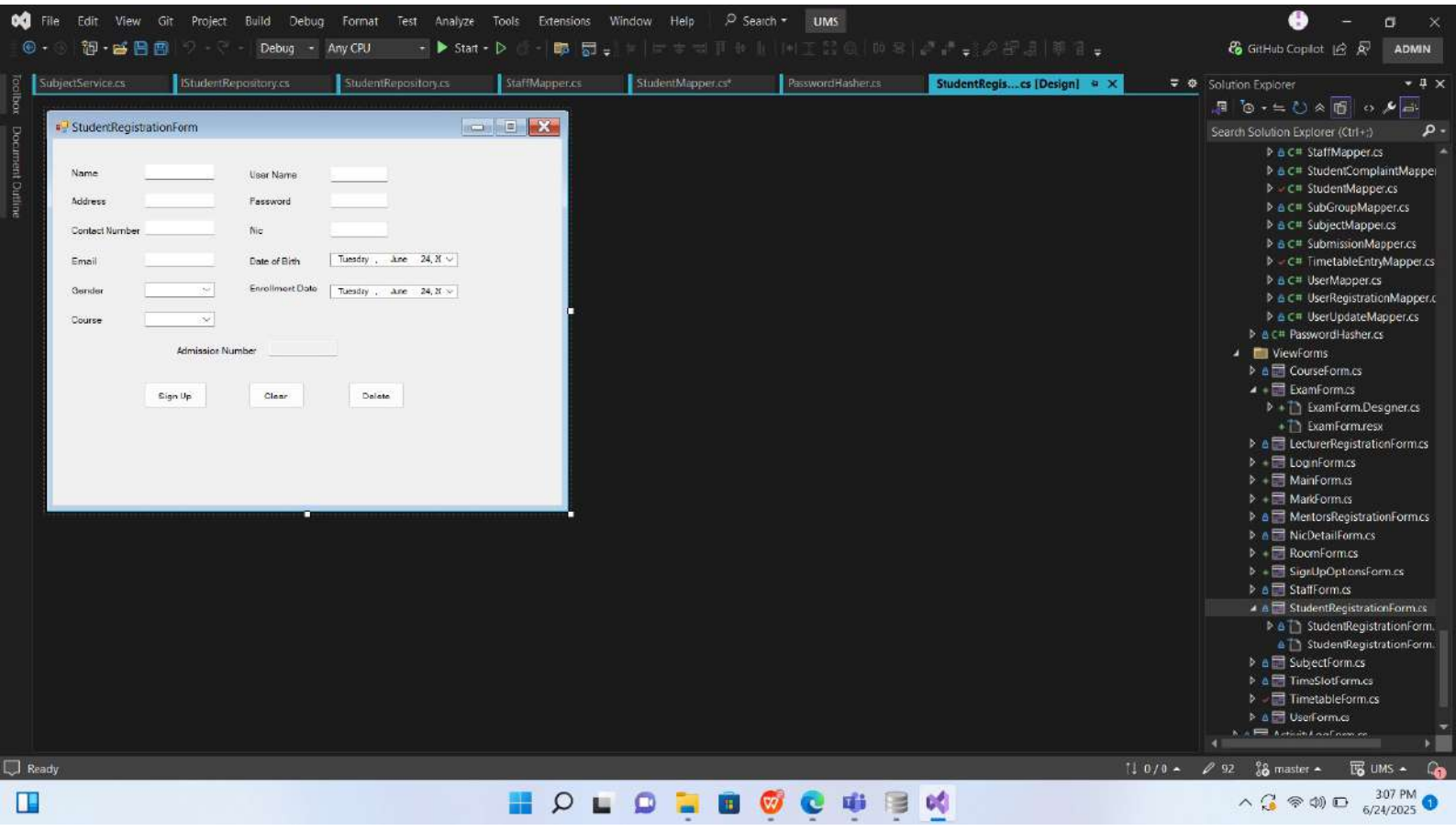

File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
ADMIN
SignUpOptionsForm.cs*
Program.cs
PasswordHasher.cs
DatabaseSeeder.cs
MainForm.cs
Student.cs
DatabaseManager.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Datas.DatabaseManager
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
0 / 0
91
master
UMS
3:01 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
302 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*
User.cs
Program.cs
GitHub Copilot Walkthrough
PasswordHasher.cs
DatabaseSeeder.cs
MainForm.cs
Student.cs
Unicom Tic Management System
Unicom_Tic_Management_System.Models.Student
UserId
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
{
48 references
internal class Student
{
14 references
public int StudentId { get; set; }
18 references
public string Name { get; set; }
14 references
public string Nic { get; set; }
16 references
public string Address { get; set; }
16 references
public string ContactNo { get; set; }
16 references
public string Email { get; set; }
19 references
public DateTime? DateOfBirth { get; set; }
16 references
public GenderType Gender { get; set; }
16 references
public DateTime EnrollmentDate { get; set; }
16 references
public int CourseId { get; set; }
16 references
public int UserId { get; set; }
14 references
public int MainGroupId { get; set; }
14 references
public int SubGroupId { get; set; }
12 references
public DateTime CreatedAt { get; set; }
14 references
public DateTime UpdatedAt { get; set; }
18 references
public string AdmissionNumber { get; internal set; }
}
}
Solution Explorer
Search Solution Explorer (Ctrl+;)
Mentor.cs
NicDetail.cs
Notification.cs
Room.cs
Staff.cs
Student.cs
StudentComplaint.cs
StudentGroup.cs
SubGroup.cs
Subject.cs
Submission.cs
TimeSlot.cs
Timetable.cs
User.cs
obj
Repositories
Interfaces
Properties
85 %
No issues found
Ln: 29
Ch: 2
SPC
CRLF
Ready
master
UMS
3:01 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
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
0 / 0
92
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
StudentRepository.cs
StaffMapper.cs
StudentMapper.cs*
PasswordHasher.cs
StudentRegist...m.cs [Design]
StudentRegistrationForm.cs
Unicom Tic Management System
Unicom_Tic_Management_System.ViewForms.StudentRegistrationF
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
0 / 0
92
master
UMS
308 PM
6/24/2025


File Edit View Git Project Build Debug Format Test Analyze Tools Extensions Window Help
Search
UMS
Debug
Any CPU
Start
GitHub Copilot
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
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
Ready
master
UMS
3:07 PM
6/24/2025