Smart Campus Portal

Term Project Report

May 27, 2025

Anood Tayyeba Imtiaz (01-135232-010) and
Tayyaab Zahoor (01-135232-070)
Course: Web Systems and Technologies

Contents

1. Executive Summary 1
2. Introduction 1
3. Methodology and Design.....2
4. Implementation Details.....2
5. Testing and Validation.....3
6. Screenshots & Walk-through.....4
7. Discussion.....6
8. Conclusion & Future Work.....6

1. Executive Summary

This report describes the design and development of the Smart Campus Portal, a web-based system for university automation. The portal allows three types of users—students, faculty, and administrators—to log in and use services specific to their roles. Key features include course registration, assignment submission and grading, announcements, and administrative user and course management. The portal uses ASP.NET Web Forms, LINQ-to-SQL, Bootstrap, and jQuery to deliver a simple and responsive user experience.

2. Introduction

Universities often use multiple systems to handle student records, courses, and administrative tasks. This can make it hard for students and staff to find information. The Smart Campus Portal aims to solve this by providing one platform where:

- Students can view their profile, register for courses, submit assignments, and check grades and fees.

- Faculty can upload course content, grade work, send announcements, and view attendance.

- Administrators can manage users, manage courses, and generate simple reports.

3. Methodology and Design

Technology Choices:

- Frontend: HTML5, CSS3, Bootstrap 5, JavaScript, jQuery.

- Backend: ASP.NET Web Forms (C#), LINQ-to-SQL for data access.

- Database: SQL Server with tables for Users, Students, Faculty, Courses, Assignments, Submissions, Grades, Announcements, and Attendance.

System Structure:

- Presentation Layer: ASPX pages with code-behind files for each role's dashboard and shared user controls for navigation.

- Business Logic: C# methods handle login, session management, and CRUD operations.

- Data Layer: A .dbml file defines a DataClassesDataContext for LINQ-to-SQL mapping.

- Session Management: Uses ASP.NET Session to track logged-in user and role.

4. Implementation Details

User Authentication & Role Management:

- Login page lets users select their role (Student, Faculty, Admin) and enter credentials.

- On successful login, the system stores UserID and Role in session and redirects to the correct dashboard.

- Each page checks session values in Page_Load to prevent unauthorized access.

Student Portal:

- Profile & Academic Record: Displays user details and simulated grade report.

- Course Registration: Shows a list of courses with checkboxes. Students can select courses and submit.

- Assignments & Grades: Students upload files for assignments and view graded marks.

- Fees: A simple table shows fee details and payment status.

Faculty Portal:

- Course Content & Assignments: Faculty can upload files and set due dates.

- Grading: Faculty view student submissions and enter marks.

- Announcements: Faculty can post messages that appear on student dashboards.

- Attendance Reports: A grid lists students and allows marking attendance for a date.

Admin Panel:

- Manage Users: Admin can add or remove students and faculty via forms and view lists.

- Manage Courses: Admin can create, update, or delete courses using a data grid.

- Reports: A summary tab shows basic counts of users, courses, and registrations.

5. Testing and Validation

Functional Testing:

- Verified login for each role and proper redirection.

- Tested course registration workflow and data persistence.

- Uploaded and retrieved assignment files; checked file type validation.

- Graded submissions and confirmed grade display on student side.

Security and Error Handling:

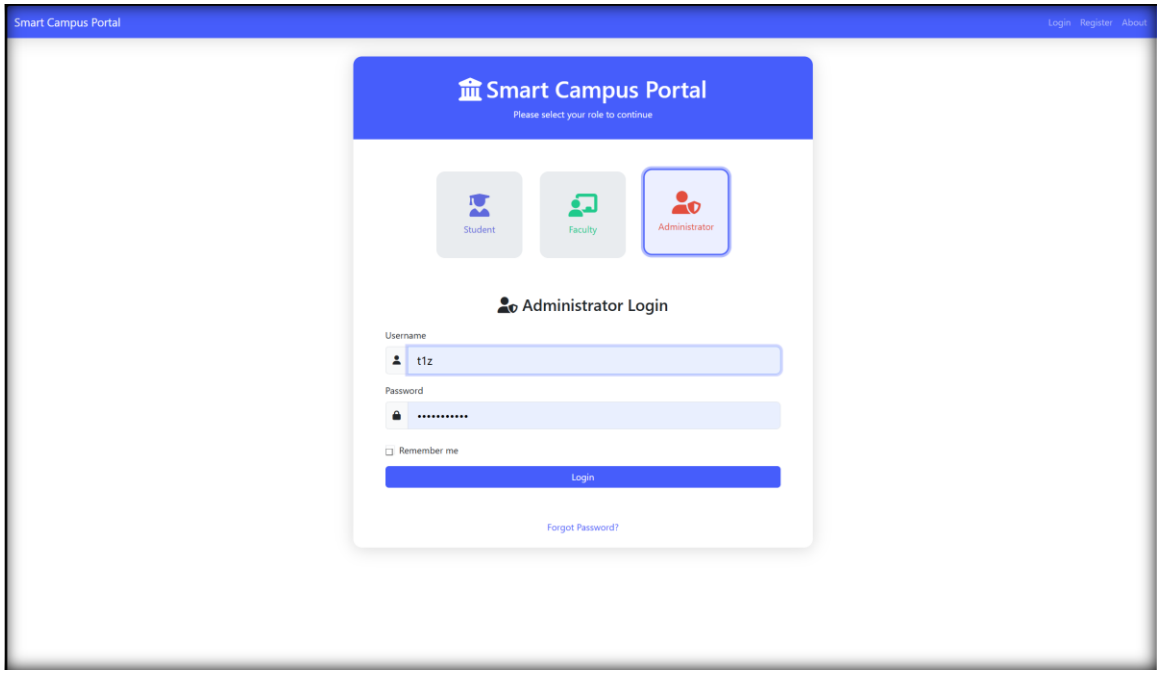
- Checked that pages redirect to login if session is missing.

- Basic input validation on forms (required fields, file types).

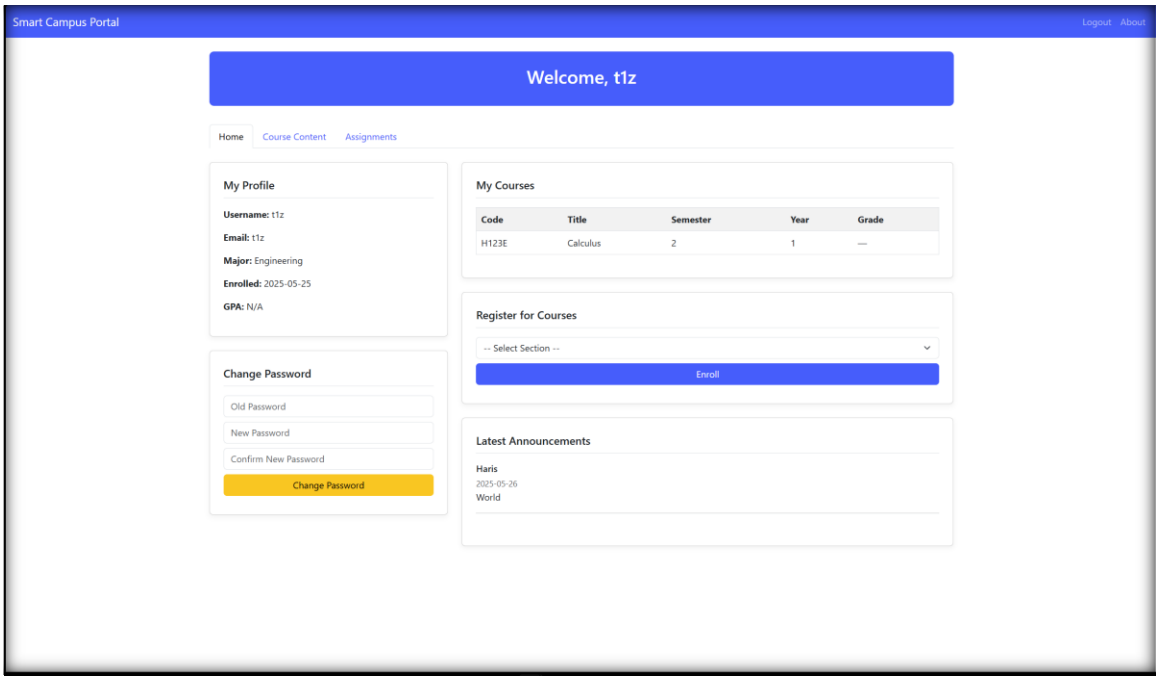
- Handled LINQ exceptions with try-catch blocks and user-friendly messages.

6. Screenshots & Walk-through

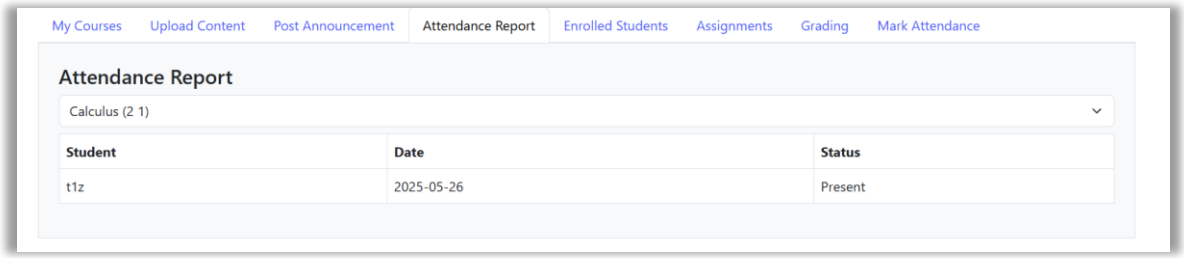
1. Login Page:



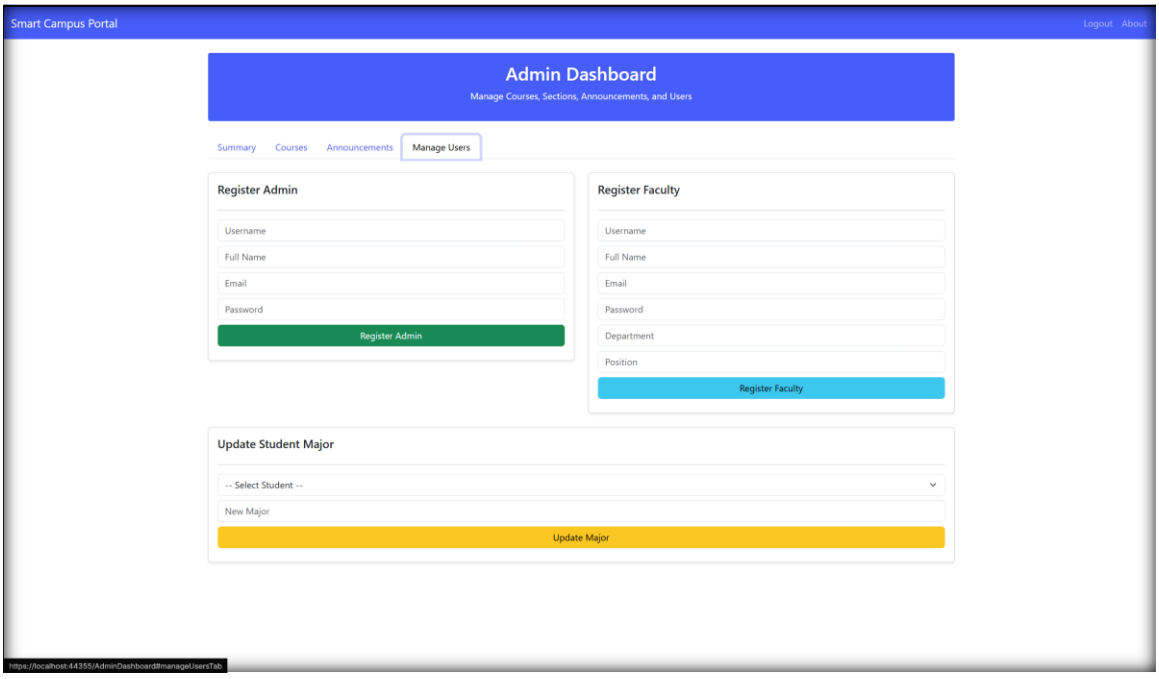
2. Student Dashboard:



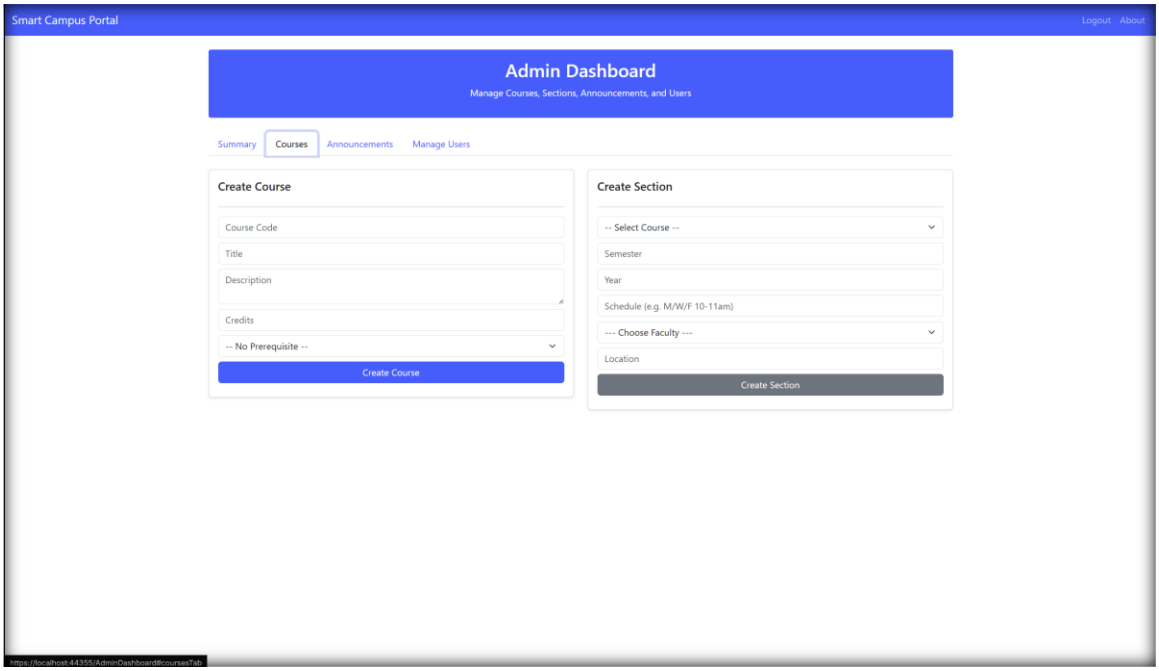
3. Faculty Dashboard:



4. Admin Manage Users:



5. Admin Manage Courses:



7. Discussion

The project meets the main requirements by offering role-based dashboards and core features for each user type. Using ASP.NET Web Forms allowed quick page events and session handling, while LINQ-to-SQL simplified data access. Some areas were challenging, such as preserving UI state on postback and validating file uploads. These were solved with JavaScript for tabs and server-side checks for files.

8. Conclusion & Future Work

The Smart Campus Portal is a functioning proof of concept that centralizes key campus services for students, faculty, and administrators.

Future Work: We will improve the UI and add the features that don't exist, such as two-factor authentication, real-time notifications, and advanced reporting dashboards.