BULMS Project Report

MUHAMMAD AHMED AMJAD	01 – 135232-051
UMAMA KHALID	01 – 135232-104
HAMZA ALI KAZMI	01 – 135232-096
AREEBA WAHAB	01 – 135232-011

DATABASE: SKAHMEDD\SQLEXPRESS

PROJECT: BULMS (A Basic University Learning Management System

PLATFORM: ASP.NET Web Forms, SQL Server

1. INTRODUCTION

BULMS is a web-based Learning Management System designed for universities to manage academic activities for students, faculty, and administrators. The system streamlines course registration, assignment submission, grading, announcements, and fee management, with a robust semester-based filtering system.

2. SYSTEM DESIGN

2.1. User Roles

- **Student:** Registers for courses, submits assignments, views grade, pays fees.
- **Faculty:** Manages course content, uploads assignments, grades submissions, posts announcements.
- **Admin:** Manages users, courses, semesters, course offerings, and generates reports.

2.2. Main Features

- **Authentication:** Secure login with role-based dashboards.
- **Course Registration:** Students register for courses, with prerequisite checks and semester filtering.
- **Assignment Management:** Faculty uploads assignments; students submit work; faculty grades submissions.
- **Announcements:** Faculty and admin can post announcements visible to relevant users.

- **Fee Management:** Students can view and pay fees; admin can generate fee reports.
- **Semester System:** All academic activities are filtered by the selected semester.
- **Reporting:** Admin can generate reports on course registrations, grades, and fees.

3. System Architecture

3.1. Technology Stack

- **Frontend:** ASP.NET Web Forms (ASPX, C# code-behind)
- **Backend:** SQL Server (Database: SKAHMED)
- **Data Access:** LINQ to SQL (.dbml)
- **File Storage:** Assignment files stored on server

3.2. Database Design

Key Tables:

- Users (UserID, Name, Email, PasswordHash, RoleID, etc.)
- Roles (RoleID, RoleName)
- Courses (CourseID, CourseName, PrerequisiteID, etc.)
- Semesters (SemesterID, Name, StartDate, EndDate)
- CourseOfferings (OfferingID, CourseID, SemesterID, FacultyID)
- CourseRegistrations (RegistrationID, StudentID, OfferingID)
- Assignments (AssignmentID, OfferingID, Title, DueDate, FilePath)
- Submissions (SubmissionID, AssignmentID, StudentID, FilePath, Grade)
- Announcements (AnnouncementID, UserID, Message, Date)
- Fees (FeeID, StudentID, Amount, Status, SemesterID)

Relationships:

- Users have Roles.
- Courses can have prerequisites (self-referencing).
- Courses are offered per semester (CourseOfferings).
- Students register for course offerings.
- Assignments are linked to course offerings.
- Submissions are linked to assignments and students.

3.3. Application Structure

- **Master Page:** Contains navigation and a global semester dropdown for filtering.
- **Role-based Pages:** Separate dashboards and features for students, faculty, and admin.
- **Data Access Layer:** LINQ to SQL for CRUD operations.
- **Validation:** Server-side and client-side validation for forms and file uploads.

4. System Architecture

4.1. High-Level Architecture Diagram

[User (Student/Faculty/Admin)]

[ASP.NET Web Forms Application]

[LINQ to SQL Data Access Layer]

[SQL Server Database (SKAHMED)]

- **Presentation Layer: ASP.NET Web Forms (ASPX pages, Master Page for navigation and semester selection)**
- **Business Logic Layer: C# code-behind, role-based logic, validation**
- **Data Access Layer: LINQ to SQL (.dbml)**
- **Database Layer: SQL Server (SKAHMED)**

5. Key Implementation Details

- **Global Semester Selection:**

The master page includes a semester dropdown.
All student and faculty features (course registration, assignments, grades)
are filtered by the selected semester.

- **Course Registration with Prerequisites:**

When registering, the system checks if the student has completed required prerequisites.

- **Assignment Submission:**

Students can upload files (with type/size validation). Faculty can download and grade submissions.

- **Admin Management:**

Admin can add/edit/delete users, courses, semesters, and assign faculty to courses per semester.

- **Reporting:**

Admin can generate reports on registrations, grades, and fees, filtered by semester.

6. Key Challenges and Solutions

6.1. Semester-Based Filtering

Challenge: Ensuring all features (registration, assignments, grades) are filtered by the selected semester.

Solution: Implemented a global semester dropdown in the master page. Used session variables to store the selected semester and filtered all queries accordingly.

6.2. Prerequisite Enforcement

Challenge: Preventing students from registering for courses without completing prerequisites.

Solution: Added prerequisite checks in the course registration logic, using SQL joins to verify completion.

6.3. Data Consistency and Foreign Keys

Challenge: Handling foreign key constraints, especially when deleting courses or users with dependencies.

Solution: Implemented checks and restricted deletions in the admin panel. Provided user-friendly error messages.

6.4. File Uploads and Validation

Challenge: Ensuring only valid files are uploaded for assignments (type, size).

Solution: Added server-side and client-side validation for file uploads. Stored files in organized server directories.

6.5. LINQ to SQL Model Updates

Challenge: Keeping the .dbml model in sync with database schema changes.

Solution: Regularly refreshed the LINQ to SQL model after any schema update.

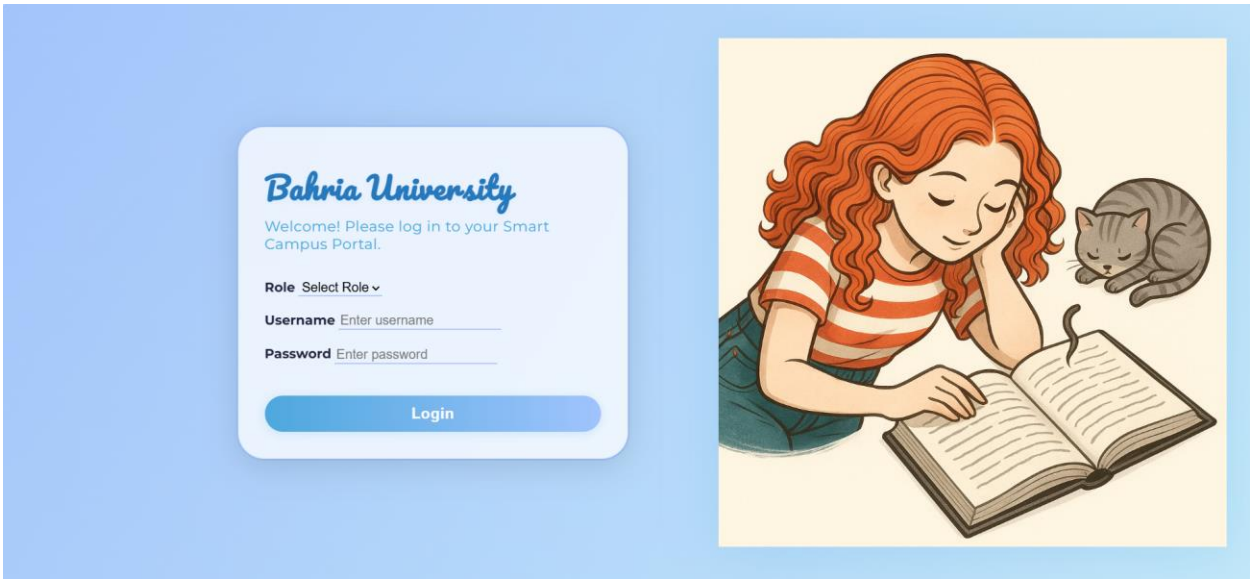
6.6. Role-Based Access Control

Challenge: Ensuring users only access features relevant to their role.

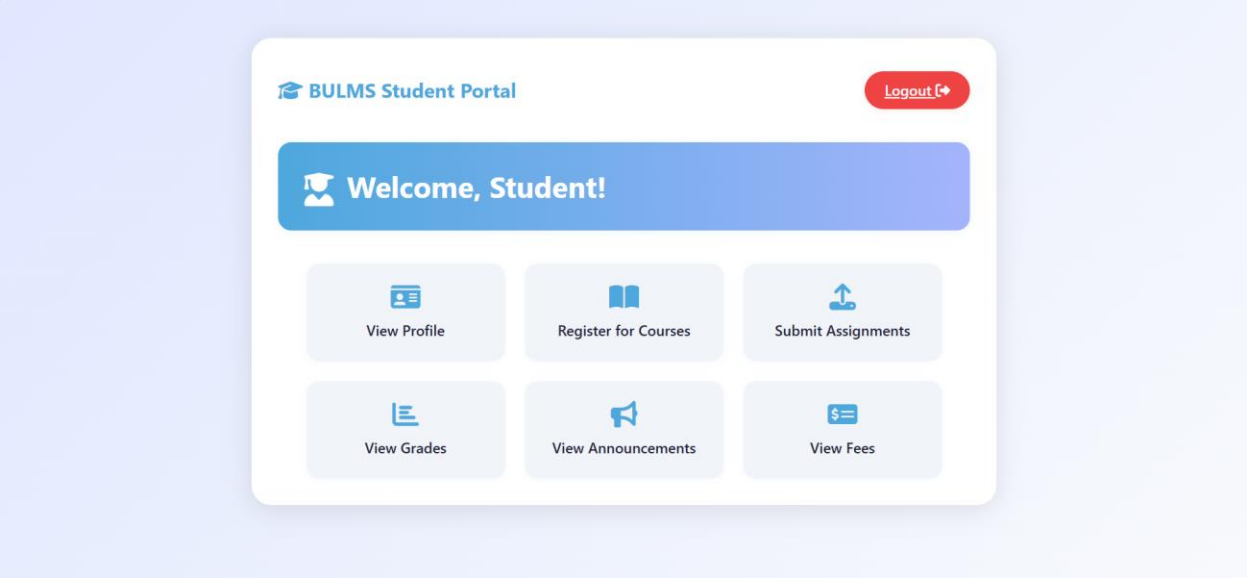
Solution: Implemented role checks on each page and redirected unauthorized users.

7. SCREENSHOTS

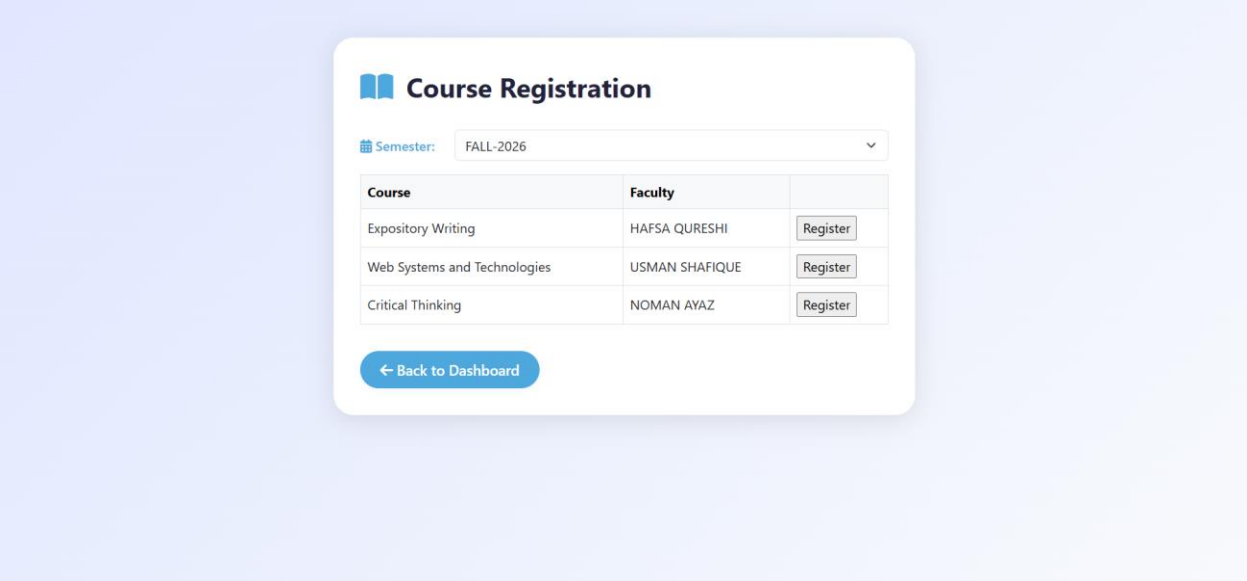
Login



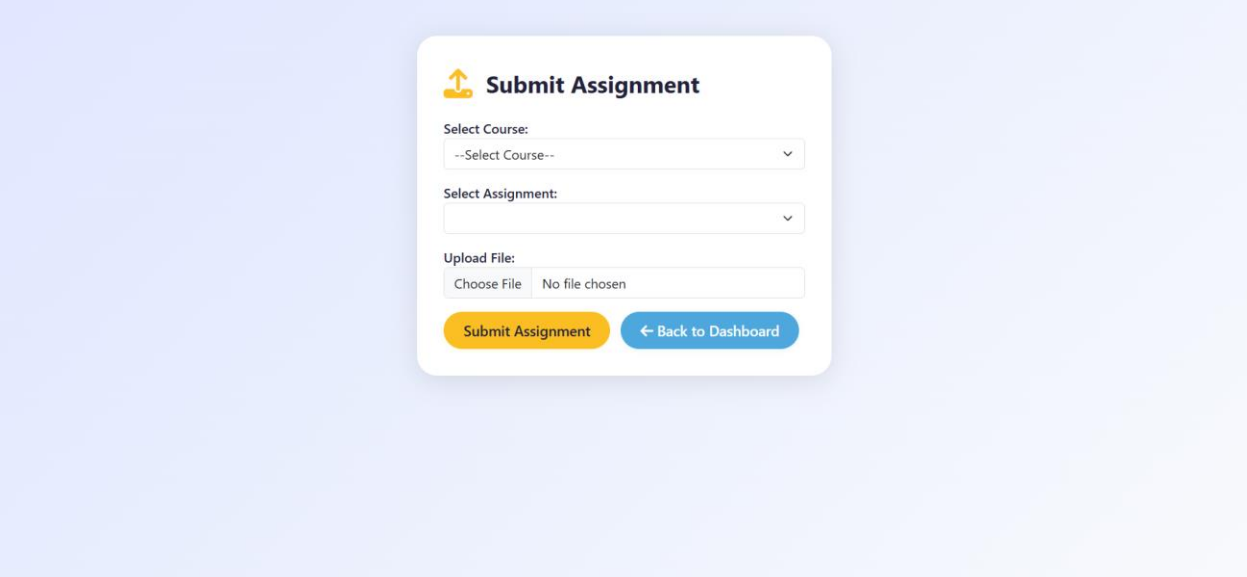
Student Dashboard:



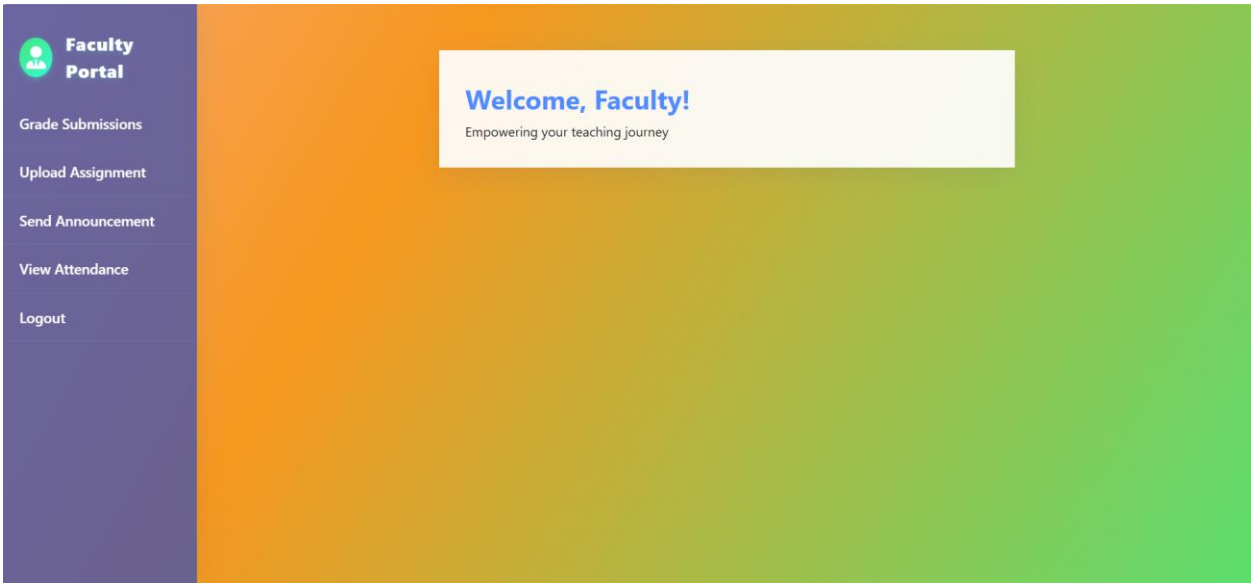
Course Registration:



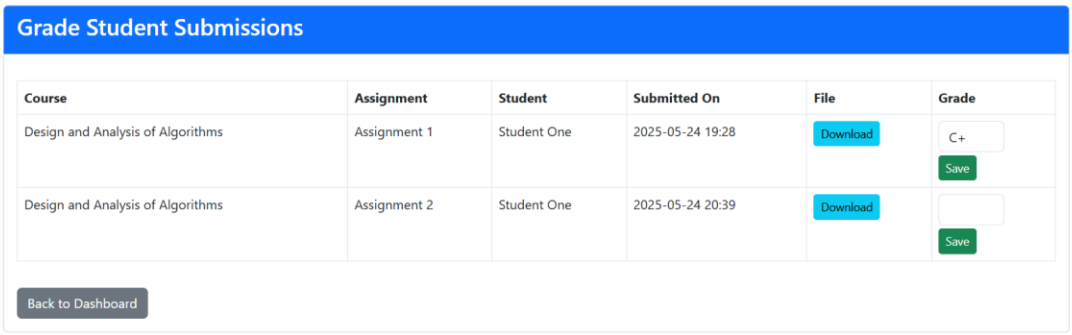
Assignment Submission:



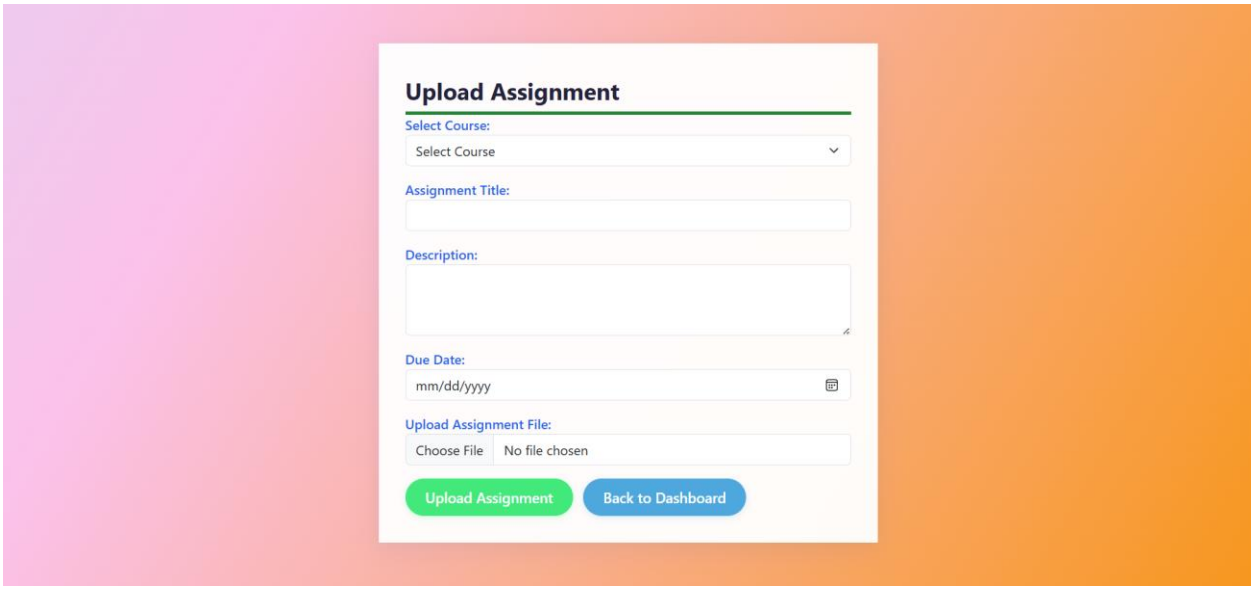
Faculty Dashboard:



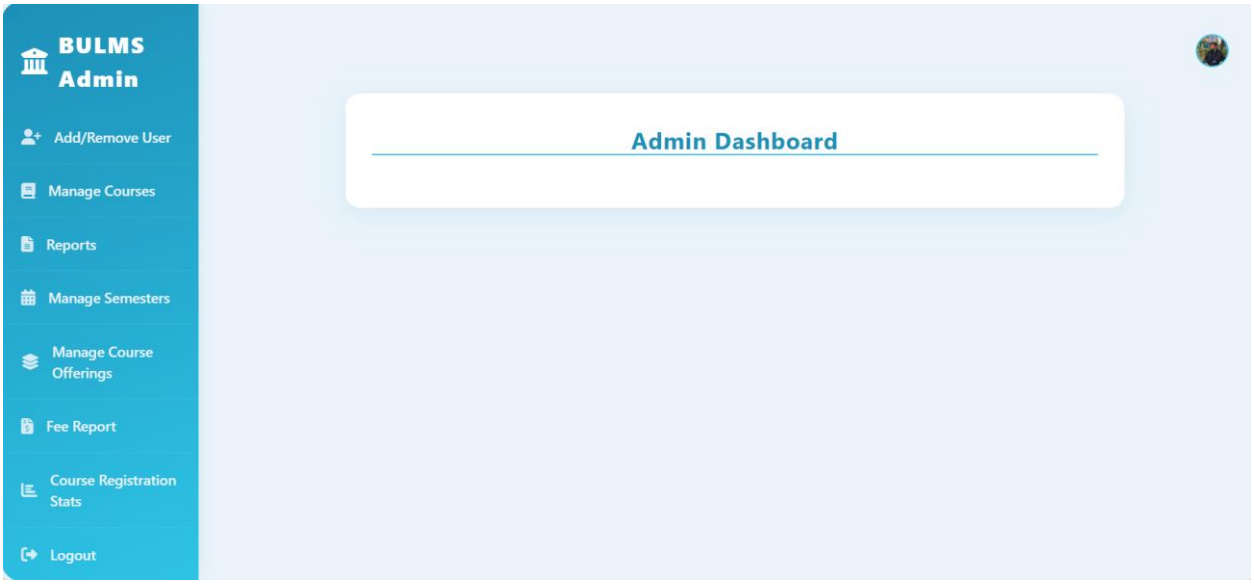
Grade Submission:



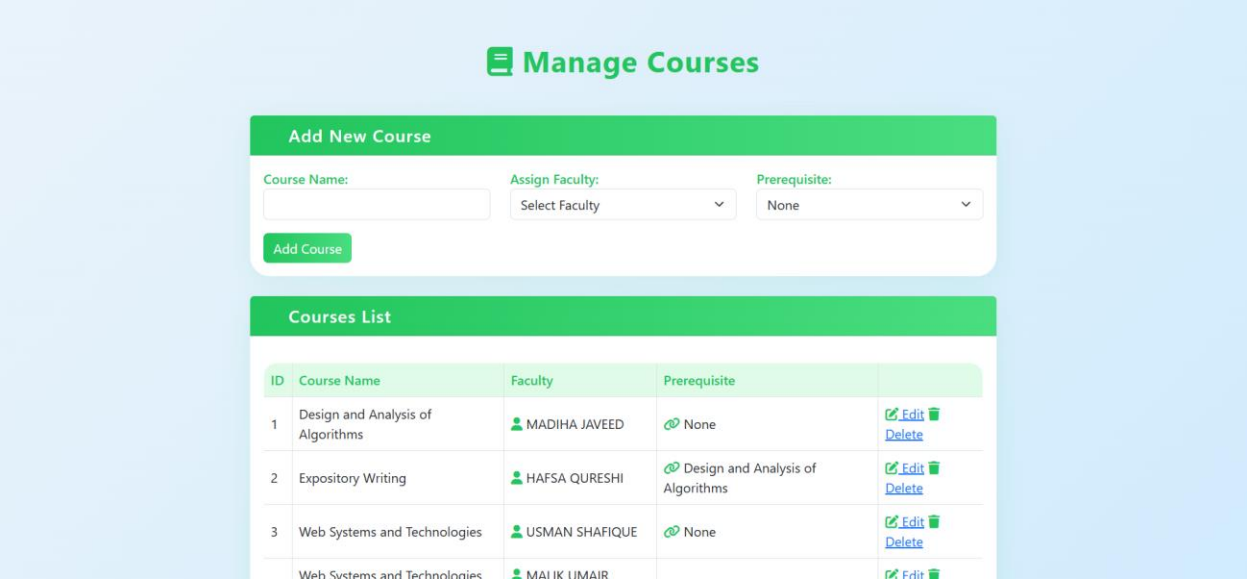
Upload Assignment:



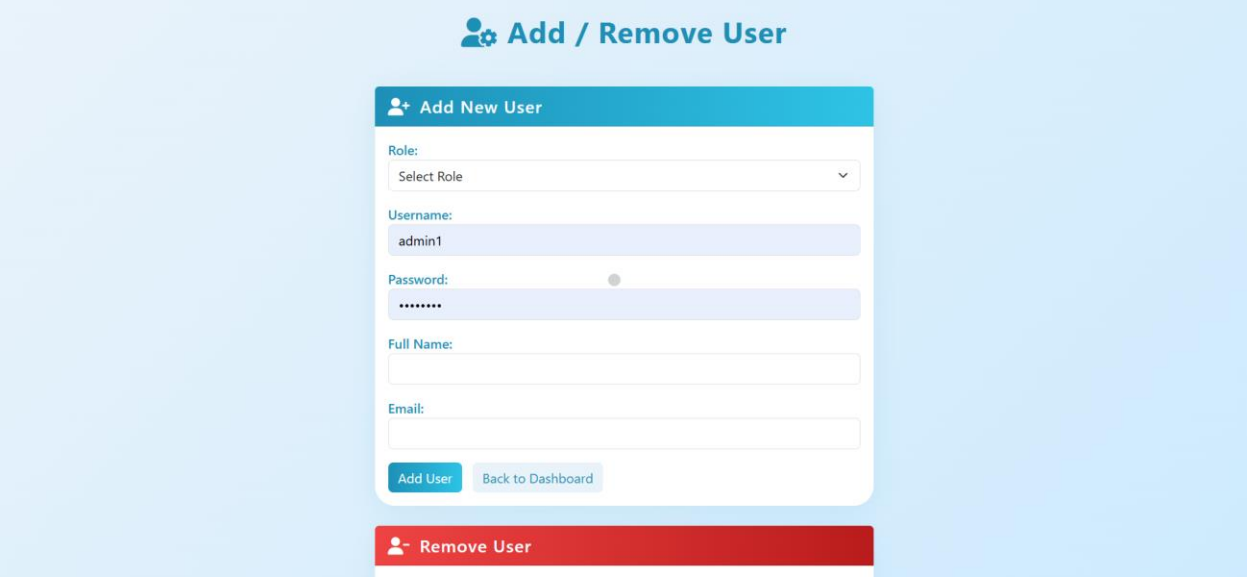
Admin Panel:



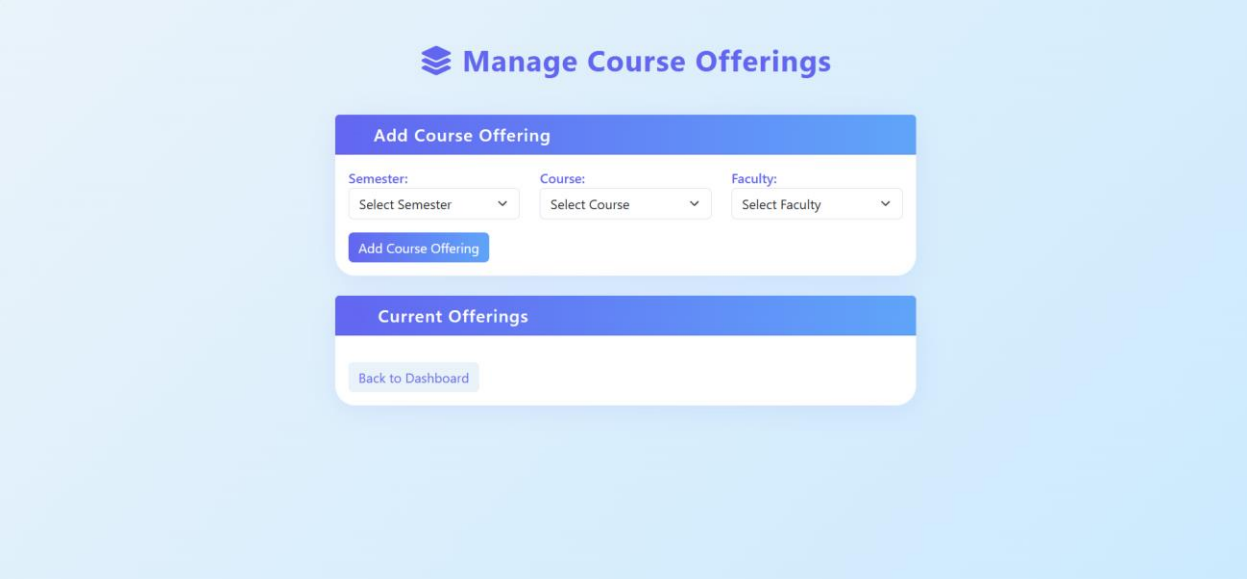
Manage Courses:



Add/ Remove User:



Manage Course Offerings:



8. Tools and Technologies Used

- **Frontend:** ASP.NET Web Forms (C#, .aspx, .aspx.cs)
- **Backend:** SQL Server (Database: SKAHMED)
- **Data Access:** LINQ to SQL (.dbml)
- **IDE:** Visual Studio
- **File Storage:** Server file system for assignment uploads
- **Other:** HTML, CSS, JavaScript (for validation and UI)

9. Future Improvements

- Implement notification/email system for deadlines and announcements.
- Add RESTful APIs for mobile app integration.
- Enhance UI/UX with modern frameworks (e.g., Bootstrap).
- Implement advanced reporting and analytics.

10. Conclusion

BULMS is a robust, semester-based university management system supporting students, faculty, and admin. The project demonstrates best practices in ASP.NET Web Forms, SQL Server database design, and user-centric feature implementation. Key challenges were addressed with practical solutions, resulting in a scalable and maintainable LMS.