

**Project Report**

Database Management System (BM)

Submitted to: Miss Quratulain Faizan & Sir Haider Imam Jaffery

Submitted by:

FA23-BSCS-0134 | Hasnain Ahmed Ansari

Project Report: **UniTrack** – University Course and Enrollment Management System

# 1. Title of the Project

**UniTrack**:

A Full-Stack Web Application for University Course and Enrollment Management

# 2. Objective

To develop a web-based system that streamlines and digitizes university operations including course management, student enrollment, instructor assignment, and grade tracking, ensuring better transparency and efficiency for all stakeholders.

# 3. Abstract

UniTrack is a comprehensive, role-based course and enrollment management system built with React.js (Frontend), Node.js/Express (Backend), and Microsoft SQL Server (Database). The system provides secure login for three roles — Admin, Student, and Instructor — and enables functions such as course creation, enrollment, grade management, and reporting. The project demonstrates the practical application of database management, RESTful API development, and front-end design using modern web technologies.

# 4. Modules and Functionalities

|  |  |
| --- | --- |
| Module | Functionalities |
| Admin | Login with hardcoded credentials Add/update/delete students, Courses, Department, and instructors  Assign instructors Toggle course availability  View Enrollment History |
| Student | Login using ID and password View open courses Enroll in available courses View enrolled courses Check grades |
| Instructor | Login using ID and password View assigned courses Assign/update grades for enrolled students |
| Course | Linked to departments and instructors Track availability for enrollment |
| Enrollment | Student-course relationship table Stores student’s course enrollments |
| Grades | Managed via a view (StudentGradeReport) for reporting Inserted/updated by instructors only |

# 5. Technical Stack

* Frontend: React.js (with Vite), Tailwind CSS
* Backend: Node.js, Express.js
* Database: Microsoft SQL Server
* Connectivity: mssql Node.js library for database queries
* Security: Role-based login, plain-text password (for simplicity)

# 6. Database Design Overview

- Tables: Student, Instructor, Course, Department, Enrollments, Teaches, Grades, Users  
- Views: StudentGradeReport, CourseEnrollments, InstructorCourses  
- Triggers:  
 - Validate grade entries  
 - Prevent enrollment and assigning the instructor into closed courses  
 - Auto-insert into Users table with prefixed user IDs (S1, I2, etc.)

# 7. System Architecture

React.js (Frontend)  
↓  
Axios API Requests  
↓  
Express.js (Backend Controllers & Routes)  
↓  
MS SQL Server (Database with Views, Triggers, and Procedures)

# 8. Demo Flow (Presentation)

1. Start at the homepage: Select role (Admin, Student, Instructor)
2. Show Admin login and dashboard: Add courses, assign instructors
3. Show Student login: View open courses, enroll, view grades
4. Show Instructor login: View assigned courses, assign grades
5. Show backend data in SQL Server Management Studio (optional)

# 9. Challenges Faced

* Auto-generating user IDs with prefixes – Solved using database triggers and string formatting
* Managing role-based logins securely – Used a centralized Users table with role validation
* Handling many-to-many relationships – Enrollments managed efficiently via junction table
* Simplifying backend code – Modularized routes and controllers for clean separation

# 10. Future Improvements

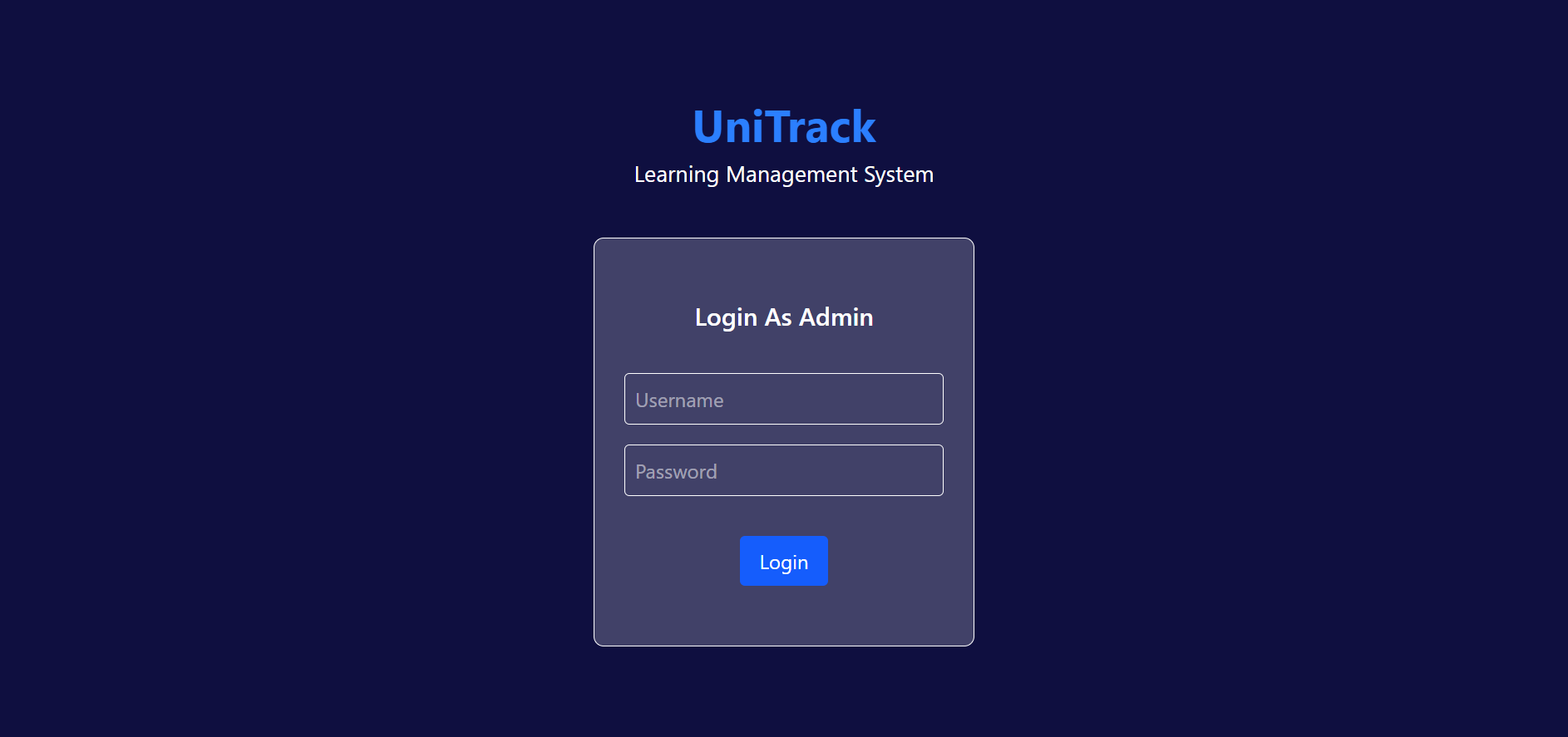
* Add class scheduling (time, day, venue) with conflict detection
* Use hashed passwords for better security
* Add pagination/search/filter on dashboards
* Email notifications for enrollment or grade updates
* Use TypeScript for better type safety in the frontend

# 11. Conclusion

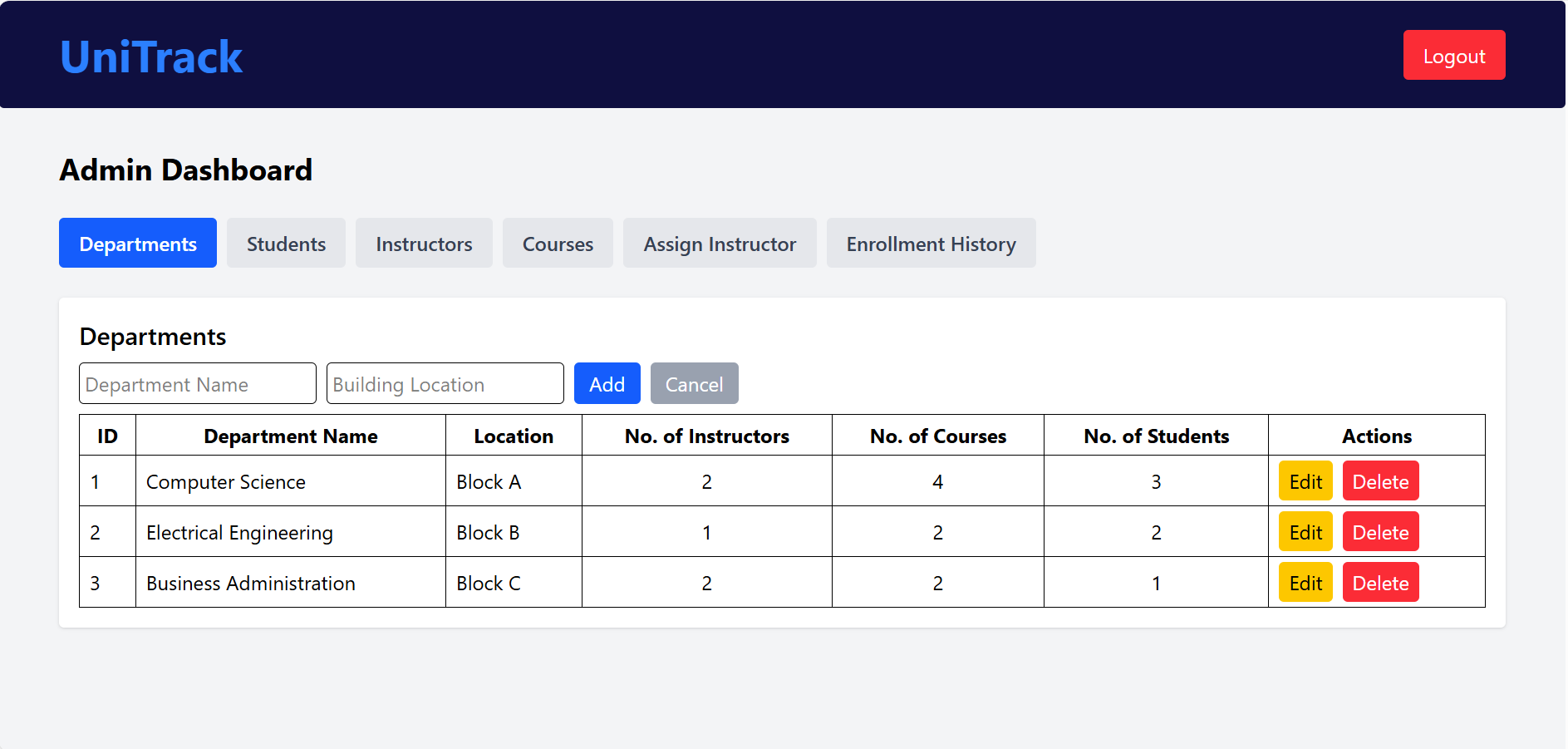
UniTrack successfully demonstrates the integration of frontend, backend, and database components to build a scalable, maintainable, and secure learning management system. It highlights core DBMS principles such as normalization, indexing, triggers, VIEW, stored procedures, and access control while also showcasing full-stack development skills.

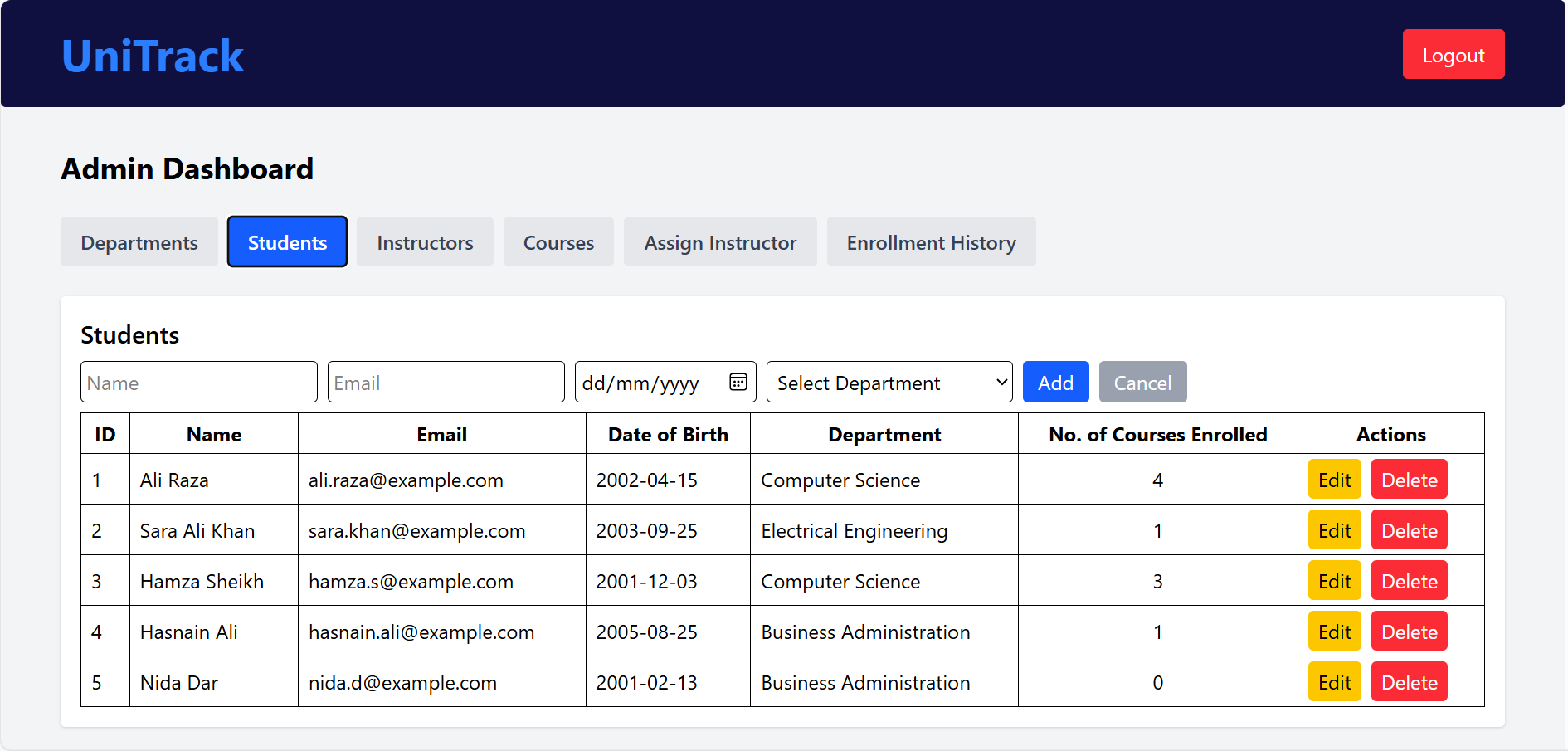
# 12. Screenshots

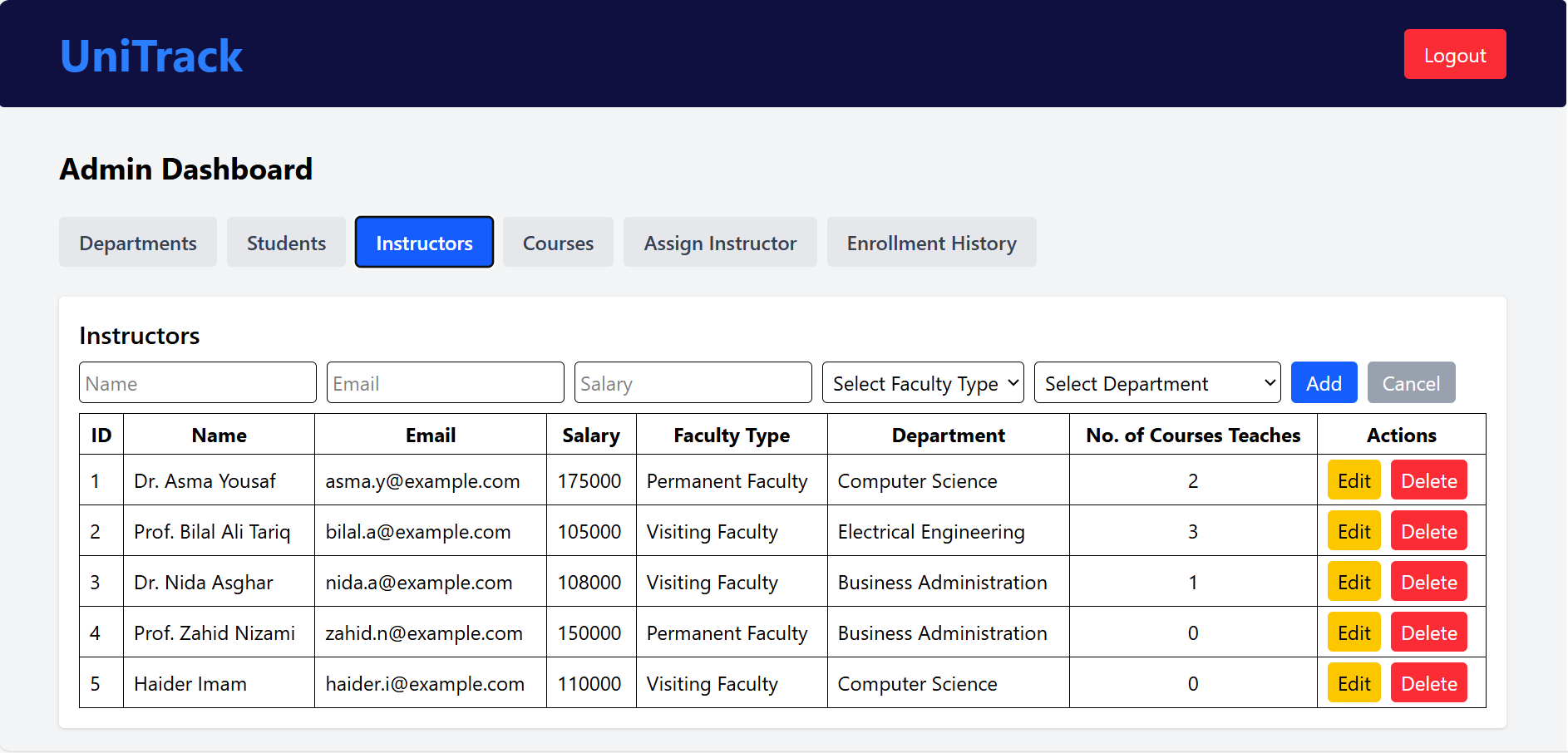
Include screenshots of the following:  
- Homepage / Role Selection:  
  
- Login Page

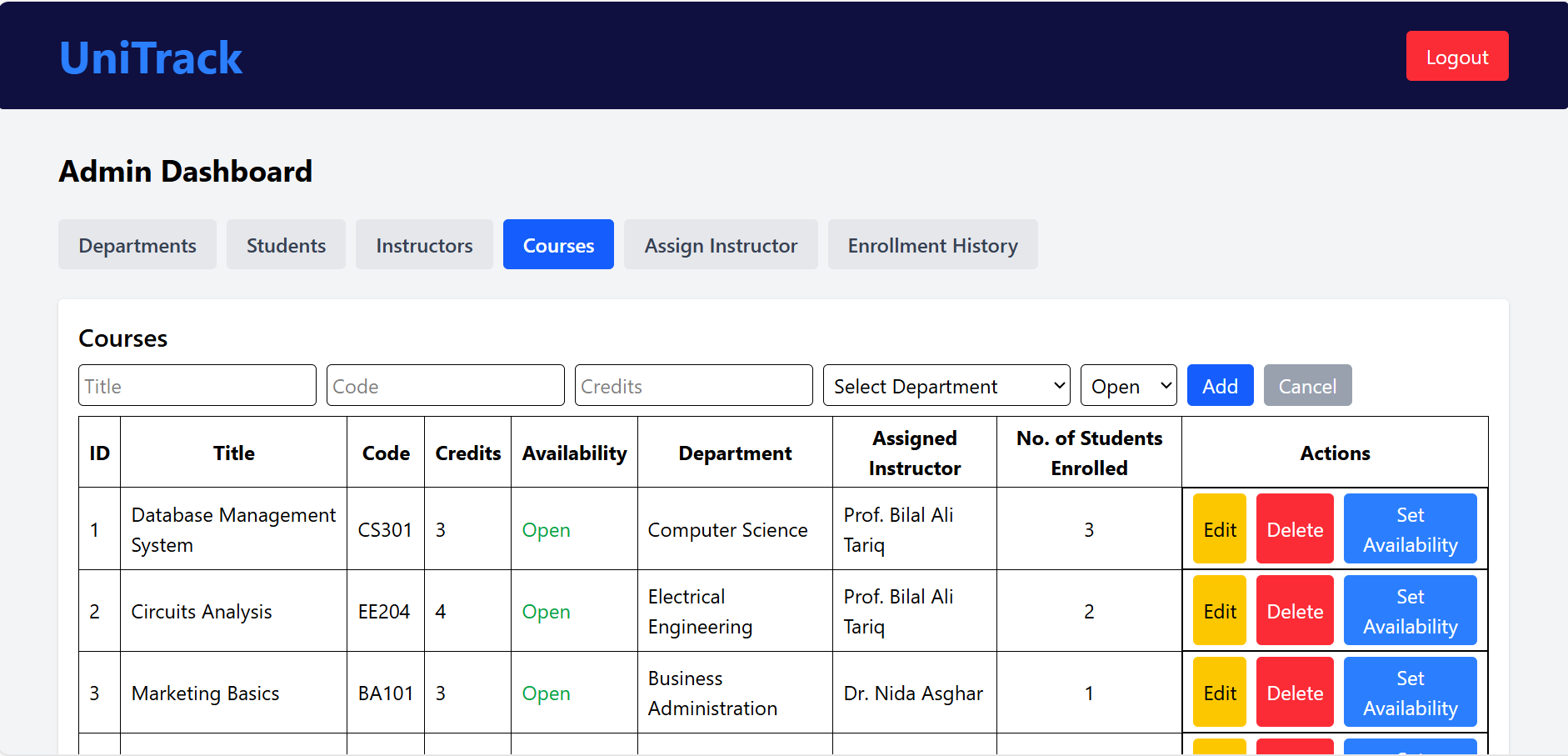


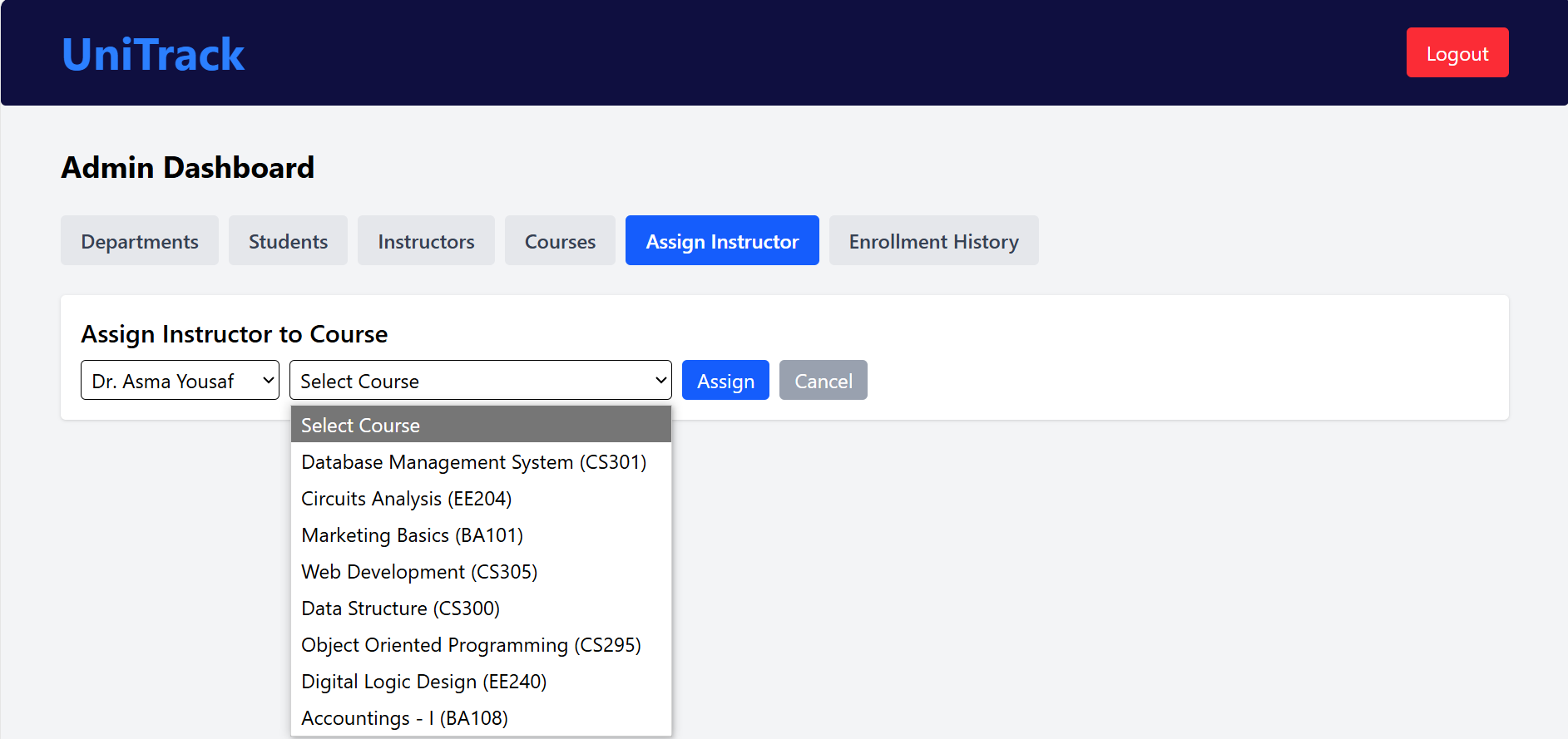
- Admin Dashboard:

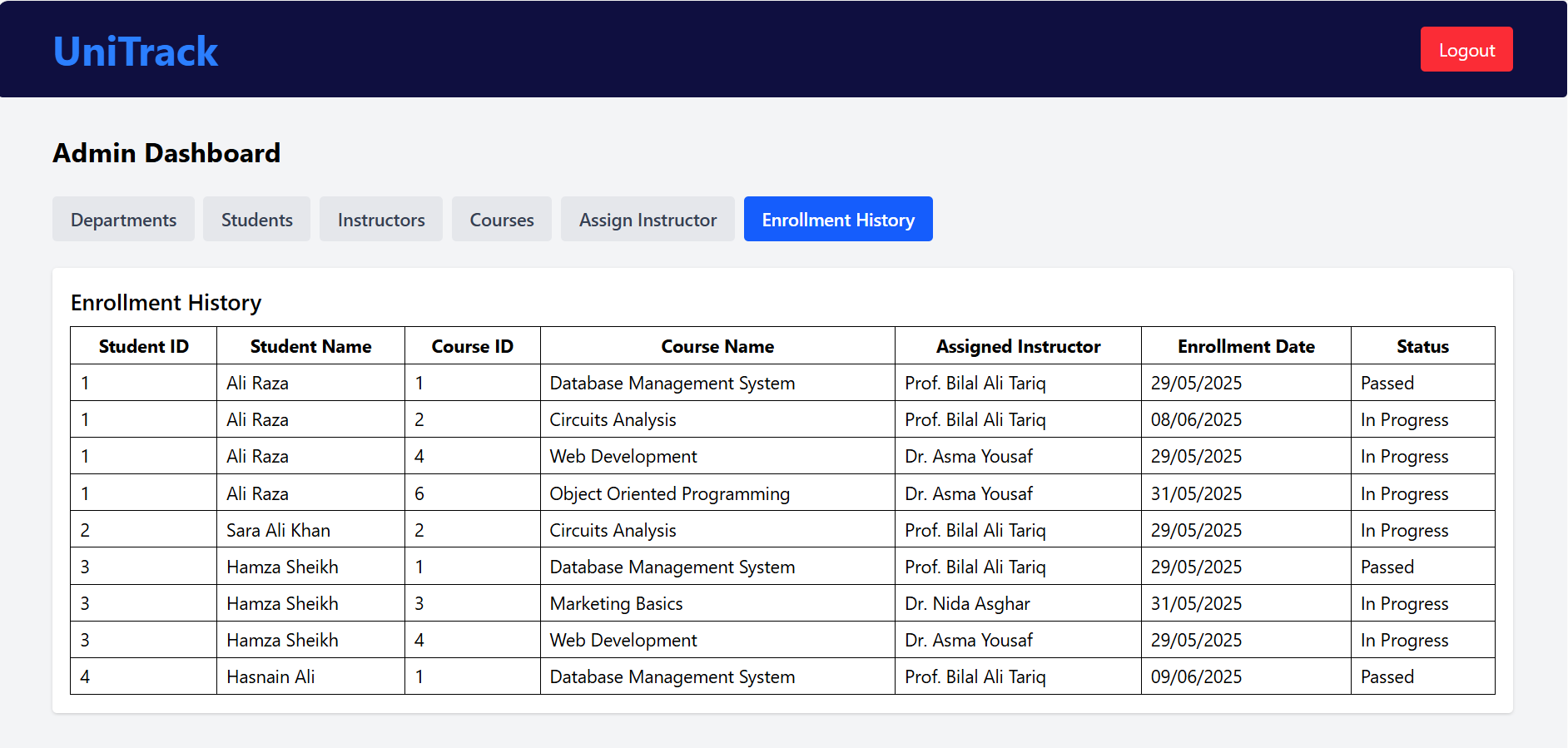




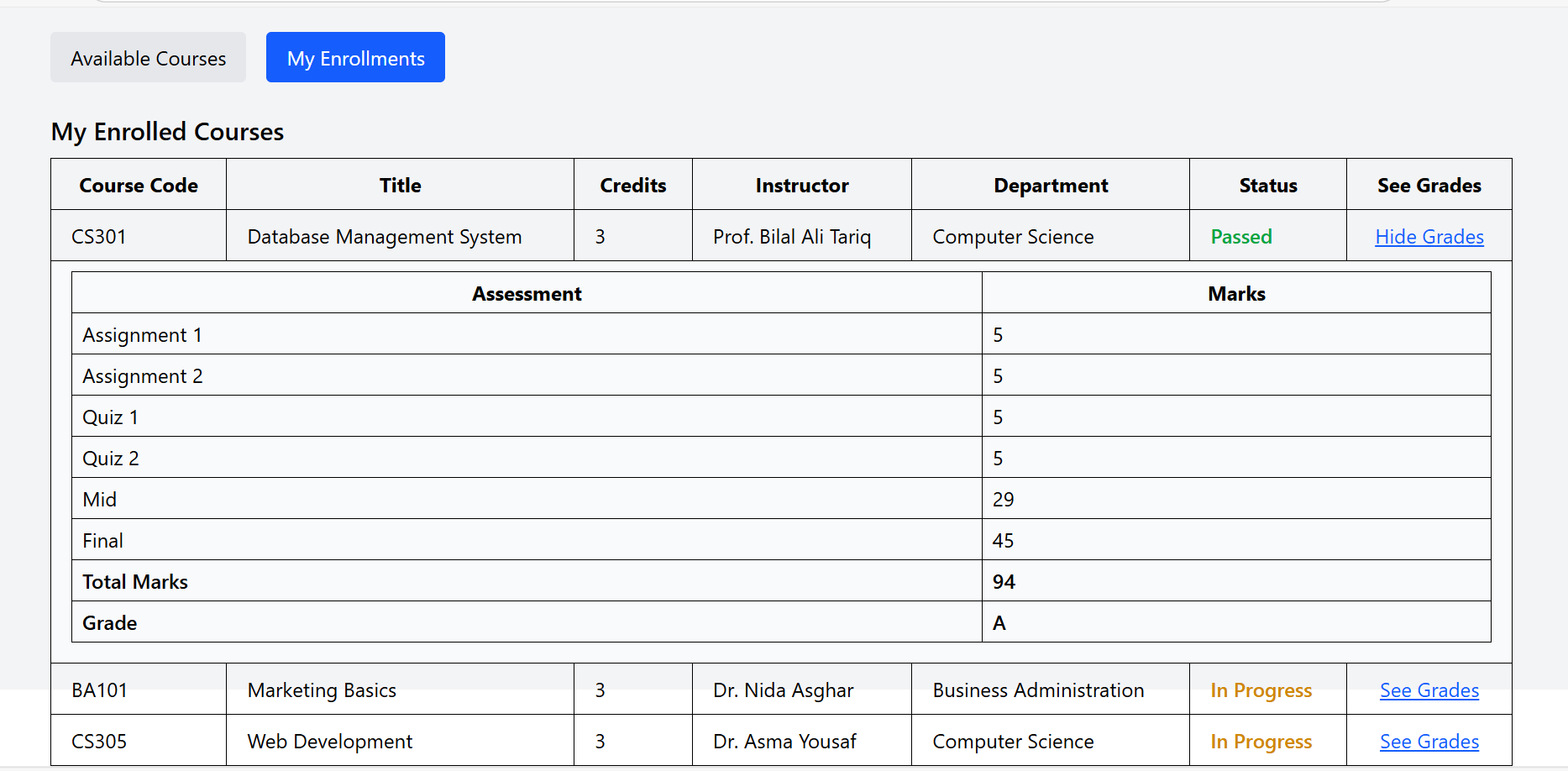


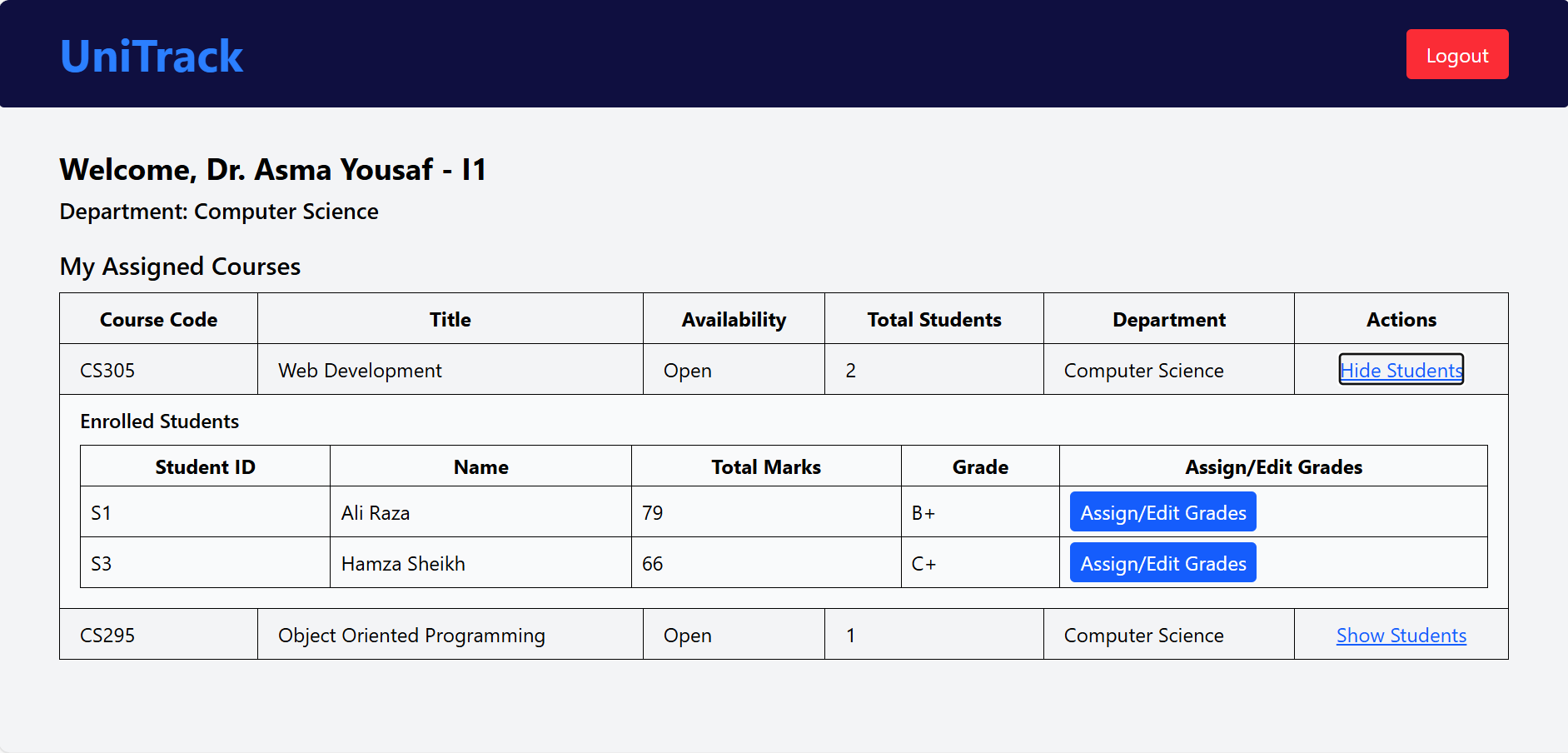


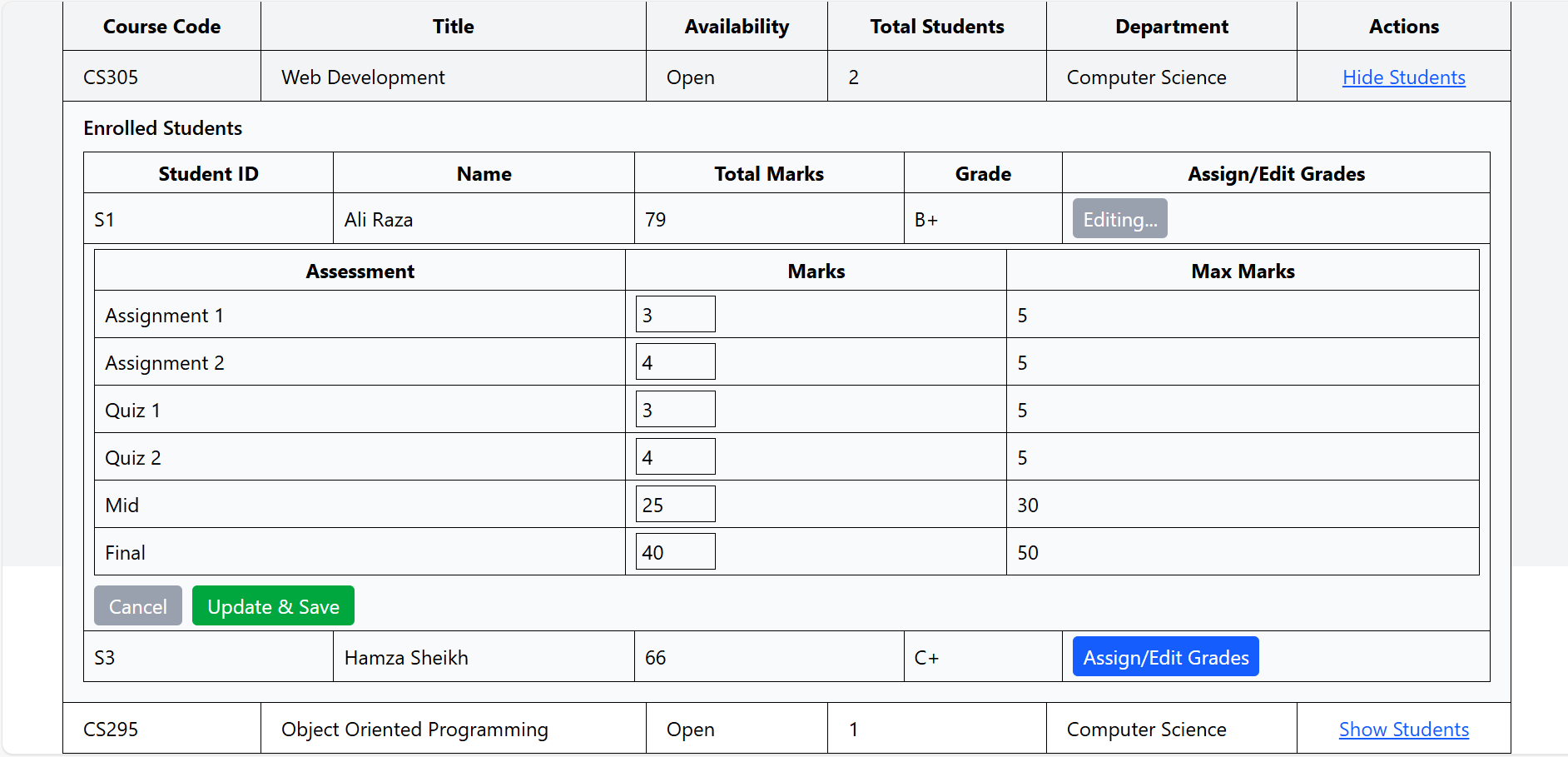




- Student Dashboard (Enroll / Grades)

   
- Instructor Dashboard (Assign Grades)





- SQL Server schema

