CS 335
Web Development
Fall 2025/2026

Milestone 4
Back-End & Database Integration

TEAM Members

Name: Khalid Afzali, Id: 2182161465
Name: Elyas Al Qasmi, Id: 22221106335
Name: Abdulwahab Alenezi, Id:225151067

Project Title: CS Scheduling System

Date	Version	Description	Author/s

1. Overview

In this milestone, we built the backend server using Node.js and Express and connected it to MongoDB using Mongoose. The backend provides REST APIs for the CS Scheduling System. It supports managing users, doctors, courses, classrooms, sections, semesters, time slots, schedules, and course materials. We added JWT login and role-based access (admin and doctor) to protect the APIs. The system can also generate schedules automatically, validate conflicts (doctor, room, capacity, and TA warnings), and publish the final schedule.

2. Server & APIs

- Technology/framework
  - Node.js
  - Express.js for REST APIs
  - MongoDB with Mongoose
  - JWT for authentication

- Implemented endpoints and CRUD operations
  - Authentication
    - Login
    - Register
    - Token verification
    - Token refresh

  - Authentication
    - Users management (CRUD)
    - Doctors management (CRUD)
    - Teaching assistants management
    - Courses management (CRUD)
    - Classrooms management (CRUD)
    - Sections management (CRUD)
    - Time slots management (CRUD)
    - Semester management
    - Schedule generation, validation, and publishing

  - Admin APIs
    - View personal profile
    - View published schedule
    - Submit availability
  -
  - Doctor APIs
    - View personal profile
    - View published schedule
    - Submit availability
  -
  - Scheduling APIs
    - Create schedules
    - Filter schedules by semester, course, or professor
    - Auto-generate schedules
    - Validate conflicts
    - Publish final schedules

- Authentication & Authorization
    - JWT-based authentication
    - Middleware to verify user identity
    - Role-based access control

    - Admin-only routes
    - Doctor-only routes
-

3. Database

- Database Used
    - MongoDB
- Final Schema & Relationships
    - User
        - One-to-one with Doctor / TeachingAssistant
- Doctor
    - Linked to User
    - Assigned to Sections
- Course
    - Linked to Sections
- Section
    - Linked to Course, Doctor, Semester
- Schedule
    - Links Course, Section, Doctor, Classroom, TimeSlot
- Classroom
    - Capacity and room type constraints
- TimeSlot
    - Day and time range
- Semester
    - Active semester control
- AuditLog
    - Tracks scheduling actions

4. Integration

- Front-End to Back-End Communication
    - Front-end communicates using HTTP REST APIs
    - JSON request/response format
    - Authorization header with JWT token
- Error Handling & Security
    - Centralized error handling middleware
    - HTTP status codes for errors
    - Input validation
    - Role-based route protection
    - Password hashing using bcrypt
    - Token expiration handling

5. Testing

- API & Database Test Cases
- User authentication (valid / invalid login)
- Role access testing (admin vs doctor)
- CRUD operations for all entities
- Schedule generation with valid data
- Schedule conflict detection:
  - Doctor double-booking
  - Classroom double-booking
  - Capacity mismatch
- Token expiration and invalid token tests
- Database seeding verification

6. Contributions

member	Contribution	
	Tasks	Percentage(%)
Khalid Afzali	Backend architecture, server setup, authentication, JWT, Database models, schema design, validation,Schedule generator,validation logic, testing and report	70%
Wahab	Testing, validation logic, validation,Schedule generator	15%
Elyas Al Qasmi	Testing, validation logic validation,Schedule generator	15%