German International University
Faculty of Informatics and Computer Science
Dr. John Zaki
Donia Ali
Nadeen Ramadan
Rahma Ameen

Software Engineering, Spring 2025
Online Event Ticketing System
Project - Task 2

1 Overview

During this milestone, you will begin developing the backend server using Node.js and Express. The following tasks should be completed.

2 Task 2 description

a) Set Up Backend Server:

- Create the Express server.
- Connect the server to the MongoDB database using mongoose.
- Organize the backend project structure (routes, controllers, middleware, models).

b) User Management:

- User Authentication and Role-Based Access Control:
  - Users, Organizers, and Admins with distinct access levels.
  - Secure login and registration using JSON Web Tokens (JWT).
  - Passwords should be stored with hashing using bcrypt to ensure data integrity.
- User Profile Management:
  - All types of users can update their profiles.

c) Events Management:

- Anyone can view posted events.
- Only organizers can post, edit *(number of tickets ,date and location of event)*, and delete events.
- Only organizers view their own events analytics.
  Hint: it will be represented as a graph containing percentage of tickets booked for each event.
- Only admins can approve or reject the event.
  - A new attribute called *status* will be added to your events schema with values [approved, pending, declined].

d) Booking Functionality:

- Authenticated standard users can view their bookings.
- Authenticated standard users can cancel their tickets of an event but keep in mind logic of deleting will increase number of tickets.

- Authenticated standard users can book tickets of an event.
- Implement logic to:
  - Check ticket availability before confirming a booking.
  - Reduce available tickets count after a successful booking.
  - Calculate total price based on ticket quantity and event price.
  - Store bookings in the database.

e) Middleware:

- Create middleware for authentication and authorization.

f) Error Handling

- Handle errors (e.g., event not found, insufficient tickets) gracefully.

g) Testing API Endpoints:

- Use tools like Postman or Thunder Client to test the developed APIs.
- Ensure all routes return the expected responses and handle edge cases.

3 API routes

Route	**Method**	**Description**	**Access**
`/api/v1/register`	POST	Register a new user	Public
`/api/v1/login`	POST	Authenticate user and return token	Public
`/api/v1/forgetPassword`	PUT	Update user password	Public
`/api/v1/users`	GET	Get a list of all users	Admin
`/api/v1/users/profile`	GET	Get current user's profile	Authenticated Users
`/api/v1/users/profile`	PUT	Update current user's profile	Authenticated Users
`/api/v1/users/:id`	GET	Get details of a single user	Admin
`/api/v1/users/:id`	PUT	Update user's role	Admin
`/api/v1/users/:id`	DELETE	Delete a user	Admin
`/api/v1/users/bookings`	GET	Get current user's bookings	Standard User
`/api/v1/users/events`	GET	Get current user's events	Event Organizer
`/api/v1/users/events/analytics`	GET	Get the analytics of the current user's events	Event Organizer
`/api/v1/bookings`	POST	Book tickets for an event	Standard User
`/api/v1/bookings/:id`	GET	Get booking details by ID	Standard User
`/api/v1/bookings/:id`	DELETE	Cancel a booking	Standard User
`/api/v1/events`	POST	Create a new event	Event Organizer
`/api/v1/events`	GET	Get list of all events	Public
`/api/v1/events/:id`	GET	Get details of a single event	Public
`/api/v1/events/:id`	PUT	Update an event	Event Organizer or Admin
`/api/v1/events/:id`	DELETE	Delete an event	Event Organizer or Admin

Table 1: Backend API Routes Overview

4 Technology Stack

- **Language:** JavaScript.
- **Backend:** Node.js, Express.
- **Database:** MongoDB (mongoose).
- **Authentication:** JSON Web Tokens (JWT) and bcrypt for secure login.

5 Deliverables

Deliverables for this milestone include:

- A working backend server connected to MongoDB.
- Implemented routes for events and bookings.
- Booking logic fully implemented and tested.
- API documentation (optional but encouraged).

6 Bonus

- Multi-Factor Authentication (MFA) in forget password either using OTP or email verification.

7 Submission Requirements

- Each team member must collaborate & commit his work.

8 Task Deadline

Task 2 deadline is Thursday, 17th April, 11:59 pm

9 Submission

a) Github repos link to be submitted through this form before the mentioned deadline:
https://forms.gle/YuFjDb9aaR8PNvZV7

b) You have to invite the following user to your Github private repository to give us access:
username: SEspring25