22i-1351_D Muqarrab Ahmad

22i-1303_D_Saad Awais

Assignment#1 DevOps

Event Booking System

Table of Contents

1. Introduction

In modern web applications, event booking platforms are widely used for managing event registrations, ticket bookings, and user interactions. This project aims to build a scalable, modular, and microservices-based event booking system that allows users to register, browse events, book tickets, and receive notifications.

The Event Booking System is designed using a microservices architecture, ensuring:
✔️ Scalability – Services can be deployed independently.
✔️ Modularity – Each service handles a specific functionality.
✔️ Flexibility – New features can be added easily.
✔️ Performance – Optimized for high-traffic applications.

---

2. Project Overview

2.1 Objectives

The primary objectives of the Event Booking System are:

1. Develop a fully functional event booking platform using React.js, Express.js, and MongoDB.
2. Implement microservices for user authentication, event management, booking, and notifications.
3. Ensure secure authentication & authorization using JWT & Role-Based Access Control (RBAC).
4. Deploy the system using Docker & Kubernetes for containerization and scalability.
5. Enable real-time notifications using RabbitMQ/Kafka.
6. Implement robust testing strategies to ensure system stability.

2.2 Technology Stack

Component	Technology Used

Frontend	**React.js**
Backend	**Node.js, Express.js**
Database	**MongoDB, PostgreSQL**
Microservices	**User, Event, Booking, Notification**
Authentication	**JWT (JSON Web Tokens)**
Messaging Queue	**RabbitMQ/Kafka**
API Gateway	**Express.js**
Testing Framewok	**Postman**

---

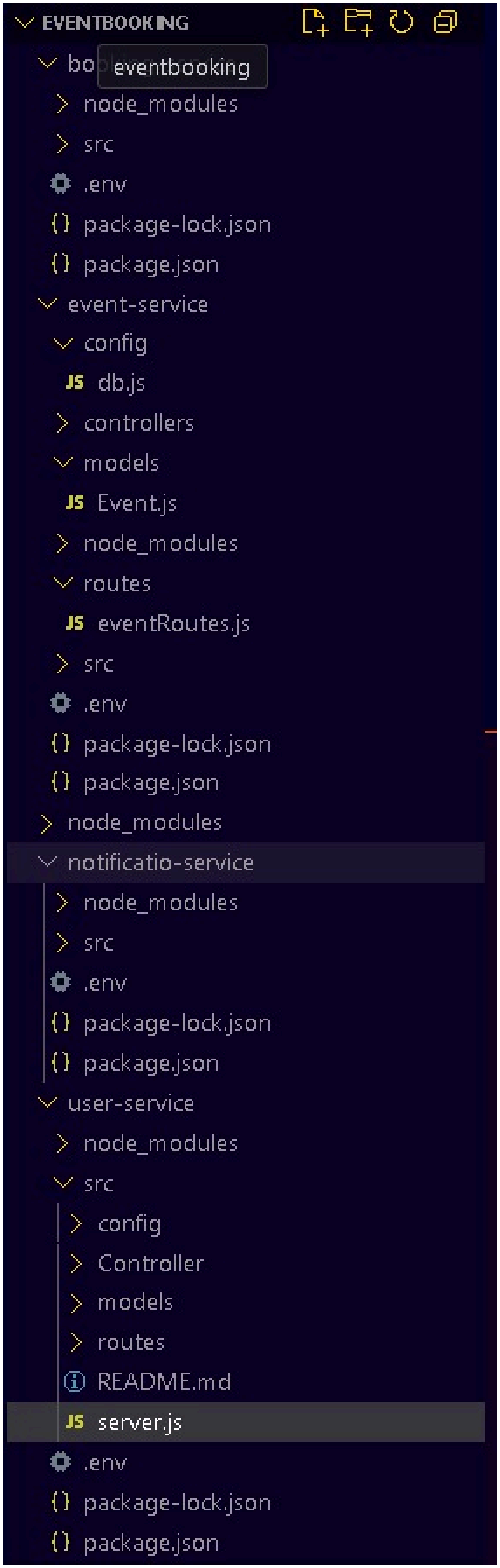
3. System Architecture

3.1 Overview of Microservices Architecture

Microservices architecture consists of independent services that handle specific functionalities. Each microservice communicates via REST APIs and message queues (RabbitMQ/Kafka).

3.2 Microservices Design & Responsibilities

Microservice	Functionality
User Service	Handles authentication, registration, login, and user management.
Event Service	Manages events, updates, and event retrieval.
Booking Service	Handles ticket reservations, payments, and confirmations.
Notification Service	Sends email/SMS notifications on successful bookings.



```
EVENTBOOKING
  bo eventbooking
    node_modules
    src
    .env
    package-lock.json
    package.json
  event-service
    config
      db.js
    controllers
    models
      Event.js
    node_modules
    routes
      eventRoutes.js
    src
    .env
    package-lock.json
    package.json
  node_modules
  notificatio-service
    node_modules
    src
    .env
    package-lock.json
    package.json
  user-service
    node_modules
    src
      config
      Controller
      models
      routes
      README.md
      server.js
    .env
    package-lock.json
    package.json
```

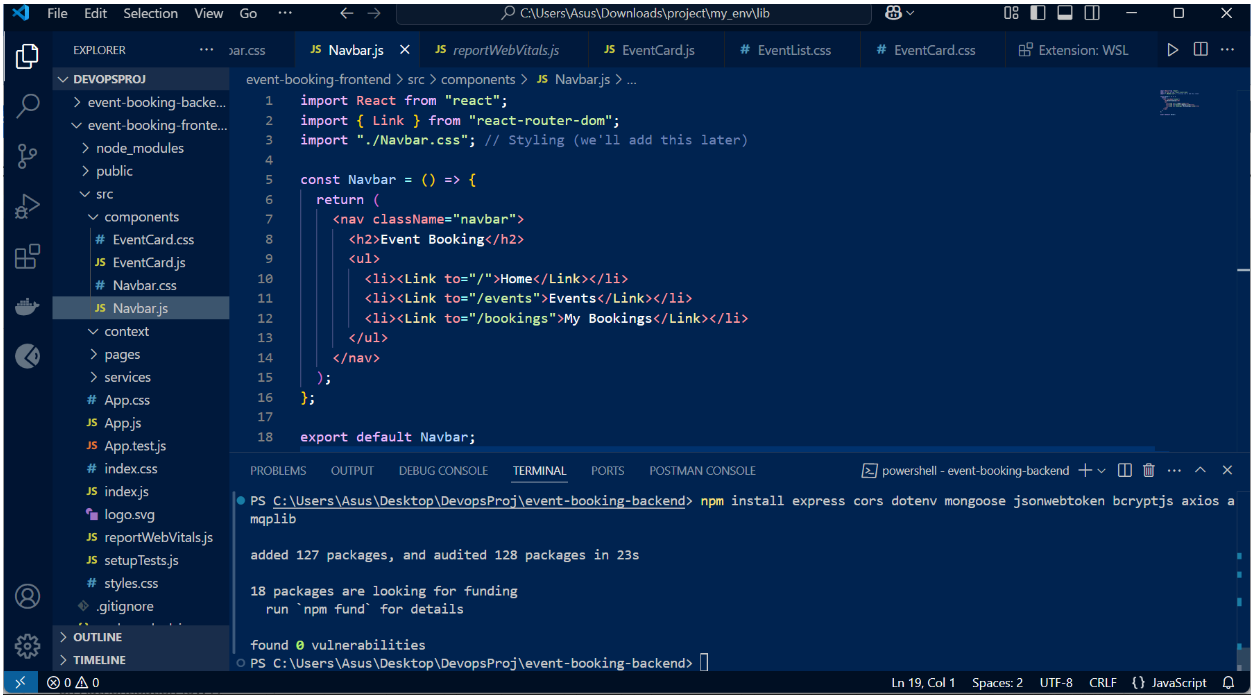
4. Implementation Details

4.1 Frontend (React.js)

The frontend is built using React.js and communicates with the backend through REST APIs.

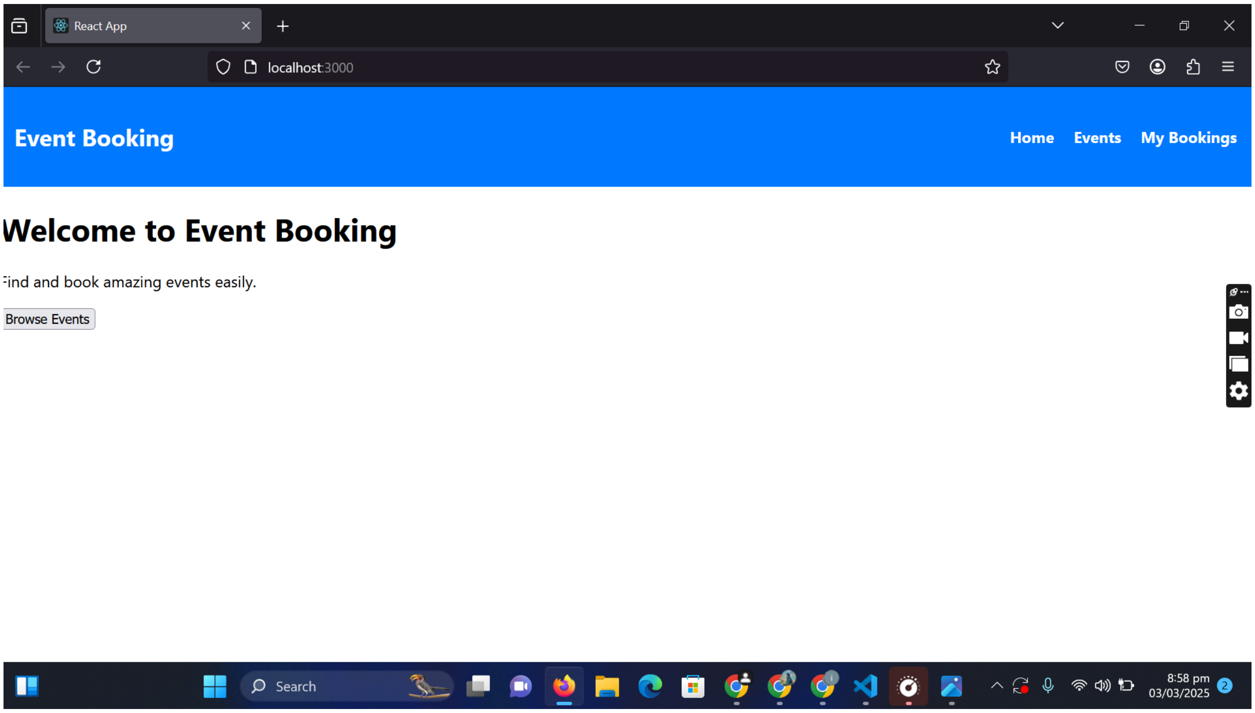
Key Features

✔️ Dark Mode Toggle
✔️ User Authentication (JWT)
✔️ Event Listing & Filtering
✔️ Booking Interface
✔️ Notifications & Alerts



```javascript
import React from "react";
import { Link } from "react-router-dom";
import "./Navbar.css"; // Styling (we'll add this later)

const Navbar = () => {
  return (
    <nav className="navbar">
      <h2>Event Booking</h2>
      <ul>
        <li><Link to="/">Home</Link></li>
        <li><Link to="/events">Events</Link></li>
        <li><Link to="/bookings">My Bookings</Link></li>
      </ul>
    </nav>
  );
};

export default Navbar;
```

```
PS C:\Users\Asus\Desktop\DevopsProj\event-booking-backend> npm install express cors dotenv mongoose jsonwebtoken bcryptjs axios a
mqplib

added 127 packages, and audited 128 packages in 23s

18 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Asus\Desktop\DevopsProj\event-booking-backend> 
```

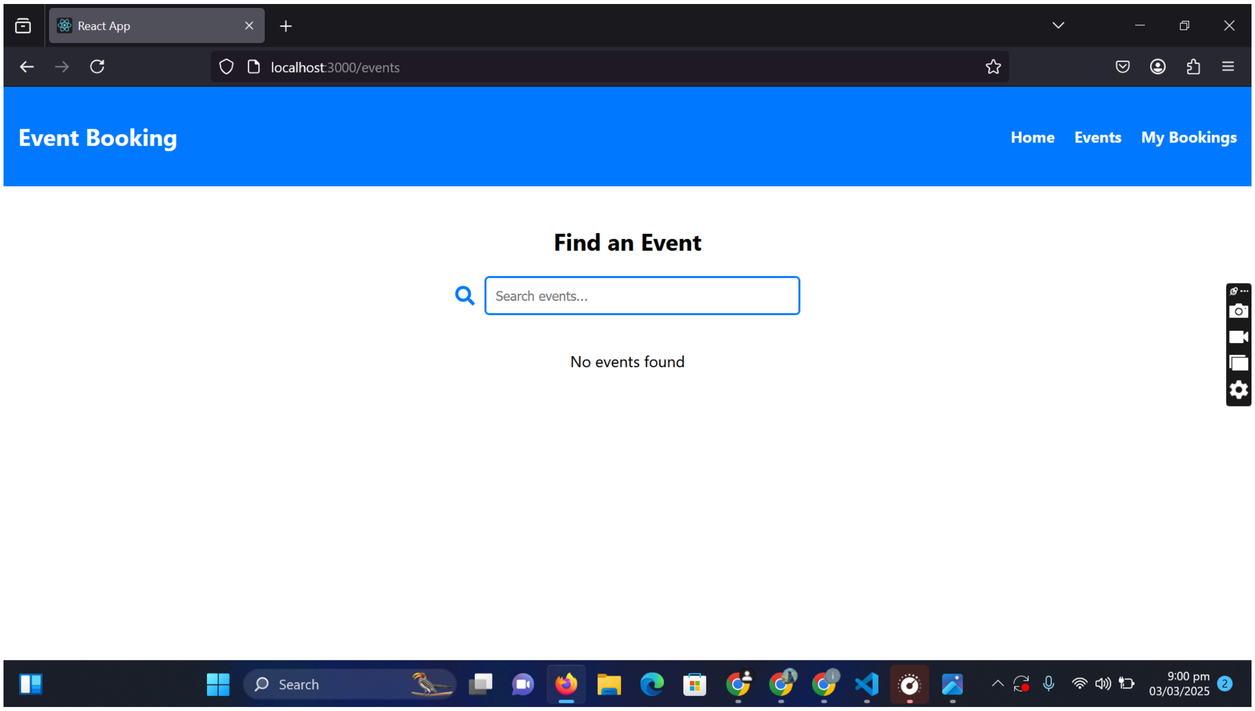


Event Booking

Welcome to Event Booking

Find and book amazing events easily.

Browse Events

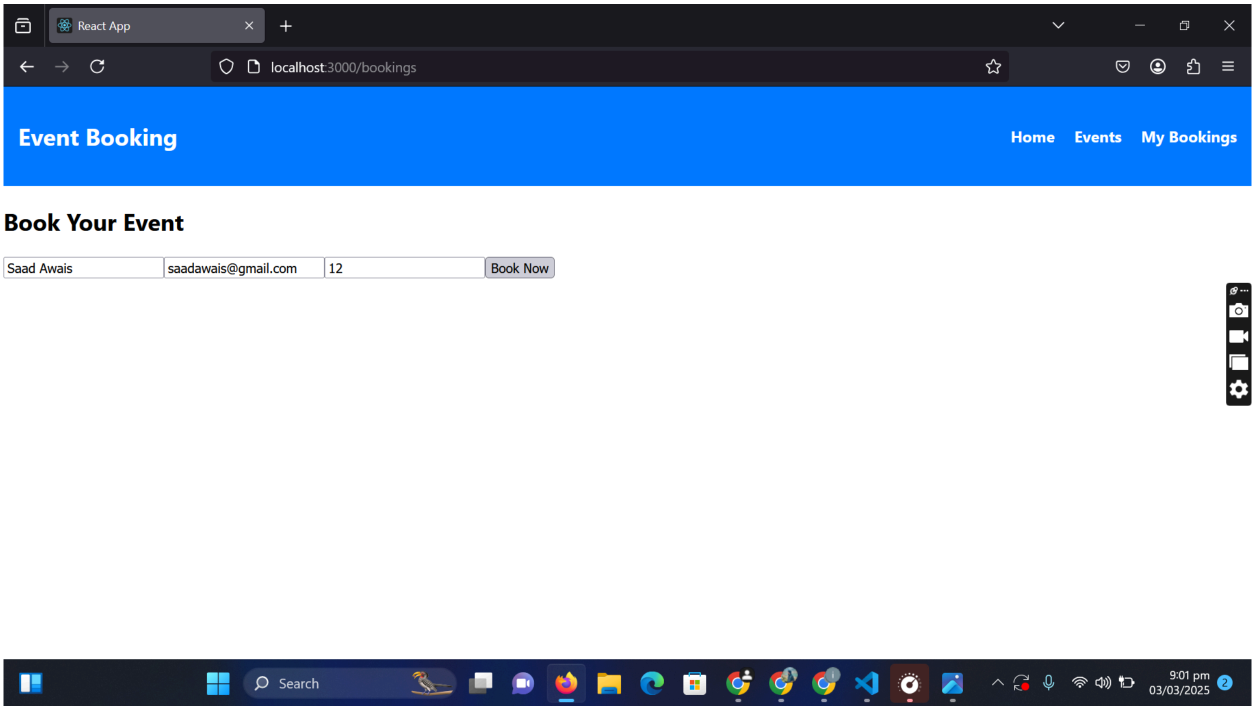


Event Booking

Find an Event

Search events...

No events found

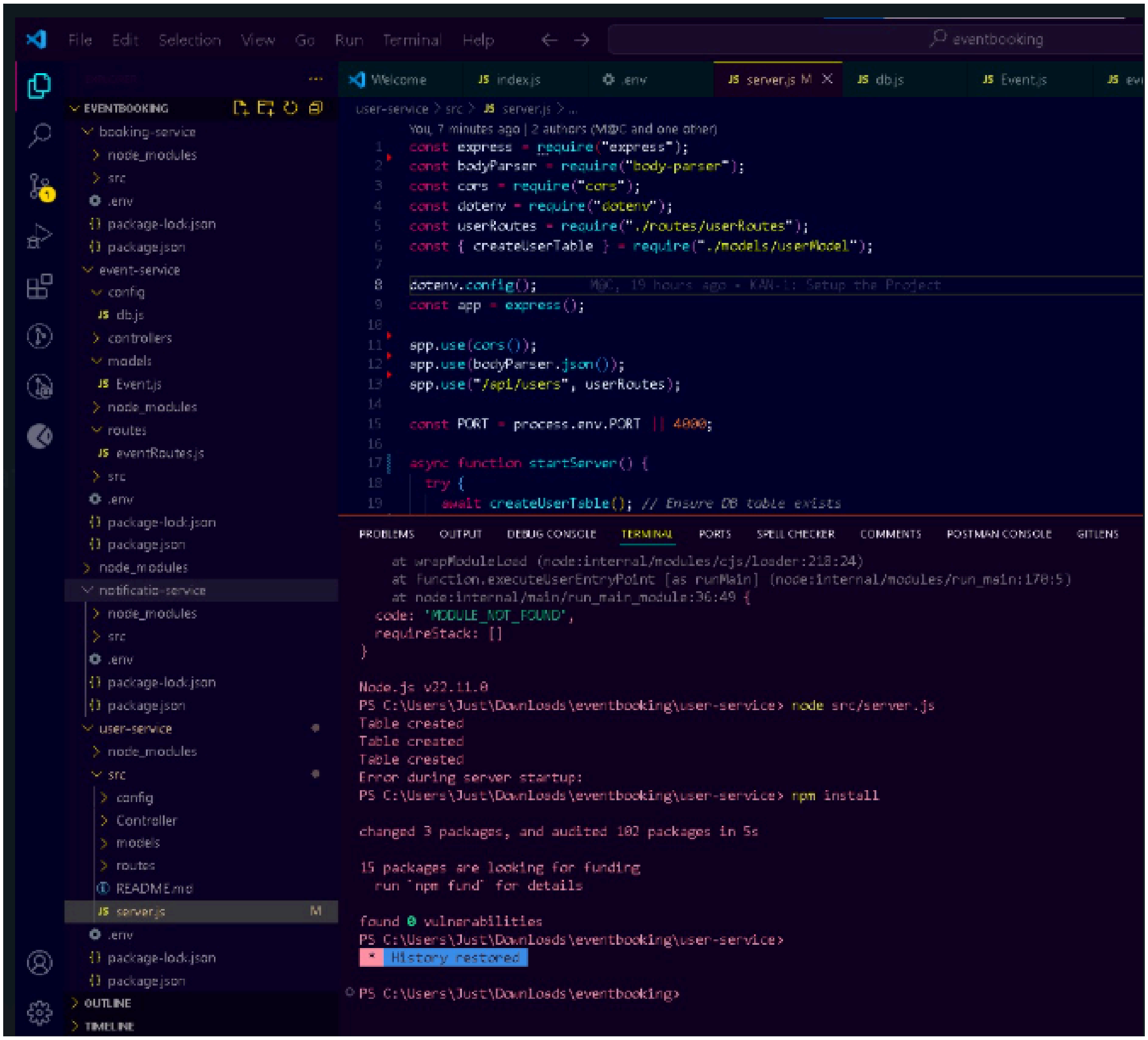


Book Your Event

Saad Awais | saadawais@gmail.com | 12

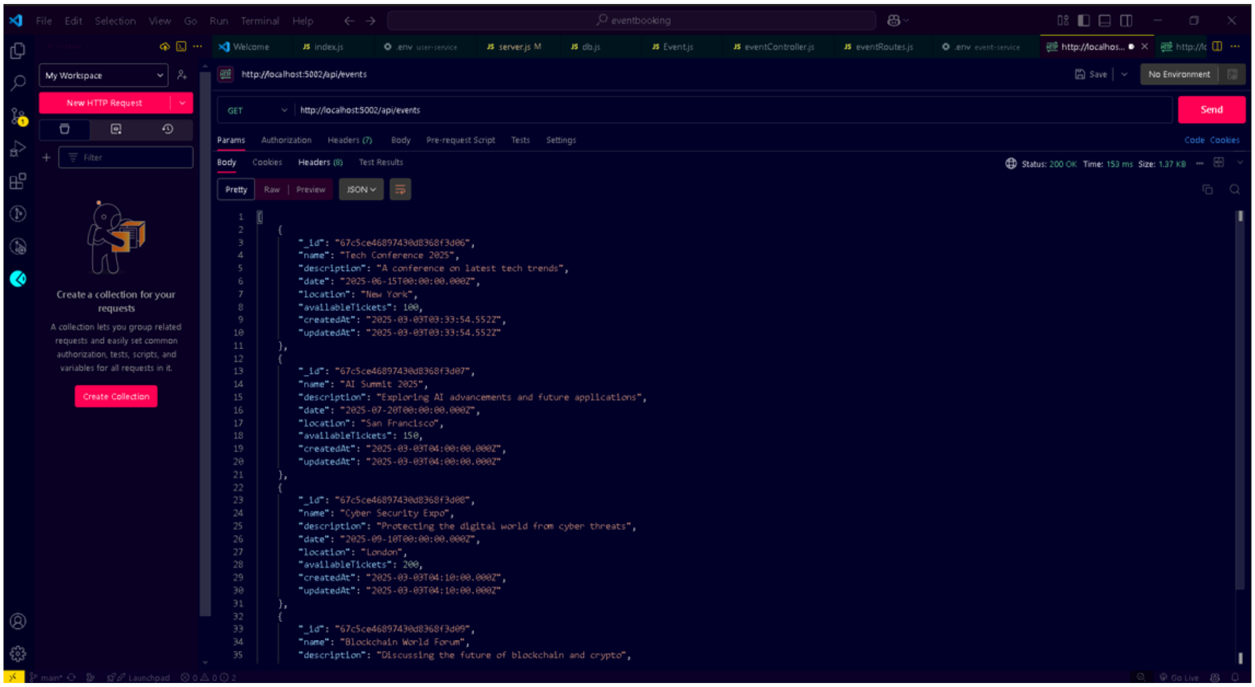
4.2 Backend (Express.js & Node.js)

Each microservice is implemented using Express.js, with MongoDB as the database.



```javascript
const express = require("express");
const bodyParser = require("body-parser");
const cors = require("cors");
const dotenv = require("dotenv");
const userRoutes = require("./routes/userRoutes");
const { createUserTable } = require("./models/userModel");

dotenv.config();
const app = express();

app.use(cors());
app.use(bodyParser.json());
app.use("/api/users", userRoutes);

const PORT = process.env.PORT || 4000;

async function startServer() {
  try {
    await createUserTable(); // Ensure DB table exists
```

```
    at wrapModuleLoad (node:internal/modules/cjs/loader:218:24)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:170:5)
    at node:internal/main/run_main_module:36:49 {
  code: 'MODULE_NOT_FOUND',
  requireStack: []
}

Node.js v22.11.0
PS C:\Users\Just\Downloads\eventbooking\user-service> node src/server.js
Table created
Table created
Table created
Error during server startup:
PS C:\Users\Just\Downloads\eventbooking\user-service> npm install

changed 3 packages, and audited 102 packages in 5s

15 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\Just\Downloads\eventbooking\user-service>
 *  History restored

PS C:\Users\Just\Downloads\eventbooking>
```



Create a collection for your requests

A collection lets you group related requests and easily set common authorization, tests, scripts, and variables for all requests in it.

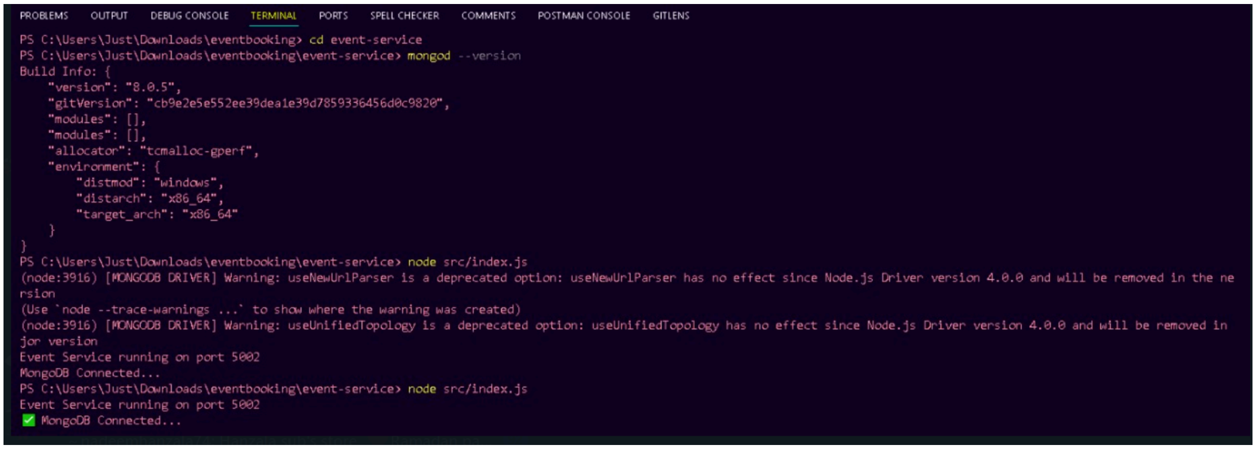
```
GET http://localhost:5002/api/events
```

Status: 200 OK Time: 153 ms Size: 1.37 KB

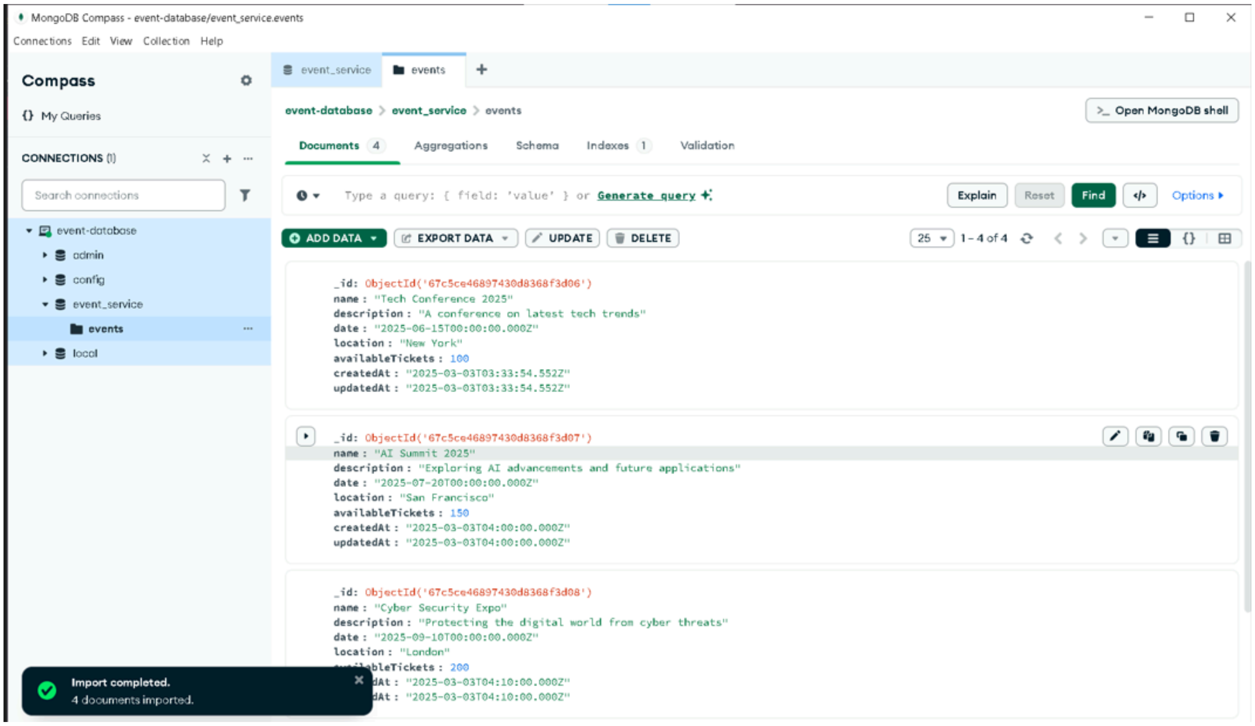
```
[
    {
        "_id": "67c5ce46897430d8368f3d06",
        "name": "Tech Conference 2025",
        "description": "A conference on latest tech trends",
        "date": "2025-06-15T00:00:00.000Z",
        "location": "New York",
        "availableTickets": 100,
        "createdAt": "2025-03-03T03:33:54.552Z",
        "updatedAt": "2025-03-03T03:33:54.552Z"
    },
    {
        "_id": "67c5ce46897430d8368f3d07",
        "name": "AI Summit 2025",
        "description": "Exploring AI advancements and future applications",
        "date": "2025-07-20T00:00:00.000Z",
        "location": "San Francisco",
        "availableTickets": 150,
        "createdAt": "2025-03-03T04:00:00.000Z",
        "updatedAt": "2025-03-03T04:00:00.000Z"
    },
    {
        "_id": "67c5ce46897430d8368f3d08",
        "name": "Cyber Security Expo",
        "description": "Protecting the digital world from cyber threats",
        "date": "2025-09-10T00:00:00.000Z",
        "location": "London",
        "availableTickets": 200,
        "createdAt": "2025-03-03T04:10:00.000Z",
        "updatedAt": "2025-03-03T04:10:00.000Z"
    },
    {
        "_id": "67c5ce46897430d8368f3d09",
        "name": "Blockchain World Forum",
        "description": "Discussing the future of blockchain and crypto",
```

4.3 Database (MongoDB and PostgreSQL

MongoDB stores user, event, and booking data.

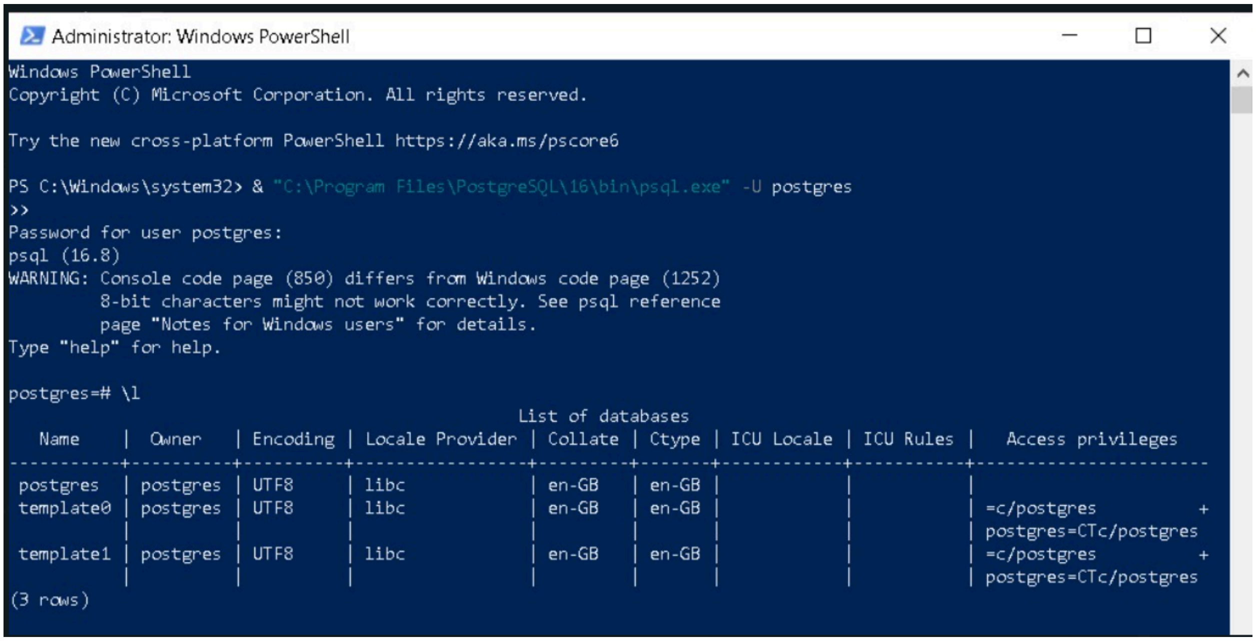


```
PS C:\Users\Just\Downloads\eventbooking> cd event-service
PS C:\Users\Just\Downloads\eventbooking\event-service> mongod --version
Build Info: {
    "version": "8.0.5",
    "gitVersion": "cb9e2e5e552ee39dea1e39d7859336456d0c9820",
    "modules": [],
    "modules": [],
    "allocator": "tcmalloc-gperf",
    "environment": {
        "distmod": "windows",
        "distarch": "x86_64",
        "target_arch": "x86_64"
    }
}
PS C:\Users\Just\Downloads\eventbooking\event-service> node src/index.js
(node:3916) [MONGODB DRIVER] Warning: useNewUrlParser is a deprecated option: useNewUrlParser has no effect since Node.js Driver version 4.0.0 and will be removed in the ne
rsion
(Use `node --trace-warnings ...` to show where the warning was created)
(node:3916) [MONGODB DRIVER] Warning: useUnifiedTopology is a deprecated option: useUnifiedTopology has no effect since Node.js Driver version 4.0.0 and will be removed in
jor version
Event Service running on port 5002
MongoDB Connected...
PS C:\Users\Just\Downloads\eventbooking\event-service> node src/index.js
Event Service running on port 5002
✅ MongoDB Connected...
```



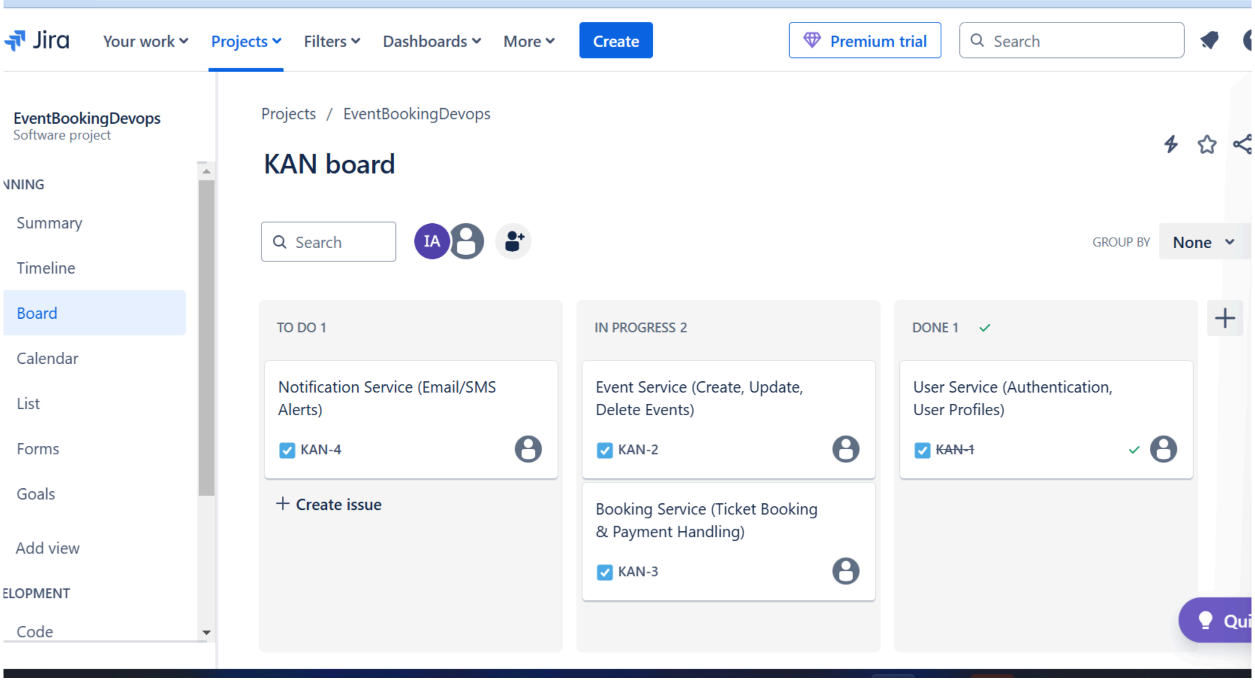
```
_id: ObjectId('67c5ce46897430d8368f3d06')
name : "Tech Conference 2025"
description : "A conference on latest tech trends"
date : "2025-06-15T00:00:00.000Z"
location : "New York"
availableTickets : 100
createdAt : "2025-03-03T03:33:54.552Z"
updatedAt : "2025-03-03T03:33:54.552Z"

_id: ObjectId('67c5ce46897430d8368f3d07')
name : "AI Summit 2025"
description : "Exploring AI advancements and future applications"
date : "2025-07-20T00:00:00.000Z"
location : "San Francisco"
availableTickets : 150
createdAt : "2025-03-03T04:00:00.000Z"
updatedAt : "2025-03-03T04:00:00.000Z"

_id: ObjectId('67c5ce46897430d8368f3d08')
name : "Cyber Security Expo"
description : "Protecting the digital world from cyber threats"
date : "2025-09-10T00:00:00.000Z"
location : "London"
availableTickets : 200
dAt : "2025-03-03T04:10:00.000Z"
dAt : "2025-03-03T04:10:00.000Z"
```

Import completed.
4 documents imported.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> & "C:\Program Files\PostgreSQL\16\bin\psql.exe" -U postgres
>>
Password for user postgres:
psql (16.8)
WARNING: Console code page (850) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=# \l
                                                   List of databases
   Name    |  Owner   | Encoding | Locale Provider | Collate | Ctype | ICU Locale | ICU Rules |   Access privileges
-----------+----------+----------+-----------------+---------+-------+------------+-----------+-----------------------
 postgres  | postgres | UTF8     | libc            | en-GB   | en-GB |            |           |
 template0 | postgres | UTF8     | libc            | en-GB   | en-GB |            |           | =c/postgres          +
           |          |          |                 |         |       |            |           | postgres=CTc/postgres
 template1 | postgres | UTF8     | libc            | en-GB   | en-GB |            |           | =c/postgres          +
           |          |          |                 |         |       |            |           | postgres=CTc/postgres
(3 rows)
```

4.4 Jira and Github screenshots:



KAN board

TO DO 1

Notification Service (Email/SMS Alerts) — KAN-4

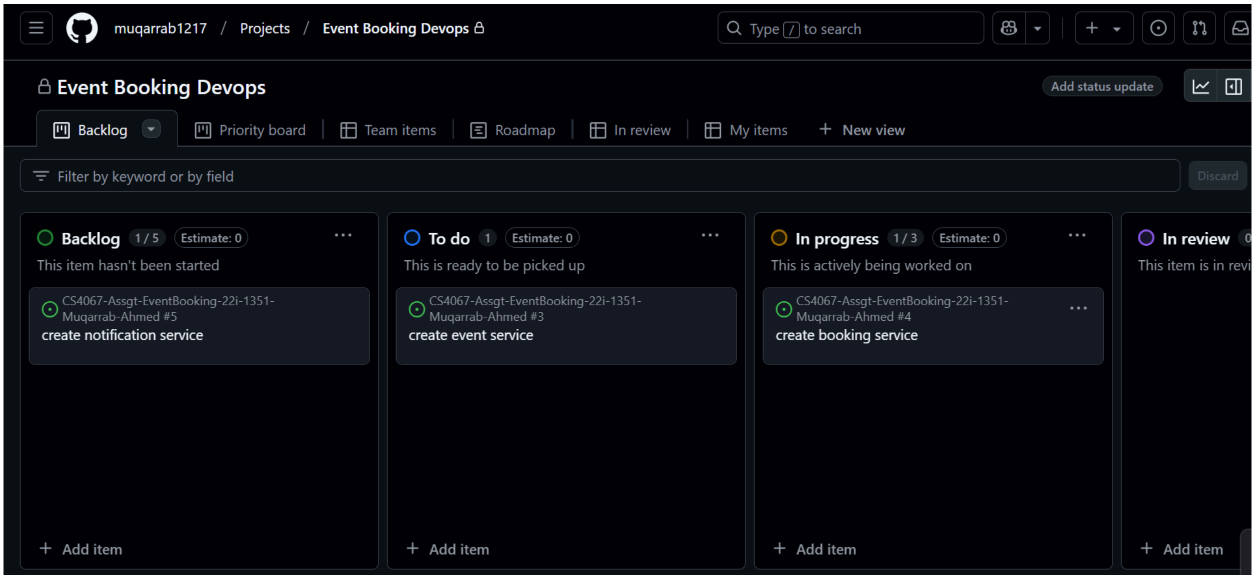
IN PROGRESS 2

Event Service (Create, Update, Delete Events) — KAN-2

Booking Service (Ticket Booking & Payment Handling) — KAN-3

DONE 1 ✓

User Service (Authentication, User Profiles) — ~~KAN-1~~ ✓



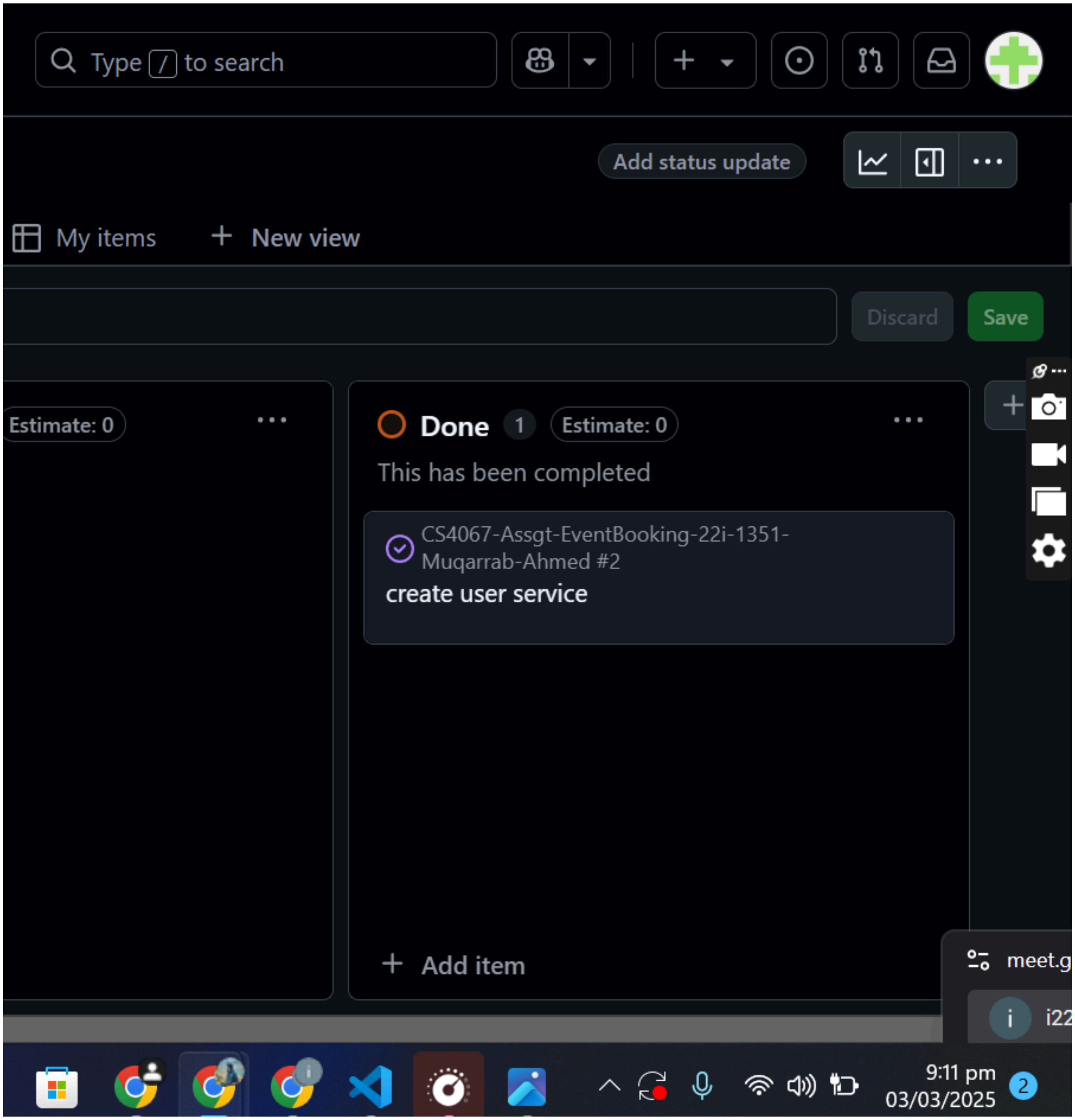
Event Booking Devops

Backlog 1 / 5 Estimate: 0
This item hasn't been started
CS4067-Assgt-EventBooking-22i-1351-Muqarrab-Ahmed #5
create notification service

To do 1 Estimate: 0
This is ready to be picked up
CS4067-Assgt-EventBooking-22i-1351-Muqarrab-Ahmed #3
create event service

In progress 1 / 3 Estimate: 0
This is actively being worked on
CS4067-Assgt-EventBooking-22i-1351-Muqarrab-Ahmed #4
create booking service

In review 0
This item is in revi

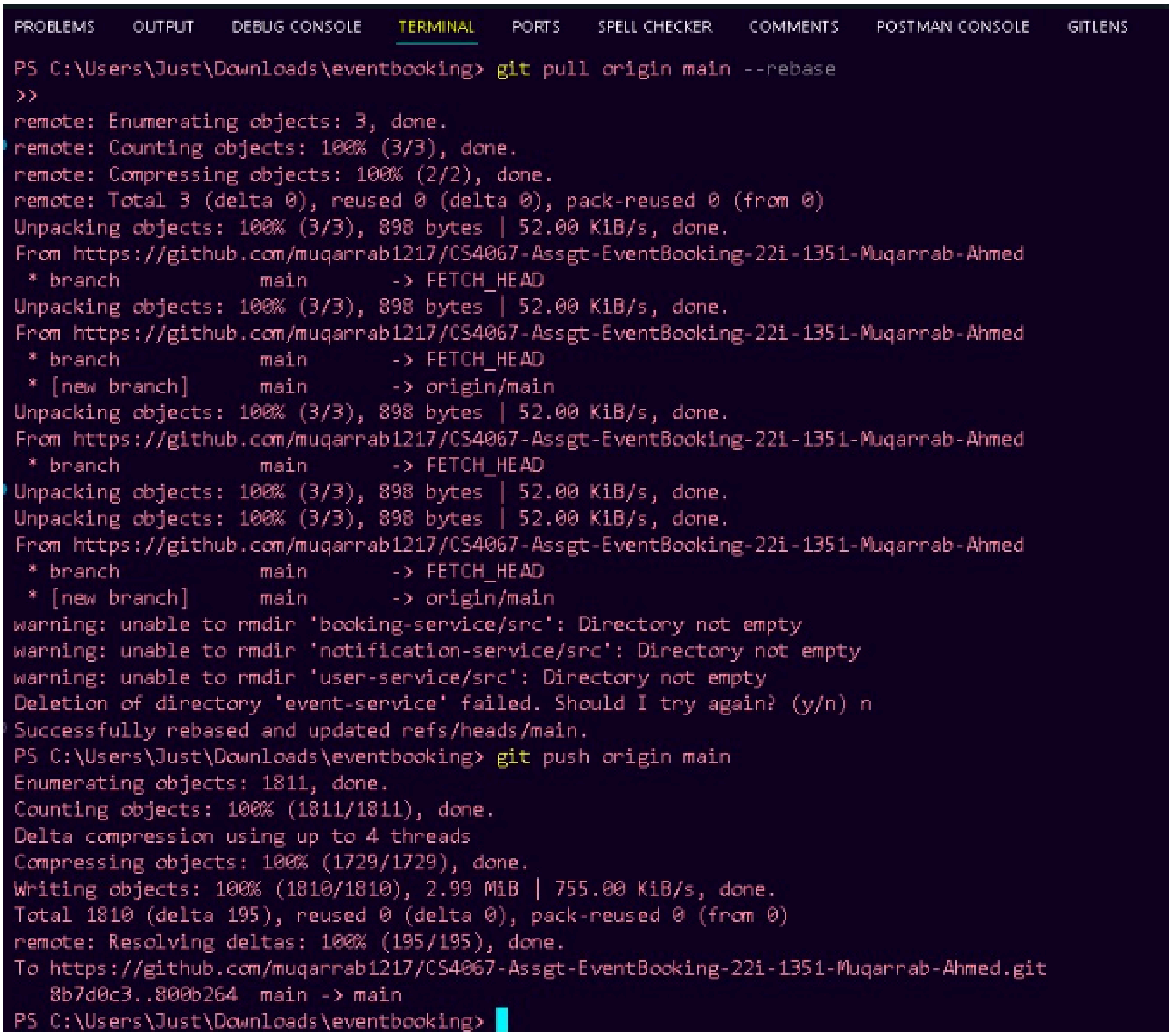


Done 1 Estimate: 0

This has been completed

CS4067-Assgt-EventBooking-22i-1351-Muqarrab-Ahmed #2

create user service



```
PS C:\Users\Just\Downloads\eventbooking> git pull origin main --rebase
>>
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 898 bytes | 52.00 KiB/s, done.
From https://github.com/muqarrab1217/CS4067-Assgt-EventBooking-22i-1351-Muqarrab-Ahmed
 * branch            main       -> FETCH_HEAD
Unpacking objects: 100% (3/3), 898 bytes | 52.00 KiB/s, done.
From https://github.com/muqarrab1217/CS4067-Assgt-EventBooking-22i-1351-Muqarrab-Ahmed
 * branch            main       -> FETCH_HEAD
 * [new branch]      main       -> origin/main
Unpacking objects: 100% (3/3), 898 bytes | 52.00 KiB/s, done.
From https://github.com/muqarrab1217/CS4067-Assgt-EventBooking-22i-1351-Muqarrab-Ahmed
 * branch            main       -> FETCH_HEAD
Unpacking objects: 100% (3/3), 898 bytes | 52.00 KiB/s, done.
Unpacking objects: 100% (3/3), 898 bytes | 52.00 KiB/s, done.
From https://github.com/muqarrab1217/CS4067-Assgt-EventBooking-22i-1351-Muqarrab-Ahmed
 * branch            main       -> FETCH_HEAD
 * [new branch]      main       -> origin/main
warning: unable to rmdir 'booking-service/src': Directory not empty
warning: unable to rmdir 'notification-service/src': Directory not empty
warning: unable to rmdir 'user-service/src': Directory not empty
Deletion of directory 'event-service' failed. Should I try again? (y/n) n
Successfully rebased and updated refs/heads/main.
PS C:\Users\Just\Downloads\eventbooking> git push origin main
Enumerating objects: 1811, done.
Counting objects: 100% (1811/1811), done.
Delta compression using up to 4 threads
Compressing objects: 100% (1729/1729), done.
Writing objects: 100% (1810/1810), 2.99 MiB | 755.00 KiB/s, done.
Total 1810 (delta 195), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (195/195), done.
To https://github.com/muqarrab1217/CS4067-Assgt-EventBooking-22i-1351-Muqarrab-Ahmed.git
   8b7d0c3..800b264  main -> main
PS C:\Users\Just\Downloads\eventbooking>
```

5. Security Implementation

5.1 Authentication (JWT)

- **Users receive a JWT token upon login.**
- **Tokens are validated for secure access.**

5.2 Role-Based Access Control (RBAC)

- **Admin – Manage events, users, and bookings.**

- **User – Book events & view details.**

---

8. Challenges & Solutions

Challenge	**Solution**
Authentication across services	**Postman**
Database scaling	**Used MongoDB sharding**

---

10. Conclusion

The Event Booking System successfully demonstrates microservices architecture, scalability, and modular deployment.