Final Project Reflection

INF653 Back-End Web Development I, Spring 2025

A) What I Learned from Building the Event Ticketing API:

1. API Architecture: Designed a modular Express application, separating controllers, routes, and models for scalability and maintainability.
2. Data Modeling: Created robust Mongoose schemas enforcing required fields, enums, and relational references between Users, Events, and Bookings.
3. Security: Implemented JWT-based authentication, role-based access control, and bcrypt.js for secure password hashing.
4. Bonus Integrations:
   • QR Codes: Used the 'qrcode' package to generate scannable tickets.
   • Email: Integrated Nodemailer with SendGrid/Ethereal for booking confirmations.
5. Deployment: Containerized the app via Render.com with environment variables, ensuring zero-downtime builds and secure secret management.

B) The Most Challenging Aspects:

1. Concurrency & Atomicity: Ensuring seat availability checks and updates ran atomically to prevent overbooking under simultaneous requests.
2. Email Deliverability: Debugging SMTP authentication errors (535/550) with SendGrid, and configuring verified sender identities.
3. Middleware Order: Orchestrating route definitions and global error handlers to correctly serve HTML/JSON 404 responses without intercepting valid routes.
4. Testing Flow: Building a reusable REST-Client script to sequentially exercise all endpoints, capture dynamic IDs, and validate error cases.

This capstone solidified my end-to-end backend development skills—from database design and API security to DevOps and continuous deployment.