EIE4432 Group Project

Student name: Yuen Pak Hei

Student ID: 23079398d

Project topic: Western Orchestral Music Performance Seat Booking System

Public URL: <http://127.0.0.1:8080/>

**(This is one-person group)**

**Table of Contents**

1. Introduction
2. Technical Stack Summary
3. Website Structure
4. Application Features Checklist
5. Database Design
6. Third-party APIs and npm Packages
7. User Test Cases
8. Project Timeline
9. Work Distribution
10. Conclusion

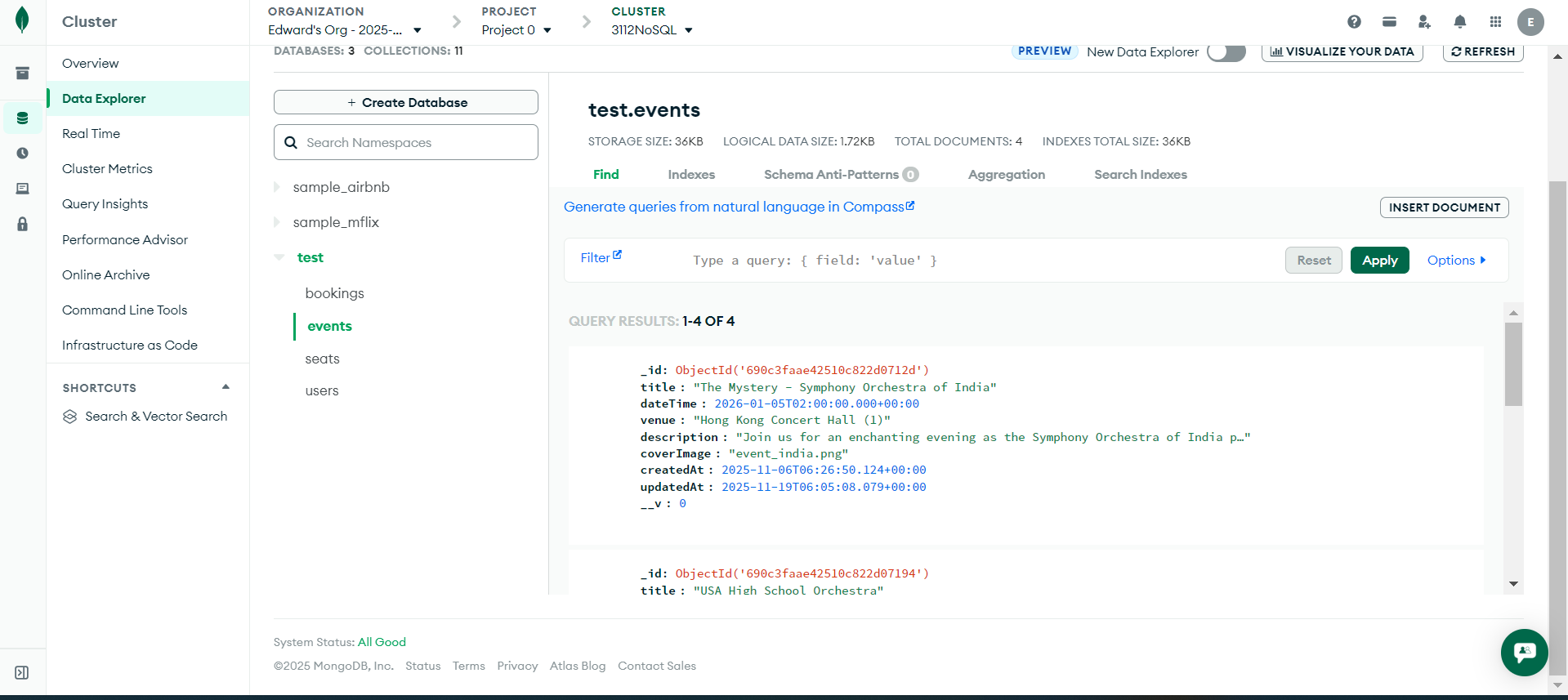
○ I declare that Generative AI tools have been used to prepare the submitted work.

The Generative AI tools used and how they were used are as follows with brief

description: \_Perplexity AI (giving ideas of the user interface & website structure)\_.

○ You must reference them according to the IEEE style.

MongoDB Atlas database:



1. **Introduction**

This project implements a web-based event management system focused on user-friendly event browsing, interactive seat selection, and user profile management. The system supports administrators managing events, seats, and users, and attendees booking seats and viewing tickets. It includes multi-platform responsive UI with collapsible menus, adaptive layouts, user-friendly prompts, validations, and consistent Bootstrap styling.

1. **Technical stack summary**

|  |  |
| --- | --- |
| Layer | Technology |
| Fronted | HTML5, CSS3, JavaScript, SVG for interactive seat maps |
| Backend | Node.js with Express.js |
| Database | MongoDB [MongoDB Atlas] (NoSQL) |
| Tools | Visual Studio Code, Mongoose, Multer (file upload), bcrypt, sessions |
| Hosting | <http://127.0.0.1:8080/> Local host or any Node.js supported cloud hosting |

1. **Web structure**

A diagram of a flowchart

Description automatically generated

1. **Application Features Checklist**

Basic features

|  |  |
| --- | --- |
| Feature | Complete or not? |
| User Account Registration | Complete |
| User Authentication | Complete |
| Event Dashboard | Complete |
| Ticket Booking Page | Complete |
| Payment Page | Complete |
| Seat Management Page (Admin Only) | Complete |
| Event Management Page (Admin Only) | Complete |

Credit Features

|  |  |
| --- | --- |
| Feature | Complete or not? |
| User Login Enhancements | Complete |
| User Account Management | Complete |
| Event Dashboard | Complete |
| Ticket Booking Page | Complete |
| Transaction History | Complete |
| Seat Management Page (Admin Only) | Complete |
| Event Management Page (Admin Only) | Complete |

Advanced Features

|  |  |
| --- | --- |
| Feature | Complete or not? |
| Enhanced User Registration | Complete |
| Interactive Seat Management (Admin Only) | Complete |

1. **Database design with diagram**

The system uses MongoDB, a document-based NoSQL database. The main collections are:

Users, Events, Seats, Bookings.

|  |  |
| --- | --- |
| Collection | Fields (Key Attributes and Special Fields) |
| Users | \_id (Primary Key, autogenerated ObjectId)  userId (unique user identifier)  nickname (string)  email (string, unique)  passwordHash (string, hashed password for security)  role (string, e.g. ‘user’ or ‘admin’)  profileImage (string, filename or URL path) |
| Events | \_id (primary key)  title (string)  dateTime (ISODate)  venue (string)  description (string)  coverImage (string, image filename) |
| Seats | \_id (primary key)  eventId (foreign key reference to Events.\_id)  seatNumber (string, e.g., 'A1')  floor (number)  status (string: 'available', 'booked', 'disabled')  priceTier (string: 'economy', 'premium')  price (number)  Optionally: bookingId (foreign key to Bookings.\_id if booked) |
| Bookings | \_id (primary key)  userId (foreign key to Users.\_id)  eventId (foreign key to Events.\_id)  seats (array of embedded seat documents with seat number and seatId)  totalCost (number)  paymentStatus (string)  transactionDate (ISODate) |

Data Flow and Relationships

Users create Bookings that link to one or more Seats for specific Events.

Seats belong to an Event; their booking status is tracked.

Passwords are stored securely in Users as hashed values in passwordHash.

Role field in Users controls authorization for admin versus regular user actions.

Diagram Description

Users (1) — (N) Bookings: One user can have many bookings.

Events (1) — (N) Seats: Each event has many seats.

Bookings contain multiple seats related to an event.

Seats optionally link to a booking if status is "booked".

1. **Third-party APIs and npm Packages**

Express for HTTP server and routing

Mongoose for MongoDB data modeling and validation

Multer for handling file uploads (profile images, event covers)

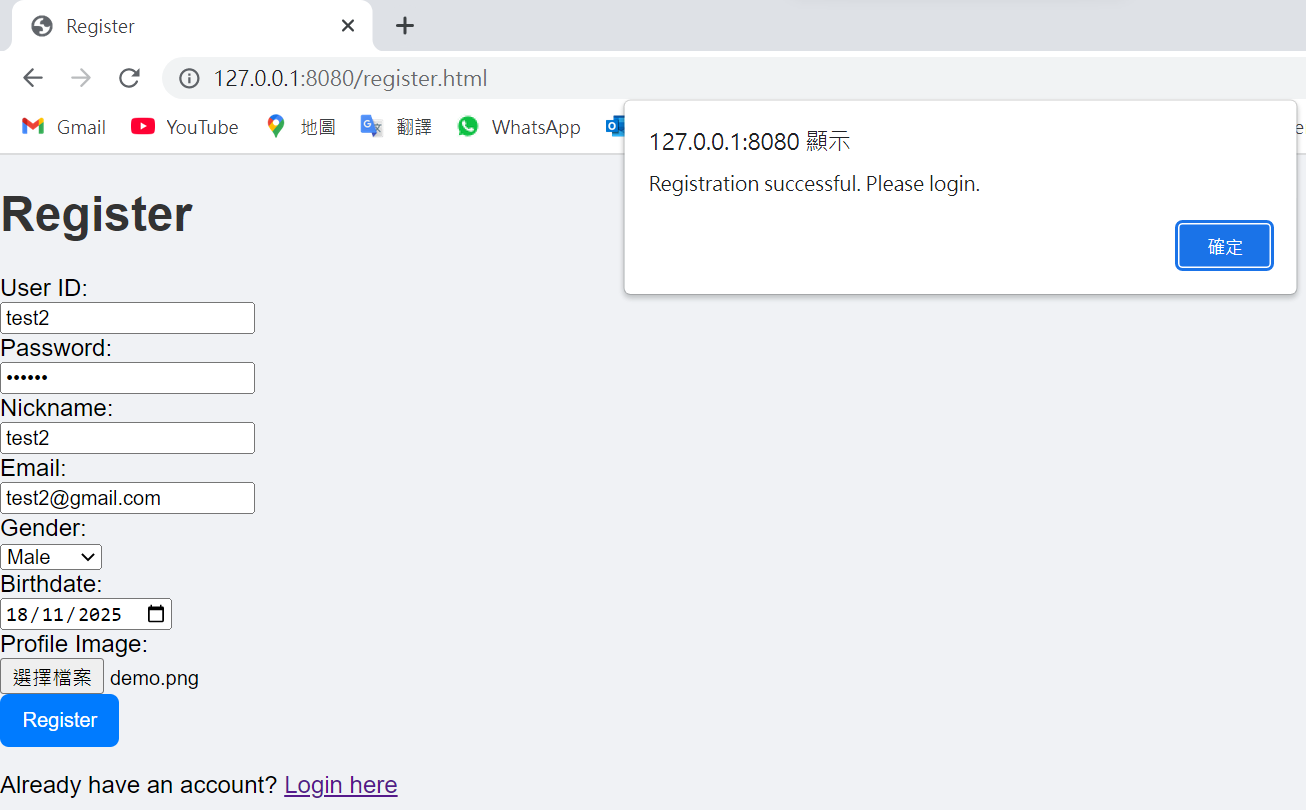
bcrypt for password hashing

JsBarcode for barcode generation on tickets

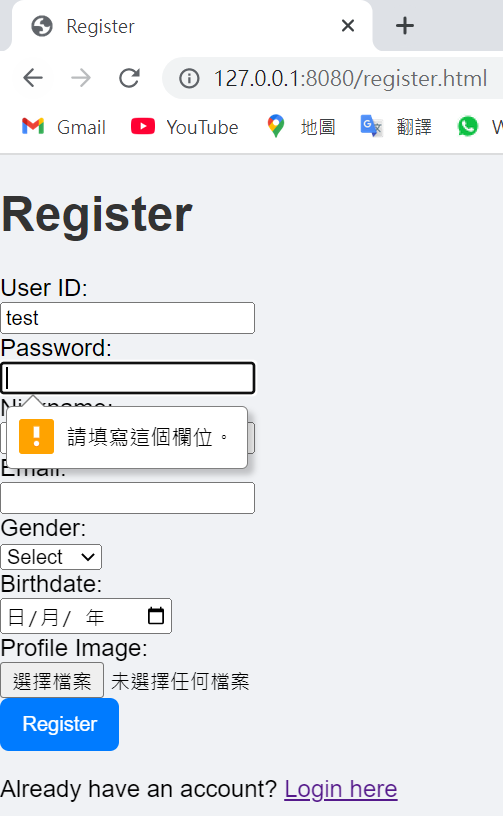
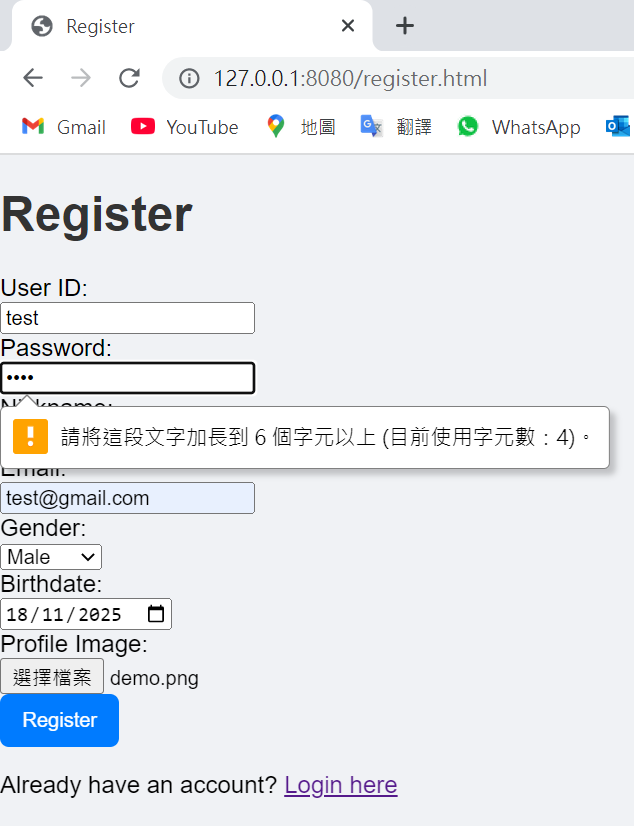
express-session for user session management

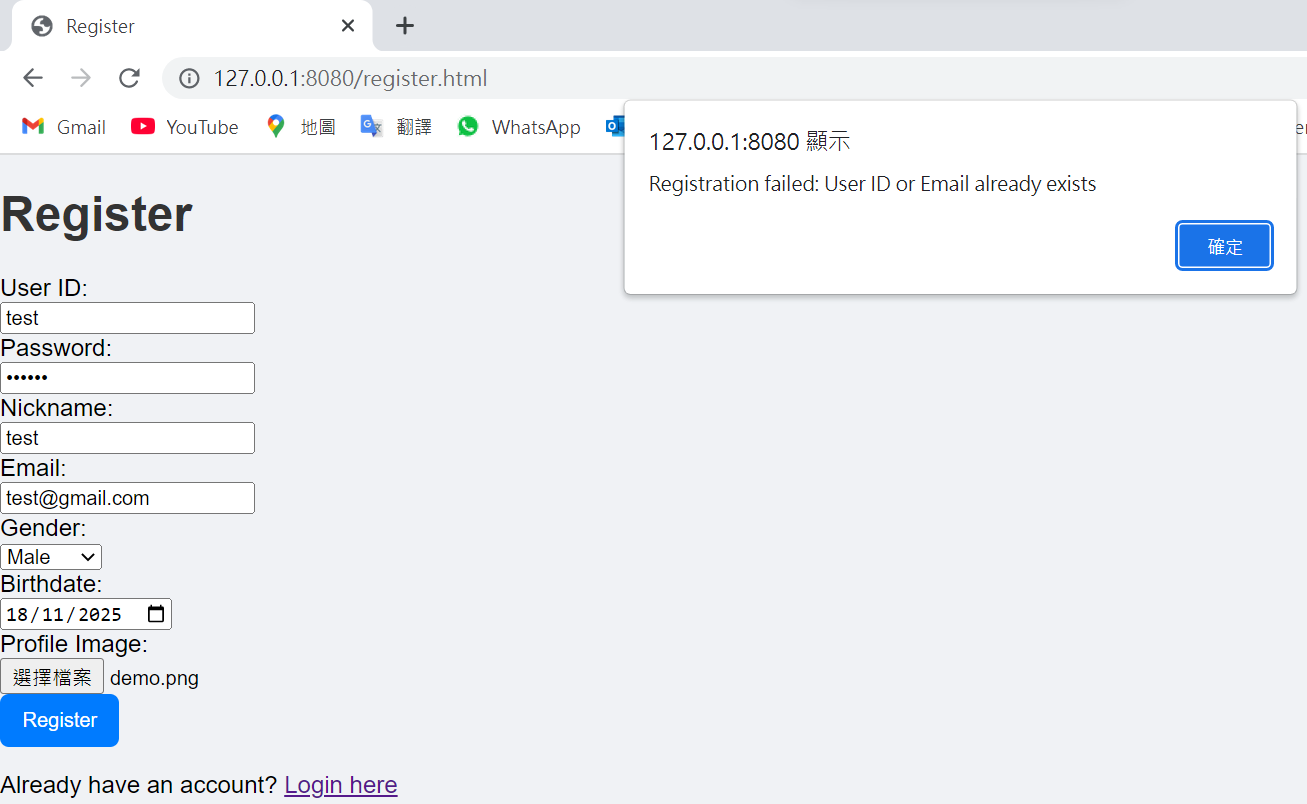
1. User test cases

Valid user account registration [userID = test2 ; password = 123456] :

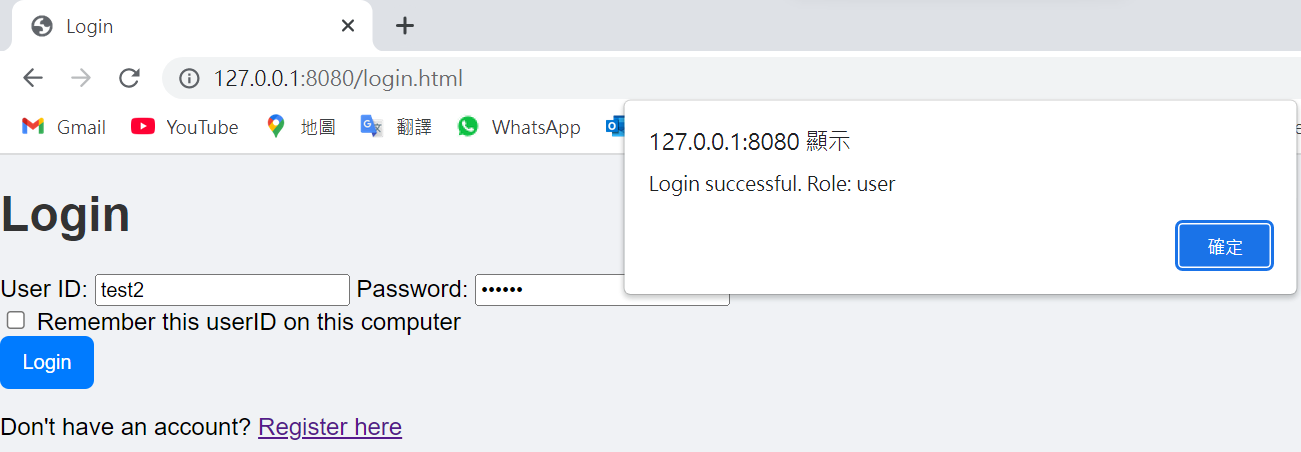


Invalid user account registration [Submit empty or invalid password (length smaller than 6)]:

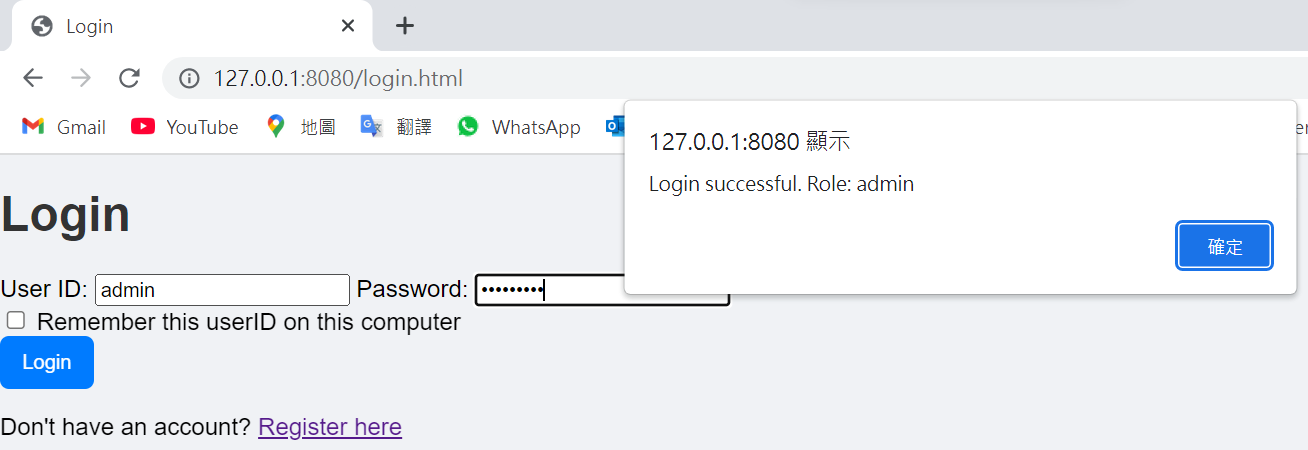
 

Invalid user account registration [Submit duplicate userID (duplicate ID checks)]:

Valid user login [role = user]:

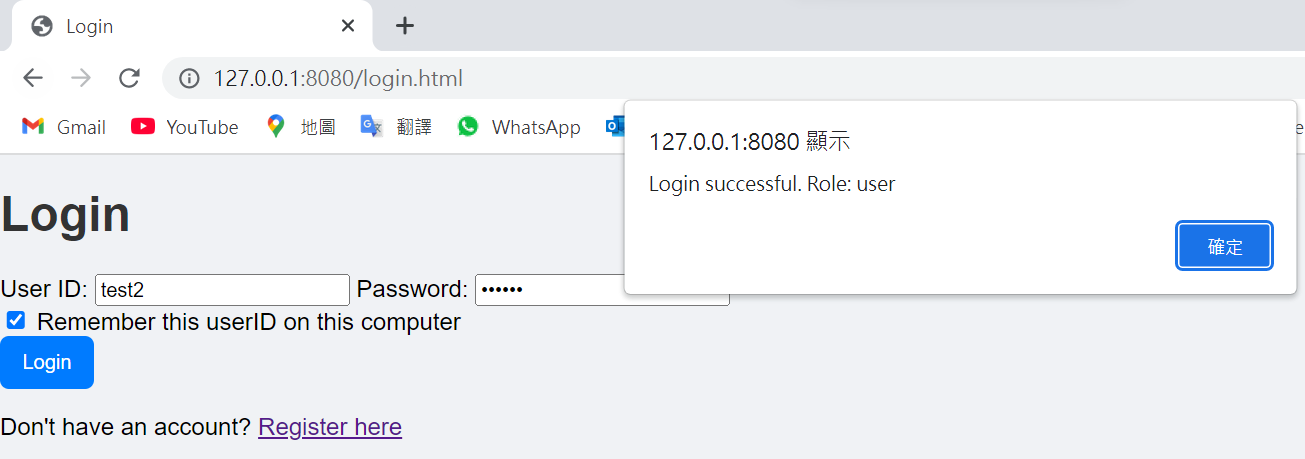
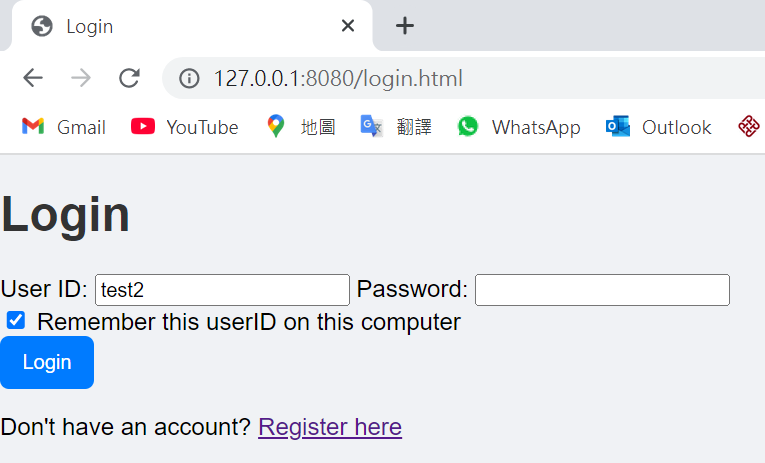


Valid admin login [role = admin]:

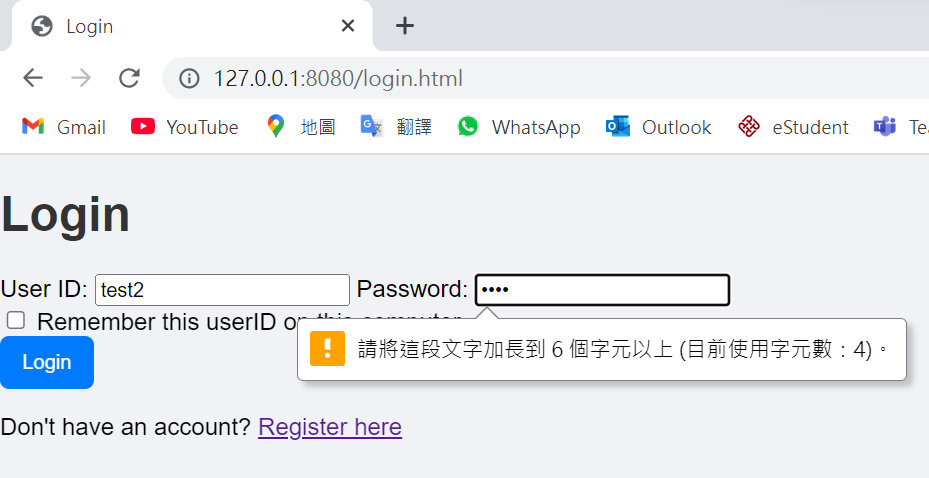
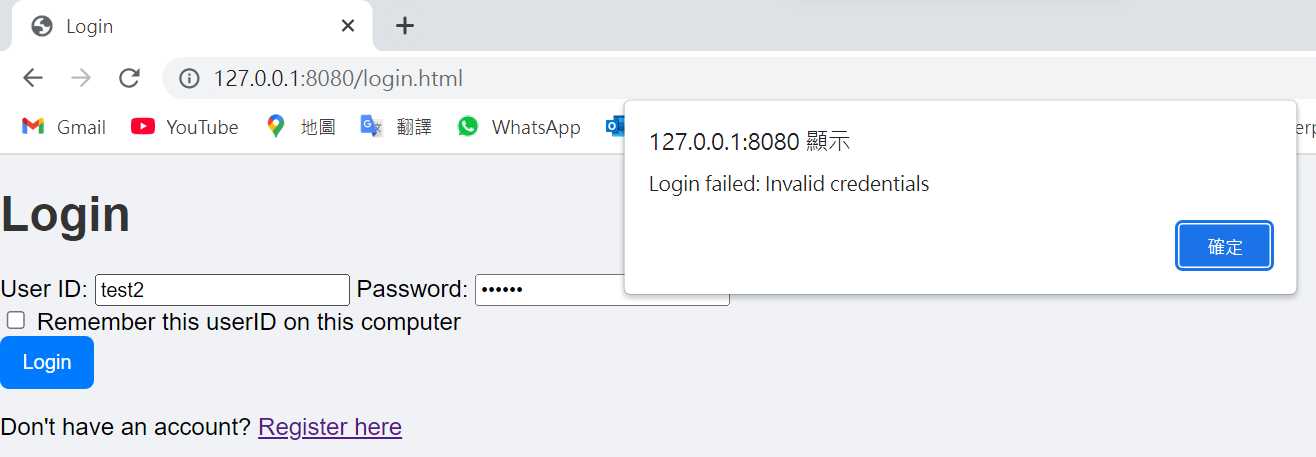


Valid user login (with ticking the option to remember login user ID):

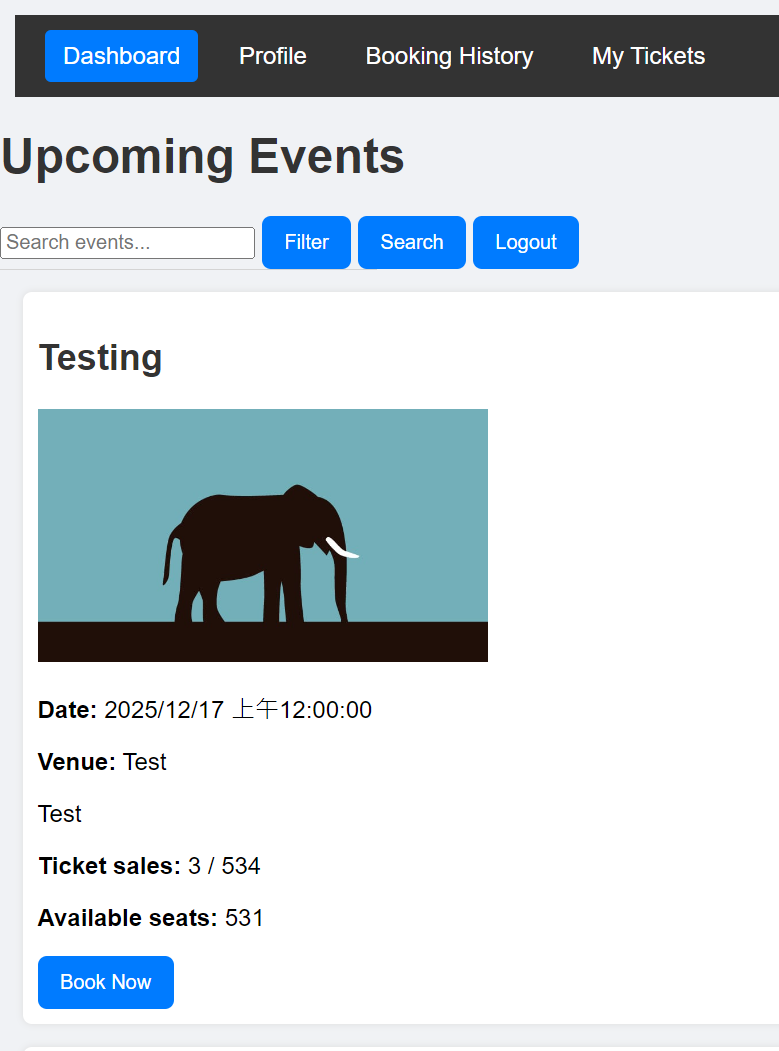
[After user clicking logout button, redirecting to login page and the user ID is generated automatically with previous login user ID.]

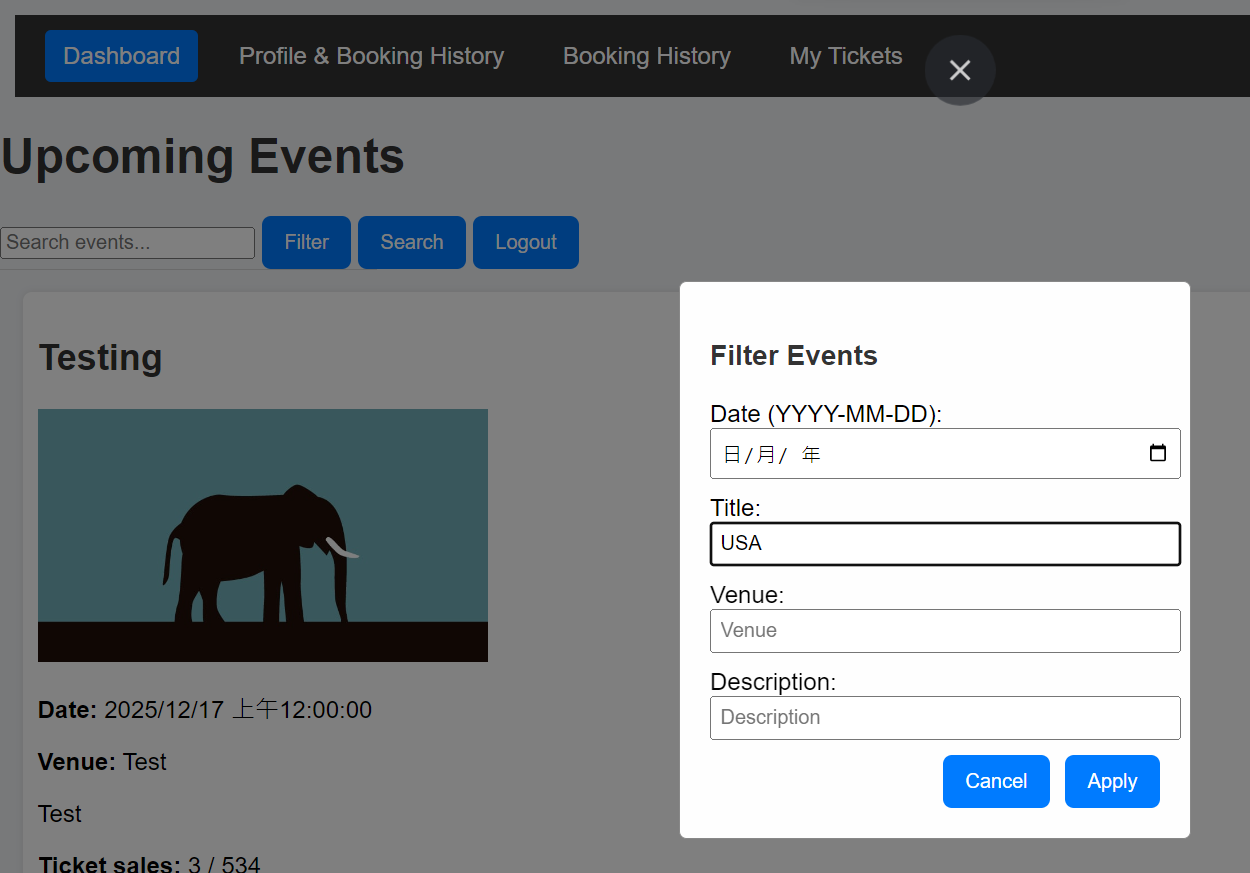
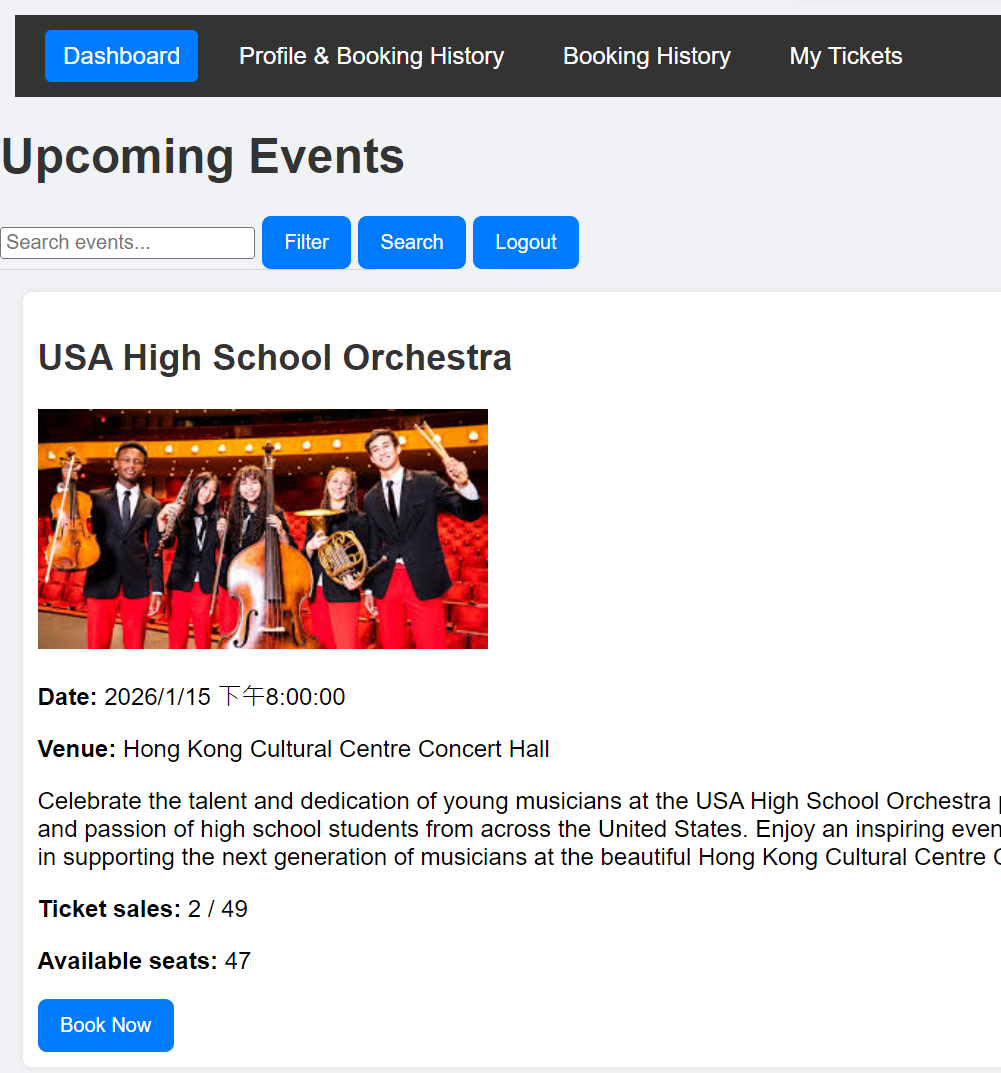
Invalid user login [password policy and wrong password]:

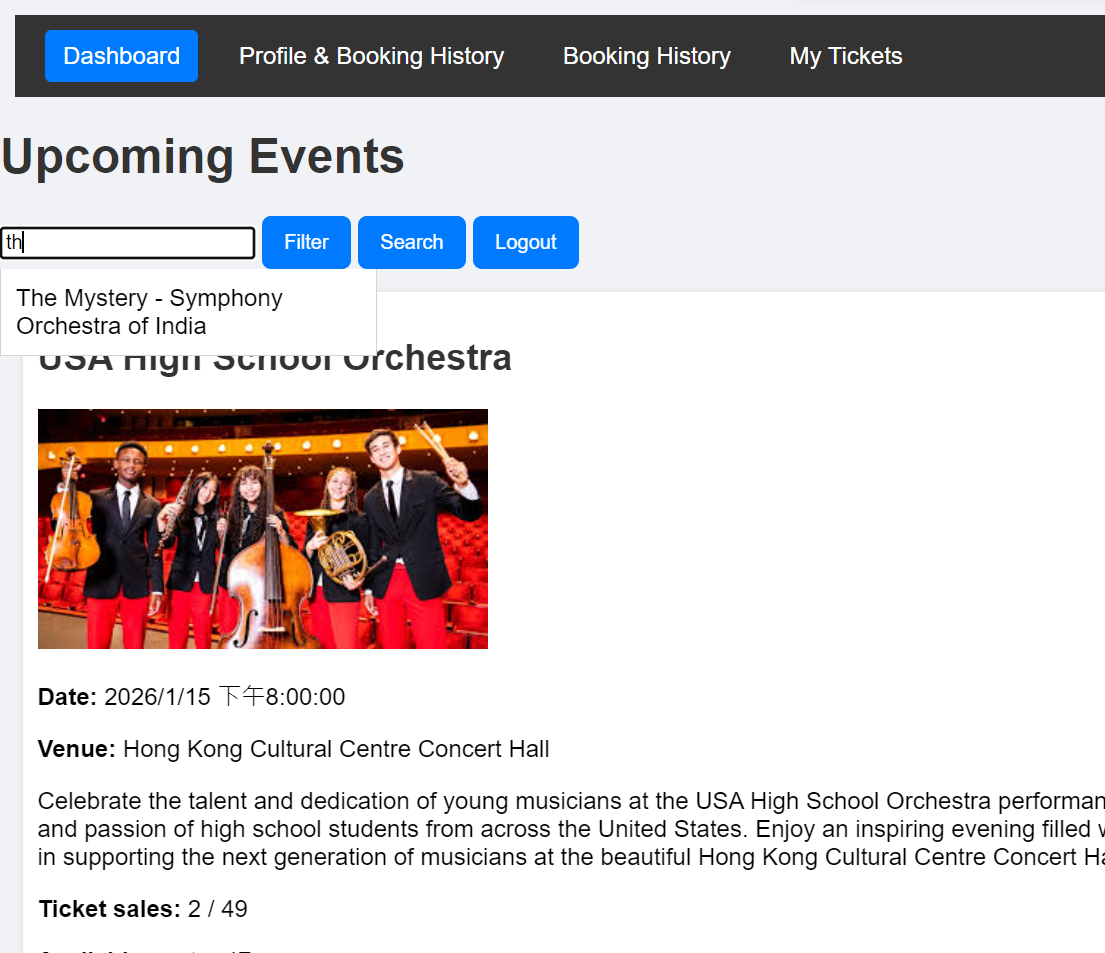
After login:



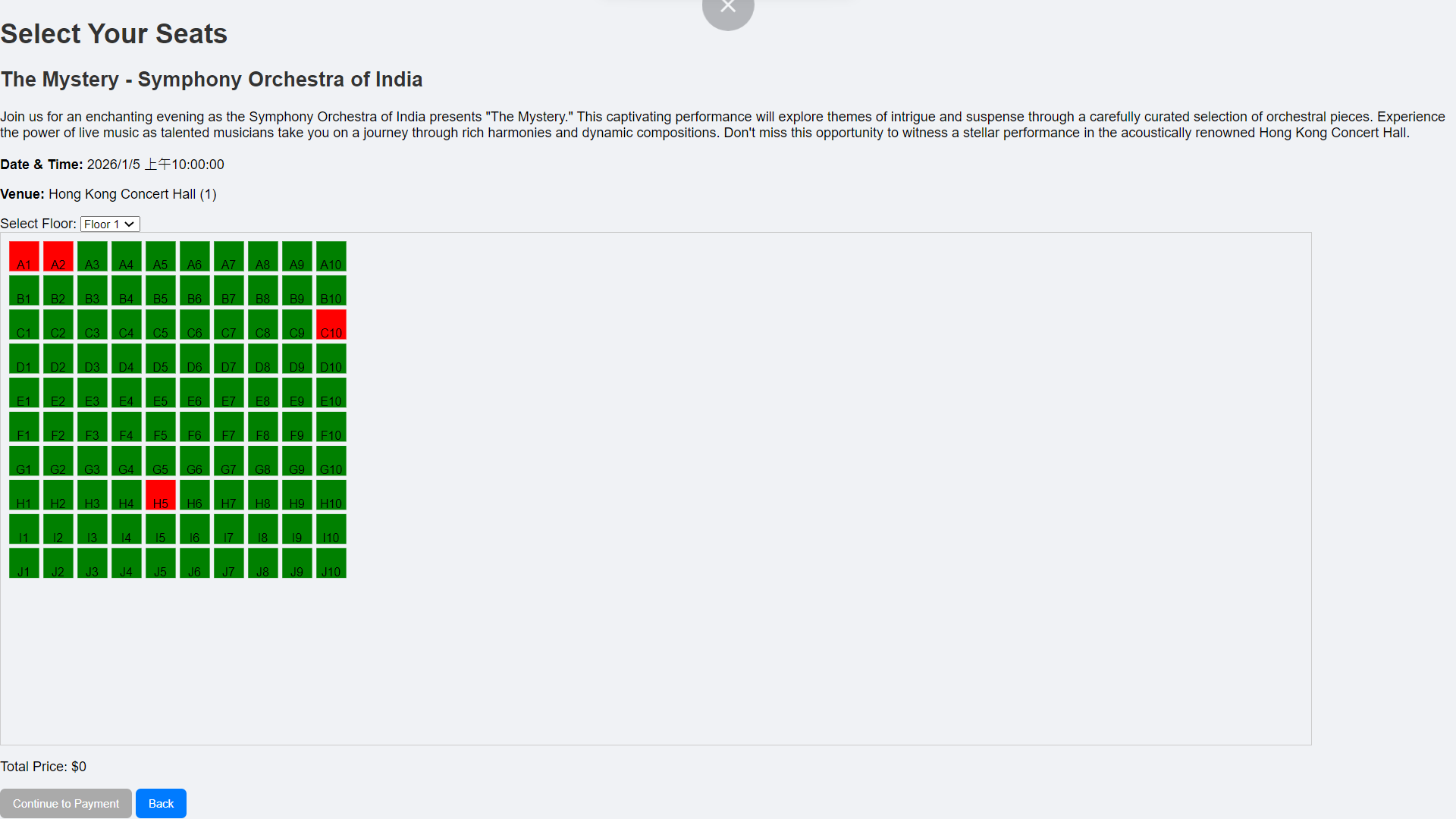
Event filter [before clicking ‘apply’ and after clicking ‘apply’ the filter]:

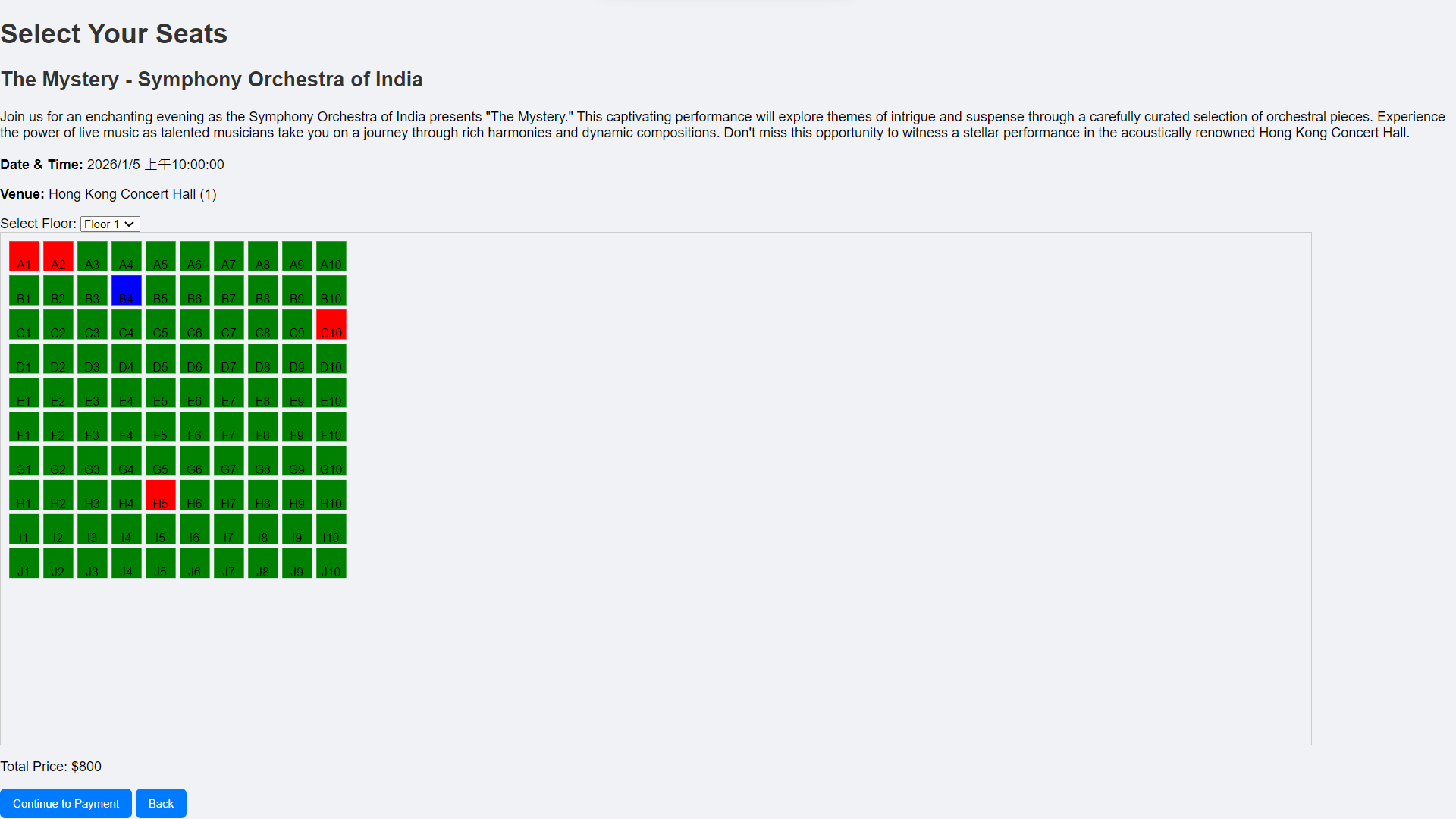
Event search [real-time event name suggestions during search – autocomplete]:

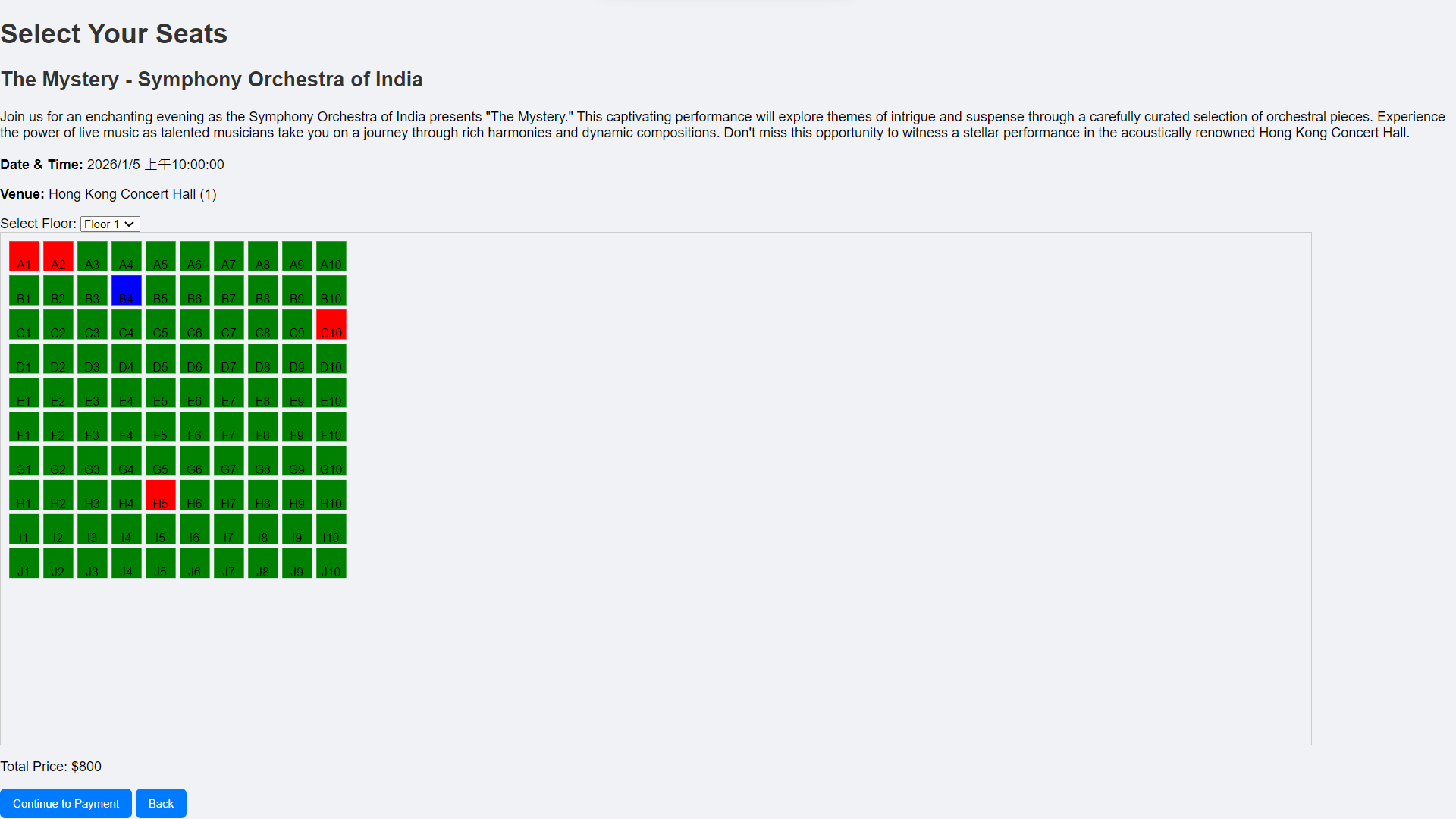
Seat selection & booking:



Selecting seat B4 (premium seat with price $800):



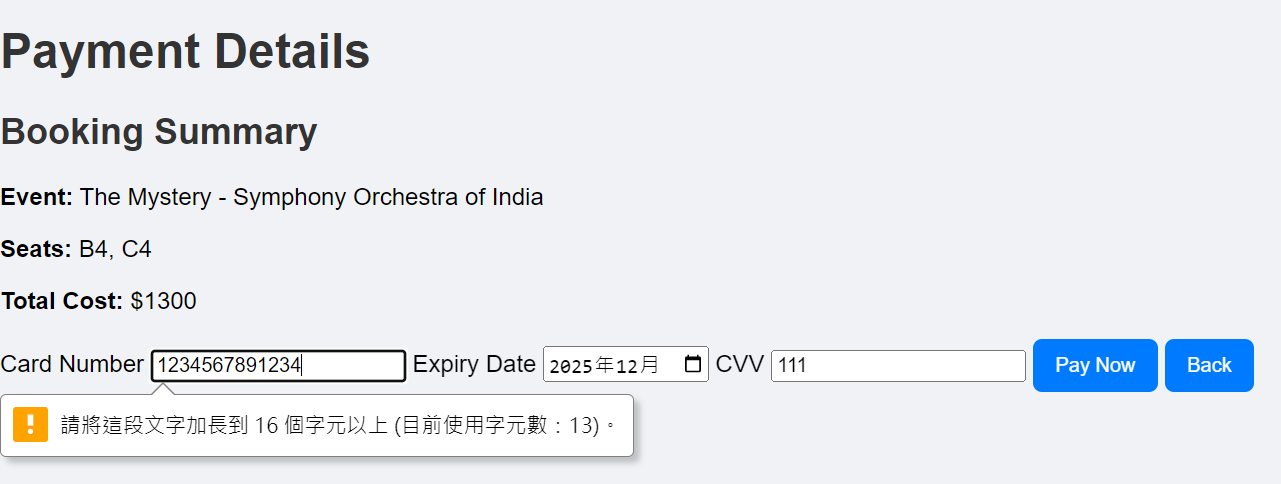
Selecting seat C4 (economy seat with price $500):



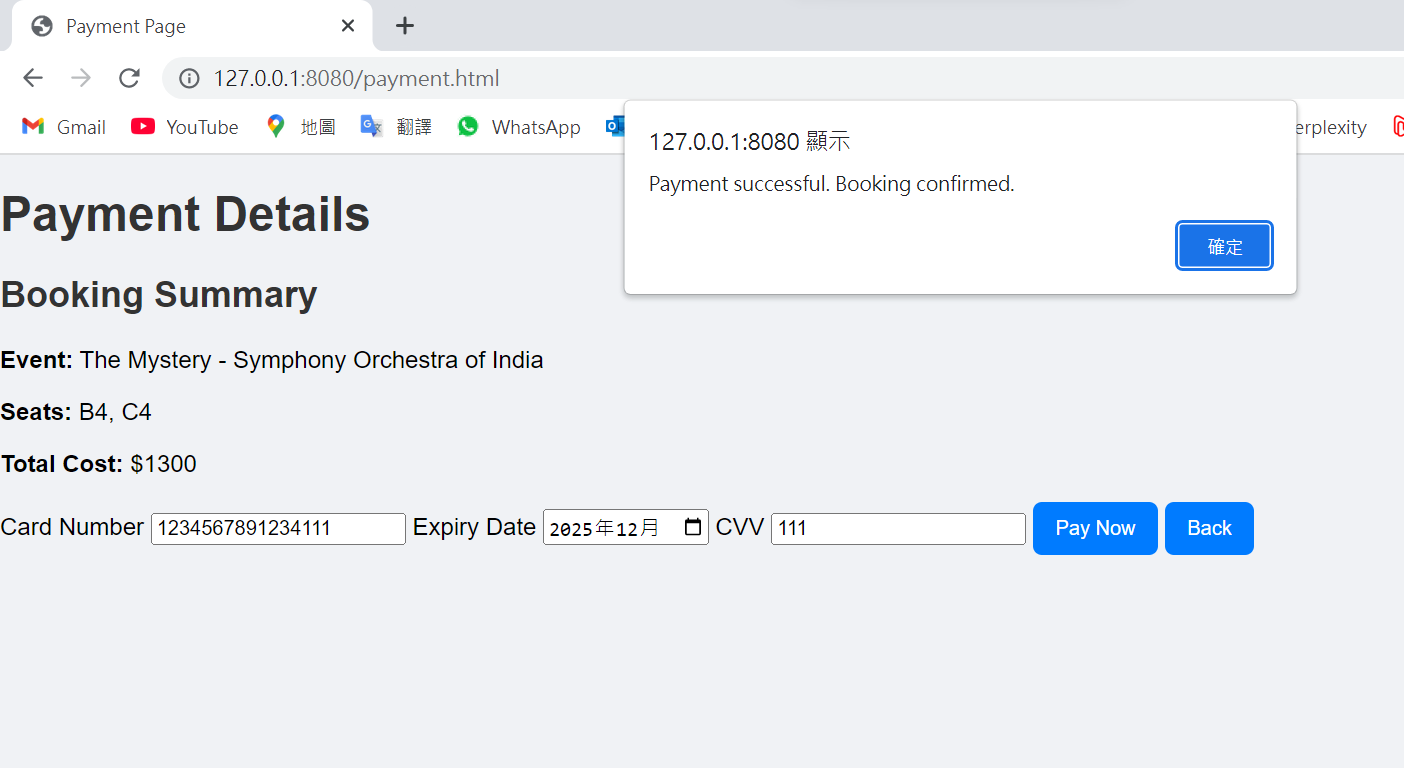
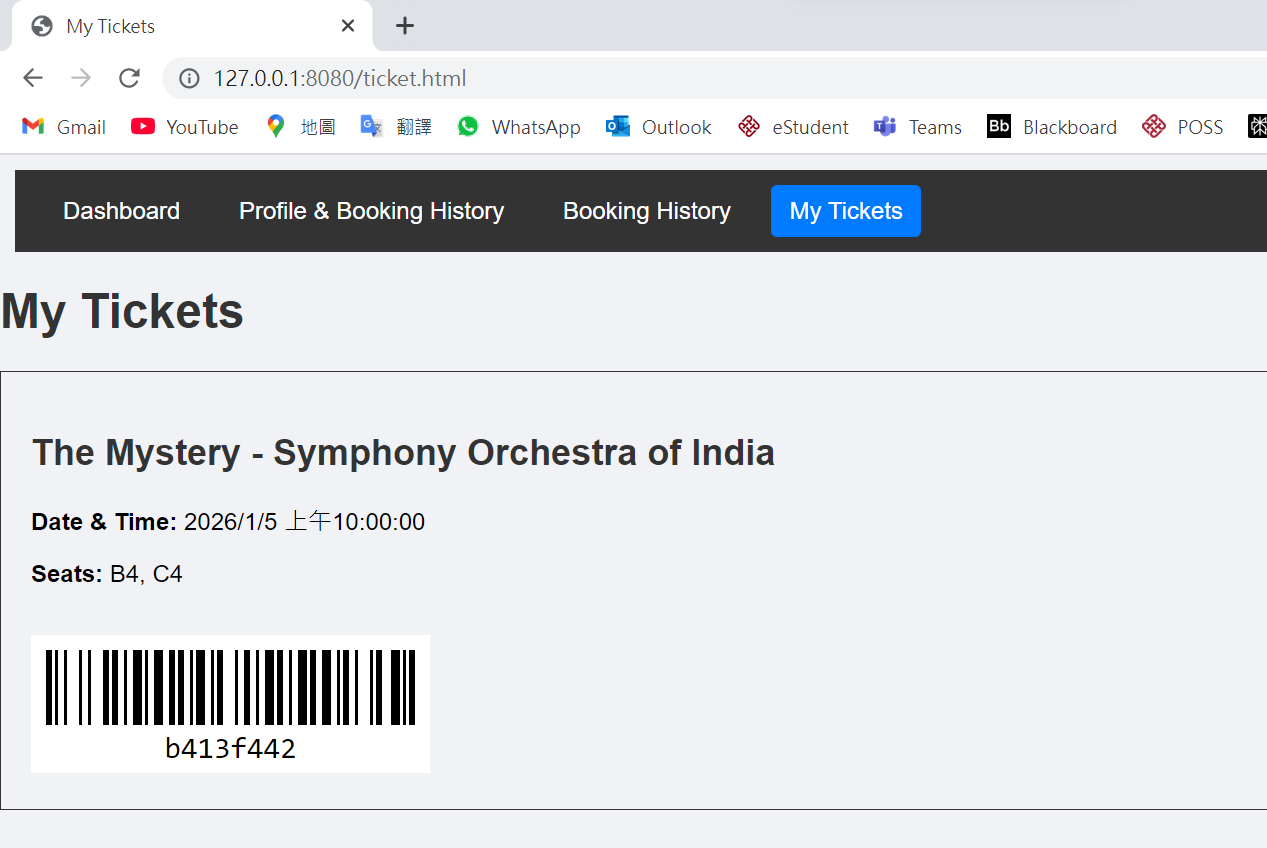
After clicking ‘continue payment’, navigate to payment page:

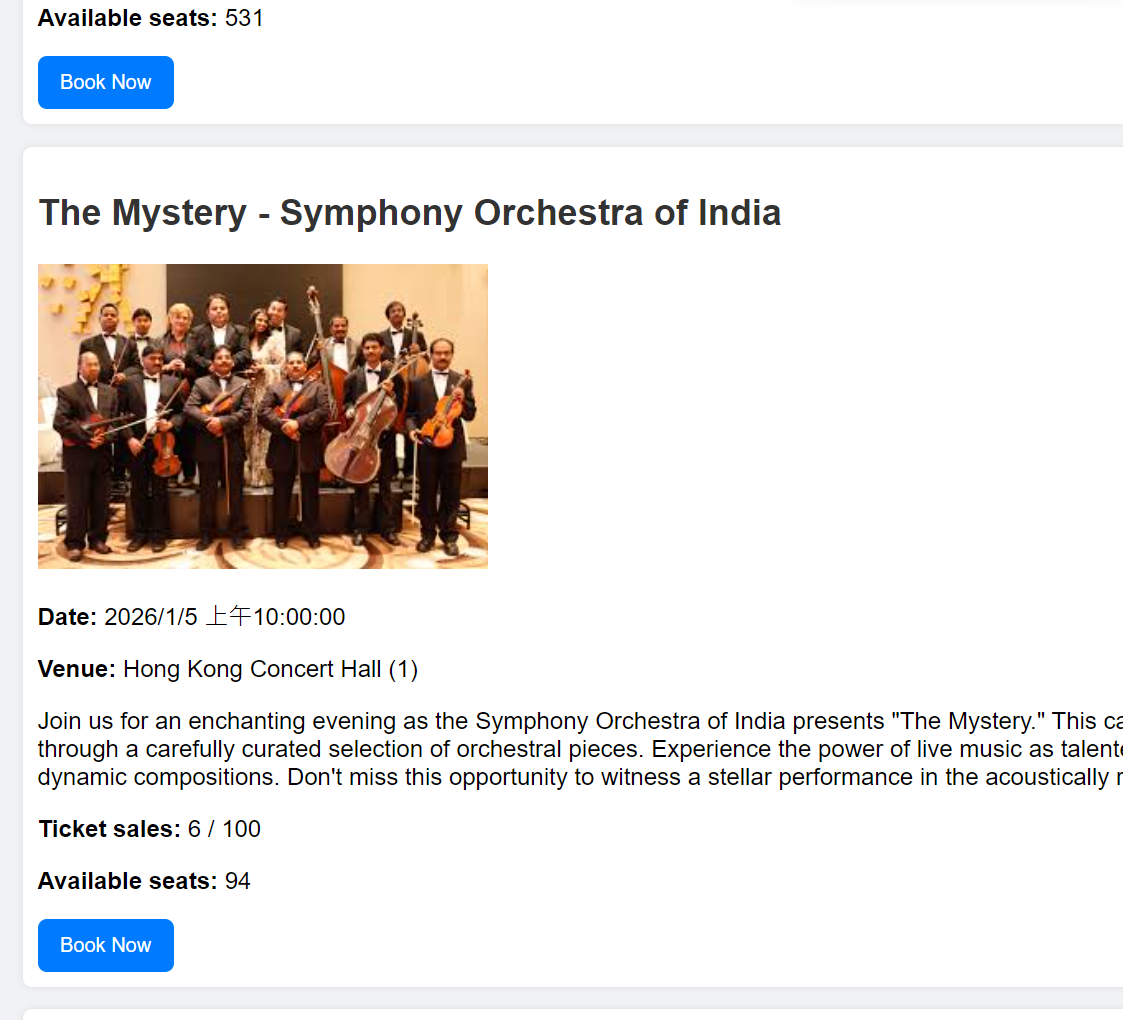


Inputting the card number fewer than 16 number:

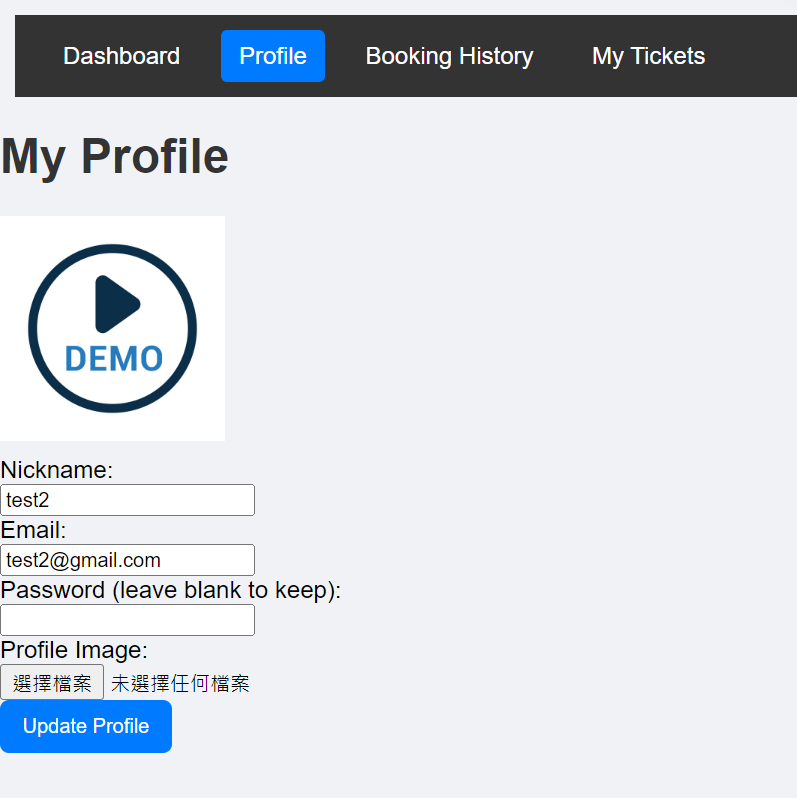


After filling in the correct format, and click ‘pay now’, it will issue a electronic tickets and redirecting to ticket page:

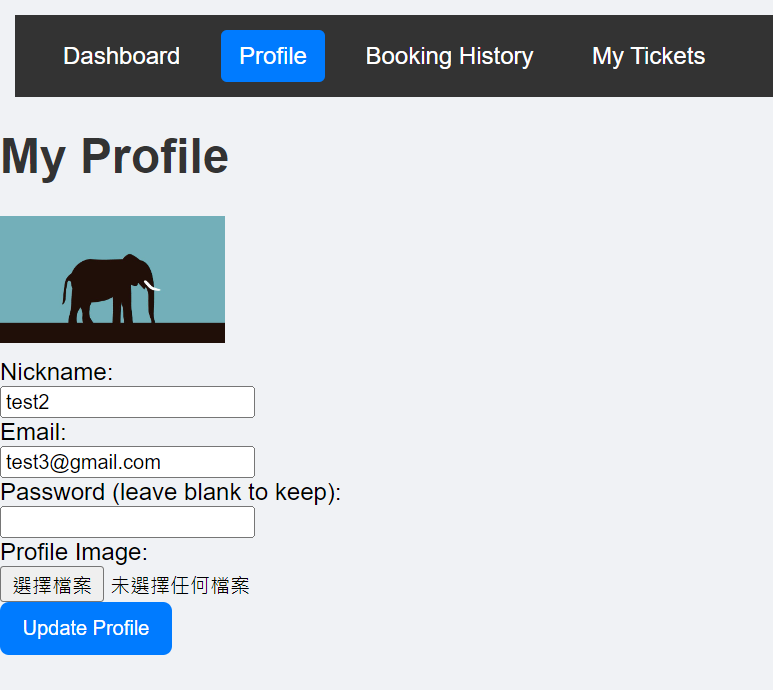
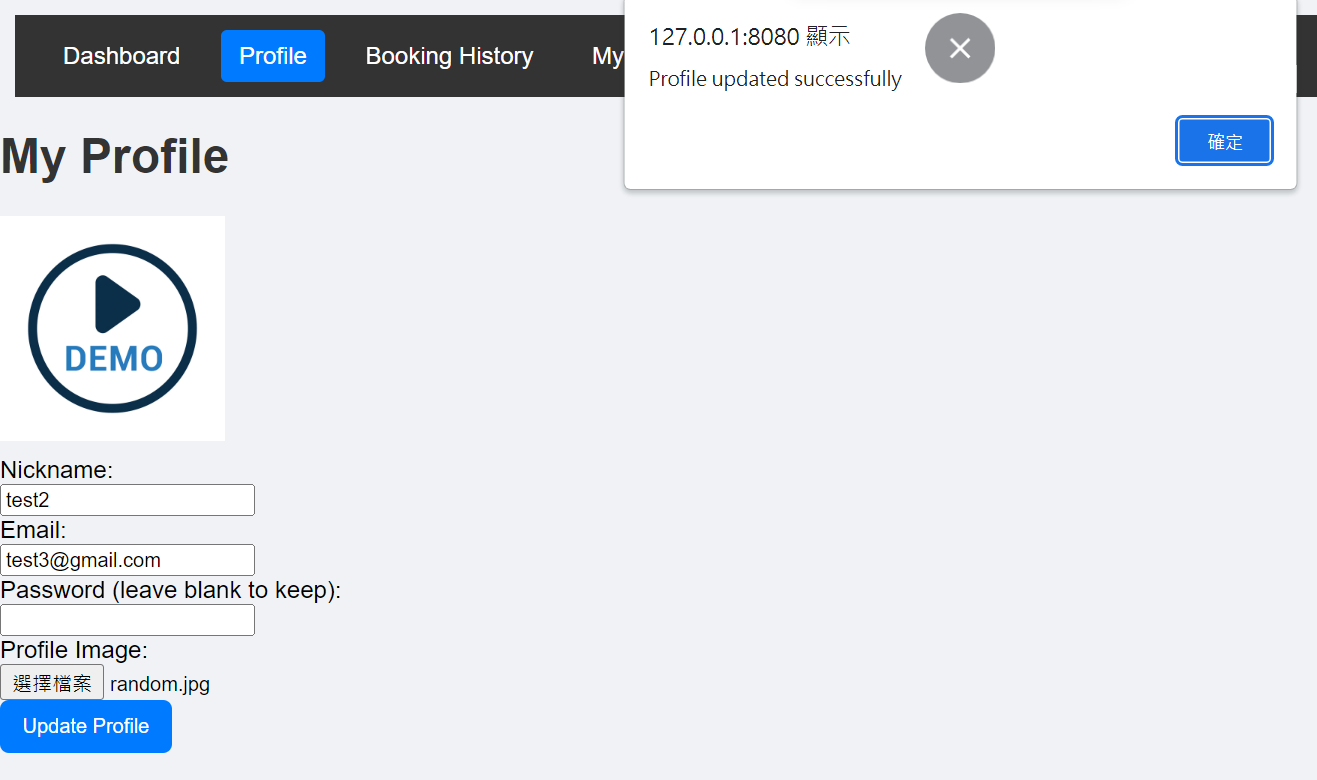
 

The booked event updated the tickets sales and available seats:  


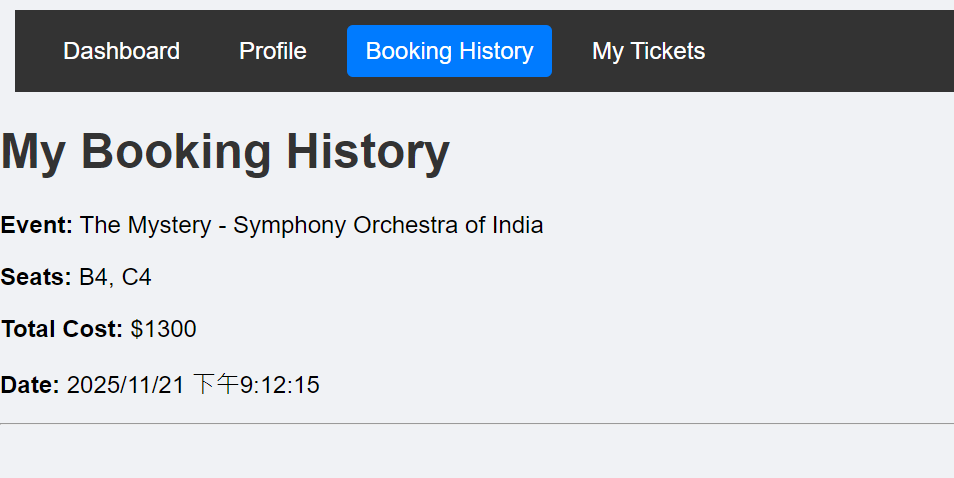
Profile page:



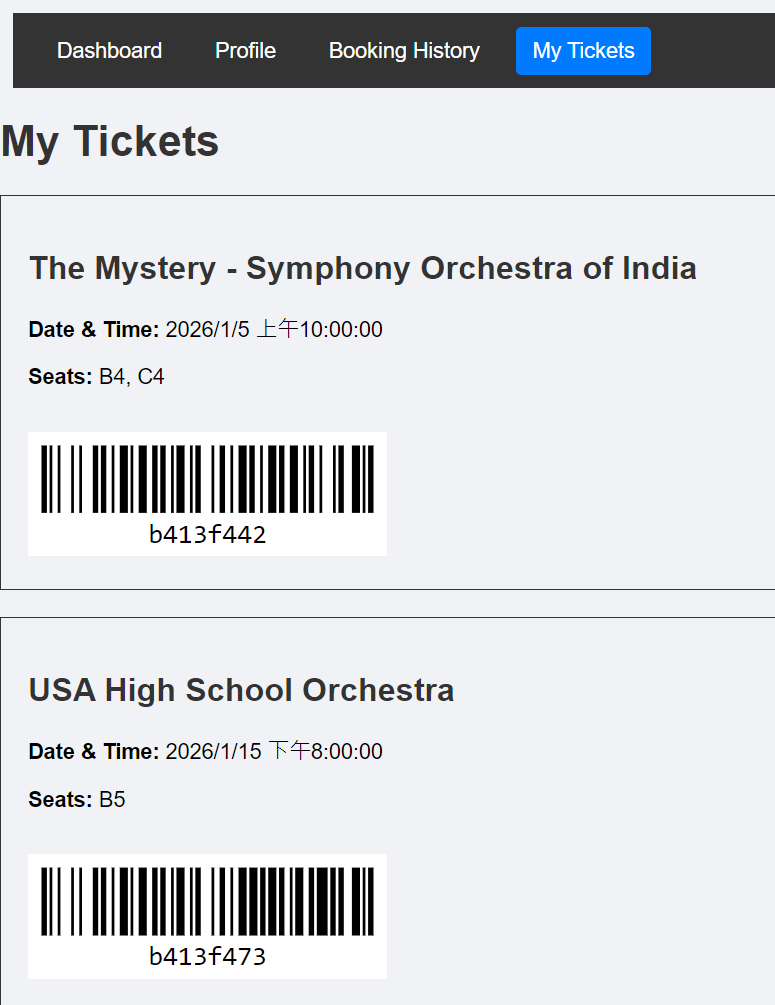
Update the email to ‘test3@gmail.com’ and update the profile to another image:



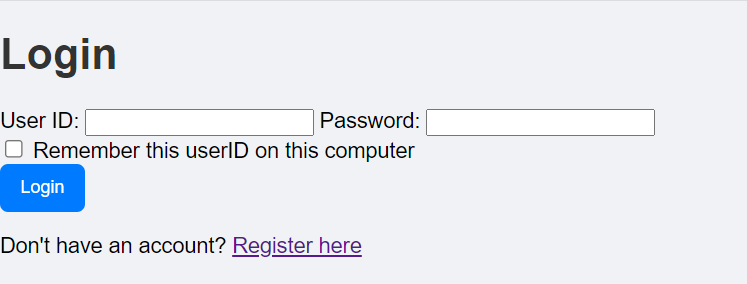
Booking history page:



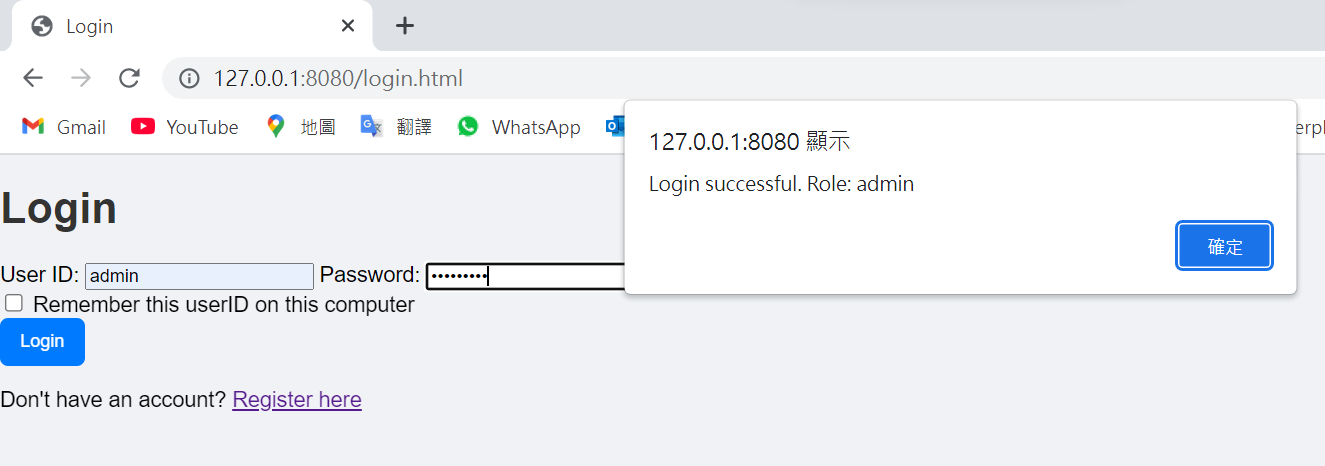
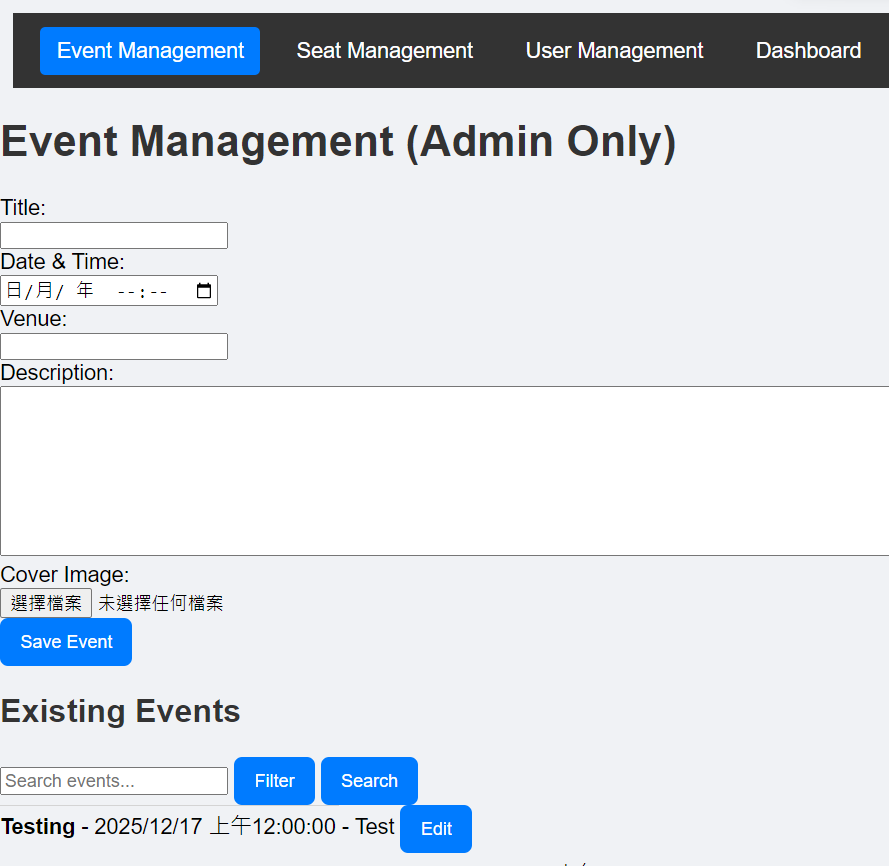
After booking one more event:

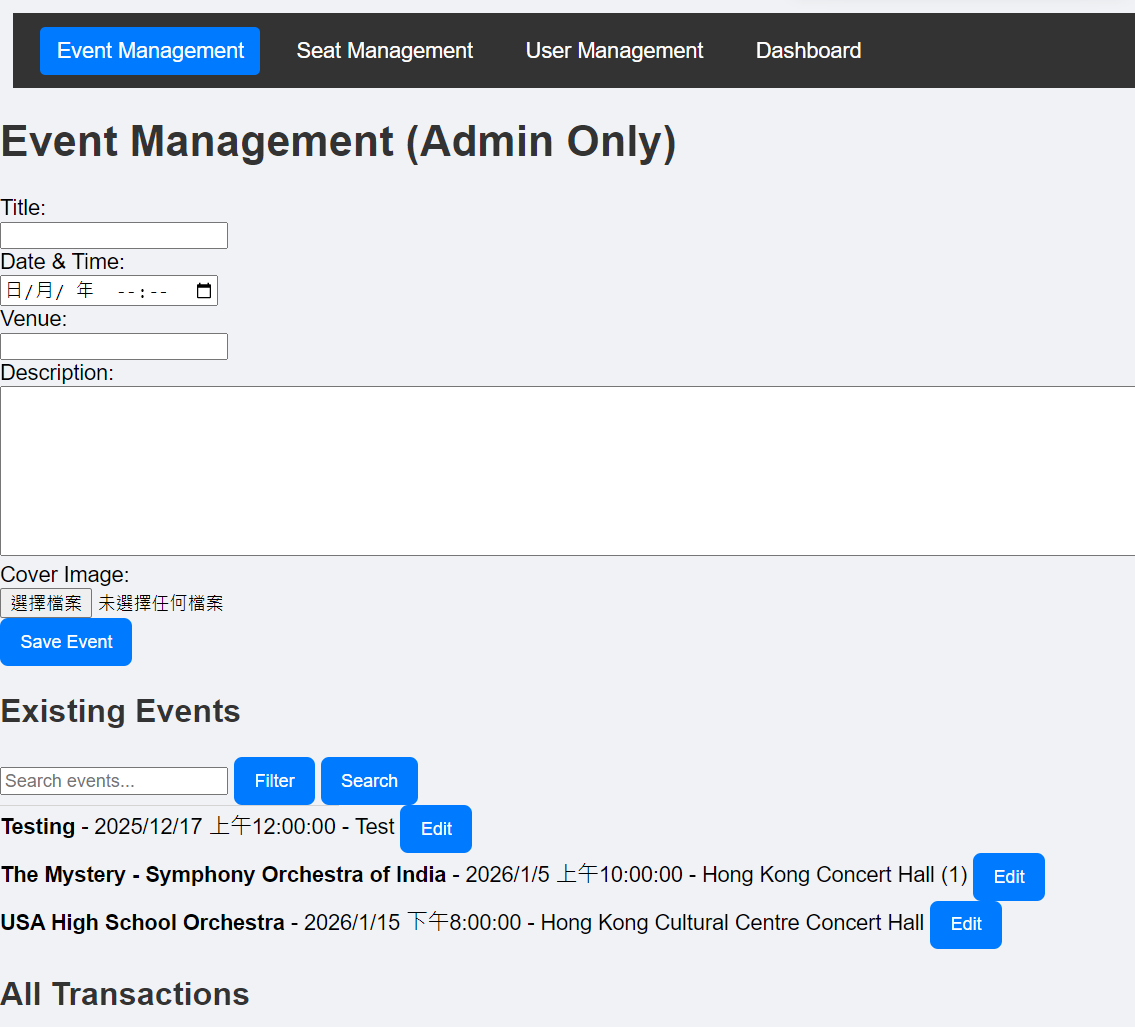
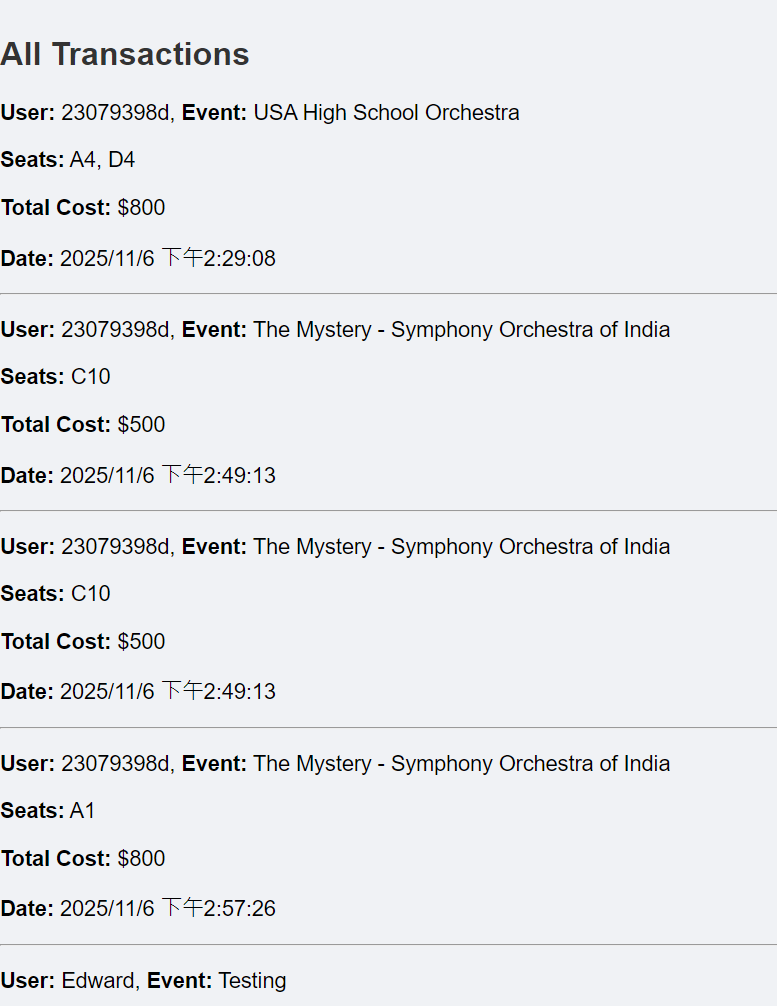
After clicking ‘logout’ in event dashboard, redirecting to login page:



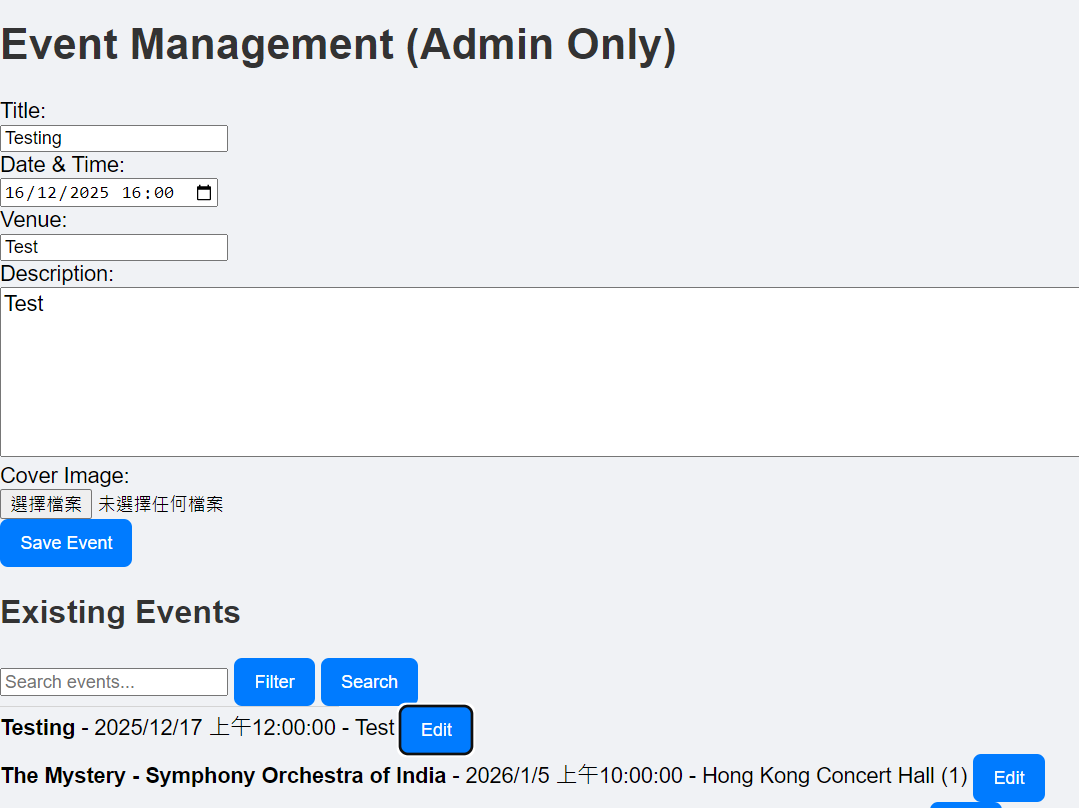
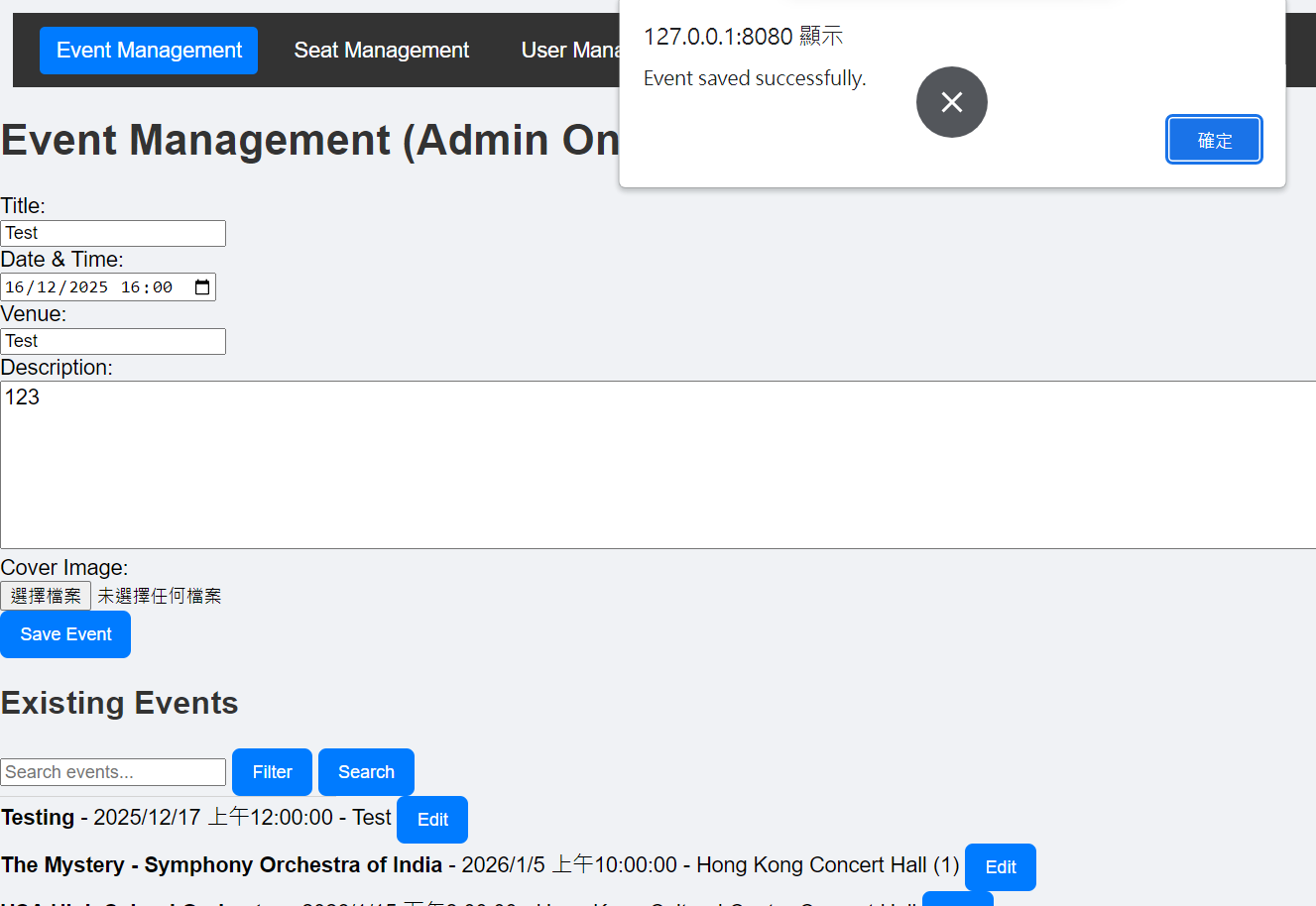
Login as admin, and you will see the admin only page:

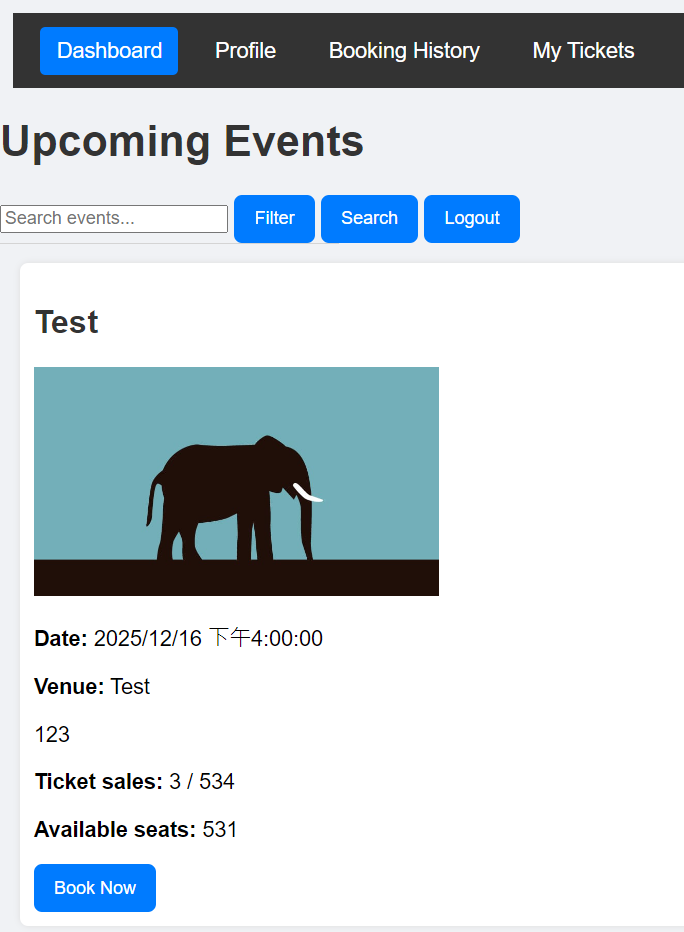
 

Event management page (admin only):

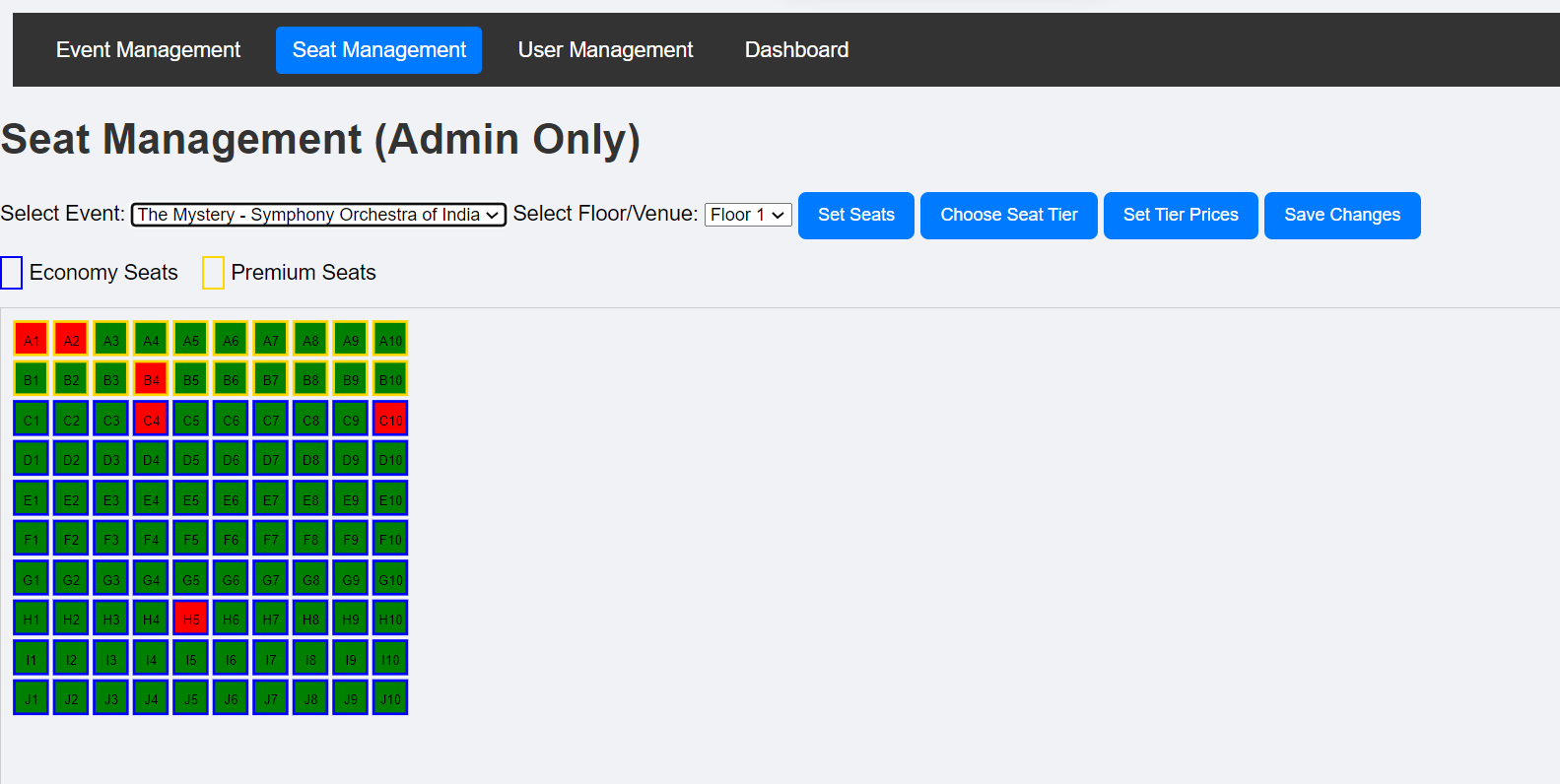
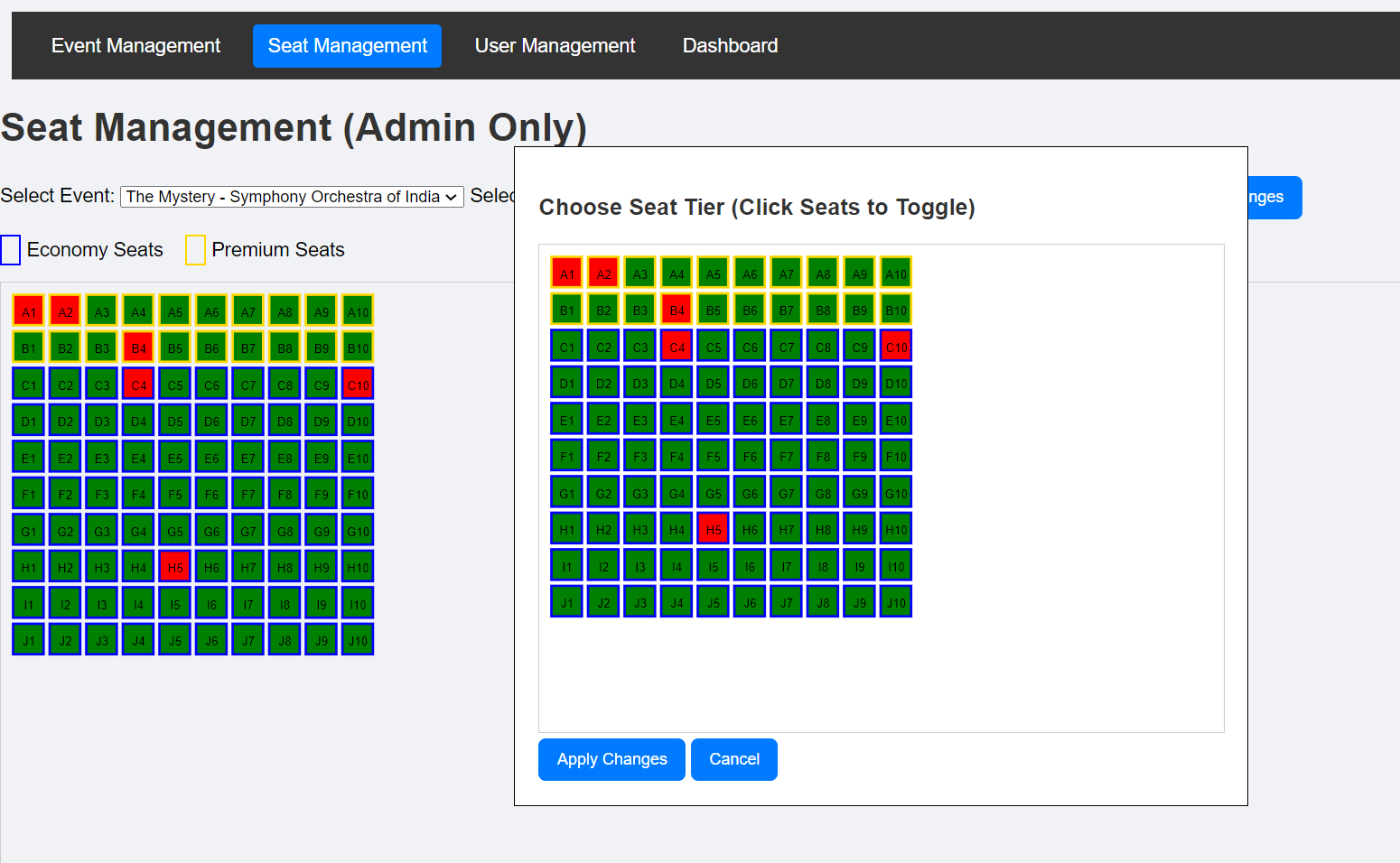
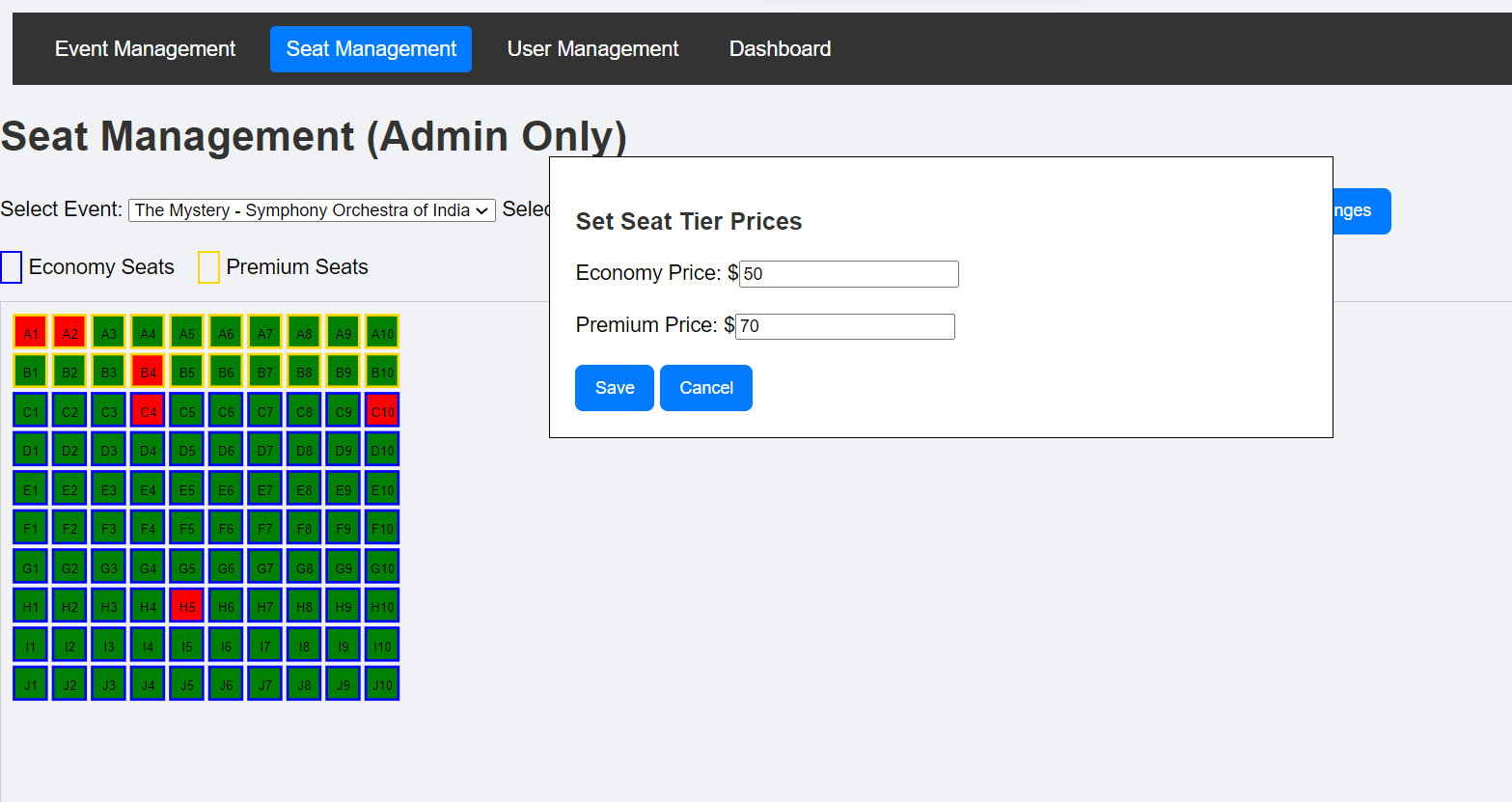
Update the event [title from ‘Testing’ to ‘Test’, and Description from ‘Test’ to ‘123’]:



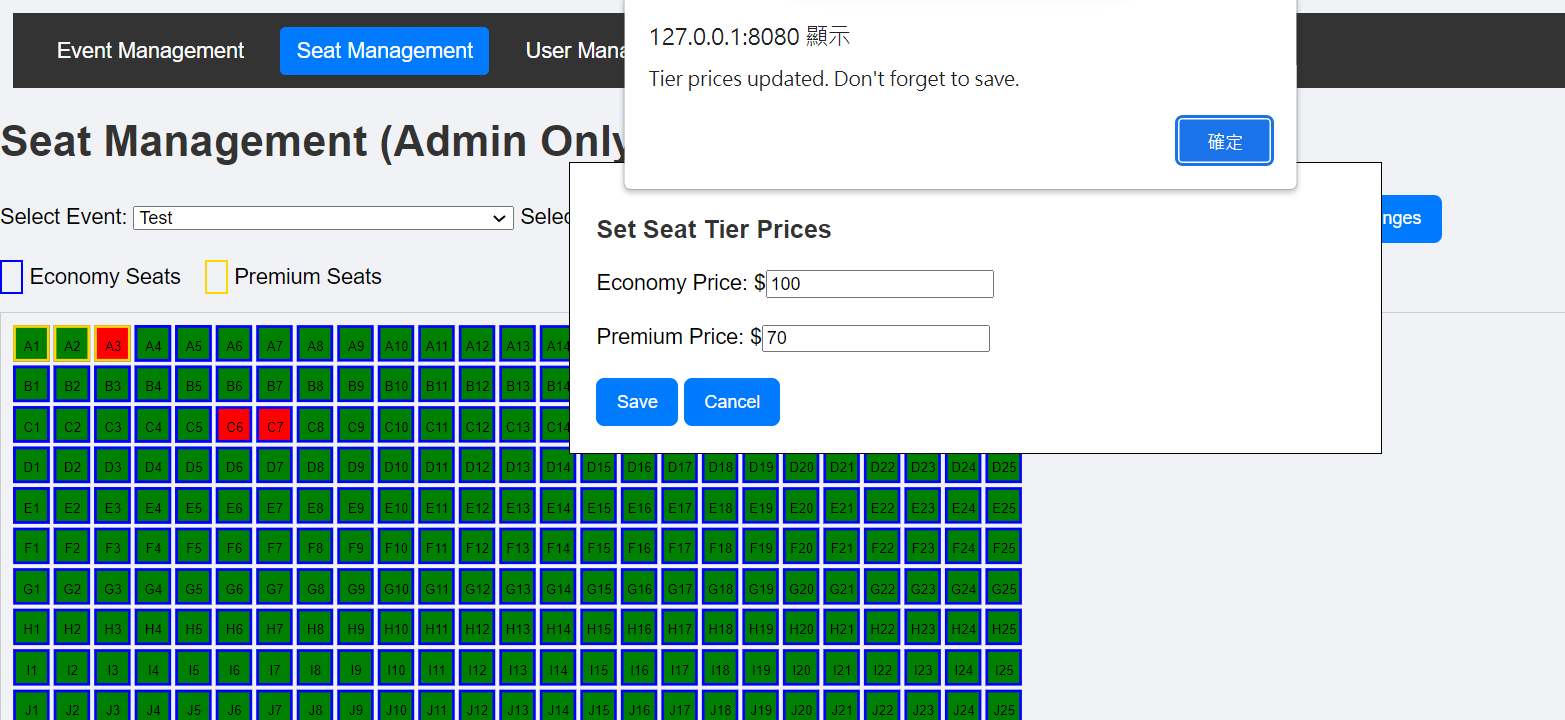
Seat management page (admin only)

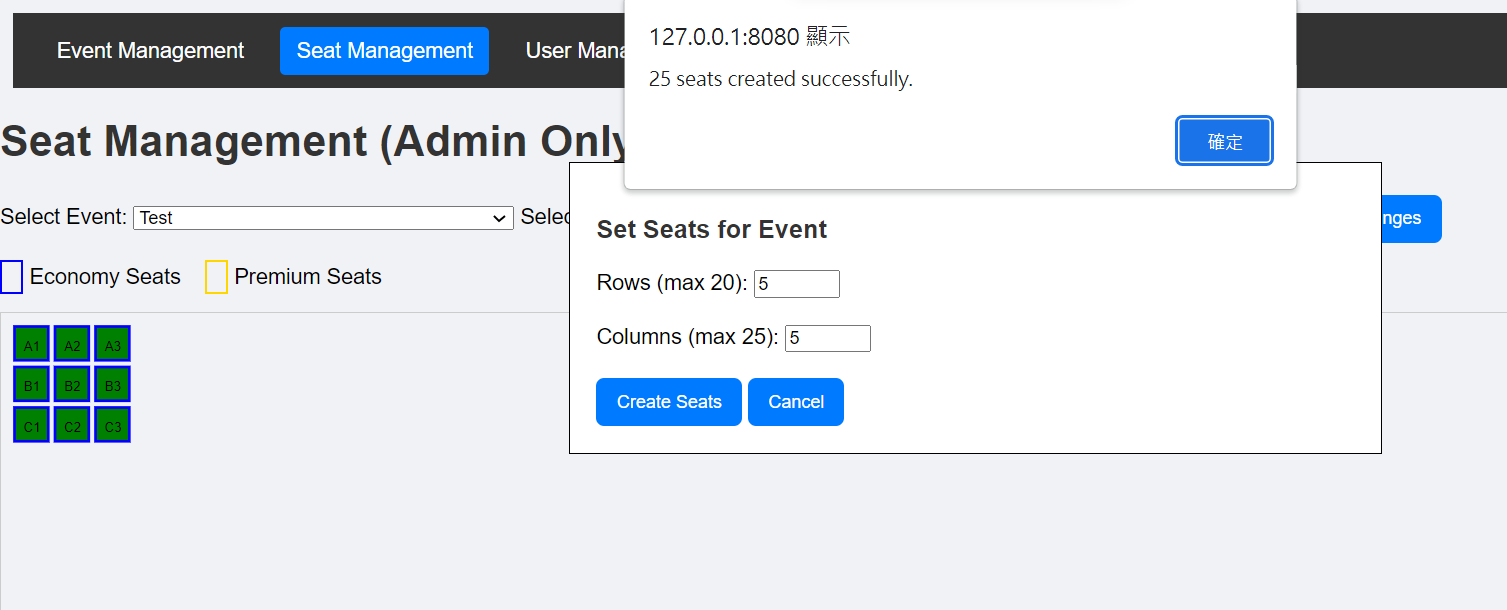
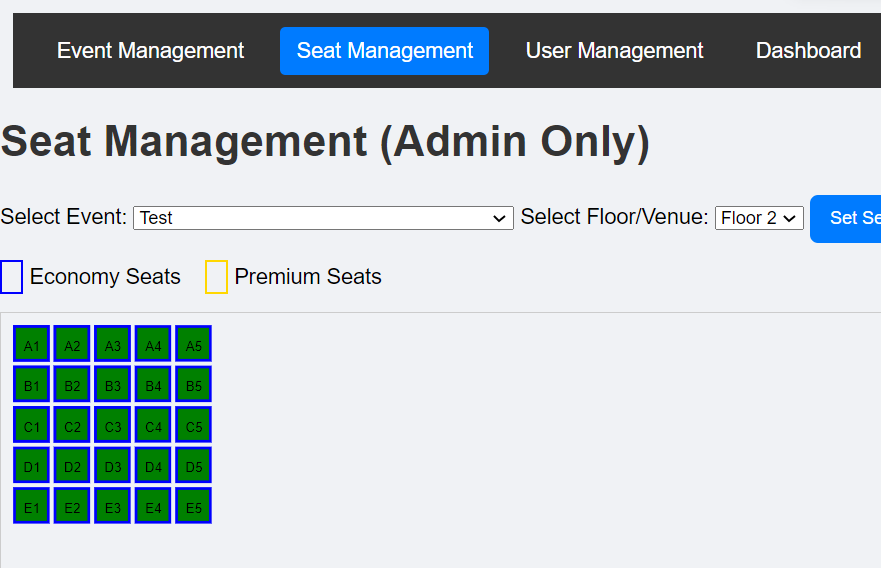
[Set seats, Choose seat tier, Set tier price]:

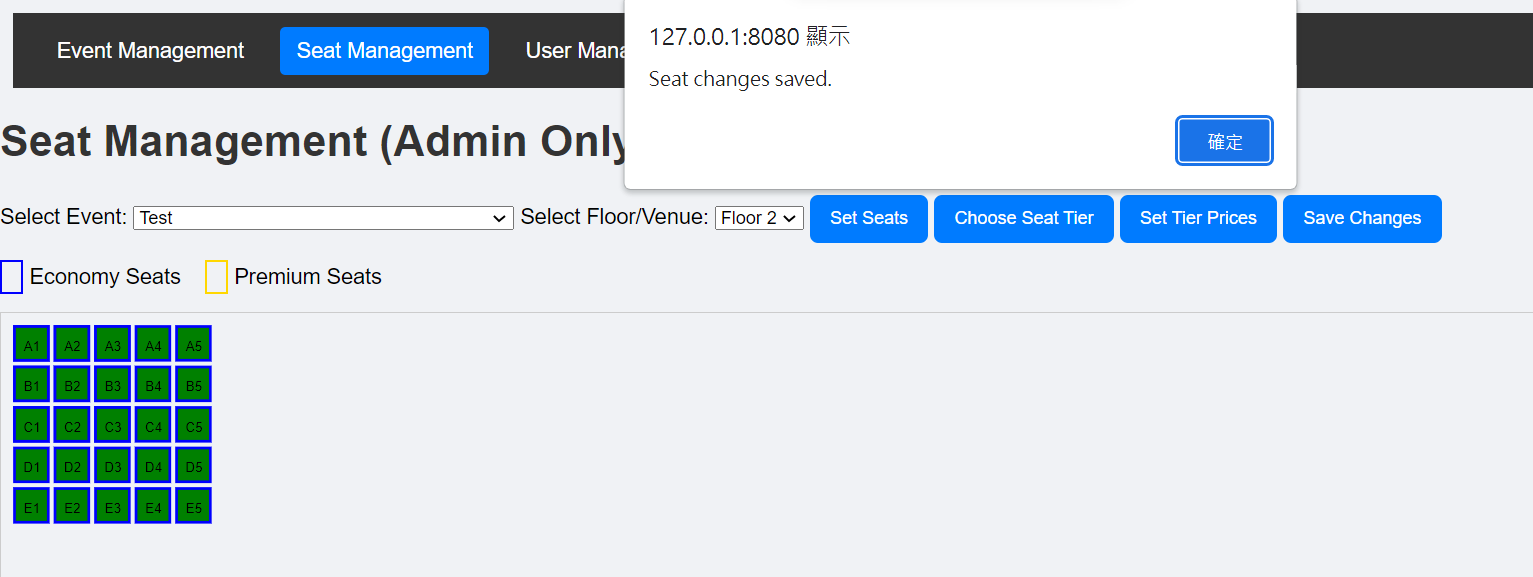
   

Seat management page (admin only)

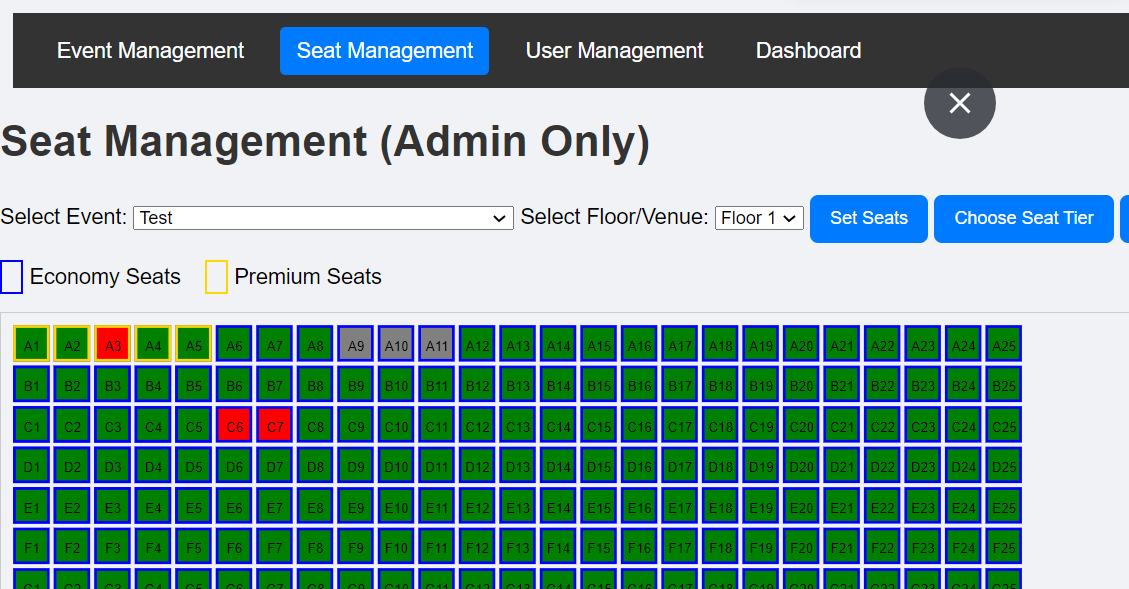
[update all options]:



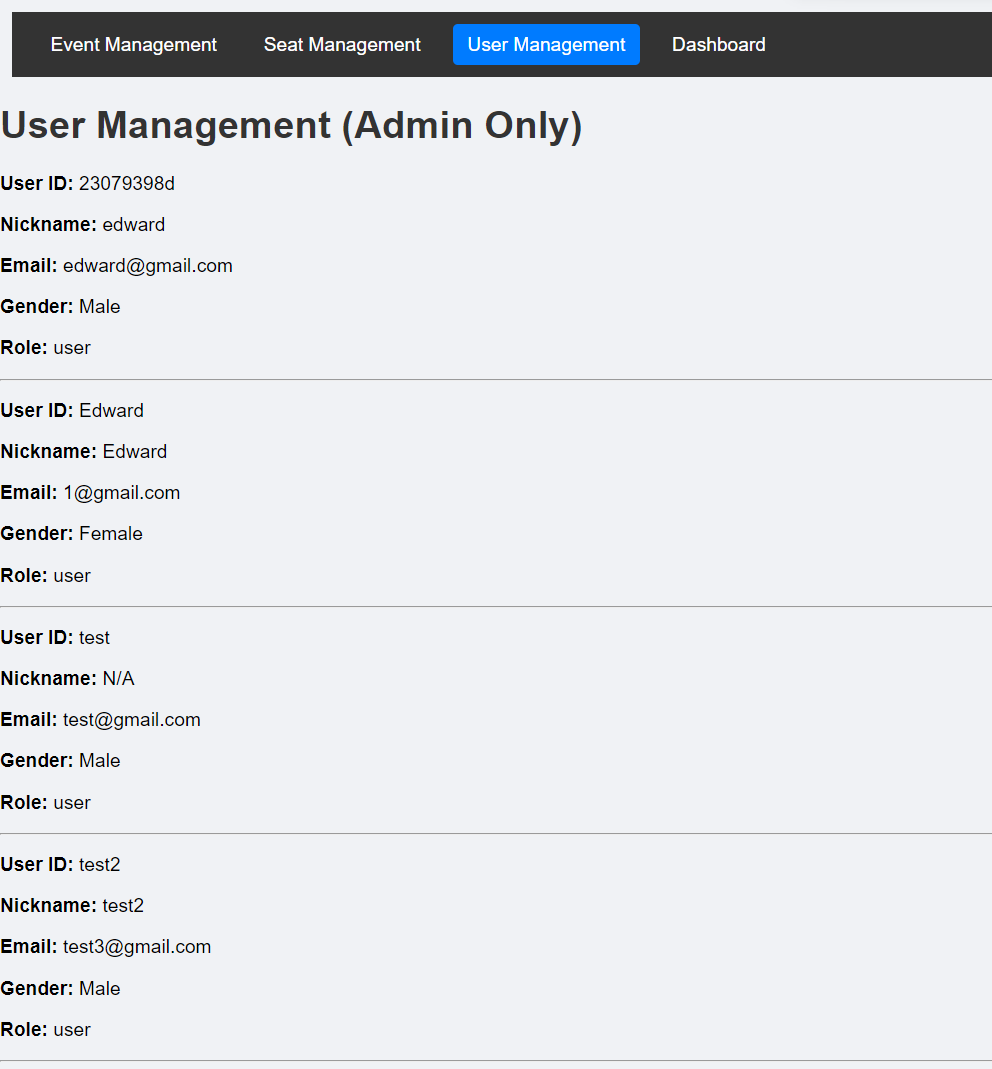
 



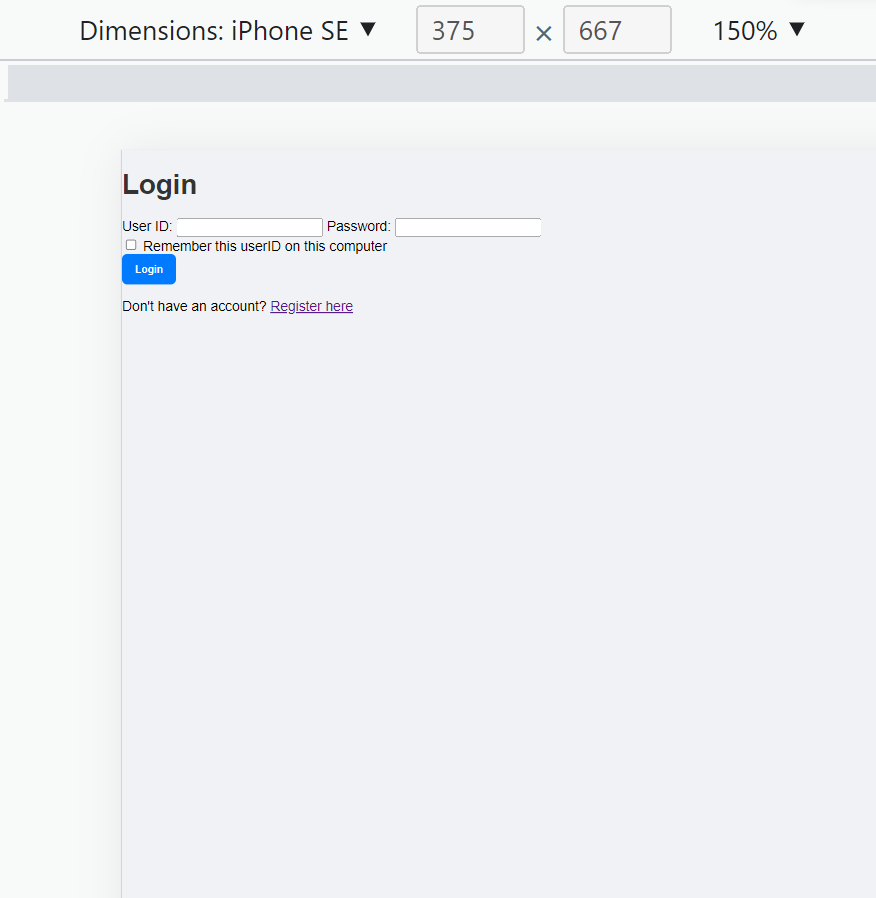
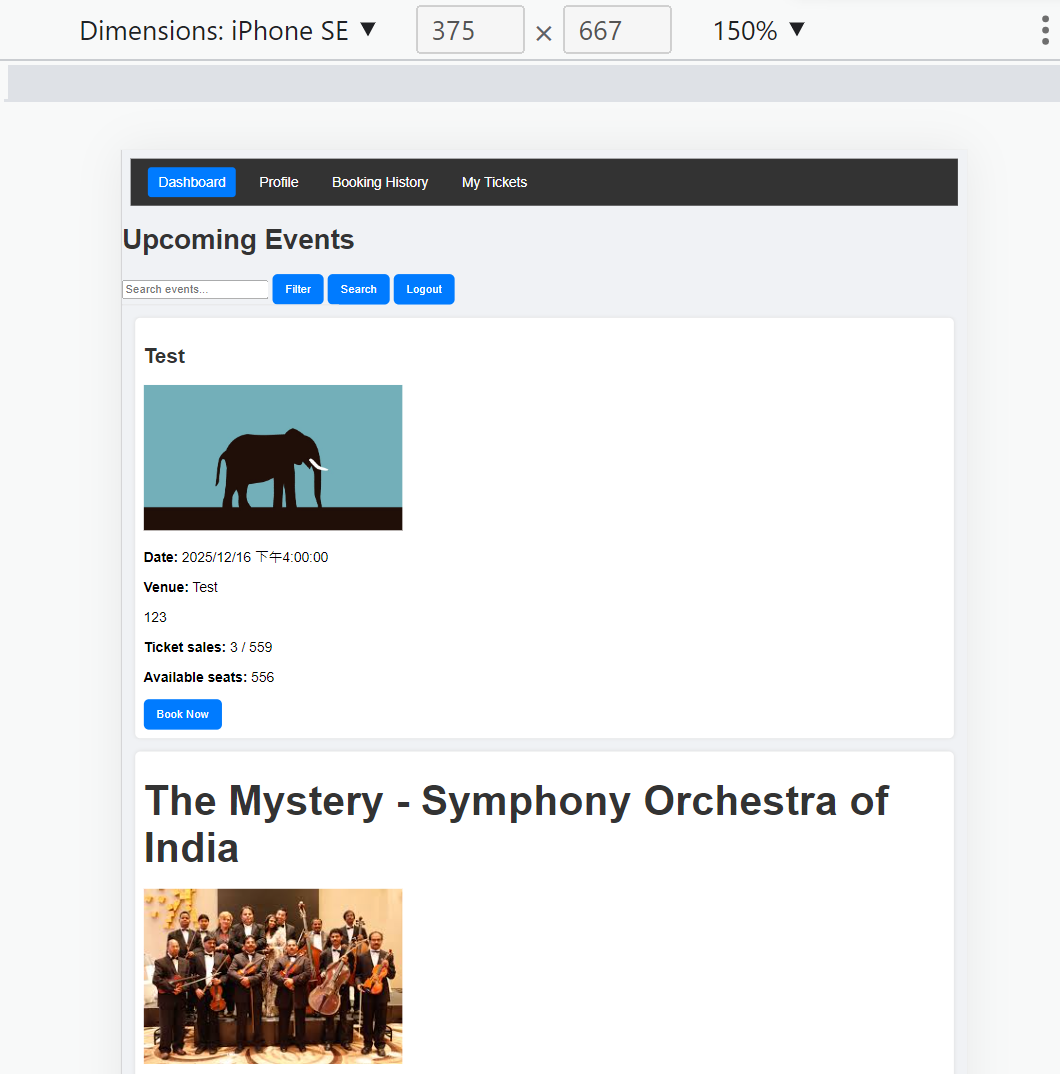
Disable seat A9 A10 A11:

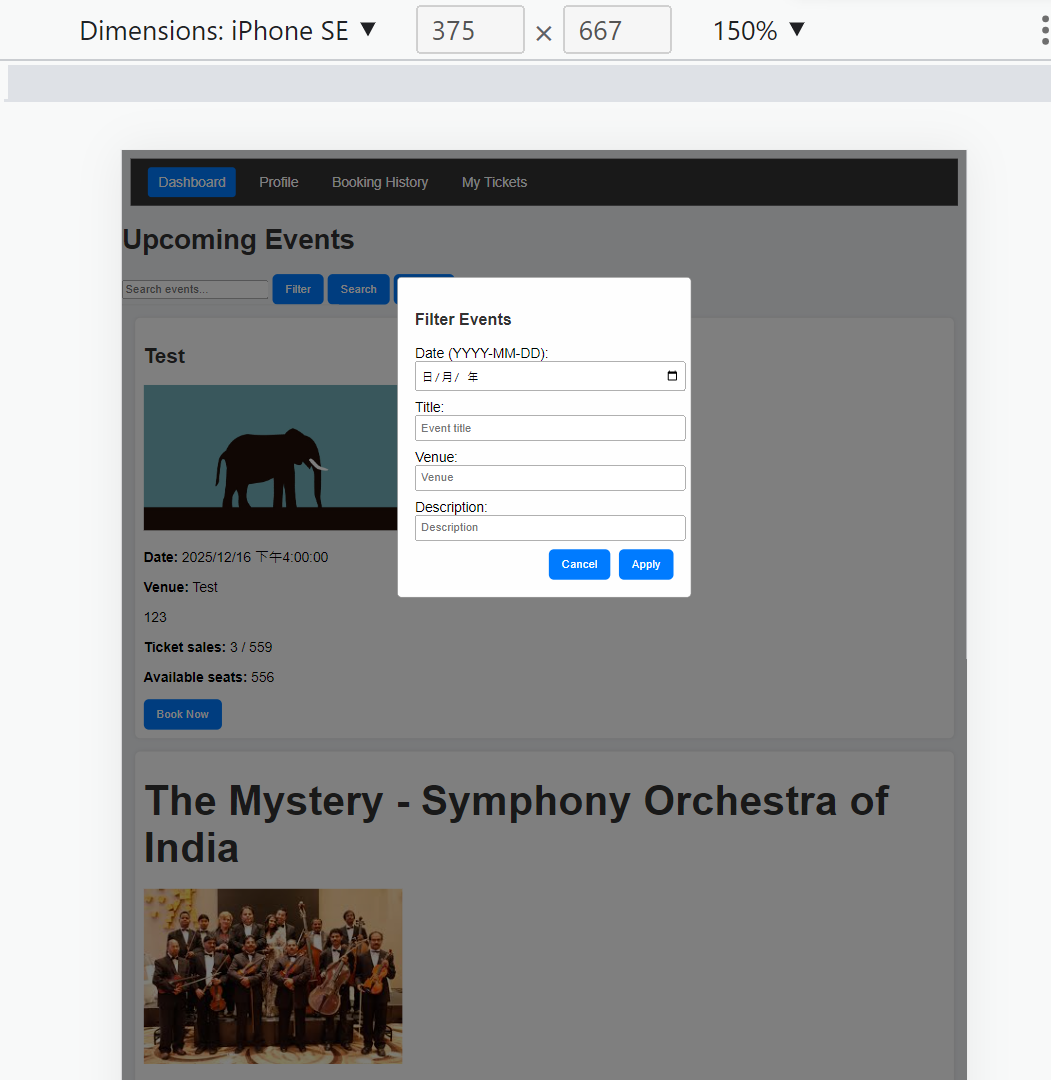
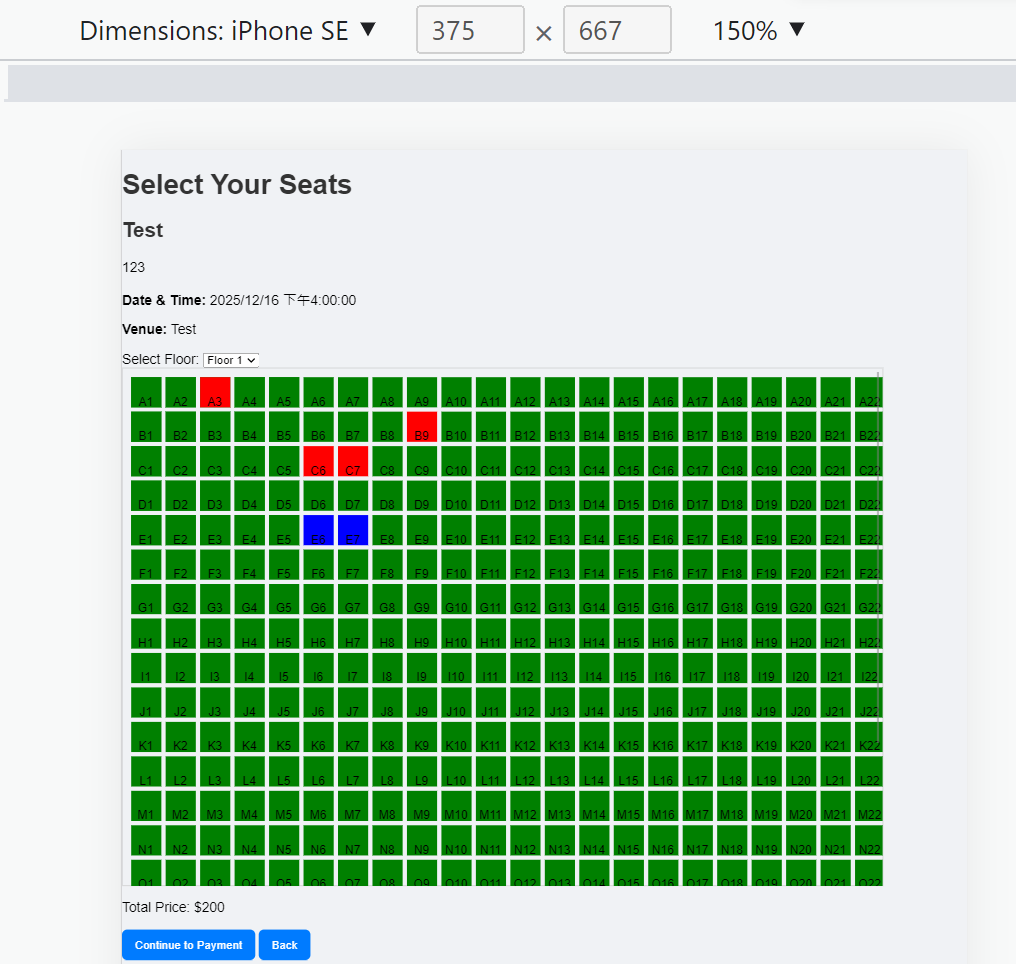


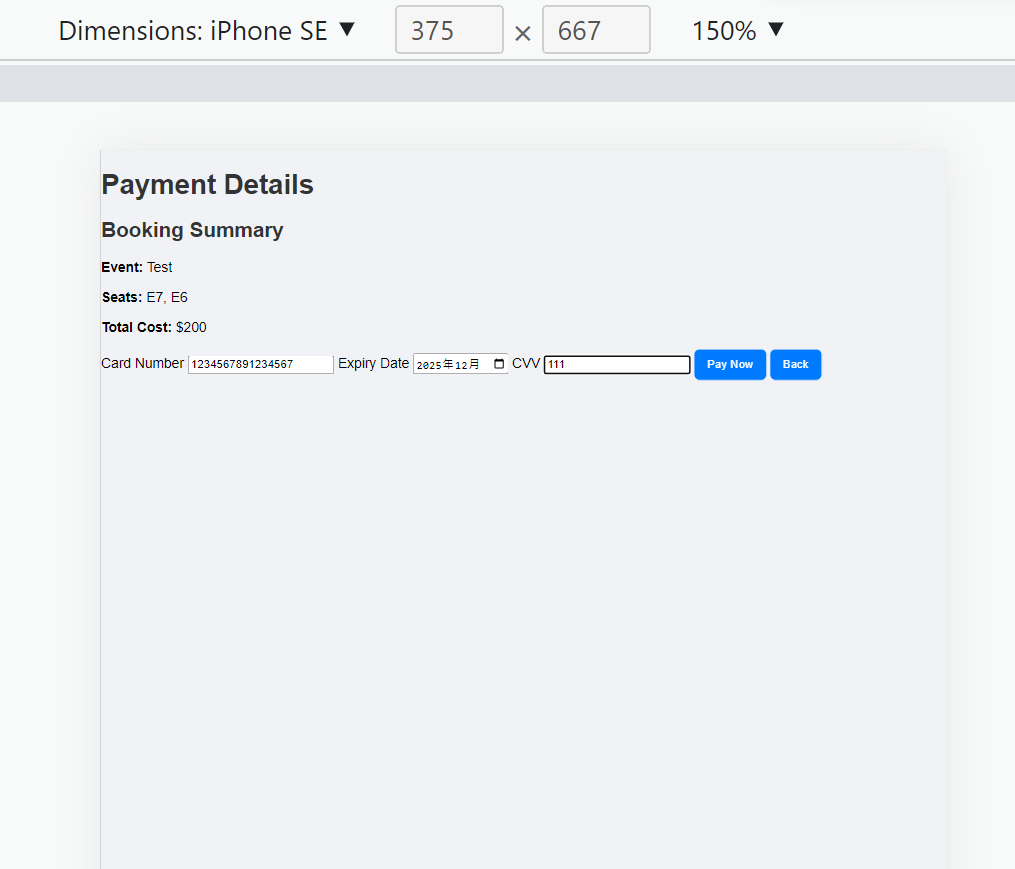
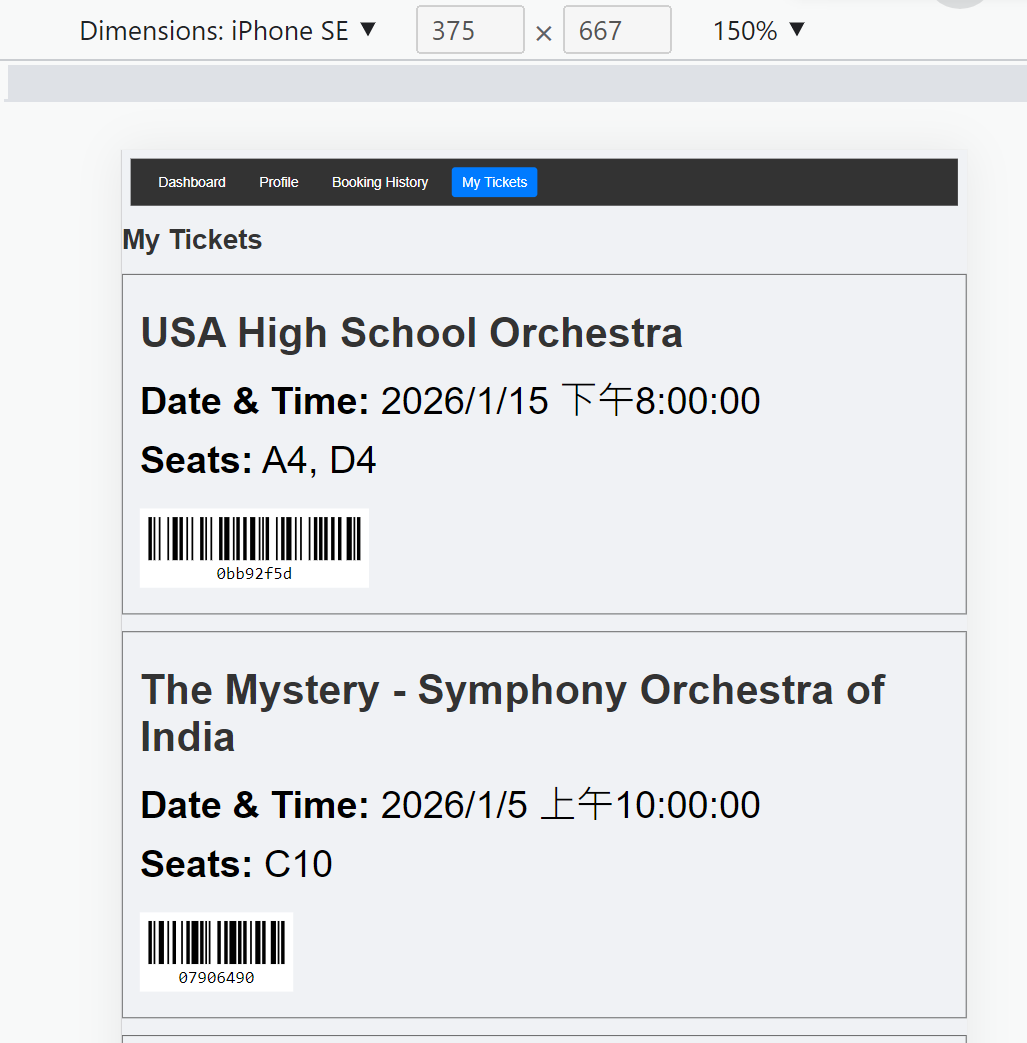
User management page (admin only):

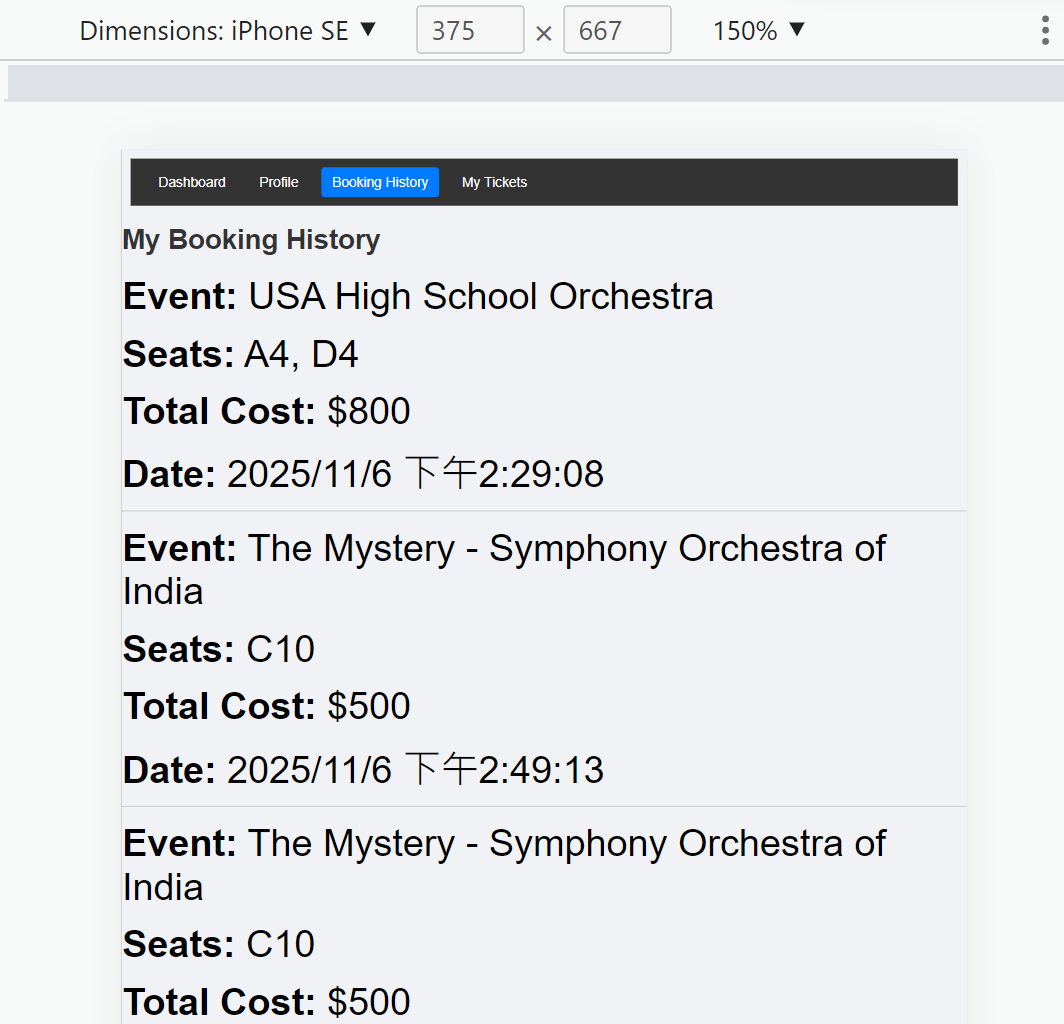


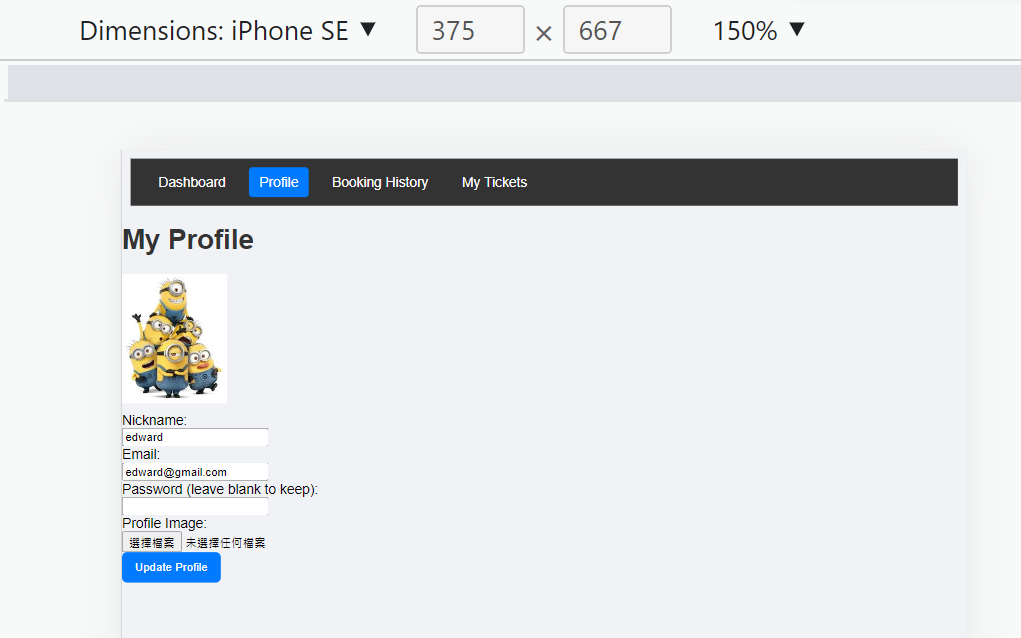
Multi-platform (dimension – IphoneSE):

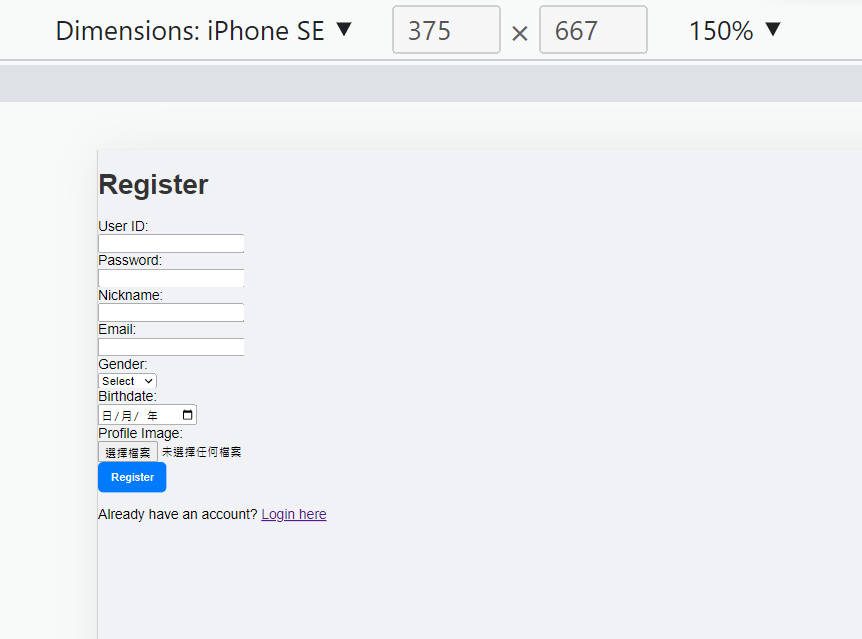
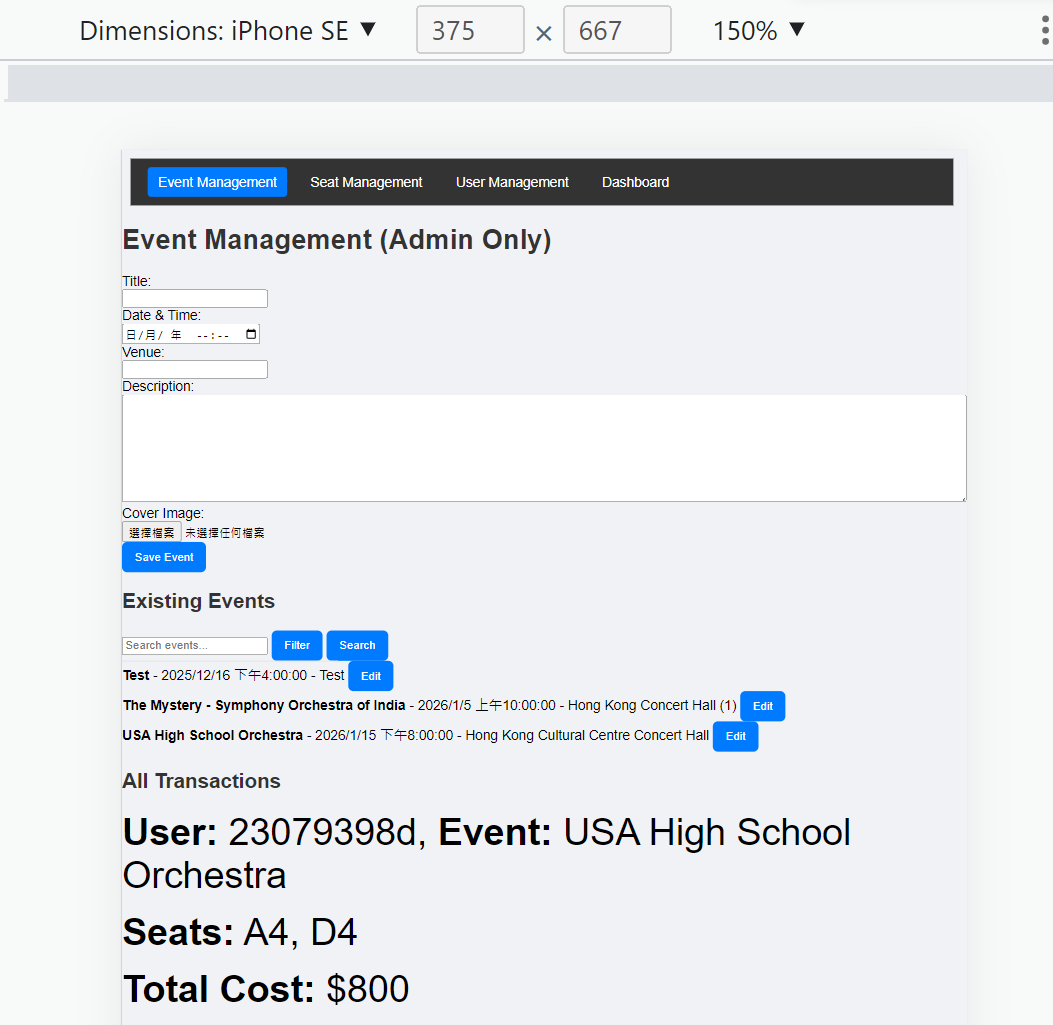
 

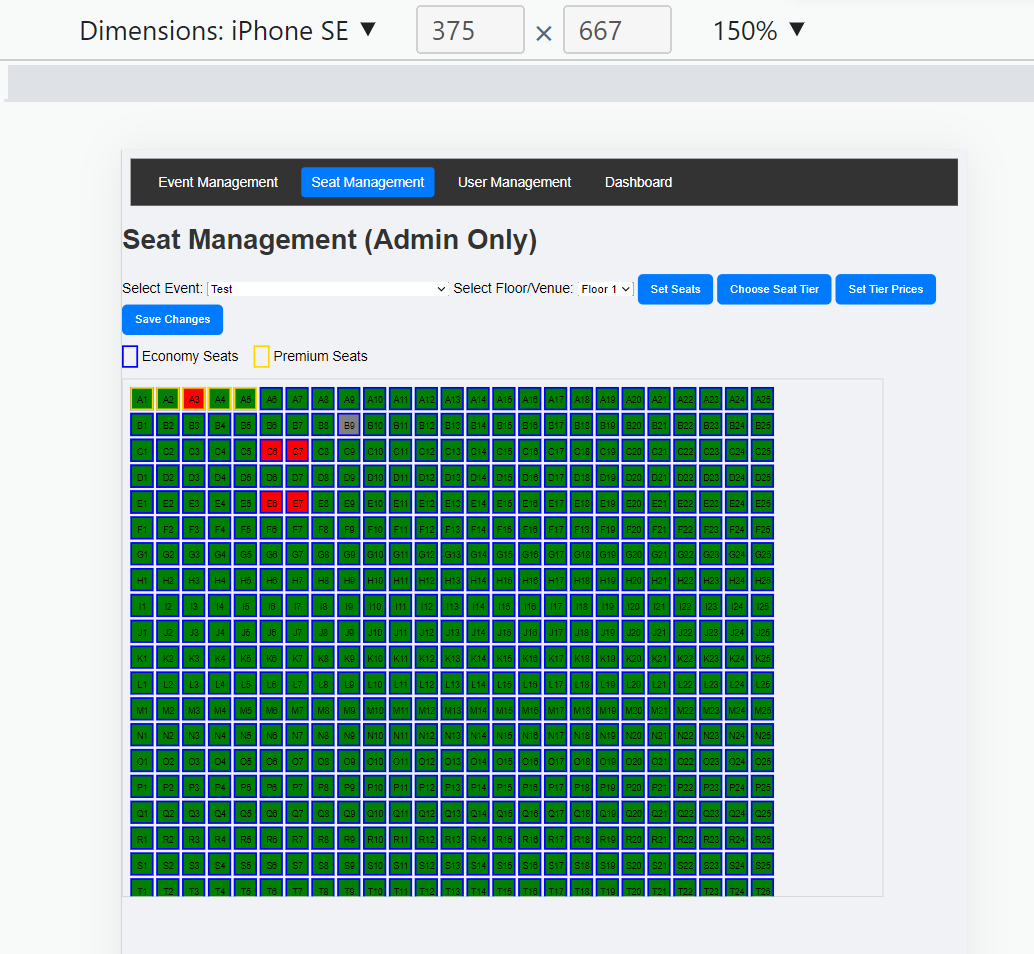
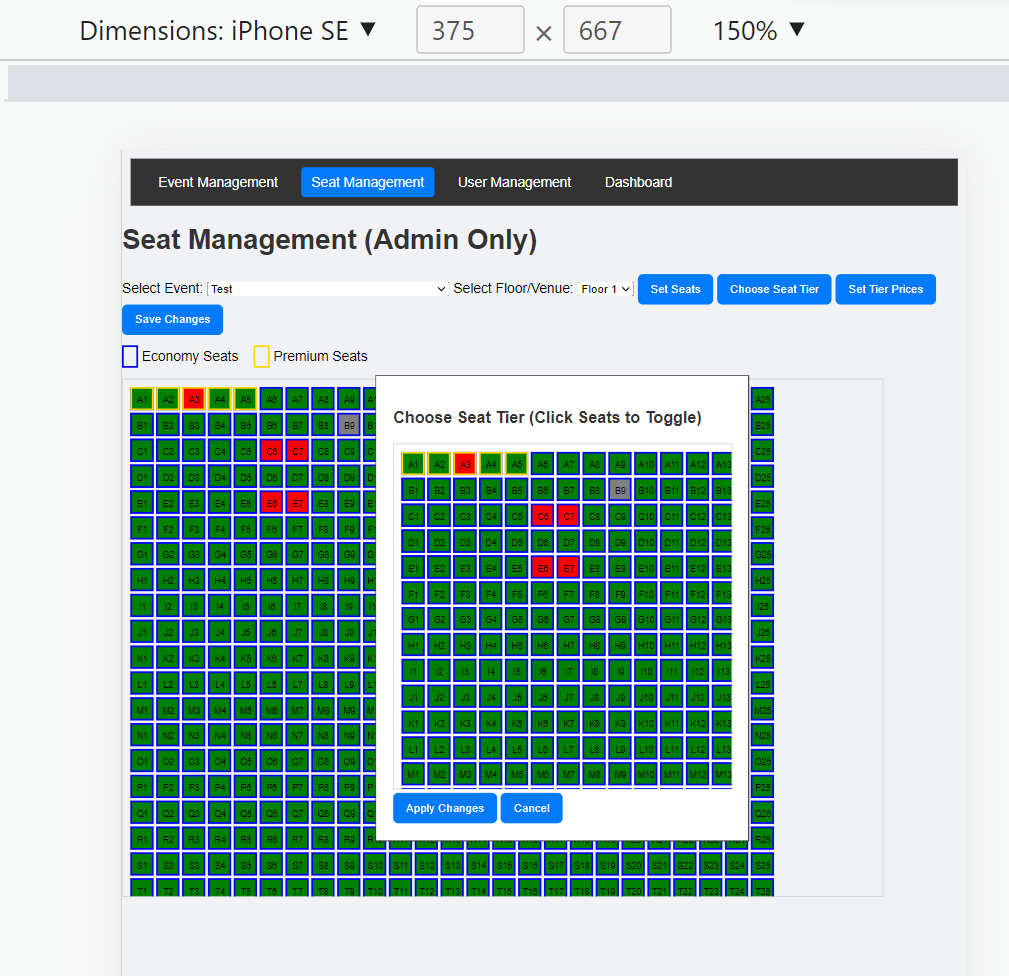
 

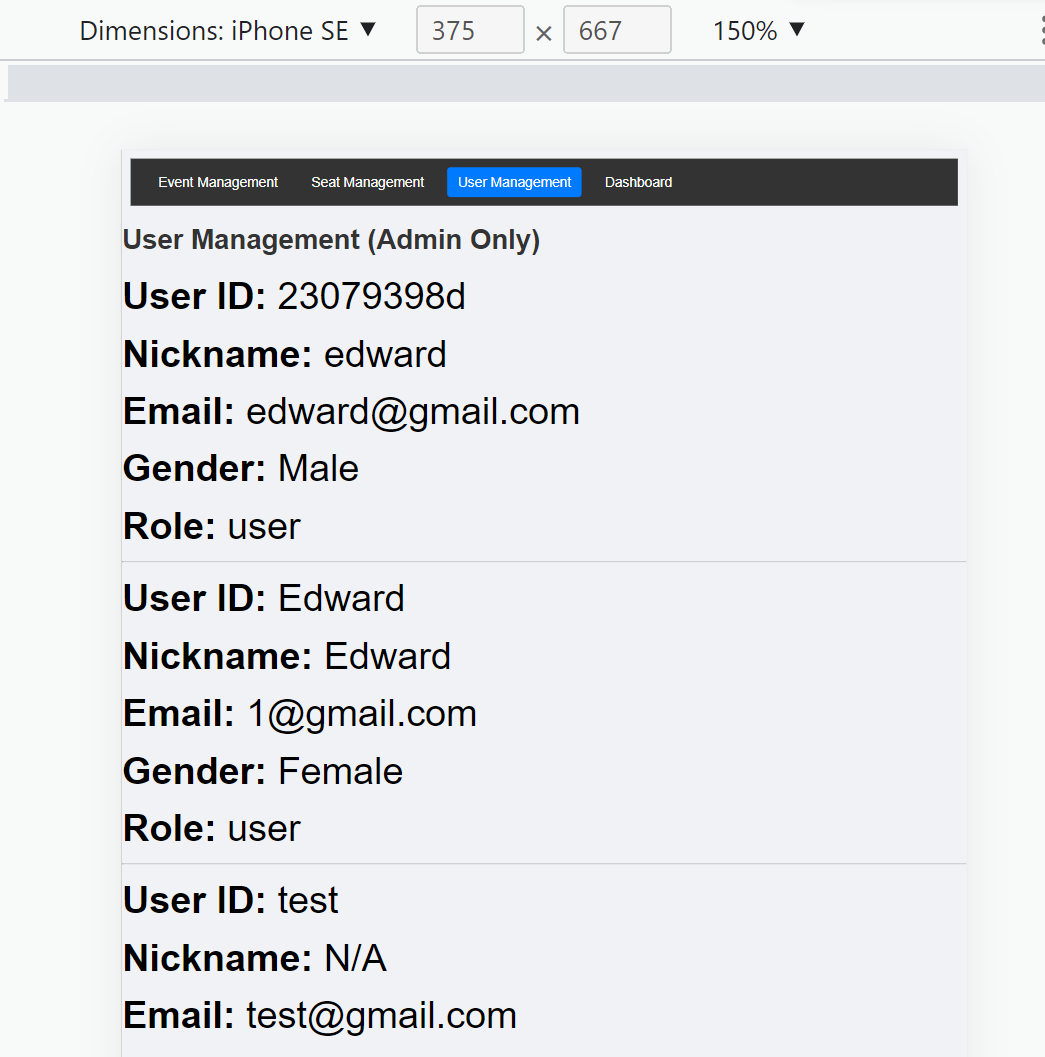
 







1. **Project Timeline**

|  |  |  |
| --- | --- | --- |
| Week | Key activities | Key deliverables |
| 6 | Web structure & User interface design | Project plan |
| 7 | Frontend pages, user authentication | Login/Register UI, event pages |
| 8 | Backend routing, MongoDB integration | Event and user APIs, DB models |
| 9 | Add interactive seat selection, booking and payment features | Seat map UI, booking & payment logic, |
| 10 | Implement user profiles, booking history, and ticket barcodes | Profile/history UI, barcode generation |
| 11 | Add admin event & seat management features | Admin UI |
| 12 | Testing, bug fixing, deployment, documentation, report writing | Deployment, README, final report |

1. **Work Distribution**

As this is an one-person group, all of the works are done by Yuen Pak Hei (23079398d)

1. **Conclusion**

The project successfully created a full-stack, user- and admin-friendly web application for orchestral event seat booking, combining modern UI, secure backend, and scalable NoSQL data handling.