

**Mukesh Patel School of Technology Management and Engineering**

**Topic: Smart Medication Reminder System**

**Course:** Web Programming

by

|  |  |  |
| --- | --- | --- |
| **Name** | Sowparnika Selvanayagam | Vishakha Khetan |
| **Roll No** | E225 | E220 |
| **SAP ID** | 70552300025 | 70552300020 |

Bachelor of Technology Computer Science

Second Year, Semester IV

**Faculty: Dr. Suraj Patil**

**AY 2024-25**

**INDEX**

| Sr. No. | Title | Page No. |
| --- | --- | --- |
| 1 | Introduction | 3 |
| 2 | Objective | 3 |
| 3 | Tools and Technology used | 4 |
| 4 | Function Modules | 4 |
| 5 | Database Design | 5 |
| 6 | Backend Overview | 6 |
| 7 | Code Files | 7 |
| 8 | Webpage Screenshots | 11 |
| 9 | Challenges Faced | 13 |
| 10 | Conclusion | 13 |
| 11 | Future Scope | 14 |
| 12 | References | 15 |

**INTRODUCTION**

In the current era of digitization, the effective management of academic events has become an integral part of college administration. However, many institutions still rely on outdated communication methods such as handwritten notices, bulletin boards, and sporadic announcements, which often result in missed information, lack of transparency, and confusion among students and staff.

To address this, we developed the **College Event Management Website**, a centralized platform designed to modernize how events are organized, announced, and accessed by students and committee members. This system not only provides a clear and accessible overview of all upcoming and past events but also empowers the event management team with the ability to control and update content dynamically.

The website includes dedicated modules tailored to different user types. Students are able to register, log in, and view event details through a clean and user-friendly interface. Meanwhile, committee members are granted access to administrative tools that enable them to log in securely, add new events, edit existing ones, and monitor event listings. The backend is built using **Node.js and Express.js**, which ensures smooth server operations and routing, while the data is stored and managed using **Oracle SQL**, offering strong relational database capabilities.

This project aims not only to improve information dissemination within a college setting but also to serve as a foundation for building more complex systems with features like event registration, QR check-ins, and real-time notifications. It also acts as a learning platform for understanding full-stack development, integrating both frontend and backend technologies in a practical real-world scenario.

**OBJECTIVE**

The primary objective of this project is to develop a fully functional and interactive web-based system that:

* Facilitates students in viewing upcoming college events.
* Allows committee members to add, edit, and view events.
* Stores data persistently and securely using Oracle SQL.
* Uses a simple and intuitive interface created with HTML, CSS, and JavaScript.
* Incorporates Node.js and Express.js for backend functionalities, including form handling, data fetching, and route management.

**TOOLS AND TECHNOLOGIES USED**

| Tool/Technology | Purpose |
| --- | --- |
| HTML | Structuring web pages |
| CSS | Styling and layout |
| JavaScript | Client-side scripting and interactivity |
| Node.js | Server-side scripting |
| Express.js | Simplifying Node.js routing and middleware |
| Oracle SQL | Backend database |
| VS Code | Development environment |
| Postman | API testing (optional) |
| Browser Dev Tools | Debugging frontend code |

**FUNCTION MODULES**

* *Welcome Page*
* Acts as the entry point.
* Asks the user to choose their role: Student or Committee.
* *Student Module*
* *Signup Page:*
* Allows students to register.
* Fields include: name, email, password.
* *Login Page:*
* Authenticates students.
* *View Events Page:*
* Displays a list of events.
* Each event has a clickable section to view more details in a modal popup.
* *Committee Module*
* *Admin Login Page:*
* Allows committee members to log in using a predefined username and password.
* *Dashboard Page:*
* Shows a list of all events.
* Each event has an Edit button beside it.
* A button at the bottom allows adding a new event.
* *Event Modal Popup*
* Displays enlarged details of a selected event including image, date, time, and venue.

**DATABASE DESIGN**

Tables Used:

1. students

| Field | Type |
| --- | --- |
| id | NUMBER (Primary Key) |
| name | VARCHAR |
| email | VARCHAR (Unique) |
| password | VARCHAR |

1. Events

| Field | Type |
| --- | --- |
| id | NUMBER (Primary Key) |
| title | VARCHAR |
| image | VARCHAR (URL or file path) |
| date | DATE |
| time | VARCHAR |
| venue | VARCHAR |
| description | VARCHAR |

**BACKEND OVERVIEW**

1. *Server.js (Main Backend File)*

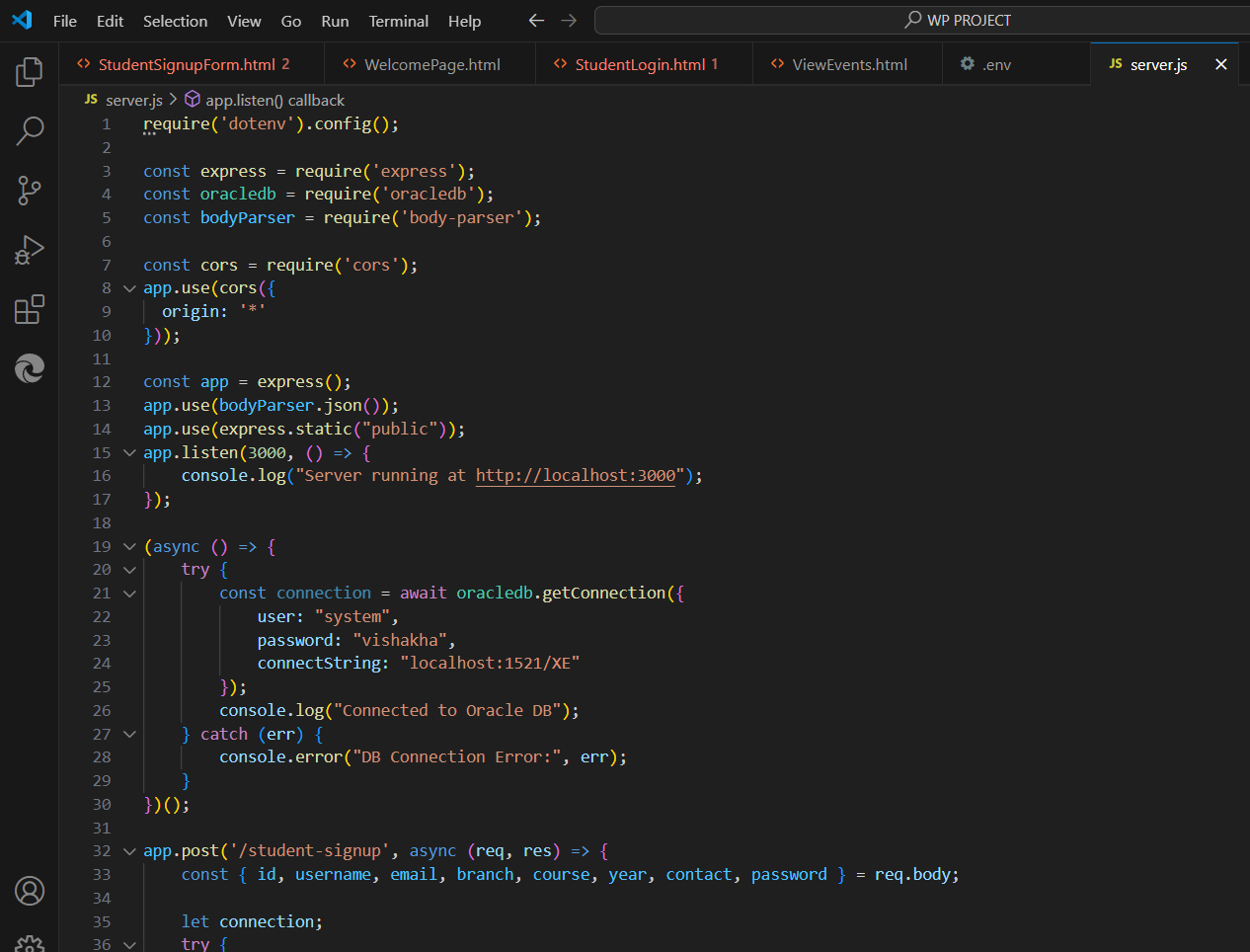
* Uses Express.js to handle routing.
* Middleware: express.json() and express.urlencoded() to parse request bodies.
* Database Connection: Uses oracledb module to interact with Oracle DB.

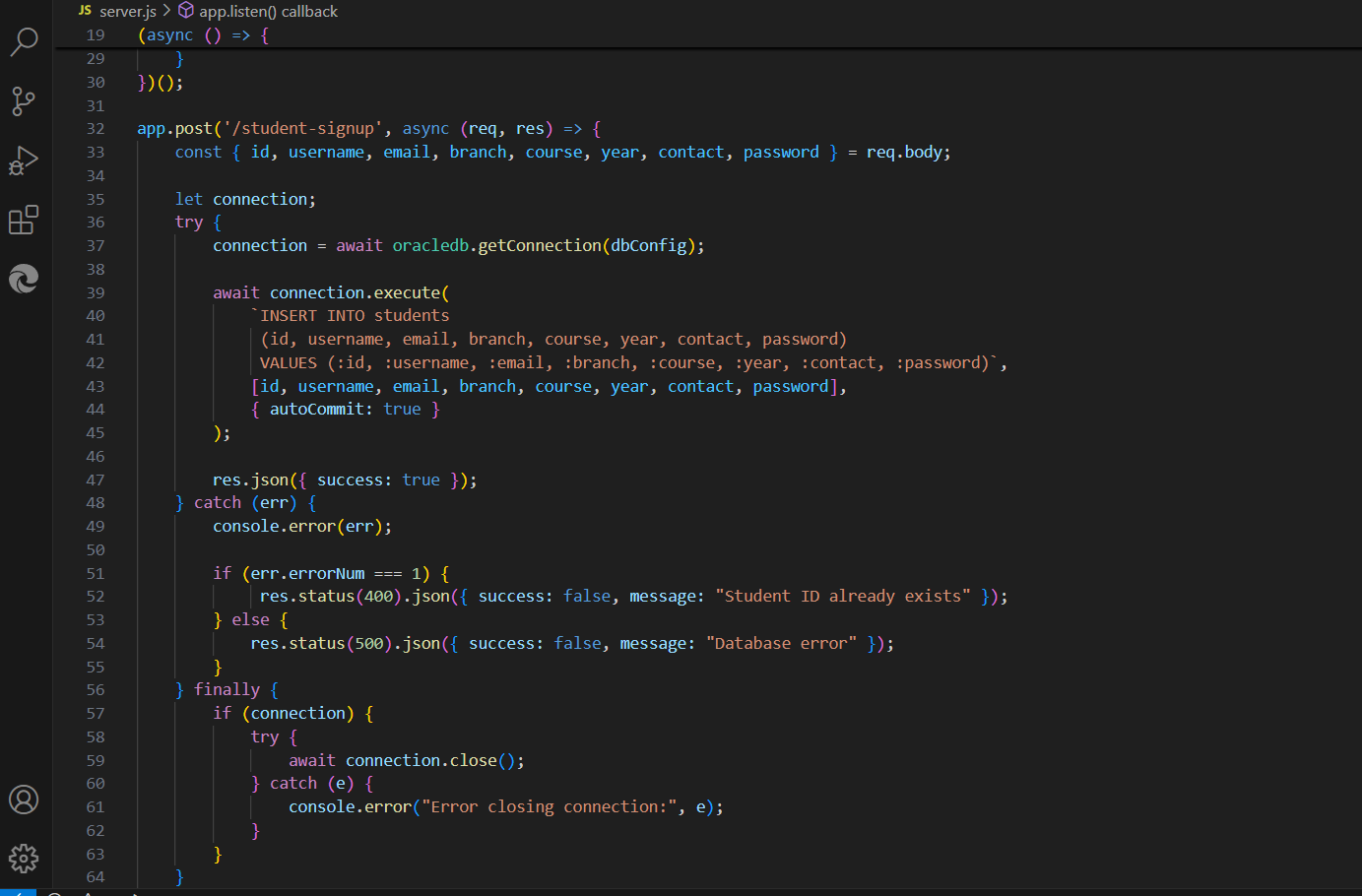
1. *Important Routes*

* POST /student-signup – Adds new student to DB.
* POST /student-login – Validates login credentials.
* POST /admin-login – Verifies admin login.
* POST /add-event – Adds new event.
* POST /edit-event – Updates an existing event.
* GET /events – Returns all events to be displayed.

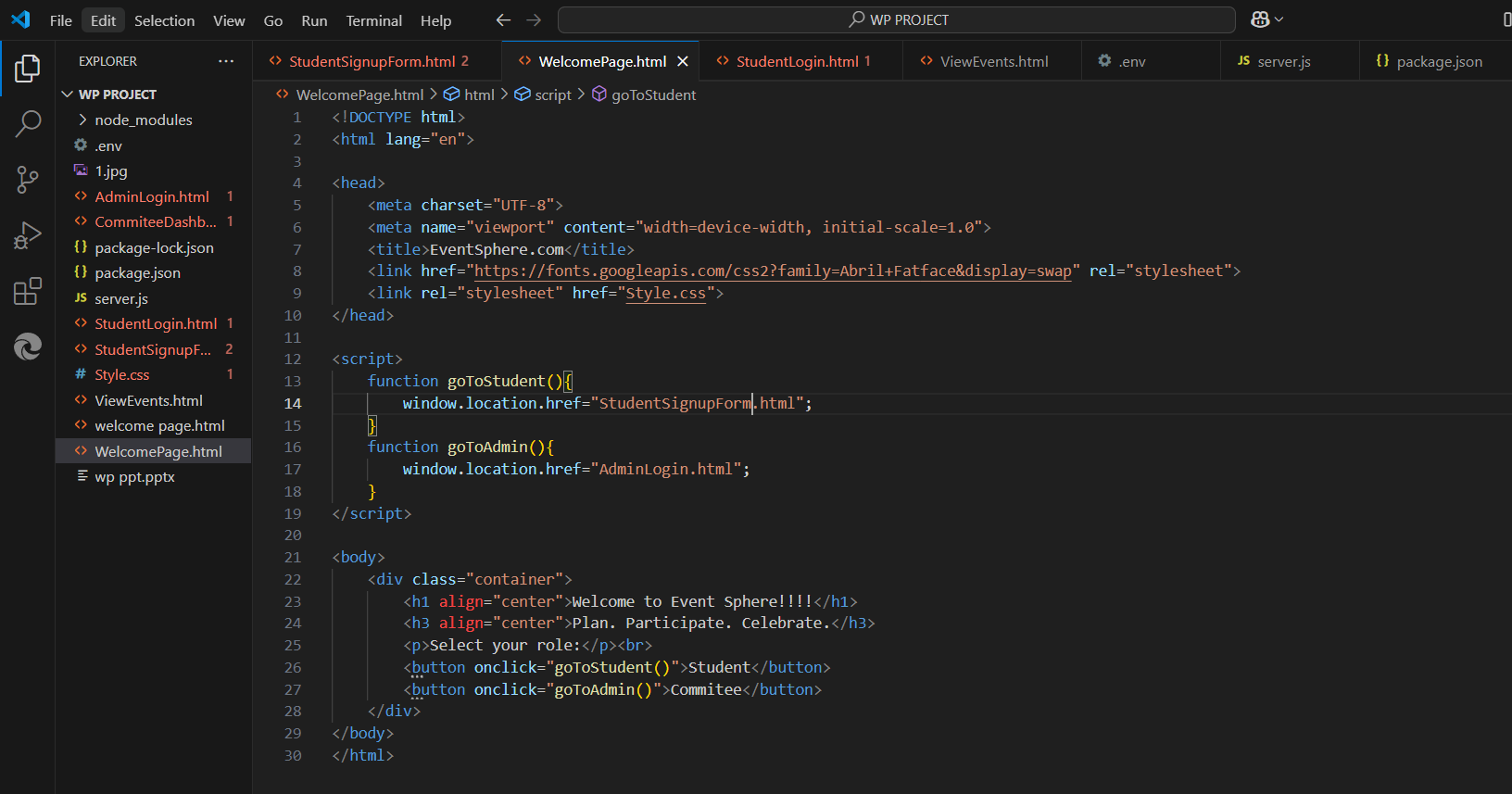
**CODE FILES**

1. server.js

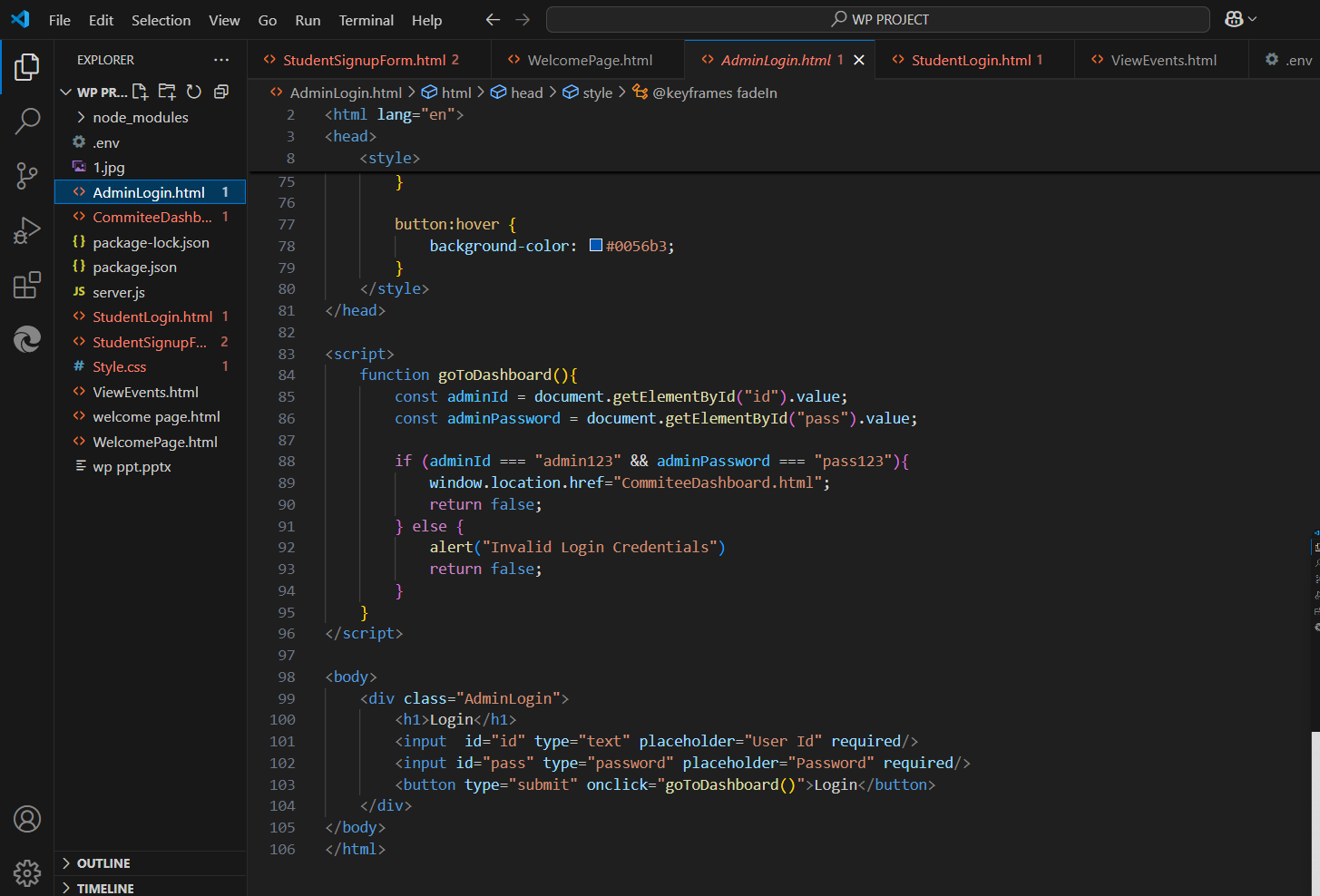




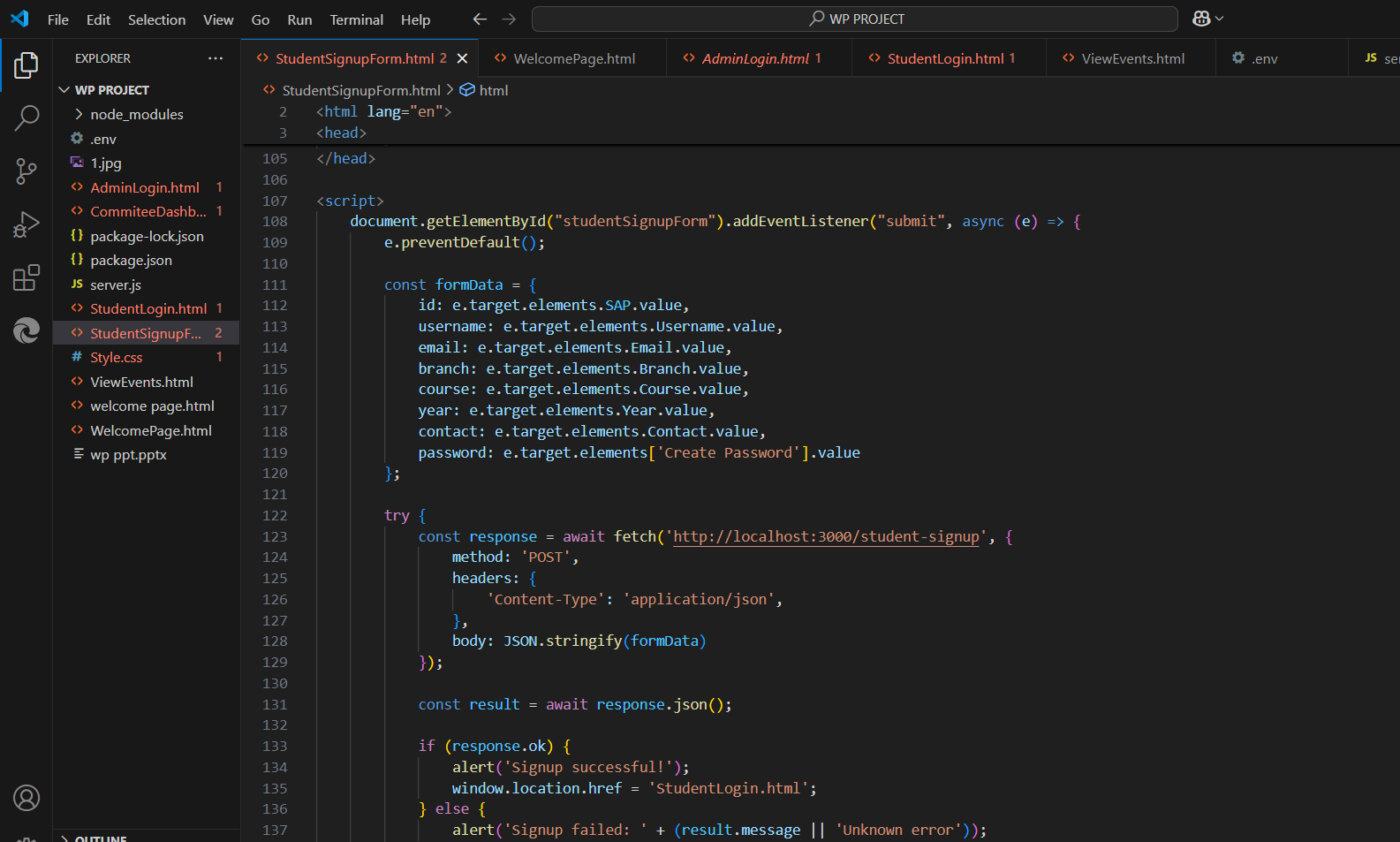
1. welcomePage.html

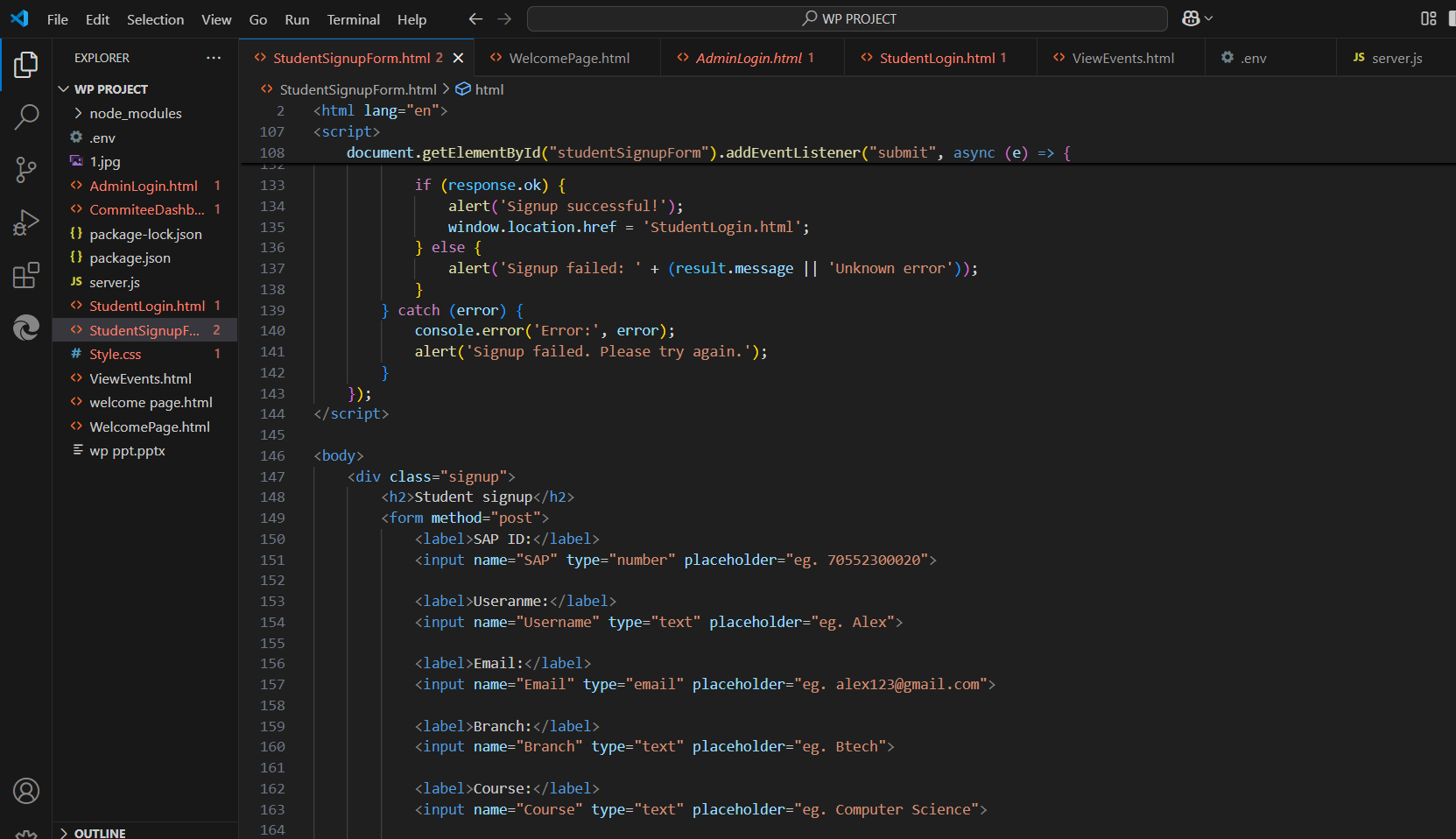


1. adminLogin.html

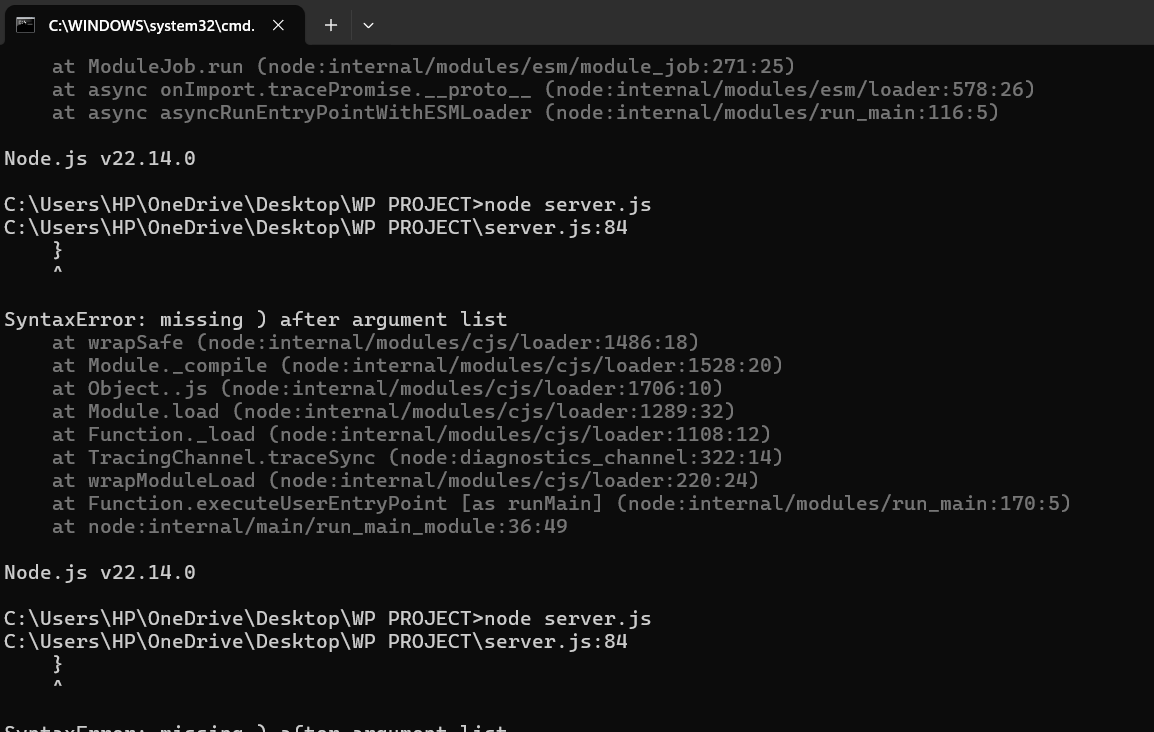
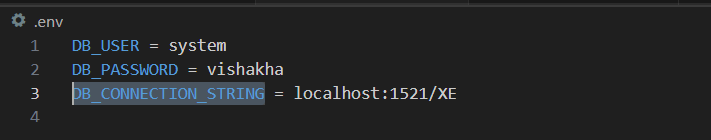


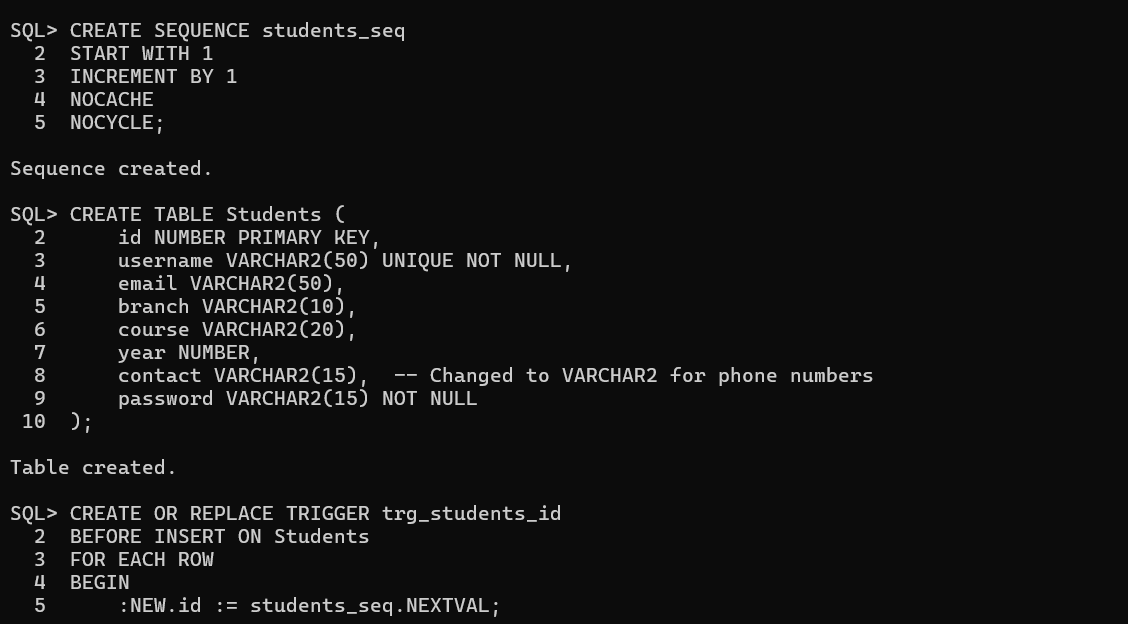
1. StudenSignupPage.html

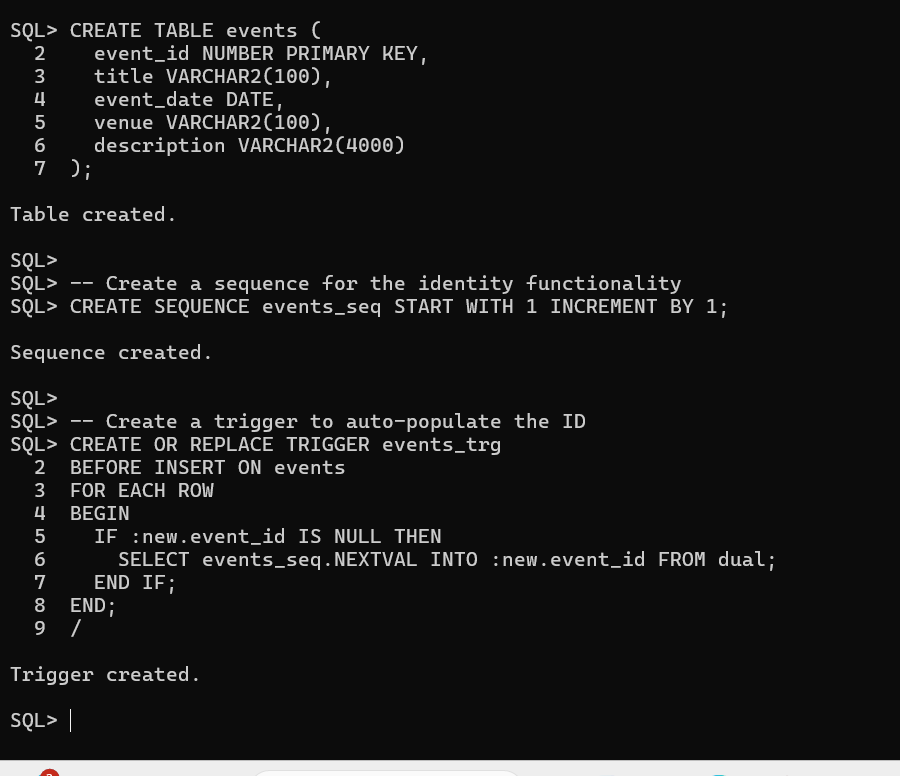




1. Additional Files

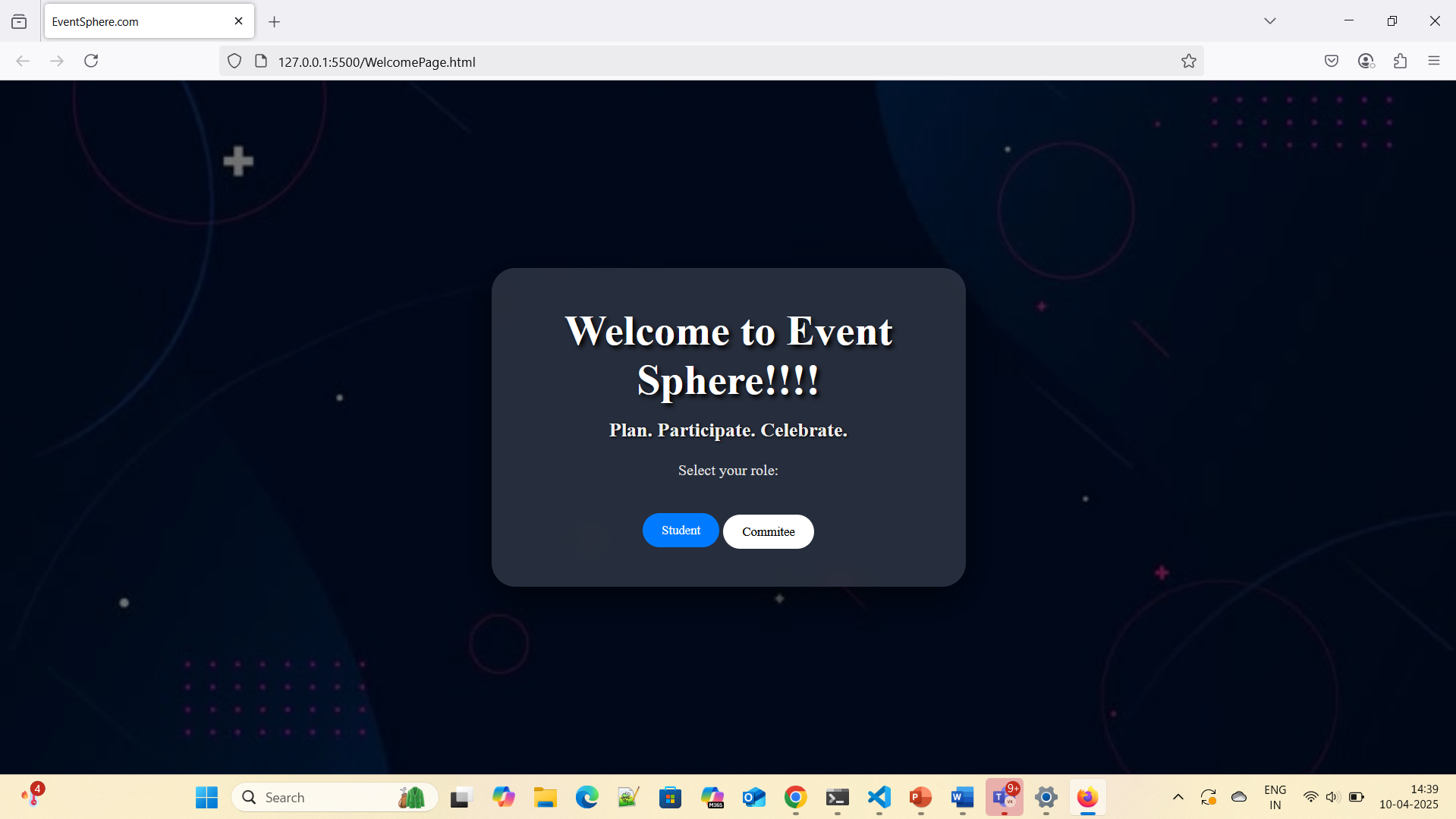




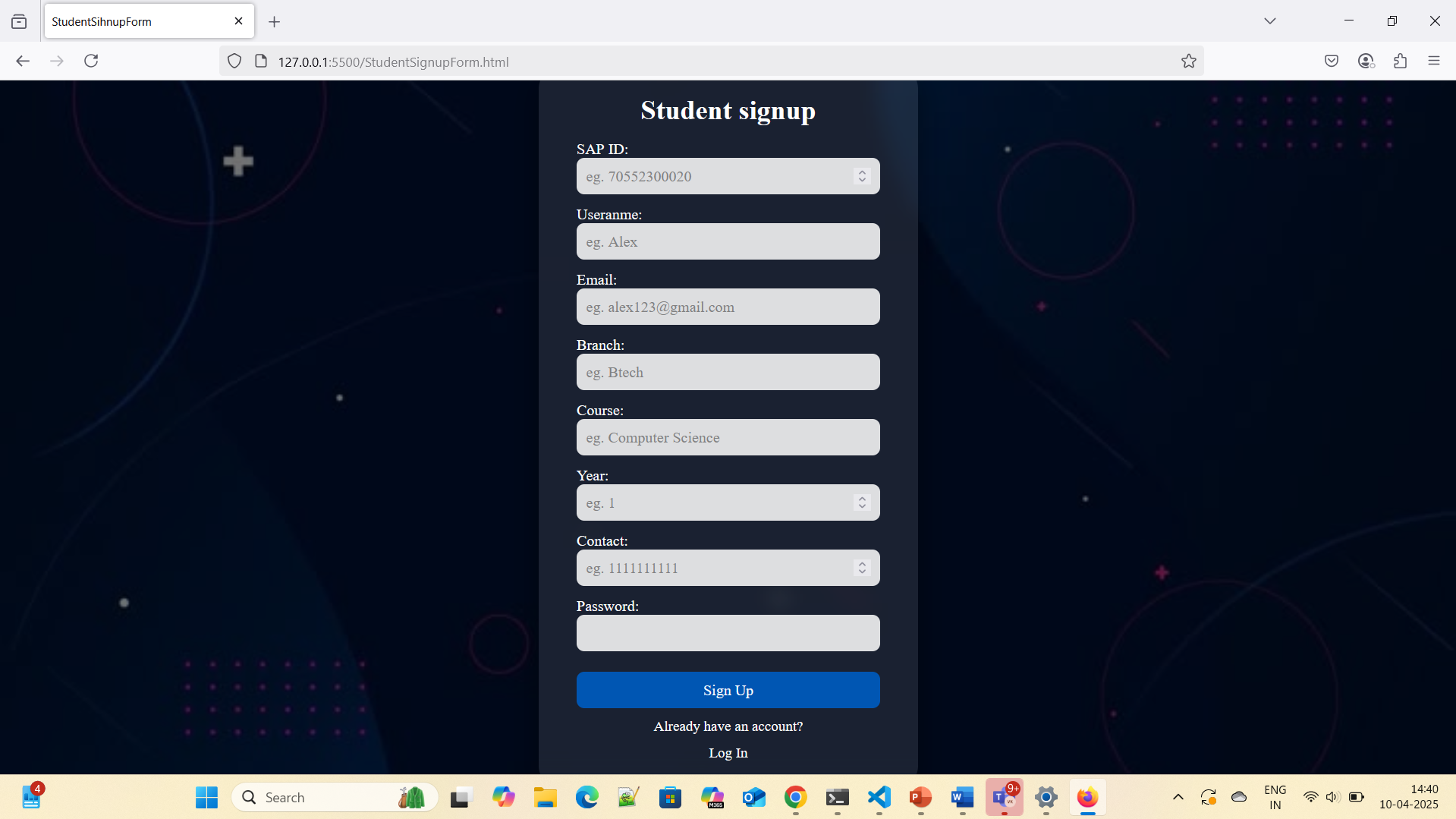


**WEBPAGES**

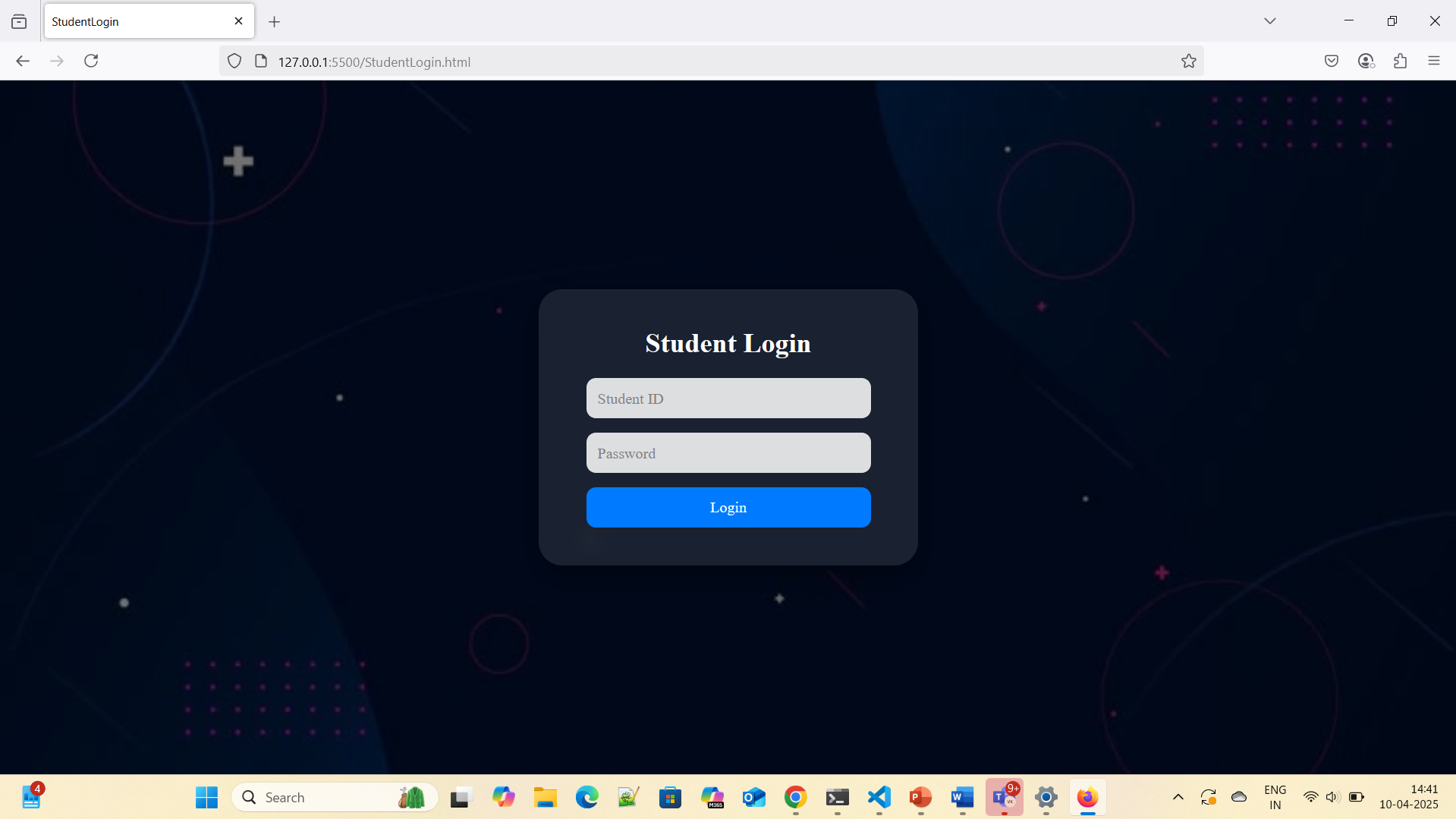
* Welcome Page



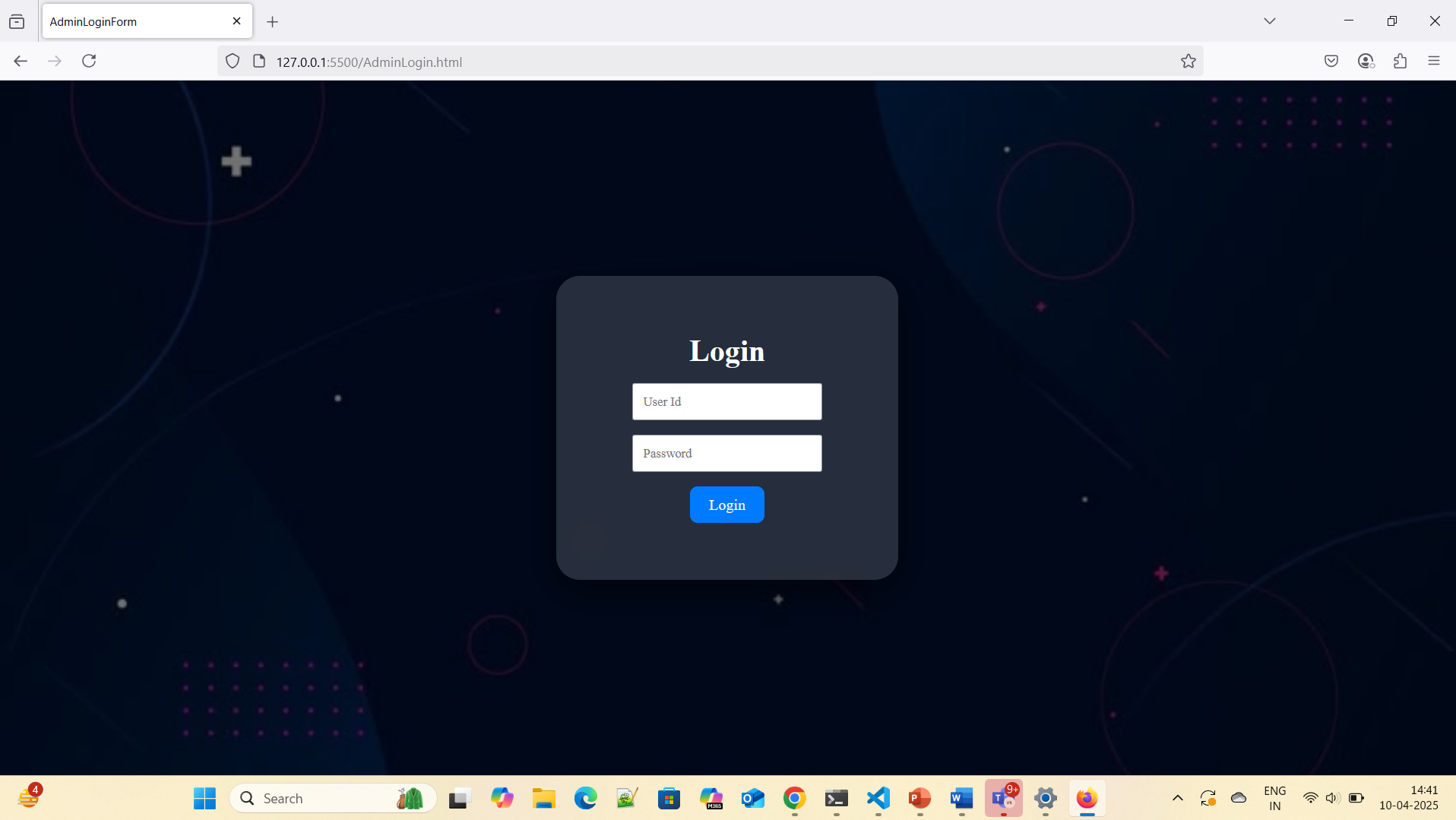
* Student Signup Page



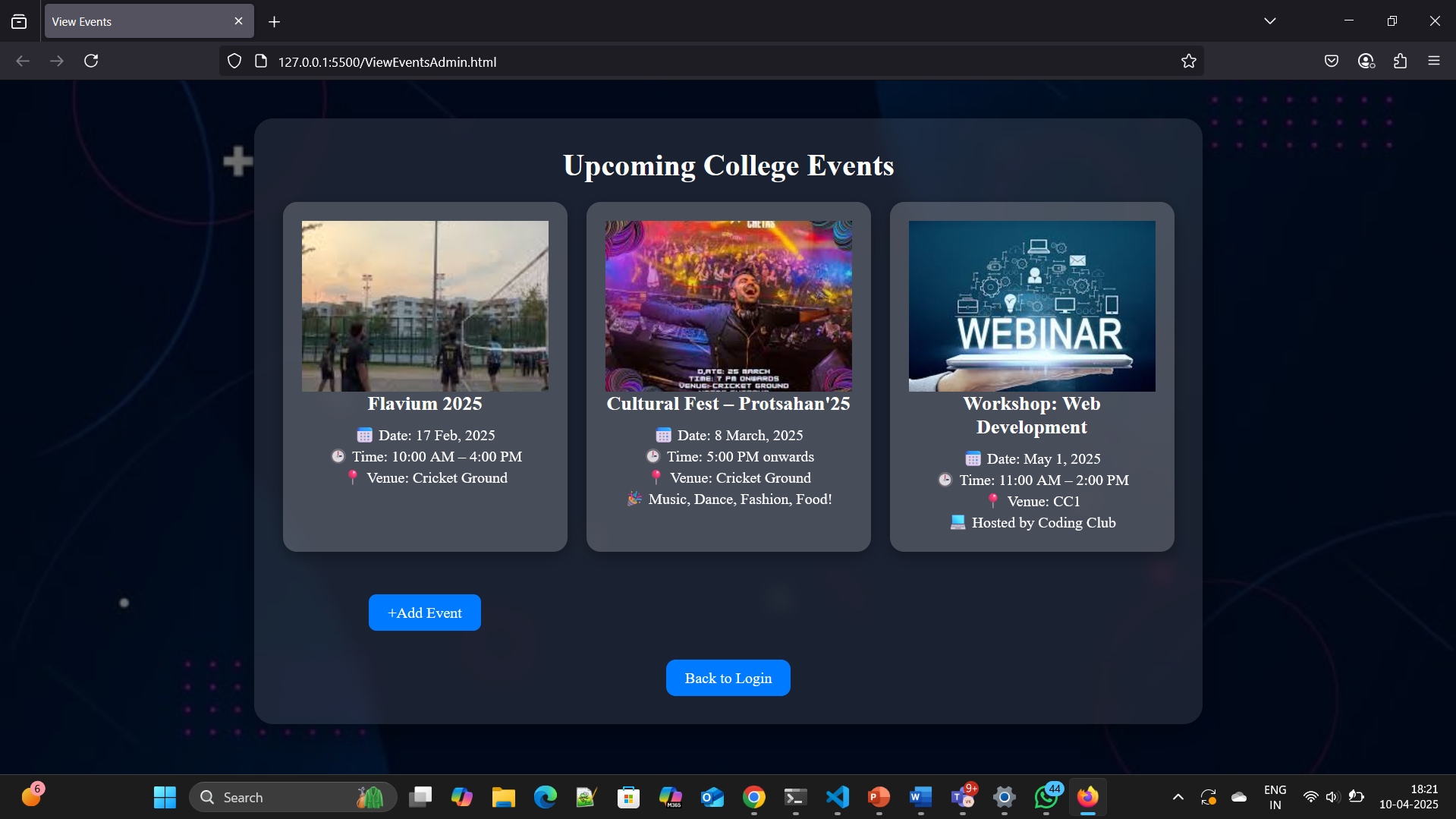
* Student Login Page



* Admin Login Page



View events page



**CHALLENGES FACED**

* Setting up Oracle DB connectivity with Node.js using oracledb.
* Resolving CORS and fetch-related issues when making requests from frontend to backend.
* Designing a responsive interface in a short amount of time.
* Handling real-time form validations and user feedback.
* Testing functionalities like editing an event and modal popups.

**CONCLUSION**

The College Event Management Website is a significant step toward creating an efficient, modern, and interactive environment for students and college committees. It simplifies the way events are communicated and managed, eliminating the dependency on traditional, error-prone, and time-consuming methods.

By successfully integrating frontend technologies such as HTML, CSS, and JavaScript with backend logic powered by Node.js, Express.js, and a robust Oracle SQL database, this project ensures that both students and committee members experience a seamless interaction. Students benefit from instant access to all event information, while administrators enjoy full control over the creation and management of events in a secure and organized manner.

In addition to technical achievement, this project represents a broader vision of campus digitization. It offers flexibility, scalability, and modularity for future enhancements, and lays a foundation for integrating more advanced technologies in the future.

The successful implementation of this website under a short timeframe also reflects the practical application of academic knowledge, teamwork, and problem-solving skills acquired during the course of development. It opens doors to potential real-world deployment with minor modifications and security upgrades.

**FUTURE SCOPE**

While the current system serves its core purpose effectively, there is ample scope for future development to enhance usability, scalability, and functionality. Some of the planned and recommended future additions include:

1. **Event Registration by Students**

Introduce functionality that allows students to register for events directly from the platform. The system could generate a unique ID or QR code for each registration to track attendance.

1. **Notification System**

Integration of a real-time notification system (via email or in-app alerts) to remind users of upcoming events, cancellations, or changes in venue/time.

1. **Mobile App Version**  
   A responsive mobile application for Android/iOS that syncs with the same database and server to provide real-time access on-the-go.
2. **Event Filters and Search Functionality**  
   Enable filtering and searching of events based on categories such as department, type (seminar, cultural, sports), date, and more.
3. **User Feedback and Ratings**  
   After attending events, students could provide feedback, rate the event, and help improve future planning.
4. **Multi-Admin Support**  
   Support different roles for admins (e.g., Department-level admins) with role-based access controls and permissions.
5. **Calendar View Integration**  
   Display upcoming events in a calendar format for better planning and visualization.
6. **Enhanced Security**  
   Encrypt sensitive information like passwords using hashing algorithms (bcrypt), implement CAPTCHA on forms, and use HTTPS for secure data transmission.
7. **Analytics Dashboard**

Add charts and reports for admins to view participation trends, popular event categories, and feedback summaries.

1. **Live Event Streaming Links**  
   Integration with platforms like YouTube or Google Meet to embed live event links for online attendees.

**REFERENCES**

* Node.js and Express documentation
* Oracle SQL Developer Docs
* MDN Web Docs (HTML/CSS/JS)
* Stack Overflow for debugging help
* YouTube tutorials for UI design and backend setup