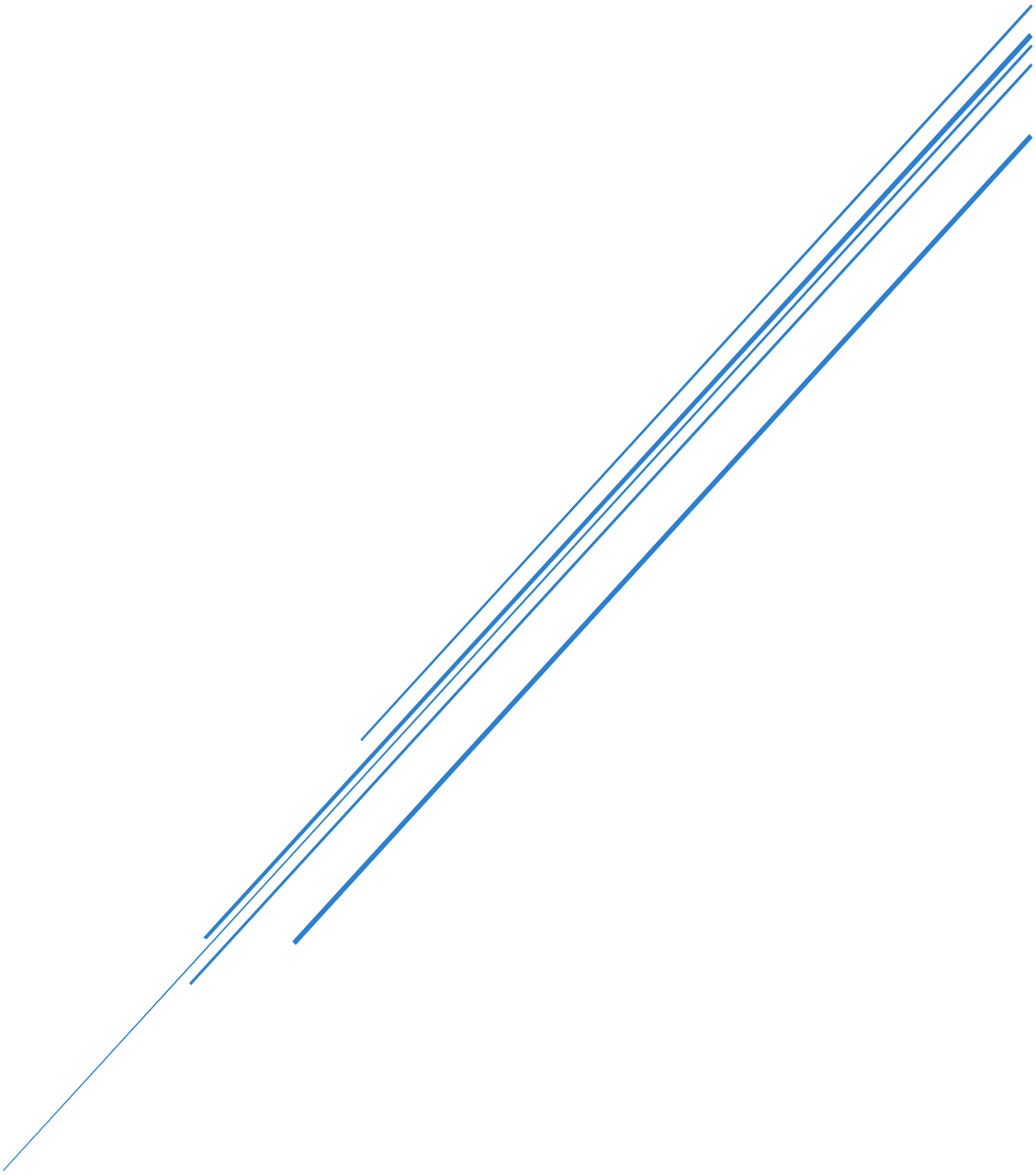
EVENT MANAGEMENT SYSTEM

Phase 5



CSAI

Software Engineering

Team Contributions:-

- Zeyad: Was mainly responsible for the core backend and system foundation throughout the project.
  - He implemented the authentication functionality, including login and signup, along with the required controllers and repositories.
  - He also worked on the base layout and user interface styling, including the base template and CSS files.
  - In addition, Ziad handled feedback functionality, database connection logic, security utilities, and application configuration files.
  - He implemented the main application setup, database initialization, and utility modules.
  - He was also responsible for writing multiple unit tests, preparing the Dockerfile, initializing the database, managing dependencies, and creating the final README file.
- Sara: Worked mainly on event registration functionality and testing.
  - She implemented the event registration logic, including its controller and repository.
  - She also worked on the event details page and ensured correct interaction between the frontend and backend.
  - In Phase 5, she focused on writing unit tests related to registration and other system components.
  - Additionally, she prepared the testing report documenting the testing process and results.
- Retaj: Was responsible for event management features and documentation.
  - She implemented event-related functionalities, including event creation, editing, listing, and organizer-specific views.
  - She also worked on the main dashboard and multiple event-related pages.
  - In addition to development tasks, she handled all documentation work, including User Documentation, Technical Documentation, and the Phase 5 Report.
  - She also prepared the final presentation slides and organized the demo flow for the project.

1. Phase 5 Requirements:-

Phase 5 focused on:

- Testing
- Documentation
- Docker deployment
- Final presentation and demo

2. Testing Summary:-

- Unit tests were implemented for repository and authentication logic.
- Integration testing was used to verify routing functionality.
- All tests passed successfully.

3. Docker Summary:-

- Dockerfile was created to containerize the application.
- The system runs inside a Docker container using Flask.
- Docker ensures environment consistency.

4. Demo Description:-

The demo demonstrates:

- User registration and login
- Event browsing and registration
- Feedback submission
- Running the application using Docker

5. Challenges & Solutions:-

Challenges:

- Database handling in Docker
- Git merge conflicts
- Flask application context during testing

Solutions:

- Proper use of Flask app context
- Git feature branches
- Clear project structure

6. Conclusion:-

The Event Management System successfully meets all requirements of Phase 5.
The project demonstrates practical application of software engineering concepts taught in the course.

7. Screenshots and Visual Evidence:-

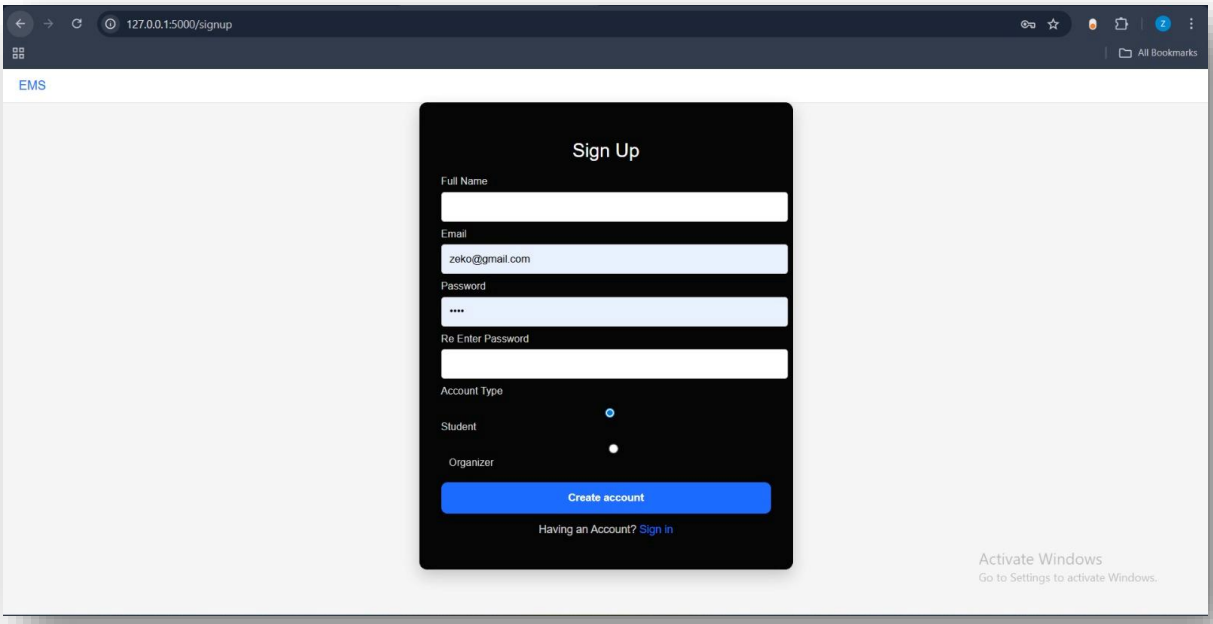
This section provides visual evidence demonstrating the successful implementation of the Event Management System and Phase 5 requirements.

The screenshots illustrate the main system features, user interactions, and Docker execution.

7.1 User Authentication Screenshots:

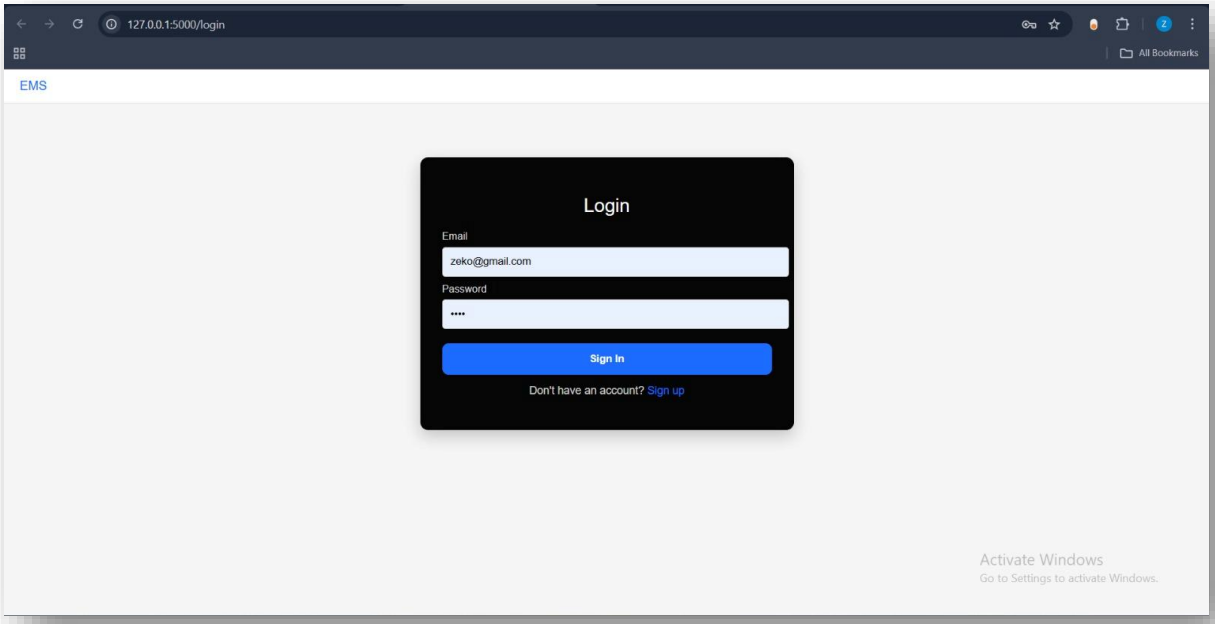
1. Signup Page

This screenshot shows the user registration page where new users can create an account by providing their basic information.



2. Login Page

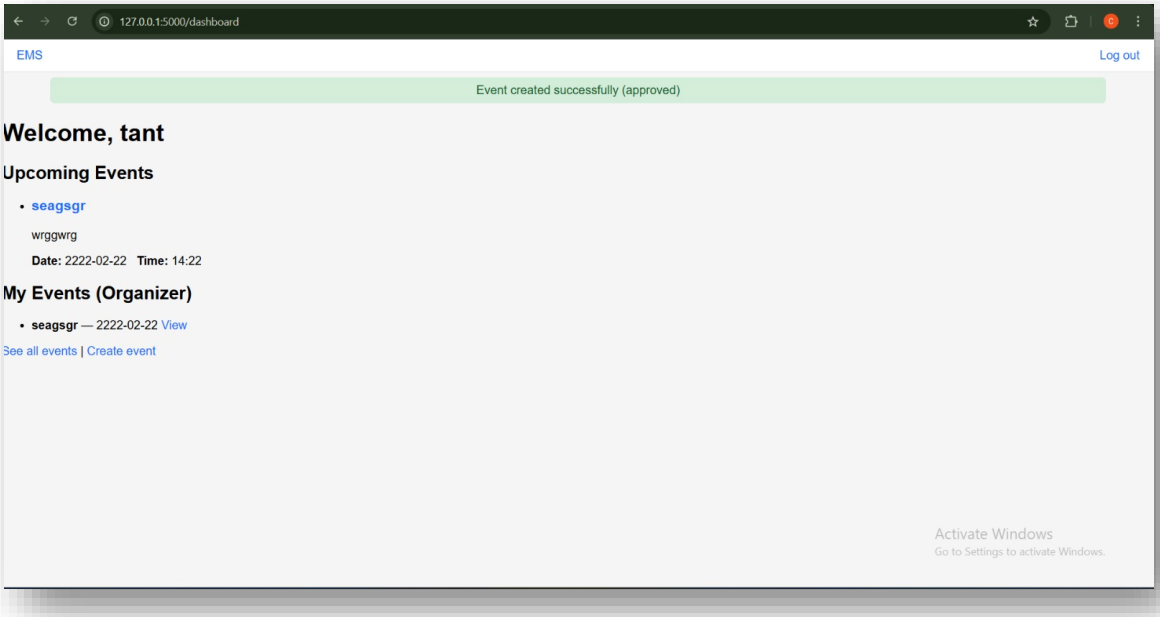
This screenshot shows the login page where registered users can enter their credentials to access the system.



7.2 Dashboard and Event Browsing:

- Event Dashboard

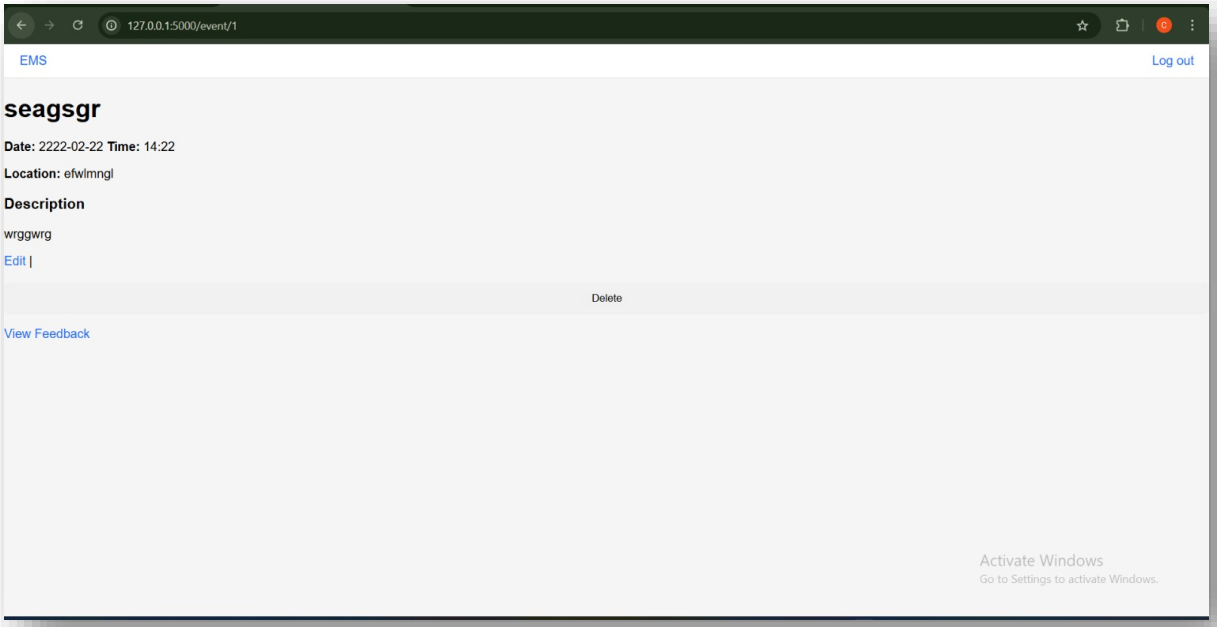
This screenshot shows the main dashboard displaying available events after the user logs in successfully.



7.3 Event Details and Registration:

- Event Details Page

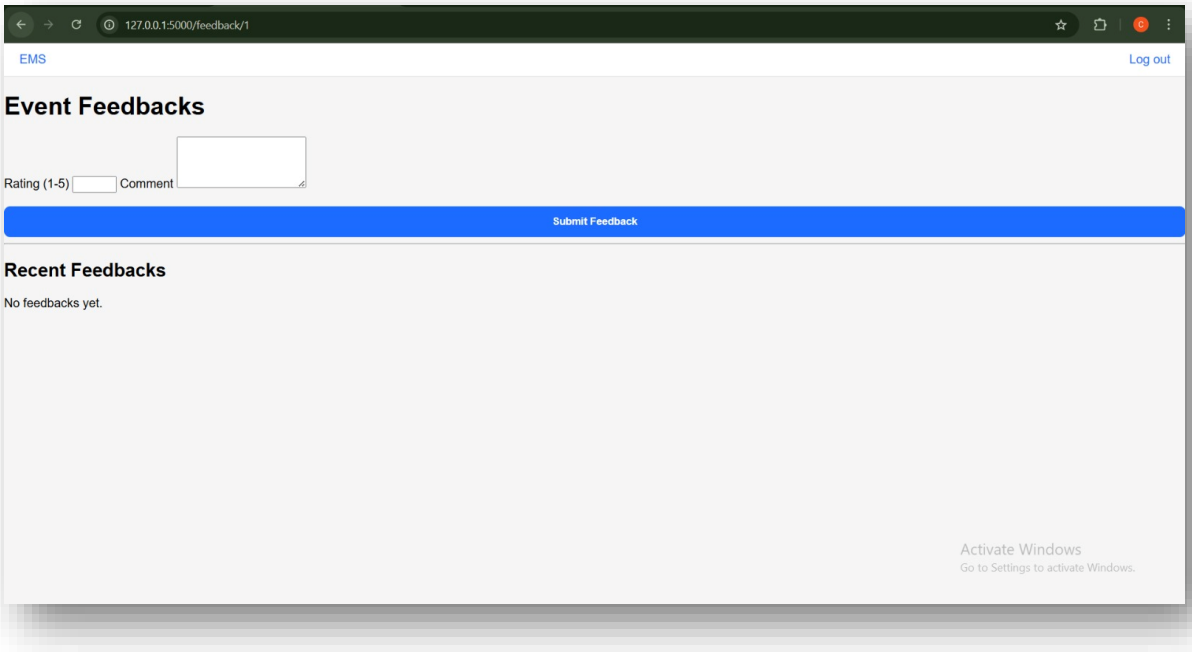
This screenshot shows the event details page, including event information and the option to register for the event.



7.4 Feedback Submission:

- Feedback Page

This screenshot demonstrates the feedback submission feature, where users can rate events and add comments.



7.5 Docker Execution:

- Application Running with Docker

This screenshot shows the application running successfully inside a Docker container, confirming Docker integration in Phase 5.