# **Event Management System Project Documentation**

## **Project Overview**

The Event Management System is a web-based application built using Django, HTML, CSS, and JavaScript. This system provides users with the ability to sign in as either an organizer or an attendee. It aims at allowing organizers create events, while attendees can RSVP to events. The application features a navigation bar with links to Home, About, Contact, Signup, and Login pages, ensuring navigation.

## **Distinctiveness and Complexity**

### **Distinctiveness:**

1. **Dual Sign-in Roles**: The system allows users to choose between signing in as an organizer or an attendee, providing two distinct user experiences.
2. **Role-specific Features**:
   * Organizers can create events, manage their details, and view RSVPs.
   * Attendees can browse events and RSVP directly to specific organizers.
3. **Custom User Experience**: The role-specific features make this application stand out compared to generic event management systems.

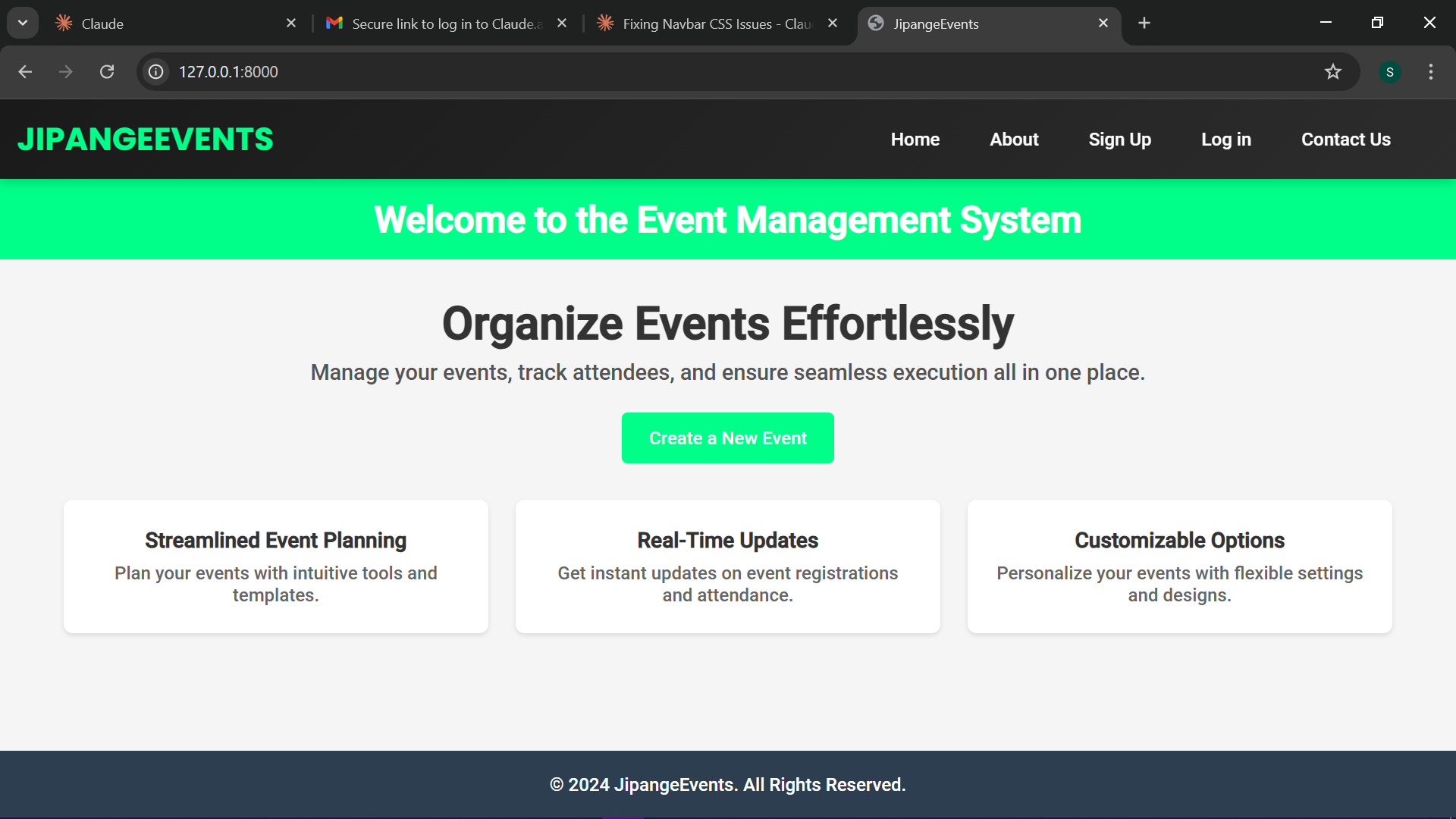
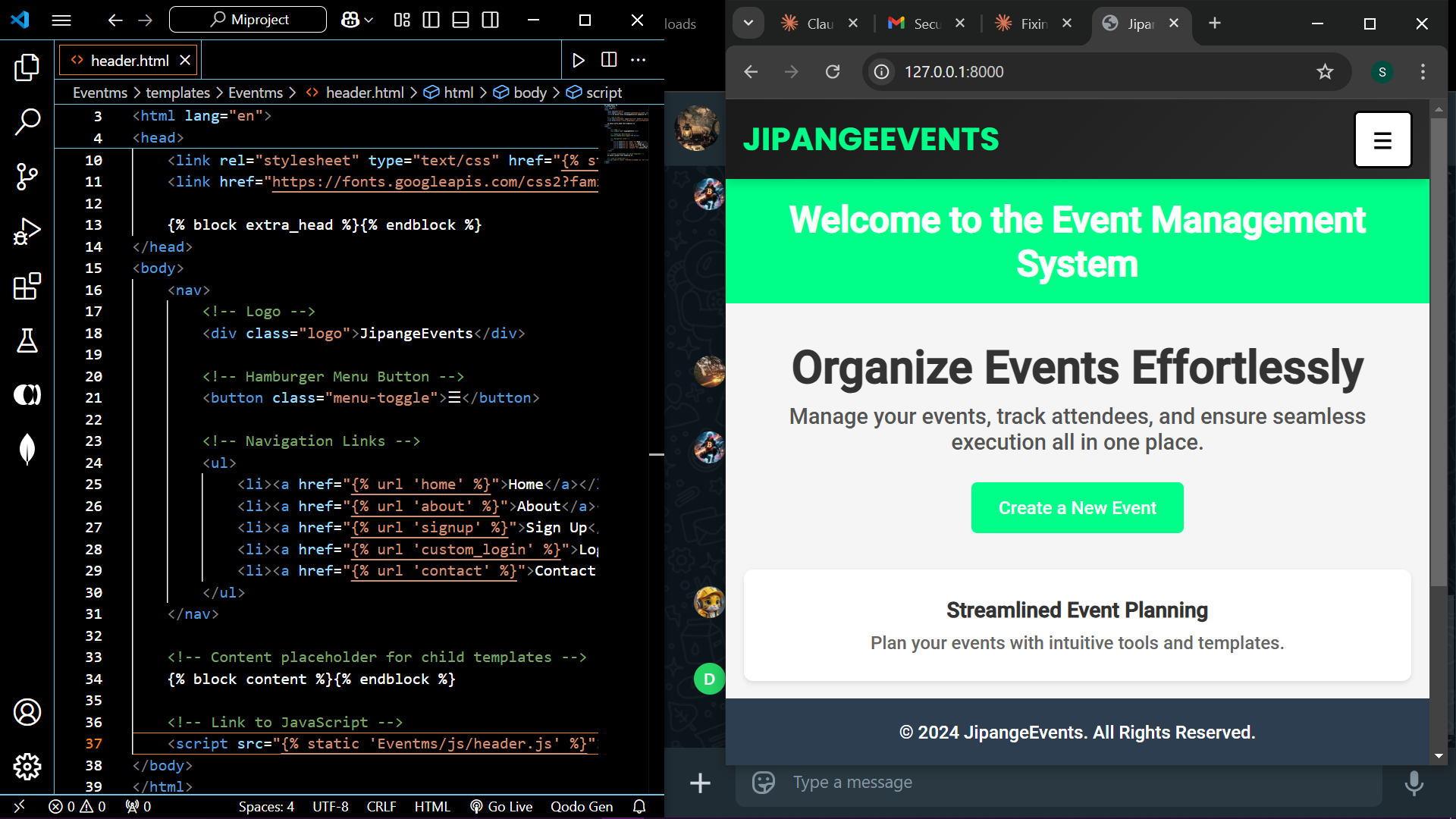
### **Complexity:**

1. **Dynamic Role Management**: Implementing logic for distinct user roles using Django’s authentication system.
2. **Database Relationships**: Properly structuring models to handle events, users, and RSVPs efficiently.
3. **Front-end Integration**: Combining Django templates with CSS and JavaScript for a smooth and responsive user interface.

## **Design Approach**

### **Why This Approach Was Used:**

1. **Django Framework**: Selected for its robustness, built-in authentication, and ease of handling database queries.
2. **Template-Based Design**: Simplifies the separation of logic and presentation, enabling cleaner code and easier maintenance.
3. **Responsive UI**: CSS and JavaScript ensure the application is visually appealing and functional across devices



## **File Overview**

1. **Eventms/static/Eventms/**:
   * **css/signupform.css**: Styles for the signup form.
   * **js/validation.js**: JavaScript file for form validation.
2. **Eventms/templates/Eventms/**:
   * **about.html**: About page describing the project.
   * **back.html**: Placeholder or additional content template for the back button in contact page.
   * **contact.html**: Contact page.
   * **footer.html**: Footer component for consistent design.
   * **header.html**: Header component used in templates.
   * **home.html**: Homepage with navigation links and introductory content.
   * **login.html**: Login page for user authentication.
   * **signup.html**: User registration page with options for organizer or attendee.
   * **signupform.html**: Template for the detailed signup form.
3. **Eventms/**:
   * **\_\_init\_\_.py**: Initializes the app as a Python package.
   * **admin.py**: Admin interface configuration.
   * **apps.py**: Application configuration.
   * **forms.py**: Custom forms for user registration and event creation.
   * **models.py**: Contains the data models for users, events, and RSVPs.
   * **tests.py**: Automated test cases for the app.
   * **urls.py**: Routes specific to the app, including role-based navigation.
   * **views.py**: Manages application logic and processes user requests.

## **How to Run the Application**

1. **Setup the Environment**:
   * Ensure Python and Django are installed.
   * Install required packages using pip install -r requirements.txt.
2. **Configure the Database**:
   * Run python manage.py makemigrations and python manage.py migrate to set up the database.
3. **Run the Server**:
   * Execute python manage.py runserver to start the development server.
   * Open http://127.0.0.1:8000/ in a web browser.
4. **Usage**:
   * Navigate through the navbar to access different pages.
   * Signup and login as an organizer or attendee to experience role-specific features.

## **Additional Information**

1. **Future Enhancements**:
   * Add event filtering and search functionality.
   * Integrate email notifications for RSVPs.
   * Implement real-time updates using WebSockets.
2. **Known Issues**:
   * No functionality for editing or deleting events yet.

This documentation ensures thorough coverage of the project’s structure and provides all necessary information for understanding, running, and extending the application.