Campus Events Website Project

Submitted by: **Honey Patel**
ID : **000798167**
Course: **1617_MCS5013_Web Server Programming**
Date: **11 November 2025**

Abstract

This project focuses on developing a simple yet functional Campus Events Website that allows students to explore, view, and respond to upcoming events on campus. The primary goal is to apply modern web development concepts such as React, Router v6+, React Query v5, Redux Toolkit, Axios, and Yup validation. The system demonstrates efficient state management, data fetching, and form handling practices, aligning with the learning objectives of modern frontend development.

1. Introduction

The Campus Events Website provides a platform where students can easily browse and RSVP to campus events. It emphasizes hands-on learning of React and its associated ecosystem, focusing on user experience, modular code structure, and seamless data synchronization. The project serves as a practical implementation of key concepts covered during the course, bridging theory with real-world frontend application development.

2. System Design and Architecture

The application follows a modular architecture composed of four main pages — Home, Event Details, My Responses, and Sign In. React Router v6+ manages navigation and routing between these views. React Query is used for fetching and caching data from a mock REST API (simulated via json-server). Redux Toolkit manages global state, including user authentication and theme preferences. Yup handles validation for the Sign-In form, ensuring robust and user-friendly input handling.

3. Implementation Details

The implementation integrates the following technologies: - **React**: Component-based UI development. - **React Router v6+**: Multi-page navigation and route protection. - **React Query v5**: Efficient server state management and caching. - **Redux Toolkit**: Global state handling for authentication and UI theme. - **Axios / Fetch API**: Data communication with the server. - **Yup Validation**: Client-side form validation for login and event creation forms. - **Custom Hooks**: Abstraction of data fetching and mutation logic (useEvents, useEvent, useRsvpMutation).

4. Sample Data and API Endpoints

The json-server simulates backend endpoints using a db.json file containing mock data for events,

RSVPs, and users. Sample endpoints include: - GET /events - GET /events/:id - GET /users/:userId/rsvps - POST /rsvps - PATCH /rsvps/:id These endpoints facilitate event listing, RSVP actions, and viewing user-specific responses.

5. User Interface Flow

The user experience begins at the Home page, which lists available events. Selecting an event navigates to its details page where RSVP actions can be performed. If the user is not signed in, they are redirected to the Sign-In page, validated via Yup. Upon successful sign-in, their responses can be viewed under the "My Responses" page. Each page is responsive, data-driven, and connected through React Router for intuitive navigation.

6. Testing and Validation

The application was tested locally using json-server and React Testing Library to ensure functional correctness of routing, state management, and data updates. Form validation and loading states were verified across different user actions to ensure smooth user interaction and error handling.

7. Results and Discussion

The project successfully demonstrates integration of key frontend technologies into a cohesive system. Students can explore event data, manage RSVPs, and experience realistic client-server interactions without needing a full backend infrastructure. The modular architecture and reusable hooks reflect good design practices suitable for scalable applications.

8. Conclusion and Future Work

This project effectively integrates theoretical knowledge with practical implementation using modern React ecosystem tools. Future enhancements may include a full backend with authentication, event creation by admins, and integration with real-time updates using WebSockets or Firebase.

References

1. React Documentation: https://react.dev/
2. React Router v6: https://reactrouter.com/
3. React Query v5: https://tanstack.com/query/v5
4. Redux Toolkit: https://redux-toolkit.js.org/
5. Yup Validation Library: https://github.com/jquense/yup