**SSU-EVENT-M**

**Version 1.0**

**WEB APPLICATION**

**Software Requirements Specification Document**

**1. Introduction**

1.1 Purpose

The purpose of this Software Requirements Specification (SRS) document is to outline the features and functionality of the Event Management Web Application. The document serves as a guide for the development team, providing a clear understanding of the project's requirements, scope, and constraints.

1.2 Scope

The Event Management Web Application is designed to facilitate the management of events by allowing users to register, submit event details, and view the status of their submissions. The application will also provide administrators with the ability to manage and review submitted forms.

1.3 Document Conventions

**User**

Individuals interacting with the application to register, login, submit event form, and check the status of their forms.

**Admin**

Individuals responsible for managing and reviewing submitted event forms.

**2. System Overview**

2.1 System Description

The Event Management Web Application leverages a modern and robust technology stack to ensure a seamless and efficient user experience. Each technology is carefully chosen to address specific aspects of the application's development, ranging from the dynamic user interface to the secure storage and retrieval of data.

2.2 Used Technologies

2.2.1 React.js

React.js is a JavaScript library for building user interfaces. It enables the creation of reusable UI components, providing a fast and interactive experience for users.

React.js is the core frontend library, responsible for rendering dynamic views and managing the application's state.

2.2.2 Tailwind CSS

Tailwind CSS is a utility-first CSS framework that provides low-level utility classes to build designs directly in the markup. It promotes a highly customizable and maintainable styling approach. Tailwind CSS is used for styling the user interface components, ensuring a responsive and visually appealing design.

2.2.3 Node.js

Node.js is a server-side JavaScript runtime. It allows the execution of JavaScript code on the server, providing a scalable and high-performance environment. Node.js powers the backend of the application, handling server-side logic, processing requests, and interacting with the database.

2.2.4 Express.js

Express.js is a web application framework for Node.js. It simplifies the development of web applications by providing a minimal and flexible set of features. Express.js serves as the backend framework, facilitating the creation of robust APIs, handling routes, and managing middleware for the application.

2.2.5 MongoDB

MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. It is designed for scalability, performance, and ease of development. MongoDB is the chosen database for storing and retrieving data related to user accounts, event details, and form submissions.

2.3 System Architecture

The application follows a client-server architecture, with the React.js frontend interacting with the Node.js and Express.js backend. The communication between the frontend and backend is handled through RESTful APIs, ensuring efficient data exchange. MongoDB serves as the persistent data storage, storing user account information, event details, and form submissions.

2.4 Development Environment

The development environment for the Event Management Web Application includes code editors for both frontend and backend development, version control systems, and continuous integration tools. Additionally, proper documentation and coding standards are followed to maintain code quality and facilitate collaborative development.

**3. Functional Requirements**

3.1 User Features

3.1.1 Registration

Users can create an account by providing necessary information.

Users should receive a confirmation email upon successful registration.

3.1.2 Login

Registered users can log in using their credentials.

Users should have access to their personalized dashboard upon successful login.

3.1.3 Logout

Users should be able to log out securely.

User sessions should be terminated upon logout.

3.1.4 Fill Event Form

Users can submit event details through a form.

The form should include fields for essential event information, such as date, time, location, etc.

3.1.5 View Form Details and Status

can view the details of the event form they submitted and its current status.

Users should see a clear representation of the form and its current status on their dashboard.

3.1.6 Download Form Details in PDF Format

Users can download a PDF document containing details of their submitted event form. The PDF document should include all information submitted by the user.

3.2 Admin Features

3.2.1 Login through 4-Digit PIN

Admins can log in using a 4-digit PIN for security.

Admins should gain access to an admin dashboard upon successful login.

3.2.2 Logout

Admins should be able to log out securely.

Admin sessions should be terminated upon logout.

3.2.3 View All Submitted Forms

Admins can view a list of all forms submitted by users.

The list should display essential details of each submitted form.

3.2.4 Edit Form

Admins can edit details of a submitted form.

Changes made by admins should reflect in the form details.

3.2.5 Download Form in PDF Format

Admins can download a PDF document containing details of a submitted event form. The PDF document should include all information submitted by the user.

**4. Future Enhancements**

4.1 Forgot Password

Users should have the ability to reset their password if forgotten.

Users should receive a password reset email with instructions.

4.2 Change Password

Users should be able to change their password.

The new password should be securely updated in the system.

**5. Non-Functional Requirements**

5.1 Performance

The application should load within 3 seconds on standard internet connections.

5.2 Security

User passwords must be stored securely using encryption.

Admin PINs must be securely hashed and stored.

5.3 Usability

The user interface should be intuitive and user-friendly.

The application should be responsive and accessible on various devices.

6. Conclusion

This Software Requirements Specification document serves as a comprehensive guide, detailing the intricate features and essential functionalities of the Event Management Web Application. By encapsulating the requirements and scope of the project, this document establishes a solid foundation for the development team to embark on the creation of a sophisticated and secure system.

6.1 Development Focus

The outlined specifications prioritize the seamless experience of both Users and Admins, ensuring that the application caters to the diverse needs of these distinct user groups. The emphasis on user-friendly interfaces and efficient workflows aims to enhance the overall satisfaction and usability of the application.

6.2 Technical Robustness

The chosen technology stack, including React.js, tailwind CSS, Node.js, Express.js, and MongoDB, has been carefully selected to provide a scalable, performant, and maintainable foundation for the application. This strategic technological alignment ensures that the system remains adaptable to future enhancements and can accommodate increasing user loads.

6.3 Collaboration and Communication

By clearly defining the roles of Users and Admins, this document establishes a framework for collaboration between these user groups. The features and functionalities specified encourage a streamlined communication channel, allowing for efficient event registration, form submission, and administrative management.

6.4 Scalability and Adaptability

The document not only addresses the immediate needs of the application but also looks ahead to future enhancements. The inclusion of future improvements such as password recovery and change functionalities demonstrates a commitment to long-term scalability and adaptability to evolving user requirements.

6.5 Continuous Improvement

As the development progresses, this document will serve as a living reference, allowing for continuous refinement and improvement. Regular reviews and updates to the SRS will enable the development team to align with evolving project dynamics, industry standards, and user feedback.

In conclusion, this Software Requirements Specification document lays the groundwork for a robust, user-centric, and adaptable Event Management Web Application. The collaboration of advanced technologies and a clear understanding of user and administrative needs positions this project for success in meeting current and future expectations.