Event Management System Requirement Document

1. Introduction

The Event Management System is a web application designed to facilitate the planning, organization, and management of events. It aims to provide event organizers and attendees with a comprehensive set of features to streamline the event management process. The system will be developed using Angular (preferred) or Razor Pages (frontend), .NET Core (backend), and SQL Server (database).

2. Features and Functionality

2.1 User Registration and Authentication

- Users & Event Organizers can create accounts and authenticate themselves to access the application.

- Implement password hashing for secure storage of user passwords.

2.2 Event Creation and Management

- Event organizers can create events by providing details such as event name, description, date, time, location, and ticketing information.

- Allow event organizers to manage their events, including editing event details, tracking ticket sales, and generating event reports.

- Event Organizers Should have the option to send Event Notification using Email to all the Registered users in the System.

2.3 Ticketing and Registration

- Users can register for events, purchase tickets, and receive e-tickets.

- Implement ticket inventory management and seat selection (if applicable).

2.4 Event Discovery and Search

- Provide a user-friendly interface for users to browse and search events based on criteria such as location, date, category, or keyword.

- Include features like event recommendations, event ratings, and user reviews to help users discover new events.

2.5 User Profiles and Social Interactions

- Allow users to create profiles with the ability to update Profile information.

- Implement feature to view and download ticket from your Account.

- Implement feature to show Past Events and Upcoming Events for the user.

2.6 Event Notifications and Reminders

- Send email to users for updates in event plans & ticket confirmations.

- Allow users to opt-in or customize their notification preferences.

2.7 Admin Panel

- Create an admin panel for administrators to manage user accounts, review events, handle reported issues, and enforce content policies.

2.8 Basic Validations

1. **User/Event Organizer Registration:**
    - Ensure that all required fields (such as name, email, and password) are filled out before allowing the user/Organizer to register.
    - Validate the email format to ensure it follows the appropriate structure.
    - Implement password strength requirements (e.g., minimum length, combination of uppercase, lowercase, and special characters).
2. **Event Creation:**
    - Validate that all mandatory fields (such as event name, date, time, and location) are provided before allowing event creation.
    - Ensure that the event date is in the future to prevent users from creating events with past dates.
3. **Ticketing and Registration:**

- Validate the number of tickets requested by the user, ensuring it is within the available ticket inventory.
- Implement checks to prevent duplicate registrations by the same user for the same event.
- Validate the payment details provided by the user to ensure they are complete and accurate.

4. **Event Search and Filtering:**
    - Validate user inputs for event search and filtering to prevent unexpected behaviors or errors.
    - Implement safeguards against SQL injection attacks by sanitizing and validating user input before using it in database queries.
5. **User Profiles:**
    - Validate user input for profile updates, ensuring that required fields are filled out and any format restrictions are enforced.
    - Sanitize user-generated content (such as profile descriptions or user-generated event reviews) to prevent cross-site scripting (XSS) attacks.
6. **Event Notifications:**
    - Verify that the user's email address is valid before sending event notifications or reminders.
    - Allow users to opt-in or customize their notification preferences to avoid unnecessary or unwanted notifications.

NOTE: Please add any valid validation that is missing in the list.

3. Technical Requirements

3.1 Frontend

- The front end will be developed using Angular or MVC framework, providing a user-friendly interface.

- Implement components, templates, and data binding to create intuitive user interfaces.

- Follow best practices for code organization, error handling, and performance optimization.

3.2 Backend

- The backend will be developed using .NET Core, providing APIs for user authentication, event creation, ticketing, and data management.

- Implement RESTful APIs for handling front-end requests and responses.

- Utilize SQL Server for data storage and retrieval, implementing efficient database schemas and queries.

3.3 Database

- SQL Server will be used as the database management system to store user accounts, event details, ticket information, and other relevant data.

- Design appropriate database schemas to ensure data integrity and performance.

- Optimize SQL queries for efficient data retrieval and storage.

4. Good to Have Features (Optional)

- As part of User Account Creation, include features like email verification and password reset for enhanced security.

- Provide event organizers with analytics and reports on ticket sales, attendance, revenue, and user demographics.

- Generate customizable event reports for event organizers to track the success and impact of their events.

- Ensure that the application is responsive and accessible on different devices, providing a seamless user experience.