

# Event Management System

## Project Specification Document

### Overview

This document outlines the development and delivery of a comprehensive Event Management System built with modern web technologies. The system will provide a complete solution for managing events, attendees, and related administrative tasks.

### Technology Stack

- **Backend Framework:** Node.js with Express.js
- **Database:** SQLite3 for data persistence
- **Frontend Template Engine:** EJS (Embedded JavaScript)
- **Additional Technologies:** HTML5, CSS3, JavaScript

### Core Features

#### 1. Event Management

- **Create Events:** Add new events with detailed information including title, description, date, time, location, and capacity
- **Edit Events:** Modify existing event details with real-time updates
- **Delete Events:** Remove events with proper confirmation dialogs
- **Event Categories:** Organize events by type (conference, workshop, seminar, social, etc.)
- **Event Status Management:** Track events as draft, published, ongoing, completed, or cancelled

#### 2. Attendee Management

- **Registration System:** Allow users to register for events with personal information
- **Attendee Profiles:** Store and manage attendee contact details and preferences
- **Registration Limits:** Enforce maximum capacity constraints per event
- **Waitlist Management:** Handle overflow registrations when events reach capacity
- **Check-in System:** Mark attendees as present during events

#### 3. User Interface

- **Responsive Design:** Mobile-friendly interface that works across all devices
- **Dashboard:** Administrative overview showing key metrics and upcoming events
- **Event Listings:** Clean, organized display of all events with filtering options

- **Registration Forms:** User-friendly forms for event registration
- **Search Functionality:** Find events by name, date, category, or location

#### 4. Administrative Features

- **User Authentication:** Secure login system for administrators
- **Role-based Access:** Different permission levels for various user types
- **Event Analytics:** Basic reporting on attendance, registration trends
- **Data Export:** Export attendee lists and event data to CSV format
- **System Configuration:** Manage application settings and preferences

### Database Schema

#### Events Table

- Event ID (Primary Key)
- Title
- Description
- Date and Time
- Location
- Maximum Capacity
- Current Registration Count
- Category
- Status
- Created/Modified Timestamps

#### Attendees Table

- Attendee ID (Primary Key)
- Full Name
- Email Address
- Phone Number
- Registration Date
- Check-in Status
- Associated Event ID (Foreign Key)

#### Users Table (Admin)

- User ID (Primary Key)

- Username
- Password (Hashed)
- Role
- Created Date

## User Experience Flow

### For Event Organizers

1. **Login** to the administrative dashboard
2. **Create Events** with comprehensive details
3. **Monitor Registrations** in real-time
4. **Manage Attendees** and handle check-ins
5. **Generate Reports** for post-event analysis

### For Event Attendees

1. **Browse Available Events** on the public listing page
2. **View Event Details** including date, location, and description
3. **Register for Events** through intuitive forms
4. **Receive Confirmation** with event details
5. **Check-in** at the event venue

## Technical Specifications

### Backend Architecture

- RESTful API design with Express.js routing
- SQLite3 database with proper indexing for performance
- Input validation and sanitization for security
- Error handling and logging mechanisms
- Session management for user authentication

### Frontend Implementation

- Server-side rendering with EJS templates
- Responsive CSS framework integration
- JavaScript for dynamic user interactions
- Form validation and AJAX requests
- Accessibility compliance (WCAG guidelines)

## Security Features

- Password hashing with bcrypt
- SQL injection prevention through parameterized queries
- Cross-Site Scripting (XSS) protection
- CSRF token implementation
- Secure session configuration

## Deliverables

### Phase 1: Core System (Week 1-2)

- Database setup and schema implementation
- Basic CRUD operations for events and attendees
- Administrative login system
- Event creation and management interface

### Phase 2: User Interface (Week 2-3)

- Public event listing page
- Registration form implementation
- Responsive design and styling
- Search and filtering functionality

### Phase 3: Advanced Features (Week 3-4)

- Check-in system
- Basic reporting and analytics
- Data export functionality
- Testing and bug fixes

## Final Delivery Package

- Complete source code with documentation
- Database setup scripts
- Installation and deployment guide
- User manual for administrators
- Technical documentation for future maintenance

## System Requirements

### Server Requirements

- Node.js 14.x or higher
- Minimum 512MB RAM
- 10GB available disk space
- Operating System: Windows, macOS, or Linux

### Browser Compatibility

- Chrome 80+
- Firefox 75+
- Safari 13+
- Edge 80+

### Post-Deployment Support

- Initial 30-day bug fix period
- Documentation for common maintenance tasks
- Guidance for future feature additions
- Database backup and recovery procedures

### Project Timeline

- **Total Duration:** 4 weeks
- **Development:** 3 weeks
- **Testing & Deployment:** 1 week
- **Client Training:** 2-3 sessions

### Investment

*[Pricing details to be discussed based on specific requirements and scope]*

---

**Next Steps:** Upon approval of this specification, we will proceed with detailed wireframes and begin the development phase. Any modifications to the scope can be accommodated with timeline and budget adjustments as needed.