PROJECT PLAN

Multi-Event Guest Management System

Introduction

The Multi-Event Guest Management System is a web-based application that allows users to create and manage multiple events, along with the guest lists linked to each event. Instead of using physical lists or spreadsheets, this project provides a structured and centralized digital interface where event information and guests can be added, viewed, edited, and removed easily.

The application is built using **Node.js**, **Express.js**, **MongoDB with Mongoose**, and **EJS**. The user interface combines **Bootstrap 5** with custom CSS, including a built-in dark/light mode toggle for modern usability.

Project Purpose

Managing an event often requires keeping track of people, contact information, and RSVP responses. When several events are happening at once, this becomes difficult to manage manually. The purpose of this project is to simplify and streamline event management.

By keeping all events and their guests in one system, the user can instantly update details, track attendance, and stay organized. This tool works well for both personal and professional events of any size.

Target Users

The system is designed for:

- Students planning school or club events
- Individuals organizing birthdays, parties, or family gatherings
- Small organizations hosting meetings or workshops
- Event planners handling multiple client events
- Anyone who needs a reliable way to track attendees

Application Features

The application includes complete CRUD (Create–Read–Update–Delete) functionality for both events and guests.

Event Features

Users can create a new event by entering basic information such as the name, date, location, description, whether RSVP is required, and the event capacity.
All events appear on the home dashboard in card format, showing a description preview, the event date and location, RSVP information, and a capacity bar.
Events can be edited or deleted, and a confirmation prompt prevents accidental deletions.

Guest Features

Each event has its own dedicated guest list.
Users can view all guests for a specific event in a table layout. Guests can be added, edited, or removed from that event. The table includes each guest's name, email, phone number, and RSVP status.

System Features

The system includes a shared header and footer, responsive layout, custom dark/light mode, and a delete confirmation message. The visual design is fully customized using Bootstrap and CSS.

Technology Stack

The project is built using:

- **Node.js** and **Express.js** for the backend
- **MongoDB Atlas** with **Mongoose** for the database
- **EJS** for server-side templating
- **Bootstrap 5** and custom CSS for the user interface
- **dotenv** for secure configuration
- **GitHub** for version control
- **Render** for deployment

This stack meets the requirements of the assignment while providing a professional structure.

Data Model

The application uses two collections: **Events** and **Guests**.

The **Event** model stores information such as the event name, date, location, description, RSVP requirement, and capacity.

The **Guest** model stores the guest’s name, email, phone number, RSVP response, and the ID of the event they belong to.

Each guest is linked to an event through its eventId, allowing the system to keep guest lists separate per event.

Page and UI Design

The user interface is built around clarity and ease of use.

Dashboard

The dashboard shows all events in a grid of responsive cards. Each card includes the event name, location, date, description snippet, RSVP status, and a visual capacity indicator. Action buttons allow the user to view guests, edit the event, or delete it.

Guest List Page

Each event has a table that displays all guests assigned to it. Action buttons allow editing or deleting guest entries.

Add/Edit Pages

Both events and guests have forms that use Bootstrap styling. These pages also support dark/light mode.

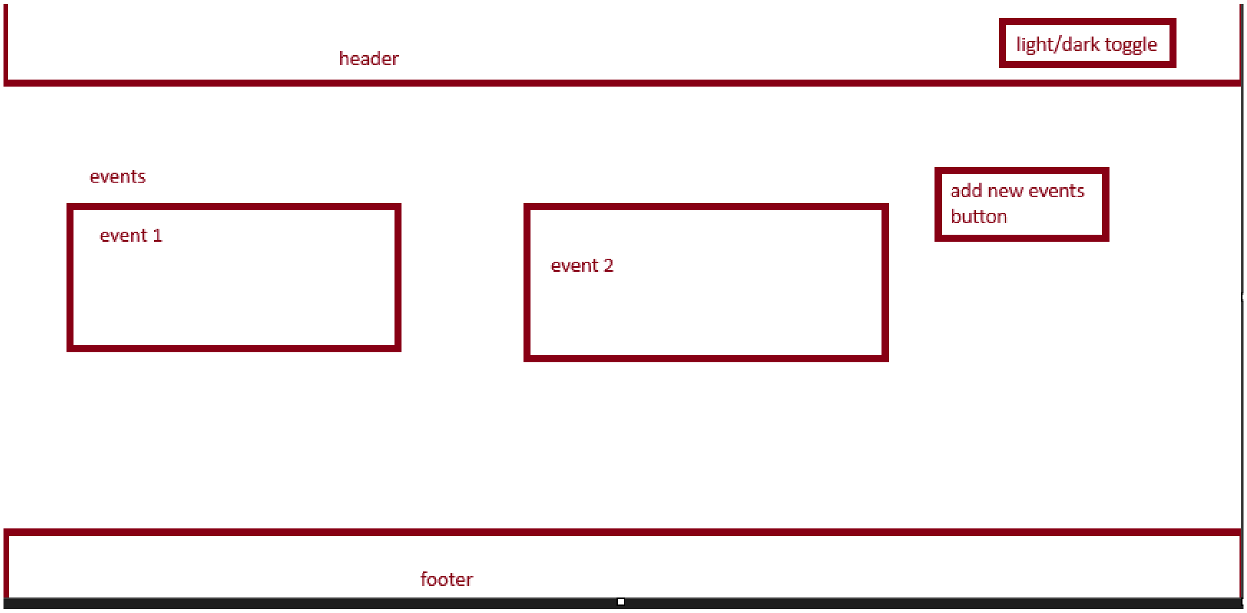
Navigation

The header includes a dark mode toggle and consistent branding. The footer appears on every page for a complete layout.

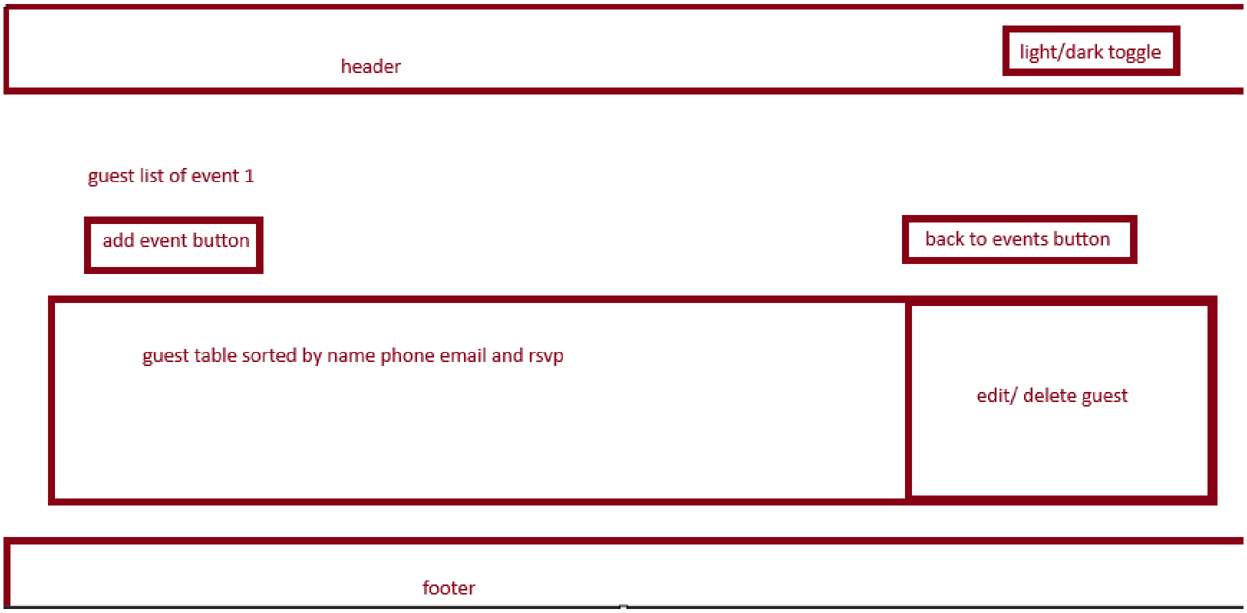
Visual Layout Sketches

Below are simple text-based representations that illustrate the planned interface:

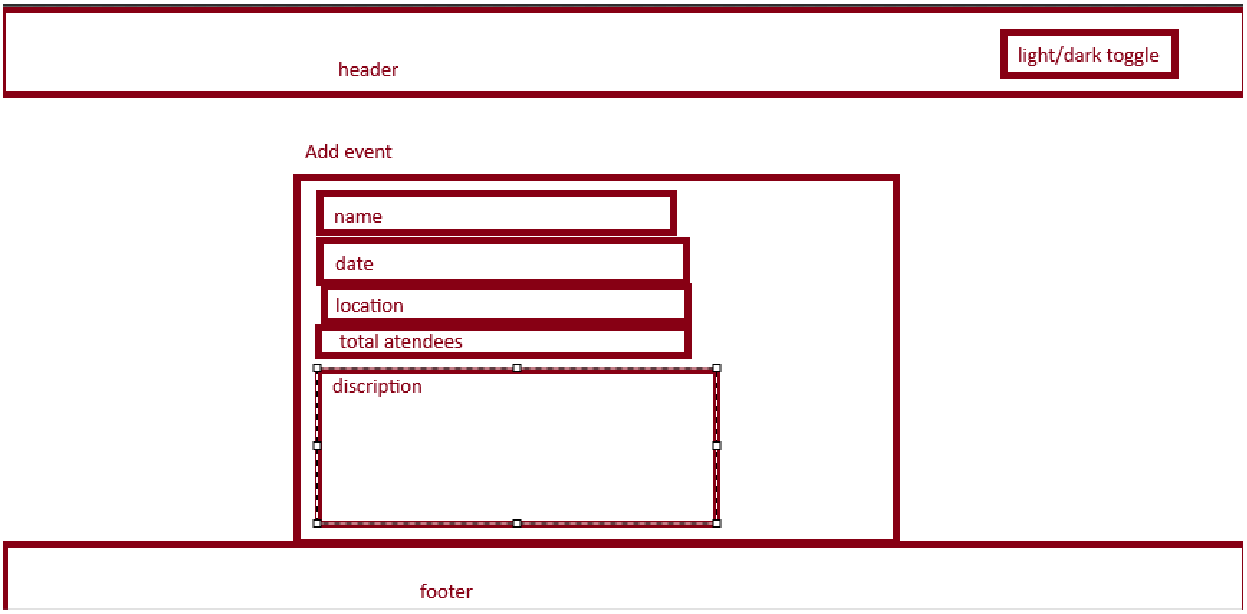
Dashboard



Guest List



Form/add event Page



Expected Challenges

Some challenges anticipated during development include designing a custom UI without copying class examples, connecting the application to MongoDB using environment variables, ensuring that the .env file remains hidden using .gitignore, and deploying the final project to a cloud hosting provider.

Another challenge was creating a system that supports **multiple events**, each with its own guest list, without overcomplicating the route structure.

Conclusion

The Multi-Event Guest Management System is a complete and practical CRUD application that meets all assignment requirements. It supports event creation, guest management, dynamic templating, a clean UI, and secure database integration.

The project demonstrates the student’s ability to build a fully functional full-stack web application with Express.js and MongoDB, while also applying UI/UX principles and proper project structure.