



# Project Document: Event Management System

## 1. Project Overview

The **Event Management System** is a web-based platform for managing events, RSVPs, and user accounts.

### Key Features:

- **Admin Functionality:**
  - Create, update, and delete events.
  - View RSVP lists for events.
- **User Functionality:**
  - View upcoming events.
  - RSVP to events.
- **Authentication:**
  - JWT-based authentication for secure access.
  - Role-based access (Admin/User).

This project uses a modern tech stack:

- **Frontend:** React (styled for a clean and crisp user experience).
- **Backend:** Node.js with Express.js.
- **Database:** MariaDB.

---

## 2. Project Setup

### 2.1 Backend Setup

1. Open the backend folder in visual studio code, open up a terminal, and Install dependencies:

```
npm install
```

2. Open up the .env folder in the backend code and update the DB\_PASSWORD and DB\_USER

**DB\_USER=root**

**DB\_PASSWORD=<your\_db\_password>**

3. Import the database:

- Open a normal terminal where the create\_database.sql file exists
- Run the SQL script (create\_database.sql) using MariaDB/MySQL:

```
mysql -u <username> -p < create_database.sql
```

if for any reason the command doesn't work the .sql file simply contains SQL commands that you can run in the mariadb terminal

4. Start the server:

```
npm run start
```

The backend will run on http://localhost:5000.

---

## 2.2 Frontend Setup

1. Open the frontend folder in a **new vscode window (keep the backend service up)**
2. Install dependencies:

```
npm install
```

The frontend will run on http://localhost:3000.

---

## 3. Project Deliverables

### 3.1 Postman Workspace

- A Postman collection that tests all backend APIs.
- The collection should includes:
  - User registration and login.
  - Admin-only APIs for event management.
  - User APIs for viewing events and RSVPing.
- **Export both the environment variables and the collection.**

### 3.2 Mocha + Selenium Test Folder

- A folder named tests that contains:
  - **Mocha + Selenium Tests:** Automated end-to-end tests for the frontend, including:

- User login and navigation.
- Admin event creation and deletion.
- RSVP functionality.

### 3.3 SQL Script

- Any SQL instructions you had to use to setup your test cases should be placed inside a pdf file along with an explanation for why you needed to use them.
- As an example: to properly test all functionality you need to setup both a normal user and an admin user on the site, so you will need to manually run a query to create an admin user. Put the query as well as why you needed to run it in the file.

**Thank You**  
**Edges For Training Team**