# Portfolio Project Report: Event Management System

Table of Contents

[Portfolio Project Report: Event Management System 1](#_Toc168740686)

[1. Features of the Project 1](#_Toc168740687)

[2. API Endpoints 1](#_Toc168740688)

[Event Management 2](#_Toc168740689)

[RSVP Management 2](#_Toc168740690)

[Notifications 2](#_Toc168740691)

[**ER Diagram** (Entity-Relationship Diagram) 4](#_Toc168740692)

# 1. Features of the Project

The Event Management System will offer the following features:

1. **User Authentication and Profile Management**
   * User registration and login.
   * User profile management (view and update profile details).
2. **Event Management**
   * Create, view, update, and delete events.
   * View all events created by a user.
   * Event categorization and search functionality.
3. **RSVP Management**
   * RSVP to an event.
   * View all RSVPs for a user.
   * Manage RSVP status (accept, decline, or tentative).
4. **Notifications**
   * Notifications for upcoming events.
   * Reminders for RSVP deadlines.

# 2. API Endpoints

Below are the API endpoints required for each feature, including the HTTP verbs, endpoint URLs, and details:

**User Authentication and Profile Management**

* **Register User**
  + **POST** /users/register
  + **Request Body**: { "username": "string", "email": "string", "password": "string" }
* **Login User**
  + **POST** /users/login
  + **Request Body**: { "email": "string", "password": "string" }
* **View User Profile**
  + **GET** /users/profile
  + **Headers**: { "Authorization": "Bearer token" }
* **Update User Profile**
  + **PUT** /users/profile
  + **Headers**: { "Authorization": "Bearer token" }
  + **Request Body**: { "username": "string", "email": "string", "password": "string" }

# Event Management

* **Create Event**
  + **POST** /events
  + **Headers**: { "Authorization": "Bearer token" }
  + **Request Body**: { "title": "string", "description": "string", "date": "datetime", "category": "string" }
* **View All Events**
  + **GET** /events
* **View Single Event**
  + **GET** /events/{event\_id}
* **Update Event**
  + **PUT** /events/{event\_id}
  + **Headers**: { "Authorization": "Bearer token" }
  + **Request Body**: { "title": "string", "description": "string", "date": "datetime", "category": "string" }
* **Delete Event**
  + **DELETE** /events/{event\_id}
  + **Headers**: { "Authorization": "Bearer token" }
* **Search Events**
  + **GET** /events/search
  + **Query Parameters**: ?title=string&category=string&date=datetime

# RSVP Management

* **RSVP to Event**
  + **POST** /events/{event\_id}/rsvp
  + **Headers**: { "Authorization": "Bearer token" }
  + **Request Body**: { "status": "string" }
* **View All RSVPs for a User**
  + **GET** /users/rsvps
  + **Headers**: { "Authorization": "Bearer token" }
* **Update RSVP Status**
  + **PUT** /events/{event\_id}/rsvp
  + **Headers**: { "Authorization": "Bearer token" }
  + **Request Body**: { "status": "string" }

# Notifications

* **View Notifications**
  + **GET** /users/notifications
  + **Headers**: { "Authorization": "Bearer token" }

**3. Database Design**

**Database Name**: event\_management\_db

**Tables**

1. **users**
   * **Columns**:
     + id (Integer, Primary Key, Auto-increment)
     + username (String, Unique, Not Null)
     + email (String, Unique, Not Null)
     + password (String, Not Null)
2. **events**
   * **Columns**:
     + id (Integer, Primary Key, Auto-increment)
     + title (String, Not Null)
     + description (Text)
     + date (DateTime, Not Null)
     + category (String)
     + user\_id (Integer, Foreign Key referencing users(id), Not Null)
3. **rsvps**
   * **Columns**:
     + id (Integer, Primary Key, Auto-increment)
     + user\_id (Integer, Foreign Key referencing users(id), Not Null)
     + event\_id (Integer, Foreign Key referencing events(id), Not Null)
     + status (String, Not Null)
     + created\_at (DateTime, Default: datetime.now(timezone.utc))
4. **notifications**
   * **Columns**:
     + id (Integer, Primary Key, Auto-increment)
     + user\_id (Integer, Foreign Key referencing users(id), Not Null)
     + message (String, Not Null)
     + created\_at (DateTime, Default: datetime.now(timezone.utc))

# **ER Diagram** (Entity-Relationship Diagram)

+------------+ +---------+ +-----------+

| users | | events | | rsvps |

+------------+ +---------+ +-----------+

| id (PK) |<----->| user\_id | | id (PK) |

| username | | id (PK) |<----->| user\_id |

| email | | title | | event\_id |

| password | | desc | | status |

+------------+ | date | | created\_at|

| category| +-----------+

+---------+

^

|

+---------+

|notifications|

+---------+

| id (PK) |

| user\_id |

| message |

| created\_at |

+---------+

