# Event Management Application Documentation

## Table of Contents

1. Introduction  
2. Prerequisites  
3. Setup Instructions  
4. Running the Project  
5. API Documentation

## Introduction

The Event Management Application is a full-stack project with a Node.js-based backend and a React-based frontend. It allows users to manage events, tasks, and attendees through a simple web interface.

## Prerequisites

• Node.js (version 16 or later)  
• npm (Node Package Manager, comes with Node.js)  
• SQLite (for database management)  
• A code editor like VS Code  
• A browser (Chrome, Firefox, etc.)

## Setup Instructions

### Backend Setup

1. Navigate to the backend folder:  
 cd event-management-backend  
2. Install the required dependencies:  
 npm install  
3. Start the backend server:  
 nodemon index.js  
By default, the backend runs on http://localhost:5000.

### Frontend Setup

1. Navigate to the frontend folder:  
 cd event-management-frontend  
2. Install the required dependencies:  
 npm install  
3. Start the frontend server:  
 npm start  
The frontend will run on http://localhost:3000.

## Running the Project

1. Ensure the backend is running by checking http://localhost:5000 in the browser. You should see 'Event Management Server is running'.  
2. Start the frontend and navigate to http://localhost:3000. The web interface should load.  
3. Use the navigation buttons to interact with events, tasks, and attendees.

## API Documentation

### Base URL

http://localhost:5000/api

### Attendees API

GET /attendees  
Description: Fetch all attendees.  
Request: No parameters required.  
Response:  
[{"id": 1, "name": "John Doe", "email": "john.doe@example.com"}, {"id": 2, "name": "Jane Smith", "email": "jane.smith@example.com"}]

GET /attendees/:id  
Description: Fetch a specific attendee by ID.  
Request Parameter:  
- id (integer): The ID of the attendee.  
Response:  
{"id": 1, "name": "John Doe", "email": "john.doe@example.com"}

### Events API

GET /events  
Description: Fetch all events.  
Request: No parameters required.  
Response:  
[{"id": 1, "name": "Conference 2024", "date": "2024-01-15", "location": "New York"}, {"id": 2, "name": "Webinar on React", "date": "2024-02-20", "location": "Online"}]

GET /events/:id  
Description: Fetch a specific event by ID.  
Request Parameter:  
- id (integer): The ID of the event.  
Response:  
{"id": 1, "name": "Conference 2024", "date": "2024-01-15", "location": "New York"}

### Tasks API

GET /tasks  
Description: Fetch all tasks.  
Request: No parameters required.  
Response:  
[{"id": 1, "description": "Prepare slides", "status": "completed"}, {"id": 2, "description": "Send invites", "status": "pending"}]

GET /tasks/:id  
Description: Fetch a specific task by ID.  
Request Parameter:  
- id (integer): The ID of the task.  
Response:  
{"id": 1, "description": "Prepare slides", "status": "completed"}