Event Management Dashboard Documentation

Instructions to Set Up and Run the Project:

1. Clone the repository from GitHub:

```
git clone <repository-link>
```

2. Navigate to the project directory:

```
cd event-management-dashboard
```

3. Install dependencies for both backend and frontend:

   - Backend (Django): pip install -r requirements.txt

   - Frontend (React): npm install

4. Set up the database:

   - Run migrations for Django: python manage.py migrate

5. Start the development servers:

   - Backend: python manage.py runserver

   - Frontend: npm start

6. Access the application at http://localhost:3000.

API Details:

Event Management API:

1. Create Event (POST /api/events/):

   - Input: { "name": "Event Name", "description": "Details", "location": "Place", "date": "YYYY-MM-DD" }

   - Output: Event details on success.

2. Get All Events (GET /api/events/):

   - Output: List of all events.

3. Update Event (PUT /api/events/{id}/):

- Input: Updated event details.

- Output: Updated event.

4. Delete Event (DELETE /api/events/{id}/):

- Output: Success message.

Attendee Management API:

1. Add Attendee (POST /api/attendees/):

- Input: { "name": "Attendee Name", "email": "Email Address" }

- Output: Attendee details on success.

2. Get All Attendees (GET /api/attendees/):

- Output: List of all attendees.

3. Delete Attendee (DELETE /api/attendees/{id}/):

- Output: Success message.

Task Management API:

1. Create Task (POST /api/tasks/):

- Input: { "name": "Task Name", "deadline": "YYYY-MM-DD", "status": "Pending", "eventId": "Event ID" }

- Output: Task details on success.

2. Get Tasks for an Event (GET /api/tasks/{eventId}/):

- Output: List of tasks for the event.

3. Update Task Status (PUT /api/tasks/{id}/status/):

- Input: { "status": "Completed" }

- Output: Updated task details.