Here’s the API documentation in \*\*Markdown\*\* format, including details about the endpoints, sample requests, and sample data for testing.

---

# \*\*Campus Event Management System API Documentation\*\*

## \*\*Base URL\*\*

`http://localhost:5000/api`

---

## \*\*Authentication Endpoints\*\*

### \*\*1. Register User\*\*

\*\*POST\*\* `/auth/register`

#### \*\*Request Body\*\*

```json

{

"username": "JohnDoe",

"email": "johndoe@example.com",

"password": "password123",

"role": "user",

"preferences": ["workshop", "seminar"]

}

```

#### \*\*Response\*\*

- \*\*201 Created\*\*

```json

{

"message": "User registered successfully"

}

```

- \*\*400 Bad Request\*\*

```json

{

"message": "Email already in use"

}

```

---

### \*\*2. Login User\*\*

\*\*POST\*\* `/auth/login`

#### \*\*Request Body\*\*

```json

{

"email": "johndoe@example.com",

"password": "password123"

}

```

#### \*\*Response\*\*

- \*\*200 OK\*\*

```json

{

"token": "JWT\_TOKEN\_HERE",

"user": {

"id": "USER\_ID\_HERE",

"username": "JohnDoe",

"role": "user"

}

}

```

- \*\*400 Bad Request\*\*

```json

{

"message": "Invalid credentials"

}

```

- \*\*404 Not Found\*\*

```json

{

"message": "User not found"

}

```

---

## \*\*Event Endpoints\*\*

### \*\*1. Get All Events\*\*

\*\*GET\*\* `/events`

#### \*\*Headers\*\*

```json

{

"Authorization": "Bearer JWT\_TOKEN\_HERE"

}

```

#### \*\*Response\*\*

- \*\*200 OK\*\*

```json

[

{

"id": "EVENT\_ID\_HERE",

"name": "Tech Workshop",

"date": "2024-12-15T09:00:00.000Z",

"time": "10:00 AM",

"location": "Room 101",

"description": "Learn about the latest tech trends",

"capacity": 50,

"availableSeats": 48,

"createdBy": "ADMIN\_ID\_HERE"

}

]

```

---

### \*\*2. RSVP to an Event\*\*

\*\*POST\*\* `/events/:id/rsvp`

#### \*\*Headers\*\*

```json

{

"Authorization": "Bearer JWT\_TOKEN\_HERE"

}

```

#### \*\*Response\*\*

- \*\*200 OK\*\*

```json

{

"message": "RSVP successful",

"event": {

"id": "EVENT\_ID\_HERE",

"name": "Tech Workshop",

"availableSeats": 47

}

}

```

- \*\*400 Bad Request\*\*

```json

{

"message": "No seats available"

}

```

- \*\*404 Not Found\*\*

```json

{

"message": "Event not found"

}

```

---

### \*\*3. Create an Event (Admin Only)\*\*

\*\*POST\*\* `/events/create`

#### \*\*Headers\*\*

```json

{

"Authorization": "Bearer JWT\_TOKEN\_HERE"

}

```

#### \*\*Request Body\*\*

```json

{

"name": "Tech Workshop",

"date": "2024-12-15",

"time": "10:00 AM",

"location": "Room 101",

"description": "Learn about the latest tech trends",

"capacity": 50

}

```

#### \*\*Response\*\*

- \*\*201 Created\*\*

```json

{

"id": "EVENT\_ID\_HERE",

"name": "Tech Workshop",

"date": "2024-12-15T09:00:00.000Z",

"time": "10:00 AM",

"location": "Room 101",

"description": "Learn about the latest tech trends",

"capacity": 50,

"availableSeats": 50,

"createdBy": "ADMIN\_ID\_HERE"

}

```

- \*\*403 Forbidden\*\*

```json

{

"message": "Access denied: Admins only"

}

```

- \*\*500 Internal Server Error\*\*

```json

{

"error": "Failed to create event"

}

```

---

## \*\*Sample Data for Testing\*\*

### \*\*Sample Users\*\*

- \*\*Admin User\*\*:

```json

{

"username": "AdminUser",

"email": "admin@example.com",

"password": "adminpassword",

"role": "admin"

}

```

- \*\*Regular User\*\*:

```json

{

"username": "RegularUser",

"email": "user@example.com",

"password": "userpassword",

"role": "user",

"preferences": ["workshop", "seminar"]

}

```

### \*\*Sample Events\*\*

- \*\*Event 1\*\*:

```json

{

"name": "Tech Workshop",

"date": "2024-12-15",

"time": "10:00 AM",

"location": "Room 101",

"description": "Learn about the latest tech trends",

"capacity": 50

}

```

- \*\*Event 2\*\*:

```json

{

"name": "Art Seminar",

"date": "2024-12-20",

"time": "2:00 PM",

"location": "Auditorium",

"description": "Explore modern art techniques",

"capacity": 30

}

```

---

## \*\*Installation and Running Locally\*\*

1. Clone the repository:

```bash

git clone https://github.com/your-repo-link.git

cd backend

```

2. Install dependencies:

```bash

npm install

```

3. Create a `.env` file and add the following:

```plaintext

MONGO\_URI=mongodb+srv://username:password@cluster0.mongodb.net/event-management

JWT\_SECRET=your\_super\_secret\_key

PORT=5000

```

4. Start the server:

```bash

npm run dev

```

5. Test the API using Postman or any API testing tool.

---

## \*\*Feature Checklist\*\*

- [x] User Registration

- [x] User Login

- [x] Admin Event Creation

- [x] RSVP Functionality

- [x] Event Listings

- [x] Role-based Access Control

---

Feel free to use this documentation in your project! Let me know if you need any more details.