**Event Management API**

This is an Event Management API built with Node.js, Express, and SQLite. The API allows users to create, edit, delete, and retrieve events. Users can also register and unregister for events. Authentication is handled using JSON Web Tokens (JWT).

**Project Structure**

.env

.gitignore

app.js

controllers/

    events-controller.js

    users-controller.js

CoPilot\_instructions.txt

database\_bkp.sqlite

database.js

database.sqlite

models/

    event.js

    user.js

package.json

public/

    images/

ReadMe.md.txt

routes/

    events.js

    users.js

util/

    auth.js

    upload.js

**Installation**

1. Clone the repository.
2. Install the dependencies:

npm install

1. Create a [.env](vscode-file://vscode-app/c:/Users/mishr/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) file with the following content:

PORT=3000

SECRET\_KEY=your\_secret\_key

1. Start the server:

npm run dev

**API Endpoints**

**User Authentication**

**Signup**

* **URL:** /users/signup
* **Method:** POST
* **Request Body:**

{

  "email": "user@example.com",

  "password": "password123"

}

* **Response:**

{

  "user": {

    "id": 1,

    "email": "user@example.com"

  },

  "token": "jwt\_token"

}

**Login**

* **URL:** /users/login
* **Method:** POST
* **Request Body:**

{

  "email": "user@example.com",

  "password": "password123"

}

* **Response:**

{

  "message": "Login successful",

  "token": "jwt\_token"

}

**Event Management**

**Create Event**

* **URL:** /events
* **Method:** POST
* **Headers:**

{

  "Authorization": "Bearer jwt\_token"

}

* **Request Body:**

{

  "title": "Event Title",

  "description": "Event Description",

  "address": "Event Address",

  "date": "2023-10-01"

}

* **Response:**

{

  "id": 1,

  "title": "Event Title",

  "description": "Event Description",

  "address": "Event Address",

  "date": "2023-10-01",

  "image": "image\_filename",

  "userId": 1

}

**Edit Event**

* **URL:** /events/:id
* **Method:** PUT
* **Headers:**

{

  "Authorization": "Bearer jwt\_token"

}

* **Request Body:**

{

  "title": "Updated Event Title",

  "description": "Updated Event Description",

  "address": "Updated Event Address",

  "date": "2023-10-02"

}

* **Response:**

{

  "message": "Event updated successfully",

  "event": {

    "id": 1,

    "title": "Updated Event Title",

    "description": "Updated Event Description",

    "address": "Updated Event Address",

    "date": "2023-10-02",

    "image": "image\_filename"

  }

}

**Delete Event**

* **URL:** /events/:id
* **Method:** DELETE
* **Headers:**

{

  "Authorization": "Bearer jwt\_token"

}

* **Response:**

{

  "message": "Event deleted successfully"

}

**Get All Events**

* **URL:** /events
* **Method:** GET
* **Response:**

[

  {

    "id": 1,

    "title": "Event Title",

    "description": "Event Description",

    "address": "Event Address",

    "date": "2023-10-01",

    "image": "image\_filename",

    "userId": 1

  }

]

**Get Event by ID**

* **URL:** /events/:id
* **Method:** GET
* **Response:**

{

  "id": 1,

  "title": "Event Title",

  "description": "Event Description",

  "address": "Event Address",

  "date": "2023-10-01",

  "image": "image\_filename",

  "userId": 1

}

**Register for Event**

* **URL:** /events/:id/register
* **Method:** POST
* **Headers:**

{

  "Authorization": "Bearer jwt\_token"

}

* **Response:**

{

  "message": "Registered to event successfully"

}

**Unregister from Event**

* **URL:** /events/:id/unregister
* **Method:** POST
* **Headers:**

{

  "Authorization": "Bearer jwt\_token"

}

* **Response:**

{

  "message": "Unregistered from event successfully"

}

**Middleware**

**Authentication**

The [authenticate](vscode-file://vscode-app/c:/Users/mishr/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) middleware verifies the JWT token and adds the user information to the request object.

import { authenticate } from '../util/auth.js';

**File Upload**

The [upload](vscode-file://vscode-app/c:/Users/mishr/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html) middleware handles file uploads using Multer.

import { upload } from '../util/upload.js';

**Database**

The database is initialized using SQLite with the following tables:

* [users](vscode-file://vscode-app/c:/Users/mishr/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Stores user information (id, email, password).
* [events](vscode-file://vscode-app/c:/Users/mishr/AppData/Local/Programs/Microsoft%20VS%20Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html): Stores event information (id, title, description, address, date, image, user\_id).
* registrations: Stores event registrations (event\_id, user\_id).

**License**

This project is licensed under the ISC License.

Similar code found with 1 license type -

View matches