

Installation Guide & User Manual

Free Food FIU

Project File Structure:

```
├── Free-Food-FIU/
│   ├── backend/
│   │   ├── config/
│   │   │   └── db.js
│   │   ├── controllers/
│   │   │   ├── eventController.js
│   │   │   └── userController.js
│   │   ├── middleware/
│   │   │   ├── authMiddleware.js
│   │   │   ├── errorMiddleware.js
│   │   │   └── fileUploadMiddleware.js
│   │   ├── models/
│   │   │   ├── eventModel.js
│   │   │   └── userModel.js
│   │   ├── routes/
│   │   │   ├── eventRoutes.js
│   │   │   └── userRoutes.js
│   │   ├── utils/
│   │   │   └── generateToken.js
│   │   └── server.js
│   ├── frontend/
│   │   ├── logo/
│   │   │   └── logo.png
│   │   ├── public/
│   │   │   └── vite.svg
│   │   ├── src/
│   │   │   ├── assets/
│   │   │   │   └── react.svg
│   │   │   ├── components/
│   │   │   │   ├── event-creation/diet-option-selectors/
│   │   │   │   │   ├── allergies/
│   │   │   │   │   │   ├── AllergyOptions.jsx
│   │   │   │   │   │   └── AllergySelectComponent.jsx
```

- | | | | | preferences/
- | | | | | | PreferenceOptions.jsx
- | | | | | | | PreferenceSelectComponent.jsx
- | | | | | | ErrorMessage.jsx
- | | | | | | EventBox.jsx
- | | | | | | FormContainer.jsx
- | | | | | | Header.jsx
- | | | | | | Hero.jsx
- | | | | | | Loader.jsx
- | | | | | | | PrivateRoute.jsx
- | | | | | | css/
- | | | | | | | EventBox.css
- | | | | | | | | EventCreationScreen.css
- | | | | | | | screens/
- | | | | | | | | EventCreationScreen.jsx
- | | | | | | | | EventsScreen.jsx
- | | | | | | | | HomeScreen.jsx
- | | | | | | | | LoginScreen.jsx
- | | | | | | | | ProfileScreen.jsx
- | | | | | | | | RegisterScreen.jsx
- | | | | | | | | | UpdateProfileScreen.jsx
- | | | | | | | | slices/
- | | | | | | | | | apiSlice.js
- | | | | | | | | | authSlice.js
- | | | | | | | | | eventApiSlice.js
- | | | | | | | | | | usersApiSlice.js
- | | | | | | | | App.jsx
- | | | | | | | index.css
- | | | | | | | main.jsx
- | | | | | | | | store.js
- | | | | | | uploads/
- | | | | | | | | placeholder.txt
- | | | | | | | | | .eslintrc.cjs
- | | | | | | | | | .gitignore
- | | | | | | | | | index.html
- | | | | | | | | | package-lock.json
- | | | | | | | | | package.json
- | | | | | | | | | | vite.config.js
- | | | | | | | | | .gitignore
- | | | | | | | | | package-lock.json

- └─ package.json
- └─ README.md

The backend and frontend directories (explained):

The backend folder contains all the resources to make a successful connexion with the Mongo Database. Also it has all the files to do the CRUD operation for the Events and Users collections.

The frontend folder contains all the resources to create the UI, such as the design of the pages. Also, it has the files necessary to connect the pages to the backend. Furthermore, the folder uploads is where the images for the events are stored locally.

Installation & Running steps:

To run the webapp it is necessary to install Node.js locally. The necessary installation file is at:

<https://nodejs.org/en/download>

After Node.js installation, we need to install the node modules for the project. There are two places where this is necessary the root folder and the frontend folder. This process is done using a terminal. We recommend using the VSCode terminal first at the root to run "npm i" and then in the frontend folder using "cd frontend" to run "npm i" again. That would be:

```
npm i
cd frontend
npm i
cd ..
```

It is necessary to create a .env file in the root folder. The structure is:

```
NODE_ENV=development
PORT=<any number but 3000>
MONGO_URI=<your connection link to your db>
JWT_SECRET=<anything>
```

Finally, run the command "npm run dev" to start the server and the client at the root folder that is why "cd .." was used previously.

Setting up MongoDB (the database):

Create a MongoDB Atlas Account:

Go to the MongoDB Atlas website (<https://www.mongodb.com/cloud/atlas>) and sign up for a new account if you don't have one.

Follow the on-screen instructions to set up your account and create a new cluster. Choose the free tier or any other suitable plan.

Create a MongoDB Atlas Database:

Once your cluster is set up, create a new database by navigating to the "Database Access" section under the "Security" menu.

Click "Add New Database User" to create a new user and assign appropriate roles (e.g., readWrite) to access the database.

Configure Network Access:

In the MongoDB Atlas dashboard, go to the "Network Access" section under the "Security" menu.

Click "Add IP Address" and select "Allow Access From Anywhere" (or specify your IP address if you want to restrict access).

Get Connection String:

Go to the "Clusters" section and click on your cluster to view its details.

Click the "Connect" button and choose "Connect Your Application."

Copy the connection string provided by MongoDB Atlas. It should look like:

```
mongodb+srv://<username>:<password>@<cluster-address>/<database>?retryWrites=true&w=
majority
```

This URI is what you will use in your .env file.