**Event Ticketing Admin Frontend**

**Documentation**

1. **Project Technical Specification**

* This project is a ReactJs/Redux/Saga based project which uses the latest Node Version (20.11.1).
* The project user interface is supported by Material UI Design library with built in user interface components, & other utilities.

For MUI library references you can visit their documentations for implementation:

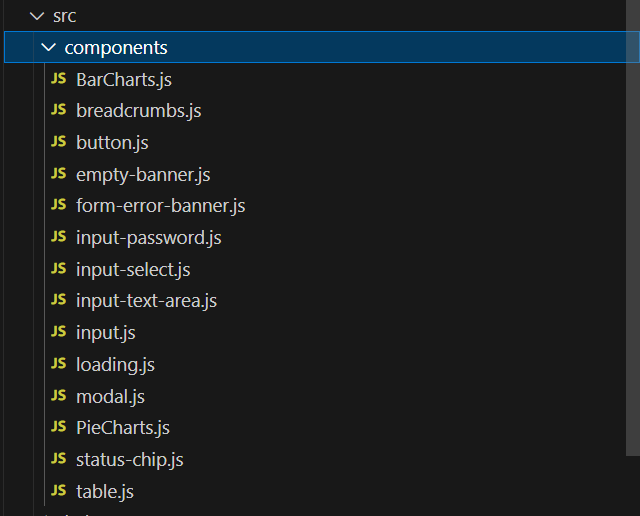
<https://mui.com/>

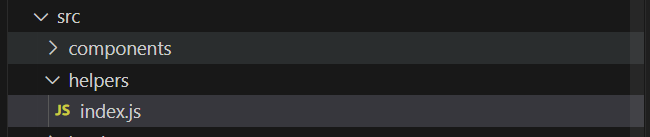
1. **Installing project specific dependency packages and tools & startup**

* To install the node packages go to the project directory & open a git bash terminal then run the command npm install.
* After installing the packages run the command npm start then you will be redirected to the default local hosting made by node ReactJs project.

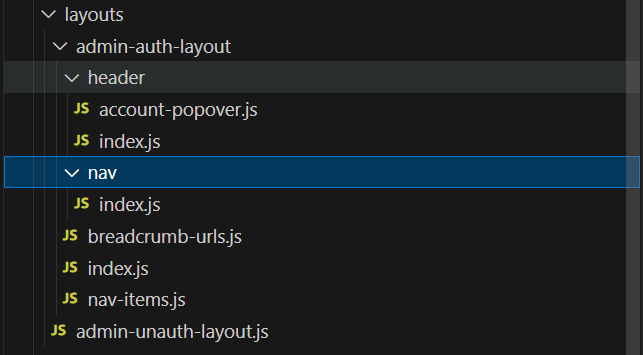
**Note:** make sure that you have installed the latest node version on your local machine to avoid packages conflict upon installation.

1. **Project folder structure**

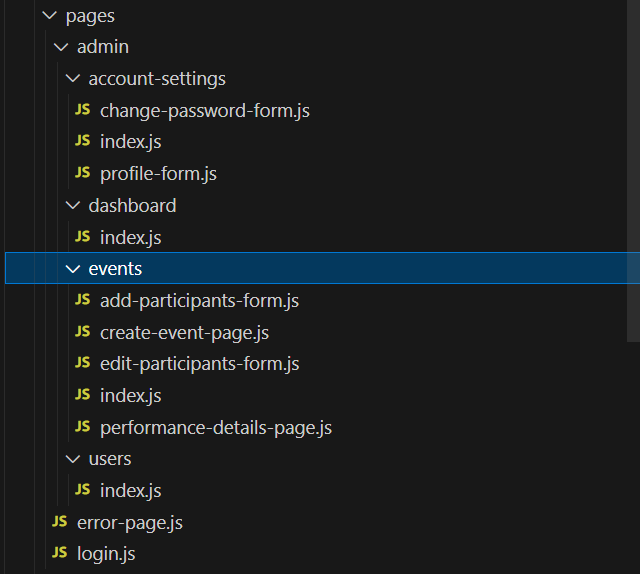
* **Components:** reusable user interface components.
* **Helpers:** reusable global function helpers.

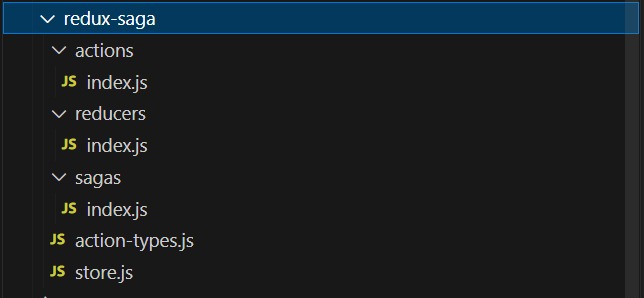
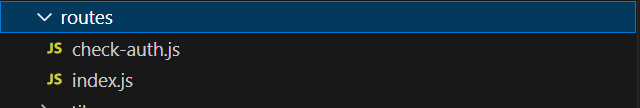


* **Layouts:**  application’s page header, footer, & navigation.



* **Pages:**  pages files for the web application.



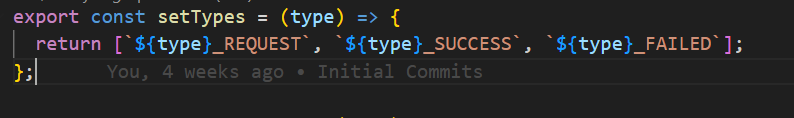
* **Redux Saga:** state managements & Axios API calls implementation.
* **Routes:** web page path locations implementation.
* **Utils:**  utilities for the project’s code implementation.

1. **Axios & API Calls flow**

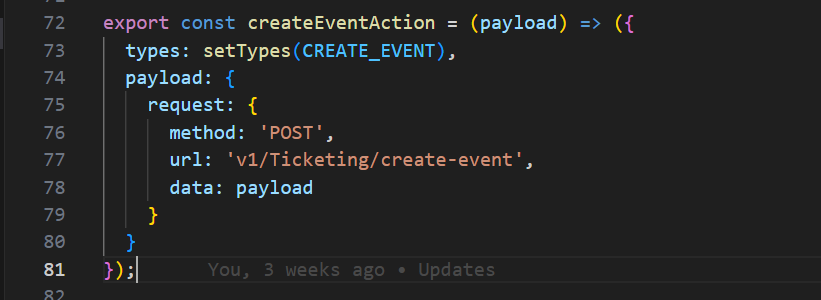
* **Storing Middlewares**: redux has a library createStore to store a middleware when the application starts, the project using this library for attaching “redux reducer “ & “axios” middlewares, when the middleware storing has been setup the next step is to attach a redux “Provider” to the application’s index file to handle the stored middlewares.

Note: you can see it on folder path: ./src/redux-saga/store.js & /src/index.js for your references

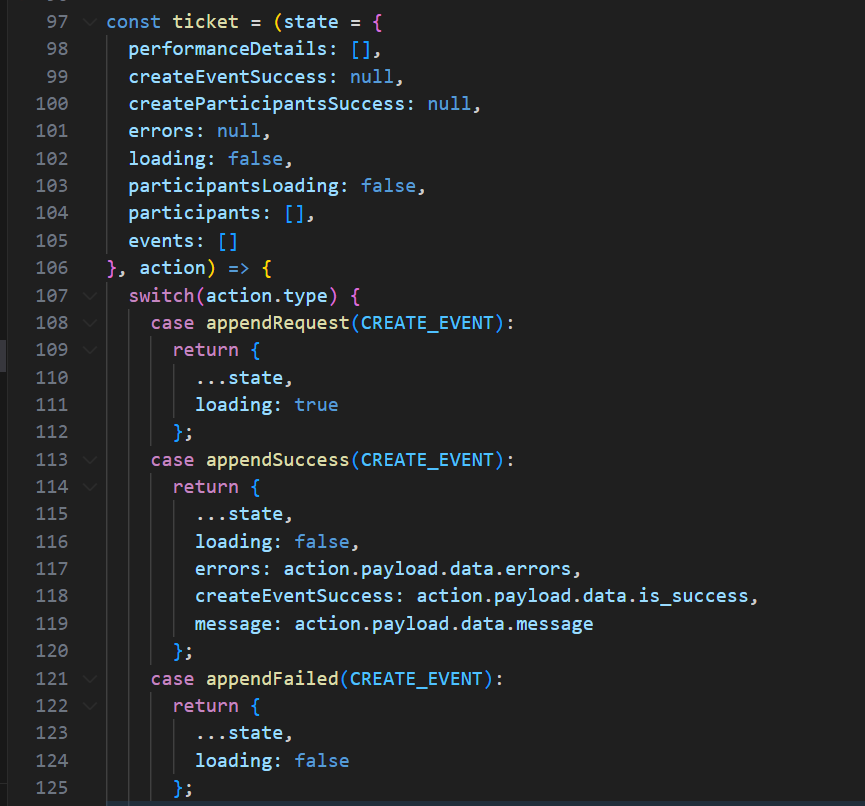
* **Action Types, Actions, Dispatching, Reducers & Sagas:**

Action Type: created action type to specify the action calls.



Action: since the project attached an axios middleware on the application calling actions with types [‘REQUEST’, ‘SUCCESS’, ‘ FAILED’] & payload requests it will handle the process of the API calls via axios middleware when dispatching an action.

Reducer: it will handle all the action responses and store it as global state and can be accessible on all react components.



Dispatching: using useDispatch library on the react component you can call the actions with API call for triggering.

Sagas: used for watching the actions states & runs a function every time when an action is triggered.

**Note:**  Action Types, Actions, Dispatching, Reducers & Sagas can be found in ./src/redux-saga folder for your references.