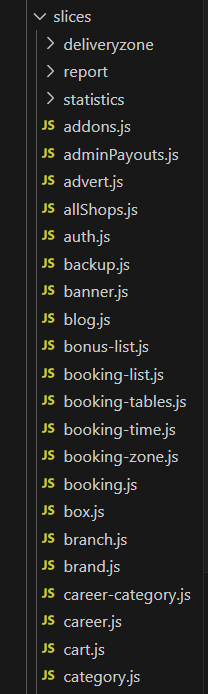
The code uses redux toolkit for the main state management. It has a store. The [rootReducer.js](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ")  file is responsible for combining all the individual slice reducers into a single root reducer, which will be used to create the Redux store. This file imports various slice reducers from different parts of the application and combines them into a single object.

* The file imports a large number of slice reducers from different directories within the slices folder. Each slice reducer is responsible for managing a specific part of the application's state.
* The imported slice reducers are combined into a single object called [rootReducer](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "). This maps each slice reducer to a specific key which represents a part of a state.
* The [rootReducer](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") object is exported as the default export of the file. This root reducer will be used to create the Redux store.

The [store.js](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") file is responsible for configuring and creating the Redux store for the application. It integrates redux-persist to persist parts of the Redux state across sessions. Combines the root reducer with the persisted reducers for [auth](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), [globalSettings](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), [orders](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), [theme](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "), and [todo](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o "). Configuring the Store: Creates the Redux store with the combined persisted reducer and default middleware



In deliveryzone folder we have 4 slices area.js, city country and region.js.

* area.js: it is found in deliveryzone folder. The area slice is a crucial part of the Redux state management for the application. It handles the fetching of area data from the API and updates the state accordingly. By using [createSlice](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") and [createAsyncThunk](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") from @reduxjs/toolkit, the code is simplified and more maintainable. This setup ensures that the state related to delivery zone areas is managed efficiently and consistently throughout the application. areaService.get Makes an API request to fetch area data, merging the initial parameters with any provided parameters.

The same goes for the rest 3 slices(country.js, region.js, city.js) in the deliveryzone folder.

* Category.js: this slice found in reports, is a crucial part of the Redux state management for the application. It handles the fetching of category report data from the API and updates the state accordingly. By using [createSlice](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") and [createAsyncThunk](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") from @reduxjs/toolkit, the code is simplified and more maintainable. This setup ensures that the state related to category reports is managed efficiently and consistently throughout the application.
* extraReducers: A field for handling actions that are not defined in the slice's reducers. In this case, it handles the lifecycle of the [fetchReportProductChart](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") thunk action.
* fetchReportProductChart.pending: Sets [loading](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") to true when the fetch request is initiated.
* fetchReportProductChart.fulfilled: Updates the state with the fetched data when the fetch request is successful.
* fetchReportProductChart.rejected: Sets [loading](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") to false and updates the [error](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") state when the fetch request fails.
* Initial State: Defines the initial state for loading, chart data, product list, and error messages.
* AsynchronousThunkAction: [fetchReportProductChart](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") is an asynchronous action that fetches category report data from the API.
* Slice Definition: Uses [createSlice](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") to define the slice, including the initial state, extra reducers to handle the lifecycle of the [fetchReportProductChart](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") action, and additional reducers for filtering the report product data.
* Exporting the Actions and Reducer: Exports the action creator for the [filterReportProduct](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") reducer and the reducer function generated by [createSlice](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ").

The same goes for all the slices in the reports folder(extras.js, order.js, overview.js, products.js, revenue.js, stock.js).

* count.js: the count slice found in the statistics folder, is responsible for managing the state related to various statistics counts, including those for admins, sellers, deliverymen, and waiters. It includes an initial state, multiple asynchronous thunk actions for fetching different types of statistics, and a slice definition with extra reducers to handle the lifecycle of these fetch actions.   
  The same for the rest of the slices in the statisctics folder.

The other slices are in the root slices folder they are not catagorized as the previous ones and we will see some of these slices.

auth.js: The auth slice in the provided code is responsible for managing the authentication state of the application. It includes an initial state, several reducers to handle different actions related to authentication, and exports the actions and reducer for use in the application.

name: The name of the slice, which will be used as a prefix for generated action types.

initialState: The initial state of the slice.

reducers: An object containing reducer functions to handle different actions related to authentication.

* setUserData: Sets the [user](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") state with the provided payload.
* updateUser: Updates the [user](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") state with the provided payload, merging it with the existing user data and setting the [fullName](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") property.
* clearUser: Clears the [user](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") state, setting it to null.
* setFirebaseToken: Sets the [firebaseToken](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") state with the provided payload.

This slice is example used in the applayout component in the layout folder with a setUserData in line13. it manages user data in the Redux state. The component fetches initial data based on the user's role and sets additional user data, including URLs, for development purposes. The [auth](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") slice, specifically the [setUserData](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") action, is crucial for updating the user data in the Redux state, ensuring that the application has access to the necessary user information.

This example demonstrates how the [auth](vscode-file://vscode-app/c:/Users/ermid/AppData/Local/Programs/Microsoft VS Code/resources/app/out/vs/code/electron-sandbox/workbench/workbench.html" \o ") slice and its actions can be integrated into a React component to handle user-related data and state management efficiently.

