

## React - Assessment 2

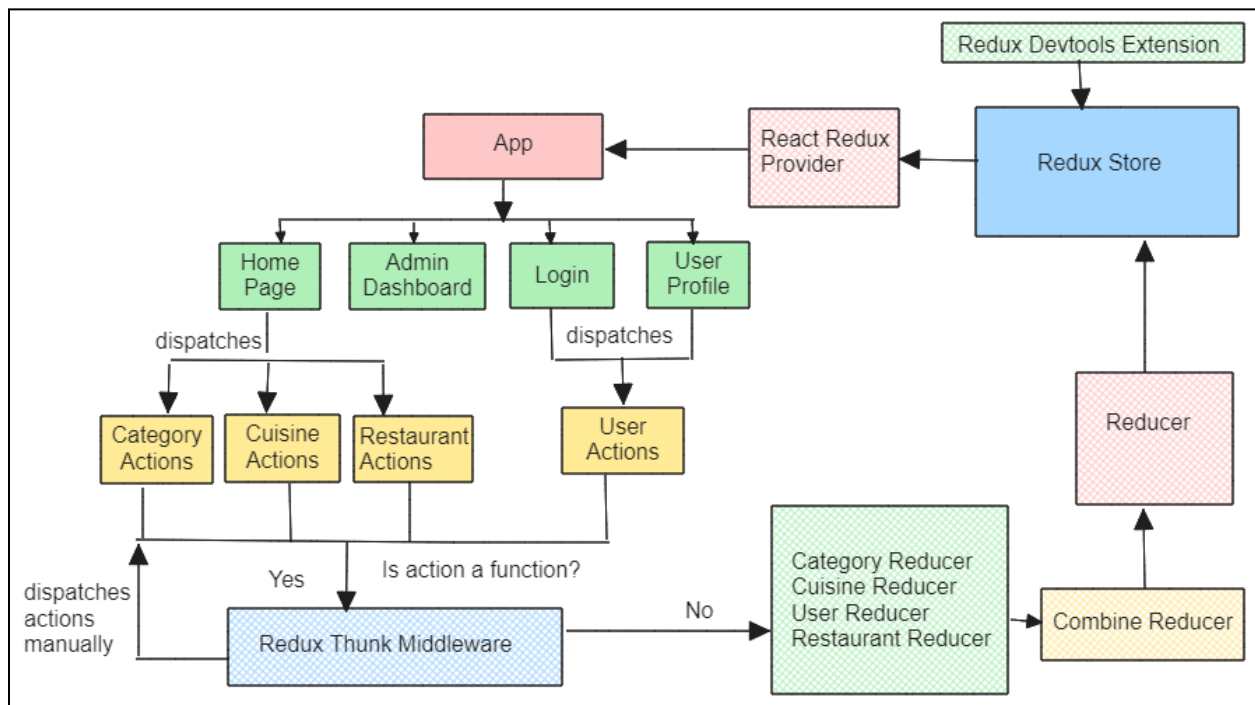
### Introduction

This week you will be building the HomePage in a Food Order App using Redux. The HomePage component is responsible for displaying a list of restaurants, categories, and cuisines available for ordering food. The implementation utilizes React, React Bootstrap, Redux, and React Router.

The API we will be using for fetching categories, cuisines, restaurants, and users data is - <https://foodorder-api-elti.onrender.com/v1/>

You can refer to the **Food Order App API Guideline.pdf** file provided to you to get detailed information about the API.

Let's also have a look at the following block diagram to understand the data flow in the Redux:



Also, refer to the **Redux Thunk pdf** to understand the working of Redux Thunk middleware in the application.

We have provided the starter code for the React app which you will be working on.

## Problem Statement

We have implemented the functionality of displaying restaurant and category details on Homepage and login functionality in the LoginPage. You can refer to the code for actions, constants, dispatch functions, and reducers for implementation of restaurantList and categoryList state.

### constants folder

In Redux, action constants are string values that define the type of action being dispatched. They are typically used to avoid hardcoding action types in multiple places within your application and to provide a centralized and standardized way of referencing actions.

The constants folder has files related to backend API constants, category, cuisines, restaurants, and users action constants.

### storage.js

This file is responsible for creating redux store. Let's understand the working step by step:

1. It first imports all the reducers.
2. The reducers are then combined using the combineReducers() function.
3. Next, we create an initial state for the store.
4. Next, we create middleware using thunk.
5. Finally, we create a store with the reducer, initial state, and middleware.
6. The Redux Dev Tools Extension is a browser extension that allows you to inspect and debug the state and actions of your Redux application in a more user-friendly way.
7. The composeWithDevTools function comes from the redux-devtools-extension package, and it helps you integrate the Redux DevTools Extension into your Redux store's middleware setup.

## Your task is as follows:

**Display Cuisine details on HomePage:** Make use of **Fetch Cuisines API** to display the name and image of all cuisines available on the Home Page.

1. Define the constants, actions, and reducer functions in cuisineActionConstants.js, cuisineActions.js, and cuisineReducers.js files provided.
2. Use the cuisineList state and dispatch actions in HomePage to display the cuisine names and images.

Here is the screenshot of the sample home page for your reference -



**!!! Happy Coding !!!**