# React Week1 - Assessment

## Introduction

This week you will be building the **home page of the user side view and the admin dashboard of the admin side view of the Food Order App** in ReactJS.

* We have provided the starter code for the React app which you will be working on.
* You will be implementing the functionality to display cuisines, categories and restaurants information on the homepage in the user side.
* For the admin side view for this week, we will only focus on creating the Admin Dashboard.
* The Login and Logout functionality are already there.
* The Header, Footer, AlertMessage and ItemCard components are already created and placed in the **components** folder.

To run the React App in the localhost kindly follow the below mentioned steps -

1. Download and extract the **Food\_Order\_App** folder from the **React Week1 Assessment\_For Coders\_Source Code** zip folder.
2. Open a terminal and navigate to the **Food\_Order\_App** folder in it.
3. Run the “*npm install*” command to install the required packages. This will create a new folder named **node\_modules** in the **Food\_Order\_App** folder.
4. Run the “*npm start*” command to run the application with the localhost.
5. To navigate to the admin side of the application, login using the username as “harry1992” and password as “harrypotter1992”. (Refer credentials.js and users.js files)
6. To navigate to the user side of the application, login using the username as “ron1993” and password as “ronweasley1993”. (Refer credentials.js and users.js files)

## Housekeeping points

* This is a minimal example and may not follow some standard practices.
* We focus on the main flow, and not much error handling.

## Problem Statement

Your task is as follows -

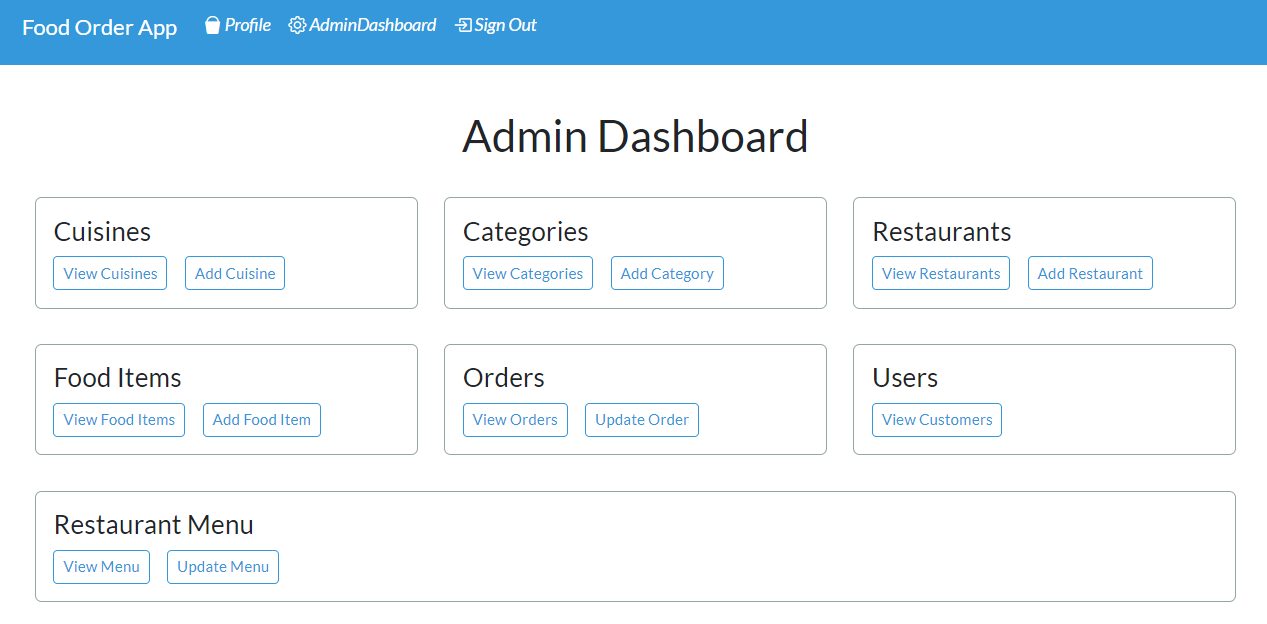
1. **Home Page ( HomePage.js in userview folder)**

* Make use of categories, cuisines and restaurants data provided in the categories.js, cuisines.js, restaurants.js files respectively and display the names along with the images on the homepage.
* You can make use of the **ItemCard(ItemCard.js)** component to display the names and images as Bootstrap cards on the home page.
* Here is the screenshot of the sample home page for your reference -



1. **Admin Dashboard ( AdminDashboard.js file in adminview folder)** 
   * The Admin Dashboard consists of an interface which an admin can use to navigate through all the admin-side pages.
   * It consists of Cuisines, Categories, Food Items, Restaurants, Restauarnt Menu, Orders and Users cards.

* The Admin Dashboard can be created using React Bootstrap components such as Container, Card, Row, Col and Button.
* A card is already added for the Cuisines (for your reference).
* Here’s a screenshot of the sample Admin Dashboard for your reference-

****

## Components Already Implemented

1. **Login Page ( LoginPage.js in pages folder)**

* The Login component consists of a form created using React Bootstrap library for the Login Page where the user can enter the username and password.
* It checks the credentials from the credentials.js file to successfully login the user.
* The users.js file contains information about the admin status of a user.
* If the user is an Admin, the Admin Dashboard is loaded on successful login.
* If the user is not an Admin, the User Home Page is loaded.

1. **Header.js:** The Header component consists of Navbar which is present on all the pages.
2. **Footer.**js: The Footer component consists of copyright which is present on all the pages.
3. **ItemCard.**js: The ItemCard.js consists of a card created from React Bootstrap used to style the categories, cuisines and restaurants.
4. **AlertMessage.**js: The AlertMessage consists of alert box created from React bootstrap used to display both success and error messages on the Login Page.

## Program Organization

* You will be getting a zip folder containing a folder named **React Week1 Assessment\_For Coders**
* The **React Week1 Assessment\_For Coders** folder contains a subfolder named **Food\_Order\_App** containing the required files and folders for the application.
* The favicon is placed in the public folder within the **Food\_Order\_App** folder.
* All the components, pages and data files are placed within the **src** subfolder of the **Food\_Order\_App** folder.
* The admin side pages are present within the **adminview** folder of the pages folder.
* The user side pages are present in the **userview** folder of the pages folder.

## Evaluation Rubric

### Total Project Points: 60

* Correctness:

Correctness of implementation

* Problem statement - point 1 : **30 Points**
* Problem statement - point 2 : **30 Points**

## Program Instructions

* Make sure that you delete the node\_modules folder from the **Food\_Order\_App** folder before submitting it for evaluation.
* Once you have deleted the node\_modules folder then you need to compress the Food\_Order\_App folder in zip/rar format.
* Project will not be evaluated if the submitted project is not in the zip/rar format.