**Assignment\_3 (React Dashboard Application)**

**Mobile Web Applications – MDEV1005**

By: Ronit Sanghvi  
Student Number: 200543133

**Overall Explanation of Project:**

This project is a React Dashboard application that provides various functionalities such as News feed, Currency converter, Country information, and a To-Do list. Users can sign in and access these features once they are logged in. The dashboard includes a header, a side drawer for navigation, and different sections for each functionality.

**API's Used:**

1. News API: Used to fetch news articles from various sources. Two news Api’s are used. One to fetch World Top Headlines and other to fetch Canada Top Headlines.

2. Currency Conversion API: Enter the amount and convert the currency as needed. All currency options are available.

3. Country API: Use this API to get details about the country’s capital, region and population.

**Details about Widgets:**

1. Card Component of MUI: Used in the News section to display individual news articles with images, titles, and descriptions and learn more links.

2. Checklist (To-Do List): Implemented as a widget to manage tasks and to-do items. Add new items and delete item functionality implemented in this checklist.

**Details about Other Implementations:-**

1. Firebase: Used for authentication purposes. Authentication is based on Email and Password. It handles user sign-in and sign-out functionalities. Also shows different errors such as wrong password, email doesn’t exist, wrong format, empty fields.

2. MUI (Material-UI): Utilized for styling and UI components, such as buttons, text fields, card component, typography, font, and the side drawer.

3. AXIOS: Utilized to make HTTP requests, most likely for fetching data from APIs.

4. React-router-dom: Used for client-side routing and to handle different sections of the dashboard and Sign In / Sign Up.

5. React-toastify: Utilized for displaying short notifications and alerts to users.

**Side Drawer Implementation**:-

The side drawer is implemented as a separate component (`Drawer.js`). It provides navigation links to different sections of the dashboard, such as News, Currency Converter, Country Information, and To-Do List.

**Firebase Authentication:-**

Users can log in to their accounts using their email and password. The **signInWithEmailAndPassword** function from Firebase authentication is used to handle the login process. Moreover, users can create new accounts by providing their email and password. The **createUserWithEmailAndPassword** function from Firebase authentication is used to handle the registration process.

**Colors and Design Description**:-

* The project follows a responsive design, making it suitable for both desktop and mobile devices. The design appears with a dark theme (`backgroundColor: #111827`) for this project. The typography uses the "Roboto" font from Google Fonts, and Material-UI components are used for consistent styling.
* The project uses the "Roboto" font family with different font weights (300, 400, 500, 700) to ensure consistent and modern typography throughout the application.
* Short messages of toastify uses Green Color background to show positive or task completion responses and Red color for error responses.
* Several **Gestalt principles** are implicitly employed to enhance the visual and cognitive organization of the user interface.
  + **Proximity**: The menu items in the side drawer are visually grouped together because they are placed close to each other, creating a clear distinction between different sections.
  + **Similarity**: The checklist items have a similar appearance with consistent styling, such as a bullet point, text alignment, and spacing. This similarity helps users recognize that the items belong to the same group.
  + **Symmetry and Balance**: The overall layout of the dashboard follows the principles of symmetry and balance, with a centered header and a symmetrical arrangement of the menu items in the side drawer.

Widgets and API’s Description

* Currency Converter: The `CurrencyConverter` component provides a currency conversion functionality. It fetches currency conversion rates from the "**exchangeratesapi**" and allows the user to enter an amount, select the "From Currency" and "To Currency," and then calculates and displays the converted amount. The converted amount is updated when the user clicks the "Convert" button.
* Checklist: The `Checklist` component allows users to add and manage a To-Do list. Users can add new items to the list using a text field and click the "Add New Data" button. Existing items can be deleted using the delete icon button.
* Country: The `CountryInfo` component displays information about the selected country. It fetches country data from the "**countryapi.io**" based on the selected country and shows details such as the capital, region, and population.

Use of Hooks (useState and useEffect)

1. **useState**:-

useState is used in multiple components for managing state variables. **App.js** and **Dashboard** used this hook to manage the auth User state to keep track of whether the user is logged in or not. **Currency Converter** use this hook to manage the conversion Rates, amount, from Currency, to Currency, and result states, which are essential for the currency conversion functionality. **Drawer** use this to manage the state for toggling the side drawer open and closed. **Country** To manage selected country state. **Checklist** use to manage list of items in To-Do list

1. **useEffect**:

useEffect is used in various components to perform side effects or manage lifecycle events. **Dashboard** Used useEffect to check whether the user is logged in by listening to authentication state changes using onAuthStateChanged. **Currency Converter** uses to fetch currency conversion rates from an API and set them in the state when the component mounts. **News.js** fetch news articles from an API based on the selected category when the component mounts or when the selectedHeadline changes. **Country** uses to fetch data for all countries from an API and then set the relevant information for the selected country in the state when the component mounts or when the selected Country changes.