This assignment is to integrate **BOOTSTRAP, Google font**, **mongoDB** together in **React Js** to build the sales app.

**Front End:**

**Component:**

**Navbar.js:**

In this page, I have created a simple **navbar** showing the **buttons** for add sales, top 5 sales, today’s total revenue, login, register and logout. Navbar is created with the help of **BOOTSTRAP 5** and **bg-danger** class is added to it for styling it. **Link tag** is used to link the pages in the respective buttons. **Ternary operator** to check if the user is logged in then log in and register page should be removed and log out button will be enabled only after the user has logged in.

**Pages:**

**Addsales.js:**

In this page, I have created a simple **form** inside **card with shadow** as styling for getting the input of sales details from the user like product name, quantity and amount then submit button is styled as **danger** to match the styling of the navbar. **Input required** is added to all the fields as a kind of **validation. addSales()** is called when the button is submitted and **protected data** is passed in the route to enable to user to add sales only after log in. **useEffect** is used to check whether the user has logged in. **Loading feature** is added after login button is clicked.

**Topsales.js:**

In this page, I have created a **table using BOOTSTRAP 5** with the styling of **table-danger and table-striped-column** to match the style with the page. This table shows the top 5 sales with their id, product name, quantity, sale amount. **topSales()** is called using **useEffect** to display the top 5 sales of the user.

**Todaysrevenue.js:**

In this page, I have just showed today’s revenue in **text** and it is **aligned center. totalRevenue()** is called using **useEffect** to display the today’s revenue of the user.

**Login.js:**

In this page, I have created a simple **form** inside **card with shadow** as styling for getting the login details of the user like email and password and then submit button is styled as **danger** to match the styling of the navbar. **Input required** is added to all the fields as a kind of **validation.** **login()** is called when the button is submitted. **If condition** is used to display what should happen if the login is success and if the login is failure. After successful login token and user details will be added in the **localstorage.**

**Register.js:**

In this page, I have created a simple **form** inside **card with shadow** as styling for getting the register details of the user like first name, last name, email and password and then submit button is styled as **danger** to match the styling of the navbar. **Input required** is added to all the fields as a kind of **validation. register()** is called when the button is submitted. **If condition** is used to display what should happen if the registration is success and if the user is already registered. After successful registration the user will be **navigated to login page.**

**Logout.js:**

When the user is logged out it displays the message that the user is logged out and the **local storage will be cleared.**

**Back End:**

**Middleware:**

Authorization is done here using **JWT token.**

**SalesModel:**

Here the schema for the sales model is defined salesId, pname, quantity, amount, author fields are defined here.

**UserModel:**

Here the schema for the user model is defined firstname, lastname, email, password fields are defined here.

**SalesRoute:**

API for different routes like addsales, top5sales, todaysrevenue, sales functionality is added here.

**UserRoute:**

API for different routes like login, register functionality is added here.

**Config:**

Links required for implementation is added here like mongoDB and JWT token links.

**Server:**

It is the main file which has all the files imported here and it enables the connection with DB.