**React JS**

Course-End Project Problem Statement



**Course-End Project**

**Creating Redux Expense Management Application**

**Project Agenda:** To create a Redux expense management application for telecom usage optimization using React and Redux with advanced concepts

**Description:**

NetCarrier a telecom company wishes to offer their customers a solution where they often face challenges in managing and optimizing their usage plans. Without real-time insights and personalized recommendations, users may not be aware of more cost-effective plans or may miss opportunities to adjust their plans based on changing usage patterns.

The goal is to create an efficient and user-friendly application that leverages cutting-edge web technologies to enhance the telecom user experience.

**Tools required:** React.js and Node.js

**Expected Deliverables:** This project addresses the challenges by providing an application that analyses historical usage data, offers real-time suggestions, and allows users to dynamically adjust their plans for optimal cost savings.

Steps to be followed:

1. Create and set up the React project
2. Add Bootstrap to the UI
3. Set up project dependencies
4. Execute the backend project
5. Set up the Redux store
6. Create a user authentication
7. Configure the navigation
8. Configure the dashboard
9. Create the usage analysis chart
10. Create the plan recommendation
11. Create the update plan

**Step 1: Create and set up the React project**

1. Open the terminal to run the following command to check that the **node** version is the latest and updated:

**node -v**

A screenshot of a computer

Description automatically generated

**Note**: If the node version is not updated then use the below command to update:

* **sudo npm cache clean -f**
* **sudo npm install -g n**
* **sudo n stable upgrade**

1. Open the terminal to run the following command to check that the **npm** version is latest and up-to-date:

**npm –version**

**A screenshot of a computer

Description automatically generated**

**Note**: If the npm version is not updated then use the below command to update:

**npm install -g npm@latest**

1. Open the terminal to run the following command to create a React app:

**npx create-react-app expense-management-application**

A screenshot of a computer

Description automatically generated

1. Open the terminal to run the following command to evaluate if the application is created and running successfully:

**npm start**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

The output appears as shown below:

A screen shot of a computer

Description automatically generated

**Step 2: Add Bootstrap to the UI**

1. Open the created React app folder (**expense-management-application**) in VS Code by clicking on **File** in the top left corner and selecting **Open Folder**

**A screenshot of a computer

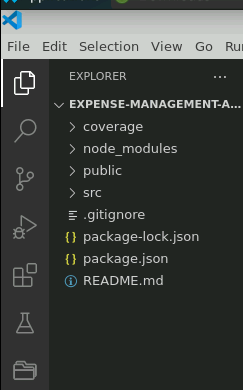
Description automatically generated**

1. Click on **Open**

**A screenshot of a computer

Description automatically generated**

The folder structure appears as shown below:



1. Open the terminal to run the following command to add Bootstrap features:

**npm install react-bootstrap** [**bootstrap@4.6.0**](mailto:bootstrap@4.6.0)

A screenshot of a computer screen

Description automatically generated

**Step 3: Set up project dependencies**

1. Open the **TERMINAL** to run the following command to add dependencies:

**npm install react react-dom react-redux redux redux-thunk axios react-router-dom**

A screenshot of a computer

Description automatically generated

1. Run the following command:

**npm install @reduxjs/toolkit react-redux**

A screen shot of a computer

Description automatically generated

1. Run the following command:

**npm install --save react-chartjs-2 chart.js**

**A screen shot of a computer

Description automatically generated**

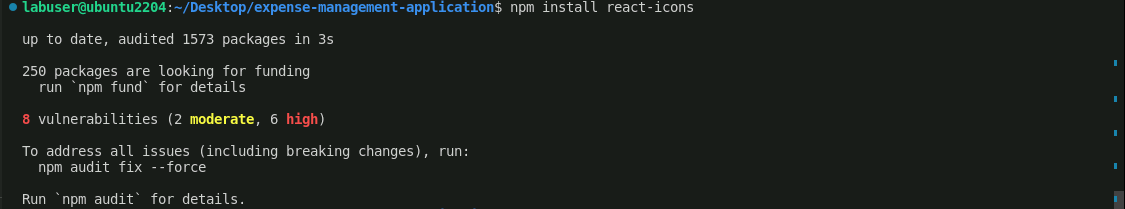
1. Run the following command:

**npm install react-toastify**

**npm install react-icons**

**A screen shot of a computer

Description automatically generated**

****

1. Run the following command to start the application:

**npm start**

**A screenshot of a computer

Description automatically generated**

The output appears as shown below:

A screenshot of a computer

Description automatically generated

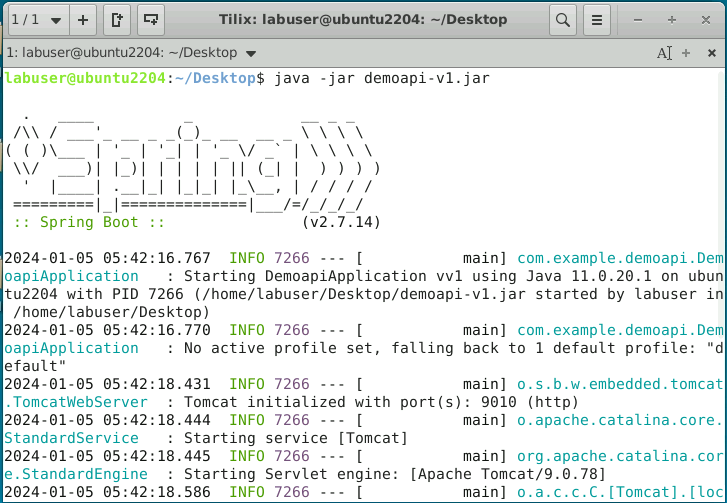
A group of icons with text

Description automatically generated

**Step 4: Execute the backend project**

1. Run the following command to execute the backend application:

**java -jar demoapi-v1.jar**

****

**Step 5: Set up the Redux store**

1. Inside the **src** folder, create subfolders named **actions**, **components**, **hocs**, **reducers**, **services**, and **store**

**A computer screen with a black screen

Description automatically generated**

1. Inside the **reducers** folder, create a file named **rootReducer.js** and enter the below code:

**import { combineReducers } from '@reduxjs/toolkit';**

**import authReducer from '../actions/authSlice';**

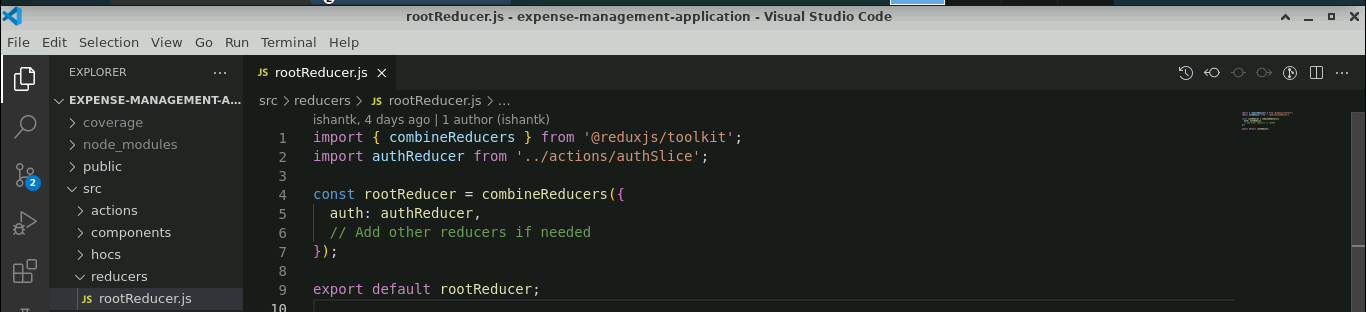
**const rootReducer = combineReducers({**

**auth: authReducer,**

**// Add other reducers if needed**

**});**

**export default rootReducer;**

****

1. Inside the **actions** folder, create a file named **authSlice.js** and enter the below code:

**import { createSlice, createAsyncThunk } from '@reduxjs/toolkit';**

**import { loginApi, updatePlanApi } from '../services/api';**

**// Create an async thunk for the login operation**

**export const loginAsync = createAsyncThunk('auth/login', async ({ email, password }, { rejectWithValue }) => {**

**try {**

**const userData = await loginApi(email, password);**

**return userData;**

**} catch (error) {**

**// Return an error with rejectWithValue**

**return rejectWithValue(error.response.data);**

**}**

**});**

**// Create an async thunk for updating the user's plan**

**export const updatePlanAsync = createAsyncThunk('auth/updatePlan', async ({ email, planId }) => {**

**try {**

**const updatedPlan = await updatePlanApi(email, planId);**

**return updatedPlan;**

**} catch (error) {**

**// Handle errors as needed**

**throw error;**

**}**

**});**

**const authSlice = createSlice({**

**name: 'auth',**

**initialState: {**

**user: null,**

**isAuthenticated: false,**

**loading: false,**

**error: null,**

**},**

**reducers: {**

**logout: (state) => {**

**state.user = null;**

**state.isAuthenticated = false;**

**},**

**},**

**extraReducers: (builder) => {**

**builder**

**.addCase(loginAsync.pending, (state) => {**

**state.loading = true;**

**state.error = null;**

**})**

**.addCase(loginAsync.fulfilled, (state, action) => {**

**state.loading = false;**

**state.isAuthenticated = true;**

**state.user = action.payload;**

**})**

**.addCase(loginAsync.rejected, (state, action) => {**

**state.loading = false;**

**state.error = action.payload; // Error details returned by rejectWithValue**

**})**

**.addCase(updatePlanAsync.fulfilled, (state, action) => {**

**// Update the user's plan in the Redux store**

**state.user = action.payload;**

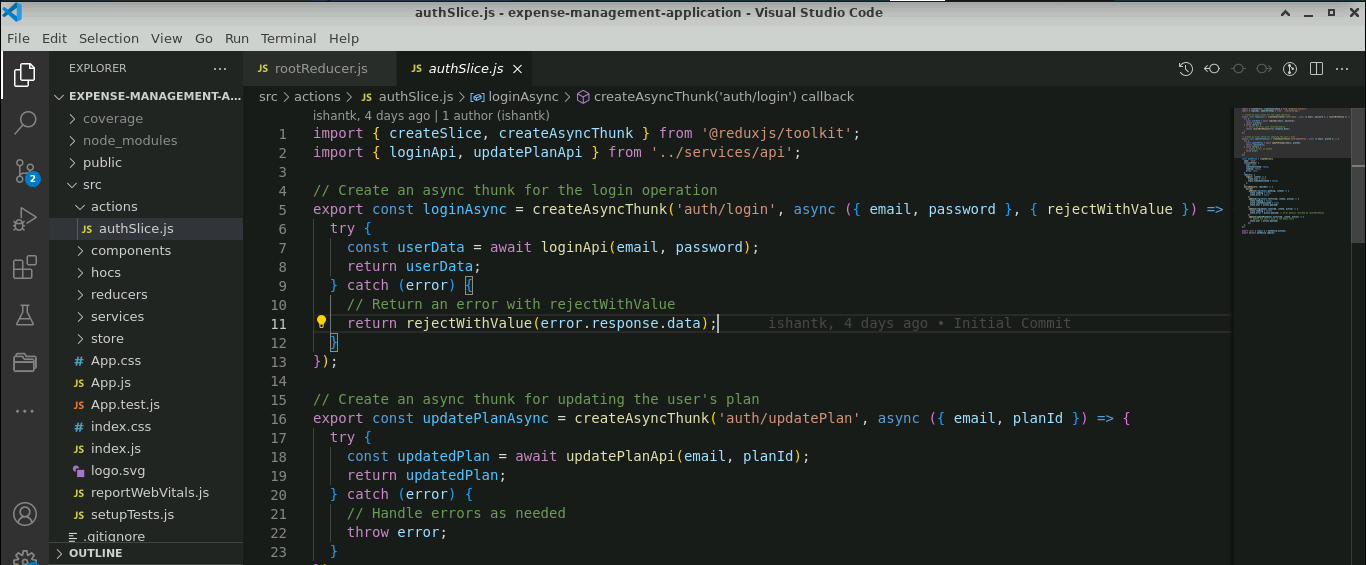
**});**

**},**

**});**

**export const { logout } = authSlice.actions;**

**export default authSlice.reducer;**

****

1. Inside the **store** folder, create a file named **store.js** and enter the below code:

**import { configureStore } from '@reduxjs/toolkit';**

**import rootReducer from '../reducers/rootReducer';**

**const store = configureStore({**

**reducer: rootReducer,**

**// Additional middleware or options if needed**

**});**

**export default store;**

**A screenshot of a computer

Description automatically generated**

1. Navigate to the **index.js** file with the code as shown below:

**import React from 'react';**

**import {createRoot} from 'react-dom/client';**

**import { Provider } from 'react-redux';**

**import store from './store/store';**

**import 'bootstrap/dist/css/bootstrap.min.css';**

**import App from './App';**

**const rootElement = document.getElementById('root');**

**const root = createRoot(rootElement);**

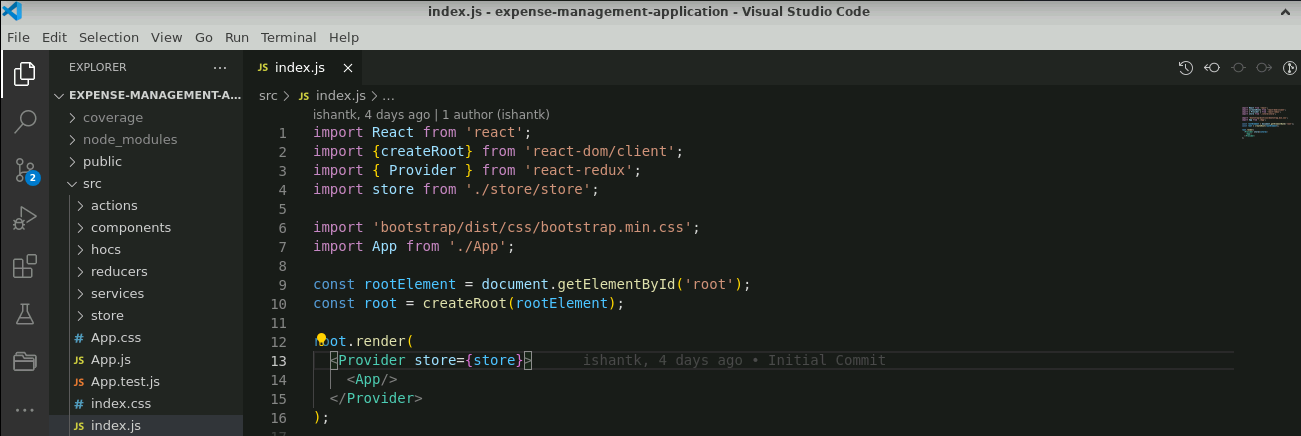
**root.render(**

**<Provider store={store}>**

**<App/>**

**</Provider>**

**);**

****

**Step 6: Create a user authentication**

* 1. Inside the **components** folder, create a file named **LoginPage.js** and enter the below code:

**import React, { useState } from 'react';**

**import { useDispatch, useSelector } from 'react-redux';**

**import { loginAsync } from '../actions/authSlice';**

**import { useNavigate } from 'react-router-dom';**

**import Button from 'react-bootstrap/Button';**

**import Form from 'react-bootstrap/Form';**

**import Container from 'react-bootstrap/Container';**

**const LoginPage = () => {**

**const navigate = useNavigate();**

**const dispatch = useDispatch();**

**const [email, setEmail] = useState('');**

**const [password, setPassword] = useState('');**

**const handleLogin = () => {**

**sessionStorage.setItem('email', email.toString());**

**dispatch(loginAsync({ email, password })).then((response) => {**

**if (response.meta.requestStatus === 'fulfilled') {**

**setTimeout(() => navigate("/home"), 400);**

**}**

**});**

**};**

**return (**

**<div>**

**<Container className="p-3">**

**<Container className="p-3" style={{backgroundColor: '#B4FFC4'}}>**

**<img src='images/login.png' alt='' width={200} height={200}/>**

**<h1 className="header">Welcome To Expense Management Application</h1>**

**<p>**

**Pocket-friendly plans at your fingertips: Revolutionizing telecom expense management for a seamless, cost-effective journey!**

**</p>**

**</Container>**

**<p></p>**

**<h3>Enter Details to Login</h3>**

**<Form>**

**<Form.Group className="mb-3" controlId="formBasicEmail">**

**<Form.Label>Email address</Form.Label>**

**<Form.Control type="email" placeholder="Enter email" value={email} onChange={(e) => setEmail(e.target.value)}/>**

**</Form.Group>**

**<Form.Group className="mb-3" controlId="formBasicPassword">**

**<Form.Label>Password</Form.Label>**

**<Form.Control type="password" placeholder="Password" value={password} onChange={(e) => setPassword(e.target.value)}/>**

**</Form.Group>**

**<Button variant="primary" onClick={handleLogin} style={{background: '#017D69'}}>**

**Login**

**</Button>**

**</Form>**

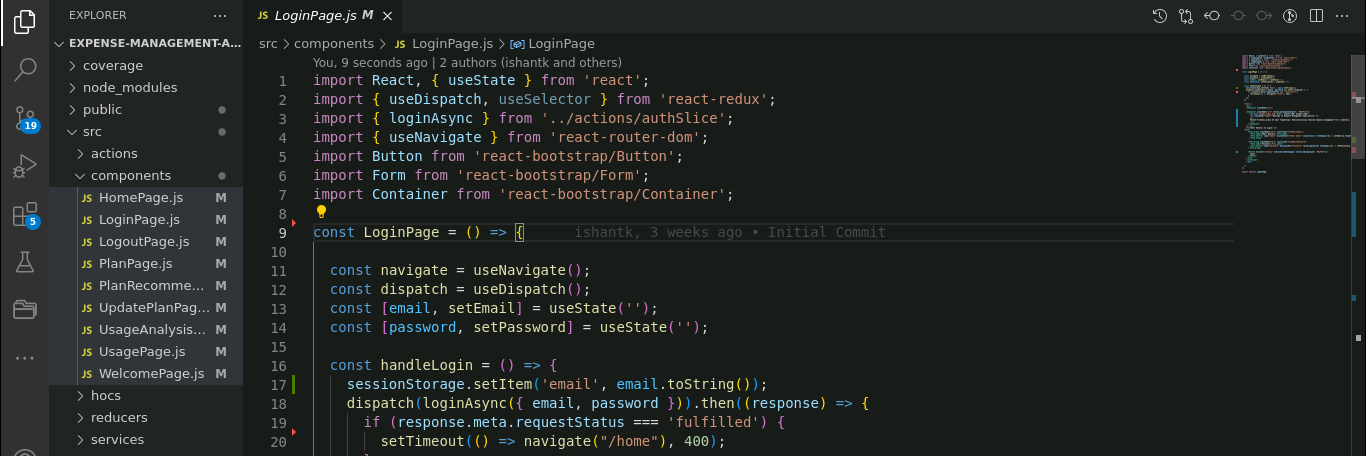
**</Container>**

**</div>**

**);**

**};**

**export default LoginPage;**

****

* 1. Create a file named **HomePage.js** and enter the below code:

**import React from 'react';**

**import { useSelector } from 'react-redux';**

**import Card from 'react-bootstrap/Card';**

**import Container from 'react-bootstrap/Container';**

**import { Link } from 'react-router-dom';**

**import Badge from 'react-bootstrap/Badge';**

**import { FaIdCard, FaCrown, FaRegAddressCard, FaChartBar, FaUser, FaCog, FaLightbulb, FaSignOutAlt, FaRegMoneyBillAlt } from 'react-icons/fa';**

**const HomePage = () => {**

**const icons = [**

**<FaRegAddressCard />,**

**<FaChartBar />,**

**<FaUser />,**

**<FaCog />,**

**<FaLightbulb />,**

**<FaSignOutAlt />,**

**];**

**const isAuthenticated = useSelector((state) => state.auth.isAuthenticated);**

**const userData = useSelector((state) => state.auth.user);**

**console.log("Authentication:"+isAuthenticated);**

**console.log("User Data:"+JSON.stringify(userData));**

**return (**

**<div>**

**{isAuthenticated ? (**

**<Container className="p-3">**

**<Container className="p-3" style={{backgroundColor: '#8CDBFF'}}>**

**<img src='images/expense-image.png' alt='' width={300} height={200}/>**

**<h1 className="header">Welcome To Expense Management Application</h1>**

**<br/>**

**<h4>Welcome, {userData.user.name}!</h4>**

**</Container>**

**<Container className="p-3">**

**<Card>**

**<Card.Body>**

**<Card.Link>{icons[0]} <Link to="/plan">My Plan</Link></Card.Link>**

**<Card.Link>{icons[1]} <Link to="/usage">Usage</Link></Card.Link>**

**<Card.Link>{icons[2]} <Link to="/usage-analysis">Usage Analysis</Link></Card.Link>**

**<Card.Link>{icons[3]} <Link to="/recommendation">Plan Recommendation</Link></Card.Link>**

**<Card.Link>{icons[4]} <Link to="/update-plan">Update Plan</Link></Card.Link>**

**<Card.Link>{icons[5]} <Link to="/logout">LogOut</Link></Card.Link>**

**</Card.Body>**

**</Card>**

**</Container>**

**<Container style={{backgroundColor: '#8CDBFF'}} className="p-3">**

**<Card className="shadow-lg rounded">**

**<Card.Body className="text-center">**

**<div className="mb-4">**

**<FaUser className="display-4 text-primary" />**

**</div>**

**<Card.Title className="h4 mb-3">{userData.user.email}</Card.Title>**

**<Card.Subtitle className="mb-2 text-muted">**

**<FaIdCard className="mr-2" /> Plan ID: {userData.user.plan.planId}**

**</Card.Subtitle>**

**<Card.Text className="lead">**

**<Badge variant="success" className="mr-2">**

**<FaCrown className="mr-1" />**

**{userData.user.plan.planName}**

**</Badge>**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2 text-info" /> Validity: </strong> {userData.user.plan.contractLengthMonths} (Months)**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2 text-success" /> Monthly Cost:</strong> ₹{userData.user.plan.monthlyCost}**

**</Card.Text>**

**</Card.Body>**

**</Card>**

**</Container>**

**</Container>**

**) : (**

**<p>Please Login First</p>**

**)}**

**</div>**

**);**

**};**

**export default HomePage;**

**A screenshot of a computer

Description automatically generated**

* 1. Create a file named **WelcomePage.js** and enter the below code:

**import React from 'react';**

**import Button from 'react-bootstrap/Button';**

**import Container from 'react-bootstrap/Container';**

**import Card from 'react-bootstrap/Card';**

**import Carousel from 'react-bootstrap/Carousel';**

**import { useNavigate } from 'react-router-dom';**

**const WelcomePage = () => {**

**const navigate = useNavigate();**

**const handleButtonClick = () => {**

**navigate('/login');**

**};**

**return (**

**<div>**

**<Container className="p-3">**

**<Container className="p-3" style={{backgroundColor: '#8CDBFF'}}>**

**<img src='images/expense-image.png' alt='' width={300} height={200}/>**

**<h1 className="header">Welcome To Expense Management Application</h1>**

**<p>**

**Pocket-friendly plans at your fingertips: Revolutionizing telecom expense management for a seamless, cost-effective journey!**

**</p>**

**<p>**

**<Button variant="primary" onClick={handleButtonClick}>Click Here to Login</Button>**

**</p>**

**</Container>**

**<Container className="p-3">**

**<div class="row">**

**<div class="col-md-4 mb-4">**

**<div class="card">**

**<div class="card-body">**

**<img className="a-block w-100" src="images/img1.jpg" alt="First slide" />**

**<p></p>**

**<h5 class="card-title">User Authentication</h5>**

**<p class="card-text">Allow users to log in and authenticate to access personalized usage data and recommendations.</p>**

**<i class="fas fa-user-lock fa-3x"></i>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-4 mb-4">**

**<div class="card">**

**<div class="card-body">**

**<img className="a-block w-100" src="images/img2.jpg" alt="First slide" />**

**<p></p>**

**<h5 class="card-title">Navigation</h5>**

**<p class="card-text">Smooth navigation between different views, ensuring a seamless user experience.</p>**

**<i class="fas fa-directions fa-3x"></i>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-4 mb-4">**

**<div class="card">**

**<div class="card-body">**

**<img className="a-block w-100" src="images/img3.jpg" alt="First slide" />**

**<p></p>**

**<h5 class="card-title">Dashboard Overview</h5>**

**<p class="card-text">Comprehensive dashboard providing an overview of historical usage data and current plan details.</p>**

**<i class="fas fa-chart-line fa-3x"></i>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-4 mb-4">**

**<div class="card">**

**<div class="card-body">**

**<img className="a-block w-100" src="images/img4.jpg" alt="First slide" />**

**<p></p>**

**<h5 class="card-title">Usage Analysis Charts</h5>**

**<p class="card-text">Interactive charts and graphs for visualizing historical usage data, helping users understand their usage patterns.</p>**

**<i class="fas fa-chart-bar fa-3x"></i>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-4 mb-4">**

**<div class="card">**

**<div class="card-body">**

**<img className="a-block w-100" src="images/img5.jpg" alt="First slide" />**

**<p></p>**

**<h5 class="card-title">Recommendations Section</h5>**

**<p class="card-text">Dedicated section for personalized recommendations based on historical data analysis.</p>**

**<i class="fas fa-lightbulb fa-3x"></i>**

**</div>**

**</div>**

**</div>**

**<div class="col-md-4 mb-4">**

**<div class="card">**

**<div class="card-body">**

**<img className="a-block w-100" src="images/img6.jpg" alt="First slide" />**

**<p></p>**

**<h5 class="card-title">Plan Adjustment Interface</h5>**

**<p class="card-text">Enable users to adjust their telecom plans directly from the dashboard, providing instant plan customization.</p>**

**<i class="fas fa-cogs fa-3x"></i>**

**</div>**

**</div>**

**</div>**

**</div>**

**</Container>**

**</Container>**

**</div>**

**);**

**};**

**export default WelcomePage;**

**A screen shot of a computer

Description automatically generated**

* 1. Inside the **services** folder, create a file named **api.js** and enter the below code:

**import axios from 'axios';**

**const instance = axios.create({**

**baseURL: 'http://localhost:9010', // API base URL**

**timeout: 5000, // Reasonable timeout**

**});**

**export const loginApi = async (email, password) => {**

**try {**

**const response = await instance.post('/login', { email, password });**

**return response.data; // Assuming the API returns user data upon successful login**

**} catch (error) {**

**throw error; // Handle errors appropriately in your application**

**}**

**};**

**export const updatePlanApi = async (email, planId) => {**

**try {**

**const response = await instance.post('/update-user-plan', { email, planId });**

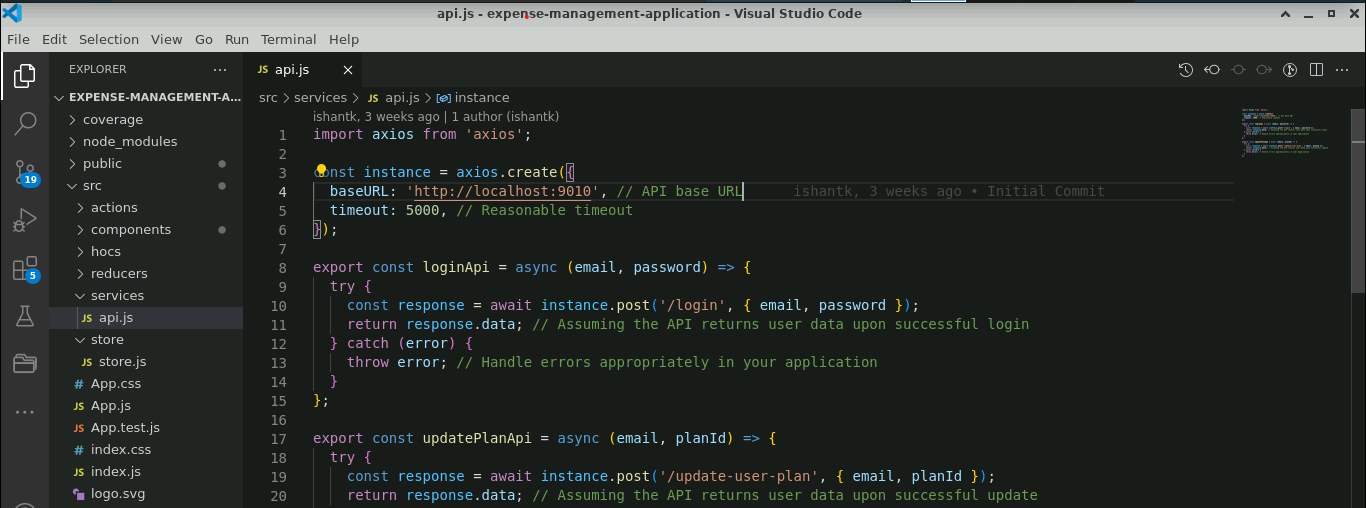
**return response.data; // Assuming the API returns user data upon successful update**

**} catch (error) {**

**throw error; // Handle errors appropriately in your application**

**}**

**};**

****

* 1. Create a file named **App.js** to configure the router for login and home page and enter the below code:

**import React from 'react';**

**import { BrowserRouter as Router , Routes, Route} from 'react-router-dom'**

**import { useSelector } from 'react-redux';**

**import WelcomePage from './components/WelcomePage';**

**import LoginPage from './components/LoginPage';**

**import HomePage from './components/HomePage';**

**import PlanPage from './components/PlanPage';**

**import UsagePage from './components/UsagePage';**

**import UsageAnalysisChartsPage from './components/UsageAnalysisChartsPage';**

**import UpdatePlanPage from './components/UpdatePlanPage';**

**import PlanRecommendationPage from './components/PlanRecommendationPage';**

**import LogoutPage from './components/LogoutPage';**

**const App = () => {**

**const isAuthenticated = useSelector((state) => state.auth.isAuthenticated);**

**const userData = useSelector((state) => state.auth.userData);**

**console.log("Authentication:"+isAuthenticated);**

**console.log("User Data:"+JSON.stringify(userData));**

**return (**

**<Router>**

**<Routes>**

**<Route path="/" element={<WelcomePage/>} />**

**<Route path="/login" element={<LoginPage/>} />**

**<Route path="/logout" element={<LogoutPage/>} />**

**<Route path="/home" element={<HomePage/>} />**

**<Route path="/plan" element={<PlanPage/>} />**

**<Route path="/usage" element={<UsagePage/>} />**

**<Route path="/usage-analysis" element={<UsageAnalysisChartsPage/>} />**

**<Route path="/recommendation" element={<PlanRecommendationPage/>} />**

**<Route path="/update-plan" element={<UpdatePlanPage/>} />**

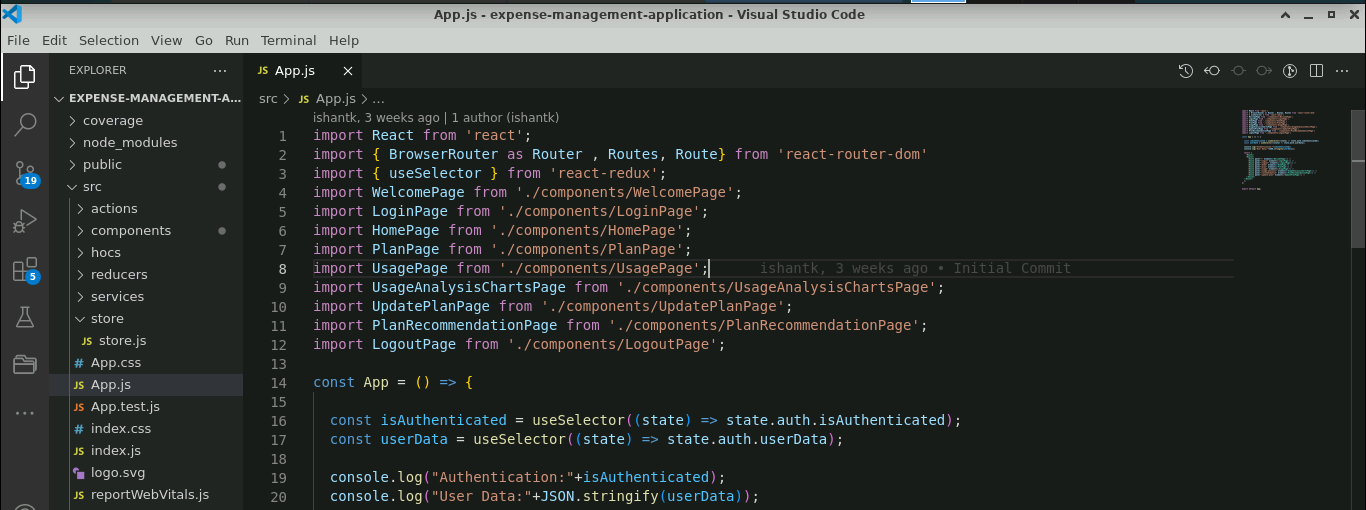
**</Routes>**

**</Router>**

**);**

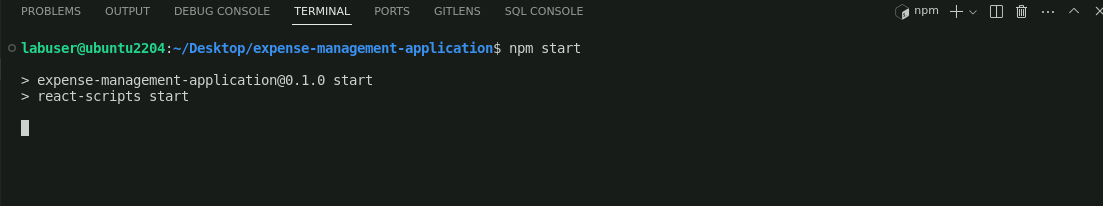
**};**

**export default App;**

****

* 1. Open the **TERMINAL** and run the below command to check the login functionality of the application:

**npm start**



The output appears as shown below:

A screenshot of a computer

Description automatically generated

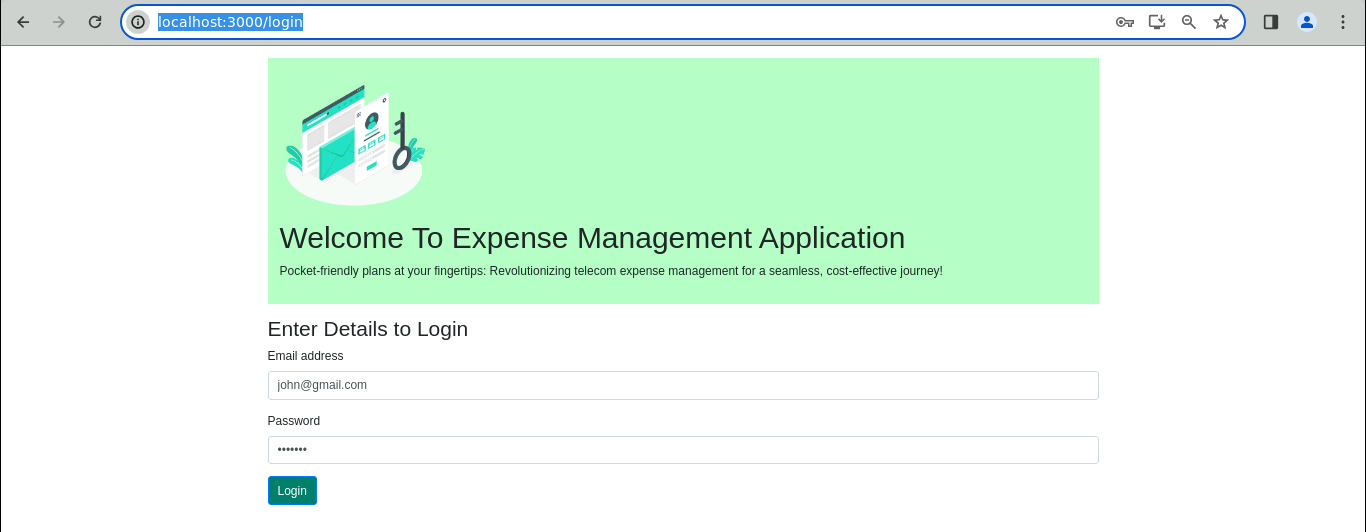
A group of icons with text

Description automatically generated

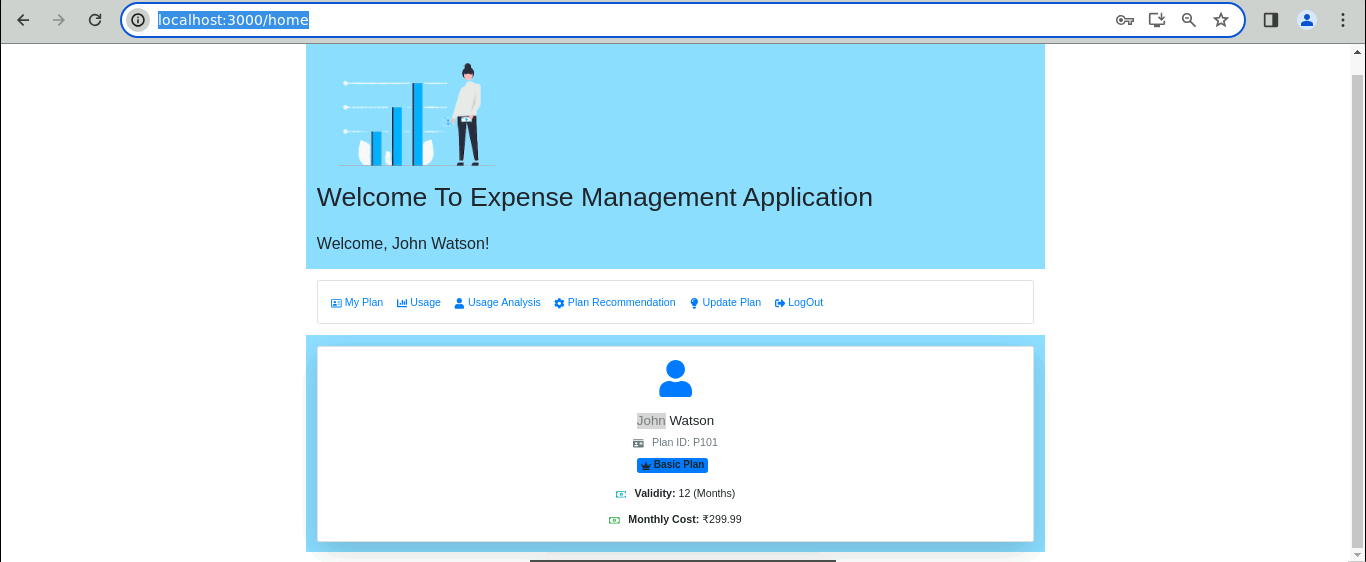
* 1. Enter the credentials to the login page and click on the **Login** button as shown below:

**Email: john@example.com**

**Password: john123**

****

The user will be directed to the home page as shown below:



**Step 7: Configure the navigation**

1. Inside the **App.js** configure the router for navigation and enter the below code:

**import React from 'react';**

**import { BrowserRouter as Router , Routes, Route} from 'react-router-dom'**

**import { useSelector } from 'react-redux';**

**import WelcomePage from './components/WelcomePage';**

**import LoginPage from './components/LoginPage';**

**import HomePage from './components/HomePage';**

**import PlanPage from './components/PlanPage';**

**import UsagePage from './components/UsagePage';**

**import UsageAnalysisChartsPage from './components/UsageAnalysisChartsPage';**

**import UpdatePlanPage from './components/UpdatePlanPage';**

**import PlanRecommendationPage from './components/PlanRecommendationPage';**

**import LogoutPage from './components/LogoutPage';**

**const App = () => {**

**const isAuthenticated = useSelector((state) => state.auth.isAuthenticated);**

**const userData = useSelector((state) => state.auth.userData);**

**console.log("Authentication:"+isAuthenticated);**

**console.log("User Data:"+JSON.stringify(userData));**

**return (**

**<Router>**

**<Routes>**

**<Route path="/" element={<WelcomePage/>} />**

**<Route path="/login" element={<LoginPage/>} />**

**<Route path="/logout" element={<LogoutPage/>} />**

**<Route path="/home" element={<HomePage/>} />**

**<Route path="/plan" element={<PlanPage/>} />**

**<Route path="/usage" element={<UsagePage/>} />**

**<Route path="/usage-analysis" element={<UsageAnalysisChartsPage/>} />**

**<Route path="/recommendation" element={<PlanRecommendationPage/>} />**

**<Route path="/update-plan" element={<UpdatePlanPage/>} />**

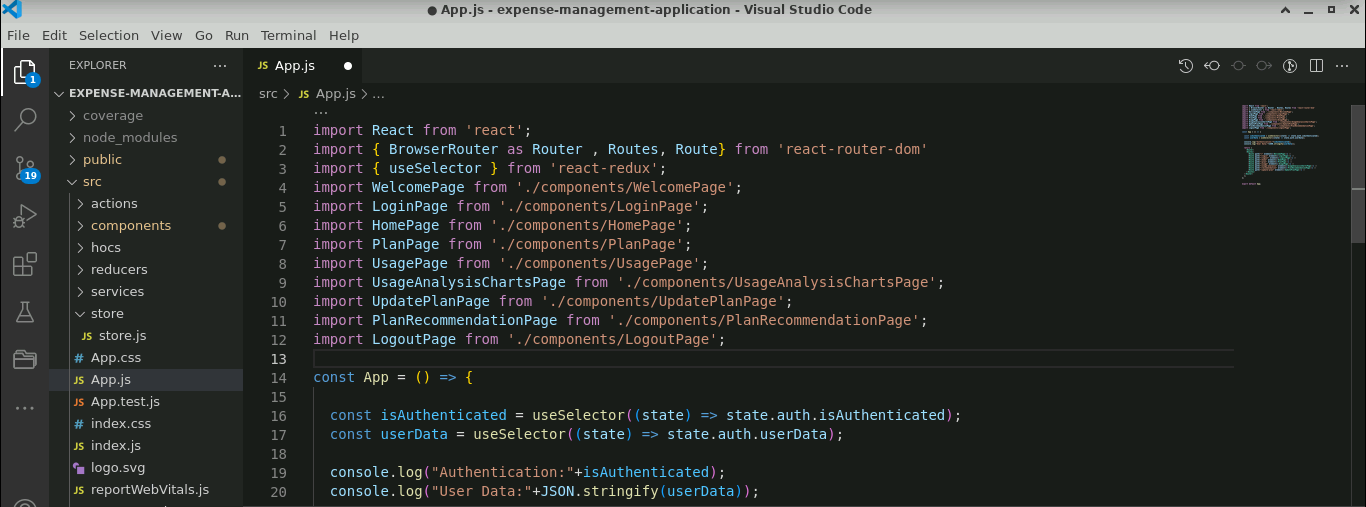
**</Routes>**

**</Router>**

**);**

**};**

**export default App;**

****

**Step 8: Configure the dashboard**

* 1. Inside the **hocs** folder, create a file named **withDataFetching.js** and enter the below code:

**import React, { Component } from 'react';**

**const withDataFetching = (url) => (WrappedComponent) => {**

**return class extends Component {**

**state = {**

**data: null,**

**loading: true,**

**error: null,**

**};**

**async componentDidMount() {**

**try {**

**const response = await fetch(url);**

**const data = await response.json();**

**this.setState({**

**data,**

**loading: false,**

**});**

**} catch (error) {**

**this.setState({**

**loading: false,**

**error: 'Error fetching data',**

**});**

**}**

**}**

**render() {**

**return <WrappedComponent {...this.props} {...this.state} />;**

**}**

**};**

**};**

**export default withDataFetching;**

**A screen shot of a computer

Description automatically generated**

* 1. Inside the **Component** folder, create a file named **PlanPage.js** and enter the below code:

**import React from 'react';**

**import { useSelector } from 'react-redux';**

**import Card from 'react-bootstrap/Card';**

**import Container from 'react-bootstrap/Container';**

**import { Link } from 'react-router-dom';**

**import { FaUser, FaRegIdCard, FaRegEnvelope, FaRegHdd, FaRegClock, FaRegComment, FaRegMoneyBillAlt } from 'react-icons/fa';**

**import { FaIdCard, FaCrown, FaRegAddressCard, FaChartBar, FaCog, FaLightbulb, FaSignOutAlt, FaHome } from 'react-icons/fa';**

**import { Badge } from 'react-bootstrap';**

**const PlanPage = () => {**

**const icons = [**

**<FaRegAddressCard />,**

**<FaChartBar />,**

**<FaUser />,**

**<FaCog />,**

**<FaLightbulb />,**

**<FaSignOutAlt />,**

**<FaHome />,**

**];**

**const isAuthenticated = useSelector((state) => state.auth.isAuthenticated);**

**const userData = useSelector((state) => state.auth.user);**

**console.log("isAuthenticated:"+isAuthenticated);**

**console.log("userData:"+userData);**

**return (**

**<div>**

**{isAuthenticated ? (**

**<Container className="p-3">**

**<Container className="p-3" style={{backgroundColor: '#8CDBFF'}}>**

**<img src='images/expense-image.png' alt='' width={300} height={200}/>**

**<h1 className="header">Welcome To Expense Management Application</h1>**

**<br/>**

**<h4>Telecom Plan For: {userData.user.name}!</h4>**

**</Container>**

**<Container className="p-3">**

**<Card>**

**<Card.Body>**

**<Card.Link>{icons[6]} <Link to="/home">Home</Link></Card.Link>**

**<Card.Link>{icons[1]} <Link to="/usage">Usage</Link></Card.Link>**

**<Card.Link>{icons[2]} <Link to="/usage-analysis">Usage Analysis</Link></Card.Link>**

**<Card.Link>{icons[3]} <Link to="/recommendation">Plan Recommendation</Link></Card.Link>**

**<Card.Link>{icons[4]} <Link to="/update-plan">Update Plan</Link></Card.Link>**

**<Card.Link>{icons[5]} <Link to="/logout">LogOut</Link></Card.Link>**

**</Card.Body>**

**</Card>**

**</Container>**

**<Container className="p-3">**

**<Container style={{backgroundColor: '#8CDBFF'}} className="p-3">**

**<Card className="shadow-lg rounded">**

**<Card.Body className="text-center">**

**<Card.Title>Plan Details</Card.Title>**

**<div className="mb-4">**

**<FaRegHdd className="display-4 text-primary" />**

**</div>**

**<Card.Title className="h4 mb-3">{icons[2]} {userData.user.email}</Card.Title>**

**<Card.Subtitle className="mb-2 text-muted">**

**<FaIdCard className="mr-2" /> Plan ID: {userData.user.plan.planId}**

**</Card.Subtitle>**

**<Card.Text className="lead">**

**<Badge variant="success" className="mr-2">**

**<FaCrown className="mr-1" />**

**{userData.user.plan.planName}**

**</Badge>**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2" /> Validity: </strong> {userData.user.plan.contractLengthMonths} (Months)**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2" /> Monthly Cost:</strong> ₹{userData.user.plan.monthlyCost}**

**</Card.Text>**

**</Card.Body>**

**</Card>**

**</Container>**

**<p />**

**<Container style={{backgroundColor: '#B4FFC4'}} className="p-3">**

**<Card className="shadow-lg rounded">**

**<Card.Body>**

**<Card.Title className="h2 text-primary mb-4"></Card.Title>**

**<Container style={{ backgroundColor: '#EEFFF1' }} className="p-3">**

**<div className="mb-4">**

**<FaRegMoneyBillAlt className="display-4 text-primary" />**

**</div>**

**<p>Basic Details:</p>**

**<Card.Text>**

**<strong><FaRegIdCard className="mr-2" /> Plan ID:</strong> {userData.user.plan.planId}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegEnvelope className="mr-2" /> Plan Name:</strong> {userData.user.plan.planName}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegHdd className="mr-2" /> Data Allowance:</strong> {userData.user.plan.dataAllowanceGB} GB**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegClock className="mr-2" /> Minutes:</strong> {userData.user.plan.minutes}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegComment className="mr-2" /> Text Messages:</strong> {userData.user.plan.textMessages}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2" /> Validity: </strong> {userData.user.plan.contractLengthMonths} (Months)**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2" /> Monthly Cost:</strong> ₹{userData.user.plan.monthlyCost}**

**</Card.Text>**

**</Container>**

**<br />**

**<Container style={{ backgroundColor: '#D8F3FF' }} className="p-3">**

**<div className="mb-4">**

**<FaRegEnvelope className="display-4 text-primary" />**

**</div>**

**<p>Extended Details:</p>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2" /> Additional Fees:</strong> ₹{userData.user.plan.additionalFees}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegHdd className="mr-2" /> Data Speed:</strong> {userData.user.plan.dataSpeed}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegEnvelope className="mr-2" /> Network Coverage:</strong> {userData.user.plan.networkCoverage}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegComment className="mr-2" /> Family Plan Options:</strong> {userData.user.plan.familyPlanOptions ? 'Yes' : 'No'}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegComment className="mr-2" /> Device Subsidies:</strong> {userData.user.plan.deviceSubsidies ? 'Yes' : 'No'}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2" /> Streaming Services:</strong> {userData.user.plan.streamingServices}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegComment className="mr-2" /> Usage Alerts:</strong> {userData.user.plan.usageAlerts ? 'Yes' : 'No'}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegComment className="mr-2" /> Early Termination Terms:</strong> {userData.user.plan.earlyTerminationTerms}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2" /> Discounts:</strong> ₹{userData.user.plan.discounts}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegEnvelope className="mr-2" /> Customer Support:</strong> {userData.user.plan.customerSupport}**

**</Card.Text>**

**</Container>**

**</Card.Body>**

**</Card>**

**</Container>**

**</Container>**

**</Container>**

**) : (**

**<p>Please Login First</p>**

**)}**

**</div>**

**);**

**};**

**export default PlanPage;**

**A screenshot of a computer program

Description automatically generated**

* 1. Click on **My Plan**

A screenshot of a computer

Description automatically generated

The output appears as shown below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

* 1. Inside the **Component** folder, create a file named **UsagePage.js** and enter the below code:

**import React from 'react';**

**import { useSelector } from 'react-redux';**

**import withDataFetching from '../hocs/withDataFetching';**

**import Card from 'react-bootstrap/Card';**

**import Container from 'react-bootstrap/Container';**

**import { Link } from 'react-router-dom';**

**import Table from 'react-bootstrap/Table';**

**import { FaRegAddressCard, FaChartBar, FaCog, FaLightbulb, FaSignOutAlt} from 'react-icons/fa';**

**import { FaCalendarAlt, FaDatabase, FaPhone, FaClock, FaHome, FaUser } from 'react-icons/fa';**

**const UsagePage = ({ data, loading, error }) => {**

**const icons = [**

**<FaRegAddressCard />,**

**<FaChartBar />,**

**<FaUser />,**

**<FaCog />,**

**<FaLightbulb />,**

**<FaSignOutAlt />,**

**<FaHome />,**

**];**

**const isAuthenticated = useSelector((state) => state.auth.isAuthenticated);**

**const userData = useSelector((state) => state.auth.user);**

**let email = userData.email;**

**console.log("Email:"+email);**

**console.log("Data:"+JSON.stringify(data));**

**if (loading) {**

**return <p>Loading...</p>;**

**}**

**if (error) {**

**return <p>Error: {error}</p>;**

**}**

**const telecomData = data.telecomPlanUsage;**

**return (**

**<div>**

**{isAuthenticated ? (**

**<Container className="p-3">**

**<Container className="p-3" style={{backgroundColor: '#8CDBFF'}}>**

**<img src='images/expense-image.png' alt='' width={300} height={200}/>**

**<h1 className="header">Welcome To Expense Management Application</h1>**

**<br/>**

**<h4 className="mb-4">Telecom Plan Usage for {userData.user.name}</h4>**

**</Container>**

**<Container className="p-3">**

**<Card>**

**<Card.Body>**

**<Card.Link>{icons[6]} <Link to="/home">Home</Link></Card.Link>**

**<Card.Link>{icons[0]} <Link to="/plan">My Plan</Link></Card.Link>**

**<Card.Link>{icons[2]} <Link to="/usage-analysis">Usage Analysis</Link></Card.Link>**

**<Card.Link>{icons[3]} <Link to="/recommendation">Plan Recommendation</Link></Card.Link>**

**<Card.Link>{icons[4]} <Link to="/update-plan">Update Plan</Link></Card.Link>**

**<Card.Link>{icons[5]} <Link to="/logout">LogOut</Link></Card.Link>**

**</Card.Body>**

**</Card>**

**</Container>**

**<br/><br/>**

**<div>**

**<Table striped bordered hover responsive>**

**<thead className="thead-dark">**

**<tr>**

**<th><FaCalendarAlt /> Date</th>**

**<th><FaDatabase /> Data Used</th>**

**<th><FaClock /> Minutes Used</th>**

**<th><FaPhone /> Text Messages Sent</th>**

**</tr>**

**</thead>**

**<tbody>**

**{telecomData.map((usage, index) => (**

**<tr key={index}>**

**<td>{usage.date}</td>**

**<td style={{ color: usage.dataUsed > 1024 ? 'red' : 'inherit' }}>{usage.dataUsed}</td>**

**<td>{usage.minutesUsed}</td>**

**<td>{usage.textMessagesSent}</td>**

**</tr>**

**))}**

**</tbody>**

**</Table>**

**</div>**

**</Container> ) : (**

**<p>Please Login First</p>**

**)}**

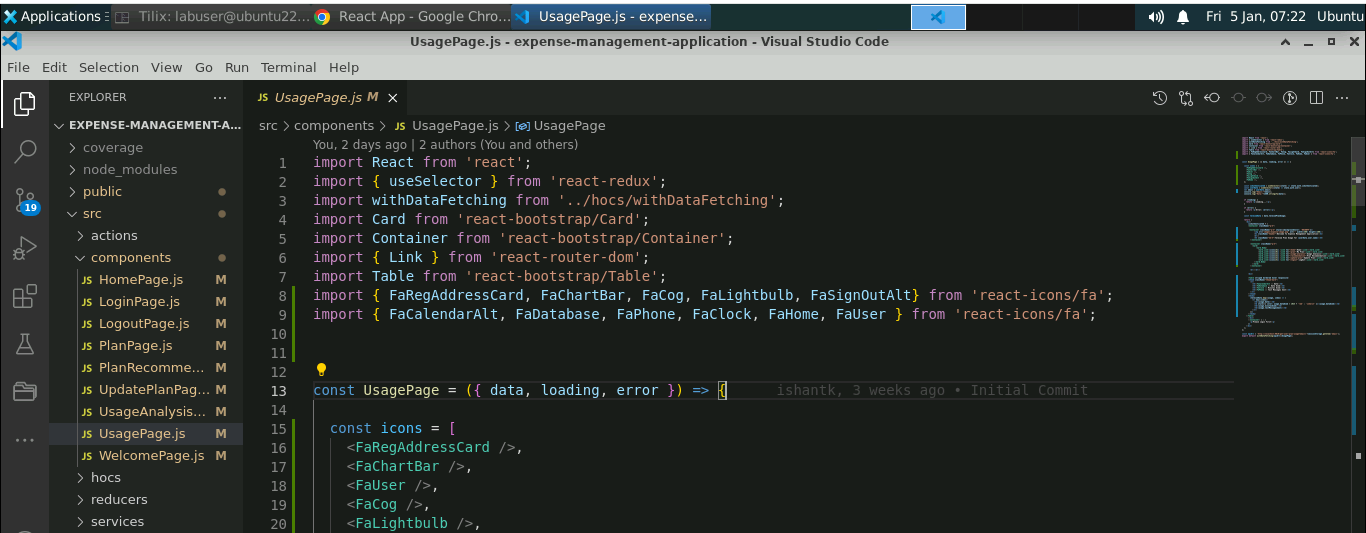
**</div>**

**);**

**};**

**const apiUrl = 'http://localhost:9010/get-user-plan-usage?email='+sessionStorage.getItem('email');**

**export default withDataFetching(apiUrl)(UsagePage);**

****

* 1. Click on the **Usage**

A screenshot of a computer

Description automatically generated

The output appears as shown below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Step 9: Create the usage analysis chart**

* 1. Inside the **components** folder, create a file named **UsageAnalysisChartsPage.js** and enter the code as shown below:

**import React from 'react';**

**import { useSelector } from 'react-redux';**

**import withDataFetching from '../hocs/withDataFetching';**

**import Card from 'react-bootstrap/Card';**

**import Container from 'react-bootstrap/Container';**

**import { Link } from 'react-router-dom';**

**import Table from 'react-bootstrap/Table';**

**import { Line } from 'react-chartjs-2';**

**import Chart from 'chart.js/auto'; // For Chart Category**

**import { FaRegAddressCard, FaChartBar, FaCog, FaLightbulb, FaSignOutAlt} from 'react-icons/fa';**

**import { FaCalendarAlt, FaDatabase, FaPhone, FaClock, FaHome, FaUser } from 'react-icons/fa';**

**const UsageAnalysisChartsPage = ({ data, loading, error }) => {**

**const icons = [**

**<FaRegAddressCard />,**

**<FaChartBar />,**

**<FaUser />,**

**<FaCog />,**

**<FaLightbulb />,**

**<FaSignOutAlt />,**

**<FaHome />,**

**];**

**const isAuthenticated = useSelector((state) => state.auth.isAuthenticated);**

**const userData = useSelector((state) => state.auth.user);**

**console.log("Data:"+JSON.stringify(data));**

**if (loading) {**

**return <p>Loading...</p>;**

**}**

**if (error) {**

**return <p>Error: {error}</p>;**

**}**

**const telecomData = data.telecomPlanUsage;**

**const labels = Array.isArray(telecomData) && telecomData.length > 0**

**? telecomData.map((usage) => usage.date)**

**: [];**

**const dataUsedList = telecomData.map((usage) => usage.dataUsed);**

**const minutesUsedList = telecomData.map((usage) => usage.minutesUsed);**

**const textMessagesSentList = telecomData.map((usage) => usage.textMessagesSent);**

**const chartData = {**

**labels: labels,**

**datasets: [**

**{**

**label: 'Data Used',**

**data: [...dataUsedList], // Convert to a list**

**fill: true,**

**backgroundColor: 'rgba(75,192,192,0.2)',**

**borderColor: 'rgba(75,192,192,1)',**

**borderWidth: 2,**

**},**

**{**

**label: 'Minutes Used',**

**data: [...minutesUsedList], // Convert to a list**

**fill: false,**

**borderColor: '#742774',**

**borderWidth: 2,**

**},**

**{**

**label: 'Text Messages Sent',**

**data: [...textMessagesSentList], // Convert to a list**

**fill: false,**

**borderColor: 'rgba(255, 205, 86, 1)',**

**borderWidth: 2,**

**},**

**],**

**};**

**return (**

**<div>**

**{isAuthenticated ? (**

**<Container className="p-3">**

**<Container className="p-3" style={{backgroundColor: '#8CDBFF'}}>**

**<img src='images/expense-image.png' alt='' width={300} height={200}/>**

**<h1 className="header">Welcome To Expense Management Application</h1>**

**<br/>**

**<h4 className="mb-4">Telecom Plan Usage</h4>**

**<h4 className="mb-4">Analysis Chart For {userData.user.name}</h4>**

**</Container>**

**<Container className="p-3">**

**<Card>**

**<Card.Body>**

**<Card.Link>{icons[6]} <Link to="/home">Home</Link></Card.Link>**

**<Card.Link>{icons[0]} <Link to="/plan">My Plan</Link></Card.Link>**

**<Card.Link>{icons[1]} <Link to="/usage">Usage</Link></Card.Link>**

**<Card.Link>{icons[3]} <Link to="/recommendation">Plan Recommendation</Link></Card.Link>**

**<Card.Link>{icons[4]} <Link to="/update-plan">Update Plan</Link></Card.Link>**

**<Card.Link>{icons[5]} <Link to="/logout">LogOut</Link></Card.Link>**

**</Card.Body>**

**</Card>**

**</Container>**

**<div>**

**<Line data={chartData} />**

**</div>**

**<br/><br/>**

**<div>**

**<Table striped bordered hover responsive>**

**<thead className="thead-dark">**

**<tr>**

**<th><FaCalendarAlt /> Date</th>**

**<th><FaDatabase /> Data Used</th>**

**<th><FaClock /> Minutes Used</th>**

**<th><FaPhone /> Text Messages Sent</th>**

**</tr>**

**</thead>**

**<tbody>**

**{telecomData.map((usage, index) => (**

**<tr key={index}>**

**<td>{usage.date}</td>**

**<td style={{ color: usage.dataUsed > 1024 ? 'red' : 'inherit' }}>{usage.dataUsed}</td>**

**<td>{usage.minutesUsed}</td>**

**<td>{usage.textMessagesSent}</td>**

**</tr>**

**))}**

**</tbody>**

**</Table>**

**</div>**

**</Container> ) : (**

**<p>Please Login First</p>**

**)}**

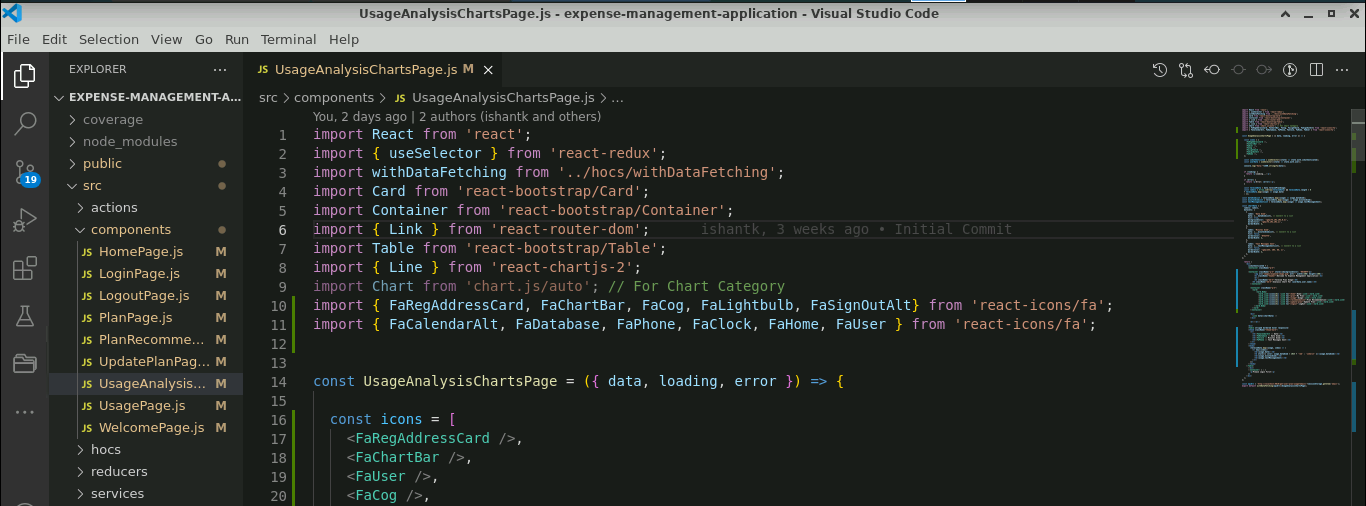
**</div>**

**);**

**};**

**const apiUrl = 'http://localhost:9010/get-user-plan-usage?email='+sessionStorage.getItem('email');**

**export default withDataFetching(apiUrl)(UsageAnalysisChartsPage);**

****

* 1. Click on the **Usage Analysis**

A screenshot of a computer

Description automatically generated

The output appears as shown below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Step 10: Create the plan recommendation**

* 1. Inside the **components** folder, create a file named **PlanRecommendationPage.js,** and add the following code as shown below:

**import React, { useState, useEffect, useMemo } from 'react';**

**import Card from 'react-bootstrap/Card';**

**import Container from 'react-bootstrap/Container';**

**import { useSelector } from 'react-redux';**

**import { Link } from 'react-router-dom';**

**import { FaRegAddressCard, FaChartBar, FaCog, FaLightbulb, FaSignOutAlt} from 'react-icons/fa';**

**import { FaHome, FaUser } from 'react-icons/fa';**

**import Badge from 'react-bootstrap/Badge';**

**import { FaIdCard, FaRegHdd, FaCrown, FaRegMoneyBillAlt } from 'react-icons/fa';**

**import { FaRegIdCard, FaDatabase, FaPhone, FaClock, FaGlobe, FaMoneyBillAlt } from 'react-icons/fa';**

**const PlanRecommendationPage = () => {**

**const icons = [**

**<FaRegAddressCard />,**

**<FaChartBar />,**

**<FaUser />,**

**<FaCog />,**

**<FaLightbulb />,**

**<FaSignOutAlt />,**

**<FaHome />,**

**];**

**const isAuthenticated = useSelector((state) => state.auth.isAuthenticated);**

**const userData = useSelector((state) => state.auth.user);**

**console.log("isAuthenticated:"+isAuthenticated);**

**console.log("userData:"+userData);**

**// State to hold user preferences**

**const [userPreferences, setUserPreferences] = useState({**

**dataAllowance: 20,**

**minutes: 500,**

**textMessages: 200,**

**internationalRoaming: true,**

**contractLengthMonths: 24,**

**monthlyCost: 0,**

**});**

**// State to hold plans data**

**const [plansData, setPlansData] = useState({ telecomPlans: [] });**

**useEffect(() => {**

**// Fetch plans data from the API endpoint**

**const fetchPlansData = async () => {**

**try {**

**const response = await fetch('http://localhost:9010/get-plans');**

**const data = await response.json();**

**console.log("Plans Data:"+JSON.stringify(data));**

**setPlansData(data);**

**} catch (error) {**

**console.error('Error fetching plans data:', error);**

**}**

**};**

**// Call the fetch function**

**fetchPlansData();**

**}, []);**

**// Function to recommend a plan based on user preferences**

**const recommendPlan = useMemo(() => {**

**return () => {**

**const { dataAllowance, minutes, textMessages, internationalRoaming, contractLengthMonths, monthlyCost } = userPreferences;**

**// Filter plans based on user preferences**

**const filteredPlans = plansData.telecomPlans.filter((plan) => {**

**return (**

**plan.dataAllowanceGB >= dataAllowance &&**

**plan.minutes >= minutes &&**

**plan.textMessages >= textMessages**

**);**

**});**

**// Sort the filtered plans by a scoring system**

**const sortedPlans = filteredPlans.sort((a, b) => {**

**// Calculate a score for each plan based on how well it matches the preferences**

**const scoreA = calculatePlanScore(a);**

**const scoreB = calculatePlanScore(b);**

**// Sort in descending order of score**

**return scoreB - scoreA;**

**});**

**console.log("sortedPlans:" + JSON.stringify(sortedPlans));**

**// Return the recommended plan (the one with the highest score)**

**return sortedPlans.length > 0 ? sortedPlans[0] : null;**

**};**

**}, [userPreferences, plansData]);**

**// Function to calculate a score for a plan based on how well it matches the preferences**

**const calculatePlanScore = (plan) => {**

**const { dataAllowance, minutes, textMessages, internationalRoaming, contractLengthMonths, monthlyCost } = userPreferences;**

**// You can adjust the weights based on the importance of each factor**

**const weightDataAllowance = 2;**

**const weightMinutes = 1.5;**

**const weightTextMessages = 1.2;**

**const weightInternationalRoaming = 1.5;**

**const weightContractLengthMonths = 1;**

**const weightMonthlyCost = 2;**

**// Calculate the score for the plan**

**const score =**

**weightDataAllowance \* (plan.dataAllowanceGB / dataAllowance) +**

**weightMinutes \* (plan.minutes / minutes) +**

**weightTextMessages \* (plan.textMessages / textMessages) +**

**weightInternationalRoaming \* (plan.internationalRoaming === internationalRoaming ? 1 : 0) +**

**weightContractLengthMonths \* (contractLengthMonths / plan.contractLengthMonths) +**

**weightMonthlyCost \* (1 - plan.monthlyCost / monthlyCost);**

**return score;**

**};**

**const recommendedPlan = recommendPlan();**

**console.log("recommendedPlan: "+recommendedPlan);**

**return (**

**<div>**

**{recommendedPlan ? (**

**<Container className="p-3">**

**<Container className="p-3" style={{backgroundColor: '#8CDBFF'}}>**

**<img src='images/expense-image.png' alt='' width={300} height={200}/>**

**<h1 className="header">Welcome To Expense Management Application</h1>**

**<br/>**

**<h2>Plan Recommendation</h2>**

**<br/>**

**<h4>Hello, {userData.user.name}!</h4>**

**</Container>**

**<Container className="p-3">**

**<Card>**

**<Card.Body>**

**<Card.Link>{icons[6]} <Link to="/home">Home</Link></Card.Link>**

**<Card.Link>{icons[0]} <Link to="/plan">My Plan</Link></Card.Link>**

**<Card.Link>{icons[2]} <Link to="/usage-analysis">Usage Analysis</Link></Card.Link>**

**<Card.Link>{icons[4]} <Link to="/update-plan">Update Plan</Link></Card.Link>**

**<Card.Link>{icons[5]} <Link to="/logout">LogOut</Link></Card.Link>**

**</Card.Body>**

**</Card>**

**</Container>**

**<Container style={{backgroundColor: '#8CDBFF'}} className="p-3">**

**<Card className="shadow-lg rounded">**

**<Card.Body className="text-center">**

**<Card.Title>Current Plan</Card.Title>**

**<div className="mb-4">**

**<FaRegHdd className="display-4 text-primary" />**

**</div>**

**<Card.Title className="h4 mb-3">{icons[2]} {userData.user.email}</Card.Title>**

**<Card.Subtitle className="mb-2 text-muted">**

**<FaIdCard className="mr-2" /> Plan ID: {userData.user.plan.planId}**

**</Card.Subtitle>**

**<Card.Text className="lead">**

**<Badge variant="success" className="mr-2">**

**<FaCrown className="mr-1" />**

**{userData.user.plan.planName}**

**</Badge>**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2" /> Validity: </strong> {userData.user.plan.contractLengthMonths} (Months)**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2" /> Monthly Cost:</strong> ₹{userData.user.plan.monthlyCost}**

**</Card.Text>**

**</Card.Body>**

**</Card>**

**</Container>**

**<br></br>**

**<Container style={{backgroundColor: '#B4FFC4'}} className="p-3">**

**<Card>**

**<Card.Body>**

**<h4>We Recommend the following Plan for you:</h4>**

**<br />**

**<Card.Title>Recommend Plan Details</Card.Title>**

**<Card.Text>**

**<strong><FaRegIdCard color="blue" /> Plan Name:</strong> {recommendedPlan.planName}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaDatabase color="green" /> Data Allowance:</strong> {recommendedPlan.dataAllowanceGB} GB**

**</Card.Text>**

**<Card.Text>**

**<strong><FaClock color="orange" /> Minutes:</strong> {recommendedPlan.minutes}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaPhone color="purple" /> Text Messages:</strong> {recommendedPlan.textMessages}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaGlobe color="red" /> International Roaming:</strong> {recommendedPlan.internationalRoaming ? 'Yes' : 'No'}**

**</Card.Text>**

**<Card.Text>**

**<strong><FaMoneyBillAlt color="black" /> Monthly Cost:</strong> ₹{recommendedPlan.monthlyCost}**

**</Card.Text>**

**</Card.Body>**

**</Card>**

**</Container>**

**</Container>**

**) : (**

**<p>No plans match your preferences. Please adjust your preferences and try again.</p>**

**)}**

**</div>**

**);**

**};**

**export default PlanRecommendationPage;**

**A screen shot of a computer

Description automatically generated**

* 1. Click on **Plan Recommendation**

A screenshot of a computer

Description automatically generated

The output appears as shown below:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Step 11: Create the update plan**

* 1. Inside the **components** folder, create a folder named **UpdatePlanPage.js** and enter the code as shown below:

**import React, { useState, useEffect } from 'react';**

**import { useSelector, useDispatch } from 'react-redux';**

**import Card from 'react-bootstrap/Card';**

**import Container from 'react-bootstrap/Container';**

**import ListGroup from 'react-bootstrap/ListGroup';**

**import Button from 'react-bootstrap/Button';**

**import { Link } from 'react-router-dom';**

**import { ToastContainer, toast } from 'react-toastify';**

**import 'react-toastify/dist/ReactToastify.css';**

**import { updatePlanAsync } from '../actions/authSlice';**

**import Badge from 'react-bootstrap/Badge';**

**import CardGroup from 'react-bootstrap/CardGroup';**

**import { FaRegHdd, FaIdCard, FaCrown, FaRegAddressCard, FaChartBar, FaUser, FaCog, FaLightbulb, FaSignOutAlt, FaRegMoneyBillAlt } from 'react-icons/fa';**

**import { FaRegIdCard, FaDatabase, FaPhone, FaClock, FaGlobe, FaMoneyBillAlt } from 'react-icons/fa';**

**const UpdatePlanPage = () => {**

**const icons = [**

**<FaRegAddressCard />,**

**<FaChartBar />,**

**<FaUser />,**

**<FaCog />,**

**<FaLightbulb />,**

**<FaSignOutAlt />,**

**];**

**const isAuthenticated = useSelector((state) => state.auth.isAuthenticated);**

**const userData = useSelector((state) => state.auth.user);**

**const dispatch = useDispatch();**

**const [plansData, setPlansData] = useState({ telecomPlans: [] });**

**useEffect(() => {**

**const fetchPlansData = async () => {**

**try {**

**const response = await fetch('http://localhost:9010/get-plans');**

**const data = await response.json();**

**setPlansData(data);**

**} catch (error) {**

**console.error('Error fetching plans data:', error);**

**}**

**};**

**fetchPlansData();**

**}, []);**

**const handleUpdatePlan = async (planId) => {**

**try {**

**// Dispatch the updatePlanAsync thunk to update the plan in Redux store**

**await dispatch(updatePlanAsync({ email: sessionStorage.getItem('email'), planId }));**

**// Show a success notification**

**toast.success('Plan updated successfully to '+planId, {**

**position: 'top-right',**

**autoClose: 3000,**

**hideProgressBar: true,**

**closeOnClick: true,**

**pauseOnHover: true,**

**draggable: true,**

**});**

**} catch (error) {**

**console.error('Error updating user plan:', error);**

**}**

**};**

**return (**

**<Container className="p-3">**

**<Container className="p-3" style={{backgroundColor: '#8CDBFF'}}>**

**<img src='images/expense-image.png' alt='' width={300} height={200}/>**

**<h1 className="header">Welcome To Expense Management Application</h1>**

**<h2>Update Plan</h2>**

**<h4>Hello, {userData.user.name}!</h4>**

**</Container>**

**<Container className="p-3">**

**<Card>**

**<Card.Body>**

**<Card.Link>{icons[0]} <Link to="/plan">My Plan</Link></Card.Link>**

**<Card.Link>{icons[1]} <Link to="/usage">Usage</Link></Card.Link>**

**<Card.Link>{icons[2]} <Link to="/usage-analysis">Usage Analysis</Link></Card.Link>**

**<Card.Link>{icons[3]} <Link to="/recommendation">Plan Recommendation</Link></Card.Link>**

**<Card.Link>{icons[5]} <Link to="/logout">LogOut</Link></Card.Link>**

**</Card.Body>**

**</Card>**

**</Container>**

**<Container style={{backgroundColor: '#B4FFC4'}} className="p-3">**

**<Card className="shadow-lg rounded">**

**<Card.Body className="text-center">**

**<Card.Title>Current Plan</Card.Title>**

**<div className="mb-4">**

**<FaRegHdd className="display-4 text-primary" />**

**</div>**

**<Card.Title className="h4 mb-3">{icons[2]} {userData.user.email}</Card.Title>**

**<Card.Subtitle className="mb-2 text-muted">**

**<FaIdCard className="mr-2" /> Plan ID: {userData.user.plan.planId}**

**</Card.Subtitle>**

**<Card.Text className="lead">**

**<Badge variant="success" className="mr-2">**

**<FaCrown className="mr-1" />**

**{userData.user.plan.planName}**

**</Badge>**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2" /> Validity: </strong> {userData.user.plan.contractLengthMonths} (Months)**

**</Card.Text>**

**<Card.Text>**

**<strong><FaRegMoneyBillAlt className="mr-2" /> Monthly Cost:</strong> ₹{userData.user.plan.monthlyCost}**

**</Card.Text>**

**</Card.Body>**

**</Card>**

**</Container>**

**<br></br>**

**<Container style={{backgroundColor: '#8CDBFF'}} className="p-3">**

**<CardGroup>**

**{plansData.telecomPlans.slice(0, 3).map((plan) => (**

**<Card key={plan.planId}>**

**<Card.Body>**

**<Card.Title>{plan.planName}</Card.Title>**

**<Card.Text><FaRegIdCard color="blue" /> Plan ID: {plan.planId}</Card.Text>**

**<Card.Text><FaDatabase color="green" /> Data Allowance: {plan.dataAllowanceGB} GB</Card.Text>**

**<Card.Text><FaClock color="orange" /> Minutes: {plan.minutes}</Card.Text>**

**<Card.Text><FaPhone color="purple" /> Text Messages: {plan.textMessages}</Card.Text>**

**<Card.Text><FaRegMoneyBillAlt color="green"/> Validity: {userData.user.plan.contractLengthMonths} (Months)</Card.Text>**

**<Card.Text><FaGlobe color="red" /> International Roaming: {plan.internationalRoaming ? 'Yes' : 'No'}</Card.Text>**

**<Card.Text><FaMoneyBillAlt color="black" /> Monthly Cost: ₹{plan.monthlyCost}</Card.Text>**

**<Button variant="info" onClick={() => handleUpdatePlan(plan.planId)}>**

**Update to {plan.planName}**

**</Button>**

**</Card.Body>**

**</Card>**

**))}**

**</CardGroup>**

**<CardGroup>**

**{plansData.telecomPlans.slice(3).map((plan) => (**

**<Card key={plan.planId}>**

**<Card.Body>**

**<Card.Title>{plan.planName}</Card.Title>**

**<Card.Text><FaRegIdCard color="blue" /> Plan ID: {plan.planId}</Card.Text>**

**<Card.Text><FaDatabase color="green" /> Data Allowance: {plan.dataAllowanceGB} GB</Card.Text>**

**<Card.Text><FaClock color="orange" /> Minutes: {plan.minutes}</Card.Text>**

**<Card.Text><FaPhone color="purple" /> Text Messages: {plan.textMessages}</Card.Text>**

**<Card.Text><FaGlobe color="red" /> International Roaming: {plan.internationalRoaming ? 'Yes' : 'No'}</Card.Text>**

**<Card.Text><FaMoneyBillAlt color="black" /> Monthly Cost: ₹{plan.monthlyCost}</Card.Text>**

**<Button variant="info" onClick={() => handleUpdatePlan(plan.planId)}>**

**Update to {plan.planName}**

**</Button>**

**</Card.Body>**

**</Card>**

**))}**

**</CardGroup>**

**{/\* ToastContainer for notifications \*/}**

**<ToastContainer />**

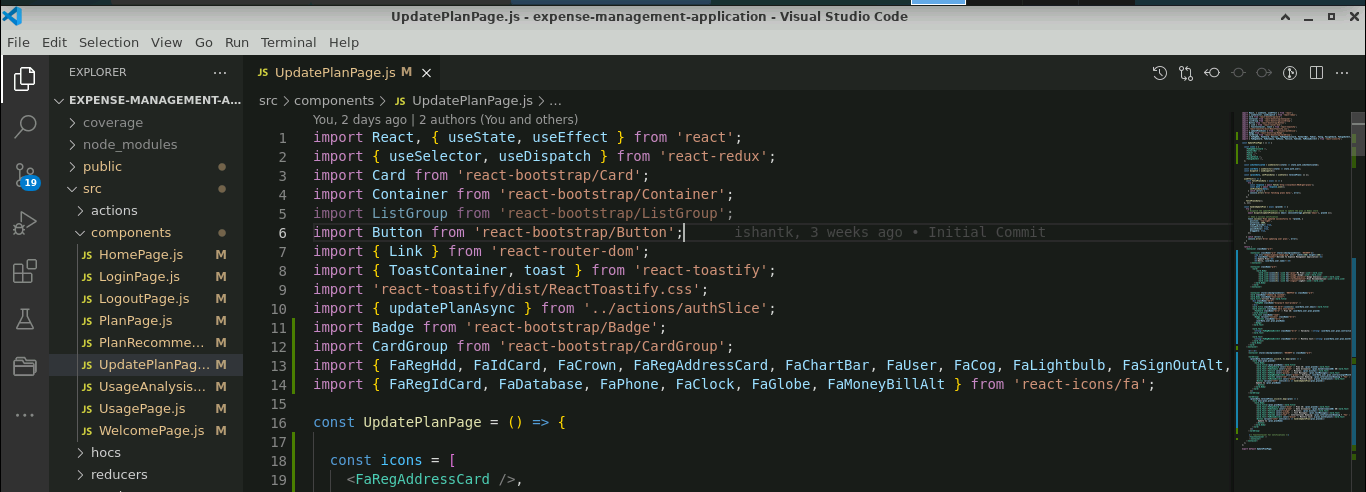
**</Container>**

**</Container>**

**);**

**};**

**export default UpdatePlanPage;**

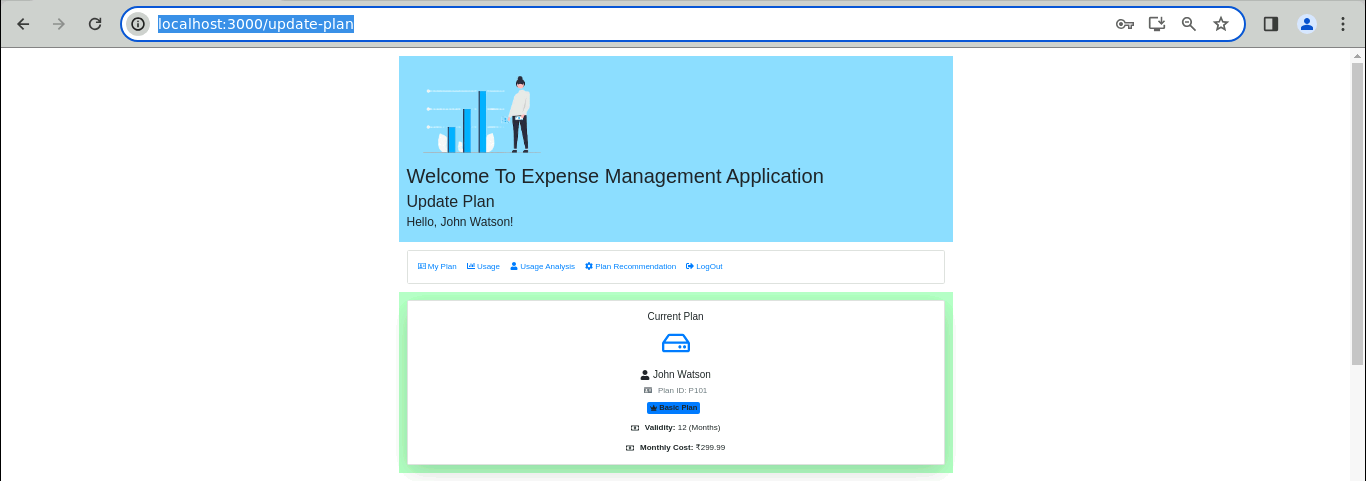
****

* 1. Click on the **Update Plan**

A screenshot of a computer

Description automatically generated

The output appears as shown below:



A screenshot of a computer

Description automatically generated

* 1. Under **Premium Plan** select **Upgrade to Premium plan**

A screenshot of a computer

Description automatically generated

The output appears as shown below:

A screenshot of a computer

Description automatically generated

**Current Plan** updated successfully as shown below:

A screenshot of a computer

Description automatically generated

* 1. Click on **LogOut** (refer to the above screenshot) to logout from the home page

A green screen with black text

Description automatically generated