Creating Project:

npx create-react-app admin-dashboard-react

Dependencies

yarn add @material-ui/core

yarn add @material-ui/icons

yarn add recharts (for charts)

yarn add react-router-dom

Style/Font style in public.html:

<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="utf-8" />

    <link rel="icon" href="%PUBLIC\_URL%/favicon.ico" />

    <meta name="viewport" content="width=device-width, initial-scale=1" />

    <meta name="theme-color" content="#000000" />

    <meta

      name="description"

      content="Web site created using create-react-app"

    />

    <link rel="preconnect" href="https://fonts.googleapis.com">

    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

    <link href="https://fonts.googleapis.com/css2?family=Source+Sans+Pro:wght@200;300;400;600;700;900&display=swap" rel="stylesheet">

    <style>

      \*{

        font-family: 'Source Sans Pro', sans-serif;

      }

    </style>

    <title>React Admin App</title>

  </head>

  <body>

    <noscript>You need to enable JavaScript to run this app.</noscript>

    <div id="root"></div>

  </body>

</html>

If top bar at 5opx then sidebar height will be:

.sidebar{

    flex: 1;

    height: calc(100vh - 50px); /\*50px for topbar\*/

    position: sticky;

    top:50px

}

Sidebar:

import React from 'react'

import './Sidebar.css'

//material ui

import {

    LineStyle,

    Timeline,

    TrendingUp,

    PermIdentity,

    Storefront,

    AttachMoney,

    BarChart,

    MailOutline,

    DynamicFeed,

    ChatBubbleOutline,

    WorkOutline,

    Report,

  } from "@material-ui/icons";

function Sidebar() {

    return (

        <div className="sidebar">

            <div className="sidebar\_\_wrapper">

                {/\*First Menu\*/}

                <div className="sidebar\_\_menu">

                    <h3 className="sidebar\_\_title">Dashboard</h3>

                    <ul className="sidebar\_\_list">

                        <li className="sidebarList\_\_item active">

                            <LineStyle className="sidebar\_\_icon" />

                            Home

                        </li>

                        <li className="sidebarList\_\_item">

                            <Timeline className="sidebar\_\_icon" />

                            Analytics

                        </li>

                        <li className="sidebarList\_\_item">

                           <TrendingUp className="sidebar\_\_icon" />

                            Sales

                        </li>

                    </ul>

                </div>

                {/\*Second Menu\*/}

                <div className="sidebar\_\_menu">

                <h3 className="sidebar\_\_title">Quick Menu</h3>

                    <ul className="sidebar\_\_list">

                        <li className="sidebarList\_\_item active">

                            <PermIdentity className="sidebar\_\_icon" />

                            Users

                        </li>

                        <li className="sidebarList\_\_item">

                            <Storefront className="sidebar\_\_icon" />

                            Products

                        </li>

                        <li className="sidebarList\_\_item">

                           <AttachMoney className="sidebar\_\_icon" />

                            Transactions

                        </li>

                    </ul>

                </div>

                {/\*Third Menu\*/}

                <div className="sidebar\_\_menu">

                    <h3 className="sidebar\_\_title">Notifications</h3>

                    <ul className="sidebar\_\_list">

                        <li className="sidebarList\_\_item active">

                            <MailOutline className="sidebar\_\_icon" />

                            Mail

                        </li>

                        <li className="sidebarList\_\_item">

                            <DynamicFeed className="sidebar\_\_icon" />

                            Feedback

                        </li>

                        <li className="sidebarList\_\_item">

                           <ChatBubbleOutline className="sidebar\_\_icon" />

                            Messages

                        </li>

                    </ul>

                </div>

                {/\*Fourth Menu\*/}

                <div className="sidebar\_\_menu">

                    <h3 className="sidebar\_\_title">Staff</h3>

                    <ul className="sidebar\_\_list">

                        <li className="sidebarList\_\_item active">

                            <WorkOutline className="sidebar\_\_icon" />

                            Manage

                        </li>

                        <li className="sidebarList\_\_item">

                            <Timeline className="sidebar\_\_icon" />

                            Analytics

                        </li>

                        <li className="sidebarList\_\_item">

                           <Report className="sidebar\_\_icon" />

                            Reports

                        </li>

                    </ul>

                </div>

            </div>

        </div>

    )

}

export default Sidebar

.sidebar{

    flex: 1;

    height: calc(100vh - 50px); /\*50px for topbar\*/

    position: sticky;

    top:50px;

    background-color: rgb(251, 251, 255);

}

.sidebar\_\_wrapper{

    padding: 20px;

    color: #555;

}

.sidebar\_\_menu{

    margin-bottom: 10px;

}

.sidebar\_\_title{

    font-size: 13px;

    color: rgb(187, 186, 186)

}

.sidebar\_\_list{

    list-style: none;

    padding: 5px;

}

.sidebarList\_\_item{

    display: flex;

    flex-direction: row;

    align-items: center;

    padding: 5px;

    cursor: pointer;

    border-radius: 10px;

}

.sidebarList\_\_item.active, .sidebarList\_\_item:hover{

    background-color: rgb(240, 240, 255);

}

.sidebar\_\_icon{

    margin-right: 5px;

    font-size: 20px !important;

}

Using flex times in sidebar:

App.js:

import React from 'react'

import Sidebar from './components/Sidebar/Sidebar';

import TopBar from  './components/TopBar/TopBar'

import './App.css'

function App() {

  return (

    <div className="App">

        <TopBar/>

        <div className="container">

          <Sidebar/>

          <div className="other">Other Pages</div>

        </div>

    </div>

  );

}

export default App;

App.css:

.container{

    display: flex;

}

.other{

    flex: 4;

}

As we say here other pages flex will be 4 times sidebar flex where sidebar flex is 1

.sidebar{

    flex: 1;

    height: calc(100vh - 50px); /\*50px for topbar\*/

    position: sticky;

    top:50px;

    background-color: rgb(251, 251, 255);

}

If in percent, then we provide sidebar flex to be 0.2 or 0.3 and rest 0.6 or 0.7 to other page in their css file not in app.css

Calling Two same level classes in CSS:

                        <li className="sidebarList\_\_item active">

                            <MailOutline className="sidebar\_\_icon" />

                            Mail

                        </li>

.sidebarList\_\_item.active, .sidebarList\_\_item:hover{

    background-color: rgb(240, 240, 255);

}

External or materialui icon with html functionality:

                           <div className="userUpdateRight\_\_upload">

                                <img className="userUpdate\_\_img" alt="" src="https://images.pexels.com/photos/1152994/pexels-photo-1152994.jpeg?auto=compress&cs=tinysrgb&dpr=2&w=500"/>

                                <label htmlFor="file"><Publish/></label>

                                <input type="file" id="file" style={{display: "none"}}/>

                            </div>

Storing different states in single state, or different values in one state:

const [movie, setMovie] = useState(null);

const handleChange = (e) => {

        const value = e.target.value;

        setMovie({...movie, [e.target.name]:value})

    }

<div className="addProduct\_\_item">

                    <label>Title</label>

                    <input type="text" placeholder="John Wick" name="title" onChange={handleChange}/>

                </div>

                <div className="addProduct\_\_item">

                    <label>Description</label>

                    <input type="text" placeholder="description" name="desc" onChange={handleChange}/>

                </div>

                <div className="addProduct\_\_item">

                    <label>Duration</label>

                    <input type="text" placeholder="duration" name="duration" onChange={handleChange}/>

                </div>

                <div className="addProduct\_\_item">

                    <label>Year</label>

                    <input type="text" placeholder="year" name="year" onChange={handleChange}/>

                </div>

                <div className="addProduct\_\_item">

                    <label>Genre</label>

                    <input type="text" placeholder="genre" name="genre" onChange={handleChange}/>

                </div>

                <div className="addProduct\_\_item">

                    <label>Limit</label>

                    <input type="text" placeholder="limit" name="limit" onChange={handleChange}/>

                </div>

                <div className="addProduct\_\_item">

                    <label>is Series?</label>

                    <select name="isSeries" id="isSeries" onChange={handleChange}>

                        <option value="false">No</option>

                        <option value="true">Yes</option>

                    </select>

                </div>

const handleSubmit = (e) => {

        e.preventDefault()

        createMovie(movie, dispatch)

    }

What happen is movie is the main state

We assign name to different inputs and handleChange passed to then onChange

In handle changes, basically it update movie by key value pair where key is the name of the input field and value is the value written on the input field. For example title field inside movie state object updates like

movie = {title: “Superman”, limit: “4”….}

and we can simply extract them like movie[title] extra

In the end we passed movie to post api inside createMovie function under movieApiCall.js

ABOUT API:

get request api always on useEffect

post on button submit with some provided state or value to be post

If get or post on same page, then no need of redux to store. If we need to populate information on different components, then comes redux or context api and then dispatch and useSelector comes so we create separate file containing api call where we use dispatch to update our states like lists or movies and then use them easily in different components or pages using useSelector