# Install components

Để cài đặt các component cần thiết cho chương trình t dùng lệnh sau:

npm i @mui/material @emotion/react @emotion/styled @mui/x-data-grid @mui/icons-material react-router-dom@6 react-pro-sidebar formik yup @fullcalendar/core @fullcalendar/daygrid @fullcalendar/timegrid @fullcalendar/list @nivo/core @nivo/pie @nivo/line @nivo/bar @nivo/geo

# Cấu trúc của chương trình

## Các folder chính

# Tổng quát các component của material UI

## Component Typography

### Prop variant

Kiểu là một cái tag ở trong html, giả sử mình truyền vào variant = ‘h1’, nó sẽ gói tất cả nội dung ở trong một cái tag <h1></h1>

# Cài đặt màu sắc cho ứng dụng

## Dùng material UI

## Using tailwind shades to create all shade of a color

Bôi đen một màu rồi ctrl+k (đối với window)

## Tạo bảng màu cho ứng dụng

export *const* tokens = (*mode*) => ({

    ...(mode === 'dark' ?

            {

                grey: {

                    100: "#e0e0e0",

                    200: "#c2c2c2",

                    300: "#a3a3a3",

                    400: "#858585",

                    500: "#666666",

                    600: "#525252",

                    700: "#3d3d3d",

                    800: "#292929",

                    900: "#141414"

                },

                primary: {

                    100: "#d0d1d5",

                    200: "#a1a4ab",

                    300: "#727681",

                    400: "#434957",

                    500: "#141b2d",

                    600: "#101624",

                    700: "#0c101b",

                    800: "#080b12",

                    900: "#040509"

                }, *//dark blue color*

                greenAccent: {

                    100: "#dbf5ee",

                    200: "#b7ebde",

                    300: "#94e2cd",

                    400: "#70d8bd",

                    500: "#4cceac",

                    600: "#3da58a",

                    700: "#2e7c67",

                    800: "#1e5245",

                    900: "#0f2922"

                }, *//green accent*

                redAccent: {

                    100: "#f8dcdb",

                    200: "#f1b9b7",

                    300: "#e99592",

                    400: "#e2726e",

                    500: "#db4f4a",

                    600: "#af3f3b",

                    700: "#832f2c",

                    800: "#58201e",

                    900: "#2c100f"

                }, *//red accent*

                blueAccent: {

                    100: "#e1e2fe",

                    200: "#c3c6fd",

                    300: "#a4a9fc",

                    400: "#868dfb",

                    500: "#6870fa",

                    600: "#535ac8",

                    700: "#3e4396",

                    800: "#2a2d64",

                    900: "#151632"

                }, *//blue accent*

            }

            :

            {

                grey: {

                    100: "#141414",

                    200: "#292929",

                    300: "#3d3d3d",

                    400: "#525252",

                    500: "#666666",

                    600: "#858585",

                    700: "#a3a3a3",

                    800: "#c2c2c2",

                    900: "#e0e0e0",

                },

                primary: {

                    100: "#040509",

                    200: "#080b12",

                    300: "#0c101b",

                    400: "#f2f0f0",

                    500: "#141b2d",

                    600: "#434957",

                    700: "#727681",

                    800: "#a1a4ab",

                    900: "#d0d1d5",

                }, *//dark blue color*

                greenAccent: {

                    100: "#0f2922",

                    200: "#1e5245",

                    300: "#2e7c67",

                    400: "#3da58a",

                    500: "#4cceac",

                    600: "#70d8bd",

                    700: "#94e2cd",

                    800: "#b7ebde",

                    900: "#dbf5ee",

                }, *//green accent*

                redAccent: {

                    100: "#2c100f",

                    200: "#58201e",

                    300: "#832f2c",

                    400: "#af3f3b",

                    500: "#db4f4a",

                    600: "#e2726e",

                    700: "#e99592",

                    800: "#f1b9b7",

                    900: "#f8dcdb",

                }, *//red accent*

                blueAccent: {

                    100: "#151632",

                    200: "#2a2d64",

                    300: "#3e4396",

                    400: "#535ac8",

                    500: "#6870fa",

                    600: "#868dfb",

                    700: "#a4a9fc",

                    800: "#c3c6fd",

                    900: "#e1e2fe",

                }, *//blue accent*

            }

    )

})

## Tạo ra một object các cài đặt màu sắc cho material ui sử dụng

*//mui theme settings*

export *const* themeSettings = (*mode*) => {

*const* colors = tokens(mode);

    return{

        pallete: {

            mode: mode,

            ...(mode === 'dark'

            ?

            {

*// palette values for dark mode*

                primary: {

                    main: colors.primary[500],

                },

                secondary: {

                    main: colors.greenAccent[500],

                },

                neutral: {

                    dark: colors.grey[700],

                    main: colors.grey[500],

                    light: colors.grey[100],

                },

                background: {

                    default: colors.primary[500],

                },

            }

            :

            {

*// palette values for light mode*

                primary: {

                  main: colors.primary[100],

                },

                secondary: {

                  main: colors.greenAccent[500],

                },

                neutral: {

                  dark: colors.grey[700],

                  main: colors.grey[500],

                  light: colors.grey[100],

                },

                background: {

                  default: "#fcfcfc",

                },

            }

            )

        },

        typography: {

            fontFamily: ["Source Sans Pro", "sans-serif"].join(","),

            fontSize: 12,

            h1: {

              fontFamily: ["Source Sans Pro", "sans-serif"].join(","),

              fontSize: 40,

            },

            h2: {

              fontFamily: ["Source Sans Pro", "sans-serif"].join(","),

              fontSize: 32,

            },

            h3: {

              fontFamily: ["Source Sans Pro", "sans-serif"].join(","),

              fontSize: 24,

            },

            h4: {

              fontFamily: ["Source Sans Pro", "sans-serif"].join(","),

              fontSize: 20,

            },

            h5: {

              fontFamily: ["Source Sans Pro", "sans-serif"].join(","),

              fontSize: 16,

            },

            h6: {

              fontFamily: ["Source Sans Pro", "sans-serif"].join(","),

              fontSize: 14,

            },

          },

    };

};

## Tạo theme và function đổi theme cho nút bấm

Mỗi lần nhấn nút bấm sẽ đổi theme của ứng dụng

*// context for color mode, don't know how the hell this work*

export *const* ColorModeContext = createContext({

    toggleColorMode: () => {},  *//this line is unnecessary*

  });

  export *const* useMode = () => {

*const* [mode, setMode] = useState("dark");

*const* colorMode = useMemo(

      () => ({

        toggleColorMode: () =>

          setMode((*prev*) => (prev === "light" ? "dark" : "light")),

      }),

      []

    );

*const* theme = useMemo(() => createTheme(themeSettings(mode)), [mode]);

    return [theme, colorMode];

  };

## Set up cho theme ở trong App.js

import { ColorModeContext, useMode } from "./theme";

import { CssBaseline, ThemeProvider } from "@mui/material";

*function* App() {

*const* [theme, colorMode] = useMode();

  return (

    <ColorModeContext.Provider *value*={colorMode}>

      <ThemeProvider *theme*={theme}>

        <CssBaseline/>

        <div *className*="app">

            <main *className*="content"></main>

        </div>

      </ThemeProvider>

    </ColorModeContext.Provider>

  );

}

export default App;

# Cài đặt App.js

App.js sẽ là nơi đặt tất cả các route của ứng dụng

Code:

import { useState } from "react";

import { Routes, Route } from "react-router-dom";

import Topbar from "./scenes/global/Topbar";

import Sidebar from "./scenes/global/Sidebar";

import Dashboard from "./scenes/dashboard";

import Team from "./scenes/team";

*// import Invoices from "./scenes/invoices";*

*// import Contacts from "./scenes/contacts";*

*// import Bar from "./scenes/bar";*

*// import Form from "./scenes/form";*

*// import Line from "./scenes/line";*

*// import Pie from "./scenes/pie";*

*// import FAQ from "./scenes/faq";*

*// import Geography from "./scenes/geography";*

import { CssBaseline, ThemeProvider } from "@mui/material";

import { ColorModeContext, useMode } from "./theme";

*function* App() {

*const* [theme, colorMode] = useMode();

*const* [isSidebar, setIsSidebar] = useState(true);

  return (

    <ColorModeContext.Provider *value*={colorMode}>

      <ThemeProvider *theme*={theme}>

        <CssBaseline/>

        <div *className*="app">

          <Sidebar *isSidebar*={isSidebar} />

            <main *className*="content">

              <Topbar/>

              <Routes>

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

                <Route *path*="/team" *element*={<Team />} />

                {*/\* <Route path="/contacts" element={<Contacts />} /> \*/*}

                {*/\* <Route path="/invoices" element={<Invoices />} /> \*/*}

                {*/\* <Route path="/form" element={<Form />} /> \*/*}

                {*/\* <Route path="/bar" element={<Bar />} /> \*/*}

                {*/\* <Route path="/pie" element={<Pie />} /> \*/*}

                {*/\* <Route path="/line" element={<Line />} /> \*/*}

                {*/\* <Route path="/faq" element={<FAQ />} /> \*/*}

                {*/\* <Route path="/calendar" element={<Calendar />} /> \*/*}

                {*/\* <Route path="/geography" element={<Geography />} /> \*/*}

              </Routes>

            </main>

        </div>

      </ThemeProvider>

    </ColorModeContext.Provider>

  );

}

export default App;

# Folder sences

## Folder landing

### Cài đặt landing

#### Set up index.jsx

Giải thích:

- Link các button tới các dashboard page của của mỗi cart

+ Code:

<Box *mt*={3} *position*="absolute" *bottom*="-80px" >

                        <Link *to*="/landing1/dashboard">

                            <Button

*sx*={{

                                backgroundColor: colors.blueAccent[700],

                                color: colors.grey[100],

                                fontSize: "14px",

                                fontWeight: "bold",

                                padding: "10px 20px",

                                }}

                            >

                                <DetailsIcon *sx*={{ mr: "10px" }} />

                                Details

                            </Button>

                        </Link>

                    </Box>

+ Dùng Link để gán url của page dashboard với lại button

Code:

import { Box, Button, IconButton, Typography, useTheme } from "@mui/material";

import { tokens } from "../../theme";

import { mockTransactions } from "../../data/mockData";

import Header from "../../components/Header";

import DownloadOutlinedIcon from "@mui/icons-material/DownloadOutlined";

import BarChart from "../../components/BarChart";

import LineChart from "../../components/LineChart";

import { LineChartApex } from "../../components/ApexChart/LineChartApex";

import { BarChartApex } from "../../components/ApexChart/BarChartApex";

import useFetch from "../../data/dataFetch";

import { useState } from "react";

import { Link } from "react-router-dom";

import { ProSidebar, Menu, MenuItem } from "react-pro-sidebar";

import HomeOutlinedIcon from "@mui/icons-material/HomeOutlined";

import DetailsIcon from '@mui/icons-material/Details';

*const* Item = ({ *title*, *to*, *icon*, *selected*, *setSelected* }) => {

*const* theme = useTheme();

*const* colors = tokens(theme.palette.mode);

    return (

      <MenuItem

*active*={selected === title}

*style*={{

          color: colors.grey[100],

        }}

*onClick*={() => setSelected(title)}

*icon*={icon}

      >

        <Typography>{title}</Typography>

        <Link *to*={to} />

      </MenuItem>

    );

  };

*const* Landing = () => {

*const* theme = useTheme();

*const* colors = tokens(theme.palette.mode);

*const* [selected, setSelected] = useState("Dashboard");

    return (

        <Box *m*="20px">

            {*/\* HEADER \*/*}

            <Box *display*="flex" *justifyContent*="space-between" *alignItems*="center">

                <Header *title*="Landing" *subtitle*="Welcome to your Landing page" />

                <Box>

                    <Button

*sx*={{

                        backgroundColor: colors.blueAccent[700],

                        color: colors.grey[100],

                        fontSize: "14px",

                        fontWeight: "bold",

                        padding: "10px 20px",

                        }}

                    >

                        <DownloadOutlinedIcon *sx*={{ mr: "10px" }} />

                        Download

                    </Button>

                </Box>

            </Box>

            {*/\* GRID & CHARTS , GRID for whole page, 12 columns\*/*}

            <Box

*display*="grid"

*gridTemplateColumns*="repeat(12, 1fr)"

*gridAutoRows*="140px"

*gap*="20px"

            >

                {*/\* Row 1  \*/*}

                <Box

*gridColumn*="span 6"

*gridRow*="span 5"

*backgroundColor*={colors.primary[400]}

                >

                    <Typography

*variant*="h5"

*fontWeight*="600"

*sx*={{ padding: "30px 30px 0 30px" }}

                    >

                        Cart 1

                    </Typography>

                    <Box *height*="400px" *mt*="-100px">

                        <Box *display*="flex" *justifyContent*="center" *alignItems*="center">

                            <img

*alt*="profile-user"

*width*="full"

*height*="full"

*src*={`../../assets/room1.svg`}

*// style={{ cursor: "pointer", borderRadius: "50%" }}*

                            />

                        </Box>

                    </Box>

                    <Box *mt*={3} *position*="absolute" *bottom*="-80px" >

                        <Link *to*="/landing1/dashboard">

                            <Button

*sx*={{

                                backgroundColor: colors.blueAccent[700],

                                color: colors.grey[100],

                                fontSize: "14px",

                                fontWeight: "bold",

                                padding: "10px 20px",

                                }}

                            >

                                <DetailsIcon *sx*={{ mr: "10px" }} />

                                Details

                            </Button>

                        </Link>

                    </Box>

                </Box>

                <Box

*gridColumn*="span 6"

*gridRow*="span 5"

*backgroundColor*={colors.primary[400]}

                >

                    <Typography

*variant*="h5"

*fontWeight*="600"

*sx*={{ padding: "30px 30px 0 30px" }}

                    >

                        Cart 1

                    </Typography>

                    <Box *height*="400px" *mt*="-100px">

                        <Box *display*="flex" *justifyContent*="center" *alignItems*="center">

                            <img

*alt*="profile-user"

*width*="full"

*height*="full"

*src*={`../../assets/room1.svg`}

*// style={{ cursor: "pointer", borderRadius: "50%" }}*

                            />

                        </Box>

                    </Box>

                    <Box *mt*={3} *position*="absolute" *bottom*="-80px" >

                        <Link *to*="/landing1/dashboard">

                            <Button

*sx*={{

                                backgroundColor: colors.blueAccent[700],

                                color: colors.grey[100],

                                fontSize: "14px",

                                fontWeight: "bold",

                                padding: "10px 20px",

                                }}

                            >

                                <DetailsIcon *sx*={{ mr: "10px" }} />

                                Details

                            </Button>

                        </Link>

                    </Box>

                </Box>

            </Box>

        </Box>

    )

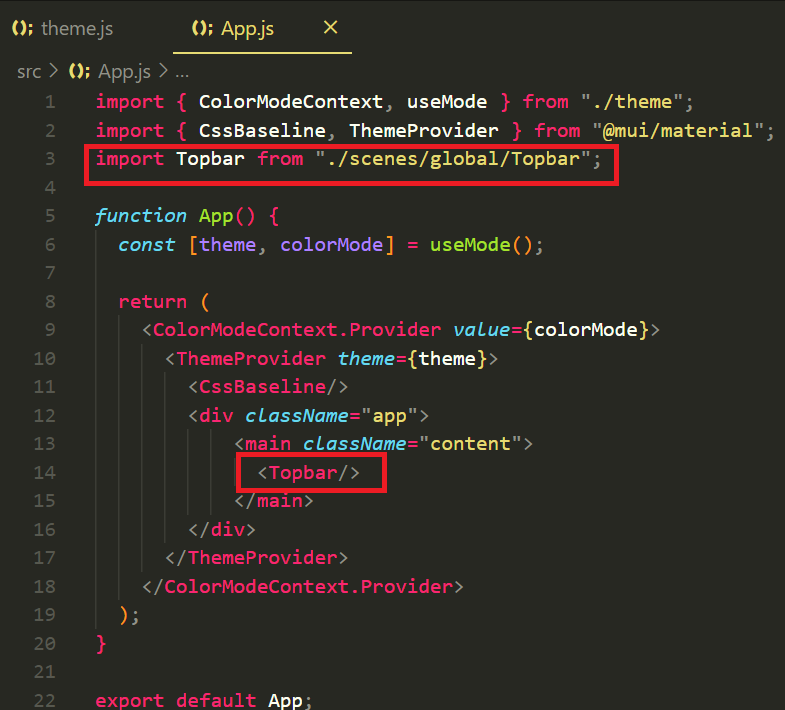
}

export default Landing;

## Folder global

### Cài đặt topbar

#### Set up trong App.js



#### Set up Topbar.jsx

import { Box, IconButton, useTheme } from "@mui/material";

import { useContext } from "react";

import { ColorModeContext, tokens } from "../../theme";

import InputBase from "@mui/material/InputBase";

import LightModeOutlinedIcon from "@mui/icons-material/LightModeOutlined";

import DarkModeOutlinedIcon from "@mui/icons-material/DarkModeOutlined";

import NotificationsOutlinedIcon from "@mui/icons-material/NotificationsOutlined";

import SettingsOutlinedIcon from "@mui/icons-material/SettingsOutlined";

import PersonOutlinedIcon from "@mui/icons-material/PersonOutlined";

import SearchIcon from "@mui/icons-material/Search";

*const* Topbar = () => {

*const* theme = useTheme();

*const* colors = tokens(theme.palette.mode);

*const* colorMode = useContext(ColorModeContext);

  return (

    <Box *display*="flex" *justifyContent*="space-between" *p*={2}>

      {*/\* SEARCH BAR \*/*}

      <Box

*display*="flex"

*backgroundColor*={colors.primary[400]}

*borderRadius*="3px"

      >

        <InputBase *sx*={{ ml: 2, flex: 1 }} *placeholder*="Search" />

        <IconButton *type*="button" *sx*={{ p: 1 }}>

          <SearchIcon />

        </IconButton>

      </Box>

      {*/\* ICONS \*/*}

      <Box *display*="flex">

        <IconButton *onClick*={colorMode.toggleColorMode}>

          {theme.palette.mode === "dark" ? (

            <DarkModeOutlinedIcon />

          ) : (

            <LightModeOutlinedIcon />

          )}

        </IconButton>

        <IconButton>

          <NotificationsOutlinedIcon />

        </IconButton>

        <IconButton>

          <SettingsOutlinedIcon />

        </IconButton>

        <IconButton>

          <PersonOutlinedIcon />

        </IconButton>

      </Box>

    </Box>

  );

};

export default Topbar;

### Cài đặt Sidebar

#### Dùng react-pro-sidebar

[react-pro-sidebar/README.md at master · azouaoui-med/react-pro-sidebar · GitHub](https://github.com/azouaoui-med/react-pro-sidebar/blob/master/README.md)

#### Cài đặt các component cho Sidebar

import { useState } from "react";

import { ProSidebar, Menu, MenuItem } from "react-pro-sidebar";

import { Box, IconButton, Typography, useTheme } from "@mui/material";

import { Link } from "react-router-dom";

import "react-pro-sidebar/dist/css/styles.css";

import { tokens } from "../../theme";

import HomeOutlinedIcon from "@mui/icons-material/HomeOutlined";

import PeopleOutlinedIcon from "@mui/icons-material/PeopleOutlined";

import ContactsOutlinedIcon from "@mui/icons-material/ContactsOutlined";

import ReceiptOutlinedIcon from "@mui/icons-material/ReceiptOutlined";

import PersonOutlinedIcon from "@mui/icons-material/PersonOutlined";

import CalendarTodayOutlinedIcon from "@mui/icons-material/CalendarTodayOutlined";

import HelpOutlineOutlinedIcon from "@mui/icons-material/HelpOutlineOutlined";

import BarChartOutlinedIcon from "@mui/icons-material/BarChartOutlined";

import PieChartOutlineOutlinedIcon from "@mui/icons-material/PieChartOutlineOutlined";

import TimelineOutlinedIcon from "@mui/icons-material/TimelineOutlined";

import MenuOutlinedIcon from "@mui/icons-material/MenuOutlined";

import MapOutlinedIcon from "@mui/icons-material/MapOutlined";

*const* Item = ({ *title*, *to*, *icon*, *selected*, *setSelected* }) => {

*const* theme = useTheme();

*const* colors = tokens(theme.palette.mode);

  return (

    <MenuItem

*active*={selected === title}

*style*={{

        color: colors.grey[100],

      }}

*onClick*={() => setSelected(title)}

*icon*={icon}

    >

      <Typography>{title}</Typography>

      <Link *to*={to} />

    </MenuItem>

  );

};

*const* Sidebar = () => {

*const* theme = useTheme();

*const* colors = tokens(theme.palette.mode);

*const* [isCollapsed, setIsCollapsed] = useState(false);

*const* [selected, setSelected] = useState("Dashboard");

  return (

    <Box

*sx*={{

        "& .pro-sidebar-inner": {

          background: `${colors.primary[400]} !important`,

        },

        "& .pro-icon-wrapper": {

          backgroundColor: "transparent !important",

        },

        "& .pro-inner-item": {

          padding: "5px 35px 5px 20px !important",

        },

        "& .pro-inner-item:hover": {

          color: "#868dfb !important",

        },

        "& .pro-menu-item.active": {

          color: "#6870fa !important",

        },

      }}

    >

      <ProSidebar *collapsed*={isCollapsed}>

        <Menu *iconShape*="square">

          {*/\* LOGO AND MENU ICON \*/*}

          <MenuItem

*onClick*={() => setIsCollapsed(!isCollapsed)}

*icon*={isCollapsed ? <MenuOutlinedIcon /> : undefined}

*style*={{

              margin: "10px 0 20px 0",

              color: colors.grey[100],

            }}

          >

            {!isCollapsed && (

              <Box

*display*="flex"

*justifyContent*="space-between"

*alignItems*="center"

*ml*="15px"

              >

                <Typography *variant*="h3" *color*={colors.grey[100]}>

                  ADMINIS

                </Typography>

                <IconButton *onClick*={() => setIsCollapsed(!isCollapsed)}>

                  <MenuOutlinedIcon />

                </IconButton>

              </Box>

            )}

          </MenuItem>

          {!isCollapsed && (

            <Box *mb*="25px">

              <Box *display*="flex" *justifyContent*="center" *alignItems*="center">

                <img

*alt*="profile-user"

*width*="100px"

*height*="100px"

*src*={`../../assets/user.png`}

*style*={{ cursor: "pointer", borderRadius: "50%" }}

                />

              </Box>

              <Box *textAlign*="center">

                <Typography

*variant*="h2"

*color*={colors.grey[100]}

*fontWeight*="bold"

*sx*={{ m: "10px 0 0 0" }}

                >

                  Ed Roh

                </Typography>

                <Typography *variant*="h5" *color*={colors.greenAccent[500]}>

                  VP Fancy Admin

                </Typography>

              </Box>

            </Box>

          )}

          <Box *paddingLeft*={isCollapsed ? undefined : "10%"}>

            <Item

*title*="Dashboard"

*to*="/"

*icon*={<HomeOutlinedIcon />}

*selected*={selected}

*setSelected*={setSelected}

            />

            <Typography

*variant*="h6"

*color*={colors.grey[300]}

*sx*={{ m: "15px 0 5px 20px" }}

            >

              Data

            </Typography>

            <Item

*title*="Manage Team"

*to*="/team"

*icon*={<PeopleOutlinedIcon />}

*selected*={selected}

*setSelected*={setSelected}

            />

            <Item

*title*="Contacts Information"

*to*="/contacts"

*icon*={<ContactsOutlinedIcon />}

*selected*={selected}

*setSelected*={setSelected}

            />

            <Item

*title*="Invoices Balances"

*to*="/invoices"

*icon*={<ReceiptOutlinedIcon />}

*selected*={selected}

*setSelected*={setSelected}

            />

            <Typography

*variant*="h6"

*color*={colors.grey[300]}

*sx*={{ m: "15px 0 5px 20px" }}

            >

              Pages

            </Typography>

            <Item

*title*="Profile Form"

*to*="/form"

*icon*={<PersonOutlinedIcon />}

*selected*={selected}

*setSelected*={setSelected}

            />

            <Item

*title*="Calendar"

*to*="/calendar"

*icon*={<CalendarTodayOutlinedIcon />}

*selected*={selected}

*setSelected*={setSelected}

            />

            <Item

*title*="FAQ Page"

*to*="/faq"

*icon*={<HelpOutlineOutlinedIcon />}

*selected*={selected}

*setSelected*={setSelected}

            />

            <Typography

*variant*="h6"

*color*={colors.grey[300]}

*sx*={{ m: "15px 0 5px 20px" }}

            >

              Charts

            </Typography>

            <Item

*title*="Bar Chart"

*to*="/bar"

*icon*={<BarChartOutlinedIcon />}

*selected*={selected}

*setSelected*={setSelected}

            />

            <Item

*title*="Pie Chart"

*to*="/pie"

*icon*={<PieChartOutlineOutlinedIcon />}

*selected*={selected}

*setSelected*={setSelected}

            />

            <Item

*title*="Line Chart"

*to*="/line"

*icon*={<TimelineOutlinedIcon />}

*selected*={selected}

*setSelected*={setSelected}

            />

            <Item

*title*="Geography Chart"

*to*="/geography"

*icon*={<MapOutlinedIcon />}

*selected*={selected}

*setSelected*={setSelected}

            />

          </Box>

        </Menu>

      </ProSidebar>

    </Box>

  );

};

export default Sidebar;

#### Giải thích

- isCollapsed là một state của useState quyết định xem sidebar có mở ra hay không

- Item là một function có các props truyền vào và sẽ trả về một Item với tên và icon được chỉnh theo các prop tương ứng và nhúng bên dưới là đường link tương ứng

### Cài đặt dashboard component dùng cho dashboard page

Dashboard sẽ có một url riêng

#### Giải thích:

##### Grid dùng cho page dashboard, box tổng thể cả page

- Tất cả các component chính sẽ được gói trong một component <Box></Box>, với các prop truyền vào để tạo ra các gird ở trong component to này <Box></Box> ví dụ như sau:

<Box

*display*="grid"

*gridTemplateColumns*="repeat(12, 1fr)"

*gridAutoRows*="140px"

*gap*="20px"

            >

            </Box>

- Các prop truyền vào trong component

+ *display*="grid" nghĩa là dùng grid system

+ *gridTemplateColumns*="repeat(12, 1fr)" nghĩa là dùng hệ thống 12 cột

+ *gridAutoRows*="140px" nghĩa là độ rộng mỗi cột là 140px

+ *gap*="20px" nghĩa là khoảng cách giữa hai cột là 20px

##### Một box code một phần trong box tổng

- Tạo một box trong một dòng ví dụ, ý nghĩa của các component như sau:

<Box

*gridColumn*="span 3"

*backgroundColor*={colors.primary[400]}

*display*="flex"

*alignItems*="center"

*justifyContent*="center"

        >

+ *gridColumn*="span 3" nghĩa là sẽ dùng 3 cột

#### Code:

import { Box } from "@mui/material";

import Header from "../../components/Header";

*const* Dashboard = () => {

    return (

        <Box *m*="20px">

            <Box *display*="flex" *justifyContent*="space-between" *alignItems*="center">

                <Header *title*="Dashboard" *subtitle*="Welcome to your dashboard" />

            </Box>

        </Box>

    )

}

export default Dashboard;

## Cài đặt team component dùng cho team page

Team sẽ có một page riêng

……………………………………………………………………………………………………………………………………………………………………………….

## Cài đặt form để điền truyền về back-end

Dùng formic [Material UI | Formik](https://formik.org/docs/examples/with-material-ui)

# Folder components

## Cài đặt tên Header cho mỗi page

### Cài đặt component Header tổng dùng cho tất cả các page mà import nó

Header component nằm trong folder components, nếu page nào muốn có header thì có thể import component header này vào và truyền vào các prop title và subtitle vào trong component

Code:

import { Typography, Box, useTheme } from "@mui/material";

import { tokens } from "../theme";

*const* Header = ({ *title*, *subtitle* }) => {

*const* theme = useTheme();

*const* colors = tokens(theme.palette.mode);

  return (

    <Box *mb*="30px">

      <Typography

*variant*="h2"

*color*={colors.grey[100]}

*fontWeight*="bold"

*sx*={{ m: "0 0 5px 0" }}

      >

        {title}

      </Typography>

      <Typography *variant*="h5" *color*={colors.greenAccent[400]}>

        {subtitle}

      </Typography>

    </Box>

  );

};

export default Header;

## Cài đặt component Control

Sử dụng thư viện react-gauge-chart

Code:

import React from "react";

import GaugeChart from "react-gauge-chart";

import { useState } from "react";

import { Box, Button, IconButton, Typography, useTheme } from "@mui/material";

import { tokens } from "../../theme";

*const* Control = React.memo(() => {

*const* theme = useTheme();

*const* colors = tokens(theme.palette.mode);

*const* [speed, setSpeed] = useState(0);

*const* Increase = () => {

        setSpeed(speed < 100 ? (speed + 1) : 100);

    };

*const* Decrease = () => {

        setSpeed(speed > 0 ? (speed - 1) : 0);

    };

    return (

        <Box *m*="20px">

        <GaugeChart *id*="gauge control"

*nrOfLevels*={20}

*percent*={speed/100}

*colors*={['#5BE12C', '#F5CD19', '#EA4228']}

*textColor*={"#888888"}

*formatTextValue*={*value* => value + "%"}

        />

        <Box *display*="flex" *justifyContent*="space-between" *alignItems*="center">

            <Button

*sx*={{

                backgroundColor: colors.blueAccent[700],

                color: colors.grey[100],

                fontSize: "14px",

                fontWeight: "bold",

                padding: "10px 20px",

                }}

*onClick*={Increase}

            >

                        {*/\* <DownloadOutlinedIcon sx={{ mr: "10px" }} /> \*/*}

                        +

            </Button>

            <Button

*sx*={{

                backgroundColor: colors.blueAccent[700],

                color: colors.grey[100],

                fontSize: "14px",

                fontWeight: "bold",

                padding: "10px 20px",

                }}

*onClick*={Decrease}

            >

                        {*/\* <DownloadOutlinedIcon sx={{ mr: "10px" }} /> \*/*}

                        -

            </Button>

        </Box>

        <Box *display*="flex" *justifyContent*="center" *alignItems*="center">

            <Button

*sx*={{

                backgroundColor: colors.blueAccent[700],

                color: colors.grey[100],

                fontSize: "14px",

                fontWeight: "bold",

                padding: "10px 20px",

                }}

*onClick*={

*async*(*e*) => {

*const* url\_sending\_set\_point = `http://127.0.0.1:8000/api/set/${speed}`;

*const* response = await fetch(url\_sending\_set\_point,{

                        'method':'GET',

                        headers: {

                            'Content-Type':'application/json',

                            }

                        });

*const* new\_data = await response.json();

                        alert(new\_data['data']);

                    }

                }

            >

                {*/\* <DownloadOutlinedIcon sx={{ mr: "10px" }} /> \*/*}

                Send

            </Button>

        {*/\* <button className="col" onClick={*

*async(e) => {*

*const url\_sending\_set\_point = `http://127.0.0.1:8000/api/set/${speed}`;*

*const response = await fetch(url\_sending\_set\_point,{*

*'method':'GET',*

*headers: {*

*'Content-Type':'application/json',*

*}*

*});*

*const new\_data = await response.json();*

*alert(new\_data['data']);*

*}*

*}>*

*SEND*

*</button> \*/*}

        </Box>

    </Box>

    );

}

)

export default Control;

Giải thích:

Hàm onClick của button là hàm gọi đến API của back-end kèm theo đó là một biến speed có nhiệm vụ truyền một bản tin điều khiển với thông số speed lên mqtt

## Cài đặt component LineChart.jsx

Dùng nico chart: [Home | nivo](https://nivo.rocks/)

### Giải thích:

### Code:

import { ResponsiveLine } from "@nivo/line";

import { useTheme } from "@mui/material";

import { tokens } from "../theme";

import { mockLineData as data } from "../data/mockData";

*const* LineChart = ({ *isCustomLineColors* = false, *isDashboard* = false }) => {

*const* theme = useTheme();

*const* colors = tokens(theme.palette.mode);

  return (

    <ResponsiveLine

*data*={data}

*theme*={{

        axis: {

          domain: {

            line: {

              stroke: colors.grey[100],

            },

          },

          legend: {

            text: {

              fill: colors.grey[100],

            },

          },

          ticks: {

            line: {

              stroke: colors.grey[100],

              strokeWidth: 1,

            },

            text: {

              fill: colors.grey[100],

            },

          },

        },

        legends: {

          text: {

            fill: colors.grey[100],

          },

        },

        tooltip: {

          container: {

            color: colors.primary[500],

          },

        },

      }}

*colors*={isDashboard ? { datum: "color" } : { scheme: "nivo" }} *// added*

*margin*={{ top: 50, right: 110, bottom: 50, left: 60 }}

*xScale*={{ type: "point" }}

*yScale*={{

        type: "linear",

        min: "auto",

        max: "auto",

        stacked: true,

        reverse: false,

      }}

*yFormat*=" >-.2f"

*curve*="catmullRom"

*axisTop*={null}

*axisRight*={null}

*axisBottom*={{

        orient: "bottom",

        tickSize: 0,

        tickPadding: 5,

        tickRotation: 0,

        legend: isDashboard ? undefined : "transportation", *// added*

        legendOffset: 36,

        legendPosition: "middle",

      }}

*axisLeft*={{

        orient: "left",

        tickValues: 5, *// added*

        tickSize: 3,

        tickPadding: 5,

        tickRotation: 0,

        legend: isDashboard ? undefined : "count", *// added*

        legendOffset: -40,

        legendPosition: "middle",

      }}

*enableGridX*={false}

*enableGridY*={false}

*pointSize*={8}

*pointColor*={{ theme: "background" }}

*pointBorderWidth*={2}

*pointBorderColor*={{ from: "serieColor" }}

*pointLabelYOffset*={-12}

*useMesh*={true}

*legends*={[

        {

          anchor: "bottom-right",

          direction: "column",

          justify: false,

          translateX: 100,

          translateY: 0,

          itemsSpacing: 0,

          itemDirection: "left-to-right",

          itemWidth: 80,

          itemHeight: 20,

          itemOpacity: 0.75,

          symbolSize: 12,

          symbolShape: "circle",

          symbolBorderColor: "rgba(0, 0, 0, .5)",

          effects: [

            {

              on: "hover",

              style: {

                itemBackground: "rgba(0, 0, 0, .03)",

                itemOpacity: 1,

              },

            },

          ],

        },

      ]}

    />

  );

};

export default LineChart;

## Cài đặt component BarChart.jsx

Dùng nico chart: [Home | nivo](https://nivo.rocks/)

Ở đâu muốn dùng barchart thì chỉ cần import component này

Code:

import { useTheme } from "@mui/material";

import { ResponsiveBar } from "@nivo/bar";

import { tokens } from "../theme";

import { mockBarData as data } from "../data/mockData";

*const* BarChart = ({ *isDashboard* = false }) => {

*const* theme = useTheme();

*const* colors = tokens(theme.palette.mode);

  return (

    <ResponsiveBar

*data*={data}

*theme*={{

*// added*

        axis: {

          domain: {

            line: {

              stroke: colors.grey[100],

            },

          },

          legend: {

            text: {

              fill: colors.grey[100],

            },

          },

          ticks: {

            line: {

              stroke: colors.grey[100],

              strokeWidth: 1,

            },

            text: {

              fill: colors.grey[100],

            },

          },

        },

        legends: {

          text: {

            fill: colors.grey[100],

          },

        },

      }}

*keys*={["hot dog", "burger", "sandwich", "kebab", "fries", "donut"]}

*indexBy*="country"

*margin*={{ top: 50, right: 130, bottom: 50, left: 60 }}

*padding*={0.3}

*valueScale*={{ type: "linear" }}

*indexScale*={{ type: "band", round: true }}

*colors*={{ scheme: "nivo" }}

*defs*={[

        {

          id: "dots",

          type: "patternDots",

          background: "inherit",

          color: "#38bcb2",

          size: 4,

          padding: 1,

          stagger: true,

        },

        {

          id: "lines",

          type: "patternLines",

          background: "inherit",

          color: "#eed312",

          rotation: -45,

          lineWidth: 6,

          spacing: 10,

        },

      ]}

*borderColor*={{

        from: "color",

        modifiers: [["darker", "1.6"]],

      }}

*axisTop*={null}

*axisRight*={null}

*axisBottom*={{

        tickSize: 5,

        tickPadding: 5,

        tickRotation: 0,

        legend: isDashboard ? undefined : "country", *// changed*

        legendPosition: "middle",

        legendOffset: 32,

      }}

*axisLeft*={{

        tickSize: 5,

        tickPadding: 5,

        tickRotation: 0,

        legend: isDashboard ? undefined : "food", *// changed*

        legendPosition: "middle",

        legendOffset: -40,

      }}

*enableLabel*={false}

*labelSkipWidth*={12}

*labelSkipHeight*={12}

*labelTextColor*={{

        from: "color",

        modifiers: [["darker", 1.6]],

      }}

*legends*={[

        {

          dataFrom: "keys",

          anchor: "bottom-right",

          direction: "column",

          justify: false,

          translateX: 120,

          translateY: 0,

          itemsSpacing: 2,

          itemWidth: 100,

          itemHeight: 20,

          itemDirection: "left-to-right",

          itemOpacity: 0.85,

          symbolSize: 20,

          effects: [

            {

              on: "hover",

              style: {

                itemOpacity: 1,

              },

            },

          ],

        },

      ]}

*role*="application"

*barAriaLabel*={*function* (*e*) {

        return e.id + ": " + e.formattedValue + " in country: " + e.indexValue;

      }}

    />

  );

};

export default BarChart;

## Cài đặt component GeographyChart.jsx

Nếu ở đâu muốn dùng component này thì import component này ra

## Giải thích:

### prop feature

geoFeatures là những feature cần thiết để mà cái bản đồ được tạo ra được

### prop data

data là data để bản đồ được vẽ data theo

### prop theme

- là những cài đặt về màu sắc mà mình sử dụng cho bản đồ

- theme mình sử dụng sẽ như sau:

*theme*={{

        axis: {

          domain: {

            line: {

              stroke: colors.grey[100],

            },

          },

          legend: {

            text: {

              fill: colors.grey[100],

            },

          },

          ticks: {

            line: {

              stroke: colors.grey[100],

              strokeWidth: 1,

            },

            text: {

              fill: colors.grey[100],

            },

          },

        },

        legends: {

          text: {

            fill: colors.grey[100],

          },

        },

      }}

### prop projectionScale

Để điều chỉnh độ to lớn scale của bản đồ

### prop lengend

Nếu mà ở trong page dashboard thì sẽ không có legend còn nếu ở page riêng thì có

## Code:

import { useTheme } from "@mui/material";

import { ResponsiveChoropleth } from "@nivo/geo";

import { geoFeatures } from "../data/mockGeoFeatures";

import { tokens } from "../theme";

import { mockGeographyData as data } from "../data/mockData";

*const* GeographyChart = ({ *isDashboard* = false }) => {

*const* theme = useTheme();

*const* colors = tokens(theme.palette.mode);

  return (

    <ResponsiveChoropleth

*data*={data}

*theme*={{

        axis: {

          domain: {

            line: {

              stroke: colors.grey[100],

            },

          },

          legend: {

            text: {

              fill: colors.grey[100],

            },

          },

          ticks: {

            line: {

              stroke: colors.grey[100],

              strokeWidth: 1,

            },

            text: {

              fill: colors.grey[100],

            },

          },

        },

        legends: {

          text: {

            fill: colors.grey[100],

          },

        },

      }}

*features*={geoFeatures.features}

*margin*={{ top: 0, right: 0, bottom: 0, left: 0 }}

*domain*={[0, 1000000]}

*unknownColor*="#666666"

*label*="properties.name"

*valueFormat*=".2s"

*projectionScale*={isDashboard ? 40 : 150}

*projectionTranslation*={isDashboard ? [0.49, 0.6] : [0.5, 0.5]}

*projectionRotation*={[0, 0, 0]}

*borderWidth*={1.5}

*borderColor*="#ffffff"

*legends*={

        !isDashboard

          ? [

              {

                anchor: "bottom-left",

                direction: "column",

                justify: true,

                translateX: 20,

                translateY: -100,

                itemsSpacing: 0,

                itemWidth: 94,

                itemHeight: 18,

                itemDirection: "left-to-right",

                itemTextColor: colors.grey[100],

                itemOpacity: 0.85,

                symbolSize: 18,

                effects: [

                  {

                    on: "hover",

                    style: {

                      itemTextColor: "#ffffff",

                      itemOpacity: 1,

                    },

                  },

                ],

              },

            ]

          : undefined

      }

    />

  );

};

export default GeographyChart;