import React, { memo } from "react";

//HOME PAGE NAVBAR

import PropTypes from "prop-types";

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

import {

  AppBar,

  Toolbar,

  Typography,

  Button,

  Hidden,

  IconButton,

  withStyles

} from "@material-ui/core";

import MenuIcon from "@material-ui/icons/Menu";

import HomeIcon from "@material-ui/icons/Home";

import HowToRegIcon from "@material-ui/icons/HowToReg";

import LockOpenIcon from "@material-ui/icons/LockOpen";

//import BookIcon from "@material-ui/icons/Book";

import NavigationDrawer from "../../../shared/components/NavigationDrawer";

import Logo from "../../../logged\_out/dummy\_data/images/TM2D.jpg";

const styles = theme => ({

  appBar: {

    boxShadow: theme.shadows[6],

    backgroundColor: theme.palette.common.white

  },

  toolbar: {

    display: "flex",

    justifyContent: "space-between"

  },

  menuButtonText: {

    fontSize: theme.typography.body1.fontSize,

    fontWeight: theme.typography.h6.fontWeight

  },

  brandText: {

    fontFamily: "'Baloo Bhaijaan', cursive",

    fontWeight: 400

  },

  noDecoration: {

    textDecoration: "none !important"

  }

});

function NavBar(props) {

  const {

    classes,

    openRegisterDialog,

    openLoginDialog,

    handleMobileDrawerOpen,

    handleMobileDrawerClose,

    mobileDrawerOpen,

    selectedTab

  } = props;

  const menuItems = [

    {

      link: "/",

      name: "Home",

      icon: <HomeIcon className="text-white" />

    },

    {

      name: "Register",

      onClick: openRegisterDialog,

      icon: <HowToRegIcon className="text-white" />

    },

    {

      name: "Test Tips",

      onClick: openLoginDialog,

      icon: <LockOpenIcon className="text-white" />

    }

  ];

  return (

    //HOME PAGE NAVBAR

    <div className={classes.root}>

      <AppBar position="fixed" className={classes.appBar}>

        <Toolbar className={classes.toolbar}>

          <div>

            <img src={Logo} alt="Logo" />

          </div>

          <div>

            <Typography

              variant="h4"

              className={classes.brandText}

              display="inline"

              color="secondary"

            >

              TEACH ME 2 DRIVE

            </Typography>

          </div>

          <div>

            <Hidden mdUp>

              <IconButton

                className={classes.menuButton}

                onClick={handleMobileDrawerOpen}

                aria-label="Open Navigation"

              >

                <MenuIcon color="primary" />

              </IconButton>

            </Hidden>

            <Hidden smDown>

              {menuItems.map(element => {

                if (element.link) {

                  return (

                    <Link

                      key={element.name}

                      to={element.link}

                      className={classes.noDecoration}

                      onClick={handleMobileDrawerClose}

                    >

                      <Button

                        color="secondary"

                        size="large"

                        classes={{ text: classes.menuButtonText }}

                      >

                        {element.name}

                      </Button>

                    </Link>

                  );

                }

                return (

                  <Button

                    color="secondary"

                    size="large"

                    onClick={element.onClick}

                    classes={{ text: classes.menuButtonText }}

                    key={element.name}

                  >

                    {element.name}

                  </Button>

                );

              })}

            </Hidden>

          </div>

        </Toolbar>

      </AppBar>

      <NavigationDrawer

        menuItems={menuItems}

        anchor="right"

        open={mobileDrawerOpen}

        selectedItem={selectedTab}

        onClose={handleMobileDrawerClose}

      />

    </div>

  );

}

NavBar.propTypes = {

  classes: PropTypes.object.isRequired,

  handleMobileDrawerOpen: PropTypes.func,

  handleMobileDrawerClose: PropTypes.func,

  mobileDrawerOpen: PropTypes.bool,

  selectedTab: PropTypes.string,

  //openRegisterDialog: PropTypes.func.isRequired,

  openRegisterForm: PropTypes.func.isRequired,

  openLoginDialog: PropTypes.func.isRequired

};

export default withStyles(styles, { withTheme: true })(memo(NavBar));

import React, { Fragment, useRef, useCallback, useState } from "react";

//LOGIN PAGE NAVBAR

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

import PropTypes from "prop-types";

import classNames from "classnames";

import {

  AppBar,

  Toolbar,

  Typography,

  Drawer,

  List,

  IconButton,

  ListItem,

  ListItemIcon,

  ListItemText,

  Hidden,

  Tooltip,

  Box,

  withStyles,

  isWidthUp,

  withWidth

} from "@material-ui/core";

import DashboardIcon from "@material-ui/icons/Dashboard";

import ImageIcon from "@material-ui/icons/Image";

//import AccountBalanceIcon from "@material-ui/icons/AccountBalance";

import PowerSettingsNewIcon from "@material-ui/icons/PowerSettingsNew";

import MenuIcon from "@material-ui/icons/Menu";

//import SupervisorAccountIcon from "@material-ui/icons/SupervisorAccount";

//import MessagePopperButton from "./MessagePopperButton";

//import SideDrawer from "./SideDrawer";

//import Balance from "./Balance";

import NavigationDrawer from "../../../shared/components/NavigationDrawer";

import Logo from "../../dummy\_data/images/TM2D.jpg";

const styles = theme => ({

  appBar: {

    boxShadow: theme.shadows[6],

    backgroundColor: theme.palette.common.white,

    transition: theme.transitions.create(["width", "margin"], {

      easing: theme.transitions.easing.sharp,

      duration: theme.transitions.duration.leavingScreen

    }),

    [theme.breakpoints.down("xs")]: {

      width: "100%",

      marginLeft: 0

    }

  },

  appBarToolbar: {

    display: "flex",

    justifyContent: "space-between",

    paddingLeft: theme.spacing(1),

    paddingRight: theme.spacing(1),

    [theme.breakpoints.up("sm")]: {

      paddingLeft: theme.spacing(2),

      paddingRight: theme.spacing(2)

    },

    [theme.breakpoints.up("md")]: {

      paddingLeft: theme.spacing(3),

      paddingRight: theme.spacing(3)

    },

    [theme.breakpoints.up("lg")]: {

      paddingLeft: theme.spacing(4),

      paddingRight: theme.spacing(4)

    }

  },

  accountAvatar: {

    backgroundColor: theme.palette.secondary.main,

    height: 24,

    width: 24,

    marginLeft: theme.spacing(2),

    marginRight: theme.spacing(2),

    [theme.breakpoints.down("xs")]: {

      marginLeft: theme.spacing(1.5),

      marginRight: theme.spacing(1.5)

    }

  },

  drawerPaper: {

    height: "100%vh",

    whiteSpace: "nowrap",

    border: 0,

    width: theme.spacing(7),

    overflowX: "hidden",

    marginTop: theme.spacing(8),

    [theme.breakpoints.up("sm")]: {

      width: theme.spacing(9)

    },

    backgroundColor: theme.palette.common.black

  },

  smBordered: {

    [theme.breakpoints.down("xs")]: {

      borderRadius: "50% !important"

    }

  },

  menuLink: {

    textDecoration: "none",

    color: theme.palette.text.primary

  },

  iconListItem: {

    width: "auto",

    borderRadius: theme.shape.borderRadius,

    paddingTop: 11,

    paddingBottom: 11,

    marginLeft: theme.spacing(1),

    marginRight: theme.spacing(1)

  },

  textPrimary: {

    color: theme.palette.primary.main

  },

  mobileItemSelected: {

    backgroundColor: `${theme.palette.primary.main} !important`

  },

  brandText: {

    fontFamily: "'Baloo Bhaijaan', cursive",

    fontWeight: 400

  },

  username: {

    paddingLeft: 0,

    paddingRight: theme.spacing(2)

  },

  justifyCenter: {

    justifyContent: "center"

  },

  permanentDrawerListItem: {

    justifyContent: "center",

    paddingTop: theme.spacing(2),

    paddingBottom: theme.spacing(2)

  }

});

function NavBar(props) {

  const { selectedTab, classes, width } = props;

  // Will be use to make website more accessible by screen readers

  const links = useRef([]);

  const [isMobileOpen, setIsMobileOpen] = useState(false);

  //const [setIsSideDrawerOpen] = useState(false);

  const openMobileDrawer = useCallback(() => {

    setIsMobileOpen(true);

  }, [setIsMobileOpen]);

  const closeMobileDrawer = useCallback(() => {

    setIsMobileOpen(false);

  }, [setIsMobileOpen]);

  const menuItems = [

    {

      link: "/c/dashboard",

      name: "Drive Test Tips",

      onClick: closeMobileDrawer,

      icon: {

        desktop: (

          <DashboardIcon

            className={

              selectedTab === "Dashboard" ? classes.textPrimary : "text-white"

            }

            fontSize="small"

          />

        ),

        mobile: <DashboardIcon className="text-white" />

      }

    },

    {

      link: "/c/posts",

      name: "Successful Learners",

      onClick: closeMobileDrawer,

      icon: {

        desktop: (

          <ImageIcon

            className={

              selectedTab === "Posts" ? classes.textPrimary : "text-white"

            }

            fontSize="small"

          />

        ),

        mobile: <ImageIcon className="text-white" />

      }

    },

    {

      link: "/",

      name: "Logout",

      icon: {

        desktop: (

          <PowerSettingsNewIcon className="text-white" fontSize="small" />

        ),

        mobile: <PowerSettingsNewIcon className="text-white" />

      }

    }

  ];

  return (

    <Fragment>

      <AppBar position="sticky" className={classes.appBar}>

        <Toolbar className={classes.appBarToolbar}>

          <div>

            <img src={Logo} alt="Logo" />

          </div>

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

            <Hidden smUp>

              <Box mr={1}>

                <IconButton

                  aria-label="Open Navigation"

                  onClick={openMobileDrawer}

                  color="primary"

                >

                  <MenuIcon />

                </IconButton>

              </Box>

            </Hidden>

          </Box>

          <Box

            display="flex"

            justifyContent="flex-end"

            alignItems="center"

            width="100%"

          >

            {/\*<MessagePopperButton messages={messages} /> \*/}

            <ListItem

              disableGutters

              className={classNames(classes.iconListItem, classes.smBordered)}

            >

              {isWidthUp("sm", width) && (

                <ListItemText

                  className={classes.username}

                />

              )}

            </ListItem>

          </Box>

          <IconButton

            color="primary"

            aria-label="Open Sidedrawer"

          >

            <Typography color="textPrimary">Username</Typography>

          </IconButton>

          {/\*<SideDrawer open={isSideDrawerOpen} onClose={closeDrawer} /> \*/}

        </Toolbar>

      </AppBar>

      <Hidden xsDown>

        <Drawer //  both drawers can be combined into one for performance

          variant="permanent"

          classes={{

            paper: classes.drawerPaper

          }}

          open={false}

        >

          <List>

            {menuItems.map((element, index) => (

              <Link

                to={element.link}

                className={classes.menuLink}

                onClick={element.onClick}

                key={index}

                ref={node => {

                  links.current[index] = node;

                }}

              >

                <Tooltip

                  title={element.name}

                  placement="right"

                  key={element.name}

                >

                  <ListItem

                    selected={selectedTab === element.name}

                    button

                    divider={index !== menuItems.length - 1}

                    className={classes.permanentDrawerListItem}

                    onClick={() => {

                      links.current[index].click();

                    }}

                    aria-label={

                      element.name === "Logout"

                        ? "Logout"

                        : `Go to ${element.name}`

                    }

                  >

                    <ListItemIcon className={classes.justifyCenter}>

                      {element.icon.desktop}

                    </ListItemIcon>

                  </ListItem>

                </Tooltip>

              </Link>

            ))}

          </List>

        </Drawer>

      </Hidden>

      <NavigationDrawer

        menuItems={menuItems.map(element => ({

          link: element.link,

          name: element.name,

          icon: element.icon.mobile,

          onClick: element.onClick

        }))}

        anchor="left"

        open={isMobileOpen}

        selectedItem={selectedTab}

        onClose={closeMobileDrawer}

      />

    </Fragment>

  );

}

NavBar.propTypes = {

  messages: PropTypes.arrayOf(PropTypes.object).isRequired,

  selectedTab: PropTypes.string.isRequired,

  width: PropTypes.string.isRequired,

  classes: PropTypes.object.isRequired,

  openAddBalanceDialog: PropTypes.func.isRequired

};

export default withWidth()(withStyles(styles, { withTheme: true })(NavBar));