```
import React, { useState, useEffect } from "react";
import { useSelector, useDispatch } from "react-redux";
import { Layout } from "antd";
import Logo from "./Logo";
import NavNotification from "./NavNotification";
import NavProfile from "./NavProfile";
//import NavLanguage from "./NavLanguage";
//import NavPanel from "./NavPanel";
import NavSearch from "./NavSearch";
import { toggleCollapsedNav, onMobileNavToggle } from "store/slices/themeSlice";
import {
  NAV_TYPE_TOP,
  SIDE_NAV_COLLAPSED_WIDTH,
  SIDE_NAV_WIDTH,
} from "constants/ThemeConstant";
import utils from "utils";
import { BRANCH_ID } from "constants/AuthConstant";
import { getProfileApi, setBranchId } from "store/slices/authSlice";
import { isEmpty } from "lodash";
import { getAllEmployeeCate, getHospitalApi } from "store/slices/categorySlice";
import { getBagetApi } from "store/slices/categorySlice";
import MediSelect from "components/Custom/MediSelect";
import { OfficeSvg } from "components/Icon";
import NavAppStore from "./NavAppStore";
import NavSign from "./NavSign";

const { Header } = Layout;

export const HeaderNav = (props) => {
  const { isMobile } = props;
  const [searchActive, setSearchActive] = useState(false);
  const dispatch = useDispatch();

  const navCollapsed = useSelector((state) => state.theme.navCollapsed);
  const mobileNav = useSelector((state) => state.theme.mobileNav);
  const navType = useSelector((state) => state.theme.navType);
  const headerNavColor = useSelector((state) => state.theme.headerNavColor);
  const currentTheme = useSelector((state) => state.theme.currentTheme);
  const [branchInfoList, setBranchInfoList] = useState([]);
  const { branchId, profile } = useSelector((state) => state.auth);

  useEffect(() => {
    dispatch(getProfileApi());
    dispatch(getHospitalApi());
    dispatch(getBagetApi());
    dispatch(getAllEmployeeCate());
  }, [dispatch]);

  useEffect(() => {
    const arrBranch = profile?.branchInfos?.filter((i) => i.allowAccess);
    setBranchInfoList(arrBranch || []);
    if (isEmpty(branchId) && !isEmpty(arrBranch)) {
      sessionStorage.setItem(BRANCH_ID, arrBranch[0]?.branchId);
      dispatch(setBranchId(arrBranch[0]?.branchId));
    }
    // eslint-disable-next-line react-hooks/exhaustive-deps
  }, [dispatch, profile]);

  const onSearchClose = () => {
    setSearchActive(false);
  };

  const onToggle = () => {
    if (!isMobile) {
      dispatch(toggleCollapsedNav(!navCollapsed));
    } else {
      dispatch(onMobileNavToggle(!mobileNav));
    }
  };

  const isNavTop = navType === NAV_TYPE_TOP ? true : false;
```

```
const mode = () => {
  if (!headerNavColor) {
    return utils.getColorContrast(
      currentTheme === "dark" ? "#00000" : "#ffffff"
    );
  }
  return utils.getColorContrast(headerNavColor);
};

const navMode = mode();

const getNavWidth = () => {
  if (isNavTop || isMobile) {
    return "0px";
  }
  if (navCollapsed) {
    return `${SIDE_NAV_COLLAPSED_WIDTH}px`;
  } else {
    return `${SIDE_NAV_WIDTH}px`;
  }
};

useEffect(() => {
  if (!isMobile) {
    onSearchClose();
  }
});

return (
  <Header
    className={`app-header ${navMode}`}
    style={{ backgroundColor: headerNavColor }}
  >
    <div className={`app-header-wrapper ${isNavTop ? "layout-top-nav" : ""}`}>
      <Logo logoType={navMode} />
      <div className="nav" style={{ width: `calc(100% - ${getNavWidth()})` }}>
        <div className="nav-left">
          <div className="nav-item" onClick={onToggle}>
            <div className="d-flex align-items-center">
              <img src="/img/icons/menu-4 1.svg" title="logo" alt="logo" />
              {/* {navCollapsed || isMobile ? (
                <MenuUnfoldOutlined className="nav-icon" />
              ) : (
                <MenuFoldOutlined className="nav-icon" />
              )} */}
            </div>
          </div>
          <ul className="ant-menu ant-menu-root ant-menu-horizontal">
            <MediSelect
              className="select-branch"
              optionLabelProp="title"
              value={branchId}
              showSearch
              options={branchInfoList?.map((x) => ({
                title: (
                  <>
                    <OfficeSvg className="mr-2" /> {x.branchName}
                  </>
                ),
                label: x.branchName,
                value: x.branchId,
              }))}
              filterOption={(input, option) =>
                (option?.label ?? "")
                  .toLowerCase()
                  .includes(input.toLowerCase())
              }
              onChange={(value) => {
                sessionStorage.setItem(BRANCH_ID, value);
                localStorage.setItem(BRANCH_ID, value);
                dispatch(setBranchId(value));
              }}
              placeholder="Chọn chi nhánh"
```

```
              style={{ width: isMobile ? 200 : 300 }}
            />
          </ul>
        </div>
        <div className="nav-right">
          <NavAppStore />
          <NavSign />
          <NavNotification />
          {/* <NavLanguage /> */}
          {/* <NavPanel direction={direction} /> */}
          <NavProfile />
        </div>
        <NavSearch active={searchActive} close={onSearchClose} />
      </div>
    </div>
  </Header>
 );
};

export default HeaderNav;
```