```
import React, { useState, useRef } from 'react';
import {
        DashboardOutlined,
        AppstoreOutlined,
        AntDesignOutlined,
        FileTextOutlined,
        SearchOutlined
} from '@ant-design/icons';
import { Link } from "react-router-dom";
import { AutoComplete, Input } from 'antd';
import IntlMessage from 'components/util-components/IntlMessage';
import navigationConfig from "configs/NavigationConfig";

function getOptionList (navigationTree, optionTree) {
        optionTree = optionTree ? optionTree : [];
        for ( const navItem of navigationTree ) {
                if(navItem.submenu.length === 0) {
                        optionTree.push(navItem)
                }
                if(navItem.submenu.length > 0) {
                        getOptionList(navItem.submenu, optionTree);
                }
        }
        return optionTree
}

const optionList = getOptionList(navigationConfig)

const getCategoryIcon = category => {
        switch (category) {
                case 'dashboards':
                        return <DashboardOutlined className="text-success"/>;
                case 'apps':
                        return <AppstoreOutlined className="text-danger"/>;
                case 'components':
                        return <AntDesignOutlined className="text-primary"/>;
                case 'extra':
                        return <FileTextOutlined className="text-warning"/>;
                default:
                        return null;
        }
}

const searchResult = () => optionList.map((item) => {
        const category = item.key.split('-')[0]
        return {
                value: item.path,
                label: (
                        <Link to={item.path}>
                                <div className="search-list-item">
                                        <div className="icon">
                                                {getCategoryIcon(category)}
                                        </div>
                                        <div>
                                                <div className="font-weight-semibold"><IntlMessage id={item.title} /></div>
                                                <div className="font-size-sm text-muted">{category} </div>
                                        </div>
                                </div>
                        </Link>
                ),
        };
});

const SearchInput = props => {
        const { active, close, isMobile, mode } = props
        const [value, setValue] = useState('');
        const [options, setOptions] = useState([])
        const inputRef = useRef(null);

        const onSelect = () => {
                setValue('')
                setOptions([])
                if(close) {
                        close()
                }
  };

        const onSearch = searchText => {
                setValue(searchText)
```

```
                setOptions(!searchText ? [] : searchResult(searchText))
        };

        const autofocus = () => {
                inputRef.current.focus();
        }

        if(active) {
                autofocus()
        }

        return (
                <AutoComplete
                        ref={inputRef}
                        className={`nav-search-input ${isMobile? 'is-mobile' : ''} ${mode === 'light' ? 'light' : ''}`}
                        popupClassName="nav-search-dropdown"
                        options={options}
                        onSelect={onSelect}
                        onSearch={onSearch}
                        value={value}
                        filterOption={(inputValue, option) =>
                                option.value.toUpperCase().indexOf(inputValue.toUpperCase()) !== -1
                        }
                >
                        <Input placeholder="Search..."  prefix={<SearchOutlined className="mr-0" />} />
                </AutoComplete>
        )
}

export default SearchInput
```