```
import React, { Fragment, useEffect, useState } from "react";
import { Button, Select, Popover, Badge } from "antd";
import { PlusOutlined } from "@ant-design/icons";
import MediSelect from "components/Custom/MediSelect";

const FilterMoreView = (props) => {
    const { children, dataSource, onClearFilter, onSearchHandle } = props;
    const [formData, setFormData] = useState({ ...dataSource });
    const [open, setOpen] = useState(false);
    const [filterCount, setFilterCount] = useState(0);

    useEffect(() => {
        setFormData({ ...dataSource });
    }, [dataSource]);

    useEffect(() => {
        // Đếm số lượng các trường dữ liệu có giá trị trong children
        if (children) {
            let count = 0;
            if (children.length > 0) {
                children.forEach((item) => {
                    const { "data-field": dataField } = item.props;
                    if (dataSource[dataField] || dataSource[dataField] === 0) {
                        count++;
                    }
                });
            } else {
                const { "data-field": dataField } = children.props;
                if (dataSource[dataField]) {
                    count++;
                }
            }

            setFilterCount(count);
        }

        // eslint-disable-next-line react-hooks/exhaustive-deps
    }, [dataSource]);

    const onItemValueChanged = (child) => {
        const handleValueChange = (value, dataField) => {
            setFormData({
                ...formData,
                [dataField]: value,
            });
        };

        if (child.type === Select || child.type === MediSelect) {
            return (value) => {
                handleValueChange(value, child.props["data-field"]);
            };
        }
        //else if to-do
        else {
            return (e) => {
                let { value } = e.target;
                handleValueChange(value, child.props["data-field"]);
            };
        }
    };

    const renderChildren = (item) => {
        if (!item) return null;

        const { "data-field": dataField, label } = item.props;
        let child;

        child = React.cloneElement(item, {
            value: formData[dataField],
            onChange: onItemValueChanged(item),
        });
```

```
    return (
        <Fragment key={item.key}>
            <div className={label ? "mb-4" : "mb-3"}>
                {label && <label className="d-block mb-1">{label}</label>}
                {child}
            </div>
        </Fragment>
    );
};

const hide = () => {
    setOpen(false);
};

const handleOpenChange = (newOpen) => {
    setOpen(newOpen);
};

const onClickHandle = () => {
    onSearchHandle(formData);
    hide();
};

const onClearClickHandle = () => {
    onClearFilter();
    hide();
};

const content = (
    <div>
        <div>
            {React.Children.map(children, (childItem) => renderChildren(childItem))}
        </div>
        <div className="d-flex mt-3">
            <Button
                type="link"
                className="px-0 mr-auto"
                danger
                onClick={onClearClickHandle}
            >
                Bỏ lọc
            </Button>
            <Button type="primary" ghost style={{ width: "86px" }} onClick={hide}>
                Hủy
            </Button>
            <Button
                type="primary"
                className="ml-2"
                style={{ width: "86px" }}
                onClick={onClickHandle}
            >
                Áp dụng
            </Button>
        </div>
    </div>
);

return (
    <Popover
        placement="bottomRight"
        content={content}
        trigger="click"
        open={open}
        onOpenChange={handleOpenChange}
    >
        <Button
            type="link"
            shape="round"
            className={
                filterCount > 0 ? "filter-showbtn filtered" : "filter-showbtn"
            }
            icon={<PlusOutlined style={{ fontSize: "11px" }} />}
        >
```

```
                Thêm lọc{" "}
                {filterCount > 0 && (
                    <Badge className="ml-1" count={filterCount} size="small"></Badge>
                )}
            </Button>
        </Popover>
    );
};

export default FilterMoreView;
```