```
import React, { Fragment, useEffect, useState } from "react";
import { DatePicker, Input, Select, Row, Col } from "antd";
import Flex from "components/shared-components/Flex";
import { isEmpty } from "lodash";
import { SearchOutlined } from "@ant-design/icons";
import { MediSelect } from "components/Custom";

const FilterComponent = (props) => {
  const {
    children,
    datasource,
    onSearchImmediate,
    // onClearFilter,
    renderFilterMoreView,
    renderInputSearch,
    renderDatePicker,
    placeholderSearh = "Tìm kiếm",
    placeholderTimeRange = ["Từ ngày", "đến ngày"]
  } = props;

  const dateFormat = "DD/MM/YYYY";

  const [formData, setFormData] = useState({ ...datasource });

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

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

    if (child.type === Select || child.type === DatePicker || 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 onChangeSearch = (text) => {
    let newData;
    newData = {
      ...formData,
      searchText: text,
    };

    setFormData(newData);

    if (onSearchImmediate) {
      onSearchImmediate(newData);
    }
  };

  const handleSearchChange = (e) => {
    const { value } = e.target;
    if (!value) {
```

```
    onChangeSearch(value);
  }
};

const onDefaultItemValueChange = (field) => (e) => {
  let newData;

  if (field === "DatePicker") {
    const [startDate, endDate] = isEmpty(e)
      ? [null, null]
      : [e[0].toISOString(), e[1].toISOString()];

    newData = {
      ...formData,
      startDate,
      endDate,
    };
  } else {
    const { target } = e;

    newData = {
      ...formData,
      [field]: target.value,
    };
  }

  setFormData(newData);

  if (onSearchImmediate) {
    onSearchImmediate(newData);
  }
};

//Custom render children
const renderChildren = (item) => {
  if (!item) return null;

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

  //Add event listener to children
  let child;

  child = React.cloneElement(item, {
    value: formData[dataField],
    className: "mb-1",
    onChange: onItemValueChanged(item),
  });

  //add custom title
  if (label && label.trim().length > 0) {
    return (
      <Fragment key={item.key}>
        <Col>
          <label className="d-block mb-1">{label}</label>
          {child}
        </Col>
      </Fragment>
    );
  }

  return (
    <Fragment key={item.key}>
      <Col>{child}</Col>
    </Fragment>
  );
};

return (
  <Flex>
    <Row gutter={16} align="bottom">
      {renderInputSearch && (
        <Col className="mb-1">
          <Input
            type="string"
```

```
            key="searchText"
            data-field="searchText"
            defaultValue={formData.searchText}
            placeholder={placeholderSearh}
            allowClear
            style={{ width: "224px" }}
            prefix={<SearchOutlined />}
            onPressEnter={onDefaultItemValueChange("searchText")}
            onChange={handleSearchChange}
            // onSearch={onChangeSearch}
          />
        </Col>
      )}
      {renderDatePicker && (
        <Col className="mb-1">
          <DatePicker.RangePicker
            defaultValue={[formData.startDate, formData.endDate]}
            format={dateFormat}
            onChange={onDefaultItemValueChange("DatePicker")}
            placeholder={placeholderTimeRange}
          />
        </Col>
      )}
      {React.Children.map(children, (childItem) => renderChildren(childItem))}
      <Col className="mb-1">
        {renderFilterMoreView && renderFilterMoreView()}
      </Col>
    </Row>
  </Flex>
 );
};

export default FilterComponent;
```