

```

import { DatePicker, Select } from "antd";
import Flex from "components/shared-components/Flex";
import { isEmpty } from "lodash";
import moment from "moment";
import React, { useState } from "react";

export default function FilterBarReport({
  CustomComponent,
  filter,
  setFilter,
  FILTER_TIME = [],
}) {
  const [isOption, setIsOption] = useState(true);

  const handleChange = (value) => {
    let startDate = moment();
    let endDate = moment();
    if (value === -1) setIsOption(!isOption);
    else {
      switch (value) {
        case 1:
          startDate = moment().startOf("day").subtract(1, "days");
          endDate = moment().endOf("day").subtract(1, "days");
          break;
        case 2:
          startDate = moment().startOf("day");
          endDate = moment().endOf("day");
          break;
        case 3:
          startDate = moment().startOf("isoweek");
          endDate = moment().endOf("isoweek");
          break;
        case 4:
          startDate = moment().startOf("month");
          endDate = moment().endOf("month");
          break;
        case 5:
          startDate = moment().startOf("month").subtract(1, "months");
          endDate = moment().endOf("month").subtract(1, "months");
          break;
        case 6:
          startDate = moment().startOf("year");
          endDate = moment().endOf("year");
          break;
        case 7:
          startDate = moment().startOf("isoweek").subtract(1, "weeks");
          endDate = moment().endOf("isoweek").subtract(1, "weeks");
          break;
        default:
          break;
      }
      setFilter({
        ...filter,
        startDate,
        endDate,
      });
      setIsOption(true);
    }
  };

  const handleChangeDatePicker = (value) => {
    let startDate = "";
    let endDate = "";
    if (!isEmpty(value)) {
      startDate = value[0].startOf("day");
      endDate = value[1].endOf("day");
    } else {
      startDate = null;
      endDate = null;
    }
    setFilter({ ...filter, startDate, endDate });
  };

```

```

};

return (
  <Flex alignItems="center" mobileFlex={false}>
    <Flex className="mb-1" mobileFlex={false}>
      {CustomComponent}
      <div className="mr-md-3 mb-3">
        <Select
          onChange={handleChange}
          defaultValue={3}
          style={{ width: 150 }}
          optionFilterProp="children"
          options={FILTER_TIME.concat({ value: -1, label: "Tùy chọn" })}
          filterOption={(input, option) =>
            (option?.label ?? "").toLowerCase().includes(input.toLowerCase())
          }
        />
      </div>
      <div className="mr-md-3 mb-3">
        <DatePicker.RangePicker
          disabled={isOption}
          onChange={handleChangeDatePicker}
        />
      </div>
    </Flex>
  </Flex>
);
}

```