```
import { Badge, Button, Card, Col, Row, Select, Spin } from "antd";
import ChartWidget from "components/shared-components/ChartWidget";
import DonutChartWidget from "components/shared-components/DonutChartWidget";
import Flex from "components/shared-components/Flex";
import { COLORS } from "constants/ChartConstant";
import React, { useEffect, useState } from "react";
import { useDispatch, useSelector } from "react-redux";
import TablePatientRating from "./components/TablePatientRating";
import FilterBarReport from "./components/FilterBarReport";
import { FILTER_TIME_BY_ALL_BRANCH, FILTER_TIME_BY_BRANCH } from "constants";
import {
  getAllDoctorReport,
  getFeedbackFromCustomer,
  getFeedbackFromCustomerExport,
  getReportAllHospital,
  getReportThisHospital,
} from "store/slices/reportSlice";
import FilterBar from "components/searchBar/FilterBar";
import { STATUS_RATING } from "constants";
import moment from "moment";
import { isEmpty } from "lodash";

const initialSearch = {
  startDate: null,
  endDate: null,
  hospitalId: null,
  searchText: null,
  ratingNumber: null,
  doctorId: null,
};

const initHospital = {
  startDate: moment().startOf("isoweek"),
  endDate: moment().endOf("isoweek"),
  hospitalId: null,
};

const initAllHospital = {
  startDate: moment().startOf("isoweek"),
  endDate: moment().endOf("isoweek"),
  arrId: [],
};

const DefaultDashboard = () => {
  const dispatch = useDispatch();
  const { direction } = useSelector((state) => state.theme);
  const { loading, reportHospital, reportAllHospital, doctorList } =
    useSelector((state) => state.report);
  const { hospitalList } = useSelector((state) => state.category);
  const [filterCustomer, setFilterCustomer] = useState(initialSearch);
  const [filterHospital, setFilterHospital] = useState(initHospital);
  const [filterAllHospital, setFilterAllHospital] = useState(initAllHospital);

  useEffect(() => {
    dispatch(getFeedbackFromCustomer({ ...filterCustomer }));
  }, [dispatch, filterCustomer]);

  useEffect(() => {
    dispatch(getReportThisHospital({ ...filterHospital }));
  }, [dispatch, filterHospital]);

  useEffect(() => {
    if (hospitalList.length > 0) {
      dispatch(getAllDoctorReport());
      setFilterCustomer((prev) => ({
        ...prev,
        hospitalId: hospitalList[0].id,
      }));
      setFilterHospital((prev) => ({
        ...prev,
        hospitalId: hospitalList[0].id,
    }));
    setFilterCustomer((prev) => ({
      ...prev,
      hospitalId: hospitalList[0].id,
    }));
  }
}, [dispatch, hospitalList]);

useEffect(() => {
  dispatch(getReportAllHospital(filterAllHospital));
}, [dispatch, filterAllHospital]);

const onChangeHospital = (value) => {
  if (isEmpty(value)) return;
  let arrId = value;
  if (arrId.includes("all")) {
    arrId = hospitalList.map((i) => i.id);
  }
  setFilterAllHospital({ ...filterAllHospital, arrId });
};

const CustomComponent = () => {
  return (
    <>
      <Col>
        <Select
          onChange={(value) =>
            setFilterCustomer({ ...filterCustomer, ratingNumber: value })
          }
          defaultValue={null}
          style={{ width: 150 }}
          optionFilterProp="children"
          options={[{ value: null, label: "Tất cả" }].concat(STATUS_RATING)}
          filterOption={(input, option) =>
            (option?.label ?? "").toLowerCase().includes(input.toLowerCase())
          }
        />
      </Col>
      <Col>
        <Select
          showSearch
          onChange={(value) =>
            setFilterCustomer({ ...filterCustomer, doctorId: value })
          }
          defaultValue={null}
          style={{ width: 200 }}
          placeholder="Chọn bác sỹ"
          optionFilterProp="children"
          allowClear
          options={doctorList
            .filter((x) => x.hospitalId === filterCustomer.hospitalId)
            .map((x) => ({ value: x.id, label: x.name }))}
          filterOption={(input, option) =>
            (option?.label ?? "").toLowerCase().includes(input.toLowerCase())
          }
        />
      </Col>
      <Col>
        <Button
          onClick={() => {
            dispatch(
              getFeedbackFromCustomerExport({
                ...filterCustomer,
                onSuccess: ({ data }) => {
                  const link = document.createElement("a");
                  const linkUrl = data.replace("http:", "https:");
                  link.href = linkUrl;
                  link.click();
                },
              })
            );
          }}
        >
          Xuất excel
        </Button>
      </Col>
    </>
  );
};

const CustomComponentHospital = () => {
  var data = hospitalList?.map((x) => ({
    label: x.name,
    value: x.id,
  }));
  return (
    <Col>
      <Select
        mode="multiple"
        allowClear
        onChange={onChangeHospital}
        defaultValue="all"
        style={{ width: 300 }}
        optionFilterProp="children"
        options={[{ value: "all", label: "Chọn tất cả" }].concat(data)}
        filterOption={(input, option) =>
          (option?.label ?? "").toLowerCase().includes(input.toLowerCase())
        }
      />
    </Col>
  );
};

const getDataTotalView = (type) => {
  if (isEmpty(reportHospital)) return [];
  let resurt = [];
  if (type) resurt = reportHospital?.totalReviewData.map((i) => i.name);
  else resurt = reportHospital?.totalReviewData.map((i) => i.total);
  return resurt;
};

return (
  <Spin tip="Đang tải..." spinning={loading}>
    <Card>
      <FilterBarReport
        CustomComponent={CustomComponentHospital()}
        filter={filterAllHospital}
        setFilter={setFilterAllHospital}
        FILTER_TIME={FILTER_TIME_BY_ALL_BRANCH}
      />
      <ChartWidget
        series={reportAllHospital?.chartData || []}
        xAxis={reportAllHospital?.labelData || []}
        title="SO SÁNH MỨC ĐỘ HÀI LÒNG CỦA BỆNH NHÂN GIỮA CÁC CHI NHÁNH"
        height={410}
        type="bar"
        direction={direction}
      />
    </Card>
    <Row gutter={16}>
      <Col xs={24} sm={24} md={24} lg={24} xxl={18}>
        <Card>
          <FilterBarReport
            CustomComponent={
              <div className="mr-md-3 mb-3">
                <Select
                  placeholder="Chọn chi nhánh"
                  onChange={(value) => {
                    setFilterHospital((prev) => ({
                      ...prev,
                      hospitalId: value,
                    }));
                    setFilterCustomer((prev) => ({
                      ...prev,
                      hospitalId: value,
                    }));
                  }}
                  value={filterHospital.hospitalId}
              style={{ width: 300 }}
              options={hospitalList?.map((x) => ({
                label: x.name,
                value: x.id,
              }))}
            />
          </div>
        }
        filter={filterHospital}
        setFilter={setFilterHospital}
        FILTER_TIME={FILTER_TIME_BY_BRANCH}
      />
      <ChartWidget
        series={reportHospital?.chartData || []}
        xAxis={reportHospital?.labelData || []}
        title="BÁO CÁO MỨC ĐỘ HÀI LÒNG CỦA BỆNH NHÂN"
        height={360}
        type="bar"
        direction={direction}
        customOptions={{
          chart: {
            stacked: true,
            toolbar: {
              show: false,
            },
          },
        }}
      />
    </Card>
  </Col>
  <Col xs={24} sm={24} md={24} lg={24} xxl={6}>
    <DonutChartWidget
      series={getDataTotalView(false)}
      labels={getDataTotalView(true)}
      title="TỔNG ĐÁNH GIÁ"
      bodyClass="mb-2 mt-3"
      extra={
        <Row justify="center">
          <Col xs={20} sm={20} md={20} lg={24}>
            <div className="mt-4 mx-auto" style={{ maxWidth: 200 }}>
              {reportHospital?.totalReviewData?.map((elm, idx) => (
                <Flex
                  alignItems="center"
                  justifyContent="between"
                  className="mb-3"
                  key={elm.value}
                >
                  <div>
                    <Badge
                      color={COLORS.find((_, index) => index === idx)}
                      style={{ marginRight: 5 }}
                    />
                    <span className="text-gray-light">{elm.name}</span>
                  </div>
                  <span className="font-weight-bold text-dark">
                    {elm.total}
                  </span>
                </Flex>
              ))}
            </div>
          </Col>
        </Row>
      }
    />
  </Col>
</Row>
<Card>
  <h4 style={{ fontWeight: "bold" }}>
    BẢNG TỔNG HỢP PHẢN HỒI Ý KIẾN KHÁCH HÀNG
  </h4>
  <FilterBar
    CustomComponent={CustomComponent()}
    filter={filterCustomer}
    setFilter={setFilterCustomer}
```

```
        />
        <TablePatientRating />
      </Card>
    </Spin>
  );
};

export default DefaultDashboard;
```