

```
import { Col, Form, Input, Modal, notification, Row, Select } from "antd";
import { useEffect, useState } from "react";
import { useDispatch, useSelector } from "react-redux";
import {
  GetAllDonHangKhamSuckhoeNhanSu,
  UpsertDonHangKhamSuckhoeKhac,
} from "store/slices/khamSuckhoeDoanSlice";
import ModalEmployeeGetAll from "views/app-views/hrm/employee/components/ModalEmployeeGetAll";
```

```
const ModelNhanSu = ({ isVisibleNhanSu, setIsVisibleNhanSu, donHangId }) => {
  const dispatch = useDispatch();
  const [form] = Form.useForm();
  const [visibleEmployee, setVisibleEmployee] = useState(false);
  const [employeeIds, setEmployeeIds] = useState(null);
  const { employeeList } = useSelector((state) => state.category);
  const { employeeAllList } = useSelector((state) => state.employee);
```

```
  useEffect(() => {
    if (isVisibleNhanSu) form.resetFields();
  }, [form, isVisibleNhanSu]);
```

```
  const onFinish = (values) => {
    try {
      const payload = {
        id: values?.id || null,
        nhanSuId: values?.nhanSuId,
        loai: 1,
        chuyenMon: values?.chuyenMon,
        ghiChu: values?.ghiChu,
        khoaPhong: values?.khoaPhong,
        donHangId: donHangId,
        ...values,
        onSuccess: (data) => {
          if (data.data.isOk) {
            dispatch(
              GetAllDonHangKhamSuckhoeNhanSu({
                id: donHangId,
                loai: 1,
              })
            );
          } else {
            notification.error({
              message: "Lưu chi tiết",
              description: data.description,
            });
          }
        },
      };

      dispatch(UpsertDonHangKhamSuckhoeKhac(payload));
    } catch (error) {}
  };

  return (
```

```
    <>
      <Modal
        title={"Thêm/sửa nhân sự tham gia"}
        open={isVisibleNhanSu}
        onCancel={() => {
          form.resetFields();
          setIsVisibleNhanSu(!isVisibleNhanSu);
        }}
        onOk={() => {
          form.submit();

          setIsVisibleNhanSu(!isVisibleNhanSu);
        }}
        //footer={renderSignProcess()}
        width={600}
        height={"80vh"}
        className="sign-modal"
```

```

// bodyStyle={{ overflowY: 'scroll' }}
maskClosable={false}
>
<Form form={form} layout="vertical" onFinish={onFinish} className="p-3">
  <Row gutter={24}>
    <Col sm={24}>
      <Form.Item
        name="nhanSuId"
        label="Tên nhân sự"
        rules={[
          {
            required: true,
            message: "Vui lòng chọn nhân sự",
          },
        ]}
      >
        <Select
          showSearch
          filterOption={(input, option) =>
            (option?.label ?? "").
              .toLowerCase()
              .includes(input.toLowerCase())
          }
          options={employeeList}
          optionLabelProp="label"
          open={false}
          mode="single"
          onChange={(e) => {}}
          onClick={() => {
            setVisibleEmployee((prev) => !prev);
            setEmployeeIds(form.getFieldValue("nhanSuId"));
          }}
        />
      </Form.Item>
    </Col>
  </Row>

  <Row gutter={24}>
    <Col sm={24}>
      <Form.Item name="khoaPhong" label="Khoa/phòng">
        <Input disabled />
      </Form.Item>
    </Col>
  </Row>

  <Row gutter={24}>
    <Col sm={24}>
      <Form.Item name="chuyenMon" label="Nhiệm vụ">
        <Input />
      </Form.Item>
    </Col>
  </Row>

  <Row gutter={24}>
    <Col sm={24}>
      <Form.Item name="ghiChu" label="Ghi chú">
        <Input />
      </Form.Item>
    </Col>
  </Row>

  <ModalEmployeeGetAll
    visibleModal={visibleEmployee}
    selectedRowKeys={employeeIds}
    setSelectedRowKeys={setEmployeeIds}
    onCancel={() => {
      setVisibleEmployee((prev) => !prev);
    }}
    onOk={() => {
      setVisibleEmployee((prev) => !prev);
      const empl = employeeList.find((x) => x.value === employeeIds[0]);
      form.setFieldValue("nhanSuId", empl.value);

      setEmployeeIds(employeeIds);
    }}
  />

```

```
        var kp = employeeAllList.find(
          (f) => f.id === empl.value
        ).departmentName;

        form.setFieldValue("khoaPhong", kp);
      }}
      isDeptByBranch
      type="radio"
    />
  </Form>
</Modal>
</>
);
};
export default ModelNhanSu;
```