

```

import React, { memo, useRef, useState } from "react";
import {
  Button,
  Form,
  DatePicker,
  Card,
  Col,
  Checkbox,
  Input,
  Row,
  Spin,
  Result,
  Modal,
} from "antd";
import { dateFormatList } from "constants";
import {
  UserOutlined,
  IdcardOutlined,
  ExclamationCircleOutlined,
} from "@ant-design/icons";
import { useSelector, useDispatch } from "react-redux";
import styled from "styled-components";
import { useEffect } from "react";
import { logoMedic } from "assets/images";
import { Steps } from "antd";
import moment from "moment";
import SignaturePad from "react-signature-canvas";
import { checkTheApi, esignSubmitApi } from "store/slices/ratingSlice";
import styles from "../styles.module.css";
const { Step } = Steps;
const { confirm } = Modal;

const ESign = () => {
  const dispatch = useDispatch();
  const { loading } = useSelector((state) => state.rating);

  const signPadRef = useRef(null);
  const [currentStep, setCurrentStep] = useState(0);
  const [form] = Form.useForm();
  const [landScape, setLandScap] = useState(false);
  const [endSign, setEndSign] = useState(false);
  useEffect(() => {
    form.setFieldsValue("fullName", "");
    form.setFieldsValue("birthday", "");
    form.setFieldsValue("identityNo", "");
  }, [dispatch, form]);

  const handleSubmit = () => {
    if (currentStep === 0) {
      form.submit();
      // setCurrentStep(1);
    } else if (currentStep === 1) {
      // get form data
      completeSign();
      // setCurrentStep(2);
    } else if (currentStep === 2) {
      setCurrentStep(3);
    }
    // if (ratingValue > 0) {
    //   var data = {
    //     id: customerInfo?.ratingInfo?.id,
    //     comment: comment,
    //     rate: ratingValue,
    //     onSuccess: () => {
    //       setComment("");
    //       setRatingValue(0);
    //       // dispatch(getByCodeRatingApi(code));
    //     },
    //   };
    //   dispatch(addRatingApi(data));
    // } else {
    //   notification.warning({
    //     message: "Thông báo",
    //     description:
    //       "Xin vui lòng đánh giá trải nghiệm trước khi gửi thông tin",
    //   });
    // }
  };

  const handleBack = () => {
    setCurrentStep(currentStep - 1);
  };

  const onChangeStep = (current) => {
    console.log("onChange:", current);
    setCurrentStep(current);
  };

```

```
};
const showPromiseConfirm = (checkTheMsg) => {
  confirm({
    title: "Có lỗi xảy ra khi định danh người dùng?",
    icon: <ExclamationCircleOutlined />,
    content: ` ${checkTheMsg}. Bạn có muốn tiếp tục?`,
    onOk() {
      setCurrentStep(1);
    },
    onCancel() {},
  });
};

const handleFormSubmit = (values) => {
  const payload = {
    NgaySinh: moment(values.birthday).format("DD/MM/YYYY"),
    HoTen: values.fullName,
    MaThe: values.identityNo,
    onSuccess: () => {
      setCurrentStep(1);
    },
    onReject: (errMsg) => {
      showPromiseConfirm(errMsg);
    },
  };
  dispatch(checkTheApi(payload));
};

const completeSign = () => {
  const esignData = signPadRef.current
    .getTrimmedCanvas()
    .toDataURL("image/png");
  const payload = {
    FullName: form.getFieldValue("fullName"),
    Dob: moment(form.getFieldValue("birthday")).format("DD/MM/YYYY"),
    IdentityNo: form.getFieldValue("identityNo"),
    EsignData: esignData,
    onSuccess: () => {
      setCurrentStep(2);
    },
  };
  dispatch(esignSubmitApi(payload));
};

const renderForm = () => {
  return (
    <Form form={form} layout="vertical" onFinish={handleFormSubmit}>
      <Form.Item
        name="fullName"
        label="Họ và tên"
        rules={[
          {
            required: true,
            message: "Họ và tên là bắt buộc!",
          },
        ]}>
        <Input
          prefix={<UserOutlined className="text-primary" />}
          placeholder="Nhập họ tên của bạn"
        />
      </Form.Item>
      <Form.Item
        name="birthday"
        label="Ngày sinh"
        rules={[
          {
            required: true,
            message: "Ngày sinh là bắt buộc!",
          },
        ]}>
        <DatePicker
          format={dateFormatList}
          style={{ width: "100%" }}
          placeholder="Chọn ngày sinh"
        />
      </Form.Item>
      <Form.Item
        name="identityNo"
        label="Căn cước công dân"
        rules={[
          {
            required: true,
            message: "Căn cước công dân là bắt buộc!",
          },
        ]}>
        <Input
          prefix={<IdcardOutlined className="text-primary" />}

```

placeholder="Nhập số căn cước công dân của bạn"

```
    />
  </Form.Item>
</Form>
```

```
);
};
```

```
const renderSignature = () => {
  return (
    <div style={{ height: "calc(100vh - 130px)", border: "1px solid black" }}>
      <SignaturePad
        ref={signPadRef}
        penColor="blue"
        canvasProps={{ className: styles.sigPad }}
        onEnd={() => {
          setEndSign(true);
        }}
      />
    </div>
  );
};
```

```
};
return (
  <Spin tip="Đang tải..." spinning={loading}>
    <Row>
      <Col xs={24} md={24} lg={6}></Col>
      <Col xs={24} md={24} lg={12}>
        <CustomDiv style={{ margin: 20 }}>
          {currentStep !== 1 && (
            <>
              <div style={{ display: "flex" }}>
                <CustomImg>
                  <img src={logoMedic} alt={logoMedic} />
                </CustomImg>
                <h3 style={{ marginLeft: "10px", color: "#149b4f" }}>
                  HỆ THỐNG Y TẾ MEDIC
                </h3>
              </div>
              <h4 className="title">
                PHẦN MỀM LẤY CHỮ KÝ DÀNH CHO KHÁM SỨC KHỎE ĐOÀN
              </h4>
              <Steps
                current={currentStep}
                onChange={onChangeStep}
                style={{ marginTop: "50px", marginBottom: "10px" }}
              >
                <Step title="Nhập thông tin" />
                <Step title="Ký" />
                <Step title="Hoàn thành" />
              </Steps>
            </>
          )}
        </CustomDiv>
      </Col>
    </Row>
    {currentStep === 2 && (
      <Result
        status="success"
        title="Ký thành công trên hệ thống ý tế Medic"
        subTitle="Cảm ơn bạn đã ký hồ sơ trên hệ thống y tế Medic. Chữ ký của bạn sẽ được sử dụng trong hồ sơ khám sức khỏe"
        extra={[
          <Button
            key="buy"
            onClick={() => {
              setCurrentStep(0);
              setEndSign(false);
            }}
          >
            Ký lại
          </Button>,
        ]}
      />
    )}
    {currentStep === 1 && renderSignature()}
    {currentStep === 0 && <Card>{renderForm()}</Card>}

    <div style={{ display: "flex", justifyContent: "center" }}>
      {(currentStep === 0 || currentStep === 1) && (
        <CustomButton
          type="text"
          onClick={handleBack}
          style={{ marginRight: "10px" }}
        >
          Quay lại
        </CustomButton>
      )}
      {currentStep < 2 && (
        <CustomButton
          type="primary"
          onClick={handleSubmit}
          disabled={currentStep === 1 && !endSign}
        >
          {currentStep === 1 ? "Hoàn thành" : "Tiếp tục"}
        </CustomButton>
      )}
    </div>
  </Spin>
);
```

```

        </CustomButton>
    )}
    {currentStep === 1 && (
        <CustomButton
            danger
            type="text"
            onClick={() => {
                if (signPadRef && signPadRef.current)
                    signPadRef.current.clear();
                setEndSign(false);
            }}
        >
            Xóa
        </CustomButton>
    )}
</div>
{currentStep === 1 && (
    <div
        style={{
            display: "flex",
            justifyContent: "center",
            marginTop: "5px",
        }}
    >
        <Checkbox
            checked={landScape}
            onChange={(e) => setLandScap(e.target.checked)}
        >
            Ký ngang
        </Checkbox>
    </div>
)}
</CustomDiv>
</Col>
<Col xs={24} md={24} lg={6}></Col>
</Row>
</Spin>
);
};
export default memo(ESign);

```

```

const CustomDiv = styled.div`
    .title {
        display: flex;
        justify-content: center;
        align-items: center;
        text-align: center;
        margin: 15px 0;
    }
    .info {
        font-size: 16px;
    }
    .ant-image,
    .rate {
        width: 100%;
        display: flex;
        justify-content: center;
    }
    .ant-rate {
        font-size: 36px;
    }
    .ant-rate-star-second {
        color: #d3d3d3;
    }
    .ant-rate-star-full .ant-rate-star-second,
    .ant-rate-star-half .ant-rate-star-first {
        color: #f7ca18;
    }
    .ant-input,
    .ant-select {
        border: 1px solid gray;
        border-radius: 0.625rem;
    }
`;
const CustomButton = styled(Button)`
    margin-top: 20px;
`;
const CustomImg = styled.div`
    display: flex;
    img {
        margin: auto;
        width: 40px;
    }
`;

```

