```
import { Form, Spin } from "antd";
import React, { useMemo } from "react";

import { useSelector } from "react-redux";
import Utils from "utils";
import MediTable from "components/Custom";
import EditableCell from "components/EditableCell";

export const initHoSo = {
  action: "initial",
  isRequired: true,
};

const ChiTietPhuongAn = ({
  ThanhLyHopDongKskDetail,
  ptChietKhau,
  ptTyLeChenh,
}) => {
  const { loading, ThanhLyHopDongKskChiTietList, DmDichVuList } = useSelector(
    (state) => state.ksk
  );
  const [form] = Form.useForm();

  const Footer = useMemo(() => {
    let tongCong = 0;
    for (
      let index = 0;
      index <= ThanhLyHopDongKskChiTietList.length - 1;
      index++
    ) {
      tongCong += ThanhLyHopDongKskChiTietList[index].thanhTien;
    }

    let tongCongThucTe = 0;
    let chietKhauThucTe = 0;
    for (
      let index = 0;
      index <= ThanhLyHopDongKskChiTietList.length - 1;
      index++
    ) {
      tongCongThucTe += ThanhLyHopDongKskChiTietList[index].thanhTienThucTe;
    }

    if (ptChietKhau > 0) {
      chietKhauThucTe = (tongCongThucTe * ptChietKhau) / 100;
    }

    return (
      <>
        <h5>Tổng cộng: ({Utils.formatterNumber(tongCong)})</h5>
        <h5>Tổng cộng thực tế: ({Utils.formatterNumber(tongCongThucTe)})</h5>
        <h5>
          Chênh lệch HĐ: ({Utils.formatterNumber(tongCong - tongCongThucTe)})
        </h5>
        <h5>
          Phí chênh lệch HĐ: (
          {ptTyLeChenh > 0
            ? Utils.formatterNumber(
                (ptTyLeChenh * (tongCong - tongCongThucTe)) / 100
              )
            : "0"}
          )
        </h5>
        <h5>
          Chiết khấu
          {` (${ptChietKhau}%)`}: (
          {ptChietKhau > 0 ? Utils.formatterNumber(chietKhauThucTe) : "0"})
        </h5>
        <h5>
          Thu thực tế: (
          {Utils.formatterNumber(
          tongCongThucTe + (ptTyLeChenh * (tongCong - tongCongThucTe)) / 100
        )}
        )
      </h5>
    </>
  );
}, [ThanhLyHopDongKskChiTietList, ptChietKhau, ptTyLeChenh]);

const tableColumns = useMemo(
  () => [
    {
      title: "STT",
      align: "center",
      render: (_, __, index) => index + 1,
      width: "50px",
    },
    {
      title: "Dịch vụ",
      dataIndex: "dichVuId",
      editable: true,
      inputType: "select",
      width: "350px",
      options: DmDichVuList.map((i) => ({
        id: i.id,
        name: i.tenDichVu,
      })),
      render: (value, record) => record.tenDichVu,
    },
    {
      title: "Đơn giá HĐ",
      dataIndex: "donGia",
      editable: true,
      inputType: "number",
      width: "110px",
      render: (value, record) => Utils.formatterNumber(record.donGia),
    },
    {
      title: "Thành tiền HĐ",
      dataIndex: "thanhTien",
      editable: false,
      width: "110px",
      render: (value, record) => Utils.formatterNumber(record.thanhTien),
    },
    {
      title: "Giá thực tế",
      dataIndex: "donGiaThucTe",
      editable: true,
      inputType: "number",
      width: "110px",
      render: (value, record) => Utils.formatterNumber(record.donGiaThucTe),
    },
    {
      title: "Tiền thực tế",
      dataIndex: "thanhTienThucTe",
      editable: false,
      width: "110px",
      render: (value, record) =>
        Utils.formatterNumber(record.thanhTienThucTe),
    },
  ],
  [DmDichVuList]
);

const mergedColumns = tableColumns.map((col) => {
  if (!col.editable) {
    return col;
  }

  return {
    ...col,
    onCell: (record) => ({
      record,
      title: col.title,
      dataIndex: col.dataIndex,
        editing: false,
        inputType: col?.inputType,
        isRequired: col?.isRequired,

        options: col?.options,
      }),
    };
  });

  return (
    <Spin tip="Đang tải..." spinning={loading}>
      <Form form={form} component={false}>
        <MediTable
          components={{
            body: {
              cell: EditableCell,
            },
          }}
          tableColumns={mergedColumns}
          dataSource={ThanhLyHopDongKskChiTietList}
          scroll={{ x: "max-content" }}
          rowKey={(item) => item?.id}
          onRenderFooter={() => Footer}
        />
      </Form>
    </Spin>
  );
};

export default ChiTietPhuongAn;
```