

```

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

import { useDispatch, useSelector } from "react-redux";
import { GetPhuongAnKinhDoanhChiTiet } from "store/slices/khamSucKhoeDoanSlice";
import Utils from "utils";
import MediTable from "components/Custom";
import EditableCell from "components/EditableCell";
import { guidEmpty } from "constants";

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

const ChiTietPhuongAn = ({ phuongAnId, PhuongAnKinhDoanhDetail }) => {
  const dispatch = useDispatch();
  const { loading, PhuongAnKinhDoanhChiTietList, DmDichVuList } = useSelector(
    (state) => state.ksk
  );

  const [form] = Form.useForm();

  useEffect(() => {
    if (phuongAnId) {
      dispatch(
        GetPhuongAnKinhDoanhChiTiet(
          phuongAnId !== null && phuongAnId !== undefined
            ? phuongAnId
            : guidEmpty
        )
      );
    }
  }, [dispatch, phuongAnId]);

  // DmDichVuList;
}, [dispatch, phuongAnId]);

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

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

  if (PhuongAnKinhDoanhDetail?.ptChietKhau > 0) {
    chietKhauThucTe =
      (tongCongThucTe * PhuongAnKinhDoanhDetail?.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: (

```

```

    {PhuongAnKinhDoanhDetail?.ptTyLeChenh > 0
      ? Utils.formatterNumber(
        (PhuongAnKinhDoanhDetail?.ptTyLeChenh *
          (tongCong - tongCongThucTe)) /
          100
      )
      : "0"}
  )
</h5>
<h5>
  Chiết khấu
  {` (${PhuongAnKinhDoanhDetail?.ptChietKhau}%)`}: (
    {PhuongAnKinhDoanhDetail?.ptChietKhau > 0
      ? Utils.formatterNumber(chietKhauThucTe)
      : "0"}
  )
</h5>
<h5>
  Thu thực tế: (
    {Utils.formatterNumber(
      tongCongThucTe +
      (PhuongAnKinhDoanhDetail?.ptTyLeChenh *
        (tongCong - tongCongThucTe)) /
        100
    )}
  )
</h5>
</>
);
};

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: isEditing(record),
      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={PhuongAnKinhDoanhChiTietList}
        // dataSource=[[{ action: "initial" }].concat(
        //   PhuongAnKinhDoanhChiTietList.map((x) => ({
        //     ...x,
        //   }))]
        // )}
        scroll={{ x: "max-content" }}
        rowKey={(item) => item?.id}
        onRenderFooter={() => Footer()}
      />
    </Form>
  </Spin>
);
};

```

```

export default ChiTietPhuongAn;

```