import React from "react";

import {

Box,

CssBaseline,

ThemeProvider,

Drawer,

List,

ListItemButton,

ListItemIcon,

ListItemText,

IconButton,

Button,

TextField,

Dialog,

DialogActions,

DialogContent,

DialogTitle,

Snackbar,

Alert,

Autocomplete,

} from "@mui/material";

import LocalPharmacyIcon from "@mui/icons-material/LocalPharmacy";

import LogoutIcon from "@mui/icons-material/Logout";

import DarkModeIcon from "@mui/icons-material/DarkMode";

import LightModeIcon from "@mui/icons-material/LightMode";

import MenuIcon from "@mui/icons-material/Menu";

import { DataGrid, GridColDef } from "@mui/x-data-grid";

import PharmacistToaManager from "./PharmacistToaManager";

import Sidebar from "./components/Sidebar";

import RotatingBanner from "./RotatingBanner";

import Topbar from "./components/Topbar";

import { Theme } from "@mui/material/styles";

import StatisticsBoard from "./StatisticsBoard";

import "./styles.css";

interface Thuoc {

id?: string;

ten\_thuoc: string;

don\_vi: string;

so\_luong\_ton: number;

gia\_nhap: number;

gia\_ban: number;

so\_luong\_dat\_hang?: number;

duong\_dung?: string;

}

interface PharmacistViewUIProps {

thuocList: Thuoc[];

columns: GridColDef[];

openDialog: boolean;

mode: "light" | "dark";

drawerOpen: boolean;

setDrawerOpen: React.Dispatch<React.SetStateAction<boolean>>;

newThuoc: Thuoc;

handleLogout: () => void;

setOpenDialog: React.Dispatch<React.SetStateAction<boolean>>;

handleChange: (e: React.ChangeEvent<HTMLInputElement>) => void;

handleAddThuoc: () => void;

setMode: React.Dispatch<React.SetStateAction<"light" | "dark">>;

theme: Theme;

handleKeyDown: (

e: React.KeyboardEvent,

nextRef: React.RefObject<HTMLInputElement> | null

) => void;

tenThuocRef: React.RefObject<HTMLInputElement>;

donViRef: React.RefObject<HTMLInputElement>;

soLuongTonRef: React.RefObject<HTMLInputElement>;

giaNhapRef: React.RefObject<HTMLInputElement>;

giaBanRef: React.RefObject<HTMLInputElement>;

successMessage: string;

setSuccessMessage: React.Dispatch<React.SetStateAction<string>>;

statisticsData: { label: string; value: string | number }[];

modeDialog: "new" | "update";

searchOptions: { ten\_thuoc: string; data: any }[];

handleSearchThuoc: (input: string) => void;

handleSelectThuocCu: (thuoc: any) => void;

}

const PharmacistViewUI: React.FC<PharmacistViewUIProps> = ({

thuocList,

columns,

openDialog,

mode,

drawerOpen,

setDrawerOpen,

newThuoc,

handleLogout,

setOpenDialog,

handleChange,

handleAddThuoc,

setMode,

theme,

handleKeyDown,

tenThuocRef,

donViRef,

soLuongTonRef,

giaNhapRef,

giaBanRef,

successMessage,

setSuccessMessage,

statisticsData,

modeDialog,

searchOptions,

handleSearchThuoc,

handleSelectThuocCu,

}) => {

const sortedThuocList = [...thuocList].sort((a, b) =>

a.ten\_thuoc.localeCompare(b.ten\_thuoc)

);

return (

<ThemeProvider theme={theme}>

<Box sx={{ display: "flex", height: "100vh" }}>

<CssBaseline />

<Topbar>

<IconButton

edge="start"

color="inherit"

aria-label="menu"

onClick={() => setDrawerOpen(!drawerOpen)}

sx={{ mr: 2 }}

>

<MenuIcon />

</IconButton>

</Topbar>

<Drawer

variant="persistent"

open={drawerOpen}

sx={{

width: 240,

flexShrink: 0,

[`& .MuiDrawer-paper`]: { width: 240, boxSizing: "border-box" },

}}

>

<List>

<ListItemButton onClick={() => setOpenDialog(true)}>

<ListItemIcon>

<LocalPharmacyIcon />

</ListItemIcon>

<ListItemText primary="Thêm thuốc" />

</ListItemButton>

<ListItemButton

onClick={() => setMode(mode === "light" ? "dark" : "light")}

>

<ListItemIcon>

{mode === "light" ? <DarkModeIcon /> : <LightModeIcon />}

</ListItemIcon>

<ListItemText

primary={mode === "light" ? "Dark mode" : "Light mode"}

/>

</ListItemButton>

<ListItemButton onClick={handleLogout}>

<ListItemIcon>

<LogoutIcon />

</ListItemIcon>

<ListItemText primary="Đăng xuất" />

</ListItemButton>

</List>

<Box sx={{ p: 2, borderTop: "1px solid #ddd" }}>

<StatisticsBoard data={statisticsData} />

</Box>

</Drawer>

<Box

sx={{

flexGrow: 1,

p: 3,

width: drawerOpen ? `calc(100% - 240px)` : "100%",

transition: "width 0.3s",

overflow: "auto",

}}

>

<Box sx={{ height: 500, width: "100%", mb: 0.5 }}>

<DataGrid

rows={sortedThuocList}

columns={columns}

rowHeight={25}

getRowId={(row) => row.id || Math.random().toString()}

getRowClassName={(params) =>

params.row.so\_luong\_ton <= (params.row.so\_luong\_dat\_hang ?? 0)

? "orange-row"

: ""

}

hideFooter

sortingOrder={["asc", "desc"]}

disableColumnMenu

/>

</Box>

<PharmacistToaManager />

</Box>

<Box sx={{ width: "20%", borderLeft: "1px solid #ccc", p: 2 }}>

<Sidebar role="pharmacist" />

</Box>

<RotatingBanner />

</Box>

<Dialog open={openDialog} onClose={() => setOpenDialog(false)} fullWidth>

<DialogTitle>

{modeDialog === "new" ? "Thêm thuốc mới" : "Nhập thêm số lượng thuốc"}

</DialogTitle>

<DialogContent>

{modeDialog === "update" ? (

<Autocomplete

freeSolo

options={searchOptions}

getOptionLabel={(option) =>

typeof option === "string" ? option : option.ten\_thuoc

}

onInputChange={(e, value) => handleSearchThuoc(value)}

onChange={(e, value) => {

if (value?.data) handleSelectThuocCu(value.data);

}}

renderInput={(params) => (

<TextField

{...params}

margin="dense"

label="Tên thuốc (cũ)"

name="ten\_thuoc"

fullWidth

value={newThuoc.ten\_thuoc}

onChange={handleChange}

inputRef={tenThuocRef}

/>

)}

/>

) : (

<TextField

margin="dense"

label="Tên thuốc"

name="ten\_thuoc"

fullWidth

value={newThuoc.ten\_thuoc}

onChange={handleChange}

inputRef={tenThuocRef}

onKeyDown={(e) => handleKeyDown(e, donViRef)}

/>

)}

<TextField

margin="dense"

label="Đơn vị"

name="don\_vi"

fullWidth

value={newThuoc.don\_vi}

onChange={handleChange}

inputRef={donViRef}

onKeyDown={(e) => handleKeyDown(e, soLuongTonRef)}

/>

<TextField

margin="dense"

label="Số lượng"

name="so\_luong\_ton"

type="number"

fullWidth

value={newThuoc.so\_luong\_ton}

onChange={handleChange}

inputRef={soLuongTonRef}

onKeyDown={(e) => handleKeyDown(e, giaNhapRef)}

/>

<TextField

margin="dense"

label="Giá nhập"

name="gia\_nhap"

type="number"

fullWidth

value={newThuoc.gia\_nhap}

onChange={handleChange}

inputRef={giaNhapRef}

onKeyDown={(e) => handleKeyDown(e, giaBanRef)}

/>

<TextField

margin="dense"

label="Giá bán"

name="gia\_ban"

type="number"

fullWidth

value={newThuoc.gia\_ban}

onChange={handleChange}

inputRef={giaBanRef}

onKeyDown={(e) => handleKeyDown(e, null)}

/>

</DialogContent>

<DialogActions>

<Button onClick={() => setOpenDialog(false)}>Hủy</Button>

<Button variant="contained" onClick={handleAddThuoc}>

Lưu

</Button>

</DialogActions>

</Dialog>

{successMessage && (

<Snackbar

open={Boolean(successMessage)}

autoHideDuration={3000}

onClose={() => setSuccessMessage("")}

>

<Alert onClose={() => setSuccessMessage("")} severity="success">

{successMessage}

</Alert>

</Snackbar>

)}

</ThemeProvider>

);

};

export default PharmacistViewUI;