

ĐỀ ÔN TẬP REACT – CHỦ ĐỀ “TASK TRACKER APP”

Huỳnh Đức Phú – 21/5/2025

I. JSON SERVER

- Sử dụng file db.json có sẵn với mục tasks.
- Khởi động JSON Server trên cổng [4 số cuối MSSV].

II. KIẾN TRÚC ỨNG DỤNG (Redux Toolkit + React Router + Tailwind CSS)

- Khởi tạo store bằng configureStore và import taskSlice.
- Trong taskSlice, định nghĩa state gồm tasks, loading, error.
- Tạo Header component chứa hai link “Home” (/) và “Tasks” (/tasks); sử dụng Tailwind để style link active.
- Tạo Footer component hiển thị Họ tên và MSSV.
- Trang Landing (/): hiển thị text “Trang Home”.

f. Trang danh sách (/tasks):

- Trước khi render, dispatch async thunk fetchTasks để load danh sách và lưu vào state.
- Hiển thị danh sách công việc (tiêu đề, trạng thái, ngày hết hạn).
- Tại mỗi item, có nút xóa dispatch async thunk deleteTask(id).
- Chèn <Outlet> để render các view phụ.

*Giải thích thêm: Mục đích chèn Outlet là sẽ có **trang thêm task**, trang đó là child trong route /tasks*

g. Trang thêm (/tasks/add):

- Render form thêm công việc (tiêu đề, mô tả, ngày hết hạn) trong <Outlet> của /tasks.
- Khi submit, dispatch async thunk addTask(data); sau khi thành công, navigate về /tasks.

h. Trang chi tiết (/tasks/:id):

- Sử dụng fetch (GET) để load chi tiết công việc.
- Hiển thị tiêu đề, mô tả, trạng thái, ngày hết hạn.
- Nút “Quay lại” về /tasks và nút chỉnh sửa để qua trang chỉnh sửa.

i. Trang chỉnh sửa (/tasks/:id/edit):

- Sử dụng fetch (GET) để load dữ liệu ban đầu vào form.
- Khi submit, dispatch async thunk updateTask(id, data); sau khi thành công, navigate về /tasks/:id.

III. CUSTOM HOOK CHO FETCH API

- Tạo hook useFetch(url) chỉ thực hiện GET; hook trả về { data, loading }.

b. Ứng dụng useFetch trong các trang chi tiết và edit.

IV. CUSTOM HOOK CHO TASK

a. Tạo hook useTask; trong hook sử dụng useDispatch để dispatch các async thunk từ taskSlice (trừ fetchTasks), và useSelector để lấy tasks, loading, error từ state.

b. Hook useTask trả về object { tasks, loading, error, createTask, editTask, removeTask }.

- createTask là hàm dispatch đến async thunk addTask
- editTask là hàm dispatch đến async thunk updateTask
- removeTask là hàm dispatch đến async thunk deleteTask

c. Ứng dụng hook vào các trang trên

V. CẤU TRÚC THƯ MỤC src/ (tham khảo)

src/

```
├─ features/
│   └─ store.js
│   └─ taskSlice.js          # slice, async thunks và logic API
├─ components/
│   └─ Header.js             # header, navigation
│   └─ Footer.js             # footer với Họ tên, MSSV
│   └─ TaskCard.js
│   └─ TaskForm.js
├─ hooks/
│   └─ useFetch.js           # custom hook GET
│   └─ useTask.js            # custom hook Redux
├─ pages/
│   └─ LandingPage.js
│   └─ TasksPage.js
│   └─ AddTaskPage.js
│   └─ TaskDetailPage.js
│   └─ TaskEditPage.js
│   └─ RootPage.js
├─ routes/
│   └─ AppRoutes.js          # định nghĩa các route
└─ index.js
```