

Theory 10

Global State Management in ReactJS

WEEK 03

Nội dung chính

❑ TỔNG QUAN VỀ REDUX	03
❑ REDUX WORKFLOW	08
❑ REDUX MIDDLEWARES	09
❑ TRIỂN KHAI REDUX THUNK	10
❑ TỔNG QUAN REDUX TOOLKIT	13
❑ TRIỂN KHAI REDUX TOOLKIT	15
❑ REDUX TOOLKIT QUERY	18
❑ TRIỂN KHAI RTK QUERY	19

AGENDA



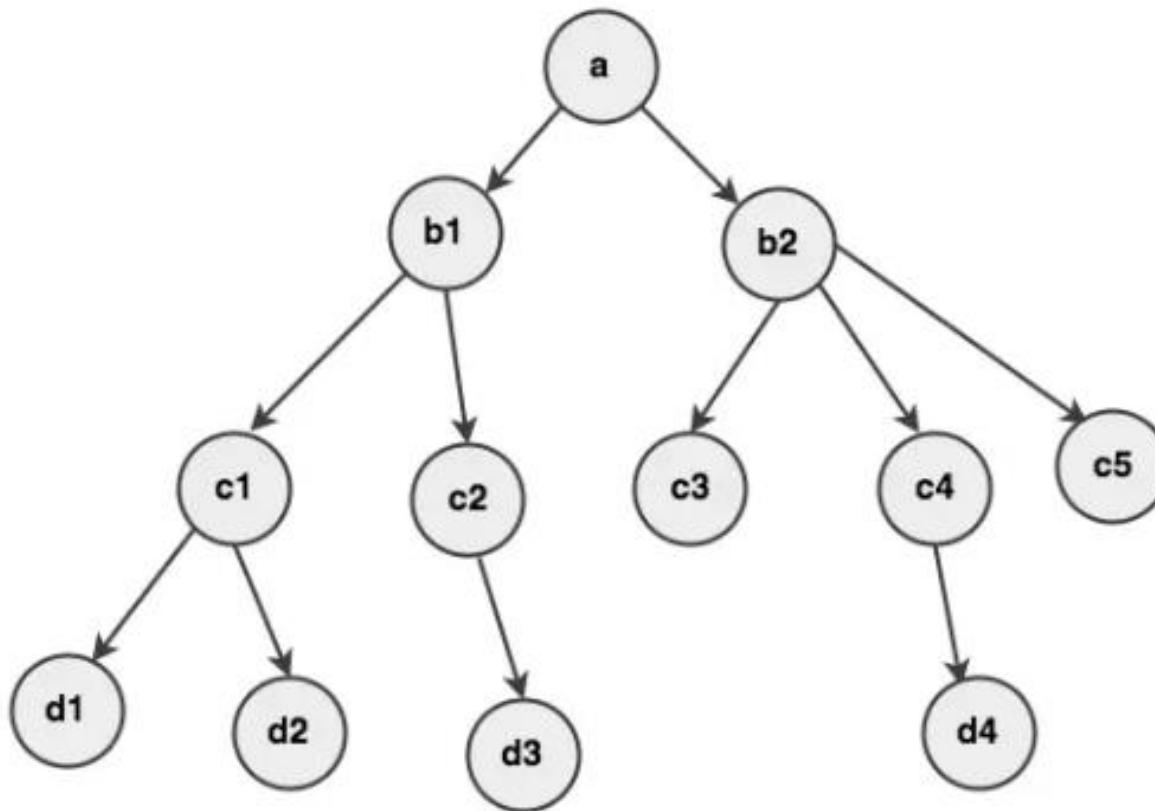
Tổng quan về Redux

- ❑ **Redux** là một thư viện **JavaScript** dùng để quản lý các state mà các state này có thể dự đoán được (**predictable state container**).
- ❑ **Redux** là sử dụng kiến trúc dữ liệu một chiều (**uni-directional data flow**).



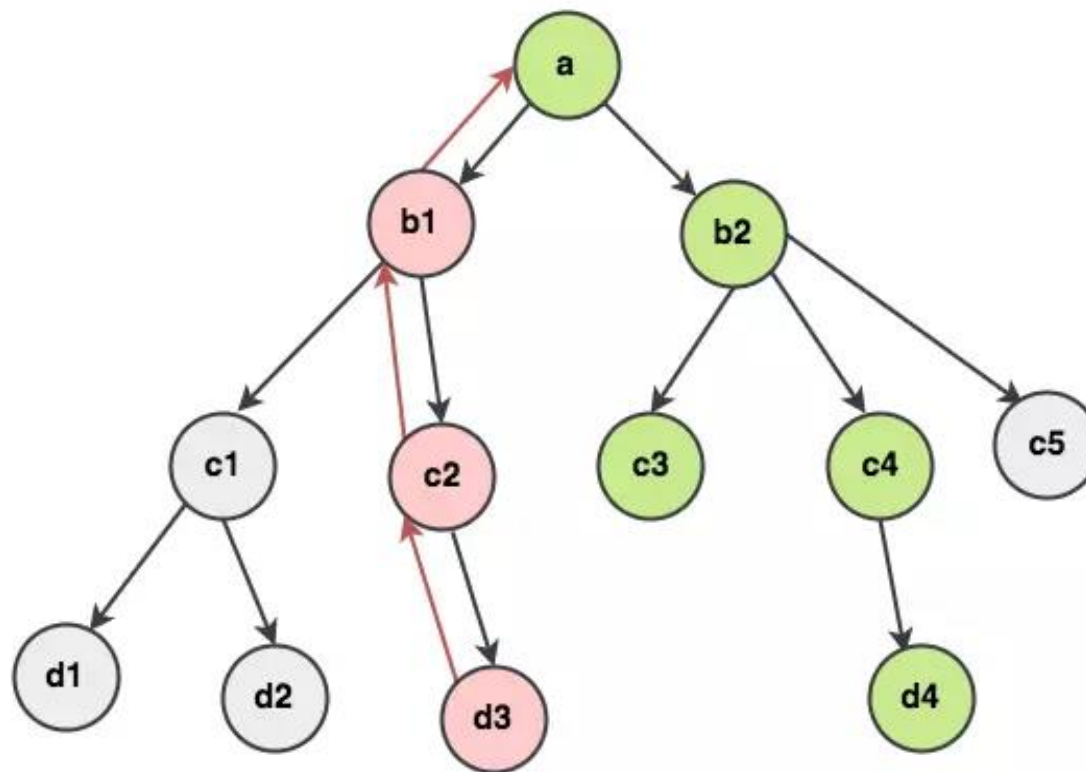
Tổng quan về Redux

❑ Tại sao nên sử dụng Redux?



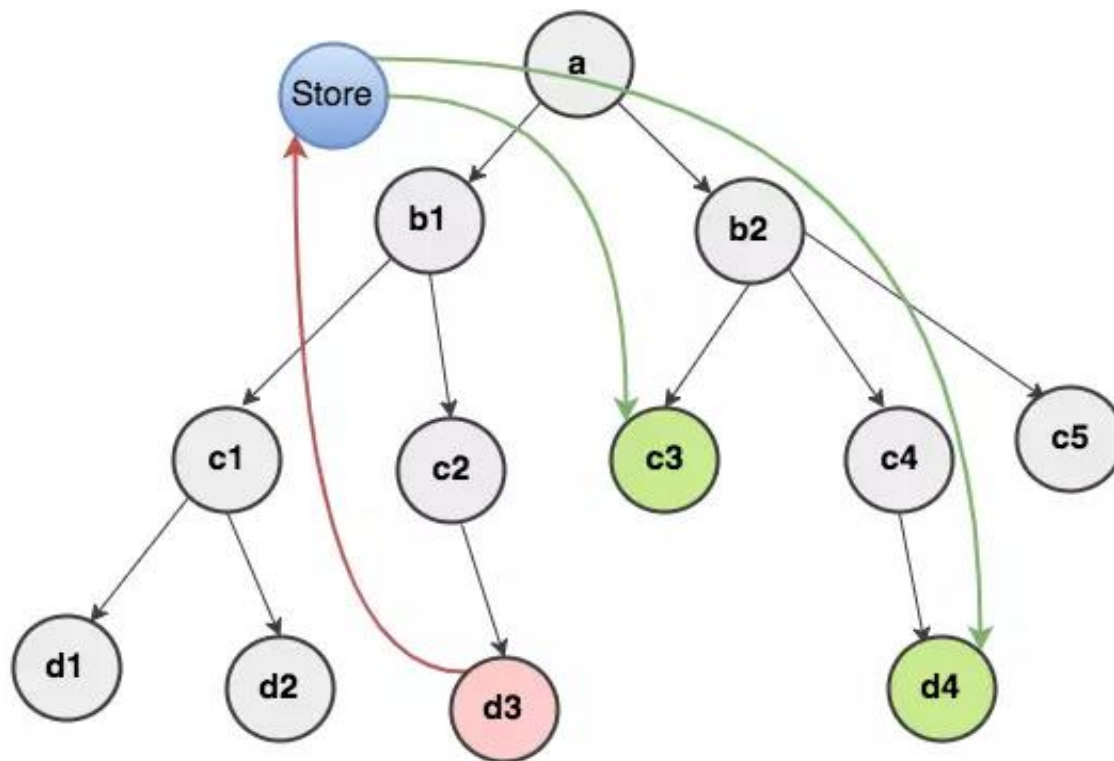
Tổng quan về Redux

❑ Tại sao nên sử dụng Redux?



Tổng quan về Redux

❑ Tại sao nên sử dụng Redux?

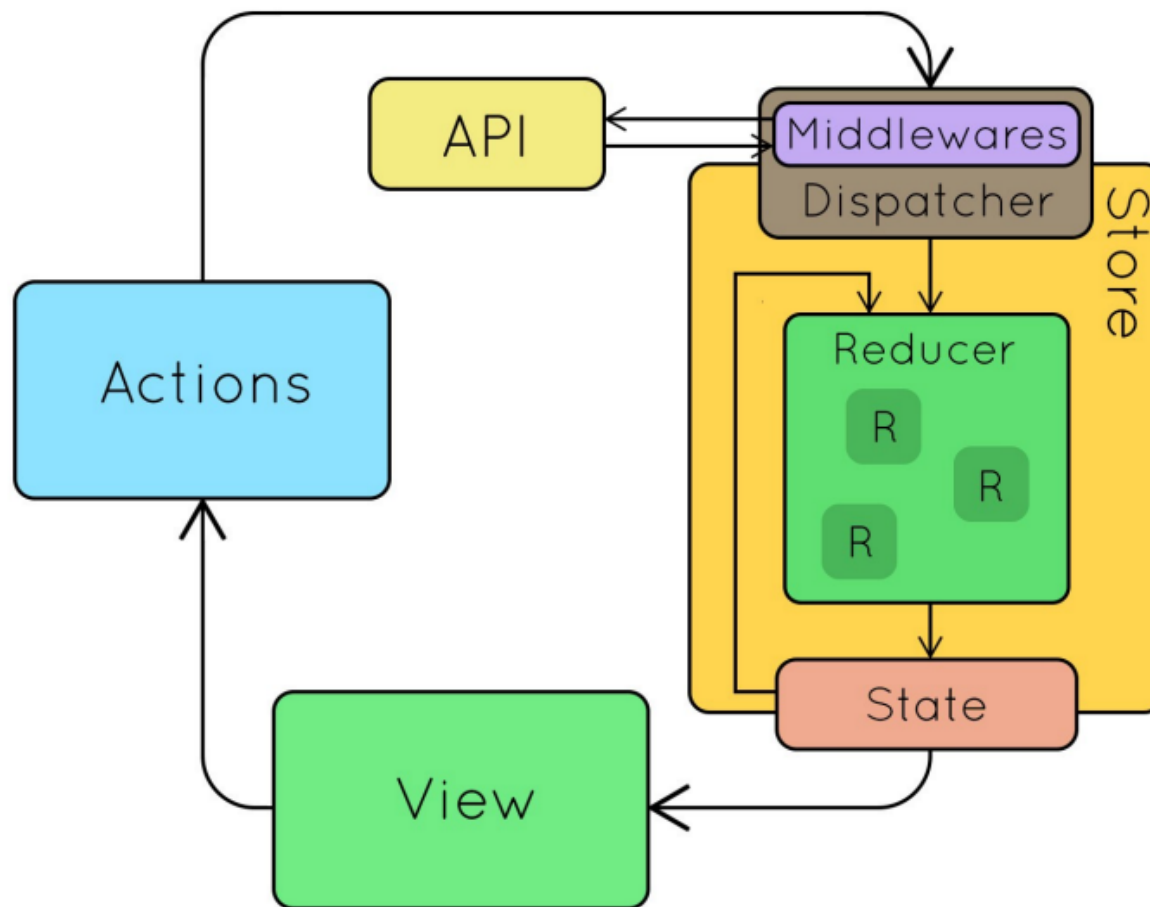


Tổng quan về Redux

□ Redux bao gồm:

- **store**: nơi quản lý các **state** và các **reducer** của ứng dụng.
 - **state**: là biến lưu trữ các dữ liệu hiện tại của ứng dụng.
 - **reducer**: là biến lưu trữ các dữ liệu hiện tại của ứng dụng.
 - **state**: là đối tượng chuyển dữ liệu đến **reducer**, quản lý **middlewares**.
- **action**: là đối tượng javascript đơn giản (**plain js object**) mô tả hành động.
- **view**: là thành phần mô tả phần giao diện của một ứng **JavaScript** bất kỳ (có thể sử dụng **React, Angular, Vue,...** hoặc **JavaScript** thuần cũng được)

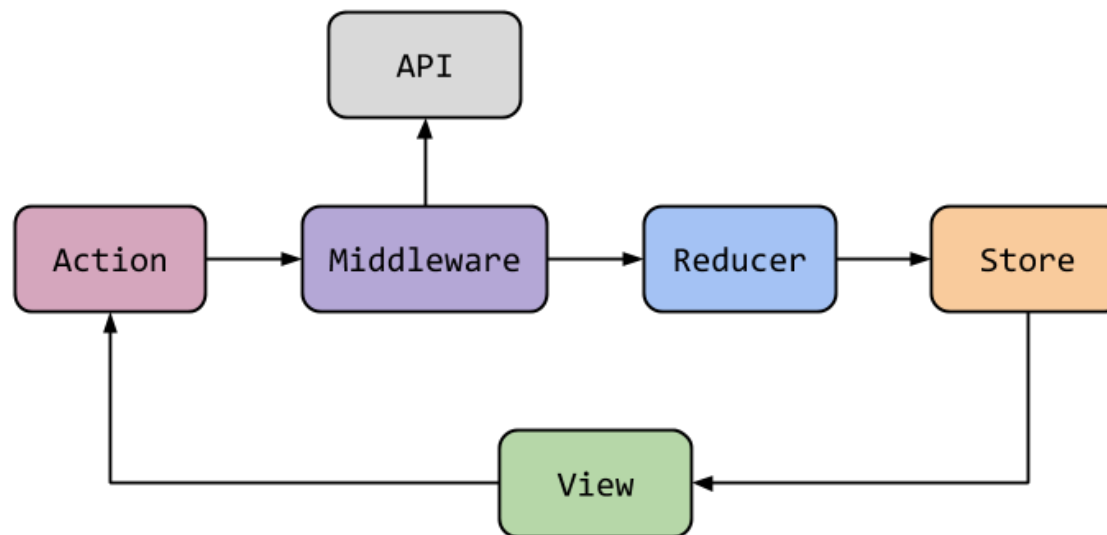
Redux Workflow



Redux Middlewares

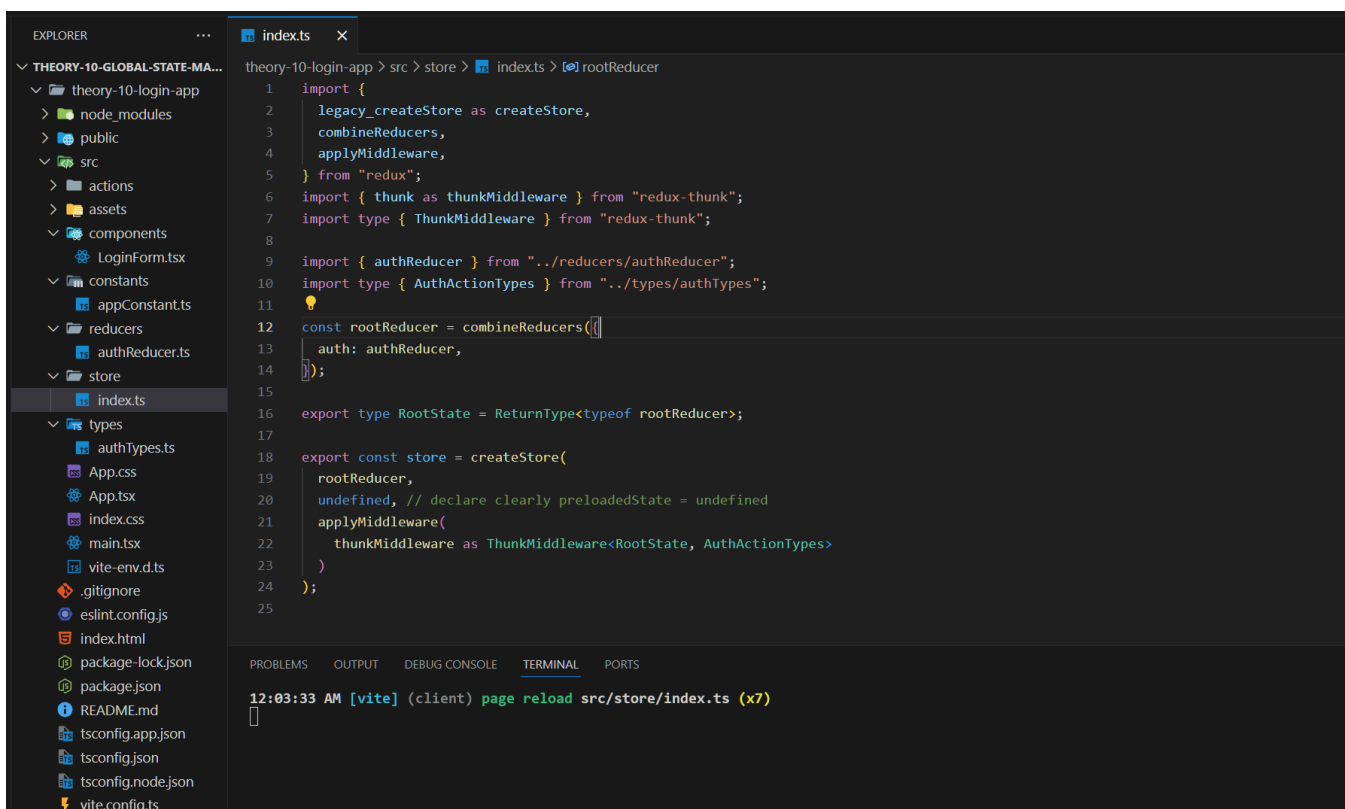
❑ Middleware là gì?

- **Middleware** là một thành phần (tầng) cho phép ta can thiệp vào giữa thời điểm **dispatch** một **action** và thời điểm **action** đến được **reducer**.
- Một số thư viện để làm việc với **Middleware**: **redux-thunk**, **redux-saga**,...



Triển khai Redux Thunk

❑ Hiện thực Login App với Redux Thunk:



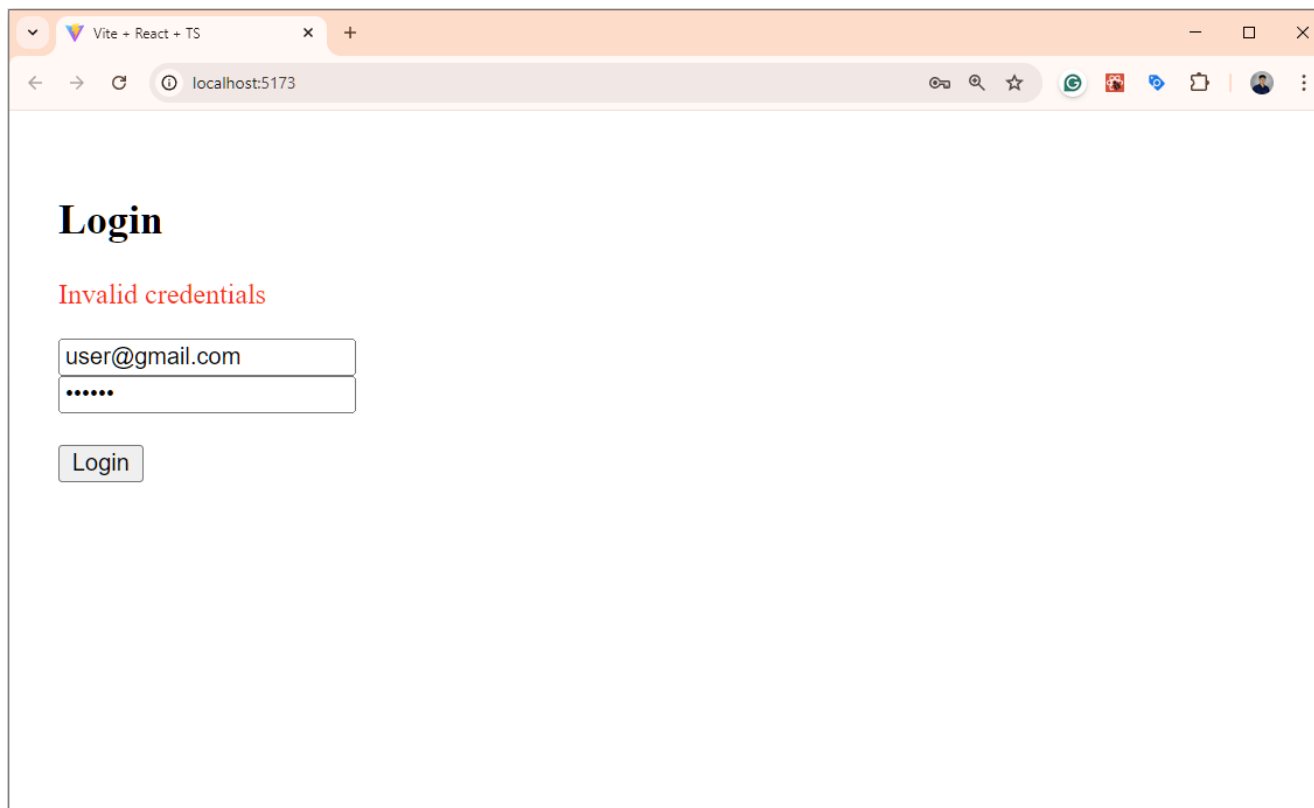
```
EXPLORER
...
theory-10-global-state-MA...
  theory-10-login-app
    node_modules
    public
    src
      actions
      assets
      components
        LoginForm.tsx
      constants
        appConstant.ts
      reducers
        authReducer.ts
      store
        index.ts
      types
        authTypes.ts
        App.css
        App.tsx
        index.css
        main.tsx
        vite-env.d.ts
      .gitignore
      eslint.config.js
      index.html
      package-lock.json
      package.json
      README.md
      tsconfig.app.json
      tsconfig.json
      tsconfig.node.json
      vite.config.ts

theory-10-login-app > src > store > index.ts > rootReducer
1 import {
2   legacy_createStore as createStore,
3   combineReducers,
4   applyMiddleware,
5 } from "redux";
6 import { thunk as thunkMiddleware } from "redux-thunk";
7 import type { ThunkMiddleware } from "redux-thunk";
8
9 import { authReducer } from "../reducers/authReducer";
10 import type { AuthActionTypes } from "../types/authTypes";
11
12 const rootReducer = combineReducers({
13   auth: authReducer,
14 });
15
16 export type RootState = ReturnType<typeof rootReducer>;
17
18 export const store = createStore(
19   rootReducer,
20   undefined, // declare clearly preloadedState = undefined
21   applyMiddleware(
22     thunkMiddleware as ThunkMiddleware<RootState, AuthActionTypes>
23   )
24 );
25
```

12:03:33 AM [vite] (client) page reload src/store/index.ts (x7)

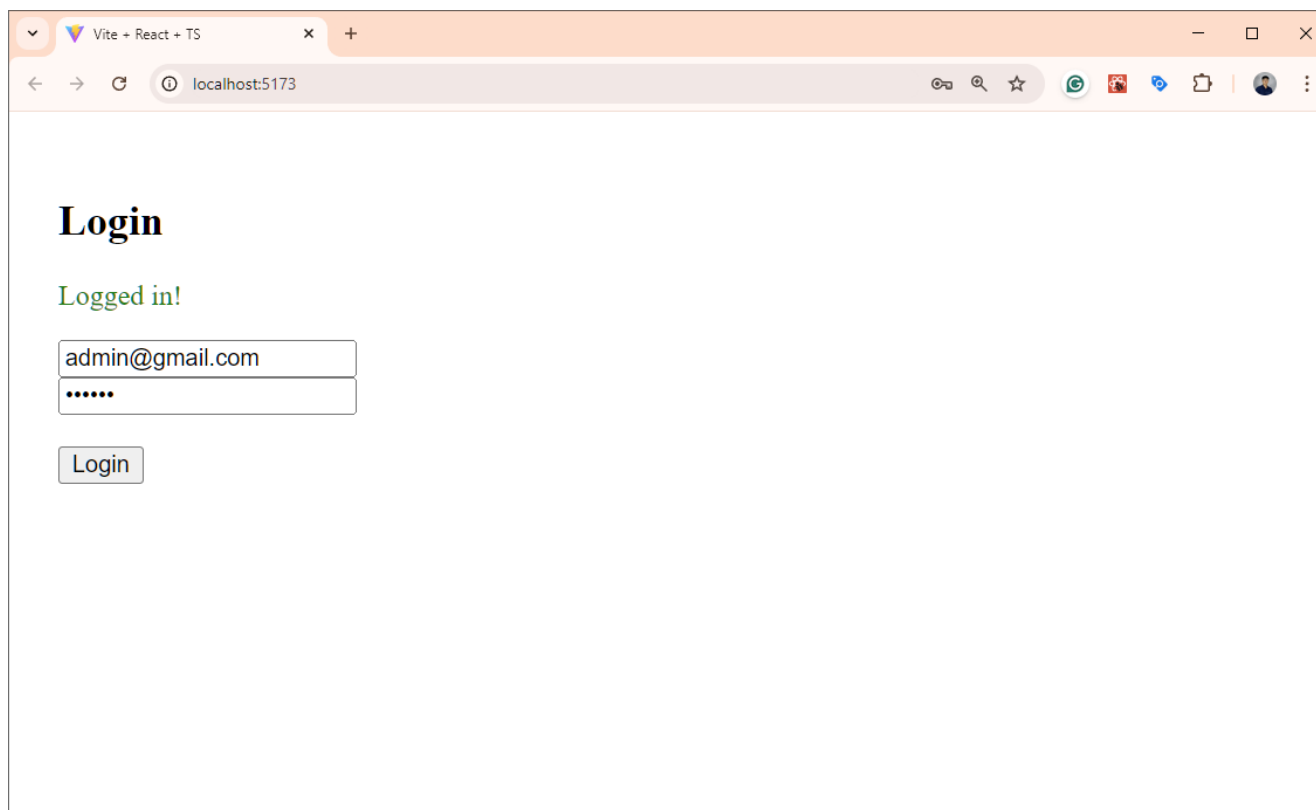
Triển khai Redux Thunk

❑ Hiện thực Login App với Redux Thunk:



Triển khai Redux Thunk

❑ Hiện thực Login App với Redux Thunk:



Tổng quan về Redux Toolkit

❑ Giới thiệu Redux Toolkit:

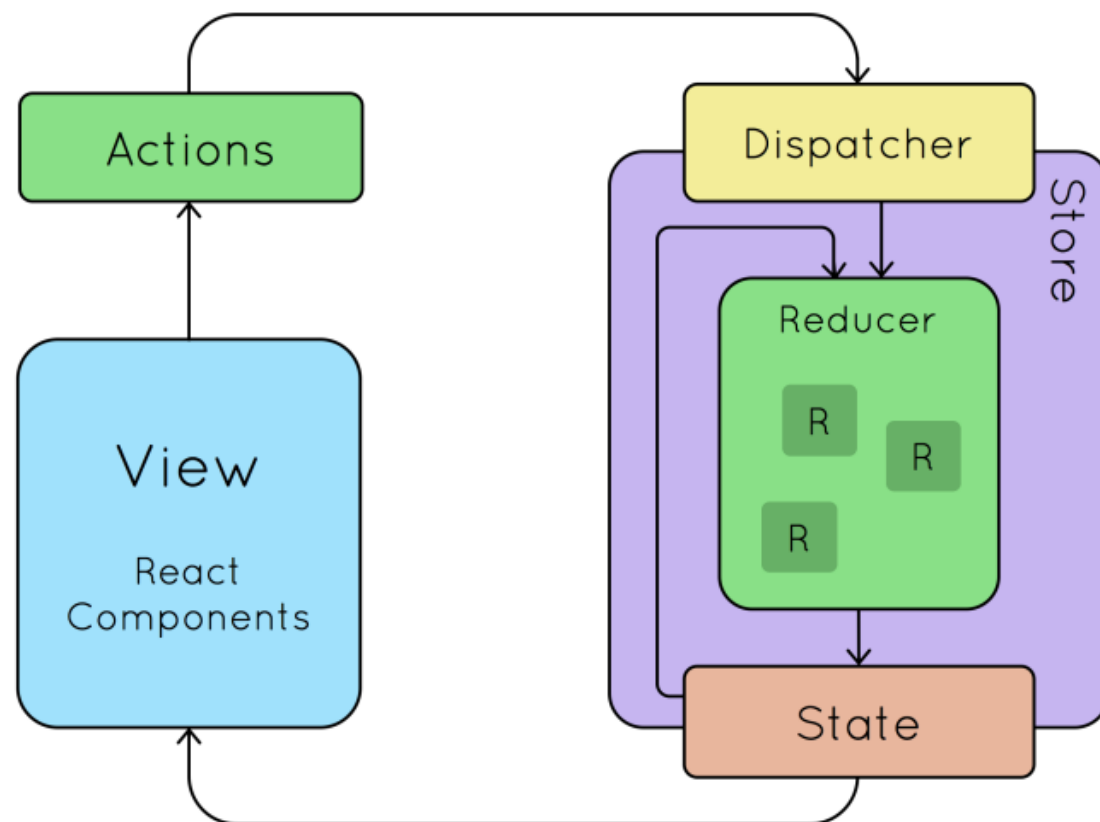
○ **Redux Toolkit** (RTK) là một thư viện **JavaScript** giúp triển khai **Redux** tốt hơn, dễ dàng và đơn giản hơn **Redux** thuần hoặc Redux với các **Middleware**.



Tổng quan về Redux Toolkit

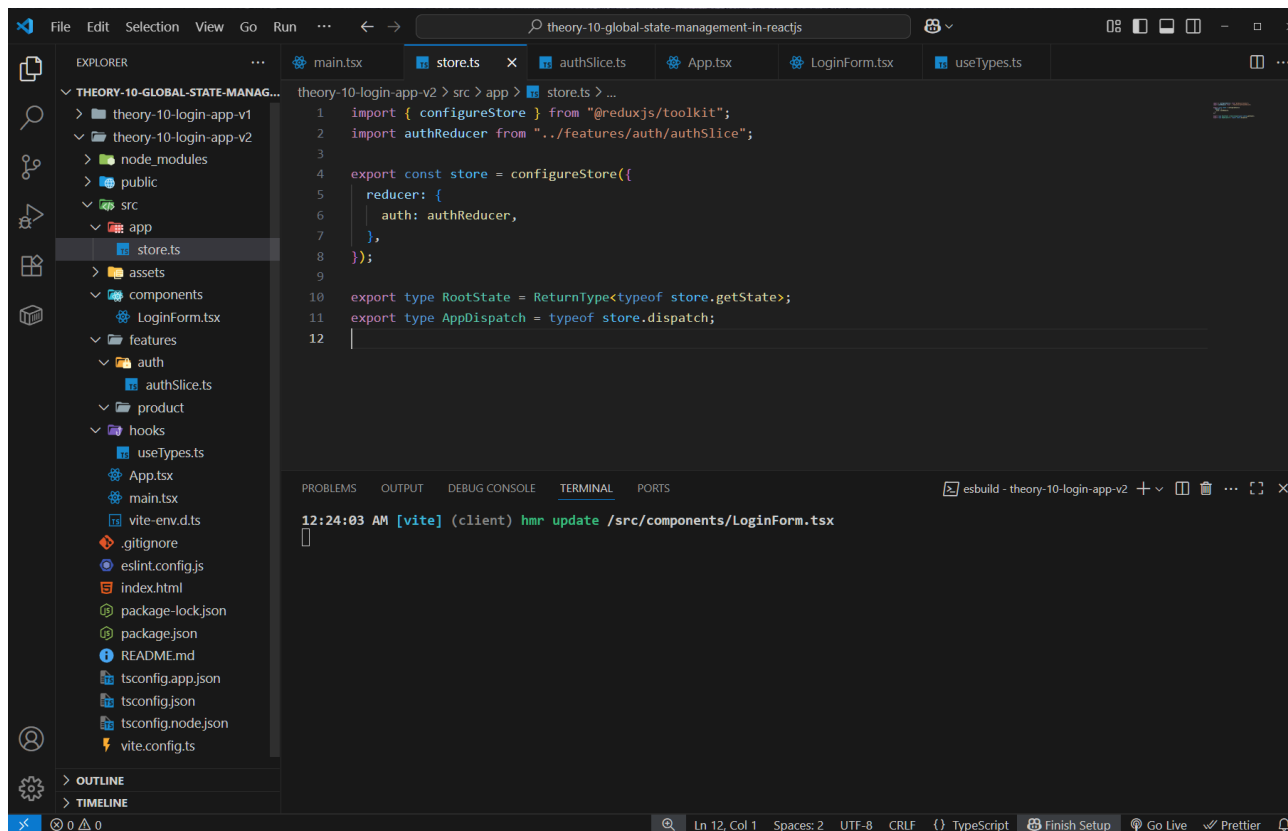
❑ Vì sao nên sử dụng Redux Toolkit?

- Việc cấu hình một **Redux store** quá nhiều thành phần phức tạp.
- Phải cài đặt thêm nhiều thư viện **Redux** khác để sử dụng một số tác vụ bổ sung.
- **Redux** yêu cầu quá nhiều mã soạn sẵn, mã nguồn còn dài.



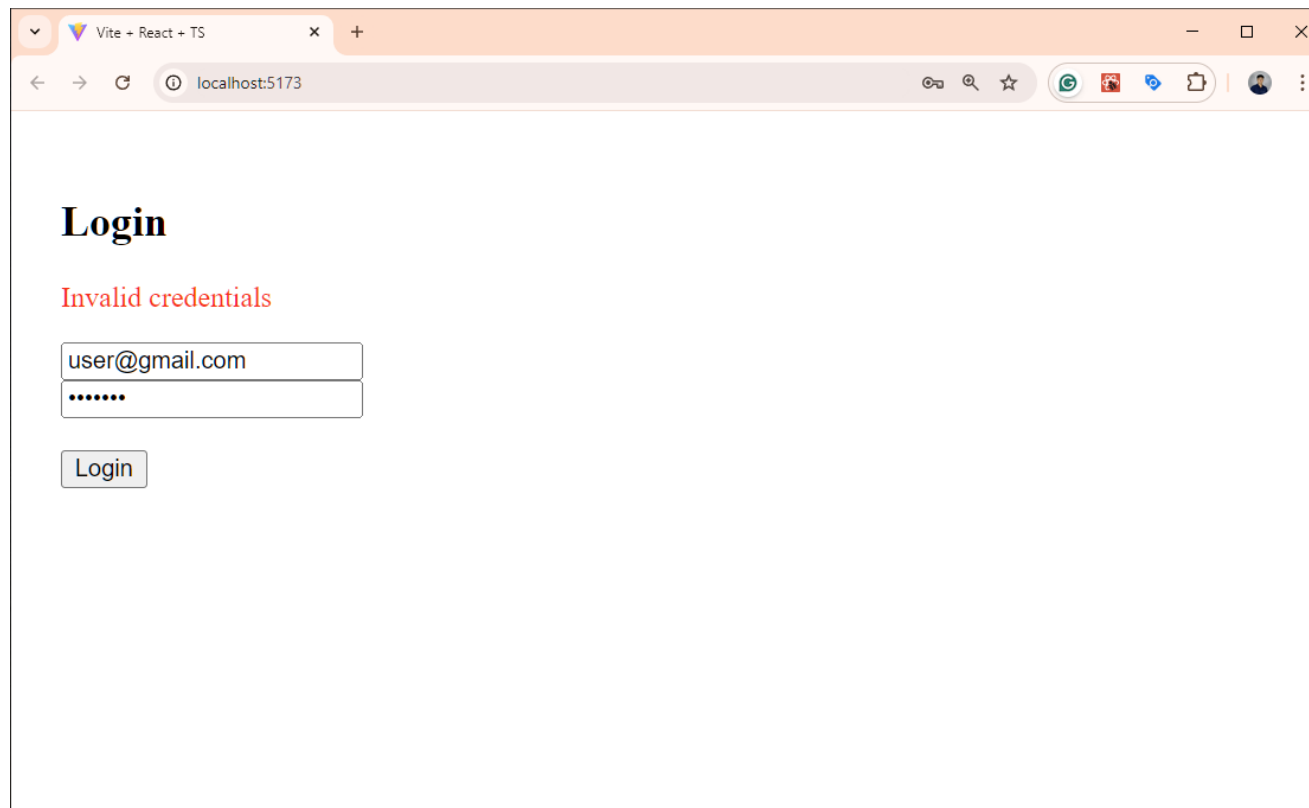
Triển khai Redux Toolkit

❑ Hiện thực Login App với Redux Toolkit:



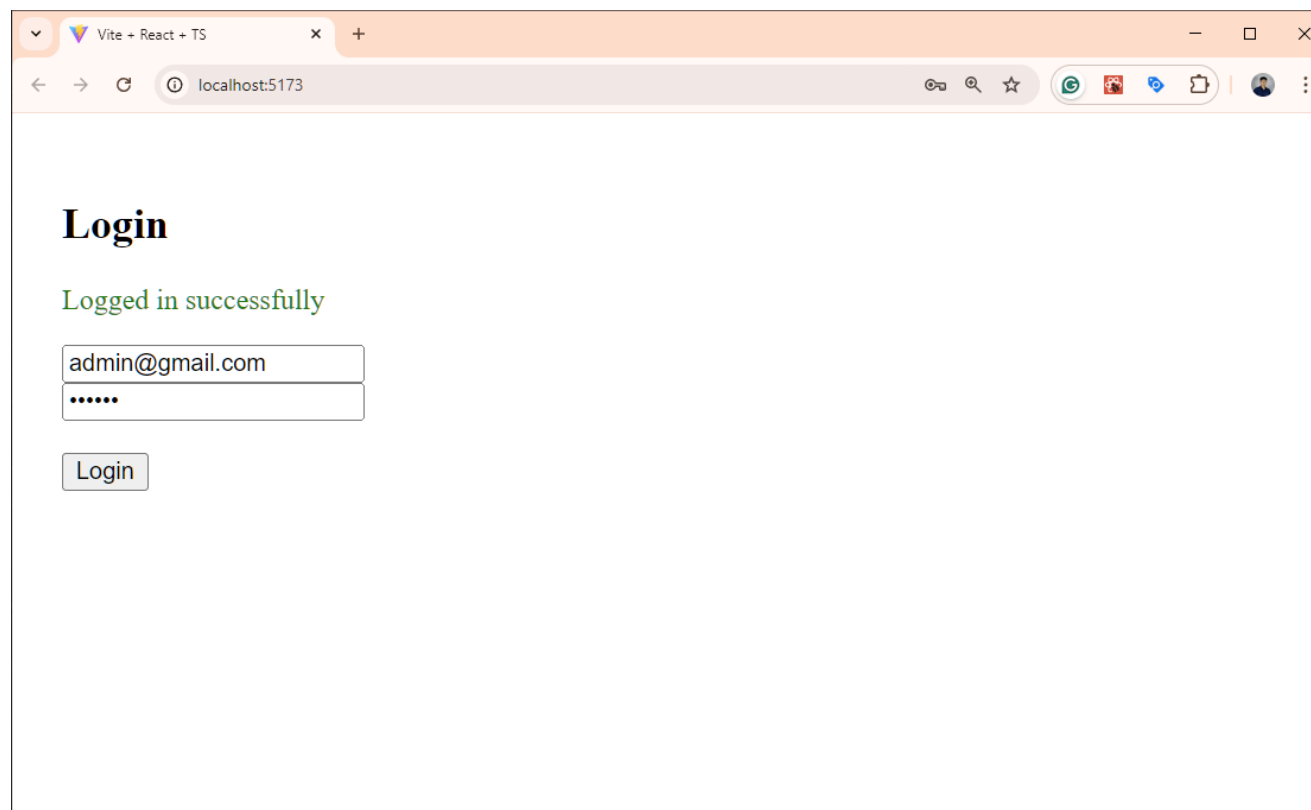
Triển khai Redux Toolkit

❑ Hiện thực Login App với Redux Toolkit:



Triển khai Redux Toolkit

❑ Hiện thực Login App với Redux Toolkit:



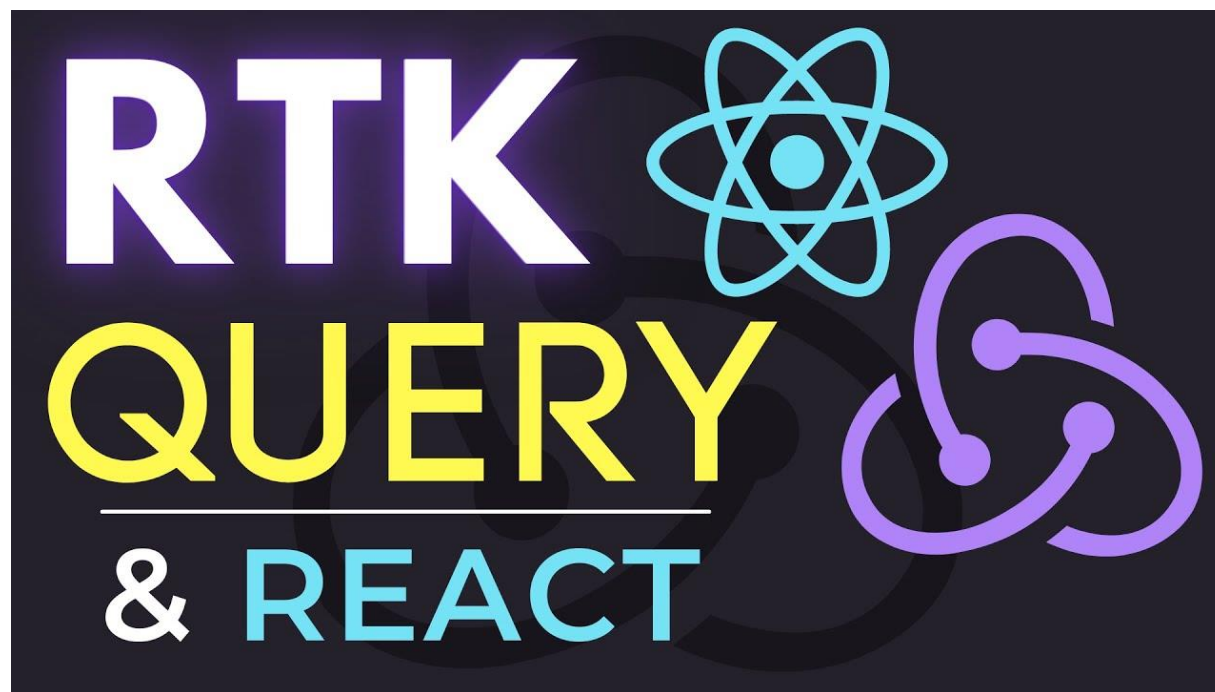
Redux Toolkit Query

❑ Giới thiệu Redux Toolkit Query:

- **RTK Query** là một addon trong bộ thư viện **Redux Toolkit**.

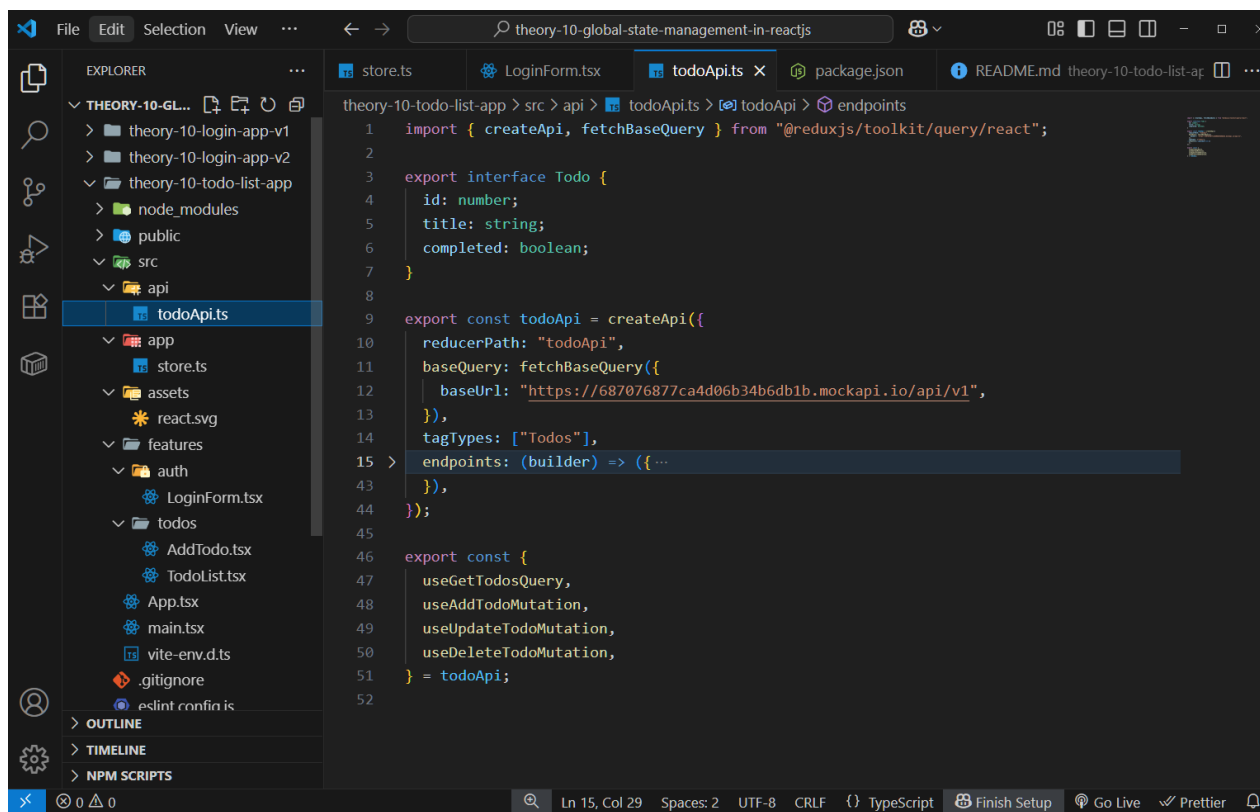
- Nó giúp chúng ta thực hiện data fetching một cách đơn giản hơn thay vì sử dụng **createAsyncThunk** để thực hiện async action.

- Chú ý **RTK Query** là dùng để query (kết nối API), chứ không phải dùng để code async trong Redux thay cho **createAsyncThunk**



Triển khai Redux Toolkit Query

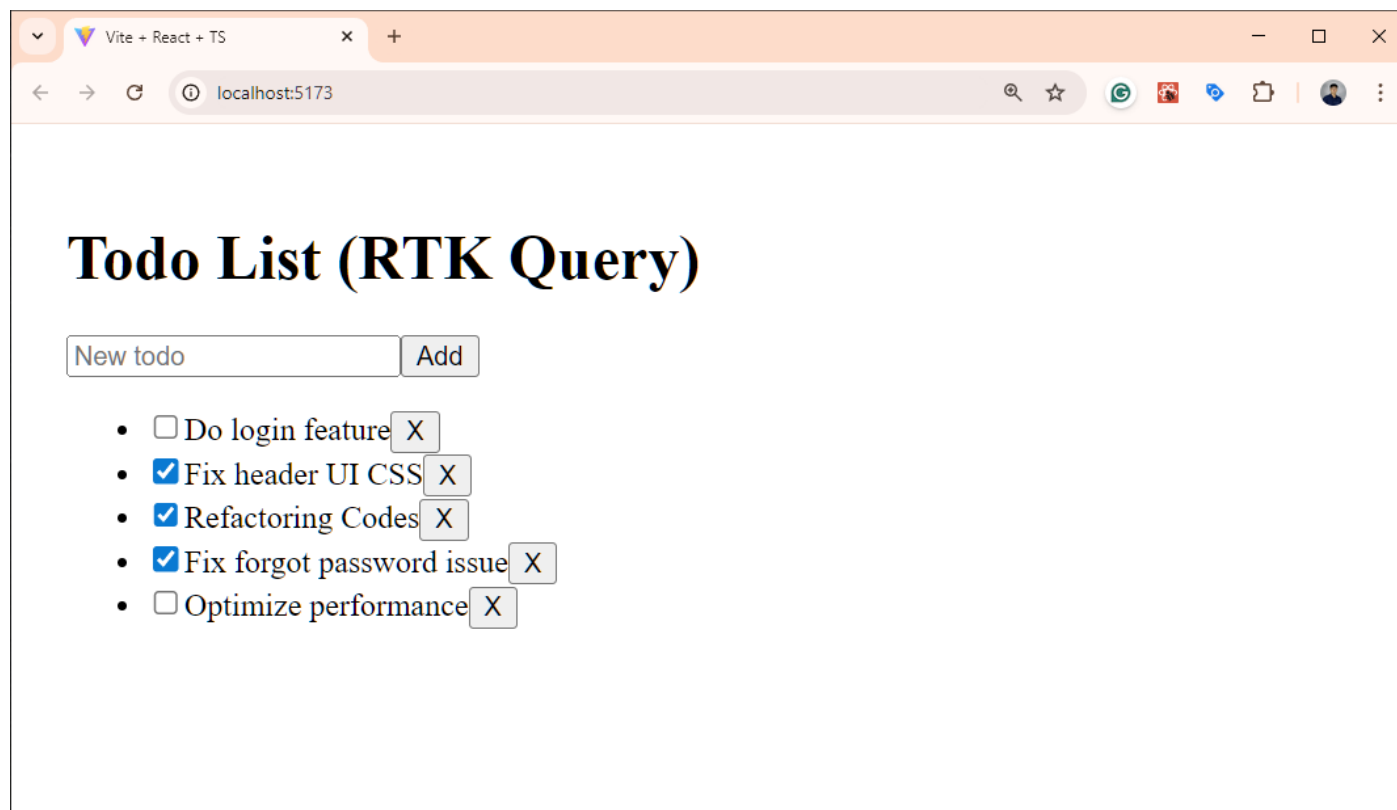
❑ Hiện thực Login App với Redux Toolkit Query:



```
1 import { createApi, fetchBaseQuery } from "@reduxjs/toolkit/query/react";
2
3 export interface Todo {
4   id: number;
5   title: string;
6   completed: boolean;
7 }
8
9 export const todoApi = createApi({
10   reducerPath: "todoApi",
11   baseQuery: fetchBaseQuery({
12     baseUrl: "https://687076877ca4d06b34b6db1b.mockapi.io/api/v1",
13   }),
14   tagTypes: ["Todos"],
15   endpoints: (builder) => ({
16     // ...
17   }),
18 });
19
20 export const {
21   useGetTodosQuery,
22   useAddTodoMutation,
23   useUpdateTodoMutation,
24   useDeleteTodoMutation,
25 } = todoApi;
```

Triển khai Redux Toolkit Query

❑ Hiện thực Login App với Redux Toolkit Query:



Tóm tắt bài học

- ☐ Tổng quan về Redux
- ☐ Redux Workflow
- ☐ Triển khai React Redux
- ☐ Redux Middlewares
- ☐ Triển khai Redux Thunk
- ☐ Tổng quan về Redux Toolkit
- ☐ Redux Toolkit Query
- ☐ Triển khai RTK Query

