

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO
THỰC TẬP CƠ SỞ

Giảng viên: Kim Ngọc Bách

Tuần 2: 22/3/2025

Sinh viên: Nguyễn Khắc Trường B22DCCN884

Hà Nội, 3/2024

MỤC LỤC

A.	Kiến thức học được ở tuần này.	1
	Giới Thiệu	1
	Phần 1: Khái niệm về VueJS	1
	Phần 2: Single-page application	2
	Phần 3: Cấu trúc thư mục VueJS.....	2
B.	Tiến độ dự án	3

A.Kiến thức học được ở tuần này.

Giới Thiệu

Dự án mà em định triển khai sẽ có phần Frontend sử dụng framework VueJS vậy nên tuần này em đã tìm hiểu qua framework này.

Phần 1: Khái niệm về VueJS

1.Khái niệm:

VueJS, hay Vue.js, là một framework JavaScript tiên tiến được dùng để xây dựng giao diện người dùng (UI). Không giống như các framework monolithic, ngay từ đầu Vue.js đã được định hình để lập trình viên có thể áp dụng từng bước. Thư viện cốt lõi chỉ tập trung vào lớp view (view layer), dễ dàng học và tích hợp với nhiều thư viện hoặc dự án khác. Ngoài ra, Vue.js còn có khả năng cung cấp các ứng dụng đơn trang (Single-Page Applications) phức tạp khi được sử dụng kết hợp với các công cụ hiện đại và thư viện hỗ trợ.

2.Uưu nhược điểm:

- Ưu điểm:

- Nhẹ và hiệu quả: Vue.js có kích thước nhỏ gọn, giúp ứng dụng tải nhanh, tăng trải nghiệm người dùng và hiệu suất.
- Dễ học và sử dụng: Cú pháp đơn giản, tài liệu rõ ràng và cấu trúc linh hoạt giúp Vue.js trở thành lựa chọn lý tưởng cho cả người mới bắt đầu và lập trình viên có kinh nghiệm.
- Cấu trúc component: Việc chia ứng dụng thành các thành phần nhỏ giúp quản lý code dễ dàng hơn, tăng tính tái sử dụng và khả năng bảo trì.

- Nhược điểm:

- Cộng đồng và tài liệu: Cộng đồng Vue.js chưa phát triển rộng rãi như các framework khác, đặc biệt là về tài liệu tiếng Anh. Điều này gây khó khăn trong việc giải quyết vấn đề.

- Khả năng mở rộng: Vue.js phù hợp hơn với các dự án nhỏ và vừa. Khi quy mô dự án lớn lên, việc thiếu hỗ trợ từ các công ty lớn và cộng đồng có thể gây ra một số khó khăn trong việc bảo trì và phát triển.
- Hệ sinh thái plugin: So với các framework như Angular và React, Vue.js có ít plugin hơn. Điều này có thể hạn chế các tính năng và chức năng mà bạn muốn tích hợp vào ứng dụng.

Phần 2: Single-page application

Single-page Application là một ứng dụng hoạt động trên browser và không cần bắt buộc phải reload lại khi sử dụng. Trang web chỉ cần load trang một lần, sau đó sẽ sử dụng javascript để load những dữ liệu mà bạn cần dùng. Trái ngược với Multi-page Application, chúng ta có một phong cách hiển thị rất "truyền thống" là mọi dữ liệu sẽ được render ra từ mã HTML, CSS và cả Javascript và trả lại về người dùng hay được gọi là Server Side Rendering. Mỗi lần chuyển trang, dù tốc độ mạng đến cỡ nào, chúng ta vẫn có cảm giác khựng lại do mỗi lần gửi request, sever đều phải thực hiện công việc như trên để trả lại cho người dùng. Bất cứ thay đổi nhỏ nhất cũng sẽ bắt buộc trang web phải render lại trang. Và VueJs cũng là framework theo kiểu Single-page Application.

Phần 3: Cấu trúc thư mục VueJS

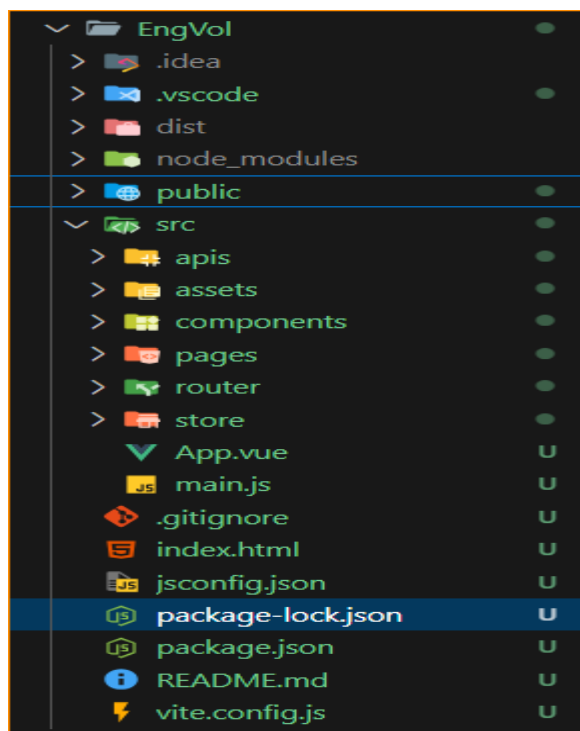
Khi bắt đầu một dự án Vue, cấu trúc thư mục thường được tạo ra bởi Vue CLI (Command Line Interface). Cấu trúc này giúp tổ chức mã nguồn một cách rõ ràng và dễ quản lý. Dưới đây là một số thư mục và tệp tin quan trọng:

- **node_modules/:**
 - Đây là thư mục chứa tất cả các thư viện và gói phụ thuộc mà dự án cần.
 - Không nên chỉnh sửa trực tiếp nội dung trong thư mục này.
- **public/:**
 - Chứa các tệp tĩnh (static assets) như index.html, favicon, v.v.
 - Các tệp tin trong thư mục này sẽ được sao chép trực tiếp vào thư mục build mà không qua xử lý của Webpack.
- **src/:**
 - Đây là thư mục chứa mã nguồn chính của ứng dụng.
 - Các thư mục và tệp tin quan trọng trong src/ bao gồm:
 - **assets/:** Chứa các tệp như hình ảnh, phông chữ, v.v.

- **components/**: Chứa các thành phần (components) Vue có thể tái sử dụng.
- **views/ hoặc pages/**: Chứa các thành phần đại diện cho các trang hoặc khung nhìn của ứng dụng.
- **router/**: Chứa các cấu hình định tuyến (routes) của ứng dụng.
- **store/**: (Nếu sử dụng Vuex) Chứa các tệp tin quản lý trạng thái của ứng dụng.
- **App.vue**: Thành phần gốc của ứng dụng.
- **main.js**: Điểm vào của ứng dụng, nơi Vue được khởi tạo và gắn vào DOM.
- **apis/**: Chứa các hàm axios call api đến backend.
- **package.json**:
 - Chứa thông tin về dự án, các gói phụ thuộc và các script.
- **vue.config.js**:
 - Tệp cấu hình cho dự án Vue.

B. Tiến độ dự án

- Tạo được cấu trúc thư mục vueJS



- Phác họa phần Frontend mà em định xây dựng ([Link đến Figma](#))
- Dưới đây là 1 số page chính:

+ Đăng ký, đăng nhập

1. Login

Login

Email

Password

[forgot password?](#)

Login

[Forgot your password?](#)

2. Sign in

Sign up

Email

Password

Repeat the password

Birthday

Country

Sign in

Already have an account? [Login](#)

+ Trang chủ

Search for flashcards

+

It's time to review...

Recent

Environment
5 terms
ptitstudent1

Environment
5 terms
ptitstudent1

Environment
5 terms
ptitstudent1

More...

Your Library

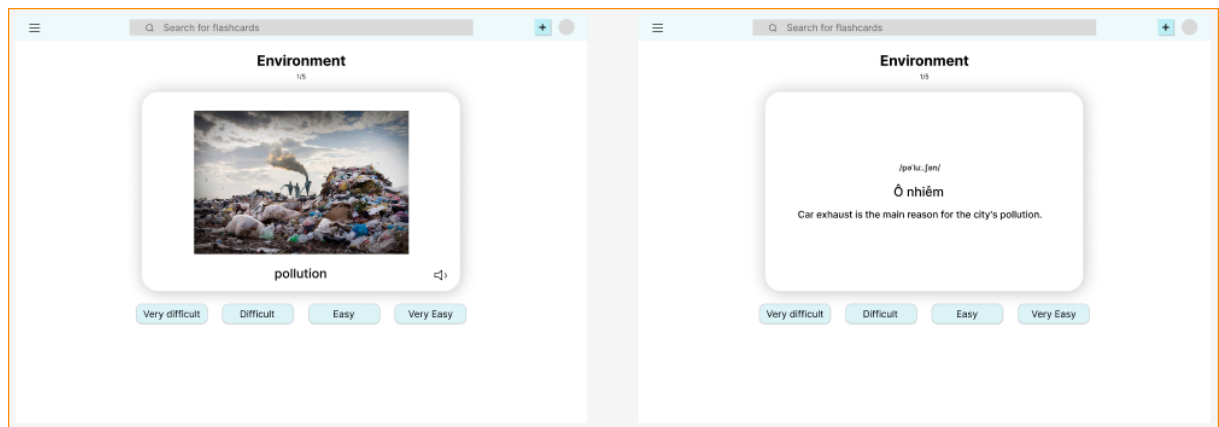
Environment
5 terms
ptitstudent1

Environment
5 terms
ptitstudent1

Environment
5 terms
ptitstudent1

More...

+ Page học



+Page game

