**Lê Quốc Dũng – 21520739**

**Phạm Thanh Đồng - 21520724**

**Proposal: Micro Frontend Architecture**

1. **Giới thiệu**

**Micro Frontend Architecture** là một phương pháp tiếp cận hiện đại trong phát triển ứng dụng web, mở rộng từ mô hình **Microservices** ở **backend**, nhưng áp dụng cho **frontend**. Thay vì phát triển một ứng dụng frontend lớn và phức tạp, kiến trúc này cho phép chia nhỏ ứng dụng thành các phần độc lập gọi là micro frontends, mỗi phần có thể được phát triển, triển khai và duy trì một cách riêng biệt.

Mỗi **micro frontend** chịu trách nhiệm cho một **chức năng** hoặc **module** cụ thể trong ứng dụng, có thể được phát triển bởi các đội nhóm khác nhau, sử dụng các công nghệ hoặc framework frontend khác nhau như React, Vue, Angular... Điều này giúp tối ưu hóa quá trình phát triển, tăng khả năng mở rộng và dễ dàng bảo trì, đồng thời cho phép các thành phần của ứng dụng phát triển và cải tiến một cách độc lập.

**Mục tiêu nghiên cứu:**

* Nghiên cứu nguyên lý của Micro Frontend Architecture: Tìm hiểu cách thức hoạt động, lợi ích và thách thức của kiến trúc Micro Frontend trong việc xây dựng các ứng dụng web lớn.
* Ứng dụng Micro Frontend trong phát triển web: Khám phá các phương pháp chia nhỏ ứng dụng frontend thành các phần độc lập, có thể triển khai và phát triển riêng lẻ.
* So sánh Micro Frontend với kiến trúc monolithic:
  + Monolithic Frontend: toàn bộ ứng dụng đều được xây dựng và triển khai dưới dạng một khối duy nhất. Có nghĩa là nếu có thay đổi nhỏ thì cần phải xây dựng và triển khai lại toàn bộ ứng dụng
  + Micro Frontend: ứng dụng được chia nhỏ ra thành các module có thể triển khai và phát triển độc lập, không làm ảnh hưởng đến module khác.
* Để áp dụng Micro Frontend thì chúng ta có thể sử dụng nhiều cách khác nhau
  + Sử dụng **iframe**: mỗi Micro Frontend được tải trong một iframe riêng
  + Sử dụng **Web Components**: là tập hợp các tiêu chuẩn của web để tạo ra các phần tử HTML tùy chỉnh có thể tái sử dụng trong bất kỳ ứng dụng nào.
  + Sử dụng **Server side Includes**: là kỹ thuật cho phép server kết hợp nội dung từ các phần khác nhau của ứng dụng thành một trang duy nhất trước khi gửi đến client.
  + Sử dụng **Webpack và Module Federation**: ở phương pháp này **Webpack** sẽ đảm nhiệm việc liên kết các các module Micro Frontend riêng biệt lại với nhau trong quá trình runtime. **Module Federation** cho phép chia sẻ các thư viện, code giữa các module với nhau.
* Đánh giá hiệu quả triển khai thực tiễn: Thử nghiệm và đánh giá tính hiệu quả của Micro Frontend qua các tình huống thực tế.

1. **Lý do chọn đề tài**

* **Giải quyết vấn đề của kiến trúc monolithic**:

Đối với những dự án có nhiều teams làm việc chung ở một code base sẽ xảy ra những conflicts, khi dự án lớn ra thì thời gian build và deploy một dự án sẽ kéo dài. Kiến trúc Micro Frontend giúp chia nhỏ ứng dụng thành các module độc lập, giảm thiểu khó khăn trong phát triển và bảo trì các ứng dụng frontend lớn.

* **Tối ưu hóa quy trình phát triển:** Micro Frontend cho phép các teams phát triển hoạt động độc lập, tăng tốc độ phát triển và triển khai, từ đó nâng cao hiệu suất làm việc và khả năng mở rộng của hệ thống.
* **Xu hướng phát triển web hiện đại:** Micro Frontend ngày càng trở nên phổ biến trong các tổ chức lớn để xử lý các hệ thống phức tạp, phù hợp với nhu cầu phát triển ứng dụng linh hoạt và dễ dàng nâng cấp.
* **Khả năng sử dụng đa dạng công nghệ:** Cho phép sử dụng nhiều framework khác nhau trong cùng một ứng dụng, giúp tận dụng tối đa lợi thế của từng công nghệ phù hợp với từng phần cụ thể của hệ thống.

1. **Ứng dụng Demo**
   1. **Giới thiệu ứng dụng**

* Để có thể demo cho mô hình MicroFrontend nhóm đã chọn xây dựng Website thương mại điện tử.
  1. **Lý do chọn dự án**
* **Độc lập triển khai**: Cập nhật từng phần mà không ảnh hưởng đến toàn bộ ứng dụng.
* **Tối ưu cập nhật sản phẩm**: Phần "Danh sách sản phẩm" và "Chi tiết sản phẩm" có thể được phát triển và cập nhật riêng lẻ mà không ảnh hưởng đến giỏ hàng hoặc thanh toán, giúp cập nhật sản phẩm nhanh chóng khi cần.
* **Quản lý quảng cáo và khuyến mãi:** Phần banner khuyến mãi hoặc tính năng "Sản phẩm đề xuất" có thể được phát triển dưới dạng micro-frontend riêng, cho phép thay đổi nhanh mà không cần can thiệp vào hệ thống khác.
* **Phân chia công việc phát triển hiệu quả:** Có thể làm việc đồng thời trên giỏ hàng, thanh toán, và đăng nhập mà không gây ảnh hưởng lẫn nhau, giúp website thương mại điện tử mở rộng và thay đổi linh hoạt hơn..
  1. **Phạm vi ứng dụng**
* Nền tảng hỗ trợ: Triển khai ứng dụng trên nền tảng Website.
* Chức năng chính:
  + **Danh sách sản phẩm**: Hiển thị tất cả các sản phẩm.
  + **Chi tiết sản phẩm**: Hiển thị thông tin chi tiết bao gồm: hình ảnh, mô tả, giá, màu sắc, kích cỡ của sản phẩm đó.
  + **Giỏ hàng**: Cho phép người dùng thêm sản phẩm vào giỏ hàng. Hiển thị tóm tắt giỏ hàng với tổng giá tiền và danh sách các sản phẩm đã chọn.
  + **Trang thanh toán**: Thu thập thông tin vận chuyển và thanh toán từ người dùng. Tóm tắt đơn hàng trước khi xác nhận mua hàng.
  + **Xác thực người dùng**: Cho phép người dùng đăng ký, đăng nhập và quản lý hồ sơ cá nhân. Hỗ trợ đăng nhập qua tài khoản Google.
  + **Tìm kiếm và lọc:** Cung cấp thanh tìm kiếm cho người dùng tìm sản phẩm bằng từ khóa. Cho phép lọc theo danh mục, màu sắc, kích cỡ.
  1. **Áp dụng với Micro Front-end**
* Chia ứng dụng thành các Micro Front-end khác nhau:
  + Micro Front-end Products: đảm nhiệm các trang Home, Danh sách sản phẩm, lọc sản phẩm, Header và Footer, xác thực người dùng.
  + Micro Front-end Details: đảm nhiệm trang chi tiết sản phẩm.
  + Micro Front-end Checkout: đảm nhiệm các trang giỏ hàng, thanh toán.
* Trong mỗi trang đề có sự xuất hiện của những Micro Front-end khác nhau:
  + Trang Home: yêu cầu Micro Front-end Products, Checkout.
  + Trang Danh sách sản phẩm: yêu cầu Micro Front-end Products, Checkout.
  + Trang chi tiết sản phẩm: yêu cầu Micro Front-end Details, Products, Checkout.
  + Trang giỏ hàng, thanh toán: yêu cầu Micro Front-end Products, Checkout.

|  |  |  |
| --- | --- | --- |
| **Trang** | **Chức năng chính** | **Micro Front-end cần thiết** |
| Home | Trang chào mừng, hiển thị các sản phẩm nổi bật. | Products, Checkout |
| Danh sách sản phẩm | Hiển thị danh sách sản phẩm có trong cửa hàng, cho phép lọc sản phẩm. | Products, Checkout |
| Chi tiết sản phẩm | Hiển thị chi tiết sản phẩm | Products, Checkout, Details |
| Giỏ hàng, thanh toán | Cho phép thanh toán sản phẩm | Products, Checkout. |

* 1. **Mục tiêu dự án**
* Xây dựng ứng dụng giúp người dùng dễ dàng mua sắm trực tuyến với các tính năng như tìm kiếm sản phẩm, thêm vào giỏ hàng, thanh toán nhanh chóng, và quản lý lịch sử đơn hàng.
* Giúp nhóm tìm hiểu được Micro Frontend và ứng dụng vào dự án thực tế.
  1. **Đối tượng sử dụng**
* Người dùng (Khách hàng):
  + Quản lý thông tin cá nhân, bao gồm cập nhật, chỉnh sửa thông tin tài khoản, địa chỉ giao hàng, và phương thức thanh toán.
  + Xem và quản lý giỏ hàng, theo dõi lịch sử đơn hàng và trạng thái các đơn hàng đã đặt.
* Quản trị viên (Admin):
  + Quản lý thông tin của toàn bộ người dùng, bao gồm khả năng thêm, sửa, hoặc xóa tài khoản người dùng.
  + Quản lý sản phẩm, danh mục sản phẩm, các chương trình khuyến mãi, và theo dõi doanh số bán hàng.
  + Quản lý đơn hàng và theo dõi tình trạng giao hàng.
  1. **Phương pháp thực hiện**
* Tìm hiểu về Micro Frontend, Reactjs, các công cụ hosting.
* Tìm hiểu các công cụ hỗ trợ: Github, Trello.
* Tìm hiểu về **Webpack5**: công cụ giúp bundling (gói gọi) các tệp js, html, css, và các tài nguyên khác trong ứng dụng web. Giúp tổ chức và tối ưu hóa mã nguồn, tạo ra một bundle cuối cùng nhỏ gọn có thể tải xuống nhanh chóng khi người dùng truy cập trang web.
* Tìm hiểu về **Module Federation**: là tính năng của Webpack5 cho phép các ứng dụng web có thể chia sẻ các phần module Javascript với nhau trong quá trình runtime mà không cần phải xây dựng lại tất cả từ đầu.
* Thu thập yêu cầu thông qua các ứng dụng phổ biến trên website, các ứng dụng có trên thị trường.
* Phân tích và thiết kế ứng dụng.
* Cài đặt và kiểm thử.
  1. **Nền tảng công nghệ**
* Công cụ thiết kế UI: Figma.
* Quản lý source code: Github.
* Ngôn ngữ chính: Javascript.
* Frameworks: React.
* Application shell: Webpack
* Client-Side Integration: Module Federation
* Cơ sở dữ liệu: MongoDB.
  1. **Kết quả mong đợi**
* Hoàn thiện được các chức năng đã đề ra.
* Triển khai ứng dụng được trên website.

1. **Kế hoạch làm việc**

09/09/2024 đến ngày10/12/2024

|  |  |  |  |
| --- | --- | --- | --- |
| **Giai đoạn** | **Thời gian** | **Công việc** | **Kết quả** |
| **Tìm hiểu chủ đề seminar** | **09/09 – 23/09** | * Tìm hiểu các chủ đề công nghệ nghiên cứu để thực hiện * Lên kế hoạch cho các công việc về sau của môn học. | - Viết Proposal về chủ đề công nghệ mình nghiên cứu và ứng dụng phù hợp cho việc xây dựng ứng dụng demo cho dự án. |
| **Nghiên cứu kỹ chủ đề seminar về công nghệ Micro Frontend và ứng dụng nó trong việc xây dựng một ứng dụng.** | **24/09 - 02/10** | * Khảo sát qua những ứng dụng sử dụng MicroFrontend trên thị trường và nghiên cứu cách hoạt động của các ứng dụng đó. * Tìm hiểu cách sử dụng và ứng dụng MicroFrontend cho ứng dụng. | Tìm hiểu Webpack Module Federation, App Shelf, Widget và tìm ra ứng dụng demo phù hợp với việc ứng dụng MicroFrontEnd trong việc xây dựng nó. |
| **Thiết kế hệ thống ứng dụng và phân tích đặc tả yêu cầu ứng dụng** | **03/10 – 17/10** | * Lên thiết kế hệ thống cho việc xây dựng ứng dụng Quản lý nhân sự. * Lên các usecase và đặc tả usecase cho ứng dụng dự án. | Có được bản thiết kế hệ thống ứng dụng cũng như danh sách các usecase cho dự án đó. |
| **Cấu hình cơ bản front-end và thiết kế bảng cơ sở dữ liệu.** | **18/10 – 20/10** | * Cấu hình và setup dự án phù hợp cho việc xây dựng ứng dụng demo. * Xây dựng các sơ đồ lớp, sơ đồ quan hệ thực thể và chi tiết cho các đối tượng được sử dụng bên trong csdl | * Chuẩn bị setup dự án cho ứng dụng. * Bảng mô tả đối tượng trong sơ đồ lớp. |
| **Thiết kế giao diện UI/UX** | **21/10 – 27/10** | * Thiết kế giao diện người dùng, tạo Wireframes | Giao diện người dùng được chia theo các màn hình rõ ràng. |
| **Phát triển giao diện front-end của ứng dụng** | **28/10 – 09/11** | * Triển khai code giao diện theo bản thiết kế Figma. * Cài đặt xử lý luồng đi của các màn hình. | - Các màn hình của ứng dụng.  - Luồng đi, xử lý chuyển trang hợp lý. |
| **Phát triển phần back-end của ứng dụng** | **10/11 – 24/11** | * Xử lý logic cũng như lưu trữ dữ liệu trên các tính năng và màn hình tương ứng. | Các tính năng cơ bản được hoàn thiện. |
| **Kiểm thử, sửa lỗi** | **25/11 – 02/12** | * Kiểm tra hệ thống để tìm lỗi và sự cố tương thích, khắc phục mọi sự cố được tìm thấy. * Tiếp tục thực hiện kiểm tra hệ thống nếu còn phát hiện lỗi. | Hệ thống được gỡ lỗi và thử nghiệm. |
| **Làm slide và chuẩn bị báo cáo cho seminar về Micro FrontEnd và kiểm thử hệ thống lần cuối.** | **03/12 – 10/12** | * Lên kế hoạch báo cáo cuối kì dự án, chuẩn bị slide cho việc seminar cuối kì | * Hệ thống được kiểm tra lại chặt chẽ. * Có slide thuyết trình để báo cáo công việc đã hoàn thành. |

1. **Tài liệu tham khảo**

**(1)** [**https://euristiq.com/micro-frontend-architecture/**](https://euristiq.com/micro-frontend-architecture/)

**(2)**  [**Micro Frontends - extending the microservice idea to frontend development (micro-frontends.org)**](https://micro-frontends.org/)