06-Phát triển và tối ưu hóa website thương mại điện tử về thời trang trẻ em tích hợp gợi ý sản phẩm

1. **Về lựa chọn công nghệ**:

Tại sao các bạn chọn sử dụng MERN Stack (MongoDB, ExpressJS, ReactJS, NodeJS) cho dự án này? Các bạn có thể so sánh các ưu và nhược điểm của MERN Stack với các stack khác như MEAN Stack (MongoDB, ExpressJS, Angular, NodeJS) hoặc LAMP Stack (Linux, Apache, MySQL, PHP)?

1. **Về bảo mật hệ thống**:

Các bạn đã triển khai những biện pháp bảo mật nào để bảo vệ thông tin người dùng và hệ thống? Các bạn có thể giải thích chi tiết về cách sử dụng JSON Web Token (JWT) và những biện pháp bảo mật nào khác đã được áp dụng?

1. **Về hiệu suất và tối ưu hóa hệ thống**:

Trong quá trình phát triển, các bạn đã thực hiện những biện pháp tối ưu hóa hiệu suất nào cho website? Làm thế nào để các bạn đảm bảo rằng hệ thống có thể xử lý lưu lượng truy cập lớn mà không gặp sự cố?

1. **Về tính năng gợi ý sản phẩm**:

Các bạn đã tích hợp tính năng gợi ý sản phẩm dựa trên giới tính và các thuộc tính khác. Các bạn có thể giải thích chi tiết về thuật toán hoặc phương pháp mà các bạn đã sử dụng để triển khai tính năng này? Làm thế nào để đảm bảo tính chính xác và hiệu quả của gợi ý sản phẩm?

1. **Về trải nghiệm người dùng (UX/UI)**:

Các bạn có thể mô tả chi tiết quá trình thiết kế giao diện người dùng và trải nghiệm người dùng? Các bạn đã thực hiện các khảo sát người dùng hoặc thử nghiệm nào để đảm bảo giao diện dễ sử dụng và thân thiện với người dùng?

1. **Về khả năng mở rộng và quản lý biến thể sản phẩm**:

Các bạn đã đề cập đến việc mở rộng khả năng quản lý biến thể sản phẩm. Các bạn có thể giải thích chi tiết về cách hệ thống quản lý các biến thể này? Làm thế nào để đảm bảo tính nhất quán và dễ quản lý khi số lượng sản phẩm và biến thể tăng lên?

1. **Về tích hợp và triển khai hệ thống**:

Các bạn có thể mô tả quy trình triển khai hệ thống lên các môi trường khác nhau như máy chủ trực tuyến và máy chủ cục bộ? Các bạn đã sử dụng các công cụ và phương pháp gì (ví dụ: Docker, CI/CD) để đảm bảo hệ thống hoạt động hiệu quả và dễ bảo trì?

**Trả lời:**

1. **Về lựa chọn công nghệ**:

MERN Stack được chọn vì tính đồng nhất với JavaScript, cộng đồng lớn, và thư viện phong phú, làm tăng hiệu quả phát triển. MEAN có kiến trúc phức tạp do Angular, thích hợp cho doanh nghiệp lớn. LAMP, dù ổn định, sử dụng công nghệ cũ và hạn chế trong xử lý dữ liệu không cấu trúc như NoSQL.

1. **Về bảo mật hệ thống**:

Chúng em đã triển khai các biện pháp bảo mật như sau:

* Xác thực và ủy quyền người dùng bằng cách sử dụng JWT
* Bảo mật mật khẩu:
* Phân quyền truy cập và bảo mật cơ sở dữ liệu
* Kiểm soát phiên làm việc

**Xác thực và Ủy quyền người dùng bằng JWT**

* **Tạo và Sử dụng JWT:** Khi người dùng đăng nhập thành công, máy chủ sẽ tạo ra một JWT chứa các thông tin của người dùng. JWT này được mã hóa bằng một khóa bí mật và có thời gian sử dụng nhất định.
* **Lưu trữ JWT:** JWT sau đó được gửi từ máy chủ đến máy khách và lưu trữ dưới dạng cookie với các thuộc tính bảo mật như HttpOnly, Secure, và SameSite để ngăn chặn các cuộc tấn công XSS và CSRF.
* **Xác thực Yêu cầu:** Trong các yêu cầu HTTP tiếp theo, máy chủ sẽ kiểm tra tính hợp lệ của JWT từ cookie để xác nhận rằng yêu cầu đến từ người dùng đã xác thực.
* **Phân quyền:** Máy chủ kiểm tra vai trò của người dùng từ JWT để đảm bảo họ có quyền truy cập vào các tài nguyên yêu cầu quyền hạn tương ứng.

**Bảo mật mật khẩu**

* **Mã hóa Mật khẩu:** Mật khẩu của người dùng được mã hóa bằng bcrypt trước khi được lưu vào cơ sở dữ liệu. Bcrypt là một thuật toán mã hóa mạnh, giúp bảo vệ mật khẩu khỏi các cuộc tấn công brute-force.

**Phân quyền truy cập và Bảo mật cơ sở dữ liệu**

* **Thiết lập Quyền Truy cập:** Các quyền truy cập cụ thể được thiết lập cho từng tài khoản trong cơ sở dữ liệu. Chỉ những người dùng có quyền hạn tương ứng mới được phép truy cập vào các tài nguyên cụ thể.
* **Bảo vệ Dữ liệu:** Việc phân quyền đảm bảo rằng dữ liệu chỉ có thể được truy cập và thao tác bởi những người dùng có quyền hạn, giảm thiểu nguy cơ rò rỉ thông tin và các cuộc tấn công từ bên trong.

**Kiểm soát phiên làm việc**

* **Thời gian Phiên:** Thiết lập thời gian hết hạn cho các phiên làm việc để ngăn chặn các cuộc tấn công sử dụng phiên đã hết hạn.
* **Bảo vệ Cookie:** Cookie chứa JWT được bảo vệ bằng các thuộc tính HttpOnly, Secure, và SameSite để ngăn chặn các cuộc tấn công XSS và CSRF, đảm bảo rằng thông tin phiên làm việc luôn an toàn.

1. **Về hiệu suất và tối ưu hóa hệ thống**:

* Single Page Application (SPA): React cho phép xây dựng SPA, nơi các tài nguyên chỉ tải một lần, và DOM ảo cập nhật phần hiển thị thay vì tải lại trang.
* Lazy Loading: Lazy loading trong React sử dụng dynamic imports để tải các component và module chỉ khi được yêu cầu, giảm thời gian tải trang đầu tiên.
* Lưu Trữ Cache: Dữ liệu thường xuyên được truy cập được lưu trong local storage hoặc session storage, giảm độ trễ truy vấn và giảm tải trên cơ sở dữ liệu.
* Cơ chế Bất Đồng Bộ của React: React sử dụng JavaScript Promises và async/await để quản lý các hoạt động bất đồng bộ, đảm bảo UI không bị block trong khi xử lý dữ liệu.

1. **Về tính năng gợi ý sản phẩm**:

* Với tính năng gợi ý sản phẩm chúng em sử dụng thuật toán deepSearch để tìm kiếm trong cấu trúc dữ liệu của sản phẩm, kiểm tra tất cả các thuộc tính tìm kiếm từ khóa liên quan. Ngoài ra chúng em cũng sử dụng các từ khóa liên quan tới giới tính của sản phẩm để xác định các sản phẩm dựa trên giới tính và còn ánh xạ loại sản phẩm hiện tại với các loại sản phẩm bổ sung có thể được gợi ý cùng nhau. Ví dụ nếu sản phẩm có từ khóa là “đầm váy”, chúng em có thể gợi ý các sản phẩm phối cùng như “quần short”, “chân váy” và “áo thun”

1. **Về trải nghiệm người dùng (UX/UI)**:

* UI: Giao diện người dùng được thiết kế bằng cách tham khảo các template từ ThemeForest và TemplateMonster, những trang web cung cấp giải pháp thiết kế đã có bề dày hơn 20 năm, đảm bảo tính chuyên nghiệp và hiện đại
* UX: Chúng em đã phát triển trải nghiệm người dùng bằng cách học hỏi từ Shopee, trang thương mại điện tử với thị phần lớn nhất, và Rabity, website thời trang trẻ em hàng đầu có chứng nhận từ Bộ Công Thương. Điều này đảm bảo rằng UX của chúng em cảm đạt những chuẩn mực cao nhất của ngành.

Khải

1. **Về khả năng mở rộng và quản lý biến thể sản phẩm**:

Chúng em sử dụng hai mảng một chiều để quản lý các biến thể sản phẩm. Có ba trường hợp biến thể:

* Không có biến thể: Ví dụ như các sản phẩm phụ kiện.
* Một biến thể: Ví dụ như sản phẩm có biến thể màu sắc.
* Hai biến thể: Ví dụ như sản phẩm có biến thể màu sắc và kích thước trong mỗi màu sắc.

Chúng em sử dụng cơ chế kết hợp hai mảng thành một cấu trúc dữ liệu thống nhất. Cụ thể, chúng em dùng hàm merge2Arrays để kết hợp các biến thể từ hai mảng thành một danh sách sản phẩm có biến thể tổ hợp.

Để đảm bảo tính nhất quán và dễ quản lý:

* React và Reactstrap: Tạo danh sách động và quản lý trạng thái sản phẩm.
* Hooks của React (useState, useEffect): Quản lý trạng thái và phản ứng với các thay đổi trong UI, đảm bảo các cập nhật từ người dùng được xử lý nhất quán và hiệu quả.
* Giao Diện Kéo và Thả: Giúp người dùng dễ dàng sắp xếp các biến thể sản phẩm, cập nhật trạng thái ngay lập tức.

Nhờ vào cơ chế này, hệ thống của chúng em có thể quản lý hiệu quả số lượng lớn biến thể sản phẩm mà không làm giảm hiệu suất hoặc trải nghiệm người dùng.

1. **Về tích hợp và triển khai hệ thống**:

* Hiện tại, chúng em triển khai hệ thống thủ công trên máy chủ cục bộ bằng cách cài đặt Node.js, MongoDB, và React. Sử dụng MongoDB Compass để quản lý cơ sở dữ liệu và khởi chạy ứng dụng React và Node.js trên localhost.
* Chúng em đã nghiên cứu về Docker để triển khai lên môi trường trực tuyến nhằm cải thiện tính nhất quán và dễ quản lý. Tuy nhiên, do giới hạn về tài nguyên và ưu tiên thời gian hoàn thiện các tính năng chính, chúng em chưa áp dụng triển khai này.