

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



ĐỖ NHẬT THANH
HÌNH VĨNH PHÁT
NGUYỄN MINH PHÚ
NGUYỄN TUẤN KIỆT
HOÀNG MINH HIẾU

ĐỀ TÀI: TRANG WEB HỌC TẬP TRỰC TUYẾN

TP. Hồ Chí Minh – 2025

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
PHÂN HIỆU TẠI TP. HỒ CHÍ MINH
BỘ MÔN CÔNG NGHỆ THÔNG TIN



ĐỖ NHẬT THANH
HÌNH VĨNH PHÁT
NGUYỄN MINH PHÚ
NGUYỄN TUẤN KIỆT
HOÀNG MINH HIẾU

ĐỀ TÀI: TRANG WEB HỌC TẬP TRỰC TUYẾN

Lớp: CNTT

Khóa: 65

NGƯỜI HƯỚNG DẪN KHÓA HỌC

1. HDC: Ths. Nguyễn Thiện Dương
2. HDP: Ths. Nguyễn Thiện Dương

TP. Hồ Chí Minh – 2025

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TpHCM, ngày tháng năm.....

Giảng viên hướng dẫn

ThS. Nguyễn Thiện Dương

LỜI CẢM ƠN

Lời nói đầu tiên, em xin gửi tới Quý Thầy Cô Bộ môn Công nghệ Thông tin Trường Đại học Giao thông vận tải phân hiệu tại thành phố Hồ Chí Minh lời chúc sức khỏe và lòng biết ơn sâu sắc.

Trong suốt quá trình học tập và thực hiện bài tập lớn môn Thiết kế web, chúng em đã nhận được rất nhiều sự hỗ trợ và động viên quý báu từ thầy Nguyễn Thiện Dương – người đã trực tiếp giảng dạy và hướng dẫn chúng em trong suốt học kỳ vừa qua. Tuy đã cố gắng trong quá trình nghiên cứu tìm hiểu tuy nhiên do kiến thức còn hạn chế nên vẫn còn tồn tại nhiều thiếu sót. Vì vậy em rất mong nhận được sự đóng góp ý kiến của Quý thầy cô bộ môn để đề tài của em có thể hoàn thiện hơn.

Trong quá trình thực hiện bài tập lớn, mặc dù còn gặp nhiều khó khăn và thiếu sót, nhưng nhờ có sự chỉ bảo nhiệt tình, sự kiên nhẫn giải đáp những thắc mắc của thầy, chúng em đã từng bước khắc phục những hạn chế của bản thân, hoàn thiện sản phẩm một cách tốt nhất có thể.

Chúng em xin chân thành cảm ơn thầy vì không chỉ dừng lại ở việc truyền đạt kiến thức, mà còn luôn động viên, tạo điều kiện kết nối với các anh chị cựu sinh viên để khích lệ tinh thần học hỏi, sáng tạo của chúng em. Sự tận tâm và nhiệt huyết của thầy là nguồn động lực lớn giúp chúng em không ngừng cố gắng và hoàn thiện mình.

Một lần nữa, chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến thầy Nguyễn Thiện Dương. Kính chúc thầy luôn mạnh khỏe, hạnh phúc và đạt nhiều thành công trong sự nghiệp trồng người cao quý.

Sinh viên

Đỗ Nhật Thanh

Hình Vĩnh Phát

Nguyễn Minh Phú

Nguyễn Tuấn Kiệt

Hoàng Minh Hiếu

MỤC LỤC

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN.....	i
LỜI CẢM ƠN	ii
MỤC LỤC	iii
DANH MỤC HÌNH ẢNH	v
CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI VÀ PHÂN TÍCH YÊU CẦU	1
1.1 Giới thiệu chung.....	1
1.2 Mục tiêu và nhiệm vụ của đề tài.....	2
1.3 Phạm vi và giới hạn	4
1.4 Phân tích yêu cầu	6
1.5 Cấu trúc báo cáo	7
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ THỰC NGHIỆM.....	8
2.1 Cơ sở lý thuyết.....	8
2.2 Khảo sát các sản phẩm tương tự	27
2.2.1 Quizizz	27
2.2.2 Kahoot!	28
2.2.3 Google Forms Quiz.....	29
2.3 Tổng hợp và định hướng thiết kế	31
CHƯƠNG 3: THIẾT KẾ VÀ TRIỂN KHAI WEBSITE	34
3.1 Thiết kế giao diện (UI Design)	34
3.2 Chức năng và tương tác (UX Design)	47
3.3 Mã nguồn và cấu trúc dự án	50
CHƯƠNG 4: KIỂM THỬ, TỐI ƯU HÓA VÀ TRIỂN KHAI.....	52
4.1 Kiểm thử giao diện và chức năng	52
4.2 Tối ưu hóa hiệu suất	52
4.3 Triển khai website lên host và server.....	53
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	55
5.1 Kết quả đạt được.....	55
5.2 Hạn chế	55
5.3 Hướng phát triển	56

TÀI LIỆU THAM KHẢO.....	58
PHỤ LỤC	60

DANH MỤC HÌNH ẢNH

STT	Tên hình	Trang
2.1	Ví dụ các thẻ định dạng văn bản	9
2.2	Ví dụ các thẻ định dạng văn bản 2	10
2.3	Danh sách 1	11
2.4	Danh sách 2	11
2.5	Danh sách 3	12
2.6	Bảng	13
3.1	Sitemap	34
3.2	Login – SignUp	35
3.3	Home	36
3.4	Kho đề	37
3.5	Môn học	38
3.6	Tạo câu hỏi	39
3.7	Bảng xếp hạng	40
3.8	Tài khoản	41
3.9	Desktop	42
3.10	Tablet	43
3.11	Samsung	44
3.12	Chrome	45
3.13	Edge	45
3.14	Firefox	46
3.15	Safari	46

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI VÀ PHÂN TÍCH YÊU CẦU

1.1 Giới thiệu chung

➤ Bối cảnh và lý do chọn đề tài

Trong xu hướng học tập và giải trí trực tuyến ngày càng phát triển, đặc biệt là các ứng dụng quiz (trắc nghiệm) giúp người dùng vừa học vừa chơi. Cũng như công nghệ web front-end (HTML, CSS, JavaScript) hiện nay rất phổ biến, dễ tiếp cận, và là nền tảng cơ bản để xây dựng các ứng dụng web tương tác, thì nhu cầu tạo ra các trang web đơn giản, trực quan, dễ sử dụng để kiểm tra kiến thức hoặc giải trí đang tăng cao, phù hợp cho cả học sinh, sinh viên và người dùng phổ thông.

Đề tài “Thiết kế web học tập trực tuyến” được lựa chọn với các lý do sau:

- Giúp củng cố và vận dụng kiến thức nền tảng về lập trình web.
- Tạo ra một sản phẩm trực quan, dễ sử dụng, có thể áp dụng trong học tập hoặc giải trí.
- Không yêu cầu hệ thống phức tạp, chỉ cần trình duyệt để chạy, phù hợp với phạm vi nghiên cứu và thực hành của sinh viên.
- Có khả năng mở rộng, nâng cấp thêm các tính năng như lưu điểm, phân loại câu hỏi, hoặc kết nối cơ sở dữ liệu trong tương lai.

➤ Ý tưởng và lĩnh vực của website

Đề tài hướng đến việc xây dựng một website với giao diện đơn giản, thân thiện. Ý tưởng chính là tạo ra một công cụ kiểm tra kiến thức theo hình thức trắc nghiệm, giúp người dùng vừa học vừa chơi, đồng thời có thể đánh giá mức độ hiểu biết của bản thân ngay lập tức.

Lĩnh vực áp dụng của website:

- **Giáo dục:** hỗ trợ học sinh, sinh viên ôn tập kiến thức qua các bài quiz trực tuyến.
- **Giải trí:** cung cấp các quiz vui về chủ đề đời sống, văn hóa, lịch sử, hoặc sở thích cá nhân.

- **Phát triển kỹ năng web:** là sản phẩm minh chứng cho khả năng lập trình front-end, phù hợp để đưa vào portfolio cá nhân.

➤ Ý nghĩa thực tế và ứng dụng của đề tài trong học tập hoặc công việc

Việc xây dựng một website mang nhiều ý nghĩa thực tế và có khả năng ứng dụng rộng rãi trong học tập cũng như công việc:

- **Trong học tập:**
 - Giúp sinh viên, học sinh ôn tập kiến thức một cách trực quan, sinh động thông qua các câu hỏi trắc nghiệm.
 - Tạo môi trường học tập tương tác, giúp người học tự đánh giá năng lực và mức độ hiểu biết của bản thân.
 - Là một sản phẩm thực hành giúp người học rèn luyện kỹ năng lập trình web cơ bản, từ đó làm nền tảng cho các dự án phức tạp hơn.
- **Trong công việc:**
 - Có thể áp dụng trong đào tạo nội bộ của doanh nghiệp, giúp kiểm tra nhanh kiến thức nhân viên về quy trình, sản phẩm hoặc kỹ năng mềm.
 - Dùng làm công cụ khảo sát ý kiến hoặc tạo trò chơi tương tác trong các sự kiện, hội thảo.
 - Là sản phẩm minh chứng năng lực lập trình front-end, có thể đưa vào portfolio cá nhân để phục vụ xin việc hoặc thực tập.
- **Ý nghĩa thực tiễn:**
 - Góp phần thúc đẩy việc ứng dụng công nghệ thông tin trong giáo dục và đào tạo.
 - Khuyến khích tinh thần tự học, tự nghiên cứu thông qua việc vừa học vừa chơi.
 - Tạo ra một sản phẩm nhỏ gọn, dễ triển khai, phù hợp với nhu cầu thực tế của nhiều đối tượng khác nhau.

1.2 Mục tiêu và nhiệm vụ của đề tài

➤ Mục tiêu tổng quát của website

- **Về thiết kế**
 - Xây dựng giao diện trực quan, hiện đại và tối giản, tập trung vào nội dung câu hỏi.

- Đảm bảo tính responsive, hiển thị tốt trên nhiều thiết bị (máy tính, tablet, điện thoại).
- Sử dụng màu sắc hài hòa, font chữ dễ đọc, bố cục rõ ràng để tạo cảm giác thoải mái cho người dùng.
- Tạo sự nhất quán và nhận diện thông qua màu chủ đạo và phong cách thiết kế.
- **Về trải nghiệm người dùng**
 - Người dùng có thể bắt đầu làm quiz ngay lập tức mà không cần thao tác phức tạp.
 - Hệ thống cung cấp phản hồi tức thì: hiển thị đúng/sai ngay sau khi chọn đáp án hoặc tổng kết cuối bài.
 - Có thanh tiến trình (progress bar) để người học theo dõi số câu đã làm và còn lại.
 - Tạo hiệu ứng chuyển câu hỏi mượt mà, giúp trải nghiệm học tập trở nên sinh động và hấp dẫn.
- **Về chức năng**
 - Hiển thị câu hỏi trắc nghiệm với nhiều lựa chọn, có thể random thứ tự câu hỏi/đáp án.
 - Tính điểm tự động và hiển thị kết quả cuối cùng.
 - Điều hướng câu hỏi bằng nút Next/Previous hoặc tự động chuyển sang câu tiếp theo.
 - Sử dụng JavaScript để tạo hiệu ứng động: highlight đáp án, hiển thị thông báo kết quả.
 - Hỗ trợ lưu trữ tạm thời bằng localStorage: ghi nhớ điểm số hoặc tiến trình làm quiz mà không cần backend.
 - Cho phép dễ dàng tùy biến nội dung quiz bằng cách chỉnh sửa file JS/JSON.

➤ **Các nhiệm vụ chính cần thực hiện**

- **Thiết kế giao diện và bố cục hợp lý**
 - Xác định bố cục tổng thể của website: khu vực hiển thị câu hỏi, đáp án, thanh tiến trình, và kết quả.

- Lựa chọn màu sắc, font chữ, icon phù hợp nhằm tạo cảm giác trực quan, dễ sử dụng.
- Thiết kế giao diện theo hướng tối giản, hiện đại, tập trung vào nội dung học tập.
- Đảm bảo tính thống nhất trong toàn bộ trang web (header, footer, nút chức năng).
- **Phát triển các chức năng cơ bản**
 - Hiện thị câu hỏi trắc nghiệm với nhiều lựa chọn.
 - Cho phép người dùng chọn đáp án và nhận phản hồi tức thì (đúng/sai).
 - Cung cấp nút điều hướng (Next/Previous) để di chuyển giữa các câu hỏi.
 - Tích hợp hiệu ứng động bằng JavaScript: highlight đáp án, thông báo kết quả.
- **Đảm bảo responsive và tối ưu hiệu suất**
 - Thiết kế website responsive, hiển thị tốt trên nhiều thiết bị (PC, tablet, smartphone).
 - Sử dụng CSS linh hoạt để thích ứng với kích thước màn hình.
 - Tối ưu mã nguồn HTML, CSS, JS nhằm giảm dung lượng tải trang và tăng tốc độ phản hồi.
 - Kiểm tra hiệu suất để cải thiện trải nghiệm người dùng.
- **Triển khai website lên host để truy cập trực tuyến**
 - Lựa chọn dịch vụ hosting phù hợp (ví dụ: GitHub Pages, Netlify, Vercel).
 - Thực hiện deploy website để người dùng có thể truy cập trực tuyến mọi lúc, mọi nơi.
 - Kiểm tra tính ổn định sau khi triển khai, đảm bảo website hoạt động đúng như thiết kế.
 - Cập nhật và bảo trì định kỳ để nâng cao chất lượng và trải nghiệm người dùng.

1.3 Phạm vi và giới hạn

➤ Giới hạn về nội dung, thời gian, kỹ thuật và công cụ được sử dụng

- **Về nội dung**
 - Website chỉ tập trung vào quiz trắc nghiệm cơ bản.

- Nội dung câu hỏi mang tính minh họa, thử nghiệm, chưa đa dạng theo nhiều lĩnh vực chuyên sâu.
- Không tích hợp chức năng quản lý người dùng, lưu trữ dữ liệu lâu dài hay hệ thống phân quyền.

- **Về thời gian**

- Thời gian thực hiện đề tài có hạn, nên chỉ tập trung vào các chức năng cốt lõi.
- Chưa có điều kiện để mở rộng thêm nhiều bộ câu hỏi và phát triển tính năng nâng cao.
- Quá trình kiểm thử chỉ thực hiện trên một số thiết bị phổ biến, chưa bao quát toàn bộ môi trường thực tế.

- **Về kỹ thuật**

- Website chỉ sử dụng HTML, CSS, JavaScript ở mức front-end, không có backend xử lý dữ liệu.
- Dữ liệu câu hỏi và kết quả chỉ được lưu trữ tạm thời bằng localStorage, không có cơ sở dữ liệu chuyên nghiệp.
- Các hiệu ứng và chức năng được xây dựng ở mức cơ bản, chưa tối ưu cho hệ thống lớn hoặc nhiều người dùng đồng thời.

- **Về công cụ được sử dụng**

- Công cụ chính: Visual Studio Code để viết mã.
- Ngôn ngữ lập trình: HTML, CSS, JavaScript.
- Công cụ kiểm thử: trình duyệt web.
- Công cụ triển khai: các dịch vụ hosting miễn phí như GitHub Pages, Netlify, Vercel.

➤ **Các phần sẽ phát triển thêm trong tương lai**

- **Về nội dung**

- Mở rộng ngân hàng câu hỏi đa dạng hơn, bao gồm nhiều lĩnh vực (toán, lịch sử, khoa học, ngôn ngữ...).

- Cho phép phân loại quiz theo chủ đề, cấp độ khó để phù hợp với từng đối tượng người học.
- Tích hợp thêm giải thích chi tiết cho từng câu hỏi nhằm giúp người học hiểu rõ kiến thức.
- **Về chức năng**
 - Tích hợp cơ sở dữ liệu backend (MySQL, Firebase, MongoDB) để lưu trữ lâu dài kết quả và nội dung quiz.
 - Thêm tính năng thi đua trực tuyến (real-time multiplayer quiz) để tăng tính tương tác.
 - Hỗ trợ xuất báo cáo kết quả dưới dạng file (PDF, Excel).
- **Về trải nghiệm người dùng (UX, UI)**
 - Cải thiện giao diện với animation, gamification (huy hiệu, cấp độ, bảng xếp hạng).
 - Hỗ trợ đa ngôn ngữ để mở rộng đối tượng sử dụng.
- **Về kỹ thuật và công cụ**
 - Ứng dụng các framework hiện đại như React, Vue hoặc Angular để quản lý giao diện hiệu quả hơn.
 - Sử dụng API để lấy dữ liệu câu hỏi từ nguồn bên ngoài.
 - Tích hợp AI/ML để gợi ý quiz phù hợp với năng lực người học.
- **Về triển khai và mở rộng**
 - Triển khai trên máy chủ chuyên nghiệp để đảm bảo tốc độ và độ ổn định.
 - Phát triển ứng dụng di động (Android/iOS) dựa trên nền tảng web.
 - Tích hợp với các hệ thống LMS (Learning Management System) để phục vụ môi trường giáo dục chính quy.

1.4 Phân tích yêu cầu

➤ Yêu cầu chức năng (Functional Requirements)

- **Menu điều hướng:** thanh menu chính giúp người dùng truy cập nhanh đến các trang: Trang chủ, Câu hỏi, Bảng xếp hạng, Hồ sơ, v.v. Hiển thị rõ ràng, dễ sử dụng, có thể mở rộng (dropdown) nếu có nhiều mục.

➤ Yêu cầu phi chức năng (Non-functional Requirements)

- **Tốc độ tải trang**
 - Website phải có thời gian tải nhanh, hạn chế tình trạng chậm trễ khi truy cập.
 - Hình ảnh, CSS và JavaScript cần được tối ưu dung lượng để giảm thời gian tải.
- **Thẩm mỹ**
 - Giao diện cần có màu sắc hài hòa, bố cục hợp lý, tạo cảm giác dễ chịu cho người dùng.
 - Font chữ rõ ràng, dễ đọc, phù hợp với nội dung học tập.
 - Thiết kế theo phong cách tối giản, hiện đại, tránh rườm rà nhưng vẫn đảm bảo tính trực quan.
- **Độ tương thích**
 - Website phải hoạt động tốt trên nhiều trình duyệt phổ biến (Chrome, Edge, Firefox, Safari).
 - Tương thích với nhiều hệ điều hành (Windows, macOS, Android, iOS).
 - Đảm bảo không bị lỗi hiển thị khi chạy trên các phiên bản trình duyệt khác nhau.
- **Responsive**
 - Website cần tự động điều chỉnh bố cục theo kích thước màn hình (desktop, tablet, smartphone).
 - Đảm bảo trải nghiệm người dùng nhất quán trên mọi thiết bị.

1.5 Cấu trúc báo cáo

- Giới thiệu ngắn gọn các chương tiếp theo:
 - **Chương 2:** Khảo sát cơ sở lý thuyết và các thực nghiệm đã có.
 - **Chương 3:** Thiết kế và triển khai website.
 - **Chương 4:** Kiểm thử, tối ưu và triển khai website lên host.
 - **Chương 5:** Kết luận và hướng phát triển.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ THỰC NGHIỆM

2.1 Cơ sở lý thuyết

➤ Kiến thức nền tảng

- **HTML5**

- HTML được viết tắt từ Hyper Text Markup Language (Ngôn ngữ đánh dấu siêu văn bản). Đây là ngôn ngữ chính của WordWide Web, hầu hết các trang web được host trên mạng được viết bởi các phiên bản của HTML. HTML được xem như các khối xây dựng cơ bản cho website, được tạo bởi Tim Berners-Lee (người sáng lập World Wide Web) vào năm 1991.

- Ngôn ngữ này gồm các thẻ HTML để nói cho trình duyệt biết các thẻ này chứa loại nội dung gì, thuộc tính của chúng và trình tự sắp xếp. Tập tin đa phương tiện, văn bản, hyperlink... hiển thị trên trình duyệt web là thành phần của HTML. Người dùng sẽ không thấy hiển thị thẻ HTML trên web mà chúng được dùng để quyết định nội dung hiển thị theo cách nào.

- HTML5 là phiên bản thứ 5 của HTML, được phát hành năm 2004. Sau nhiều lần cập nhật HTML, hiện tại HTML5 là chuẩn mới nhất của HTML. Về bản chất, HTML5 vẫn là ngôn ngữ markup chính, tuy nhiên nó có nhiều tính năng hơn HTML, xóa một số tính nghiêm ngặt trong XHTML.

- Cấu trúc cơ bản của trang HTML5:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Tiêu đề trang</title>
</head>
<body>
  Nội dung trang web
</body>
</html>
```

Chú thích:

<!DOCTYPE html>: Khai báo chuẩn HTML5.

<html> </html>: Xác định phạm vi của tài liệu HTML.

<head> </head>: Chứa thông tin mô tả trang (metadata).

<title> </title>: Tiêu đề hiển thị trên tab trình duyệt.

<body> </body>: Chứa toàn bộ nội dung hiển thị trên trang web.

- Các thẻ định dạng văn bản:

Tiêu đề (Heading): <h1>, <h2>, ..., <h6>.

Đoạn văn: <p> Nội dung </p>

Xuống dòng:


```
<body>
  <h1>VÍ DỤ</h1>
  <h2>VÍ DỤ</h2>
  <h3>VÍ DỤ</h3>
  <h4>VÍ DỤ</h4>
  <h5>VÍ DỤ</h5>
  <h6>VÍ DỤ</h6>
  <p>ĐỒ NHẬT THANH</p>
  <p>ĐỒ NHẬT <br> THANH</p>
</body>
```

VÍ DỤ

VÍ DỤ

VÍ DỤ

VÍ DỤ

VÍ DỤ

VÍ DỤ

ĐỒ NHẬT THANH

**ĐỒ NHẬT
THANH**

Hình 2.1 Ví dụ các thẻ định dạng văn bản

Chữ đậm:

Nhấn mạnh quan trọng:

Chữ nghiêng: <i>

Tô sáng: <mark>

Chữ nhỏ: <small>

```
<body>  
  <b>VÍ DỤ</b>  
  <br>  
  <strong>VÍ DỤ</strong>  
  <br>  
  <i>VÍ DỤ</i>  
  <br>  
  <mark>VÍ DỤ</mark>  
  <br>  
  <small>VÍ DỤ</small>  
</body>
```

VÍ DỤ
VÍ DỤ
VÍ DỤ
VÍ DỤ
VÍ DỤ

Hình 2.2 Ví dụ các thẻ định dạng văn bản 2

- Các thẻ danh sách:

Danh sách không thứ tự:

```
<body>
  <ul>
    <li>Mục 1</li>
    <li>Mục 2</li>
  </ul>
</body>
```

- Mục 1
- Mục 2

Hình 2.3 Danh sách 1

Danh sách có thứ tự:

```
<body>
  <ol>
    <li>Mục 1</li>
    <li>Mục 2</li>
  </ol>
</body>
```

1. Mục 1
2. Mục 2

Hình 2.4 Danh sách 2

Danh sách mô tả:

```
<body>
  <dl>
    <dt>Thuật ngữ</dt>
    <dd>Mô tả của thuật ngữ</dd>
  </dl>
</body>
```

Thuật ngữ

Mô tả của thuật ngữ

Hình 2.5 Danh sách 3

- Hình ảnh trong HTML

```

```

src: Đường dẫn file ảnh.

alt: Mô tả ảnh (SEO & hỗ trợ khi ảnh lỗi).

- Liên kết – Thẻ <a>

```
<a href="https://example.com" target="_blank">Nhấn vào đây</a>
```

- Bảng trong HTML

```
<body>
  <table>
    <tr>
      <th>Cột 1</th>
      <th>Cột 2</th>
    </tr>
    <tr>
      <td>Dữ liệu 1</td>
      <td>Dữ liệu 2</td>
    </tr>
  </table>
</body>
```

Cột 1	Cột 2
Dữ liệu 1	Dữ liệu 2

Hình 2.6 Bảng

- Thẻ form và các phương thức nhập liệu

```
<body>
  <form action="xu-ly.php" method="post">
    ...
  </form>
</body>
```

Ví dụ: `<input type="text">`, text, number, email, ...

- Các thẻ semantic (ngữ nghĩa) trong HTML5

Đây là phần quan trọng nhất của HTML5:

`<header>`: phần đầu trang

<footer>: phần cuối trang
<nav>: thanh điều hướng
<section>: phân vùng nội dung
<article>: một bài viết độc lập
<aside>: lề phụ, quảng cáo
<main>: nội dung chính
<figure>: khối hình ảnh
<figcaption>: chú thích hình

- Các thẻ đa phương tiện trong HTML5

Audio:

```
<audio controls>  
    <source src="nhac.mp3" type="audio/mpeg">  
</audio>
```

Video:

```
<video controls width="500">  
    <source src="video.mp4" type="video/mp4">  
</video>
```

Canvas:

```
<canvas id="myCanvas"></canvas>
```

- Địa chỉ URL trong HTML5

Địa chỉ tuyệt đối: gồm đầy đủ giao thức, tên miền. Ví dụ: https://...

Địa chỉ tương đối: dựa trên vị trí file hiện tại. Ví dụ: images/logo.png

- **CSS3**

- CSS meaning là từ viết tắt của Cascading Style Sheets, tức là ngôn ngữ được dùng để định dạng các phần tử do ngôn ngữ đánh dấu (như HTML) tạo ra. Hay nói cách khác, HTML định dạng các phần tử có trên website như tiêu đề, bảng biểu, tạo đoạn văn bản..., còn CSS giúp cho các phần tử của HTML trở nên phong phú, nổi bật hơn, bằng cách trang trí, đổi màu chữ, thêm màu sắc cho trang hay thay đổi cấu trúc trang...

- CSS3 là phiên bản 3 của CSS và cũng là phiên bản mới nhất của ngôn ngữ này. CSS3 không chỉ thừa hưởng tất cả các ưu điểm của những phiên bản trước, mà nó còn có thêm các tính năng cải tiến hơn. Từ đó, CSS3 mang lại nhiều sự tiện lợi cho người dùng. Vì thế, đây cũng là phiên bản được giới lập trình ưa chuộng và sử dụng nhiều trong quá trình thiết kế website.

- Cách nhúng CSS vào trang HTML5:

Inline CSS: `<p style="color: red;">Nội dung</p>`

Internal CSS: `<style> p { color: blue; } </style>`

External CSS: `<link rel="stylesheet" href="styles.css">`

- Cấu trúc một quy tắc CSS:

```
selector {  
    property: value;  
}
```

Selector: bộ chọn

Property: thuộc tính

Value: giá trị

- Bộ chọn trong CSS3:

- Bộ chọn cơ bản
 - Bộ chọn theo thẻ: `p, h1, div...`
 - Bộ chọn theo class: `.ten-class`
 - Bộ chọn theo id: `#ten-id`
- Bộ chọn kết hợp
 - Div p: chọn tất cả p bên trong div
 - Div > p: p là con trực tiếp
 - Div + p: p đứng sau div
 - Div ~ p: tất cả p cùng cấp đứng sau div
- Bộ chọn thuộc tính
 - `input[type="text"] { color: blue; }`
- Bộ chọn pseudo-class

:hover – rê chuột

:focus – đang nhập

:active – đang bấm

:first – child

:last – child

- Bộ chọn pseudo-element

::before

::after

::first – letter

::selection

- Các thuộc tính CSS3 quan trọng

- Thuộc tính văn bản (Text):

color

font-size

font-family

font-weight

text-align

text-decoration

line-height

letter-spacing

- Thuộc tính nền (Background):

background-color

background-image

background-repeat

background-size

background-position

background-attachment

- Kích thước – khoảng cách (Box Model):

width

height
padding
margin
border
border-radius
box-shadow

- Thuộc tính hiển thị và bố cục

- Display
 - display: block;
 - display: inline;
 - display: inline-block;
 - display: none; /* ẩn */
- Position
 - position: static;
 - position: relative;
 - position: absolute;
 - position: fixed;
 - position: sticky;
- Float
 - float: left;
 - float: right;

- Flexbox trong CSS3

- Khởi tạo flex container
 - .container {
 - display: flex;
 - }
- Các thuộc tính chính
 - flex-direction
 - justify-content

align-items

flex-wrap

gap

- Ví dụ

```
.container {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    gap: 20px;  
}
```

- CSS Grid Layout

- Khởi tạo Grid

```
.container {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr;  
    gap: 20px;  
}
```

- Quy định hàng và cột

grid-template-rows

grid-template-areas

grid-column

grid-row

- Hiệu ứng trong CSS3

- Transition

```
button {  
    transition: 0.3s;  
}  
  
button:hover {  
    transform: scale(1.1);  
}
```

```
}
```

- Transform

```
transform: rotate(45deg);
```

```
transform: scale(1.2);
```

```
transform: translateX(50px);
```

- Animation

```
@keyframes fadeIn {
```

```
  from { opacity: 0; }
```

```
  to { opacity: 1; }
```

```
}
```

```
.box {
```

```
  animation: fadeIn 1s ease-in-out;
```

```
}
```

- Responsive Design với CSS3

- Media Queries

```
@media (max-width: 768px) {
```

```
  div {
```

```
    width: 100%;
```

```
    font-size: 14px;
```

```
  }
```

```
}
```

- CSS3 và màu sắc

- Các dạng màu hỗ trợ:

```
Rbg(255, 0, 0)
```

```
Rbga(255, 0, 0, 0.5)
```

```
#ff0000
```

```
Hsl(0, 100%, 50%)
```

- Đơn vị đo trong CSS3

Tuyệt đối: px, pt

Tương đối: %, em, rem, vw, vh (viewport)

- **JavaScript**

- JavaScript, còn được viết tắt là JS, là một ngôn ngữ lập trình, chạy dưới dạng ngôn ngữ kịch bản phía máy khách trong môi trường hosting. Môi trường hosting phổ biến nhất là trình duyệt web, dùng ngôn ngữ này để tạo tính tương tác cho trang web.

- JavaScript được phát triển vào những năm 1990 bởi Brendan Eich – người cũng là co-founder của Mozilla & Firefox. Kể từ đó, ngôn ngữ này đã trở thành một trong những công nghệ cốt lõi duy trì World Wide Web.

- Cách nhúng JS vào trang HTML:

Inline: `<button onclick="alert('Hello')">Click</button>`

Internal: `<script> console.log("Hello JS"); </script>`

External: `<script src="main.js"></script>`

- Kiểu dữ liệu trong JavaScript

JavaScript có 7 kiểu dữ liệu nguyên thủy (primitive): number, string, boolean, null, undefined, symbol, BigInt và 1 kiểu dữ liệu phức tạp: Object,

- Biến trong JS

Var: biến toàn cục, ít dùng

Let: biến cục bộ

Const: không thể gán lại

Ví dụ:

`let a = 10;`

`const PI = 3.14;`

- Toán tử trong JavaScript

Toán tử số học: +, -, *, /, %, **

So sánh: ==, ===, !=, !==, >, <, >=, <=

Logic: &&, ||, !

Gán: =, +=, -=, *=, ...

- Cấu trúc điều khiển

If – else:

```
if (a > 10) {
```

```
...
```

```
} else {
```

```
...
```

```
}
```

Switch:

```
switch (key) {
```

```
    case 1: break;
```

```
    case 2: break;
```

```
}
```

Vòng lặp:

For

While

Do ... while

Ví dụ:

```
for (let i = 0; i < 5; i++) {
```

```
    console.log(i);
```

```
}
```

- Hàm trong JS

Hàm thường:

```
function greet(name) {
```

```
    return "Hello " + name;
```

```
}
```

Arow function:

```
const greet = (name) => "Hello " + name;
```

- Object trong JS

```
const sinhVien = {
```

```
name: "Nam",  
age: 18,  
study: function() {  
    console.log("Đang học");  
}  
};
```

- Mảng (Array)

```
let arr = [1, 2, 3];
```

Các phương thức quan trọng: push(), pop(), shift(), unshift(), map(), filter(), reduce(), forEach()

- DOM (Document Object Model)

DOM cho phép JS tương tác với HTML/CSS.

- Truy xuất phần tử
document.getElementById("id")
document.querySelector(".class")
document.querySelectorAll("p")
- Thay đổi nội dung
element.innerHTML = "Nội dung mới";
element.textContent = "Text mới";
- Thay đổi CSS
element.style.color = "red";
- Tạo phần tử
const div = document.createElement("div");
document.body.appendChild(div);

- Xử lý sự kiện (Events)

```
button.addEventListener("click", function() {  
    alert("Bạn đã bấm!");  
});
```

- JSON trong JavaScript

```
let data = '{ "name": "An", "age": 18 }';
```

```
let obj = JSON.parse(data); // JSON → Object
```

```
let json = JSON.stringify(obj); // Object → JSON
```

- **LocalStorage**

```
localStorage.setItem("name", "Nam");
```

```
localStorage.getItem("name");
```

```
localStorage.removeItem("name");
```

```
localStorage.clear();
```

- **Hàm xử lý thời gian**

```
setTimeout(function() {  
    console.log("Delay 2 giây");  
}, 2000);
```

```
setInterval(() => {  
    console.log("Lặp mỗi giây");  
}, 1000);
```

➤ **Khái niệm và nguyên tắc UI/UX, responsive design, accessibility**

- **UI – User Interface (Giao diện người dùng)**

- UI là hình thức, giao diện, màu sắc, kiểu chữ, bố cục mà người dùng nhìn thấy và tương tác, tập trung vào tính thẩm mỹ và tính nhất quán.

- Về cơ bản, chúng ta có thể hiểu UI là sự liên kết giữa người dùng và chức năng của sản phẩm. UI giúp đạt được kết quả cuối cùng thông qua một loạt tương tác giữa người dùng và máy. Người dùng sẽ truyền đạt nhu cầu, mong muốn của họ và sản phẩm sẽ đưa ra giải pháp cho những nhu cầu, mong muốn đó.

- **UX – User Experience (Trải nghiệm người dùng)**

- UX là toàn bộ **trải nghiệm của người dùng khi tương tác với sản phẩm** (website, app...), tập trung vào **tính hữu dụng, dễ dùng, hiệu quả, tối ưu thao tác**.

- Quy trình 8 bước thiết kế UX :

○ **Bước 1: Xác định (Define)**

Trong giai đoạn đầu tiên, bạn cần xác định chính xác mình cần tạo những gì và vì sao. Tại sao cần tạo ra sản phẩm này? Sản phẩm này phục vụ cho ai? Nó có thể giải quyết những vấn đề nào? Các cuộc họp với các bên liên quan sẽ giúp bạn trả lời những câu hỏi này.

○ **Bước 2: Nghiên cứu (Research)**

Trong giai đoạn này, các designer tiến hành nghiên cứu để hiểu sâu hơn về người dùng và nhu cầu của họ. Nghiên cứu này giúp họ tạo ra sự đồng cảm với người dùng và hiểu những gì họ cần từ sản phẩm hoặc dịch vụ.

○ **Bước 3: Phân tích & Lập kế hoạch (Analysis & Planning)**

Trong giai đoạn lập kế hoạch, các designer lấy tất cả thông tin họ thu thập được trong giai đoạn nghiên cứu và bắt đầu lên kế hoạch về cách họ sẽ đáp ứng những nhu cầu đó. Designer sẽ phát triển chân dung người dùng (user personas), câu chuyện của người dùng (user stories), wireframe và các kế hoạch cấp cao khác trong giai đoạn này.

○ **Bước 4: Thiết kế (Design)**

Khi đã hiểu rõ về người dùng của mình và có kế hoạch phát triển, designer sẽ bắt đầu phác thảo một số ý tưởng về cách người dùng sẽ tương tác với giao diện của họ. Ở giai đoạn này, designer sẽ nghĩ về những thứ như bố cục tổng thể, điều hướng và các yếu tố cụ thể trên mỗi trang.

○ **Bước 5: Tạo mẫu (Prototyping)**

Sau khi có giao diện người dùng (UI), giờ đây bạn sẽ biến nó thành một nguyên mẫu hoạt động. Tạo nguyên mẫu (prototyping) cho phép bạn trình bày trải nghiệm thực tế hơn cho thử nghiệm khả năng sử dụng (usability testing) của mình, sau đó có thể cung cấp phản hồi chính xác hơn và thông tin chi tiết về “những gì đang hoạt động” và “những gì không hoạt động”.

- **Bước 6: Thử nghiệm (Testing)**

Trước khi ra mắt, bạn cần phải kiểm tra giao diện với người dùng thực. Thử nghiệm khả năng sử dụng (usability testing) giúp xác định bất kỳ điều gì cần cải thiện trước khi sản phẩm cuối cùng được đưa vào hoạt động và cung cấp phản hồi từ người dùng.

- **Bước 7: Khởi chạy (Launch)**

Sau khi thử nghiệm hoàn tất và tất cả các thay đổi cần thiết đã được thực hiện đối với hi-fi UI, sản phẩm đã sẵn sàng để bàn giao cho nhóm phát triển triển khai.

- **Bước 8: Cải tiến (Iteration)**

Sau khi ra mắt, sản phẩm (dù là trang web, ứng dụng hay sản phẩm kỹ thuật số khác) vẫn chưa hoàn thành. Quá trình thiết kế là một chu kỳ liên tục được cải tiến khi người dùng tương tác và cung cấp phản hồi về sản phẩm.

- **Responsive Design (Thiết kế đáp ứng)**

- Responsive Web Design (RWD) hay Thiết kế Web đáp ứng là một cách tiếp cận trong thiết kế, giúp các website hiển thị tốt trên nhiều thiết bị và kích thước màn hình (điện thoại, laptop, desktop,...) nhằm đảm bảo tính khả dụng và sự hài lòng của người dùng

- **Accessibility (Khả năng tiếp cận)**

- Hỗ trợ truy cập vào internet thuận tiện nhất cho người khiếm thị, người khiếm thính, người khuyết tật vận động, người bị suy giảm nhận thức...

- **Nguyên tắc phối màu**

- Phối màu đơn sắc (Monochromatic)
- Phối màu tương đồng (Analogous)
- Phối màu tương phản (Complementary)
- Phối màu bộ ba (Triadic)
- Phối màu bổ túc xen kẽ (Split – complementary)
- Phối màu bổ túc bộ bốn (Rectangular Tetradic/Compound Complementary)

- **Những cách sắp xếp bố cục**

- Bố cục cân bằng
- Bố cục tương phản
- Bố cục đồng nhất
- Bố cục dòng chảy thị giác
- Bố cục nhấn mạnh

- **Typography**

- **Typography** – Nghệ thuật trình bày chữ có thể đã xuất hiện từ thế kỷ XI. Trước thời đại kỹ thuật số 4.0 như hiện nay, sáng tạo về Typography là một nghề chuyên biệt gắn liền với sách và tạp chí hay các tác phẩm được xuất bản rộng rãi. Ví dụ đầu tiên về Typography có thể được nhìn thấy trong Kinh thánh Gutenberg, cuốn sách đã khởi động một cuộc cách mạng về nghệ thuật trình bày chữ ở phương Tây. Và Typography được sử dụng trong Kinh thánh Gutenberg hiện được gọi là Textura. Bạn có thể tìm thấy nó trong menu thả xuống phong chữ trên các ứng dụng máy tính để bàn lớn hiện nay.

Font chữ ấn tượng: Acropolis, Cyclone, Font zag free, Gotham, Ziggurat,...

- **Kiến thức cơ bản về hosting, domain, server**

- **Hosting** là dịch vụ lưu trữ dữ liệu website trên máy chủ, nơi chứa toàn bộ nội dung của trang web như tệp tin, mã nguồn, hình ảnh, và cơ sở dữ liệu.

- Có thể nói dễ hiểu là Hosting giống như mảnh đất, nơi bạn đặt: Nhà (Website), bàn ghế (nội dung bài viết), tranh ảnh (hình ảnh), tủ bếp (cơ sở dữ liệu) Tivi, đèn, quạt... (mã nguồn và chức năng website)

- Vai trò:

Lưu trữ dữ liệu

Đảm bảo website hoạt động liên tục

- **Domain** (tên miền) là địa chỉ của website trên Internet, ví dụ: example.com. Nó giống như "địa chỉ nhà" giúp người dùng dễ dàng truy cập trang web thông qua trình duyệt mà không cần nhớ địa chỉ IP phức tạp.

- Vai trò

Cho phép người dùng tìm và truy cập website nhanh chóng

Xây dựng thương hiệu trực tuyến (ví dụ: google.com, tiki.vn)

- Cấu trúc:

Tên chính: Phần do bạn chọn, ví dụ: "index"

Tiện ích (đuôi mở rộng): Như .com, .vn, .org, v.v.

- Cách hoạt động: Domain liên kết với Hosting thông qua **DNS (Domain Name System)**, chuyển tên miền thành địa chỉ IP của máy chủ để hiển thị website.

- **Server** hay còn được gọi là máy chủ là một hệ thống (bao gồm phần mềm và phần cứng máy tính tương ứng). Máy chủ (Server) có thể chạy trên một máy tính hoặc nhiều máy tính. Máy chủ Server được kết nối với mạng máy tính hoặc Internet, có năng lực xử lý cao và có IP tĩnh. Trên máy chủ sẽ được cài đặt các phần mềm nhằm phục vụ cho các máy tính khác truy cập để đáp ứng các yêu cầu cung cấp các dịch vụ và tài nguyên.

- Phân loại:

Máy chủ riêng – Dedicated

Máy chủ ảo – Virtual Private Server – VPS

Máy chủ đám mây – Cloud Server

2.2 Khảo sát các sản phẩm tương tự

2.2.1 Quizizz

- **Giới thiệu chung:** Quizizz là nền tảng tạo bài kiểm tra (quiz), học tập tương tác và chơi theo dạng trò chơi. Người dùng có thể làm bài trực tuyến theo thời gian thực hoặc tự học.

- **Giao diện (UI):**

- Màu sắc tươi sáng, chủ đạo là tím – xanh.
- Nhiều hình minh họa dạng cartoon, tạo cảm giác vui vẻ.
- Layout dạng card trực quan, dễ nhìn.
- Thanh menu trái + nội dung chính hiển thị bên phải, tối ưu không gian.

- **Trải nghiệm người dùng (UX):**

- Dễ thao tác: tạo quiz → chia sẻ mã → người dùng làm bài.
- Có chế độ luyện tập, chế độ chơi theo lớp học.

- Tự động thống kê điểm, phân loại câu đúng/sai.
- Hiệu ứng hoạt hình tạo cảm giác gamification → tăng hứng thú.

- Tính năng chính

- Tạo quiz nhiều dạng câu hỏi (trắc nghiệm, điền từ, sắp xếp...).
- Ngân hàng câu hỏi phong phú.
- Báo cáo kết quả chi tiết.
- Hỗ trợ đăng nhập bằng Google, Microsoft.

- Điểm mạnh

- UI đẹp, hiện đại.
- Tính năng đa dạng, dễ dùng.
- Gamification tốt (xếp hạng, điểm, âm thanh...).

- Điểm hạn chế

- Một số tính năng nâng cao phải trả phí.
- Tải trang lâu khi mạng yếu (do nhiều hình ảnh, animation).

2.2.2 Kahoot!

- Giới thiệu chung: Kahoot là nền tảng tạo quiz theo dạng game, phổ biến trong giáo dục. Người học tham gia bằng mã PIN.

- Giao diện (UI):

- Thiết kế tối giản hơn Quizizz.
- Màu sắc chủ đạo tím và trắng.
- Giao diện tập trung vào màn hình lớn (lớp học xem trên máy chiếu).

- Trải nghiệm người dùng (UX):

- Quy trình sử dụng đơn giản: Create → Host → Join.
- Người học trả lời bằng điện thoại → đồng bộ thời gian thực.
- Hệ thống tính điểm theo tốc độ trả lời → tạo cảm giác cạnh tranh.

- Tính năng chính

- Tạo quiz trắc nghiệm, đúng/sai.
- Thống kê kết quả.
- Chế độ chơi trực tiếp hoặc giao bài tập.

- Điểm mạnh

- Trải nghiệm “game show” thú vị.
- Dùng tốt cho lớp học trực tiếp.
- Tương thích nhiều thiết bị.

- Điểm hạn chế

- Ít dạng câu hỏi so với Quizizz.
- Giao diện di động đôi khi nhỏ, khó bấm.
- Nhiều giới hạn với tài khoản miễn phí.

2.2.3 Google Forms Quiz

- Giới thiệu chung: Google Forms là công cụ tạo biểu mẫu, khảo sát, quiz đơn giản, phù hợp cho giáo viên hoặc sinh viên.

- Giao diện (UI):

- Tối giản, đơn sắc, dễ dùng.
- Sử dụng layout theo cột, rất rõ ràng.
- Tùy chỉnh giao diện ít (chủ yếu chỉnh theme màu).

- Trải nghiệm người dùng (UX):

- Làm quiz trực tiếp trên mọi thiết bị.
- Tự động chấm điểm, thống kê bằng Google Sheets.
- Tốc độ tải nhanh, nhẹ.

- Tính năng chính

- Tạo các loại câu hỏi: trắc nghiệm, hộp kiểm, thang đo, đoạn văn...
- Tự động tính điểm bài quiz.
- Xuất kết quả sang Google Sheets.

- Điểm mạnh

- Giao diện nhanh, nhẹ, dễ dùng.
- Hoạt động tốt trên mọi thiết bị.
- Tích hợp hệ sinh thái Google.

- Điểm hạn chế

- Không có gamification.

- Ít hiệu ứng, không hấp dẫn bằng Kahoot/Quizizz.
- Khó dùng để cạnh tranh theo thời gian thực.

➤ **So sánh về giao diện, chức năng, UX và tốc độ tải**

1 – Tốt nhất; 3 – Thấp nhất

	Giao diện	Chức năng	UX	Tốc độ tải
Quizizz	1	1	1	3
Kahoot!	2	2	2	2
GG Form Quiz	3	3	3	1

➤ **Nhận xét, rút ra bài học, định hướng cho sản phẩm nhóm**

• **Nhận xét chung sau khi phân tích website cùng lĩnh vực**

- Qua việc phân tích 3 nền tảng quiz tiêu biểu (Quizizz, Kahoot và Google Forms), nhóm nhận thấy:

- Mỗi nền tảng có thế mạnh khác nhau: Quizizz nổi bật về giao diện và gamification; Kahoot mạnh về tương tác thời gian thực; Google Forms tối ưu tốc độ và tính đơn giản.
- UI/UX ảnh hưởng trực tiếp đến trải nghiệm làm bài. Những nền tảng có giao diện đẹp, sinh động tạo hứng thú học tập rõ rệt.
- Tính năng phong phú không phải lúc nào cũng cần thiết, quan trọng là phù hợp với mục tiêu của người dùng.
- Tốc độ tải và khả năng sử dụng trên nhiều thiết bị là yếu tố quan trọng, đặc biệt với học sinh – sinh viên sử dụng điện thoại.

• **Bài học rút ra cho quá trình phát triển sản phẩm**

- Từ quá trình phân tích, nhóm rút ra một số bài học quan trọng:

- Giao diện cần trực quan, đơn giản, tránh rối mắt; ưu tiên bố cục rõ ràng như Google Forms nhưng có điểm nhấn sinh động giống Quizizz.
- Nên có mức độ gamification vừa phải để tăng hứng thú, nhưng không quá phức tạp.

- Tốc độ tải phải được ưu tiên, tránh sử dụng quá nhiều hình ảnh dung lượng lớn gây nặng trang.
- Cần có chức năng thống kê điểm, câu đúng/sai để người dùng thấy được kết quả sau mỗi bài quiz.
- Trải nghiệm nhất quán trên các thiết bị (desktop – mobile) để phù hợp đối tượng học sinh, sinh viên.
- Tập trung vào mục tiêu học tập, không chỉ là “game chơi vui”.
- **Định hướng phát triển cho sản phẩm của nhóm**
 - **Giao diện (UI):**
 - Sử dụng thiết kế phẳng (flat design) kết hợp màu sắc tươi sáng nhưng không quá sặc sỡ.
 - Bố cục dạng card cho danh sách quiz.
 - Tối ưu hiển thị cho cả desktop và mobile.
 - **Chức năng**
 - Tạo quiz với các dạng câu hỏi cơ bản: trắc nghiệm, đúng/sai.
 - Tính điểm tự động và hiển thị báo cáo sau khi làm bài.
 - Lưu lịch sử điểm theo từng bài để người học theo dõi tiến độ.
 - Thêm đếm giờ, thanh tiến trình, hoặc hiệu ứng nhẹ để tăng sự hấp dẫn giống Quizizz nhưng phù hợp khả năng nhóm.
 - **Trải nghiệm người dùng (UX)**
 - Quy trình làm bài rõ ràng: Chọn quiz → Bắt đầu → Trả lời → Xem kết quả.
 - Giao diện tối giản khi làm quiz để không phân tâm.
 - Phản hồi nhanh sau khi chọn đáp án.
 - **Tốc độ và hiệu suất**
 - Hạn chế dùng ảnh dung lượng lớn, tối ưu CSS/JS.
 - Tận dụng cache trình duyệt để tăng tốc độ tải.
 - Ưu tiên nền tảng nhẹ giống Google Forms.

2.3 Tổng hợp và định hướng thiết kế

➤ Tóm tắt các yếu tố sẽ áp dụng cho dự án “QUIZZKIT”

- **Về giao diện**
 - Thiết kế đơn giản – rõ ràng – hiện đại, tránh rối mắt.
 - Sử dụng màu sắc tươi nhưng nhẹ, tạo cảm giác học tập thân thiện.
 - Bố cục theo dạng thẻ (card) giúp người dùng dễ chọn quiz.
 - Tối ưu hiển thị tốt cho mobile và desktop.
- **Về chức năng**
 - Hỗ trợ các dạng câu hỏi cơ bản: trắc nghiệm, đúng/sai.
 - Chức năng xem lại câu đúng/sai và thống kê điểm.
 - Tạo trang quản lý quiz đơn giản (thêm/xóa/sửa câu hỏi).
- **Về trải nghiệm người dùng**
 - Quy trình làm bài rõ ràng: Chọn quiz → Làm bài → Kết quả.
 - Giao diện làm bài tối giản, hạn chế yếu tố gây phân tâm.
 - Hiệu ứng chuyển trang nhẹ (fade, slide) để trải nghiệm mượt hơn.
 - Nút bấm, câu hỏi, đáp án được bố trí dễ thao tác trên cả điện thoại.
- **Về tốc độ và hiệu suất**
 - Hạn chế ảnh và file nặng; ưu tiên SVG hoặc icon nhỏ.
 - Sử dụng phương thức lưu trữ nhẹ (localStorage) thay vì backend phức tạp.
- **Về hướng tiếp cận**
 - Ưu tiên tính dễ dùng cho người mới và phù hợp học tập.
 - Đảm bảo web dễ mở rộng thêm chức năng về sau.

➤ **Lý do chọn framework, công cụ và phong cách thiết kế cụ thể**

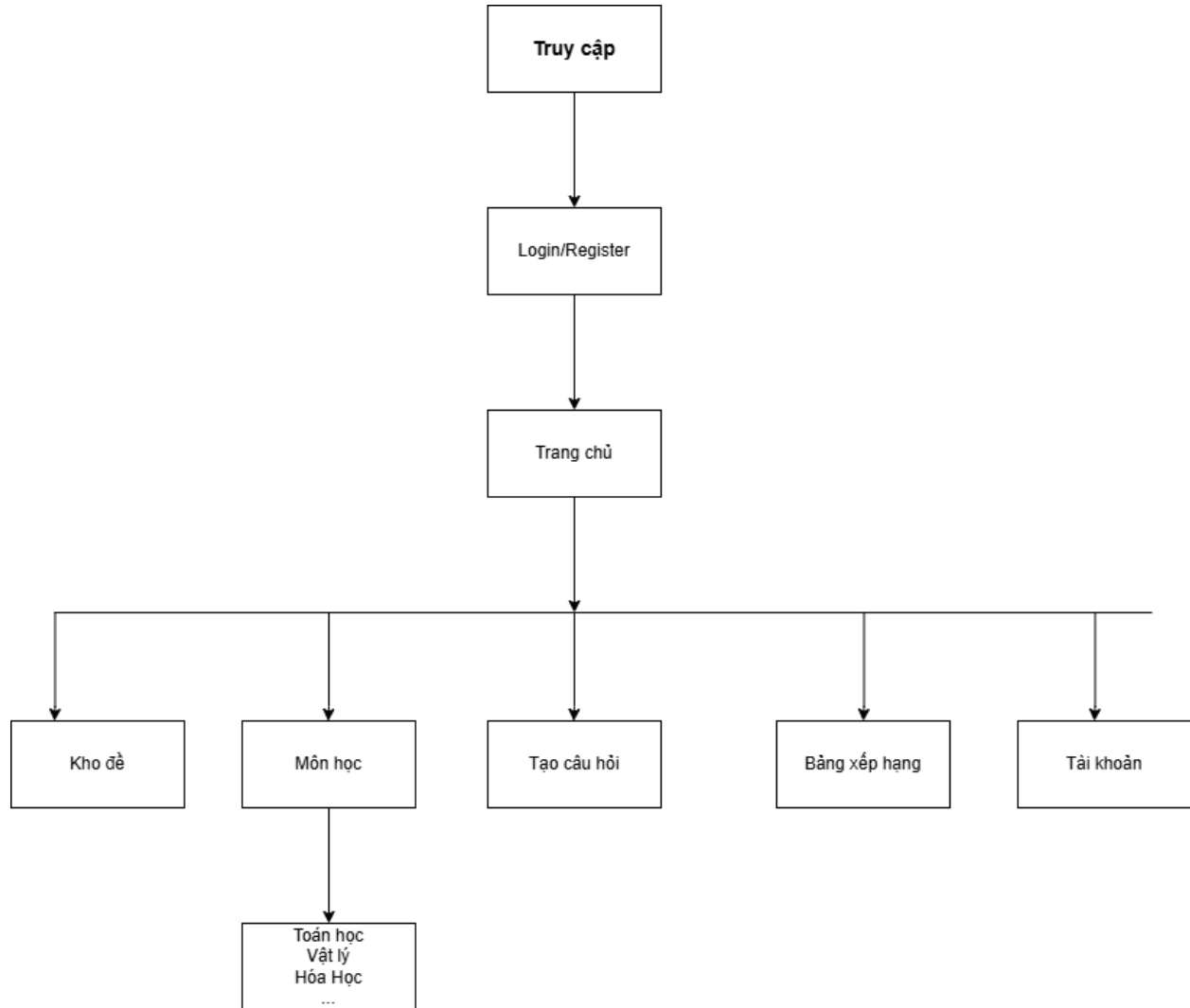
- **Lý do chọn ngôn ngữ & framework**
 - HTML5 – CSS3 – JavaScript:
 - Dễ học – dễ triển khai, phù hợp với phạm vi môn học và thời gian dự án.
 - Không yêu cầu backend phức tạp, chỉ cần xử lý phía client.
 - Tải nhanh, không phụ thuộc thư viện lớn.
 - Dễ tùy chỉnh theo ý đồ thiết kế của nhóm.
 - Hoạt động tốt trên mọi trình duyệt hiện nay.

- Không dùng backend mà sử dụng LocalStorage:
 - LocalStorage đáp ứng tốt việc: lưu điểm, lưu tiến trình, lưu dữ liệu.
 - Nhanh, gọn, không tốn tài nguyên.
- **Lý do chọn các công cụ hỗ trợ**
 - VS Code
 - Giao diện trực quan, nhiều extension hỗ trợ HTML/CSS/JS.
 - Tự động gợi ý code, linting, live server.
 - Nhẹ, phù hợp sinh viên.
 - Live Server
 - Xem kết quả trực tiếp khi code.
 - Tiết kiệm thời gian build.
 - GitHub
 - Quản lý phiên bản code.
 - Dễ phân chia công việc giữa các thành viên nhóm.
- **Lý do chọn phong cách thiết kế (UI/UX)**
 - Màu sắc nhẹ, tươi sáng
 - Tạo cảm giác thân thiện, phù hợp website học tập.
 - Không gây mỏi mắt khi làm quiz lâu.
 - Giúp các nút, card, câu hỏi nổi bật tự nhiên.
 - Ưu tiên trải nghiệm người dùng
 - Bố cục rõ ràng: Trang chủ → Chọn quiz → Làm bài → Xem kết quả.
 - Tương tác đơn giản → thao tác nhanh nhất có thể.
 - Hiện thị kết quả trực quan, giúp người học dễ nhận phản hồi.
 - Tối ưu mobile-first
 - Đa số sinh viên và người học làm quiz bằng điện thoại.
 - Giao diện co giãn tốt trên mọi kích thước màn hình.

CHƯƠNG 3: THIẾT KẾ VÀ TRIỂN KHAI WEBSITE

3.1 Thiết kế giao diện (UI Design)

Sitemap:



Hình 3.1 Sitemap

Bố cục tổng thể:

Header cố định: navbar, sidebar, avatar.

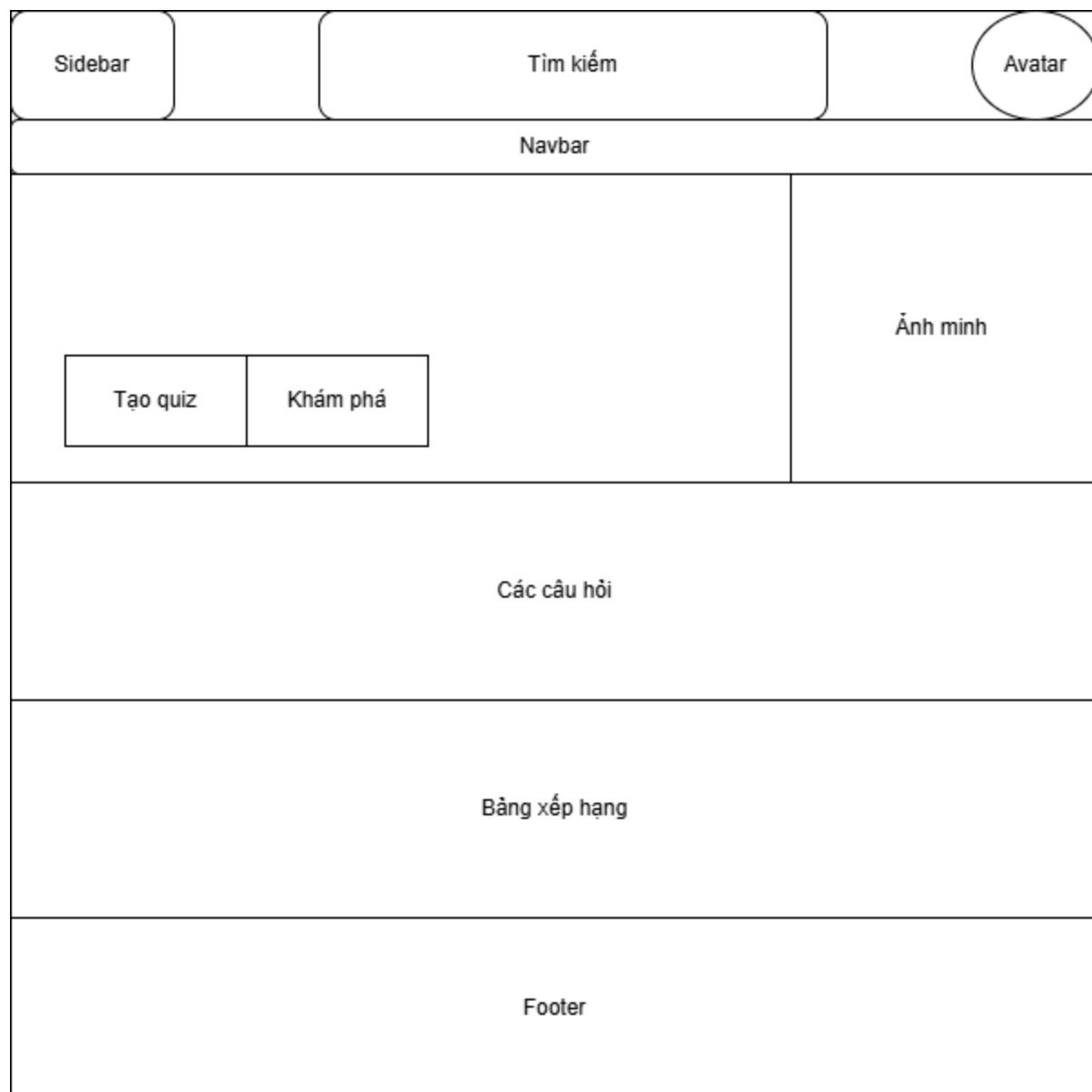
Body: Trang chủ, Kho đề, Môn học, Tạo câu hỏi, Bảng xếp hạng, Tài khoản.

Footer cố định: giới thiệu, liên hệ, nhóm.

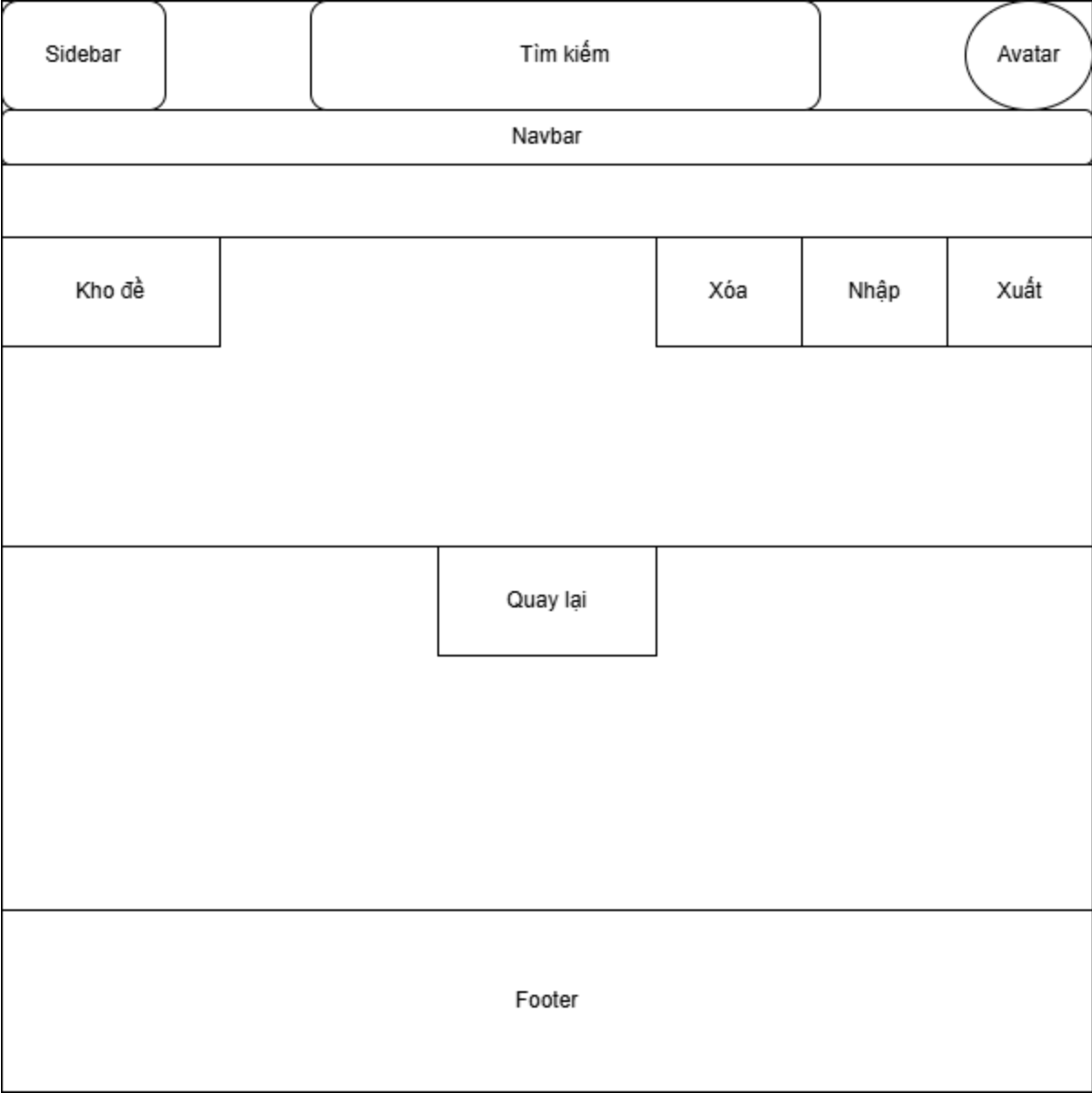
Wifeframe:

<div><p>Sign Up</p><p>Email</p><p>Username</p><p>Password</p><div><p>Sign Up</p><p>Login</p></div></div>	<div><p>Login</p><p>Username</p><p>Password</p><div><p>Login</p><p>Sign Up</p></div></div>
--	--

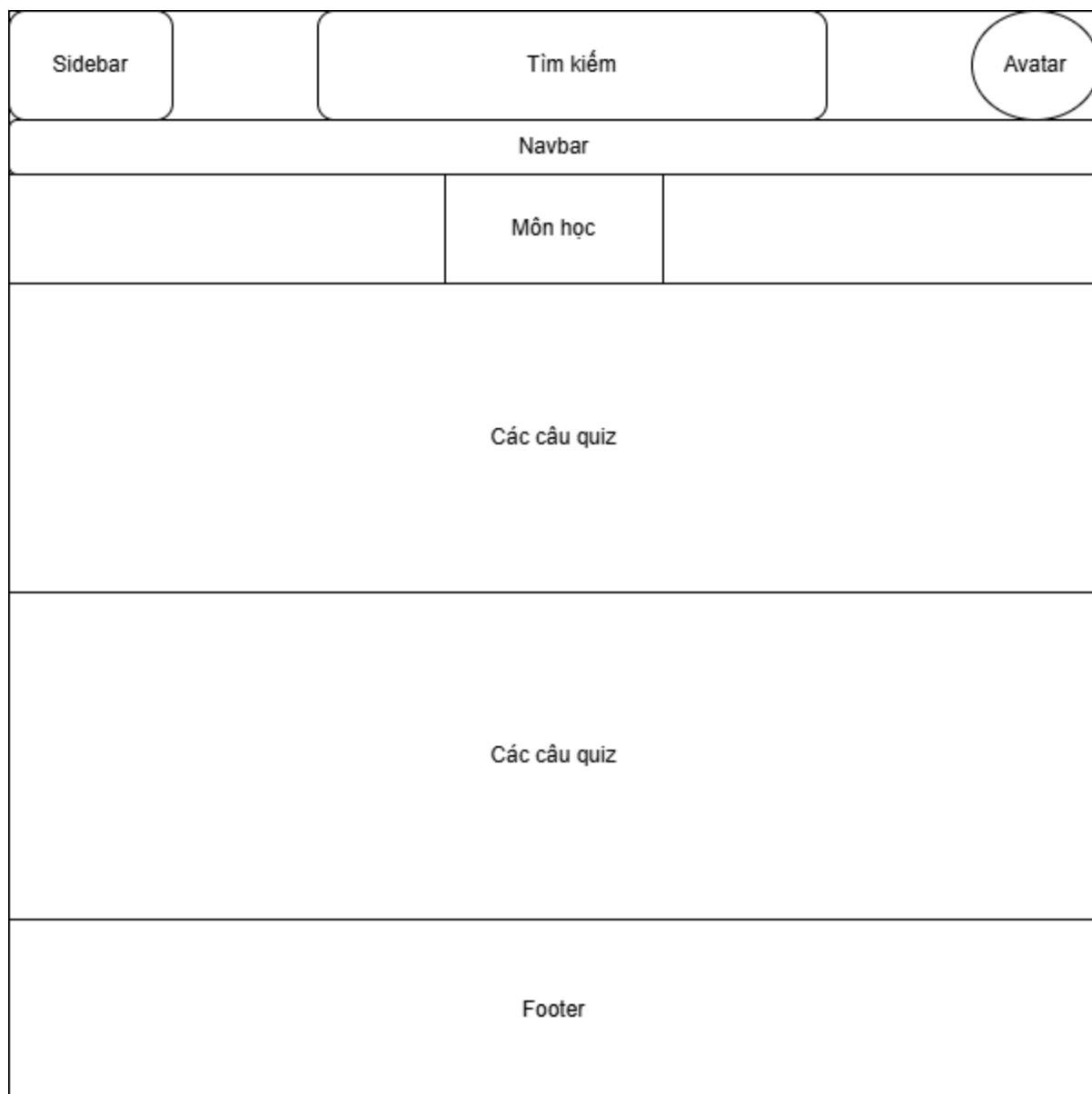
Hình 3.2 Login-SignUp



Hình 3.3 Home



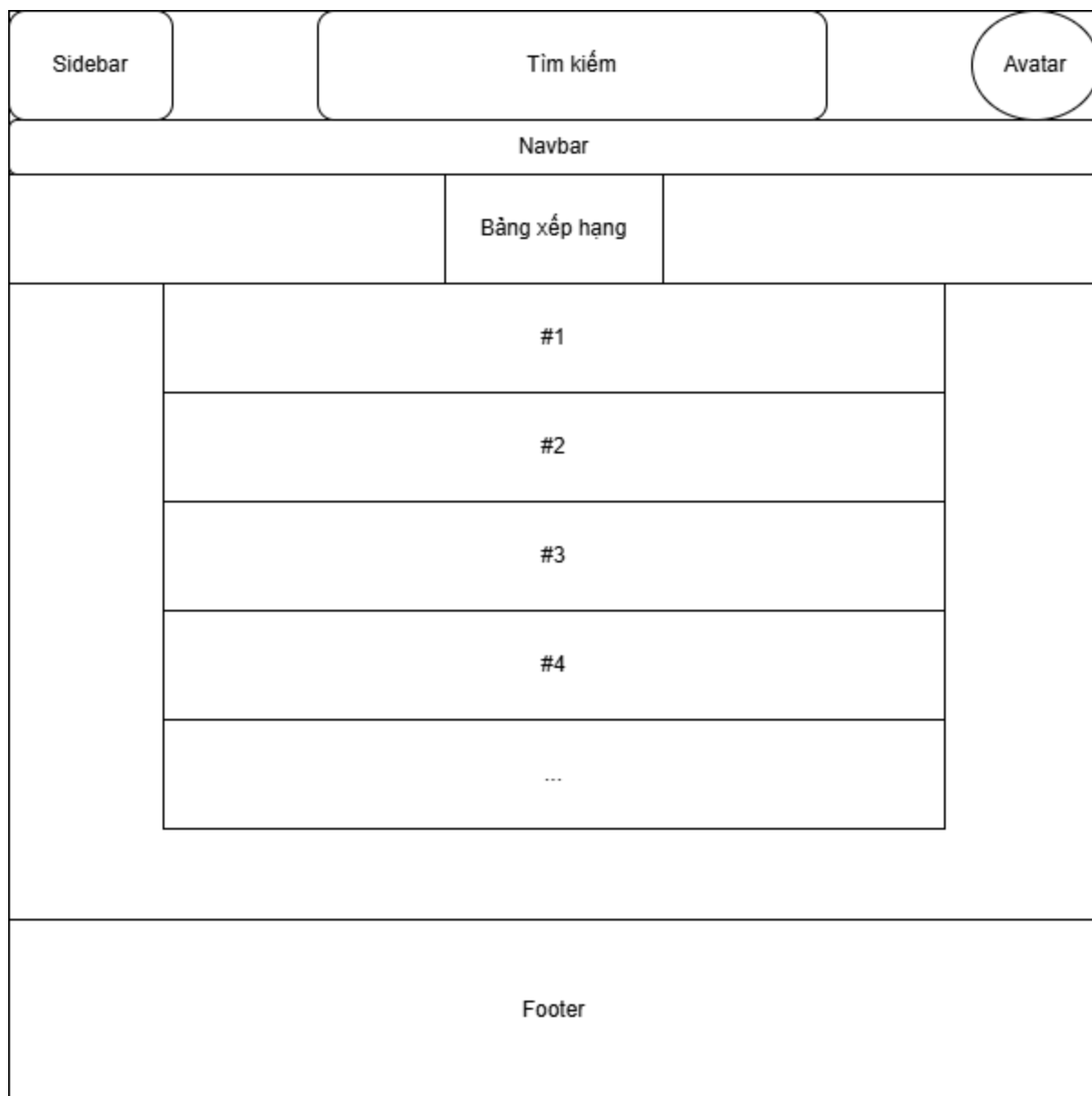
Hình 3.4 Kho đề



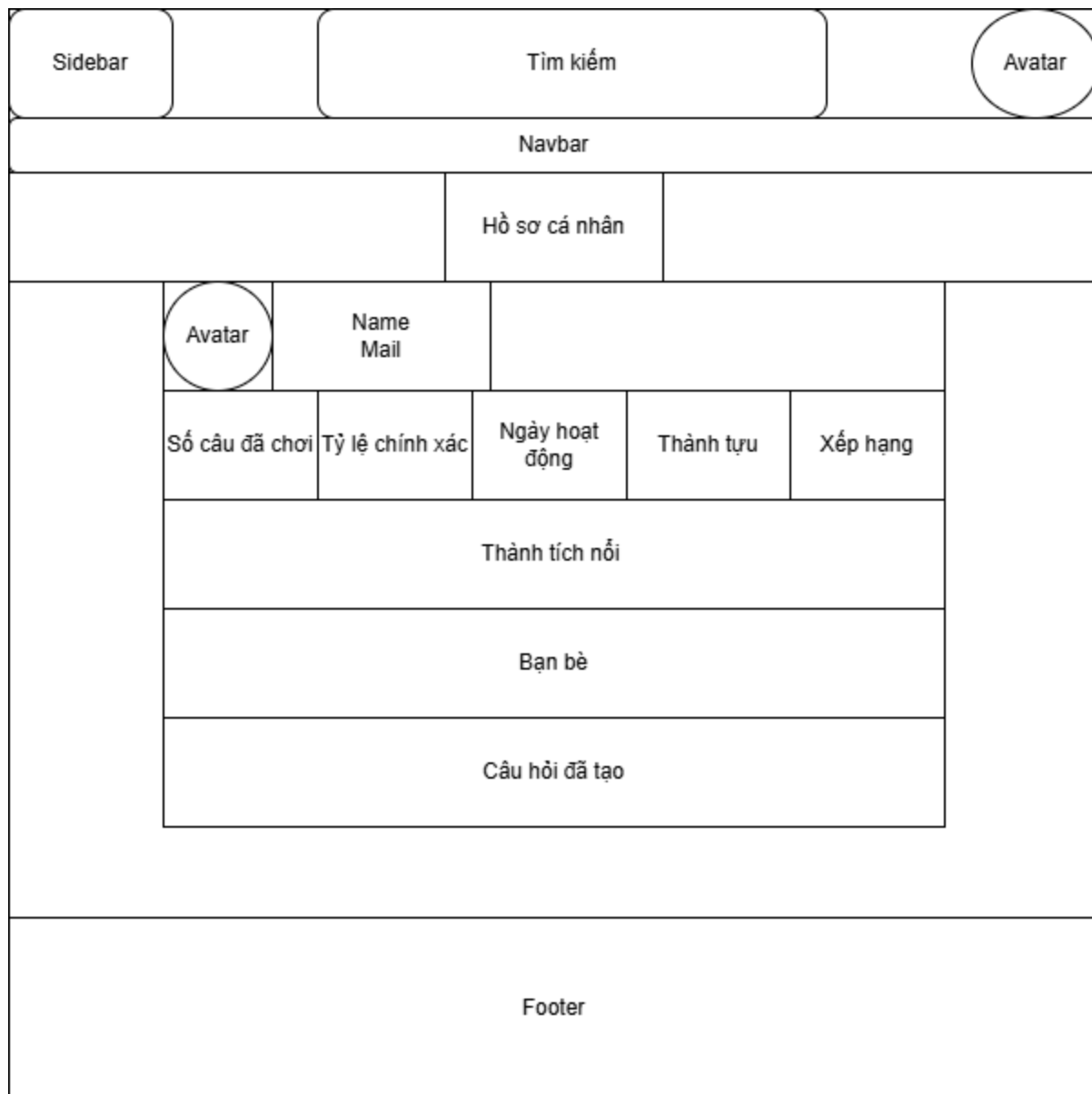
Hình 3.5 Môn học

Sidebar	Tìm kiếm		Avatar
Navbar			
	Tạo câu hỏi		
	Tên câu hỏi		
	Môn học		
	Số câu		
	Mô tả		
	Ảnh tùy chọn		
Footer			

Hình 3.6 Tạo câu hỏi



Hình 3.7 Bảng xếp hạng



Hình 3.8 Tài khoản

Bộ màu:

Màu chính: #e07b39, #5c4033

Màu phụ: #ffffff, #f6f7fb

Màu dark mode: #111827, #1f2937, #e5e7eb, #918521

Font chữ: Segoe UI

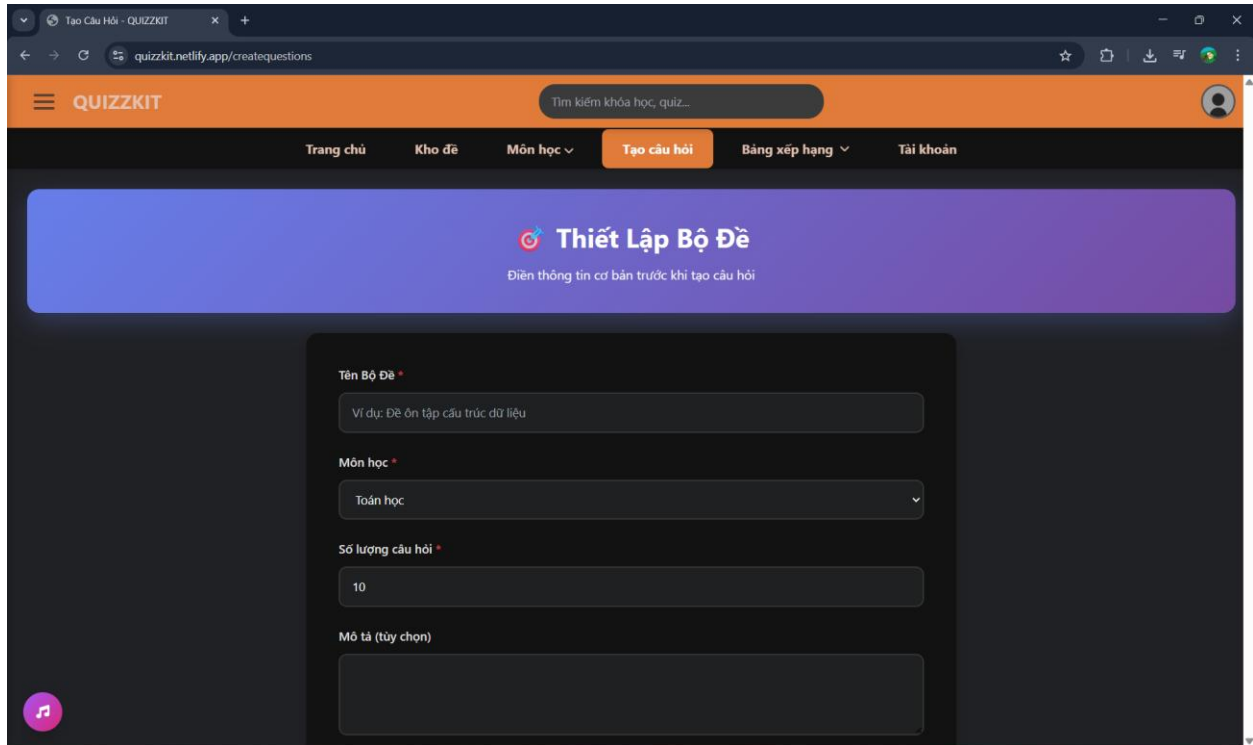
Icon: awesome

Nguyên tắc UX/UI được áp dụng:

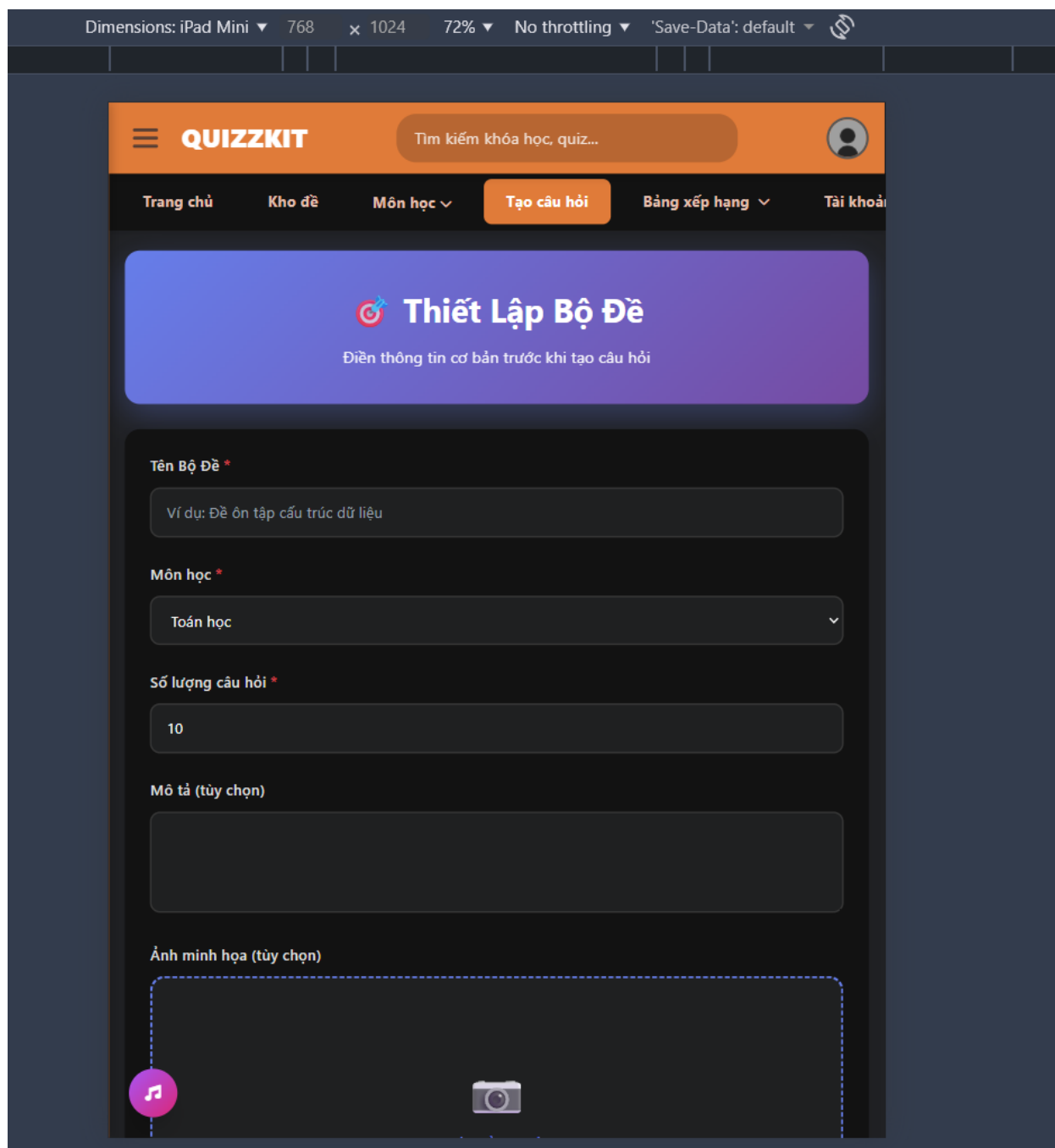
- **Đơn giản & dễ sử dụng:** Ít bước, thao tác nhanh.

- **Tính nhất quán:** Màu, font, nút được thống nhất trên toàn trang.
- **Tập trung vào nội dung chính:** Câu hỏi và đáp án hiển thị rõ ràng.
- Feedback rõ ràng: hover button, đáp án đúng sai có màu, thanh tiến trình
- **Bố cục trực quan:** Người dùng dễ hiểu luồng quiz.
- **Thân thiện người mới:** Không gây rối hoặc quá nhiều thông tin.

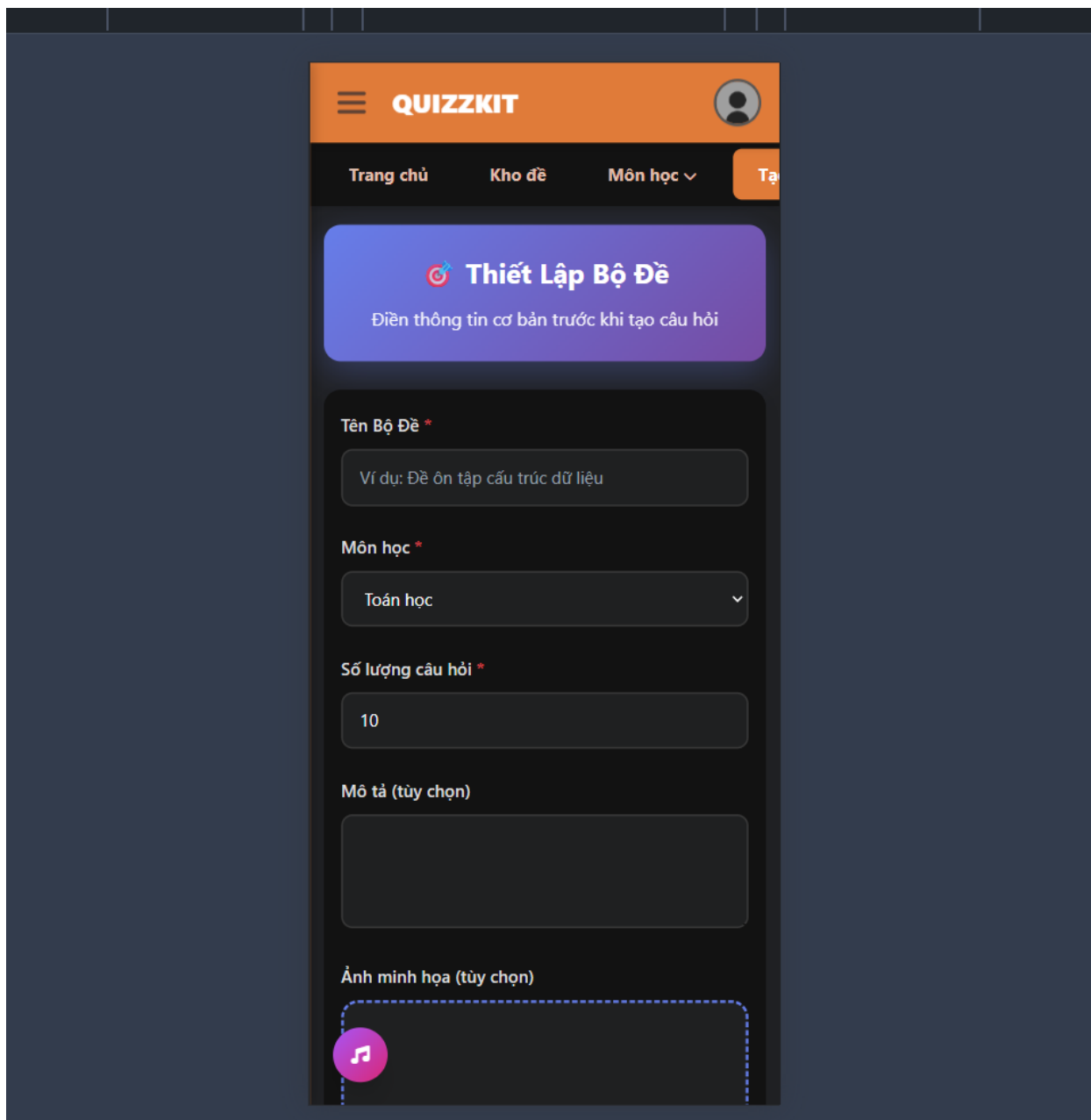
Responsive Design:



Hình 3.9 Desktop

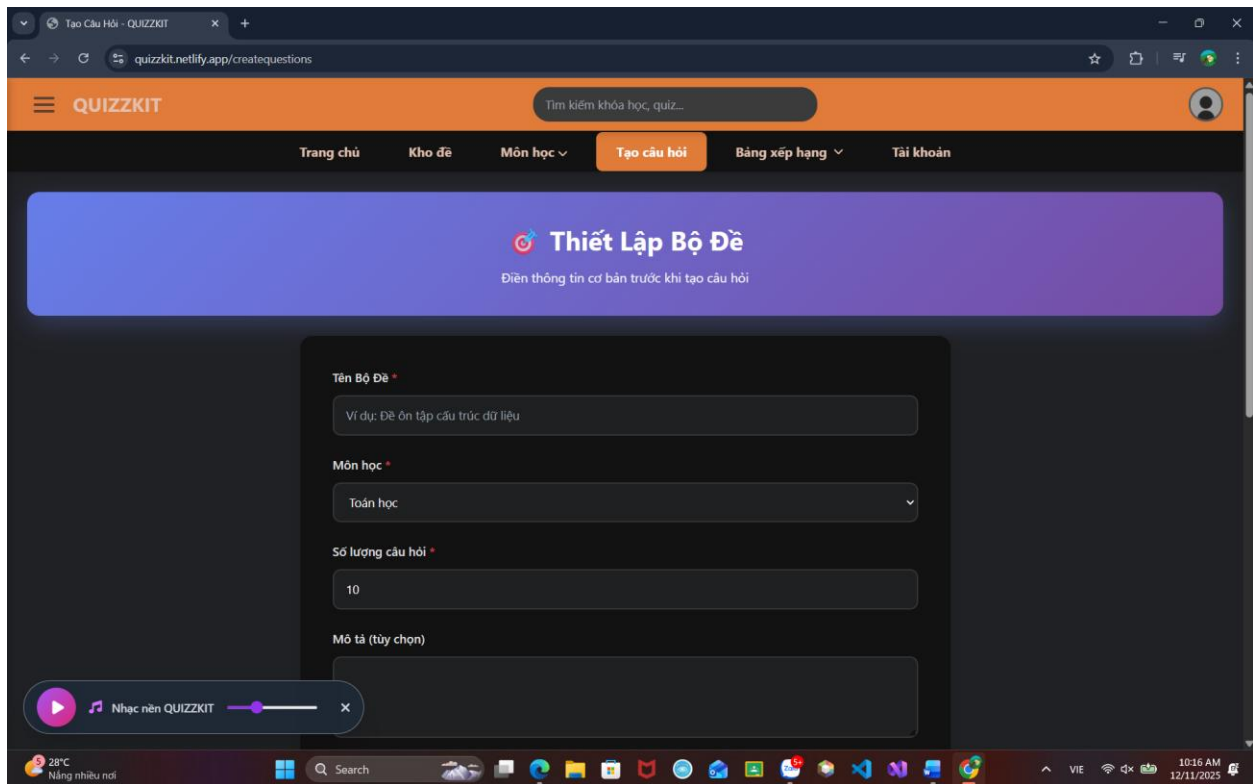


Hình 3.10 Tablet

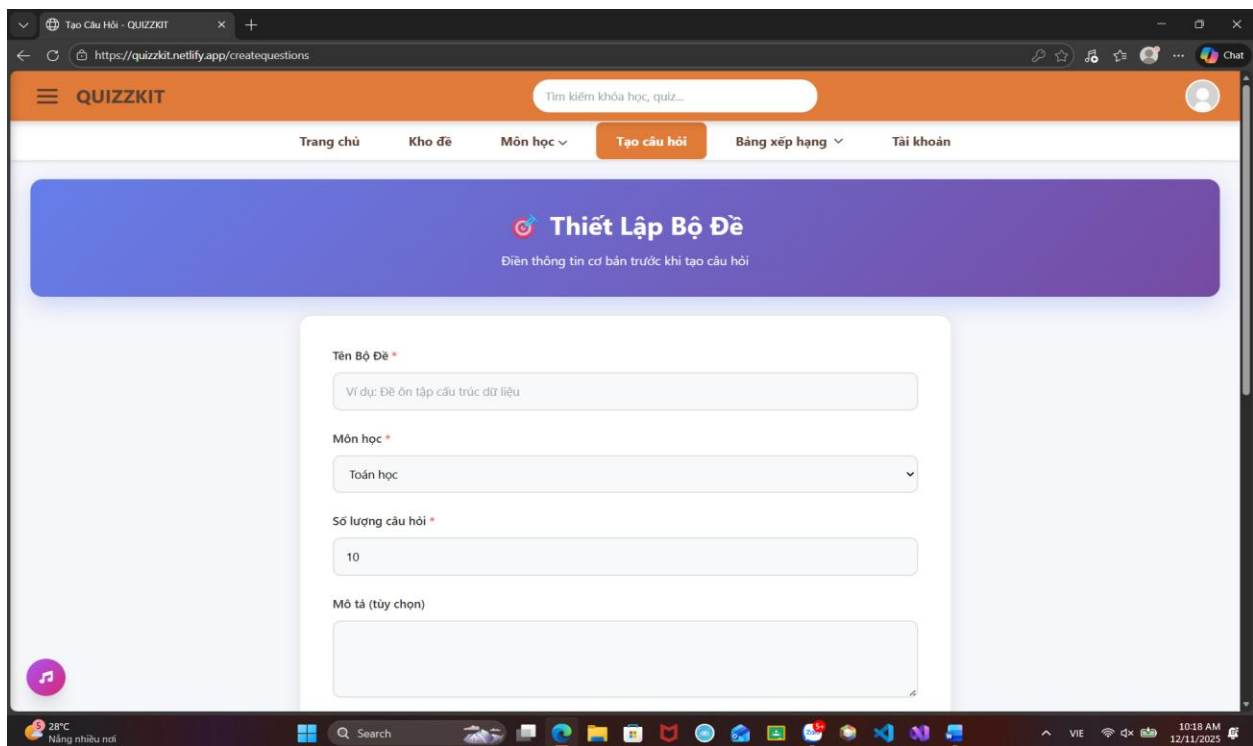


Hình 3.11 Samsung

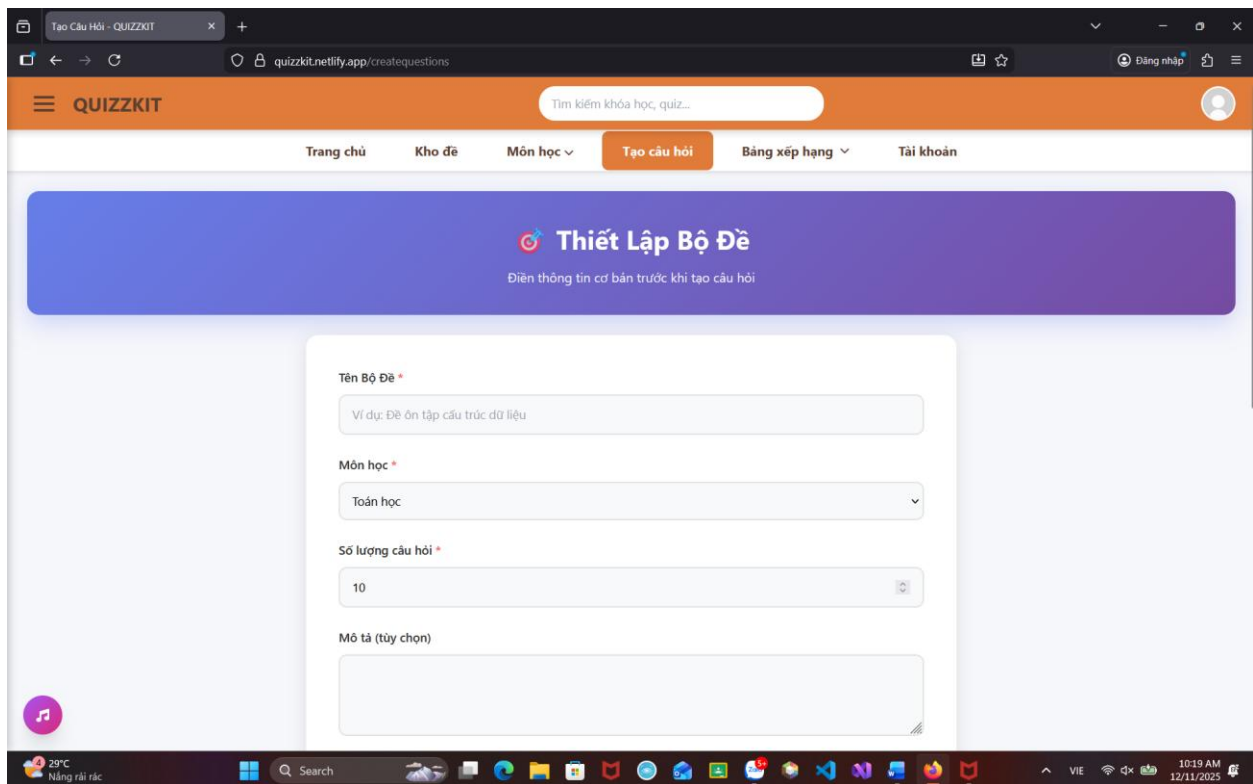
Tương thích trình duyệt:



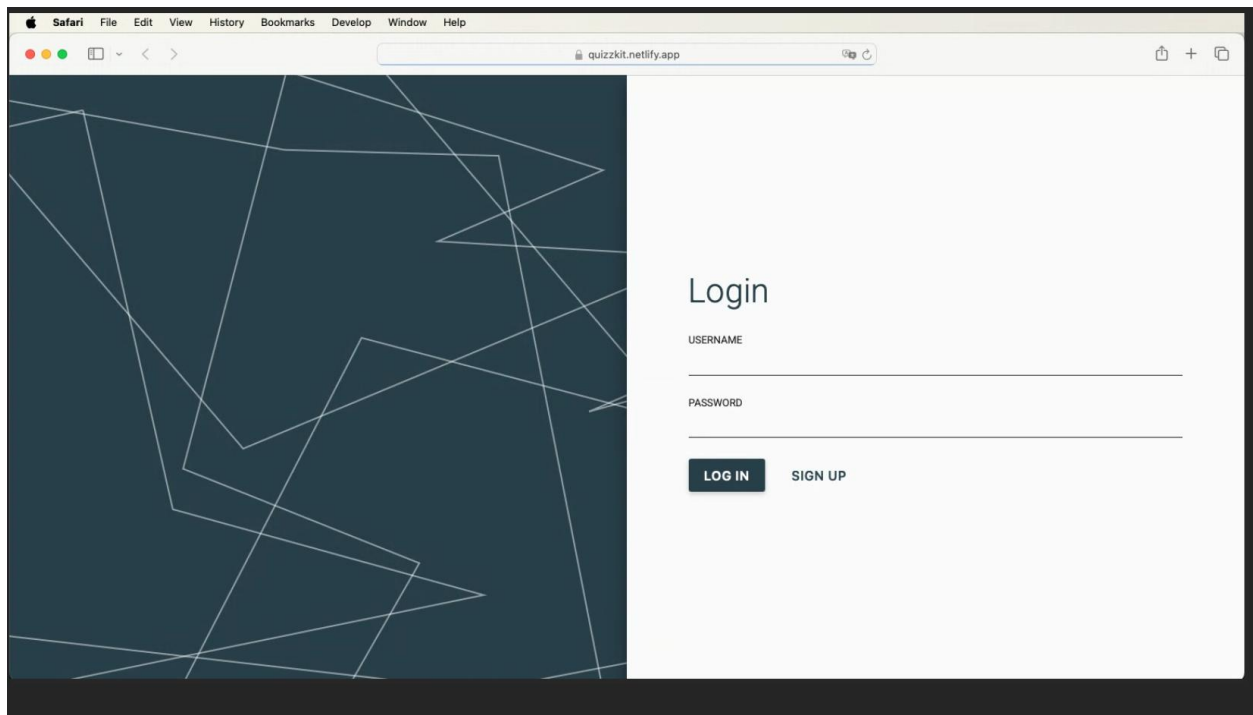
Hình 3.12 Chrome



Hình 3.13 Edge



Hình 3.14 Firefox



Hình 3.15 Safari

3.2 Chức năng và tương tác (UX Design)

➤ Các chức năng chính: tạo câu hỏi, kho đề và xem câu hỏi

- **Trang Tạo Câu Hỏi (taocauhoi.html):**

- Trang này cho phép người dùng xây dựng một bộ đề hoàn chỉnh từng bước, gồm các chức năng:

Nhập thông tin bộ đề:

- Tên bộ đề: Xác định tiêu đề của đề thi.
- Môn học: Lựa chọn lĩnh vực (Toán, Lý, Hóa, Tin học, ...).
- Số lượng câu hỏi: Xác định tổng số câu.
- Mô tả bộ đề: Mô tả nội dung, mức độ hoặc mục đích sử dụng.

Thêm ảnh đại diện:

- Nhấn vào khung ảnh → chọn ảnh từ máy.
- Ảnh sẽ được hiển thị xem trước (preview).
- Có nút xóa ảnh nếu chọn sai.
- Tính năng này giúp bộ đề trực quan, sinh động hơn khi hiển thị trong Kho đề.

Thêm câu hỏi trắc nghiệm:

- Nội dung câu hỏi.
- 4 đáp án: A, B, C, D.
- Người dùng chọn 1 đáp án đúng bằng radio button.

Hệ thống kiểm tra:

- Không cho lưu nếu thiếu nội dung.
- Phải có 1 đáp án đúng.

Thêm, sửa, hủy câu hỏi:

- Nút Thêm câu hỏi → thêm vào danh sách.
- Nút Sửa → chỉnh lại nội dung câu hỏi.
- Nút Hủy → hủy thao tác chỉnh sửa đang thực hiện.

Hiển thị tiến độ tạo đề:

- Thanh Progress bar hiển thị số câu đã tạo / tổng số câu.
- Người dùng dễ dàng theo dõi mức độ hoàn thành.

Lưu bộ đề vào hệ thống:

- Khi nhấn Lưu bộ đề, toàn bộ dữ liệu sẽ được lưu vào LocalStorage của trình duyệt

- Trang Kho Đề (khode.html)

- Trang này dùng để quản lý các bộ đề đã tạo, gồm các chức năng sau:

Hiển thị danh sách bộ đề:

- Tất cả bộ đề được hiển thị dưới dạng thẻ (quiz card).

Mỗi thẻ gồm:

- Ảnh đại diện, Tên bộ đề, Môn học
- Mô tả: Số câu hỏi

Xem chi tiết bộ đề:

- Nhấn nút Xem chi tiết
- Thông tin đề, Danh sách câu hỏi, Đáp án đúng được tô nổi bật

Hệ thống hỗ trợ xuất:

- File JSON: Dùng để lưu trữ, import lại sau
- File HTML: Dùng để in đề hoặc chia sẻ online.

➤ **Cách người dùng tương tác:**

Thành phần	Hình thức tương tác	Mục đích
Menu	Click	Chuyển trang
Form nhập	Gõ bàn phím	Nhập dữ liệu
Import JSON	Upload file	Thêm đề từ ngoài
Phần lớn...	Click	Thực hiện chức năng

➤ **Mô tả các đoạn mã hoặc logic quan trọng:**

- Lưu đề vào LocalStorage:

```
localStorage.setItem("quizList", JSON.stringify(data));
```

- Load đề từ kho:

```
const data = JSON.parse(localStorage.getItem("quizList")) || [];
```

- Xuất JSON:

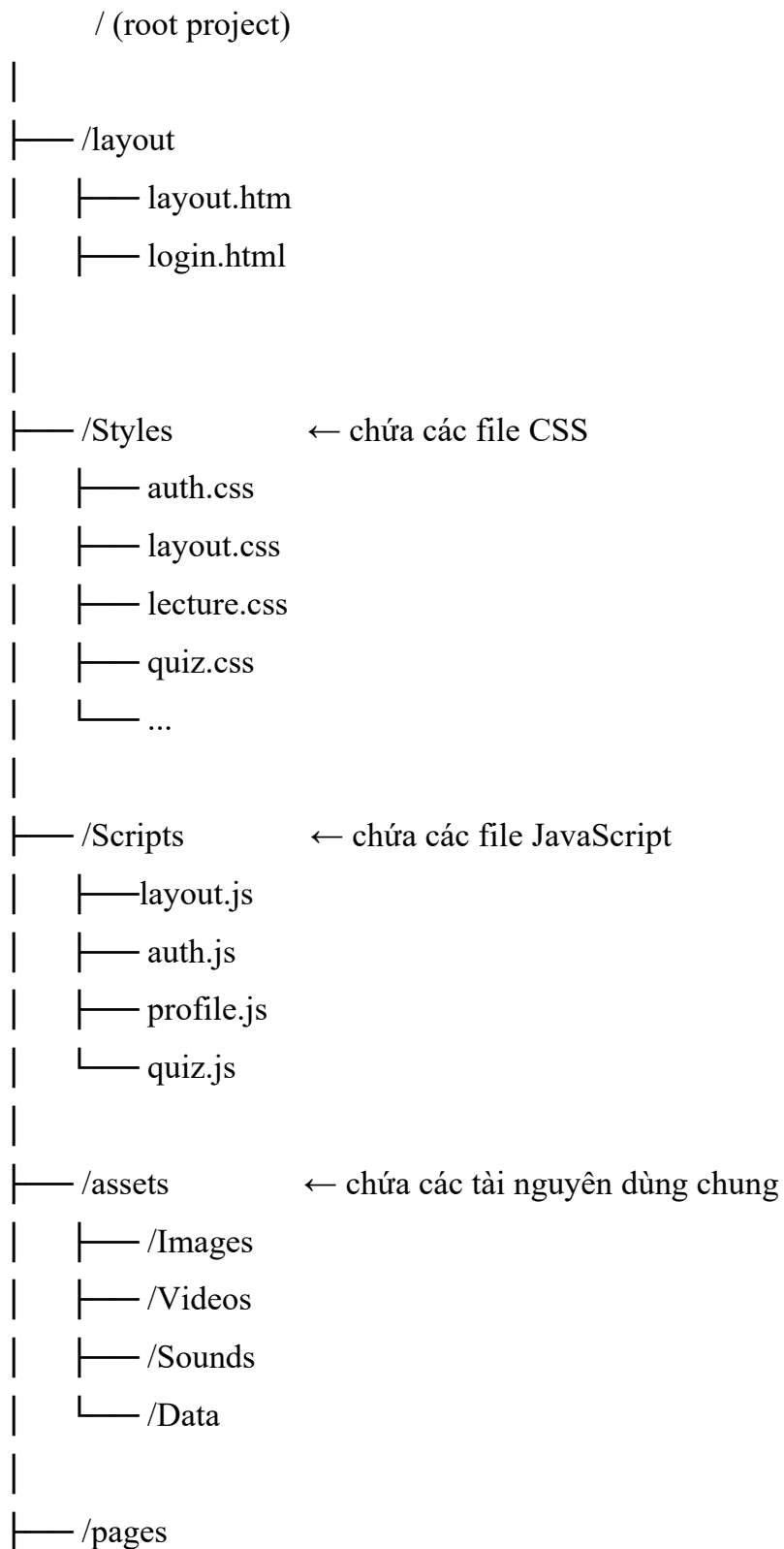
```
const blob = new Blob([JSON.stringify(quiz)], { type: "application/json" });
```

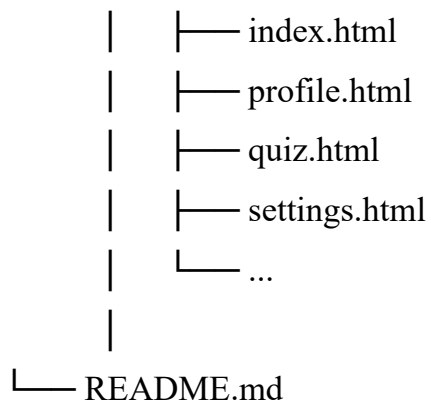
- Import đề từ file:

```
reader.onload = function(e) {  
    const importedData = JSON.parse(e.target.result);  
}
```


3.3 Mã nguồn và cấu trúc dự án

- **Cấu trúc thư mục**





- **Công cụ và thư viện sử dụng**

Công cụ	Mục đích
VS Code	Code chính
Live Server	Chạy web tĩnh
GitHub	Quản lý
Netlify	Deploy website
Thư viện	Công dụng
Font Awesome	Icon

- **Nguyên tắc đặt tên, tái sử dụng mã và comment code**

Nguyên tắc đặt tên: Biến, hàm JS đặt theo camelCase, class CSS đặt theo BEM / mô tả chức năng (Ví dụ: layout-sidebar, ...), tên thư mục viết thường, không dấu, có nghĩa.

Tái sử dụng mã: Layout sidebar – navbar – footer dùng chung cho toàn bộ trang.

Comment code: Comment chỉ mô tả mục đích – không mô tả lại code.

CHƯƠNG 4: KIỂM THỬ, TỐI ƯU HÓA VÀ TRIỂN KHAI

4.1 Kiểm thử giao diện và chức năng

➤ Kiểm tra từng chức năng

- Sau khi hoàn thành phát triển, nhóm đã tiến hành kiểm thử toàn bộ giao diện và chức năng của website. Kết quả cho thấy các thành phần giao diện được hiển thị đúng theo thiết kế, bố cục rõ ràng và không xuất hiện lỗi vỡ giao diện. Các chức năng như đăng nhập, đăng ký, chuyển trang, tải câu hỏi và làm bài quiz đều hoạt động ổn định, xử lý đúng dữ liệu và không phát sinh lỗi trong quá trình sử dụng.

➤ Kiểm thử responsive trên nhiều thiết bị

- Website hoạt động tốt trên nhiều kích thước màn hình khác nhau. Khi kiểm thử responsive trên desktop, tablet và mobile, tất cả các nội dung đều tự động co giãn hợp lý, menu và nút bấm hiển thị rõ ràng, đảm bảo trải nghiệm người dùng mượt mà.

- Ảnh minh họa chạy trên nhiều thiết bị được gắn với chương 3 (Hình 3.9, Hình 3.10, Hình 3.11).

➤ Kiểm tra tương thích trên các trình duyệt phổ biến

- Website cũng tương thích tốt với các trình duyệt phổ biến như Chrome, Edge, Firefox, Safari. Các hiệu ứng CSS, font chữ và đoạn mã JavaScript đều được trình duyệt xử lý đúng, không ghi nhận lỗi giao diện hay lỗi thực thi. Nhìn chung, kết quả kiểm thử cho thấy website đáp ứng đầy đủ yêu cầu về giao diện, chức năng và khả năng sử dụng.

- Ảnh minh họa chạy trên các trình duyệt phổ biến gắn với chương 3 (Hình 3.12, Hình 3.13, Hình 3.14, Hình 3.15).

4.2 Tối ưu hóa hiệu suất

- Trong giai đoạn hoàn thiện, nhóm đã tiến hành tối ưu hóa hiệu suất nhằm giúp website tải nhanh và hoạt động ổn định trên nhiều thiết bị. Các hình ảnh trong dự án được xử lý lại bằng cách nén dung lượng, đổi sang định dạng phù hợp và giảm kích thước không cần thiết, giúp giảm thời gian tải trang đáng kể. Toàn bộ mã CSS và JavaScript được tinh gọn, loại bỏ những đoạn mã không sử dụng và sắp xếp lại hợp lý để tăng tốc độ render của trình duyệt.

- Website được đánh giá bằng các công cụ Google PageSpeed và Lighthouse. Kết quả cho thấy tốc độ tải trang nằm ở mức tốt, các chỉ số như Largest Contentful Paint (LCP) và First Input Delay (FID) ổn định, không có lỗi nghiêm trọng ảnh hưởng đến trải nghiệm người dùng. Giao diện cũng được kiểm tra lại nhằm đảm bảo khả năng hiển thị mượt mà trên mạng yếu hoặc thiết bị cấu hình thấp.
- Bên cạnh đó, các tiêu chuẩn SEO cơ bản cũng được áp dụng để cải thiện khả năng tìm kiếm và mức độ thân thiện với bot của công cụ tìm kiếm. Nhóm đã bổ sung meta tags cho tiêu đề, mô tả trang; thêm thuộc tính alt cho toàn bộ hình ảnh; và sử dụng cấu trúc HTML ngữ nghĩa để giúp công cụ tìm kiếm hiểu rõ nội dung website. Việc tối ưu hóa này góp phần giúp website đạt chất lượng tốt về hiệu suất, khả năng truy cập và SEO cơ bản.

4.3 Triển khai website lên host và server

- Trong quá trình triển khai sản phẩm, nhóm sử dụng dịch vụ hosting miễn phí Netlify để đưa website lên môi trường hoạt động thực tế. Đây là nền tảng hỗ trợ upload nhanh các dự án tĩnh (HTML, CSS, JavaScript), khả năng cập nhật tự động mỗi khi thay đổi mã nguồn, đồng thời cung cấp tốc độ tải cao nhờ hệ thống CDN toàn cầu. Ngoài Netlify, nhóm cũng tìm hiểu các hình thức hosting phổ biến như GitHub Pages, Vercel và triển khai trên server riêng, tuy nhiên Netlify đáp ứng tốt nhất cho yêu cầu dự án nên được lựa chọn.

- Quy trình triển khai được thực hiện đơn giản gồm các bước:
 - Chuẩn bị toàn bộ mã nguồn trong một thư mục chứa HTML, CSS, JavaScript và tài nguyên.
 - Truy cập trang Netlify, sử dụng chức năng “Drag & Drop” để upload toàn bộ thư mục dự án.
 - Netlify tự động build, gán tên miền miễn phí và tạo môi trường chạy thực tế ngay lập tức.
 - Kiểm tra lại cấu trúc, đường dẫn nội bộ và tối ưu một số file tĩnh để đảm bảo tốc độ tải.
- Sau khi triển khai, website được cung cấp tại địa chỉ: <https://quizzkit.netlify.app/>
- Nhóm đã tiến hành kiểm tra truy cập từ xa trên nhiều thiết bị và mạng khác nhau (Wifi, 4G), kết quả cho thấy website hoạt động ổn định, tốc độ tải nhanh và không gặp lỗi liên kết. Hệ thống hosting của Netlify đảm bảo uptime cao, phù hợp cho việc demo và sử dụng thử nghiệm.

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết quả đạt được

- Sau quá trình xây dựng và phát triển, nhóm đã hoàn thành đầy đủ các tiêu chí theo yêu cầu của rubric. Về mặt giao diện, website đạt tính thẩm mỹ, bố cục rõ ràng, có khả năng responsive tốt trên cả desktop, tablet và mobile. Các thành phần UI/UX được áp dụng đúng nguyên tắc, đảm bảo tính trực quan và dễ sử dụng cho người dùng. Mã nguồn được tổ chức theo cấu trúc rõ ràng, tách biệt giữa HTML, CSS, JavaScript và tài nguyên, đồng thời đặt tên và comment theo đúng chuẩn để dễ bảo trì.
- Về chức năng, hệ thống đăng ký – đăng nhập hoạt động đầy đủ, trang quiz vận hành ổn định với khả năng hiển thị câu hỏi, tính điểm và chuyển trang hợp lý. Tính năng điều hướng, hiệu ứng, sidebar, dark mode và các mục nội dung hoạt động mượt mà, không phát sinh lỗi trong quá trình kiểm thử. Website cũng được tối ưu hóa tốc độ tải bằng kỹ thuật nén ảnh, tinh gọn mã CSS/JS và kiểm tra bằng các công cụ PageSpeed cho kết quả tốt.

Thiết kế & giao diện – Responsive design	Hoàn thành
Tương thích trình duyệt	
Chức năng & tương tác	Hoàn thành 80%
Mã nguồn & cấu trúc	Hoàn thành
Tối ưu hóa hiệu suất	Đã kiểm thử PageSpeed
Triển khai dự án lên host và server	Netlify

- So với mục tiêu ban đầu, nhóm đã hoàn thành phần lớn các yêu cầu đề ra: xây dựng một website học tập dạng quiz có giao diện thân thiện, dễ sử dụng và đầy đủ chức năng cơ bản. Một số tính năng nâng cao như đồng bộ dữ liệu lên backend chưa được phát triển, tuy nhiên tổng thể dự án vẫn đạt mức hoàn thiện cao và đáp ứng tốt mục tiêu của đề tài.

5.2 Hạn chế

Mặc dù QuizzKit đã hoạt động ổn định, nhưng vẫn tồn tại một số hạn chế cần cải thiện để nâng cao trải nghiệm người dùng và mở rộng quy mô:

- Giao diện chưa thật sự tối ưu cho mọi kích thước màn hình. Dù trang đã hỗ trợ responsive, nhưng ở một số thiết bị có độ phân giải đặc biệt nhỏ hoặc lớn, bố cục đôi khi bị lệch hoặc không giữ được tỷ lệ mong muốn.

- Chưa hỗ trợ nhiều dạng câu hỏi nâng cao, chẳng hạn như kéo-thả, sắp xếp thứ tự, điền khuyết theo đoạn văn, hoặc quiz dạng mini-game như đúng-sai nhanh. Các loại câu hỏi hiện tại chủ yếu là trắc nghiệm (MCQ), đơn giản và chưa tạo nhiều trải nghiệm đa dạng.

- Chưa có tính năng thống kê nâng cao, ví dụ: phân tích điểm theo từng chủ đề, theo dõi tiến độ học tập, đề xuất quiz tiếp theo...

- Chưa hỗ trợ tính năng thi theo phòng/thi theo mã phòng như các nền tảng lớn, nên chưa phù hợp với sử dụng lớp học quy mô lớn hoặc thi nhóm theo thời gian thực.

- Nền tảng hiện chỉ hỗ trợ ngôn ngữ tiếng Việt, dẫn đến việc giới hạn đối tượng người dùng quốc tế hoặc những người cần nội dung đa ngôn ngữ, khiến việc tiếp cận với cộng đồng giáo dục toàn cầu chưa khả thi.

- Giao diện web chưa được tối ưu hóa hoàn hảo cho thiết bị di động, dẫn đến trải nghiệm kém hơn trên điện thoại hoặc máy tính bảng so với máy tính để bàn; chưa có ứng dụng mobile riêng biệt.

- Kho đề thi và tài liệu lý thuyết phụ thuộc lớn vào đóng góp cộng đồng, nên hiện tại còn hạn chế về số lượng và độ đa dạng (chủ yếu tập trung vào các môn học cơ bản ở Việt Nam), có thể dẫn đến tình trạng thiếu nội dung chuyên sâu hoặc cập nhật kịp thời.

- Tính năng bảo mật dữ liệu cá nhân (như tài khoản người dùng và đề thi cá nhân hóa) chưa được tích hợp các biện pháp nâng cao như xác thực hai yếu tố hoặc mã hóa dữ liệu đầy đủ, tiềm ẩn rủi ro khi lượng người dùng tăng

- Hệ thống chưa hỗ trợ tích hợp với các công cụ bên thứ ba (như Google Classroom hoặc Zoom), làm giảm khả năng linh hoạt trong môi trường học tập kết hợp.

5.3 Hướng phát triển

Để khắc phục các hạn chế và nâng tầm QuizKit thành một nền tảng giáo dục toàn diện hơn, tôi dự định tập trung vào một số hướng phát triển chính trong tương lai gần:

- Xây dựng hệ thống tài khoản và phân quyền (người dùng – giáo viên – quản trị viên). Việc này giúp bảo mật dữ liệu, lưu trữ lâu dài và cho phép mở rộng nhiều tính năng mới như lưu lịch sử thi, xem lại bài làm, hoặc chia sẻ đề dễ dàng.

- Phát triển API riêng, cho phép người khác tích hợp hệ thống QuizKit vào website hoặc phần mềm của họ.

- Mở rộng hỗ trợ đa ngôn ngữ (bắt đầu với tiếng Anh và tiếng Trung) để thu hút người dùng quốc tế, đồng thời tối ưu hóa giao diện responsive và phát triển ứng dụng di động (iOS và Android) nhằm cải thiện trải nghiệm trên mọi thiết bị, với tính năng như thông báo đẩy cho đề thi mới hoặc cập nhật xếp hạng.

- Tích hợp trí tuệ nhân tạo (AI) để tự động hóa việc tạo câu hỏi trắc nghiệm dựa trên tài liệu đầu vào, phân tích hiệu suất học tập của người dùng, và gợi ý đề thi cá nhân hóa – điều này sẽ giúp giảm thời gian chuẩn bị cho giáo viên và tăng tính tương tác.

- Mở rộng kho đề thi bằng cách khuyến khích cộng đồng qua hệ thống phần thưởng (như badge hoặc điểm tích lũy), đồng thời hợp tác với các tổ chức giáo dục để bổ sung nội dung chuyên sâu và cập nhật định kỳ.

- Nâng cao bảo mật bằng cách áp dụng các tiêu chuẩn như OAuth cho đăng nhập, mã hóa dữ liệu, và kiểm tra định kỳ; đồng thời tích hợp với các nền tảng bên thứ ba (ví dụ: API của Google Drive để lưu trữ đề thi hoặc Zoom cho lớp học trực tuyến).

- Dự kiến triển khai mô hình freemium với các tính năng cao cấp (như phân tích dữ liệu nâng cao hoặc không giới hạn lưu trữ) để đảm bảo tính bền vững tài chính, nhằm hỗ trợ phát triển lâu dài. Các hướng này sẽ được thực hiện theo giai đoạn, bắt đầu từ việc thu thập phản hồi người dùng để ưu tiên tính năng phù hợp nhất.

TÀI LIỆU THAM KHẢO

- [1]. <https://xanhcloud.com/html5-la-gi/>, “HTML5 là gì?, truy cập ngày 25 tháng 11 năm 2025”
- [2]. <https://vietnix.vn/css3-la-gi/>, “CSS3 là gì?, truy cập ngày 25 tháng 11 năm 2025”
- [3]. <https://itviec.com/blog/javascript-la-gi/>, “JavaScript là gì?, truy cập ngày 1 tháng 12 năm 2025”
- [4]. <https://topdev.vn/blog/html-la-gi/>, “HTML là gì?, truy cập ngày 10 tháng 12 năm 2025”
- [5]. <https://topdev.vn/blog/css-la-gi/>, “CSS là gì?, truy cập ngày 10 tháng 12 năm 2025”
- [6]. <https://topdev.vn/blog/javascript-la-gi/>, “JavaScript là gì?, truy cập ngày 10 tháng 12 năm 2025”
- [7]. <https://freetuts.net/html-la-gi-2943.html>, “HTML là gì? Kiến thức cơ bản, truy cập ngày 10 tháng 12 năm 2025”
- [8]. <https://freetuts.net/css-la-gi-2936.html>, “CSS là gì? Cách dùng CSS, truy cập ngày 10 tháng 12 năm 2025”
- [9]. <https://freetuts.net/javascript-la-gi-2956.html>, “JavaScript là gì? Hướng dẫn cơ bản, truy cập ngày 10 tháng 12 năm 2025”
- [10]. <https://vietnix.vn/html-la-gi/>, “Khái niệm HTML, truy cập ngày 10 tháng 12 năm 2025”
- [11]. <https://vietnix.vn/javascript-la-gi/>, “Giới thiệu JavaScript, truy cập ngày 10 tháng 12 năm 2025”
- [12]. <https://vietnix.vn/hosting-la-gi/>, “Hosting là gì?, truy cập ngày 10 tháng 12 năm 2025”
- [13]. <https://vietnix.vn/ui-ux-la-gi/>, “UI/UX là gì?, truy cập ngày 10 tháng 12 năm 2025”
- [14]. <https://gokisoft.com/hoc-html-co-ban.html>, “Học HTML cơ bản, truy cập ngày 10 tháng 12 năm 2025”

- [15]. <https://gokisoft.com/hoc-css-co-ban.html>, “Học CSS cơ bản, truy cập ngày 10 tháng 12 năm 2025”
- [16]. <https://gokisoft.com/hoc-javascript-co-ban.html>, “Học JavaScript cơ bản, truy cập ngày 10 tháng 12 năm 2025”
- [17]. <https://monamedia.co/html-la-gi/>, “Tìm hiểu về HTML, truy cập ngày 10 tháng 12 năm 2025”
- [18]. <https://monamedia.co/css-la-gi/>, “Tìm hiểu về CSS, truy cập ngày 10 tháng 12 năm 2025”
- [19]. <https://monamedia.co/javascript-la-gi/>, “Tìm hiểu về JavaScript, truy cập ngày 10 tháng 12 năm 2025”
- [20]. <https://vietiso.com/blog/ux-la-gi.html>, “UX là gì?, truy cập ngày 10 tháng 12 năm 2025”
- [21]. <https://vietiso.com/blog/ui-la-gi.html>, “UI là gì?, truy cập ngày 10 tháng 12 năm 2025”
- [22]. <https://tenten.vn/tin-tuc/domain-la-gi/>, “Tên miền (Domain) là gì?, truy cập ngày 10 tháng 12 năm 2025”
- [23]. <https://tenten.vn/tin-tuc/hosting-la-gi/>, “Hosting – Khái niệm và chức năng, truy cập ngày 10 tháng 12 năm 2025”

PHỤ LỤC

GitHub: https://github.com/huyggutgyt898/quiz_webDesign.git

Link: <https://quizzkit.netlify.app>

Điều hướng	Pass
Đăng nhập – đăng ký	Pass
Tạo quiz	Pass
Xem quiz	Pass

Link giao diện:

https://drive.google.com/drive/folders/1pvMczbGgKZRRRNhzDuK1JzWCxo0BF-Ts?usp=drive_link