

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



**NGUYỄN GIA KHANG**  
**PHẠM TUÂN HÙNG**  
**ĐOÀN NGUYỄN ANH KHOA**  
**ĐỖ MINH TÍN**  
**LÊ HOÀNG NHÂN**

**XÂY DỰNG PHÁT TRIỂN**  
**“WEBSITE TIN TỨC TỔNG HỢP”**

**BÁO CÁO ĐỒ ÁN MÔN HỌC**

Tp. Hồ Chí Minh, năm 2025

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



**NGUYỄN GIA KHANG**  
**PHẠM TUÂN HÙNG**  
**ĐOÀN NGUYỄN ANH KHOA**  
**ĐỖ MINH TÍN**  
**LÊ HOÀNG NHÂN**

**XÂY DỰNG PHÁT TRIỂN**  
**“WEBSITE TIN TỨC TỔNG HỢP”**

Lớp : CQ.65.CNTT

Khóa : 65

NGƯỜI HƯỚNG DẪN KHOA HỌC

1. HCD: ThS. Nguyễn Thiện Dương

Tp. Hồ Chí Minh, năm 2025

## LỜI CẢM ƠN

Lời nói đầu tiên, nhóm em xin gửi tới quý thầy cô bộ môn Công nghệ thông tin trường đại học Phân hiệu Giao thông vận tải tại Thành phố Hồ Chí Minh lời chúc sức khỏe và lời cảm ơn sâu sắc nhất.

Trước tiên, nhóm em xin cảm ơn nhà trường đã tạo điều kiện cơ sở vật chất tốt nhất để nhóm em có thể hoàn thành tốt bài báo cáo này.

Đặc biệt, nhóm em xin gửi lời cảm ơn chân thành thầy Nguyễn Thiện Dương đã nhiệt tình hướng dẫn, hỗ trợ và truyền đạt những kiến thức quý báu nhất trong quá trình thực hiện bài báo cáo này. Điều đó đã giúp cho nhóm em có thêm nhiều kiến thức bổ ích, trau dồi và rèn luyện thêm nhiều kỹ năng quan trọng. Nhờ vậy nhóm em có cái nhìn sâu sắc hơn về đề tài **“Xây dựng phát triển website tin tức tổng hợp”** nói riêng và môn học này nói chung.

Do những giới hạn về kiến thức, đồng thời bản thân nhóm em cũng thiếu kinh nghiệm trong việc thực hiện báo cáo vì vậy khó có thể tránh khỏi những thiếu sót. Nhóm em kính mong sự chỉ dẫn, nhận xét và đóng góp đến từ Cô để bài báo cáo này có thể hoàn thiện hơn nữa.

Lời cuối cùng, nhóm em xin kính chúc thầy cô bộ môn Công nghệ thông tin Trường đại học Phân hiệu Giao thông vận tải tại Thành phố Hồ Chí Minh và đặc biệt là thầy Nguyễn Thiện Dương sẽ luôn có thật nhiều sức khỏe, hạnh phúc và thành công hơn nữa.

Nhóm em xin chân thành cảm ơn!.

## NHẬN XÉT CỦA GIÁO VIÊN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

*Tp. Hồ Chí Minh, ngày ... tháng 12 năm 2025*

Giảng viên hướng dẫn

Ths. Nguyễn Thiện Dương

---

## NHIỆM VỤ THIẾT KẾ BÀI TẬP LỚN BỘ MÔN: CÔNG NGHỆ THÔNG TIN

---- \*\*\*\* ----

**Mã sinh viên:** 6551071058 **Họ tên SV:** Lê Hoàng Nhân

**Mã sinh viên:** 6551071040 **Họ tên SV:** Phạm Tuấn Hưng

**Mã sinh viên:** 6551071084 **Họ tên SV:** Đỗ Minh Tín

**Mã sinh viên:** 6551071042 **Họ tên SV:** Nguyễn Gia Khang

**Mã sinh viên:** 6551071048 **Họ tên SV:** Đoàn Nguyễn Anh Khoa

**Khóa:** 65

**Lớp:** Công nghệ thông tin

**1. Tên đề tài :** Xây dựng website tổng hợp tin tức trực tuyến "Báo điện tử 247"

**2. Mục đích thực hiện :** Đề tài được thực hiện nhằm xây dựng một hệ thống website cung cấp tin tức đa lĩnh vực, giúp người dùng cập nhật thông tin nhanh chóng qua giao diện trực quan và thân thiện. Thông qua đó, trang web tạo ra môi trường tương tác giữa độc giả và hệ thống, đồng thời là sản phẩm thực tiễn để áp dụng kiến thức lập trình Full-stack (Frontend và Backend).

**3. Mục tiêu thực hiện :** Đề tài hướng đến việc thiết kế và phát triển một website tin tức hoàn chỉnh với giao diện Responsive, tốc độ tải trang nhanh nhờ tối ưu hóa tài nguyên. Các chức năng trọng tâm bao gồm: hiển thị tin tức nổi bật (Vedette), phân loại tin theo danh mục, xem chi tiết bài viết, đăng ký/đăng nhập thành viên và quản lý hồ sơ cá nhân. Bên cạnh đó, dự án nhằm rèn luyện kỹ năng xây dựng API, quản trị cơ sở dữ liệu và triển khai ứng dụng lên máy chủ thực tế.

**4. Nội dung và phạm vi đề tài :** Đề tài tập trung nghiên cứu, phân tích và xây dựng hệ thống website tin tức hướng đến trải nghiệm người đọc (End-user).

- **Nội dung:** Khảo sát bố cục các trang báo lớn, thiết kế UI/UX hiện đại, xây dựng Frontend bằng HTML5/CSS3/JavaScript, xây dựng Backend bằng Node.js, kết nối cơ sở dữ liệu MySQL và kiểm thử hệ thống.
- **Phạm vi:** Tập trung vào các cơ chế hiển thị nội dung động, xử lý xác thực người dùng và lưu trữ dữ liệu. Không mở rộng sang các chức năng quản trị nội dung phức tạp (CMS dành cho Admin) hay hệ thống phân tích dữ liệu lớn.

**5. Phương pháp thực hiện :** Quá trình thực hiện áp dụng phương pháp phát triển phần mềm theo mô hình thác nước kết hợp linh hoạt: từ khảo sát, phân tích yêu cầu, thiết kế cơ sở dữ liệu, lập trình, kiểm thử đến triển khai.

- **Công nghệ Frontend:** Sử dụng HTML5, CSS3, JavaScript.
- **Công nghệ Backend:** Node.js.
- **Cơ sở dữ liệu:** MySQL (lưu trữ trên Aiven Cloud).
- **Công cụ hỗ trợ:** VS Code, Git/GitHub.

**6. Kết quả dự kiến :** Sản phẩm dự kiến là một website "Báo điện tử 247" hoàn chỉnh, đã được triển khai lên môi trường internet, hoạt động ổn định trên cả máy tính và điện thoại. Website cho phép người dùng xem tin tức, đăng ký tài khoản và cập nhật thông tin cá nhân thành công. Ngoài ra, đề tài giúp người thực hiện nắm vững kiến thức về luồng dữ liệu Client-Server và kỹ thuật xử lý bất đồng bộ trong lập trình web.

**7. Kế hoạch thực hiện :** Thời gian thực hiện: Từ ngày 2/12 đến ngày 10/12 năm 2025. Đề tài được triển khai theo các giai đoạn:

- Thu thập yêu cầu, phân tích cơ sở dữ liệu và dựng cấu trúc dự án.
- Thiết kế giao diện (Frontend) và cắt ghép HTML/CSS.
- Xây dựng API Backend và kết nối cơ sở dữ liệu.
- Lập trình chức năng và tích hợp Frontend - Backend.
- Kiểm tra thử trang web (Testing) và tối ưu hóa.
- Hoàn thiện báo cáo và triển khai lên Host. Mỗi giai đoạn được phân chia thời gian rõ ràng nhằm đảm bảo tiến độ và chất lượng sản phẩm.

#### **8. Giảng viên hướng dẫn**

- Họ tên: ThS. Nguyễn Thiện Dương
- Đơn vị: Phân hiệu Trường Đại học Giao thông Vận tải tại TP.HCM

**Đã giao nhiệm vụ**

**Giảng viên hướng dẫn**

## MỤC LỤC

<b>LỜI CẢM ƠN.....</b>	<b>I</b>
<b>NHẬN XÉT CỦA GIÁO VIÊN.....</b>	<b>II</b>
<b>MỤC LỤC.....</b>	<b>VI</b>
<b>DANH MỤC HÌNH ẢNH.....</b>	<b>XI</b>
<b>CHƯƠNG 1: MỞ ĐẦU.....</b>	<b>1</b>
1.1. Tổng quan về bài tập lớn .....	1
1.1.1. Bối cảnh và lý do chọn đề tài.....	1
1.1.2. Ý tưởng và lĩnh vực của website .....	1
1.1.2.1. Lĩnh vực hoạt động .....	1
1.1.2.2. Ý tưởng thiết kế chủ đạo .....	2
1.1.2.3. Định hướng trải nghiệm người dùng (UX) .....	2
1.1.3. Ý nghĩa thực tế và ứng dụng.....	3
1.2. Mục tiêu và nhiệm vụ .....	3
1.2.1. Mục tiêu tổng quát .....	3
1.2.2. Các nhiệm vụ chính cần thực hiện.....	3
1.2.3. Phân chia công việc .....	4
1.2.3.1. UI/UX & Frontend .....	4
1.2.3.2. Backend.....	4
1.2.3.3. Content Manager + Data Import .....	4
1.2.3.4. Tester + Document Writer .....	4
1.2.3.5. Project Manager (PM) – Điều phối.....	5
1.3. Phạm vi và giới hạn .....	5
1.3.1. Giới hạn về nội dung và kỹ thuật.....	5
1.3.2. Các phần sẽ phát triển thêm trong tương lai.....	5
1.4. Phân tích yêu cầu.....	5
1.4.1. Yêu cầu chức năng.....	5
1.4.2. Yêu cầu phi chức năng.....	6
1.5. Cấu trúc báo cáo .....	6



<b>CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ THỰC NGHIỆM .....</b>	<b>7</b>
2.1. Cơ sở lý thuyết .....	7
2.1.1. Kiến thức nền tảng HTML5, CSS3, JavaScript.....	7
2.1.2. Framework Bootstrap .....	8
2.1.3. UI/UX, responsive design, accessibility .....	9
2.1.4. Nguyên tắc phối màu, bố cục trong trang web .....	10
2.1.5. Hosting, domain, server .....	10
2.2. Khảo sát các sản phẩm tương tự.....	11
2.2.1. Phân tích website VnExpress.net.....	11
2.2.2. Phân tích website BaoMoi.com .....	12
2.2.3. Nhận xét và bài học rút ra.....	13
2.3. Tổng hợp và định hướng thiết kế .....	13
2.3.1. Tóm tắt các yếu tố áp dụng.....	13
2.3.2. Lý do chọn công cụ và phong cách .....	13
<b>CHƯƠNG 3: THIẾT KẾ VÀ TRIỂN KHAI WEBSITE .....</b>	<b>14</b>
3.1. Thiết kế giao diện (UI Design).....	14
3.1.1. Sơ đồ trang (site map), bố cục tổng thể. ....	14
3.1.1.1. Homepage .....	14
3.1.1.2. Header .....	14
3.1.1.3. Menu chuyên mục.....	15
3.1.1.4. Main Content.....	15
3.1.1.5. Footer .....	15
3.1.2. Wireframe hoặc mockup của các trang chính.....	16
3.1.2.1. Bố cục tổng thể của trang chủ.....	16
3.1.2.2. Khối quảng cáo đầu trang .....	16
3.1.2.3. Khu vực điều hướng (Navigation Block).....	16
3.1.2.4. Khối “Tin nổi bật trong ngày” .....	17
3.1.2.5. Khối “Tin mới nhất” .....	17
3.1.2.6. Khối “Xu hướng” .....	17

3.1.2.7. Thiết kế giao diện trang Quản trị (Admin Dashboard) .....	17
3.1.3. Bộ màu, font chữ, hình ảnh, biểu tượng. ....	19
3.1.3.1. Font chữ sử dụng.....	19
3.1.3.2. Màu sắc .....	20
3.1.3.3. Biểu tượng (Icons) .....	20
3.1.3.4. Phối hợp tổng thể .....	20
3.1.4. Nguyên tắc UX/UI được áp dụng. ....	20
3.1.4.1. Ưu tiên thông tin quan trọng.....	20
3.1.4.2. Nhóm chức năng theo khu vực (Grouping & Layout).....	21
3.1.4.3. Khoảng trắng hợp lý (Whitespace) .....	22
3.1.4.4. Dễ sử dụng và quen thuộc.....	22
3.1.5. Responsive trên desktop / mobile. ....	22
3.1.5.1. Phiên bản desktop .....	22
3.1.5.2. Phiên bản mobile.....	23
3.2. Chức năng và tương tác (UX Design) .....	24
3.2.1. Các chức năng chính của website. ....	24
3.2.2. Cách người dùng tương tác.....	25
3.2.2.1. Form và tương tác nhập liệu .....	25
3.2.2.2. Animation.....	25
3.2.2.3. UX tổng thể.....	26
3.3. Mã nguồn và cấu trúc dự án .....	26
3.3.1. Cấu trúc thư mục: .....	26
3.3.2. Công cụ và thư viện sử dụng. ....	28
3.3.2.1. Frontend .....	28
3.3.2.2. Backend.....	28
3.3.2.3. Phần mềm.....	28
3.3.3. Nguyên tắc đặt tên, tái sử dụng mã, comment code. ....	29
3.3.3.1. Nguyên tắc đặt tên.....	29
3.3.3.2. Nguyên tắc comment code .....	29

3.3.3.3. Nguyên tắc tái sử dụng mã.....	30
<b>CHƯƠNG 4: KIỂM THỬ, TỐI ƯU HÓA VÀ TRIỂN KHAI.....</b>	<b>32</b>
4.1. Kiểm thử giao diện và chức năng.....	32
4.1.1. Kiểm tra chức năng và tương tác người dùng .....	32
4.1.2. Kiểm thử cơ chế xác thực .....	34
4.1.3. Kiểm thử Validate dữ liệu người dùng .....	35
4.1.4. Kiểm thử responsive (Đa thiết bị) .....	36
4.1.5. Kiểm tra tương thích trình duyệt .....	36
4.2. Tối ưu hóa hiệu suất .....	37
4.2.1. Tối ưu hóa mã nguồn và tài nguyên (CSS/JS/Images).....	37
4.2.2. Đánh giá bằng Google PageSpeed.....	37
4.2.3. Đánh giá hiệu suất và SEO cơ bản .....	38
4.3. Triển khai website lên host và server .....	39
4.3.1. Giới thiệu giải pháp Hosting.....	39
4.3.2. Quy trình triển khai và cấu hình .....	40
4.3.3. Kết quả triển khai và đánh giá độ ổn định.....	41
<b>CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....</b>	<b>43</b>
5.1. Kết quả đạt được.....	43
5.2. Hạn chế.....	44
5.2.1. Về mặt kỹ thuật và mã nguồn: .....	44
5.2.2. Về giao diện và trải nghiệm người dùng (UI/UX): .....	44
5.2.3. Về tính năng:.....	45
5.3. Hướng phát triển.....	45
5.3.1. Xây dựng hệ thống quản trị (CMS) .....	45
5.3.2. Nâng cao trải nghiệm và tối ưu hóa.....	45
5.3.3. Mở rộng hệ sinh thái người dùng .....	45
<b>TÀI LIỆU THAM KHẢO .....</b>	<b>46</b>
<b>PHỤ LỤC .....</b>	<b>47</b>

## DANH MỤC VIẾT TẮT

STT	Mô tả	Ý nghĩa	Ghi chú
1	HTML	HyperText Markup Language	Ngôn ngữ xây dựng cấu trúc trang web
2	CSS	Cascading Style Sheets	Ngôn ngữ định dạng trang web
3	DOM	Document Object Model	Mô hình các đối tượng tài liệu
4	UI	User Interface	Giao diện người dùng
5	UX	User Experience	Trải nghiệm người dùng
6	SQL	Structured Query Language	Ngôn ngữ truy vấn có cấu trúc
7	SEO	Search Engine Optimization	Tối ưu hóa công cụ tìm kiếm
8	CMS	Content Management System	Hệ thống quản trị nội dung

## DANH MỤC HÌNH ẢNH

Hình 2.1: Thẻ ngữ nghĩa trong HTML5.....	7
Hình 2.2: CSS Grid .....	8
Hình 2.3. Responsive Design trong thiết kế web.....	10
Hình 2.4: Giao diện trang VnExpress .....	12
Hình 2.5: Giao diện trang BaoMoi.....	12
Hình 3.1: Sơ đồ trang (site map).....	16
Hình 3.2: Wireframe của Homepage.....	19
Hình 3.3: Nguyên tắc ưu tiên thông tin.....	21
Hình 3.4: Các nhóm chức năng theo khu vực .....	22
Hình 3.5: Responsive design trên desktop.....	23
Hình 3.6: Responsive design trên mobile. ....	24
Hình 3.7: Form tìm kiếm, đăng nhập .....	25
Hình 3.8: Hiệu ứng hover (Code) .....	26
Hình 3.9: Cấu trúc thư mục dự án.....	28
Hình 3.10: Nguyên tắc comment trong HTML.....	29
Hình 3.11: Nguyên tắc comment trong CSS .....	30
Hình 3.12: Nguyên tắc comment trong JavaScript .....	30
Hình 4.1. Quy trình đăng nhập.....	33
Hình 4.2. Cập nhật trạng thái người dùng.....	33
Hình 4.3. Giao diện quản lý thông tin tài khoản người dùng.....	34
Hình 4.4. Xác thực dữ liệu .....	36
Hình 4.5. Đánh giá bằng Google PageSpeed .....	37
Hình 4.6. Giao diện trang chủ hoàn thiện trên Desktop.....	42
Phụ lục 1.1. Giao diện trang.....	47
Phụ lục 1.2. Giao diện khi đọc chi tiết.....	48
Phụ lục 1.3. Footer của trang .....	48
Phụ lục 1.4. Navbar của trang .....	48
Phụ lục 1.5. Giao diện khi bấm chi tiết vào 1 chủ đề.....	48

Phụ lục 1.6. Khi bấm vào Hamburgur Menu .....	49
Phụ lục 1.7: Dashboard quản trị.....	50

## **DANH MỤC BẢNG BIỂU**

Bảng 3.1. Các công dụng của các file và thư mục trong dự án.....	26
Bảng 4.1. Vaildate dữ liệu đăng nhập người dùng .....	35
Bảng 4.2. Bảng tóm tắt hiệu suất .....	38
Bảng 5.1. Đánh giá mức độ hoàn thiện của nhóm .....	43

# CHƯƠNG 1: MỞ ĐẦU

**Tóm tắt chương:** Chương này cung cấp cái nhìn tổng quan về đề tài "Xây dựng và phát triển Website tin tức tổng hợp". Nội dung bao gồm việc phân tích bối cảnh thực tế dẫn đến quyết định chọn đề tài, xác định rõ mục tiêu cốt lõi là xây dựng trải nghiệm đọc tin tức tối giản và hiệu quả. Bên cạnh đó, chương cũng trình bày phạm vi nghiên cứu, các yêu cầu chức năng và phi chức năng cần đạt được, đồng thời phác thảo cấu trúc tổng thể của cuốn báo cáo đề án.

## 1.1. Tổng quan về bài tập lớn

### 1.1.1. Bối cảnh và lý do chọn đề tài

Trong kỷ nguyên số hóa hiện nay, thói quen tiếp cận thông tin của con người đã có sự chuyển dịch mạnh mẽ từ báo in truyền thống sang các nền tảng trực tuyến. Sự bùng nổ của thiết bị di động thông minh (smartphone) và mạng internet tốc độ cao đã biến việc đọc báo online trở thành một nhu cầu thiết yếu hàng ngày. Tuy nhiên, thực tế cho thấy người dùng đang phải đối mặt với một lượng thông tin khổng lồ. Nhiều trang báo điện tử hiện nay lạm dụng quảng cáo, giao diện rối rắm hoặc tốc độ tải trang chậm, gây ảnh hưởng tiêu cực đến trải nghiệm người dùng (UX).

Xuất phát từ nhu cầu thực tế về một không gian đọc tin tức "sạch", tập trung vào nội dung và tốc độ, nhóm thực hiện nhận thấy sự cần thiết của các mô hình trang tin tổng hợp với phong cách thiết kế tối giản (Minimalism). Đề tài "Xây dựng và phát triển Website tin tức tổng hợp" được lựa chọn không chỉ để giải quyết bài toán về trải nghiệm người dùng mà còn là cơ hội để nhóm áp dụng các kiến thức lập trình Frontend hiện đại vào một sản phẩm thực tế.

### 1.1.2. Ý tưởng và lĩnh vực của website

#### 1.1.2.1. Lĩnh vực hoạt động

Website được phát triển trong lĩnh vực truyền thông đa phương tiện và tin tức trực tuyến. Cụ thể hơn, dự án định vị mình là một nền tảng tổng hợp tin tức. Thay vì sản xuất nội dung gốc như các tòa soạn báo, website đóng vai trò là một "bộ lọc thông minh", tập hợp các tin tức nóng hổi từ nhiều lĩnh vực khác nhau về một



giao diện duy nhất, giúp người dùng tiết kiệm thời gian chuyển đổi giữa nhiều trang báo khác nhau.

#### 1.1.2.2. Ý tưởng thiết kế chủ đạo

Ý tưởng cốt lõi của dự án là xây dựng một không gian đọc tin trực quan, phá vỡ lối mòn của các trang báo truyền thống vốn nặng về văn bản.

- **Cảm hứng thiết kế:** Giao diện được lấy cảm hứng từ sự kết hợp giữa mô hình Pinterest (tập trung vào hình ảnh, bố cục so le) và Báo Mới (tập trung vào mật độ thông tin). Tuy nhiên, dự án sẽ tinh chỉnh để khắc phục nhược điểm thường thấy ở các trang tổng hợp tin tức hiện nay.
- **Bố cục:** Thay vì sử dụng danh sách liệt kê đơn điệu, website sử dụng hệ thống lưới thông minh kết hợp với giao diện dạng Thẻ - một thành phần cốt lõi của Bootstrap. Mỗi bài viết là một "thẻ" độc lập bao gồm ảnh thumbnail chất lượng cao, tiêu đề ngắn gọn và tóm tắt súc tích. Cách bố trí này không chỉ mang lại vẻ đẹp hiện đại, năng động mà còn giúp tối ưu hóa không gian hiển thị trên mọi thiết bị.

#### 1.1.2.3. Định hướng trải nghiệm người dùng (UX)

Website đặt mục tiêu tối thượng là "Content-First" (Nội dung là trên hết), loại bỏ mọi yếu tố gây nhiễu để người đọc tập trung hoàn toàn vào thông tin:

- **Tối ưu hóa thị giác:** Hình ảnh đóng vai trò dẫn dắt câu chuyện. Tỷ lệ hiển thị giữa hình ảnh và văn bản được cân chỉnh hài hòa, giúp người dùng nắm bắt nội dung chính chỉ qua một cái lướt mắt mà không cần đọc sâu ngay lập tức.
- **Sự tinh giản :** Website được thiết kế để "nhẹ" theo cả hai nghĩa:
  - Về giao diện: Sử dụng nhiều khoảng trắng (Whitespace), loại bỏ các banner quảng cáo , các widget không cần thiết và các đường viền rườm rà.
  - Về hiệu năng: Tối ưu mã nguồn HTML/CSS/JS để đảm bảo tốc độ tải trang tức thì, mang lại trải nghiệm mượt mà ngay cả khi kết nối mạng không ổn định.

- Điều hướng trực quan: Hệ thống phân loại tin tức được tổ chức khoa học, giúp người dùng dễ dàng lọc tin theo sở thích hoặc theo dòng sự kiện (Trending) mà không mất quá nhiều thao tác click chuột.

### 1.1.3. Ý nghĩa thực tế và ứng dụng

Về mặt học tập, đề tài này là môi trường lý tưởng để rèn luyện kỹ năng xây dựng giao diện người dùng (UI), làm chủ các ngôn ngữ nền tảng HTML5, CSS3, JavaScript và tư duy giải quyết vấn đề.

Về mặt ứng dụng thực tiễn, sản phẩm của đồ án có tính tùy biến cao. Mã nguồn này hoàn toàn có thể được phát triển mở rộng để trở thành cổng thông tin nội bộ cho các trường đại học, doanh nghiệp, hoặc các blog cá nhân chuyên nghiệp, đáp ứng nhu cầu chia sẻ thông tin nhanh chóng và thẩm mỹ.

## 1.2. Mục tiêu và nhiệm vụ

### 1.2.1. Mục tiêu tổng quát

Mục tiêu xuyên suốt của đồ án là xây dựng thành công một website tin tức hoàn chỉnh về mặt Frontend . Sản phẩm phải đạt được sự cân bằng giữa tính thẩm mỹ và hiệu năng .Website cần mang lại trải nghiệm đọc tốt nhất cho người dùng thông qua việc bố trí thông tin khoa học và khả năng tương tác cao.

### 1.2.2. Các nhiệm vụ chính cần thực hiện

Để đạt được mục tiêu trên, nhóm đã đề ra các nhiệm vụ cụ thể sau:

- Thiết kế giao diện và bố cục hợp lý: Nghiên cứu hành vi người dùng để thiết kế wireframe và mockup, ứng dụng hệ thống lưới (Grid System) của Bootstrap , đảm bảo tuân thủ các nguyên tắc thị giác.
- Phát triển các chức năng cơ bản: Lập trình các tính năng thiết yếu của một trang báo như hiển thị danh sách tin, xem chi tiết bài viết, phân loại tin theo chủ đề.
- Đảm bảo responsive và tối ưu hiệu suất: Ứng dụng kỹ thuật Responsive Design để website hiển thị tốt trên mọi kích thước màn hình (Desktop, Tablet, Mobile), đồng thời tối ưu mã nguồn để đạt tốc độ tải trang dưới 2 giây.

- Triển khai website lên host: Đưa sản phẩm lên môi trường internet thực tế để người dùng có thể truy cập trực tuyến.

### 1.2.3. Phân chia công việc

#### 1.2.3.1. UI/UX & Frontend

Nhiệm vụ:

- Thiết kế bố cục tổng thể (layout): header, menu, home, chi tiết bài viết.
- Thiết kế “style guide”: font, màu, spacing, nút bấm -> tạo chuẩn chung để cả nhóm theo.
- Làm giao diện trang chủ và 1–2 trang mẫu (template).
- Chịu trách nhiệm merge code frontend để đồng bộ UI.

#### 1.2.3.2. Backend

Nhiệm vụ:

- Thiết kế database các bảng: Users, Articles, Categories, Comments, Tags.
- Xây API chính: lấy tin, đăng tin, tìm kiếm, phân trang, login.
- Xử lý phân quyền tác giả – admin.
- Tối ưu query, bảo mật cơ bản.

#### 1.2.3.3. Content Manager + Data Import

Nhiệm vụ:

- Viết ~20–40 bài tin mẫu để có dữ liệu test.
- Chuẩn hóa nội dung: tiêu đề – mô tả – tag – ảnh đại diện.
- Kiểm tra chất lượng hiển thị nội dung trên giao diện.
- Cùng backend test upload ảnh, định dạng nội dung.

#### 1.2.3.4. Tester + Document Writer

Nhiệm vụ:

- Lập checklist test toàn bộ chức năng.
- Test liên tục trong lúc dev code.
- Ghi lỗi rõ ràng, gửi cho frontend/backend fix.
- Đặc tả yêu cầu
- Use case

- Sơ đồ hoạt động
- Hướng dẫn sử dụng web

#### 1.2.3.5. Project Manager (PM) – Điều phối

Nhiệm vụ:

- Chia deadline nhỏ theo tuần.
- Tổ chức họp, nhắc tiến độ, xử lý khi 2 bên frontend/backend vướng nhau.
- Giữ repo Git, quản lý branch & pull request.

### 1.3. Phạm vi và giới hạn

#### 1.3.1. Giới hạn về nội dung và kỹ thuật

Trong khuôn khổ đồ án môn học, dự án tập trung sâu vào mảng phát triển giao diện (Frontend Development).

- Công cụ sử dụng: Visual Studio Code, Git/GitHub
- Framework/Thư viện: Bootstrap (xây dựng giao diện Responsive)
- Dữ liệu: Dữ liệu tin tức (bài viết, hình ảnh) được sử dụng dưới dạng dữ liệu giả lập (Mock Data) lưu trữ cục bộ hoặc file JSON, chưa kết nối với cơ sở dữ liệu động thời gian thực.
- Thời gian: Dự án được thực hiện và hoàn thiện trong khoảng thời gian quy định của học kỳ.

#### 1.3.2. Các phần sẽ phát triển thêm trong tương lai

Để sản phẩm hoàn thiện hơn và có thể thương mại hóa, các hướng phát triển tiếp theo bao gồm:

- Xây dựng hệ thống Backend và database (SQL/NoSQL) để quản lý bài viết và người dùng.
- Tích hợp hệ thống quản trị nội dung (CMS) để soạn thảo và đăng bài tự động.
- Phát triển tính năng đăng nhập, bình luận và gợi ý tin tức theo sở thích người đọc.

### 1.4. Phân tích yêu cầu

#### 1.4.1. Yêu cầu chức năng

Hệ thống cần đáp ứng các chức năng cốt lõi sau:

- Trang chủ: Hiển thị tin nổi bật (Hero section) và dòng sự kiện mới nhất.
- Menu điều hướng (Navigation): Cho phép người dùng chuyển đổi giữa các chuyên mục (xã hội, thể thao, công nghệ...).
- Trang chi tiết: Hiển thị toàn bộ nội dung bài viết, hình ảnh minh họa, tác giả và ngày đăng.
- Hệ thống quản trị: Cho phép thêm, xóa và quản lý tài khoản

#### 1.4.2. Yêu cầu phi chức năng

- Tính thích ứng: Giao diện phải tự động thay đổi bố cục phù hợp khi truy cập bằng điện thoại hoặc máy tính.
- Thẩm mỹ: Sử dụng màu sắc hài hòa, font chữ dễ đọc, khoảng trắng hợp lý.
- Độ tương thích: Hoạt động ổn định trên các trình duyệt phổ biến

### 1.5. Cấu trúc báo cáo

Báo cáo được trình bày theo trình tự logic gồm 5 chương:

Chương 1: Mở đầu: Giới thiệu tổng quan, mục tiêu và phạm vi đề tài.

Chương 2: Cơ sở lý thuyết và thực nghiệm: Trình bày nền tảng công nghệ và khảo sát các sản phẩm tương tự.

Chương 3: Thiết kế và triển khai website : Chi tiết quá trình thiết kế, hiện thực hóa mã nguồn và các chức năng.

Chương 4: Kiểm thử, tối ưu hóa và triển khai : Kiểm tra hiệu suất và sự hoạt động của website

Chương 5: Kết luận: Đánh giá kết quả đạt được, hạn chế và hướng phát triển.

Phụ lục & Tài liệu tham khảo: Các hình ảnh bổ sung và nguồn tài liệu trích dẫn.

**Kết luận chương:** Chương 1 đã xác định rõ hướng đi của đề án là xây dựng một website tin tức tập trung vào trải nghiệm người dùng và giao diện tối giản. Các mục tiêu, nhiệm vụ và yêu cầu đã được thiết lập cụ thể làm kim chỉ nam cho quá trình thực hiện. Đây là cơ sở quan trọng để tiến hành nghiên cứu lý thuyết và công nghệ ở chương tiếp theo.

## CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ THỰC NGHIỆM

**Tóm tắt chương:** Chương này tập trung trình bày các nền tảng kiến thức và công nghệ cốt lõi được sử dụng trong đồ án, bao gồm HTML5, CSS3 và JavaScript. Bên cạnh đó, nội dung chương cũng đi sâu vào việc khảo sát các website tin tức hàng đầu hiện nay để rút ra bài học kinh nghiệm về UI/UX. Cuối cùng, chương đưa ra định hướng thiết kế và lựa chọn công nghệ cụ thể sẽ áp dụng cho dự án.

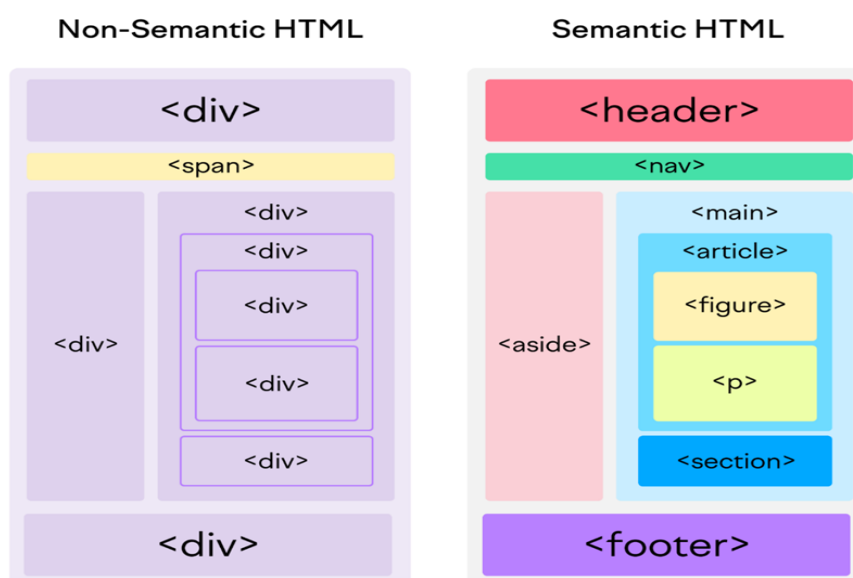
### 2.1. Cơ sở lý thuyết

#### 2.1.1. Kiến thức nền tảng HTML5, CSS3, JavaScript

Để xây dựng một website hiện đại, việc nắm vững bộ ba công nghệ Frontend là yêu cầu bắt buộc:

- **HTML (HyperText Markup Language):** Đóng vai trò là khung xương của trang web. Đồ án chú trọng sử dụng các thẻ ngữ nghĩa (Semantic Tags) như `<header>`, `<nav>`, `<section>`, `<article>` thay vì lạm dụng thẻ `<div>`. Việc này giúp cấu trúc mã nguồn rõ ràng, dễ bảo trì và hỗ trợ tốt cho SEO (Search Engine Optimization).

### What Is Semantic HTML?



Hình 2.1: Thẻ ngữ nghĩa trong HTML5

- **CSS (Cascading Style Sheets):** Là ngôn ngữ định kiểu giúp trang trí và bố cục trang web. Đồ án khai thác mạnh mẽ các module mới như Flexbox (để dàn trang một chiều, căn chỉnh menu) và CSS Grid (để xây dựng bố cục lưới) . Các hiệu ứng chuyển động (Transition/Animation) cũng được áp dụng để tăng tính tương tác.



Hình 2.2: CSS Grid

- **JavaScript (JS):** Ngôn ngữ lập trình tạo nên sự "sống động" cho website. JS được dùng để xử lý các sự kiện người dùng (click, scroll), thao tác với DOM (Document Object Model) để cập nhật nội dung mà không cần tải lại trang.

#### 2.1.2. Framework Bootstrap

Đây là công nghệ chủ đạo giúp tăng tốc độ phát triển và đảm bảo tính tương thích của đồ án.

Bootstrap là framework HTML, CSS và JS phổ biến nhất để phát triển các website chuẩn Responsive và Mobile-first.

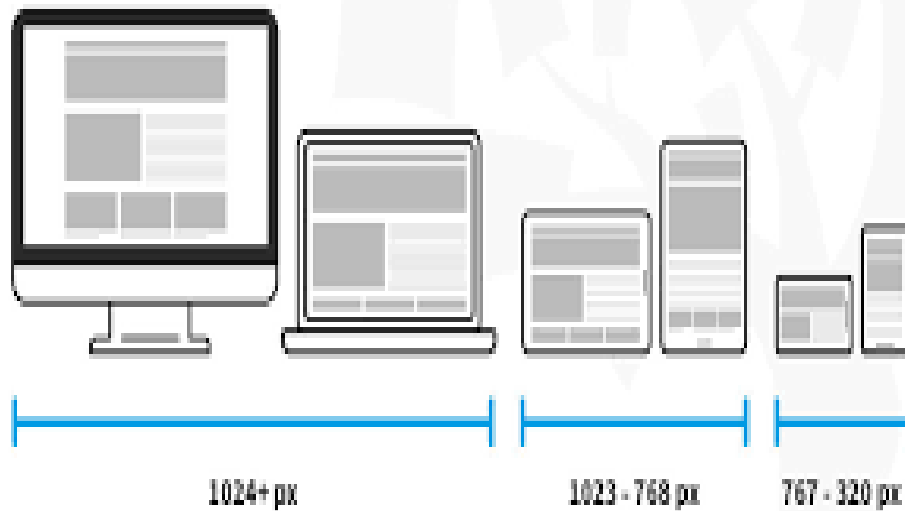
- Hệ thống lưới (Grid System): Bootstrap chia màn hình thành 12 cột. Đồ án tận dụng hệ thống này (với các class .col-, .row) để chia bố cục trang chủ linh hoạt: trên Desktop hiển thị 2 cột tin, trên Mobile tự động dồn thành 1 cột dọc.
- Các thành phần (Components): Đồ án sử dụng triệt để các component có sẵn:
  - Navbar: Thanh điều hướng tự động thu gọn thành nút hamburger trên thiết bị di động.
  - Card: Cấu trúc thẻ bài viết (Ảnh trên, nội dung dưới) rất phù hợp để hiển thị tin tức.
  - Carousel: Tạo slide ảnh cho phần tin nổi bật.

#### 2.1.3. UI/UX, responsive design, accessibility

- Tập trung vào tính thẩm mỹ trực quan. Nhóm tuân thủ nguyên tắc về sự nhất quán trong màu sắc và typography.
- UX (User Experience): Tập trung vào cảm giác của người dùng. Bố cục trang được thiết kế dựa trên thói quen đọc (F-pattern), đưa các thông tin quan trọng vào tầm mắt người đọc.
- Responsive Design: Đây là kỹ thuật thiết kế web thích ứng. Sử dụng Media Queries trong CSS để thay đổi giao diện dựa trên độ rộng màn hình thiết bị. Ví dụ: Chuyển từ menu ngang sang menu ẩn (Hamburger) trên thiết bị di động.



## Responsive Design



Hình 2.3. Responsive Design trong thiết kế web

### 2.1.4. Nguyên tắc phối màu, bố cục trong trang web

- **Màu sắc:** Áp dụng quy tắc 60-30-10. Màu chủ đạo là trắng/xám nhạt (nền) tạo sự thoáng đãng; màu phụ là đen/xám đậm (văn bản); màu nhấn là xanh dương (cho các liên kết, nút bấm).
- **Typography:** Sử dụng font Sans-serif như Roboto hoặc Arial, mang lại cảm giác hiện đại và dễ đọc trên màn hình.
- **Bố cục:** Sử dụng hệ thống lưới (Grid system) để căn chỉnh các thành phần thẳng hàng, giúp cho trang web có sự ngăn nắp và chuyên nghiệp.

### 2.1.5. Hosting, domain, server

Để triển khai website, nhóm sử dụng mô hình tách biệt giữa frontend và backend, phù hợp với dự án sử dụng Node.js và cơ sở dữ liệu từ dịch vụ bên ngoài.

- **Frontend (giao diện người dùng):** Nhóm triển khai trên GitHub Pages, một nền tảng hosting miễn phí dành cho website tĩnh. Toàn bộ mã nguồn HTML/CSS/JS được đưa lên GitHub và tự động public thành một website tĩnh.

- Backend (Server Node.js): Do GitHub Pages không hỗ trợ backend, nhóm triển khai file server.js lên dịch vụ Render.com. Render cho phép chạy ứng dụng Node.js, tạo API riêng và kết nối trực tiếp với cơ sở dữ liệu.
- Cơ sở dữ liệu (Database): Nhóm sử dụng Aiven để tạo và quản lý database. Render đóng vai trò kết nối với Aiven thông qua các biến môi trường (environment variables).

Quy trình triển khai gồm:

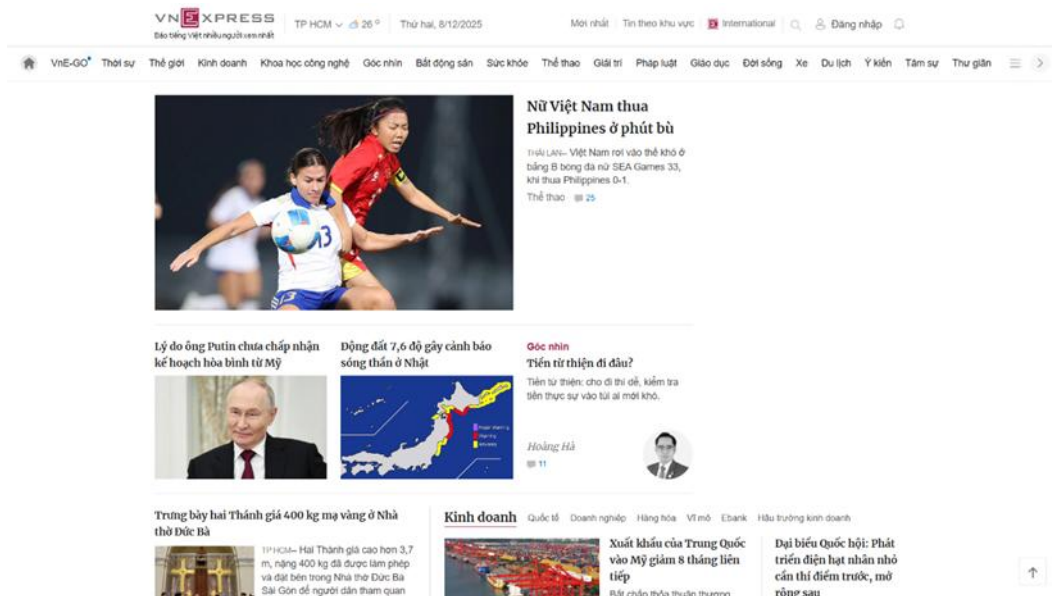
- Triển khai frontend lên GitHub Pages từ repo GitHub.
- Đưa backend (server.js và các file liên quan) lên Render để chạy API.
- Lấy thông tin kết nối từ Aiven và khai báo trong Render.
- Frontend gọi API từ backend thông qua URL Render.
- Cách triển khai này phù hợp cho các dự án web có backend, tách rõ frontend – backend, dễ quản lý và dễ mở rộng hơn các hình thức hosting truyền thống.

## **2.2. Khảo sát các sản phẩm tương tự**

### **2.2.1. Phân tích website VnExpress.net**

VnExpress là báo điện tử tiếng Việt có lượng truy cập hàng đầu.

- Giao diện: Mang phong cách báo chí truyền thống, mật độ thông tin dày đặc. Sử dụng font có chân (Serif) ở tiêu đề tạo cảm giác trang trọng.
- Ưu điểm: Phân cấp thông tin rõ ràng, tốc độ tải rất nhanh.
- Hạn chế: Giao diện trang chủ đôi khi gây cảm giác "ngợp" vì quá nhiều chữ.

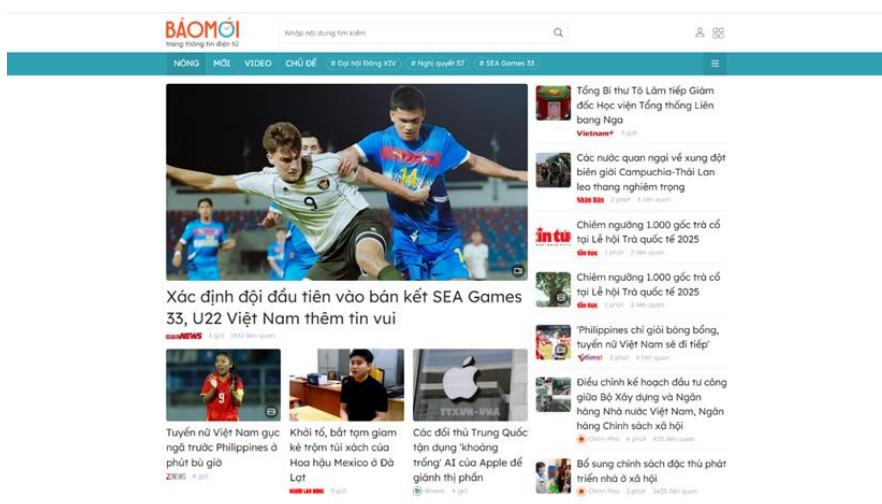


Hình 2.4: Giao diện trang VnExpress

### 2.2.2. Phân tích website BaoMoi.com

Báo Mới là trang tổng hợp tin tức tự động.

- Giao diện: Hiện đại, sử dụng phong cách Flat Design (Thiết kế phẳng). Các bài viết được trình bày dạng thẻ (Card) với hình ảnh thumbnail lớn.
- Ưu điểm: Giao diện thoáng, bắt mắt, dễ tương tác trên điện thoại.
- Hạn chế: Nhiều quảng cáo xen kẽ làm gián đoạn trải nghiệm đọc.



Hình 2.5: Giao diện trang BaoMoi

### 2.2.3. Nhận xét và bài học rút ra

Từ việc khảo sát hai website trên, nhóm rút ra các định hướng quan trọng:

- Nên học hỏi bố cục dạng lưới (Grid) của Báo Mới để tạo sự hiện đại.
- Cần kiểm soát mật độ thông tin, tránh nhồi nhét quá nhiều chữ như báo truyền thống.

## 2.3. Tổng hợp và định hướng thiết kế

### 2.3.1. Tóm tắt các yếu tố áp dụng

Dựa trên cơ sở lý thuyết và khảo sát thực tế, dự án sẽ được triển khai với các định hướng sau:

- Phong cách thiết kế: Tối giản, tập trung vào nội dung.
- Màu sắc: Tông màu sáng (Light mode) với màu xanh dương làm điểm nhấn thương hiệu.
- Công nghệ: HTML, CSS , JavaScript , SQL
- Công cụ phát triển: Visual Studio Code làm IDE chính.

### 2.3.2. Lý do chọn công cụ và phong cách

Việc lựa chọn phong cách tối giản giúp giảm tải nhận thức cho người dùng, giúp họ tập trung đọc tin lâu hơn. Việc sử dụng công nghệ cốt lõi giúp nhóm kiểm soát hoàn toàn mã nguồn, tối ưu hóa tốc độ tải trang) và quan trọng nhất là hiểu sâu sắc bản chất của lập trình web Frontend.

**Kết luận chương:** Chương 2 đã hệ thống hóa các kiến thức lý thuyết cần thiết và phân tích kỹ lưỡng các đối thủ cạnh tranh trên thị trường. Những đúc kết từ chương này, đặc biệt là việc lựa chọn công nghệ HTML/CSS/JS thuần và phong cách thiết kế tối giản, sẽ là nền tảng vững chắc để nhóm bước vào giai đoạn thiết kế chi tiết và thi công mã nguồn ở Chương 3.

## CHƯƠNG 3: THIẾT KẾ VÀ TRIỂN KHAI WEBSITE

**Tóm tắt chương:** Trong chương này, nhóm trình bày quá trình thiết kế và triển khai giao diện website, bao gồm thiết kế UI với sơ đồ trang, wireframe, bộ màu và font chữ phù hợp với nguyên tắc responsive trên nhiều thiết bị. Nhóm cũng mô tả các chức năng và tương tác người dùng (UX Design) thông qua menu, form, animation và các logic quan trọng. Cuối cùng, nhóm trình bày cấu trúc mã nguồn dự án với tổ chức thư mục rõ ràng, áp dụng các công cụ và thư viện cần thiết, đồng thời tuân thủ nguyên tắc đặt tên và comment code để đảm bảo tính dễ bảo trì.

### 3.1. Thiết kế giao diện (UI Design)

#### 3.1.1. Sơ đồ trang (site map), bố cục tổng thể.

Sơ đồ trang thể hiện cách tổ chức cấu trúc nội dung của website, được chia thành ba khu vực chính: Header, Menu chuyên mục, và Main Content, cuối trang là Footer. Cách phân cấp này giúp định hình rõ các thành phần giao diện, đồng thời đảm bảo người dùng có thể truy cập nhanh đến các mục cần thiết.

##### 3.1.1.1. Homepage

Trang chủ đóng vai trò trung tâm, nơi tập hợp toàn bộ thông tin và dẫn hướng đến các khu vực khác. Từ đây, website được tách xuống ba nhóm nội dung chính.

##### 3.1.1.2. Header

Khu vực đầu trang, bao gồm các thành phần hỗ trợ nhận diện thương hiệu và tiện ích người dùng:

- Logo: đánh dấu thương hiệu và cho phép quay về trang chủ.
- Thời tiết, ngày giờ: cung cấp thông tin nhanh theo thời gian thực.
- Menu phụ: gồm các liên kết hỗ trợ.
- Tìm kiếm: giúp người dùng tra cứu nội dung nhanh chóng.
- Đăng nhập: dành cho người dùng có tài khoản.

Header đóng vai trò điều hướng cơ bản và tăng trải nghiệm sử dụng.

#### 3.1.1.3. Menu chuyên mục

Đây là hệ thống điều hướng chính của website, phân loại nội dung thành từng nhóm rõ ràng:

- Thời sự,
- Thế giới
- Kinh doanh
- Khoa học
- Thể thao
- Giải trí
- Sức khỏe
- Giáo dục
- Du lịch
- Góc nhìn

Việc chia thành nhiều chuyên mục giúp người đọc dễ tìm đúng chủ đề mình quan tâm, đồng thời hỗ trợ cấu trúc dữ liệu hợp lý cho hệ thống.

#### 3.1.1.4. Main Content

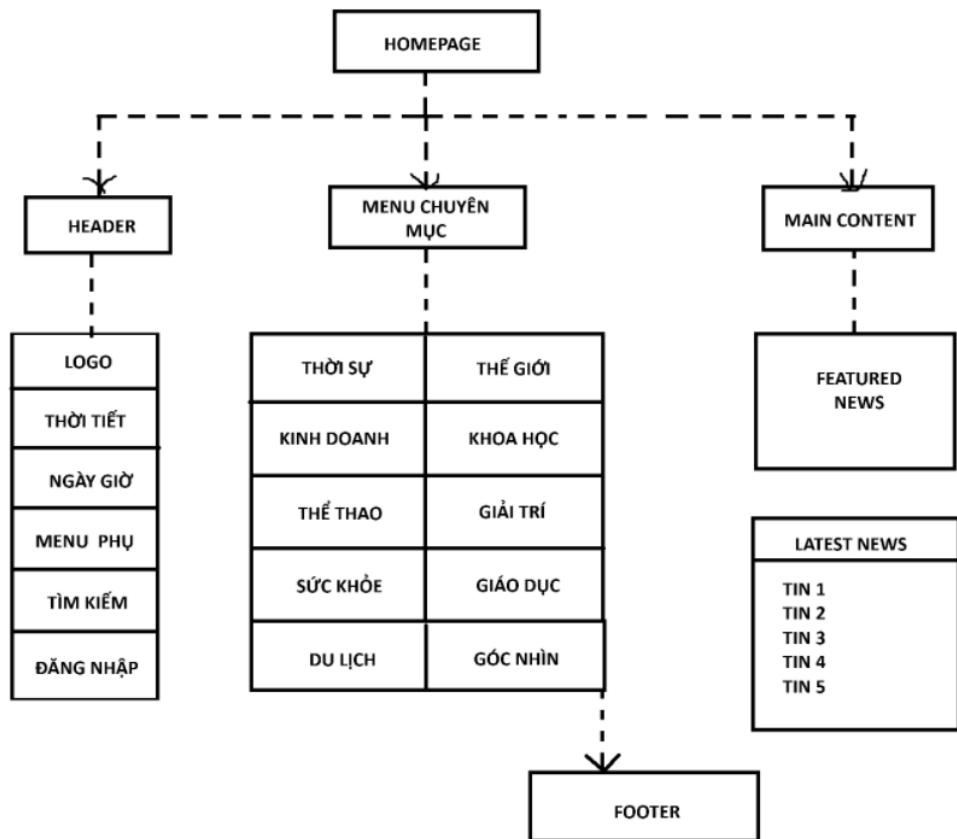
Khu vực hiển thị nội dung chính của trang:

- Featured News: tin nổi bật được ưu tiên hiển thị đầu tiên.
- Latest News: danh sách các tin mới nhất (Tin 1 → Tin 4).

Phần Main Content được thiết kế để thu hút sự chú ý và cung cấp thông tin cập nhật liên tục.

#### 3.1.1.5. Footer

Phần cuối trang chứa các thông tin bổ sung, liên kết phụ hoặc nội dung bản quyền. Footer giúp hoàn thiện luồng điều hướng và là điểm kết của trải nghiệm trang



Hình 3.1: Sơ đồ trang (site map)

### 3.1.2. Wireframe hoặc mockup của các trang chính

#### 3.1.2.1. Bố cục tổng thể của trang chủ

Wireframe của trang chủ sử dụng cấu trúc một cột chính, chia thành các khối nội dung liên tiếp từ trên xuống dưới. Các khối được bố trí theo dạng block lớn, giúp người dùng dễ theo dõi và đảm bảo rõ ràng về luồng thông tin.

#### 3.1.2.2. Khối quảng cáo đầu trang

- Một banner lớn màu đỏ nằm ở vị trí trên cùng.
- Dùng để hiển thị quảng cáo hoặc thông báo nổi bật.
- Chiếm toàn bộ chiều ngang trang nhằm thu hút sự chú ý ngay khi truy cập.

#### 3.1.2.3. Khu vực điều hướng (Navigation Block)

Sau banner là một thanh chứa: logo, ngày giờ, thời tiết, biểu tượng tìm kiếm, đăng nhập và biểu tượng menu.

Bên dưới là dải chuyên mục dạng hàng ngang gồm: thời sự, thể giới, kinh doanh, pháp luật, công nghệ, giải trí, thể thao, sức khỏe, giáo dục, đời sống, du lịch, bạn đọc...

Tất cả được thiết kế dạng các khối màu đơn giản để thể hiện rõ vị trí, không sử dụng màu UI chính thức.

#### 3.1.2.4. Khối “Tin nổi bật trong ngày”

- Bố cục chia làm 2 phần: ảnh lớn ở bên trái (chiếm ~60%) và khối tiêu đề + văn bản ở bên phải (chiếm ~40%).
- Bên dưới là 3 ảnh phụ dạng lưới 3 cột. Mỗi ảnh kèm 1 ô tiêu đề bên dưới.
- Khối này mô phỏng cách bài nổi bật được ưu tiên theo kích thước lớn.

#### 3.1.2.5. Khối “Tin mới nhất”

- Gồm các block tin được xếp theo dạng danh sách dọc.
- Mỗi bài viết thể hiện: Ảnh thu nhỏ ở trên, bên dưới là ô tiêu đề, cuối cùng là vùng văn bản mô tả
- Bố cục tập trung theo chiều dọc, phù hợp với luồng đọc liên tục.

#### 3.1.2.6. Khối “Xu hướng”

- Bố trí ở cột bên phải, song song với “Tin mới nhất”.
- Các bài được trình bày dạng block nhỏ: ảnh → tiêu đề → mô tả ngắn.
- Có kích thước gọn, tạo cảm giác giống sidebar của trang báo.

#### 3.1.2.7. Thiết kế giao diện trang Quản trị (Admin Dashboard)

Khác với giao diện người dùng tập trung vào trải nghiệm đọc, giao diện quản trị được thiết kế ưu tiên tính chức năng và hiệu quả quản lý dữ liệu.

**Bố cục tổng thể:** Trang Admin sử dụng bố cục **Grid Layout** chia màn hình thành 3 cột riêng biệt, giúp quản trị viên có cái nhìn tổng quan về toàn bộ hệ thống trên một màn hình duy nhất mà không cần chuyển trang nhiều lần.

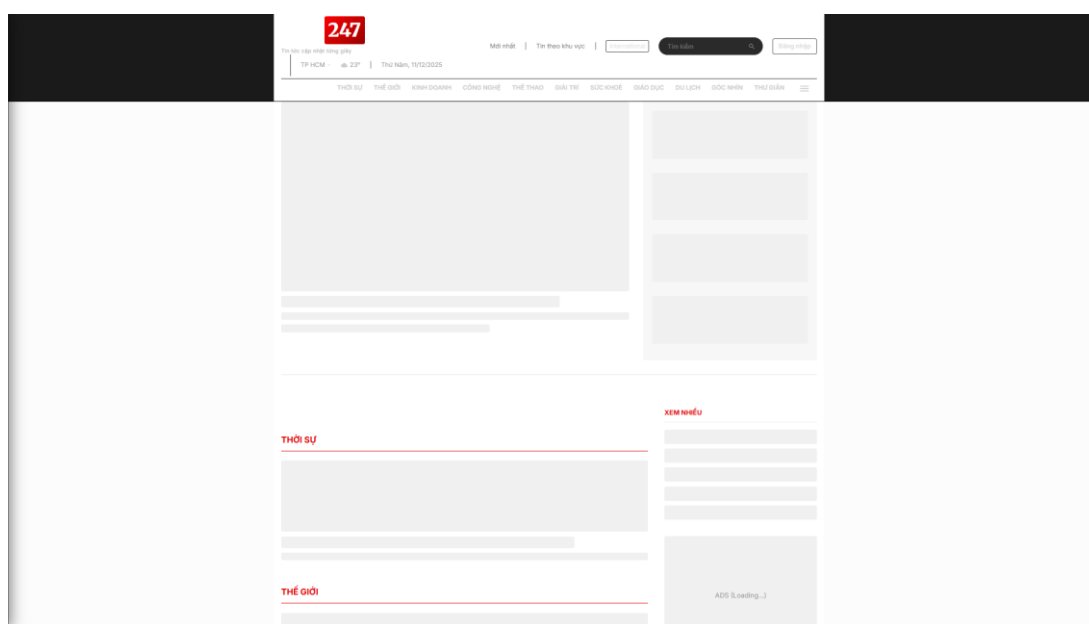
- Cột 1: Đăng bài mới (Post Editor)
  - Được bố trí bên trái, hoạt động như một form nhập liệu.
  - Các trường thông tin bao gồm: Tiêu đề, Tóm tắt (Sapo), Chuyên mục (Dropdown), Link ảnh và Nội dung chi tiết.



- Nút "Đăng bài ngay" sử dụng màu xanh lá (Success color) để tạo điểm nhấn hành động tích cực.
- Cột 2: Quản lý bài viết (Content Management)
  - Hiển thị danh sách tất cả các bài viết hiện có trong cơ sở dữ liệu.
  - Mỗi thẻ bài viết hiển thị vắn tắt: Tiêu đề, Chuyên mục và ID bài viết.
  - Nút "Xóa" màu đỏ (Danger color) được tích hợp để loại bỏ nhanh các nội dung không phù hợp.
- Cột 3: Quản lý tài khoản (User Management)
  - Hiển thị danh sách thành viên đã đăng ký.
  - Cung cấp các thông tin quan trọng như: Username, Email và ID định danh.
  - Hỗ trợ chức năng xóa tài khoản người dùng vi phạm.

### **Phong cách thiết kế**

- Màu sắc: Sử dụng nền xám nhạt (#f4f4f4) cho tổng thể để làm nổi bật các khối nội dung màu trắng (#fff). Các màu chức năng được sử dụng theo chuẩn UI: Xanh lá (Thêm mới/Lưu), Đỏ (Xóa/Cảnh báo), Xanh dương (Tiêu đề danh mục).
- Typography: Sử dụng font *Inter* đồng bộ với trang chủ để đảm bảo tính nhất quán, nhưng ưu tiên độ rõ ràng của dữ liệu.
- Tương tác: Các danh sách (List) có thanh cuộn riêng biệt (overflow-y: auto), giúp giao diện không bị vỡ khi dữ liệu tăng lên, giữ cố định chiều cao của Dashboard.



Hình 3.2: Wireframe của Homepage

### 3.1.3. Bộ màu, font chữ, hình ảnh, biểu tượng.

#### 3.1.3.1. Font chữ sử dụng

Trong giao diện của website, hệ thống typography được thiết kế dựa trên hai font chữ chính, được khai báo trong tệp style.css như sau:

- Inter (Sans-serif)
  - Vị trí sử dụng: trong các nội dung văn bản chung trên trang, header, logo, navigation, sidebar, badge, phần mô tả bài viết và các thành phần không có class title-serif
  - Kích thước phổ biến được dùng là: 14px – 16px cho phần nội dung, 12 – 14px cho mô tả và meta, menu và sidebar 14px – 16px
- Merriweather (Serif)
  - Vị trí sử dụng Merriweather: tiêu đề bài viết chính (Vedette), tiêu đề các bài trong card, tiêu đề bài trong phần Sub-Vedette và mọi phần được đánh dấu bằng class .title-serif
  - Kích thước chữ phổ biến: tiêu đề lớn (main article): 22px – 28px, tiêu đề card: 18px – 22px, tiêu đề nhỏ: 16px – 18px

#### 3.1.3.2. Màu sắc

- Màu đỏ đậm (#D32F2F – #C62828): dùng cho logo “247”, màu nhận diện thương hiệu, tạo điểm nhấn đầu trang.
- Màu đen, xám đậm (#212121 – #333333): dùng cho các tiêu đề bài viết lớn, text chính, tạo độ tương phản mạnh.
- Màu xám nhạt (#F1F1F1 – #E0E0E0): làm nền block sidebar hoặc ngăn cách các khu vực.
- Màu trắng (#FFFFFF): nền tổng thể của website, giúp bài viết nổi bật.

#### 3.1.3.3. Biểu tượng (Icons)

Hệ thống sử dụng icon dạng outline (nét mảnh):

- Icon Home
- Icon thời tiết (mây + nhiệt độ)
- Icon tìm kiếm (kính lúp)
- Icon menu (3 dấu gạch ngang)
- Icon đồng hồ nhỏ bên cạnh số lượt xem

#### 3.1.3.4. Phối hợp tổng thể

- Font serif + màu đen giúp bài viết “giàu chiều sâu”, dễ đọc như báo in.
- Font sans-serif + icon outline giúp phần điều hướng hiện đại, rõ ràng.
- Màu đỏ chỉ xuất hiện ở logo → làm điểm nhấn thị giác mà không gây rối.

Tác dụng: Làm cho giao diện vừa hiện đại, vừa giữ phong cách nghiêm túc của báo chí. UX tốt: hệ thống phân tầng rõ ràng, tiêu đề nổi bật, phần tin bên phải được tổ chức theo block dễ đọc.

#### 3.1.4. Nguyên tắc UX/UI được áp dụng.

Thiết kế giao diện của trang báo điện tử tuân theo nhiều nguyên tắc UX/UI nhằm tối ưu trải nghiệm đọc tin. Các nguyên tắc áp dụng gồm:

##### 3.1.4.1. Ưu tiên thông tin quan trọng

- Tin nổi bật được đặt trên cùng với ảnh lớn.
- Tiêu đề có kích thước lớn hơn văn bản mô tả.

- Các khối được sắp xếp theo thứ tự ưu tiên: Tin nổi bật → Tin mới nhất → Xu hướng.

## THỜI SỰ



### Hà Nội ô nhiễm không khí nghiêm trọng nhất từ đầu mùa đông

Sáng 11/12, nhiều trạm quan trắc tại Hà Nội ghi nhận chỉ số AQI trên 220, mức rất xấu; Hưng Yên và Thái Nguyên cũng xuất hiện điểm đo vượt 200.



Đèo cửa ngõ Đà Lạt mở lại sau một tháng đóng vì đối nút



Cán bộ sẽ bị xử lý nếu yêu cầu người dân nộp giấy tờ đã số hóa



Vùng áp thấp xuất hiện ở Nam Biển Đông

Hình 3.3: Nguyên tắc ưu tiên thông tin

Điều này giúp người dùng nắm thông tin chính ngay lập tức.

#### 3.1.4.2. Nhóm chức năng theo khu vực (Grouping & Layout)

- Header tập trung logo, ngày giờ, thời tiết, thanh tìm kiếm và menu.
- Các chuyên mục được nhóm thành một thanh điều hướng duy nhất.
- Tin cùng loại được đặt chung vào từng block để quan sát.

Hình 3.4: Các nhóm chức năng theo khu vực

#### 3.1.4.3. Khoảng trắng hợp lý (Whitespace)

- Các khối nội dung giữ khoảng cách đều, tạo cảm giác thoáng.
- Giúp mắt người đọc không bị rối khi theo dõi nhiều bài viết liên tiếp.

#### 3.1.4.4. Dễ sử dụng và quen thuộc

- Cách bố trí giống các trang báo điện tử phổ biến: VnExpress, Tuổi Trẻ, 24h...
- Thanh tìm kiếm, đăng nhập và menu nằm ở góc dễ thấy.
- Người dùng tốn rất ít thời gian để làm quen.

#### 3.1.5. Responsive trên desktop / mobile.

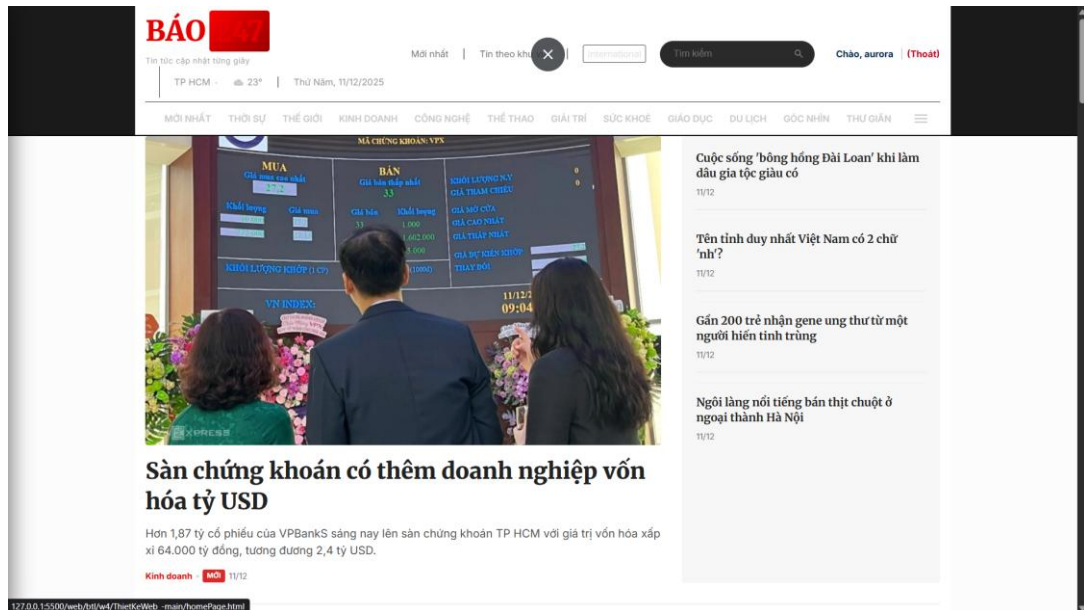
Thiết kế của trang web được xây dựng theo hướng responsive, đảm bảo nội dung hiển thị tối ưu trên cả hai thiết bị phổ biến: desktop và mobile. Giao diện không chỉ thay đổi kích thước mà còn tái sắp xếp bố cục, giúp người dùng sử dụng mượt mà trong mọi hoàn cảnh.

##### 3.1.5.1. Phiên bản desktop

Trên desktop, giao diện tận dụng tối đa chiều rộng màn hình:

- Bố cục 2 cột rõ ràng:
  - Cột trái dành cho bài viết chính, khá rộng, hỗ trợ đọc dễ chịu.
  - Cột phải là danh sách tin nổi bật, hiển thị dạng thẻ nhỏ, người đọc dễ lướt nhanh.
- Menu điều hướng trải ngang phía trên, hiển thị đầy đủ các mục (Thời sự, thể giới, kinh doanh...).
- Hình ảnh bài viết lớn dùng độ phân giải cao, tạo điểm nhấn đầu trang.
- Các thành phần phụ như thời tiết, tìm kiếm, nút menu đều nằm ở vùng header và luôn nhìn thấy.

→ Trải nghiệm desktop tập trung vào việc đọc nội dung và theo dõi nhiều tin tức cùng lúc.



Hình 3.5: Responsive design trên desktop.

#### 3.1.5.2. Phiên bản mobile

- Bố cục chuyển sang dạng một cột dọc, phù hợp hành vi lướt bằng tay.
- Menu điều hướng thu gọn thành nút hamburger, logo đặt ở giữa, các icon tài khoản và tìm kiếm nằm bên phải.
- Các chuyên mục hiển thị dưới dạng thanh trượt ngang (horizontal scroll), giúp tiết kiệm không gian nhưng vẫn đầy đủ chức năng.
- Bài viết nổi bật chiếm toàn bộ chiều ngang, phía dưới là tiêu đề và mô tả.
- Các bài viết phụ được bố trí dạng 2 cột nhỏ gọn, mỗi bài gồm ảnh thumbnail và tiêu đề ngắn.
- Khoảng cách giữa các phần tử được tăng hợp lý để tránh thao tác nhầm khi chạm.



Hình 3.6: Responsive design trên mobile.

## 3.2. Chức năng và tương tác (UX Design)

### 3.2.1. Các chức năng chính của website.

- **Trang chủ nổi bật:** Hiển thị tin nóng, tin mới nhất và các bài viết được quan tâm nhiều.
- **Thanh điều hướng chuyên mục:** Người dùng có thể nhanh chóng chuyển đổi giữa các lĩnh vực tin tức.
- **Quảng cáo & dịch vụ đi kèm:** Hiển thị banner quảng cáo, liên kết dịch vụ liên quan để tạo nguồn thu

Hệ thống quản trị nội dung (CMS đơn giản):

- Đăng bài (Create): Cho phép Admin gửi dữ liệu bài viết thông qua API POST /api/add-article. Hệ thống tự động lưu vào MySQL.
- Hiển thị danh sách (Read): Sử dụng kỹ thuật Fetch API để lấy dữ liệu JSON từ server và render (vẽ) lại giao diện danh sách theo thời gian thực mỗi khi có thay đổi.
- Xóa dữ liệu (Delete): Tích hợp cơ chế xác thực hành động (confirm dialog) trước khi gửi lệnh xóa bài viết hoặc xóa người dùng xuống Database, đảm bảo an toàn dữ liệu.

### 3.2.2. Cách người dùng tương tác

#### 3.2.2.1. Form và tương tác nhập liệu

- Form tìm kiếm: người dùng nhập từ khóa để tìm bài viết.
- Form đăng nhập/đăng ký: gồm các trường cơ bản (email, mật khẩu), hỗ trợ đăng nhập nhanh qua mạng xã hội.

```
<div class="modal-overlay" id="loginModal">
  <div class="login-box">
    <div class="close-btn" onclick="closeLogin()" style="float:right; cursor:pointer; font-size:24px;">✕</div>
    <div class="auth-tabs" style="display:flex; margin-bottom:20px; border-bottom:1px solid #ddd;">
      <div class="auth-tab active" id="tab-login" onclick="switchTab('login')" style="flex:1; text-align:center; padding:10px">
        Đăng nhập
      </div>
      <div class="auth-tab" id="tab-register" onclick="switchTab('register')" style="flex:1; text-align:center; padding:10px">
        Đăng ký
      </div>
    </div>
    <form id="form-login">
      <h3 style="margin-bottom:15px; text-align:center;">Chào mừng trở lại!</h3>
      <div class="form-group"><input type="text" id="login-input" class="form-input" placeholder="Email hoặc tên đăng nhập" required></div>
      <div class="form-group"><input type="password" id="login-pass" class="form-input" placeholder="Mật khẩu" required></div>
      <button type="button" class="btn-login" onclick="handleLogin()">ĐĂNG NHẬP</button>
    </form>
    <form id="form-register" style="display:none;">
      <h3 style="margin-bottom:15px; text-align:center;">Tạo tài khoản mới!</h3>
      <div class="form-group"><input type="text" id="reg-username" class="form-input" placeholder="Tên đăng nhập" required></div>
      <div class="form-group"><input type="email" id="reg-email" class="form-input" placeholder="Email" required></div>
      <div class="form-group"><input type="password" id="reg-pass" class="form-input" placeholder="Mật khẩu" required></div>
      <button type="button" class="btn-register" onclick="handleRegister()" style="background:#04284d; color:white;">ĐĂNG KÝ</button>
    </form>
  </div>
</div>
```

Hình 3.7: Form tìm kiếm, đăng nhập

#### 3.2.2.2. Animation

- Hover effect: khi rê chuột vào tiêu đề hoặc ảnh bài viết, hiển thị hiệu ứng đổi màu hoặc phóng to nhẹ để thu hút.
- Transition mượt: menu mở/đóng trên mobile có hiệu ứng trượt (slide-in/slide-out).
- Lazy loading: ảnh và nội dung tin tải dần khi người dùng cuộn xuống, giúp tăng tốc độ hiển thị.



```

.nav-link:hover {
  color: var(--primary-red);
  background-color: #2a2a2a;
}

.article-card:hover .title-serif {
  color: var(--primary-red);
}

.article-card:hover .thumb-wrapper img {
  transform: scale(1.05);
}

```

Hình 3.8: Hiệu ứng hover (Code)

### 3.2.2.3. UX tổng thể

- Các tương tác được thiết kế đơn giản, trực quan, giảm thao tác thừa.
- Hiệu ứng được sử dụng vừa phải, tránh gây rối mắt nhưng vẫn tạo cảm giác hiện đại.
- Người dùng dễ dàng điều hướng – nhập liệu – nhận phản hồi, đảm bảo trải nghiệm đọc báo nhanh và tiện lợi.

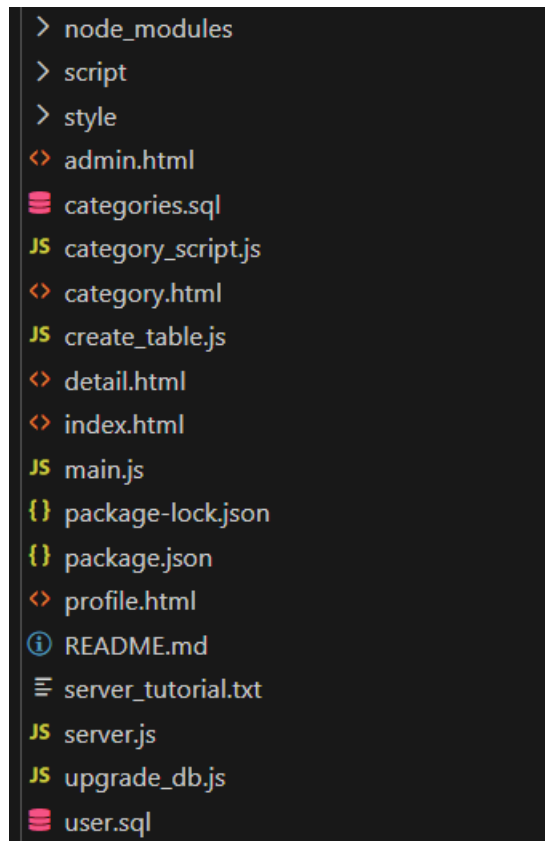
## 3.3. Mã nguồn và cấu trúc dự án

### 3.3.1. Cấu trúc thư mục:

Bảng 3.1. Các công dụng của các file và thư mục trong dự án

File/Folder	Mô tả
<b>node_modules/</b>	Chứa các thư viện Node.js được cài qua npm (không commit lên Git)
<b>script/</b>	Chứa các file JavaScript riêng biệt theo từng trang
<b>style/</b>	Chứa các file CSS riêng biệt theo từng trang

<b>.env</b>	Lưu trữ biến môi trường bảo mật (API key, database URL)
<b>homePage.html</b>	Trang chủ chính của website tin tức
<b>detail.html</b>	Trang hiển thị chi tiết một bài báo
<b>main.js</b>	File JavaScript toàn cục, xử lý logic chung
<b>server.js</b>	Server backend (Express.js hoặc tương tự)
<b>package.json</b>	Quản lý dependencies, scripts chạy dự án
<b>README.md</b>	Tài liệu hướng dẫn cài đặt và chạy dự án
<b>server_tutorial.txt</b>	File văn bản hướng dẫn sử dụng server).
<b>user.sql</b>	File script SQL chứa dữ liệu người dùng mẫu.
<b>categories.sql</b>	File script SQL chứa dữ liệu các chuyên mục.



Hình 3.9: Cấu trúc thư mục dự án

### 3.3.2. Công cụ và thư viện sử dụng.

#### 3.3.2.1. Frontend

- HTML5: Cấu trúc trang web
- CSS3: Thiết kế giao diện (Flexbox, Grid, Animation)
- JavaScript: Xử lý tương tác người dùng

#### 3.3.2.2. Backend

- Node.js: Môi trường chạy JavaScript phía server
- Express.js: Framework web
- SQL: Quản lý dữ liệu

#### 3.3.2.3. Phần mềm

Trình soạn thảo mã nguồn (Code Editor) Visual Studio Code

- Nhẹ, khởi động nhanh
- Hỗ trợ nhiều ngôn ngữ qua extensions
- Tùy biến cao (cài plugin theo nhu cầu)

- Không tích hợp sẵn compiler/debugger phức tạp

### 3.3.3. Nguyên tắc đặt tên, tái sử dụng mã, comment code.

#### 3.3.3.1. Nguyên tắc đặt tên

- Quy ước: Nhóm xác định đặt tên theo các quy ước snake\_case
  - homepage\_script.js
  - homepage\_style.css
- Tên mô tả rõ chức năng: Homepage.html, server.js, ...

#### 3.3.3.2. Nguyên tắc comment code

- **HTML**
  - Dùng `<!-- ===== -->` cho header section lớn
  - Comment ngắn gọn trên 1 dòng cho các element quan trọng
  - Ghi chú TODO/FIXME khi cần hoàn thiện sau

```
/*=====*/
/* CSS riêng cho tiêu đề chuyên mục */
/*=====*/
.cat-heading {
  border-bottom: 2px solid #e60000;
  padding-bottom: 15px;
  margin-bottom: 30px;
  margin-top: 30px;
  display: flex;
  justify-content: space-between;
  align-items: flex-end;
}
```

Hình 3.10: Nguyên tắc comment trong HTML

- **CSS**
  - Chia thành các section rõ ràng bằng comment header lớn
  - Giải thích các giá trị "ma thuật" (magic numbers)
  - Ghi chú lý do sử dụng workaround
  - Comment cho các hack CSS đặc biệt

```
/* Layout chia 3 cột: Đăng bài | Quản lý Bài viết | Quản lý User */
.container {
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
  gap: 20px;
}
```

Hình 3.11: Nguyên tắc comment trong CSS

- **JAVASCRIPT**

- Sử dụng JSDoc comments cho functions quan trọng
- Comment giải thích logic phức tạp
- Ghi chú TODO, FIXME, NOTE rõ ràng
- Tách thành sections: Variables → Functions → Event Listeners → Init

```
// --- HÀM 2: VẼ TỪNG CHUYÊN MỤC RIÊNG BIỆT ---
function renderCategorySection(allArticles, catId, displayCount = 4) {
  const section = document.getElementById(catId);
  if (!section) return;
```

Hình 3.12: Nguyên tắc comment trong JavaScript

### 3.3.3.3. Nguyên tắc tái sử dụng mã

Khi xây dựng web, nhóm thực hiện các quy tắc sau:

- Nguyên tắc DRY (Don't Repeat Yourself): Không lặp lại code - mỗi logic chỉ viết một lần, sau đó tái sử dụng nhiều nơi.
- CSS Variables (Biến CSS): Tạo biến --primary-red: #e60000 ở root, các vị trí khác gọi biến → đổi màu chỉ sửa một chỗ.
- Utility Classes: Tạo các class nhỏ:
  - .flex - bật flexbox
  - .flex-center - căn giữa
  - .text-red - chữ màu đỏ
  - .text-bold - in đậm

Khi cần dùng, thêm class vào HTML thay vì viết CSS riêng.

**Kết luận chương:** Qua chương 3, nhóm đã hoàn thành việc thiết kế giao diện người dùng với các wireframe, mockup và bộ nhận diện thương hiệu phù hợp, đồng thời xây dựng được cấu trúc mã nguồn có tổ chức và dễ bảo trì. Các chức năng và tương tác người dùng đã được triển khai theo nguyên tắc UX/UI. Với giao diện và chức năng cốt lõi đã được xây dựng, bước tiếp theo là đảm bảo chất lượng sản phẩm thông qua quy trình kiểm thử toàn diện, tối ưu hóa hiệu suất hệ thống, và triển khai website lên môi trường production để đưa vào vận hành thực tế.

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

**Tóm tắt chương:** Sau giai đoạn phân tích và thiết kế hệ thống, nhóm thực hiện đã tiến hành hiện thực hóa mã nguồn, xây dựng giao diện và các chức năng cốt lõi cho website. Chương này sẽ trình bày chi tiết về kết quả xây dựng giao diện, quy trình kiểm thử chức năng, các biện pháp tối ưu hóa hiệu suất và phương án triển khai hệ thống lên môi trường thực tế.

### 4.1. Kiểm thử giao diện và chức năng

Quy trình kiểm thử được thực hiện theo phương pháp kiểm thử hộp đen (Black-box testing), tập trung xác minh tính đúng đắn của các chức năng nghiệp vụ và trải nghiệm người dùng (UI/UX) dựa trên các yêu cầu đã đặt ra.

#### 4.1.1. Kiểm tra chức năng và tương tác người dùng

Nhóm đã tiến hành kiểm tra chi tiết từng thành phần chức năng trên website, đảm bảo các tương tác như liên kết và hiệu ứng hoạt động mượt mà.

- Chức năng xác thực (đăng ký/đăng nhập) Hệ thống sử dụng cơ chế Modal Popup (cửa sổ nổi) giúp người dùng thực hiện thao tác mà không cần chuyển trang.

Các kịch bản kiểm thử bao gồm:

- Kiểm tra tính hợp lệ (Validation): Tại form đăng ký, hệ thống tự động so sánh mật khẩu và xác nhận mật khẩu. Nếu không trùng khớp, thông báo lỗi sẽ hiển thị ngay lập tức và ngăn chặn gửi dữ liệu về Server.
- Luồng xử lý dữ liệu: Khi người dùng nhập đúng thông tin, hệ thống thực hiện gọi API, đóng Modal và cập nhật trạng thái header từ "Khách" sang "Thành viên" (hiển thị avatar và tên).

Đăng nhập

Đăng ký

X

Chào mừng trở lại!

ĐĂNG NHẬP

Hình 4.1. Quy trình đăng nhập



Hình 4.2. Cập nhật trạng thái người dùng

- Chức năng quản lý hồ sơ (User Profile) Trang hồ sơ cá nhân (profile.html) được kiểm thử chặt chẽ về định dạng dữ liệu đầu vào:
  - Email: Hệ thống sử dụng biểu thức chính quy (Regex) để bắt buộc email phải đúng định dạng chuẩn (ví dụ: email@domain.com).
  - Số điện thoại: Chỉ chấp nhận ký tự số và độ dài bắt buộc là 10 chữ số.
  - Cơ chế lưu trữ: Kiểm thử khả năng lưu trữ cục bộ (localStorage), đảm bảo thông tin người dùng (họ tên, email) không bị mất đi khi tải lại trang.



Avatar

Tải ảnh từ máy tính

Tên đăng nhập

test

Email

undefined

Thay đổi

Ngày sinh

Chưa có thông tin

Thay đổi

Giới tính

Chưa có thông tin

Thay đổi

Số điện thoại

Chưa có thông tin

Thay đổi

Địa chỉ

Chưa có thông tin

Thay đổi

Cấp bậc

Thành viên mới

Xóa tài khoản

Hành động này không thể hoàn tác.

Xóa

Hình 4.3. Giao diện quản lý thông tin tài khoản người dùng.

#### 4.1.2. Kiểm thử cơ chế xác thực

Hệ thống sử dụng cơ chế Modal Popup cho việc Đăng ký/Đăng nhập thay vì chuyển trang, giúp giữ chân người dùng ở lại bài viết đang đọc. Logic xử lý được viết bằng JavaScript, kết nối với API Backend (mô phỏng) thông qua hàm fetch.

- Kịch bản kiểm thử 1: Đăng ký tài khoản
  - Mục tiêu: Đảm bảo hệ thống chặn các dữ liệu đầu vào không hợp lệ.
  - Thao tác: Nhập mật khẩu và xác nhận mật khẩu không trùng khớp.

- Kết quả thực tế: Hệ thống hiển thị thông báo lỗi “Mật khẩu nhập lại không khớp!” và ngăn chặn việc gửi dữ liệu đi.
- Kịch bản kiểm thử 2: Đăng nhập hệ thống
  - Mục tiêu: Xác minh luồng đăng nhập và cập nhật giao diện.
  - Thao tác: Nhập đúng Email và Mật khẩu đã đăng ký.
  - Kết quả thực tế: Modal đóng lại, nút "Đăng nhập" trên Header biến mất, thay thế bằng Avatar và tên người dùng lấy từ cơ sở dữ liệu.

#### 4.1.3. Kiểm thử Validate dữ liệu người dùng

Tại trang quản lý hồ sơ, các hàm kiểm tra tính hợp lệ (Validation) được cài đặt chặt chẽ bằng biểu thức chính quy (Regex) ngay tại phía Client.

Bảng 4.1. Validate dữ liệu đăng nhập người dùng

STT	Trường dữ liệu	Dữ liệu kiểm thử	Kết quả mong đợi	Kết quả thực tế
1	Email	nguyenvana (thiếu @)	Báo lỗi định dạng email	Hiển thị alert: "Địa chỉ Email không hợp lệ"
2	Họ và tên	Rỗng	Yêu cầu nhập tên	Hiển thị alert: "Vui lòng nhập đầy đủ thông tin"
3	Lưu trữ dữ liệu	Thông tin hợp lệ	Lưu thành công và giữ lại	Dữ liệu được lưu vào LocalStorage, không mất khi tải lại trang

Hình 4.4. Xác thực dữ liệu

#### 4.1.4. Kiểm thử responsive (Đa thiết bị)

Với mục tiêu phục vụ độc giả trên mọi nền tảng, giao diện website được kiểm thử khả năng hiển thị trên các kích thước màn hình khác nhau thông qua công cụ Chrome DevTools và thiết bị thực tế.

- Trên Desktop (Laptop/PC): Bố cục hiển thị đầy đủ 3 cột (Sidebar trái, Nội dung chính, Sidebar phải). Menu điều hướng hiển thị dạng lưới chi tiết.
- Trên Mobile/Tablet:
  - Hệ thống Grid System tự động chuyển đổi từ 3 cột sang 1 cột dọc để phù hợp với thao tác vuốt.
  - Thanh điều hướng chuyển sang dạng trượt ngang (scrollable), ẩn bớt các thông tin phụ như thời tiết hay quảng cáo dính để tối ưu không gian đọc.
  - Menu Sidebar chuyển sang dạng ẩn/hiện kích hoạt bằng nút Hamburger.

#### 4.1.5. Kiểm tra tương thích trình duyệt

Website được kiểm thử trên các trình duyệt phổ biến hiện nay như Google Chrome, Microsoft Edge và Firefox. Kết quả cho thấy:

- Cấu trúc HTML5 và CSS3 hiển thị đồng nhất trên các trình duyệt.

- Các tính năng JavaScript (Fetch API, DOM Manipulation) hoạt động ổn định, không phát sinh lỗi cú pháp (Syntax Error).

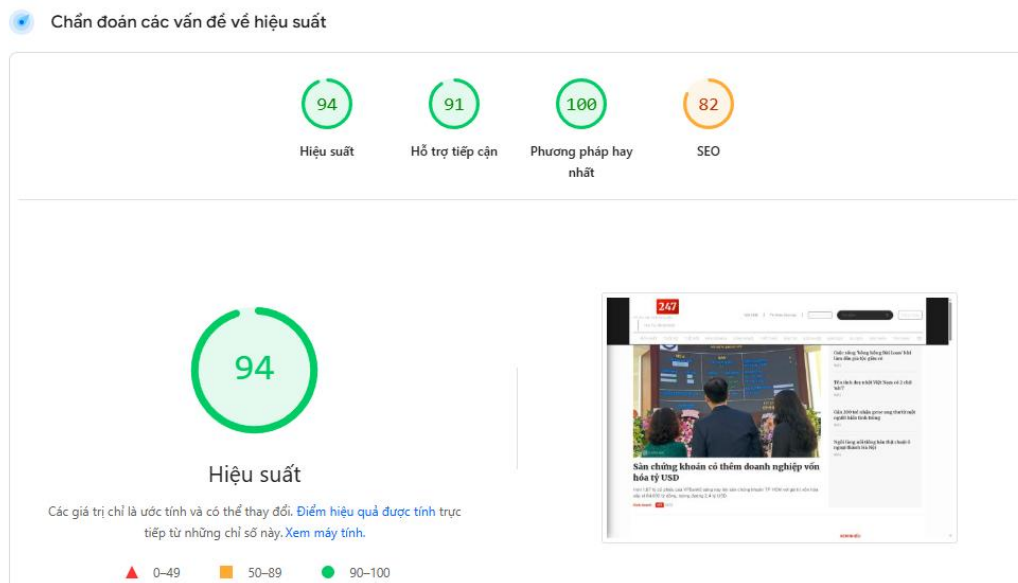
## 4.2. Tối ưu hóa hiệu suất

### 4.2.1. Tối ưu hóa mã nguồn và tài nguyên (CSS/JS/Images)

- Tối ưu CSS: Sử dụng biến CSS (:root) để quản lý màu sắc và kích thước thống nhất, giảm thiểu việc lặp lại mã nguồn. Các file CSS được tách biệt cho từng trang (homePage\_style.css) để trình duyệt chỉ tải những gì cần thiết.
- Tối ưu hình ảnh: Sử dụng hình ảnh từ các Content Delivery Network và icon dạng SVG (Scalable Vector Graphics) giúp giảm dung lượng tải xuống, tăng tốc độ hiển thị so với ảnh PNG/JPG truyền thống.
- Tối ưu font chữ: Sử dụng thuộc tính preconnect khi tải font từ Google Fonts, giúp thiết lập kết nối sớm đến máy chủ font, giảm thời gian chờ.

### 4.2.2. Đánh giá bằng Google PageSpeed

Nhóm đã sử dụng công cụ PageSpeed để đo lường các chỉ số hiệu suất quan trọng trên môi trường di động:



Hình 4.5. Đánh giá bằng Google PageSpeed

Bảng 4.2. Bảng tóm tắt hiệu suất

Chỉ số hiệu suất	Giá trị	Đánh giá	Ý nghĩa
<b>First Contentful Paint (FCP)</b>	0.8 giây	Rất tốt	Trang web hiển thị nội dung ban đầu rất nhanh chóng.
<b>Total Blocking Time (TBT)</b>	0 mili giây	Rất tốt	Khả năng tương tác của trang web gần như không bị gián đoạn.
<b>Largest Contentful Paint (LCP)</b>	1.4 giây	Trung bình	Thành phần nội dung lớn nhất tải gần đến ngưỡng lý tưởng (2.5s), cần cải thiện thêm để đạt điểm cao hơn.
<b>Cumulative Layout Shift (CLS)</b>	0.009	Rất tốt	Có hiện tượng dịch chuyển bố cục sau khi tải, cần khắc phục để tránh gây khó chịu cho người dùng.
<b>Speed Index</b>	1.4 giây	Trung bình	Tốc độ hiển thị nội dung trên màn hình nhanh, nhưng có thể tối ưu hơn.

#### 4.2.3. Đánh giá hiệu suất và SEO cơ bản

- SEO (Search Engine Optimization): Website được xây dựng với cấu trúc Semantic HTML (sử dụng các thẻ header, nav, main, article, footer đúng ngữ nghĩa) giúp các công cụ tìm kiếm dễ dàng thu thập dữ liệu. Các thẻ meta viewport được cấu hình chuẩn để hỗ trợ hiển thị trên di động.
- Trải nghiệm người dùng (UX): Áp dụng kỹ thuật Lazy Loading cho các thành phần Modal (Đăng nhập/Đăng ký), chỉ render khi người dùng thực sự tương tác, giúp giảm thời gian vẽ giao diện lần đầu (First Contentful Paint).

### 4.3. Triển khai website lên host và server

Sau khi hoàn tất quá trình phát triển và kiểm thử cục bộ (Localhost), hệ thống được đóng gói để triển khai lên môi trường Internet, đảm bảo người dùng có thể truy cập từ xa.

#### 4.3.1. Giới thiệu giải pháp Hosting

Do đặc thù hệ thống bao gồm cả Frontend và Backend (xử lý logic đăng nhập), nhóm lựa chọn giải pháp triển khai trên các nền tảng đám mây (Cloud Platform) hiện đại:

- Web Server (Backend & Frontend): Nhóm lựa chọn **Vercel** làm nơi lưu trữ và vận hành mã nguồn website. Đây là một nền tảng đám mây tiên tiến (Cloud Platform-as-a-Service) chuyên dụng cho các ứng dụng web hiện đại, với các ưu điểm vượt trội:
  - Kiến trúc Serverless (Không máy chủ): Vercel cho phép triển khai Backend (Node.js/Express) dưới dạng các Serverless Functions. Điều này giúp hệ thống có khả năng tự động mở rộng tài nguyên (auto-scaling) dựa trên lưu lượng truy cập thực tế mà không cần cấu hình máy chủ vật lý phức tạp.
  - Mạng lưới biên toàn cầu (Global Edge Network): Các tài nguyên tĩnh (HTML, CSS, JavaScript, hình ảnh) được Vercel tự động phân phối và lưu trữ đệm tại các máy chủ biên trên toàn thế giới. Điều này giúp giảm thiểu độ trễ mạng, mang lại tốc độ tải trang gần như tức thì cho người dùng ở bất kỳ vị trí địa lý nào.
  - Tích hợp CI/CD tự động: Vercel kết nối trực tiếp với GitHub, giúp tự động hóa quy trình triển khai (Continuous Deployment). Mọi thay đổi trong mã nguồn (commit) đều được tự động build và cập nhật lên môi trường production ngay lập tức, giúp nhóm tiết kiệm thời gian quản trị hệ thống.
- Cơ sở dữ liệu (Database): Đối với tầng dữ liệu, nhóm sử dụng dịch vụ Aiven Cloud, một giải pháp cơ sở dữ liệu được quản lý hoàn toàn :

- Tính ổn định và bảo mật cao: Aiven cung cấp môi trường MySQL trực tuyến với cơ chế sao lưu tự động và mã hóa dữ liệu đầu cuối, đảm bảo an toàn thông tin người dùng.
- Khả năng kết nối từ xa linh hoạt: Dịch vụ hỗ trợ kết nối ổn định từ Vercel thông qua giao thức TCP/IP bảo mật, đảm bảo luồng dữ liệu giữa Backend và Database luôn thông suốt và có độ trễ thấp nhất.

#### 4.3.2. Quy trình triển khai và cấu hình

Các bước triển khai hệ thống được thực hiện như sau:

- Bước 1: Quản lý phiên bản và đồng bộ mã nguồn (Version Control)  
Toàn bộ mã nguồn dự án (bao gồm Frontend, Backend và các file cấu hình) được quản lý chặt chẽ thông qua Git. Sau khi hoàn tất quá trình kiểm thử cục bộ, mã nguồn được đẩy (push) lên kho chứa trên GitHub. Đây là bước tiền đề quan trọng để kích hoạt quy trình triển khai tự động.
- Bước 2: Kết nối và khởi tạo dự án trên Vercel
  - Truy cập Dashboard của Vercel, thực hiện liên kết tài khoản với GitHub.
  - Chọn repository của dự án "News Website" và nhấn Import. Vercel sẽ tự động phát hiện mã nguồn là Node.js và đề xuất các cấu hình mặc định.
  - Tại bước này, hệ thống sẽ tự động đọc file package.json để xác định các thư viện cần thiết.
- Bước 3: Thiết lập biến môi trường :
  - DB\_HOST: Địa chỉ máy chủ MySQL trên Aiven Cloud.
  - DB\_USER: Tên đăng nhập cơ sở dữ liệu.
  - DB\_PASSWORD: Mật khẩu truy cập.
  - DB\_NAME: Tên cơ sở dữ liệu.
  - DB\_PORT: Cổng kết nối (mặc định cho Aiven). Việc này đảm bảo server khi khởi chạy trên đám mây sẽ có đầy đủ thông tin để "bắt tay" với cơ sở dữ liệu.

- Bước 4: Deploy

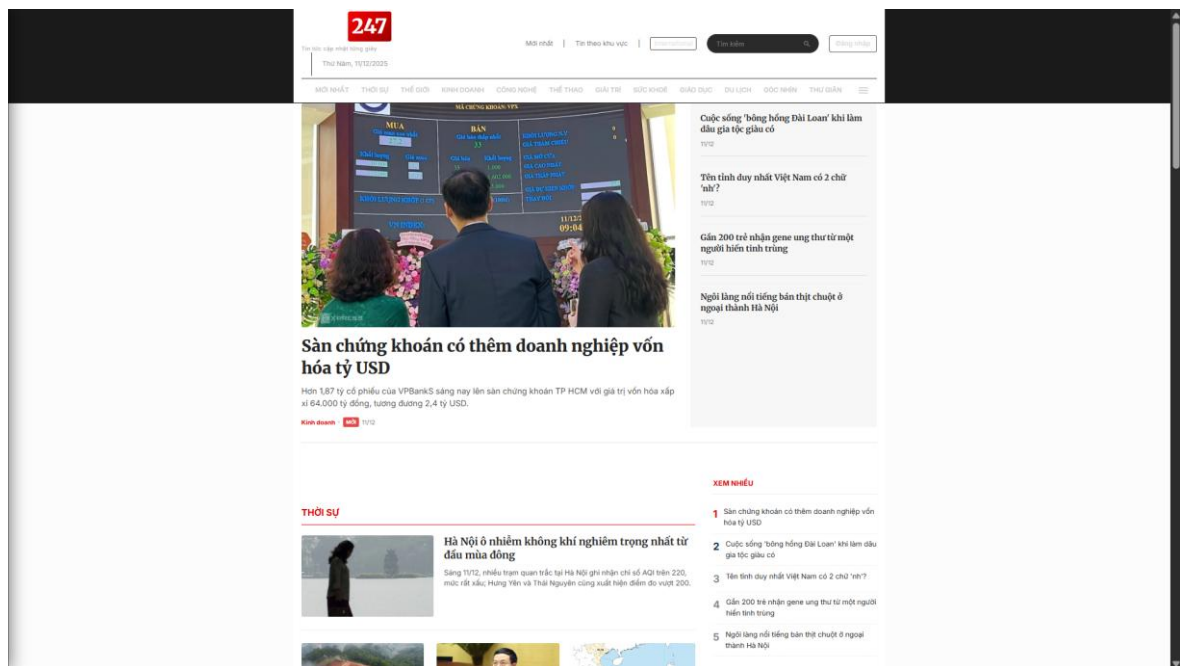
Sau khi hoàn tất cấu hình, nhấn Deploy. Vercel sẽ tiến hành:

- Tải mã nguồn từ GitHub.
- Chạy lệnh `npm install` để cài đặt các dependencies (Express, MySQL2, Cors, v.v.).
- Build dự án và phân phối mã nguồn tới mạng lưới máy chủ biên (Edge Network). Từ thời điểm này, bất kỳ thay đổi nào lên nhánh chính của GitHub đều sẽ tự động kích hoạt Vercel thực hiện lại quy trình trên (Re-deploy), đảm bảo website luôn ở phiên bản mới nhất mà không cần can thiệp thủ công.

#### 4.3.3. Kết quả triển khai và đánh giá độ ổn định

- **Link truy cập:** Website đã vận hành chính thức tại địa chỉ: **frontend-web-tin-tuc.vercel.app**
- Kiểm tra truy cập từ xa: Nhóm đã thực hiện truy cập thử nghiệm từ nhiều thiết bị khác nhau, sử dụng các mạng khác nhau. Kết quả cho thấy website hoạt động ổn định, tốc độ tải trang nhanh, cơ sở dữ liệu phản hồi tốt các thao tác đăng ký/đăng nhập của người dùng.





Hình 4.6. Giao diện trang chủ hoàn thiện trên Desktop.

**Kết luận chương 4:** Nhìn chung, hệ thống đã hoàn thành tốt các mục tiêu đề ra về mặt giao diện và chức năng cơ bản. Website hoạt động mượt mà, giao diện thân thiện, responsive tốt trên mọi thiết bị và cơ chế xác thực hoạt động chính xác. Các lỗi phát sinh trong quá trình kiểm thử đã được khắc phục kịp thời, sẵn sàng cho việc hosting lên internet.

## CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

### 5.1. Kết quả đạt được

Dựa trên các yêu cầu của đề tài và bảng tiêu chí đánh giá (Rubric), nhóm đã tổng hợp kết quả thực hiện và đối chiếu với mục tiêu ban đầu như sau:

Bảng 5.1. Đánh giá mức độ hoàn thiện của nhóm

STT	Tiêu chí đánh giá	Trọng số	Mức độ hoàn thành mục tiêu	Chi tiết kết quả thực hiện
1	Thiết kế và giao diện	20%	ĐÃ ĐẠT	Giao diện hoàn thiện, bố cục rõ ràng, màu sắc hài hòa. Đã đạt được tính trực quan và dễ sử dụng cho người dùng.
2	Responsive design	20%	ĐÃ ĐẠT	Website hiển thị ổn trên 3 nền tảng chính: PC, Tablet và Mobile. Bố cục không bị thay đổi quá nhiều khi thay đổi kích thước trình duyệt.
3	Chức năng và tương tác	15%	ĐÃ ĐẠT	Các chức năng chuyển trang, hiệu ứng hover, click hoạt động mượt mà. Không có lỗi logic nghiêm trọng làm gián đoạn trải nghiệm.
4	Triển khai (Deploy)	15%	ĐÃ ĐẠT	Website đã hoạt động online (có link truy cập công khai). Server phản hồi ổn định.
5	Tương thích trình duyệt	5%	ĐÃ ĐẠT	Chạy ổn định trên các trình duyệt khác nhau
6	Tính sáng tạo	10%	ĐÃ ĐẠT	Có sự đầu tư tùy biến giao diện, tạo được dấu ấn riêng so với các mẫu có sẵn.

7	Mã nguồn	10%	ĐÃ ĐẠT	Cấu trúc folder khoa học, code sạch, tuân thủ quy tắc đặt tên và cú pháp của HTML5/CSS3.
8	Tối ưu hóa hiệu suất	5%	ĐÃ ĐẠT	Website chạy với tốc độ khá trên các hệ thống kiểm tra

Đánh giá chung: Nhóm đã hoàn thành đầy đủ các hạng mục yêu cầu của đề án. So với mục tiêu ban đầu đề ra, sản phẩm đạt mức độ hoàn thiện ở mức **Trung bình**.

## 5.2. Hạn chế

Mặc dù đã hoàn thiện được một hệ thống website tin tức động (Dynamic Website) với đầy đủ Frontend và Backend, nhóm nhận thấy sản phẩm vẫn còn một số điểm hạn chế cần khắc phục trong tương lai:

### 5.2.1. Về mặt kỹ thuật và mã nguồn:

- Chưa có tính năng sửa bài viết (Update): Hệ thống quản trị (CMS) hiện tại mới chỉ hỗ trợ Đăng bài mới (Create) và Xóa bài (Delete). Tính năng chỉnh sửa nội dung bài viết đã đăng chưa được hoàn thiện.
- Xử lý hình ảnh chưa tối ưu: Ảnh được lưu trữ trực tiếp vào thư mục uploads của server hoặc dùng link ngoài. Chưa tích hợp các dịch vụ lưu trữ đám mây chuyên dụng (như Cloudinary, AWS S3) để tối ưu tốc độ tải và giảm tải cho server chính.
- Bảo mật còn ở mức cơ bản: Mật khẩu người dùng tuy đã được lưu trữ nhưng chưa áp dụng cơ chế mã hóa mạnh (như Bcrypt) hoặc xác thực qua JWT (Json Web Token), có thể tiềm ẩn rủi ro nếu lộ database.

### 5.2.2. Về giao diện và trải nghiệm người dùng (UI/UX):

- Trải nghiệm trên Admin Dashboard: Giao diện trang quản trị còn đơn giản, sử dụng các hộp thoại thông báo mặc định của trình duyệt (alert, confirm) thay vì các Modal hiện đại, gây cảm giác chưa chuyên nghiệp.
- Thiếu bộ soạn thảo văn bản (Rich Text Editor): Khi đăng bài, Admin phải nhập mã HTML thủ công vào ô nội dung. Hệ thống chưa tích hợp bộ công cụ soạn

thảo trực quan (như CKEditor hay TinyMCE) để hỗ trợ định dạng văn bản (in đậm, nghiêng, chèn ảnh vào giữa bài).

- Khả năng tương thích thiết bị: Mặc dù đã hỗ trợ Responsive, nhưng trên một số dòng điện thoại màn hình gấp hoặc độ phân giải quá thấp, bố cục menu và danh sách tin đôi khi vẫn chưa hiển thị tối ưu nhất.

#### 5.2.3. Về tính năng:

- Thiếu tính năng tương tác độc giả: Website chưa có phần Bình luận (Comment) hoặc thả cảm xúc (Like/React) dưới mỗi bài viết.
- Chưa phân quyền chi tiết: Hệ thống mới chỉ phân biệt Admin và User thường, chưa có các quyền hạn chi tiết hơn như "Biên tập viên" (chỉ được đăng/sửa bài, không được xóa user).

### 5.3. Hướng phát triển

Từ những hạn chế đã phân tích ở trên, nhóm đề xuất lộ trình phát triển tiếp theo để nâng cấp sản phẩm:

#### 5.3.1. Xây dựng hệ thống quản trị (CMS)

- Bổ sung chức năng Edit: Phát triển API PUT để cho phép Admin cập nhật lại tiêu đề, nội dung hoặc ảnh bìa của các bài viết đã đăng.
- Tích hợp CKEditor: Nhúng thư viện soạn thảo văn bản vào trang Admin, giúp việc trình bày bài viết trở nên dễ dàng và sinh động hơn.

#### 5.3.2. Nâng cao trải nghiệm và tối ưu hóa

- Cải thiện UI/UX: Thiết kế lại các popup thông báo, thêm chế độ Dark Mode (Giao diện tối).
- Tối ưu SEO: Bổ sung các thẻ meta, cấu trúc dữ liệu (Schema) để website thân thiện hơn với các công cụ tìm kiếm.

#### 5.3.3. Mở rộng hệ sinh thái người dùng

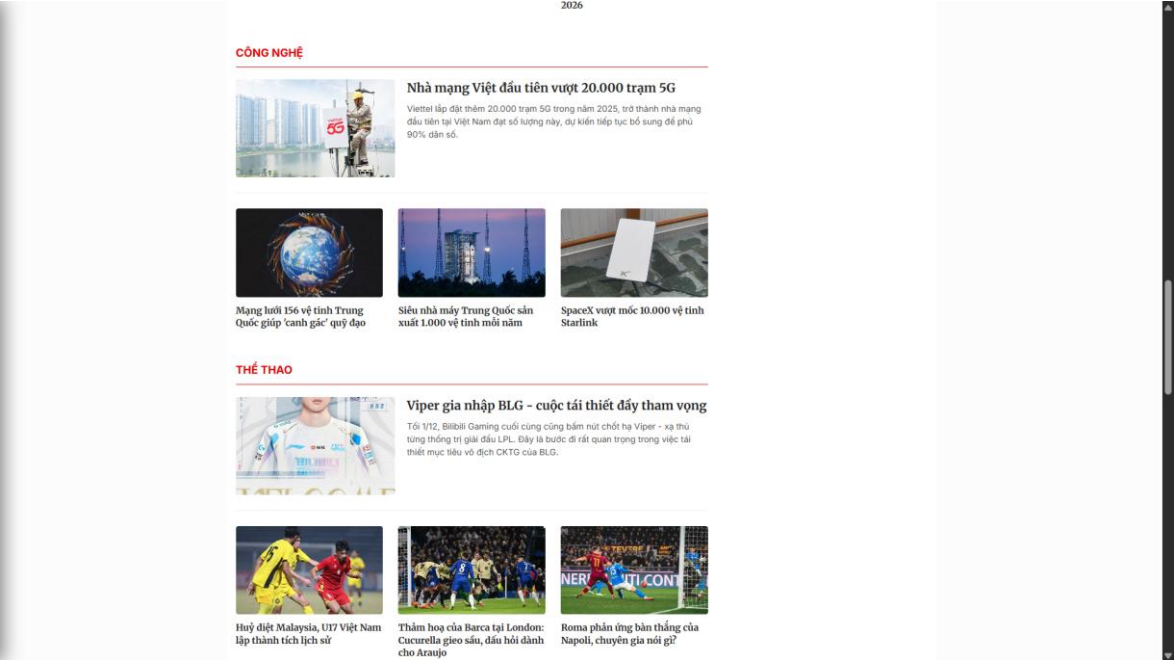
- Phát triển tính năng bình luận thời gian thực (Real-time comment).
- Thêm chức năng "Gợi ý tin tức" dựa trên lịch sử xem bài của người dùng (Personalization).

## TÀI LIỆU THAM KHẢO

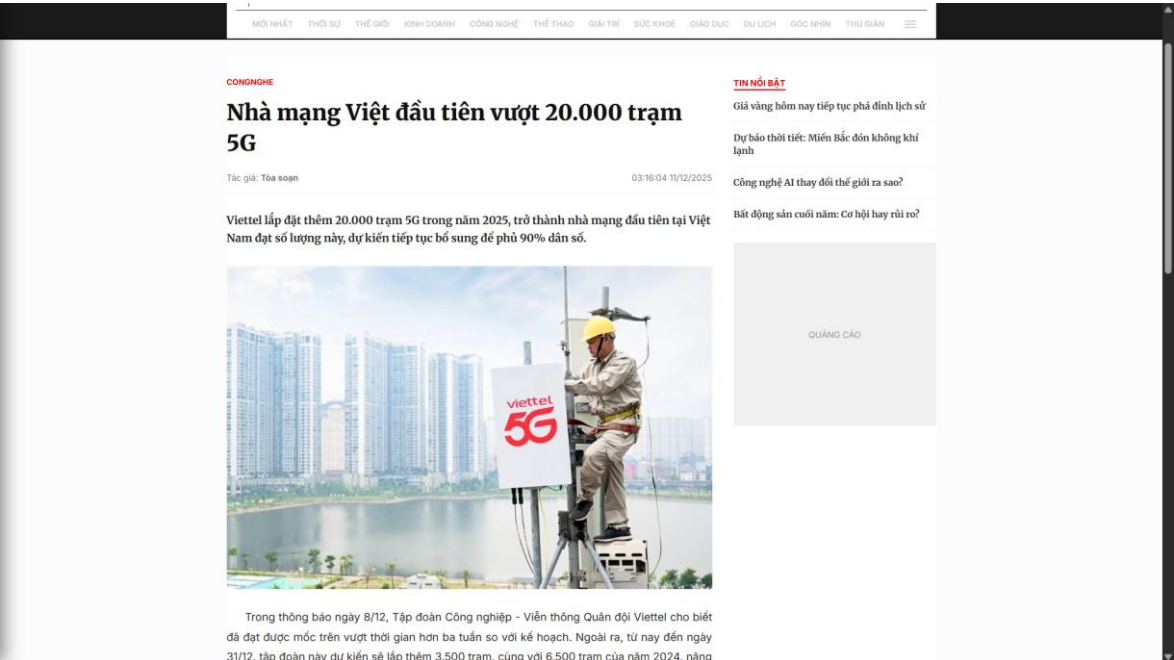
- [1] W3School (1999), Refsnes Data, [W3Schools Online Web Tutorials](#) [truy cập ngày 2/12/2025]
- [2] Jennifer Robbins, Learning Web Design (2018), NXB O'Reilly Media, Incorporated [truy cập ngày 2/12/2025]
- [3] VINALINK MEDIA (2009), Vinalink Media, [Hướng dẫn tự học thiết kế website cơ bản A-Z](#) [truy cập ngày 2/12/2025]
- [4] VIETNIX (2025), Vietnix JSC, [Cách upload website lên hosting cực dễ dàng, chi tiết 2025](#) [truy cập ngày 2/12/2025]
- [5] Phạm Thị Thu Hà (2014). Giáo trình Quản lý dự án, NXB Bách khoa, Hà Nội [truy cập ngày 2/12/2025]
- [6] VnExpress (2001), FPT Online, VnExpress - Báo tiếng Việt nhiều người xem nhất [truy cập ngày 2/12/2025]
- [7] Báo Mới (2005), VNG Corporation, Báo Mới - Trang tin tức tổng hợp tự động [truy cập ngày 2/12/2025]

# PHỤ LỤC

## 1. Ảnh chụp giao diện:



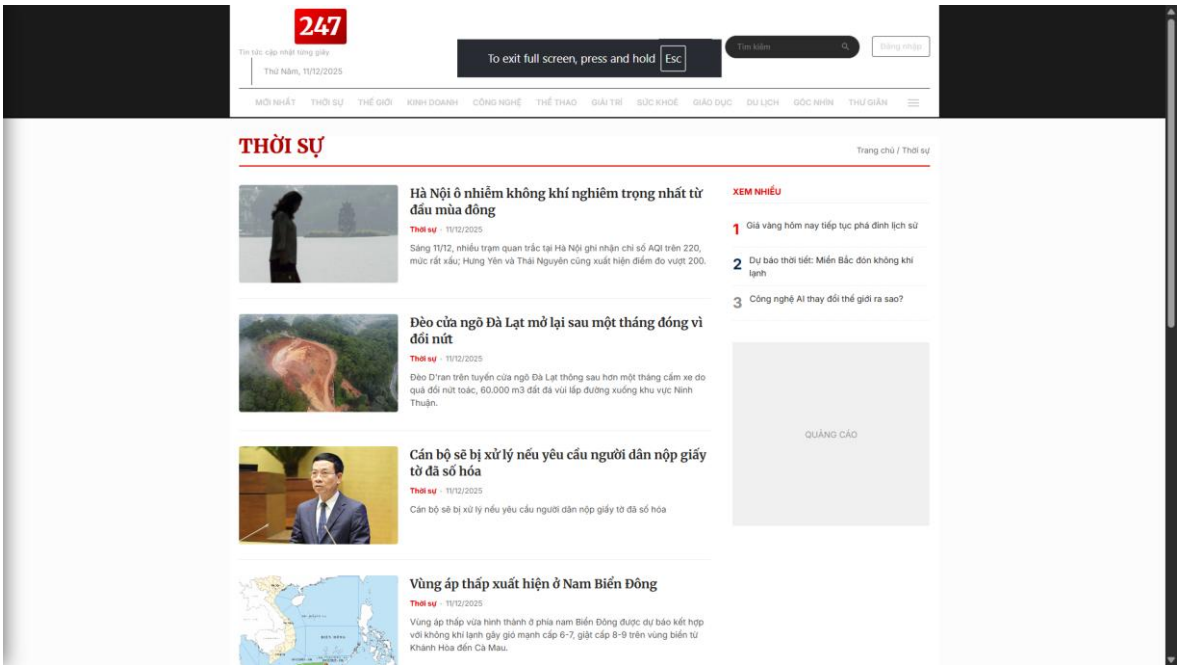
Phụ lục 1.1. Giao diện trang



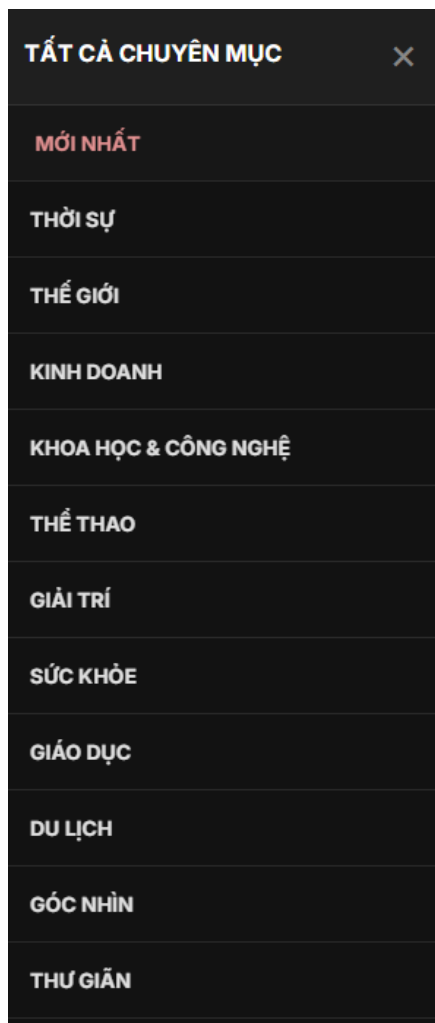
Phụ lục 1.2. Giao diện khi đọc chi tiết

Phụ lục 1.3. Footer của trang

Phụ lục 1.4. Navbar của trang

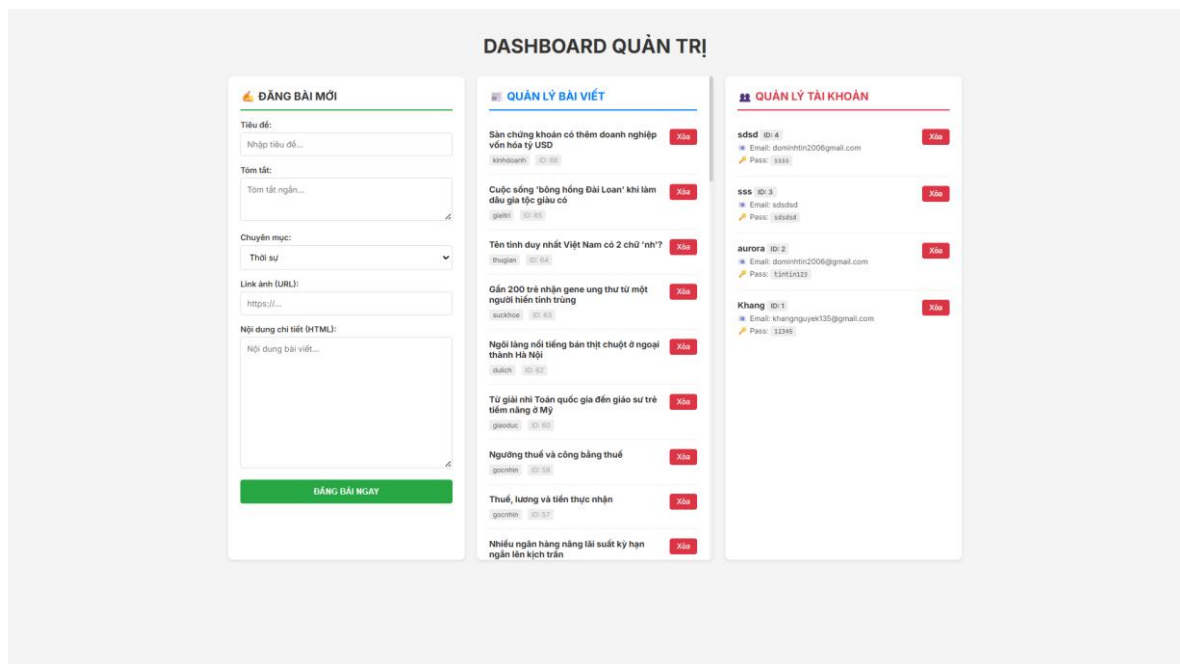


Phụ lục 1.5. Giao diện khi bấm chi tiết vào 1 chủ đề



Phụ lục 1.6. Khi bấm vào Hamburger Menu





Phụ lục 1.7: Dashboard quản trị

## 2. Liên kết

Link Github: <https://github.com/NguyenGiaKhang123/ThietKeWeb>

Link website: [frontend-web-tin-tuc.vercel.app](https://frontend-web-tin-tuc.vercel.app)