

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC NGHIỆM

Học phần: Phần mềm mã nguồn mở

**Chủ đề: Xây dựng trang web "Tìm đồ thất lạc" bằng
VueJS, Dapper và Firebase Authentication**

Giáo viên hướng dẫn : TS. Nguyễn Thái Cường

Nhóm sinh viên thực hiện:

1. Lê Đức Tiệp	2020604011
2. Nguyễn Thị Phụng	2020604425
3. Nguyễn Thị Thương	2020603814
4. Nguyễn Công Thành	2020603178
5. Phạm Trí Nghĩa	2021602169

Mã Lớp: 20224IT6030001

Nhóm : 02

Hà Nội, 2023

TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO THỰC NGHIỆM

Học phần: Phần mềm mã nguồn mở

**Chủ đề: Xây dựng trang web "Tìm đồ thất lạc" bằng
VueJS, Dapper và Firebase Authentication**

Giáo viên hướng dẫn : TS. Nguyễn Thái Cường

Nhóm sinh viên thực hiện:

1. Lê Đức Tiệp	2020604011
2. Nguyễn Thị Phụng	2020604425
3. Nguyễn Thị Thương	2020603814
4. Nguyễn Công Thành	2020603178
5. Phạm Trí Nghĩa	2021602169

Mã Lớp: 20224IT6030001

Nhóm : 02

MỤC LỤC

LỜI CẢM ƠN	4
PHẦN 1: GIỚI THIỆU ĐỀ TÀI	5
1. Tên đề tài.....	5
2. Lý do chọn	5
PHẦN 2: TÌM HIỂU CÁC CÔNG NGHỆ SỬ DỤNG TRONG DỰ ÁN	5
1. TÌM HIỂU VỀ VueJS.....	5
1.1 Giới thiệu về VueJS	5
1.2 Ưu và nhược điểm VueJS	7
1.3 Một vài tính năng đặc trưng của VueJS.....	9
2. TÌM HIỂU VỀ Dapper	11
3. Dapper là gì?.....	11
3.1 Ưu và nhược điểm.....	12
3.1.1 Ưu điểm	12
3.1.2 Nhược điểm	12
4. TÌM HIỂU VỀ Firebase.....	13
4.1 Firebase là gì?	13
4.2 Firebase hoạt động với chức năng như thế nào?.....	14
4.2.1 Firebase Authentication	14
4.2.2 Firebase Hosting	14
4.2.3 Firebase Realtime Database.....	15
4.3 Ưu nhược điểm của Firebase	15
4.3.1 Ưu điểm:	15
4.3.2 Nhược điểm	16
PHẦN 3: PHÂN TÍCH, THIẾT KẾ, PHÁT TRIỂN DỰ ÁN	22
1. Biểu đồ usecase	17
2. Các use case chính	26
2.1 Mô tả chi tiết các use case.....	28

2.1.1 Mô tả use case Cập nhật trang cá nhân (Nghĩa)	28
2.1.2 Mô tả use case Xem trang cá nhân (Nghĩa).....	29
2.1.3 Mô tả use case Duyệt bài đăng(Nguyễn Thị Phụng).....	30
2.1.4 Mô tả use case Xóa bài đăng(Nguyễn Thị Phụng).....	31
2.1.5 Mô tả use case Xem bài viết theo bộ lọc và tìm kiếm (Thương)	32
2.1.6 Mô tả UseCase Xem chi tiết bài đăng (Thương).....	32
2.1.7 Mô tả UseCase Đăng nhập (Thành)	33
2.1.8 Mô tả use case Đăng ký (Thành).....	35
Biểu đồ lớp.....	39
3: Thiết kế, xây dựng, phát triển dự án	40
1. Tìm hiểu cách hoạt động của VueJS.....	40
1.1 Cách hoạt động của VueJS	40
1.2 Cách build 1 dự án VueJS đơn giản	41
2. Thiết kế giao diện các màn hình trang web theo đề tài.	42
3. Xây dựng chương trình	51
TỔNG KẾT	56
Tài liệu tham khảo:	57

LỜI CẢM ƠN

“Đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến Trường Đại học Công Nghiệp Hà Nội đã đưa môn học Phần mềm mã nguồn mở vào chương trình giảng dạy. Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn – Thầy Nguyễn Thái Cường đã dạy, truyền đạt những kiến thức quý báu cho em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học Phần mềm mã nguồn mở của thầy, em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để em có thể vững bước sau này.

Bộ môn Phần mềm mã nguồn mở là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Tuy nhiên, do vốn kiến thức còn nhiều hạn chế và khả năng tiếp thu thực tế còn nhiều bỡ ngỡ. Mặc dù nhóm chúng em đã cố gắng hết sức nhưng chắc chắn bài báo cáo khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong thầy xem xét và góp ý để bài báo cáo của nhóm chúng em được hoàn thiện hơn.

Em xin chân thành cảm ơn!”

PHẦN 1: GIỚI THIỆU ĐỀ TÀI

1. Tên đề tài:

Xây dựng trang web "Tìm đồ thất lạc" bằng VueJS, Dapper và Firebase Authentication

2. Lý do chọn

Hiện nay, ngành Công nghệ thông tin đã có những bước phát triển nhanh chóng về ứng dụng của nó trong mọi lĩnh vực trong cuộc sống trên phạm vi toàn thế giới nói chung và Việt Nam nói riêng. Công nghệ thông tin là một phần không thể thiếu của cuộc sống văn minh, góp phần đẩy mạnh công cuộc công nghiệp hóa hiện đại hóa đất nước. Việc ứng dụng những thành quả của khoa học công nghệ vào trong đời sống, trong công tác là hết sức thiết yếu. Ứng dụng của công nghệ thông tin kết hợp với truyền thông hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của công ty, các tổ chức... nó đóng vai trò quan trọng và không thể thiếu. Công nghệ thông tin và truyền thông hóa góp phần làm thay đổi suy nghĩ, lối mòn tư duy của mỗi con người, nó giúp con người năng động hơn, kết nối nhanh hơn ở mọi lúc mọi nơi làm tăng mức độ hiệu quả, năng suất của công việc.

PHẦN 2: TÌM HIỂU CÁC CÔNG NGHỆ SỬ DỤNG TRONG DỰ ÁN

1. TÌM HIỂU VỀ VUEJS

1.1 Giới thiệu về VueJS

Đầu tiên, VueJS là gì? VueJS là một framework mã nguồn mở của JavaScript được sử dụng để phát triển các giao diện web tương tác. Nó là một trong những framework nổi tiếng được sử dụng để đơn giản hóa việc phát triển web. VueJS tập trung vào view layer. Nó có thể dễ dàng tích hợp vào các dự án lớn để phát triển front-end mà không gặp bất kỳ sự cố nào.



Hình 1.1 VueJS

1.2 Ưu và nhược điểm VueJS

1.2.1 Ưu điểm

Kích thước nhỏ: Tập zip được tải xuống của framework này chỉ nặng 18KB. Điều này khiến nó không chỉ cài đặt nhanh mà còn tác động tích cực đến SEO và UX của bạn.

Kết xuất và hiệu suất DOM ảo: Mô hình đối tượng tài liệu (DOM) là thứ bạn có thể gặp phải khi kết xuất các trang web. DOM đại diện cho một trang HTML với các kiểu, thành phần và nội dung dưới dạng cấu trúc cây của các đối tượng (nút). Các đối tượng cây DOM lưu trữ dưới dạng cây và được tạo bởi trình duyệt khi tải trang.

Khi người dùng tương tác với trang, các đối tượng sẽ thay đổi trạng thái của chúng, do đó trình duyệt sẽ phải cập nhật thông tin và hiển thị trên màn hình. Tuy nhiên, việc cập nhật toàn bộ DOM rất phức tạp. Ưu tiên tốc độ load, VueJS sử dụng DOM ảo. Hãy coi đây là một bản sao của DOM gốc giúp tìm ra những phần tử cần cập nhật mà không cần kết xuất lại toàn bộ cây nút. Cách tiếp cận này giúp hiển thị trang khá nhanh và cải thiện hiệu suất ứng dụng.

Hệ thống phản ứng và các tùy chọn ràng buộc dữ liệu: Liên kết dữ liệu là kết nối giữa mô hình dữ liệu (nguồn dữ liệu) và mẫu DOM hoặc HTML của chế độ xem. Liên kết dữ liệu một chiều cho phép thông tin truyền theo một hướng, từ mô hình sang chế độ xem hoặc ngược lại. Trong trường hợp đầu tiên, các thay đổi đối với nguồn sẽ tự động cập nhật DOM, nhưng nó không hoạt động ngược lại vì DOM có quyền truy cập chỉ đọc vào mô hình.

Liên kết dữ liệu hai chiều cho phép trao đổi dữ liệu giữa mô hình và chế độ xem theo cả hai hướng. Nói cách khác, mô hình cũng lắng nghe các sự kiện trên DOM và bất kỳ cập nhật nào ở một bên sẽ phản ánh ngay lập tức ở bên kia. Cách tiếp cận này loại bỏ mã soạn sẵn và đơn giản hóa việc phát triển ứng dụng. Tuy nhiên, việc khó Debug và dễ xảy ra lỗi khiến luồng hai chiều không phù hợp cho các dự án lớn.

1.2.2 Nhược điểm

Với nhiều ưu điểm như vậy, liệu nhược điểm của VueJS là gì? Đồng

xu luôn có hai mặt, vì vậy, hãy xem xét các điểm yếu của Vue.js.

Rào cản ngôn ngữ: Việc áp dụng VueJS của các doanh nghiệp như Xiaomi và Alibaba đã giúp phổ biến framework và tạo ra nhu cầu trên thị trường lao động. Với việc Vuejs ngày càng phổ biến ở Trung Quốc, một phần quan trọng trong nội dung và các cuộc thảo luận của nó đều là tiếng Trung.

Bức tường lửa gắt gao của Trung Quốc khiến mọi thứ ở quốc gia này hơi khác một chút vì rất nhiều tài nguyên phổ biến hiện không có sẵn ở đó. Điều đó làm cho việc học và sử dụng React hoặc Angular đối với các nhà phát triển địa phương trở nên khó khăn hơn. Chính vì vậy, đối với các nhà phát triển ở Trung Quốc, VueJS là một biến thể thích hợp hơn.

Vì vậy, khi tìm kiếm nội dung về VueJS, bạn sẽ phải đối mặt với các cuộc thảo luận diễn đàn, mô tả plugin và hướng dẫn bằng tiếng Trung. Đây có thể là một vấn đề nếu bạn không biết ngôn ngữ này.

Thiếu hỗ trợ cho các dự án quy mô lớn: Quy mô nhóm phát triển và cộng đồng của VueJS vẫn không thể so sánh với Angular hay React. Framework này cũng không được hỗ trợ tài chính từ các doanh nghiệp lớn. Để được áp dụng trong các dự án quy mô lớn, công nghệ phải ổn định và được hỗ trợ mạnh mẽ để các vấn đề có thể được giải quyết nhanh chóng. Mặc dù VueJS không gặp nhiều vấn đề và thậm chí còn có nhu cầu đến từ các doanh nghiệp như IBM và Adobe, nhưng nó chủ yếu được sử dụng trong các dự án tương đối nhỏ.

Nguy cơ đến từ việc quá linh hoạt: Tính linh hoạt là một đặc tính gây tranh cãi của một dự án lớn. Cung cấp cho nhóm phát triển của bạn quá nhiều tùy chọn có thể dẫn đến các cách tiếp cận lập trình khác nhau trong một nhóm. Và kết quả là, nó trở thành một công cụ vô hiệu hóa cuối cùng thay vì một phần mềm hoạt động.

Nguồn tài nguyên giới hạn: Mặc dù hệ sinh thái khá rộng và có tất cả các công cụ cần thiết để bắt đầu phát triển với VueJS, nhưng framework này không lớn bằng React hay Angular. Nói chính xác hơn, chỉ cần so sánh số lượng plugin có sẵn cho React và Vue.js, sự khác biệt là ở hàng trăm đơn vị. Các plugin hiện có có thể được sử dụng với các framework khác cũng thường không được hỗ trợ.

1.3 Một vài tính năng đặc trưng của VueJS

DOM ảo

VueJS sử dụng DOM ảo, cũng được sử dụng bởi các framework khác như React, Ember, v.v. Các thay đổi không được thực hiện đối với DOM, thay vào đó, một bản sao của DOM được tạo dưới dạng cấu trúc dữ liệu JavaScript. Khi nào có bất kỳ thay đổi nào, chúng sẽ được thực hiện đối với cấu trúc dữ liệu JavaScript. Cấu trúc sau được so sánh với cấu trúc dữ liệu ban đầu. Những thay đổi cuối cùng sau đó được cập nhật vào DOM thực, người dùng sẽ thấy thay đổi này. Điều này tốt về mặt tối ưu hóa, ít tốn kém hơn và các thay đổi có thể được thực hiện với tốc độ nhanh hơn.

Ràng buộc dữ liệu

Tính năng ràng buộc dữ liệu giúp thao tác hoặc gán giá trị cho các thuộc tính HTML, thay đổi kiểu, gán lớp với sự trợ giúp của chỉ thị ràng buộc có tên v-bind có sẵn trong VueJS.

Component

Component là một trong những tính năng quan trọng của VueJS giúp tạo các phần tử tùy chỉnh và có thể tái sử dụng trong HTML.

Xử lý sự kiện

v-on là thuộc tính được thêm vào các phần tử DOM để lắng nghe các sự kiện trong VueJS.

Hoạt ảnh/Chuyển tiếp

VueJS cung cấp nhiều cách khác nhau để áp dụng chuyển đổi sang các phần tử HTML khi chúng được thêm, cập nhật hoặc xóa khỏi DOM. VueJS có một thành phần chuyển tiếp tích hợp cần được bao quanh phần tử để tạo hiệu ứng chuyển tiếp. Người dùng có thể dễ dàng thêm các thư viện hoạt hình của bên thứ ba và cũng có thể thêm nhiều tương tác hơn vào giao diện.

Thuộc tính được tính toán

Đây là một trong những tính năng quan trọng của VueJS. Nó giúp lắng nghe những thay đổi được thực hiện đối với các thành phần giao diện người dùng và thực hiện các tính toán cần thiết. Không cần mã hóa bổ sung cho việc này.

Mẫu

VueJS cung cấp các mẫu dựa trên HTML liên kết DOM với dữ liệu đối tượng Vue. VueJS biên dịch các mẫu thành các hàm Kết xuất DOM ảo. Chúng ta có thể sử dụng mẫu của các chức năng kết xuất và ngược lại.

Chỉ thị

VueJS có các chỉ thị tích hợp như v-if, v-else, v-show, v-on, v-bind và v-model, được sử dụng để thực hiện các hành động khác nhau trên giao diện người dùng.

2. TÌM HIỂU VỀ Dapper

2.1 Dapper là gì?

Dapper là một thư viện ORM (Object-Relational Mapping) mã nguồn mở được viết cho ngôn ngữ lập trình C#. ORM là một phần mềm trung gian giữa cơ sở dữ liệu và mã nguồn ứng dụng, giúp ánh xạ dữ liệu từ các bảng trong cơ sở dữ liệu vào các đối tượng trong mã nguồn và ngược lại.

Thư viện Dapper giúp đơn giản hóa quá trình truy vấn và thao tác dữ liệu trong cơ sở dữ liệu SQL bằng cách cho phép bạn viết các truy vấn SQL trực tiếp và sau đó sử dụng Dapper để thực hiện truy vấn đó và ánh xạ kết quả trả về vào các đối tượng trong mã nguồn của ứng dụng.

Dapper nổi tiếng với hiệu suất cao, điều này là do nó được thiết kế theo hướng tối ưu hóa, tập trung vào việc giảm thiểu sự trễ và tối ưu hóa hiệu suất truy vấn dữ liệu. Nó không cung cấp nhiều tính năng phức tạp như một số thư viện ORM khác, nhưng lại tập trung vào cung cấp giải pháp đơn giản và hiệu quả cho việc tương tác với cơ sở dữ liệu.

Dapper được phát triển bởi Sam Saffron, một thành viên của đội ngũ Stack Overflow, và mã nguồn của nó được lưu trữ trên GitHub để mọi người có thể đóng góp và sử dụng miễn phí.

Tóm lại, Dapper là một thư viện ORM nhẹ nhàng và hiệu quả cho .NET Framework và .NET Core, giúp đơn giản hóa việc truy vấn và ánh xạ dữ liệu giữa cơ sở dữ liệu SQL và mã nguồn ứng dụng.

2.2 Ưu và nhược điểm

2.2.1 Ưu điểm

Hiệu suất cao: Dapper nổi tiếng với hiệu suất tốt. Thư viện này sử dụng cơ chế đọc dữ liệu sơ bộ (micro-ORM), giúp giảm bớt trễ và tối ưu hóa truy vấn dữ liệu. Điều này thích hợp cho các ứng dụng yêu cầu xử lý dữ liệu nhanh chóng và hiệu quả.

Đơn giản và dễ sử dụng: Dapper không yêu cầu nhiều cấu hình phức tạp và không cần tạo các lớp đối tượng kế thừa. Bạn chỉ cần viết truy vấn SQL thông thường và sử dụng Dapper để ánh xạ dữ liệu vào đối tượng.

Hỗ trợ nhiều loại cơ sở dữ liệu: Mặc dù xuất phát từ SQL Server, Dapper cũng hỗ trợ nhiều hệ quản trị cơ sở dữ liệu khác như MySQL, PostgreSQL, Oracle và SQLite.

Tích hợp tốt: Dapper hoạt động trên nhiều phiên bản .NET Framework và .NET Core, cho phép bạn sử dụng nó trong nhiều dự án khác nhau.

Linh hoạt: Dapper cho phép bạn tùy chỉnh cách ánh xạ dữ liệu bằng cách sử dụng các thuộc tính và phương thức mở rộng, điều này giúp bạn kiểm soát các dữ liệu được ánh xạ vào các đối tượng.

2.2.1 Nhược điểm

Thiếu các tính năng cao cấp của ORM khác: So với một số thư viện ORM khác, Dapper thiếu một số tính năng phức tạp như quan hệ đa mức, tự động tạo cơ sở dữ liệu, quản lý thay đổi cơ sở dữ liệu tự động và hơn thế nữa.

Phải viết truy vấn SQL thủ công: Mặc dù điều này có thể được coi là một ưu điểm nếu bạn thích kiểm soát hoàn toàn truy vấn, nhưng việc viết truy vấn SQL thủ công cũng có thể là một gánh nặng nếu bạn không quen với SQL.

Thiếu tài liệu chi tiết: Dapper có tài liệu khá đầy đủ, nhưng so với một số ORM khác, tài liệu và hướng dẫn chi tiết có thể hơi thiếu.

Tóm lại, Dapper là một thư viện ORM đơn giản và hiệu quả, phù hợp cho

các dự án cần hiệu suất cao và sự tương tác đơn giản với cơ sở dữ liệu. Tuy nhiên, nó cũng có nhược điểm là thiếu một số tính năng phức tạp mà các thư viện ORM khác có thể cung cấp.

3. TÌM HIỂU VỀ FIREBASE

3.1 Firebase là gì?

Firebase là một nền tảng dịch vụ đám mây của Google cung cấp cho các nhà phát triển các công cụ để xây dựng và triển khai ứng dụng di động và web. Nền tảng này cung cấp nhiều tính năng như cơ sở dữ liệu thời gian thực, phân tích,



lưu trữ, xác thực người dùng, push notification và nhiều hơn nữa. Với Firebase, các nhà phát triển không cần phải tự xây dựng các phần mềm phức tạp để quản lý cơ sở dữ liệu và hạ tầng, mà có thể tập trung vào việc phát triển ứng dụng của mình. Firebase được sử dụng rộng rãi trong cộng đồng phát triển ứng dụng và web hiện nay.

3.2 Firebase hoạt động với chức năng như thế nào?

Chức năng hoạt động của **Firebase là gì?** Kể từ sau khi Google chính thức mua lại và tiếp tục phát triển, cho đến nay Firebase gồm có các hoạt động điển hình như sau:

3.2.1 *Firebase Authentication*

Hoạt động nổi trội nhất của Firebase chính là xây dựng những bước xác định người dùng thông qua Email, Facebook, Twitter, GitHub hay Google. Ngoài ra, hoạt động Firebase Authentication cũng hỗ trợ xác thực nặc danh cho những ứng dụng. Hoạt động xác thực của Firebase có thể giúp cho thông tin cá nhân của những người sử dụng được an toàn hơn. Điều này cũng đảm bảo tài khoản và các thông tin cá nhân của người dùng không bị đánh cắp.

3.2.2 *Firebase Hosting*

Cách thức hoạt động tiếp theo được nhắc đến là Firebase Hosting. Đây là một hoạt động được phân phối thông qua tiêu chuẩn công nghệ bảo mật SSL từ hệ thống mạng CDN.

CDN chính là cụm từ viết tắt của Content Delivery Network chính là một mạng lưới máy chủ giúp lưu giữ lại các bản sao của các nội dung tĩnh, Những nội dung tĩnh này nằm ở bên trong website và trực tiếp phân phối đến các máy chủ PoP khác. Mạng lưới của máy chủ CDN được thiết đặt ở khắp nơi trên thế giới.

Từ máy chủ Pop – Points of Presence, nguồn dữ liệu sẽ được gửi đi đến những người dùng cuối cùng.

3.2.3 *Firebase Realtime Database*

Firebase Realtime Database có dạng một JSON đã được đồng bộ thời gian đến với tất cả các kết nối client. Để có được hoạt động này thì các lập trình viên cần phải đăng ký tài khoản ở trên Firebase. Dữ liệu ở trong database sẽ tự động cập nhật một cách liên tục khi phát triển ứng dụng. Sau khi đã được cập nhật thì những dữ liệu này sẽ được truyền tải thông qua các kết nối SSL có 2048 bit.

3.3 Ưu nhược điểm của Firebase

3.3.1 *Ưu điểm:*

- **Sử dụng dễ dàng:** Những người dùng có thể đăng ký một tài khoản Firebase thông qua tài khoản Google. Đồng thời, người dùng cũng có thể sử dụng nền tảng này trong quá trình phát triển ứng dụng một cách đơn giản nhất.
- **Tốc độ phát triển nhanh:** Ưu điểm tiếp theo chính là Firebase hỗ trợ cho việc phát triển ứng dụng rất nhanh chóng. Điều này sẽ giúp lập trình viên giảm bớt được thời gian để phát triển cũng như tiếp thị ứng dụng.
- **Cung cấp nhiều dịch vụ:** Firebase còn cung cấp đa dịch vụ cho mục đích phát triển trang web. Người dùng có thể lựa chọn database Firestore hoặc Realtime theo đúng ý muốn của mình.
- **Nền tảng cho Google phát triển:** *google firebase là gì?* Firebase được Google mua lại và trở thành một phần của Google. Ứng dụng này sẽ khai thác được triệt để sức mạnh cũng như các dịch vụ hiện đang sẵn có của Google.
- **Giao diện người dùng được chú trọng:** Firebase sẽ cho phép các lập trình viên tập trung hơn vào việc phát triển giao diện của người dùng thông qua kho Backend mẫu vô cùng đa dạng.
- **Firebase app không có máy chủ:** Chính điều này sẽ giúp cho Firebase có được khả năng tối ưu hóa nhất về hiệu suất làm việc nhờ vào việc mở rộng cụm database.
- **Học máy:** Ứng dụng Firebase sẽ cung cấp học máy cho các lập trình viên để hỗ trợ tốt nhất cho việc phát triển ứng dụng.
- **Tạo lưu lượng truy cập:** *Firebase App* sẽ hỗ trợ việc tạo lập các chỉ mục. Đồng thời, Firebase cũng sẽ giúp nâng cao thứ hạng của ứng dụng ở trên

bảng xếp hạng của Google. Nhờ vậy mà lượt traffic sẽ tăng lên.

- **Theo dõi lỗi:** Đây là một công cụ để phát triển cũng như khắc phục lỗi vô cùng tuyệt vời. Nhờ vậy khi sử dụng, bạn không cần lo lắng mình sẽ đề sớt lỗi.

- **Chức năng sao lưu:** *cách sử dụng firebase* sao lưu một cách thường xuyên và đảm bảo tính sẵn có. Đồng thời, chức năng này cũng giúp cho thông tin và dữ liệu được bảo mật một cách an toàn nhất.

3.3.2 Nhược điểm:

- **Firestore không là mã nguồn mở:** Điều này sẽ giúp cho ứng dụng trở thành một lựa chọn không quá tối ưu đối với nhiều nhà phát triển. Người dùng không thể sửa đổi được mã nguồn Firestore.
- **Người dùng không truy cập được mã nguồn:** Đối với những ứng dụng lớn thì việc chuyển đổi sang các nhà cung cấp khác thực sự không dễ dàng. Để làm được điều này thì toàn bộ Backend cần phải được xây dựng lại từ đầu.
- **Nền tảng không hoạt động nhiều quốc gia:** Firestore chính là một Subdomain của Google. Trang web chính thức của Firestore hiện tại đang bị chặn ở nhiều quốc gia trên thế giới, trong đó có cả Trung Quốc.
- **Firestore chỉ hoạt động với CSDL NoSQL:** Người dùng sẽ không thể xử lý được dữ liệu một cách nhanh chóng. Firestore chỉ sử dụng JSON và hầu như không có SQL. Chính vì vậy, để di chuyển từ cơ sở dữ liệu sẽ không hề dễ dàng.
- **Firestore chỉ chạy trên Google Cloud:** Firestore trở thành một phần của Google và tất cả cơ sở hạ tầng của ứng dụng đều hoạt động trên Google Cloud. Người dùng không thể chạy ứng dụng trên những đơn vị cung cấp đám mây khác.
- **Truy vấn kém hiệu quả:** Không phải các cơ sở dữ liệu đều phù hợp cho mọi trường hợp và tất nhiên Cloud Firestore cũng không là ngoại lệ. Điều này khiến bạn rất mất thời gian khi sử dụng ứng dụng.
- **Các dịch vụ cung cấp không phải đều miễn phí:** Không phải tất cả các dịch vụ được cung cấp đều miễn phí. Các chức năng trên đám mây sẽ chỉ có thể khả dụng trên gói Blaze và dĩ nhiên người dùng cũng không thể sử dụng được các dịch vụ ở trong gói Spark.
- **Giá sử dụng dịch vụ Firestore khá cao:** Firestore sẽ cung cấp cho người dùng rất nhiều tính năng hữu ích nhưng giá thành của chúng cũng cao hơn so với nhà cung cấp IaaS thuần túy.
- **Thiếu hợp đồng doanh nghiệp:** Firestore không có những tùy chọn Dedicated Servers hoặc các hợp đồng doanh nghiệp. Để có thể sử dụng được Firestore chính là sử dụng cấu trúc Serverless ít linh hoạt hơn.
- **Không cung cấp API GraphQL:** Ứng dụng không cung cấp API

GraphQL tương tự như một phần của quá trình thiết lập tiêu chuẩn. Mặc dù còn có những giải pháp thay thế khác thế nhưng REST vẫn là một tùy chọn mặc định của nền tảng này.

PHẦN 3: PHÂN TÍCH, THIẾT KẾ, PHÁT TRIỂN DỰ ÁN

1. Biểu đồ Use case.

2. Các use case chính

- Use case đăng nhập
- Use case đăng ký
- Use case đăng xuất
- Use case xem trang cá nhân
- Use case cập nhật trang cá nhân
- Use case đăng bài viết
- Use case xem bài viết theo bộ lọc và tìm kiếm
- Use case xem chi tiết bài đăng
- Use case admin xóa bài đăng
- Use case admin duyệt bài đăng
- Use case cập nhật trang cá nhân

2.1 Mô tả chi tiết các use case

2.2.1 Mô tả use case Cập nhật trang cá nhân (Phạm Trí Nghĩa)

Mô tả vắn tắt: Use case này cho phép người dùng chỉnh sửa thông tin cá nhân được lưu trên trang web

- Luồng sự kiện

- o Luồng cơ bản

- Use case này bắt đầu khi người dùng nhấn vào nút “Sửa thông tin” trên form thông tin tài khoản. Hệ thống sẽ lấy thông tin tài khoản từ bảng Account và hiển thị form thông tin tài khoản để người dùng cập nhật.

- Người dùng cập nhật xong thông tin cá nhân và nhấn vào nút “Lưu”. Hệ thống kiểm tra tính đúng đắn của dữ liệu và gửi lên server.

- Server kiểm tra quyền của người dùng và cập nhật lại thông tin của tài khoản.

- Hệ thống chuyển hướng đến trang thông tin tài khoản và use case kết thúc.

- o Luồng rẽ nhánh

- Tại bước 1 trong luồng cơ bản, nếu tài khoản đã bị xóa trong cơ sở dữ liệu thì hệ thống sẽ không tìm thấy tài khoản và đưa ra thông báo “Tài khoản của bạn không tồn tại” và thực hiện đăng xuất tài khoản, use case kết thúc.

- Tại bước 2 trong luồng cơ bản, nếu người dùng bỏ trống một số trường bắt buộc hoặc nhập thông tin với độ dài là quá dài thì hệ thống sẽ thông báo lỗi và yêu cầu người dùng nhập lại.

- Tại bước 3 trong luồng cơ bản, nếu server kiểm tra thấy người dùng không có quyền chỉnh sửa thông tin thì hệ thống sẽ thông báo lỗi. Use case kết thúc.

- Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối

- **Các yêu cầu đặc biệt**

Không có.

- **Tiền điều kiện**

Người dùng phải đăng nhập rồi.

- **Hậu điều kiện**

Tài khoản đã được cập nhật thành công.

- **Điểm mở rộng**

Xem thông tin tài khoản.

2.2.2 Mô tả use case Xem trang cá nhân(Phạm Trí Nghĩa)

Mô tả vắn tắt: Use case cho phép người dùng xem thông tin tài khoản của mình.

- **Luồng sự kiện:**

- **Luồng cơ bản:**

- Use case này bắt đầu khi người dùng nhấn vào biểu tượng người dùng trên thanh công cụ, hệ thống sẽ hiển thị trang thông tin người dùng
 - Use case kết thúc.

- **Luồng rẽ nhánh:**

1. Tại bước 1 trong luồng cơ bản. Nếu tài khoản người dùng đã bị khóa hoặc người dùng chưa có tài khoản thì hệ thống sẽ đưa ra thông báo không tìm thấy tài khoản và thực hiện đăng xuất.
2. Use case kết thúc..

- **Các yêu cầu đặc biệt:**

Không có

- **Tiền điều kiện:**

Người dùng đã đăng nhập vào trang web.

- **Hậu điều kiện:**

Hiển thị được trang thông tin người dùng.

- **Điểm mở rộng:**

Cập nhật trang cá nhân.

2.2.3 Mô tả use case Admin duyệt bài đăng(Nguyễn Thị Phương)

Mô tả vắn tắt: Use case này cho phép quản trị viên duyệt bài đăng của người dùng.

- **Luồng sự kiện**

- o Luồng cơ bản

- Use case này bắt đầu khi người quản lý kích vào hình ảnh của bài biết, hệ thống sẽ lấy thông tin của bài biết trong bản news và hiển thị lên màn hình.
- Người quản lý sẽ nhấn vào nút duyệt để duyệt bài viết, use case kết thúc

- o Luồng rẽ nhánh

- Tại bước 2 trong luồng cơ bản, nếu người quản lý nhấn nút “Xóa” thì hệ thống sẽ thực hiện xóa bài viết. Use case kết thúc.

- **Các yêu cầu đặc biệt**

Không có.

- **Tiền điều kiện**

Admin phải đăng nhập với tư cách người quản trị.

- **Hậu điều kiện**

Bài viết đã được duyệt.

- **Điểm mở rộng**

Xóa bài viết.

2.2.4 Mô tả use case Admin xóa bài đăng (Nguyễn Thị Phụng)

Mô tả vắn tắt: Use case này cho phép quản trị viên xóa một bài viết.

- **Luồng sự kiện**

- o Luồng cơ bản

- Use case này bắt đầu khi người quản trị nhấn vào nút xóa bài viết.
- Hệ thống hiển thị thông báo “Bạn có chắc chắn muốn xóa bài viết này?”
- Hệ thống sẽ gửi thông tin id của bài viết cần xóa lên api, server sẽ kiểm tra quyền của người dùng và thực hiện xóa.

- o Luồng rẽ nhánh

- Tại bước 2 trong luồng cơ bản, nếu người dùng chọn không thì hệ thống sẽ không làm gì cả và use case kết thúc.
- Tại bước 3 trong luồng cơ bản, nếu người dùng không đủ quyền thì hệ thống sẽ báo lỗi và use case kết thúc.
- Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với server thì hệ thống sẽ hiển thị thông báo lỗi, use case kết thúc.

- **Các yêu cầu đặc biệt**

Không có.

- **Tiền điều kiện**

Người quản trị phải đăng nhập với tư cách quản trị viên.

- **Hậu điều kiện**

Bài viết đã được xóa.

- **Điểm mở rộng**

Không có.

2.2.5 Mô tả use case Đăng bài viết (Nguyễn Thị Thương)

- **Mô tả vắn tắt:** Use case này cho phép người dùng có thể đăng bài tìm đồ hoặc đăng bài tìm được đồ.

- **Luồng sự kiện:**

- **Luồng cơ bản:**

- Use case này bắt đầu khi người dùng nhấp vào nút “Đăng bài”, hệ thống sẽ kiểm tra quyền của người dùng và thực hiện hiển thị form đăng bài viết.
- Người dùng nhập thông tin bài viết, và nhấn nút “Lưu”.
- Hệ thống sẽ kiểm tra tính đúng đắn của dữ liệu và gửi lên server.
- Server kiểm tra tính đúng đắn của dữ liệu và thực hiện thêm một bản ghi trong bảng news.
- Hệ thống hiển thị thông báo thêm thành công, use case kết thúc.

- **Luồng rẽ nhánh:**

- Tại bước 1 trong luồng cơ bản, nếu người dùng chưa đăng nhập thì hệ thống sẽ hiển thị thông báo “Bạn phải đăng nhập để thực hiện tính năng này”, sau đó hiển thị màn hình đăng nhập để người dùng đăng nhập hoặc đăng ký. Use case kết thúc.
- Tại bước 3 trong luồng cơ bản, nếu người dùng không nhập các trường bắt buộc hoặc nhập quá dài thì hệ thống sẽ báo lỗi và người dùng nhập lại thông tin, use case kết thúc.
- Tại bước 4 trong luồng cơ bản, nếu bài viết đã tồn tại trong hệ thống, hệ thống sẽ hiển thị thông báo lỗi và người dùng nhập lại thông tin.
- Tại bất kỳ bước nào trong luồng cơ bản, nếu không kết nối được với luồng

cơ bản thì hệ thống sẽ hiển thị thông báo lỗi và use case kết thúc.

- **Các yêu cầu đặc biệt:**

- Không có.

- **Tiền điều kiện:**

- Người dùng phải đã đăng nhập

- **Hậu điều kiện:**

- Bài viết mới được đăng

- **Điểm mở rộng:**

- Không có.

2.2.6 Mô tả use case Xem bài viết theo bộ lọc và tìm kiếm (Nguyễn Thị Thương)

- **Mô tả vắn tắt:** Use case này cho phép người dùng lọc và tìm kiếm bài viết.
- **Luồng sự kiện:**
 - **Luồng cơ bản:**
 - Use case này bắt đầu khi người dùng nhập từ khóa cần tìm kiếm hoặc nhập các thông tin cho bộ lọc.
 - Hệ thống sẽ kiểm tra tính hợp lệ của từ khóa, ví dụ như độ dài, nếu vi phạm thì hệ thống sẽ báo lỗi, use case kết thúc.
 - Hệ thống gửi thông tin về từ khóa và bộ lọc lên server, server lọc bài viết theo bộ lọc và từ khóa.
 - Hệ thống hiển thị danh sách bài viết, use case kết thúc
 - **Luồng rẽ nhánh:**
 - Tại bước 2 trong luồng rẽ nhánh, nếu người dùng nhập từ khóa quá dài, hệ thống sẽ hiển thị thông báo “Không thể nhập từ khóa tìm kiếm dài hơn 255 ký tự”.
 - Tại bước 3 trong luồng cơ bản, nếu người dùng không có quyền, hệ thống sẽ hiển thị thông báo “Tài khoản của bạn không có quyền” và chuyển hướng về trang đăng nhập.
 - Tại bước 4 trong luồng cơ bản, nếu không có bài viết nào thì hệ thống sẽ hiển thị thông báo “Không có bài viết nào”
 - Tại bất kỳ bước nào trong luồng cơ bản, nếu không thể kết nối với server thì hệ thống sẽ báo lỗi và use case kết thúc
- **Các yêu cầu đặc biệt:**
 - Không có.
- **Tiền điều kiện:**
 - Không có.
- **Hậu điều kiện:**

- Hiện thị được danh sách

- **Điểm mở rộng:**

- Xem chi tiết bài viết

2.2.7 Mô tả UseCase Xem chi tiết bài đăng (Nguyễn Thị Thương)

- Mô tả vắn tắt: Use case này cho phép người dùng đăng bài

- Luồng sự kiện:

- **Luồng cơ bản:**

- Use case này bắt đầu khi người dùng nhấp kích vào hình ảnh bài đăng, hệ thống sẽ lấy thông tin chi tiết của bài đăng từ bảng news sau đó hiển thị lên màn hình cho người dùng. Use case kết thúc.

- **Các yêu cầu đặc biệt:**
 - Không có.
- **Tiền điều kiện:**
 - Không có.
- **Hậu điều kiện:**
 - Hiện thị được thông tin chi tiết của bài đăng
- **Điểm mở rộng:**
 - Yêu cầu xóa bài đăng.

2.2.8 Mô tả UseCase Đăng nhập (Nguyễn Công Thành)

- **Mô tả vắn tắt:** Use case này cho phép người dùng đăng nhập với tư cách người dùng hoặc quản trị viên.
- **Luồng sự kiện:**

○ **Luồng cơ bản:**

1. - Người dùng nhấn vào biểu tượng đăng nhập và nhập thông tin đăng nhập vào form đăng nhập sau đó nhấn nút “Đăng nhập”.
2. - Người dùng cũng có thể chọn đăng nhập bằng tài khoản gmail để đăng nhập.
3. - Hệ thống sẽ kiểm tra tài khoản mật khẩu của người dùng và xác nhận quyền của người dùng, nếu người dùng có quyền đăng nhập thì hệ thống sẽ cho đăng nhập với quyền người dùng hoặc quản trị viên.
4. - Sau khi đăng nhập thành công hệ thống sẽ thông báo đăng nhập thành công và kết thúc use case

○ **Luồng rẽ nhánh:**

- Tại bước 1 trong luồng cơ bản, nếu người dùng để trống tài khoản mật khẩu hoặc nhập mật khẩu ngắn hơn 6 ký tự thì frontend sẽ tự động thông báo lỗi và không gửi request lên server.
- Tại bước 2 trong luồng cơ bản, nếu tài khoản không đúng tên đăng nhập hoặc mật khẩu thì hệ thống sẽ báo lỗi “Tài khoản hoặc mật khẩu không đúng”.
- Tại bước 2 trong luồng cơ bản, nếu tài khoản không có quyền truy cập do đã bị vô hiệu hóa thì hệ thống sẽ thông báo “Tài khoản của bạn không có quyền truy cập”.

● **Các yêu cầu đặc biệt:**

- Không có.

● **Tiền điều kiện:**

- Người dùng phải có tài khoản để đăng nhập

● **Hậu điều kiện:**

- Người dùng có thể đăng nhập vào trang web

- **Điểm mở rộng:**

- Đăng ký

2.2.9 Mô tả use case Đăng ký (Nguyễn Công Thành)

Mô tả vắn tắt: Use case này cho phép người dùng có thể đăng ký tài khoản để đăng nhập vào trang web.

- **Luồng sự kiện:**

- **Luồng cơ bản:**

- Người dùng nhấn vào nút “Đăng ký”, hệ thống sẽ hiển thị form nhập thông tin tài khoản và mật khẩu.
- Người dùng nhập thông tin tài khoản và mật khẩu, sau đó nhấn nút đăng ký.
- Hệ thống kiểm tra thông tin người dùng nhập có hợp lệ không, sau đó hệ thống gửi thông tin lên server để đăng ký.
- Hệ thống tạo xong tài khoản và chuyển hướng người dùng về trang chủ, use case kết thúc.

○ **Luồng rẽ nhánh:**

- Ở bước 2 trong luồng cơ bản, nếu người dùng không nhập đầy đủ thông tin, hệ thống sẽ hiển thị thông báo “Không được để trống tài khoản và mật khẩu”.
- Ở bước 3 trong luồng cơ bản, nếu Tài khoản đã tồn tại trong hệ thống, hệ thống sẽ báo lỗi “Tài khoản này đã tồn tại, vui lòng chọn tài khoản mới”.

● **Các yêu cầu đặc biệt:**

Không có

● **Tiền điều kiện:**

Không có

● **Hậu điều kiện:**

Người dùng tạo được tài khoản mới

● **Điểm mở rộng:**

Đăng nhập.

Biểu đồ lớp

3: Thiết kế, xây dựng, phát triển dự án

1. Tìm hiểu cách hoạt động của VueJS

1.1 Cách hoạt động của VueJS

VueJS là một framework JavaScript được sử dụng để xây dựng giao diện người dùng (UI) tương tác trong các ứng dụng web. Nó hoạt động dựa trên mô hình MVVM (Model-View-ViewModel) và sử dụng cơ chế two-way data binding để đồng bộ hóa dữ liệu giữa các thành phần trong ứng dụng.

Dưới đây là cách hoạt động cơ bản của VueJS:

Khởi tạo Vue instance: Đầu tiên, bạn cần tạo một đối tượng Vue instance bằng cách sử dụng constructor của Vue. Đối tượng này sẽ đại diện cho toàn bộ ứng dụng Vue và quản lý các thành phần và dữ liệu của nó.

Liên kết dữ liệu: Sau khi tạo Vue instance, bạn cần liên kết dữ liệu của ứng dụng với các thành phần Vue. VueJS sử dụng cú pháp đặc biệt (v-bind và v-model) để liên kết dữ liệu từ đối tượng Vue instance đến các thuộc tính và trạng thái của thành phần.

Xử lý sự kiện: VueJS cung cấp cú pháp (v-on) để xử lý sự kiện từ người dùng. Bạn có thể gắn các hàm xử lý sự kiện vào các thành phần và VueJS sẽ tự động gọi chúng khi sự kiện xảy ra.

Render giao diện người dùng: VueJS sử dụng cú pháp đặc biệt (v-if, v-for) để tạo và điều khiển cách các thành phần được hiển thị trên giao diện người dùng. Bạn có thể xác định cấu trúc HTML và các điều kiện hiển thị dựa trên trạng thái của dữ liệu.

Theo dõi thay đổi dữ liệu: VueJS tự động theo dõi các thay đổi trong dữ liệu và cập nhật giao diện người dùng một cách tự động. Khi dữ liệu thay đổi, VueJS sẽ thông báo cho các thành phần liên quan và cập nhật chỉ mục DOM tương ứng.

Lifecycle hooks: VueJS cung cấp các lifecycle hooks để bạn có thể thực hiện

các hành động tại các giai đoạn khác nhau trong vòng đời của một thành phần Vue. Ví dụ: created, mounted, updated, destroyed.

1.2 Cách build 1 dự án VueJS đơn giản

Để xây dựng một dự án đơn giản với VueJS, cần thực hiện các bước sau:

1. Tạo một thư mục mới cho dự án của bạn.
2. Mở terminal và chạy lệnh sau để tạo một file package.json:

```
csharp Copy code  
npm init -y
```

3. Cài đặt ReactJS và các thư viện phụ thuộc bằng cách chạy lệnh:

```
Copy code  
npm install react react-dom
```

4. Tạo một file index.html trong thư mục gốc của dự án và thêm mã HTML cơ bản sau:

```
html Copy code  
<!DOCTYPE html>  
<html>  
  <head>  
    <title>My React App</title>  
  </head>  
  <body>  
    <div id="root"></div>  
    <script src="./src/index.js"></script>  
  </body>  
</html>
```

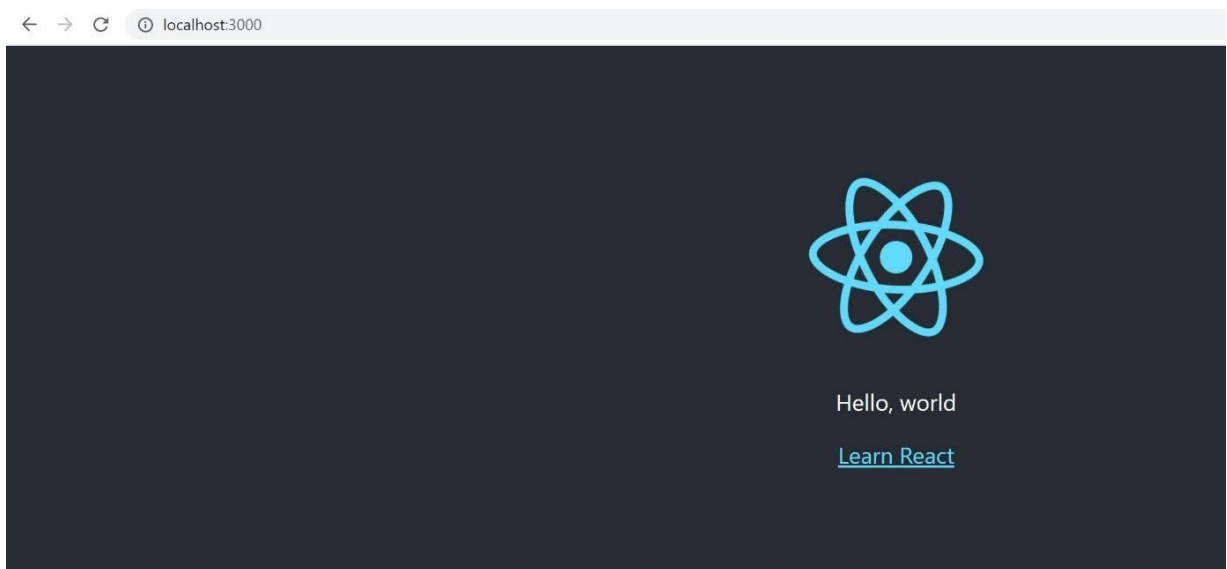

5. Tạo thư mục "src" và trong đó tạo một file index.js. Trong file index.js, thêm mã sau để hiển thị một "Hello World" message:

```
javascript Copy code  
  
import React from "react";  
import ReactDOM from "react-dom";  
  
ReactDOM.render(  
  <h1>Hello World</h1>,  
  document.getElementById("root")  
)
```

6. Chạy lệnh sau trong terminal để biên dịch mã ReactJS thành mã JavaScript:

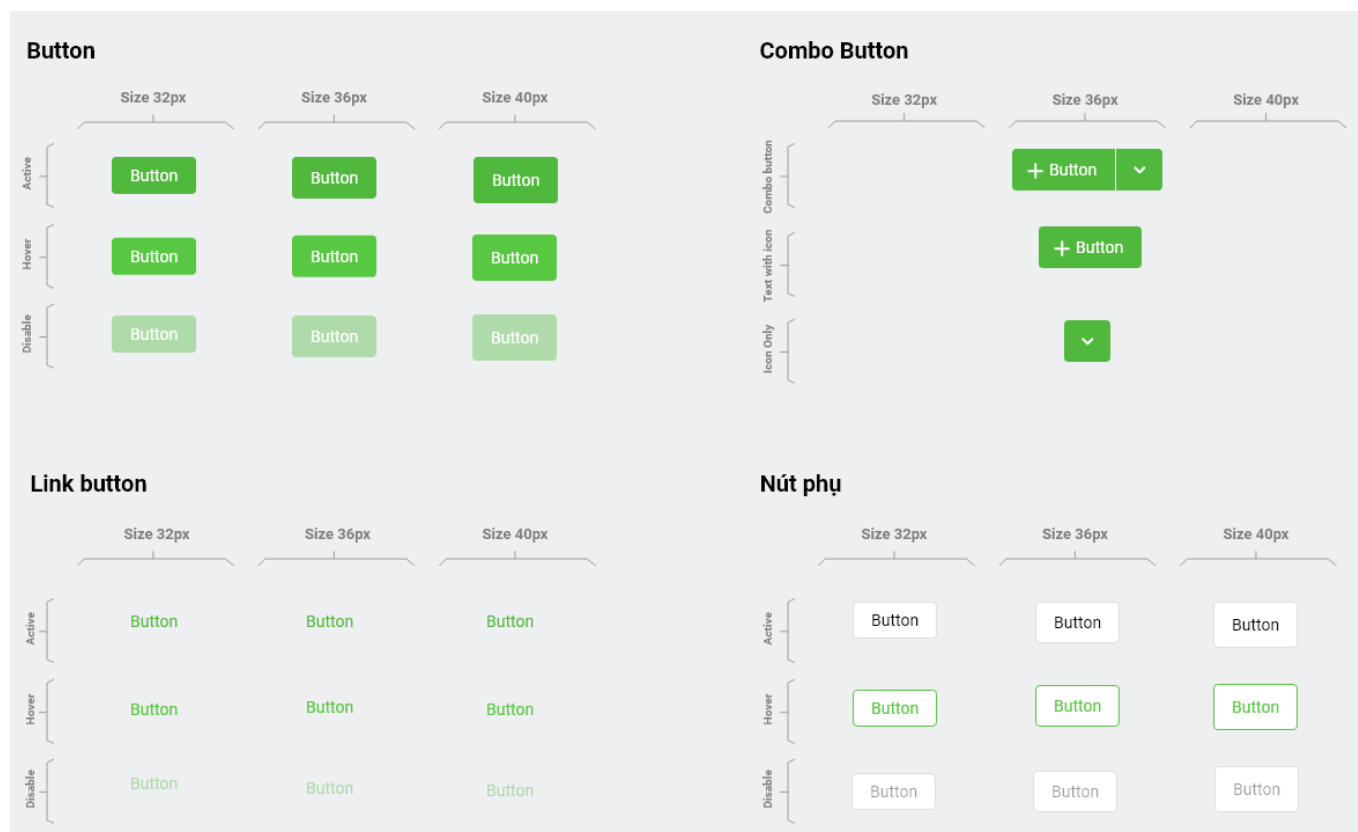
```
CSS Copy code  
npx babel src --out-dir public --presets react-app/prod
```

7. Mở file index.html và bạn sẽ thấy "Hello World" message được hiển thị trên trang web.

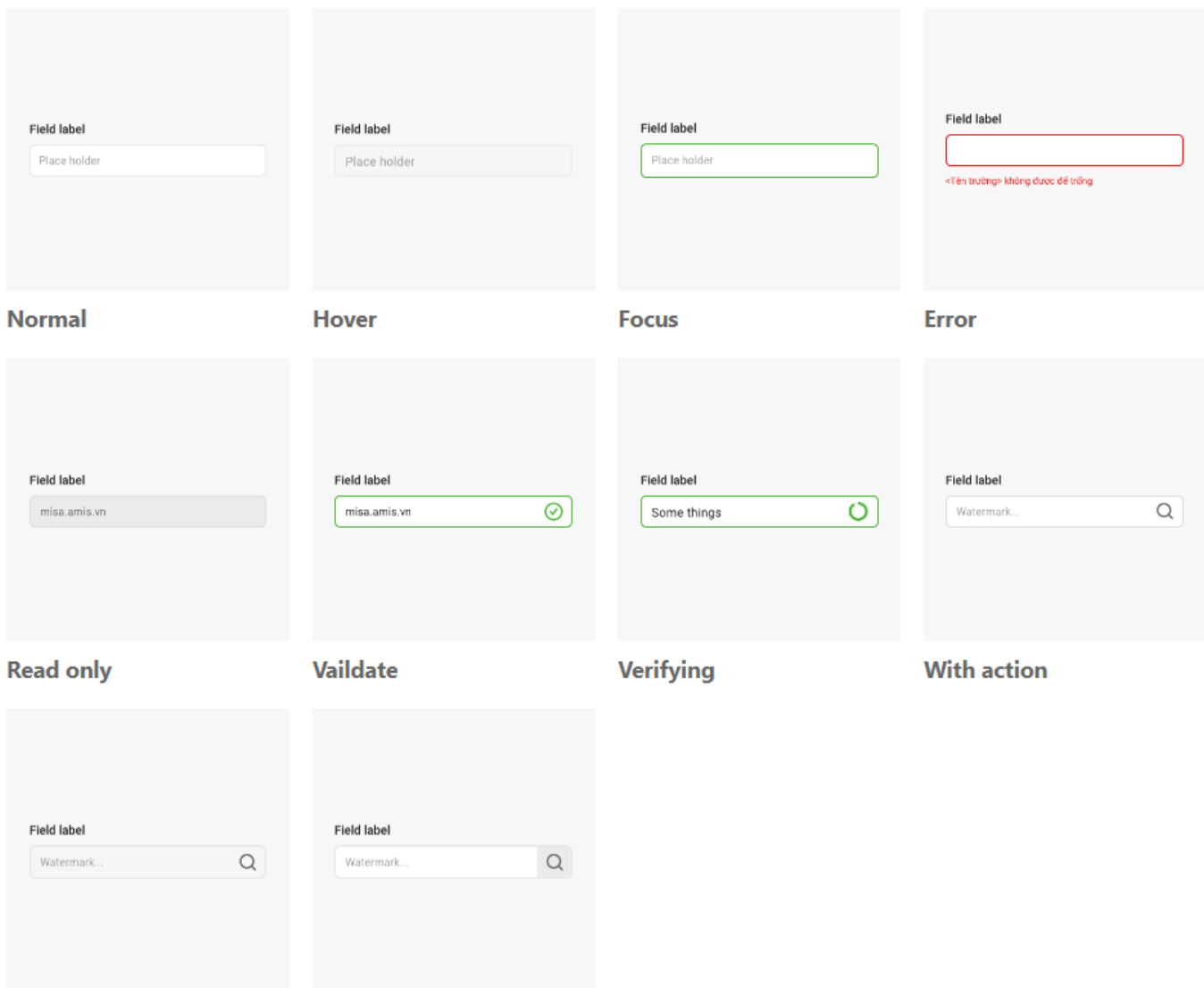


2. Thiết kế giao diện các màn hình trang web theo đề tài

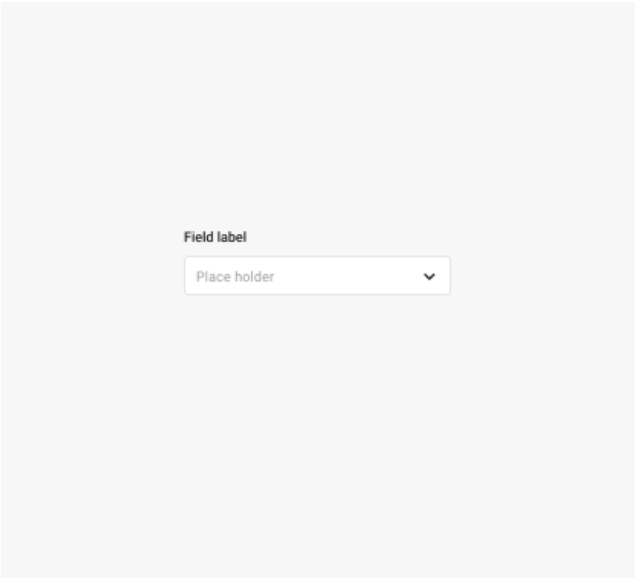
2.1. Thiết kế nút ấn



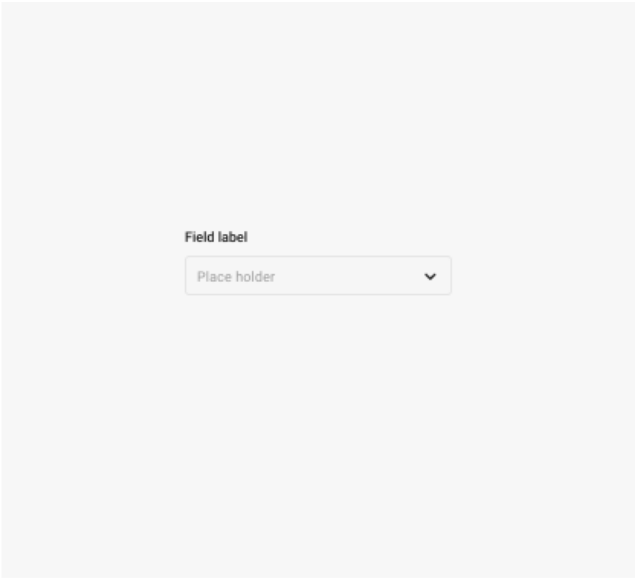
2.2. Thiết kế các trạng thái của text field.



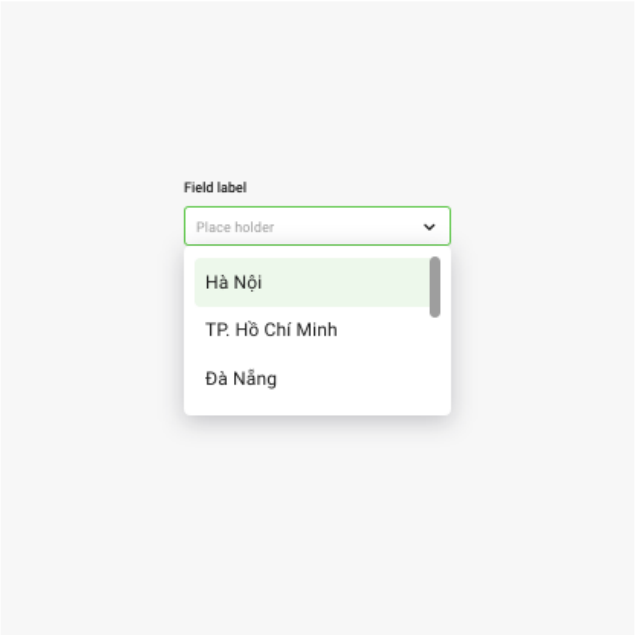
2.3. Thiết kế dropdown list.



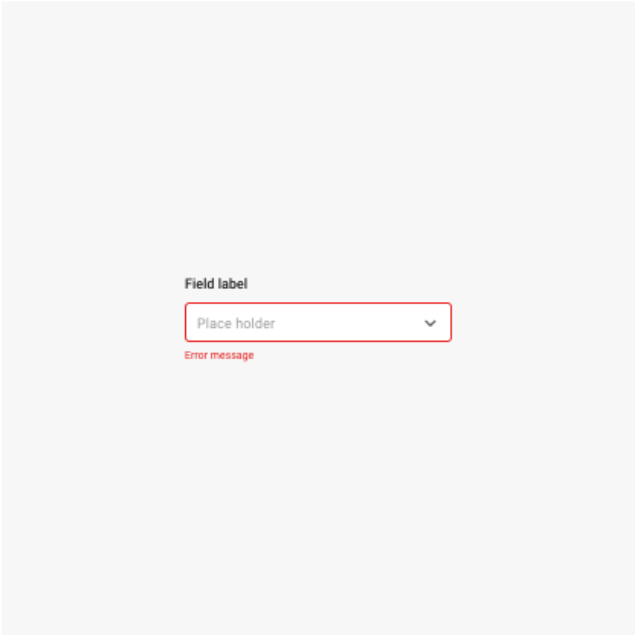
Normal



Hover



Show option



Error

Drop down

	Default	Thu gọn
Size 32px	<div><div>Field label</div><div><div>- Không chọn -</div><div><div>- Không chọn -</div><div>Hà Nội</div><div>TP. Hồ Chí Minh</div><div>Đà Nẵng</div><div>Hải Phòng</div><div>Hải Dương</div></div></div></div>	<div><div>Field label</div><div><div>Place holder</div></div></div>
Size 36px	<div><div>Field label</div><div><div>- Không chọn -</div><div><div>- Không chọn -</div><div>Hà Nội</div><div>TP. Hồ Chí Minh</div><div>Đà Nẵng</div><div>Hải Phòng</div><div>Hải Dương</div></div></div></div>	<div><div>Field label</div><div><div>Place holder</div></div></div>
Size 40px	<div><div>Field label</div><div><div>- Không chọn -</div><div><div>- Không chọn -</div><div>Hà Nội</div><div>TP. Hồ Chí Minh</div><div>Đà Nẵng</div><div>Hải Phòng</div><div>Hải Dương</div></div></div></div>	<div><div>Field label</div><div><div>Place holder</div></div></div>

Màn hình tìm danh sách bài viết

Timdothatlac.vn
Nhập tên trong giấy tờ, tên đồ vật, tên vật nuôi.
🔍

Bộ lọc

Xóa bộ lọc

Loại tìm

☐ Tin cần tìm

☐ Tin nhất được

Danh mục

☒ Vì Giấy tờ

☐ Thú cưng (Chó/Mèo)

☐ Tìm người

☐ Điện thoại/Tablet/Laptop

☐ Chìa khóa

☐ Xe máy/Ô tô

☐ Đồ vật khác

Chọn khu vực

Bà Rịa - Vũng Tàu ▼

Tìm kiếm

 Rời ví và giấy tờ tên Minh Hoàng ♡ Bà Rịa - Vũng Tàu Tin nhất được 5.7k	 Rời giấy tờ mang tên Nguyễn Đức Thuận ♡ Bà Rịa - Vũng Tàu Tin nhất được 597	 Tìm vì Hoàng Ngọc Minh Phương ở Thái Thi... ♡ Bà Rịa - Vũng Tàu Tin nhất được 2.2k	 Mình bị rời đăng ký xe tên Vũ Thị Hồng L... ♡ Bà Rịa - Vũng Tàu Tin nhất được 1k	 Rời giấy tờ mang tên Nguyễn Đức Thuận ♡ Bà Rịa - Vũng Tàu Tin cần tìm 7.1k
 Tìm vì Hoàng Ngọc Minh Phương ở Thái Thi... ♡ Bà Rịa - Vũng Tàu Tin nhất được 3.2k	 Rời sổ hộ khẩu mang tên chú họ Lương Ngụ... ♡ Bà Rịa - Vũng Tàu Tin nhất được 9	 Rời giấy tờ tên Trương Quang Hiếu tại P... ♡ Bà Rịa - Vũng Tàu Tin nhất được 1.6k	 Tìm vì Hoàng Ngọc Minh Phương ở Thái Thi... ♡ Bà Rịa - Vũng Tàu Tin cần tìm 544	 Rời một ví dài màu đen ạ.trong đó có gá... ♡ Bà Rịa - Vũng Tàu Tin cần tìm 678
 Rời ví và giấy tờ mang tên Huỳnh Duy Tha... ♡ Bà Rịa - Vũng Tàu Tin cần tìm 6.1k	 Rời ví và giấy tờ mang tên Huỳnh Duy Tha... ♡ Bà Rịa - Vũng Tàu Tin cần tìm 6.4k	 Rời ví có giấy tên Nguyễn Kim Thoa ở đườ... ♡ Bà Rịa - Vũng Tàu Tin cần tìm 6.6k	 Rời giấy tờ mang tên CAO VĂN CƯỜNG ♡ Bà Rịa - Vũng Tàu Tin nhất được 2.6k	 Rời ví có giấy tên Nguyễn Kim Thoa ở đườ... ♡ Bà Rịa - Vũng Tàu Tin nhất được 229

Màn hình đăng nhập

Timdothatlac.vn

Nhập tên trong giấy tờ, tên đồ vật, tên vật nuôi ...

Đăng nhập

×

Timdothatlac.vn

Email *

Mật khẩu *

Đăng nhập

Hoặc

Đăng ký

Đăng nhập bằng Google

Màn hình đăng ký

Đăng nhập



Timdothatlac.vn

Email *

Mật khẩu *

Đăng ký

Hoặc

Đăng nhập

Đăng nhập bằng Google

Màn hình thông người dùng

Thông tin tài khoản

Họ và tên

Lê Đức Tiếp

Số điện thoại

0988888888

Giới tính

☐ Nam☒ Nữ

Năm sinh

2002

Địa chỉ

Số 123, Nguyễn Xá, Nhón, Hà Nội

Chỉnh sửa

Màn hình sửa thông tin người dùng

Timdothatlac.vn

Nhập tên trong giấy tờ, tên đồ vật, tên vật nuôi ...

Thông tin tài khoản

Họ và tên

Lê Đức Tiếp

Số điện thoại

09888888888

Giới tính

☐ Nam ☒ Nữ

Năm sinh

2002

Địa chỉ

Số 123, Nguyễn Xá, Nhón, Hà Nội

Lưu

Màn hình xem chi tiết một bài viết

Timdothatlac.vn

Nhập tên trong giấy tờ, tên đồ vật, tên vật nuôi ...

Lê Đức Tiếp

14/8/2023

Tin cần tìm


Thú cưng(Chó/Mèo)

Rơi giấy tờ mang tên Nguyễn Đức Thuận

- tiền mất và một số giấy tờ quan trọng khác tất cả mang tên: Đỗ Thị Văn Anh \n Thứ 2 ngày 28/3 vào khoảng 23h15 trước cửa CircleK 366 Nguyễn Trãi, mình có đánh rơi 1 cái ví màu đen trong đó



Nếu nhặt được xin liên hệ 0349834837.

Đà Nẵng



Màn hình bình luận

0 bình luận 6 lượt xem

Share:  

Bình luận

Gửi

Màn hình admin

Timdothatlac.vn									
Nhập tên trong giấy tờ, tên đồ vật, tên vật nuôi ...									
Đã chọn 4									
Bỏ chọn									
Duyệt									
Án									
Nhập theo mã, họ tên									
Tiện ích									
	TRẠNG THÁI	TİEU ĐỀ	LOẠI TIN	NỘI DUNG	THÔNG TIN LIÊN HỆ	THÀNH PHỐ	HUYỆN	XÃ	CHỨC NĂNG
<input type="checkbox"/>	Án	Tôi bị mất một chiếc bóp, ai tìm thấy ...	Tin cần tìm	- Bị mất hôm trước vào buổi đi chơi về...	- Số điện thoại liên hệ: 0368542562	Bắc Giang	Huyện Yên Dũng	Thị trấn Nam Điền	Xem
<input checked="" type="checkbox"/>	Đăng tin	Tôi bị rơi giấy tờ tên Lê Đức Tiếp	Tin cần tìm	- Rơi ở phố Nhân	- Vui lòng liên hệ số điện thoại 03215...	An Giang			Hiện Xóa
<input checked="" type="checkbox"/>	Đăng tin	Tôi nhặt được một con mèo	Tin nhặt được						Xem
<input checked="" type="checkbox"/>	Đăng tin	Mất Balo và giấy tờ mang tên Trần Dũng...	Tin cần tìm	Tên tôi là : Đoàn Duy Tiến sinh năm 1...	Vào lúc 13h10-13h30 ngày 23/3/2022, mi...	Hà Nội			Xem
<input checked="" type="checkbox"/>	Đăng tin	Rơi ví và giấy tờ tên Minh Hoàng	Tin nhặt được	- nhiều giấy tờ quan trọng có tên NGU...	Số 0932277239 có bị mất bóp da nâu cá...	Đắk Nông			Xem
<input type="checkbox"/>	Đăng tin	Tha thiết tìm lại mèo màu đen trắng bị...	Tin nhặt được	Hôm nay tầm 7h15 em bị mất ví trên đoa...	Khu vực Dragon Fitness (quán cà phê Ôn...	Bắc Ninh			Xem
<input type="checkbox"/>	Đăng tin	Tìm ví Hoàng Ngọc Minh Phương ở Thái T...	Tin nhặt được	Hôm nay tầm 7h15 em bị mất ví trên đoa...	Xin cảm ơn và hậu tạ	Quảng Ngãi			Xem
<input type="checkbox"/>	Đăng tin	Rơi giấy tờ mang tên Nguyễn Đức Thuận	Tin cần tìm	- tiền mặt và một số giấy tờ quan tro...	Nếu nhặt được xin liên hệ 0349834837.	Đà Nẵng			Xem
<input type="checkbox"/>	Đăng tin	Rơi ví có giấy tờ Nguyễn Kim Thoa ở đ...	Tin nhặt được	chợ Mỹ Đình tôi có làm rơi một tập h...	0928312343	Đồng Nai			Xem
<input type="checkbox"/>	Đăng tin	Mất ví và giấy tờ tên Lê Quốc Bình ở V...	Tin nhặt được	Tên tôi là : Đoàn Duy Tiến sinh năm 1...	Quảng đường từ ngân hàng Sacombank đượ...	Đà Nẵng			Xem
<input type="checkbox"/>	Đăng tin	Rơi ví và giấy tờ mang tên Huỳnh Duy T...	Tin cần tìm	Hì vọng có người nhặt được cho tôi chu...	Họ Tên: Đoàn Duy Tiến	Long An			Xem
<input type="checkbox"/>	Đăng tin	Mình bị rơi đăng ký xe tên Vũ Thị Hồng...	Tin cần tìm	chợ Mỹ Đình tôi có làm rơi một tập h...	Bên trong có :	Hà Tĩnh			Xem
<input type="checkbox"/>	Đăng tin	Tìm chó đã bị bắt tối ngày 18/3/2022 s...	Tin nhặt được	- nhiều giấy tờ quan trọng có tên NGU...	Hoặc liên hệ: 0362130195	Lâm Đồng			Xem
<input type="checkbox"/>	Đăng tin	Rơi số hồ khẩu mang tên chủ hộ Lương N...	Tin nhặt được	Hì vọng có người nhặt được cho tôi chu...	Hoặc liên hệ: 0362130195	Bắc Ninh			Xem
<input type="checkbox"/>	Đăng tin	Rơi ví và giấy tờ tên Minh Hoàng	Tin cần tìm	chợ Mỹ Đình tôi có làm rơi một tập h...	Ái nhặt được chiếc ví đỏ trên đoạn đườ...	Cà Mau			Xem
Tổng số: 21.002 bản ghi									
Số bản ghi/trang: 20 20 - 21.002 bản ghi									

3. Xây dựng chương trình

- Link github sản phẩm:
<https://github.com/LeDucTiep/BaiTapLonNhom2.git>

4. Kiểm thử

Test case	Trạng thái
Đăng nhập điền đầy đủ thông tin nhưng sai mật khẩu	Test case passed
Đăng nhập nhưng không điền thông tin	Test case passed
Đăng nhập nhưng tài khoản không tồn tại	Test case passed
Thêm bài viết và điền đầy đủ thông tin	Test case passed
Thêm bài viết nhưng để trống thông tin được phép để trống	Test case passed
Thêm bài viết nhưng điền thiếu 1 trường bắt buộc	Test case passed
Thêm bài viết và có tải ảnh lên	Test case passed

Chỉnh sửa thông tin tài khoản đầy đủ thông tin và chỉnh sửa thành công	Test case passed
Thêm bài viết nhưng không điền đầy đủ thông tin	Test case passed
Thêm bài viết nhưng không tải thêm ảnh	Test case passed
Đăng tin: Chưa chọn thành phố không cho chọn huyện và phường	Test case passed
Chưa chọn huyện thì không cho chọn phường	Test case passed
Thực hiện tìm kiếm theo bộ lọc với thông tin điền đầy đủ	Test case passed
Chỉ lọc tin tức theo loại tin	Test case passed
Chỉ lọc tin tức theo danh mục	Test case passed
Chỉ lọc tin tức theo khu vực	Test case passed
Tìm kiếm với từ khóa bất kỳ thì hiển thị bài viết có tiêu đề hoặc thông tin liên lạc chứa từ khóa đó	Test case passed
Admin thực hiện xóa nhiều bài viết	Test case passed
Admin thực hiện duyệt nhiều bài viết	Test case passed
Admin thực hiện ẩn nhiều bài viết	Test case passed
Admin thực hiện duyệt một bài viết	Test case passed
Admin thực hiện xóa một bài viết	Test case passed
Admin thực hiện ẩn một bài viết	Test case passed
Phân trang và chuyển trang bài viết tại màn hình admin	Test case passed
Quản xuống tải thêm bài viết ở màn hình	Test case passed

danh sách bài viết	
Thực hiện lọc và tìm kiếm ở màn hình Admin	Test case passed
Thực hiện comment với nội dung là rỗng	Test case passed
Thực hiện comment với nội dung quá dài	Test case passed
Thực hiện comment với nội dung nhỏ hơn 255 ký tự	Test case passed
Nhấn chia sẻ lên Facebook thì tự động sẽ chuyển đến trang chia sẻ của Facebook	Test case passed

TỔNG KẾT

Qua bài tập lớn đã giúp sinh viên có cái nhìn tổng thể hơn về một dự án hoàn chỉnh và những kỹ năng như làm việc nhóm, kỹ năng đọc hiểu document để sử dụng các công nghệ mới nhất. Giúp sinh viên tiếp cận được với quy trình cơ bản để tạo ra sản phẩm.

Đến đây cũng đã kết thúc bài báo cáo bài tập lớn của nhóm về đề tài **"Xây dựng trang web "Tìm đồ thất lạc" bằng VueJS, Dapper và Firebase Authentication"** với sự đóng góp của các thành viên trong nhóm thực hiện. Nhìn lại quá trình thực hiện bài báo cáo này, nhóm thấy được những điểm mạnh và điểm yếu còn vướng mắc về kiến thức cũng như kỹ năng trình bày và tìm hiểu các tài liệu liên quan. Để rút kinh nghiệm cho những lần thực hiện sau và cải thiện chất lượng của bài báo cáo sau sửa đổi, nhóm đi đến vạch ra những ưu nhược điểm của cá nhân cũng như tập thể nhóm thực hiện đề tài về những gì nhóm làm, những gì nhóm đã đạt được, những gì còn thiếu sót và định hướng sẽ thực hiện trong tương lai.

Những điều nhóm còn thiếu sót, chưa thực hiện được. Tuy có những đóng góp tích cực nhằm hoàn thiện báo cáo, tuy nhiên không thể tránh khỏi những thiếu sót, hạn chế. Bên cạnh đó, việc thiếu dữ liệu là điều nhóm còn hạn chế. Bởi các thành viên nhóm còn đang là sinh viên, không đủ nguồn lực để xây dựng một bộ dữ liệu hoàn chỉnh, đảm bảo chất lượng

Cuối cùng với những đóng góp, cố gắng để hoàn thiện báo cáo, nhóm rất mong được sự góp ý chân thành, quý báu từ quý thầy cô và các bạn đọc. Nhóm xin chân thành cảm ơn!

Nhóm thực hiện đề tài!

Tài liệu tham khảo:

- <https://viblo.asia/p/tim-hieu-ve-vuejs-co-ban-p1-Do754bpWZM6>
- <https://vi.vuejs.org/v2/guide/>
- <https://viblo.asia/p/nhung-kien-thuc-vuejs-co-ban-p1-63vKjOpNZ2R>
- <https://glints.com/vn/blog/vuejs-la-gi/>
- <https://viblo.asia/p/vuejs-va-nhung-kien-thuc-can-biet-p1-ByEZkANg5Q0>
- <https://toidicode.com/hoc-vuejs-co-ban>
- <https://hoc.tv/hoc-vuejs>
- <https://viblo.asia/p/dapper-orm-awesome-extensions-for-net-developer-Qpmlcd07Zrd>
- <https://tedu.com.vn/khoa-hoc/thiet-ke-restful-api-voi-aspnet-core-va-dapper-orm-24.html>
- <https://nhasachtinhoc.blogspot.com/2020/11/chia-se-khoa-hoc-dapper-co-ban-den-nang-cao-8945.html>
- <https://hubm.edu.vn/dapper-c-la-gi-micro-orm-trong-net-hanh-trang-lap-trinh-blog>
- <https://firebase.google.com/docs/auth>
- https://pub.dev/packages/firebase_auth
- <https://viblo.asia/p/firebase-login-and-registration-authentication-aWj53VWol6m>
- <https://anonymystick.com/blog-developer/firebase-huong-dan-login-bang-tai-khoan-google-facebook-su-dung-firebase-tren-website-g67mjfqq>