

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

**--------------------**



**HỒ VĂN PHONG : 15110277**

**TRẦN QUANG MINH : 15110250**

Đề tài:

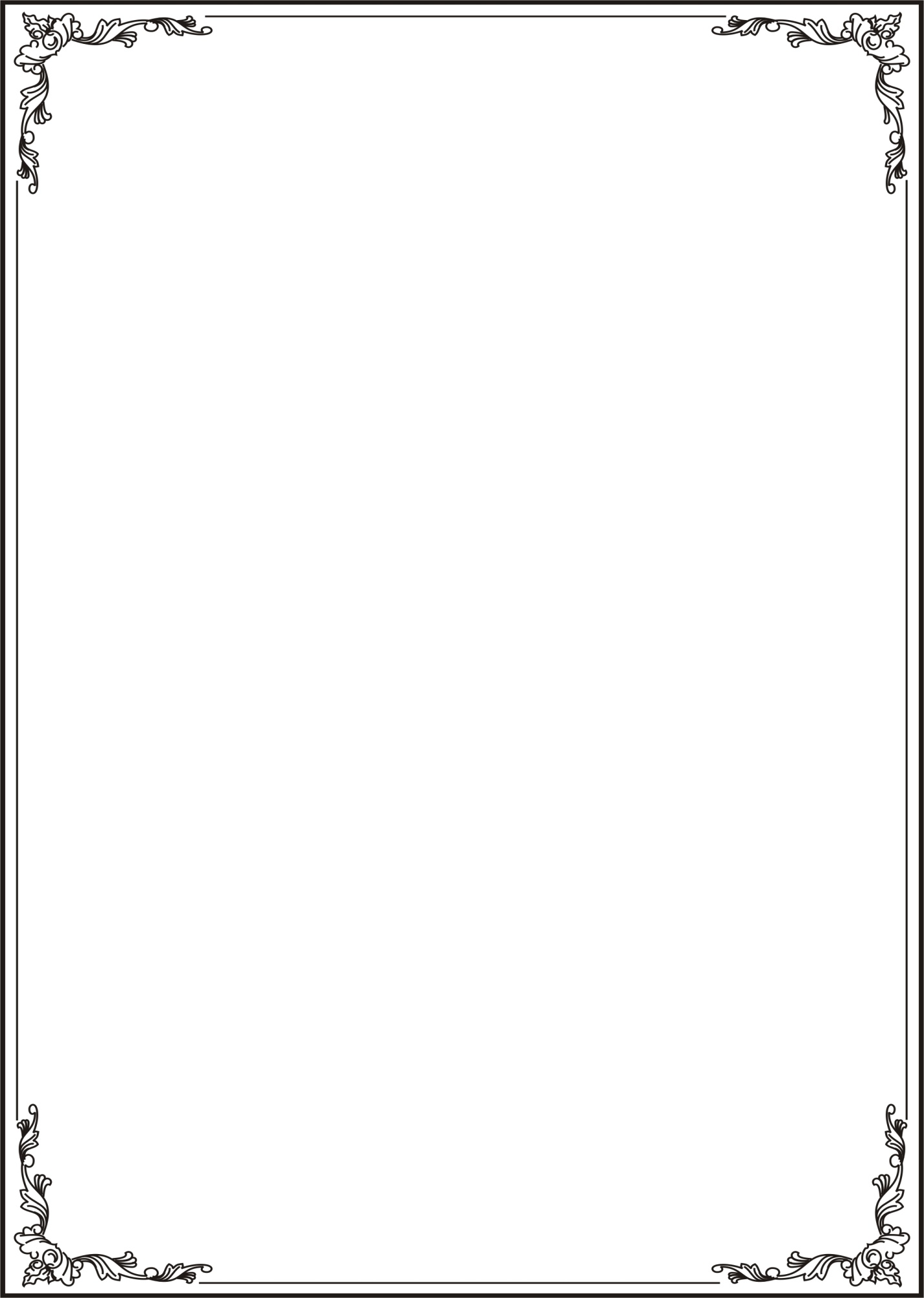
**TÌM HIỂU VÀ XÂY DỰNG PROGRESSIVE WEB APPLICATIONS**

**TIỂU LUẬN CHUYÊN NGÀNH HỆ THỐNG THÔNG TIN**

**GIÁO VIÊN HƯỚNG DẪN**

**THS. LÊ THỊ MINH CHÂU**

**KHOÁ 2015 - 2019**



**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**BỘ MÔN HỆ THỐNG THÔNG TIN**

**--------------------**



**HỒ VĂN PHONG : 15110277**

**TRẦN QUANG MINH : 15110250**

Đề tài:

**TÌM HIỂU VÀ XÂY DỰNG PROGRESSIVE WEB APPLICATIONS**

**TIỂU LUẬN CHUYÊN NGÀNH HỆ THỐNG THÔNG TIN**

**GIÁO VIÊN HƯỚNG DẪN**

**THS. LÊ THỊ MINH CHÂU**

**KHOÁ 2015 - 2019**

ĐH SƯ PHẠM KỸ THUẬT TP.HCM CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

**KHOA CÔNG NGHỆ THÔNG TIN Độc lập – Tự do – Hạnh phúc**

**------------ ------------**

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên 1: Trần Quang Minh MSSV: 15110250

Họ và tên Sinh viên 2: Hồ Văn Phong MSSV: 15110277

Ngành: Công nghệ Thông tin

Tên đề tài: Tìm hiểu và xây dựng Progressive Web Applications

Họ và tên Giáo viên hướng dẫn: Ths. Lê Thị Minh Châu

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

.....................................................................................................................................

.....................................................................................................................................

.....................................................................................................................................

1. Ưu điểm:

.....................................................................................................................................

.....................................................................................................................................

1. Khuyết điểm

.....................................................................................................................................

.....................................................................................................................................

1. Điểm :

*TP.HCM, ngày 20 tháng 12 năm 2018*

Giáo viên hướng dẫn

Ths. Lê Thị Minh Châu

ĐH SƯ PHẠM KỸ THUẬT TP.HCM CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

**KHOA CÔNG NGHỆ THÔNG TIN Độc lập – Tự do – Hạnh phúc**

**------------ ------------**

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

Họ và tên Sinh viên 1: Trần Quang Minh MSSV: 15110250

Họ và tên Sinh viên 1: Hồ Văn Phong MSSV: 15110277

Ngành: Công nghệ Thông tin

Tên đề tài: Tìm hiểu và xây dựng Progressive Web Applications

Họ và tên Giáo viên phản biện: Ths. Quách Đình Hoàng

**NHẬN XÉT**

1. Về nội dung đề tài & khối lượng thực hiện:

.....................................................................................................................................

.....................................................................................................................................

.....................................................................................................................................

1. Ưu điểm:

.....................................................................................................................................

.....................................................................................................................................

1. Khuyết điểm

.....................................................................................................................................

.....................................................................................................................................

1. Điểm :

*TP.HCM, ngày 20 tháng 12 năm 2018*

Giáo viên phản biện

Ths. Quách Đình Hoàng

**LỜI CẢM ƠN**

Nhóm em xin chân thành cám ơn khoa Công Nghệ Thông Tin, Bộ Môn Hệ Thống Thông Tin, trường Đại học Sư Phạm Kỹ Thuật TP.Hồ Chí Minh đã tạo điều kiện thuận lợi cho chúng em thực hiện đề tài này.

Chúng em cũng xin được gửi lời cám ơn chân thành nhất đến cô Lê Thị Minh Châu, người đã tận tình chỉ bảo và hướng dẫn nhóm em thực hiện đề tài này.

Bên cạnh đó, chúng em xin gửi lời cám ơn đến các thầy cô giảng viên của trường Đại Học Sư Phạm Kỹ Thuật TP.Hồ Chí Minh nói chung cũng như các thầy cô giảng viên khoa Công Nghệ Thông Tin nói riêng, những người đã giảng dạy, tạo điều kiện cho em tích lũy được những kiến thức quý báu trong những năm học qua.

Dù đã cố gắng hoàn thành đề tài tiểu luận đúng yêu cầu, nhưng do thời gian hạn hẹp và khả năng còn hạn chế nên chắc chắn sẽ không tránh khỏi thiếu sót. Chúng em mong nhận được sự thông cảm và tận tình chỉ bảo của các thầy cô và các bạn.

**Nhóm em xin chân thành cảm ơn**

**KẾ HOẠCH THỰC HIỆN**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Thời gian** | **Công Việc** | **Ghi chú** |
| **1** | Từ 30/09/2018  Đến 06/10/2018 | Tìm hiểu về công nghệ Progressive Web Applications. |  |
| **2** | Từ 07/10/2018  Đến 13/10/2018 | Tìm hiểu framework ReactJS. |  |
| **3** | Từ 14/10/2018  Đến 20/10/2018 | Tiếp tục tìm hiểu về framework ReactJS.  Tìm hiểu về NodeJS.  Tìm hiểu về MySQL. |  |
| **4** | Từ 21/10/2018  Đến 27/10/2018 |
| **5** | Từ 28/10/2018  Đến 03/11/2018 | Tìm hiểu cách cài đặt PWA với ReactJS. |  |
| **6** | Từ 04/11/2018  Đến 10/11/2018 | Khảo sát hiện trạng và phân tích yêu cầu. |  |
| **7** | Từ 11/11/2018  Đến 17/11/2018 | Thiết kế cơ sở dữ liệu.. |  |
| **8** | Từ 18/11/2018  Đến 24/11/2018 | Thiết kế giao diện. |  |
| **9** | Từ 25/11/2018  Đến 01/11/2018 | Triển khai ứng dụng. |  |
| **10** | Từ 02/12/2018  Đến 08/12/2018 | Hoàn thiện ứng dụng. |  |
| **11** | Từ 09/12/2018  Đến 20/12/2018 | Viết báo cáo tiểu luận. |  |

Ý kiến giảng viên hướng dẫn TP. Hồ Chí Minh, ngày…tháng … năm 2018

(Ký và ghi rõ họ tên) Người viết đề cương

MỤC LỤC

[ĐỀ CƯƠNG TIỂU LUẬN CHUYÊN NGÀNH 1](#_Toc533239089)

[DANH MỤC BẢNG BIỂU 2](#_Toc533239090)

[DANH MỤC HÌNH ẢNH 2](#_Toc533239091)

[DANH MỤC CÁC TỪ VIẾT TẮT 3](#_Toc533239092)

[MỞ ĐẦU 4](#_Toc533239093)

[1. GIỚI THIỆU ĐỀ TÀI 4](#_Toc533239094)

[2. TÍNH CẤP THIẾT CỦA ĐỀ TÀI 4](#_Toc533239095)

[3. MỤC TIÊU CỦA ĐỀ TÀI 4](#_Toc533239096)

[4. KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC 5](#_Toc533239097)

[NỘI DUNG 6](#_Toc533239098)

[Chương 1: CÁC CÔNG NGHỆ TÌM HIỂU 6](#_Toc533239099)

[1.1. Progressive Web Applications 6](#_Toc533239100)

[1.1.1. Khái niêm 6](#_Toc533239101)

[1.1.2. Một số yếu tố quan trọng 6](#_Toc533239102)

[1.1.3. Lợi ích 6](#_Toc533239103)

[1.1.4. Các công nghệ nổi bật 7](#_Toc533239104)

[1.2. Client-Side Storage 7](#_Toc533239105)

[1.2.1. Web storage 7](#_Toc533239106)

[1.2.2. IndexedDB 7](#_Toc533239107)

[1.3. Cache API 9](#_Toc533239108)

[1.2. Service Worker 11](#_Toc533239109)

[1.3. Notification API & Push Notification 14](#_Toc533239110)

[1.4. Responsive Web 15](#_Toc533239111)

[1.5. Web App Manifest 16](#_Toc533239112)

[1.8. ReactJS 17](#_Toc533239113)

[1.8.1. Giới thiệu 17](#_Toc533239114)

[1.8.2. Các khái niệm quan trọng 17](#_Toc533239115)

[1.8.2.1. Component 17](#_Toc533239119)

[1.8.2.2. JSX — Javascript Syntax Extension 18](#_Toc533239120)

[1.8.2.3. Props & State 19](#_Toc533239121)

[1.8.2.4.Thuộc tính key 19](#_Toc533239122)

[Chương 2: KHẢO SÁT HIỆN TRẠNG VÀ PHÂN TÍCH YÊU CẦU 20](#_Toc533239123)

[2.1. Khảo sát hiện trạng 20](#_Toc533239124)

[2.1.1. Forbes (https://www.forbes.com/) 20](#_Toc533239125)

[2.1.2. Báo mới (https://baomoi.com/) 21](#_Toc533239126)

[2.1.3. AliExpress (https://www.aliexpress.com/) 22](#_Toc533239127)

[2.1.4. Twitter Lite 22](#_Toc533239128)

[2.1.5 MakeMyTrip 23](#_Toc533239129)

[2.2. Phân tích yêu cầu 24](#_Toc533239130)

[2.2.1. Mô tả yêu cầu nghiệp vụ 24](#_Toc533239131)

[2.2.2. Usecase Diagram 25](#_Toc533239132)

[2.2.3. Bảng mô tả các use case 25](#_Toc533239133)

[2.2.4. Mô tả các actor 26](#_Toc533239134)

[Chương 3: THIẾT KẾ CƠ SỞ DỮ LIỆU 28](#_Toc533239135)

[3.1. Mô hình ERD 28](#_Toc533239136)

[3.2. Class Diagram 29](#_Toc533239137)

[3.3. Mô tả dữ liệu chi tiết 30](#_Toc533239138)

[3.3.1. Bảng mô tả các trường trong User 30](#_Toc533239139)

[3.3.2. Bảng mô tả các trường trong Post 31](#_Toc533239140)

[3.3.3. Bảng mô tả các trường trong Category 32](#_Toc533239141)

[3.3.4. Bảng mô tả các trường trong Clap 32](#_Toc533239142)

[3.3.5. Bảng mô tả các trường trong Bookmark 32](#_Toc533239143)

[3.3.6. Bảng mô tả các trường trong Subscriptions 33](#_Toc533239144)

[Chương 4: THIẾT KẾ GIAO DIỆN 34](#_Toc533239145)

[4.1. HomePage 34](#_Toc533239146)

[4.2. CategoryPage 35](#_Toc533239147)

[4.3. PostPage 35](#_Toc533239148)

[4.4. UserPage 36](#_Toc533239149)

[4.5. NewPostPage 36](#_Toc533239150)

[4.6. EditPostPage 37](#_Toc533239151)

[4.7. EditProfilePage 37](#_Toc533239152)

[Chương 5: CÀI ĐẶT 38](#_Toc533239153)

[5.1. NodeJS 38](#_Toc533239154)

[5.2. Hệ quản trị cơ sở dữ liệu MySQL 38](#_Toc533239155)

[5.3 Cấu hình Progressive Web Applications 39](#_Toc533239156)

[5.3.1. Cấu hình Manifest: 39](#_Toc533239157)

[5.3.2. Cấu hình Service Worker 41](#_Toc533239158)

[5.3.3. Cấu hình Cache API 45](#_Toc533239159)

[5.3.4. Upload bài viết ở chế độ BackgroundSync 47](#_Toc533239160)

[5.3.5. Cấu hình Bookmark người dùng 52](#_Toc533239161)

[5.3.6 Cấu hình cho người dùng nhận được thông báo khi có bài viết mới 55](#_Toc533239162)

[KẾT LUẬN 57](#_Toc533239163)

[1. KIẾN THỨC VÀ KINH NGHIỆM ĐẠT ĐƯỢC 57](#_Toc533239164)

[2. KHÓ KHĂN GẶP PHẢI 57](#_Toc533239165)

[3. ƯU ĐIỂM VÀ NHƯỢC ĐIẺM 57](#_Toc533239166)

[4. HƯỚNG PHÁT TRIỂN ĐỀ TÀI TRONG TƯƠNG LAI 58](#_Toc533239167)

[TÀI LIỆU THAM KHẢO 59](#_Toc533239168)

ĐH SƯ PHẠM KỸ THUẬT TP.HCM CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

**KHOA CÔNG NGHỆ THÔNG TIN Độc lập – Tự do – Hạnh phúc**

**------------ ------------**

# ĐỀ CƯƠNG TIỂU LUẬN CHUYÊN NGÀNH

Họ và tên: Trần Quang Minh MSSV: 15110250

Họ và tên: Hồ Văn Phong MSSV: 15110277

Ngành: Công nghệ Thông tin

Tên đề tài: Tìm hiểu và xây dựng Progressive Web Applications

Họ và tên Giáo viên hướng dẫn: Ths. Lê Thị Minh Châu

**Nội dung thực hiện:**

*Lý thuyết:*

- Tìm hiểu công nghệ Progressive Web Applications.

**-** Tìm hiểu một số framework hỗ trợ PWA: ReactJS.

*Thực hành:* Áp dụng xây dựng web app mô phỏng.

**Thời gian thực hiện:** Bắt đầu từ 30/9/2018 đến 22/12/2018

Chữ ký của SV:

Chữ ký của SV:

*TP.HCM, Ngày 20 Tháng 12 năm 2018.*

|  |  |  |
| --- | --- | --- |
| TRƯỞNG KHOA CNTT  (Ký và ghi rõ họ tên) |  | GIẢNG VIÊN HƯỚNG DẪN |
| (Ký và ghi rõ họ tên) |

# DANH MỤC BẢNG BIỂU

Bảng 1 Mô tả các bảng trong cơ sở dữ liệu 30

Bảng 2 Mô tả các trường trong User 31

Bảng 3 Mô tả các trường trong Post 31

Bảng 4 Mô tả các trường trong Category 32

Bảng 5 Mô tả các trường trong Clap 32

Bảng 6 Mô tả các trường trong Bookmark 32

Bảng 7 Mô tả các trường trong Subscriptions 33

Bảng 8 Danh mục các màn hình trong ứng dụng 34

# DANH MỤC HÌNH ẢNH

[Hình 1 Web Storage 7](#_Toc533238873)

[Hình 2 Khởi tạo IndexedDB 8](#_Toc533238874)

[Hình 3 Thêm data cho IndexedDB 8](#_Toc533238875)

[Hình 4 Xóa IndexDB 9](#_Toc533238876)

[Hình 5 Khai báo một Cache API 10](#_Toc533238877)

[Hình 6 Sử dụng Cache API để lưu trữ 11](#_Toc533238878)

[Hình 7 Service Worker Lifecycle 12](#_Toc533238879)

[Hình 8 Servive Worker Events 13](#_Toc533238880)

[Hình 9 Service Worker Fetch 14](#_Toc533238881)

[Hình 10 Kích thước màn hình 16](#_Toc533238882)

[Hình 11 React Component 17](#_Toc533238883)

[Hình 12 React Component Lifecycle 18](#_Toc533238884)

[Hình 13 React Prop & State 19](#_Toc533238885)

[Hình 14 React thuộc tính key 20](#_Toc533238886)

[Hình 15 Forbes (https://www.forbes.com/) 20](#_Toc533238887)

[Hình 16 Báo mới (https://baomoi.com/) 21](#_Toc533238888)

[Hình 17 AliExpress (https://www.aliexpress.com/) 22](#_Toc533238889)

[Hình 18 Twitter Lite 22](#_Toc533238890)

[Hình 19 MakeMyTrip 23](#_Toc533238891)

[Hình 20 Usecase Diagram 25](#_Toc533238892)

[Hình 21 Mô hình ERD 28](#_Toc533238893)

[Hình 22 Class Diagram 29](#_Toc533238894)

[Hình 23 HomePage 34](#_Toc533238895)

[Hình 24 CategoryPage 35](#_Toc533238896)

[Hình 25 PostPage 35](#_Toc533238897)

[Hình 26 UserPage 36](#_Toc533238898)

[Hình 27 NewPostPage 36](#_Toc533238899)

[Hình 28 EditPostPage 37](#_Toc533238900)

[Hình 29 EditProfilePage 37](#_Toc533238901)

[Hình 30 Cấu trúc thư mục ứng dụng 39](#_Toc533238902)

[Hình 31 Sơ đồ Cache dữ liệu 45](#_Toc533238903)

[Hình 32 Sơ đồ BackgroundSync 47](#_Toc533238904)

[Hình 33 Sơ đồ Push Notification 55](#_Toc533238905)

# DANH MỤC CÁC TỪ VIẾT TẮT

PWA – Progressive Web Applications

SW - service workers

# MỞ ĐẦU

## GIỚI THIỆU ĐỀ TÀI

Hiện nay, thế giới đang chuyển dịch vào thời đại cách mạng công nghiệp 4.0 mà trọng điểm là ngành công nghệ thông tin đang có những bước đi mạnh mẽ từng ngày. Việt Nam ta hội nhập để bắt kịp xu thế, các thiết bị công nghệ kết nối Internet (PC, Laptop, Smartphone…) ngày càng phổ biến rộng rãi. Vì thế nhu cầu truy cập web là không thể thiếu nên một ứng dụng web lý tưởng phải được tạo ra. **Một ứng dụng web có thể hoạt động được offline hoàn toàn, có thời gian tải trang gần như tức thì và chạy một cách linh hoạt ngay cả khi đường truyền mạng thiếu ổn định kèm với tính an toàn… cho người dùng.** Progressive Web Applications được sinh ra cho phép chúng ta **tận dụng được những tính năng mạnh mẽ mà trình duyệt đã cung cấp sẵn để xây dựng những ứng dụng web như trên. Tóm lại, việc phát triển PWA là một nhu cầu lớn.**

## TÍNH CẤP THIẾT CỦA ĐỀ TÀI

Một ứng dụng web có những tính năng tương tự như một app thật thụ: trong điện thoại di động, người dùng dành nhiều thời gian trong các App, chứ không phải là web. Chỉ có 13% thời gian điện thoại di động của người sử dụng là trên web di động, và 87% chi tiêu trong App (2015). App có một số tính năng tương tác lại tuyệt vời - như biểu tượng màn hình làm cho việc truy cập trở lại ứng dụng dễ dàng, và thông báo kết nối người dùng trở lại ngay cả khi ứng dụng được đóng lại. Với việc tận dụng tài nguyên trên trình duyệt cung cấp mang đến tốc độ tải vượt trội.

Bên cạnh đó, người dùng có thể nhận thông báo ngay cả khi ứng dụng web hiện không chạy và trình duyệt không mở.

Vì vậy, việc áp dụng PWA vào ứng dụng web là rất cần thiết.

## MỤC TIÊU CỦA ĐỀ TÀI

* Tìm hiểu về công nghệ Progressive Web Applications.
* Tìm hiểu về framework hỗ trợ Progressive Web Applications: ReactJS.
* Xây dựng một ứng dụng web áp dụng các tính năng của Progressive Web Applications.

## 4. KẾT QUẢ DỰ KIẾN ĐẠT ĐƯỢC

* Tìm hiểu về công nghệ Progressive Web Applications.
* Hiểu rõ cách vận hành của các tính năng của Progressive Web Applications.
* Cài đặt và áp dụng một số tính năng của Progressive Web Applications.
* Tìm hiểu về framework hỗ trợ Progressive Web Applications: ReactJS.
* Ứng dụng phù hợp với bất kỳ: máy tính để bàn, điện thoại di động, máy tính bảng hay bất cứ thiết bị thông minh.
* Xây dựng một ứng dụng web áp dụng các tính năng của Progressive Web Applications.

# NỘI DUNG

## Chương 1: CÁC CÔNG NGHỆ TÌM HIỂU

### 1.1. Progressive Web Applications

#### 1.1.1. Khái niêm

PWA là việc sử dụng các công nghệ web hiện đại để biến web apps trong giống native app hơn. Vì với native apps nó sẽ tích hợp tốt hơn với hệ điều hành do đó mang nhiều trải nhiệm tốt hơn cho người dùng. Chúng ta có thể cài đặt native apps do đó chúng ta có thể sử offline, hơn nữa người dùng yêu thích việc mở các ứng dụng tại homescreen để kết nối nhanh chóng tới các ứng dụng yêu thích hơn việc phải điều hướng bằng cách sử dụng trình duyệt web.

Với những ưu điểm của native app, do đó việc sử dụng các công nghệ để biến web apps thành PWA sẽ mang đến người dùng nhiều trải nghiệm tốt hơn

#### 1.1.2. Một số yếu tố quan trọng

- Discoverable: Có thể tìm kiếm bằng máy tìm kiếm như Google, Bing, ….

- Installable: Có thể cài đăt trên home screen của thiết bị giống như việc cài đặt một native apps có icon và có thể đưa ra homescreen.

- Linkable: Có thể chia sẽ bằng cách gửi URL và từ URL đó có thể kết nối tới web apps.

- Network indepentdent: Có thể hoạt động ofline hoặc đường truyền kết nối internet yếu.

- Progressive: Có thể hoạt động tốt trên các browers cũ.

- Re-engageable: Có thể thông báo cho người dung khi có nội dung mới.

- Responsive: Có thể hiển thị tốt trên nhiều thiết bị có kích thước khác nhau như hiển thị trên desktop hay trên mobile phone.

- Safe: Việc kết nối tới web app phải được bảo mật.

#### 1.1.3. Lợi ích

- Giúp làm giảm thời gian load sau khi app web được cài đặt nhờ vào công nghệ Caching với Service Workers.

- Có khả năng update chỉ những dữ liệu thay đổi khi app update sẵn sàng. Bằng việc này giúp cho người dùng nhận thấy và quay lại web app

- Có giao diện và cảm giác rằng tích hợp nhiều tính năng của native platform

- Lôi kéo người dùng quay lại app bằng việc sử dụng system notifications and push messages.

#### 1.1.4. Các công nghệ nổi bật

* Client-side storage
* Service workers
* Noitification API và Push API
* Responsive web apps
* Web app manifest

### 1.2. Client-Side Storage

Định Nghĩa: Là một API của WebBrowers cho phép website có thể lưu trữ dữ liệu tại phía client.

#### 1.2.1. Web storage

* Dùng để lưu trữ những thông tin cơ bản, được thiết đặt bằng một cặp gồm key và value.
* Set dữ liệu: localStorage.setItem(‘key’, ‘value’).
* Get dữ liệu:localStorage.getItem(‘key’).
* Ví dụ:



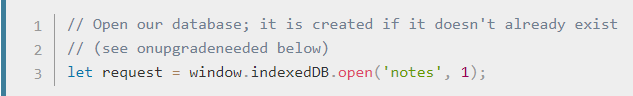
Hình 1 Web Storage

#### 1.2.2. IndexedDB

* Dùng để lưu trữ những dữ liệu phức tạp không giớí hạn dung lượng cũng như chiều dài của dữ liệu.

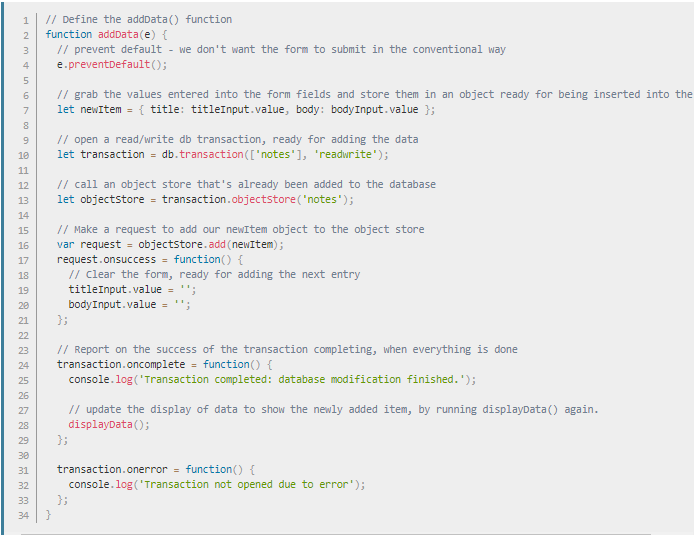
- Một số phương thức:

+ Khởi tạo:



Hình 2 Khởi tạo IndexedDB

+ Thêm data:



Hình 3 Thêm data cho IndexedDB

+ Xoá



Hình 4 Xóa IndexDB

### 1.3. Cache API

- Là một cách lưu trữ phía client bằng cách lưu 1 cặp request và response.

- Các phương thức:

1. Cache.match(request, options).
2. Cache.matchAll(request, options).
3. Cache.add(request).
4. Cache.addAll(requests).
5. Cache.put(request, response).
6. Cache.delete(request, options).
7. Cache.keys(request, options).

**Ví dụ:**



Hình 5 Khai báo một Cache API



Hình 6 Sử dụng Cache API để lưu trữ

### Service Worker

Là một file javascripts được chạy ở một thread riêng biệt tách biệt với HTML DOM. Nó được sử dụng trong việc tiếp nhận xử lí, giúp việc quản lí cache của web app.

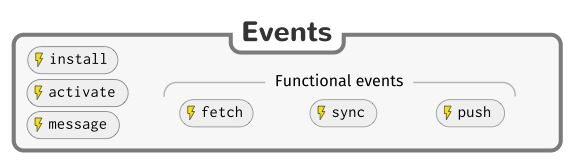
Vòng đời cơ bản của service workers:



Hình 7 Service Worker Lifecycle

* Service worker URL được tải về và được đăng kí trong trình duyệt.
* Nếu thành công service worker được thực thi trong ServiceWorkerGlobalScope , được chạy tách biệt với main script thread không có sử dung DOM access.
* Lúc này service worker bây giờ đã sẵn sàng cho việc tiếp nhận xử lí các event.
* Cài đặt service worker được thực thi và kiểm soát các file nằm trong chung 1 scope của file SW.
* Khi oninstall handler hoàn thành, SW được cho rằng đã được cài đặt.
* Tiếp theo là active SW.

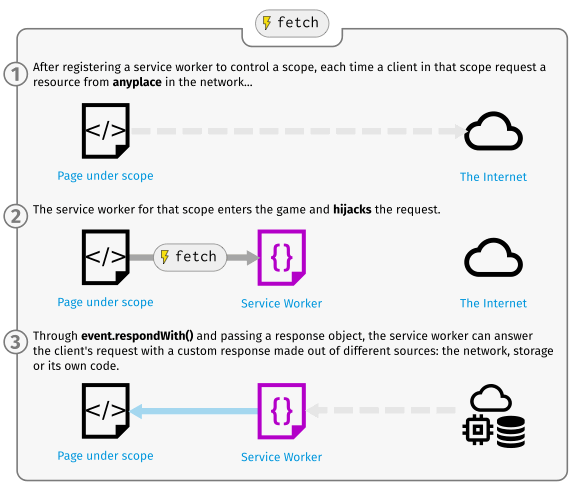
Các sự kiện của service worker:



Hình 8 Servive Worker Events

* + - Install: Thông báo service workers đã được cài dặt.
    - Activate: Thông báo sự kiện service workers đã được kích hoạt.
    - Fecth: Sự kiện xảy ra khi có một http request sinh ra
    - Sync: Khi có một sự kiên Sync phát sinh
    - Push: Sự kiên xảy ra khi có Push notification từ server.

Cách thức hoạt động của service worker



Hình 9 Service Worker Fetch

Bước 1: Sau khi một service workers đã được cài đặt và kích hoạt thì nó sẽ kiểm soát toàn bộ http resquest phát sinh tại những file mà nó quản lí.

Bước 2: Khi có một sự kiện fetch phát sinh service workers nó sẽ tiếp nhận.

Bước 3: Sau khi tiếp nhận nó sẽ bắt đầu xử lí và trả về response thích hợp mà ta đã cấu hình ở service workers.

### Notification API & Push Notification

Notification là một thông báo bật lên trên thiết bị người dùng. Notification có thể được kích hoạt cục bộ bởi một ứng dụng mở hoặc chúng có thể được “pushed” từ máy chủ đến người dùng ngày cả khi ứng dụng không chạy. Chúng giúp người dùng cập nhật notification kịp thời và ứng dụng thêm thu hút người sử dụng.

Push notification có thể dưới 2 dạng: Notification API và Push API. Notification API cho phép ứng dụng hiển thị thông báo hệ thống cho người dùng. Push API cho phép service worker xử lý “Push Messages” từ máy chủ, ngay cả khi ứng dụng không hoạt động.

### Responsive Web

Responsive Web Design (RWD) là xu hướng mới theo đó quy trình thiết kế và phát triển web sẽ đáp ứng mọi thiết bị và môi trường của người dùng theo các tiêu chí kích thước và chiều của màn hình thiết bị. Để làm được điều đó chúng ta sẽ sử dụng linh hoạt kết hợp các kỹ thuật bao gồm flexible grid, responsive image và CSS media query. Khi người dùng chuyển từ máy tính xách tay của họ sang iPad hay iPhone, trang web sẽ tự động chuyển đổi để phù hợp với kích thước màn hình và kịch bản xử lý. Nói cách khác, các trang web cần phải có công nghệ tự động đáp ứng theo thiết bị của người dùng. Điều này sẽ loại bỏ sự cần thiết cho nhiều thiết kế web khác nhau và giảm thiểu thời gian cũng như chi phí thiết kế web.

Responsive Web Design là một điều tất yếu phải có trong thời đại bây giờ. Tuy là nói chạy trên nhiều chế độ phân giải màn hình tuy nhiên bạn chỉ cần một CSDL, một layout website tất cả chỉ là CSS làm việc.

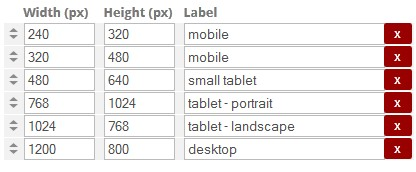
Có thể nói Responsive Web Design sẽ làm cho website chạy tốt trên mọi thiết bị di động, tăng tính tương thích cho website của bạn, tạo độ tin cậy và sự chuyên nghiệp với khách hàng.

Bạn có thể sử dụng Responsive Web Design ở bất kỳ dự án website nào, bằng bất kỳ ngôn ngữ thiết kế nào, hay bất kỳ một mã nguồn mở nào cũng được vì cốt lõi của nó chỉ là HTML và CSS.

Cách thức hoạt động:

Để thực hiện được những gì mà ta mong muốn bên trên thì chúng ta cần phân ra các cỡ màn hình chuẩn cơ bản và set theo các kích thước đó.

Độ phân giải màn hình thường gặp: 320px, 480px, 600px, 768px, 900px, 1200px, 1680px, 1050px …. có thể xep thành các cặp như: 320 x 480, 1024 x 768, 1680 1050..



Hình 10 Kích thước màn hình

### Web App Manifest

Định nghĩa:là những đặc tả web app được google sử dụng cho trình duyệt chrome.

Cách khai báo:

* Tạo một tập tin có tên là manifest.json.
* Thêm các thuộc tính vào tập tin.
* Liên kết tập tin vào trang html.

Một số thuộc tính của manifest:

* Name: Tên của web app.
* Short\_name: Tên sẽ hiển thị ngoài màn hình home.
* Icons: thiết đặt các thuộc tính icon cho web app.
* Start\_url: Chỉ định trang nào sẽ khởi chạy khi khởi động web app.
* Scope: Phân vùng.
* Display: Thiết đặt chế độ hiển thị.
* Orientation: Thiết đặt chế độ xem khung hình.
* Background\_color: Thiết lập màu background.
* Theme\_color: Thiết lập màu cho theme.
* Description: Miêu tả app web.
* Dir: Thiết lập chế đọ xem nội dung.
* Lang: Thiết đặt ngôn ngữ mặc định.

### 1.8. ReactJS

#### 1.8.1. Giới thiệu

- ReactJS là một thư viện JavaScript để xây dựng giao diện người dùng.

- Nó được duy trì bởi Facebook và một cộng đồng các nhà phát triển và công ty riêng lẻ.

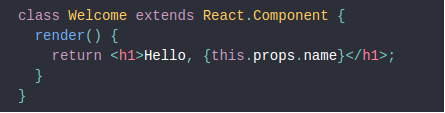
- Reactjs dùng virtual DOM.

#### 1.8.2. Các khái niệm quan trọng



##### 1.8.2.1. Component

- React được xây dựng quanh các component.



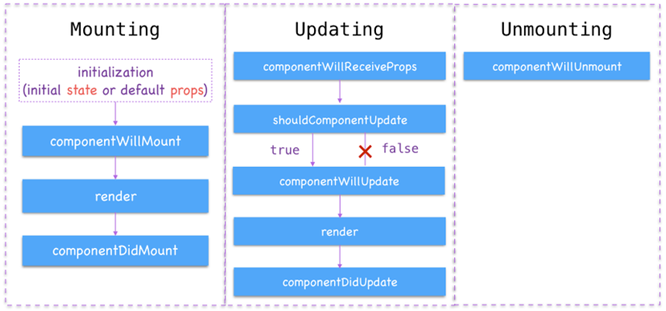
Hình 11 React Component

- Phương thức createClass trong React.Component nhận vào tham số, là đối tượng mô tả đặc tính của component.

- Đối tượng này bao gồm tất cả các phương thức để hình thành nên component.

- Phương thức quan trọng nhất là render, phương thức này được trigger khi component đã sẵn sàng để được render lên trên page.

- **Lifecycle Methods:**



Hình 12 React Component Lifecycle

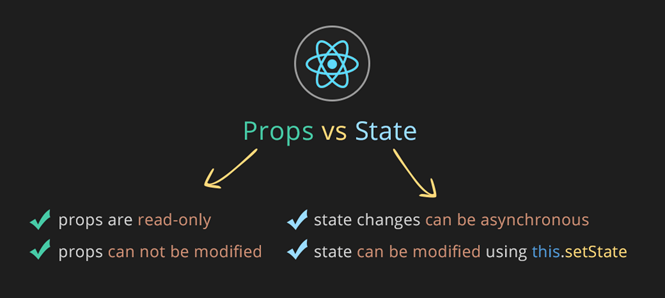
* ComponentWillMountđây là method sẽ được thực thi trước khi một component được render trên cả server side và client side.
* **ComponentDidMount** method này được thực thi khi 1 component được render trên client side.
* **ComponentWillReceiveProps** sẽ được thực thi ngay khi thuộc tính props được update và trước khi component được render lại.
* **ShouldComponentUpdate** sẽ trả về kết quả true or false. Phương thức này sẽ xác định 1 component có được update hay không. Mặc định giá trị này là true.
* **ComponentWillUpdate** được gọi khi chúng ta update state của component trước khi nó render lại.
* **ComponentDidUpdate** sau khi componentWillUpdate ở trên được gọi xong thì đến lượt thằng này được goi.
* **ComponentWillUnmount** được gọi khi chúng ta unmout 1 component kiểu như xóa nó khỏi react.

##### 1.8.2.2. JSX — Javascript Syntax Extension

- Đây đơn giản là một syntax extension của Javascript.

- Về bản chất, các tag thực sự là những lời gọi hàm, sẽ được chuyển đổi trong React code và end up dưới dạng HTML và Javascript trong cây DOM.

##### 1.8.2.3. Props & State



Hình 13 React Prop & State

- Props & State là 2 kiểu data của React.- State: private và chỉ có thể được thay đổi bên trong bản thân component.

- Props: mang tính external, và không bị kiểm soát bởi bản thân component. Nó được truyền từ component cao hơn theo phân cấp, hay có thể hiểu đơn giản là truyền từ component cha xuống component con, cái mà điều khiển dữ liệu trước khi truyền xuống.

##### Thuộc tính key

- Khi tạo các component một cách dynamically, mỗi thành phần đều cần thuộc tính key, và thuộc tính này là duy nhất (unique).

- Trong suốt quá trình rendering, các component sẽ bị xáo trộn, chúng cũng có thể bị destroy hay recreate tùy vào sự khác nhau của mỗi giải thuật, việc gán cho nó một key để định danh và đảm bảo rằng các component đều ở đúng vị trí của nó, tối ưu hóa việc rendering.

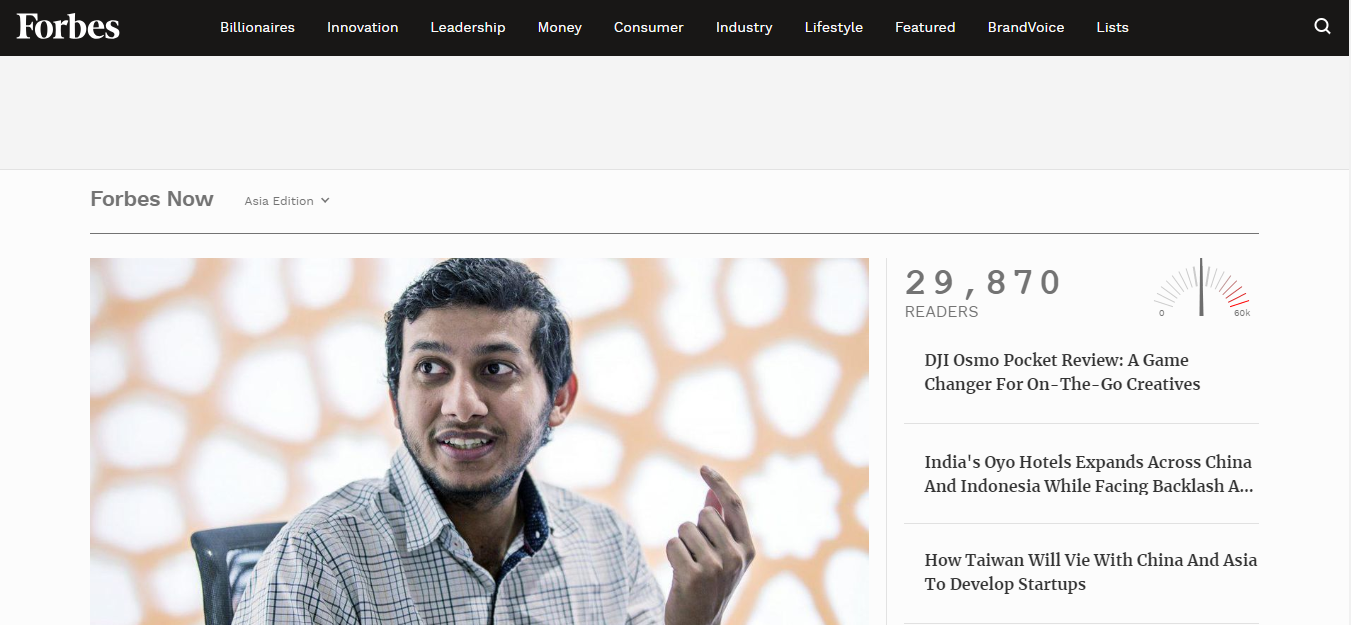


Hình 14 React thuộc tính key

## Chương 2: KHẢO SÁT HIỆN TRẠNG VÀ PHÂN TÍCH YÊU CẦU

### 2.1. Khảo sát hiện trạng

#### 2.1.1. Forbes (<https://www.forbes.com/>)



Hình 15 Forbes (<https://www.forbes.com/>)

Forbes là một tạp chí kinh doanh hàng đầu của Mỹ chuyên sản xuất các bài viết về các chủ đề tài chính, tiếp thị, đầu tư và công nghệ.

Ứng dụng web của Forbes khi áp dụng PWA đã làm tăng thời gian truy cần và tăng gấp đôi tỉ lệ tương tác thông qua việc ‘push notification’ và tùy chọn thêm cài đặt vào màn hình chính. Nhờ đó kết quả đạt được:

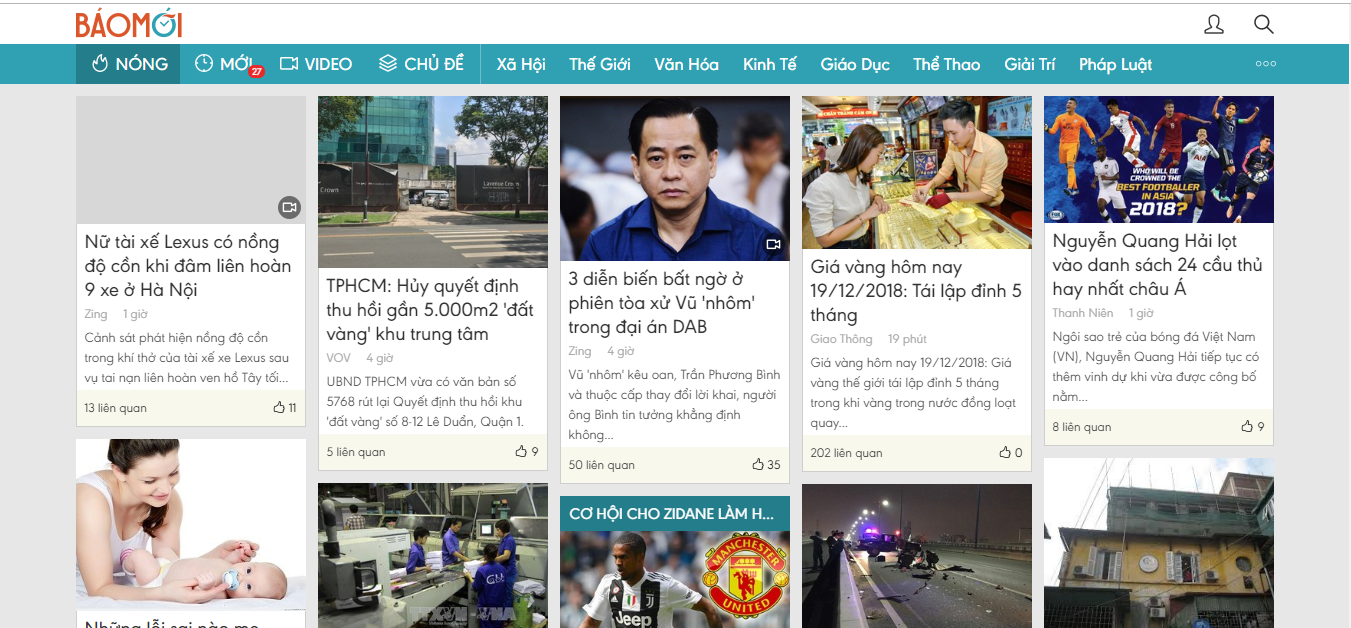
- Tăng 43% số thời gian truy cập trên một user.

- Tăng 20% hiệu suất trên mỗi một trang.

- Tăng 100% tỉ lệ người tham gia.

- Tăng đến 6 lần số độc giả đọc hết một bài viết.

#### 2.1.2. Báo mới (<https://baomoi.com/>)

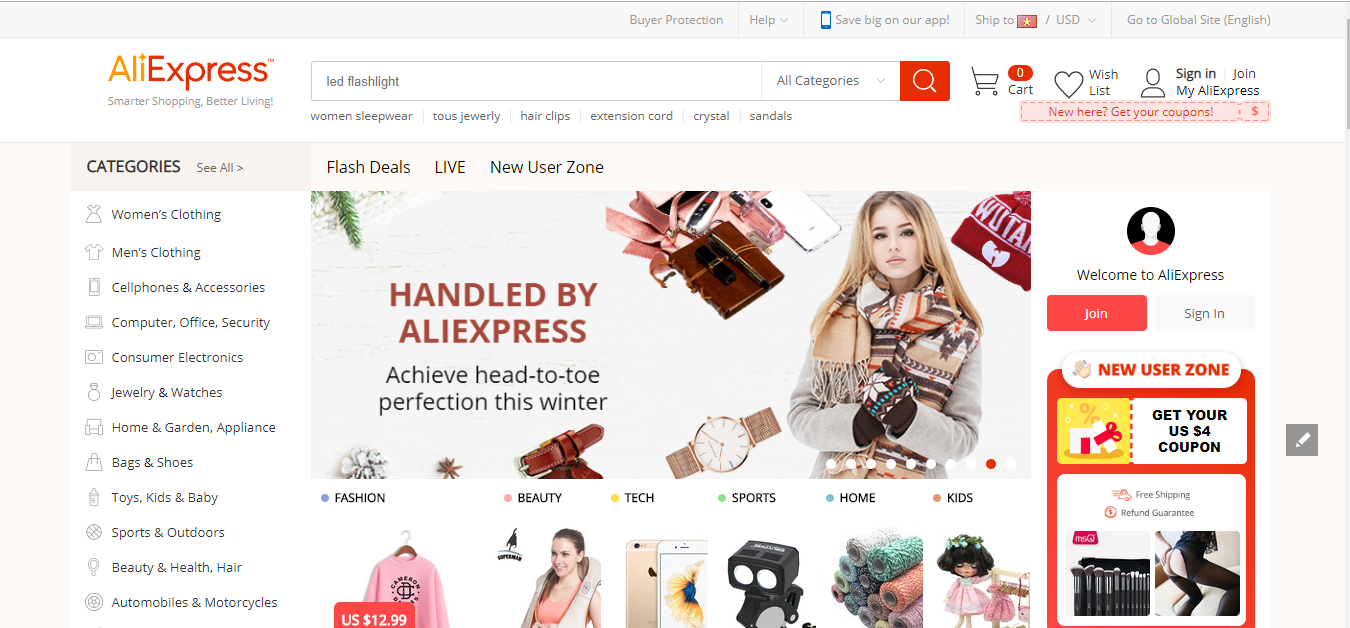


Hình 16 Báo mới (<https://baomoi.com/>)

Báo mới có áp dụng PWA cho ứng dụng web của mình làm tốc độ xử lí trang tốt và tăng trải nghiệm người dùng qua việc responsive và service worker.

Tuy nhiên, báo mới chưa hoàn toàn phát triển hết PWA và đồng thời còn một số hạn chế như manifest icon bị lỗi, chưa tốt trên mobile (có app mobile),…

#### 2.1.3. AliExpress (<https://www.aliexpress.com/>)



Hình 17 AliExpress (<https://www.aliexpress.com/>)

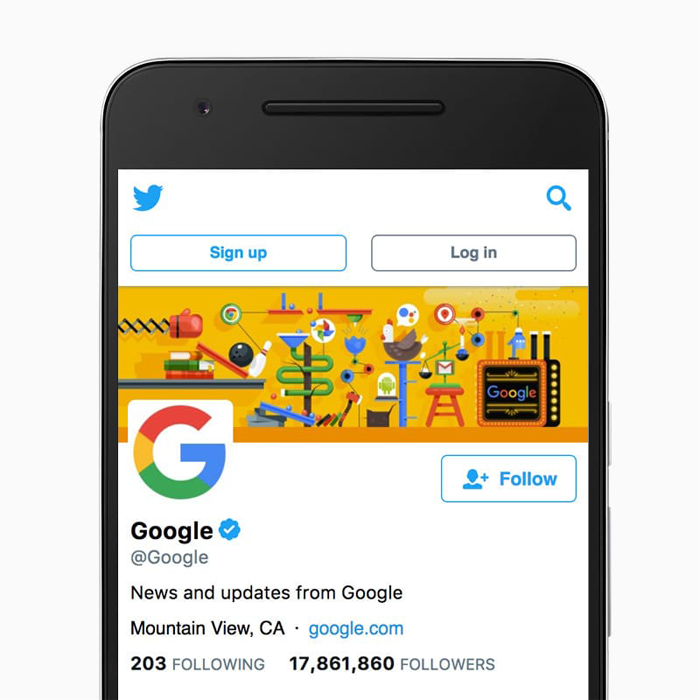
AliExpress là một trang thương mại điện tử phổ biến của tập đoàn Alibaba. Vài năm trước, họ đã chuyển đổi trang web di động của mình thành PWA.

Với công nghệ này, trang web AliExpress đã nâng cao đáng kể trải nghiệm người dùng và hiệu suất của mình. Và kết quả đạt được:

- Tăng 2 lần số lượng người truy cập trên một trang.

- Tăng 74% số thời gian truy cập trong mỗi phiên truy cập.

#### 2.1.4. Twitter Lite



Hình 18 Twitter Lite

Năm 2017, Twitter tung ra Progressive Web Applications ‘Twitter Lite’ là trải nghiệm web mặc định cho tất cả người dùng trên thế giới.

Với hơn 80% người dùng trên thiết bị di động, Twitter muốn cải thiện trải nghiệm web trên thiết bị di động của mình bằng cách làm cho nó nhanh hơn và hấp dẫn hơn, đặc biệt là vì nhiều người dùng của họ nằm ở khu vực có điều kiện mạng chậm, không ổn định.

Twitter Lite  tải các trang ngay lập tức và giảm sử dụng dữ liệu bằng cách tối ưu hóa hình ảnh và dựa vào dữ liệu được lưu trong bộ nhớ cache.

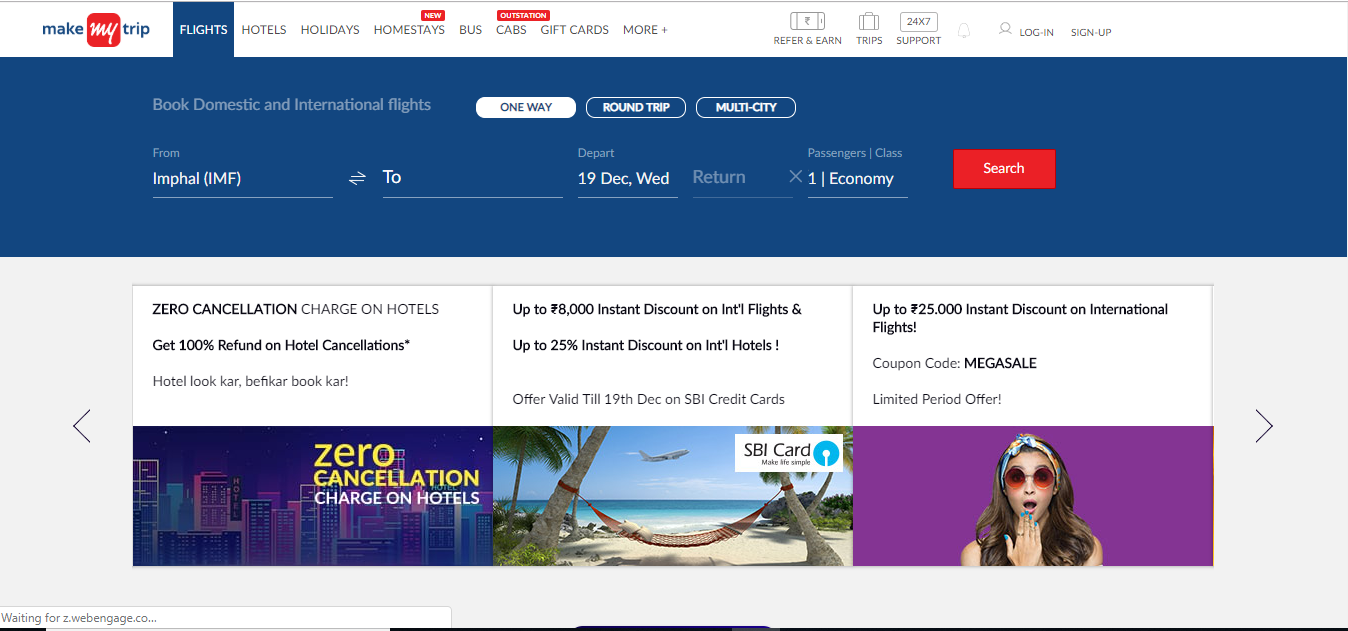
Nó cũng thu hút người dùng bằng ‘push notification’ và cho phép người dùng thêm ứng dụng vào màn hình chính của họ. Qua đó, kết quả đạt được:

- Tăng 75% Tweets gửi đi.

- Giảm 20% tỉ lệ thoát ứng dụng.

- Tăng 65% số trang được xem trong mỗi lần truy cập.

#### 2.1.5 MakeMyTrip



Hình 19 MakeMyTrip

MakeMyTrip là công ty du lịch trực tuyến hàng đầu Ấn Độ.

Ứng dụng web của MakeMyTrip khi áp dụng PWA mang đến sự dễ dàng và trải nghiệm ứng dụng gốc cho người dùng trên các trình duyệt di động, tăng tỷ lệ chuyển đổi và thời gian tải trang của công ty và cải thiện trải nghiệm người dùng nói chung.

Dưới đây là một số thành công khi áp dụng PWA:

- Tăng 3 lần tỉ lệ chuyển đổi.

- Tăng 35% tốc độ tải trang.

- Tăng 150% số người truy cập.

### 2.2. Phân tích yêu cầu

#### 2.2.1. Mô tả yêu cầu nghiệp vụ

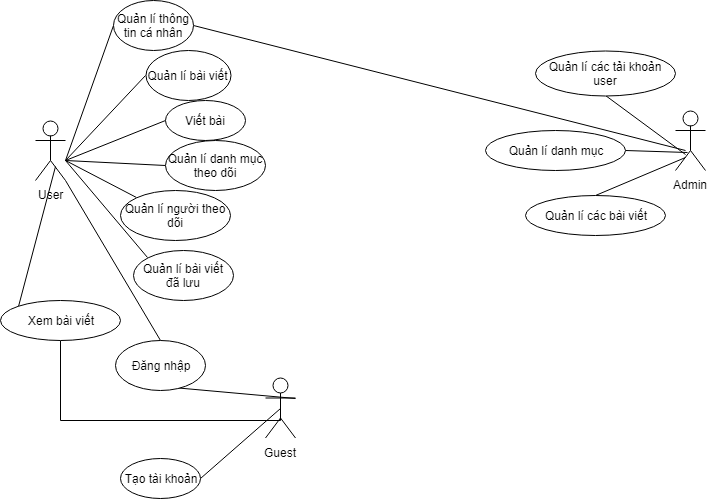
- Ứng dụng web cung cấp cho người dùng có thể xem các bài viết.

- Thành viên có thể soạn thảo và đăng các bài viết của bản thân. Không những thế, họ còn có thể lưu lại các bài viết mình yêu thích vào bookmark và nhân được thông báo khi có bài viết mới.

- Thành viên được cung cấp một trang user page quản lí các thông tin cá nhân, bài viết của mình và tìm kiếm lại các bài viết đã lưu.

- Về mặt công nghệ, ứng dụng áp dụng đầy đủ các tính năng của PWA vào hệ thống. Nó được xây dựng trên framework ReactJS kết hợp NodeJS + ExpressJS và hệ quản trị cơ sở dữ liệu MySQL.

#### 2.2.2. Usecase Diagram



Hình 20 Usecase Diagram

#### 2.2.3. Bảng mô tả các use case

|  |  |  |
| --- | --- | --- |
| TT | Tên use case | Ý nghĩa |
| 1 | Quản lí thông tin cá nhân | Dùng để chỉnh sửa thông tin cá nhân, chẳng hạn như cập nhật tên, ngày sinh… |
| 2 | Quản lí bài viết | Dùng để thêm , xóa , sửa bài viết mà cá nhân đó đã viết. |
| 3 | Viết bài | Dùng để viết bài để đăng lên web |
| 4 | Quản lí danh mục theo dõi | Dùng để thêm hoặc sửa xóa các danh mục đang theo dõi |
| 5 | Quản lí người theo dõi | Dùng để kiểm tra và tùy chình những cá nhân mà đang theo dõi |
| 6 | Quản lí bài viết đã lưu | Dùng để thêm, xóa các bài viết mà cá nhân đã lưu |
| 7 | Xem bài viết | Xem các nôi dung liên quan của một bài viết |
| 8 | Đăng nhập | Dùng tài khoản để đăng nhập vào hệ thống |
| 9 | Tạo tài khoản | Tạo một tài khoản để đăng nhập vào hệ thống |
| 10 | Quản lí các tài khoản | Thêm, sửa, xóa các tài khoản đang có trong hệ thống |
| 11 | Quản lí danh mục | Dùng để thêm, sửa, xóa các danh mục của các bài viết |
| 12 | Quản lí bài viết | Dùng để thêm, sửa, xóa các bài viết đã được lên hệ thống |

Bảng 2.2.3 Mô tả các use case

#### 2.2.4. Mô tả các actor

- Guest:

+ Đăng nhập.

+ Tạo tài khoản

+ Xem các bài viết

- User:

+ Quản lí thông tin cá nhân

+ Quản lí bài viết

+ Viết bài

+ Quản lí danh mục theo dõi

+ Quản lí người theo dõi

+ Quản lí bài viết đã lưu

+ Xem bài viết

- Admin:

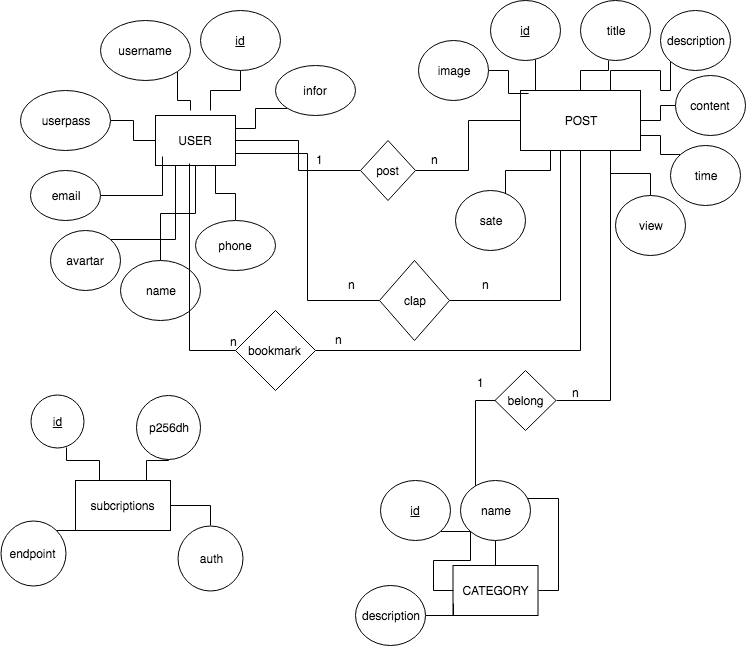
+ Quản lí các tài khoản

+ Quản lí danh mục

+ Quản lí bài viết

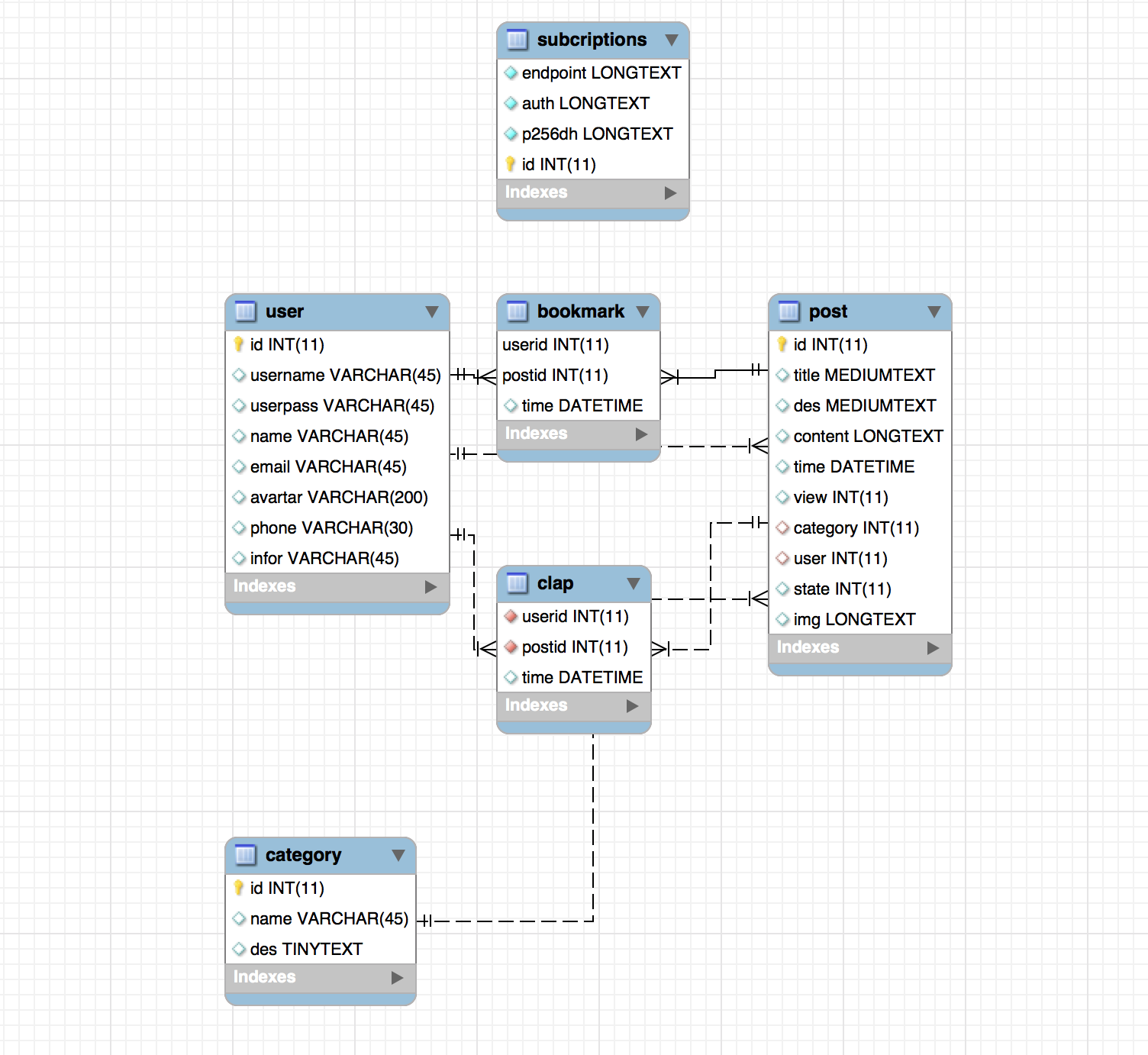
## Chương 3: THIẾT KẾ CƠ SỞ DỮ LIỆU

### 3.1. Mô hình ERD



Hình 21 Mô hình ERD

### 3.2. Class Diagram



Hình 22 Class Diagram

### Mô tả dữ liệu chi tiết

|  |  |  |
| --- | --- | --- |
| TT | Tên bảng | Mục đích của bảng trong chương trình |
| 1 | User | Chứa nội dung các thông tin đăng nhập và phân quyền của user |
| 2 | Post | Chứa nội dung các thông tin cá nhân của user |
| 3 | Category | Chứa nội dung các bình luận trong diễn đàn |
| 4 | Clap | Chứa nội dung các thông báo |
| 5 | Bookmark | Chứa nội dung thông tin các khóa học |
| 6 | Subcriptions | Chứa các thông tin về những clien đăng ký subcribe |

Bảng 1 Mô tả các bảng trong cơ sở dữ liệu

#### 3.3.1. Bảng mô tả các trường trong User

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | id | ID của user(duy nhất,khóa chính) |
| 2 | Username | Tên đăng nhập |
| 3 | Userpass | Mật khảo đăng nhập |
| 4 | Name | Tên thật của user |
| 5 | Email | Dịa chỉ email của user |
| 6 | Avartar | Ảnh đại diện của user |
| 7 | Phone | Số điện thoại của user |
| 8 | Infor | Thông tin thêm hay dùng để miêu tả thêm user |

Bảng 2 Mô tả các trường trong User

#### 3.3.2. Bảng mô tả các trường trong Post

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | Id | Mã xác định cho bài post(duy nhất, khóa chính) |
| 2 | Title | Tiêu đề cho bài viết |
| 3 | Des | Miêu tả, tóm lược cho bài viết |
| 4 | Content | Nội dung của bài viết |
| 5 | Time | Thời gian mà bài đó được lư vào cơ sở dữ liệu |
| 6 | View | Số lượt xem của bài viết |
| 7 | Category | Id của category mà bài viết đó thuộc vào |
| 8 | User | Id người upload bài viết |
| 9 | State | Trạng thái của bài viết |
| 10 | Img | Ảnh bìa của bài viết |

Bảng 3 Mô tả các trường trong Post

#### 3.3.3. Bảng mô tả các trường trong Category

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | Id | Mã định danh cho Category(duy nhất, khóa chính) |
| 2 | Name | Tên của category này |
| 3 | Des | Miêu tả chi tiết của category |

Bảng 4 Mô tả các trường trong Category

#### 3.3.4. Bảng mô tả các trường trong Clap

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | Userid | Đinh danh của người thích bài |
| 2 | Postid | Định danh cho bài post |
| 3 | Time | Thời gian lưu vào cơ sở dữ liệu |

Bảng 5 Mô tả các trường trong Clap

#### 3.3.5. Bảng mô tả các trường trong Bookmark

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | Userid | Đinh danh của người thích bài |
| 2 | Postid | Định danh cho bài post |
| 3 | Time | Thời gian lưu vào cơ sở dữ liệu |

Bảng 6 Mô tả các trường trong Bookmark

#### 3.3.6. Bảng mô tả các trường trong Subscriptions

|  |  |  |
| --- | --- | --- |
| TT | Tên trường | Mục đích của trường trong chương trình |
| 1 | id | Đinh danh của subscribe client |
| 2 | endpoint | Địa chỉ của một client subscribe |
| 3 | auth | Xác minh client |
| 4 | p256dh | Mã băm |

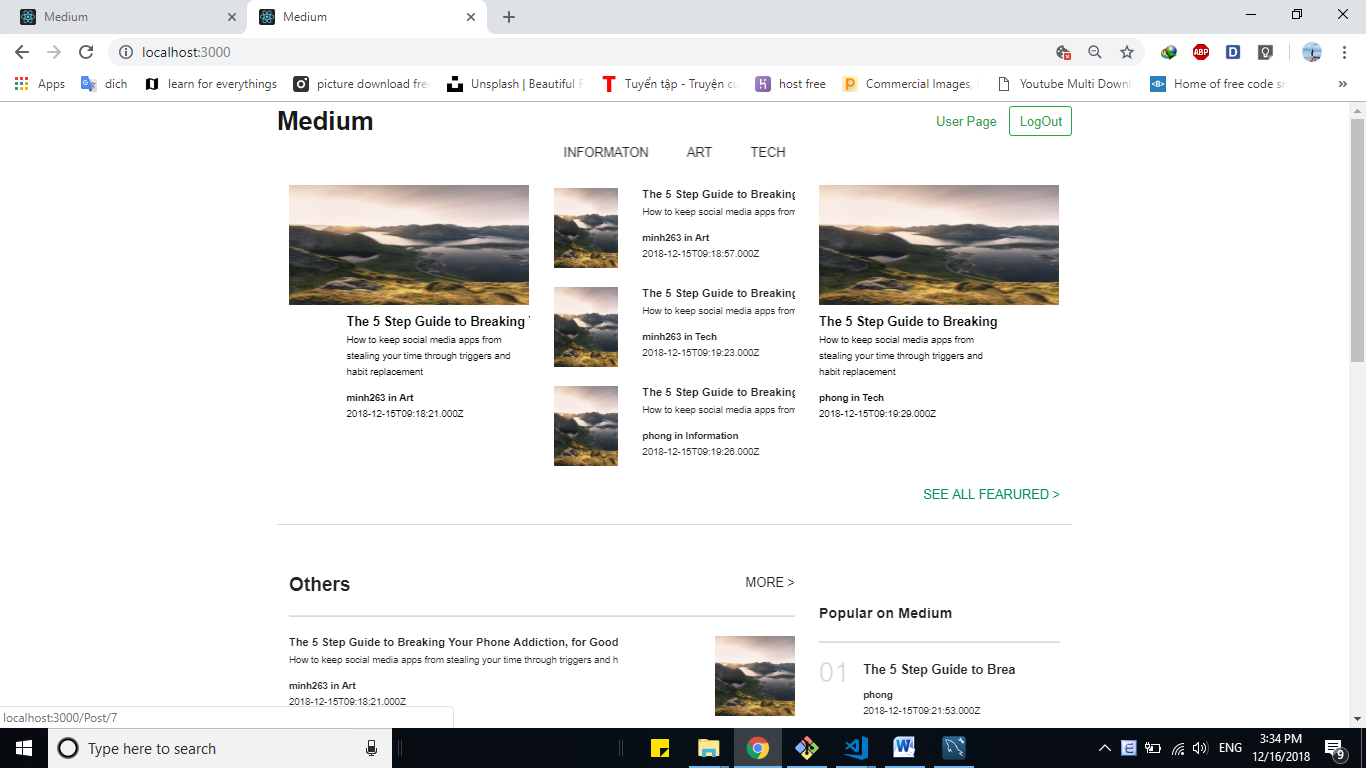
Bảng 7 Mô tả các trường trong Subscriptions

## Chương 4: THIẾT KẾ GIAO DIỆN

|  |  |  |
| --- | --- | --- |
| TT | Màn hình chính/Cửa sổ/Dialog | Mục đích |
| 1 | MH 1: HomePage | Toàn bộ thông tin cần thiết của webside khi chưa đăng nhập |
| 2 | MH 2: CategoryPage | Toàn bộ thông tin cần thiết của webside khi đăng nhập |
| 3 | MH 3: PostPage | Đăng nhập vào weside |
| 4 | MH 4: UserPage | Đăng ký tài khoản |
| 5 | MH 5: NewPostPage | Thông tin cá nhân |
| 6 | MH 6: EditPostPage | Tất cả khóa học đã và đang học |
| 7 | MH 7: EditProfilePage | Thay đổi mật khẩu cá nhân |

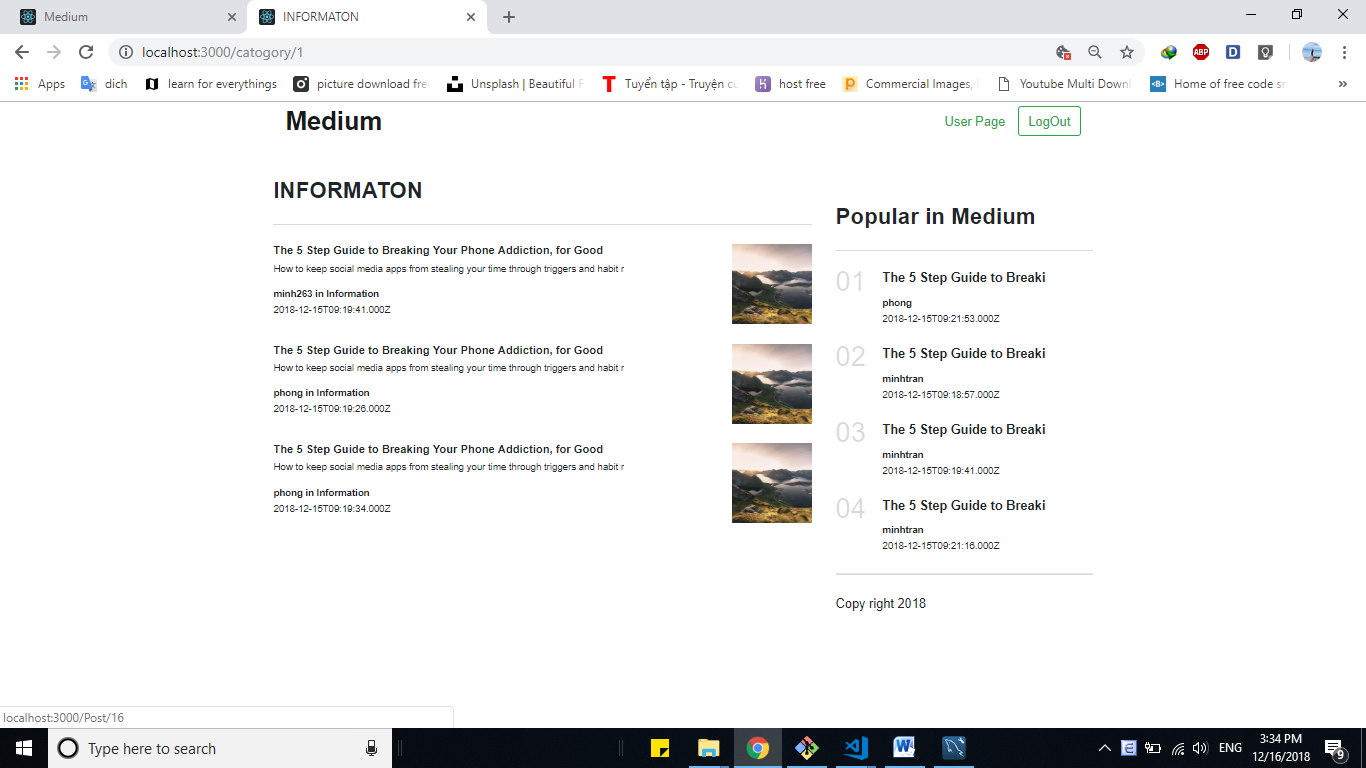
Bảng 8 Danh mục các màn hình trong ứng dụng

### 4.1. HomePage



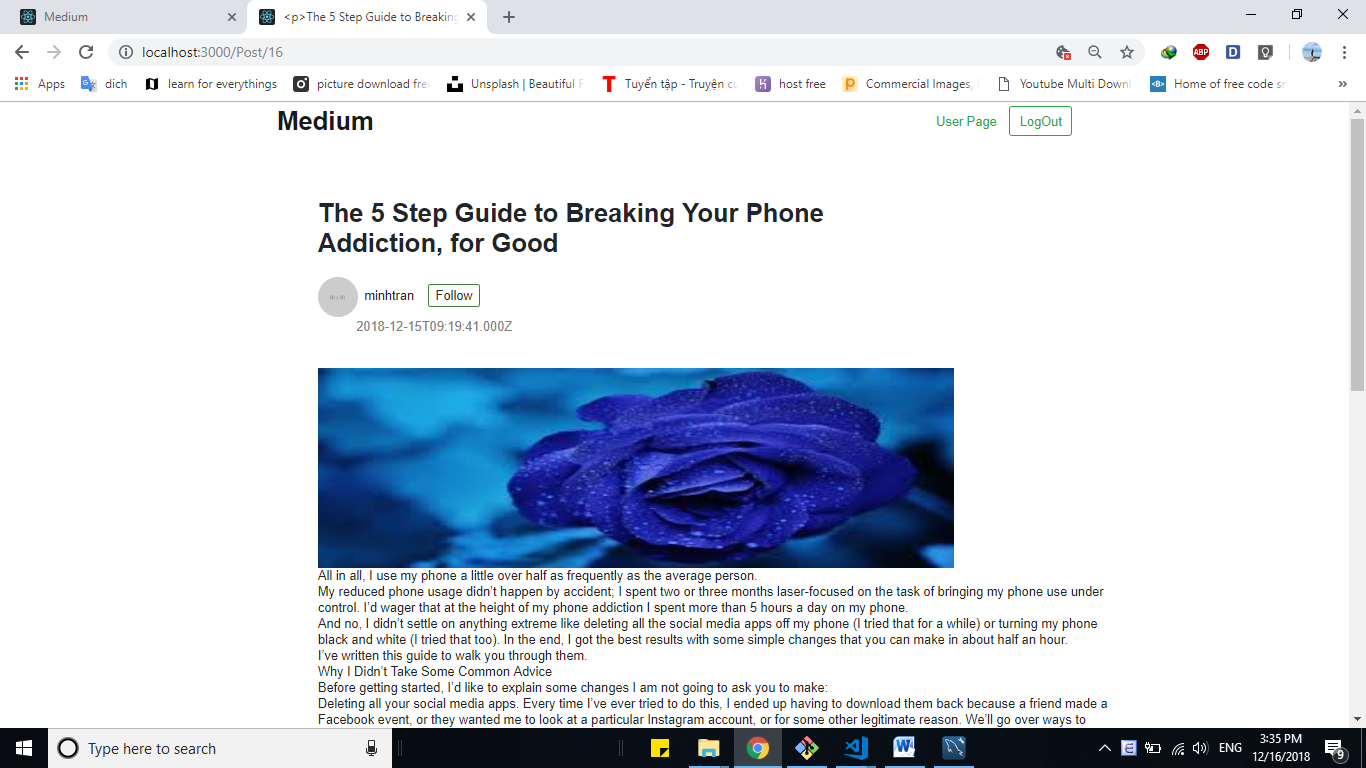
Hình 23 HomePage

### 4.2. CategoryPage



Hình 24 CategoryPage

### 4.3. PostPage



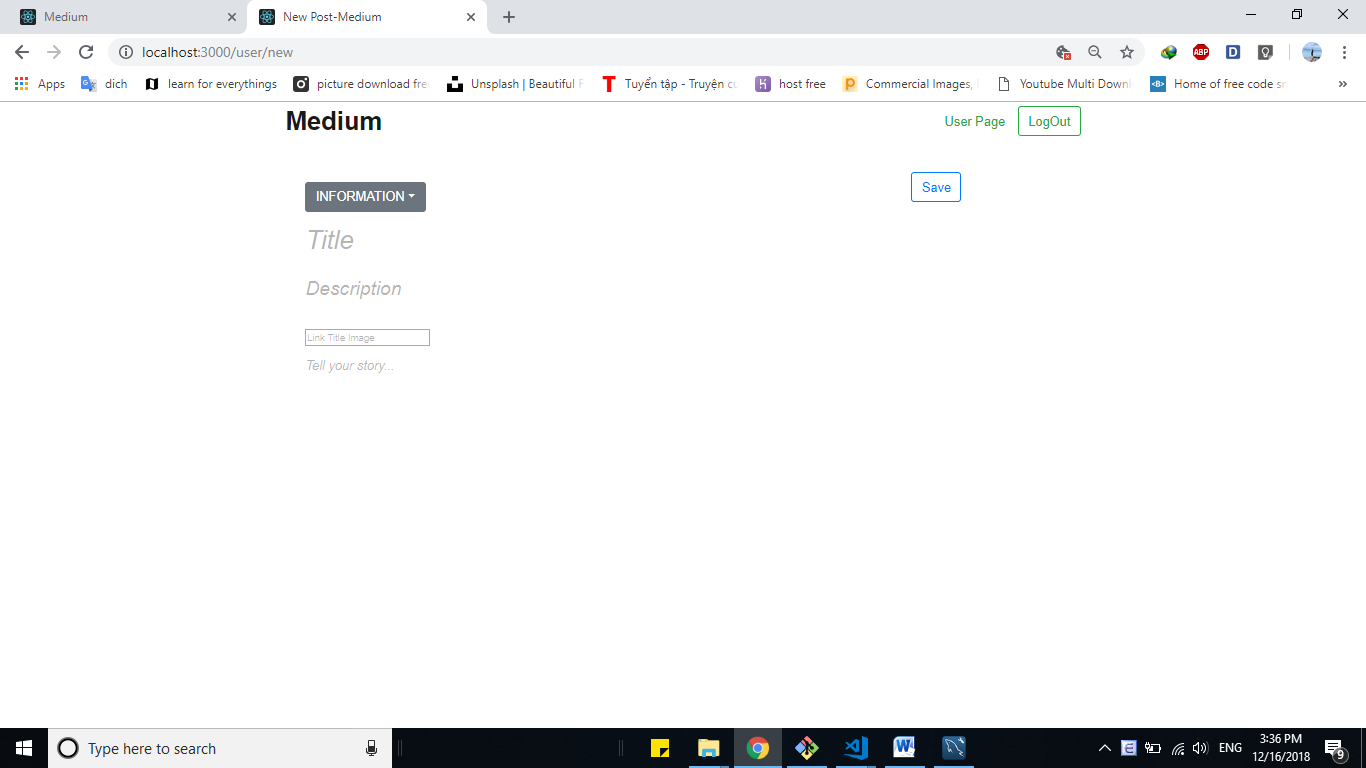
Hình 25 PostPage

### 4.4. UserPage



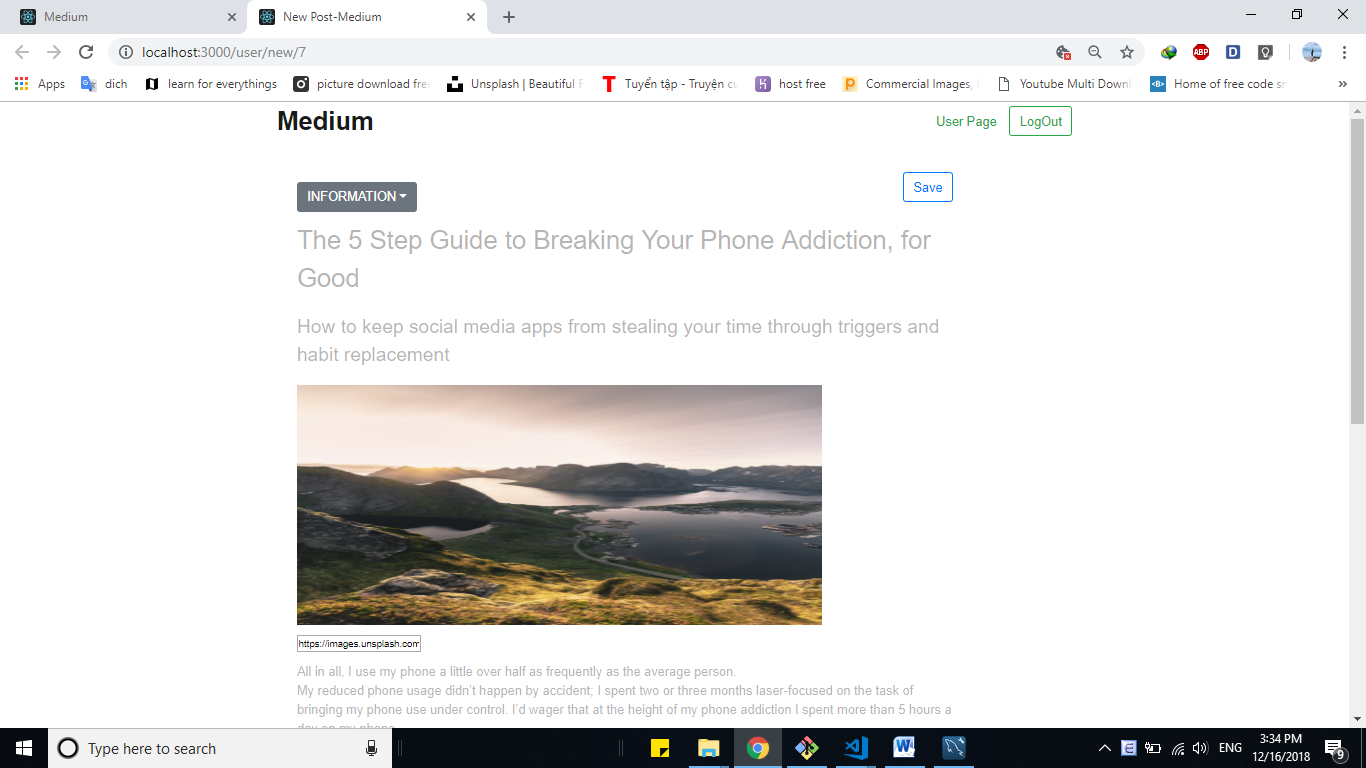
Hình 26 UserPage

### 4.5. NewPostPage



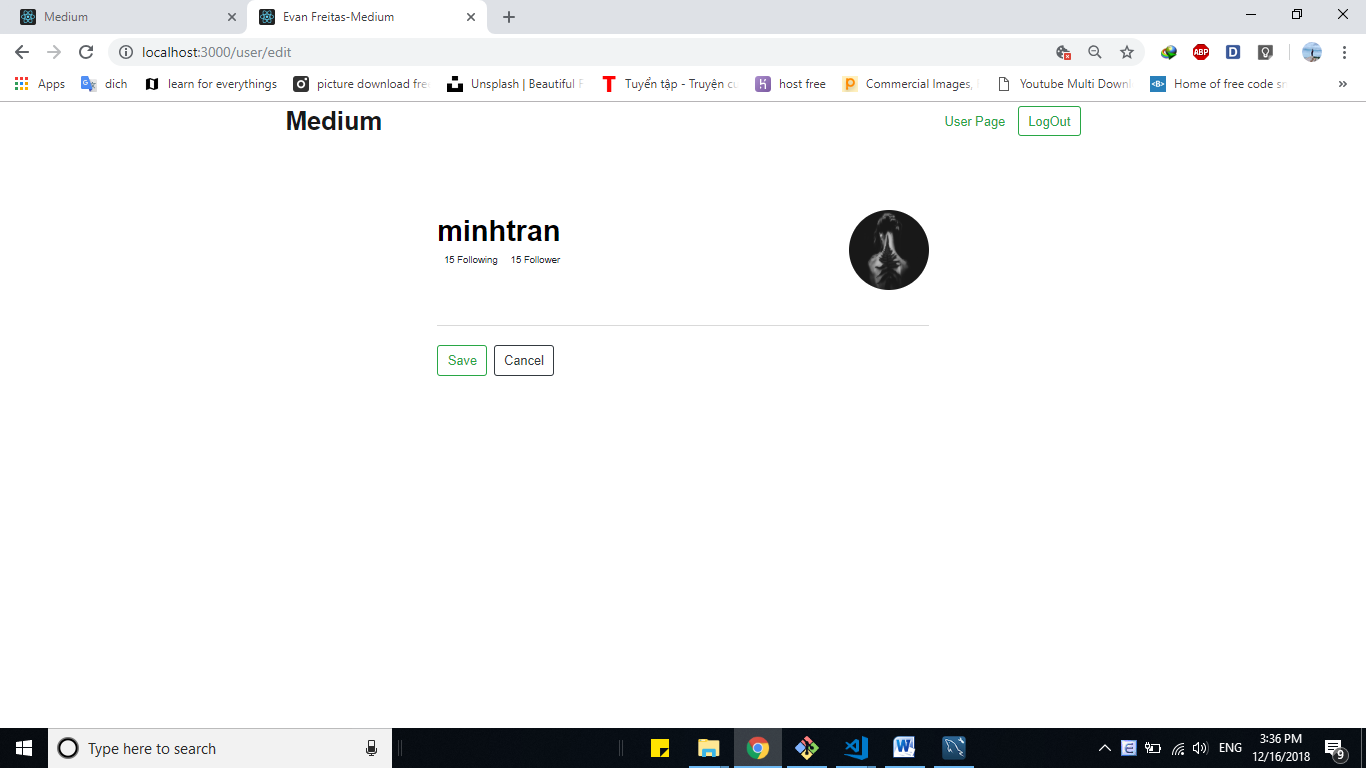
Hình 27 NewPostPage

### 4.6. EditPostPage



Hình 28 EditPostPage

### 4.7. EditProfilePage



Hình 29 EditProfilePage

## Chương 5: CÀI ĐẶT

### 5.1. NodeJS

* Node.js là một mã nguồn mở, một môi trường cho các máy chủ và ứng dụng mạng.
* Node.js sử dụng Google V8 JavaScript engine để thực thi mã, và một tỷ lệ lớn các mô-đun cơ bản được viết bằng JavaScript. Các ứng dụng node.js thì được viết bằn JavaScript.
* Node.js chứa một thư viện built-in cho phép các ứng dụng hoạt động như một Webserver mà không cần phần mềm như Nginx, Apache HTTP Server hoặc IIS.
* Node.js cung cấp kiến trúc hướng sự kiện (event-driven) và non-blocking I/O API, tối ưu hóa thông lượng của ứng dụng và có khả năng mở rộng cao.
* Mọi hàm trong Node.js là không đồng bộ (asynchronous). Do đó, các tác vụ đều được xử lý và thực thi ở chế độ nền (background processing).

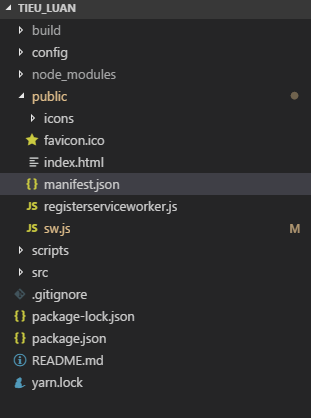
### 5.2. Hệ quản trị cơ sở dữ liệu MySQL

MySQL là [hệ quản trị cơ sở dữ liệu](https://vi.wikipedia.org/wiki/H%E1%BB%87_qu%E1%BA%A3n_tr%E1%BB%8B_c%C6%A1_s%E1%BB%9F_d%E1%BB%AF_li%E1%BB%87u) [tự do nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F) phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. Người dùng có thể tải về MySQL miễn phí từ trang chủ. MySQL có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux), [Mac OS X](https://vi.wikipedia.org/wiki/Mac_OS_X), [Unix](https://vi.wikipedia.org/wiki/Unix), [FreeBSD](https://vi.wikipedia.org/wiki/FreeBSD), [NetBSD](https://vi.wikipedia.org/w/index.php?title=NetBSD&action=edit&redlink=1), [Novell NetWare](https://vi.wikipedia.org/w/index.php?title=Novell_NetWare&action=edit&redlink=1), [SGI Irix](https://vi.wikipedia.org/w/index.php?title=SGI_Irix&action=edit&redlink=1), [Solaris](https://vi.wikipedia.org/wiki/Solaris), [SunOS](https://vi.wikipedia.org/w/index.php?title=SunOS&action=edit&redlink=1),..

MySQL là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ NodeJs, [PHP](https://vi.wikipedia.org/wiki/PHP), [Perl](https://vi.wikipedia.org/wiki/Perl), và nhiều ngôn ngữ khác, làm nơi lưu trữ những thông tin trên các trang web viết bằng NodeJs, PHP hay Perl,...

### 5.3 Cấu hình Progressive Web Applications



Hình 30 Cấu trúc thư mục ứng dụng

#### 5.3.1. Cấu hình Manifest:

- Lưu file manifest.json cùng thư mục với trang index.html.

- Nội dung manifest.json:

{

"short\_name": "Medium",

"name": "Demon Medium App",

"icons": [

{

"src": "/icons/icon-48x48.png",

"sizes": "48x48",

"type": "image/png"

},

{

"src": "/icons/icon-57x57.png",

"sizes": "57x57",

"type": "image/png"

},

{

"src": "/icons/icon-72x72.png",

"sizes": "72x72",

"type": "image/png"

},

{

"src": "/icons/icon-76x76.png",

"sizes": "76x76",

"type": "image/png"

},

{

"src": "/icons/icon-96x96.png",

"sizes": "96x96",

"type": "image/png"

},

{

"src": "/icons/icon-114x114.png",

"sizes": "114x114",

"type": "image/png"

},

{

"src": "/icons/icon-144x144.png",

"sizes": "144x144",

"type": "image/png"

},

{

"src": "/icons/icon-192x192.png",

"sizes": "192x192",

"type": "image/png"

},

{

"src": "/icons/icon-256x256.png",

"sizes": "256x256",

"type": "image/png"

},

{

"src": "/icons/icon-384x384.png",

"sizes": "384x384",

"type": "image/png"

},

{

"src": "/icons/icon-512x512.png",

"sizes": "512x512",

"type": "image/png"

}

],

"start\_url": "./index.html",

"display": "standalone",

"theme\_color": "#000000",

"background\_color": "#ffffff"

}

- Thư mục icon chứa tất các ảnh icon cho ứng dụng nằm cùng thư mục với file manifest.json.

#### 5.3.2. Cấu hình Service Worker

- Nội dung file registerserviceworker.js:

if('serviceWorker' in navigator){

navigator.serviceWorker.register('/sw.js',{scope:'/'})

.then(function(reg){

console.log('Registion succeed. Scope is '+reg.scope);

}).catch(function(error){

console.log('Registration faied with' + error)

});

//register sync

Notification.requestPermission(permission=>{

console.log('permission:',permission)

})

}

- Nội dung file sw.js:

const cacheName='v1';

var db;

const urlCache=[

'/'

]

self.addEventListener('install',function(e){

e.waitUntil(

caches.open(cacheName).then(function(cache){

return cache.addAll(urlCache);

})

)

})

// self.addEventListener(active)

self.addEventListener('fetch',function(e){

var type=['image','style','script'];

var request=e.request;

console.log(request.destination);

var findReponse=caches.open(cacheName)

.then(cache=>cache.match(request))

.then(response=>{

if(response){

return response;

}else{//cache neu tap tin la image hoac cac file tinh

if(type.includes(e.request.destination)){

return fetch(request)

.then(function(res){

return caches.open(cacheName)

.then (cache=>{

cache.put(e.request.url,res.clone());

return res;

})

})

}else{

return fetch(request);

}

}

})

e.respondWith(findReponse);

});

self.addEventListener('sync',function(e){

if(e.tag==='submit'){

console.log('sync!');

var request = indexedDB.open("PostDatabase",6);

request.onerror = function(event) {

window.alert('index DB is wrong');

};

// nếu thành công

request.onsuccess = function(event) {

console.log("running onsuccess");

db = event.target.result;

readpost();

};

}

if(e.tag=='bookmark'){

console.log('sync!');

var request = indexedDB.open("PostBookmark",1);

request.onerror = function(event) {

window.alert('index DB is wrong');

};

// nếu thành công

request.onsuccess = function(event) {

console.log("running onsuccess");

db = event.target.result;

readbookmark();

};

}

function readpost(){

db.transaction("posts").objectStore("posts").get("1").onsuccess = function(event) {

let result=event.target.result;

const data = {

"title":result.title,

"des": result.des,

"content": result.content,

"view" : 0,

"category":parseInt(result.category),

"user": parseInt(result.user),

"state": 1,

"img": result.img

};

let url='http://localhost:3001/post/upload';

fetch(url,{

method:'POST',

body:JSON.stringify(data),

headers:{

'Content-Type': 'application/json'

}

}).then(res => res)

.then(response => console.log('Success:', response))

.catch(error => console.error('Error:', error));

};

}

function readbookmark(){

var objectStore = db.transaction(["bookmark"],"readwrite").objectStore("bookmark");

var postbookmark = [];

objectStore.openCursor().onsuccess = function(event) {

var cursor = event.target.result;

if (cursor) {

postbookmark.push(cursor.value);

cursor.continue();

}

else {//sync bookmark and cache bookmark

postbookmark.forEach(data=>{

let urlpage='/Post/'+data.ppostid;

let url='http://localhost:3001/post/load\_post/'+data.ppostid+'/';

fetch(url).then(res =>{

caches.open(cacheName)

.then(cache=>{

cache.put(url,res.clone())

})

})

fetch(urlpage).then(res =>{

caches.open(cacheName)

.then(cache=>{

cache.put(urlpage,res.clone())

})

})

});

//Xoa indexdb

var requestdelete=objectStore.clear();

requestdelete.onsuccess=function(event){

console.log('Xoa thanh cong');

}

requestdelete.onerror=function(){

window.alert('xoa that bai');

}

}

}

}

})

self.addEventListener('activate', function(event) {

var cacheKeeplist = [];

event.waitUntil(

caches.keys().then(function(keyList) {

return Promise.all(keyList.map(function(key) {

if (cacheKeeplist.indexOf(key) === -1) {

return caches.delete(key);

}

}));

})

);

});

#### 5.3.3. Cấu hình Cache API



Hình 31 Sơ đồ Cache dữ liệu

Đầu tiên web apps sẽ phát sinh một http request, tiếp đến service worker sẽ bắt sự kiện này và xử lí xem nếu có dữ liệu cache tương ứng với http request trên thì service worker sẽ trả lại response tương ứng được chứa trong cache data. Nếu service worker không phát hiện thấy dữ liệu tương ứng không có trong cache thì nó sẽ gửi http request ấy lên server và server trả lại một response tương ứng. Khi response được trả về thì service worker xem xét xem dữ liệu trả về có cần được cache lại hay không, nếu có thì cache lại và trả lại response cho người dùng, còn nếu không thì chỉ cần trả response cho người dùng.

- Cache bộ khung HomePage

const cacheName='v1';

var db;

const urlCache=[

'/'

]

);

});

- Tự động cache những file chỉ định gồm: .js, .css, hình ảnh,…

self.addEventListener('fetch',function(e){

var type=['image','style','script'];

var request=e.request;

console.log(request.destination);

var findReponse=caches.open(cacheName)

.then(cache=>cache.match(request))

.then(response=>{

if(response){

return response;

}else{//cache neu tap tin la image hoac cac file tinh

if(type.includes(e.request.destination)){

return fetch(request)

.then(function(res){

return caches.open(cacheName)

.then (cache=>{

cache.put(e.request.url,res.clone());

return res;

})

})

}else{

return fetch(request);

}

}

})

e.respondWith(findReponse);

});

- Tự động xóa cache khi có một phiên bản service worker mới

self.addEventListener('activate', function(event) {

var cacheKeeplist = [];

event.waitUntil(

caches.keys().then(function(keyList) {

return Promise.all(keyList.map(function(key) {

if (cacheKeeplist.indexOf(key) === -1) {

return caches.delete(key);

}

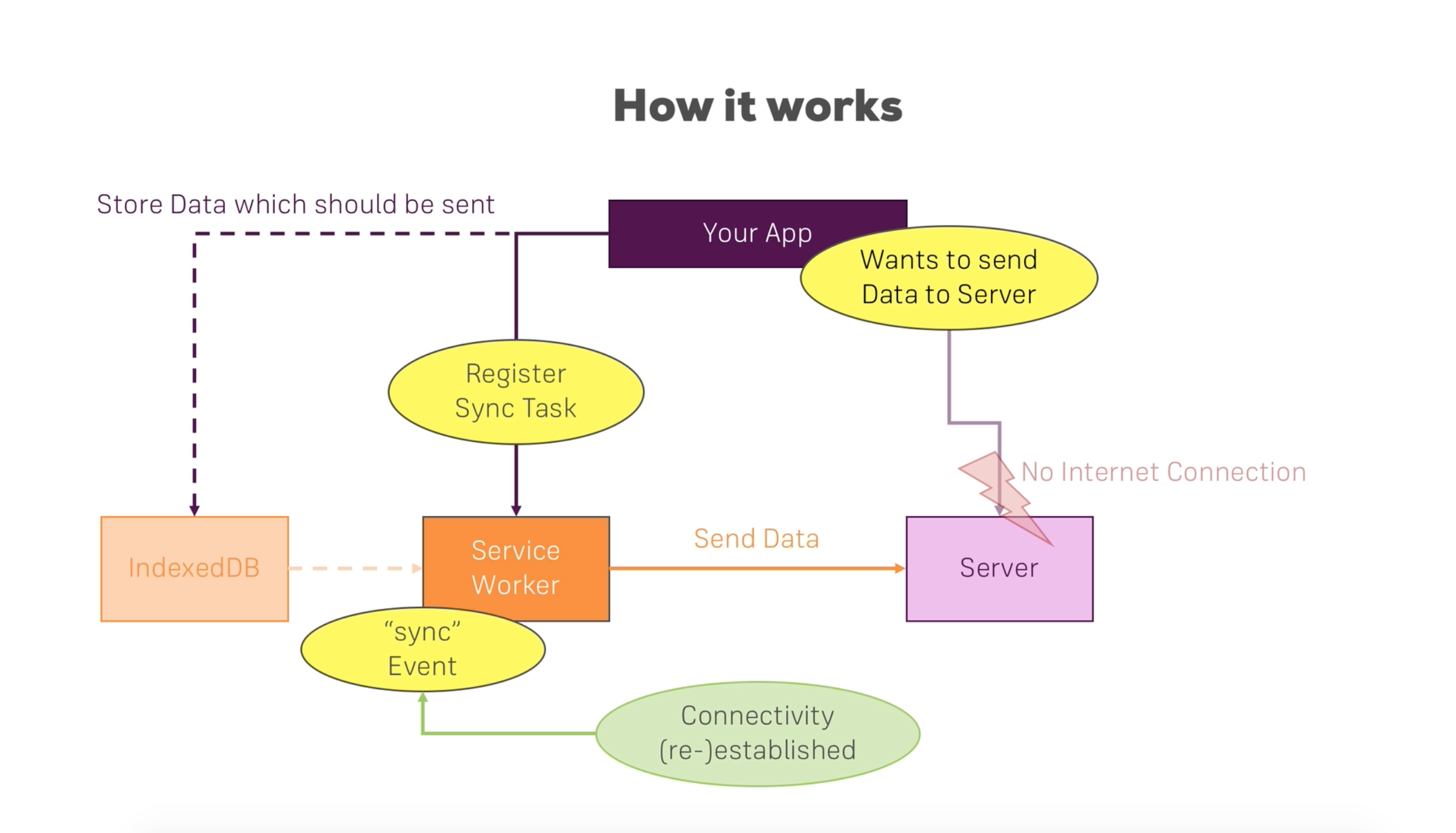
}));

})

);

});

#### 5.3.4. Upload bài viết ở chế độ BackgroundSync



Hình 32 Sơ đồ BackgroundSync

Với chiến lược này thì việc upload dữ liệu lên server sẽ được thực hiện tại một thread riêng không ảnh hưởng tới thread chính của chương trình và cho phép lưu trữ tạm dữ liệu khi mất kết nối internet rồi cho phép tự động thực hiện lại khi có internet trở lại.

Đầu tiên khi Web apps của bạn cần upload một dữ liệu nào đó lên server thì nó sẽ phát sinh một sự kiện “sync” và đồng thời đăng kí với service worker một sync tag nhằm chỉ rõ đó là sự kiện của ai. Tiếp đến dữ liệu sẽ được lưu trữ tại indexedDB. Khi có kết nối internet lập tức sự kiện “sync” của service worker được sinh ra ứng với tag được đăng kí, lúc này service worker sẽ lấy dữ liệu từ indexDB và thực hiện quá trình gửi http request và nhận response tương ứng.

- Bước 1: Lưu dữ liệu từ form vào indexdb.

NewPost/index.js

handledb() {

var request = window.indexedDB.open("PostDatabase", 6);

request.onerror = function (event) {

window.alert('index DB is wrong');

};

// nếu thành công

request.onsuccess = function (event) {

console.log("running onsuccess");

db = event.target.result;

addPost();

};

request.onupgradeneeded = function (event) {

window.alert('add data');

var db = event.target.result;

console.log("running onupgradeneeded");

if (!db.objectStoreNames.contains("posts")) {

var objectStore = db.createObjectStore("posts", { keyPath: "id" });

objectStore.createIndex("title", "title", { unique: false });

objectStore.createIndex("des", "des", { unique: false });

objectStore.createIndex("content", "content", { unique: true });

objectStore.createIndex("view", "", { unique: true });

objectStore.createIndex("category", "category", { unique: true });

objectStore.createIndex("user", "user", { unique: true });

objectStore.createIndex("state", "state", { unique: true });

objectStore.createIndex("img", "img", { unique: true });

}

};

function addPost() {

const medium = document.getElementsByClassName('medium-editor-hidden');

const title = medium[0].value;

const description = medium[1].value;

const content = medium[2].value;

const selected= selectedcategory;

const image=urlimage;

const user = localStorage.getItem('userid');

const data = {

id:"1",

title: title,

des: description,

content: content,

view : 0,

category: selected,

user: user,

state: 1,

img: image

};

//Ask for the objectStore

var objectStore = transaction.objectStore("posts");

var requestdelete=objectStore.clear();

requestdelete.onsuccess=function(event){

window.alert('Xoa thanh cong');

}

requestdelete.onerror=function(){

window.alert('xoa that bai');

}

var request = objectStore.add(data);

request.onsuccess = function (event) {

// event.target.result === customer.ssn;

};

request.onerror = function (event) {

console.log("Error", event.target.error.name);

//some type of error handler

}

}

}

- Bước 2: Gửi thông báo Sync cho service worker.

var transaction = db.transaction(["posts"], "readwrite");

transaction.oncomplete = function (event) {

console.log("All done!");

navigator.serviceWorker.ready

.then(registration => {

registration.sync.register('submit').then(() => {

console.log('sync registered')

});

});

};

- Bước 3: Thực hiện upload dữ liệu từ service worker lên server.

if(e.tag==='submit'){

console.log('sync!');

var request = indexedDB.open("PostDatabase",6);

request.onerror = function(event) {

window.alert('index DB is wrong');

};

request.onsuccess = function(event) {

console.log("running onsuccess");

db = event.target.result;

readpost();

};

}

function readpost(){

db.transaction("posts").objectStore("posts").get("1").onsuccess = function(event) {

let result=event.target.result;

const data = {

"title":result.title,

"des": result.des,

"content": result.content,

"view" : 0,

"category":parseInt(result.category),

"user": parseInt(result.user),

"state": 1,

"img": result.img

};

let url='http://localhost:3001/post/upload';

fetch(url,{

method:'POST',

body:JSON.stringify(data),

headers:{

'Content-Type': 'application/json'

}

}).then(res => res)

.then(response => console.log('Success:', response))

.catch(error => console.error('Error:', error));

};

}

}

#### 5.3.5. Cấu hình Bookmark người dùng

- Bước 1: Gửi thông tin bookmark lên server

Post/index.js

handleBookmarkwithSW=()=>{

const postid = window.location.href.split('/')[4];

const userid = localStorage.getItem('userid');

if (userid) {

axios.post('http://localhost:3001/user/bookmark', { postid, userid }).then(

res => {

const state = res.data[0].state;

if (state) {

alert('Add BookMark Successful!');

this.setState({ bookmark: 'red' });

} else {

alert('Remove BookMark Successful');

this.setState({ bookmark: '' });

}

}

);

}

- Bước 2: Đưa thông tin lên quan xuống indexdb

//Dua xuong indexdb de bat dau dong bo

var request = window.indexedDB.open("PostBookmark", 1);

// var sel = this;

request.onerror = function (event) {

window.alert('index DB is wrong');

};

// nếu thành công

request.onsuccess = function (event) {

console.log("running onsuccess");

db = event.target.result;

addBookmark();

};

request.onupgradeneeded = function (event) {

window.alert('add bookmark');

var db = event.target.result;

console.log("running onupgradeneeded");

if (!db.objectStoreNames.contains("bookmark")) {

var objectStore = db.createObjectStore("bookmark", { keyPath: "ppostid" });

}

};

function addBookmark() {

const data = {

ppostid:postid,

puserid:userid

};

transaction.onerror = function (event) {

// Don't forget to handle errors

console.log('erro');

};

//Ask for the objectStore

var objectStore = transaction.objectStore('bookmark');

var request = objectStore.add(data);

request.onsuccess = function (event) {

// event.target.result === customer.ssn;

};

request.onerror = function (event) {

console.log("Error", event.target.error.name);

//some type of error handler

}

}

} else {

alert('Please Login!');

}

}

- Bước 3: Thông báo sự kiện cho service worker

//Get a transaction

//default for OS list is all, default for type is read

var transaction = db.transaction(["bookmark"], "readwrite");

transaction.oncomplete = function (event) {

console.log("All done!");

navigator.serviceWorker.ready

.then(registration => {

registration.sync.register('bookmark').then(() => {

console.log('sync registered')

});

});

};

- Bước 4: Cache dữ liệu liên quan đến bài viết người dùng đã lưu bookmark

if(e.tag=='bookmark'){

console.log('sync!');

var request = indexedDB.open("PostBookmark",1);

request.onerror = function(event) {

window.alert('index DB is wrong');

};

// nếu thành công

request.onsuccess = function(event) {

console.log("running onsuccess");

db = event.target.result;

readbookmark();

};

}

function readbookmark(){

var objectStore = db.transaction(["bookmark"],"readwrite").objectStore("bookmark");

var postbookmark = [];

objectStore.openCursor().onsuccess = function(event) {

var cursor = event.target.result;

if (cursor) {

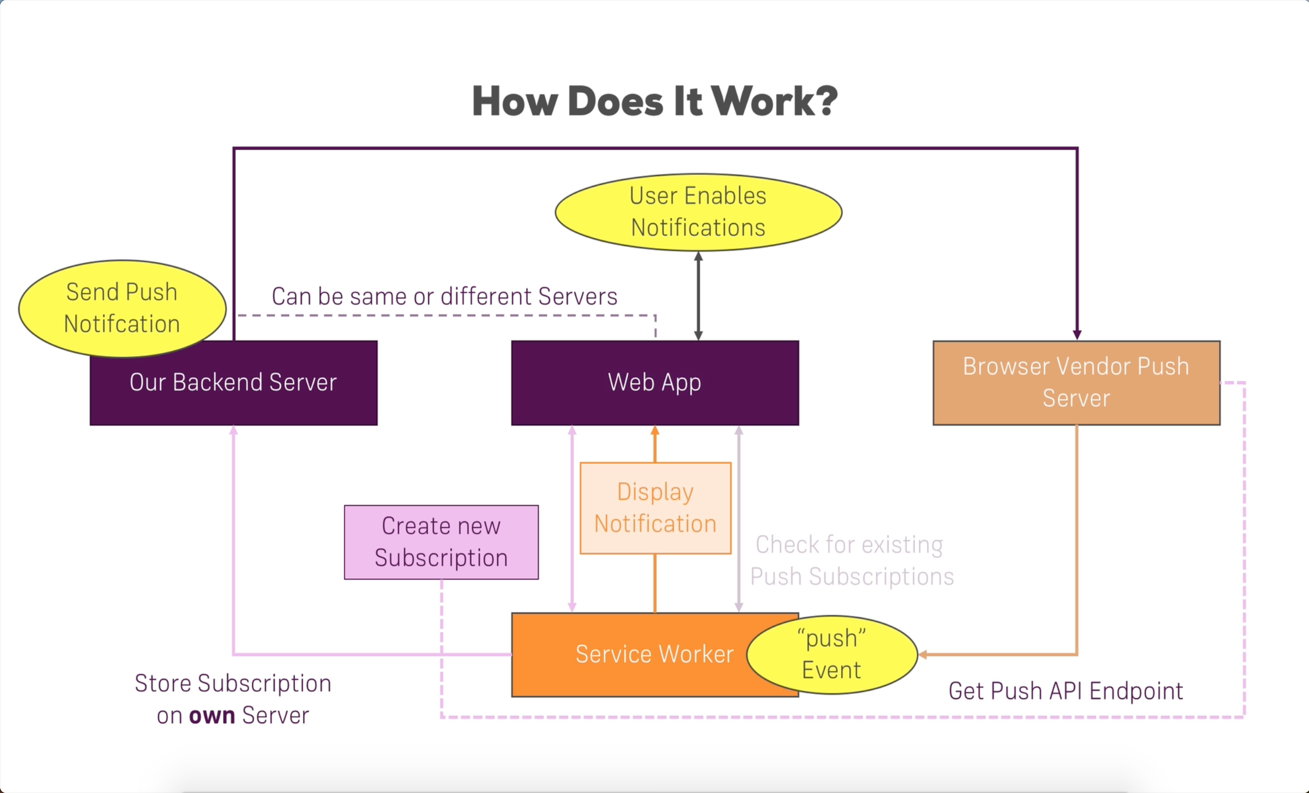
postbookmark.push(cursor.value);

cursor.continue();

}

#### 5.3.6 Cấu hình cho người dùng nhận được thông báo khi có bài viết mới

Sơ đồ hoạt động:



Hình 33 Sơ đồ Push Notification

Với chiến lược này, bất kể người dùng đang ở trong hay đã thoát ra khỏi web apps thì đều nhận được thông báo khi có một sự kiện đã được chỉ định từ server. Đầu tiền người dùng sẽ đăng kí một subscription với server lúc này server sẽ nhận các tham số endpoint, auth, p256dh rồi lưu vào database. Khi có một sự kiện được chỉ định từ server thì server sẽ thực hiện thao tác push notification xuống browser tương ứng, tiếp đến sẽ kích hoạt sự kiện “push” trong service worker và trả về thông báo tương ứng cho người dùng. Lưu ý đây không phải thông báo của browser mà là thông báo của hệ thống, để sử dụng được tính năng này thì người dùng bắt buộc phải kích hoạt quyền notification cho ứng dụng. Tương tự ta cài đặt cho việc khi có một bài mới được upload lên, thì server sẽ gửi một push notification xuống phía service worker, nội dụng của push notification này bào gồm tựa để, đường dẫn tới bài viết khi người dùng nhấn vào thông báo.

Code:

self.addEventListener('notificationclick',function(e){

var notification=e.notification;

var action=e.action;

console.log(notification);

if(action=='confirm'){

console.log('confirm was chosen');

notification.close();

}else{

e.waitUntil(

clients.matchAll()

.then(clis=>{

var client=clis.find(c=>{

return c.visibilityState==='visible';

});

if (client!==undefined){

client.navigate(notification.data.openurl);

client.focus();

}else{

clients.openWindow(notification.data.openurl);

}

notification.close();

})

)

}

});

self.addEventListener('notificationclose',function(event){

console.log('notification was close:',event)

})

self.addEventListener('push',event=>{

console.log('push notification recive');

var data={title:'', content:'',openurl:''};

if(event.data){

data=JSON.parse(event.data.text());

}

var option={

body:data.content,

icon:'/Rocket-icon-blue.png',

image:'/Rocket-icon-blue.png',

data:{

openurl:data.openurl

}

};

event.waitUntil(

self.registration.showNotification(data.title,option)

);

})

connection.query(query2,

[], function (error, results, fields) {

if (error) throw error;

Object.keys(results).forEach(function(key) {

var row = results[key];

var pushConfig={

"endpoint":row.endpoint,

"keys":{

"auth":row.auth,

"p256dh":row.p256dh

}

};

connection.query('select max(id) as id from post',

[],

function (error, results, fields) {

if (error) throw error;

webpush.sendNotification(pushConfig,JSON.stringify({title:'New Post', content:'New post added',openurl:'/post/'+results[0].id}))

.catch(err=>{

console.log(err);

})

});

});

res.json({"ok":"ok"});

});

# KẾT LUẬN

## 1. KIẾN THỨC VÀ KINH NGHIỆM ĐẠT ĐƯỢC

- Hiểu rõ về kiến thức Progressive Web Applications và cách áp dụng nó vào thực tế.

- Tìm hiểu và biết xây dựng một trang web với framework ReactJS.

- Tìm hiểu được cách áp dụng Progressive Web Applications với framework ReactJS để xây dựng được một ứng dụng demo.

- Nâng cao kĩ năng khảo sát hiện trang, phân tích yêu cầu, xác định yêu cầu từ đó thiết kế cơ sở dữ liệu, giao diện và thiết kế xử lý một cách khoa học, bài bản.

- Tìm hiểu thêm một số kiến thức liên quan như ngôn ngữ NodeJS, framework ExpressJS,…

- Luyện tập kĩ năng làm việc nhóm như phân công công việc, lên lịch cụ thể cho các yêu cầu, nghe và nói ý kiến của nhau,…

## 2. KHÓ KHĂN GẶP PHẢI

- Do kiến thức còn nhiều hạn chế và là ngôn ngữ mới đối với nhóm nên trong quá trình thực hiện gặp một vài trục trặc và lỗi.

- Thiết kế giao diện là một trong những điểm khó khăn nữa của nhóm.

- Do thời gian còn hạn hết, nhóm chưa đủ khả năng để trình bày tốt tất cả đề tài này.

## 3. ƯU ĐIỂM VÀ NHƯỢC ĐIỂM

**Ưu điểm**

- Ứng dụng vận dụng tốt các tính năng của PWA.

- Tính năng đơn giản dễ sử dụng.

- Đảm bảo ứng dụng hoạt động mượt mà, hạn chế tối đa lỗi có thể xảy ra.

**Nhược điểm**

- Ứng dụng khá đơn giản, giao diện chưa được sinh động.

- Ứng dụng còn thiếu xót một vài tính năng.

- Áp dụng chưa được nhiều tính năng của PWA.

## 4. HƯỚNG PHÁT TRIỂN ĐỀ TÀI TRONG TƯƠNG LAI

- Tiếp tục tìm hiểu thêm về PWA và framework ReactJS.

- Tìm hiểu và phát triển các tính năng nâng cao hơn trong lĩnh vực đề tài.

- Tùy chỉnh code thêm tường minh và dễ hiểu.

- Xây dựng hoàn thiện các tính năng của ứng dụng Demo không chỉ thể hiện đủ tính chất của đề tài mà còn mang đến ux/ui và có ý nghĩa thực tiễn trong xã hội.

# TÀI LIỆU THAM KHẢO

[1]. <https://developer.mozilla.org/en-US/docs/Web/Apps/Progressive>

[2]. <https://developers.google.com/web/progressive-web-apps>

[3]. <https://reactjs.org>

[4]. Scott Domes, *Progressive Web Apps with React: Create lightning fast web apps with native power using React and Firebase*, Paperback, 2017.

[5]. <https://nodejs.org>

[6]. <https://getbootstrap.com>

[7]. <https://medium.com>

[8]. <https://github.com/yabwe/medium-editor>

[9]. <https://github.com/axios/axios>

[10]. <https://developers.google.com/web/fundamentals/web-app-manifest>

[11]. https://medium.com/@the\_manifest/11-examples-of-progressive-web-apps-944f6db25a5a