

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

ĐỒ ÁN TỐT NGHIỆP

**Hệ thống thu thập, phân tích và thống nhất dữ liệu
người dùng**

NGUYỄN QUANG LỘC
locnq176803@sis.hust.edu.vn

Ngành Công nghệ thông tin
Chương trình tiên tiến Việt Nhật

Giảng viên hướng dẫn: TS. NGUYỄN THANH HÙNG

Chữ kí GVHD

Khoa: Khoa học máy tính

Trường: Công nghệ thông tin và Truyền thông

HÀ NỘI, 08/2022

LỜI CẢM ƠN

Đầu tiên, em xin phép được gửi lời cảm ơn chân thành đến các thầy cô trong trường Đại học Bách Khoa Hà Nội nói chung và các thầy cô trong viện Công nghệ Thông tin và Truyền thông nói riêng, vì những kiến thức, những kỹ năng mà em đã nhận được trong suốt 5 năm ngồi trên ghế nhà trường.

Em xin được gửi lời cảm ơn đến thầy giáo TS. NGUYỄN THANH HÙNG không những đã giúp em định hướng được mục tiêu mà còn tận tình hướng dẫn, giúp đỡ em trong suốt quá trình làm đồ án.

Cuối cùng, xin được gửi lời cảm ơn đến toàn thể bạn bè, những người anh em đã cùng nhau học tập, nghiên cứu, các bạn đã luôn luôn ở bên tôi suốt những năm tháng sinh viên. 5 năm, không quá ngắn nhưng cũng không dài, quãng thời gian đó em đã học được rất nhiều từ các bạn, không chỉ kiến thức mà những kỹ năng sống rất tuyệt vời.

Trong quá trình phát triển sản phẩm cũng như viết báo cáo, mặc dù em đã cố gắng hết sức nhưng vẫn không thể tránh khỏi những sai sót, mong các thầy cô và bạn đọc thông cảm và góp ý để em có thể chỉnh sửa và hoàn thiện hơn nữa sản phẩm này.

TÓM TẮT NỘI DUNG ĐỒ ÁN

Trong thế giới hiện đại ngày nay chúng ta luôn tràn ngập trong 1 số lượng lớn các thông tin từ rất nhiều người khác nhau như tivi, máy tính, sách báo, truyền thông, người thân, bạn bè, John Naisbitt- nhà tương lai học đã từng nhận định rằng: Chúng ta chết đuối trong thông tin, nhưng chết đói về tri thức. Nhờ các phương tiện truyền thông, sự phát triển của công nghệ, hàng ngày chúng ta cập nhật rất nhiều thông tin khác nhau (thậm chí bị bão hòa thông tin) nhưng thường lại là thông tin bề nổi, mà thiếu sự hiểu sâu và bản chất vấn đề.

Vì vậy mỗi người cần xác định đâu là thông tin hữu ích với mình và nên tiếp cận nó từ đâu. Và đối với các doanh nghiệp các marketer cũng vậy họ cũng cần biết đâu là thông tin hữu dụng với mình và đó là thông tin và người dùng. Mặc dù nghe có vẻ hiển nhiên, nhưng việc thấu hiểu khách hàng của bạn là ai, họ ở đâu, họ muốn gì và cách họ sử dụng sản phẩm của bạn là cực kì có giá trị khi nói đến việc mở rộng quy mô kinh doanh. Những thông tin chi tiết này có thể khó thu thập, nhưng cho phép bạn phân khúc khách hàng và cải thiện hiệu quả công ty một cách dễ dàng. Theo dõi siêu dữ liệu khách hàng giúp bạn xác định các cơ hội để cải thiện hiệu quả bán hàng, mở rộng doanh thu và tăng tỷ lệ giữ chân khách hàng.

Với mong muốn giúp các doanh nghiệp hiểu rõ người dùng của họ hơn, đồ án tốt nghiệp của em tập trung vào việc xây dựng 1 hệ thống thu thập các dữ liệu của người dùng khi họ truy cập vào website của doanh nghiệp bằng trình duyệt. Các dữ liệu của người dùng sau khi được thu thập sẽ được biểu diễn bằng các biểu đồ, sơ đồ, bảng biểu trực quan dễ tương tác đem lại cho các doanh nghiệp 1 cách nhìn mới mẻ về người dùng của họ. Sau đó các doanh nghiệp có thể tiến hành thống nhất các dữ liệu đó và dữ liệu của chính hệ thống của mình để tạo ra 1 dữ liệu hoàn chỉnh thống nhất về từng người dùng. Sau khi đã có được cái nhìn sâu sắc về thông tin địa lý, hành vi, sở thích, thói quen của khách hàng các doanh nghiệp có thể tiến hành nâng cấp website tạo trải nghiệm tốt hơn cho người dùng. Không những vậy họ còn có thể tiến hành các chiến dịch marketing, quảng cáo nhằm vào 1 đối tượng khách hàng nhất định. Vì nắm rõ thông tin người dùng nên các chiến dịch đó sẽ đạt kết quả cao hơn và người dùng cũng cảm thấy vui vẻ khi nhận được các thông tin về sản phẩm mình yêu thích.

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI.....	1
1.1 Đặt vấn đề.....	1
1.2 Mục tiêu và phạm vi đề tài.....	1
1.3 Định hướng giải pháp.....	2
1.4 Bố cục đồ án	2
CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU.....	3
2.1 Khảo sát hiện trạng	3
2.2 Tổng quan chức năng	3
2.2.1 Biểu đồ use case tổng quát	3
2.2.2 Biểu đồ use case phân rã View Project Report	4
2.2.3 Biểu đồ use case phân rã Activate Data	4
2.3 Đặc tả chức năng	5
2.3.1 Đặc tả use case Setup tracking package	5
2.3.2 Đặc tả use case View Project Report	6
2.3.3 Đặc tả use case Unify data	7
2.3.4 Đặc tả use case Send Email.....	8
2.3.5 Đặc tả use case Send Notification	9
2.3.6 Đặc tả use case Send SMS	10
CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG.....	11
3.1 Công nghệ FrontEnd.....	11
3.1.1 ReactJS	11
3.1.2 TailwindCSS	12
3.1.3 Context API.....	12
3.1.4 Các thư viện khác.....	13

3.2 Công nghệ BackEnd	14
3.2.1 NodeJS	14
3.2.2 ExpressJS.....	14
3.2.3 MySQL.....	15
3.2.4 Prisma	15
3.2.5 GraphQL.....	15
3.2.6 Các thư viện khác.....	16
CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNG GIÁ	18
4.1 Thiết kế kiến trúc.....	18
4.1.1 Thiết kế tổng quan.....	18
4.1.2 Thiết kế kiến trúc frontend.....	19
4.1.3 Thiết kế kiến trúc backend	20
4.2 Thiết kế chi tiết.....	21
4.2.1 Thiết kế giao diện	21
4.2.2 Thiết kế cơ sở dữ liệu	26
4.3 Xây dựng ứng dụng.....	29
4.3.1 Thư viện và công cụ sử dụng.....	29
4.3.2 Kết quả đạt được	29
4.3.3 Minh họa các chức năng chính	30
4.4 Kiểm thử.....	34
4.5 Triển khai	35
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	36
5.1 Các giải pháp và đóng góp nổi bật.....	36
5.2 Kết luận	38
5.3 Hướng phát triển.....	38
TÀI LIỆU THAM KHẢO.....	39

DANH MỤC HÌNH VẼ

Hình 2.1	Biểu đồ use case tổng quát	3
Hình 2.2	Biểu đồ use case phân rã View Project Reportt	4
Hình 2.3	Biểu đồ use case phân rã Activate Data	4
Hình 4.1	Kiến trúc hệ thống	18
Hình 4.2	Kiến trúc Frontend	19
Hình 4.3	Kiến trúc Backend	20
Hình 4.4	Màn hình dashboard overview	21
Hình 4.5	Màn hình geographic report	21
Hình 4.6	Màn hình techonographic report	22
Hình 4.7	Màn hình behavioral report	22
Hình 4.8	Màn hình send email	23
Hình 4.9	Màn hình send notification	23
Hình 4.10	Màn hình send sms	24
Hình 4.11	Màn hình create project	24
Hình 4.12	Màn hình user profile	25
Hình 4.13	Màn hình chức năng xem báo cáo tổng quát	30
Hình 4.14	Màn hình chức năng xem báo cáo về địa lý	30
Hình 4.15	Màn hình chức năng xem báo cáo về hành vi	31
Hình 4.16	Màn hình chức năng tải về báo cáo	31
Hình 4.17	Màn hình báo cáo được tải về	31
Hình 4.18	Màn hình thể hiện dữ liệu đã được thống nhất của người dùng	32
Hình 4.19	Màn hình thể hiện chức năng gửi thông báo	32
Hình 4.20	Màn hình thể hiện chức năng gửi email	33
Hình 4.21	Màn hình hiển thị email đã được gửi	33

DANH MỤC BẢNG BIỂU

Bảng 2.1	Đặc tả use case Setup tracking package	5
Bảng 2.2	Đặc tả use case View Project Report	6
Bảng 2.3	Đặc tả use case Unify data	7
Bảng 2.4	Đặc tả use case Send Email	8
Bảng 2.5	Đặc tả use case Send Notification	9
Bảng 2.6	Đặc tả use case Send SMS	10
Bảng 4.1	Thiết kế chi tiết bảng Customer	26
Bảng 4.2	Thiết kế chi tiết bảng Project	26
Bảng 4.3	Thiết kế chi tiết bảng User	26
Bảng 4.4	Thiết kế chi tiết bảng UsersOnProjects	26
Bảng 4.5	Thiết kế chi tiết bảng Event	27
Bảng 4.6	Thiết kế chi tiết bảng Notification	28
Bảng 4.7	Thiết kế chi tiết bảng UsersReceiveNotification	28
Bảng 4.8	Thiết kế chi tiết bảng Email	28
Bảng 4.9	Thiết kế chi tiết bảng UsersReceiveEmail	28
Bảng 4.10	Danh sách thư viện và công cụ sử dụng	29
Bảng 4.11	Bảng thống kê các thông số của hệ thống	29
Bảng 4.12	Đặc tả use case Setup tracking package	34
Bảng 4.13	Đặc tả use case Setup tracking package	34
Bảng 4.14	Đặc tả use case Setup tracking package	35
Bảng 4.15	Chi tiết cấu hình hệ thống	35

DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
API	Giao diện lập trình ứng dụng (Application Programming Interface)
CNTT	Công nghệ thông tin
CSS	Ngôn ngữ mô tả giao diện và định dạng(Cascading Style Sheets)
DOM	Mô hình Các Đối tượng Tài liệu(Document Object Model)
HTML	Ngôn ngữ đánh dấu siêu văn bản (HyperText Markup Language)
RDBMS	Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System)
SMS	Dịch vụ tin nhắn (Short Message Service)

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

1.1 Đặt vấn đề

Trong thời đại hiện nay thì dữ liệu rất quan trọng, từ dữ liệu thu thập được có thể suy ra được nhiều thông tin có giá trị. Vì vậy dữ liệu của người dùng truy cập vào 1 website luôn được các quản trị viên website quan tâm. Thu thập dữ liệu rất hữu ích cho các doanh nghiệp muốn hiểu rõ hơn về hành vi của người dùng. Nó cung cấp cho các doanh nghiệp cái nhìn sâu sắc về hành vi của khách hàng trên các trang web, các bài viết trên blog của họ, v.v. để họ có thể mang lại trải nghiệm được nhắm mục tiêu, được cá nhân hóa và thú vị. Tuy rằng hiện nay có nhiều công cụ giúp thu thập dữ liệu người dùng nhưng những dữ liệu đó vẫn còn rời rạc và chưa được khai thác hết các giá trị ẩn sâu từ người dùng.

1.2 Mục tiêu và phạm vi đề tài

Dựa vào những vấn đề đã phân tích nêu trên, mục tiêu của đề án sẽ bao gồm 4 mục dưới đây: Thứ nhất hệ thống của em sẽ tiến hành thu nhập các dữ liệu của người dùng truy cập vào trang web của khách hàng, các dữ liệu thu thập được sẽ được chia làm 3 nhóm chính: dữ liệu về vị trí địa lý; Dữ liệu về thiết bị, công nghệ; Dữ liệu về hành vi. Đặc biệt các dữ liệu sẽ thông qua 1 hàm hash để tạo ra 1 ID (browser fingerprint) giúp nhận diện người dùng kể cả khi họ chưa đăng nhập.

Thứ hai hệ thống sẽ dựa trên những dữ liệu đã thu thập được tiến hành phân tích, thống kê các dữ liệu, xây dựng nên các biểu đồ mô hình. Các biểu đồ mô hình này sẽ giúp người quản trị website biết được số lượng người truy cập hệ thống, đó là những người như thế nào, họ truy cập vào những trang nào và họ đến từ đâu,Không những thế hiệu năng của hệ thống những chỗ tốt và chưa tốt cũng được đánh giá.

Thứ ba hệ thống sẽ thống nhất các dữ liệu về người dùng thu thập được với dữ liệu về người dùng của hệ thống khách hàng. Từ đó đưa ra 1 cái nhìn tổng quan cho người quản trị website về các người dùng của mình. Ở bước này các dữ liệu về hành vi của người dùng ở các website khác cũng sẽ được thống nhất.

Thứ tư sau khi có được dữ liệu tổng quát thống nhất về người dùng, người quản trị có thể tiến hành phân loại người dùng theo vị trí địa lý, công nghệ, hành vi, sở thích, mối quan tâm... rồi tiến hành các chiến dịch marketing, quảng cáo,... đến các người dùng tương ứng thông qua email, thông báo và sms.

1.3 Định hướng giải pháp

Từ việc xác định rõ nhiệm vụ cần giải quyết ở phần 1.2, em xin được đề xuất định hướng giải pháp của mình như sau: Em sẽ chia hệ thống của mình thành 3 module nhỏ:

(i) Module thứ nhất có chức năng thu thập dữ liệu người dùng được viết dưới dạng 1 package để có thể nhúng vào trang web của khách hàng. Module này sẽ thu thập các dữ liệu sẵn có từ trình duyệt, các dữ liệu được người dùng tạo ra khi truy cập trang web và gửi về phía module thứ hai.

(ii) Module thứ 2 là 1 server được viết bằng NodeJs có 2 nhiệm vụ chính: Một là nhận dữ liệu thu thập được từ module thứ nhất và lưu vào cơ sở dữ liệu. Hai là đảm nhận việc xử lý các request đến từ module thứ 3.

(iii) Module thứ 3 là hệ thống được viết bằng ReactJs chạy trên trình duyệt giúp khách hàng tương tác được với hệ thống 1 cách trực quan.

Đi theo định hướng giải pháp đó hệ thống sẽ đem lại cho các doanh nghiệp các dữ liệu, thông tin về người sử dụng, từ đó hiểu hơn về người sử dụng và có thể tiến hành các chiến dịch marketing, quảng cáo 1 cách hiệu quả nhất.

1.4 Bố cục đồ án

Phần còn lại của báo cáo đồ án tốt nghiệp này được tổ chức như sau.

Chương 2 trình bày về vấn đề khảo sát hiện trạng và yêu cầu của hệ thống dựa trên các hệ thống đã có. Sau đó em sẽ trình bày về các chức năng chính của hệ thống bằng biểu đồ use case tổng quát. Rồi tiếp đó sẽ đi sâu phân tích 1 số use case quan trọng của hệ thống.

Trong Chương 3, em xin được giới thiệu về các công nghệ, nền tảng sử dụng trong đồ án.

Tiếp sau là Chương 4, chương này em sẽ trình bày về thiết kế của hệ thống bao gồm: Thiết kế về kiến trúc của hệ thống, thiết kế về giao diện và thiết kế về cơ sở dữ liệu.

Và cuối cùng, trong chương 5 em sẽ trình bày về sẽ nói về các vấn đề khó khăn gặp phải khi thực hiện đồ án lần này và những điều đã làm được và chưa làm được trong đồ án cũng như là hướng phát triển đồ án trong tương lai.

CHƯƠNG 2. KHẢO SÁT VÀ PHÂN TÍCH YÊU CẦU

2.1 Khảo sát hiện trạng

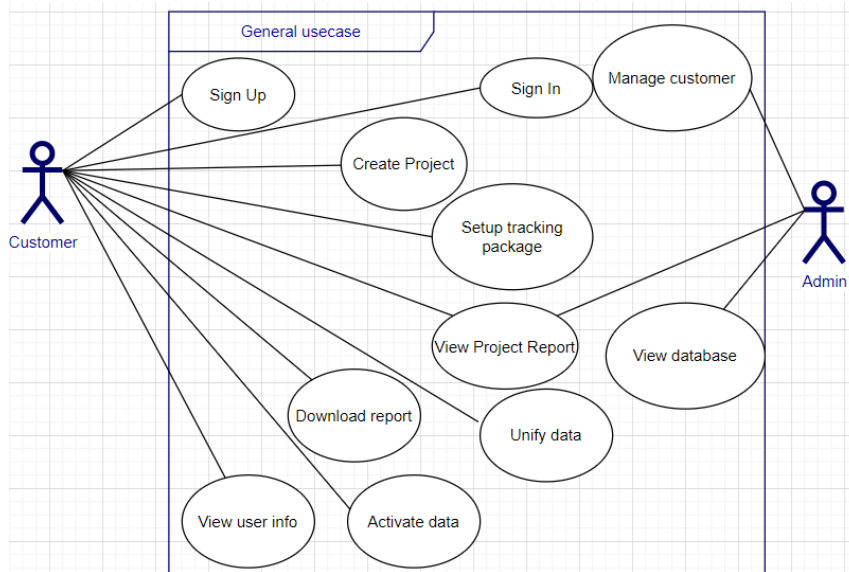
Sau khi khảo sát 1 số phần mềm, ứng dụng thu thập và phân tích dữ liệu của người dùng tiêu biểu như Google Analytics, Matomo, KeenJs,.. thì em nhận thấy hầu như các hệ thống này sau khi thu thập dữ liệu thì chỉ đưa ra các báo cáo bằng text và đồ thị, mô hình về các dữ liệu đã thu thập được mà chưa biết làm gì tiếp theo với lượng dữ liệu đã thu thập được. Bên cạnh đó thì các dữ liệu khá rời rạc chưa có sự thống nhất.

Bên cạnh đó cũng có 1 số hệ thống như Adobe Analytics, Exponea,.. có phân tích hợp marketing, xử lý dữ liệu nhưng khá là phức tạp và lằng nhằng.

Với mong muốn tạo được hệ thống có quy mô vừa phải, phù hợp với các cá nhân, doanh nghiệp nhỏ, thì em đã phát triển hệ thống này.

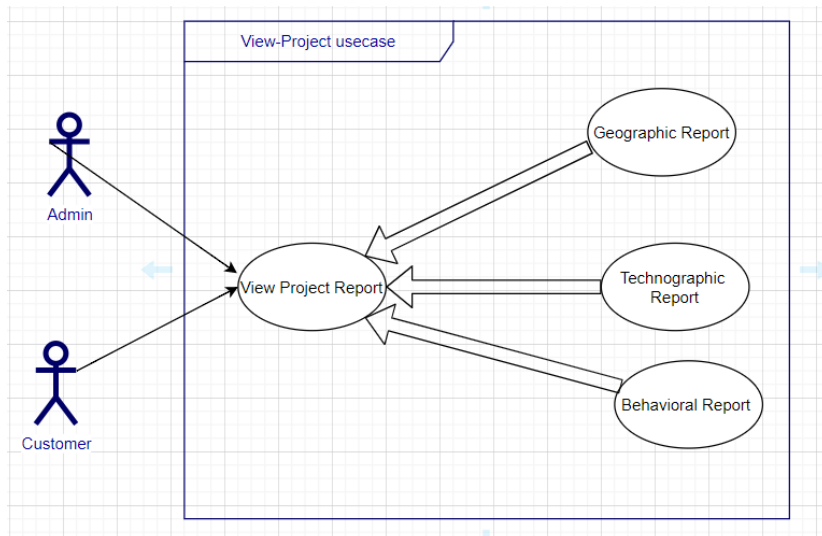
2.2 Tổng quan chức năng

2.2.1 Biểu đồ use case tổng quát



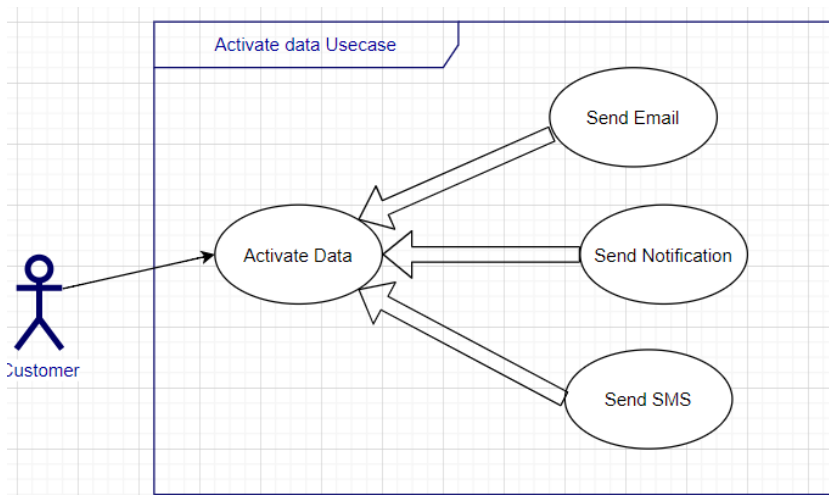
Hình 2.1: Biểu đồ use case tổng quát

2.2.2 Biểu đồ use case phân rã View Project Report



Hình 2.2: Biểu đồ use case phân rã View Project Reportt

2.2.3 Biểu đồ use case phân rã Activate Data



Hình 2.3: Biểu đồ use case phân rã Activate Data

2.3 Đặc tả chức năng

2.3.1 Đặc tả use case Setup tracking package

Bảng 2.1: Đặc tả use case Setup tracking package

Tên Use case:	Setup tracking package
Tác nhân	Customer
Mô tả:	Khách hàng cài đặt package thu thập dữ liệu vào website của mình.
Tiền điều kiện:	<ol style="list-style-type: none"> 1. Khách hàng cần tạo tài khoản. 2. Khách hàng cần tạo project trên hệ thống, yêu cầu cung cấp domain của website để xác thực.
Hậu điều kiện:	
Kích hoạt:	Không
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Khách hàng tạo project trên hệ thống ; 2. Khách hàng nhập link package vào website của mình; 3. Package tiến hành thu thập dữ liệu và gửi về server nếu như domain của website đã được đăng ký trong cơ sở dữ liệu. 4. Khách hàng vào hệ thống và xác nhận xem hệ thống đã nhận được dữ liệu từ package chưa
Luồng sự kiện thay thế:	<ol style="list-style-type: none"> 1. Khách hàng kiểm tra lại các bước cài đặt package đã đúng chưa ; 2. Khách hàng kiểm tra lại domain đã đăng ký với domain website có giống nhau không;

2.3.2 Đặc tả use case View Project Report

Bảng 2.2: Đặc tả use case View Project Report

Tên Use case:	View Project Report
Tác nhân	Customer, Admin
Mô tả:	Xem các báo cáo (mô hình, biểu đồ) về các dữ liệu đã thu thập được của người dùng .
Tiền điều kiện:	1. Tác nhân cần đăng nhập vào hệ thốngthống dụng.
Hậu điều kiện:	
Kích hoạt:	Không
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Tác nhân truy cập vào dashboard ; 2. Tác nhân chọn project cần xem; 3. Tác nhân chọn loại báo cáo muốn xem (Overview, Geographic, Behavioral);
Luồng sự kiện thay thế:	Không

2.3.3 Đặc tả use case Unify data

Bảng 2.3: Đặc tả use case Unify data

Tên Use case:	Unify data
Tác nhân	Customer
Mô tả:	Khách hàng hợp nhất dữ liệu của người dùng ở hệ thống và dữ liệu người dùng trên website của mình .
Tiền điều kiện:	Website của khách hàng cần có hệ thống xác thực người dùng và cần có id của người dùng đã đăng nhập
Hậu điều kiện:	
Kích hoạt:	Không
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Khách hàng truy cập vào phần Unify Data ; 2. Khách hàng làm theo hướng dẫn và nhúng đoạn code có tác dụng gửi id của người dùng về server ; 3. Hệ thống xác định những người dùng sau khi đăng nhập website là cùng 1 người dùng. 4. Hệ thống tiến hành hợp nhất dữ liệu của những người dùng đó thành 1 thể thống nhất. 5. Khách hàng xác nhận thông tin của những người dùng đã được thống nhất dữ liệu trên hệ thống .
Luồng sự kiện thay thế:	Khách hàng kiểm tra lại các bước cài đặt trên website đã đúng hay.

2.3.4 Đặc tả use case Send Email

Bảng 2.4: Đặc tả use case Send Email

Tên Use case:	Send Email
Tác nhân	Customer
Mô tả:	Khách hàng tiến hành các chiến dịch quảng cáo, marketing thông qua mail .
Tiền điều kiện:	Phải thu thập được địa chỉ email của người dùng.
Hậu điều kiện:	
Kích hoạt:	Không
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Khách hàng truy cập vào phần Send Email trên hệ thống ; 2. Khách hàng viết tiêu đề và nội dung của mail ; 3. Khách hàng lọc những người dùng mình muốn gửi mail; 4. Hệ thống trả về số lượng người dùng đã được lọc; 5. Khách hàng xác nhận có muốn gửi email cho những người dùng đó hay không; 6. Hệ thống gửi mail đến những người dùng đó; 7. Khách hàng xác nhận tình trạng của email;
Luồng sự kiện thay thế:	<ol style="list-style-type: none"> 1. Khách hàng kiểm tra lại package đã được cài đặt và chạy ổn định hay chưa ; 2. Khách hàng kiểm tra lại địa chỉ email của người dùng đã được lưu và gửi về hệ thống chưa ;

2.3.5 Đặc tả use case Send Notification

Bảng 2.5: Đặc tả use case Send Notification

Tên Use case:	Send Notification
Tác nhân	Customer
Mô tả:	Khách hàng tiến hành các chiến dịch quảng cáo, marketing thông qua thông báo .
Tiền điều kiện:	Người dùng phải cho phép hiển thị thông báo trên trình duyệt.
Hậu điều kiện:	
Kích hoạt:	Không
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Khách hàng truy cập vào phần Send Notification trên hệ thống ; 2. Khách hàng viết tiêu đề và nội dung của thông báo ; 3. Khách hàng lọc những người dùng mình muốn gửi thông báo ; 4. Hệ thống trả về số lượng người dùng đã được lọc ; 5. Khách hàng xác nhận có muốn gửi thông báo cho những người dùng đó hay không; 6. Hệ thống gửi thông báo đến những người dùng đó; 7. Khách hàng xác nhận tình trạng của thông báo .
Luồng sự kiện thay thế:	Khách hàng kiểm tra lại package đã được cài đặt và chạy ổn định hay chưa ;

2.3.6 Đặc tả use case Send SMS

Bảng 2.6: Đặc tả use case Send SMS

Tên Use case:	Send SMS
Tác nhân	Customer
Mô tả:	Khách hàng tiến hành các chiến dịch quảng cáo, marketing thông qua SMS .
Tiền điều kiện:	Phải thu thập được số điện thoại của người dùng.
Hậu điều kiện:	
Kích hoạt:	Không
Luồng sự kiện chính:	<ol style="list-style-type: none"> 1. Khách hàng truy cập vào phần Send SMS trên hệ thống ; 2. Khách hàng viết nội dung của SMS ; 3. Khách hàng lọc những người dùng mình muốn gửi SMS; 4. Hệ thống trả về số lượng người dùng đã được lọc; 5. Khách hàng xác nhận có muốn gửi SMS cho những người dùng đó hay không; 6. Hệ thống gửi mail đến những người dùng đó; 7. Khách hàng xác nhận tình trạng của SMS;
Luồng sự kiện thay thế:	<ol style="list-style-type: none"> 1. Khách hàng kiểm tra lại package đã được cài đặt và chạy ổn định hay chưa ; 2. Khách hàng kiểm tra lại số điện thoại của người dùng đã được lưu và gửi về hệ thống chưa ;

CHƯƠNG 3. CÔNG NGHỆ SỬ DỤNG

3.1 Công nghệ FrontEnd

3.1.1 ReactJS

SSR (Server side rendering) là kỹ thuật không còn xa lạ gì với các bạn lập trình viên nữa. Nó có từ từ thừa sơ khai của website, và còn được sử dụng rộng rãi cho đến ngày nay và chưa có dấu hiệu bị thay thế hoàn toàn. Nhưng càng ngày, những trang web càng giống ứng dụng hơn là nơi hiển thị nội dung, bạn có thể xử lý ảnh, chat, soạn thảo văn bản ... Dần dần cách thức hoạt động của SSR tỏ ra nặng nề hơn và bộc lộ những nhược điểm nhất định. Đầu tiên nói về cách hoạt động của SSR: Đầu tiên: Browser sẽ gửi một request lên web server; Tiếp theo: Web server sẽ trả về toàn bộ code bao gồm html js css đã được render sẵn; Lúc này nội dung HTML đã hoàn chỉnh và được hiển thị ngay khi nó được load về máy. Tuy nhiên lúc này trang web mới chỉ xem được nội dung, người dùng chưa thể tương tác được. Ngay trong lúc này, trình duyệt vẫn âm thầm tải tiếp JS và thực thi nó ngay khi hoàn tất, một khi hoàn tất quá trình này người dùng mới có thể tương tác được.

Phương pháp này có nhiều lợi ích trong đó phải kể đến việc load website lần đầu tiên khá là nhanh giúp người dùng không phải chờ đợi lâu. Tiếp đến là khả năng tối ưu SEO, website của bạn do được render trên server nên khi những công cụ tìm kiếm như Google, Bing, ... crawl qua thì sẽ đọc được hết nội dung website. Bên cạnh đó thì còn có cả những điểm mà nó chưa làm tốt. Có thể kể đến như, do render trên server nên thời gian phản hồi TTFB (Time To First Byte) sẽ chậm. Web server của bạn sẽ tốn băng thông hơn vì truyền tải lượng thông tin lớn hơn. Mọi thay đổi dù nhỏ nhất đều phải load lại trang, do việc xử lý render do server đảm nhận.

CSR ra đời sau SSR để giải quyết các vấn đề bất cập ở SSR. Ngược lại với SSR, CSR thực hiện rendering chủ yếu ở phía client. Phía client thì chỉ nhận được 1 trang html trống và 1 tệp tin javascript sau đó trình duyệt sẽ bắt đầu chạy tệp tin Reactjs, và xong xuôi hết thì mới hiển thị giao diện ra cho người dùng. Dù cho lần đầu load sẽ hơi lâu nhưng những lần load trang tiếp theo sẽ nhanh hơn nhiều và đem lại trải nghiệm người dùng tốt hơn nhiều so với SSR. ReactJS là 1 thư viện javascript tiên phong trong việc sử dụng công nghệ CSR đó để nâng cao trải nghiệm và kết quả đã được chứng minh qua facebook (Cha đẻ của ReactJS).

ReactJS là một thư viện Javascript với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác. Nếu như AngularJS là một Framework cho phép nhúng code javascript trong

code html thông qua các attribute như ng-model, ng-repeat...thì với react là một library cho phép nhúng code html trong code javascript nhờ vào JSX, bạn có thể dễ dàng lồng các đoạn HTML vào trong JS. Tích hợp giữa javascript và HTML vào trong JSX làm cho các component dễ hiểu hơn. ReactJs cũng nổi tiếng với công nghệ DOM ảo giúp tăng hiệu năng cho ứng dụng. Việc chỉ node gốc mới có trạng thái và khi nó thay đổi sẽ tái cấu trúc lại toàn bộ, đồng nghĩa với việc DOM tree cũng sẽ phải thay đổi một phần, điều này sẽ ảnh hưởng đến tốc độ xử lý. React JS sử dụng Virtual DOM (DOM ảo) để cải thiện vấn đề này. Virtual DOM là một object Javascript, mỗi object chứa đầy đủ thông tin cần thiết để tạo ra một DOM, khi dữ liệu thay đổi nó sẽ tính toán sự thay đổi giữa object và tree thật, điều này sẽ giúp tối ưu hoá việc re-render DOM tree thật.

3.1.2 TailwindCSS

Với nền tảng web hiện đại bây giờ có một yếu tố rất quan trọng ảnh hưởng trực tiếp đến sự thành công của trang web đấy chính là giao diện. Trong quá trình phát triển chúng ta đã quá quen với những framework như là Bootstrap, Foundation, Material UI, Ant Design, ...Nhưng những framework này có khả năng tùy biến không cao, gần như nếu không thay đổi gì thì gần như là các trang web sẽ có giao diện y hệt mặc định của framework. Còn nếu muốn tùy biến lại giao diện ta lại phải viết lại các class và viết lại các thuộc tính css cho chúng. Nhưng với TailwindCSS, chỉ dựa vào các class đã được tạo ra ta vẫn có thể tùy biến các thuộc tính css 1 cách tự do, nhanh chóng và không cần tạo thêm class cũng như viết thêm file css.

TailwindCSS là một utility-first CSS framework, nó cũng giống như Bootstrap, nó có những class built-in mà chúng ta có thể dùng. Tailwind CSS có nhiều các class bao gồm các thuộc tính CSS khác nhau và quan trọng, chúng ta có thể dễ dàng mở rộng tạo mới ra những class bằng chính những class của nó.

Ưu điểm nhìn thấy rõ nhất của Tailwind CSS đấy là:Người sử dụng có thể không cần phải viết đến 1 dòng css nào mà vẫn có giao diện tùy biến theo mong muốn;Cách đặt tên class dễ hiểu, 1 class đại diện cho 1 thuộc tính. Tailwind CSS có gần như đủ gần 85 phần trăm thuộc tính css.Dễ cài đặt, dễ sử dụng, document của Tailwind rất dễ hiểu. Tailwind CSS phù hợp cho các dự án nhỏ, người dùng tùy biến nhiều, cần làm nhanh giao diện.

3.1.3 Context API

State là thế mạnh của react nhưng nó cũng đem lại nhiều sự phiền phức. Thông thường các state sẽ được truyền qua các component thông props của component con. Nếu ứng dụng nhỏ thì không vấn đề gì nhưng nếu ứng dụng to và gồm nhiều component thì việc truyền state qua hàng chục component hay truyền state từ 1

nhánh virtual dom này tới virtual dom khác thì thật là khó khăn. Từ đó các công nghệ để quản lý state đã được sinh ra, phải kể đến như Redux, vuex, recoil, zustand, ... trong số đó phải kể đến Context API.

React Context tồn tại để chúng ta cần truyền dữ liệu một cách thủ công bằng việc sử dụng props ở tất cả các cấp của component. Context chia sử dữ liệu cho nhiều các component khác nhau. Việc truyền dữ liệu từ component cha xuống component con thông qua props là tương đối dài dòng và khó kiểm soát so với việc sử dụng Context API. Bằng việc sử dụng Context API, chúng ta không còn cần phải truyền các dữ liệu muốn chia sẻ với nhau thông qua việc dùng props.

3.1.4 Các thư viện khác

ChartJs Đây là một trong những dự án mã nguồn mở giúp cho mọi người có thể vẽ những biểu đồ thể hiện số liệu trên website một cách dễ dàng và đẹp nhất. Dự án này hiện tại đã có đến hơn 57.700 stars và 4200 lượt commit trên Github và được cập nhật thường xuyên. 4 điểm mạnh nhất của Chart.js là: – Dự án mã nguồn mở: cả cộng đồng phát triển và khắc phục lỗi. – Tương thích tốt với HTML 5 cái này gần như bắt buộc ở hiện tại – Hơn 8 kiểu biểu đồ phổ biến nhất hiện nay – Responsive: hiển thị đẹp nhất trên tất cả các thiết bị từ Desktop, Tablet, Mobile. Có 3 bước cơ bản trong việc tạo một biểu đồ bằng thư viện Chart.js. – Bước 1: khai báo thư viện Chart.js và BootstrapCDN – Bước 2: Tạo một thẻ với <canvas> bên trong để hứng biểu đồ – Bước 3: Tùy biến biểu đồ và thay đổi số liệu

Apollo client Apollo Client là một thư viện quản lý state toàn diện cho JavaScript cho phép bạn quản lý cả dữ liệu ylocal(data ở client) và remote(data liên quan đến server) với GraphQL. Sử dụng nó để fetch, cache và modify dữ liệu ứng dụng, tất cả tự động cập nhật lại UI.

Apollo Client giúp bạn cấu trúc code theo cách tiết kiệm, có thể dự đoán và khai báo phù hợp với thực tiễn phát triển hiện đại. Thư viện core @apollo/client cung cấp tích hợp sẵn với React và cộng đồng Apollo lớn hơn duy trì tích hợp cho các lớp chế độ xem phổ biến khác.

Font Awesome Trước đây để tạo các ICON đó ta phải thực hiện cắt chúng ra từ file PSD và dùng CSS để gán background hoặc dùng thẻ img để đưa ICON đó vào. Tuy nhiên hiện nay chúng ta đã có giải pháp khác đó là sử dụng Symbol Font (font chữ kiểu ký hiệu). Hiện nay có nhiều thư viện Symbol Font và Font Awesome là 1 thư viện cực kỳ nổi tiếng và được nhiều người sử dụng trong số đó.

3.2 Công nghệ BackEnd

3.2.1 NodeJS

NodeJS (hay Node) không phải là một ngôn ngữ, thư viện hay một framework, Node là môi trường thực thi (Runtime Environment) cho JavaScript. Mặc dù JS thường cần trình duyệt để hoạt động, nhưng Node.js tạo các cài đặt phù hợp để JS chạy bên ngoài trình duyệt.

Các đặc điểm của NodeJs phải kể đến như: 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 bằng 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);

3.2.2 ExpressJS

Expressjs là một framework mã nguồn mở miễn phí cho Node.js. Express.js được sử dụng trong thiết kế và xây dựng các ứng dụng web một cách đơn giản và nhanh chóng.

Vì Express js chỉ yêu cầu ngôn ngữ lập trình Javascript nên việc xây dựng các ứng dụng web và API trở nên đơn giản hơn với các lập trình viên và nhà phát triển. Expressjs cũng là một khuôn khổ của Node.js do đó hầu hết các mã code đã được viết sẵn cho các lập trình viên có thể làm việc.

Nhờ có Expressjs mà các nhà lập trình có thể dễ dàng tạo các ứng dụng 1 web, nhiều web hoặc kết hợp. Do có dung lượng khá nhẹ, Expressjs giúp cho việc tổ chức các ứng dụng web thành một kiến trúc MVC có tổ chức hơn. Để có thể sử dụng được mã nguồn này, chúng ta cần phải biết về Javascript và HTML.

Expressjs cũng là một phần của công nghệ giúp quản lý các ứng dụng web một cách dễ dàng hơn hay còn được gọi là ngăn xếp phần mềm MEAN. Nhờ có thư viện Javascript của Express js đã giúp cho các nhà lập trình xây dựng nên các ứng dụng web hiệu quả và nhanh chóng hơn. Expressjs cũng được sử dụng để nâng cao các chức năng của Node.js.

Trên thực tế, nếu không sử dụng Express.js, bạn sẽ phải thực hiện rất nhiều bước lập trình phức tạp để xây dựng nên một API hiệu quả. Express js đã giúp cho việc

lập trình trong Node.js trở nên dễ dàng hơn và có nhiều tính năng mới bổ sung.

3.2.3 MySQL

MySQL là một hệ thống quản trị cơ sở dữ liệu mã nguồn mở (gọi tắt là RDBMS) hoạt động theo mô hình client-server. Với RDBMS là viết tắt của Relational Database Management System. Mỗi cơ sở dữ liệu có thể có nhiều bảng quan hệ chứa dữ liệu. MySQL cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL. MySQL được phát hành từ thập niên 90s.

Ưu điểm của MySQL: Dễ sử dụng: MySQL là cơ sở dữ liệu tốc độ cao, ổn định, dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh; Độ bảo mật cao: MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet khi sở hữu nhiều tính năng bảo mật thậm chí là ở cấp cao; Đa tính năng: MySQL hỗ trợ rất nhiều chức năng SQL được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ cả trực tiếp lẫn gián tiếp ; Khả năng mở rộng và mạnh mẽ: MySQL có thể xử lý rất nhiều dữ liệu và hơn thế nữa nó có thể được mở rộng nếu cần thiết; Nhanh chóng: Việc đưa ra một số tiêu chuẩn cho phép MySQL để làm việc rất hiệu quả và tiết kiệm chi phí, do đó nó làm tăng tốc độ thực thi.

3.2.4 Prisma

Prisma là một Object Relational Mapping (ORMs) được dùng để xây dựng các máy chủ như GraphQL Server, RESTful APIs, microservice, ... Prisma đơn giản là 1 layer nằm giữa Webserver và Database. Prisma giúp chúng ta giao tiếp với db một cách dễ dàng hơn.

Prisma bao gồm ba phần chính:

Prisma Client : Trình tạo truy vấn an toàn và được tạo tự động cho Node.js và TypeScript.

Prisma Migrate : Một hệ thống di chuyển và mô hình hóa dữ liệu khai báo.

Prisma Studio : Một GUI để xem và chỉnh sửa dữ liệu trong cơ sở dữ liệu của bạn.

3.2.5 GraphQL

Trải qua nhiều thập kỉ, REST đã trở thành một tiêu chuẩn cho thiết kế Web API. REST có những ý tưởng rất tuyệt vời như là stateless servers hay là structured access to resources. Tuy nhiên REST APIs vẫn còn chưa linh hoạt để bắt kịp với những yêu cầu thay đổi nhanh chóng phía client. GraphQL được phát triển để đáp ứng nhu cầu linh hoạt hơn và hiệu quả hơn. Nó giải quyết nhiều những thiếu sót và kém hiệu quả mà developer đã trải nghiệm được sau một thời gian làm việc với

REST API.

GraphQL là ngôn ngữ thao tác và truy vấn dữ liệu nguồn mở cho API, cung cấp cho client 1 cách thức dễ dàng để request chính xác những gì họ cần, giúp việc phát triển API dễ dàng hơn theo thời gian. GraphQL được Facebook phát triển nội bộ vào năm 2012 trước khi phát hành công khai vào năm 2015.

GraphQL bao gồm 3 điểm đặc trưng bao gồm: Cho phép client xác định chính xác những dữ liệu gì họ cần; GraphQL làm cho việc tổng hợp dữ liệu từ nhiều nguồn dễ dàng hơn; Sử dụng một type system để khai báo dữ liệu.

Ưu điểm của graphql: GraphQL schema tự động tạo ra nguồn đáng tin cậy trong ứng dụng của GraphQL; Chỉ với một request, client có thể nguồn thông tin họ cần làm giảm số lượng requests; Hỗ trợ tối đa việc kiểm soát và xử lý data type. Từ đó có thể hạn chế sự sai lệch trong giao tiếp giữa server và client; Giúp ứng dụng có thể phát triển thêm API mà không làm ảnh hưởng đến các truy vấn đã có; Không yêu cầu một kiến trúc ứng dụng cụ thể mà vẫn hoạt động như một Rest API; Đồng thời bạn có thể làm việc với các công cụ API hiện có.

Bên cạnh đó graphql vẫn còn tồn tại 1 số khuyết điểm: Nhiều extension mã nguồn mở của GraphQL không tương thích hoạt động được với Rest API; Nhiều truy vấn bị GraphQL chuyển lên server; Và điều này khiến server chịu thêm nhiều công việc và trở nên phức tạp hơn; Công việc để triển khai GraphQL và server sẽ có thể nhiều hơn việc phát triển một Rest API; Bộ nhớ đệm phức tạp hơn rất nhiều lần khi so với Rest. Người viết API sẽ có thể phải đảm nhiệm cả việc bảo trì cho GraphQL schema.

Mặc dù vậy em vẫn quyết định sử dụng graphql vì sự tường minh của các request. Tất cả các tham số và kiểu dữ liệu đều được khai báo 1 cách chặt chẽ khiến cho việc gửi và nhận api dễ dàng hơn. Mặc dù việc thiết lập và cài đặt phức tạp hơn rest API nhưng để phát triển và bảo trì thì em nghĩ là nó xứng đáng với công sức mình bỏ ra.

3.2.6 Các thư viện khác

Apollo Server Apollo Server là một triển khai máy chủ GraphQL cho JavaScript, đặc biệt là cho Node.js.

Bcryptjs Bcryptjs là một dạng thư viện cung cấp hash function thường sử dụng để mã hóa password với mục đích tăng cường bảo mật.

Sendgrid Sendgrid là phần mềm tiếp thị qua Email, Sendgrid cung cấp giải pháp Email dựa trên nền tảng đám mây, thay thế cho các Email truyền thống.

Pusher Thư viện cung cấp khả năng gửi push notification (Thông báo đẩy) trên

trình duyệt.

Twilio Twilio là dịch vụ cho phép xây dựng ứng dụng liên quan đến việc gửi và nhận các tin nhắn và cuộc gọi trên điện thoại.

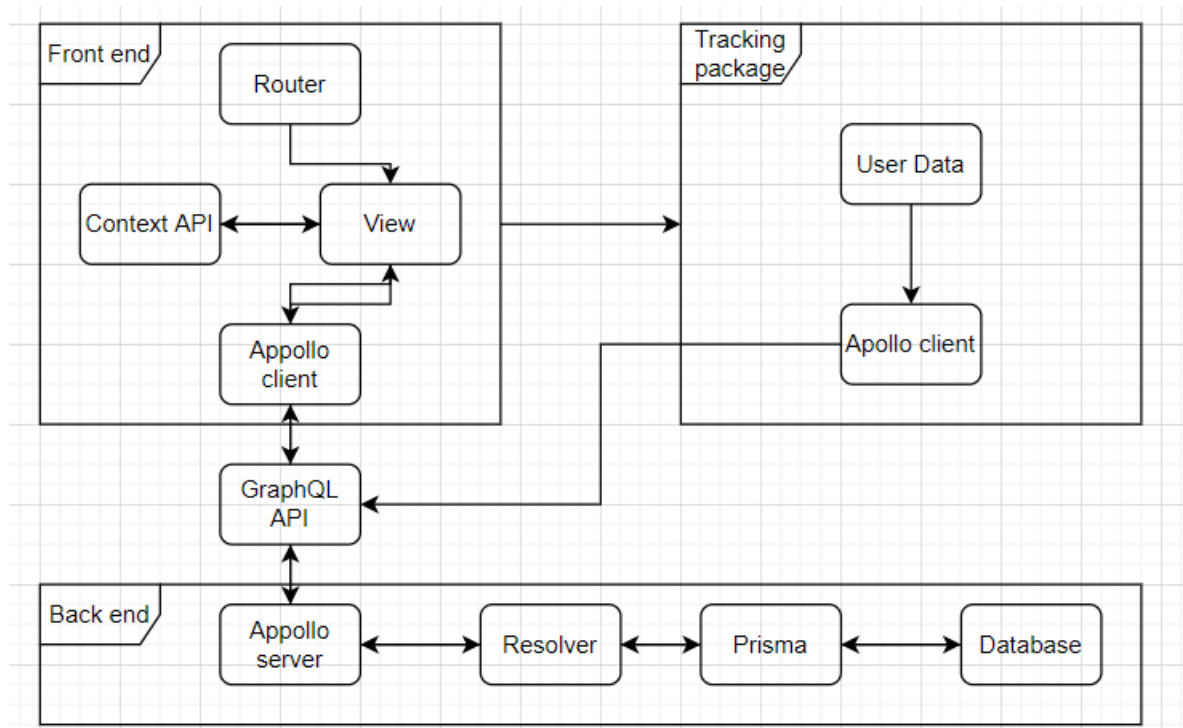
Whatmyuseragent.com whatmyuseragent cung cấp api giúp xác định thông tin về loại trình duyệt, loại điện thoại cũng như hệ điều hành dựa trên dữ liệu từ useragent lưu trên trình duyệt.

ip-api.com ip-api.com cung cấp api miễn phí giúp truy xuất các thông tin về vị trí địa lý của người dùng như tên quốc gia, vùng lãnh thổ,... từ địa chỉ IP của họ.

CHƯƠNG 4. THỰC NGHIỆM VÀ ĐÁNG GIÁ

4.1 Thiết kế kiến trúc

4.1.1 Thiết kế tổng quan

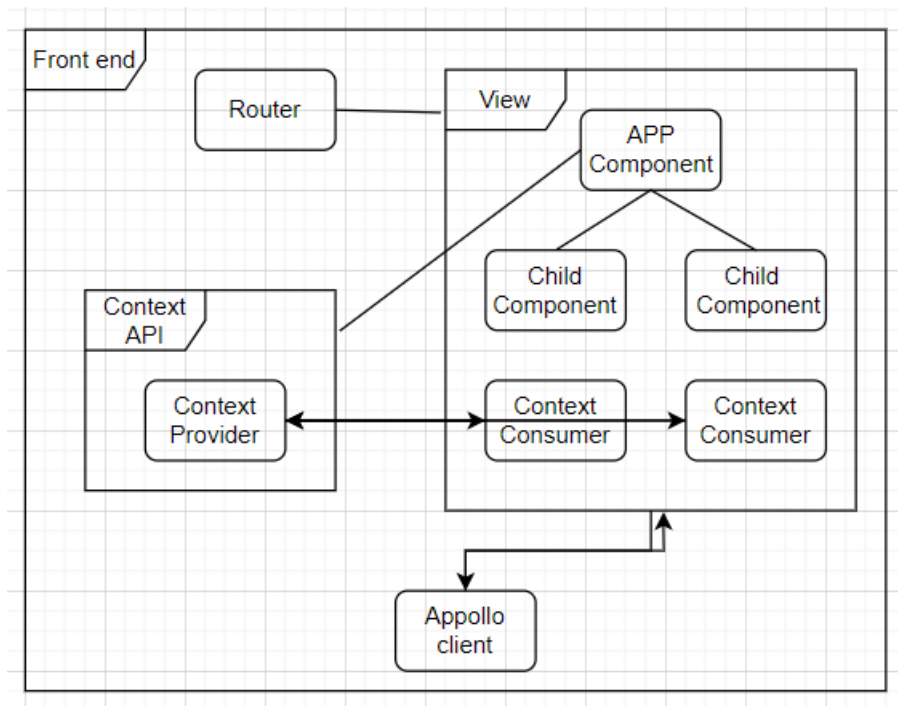


Hình 4.1: Kiến trúc hệ thống

Hình 4.1 mô tả kiến trúc tổng quan của hệ thống. Hệ thống sẽ được chia làm 3 phần riêng biệt: Bao gồm tracking-package, frontend và backend. Các phần này sẽ giao tiếp với nhau qua api trên nền tảng grapQL.

Phía backend sẽ được cài đặt apollo server giúp đảm nhận việc xử lý các request của frontend và tracking-package.

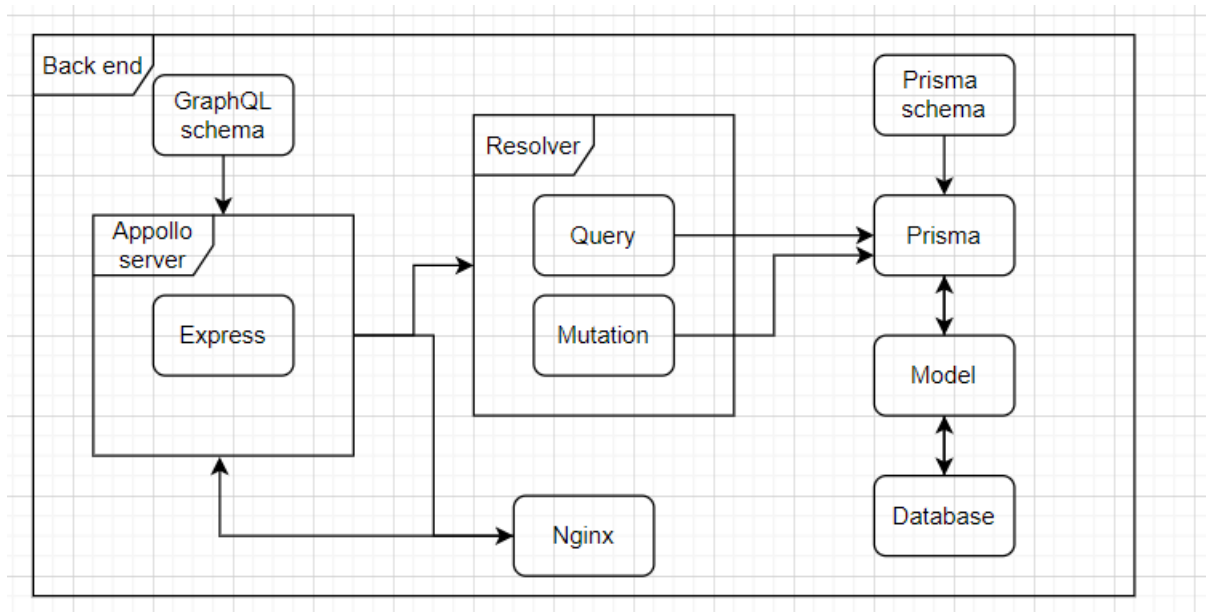
4.1.2 Thiết kế kiến trúc frontend



Hình 4.2: Kiến trúc Frontend

Hình 4.2 mô tả kiến trúc chi tiết của frontend. Khi người dùng request 1 router dữ liệu tại client sẽ được lấy từ context Provider trực tiếp từ component con mà không cần thông qua component cha. Dữ liệu từ phía server sẽ được lấy qua apollo client.

4.1.3 Thiết kế kiến trúc backend

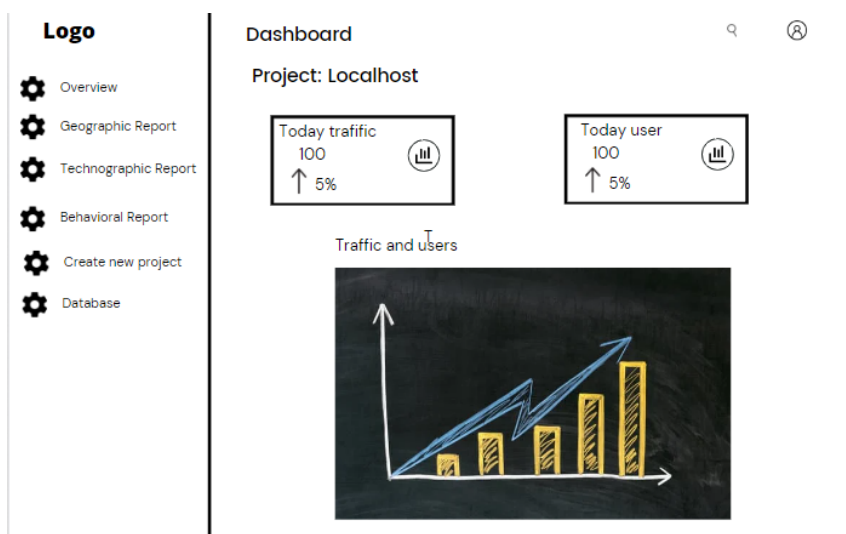


Hình 4.3: Kiến trúc Backend

Hình 4.3 mô tả kiến trúc chi tiết của backend. Đầu tiên prisma sẽ tạo ra CSDL và model tương ứng theo file schema của prisma. Khi web server nhận được request từ client nó sẽ forward cho apolloserver. Apolloserver dựa theo file schema của graphql mà xác định request đó cần xử lý như thế nào và forward cho resolver. Resolver có nhiệm vụ xử lý các request từ phía client, các request này được chia làm 2 loại chính: 1. Query: các request yêu cầu đọc dữ liệu; 2. Mutation: các request yêu cầu ghi dữ liệu. Rồi tiếp theo thông qua prisma mà đọc ghi dữ liệu tương ứng trong cơ sở dữ liệu.

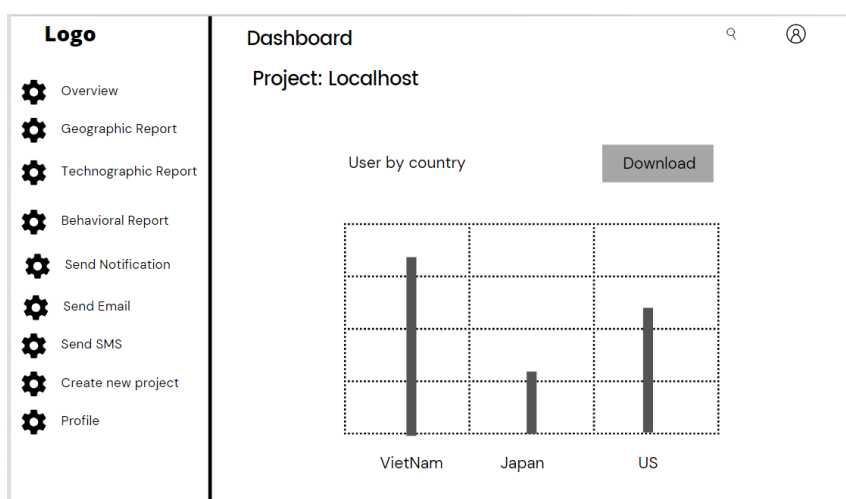
4.2 Thiết kế chi tiết

4.2.1 Thiết kế giao diện



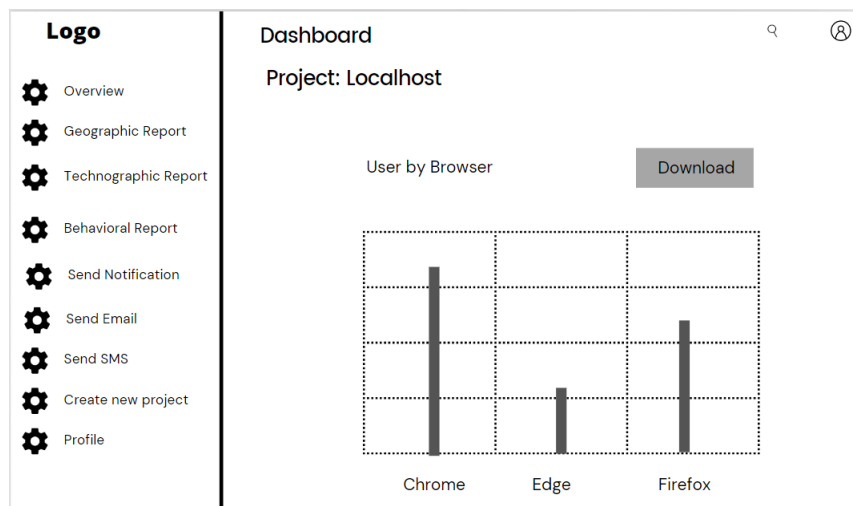
Hình 4.4: Màn hình dashboard overview

Hình 4.4 mô tả màn hình chính của hệ thống hiển thị các thống kê tổng quan về lượt truy cập và người sử dụng



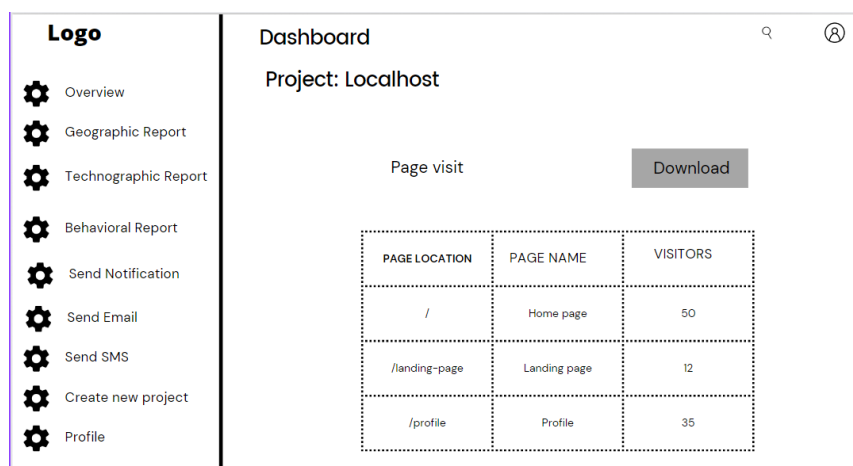
Hình 4.5: Màn hình geographic report

Hình 4.5 mô tả màn hình hiển thị báo vào về dữ liệu địa lý. Bên phải là thanh điều hướng sang các chức năng khác. Ở giữa màn hình là các biểu đồ cột. Phía trên biểu đồ là nút download giúp người dùng download dữ liệu về máy mình dưới dạng file csv.



Hình 4.6: Màn hình technographic report

Hình 4.6 mô tả màn hình hiển thị báo vào về dữ liệu công nghệ. Bên phải là thanh điều hướng sang các chức năng khác. Ở giữa màn hình là các biểu đồ cột. Phía trên biểu đồ là nút download giúp người dùng download dữ liệu về máy mình dưới dạng file csv



Hình 4.7: Màn hình behavioral report

Hình 4.7 mô tả màn hình hiển thị báo vào về dữ liệu hành vi người dùng. Bên phải là thanh điều hướng sang các chức năng khác. Ở giữa màn hình là các biểu đồ cột. Phía trên biểu đồ là nút download giúp người dùng download dữ liệu về máy mình dưới dạng file csv

The screenshot shows a web application interface. On the left is a sidebar with a 'Logo' and a list of menu items: Overview, Geographic Report, Technographic Report, Behavioral Report, Send Notification, Send Email, Send SMS, Create new project, and Profile. The main content area is titled 'Dashboard' and 'Project: Localhost'. It features a 'send notification' section with 'NOTIFICATION INFORMATION' containing 'TITLE' (input: 'Welcome to our website') and 'BODY' (input: 'Let's get started'). Below this is a 'USER SEGMENT' section with 'Countries' (input: 'VietNam') and 'Browsers' (input: 'Chrome'). A 'Send notification' button is at the bottom.

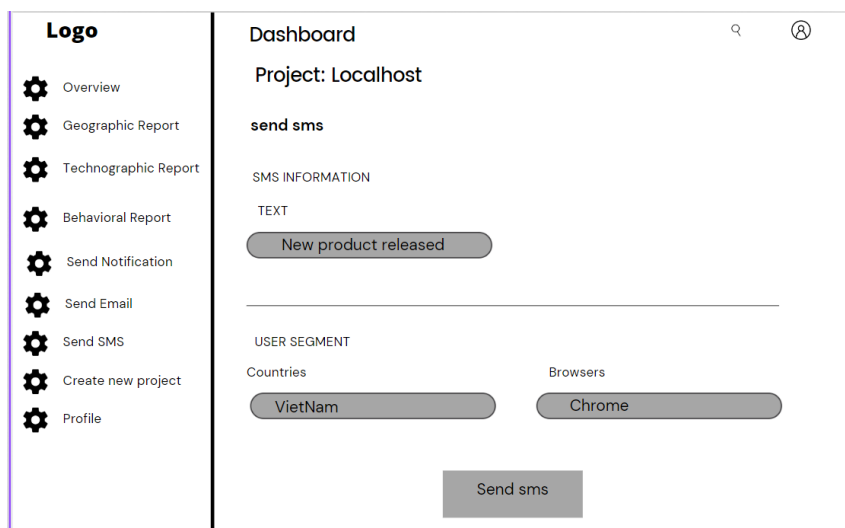
Hình 4.8: Màn hình send email

Hình 4.8 mô tả màn hình hiển thị chức năng gửi email cho người dùng. Bên phải là thanh điều hướng sang các chức năng khác. Ở giữa màn hình là các ô input để nhập nội dung email và lọc các người dùng muốn gửi email.

The screenshot shows a similar web application interface. The sidebar is identical. The main content area is titled 'Dashboard' and 'Project: Localhost'. It features a 'send email' section with 'EMAIL INFORMATION' containing 'SUBJECT' (input: 'Welcome to our website') and 'TEXT' (input: 'Let's get started'). Below this is a 'USER SEGMENT' section with 'Countries' (input: 'VietNam') and 'Browsers' (input: 'Chrome'). A 'Send email' button is at the bottom.

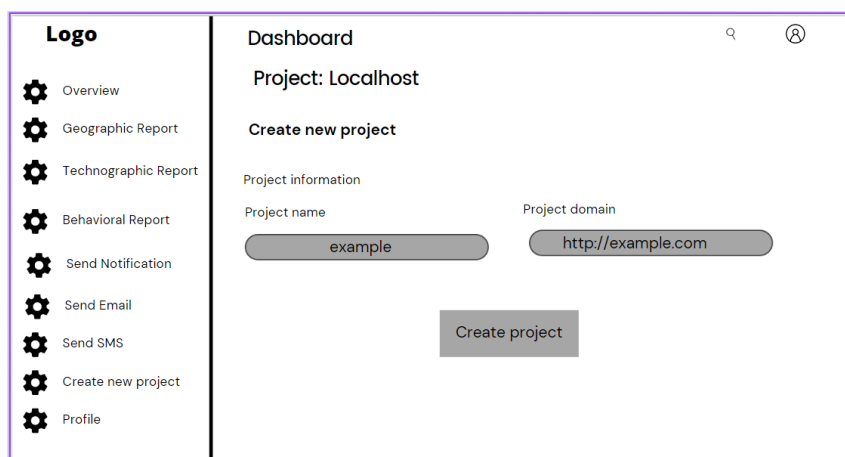
Hình 4.9: Màn hình send notification

Hình 4.9 mô tả màn hình hiển thị chức năng gửi thông báo cho người dùng. Bên phải là thanh điều hướng sang các chức năng khác. Ở giữa màn hình là các ô input để nhập nội dung thông báo và lọc các người dùng muốn gửi thông báo.



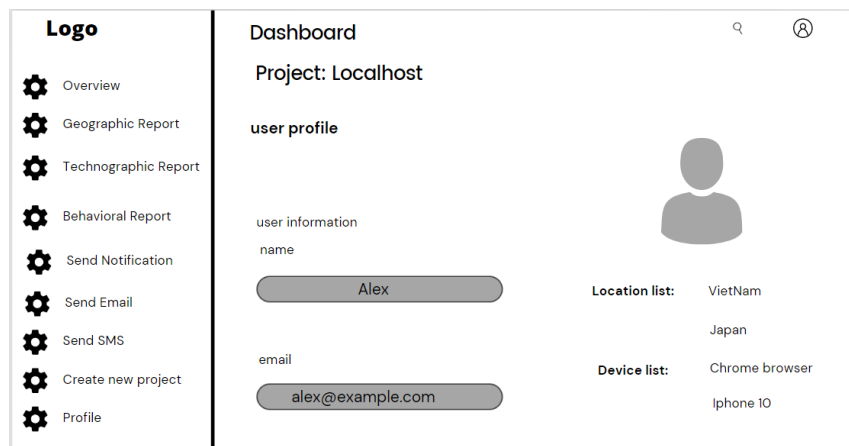
Hình 4.10: Màn hình send sms

Hình 4.10 mô tả màn hình hiển thị chức năng gửi tin nhắn cho người dùng. Bên phải là thanh điều hướng sang các chức năng khác. Ở giữa màn hình là các ô input để nhập nội dung tin nhắn và lọc các người dùng muốn gửi tin nhắn.



Hình 4.11: Màn hình create project

Hình 4.11 mô tả màn hình hiển thị chức năng tạo 1 project mới. Bên phải là thanh điều hướng sang các chức năng khác. Ở giữa màn hình là các ô input để nhập các thuộc tính của project.



Hình 4.12: Màn hình user profile

Hình 4.12 mô tả màn hình chính hiển thị thông tin của từng người sử dụng sau khi đã được thống nhất dữ liệu.

4.2.2 Thiết kế cơ sở dữ liệu

Thiết kế chi tiết cơ sở dữ liệu:

Bảng 4.1: Thiết kế chi tiết bảng Customer

Tên trường :	Kiểu dữ liệu	Ý nghĩa
id	Int	id, khóa chính của bảng
username	String	Tên đăng nhập của khách hàng (doanh nghiệp)
password	String	Mật khẩu của khách hàng (doanh nghiệp)

Bảng 4.2: Thiết kế chi tiết bảng Project

Tên trường :	Kiểu dữ liệu	Ý nghĩa
id	Int	id, khóa chính của bảng
project _{name}	String	Tên Project
origin	String	URL của trang web (bao gồm cả port)
CustomerId	String	Id của khách hàng (doanh nghiệp)

Bảng 4.3: Thiết kế chi tiết bảng User

Tên trường :	Kiểu dữ liệu	Ý nghĩa
id	String	id, khóa chính của bảng
name	String	Tên người dùng

Bảng 4.4: Thiết kế chi tiết bảng UsersOnProjects

Tên trường :	Kiểu dữ liệu	Ý nghĩa
userId	String	user ID của người dùng
projectId	Int	ID của project
email	String	Email của người dùng tại trang web này
first visit	Date	Thời gian lần đầu người dùng truy cập trang web này

Bảng 4.5: Thiết kế chi tiết bảng Event

Tên trường :	Kiểu dữ liệu	Ý nghĩa
id	Int	user ID của người dùng
event date	Date	ID của project
event name	String	Tên event
event value	Json	Giá trị của event
session id	String	id của session
category	String	loại thiết bị
mobile brand name	String	tên thiết bị điện thoại
mobile model name	String	model của thiết bị điện thoại
operating system	String	Hệ điều hành của thiết bị
operating system version	String	Phiên bản hệ điều hành của thiết bị
advertising id	String	advertising id của thiết bị
device language	String	Ngôn ngữ của thiết bị
is limited ad tracking	Boolean	Giá trị cho biết thiết bị có chặn thu thập dữ liệu hay không
time zone	String	Time zone của thiết bị
browser	String	Tên trình duyệt
browser version	String	Version của trình duyệt
ip address	String	Địa chỉ IP
continent	String	Tên lục địa người dùng đang ở
country	String	Tên quốc gia người dùng đang ở
region	String	Tên vùng người dùng đang ở
city	String	Tên thành phố người dùng đang ở
sub continent	String	Tên lục địa người dùng đang ở
referrer	String	Tên trang web đưa người tới trang web hiện tại
UserId	String	Id của người dùng tạo ra event này
ProjectId		Id của project mà event được tạo ra

Bảng 4.6: Thiết kế chi tiết bảng Notification

Tên trường :	Kiểu dữ liệu	Ý nghĩa
id	Int	id của thông báo
title	String	tiêu đề của thông báo
body	String	nội dung của thông báo
ProjectId	Int	Id của project mà thông báo thuộc về

Bảng 4.7: Thiết kế chi tiết bảng UsersReceiveNotification

Tên trường :	Kiểu dữ liệu	Ý nghĩa
userId	String	id của người dùng
notificationId	Int	id của thông báo
isReceived	Boolean	thể hiện thông báo đã nhận được
isOpened	Boolean	thể hiện thông báo đã được mở chưa

Bảng 4.8: Thiết kế chi tiết bảng Email

Tên trường :	Kiểu dữ liệu	Ý nghĩa
id	String	id của email
subject	String	tiêu đề của email
text	String	nội dung của email
filter	Json	thể hiện mail đã nhận được chưa

Bảng 4.9: Thiết kế chi tiết bảng UsersReceiveEmail

Tên trường :	Kiểu dữ liệu	Ý nghĩa
userId	String	id của người dùng
emailId	Int	id của email
isReceived	Boolean	nội dung của email
isOpened	Boolean	bộ lọc các người dùng nhận được mail

4.3 Xây dựng ứng dụng

4.3.1 Thư viện và công cụ sử dụng

Các công cụ, ngôn ngữ lập trình, API, thư viện, IDE, công cụ kiểm thử, v.v. mà em sử dụng để phát triển ứng dụng.

Mục đích	Công cụ	Địa chỉ URL
IDE lập trình	Visual studio code	https://code.visualstudio.com/
Ngôn ngữ lập trình	Javascript	https://www.javascript.com/
Thư việc frontend	React	https://reactjs.org/
Thư viện đảm nhận gửi request cho server	Apollo client	https://www.apollographql.com/
Thư viện vẽ đồ thị	ChartJS	https://www.chartjs.org/
Thư viện backend	NodeJS	https://nodejs.org/en/
Thư viện backend	ExpressJS	https://expressjs.com/
Thư viện đảm nhận xử lý request từ client	Apollo Server	https://www.apollographql.com/
Thư viện giúp gửi thông báo	Pusher	https://pusher.com/
Thư viện giúp gửi email	SendGrid	https://sendgrid.com/
Thư viện giúp gửi SMS	Twilio	https://www.twilio.com/
Hệ quản trị cơ sở dữ liệu	MySQL	https://www.mysql.com/

Bảng 4.10: Danh sách thư viện và công cụ sử dụng

4.3.2 Kết quả đạt được

Kết quả thu được trong đồ án lần này là một hệ thống gồm có 3 thành phần chính: Package giúp thu thập dữ liệu người dùng gửi về server; 1 server viết bằng nodejs thực hiện xử lý các request từ package trên và từ client; 1 ứng dụng reactjs giúp người sử dụng tương tác với hệ thống. Chi tiết về các thành phần được biểu diễn trong bảng sau.

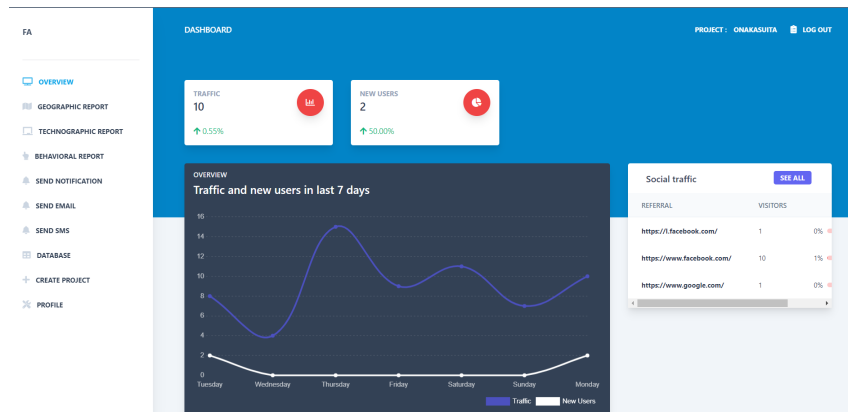
Thông số	Giá trị
Số lượng file mã nguồn ở tracking package	35
Số lượng file mã nguồn ở frontend	80
Số lượng file mã nguồn ở backend	65
Số lượng api	50
Dung lượng mã nguồn ở tracking package	70kb
Dung lượng mã nguồn ở frontend	230kb
Dung lượng mã nguồn ở backend	528kb

Bảng 4.11: Bảng thống kê các thông số của hệ thống

4.3.3 Minh họa các chức năng chính

Chức năng xem các báo cáo

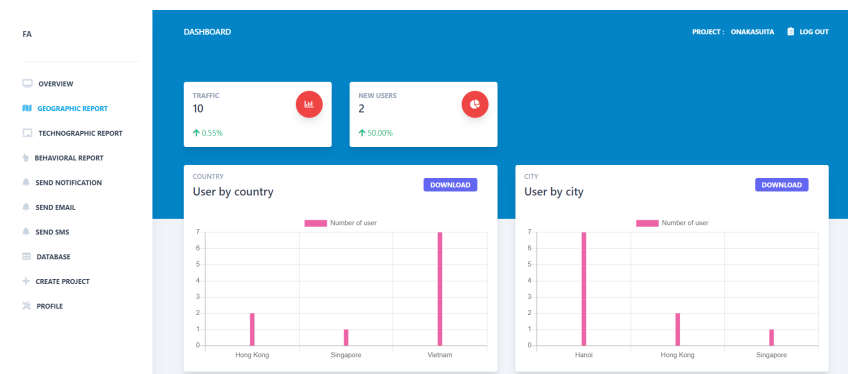
1. Báo cáo tổng quát.



Hình 4.13: Màn hình chức năng xem báo cáo tổng quát

Hình 4.13 Sau khi thực hiện cài đặt tracking package thành công, hệ thống sẽ gửi dữ liệu về server, server sau đó hiển thị dữ liệu thu thập được như hình trên. Trên cùng là dữ liệu về số lượt truy cập và người dùng mới trong ngày. Bên dưới phía tay trái là biểu đồ thể hiện lượt truy cập và người dùng trong 7 ngày gần nhất. Phía bên tay phải là bảng thể hiện các nguồn mà người dùng biết đến trang web

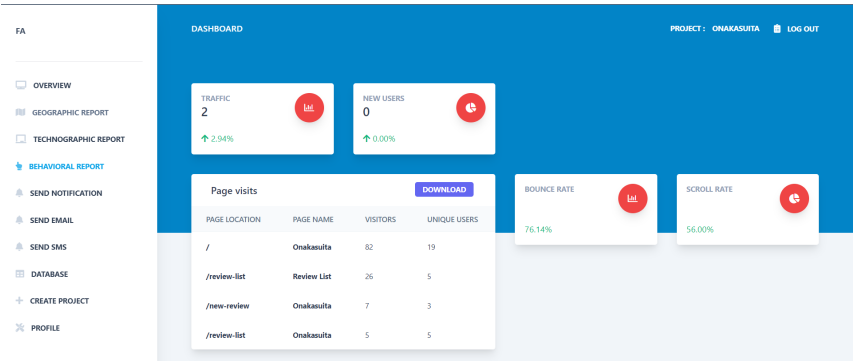
2. Báo cáo về địa lý.



Hình 4.14: Màn hình chức năng xem báo cáo về địa lý

Hình 4.14 Bên dưới phía tay trái là biểu đồ thể hiện số lượng người dùng được phân chia theo từng quốc gia. Phía bên tay phải là biểu đồ thể hiện số lượng người dùng được phân chia theo từng thành phố.

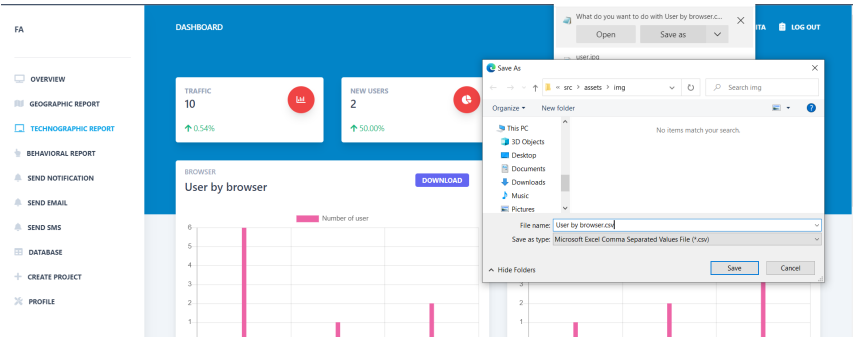
3. Báo cáo về hành vi.



Hình 4.15: Màn hình chức năng xem báo cáo về hành vi

Hình 4.15 Bên dưới phía tay trái là bảng thể hiện số lượng người dùng truy cập với mỗi page của website. Phía bên tay phải thể hiện bounce-rate và scorll-rate của website.

Chức năng tải về các báo cáo



Hình 4.16: Màn hình chức năng tải về báo cáo

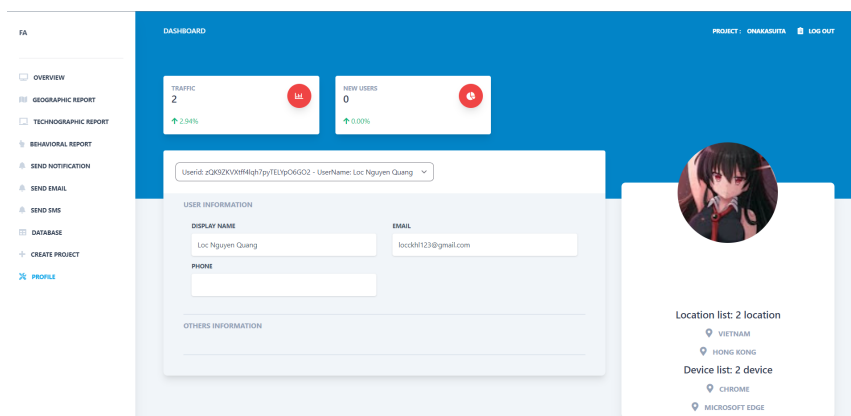
Hình 4.16 mô tả hành động người dùng muốn tải về 1 báo cáo dưới dạng file csv.Ở đây là báo cáo số lượng người dùng truy cập được phân chia theo các loại trình duyệt.

	A	B
1	BROWSER	NUMBER OF USER
2	Chrome	6
3	Chrome Mobile	1
4	Microsoft Edge	2

Hình 4.17: Màn hình báo cáo được tải về

4.17 mô tả file được tải ở trên được mở bởi phần mềm Microsoft Excel.

Chức năng thống nhất dữ liệu người dùng

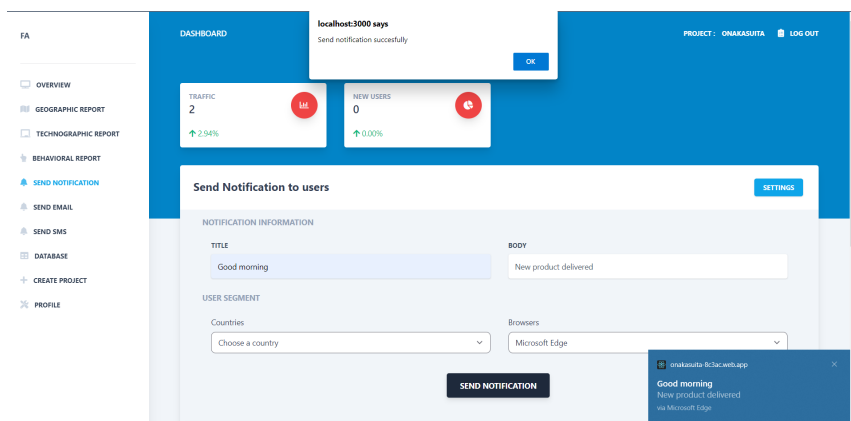


Hình 4.18: Màn hình thể hiện dữ liệu đã được thống nhất của người dùng

Hình 4.18 Sau khi khách hàng tiến hành thống nhất dữ liệu người dùng, dữ liệu của từng người dùng sẽ được hiển thị.

Chức năng gửi thông báo, email, , sms cho người dùng

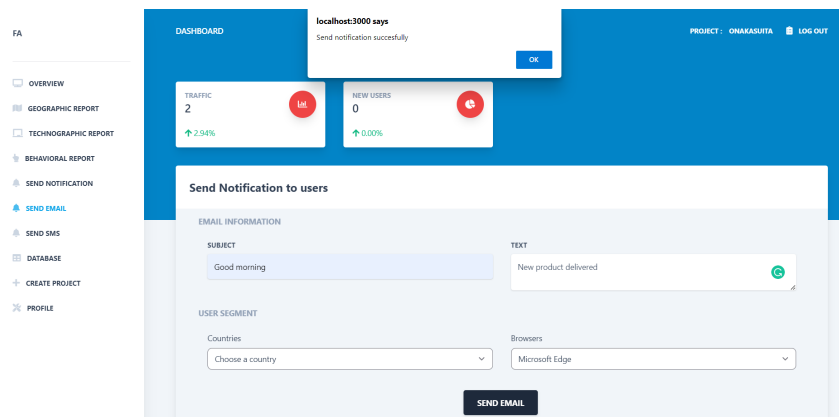
1. Chức năng gửi thông báo



Hình 4.19: Màn hình thể hiện chức năng gửi thông báo

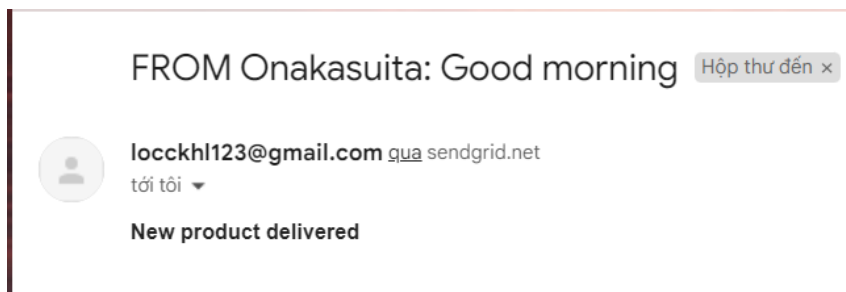
Hình 4.19 Sau khi nhập nội dung thông báo, thông báo sẽ được gửi tới người dùng ngay lập tức.

2. Chức năng gửi email



Hình 4.20: Màn hình thể hiện chức năng gửi email

Hình 4.20 Hệ thống thông báo email đã được gửi thành công.



Hình 4.21: Màn hình hiển thị email đã được gửi

4.4 Kiểm thử**Cài đặt tracking package****Bảng 4.12:** Đặc tả use case Setup tracking package

Mã Test case:	TC01
Tên Test case	Cài đặt tracking package
Các bước thực hiện:	<ol style="list-style-type: none"> 1. Khách hàng tạo tài khoản và đăng nhập vào hệ thống ; 2. Khách hàng tạo project trên hệ thống ; 3. Khách hàng nhúng link hoặc cài đặt package vào website của mình;
Kết quả mong đợi	Dữ liệu được hiển thị đầy đủ trên hệ thống.
Kết quả thực tế	Không có lỗi nào

Thông nhất dữ liệu người dùng**Bảng 4.13:** Đặc tả use case Setup tracking package

Mã Test case:	TC02
Tên Test case	Thông nhất dữ liệu người dùng
Các bước thực hiện:	<ol style="list-style-type: none"> 1. Khách hàng tìm chỗ mà chức năng xác thực người dùng được triển khai ; 2. Khách hàng gửi thông tin của người dùng về hệ thống khi người dùng đăng nhập ;
Kết quả mong đợi	Thông tin của người dùng được thống nhất và hiển thị đầy đủ .
Kết quả thực tế	Không có lỗi nào

Gửi thông báo cho người dùng**Bảng 4.14:** Đặc tả use case Setup tracking package

Mã Test case:	TC03
Tên Test case	Gửi thông báo cho người dùng
Các bước thực hiện:	<ol style="list-style-type: none"> 1. Khách hàng truy cập vào phần gửi thông báo trên hệ thống ; 2. Khách hàng nhập tiêu đề và nội dung của thông báo ; 3. Khách hàng lọc ra những người dùng mình muốn gửi thông báo đến;
Kết quả mong đợi	Thông báo được gửi đến người dùng.
Kết quả thực tế	Không có lỗi nào

4.5 Triển khai

Ứng dụng của em được triển khai trên nền tảng ubuntu cùng với nginx là webserver, chi tiết như sau:

Bảng 4.15: Chi tiết cấu hình hệ thống

Tên cấu hình:	Thông số
VPS service	Microsoft Azure
Hệ điều hành:	Linux (ubuntu 18.04)
Web server	nginx
RAM	8GiB
Public IP address	20.89.51.84
Domain name	freeanalytics.one (Mua ở name cheap)

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Các giải pháp và đóng góp nổi bật

Vấn đề về xác thực người dùng ẩn danh Lâu nay để xác thực người dùng thì phương pháp phổ biến nhất chính là dùng cookie. Cookie là một đoạn văn bản mà một Web server có thể lưu trên ổ cứng của người dùng với các tệp được trang web người dùng truy cập tạo ra. Cookie cho phép website lưu thông tin của người dùng trên máy tính và sau đó lấy lại nó. Các mẫu thông tin sẽ được lưu dưới dạng cặp tên – giá trị (name-value).

Cookie được chia thành 2 loại: first-party cookies: do trang web người dùng truy cập tạo ra, trang web được hiển thị trên thanh địa chỉ. Cookie bên thứ nhất được thu thập bởi Session Cookie và Persistent Cookie. Với first-party cookie, các trang web có thể duy trì trạng thái đăng nhập, ghi nhớ tùy chọn và cung cấp nội dung phù hợp với vị trí và thói quen của người dùng.

third-party cookies: do các trang web khác tạo ra. Trên trang web mà người dùng truy cập chứa các nội dung như quảng cáo hoặc hình ảnh được sở hữu bởi bên thứ ba. Đây là Cookie cho phép tiếp thị và quảng cáo.

Với các tools, hệ thống thu thập dữ liệu hay tiếp thị quảng cáo thì third-party cookies rất quan trọng. Theo như google thì có thể trong cuối năm 2023 họ sẽ không hỗ trợ và loại bỏ đi third-party cookies. Google không phải công ty đầu tiên đưa ra quyết định này, trước đó safari và firefox đã có những biện pháp, quy định chặt chẽ về cookie để tăng cường bảo mật cho người dùng. Trong bối cảnh này thì các nhà tiếp thị cần những giải pháp thay thế khác, và 1 trong số đó là browser fingerprint. Ở đồ án này, em đã sử dụng công nghệ này để có thể xác thực được người dùng ẩn danh.

browser fingerprint là công nghệ dựa theo các cấu hình, cài đặt của trình duyệt mà người dùng truy cập từ đó tạo ra 1 mã hash dùng để làm id giúp xác thực người dùng đó. Các thông số phải kể đến như loại, phiên bản trình duyệt, loại, phiên bản hệ điều hành, ngôn ngữ, độ phân giải màn hình, màu sắc, phông chữ trình duyệt hỗ trợ, Có thể thấy càng dùng thông số để hash thì càng có thể làm tăng độ chính xác của id. Tuy các thông số này nhiều nhưng tỷ lệ để 2 người trên thế giới có thiết lập thông số trình duyệt giống nhau không phải là thấp. Vì thế những thông số thông thường này là chưa đủ.

Để tăng độ chính xác ta phải sử dụng thêm các thông số đặc biệt khác, tiêu biểu như là Canvas Fingerprinting. Canvas Fingerprinting bản thân nó cũng được coi là

1 loại browser fingerprint. Canvas Fingerprinting sử dụng thẻ canvas của html5 để vẽ nên 1 hình bất kỳ trên trình duyệt mà không hiển thị nó cho người dùng. Hình vẽ được tạo ra sẽ không giống nhau hoàn toàn do phụ thuộc kiến trúc của cpu cũng như của gpu.

Ngoài ra còn 1 thông số khác khá hữu dụng, đó là thông tin và các extension được cài đặt trên trình duyệt. Mặc dù cách này khá hay nhưng dò người dùng có thể cài đặt hoặc xóa extension sau 1 khoảng thời gian và cách này cũng không phải sử dụng api chính thống của trình duyệt nên em không áp dụng.

Kết quả là sau khi áp dụng kỹ thuật browser fingerprint em đã thu được id giống nhau kể cả khi người dùng sử dụng chế độ ẩn danh hay tắt cookie.

5.2 Kết luận

So với các hệ thống tracking, analytic, crm, hay cdp,... lớn hiện nay thì đồ án này quả thật vẫn còn nhiều thiếu sót. Nhưng trái lại thì hệ thống của em đơn giản và dễ dàng tích hợp nên khá phù hợp cho các website nhỏ và vừa. Mặc dù không được nhiều tính năng như các hệ thống lớn khác nhưng những tính năng mà hệ thống này có có thể nói là đủ dùng.

Khi làm đồ án, bản thân em đã đạt được những kết quả sau:

- Học được cách thiết kế 1 hệ thống từ đầu đến cuối.
- Học được cách tự tìm kiếm tài liệu, sách tham khảo ở nhiều nguồn uy tín.
- Biết được cách triển khai 1 hệ thống thật trên vps.
- Rèn luyện được kỹ năng viết báo cáo 1 cách đầy đủ và chi tiết.
- Nâng cao kỹ năng quản lý thời gian để giải quyết công .
- Hiểu được thêm nhiều kiến thức và các công nghệ hiện tại.

5.3 Hướng phát triển

Hệ thống của em còn nhiều thiếu sót nên cần phải cải thiện nhiều.

- Trước hết là giao diện và trải nghiệm người dùng cần phải được nâng cao.
- Từ các dữ liệu đã thu thập được cần phải tìm ra được nhiều thông tin có giá trị hơn.
- Cấu trúc và logic của code nhiều chỗ còn chưa hợp lý cần phải được sửa lại.

Tiếp đó là hướng phát triển thêm các tính năng mới:

- Tiến hành thu thập thêm nhiều dữ liệu có ích từ người dùng hơn.
- Phát triển tính năng tự động hóa marketing trên nền tảng các chức năng gửi thông báo, email, sms cho người dùng.
- Cải thiện doc để giúp người dùng cài đặt hệ thống dễ dàng hơn .
- Hướng tới tự xây dựng 1 smtp server để giúp mỗi website đều có một email riêng.

TÀI LIỆU THAM KHẢO

- [1] *React*. [Online]. Available: <https://reactjs.org/> (visited on 07/2022).
- [2] *Tailwindcss*. [Online]. Available: <https://tailwindcss.com/> (visited on 07/2022).
- [3] *Context api*. [Online]. Available: <https://reactjs.org/docs/context.html> (visited on 07/2022).
- [4] *Chartjs*. [Online]. Available: <https://www.chartjs.org/> (visited on 07/2022).
- [5] *Apollo docs home*. [Online]. Available: <https://www.apollographql.com/docs/> (visited on 07/2022).
- [6] *Fontawesome*. [Online]. Available: <https://fontawesome.com/> (visited on 07/2022).
- [7] *Nodejs*. [Online]. Available: <https://nodejs.org/en/> (visited on 07/2022).
- [8] *Expressjs*. [Online]. Available: <https://expressjs.com/> (visited on 07/2022).
- [9] *Mysql*. [Online]. Available: <https://www.mysql.com/> (visited on 07/2022).
- [10] *Prisma*. [Online]. Available: <https://www.prisma.io/> (visited on 07/2022).
- [11] *Sendgrid*. [Online]. Available: <https://sendgrid.com/> (visited on 07/2022).
- [12] *Pusher*. [Online]. Available: <https://pusher.com/> (visited on 07/2022).
- [13] *Twilio*. [Online]. Available: <https://www.twilio.com/> (visited on 07/2022).