

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP LỚN LẬP TRÌNH MẠNG
GỬI EMAIL - EMAIL MARKETING
Giảng viên hướng dẫn: TS. ĐẶNG NHÂN CÁCH

Sinh viên thực hiện	Mã số sinh viên
Huỳnh Công Nam Dương	2251330006
Hà Tuấn Khang	2251330021
Đoàn Bảo Long	2251120427
Nguyễn Hữu Bo	22H1120033
Huỳnh Hùng Thiên Đức	2251120083
Huỳnh Công Đạt	2251120079

TP.HỒ CHÍ MINH, ngày 30/11/2024

LỜI CẢM ƠN

Lời nói đầu tiên cho phép em gửi lời cảm ơn tới các Thầy Cô giáo cùng các cán bộ công tác tại Trường Đại học Giao Thông Vận Tải TP.HCM đã tạo mọi điều kiện giúp đỡ chúng em trong thời gian xây dựng.

Đặc biệt em xin bày tỏ lòng biết ơn sâu sắc tới thầy TS. Đặng Nhân Cách người đã tận tình giúp đỡ, chỉ bảo về nghiệp vụ và trực tiếp hướng dẫn nhóm chúng em trong suốt quá trình hoàn thành bài tập lớn này.

Tuy nhiên do thời gian có hạn và cùng với nhiều nguyên nhân khác, mặc dù chúng em đã nỗ lực hết mình xong đồ án của nhóm, tuy vẫn còn mắc phải những thiếu sót và hạn chế. Em rất mong nhận được sự thông cảm và chỉ bảo của các thầy cô cùng tất cả các bạn.

Chúng em xin trân trọng cảm ơn!

MỤC LỤC

MỞ ĐẦU	3
1. Giới thiệu	3
2. Mục tiêu của đề tài	4
3. Nhiệm vụ của đề tài	4
CHƯƠNG I: SƠ LƯỢC VỀ LẬP TRÌNH MẠNG	5
1. Khái niệm về lập trình mạng	5
2. Mục đích của lập trình mạng	6
3. Ứng dụng lập trình mạng	6
4. Mô hình mạng 7 tầng OSI	6
CHƯƠNG II PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	8
1. Phân tích yêu cầu bài toán	8
2. Thiết kế mô hình hệ thống	8
3. Mục tiêu và mục đích	11
CHƯƠNG III GIỚI THIỆU BACKEND	12
1. Giới thiệu về thư viện Nodemailer :	13
1.1. Tính Năng Nổi Bật	13
1.2 . Chức năng module	14
2. Middleware: verifyToken	15
3. Auth	16
3.1. Đăng ký người dùng	16
3.2. Đăng nhập người dùng	16
4. Campaign	17
4.1. Lấy danh sách chiến dịch hoặc thông tin chiến dịch theo ID	17
4.2. Thêm chiến dịch mới	17
4.3. Cập nhật chiến dịch	18
4.4. Xóa chiến dịch	18
4.5. Bắt đầu chiến dịch	19
5. Contacts	19
5.1. Lấy danh sách liên hệ	19
5.2. Thêm liên hệ mới	20
5.3. Lấy chi tiết liên hệ	20
5.4. Cập nhật liên hệ	20
5.5. Xóa liên hệ	21
6: Groups	21
6.1. Lấy danh sách nhóm	21
6.2. Thêm nhóm mới	22
6.3. Lấy chi tiết nhóm	22
6.4. Cập nhật nhóm	22
6.5. Xóa nhóm	23
6.6. Thêm liên hệ vào nhóm	23
6.7. Lấy danh sách liên hệ của một nhóm	24
7. Mail	24
7.1. Lấy danh sách email	24
7.2. Thêm email mới	25
7.3. Xóa email	25
7.4. Cập nhật email	25
CHƯƠNG IV XÂY DỰNG ỨNG DỤNG	26
Kết luận	30

MỞ ĐẦU

1. Giới thiệu

Ngày nay, khi nói đến phát triển các ứng dụng phần mềm, hầu hết mọi người đều nhắc đến các chương trình có khả năng hoạt động trong môi trường mạng tích hợp, đặc biệt là mạng máy tính. Từ các hệ thống quản lý doanh nghiệp, thương mại điện tử, quảng cáo, cho đến chăm sóc khách hàng đều cần đến các ứng dụng mạng.

Nhu cầu về công nghệ thông tin trong lĩnh vực quảng cáo và tiếp thị ngày càng đa dạng. Với sự mở rộng của các hệ thống truyền thông, cùng với việc ngày càng có nhiều thiết bị kết nối vào Internet, khả năng tiếp cận khách hàng mục tiêu qua email trở nên phổ biến và hiệu quả hơn bao giờ hết. Sự ứng dụng giao thức SMTP đã làm nền tảng cho các hệ thống mạng phát triển vượt bậc, giúp việc truyền tải dữ liệu diễn ra nhanh chóng, an toàn và ổn định. Các vấn đề về an ninh, bảo mật cũng được giao thức này xử lý hiệu quả, trở thành một ưu điểm lớn trong lĩnh vực truyền thông và quảng cáo trực tuyến.

Bên cạnh sự phát triển mạnh mẽ của các nền tảng email hiện nay (như Gmail, Outlook, Yahoo Mail...), nhu cầu gửi email quảng cáo hàng loạt và quản lý hiệu quả chiến dịch marketing qua email đã trở thành một phần không thể thiếu đối với doanh nghiệp. Việc cá nhân hóa nội dung email, tự động hóa chiến dịch, đo lường hiệu quả và đảm bảo an toàn thông tin là những yếu tố quan trọng cần được chú trọng trong hệ thống này.

Xuất phát từ thực tế đó, em chọn thực hiện đề tài: **“Sử dụng giao thức SMTP để xây dựng hệ thống gửi email marketing quảng cáo”**.

2. Mục tiêu của đề tài

- Tìm hiểu giao thức SMTP và các công nghệ liên quan.
- Xây dựng hệ thống gửi email marketing quảng cáo.

3. Nhiệm vụ của đề tài.

- Ngôn ngữ lập trình: Node.js
- Công cụ phát triển: Visual Studio Code.

- Sử dụng giao thức SMTP
- Tích hợp API gửi email và cơ sở dữ liệu.

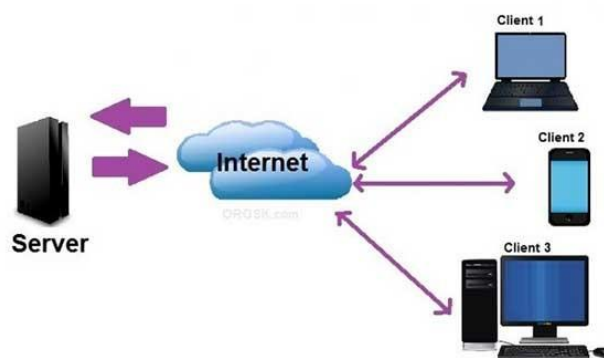
CHƯƠNG I: SƠ LƯỢC VỀ LẬP TRÌNH MẠNG

1. Khái niệm về lập trình mạng

Mạng máy tính được phát triển nhằm mục đích nghiên cứu và phân tích quá trình giao tiếp. Nó hỗ trợ giao tiếp các hệ thống máy tính với nhau để trao đổi thông tin và tài nguyên. Cho nên, để thực hiện việc kết nối mạng bạn cần phải có một mạng riêng của nó.

Khi nói đến phát triển các ứng dụng phần mềm, đa số là người ta muốn nói đến chương trình có khả năng làm việc trong môi trường mạng tích hợp nói chung và mạng máy tính nói riêng. Từ các chương trình kế toán doanh nghiệp, quản lý, trò chơi, điều khiển...

Vấn đề lập trình mạng liên quan đến nhiều lĩnh vực kiến thức khác nhau. Từ kiến thức sử dụng ngôn ngữ lập trình, phân tích thiết kế hệ thống, kiến thức hệ thống mạng, mô hình xây dựng chương trình ứng dụng mạng, kiến thức về cơ sở dữ liệu... cho đến kiến thức truyền thông, các kiến thức các lĩnh vực liên quan khác như mạng điện thoại di động, PSTN, hệ thống GPS, các mạng như Bluetooth, WUSB, mạng sensor... đều là các chương trình ứng dụng mạng.



Hình 1: Mô hình lập trình mạng

2. Mục đích của lập trình mạng

- Giao tiếp dữ liệu: Lập trình mạng giúp các ứng dụng trao đổi thông tin với nhau qua mạng máy tính, từ đó các thiết bị hoặc phần mềm có thể chia sẻ dữ liệu, tệp tin, tin nhắn, v.v.

- Mở rộng khả năng ứng dụng: Thay vì chỉ hoạt động trên một hệ thống cục bộ, các ứng dụng có thể được truy cập và hoạt động trên mạng lưới rộng lớn, bao gồm mạng nội bộ (LAN) và mạng toàn cầu (Internet).

- Phát triển dịch vụ mạng: Cung cấp các dịch vụ trực tuyến như email, web, dịch vụ đám mây, hoặc các ứng dụng có khả năng tương tác qua mạng.

3. Ứng dụng lập trình mạng

- Dịch vụ web: Tạo và phát triển các trang web và ứng dụng web cho phép người dùng giao tiếp qua Internet.

- Ứng dụng chat và hội nghị truyền hình: Phát triển các phần mềm như Skype, Zoom, Messenger, cho phép người dùng gửi tin nhắn, gọi điện hoặc thực hiện video call.

- Trò chơi trực tuyến: Nhiều trò chơi yêu cầu giao tiếp thời gian thực giữa các máy tính qua mạng, chẳng hạn như các trò chơi multiplayer (nhiều người chơi)

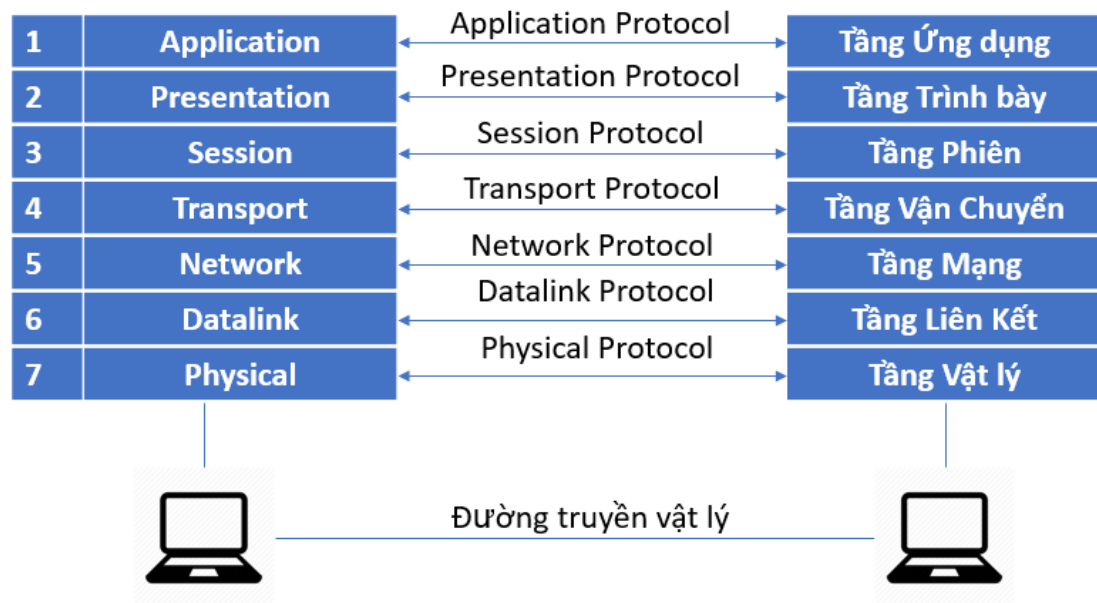
- Hệ thống phân phối tệp tin: Phát triển các ứng dụng cho phép truyền tải dữ liệu lớn qua mạng, ví dụ như FTP server hoặc các dịch vụ đám mây (Google Drive, Dropbox).

- Internet of Things (IoT): Lập trình mạng được sử dụng để kết nối các thiết bị thông minh như đèn, tủ lạnh, máy điều hòa vào mạng để người dùng điều khiển từ xa.

4. Mô hình mạng 7 tầng OSI

Mô hình OSI (Open Systems Interconnection) là mô hình tham chiếu chuẩn do ISO (Tổ chức Tiêu chuẩn Quốc tế) phát triển nhằm chuẩn hóa cách các hệ thống mạng giao tiếp với nhau. Mô hình này chia quy trình giao tiếp thành 7 tầng độc lập, mỗi tầng thực

hiện các chức năng khác nhau. Mô hình này giúp dễ dàng hiểu và khắc phục sự cố trong mạng.



Hình 2: Mô hình OSI

- Tầng 1: Vật lý (Physical Layer):

Đây là tầng thấp nhất, chịu trách nhiệm truyền tải các tín hiệu điện tử, quang học hoặc tín hiệu vô tuyến giữa các thiết bị mạng. Nó quy định các yếu tố như loại cáp, kết nối vật lý, tín hiệu điện áp.

- Tầng 2: Liên kết dữ liệu (Data Link Layer)

Tầng này chịu trách nhiệm điều khiển truy cập vào phương tiện truyền dẫn, phát hiện và sửa lỗi, đảm bảo dữ liệu được truyền chính xác giữa hai thiết bị. Nó được chia thành hai tầng phụ: MAC (Medium Access Control) và LLC (Logical Link Control).

- Tầng 3: Mạng (Network Layer)

Đây là tầng định tuyến, giúp xác định đường đi của gói tin từ nguồn đến đích. Giao thức phổ biến ở tầng này là IP (Internet Protocol).

- Tầng 4: Giao vận (Transport Layer)

Chịu trách nhiệm đảm bảo dữ liệu được truyền đến đúng ứng dụng ở máy nhận. Giao thức phổ biến là TCP (Transmission Control Protocol) và UDP (User Datagram Protocol).

- Tầng 5: Phiên (Session Layer)

Quản lý các phiên kết nối giữa hai thiết bị, bao gồm việc thiết lập, duy trì và kết thúc phiên giao tiếp.

- Tầng 6: Trình bày (Presentation Layer)

Chịu trách nhiệm dịch dữ liệu giữa định dạng được mạng sử dụng và định dạng mà ứng dụng có thể hiểu. Nó bao gồm việc mã hóa, giải mã, nén và giải nén dữ liệu.

- Tầng 7: Ứng dụng (Application Layer)

Đây là tầng gần người dùng nhất, cung cấp giao diện trực tiếp giữa người dùng và mạng. Các giao thức ở tầng này bao gồm HTTP, FTP, SMTP, DNS, v.v.

CHƯƠNG II PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

1. Phân tích yêu cầu bài toán

Yêu cầu bài toán: Sử dụng giao thức SMTP để xây dựng hệ thống gửi email marketing quảng cáo”

2. Thiết kế mô hình hệ thống

- **Quản Lý Người Dùng:**

- **Đăng ký:** Cho phép người dùng tạo tài khoản mới bằng cách cung cấp thông tin như tên người dùng, email và mật khẩu.
- **Đăng nhập:** Người dùng có thể đăng nhập vào tài khoản của mình để truy cập các chức năng của ứng dụng.
- **Đăng xuất:** Cho phép người dùng đăng xuất khỏi tài khoản để bảo vệ thông tin cá nhân.

- **Quản Lý Chiến Dịch Email:**
 - **Tạo Chiến Dịch:** Người dùng có thể tạo chiến dịch email mới bằng cách cung cấp thông tin như tên chiến dịch, mẫu email và danh sách người nhận.
 - **Chỉnh sửa Chiến Dịch:** Cho phép người dùng cập nhật thông tin của các chiến dịch đã tạo.
 - **Xóa Chiến Dịch:** Người dùng có thể xóa các chiến dịch không còn cần thiết.
 - **Bắt đầu Chiến Dịch:** Khởi động chiến dịch để gửi email đến danh sách người nhận đã chọn.
- **Quản Lý Mẫu Email:**
 - **Tạo và Chỉnh sửa Mẫu Email:** Người dùng có thể tạo và chỉnh sửa các mẫu email để sử dụng trong các chiến dịch.
 - **Danh Sách Mẫu Email:** Hiện thị danh sách các mẫu email đã tạo, cho phép người dùng chọn mẫu để sử dụng trong chiến dịch.
- **Quản Lý Danh Bạ:**
 - **Thêm Danh Bạ:** Người dùng có thể thêm liên hệ mới vào danh sách danh bạ.
 - **Chỉnh sửa Danh Bạ:** Cập nhật thông tin liên hệ như tên, email và số điện thoại.
 - **Xóa Danh Bạ:** Xóa các liên hệ không còn cần thiết.
- **Quản Lý Nhóm Liên Hệ:**
 - **Tạo Nhóm:** Người dùng có thể tạo các nhóm liên hệ để tổ chức danh bạ.
 - **Thêm và Xóa Liên Hệ trong Nhóm:** Quản lý các liên hệ trong từng nhóm để dễ dàng gửi email đến nhóm cụ thể.
- **Thống Kê Chiến Dịch:**
 - **Theo Dõi Hiệu Suất:** Cung cấp thông tin thống kê về các chiến dịch đã gửi, bao gồm số lượng email đã gửi, tỷ lệ mở email, và tỷ lệ nhấp chuột.
 - **Báo cáo Chiến Dịch:** Hiện thị báo cáo chi tiết về hiệu suất của từng chiến dịch để người dùng có thể phân tích và cải thiện chiến dịch trong tương lai.
- **Bảo Mật và Quản Lý Phiên:**
 - **Xác Thực Người Dùng:** Sử dụng token JWT để xác thực người dùng và bảo vệ các route quan trọng.

- **Quản Lý Cookie:** Lưu trữ thông tin phiên làm việc của người dùng để duy trì trạng thái đăng nhập.
- **Giao Diện Người Dùng Thân Thiện:**
 - **Thiết Kế Responsive:** Giao diện được tối ưu hóa cho cả máy tính và thiết bị di động, giúp người dùng dễ dàng truy cập từ bất kỳ đâu.
 - **Thành Phần Giao Diện MUI:** Sử dụng Material-UI để tạo ra các thành phần giao diện đẹp mắt và dễ sử dụng.

-Khái quát về Tiny Cloud :

Tiny Cloud là một dịch vụ cung cấp nền tảng cho việc phát triển và triển khai các ứng dụng web, đặc biệt là trong lĩnh vực chỉnh sửa văn bản và nội dung. Nó thường được sử dụng kết hợp với TinyMCE, một trình soạn thảo WYSIWYG (What You See Is What You Get) phổ biến, cho phép người dùng tạo và chỉnh sửa nội dung một cách trực quan.

Dưới đây là một số điểm nổi bật của Tiny Cloud:

- **Dễ Dàng Sử Dụng:** Tiny Cloud cung cấp một API đơn giản và dễ sử dụng, giúp các nhà phát triển tích hợp trình soạn thảo vào ứng dụng của họ một cách nhanh chóng.
- **Lưu Trữ Dữ Liệu:** Dịch vụ này cho phép lưu trữ và quản lý tài liệu trên đám mây, giúp người dùng dễ dàng truy cập và chia sẻ nội dung của họ từ bất kỳ đâu.
- **Tùy Chỉnh Linh Hoạt:** Tiny Cloud hỗ trợ nhiều tùy chọn tùy chỉnh cho trình soạn thảo, cho phép người dùng điều chỉnh giao diện và tính năng theo nhu cầu của họ.
- **Bảo Mật:** Dịch vụ này cung cấp các biện pháp bảo mật để bảo vệ dữ liệu của người dùng, bao gồm mã hóa và xác thực.
- **Hỗ Trợ Nhiều Nền Tảng:** Tiny Cloud có thể được tích hợp vào nhiều loại ứng dụng và nền tảng khác nhau, từ trang web đến ứng dụng di động.

- **Cộng Đồng và Tài Nguyên:** Tiny Cloud có một cộng đồng lớn và nhiều tài nguyên hỗ trợ, bao gồm tài liệu hướng dẫn, diễn đàn và ví dụ mã.

Tiny Cloud là một giải pháp lý tưởng cho các doanh nghiệp và nhà phát triển muốn tích hợp chức năng chỉnh sửa văn bản mạnh mẽ vào ứng dụng của họ mà không cần phải xây dựng mọi thứ từ đầu.

3. Mục tiêu và mục đích

Mục tiêu và mục đích của người gửi Gmail có thể được hiểu ở nhiều góc độ khác nhau, phụ thuộc vào mục đích sử dụng của người gửi. Dưới đây là một số mục tiêu và phổ biến mục tiêu khi sử dụng Gmail để gửi email:

a. Giao thức và hiệu quả nhanh

- Mục tiêu: Cung cấp một phương thức giao tiếp nhanh chóng giữa cá nhân, nhóm hoặc doanh nghiệp.

- Mục đích : Giúp người dùng trao đổi thông tin, giải quyết công việc hoặc trao đổi ý tưởng mà không cần thiết bị trực tiếp.

b. Chia sẻ thông tin và tài liệu

- Mục tiêu : Truyền đạt thông tin, tài liệu, file đính kèm một cách thuận tiện.

- Mục đích : Người gửi có thể chia sẻ các văn bản, hình ảnh, video, tài liệu quan trọng, báo cáo, hợp đồng,... cho người nhận.

c. Xác định và yêu cầu thông tin

- Mục tiêu : Gửi email để xác nhận thông tin hoặc yêu cầu người nhận cung cấp dữ liệu, báo cáo hoặc công việc gì đó.

- Mục đích : Cung cấp xác thực về một giao dịch, yêu cầu cung cấp thêm thông tin hoặc nhận phản hồi từ người khác.

d. Xây dựng và duy trì mối quan hệ

- Mục tiêu : Giữ liên lạc với bạn bè, gia đình, đối tác, hoặc đồng nghiệp.

- Mục đích : Tạo và duy trì mối quan hệ cá nhân hoặc chuyên nghiệp, đặc biệt khi không thể gặp mặt trực tiếp.

e. Thực hiện các giao dịch hoặc thủ tục chính

- Mục tiêu : Gửi thông tin hoặc yêu cầu liên kết đến các tài khoản giao dịch chính, thủ tục hành chính.

- Mục đích : Gửi email xác nhận thanh toán, phụ hồ sơ xin việc, làm thủ tục hành chính hoặc liên hệ với cơ quan chức năng.

f. Giải trí và thư giãn

- Mục tiêu : Gửi các email mang tính chất thư giãn như chia sẻ hình ảnh, video hài hước, hoặc các bài viết thú vị.

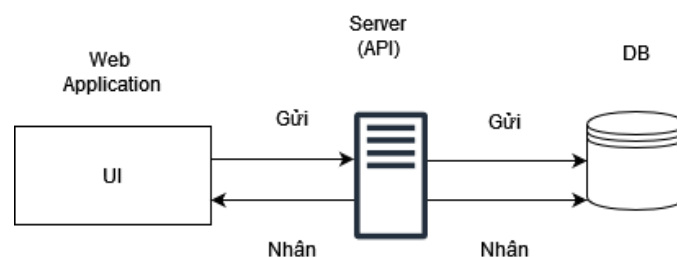
- Mục đích : Duy trì sự gắn kết xã hội qua việc chia sẻ những điều thú vị hoặc giải trí tính năng.

g. Thư thông báo công việc hoặc nhắc nhở

- Mục tiêu : nhắc nhở, báo cáo tiến trình công việc, hoặc đưa ra các thông báo quan trọng liên quan đến công việc.

- Mục đích : Đảm bảo các công việc được thực hiện đúng hạn, nhắc nhở về các cuộc thi hoặc nhiệm vụ cần hoàn thành.

CHƯƠNG III GIỚI THIỆU BACKEND



Hình 3 :Mô hình ứng dụng

Web Application: Giao diện người dùng, cho phép dễ dàng thao tác với API
Server(API): xử lý yêu cầu và trả về dữ liệu.

1.Giới thiệu về thư viện Nodemailer :

Nodemailer là một thư viện phổ biến trong Node.js được sử dụng để gửi email một cách dễ dàng và hiệu quả. Nó hỗ trợ nhiều giao thức email khác nhau và cho phép các nhà phát triển gửi email từ ứng dụng của họ mà không cần phải cấu hình phức tạp. Dưới đây là một số điểm nổi bật về Nodemailer:

1.1.Tính Năng Nổi Bật

- **Gửi Email Đơn Giản:** Nodemailer cung cấp một API đơn giản giúp việc gửi email trở nên dễ dàng. Bạn chỉ cần tạo một đối tượng transporter và sử dụng nó để gửi email.
- **Hỗ Trợ Nhiều Giao Thức:** Nodemailer hỗ trợ nhiều giao thức gửi email như SMTP, Sendmail, và Amazon SES. Điều này giúp bạn linh hoạt trong việc chọn phương thức gửi email phù hợp với nhu cầu của mình.
- **Tùy Chỉnh Nội Dung Email:** Bạn có thể tùy chỉnh nội dung email với HTML, văn bản thuần túy, tệp đính kèm, và nhiều tùy chọn khác. Nodemailer cho phép bạn gửi email với định dạng phong phú và chuyên nghiệp.
- **Hỗ Trợ Đính Kèm Tệp:** Thư viện này cho phép bạn dễ dàng đính kèm tệp vào email, bao gồm cả hình ảnh, tài liệu, và nhiều định dạng khác.
- **Bảo Mật:** Nodemailer hỗ trợ các phương thức xác thực mạnh mẽ, bao gồm OAuth2, giúp bảo vệ thông tin đăng nhập của bạn khi gửi email.
- **Hỗ Trợ Mã Hóa:** Bạn có thể cấu hình Nodemailer để sử dụng SSL/TLS, đảm bảo rằng thông tin email được gửi đi một cách an toàn.

- **Tích Hợp Dễ Dàng:** Nodemailer có thể dễ dàng tích hợp vào các ứng dụng Node.js, cho phép bạn gửi email từ bất kỳ đâu trong mã của bạn.

1.2 . Chức năng module

- **Nodemailer:**
 - **Chức năng:** Gửi email từ ứng dụng Node.js.
 - **Tại sao cần:** Dùng để gửi thông báo hoặc email marketing đến người dùng một cách tự động, giúp cải thiện khả năng giao tiếp với khách hàng.
- **Axios:**
 - **Chức năng:** Gửi các yêu cầu HTTP từ phía client (frontend) đến server (backend).
 - **Tại sao cần:** Để thực hiện các cuộc gọi API, nhận và gửi dữ liệu giữa frontend và backend một cách dễ dàng và hiệu quả.
- **MUI (Material-UI):**
 - **Chức năng:** Thư viện giao diện người dùng cho React, cung cấp các thành phần UI như Button, TextField, Table, v.v.
 - **Tại sao cần:** Để xây dựng giao diện người dùng đẹp mắt, responsive và dễ sử dụng, giúp cải thiện trải nghiệm người dùng.
- **jsonwebtoken:**
 - **Chức năng:** Tạo và xác thực token JWT (JSON Web Token).
 - **Tại sao cần:** Để bảo vệ các route trong ứng dụng bằng cách xác thực người dùng, đảm bảo chỉ những người dùng đã đăng nhập mới có thể truy cập vào các chức năng nhất định.
- **bcrypt:**
 - **Chức năng:** Mã hóa và kiểm tra mật khẩu.
 - **Tại sao cần:** Để bảo mật thông tin người dùng bằng cách mã hóa mật khẩu trước khi lưu vào cơ sở dữ liệu, giảm thiểu rủi ro bị lộ thông tin.
- **dotenv:**

- **Chức năng:** Quản lý các biến môi trường.
- **Tại sao cần:** Để lưu trữ các thông tin nhạy cảm như mật khẩu cơ sở dữ liệu, khóa bí mật, v.v., giúp dễ dàng cấu hình và bảo mật ứng dụng.
- **cors:**
 - **Chức năng:** Cấu hình Cross-Origin Resource Sharing (CORS).
 - **Tại sao cần:** Để cho phép frontend (client) truy cập vào backend (server) từ các nguồn khác nhau, giúp giải quyết vấn đề bảo mật khi gửi yêu cầu giữa các miền khác nhau.
- **mysql2:**
 - **Chức năng:** Kết nối và tương tác với cơ sở dữ liệu MySQL.
 - **Tại sao cần:** Để lưu trữ và truy xuất dữ liệu trong ứng dụng, hỗ trợ các thao tác như thêm, sửa, xóa và lấy dữ liệu từ cơ sở dữ liệu.
- **react-router-dom:**
 - **Chức năng:** Quản lý điều hướng trong ứng dụng React.
 - **Tại sao cần:** Để tạo ra các route cho các trang khác nhau trong ứng dụng, giúp người dùng dễ dàng điều hướng giữa các phần của ứng dụng.
- **js-cookie:**
 - **Chức năng:** Quản lý cookie trong trình duyệt.
 - **Tại sao cần:** Để lưu trữ thông tin phiên làm việc của người dùng, chẳng hạn như token đăng nhập, giúp duy trì trạng thái đăng nhập giữa các lần truy cập.

2. Middleware: verifyToken

Kiểm tra tính hợp lệ của JWT token trong request.

- Middleware lấy token từ header Authorization theo định dạng Bearer <token>.
- Nếu không có token, trả về lỗi 401 Unauthorized với thông báo "Token không được cung cấp".

- Dùng `jwt.verify()` để xác minh token với khóa bí mật ('hello_world').
- Nếu token không hợp lệ hoặc đã hết hạn, trả về lỗi 401 Unauthorized với thông báo "Token không hợp lệ hoặc đã hết hạn".
- Nếu token hợp lệ, thông tin người dùng sẽ được lưu vào `req.user`, và tiếp tục xử lý request.

Response:

- 200 OK: Token hợp lệ và thông tin người dùng được lưu vào `req.user`.
- 401 Unauthorized: Token không hợp lệ hoặc không được cung cấp.

3. Auth

3.1. Đăng ký người dùng

Method: POST

URL: /auth/register

Body:

- `name (required)`: Tên người dùng.
- `email (required)`: Email người dùng.
- `password (required)`: Mật khẩu.

Response:

- 201 Created: Đăng ký thành công.
- 400 Bad Request: Thiếu thông tin bắt buộc.
- 500 Internal Server Error: Lỗi hệ thống.

3.2. Đăng nhập người dùng

Method: POST

URL: /auth/login

Body:

- email (*required*): Email người dùng.
- password (*required*): Mật khẩu.

Response:

- 200 OK: Đăng nhập thành công, trả về token và thông tin người dùng.
- 400 Bad Request: Email không tồn tại hoặc mật khẩu sai.
- 500 Internal Server Error: Lỗi hệ thống.

4. Campaign

4.1. Lấy danh sách chiến dịch hoặc thông tin chiến dịch theo ID

Method: GET

URL: /campaigns/get

Middleware: verifyToken

Query Params:

- campaign_id (*optional*): ID chiến dịch cần lấy.

Response:

- 200 OK: Danh sách hoặc thông tin chiến dịch.
- 404 Not Found: Chiến dịch không tồn tại.
- 500 Internal Server Error: Lỗi hệ thống.

4.2. Thêm chiến dịch mới

Method: POST

URL: /campaigns/add

Middleware: verifyToken

Body:

- name (*required*): Tên chiến dịch.
- email_id (*optional*): ID email.

- `send_at` (*optional*): Thời gian gửi.
- `status` (*optional*): Trạng thái chiến dịch.

Response:

- 201 Created: Thêm thành công, trả về `campaign_id`.
- 500 Internal Server Error: Lỗi hệ thống.

4.3. Cập nhật chiến dịch

Method: PUT

URL: /campaigns/update

Middleware: verifyToken

Body:

- `campaign_id` (*required*): ID chiến dịch.
- `name`, `email_id` (*required*): Tên và email ID.
- `groups_ids` (*required*): Mảng các group ID.

Response:

- 200 OK: Cập nhật thành công.
- 404 Not Found: Chiến dịch không tồn tại.
- 500 Internal Server Error: Lỗi hệ thống.

4.4. Xóa chiến dịch

Method: DELETE

URL: /campaigns/delete

Middleware: verifyToken

Body:

- `campaign_id` (*required*): ID chiến dịch.

Response:

- 200 OK: Xóa thành công.
- 404 Not Found: Chiến dịch không tồn tại.
- 500 Internal Server Error: Lỗi hệ thống.

4.5. Bắt đầu chiến dịch

Method: POST

URL: /campaigns/start

Middleware: verifyToken

Body:

- campaign_id (*required*): ID chiến dịch cần bắt đầu.

Response:

- 200 OK: Chiến dịch đã bắt đầu.
- 404 Not Found: Không có người nhận hợp lệ hoặc email template không tồn tại.
- 500 Internal Server Error: Lỗi hệ thống.

5. Contacts

5.1. Lấy danh sách liên hệ

Method: GET

URL: /contacts

Middleware: verifyToken

Response:

- 200 OK: Trả về danh sách tất cả liên hệ.
- 500 Internal Server Error: Lỗi hệ thống.

5.2. Thêm liên hệ mới

Method: POST

URL: /contacts

Middleware: verifyToken

Body:

- email (*required*): Email của liên hệ.
- name (*required*): Tên của liên hệ.
- phone (*optional*): Số điện thoại của liên hệ.

Response:

- 201 Created: Thêm liên hệ thành công.
- 400 Bad Request: Thiếu thông tin bắt buộc.
- 500 Internal Server Error: Lỗi hệ thống.

5.3. Lấy chi tiết liên hệ

Method: GET

URL: /contacts/:id

Middleware: verifyToken

Response:

- 200 OK: Trả về chi tiết liên hệ.
- 404 Not Found: Không tìm thấy liên hệ.
- 500 Internal Server Error: Lỗi hệ thống.

5.4. Cập nhật liên hệ

Method: PUT

URL: /contacts/:id

Middleware: verifyToken

Body:

- email (*optional*): Email của liên hệ.
- name (*optional*): Tên của liên hệ.
- phone (*optional*): Số điện thoại của liên hệ.

Response:

- 200 OK: Cập nhật liên hệ thành công.
- 404 Not Found: Không tìm thấy liên hệ để cập nhật.
- 500 Internal Server Error: Lỗi hệ thống.

5.5. Xóa liên hệ

Method: DELETE

URL: /contacts/:id

Middleware: verifyToken

Response:

- 200 OK: Xóa liên hệ thành công.
- 404 Not Found: Không tìm thấy liên hệ để xóa.
- 500 Internal Server Error: Lỗi hệ thống.

6: Groups

6.1. Lấy danh sách nhóm

Method: GET

URL: /groups

Middleware: verifyToken

Response:

- 200 OK: Trả về danh sách tất cả nhóm.
- 500 Internal Server Error: Lỗi hệ thống.

6.2. Thêm nhóm mới

Method: POST

URL: /groups

Middleware: verifyToken

Body:

- name (*required*): Tên nhóm.

Response:

- 201 Created: Thêm nhóm thành công.
- 400 Bad Request: Thiếu tên nhóm.
- 500 Internal Server Error: Lỗi hệ thống.

6.3. Lấy chi tiết nhóm

Method: GET

URL: /groups/:id

Middleware: verifyToken

Response:

- 200 OK: Trả về chi tiết nhóm.
- 404 Not Found: Không tìm thấy nhóm.
- 500 Internal Server Error: Lỗi hệ thống.

6.4. Cập nhật nhóm

Method: PUT

URL: /groups/:id

Middleware: verifyToken

Body:

- name (*required*): Tên nhóm.

Response:

- 200 OK: Cập nhật nhóm thành công.
- 404 Not Found: Không tìm thấy nhóm để cập nhật.
- 500 Internal Server Error: Lỗi hệ thống.

6.5. Xóa nhóm

Method: DELETE

URL: /groups/:id

Middleware: verifyToken

Response:

- 200 OK: Xóa nhóm thành công.
- 404 Not Found: Không tìm thấy nhóm để xóa.
- 500 Internal Server Error: Lỗi hệ thống.

6.6. Thêm liên hệ vào nhóm

Method: POST

URL: /groups/:id/contacts

Middleware: verifyToken

Body:

- contactId (*required*): ID của liên hệ.

Response:

- 201 Created: Thêm liên hệ vào nhóm thành công.

- 400 Bad Request: Thiếu thông tin bắt buộc.
- 200 OK: Liên hệ đã tồn tại trong nhóm.
- 500 Internal Server Error: Lỗi hệ thống.

6.7. Lấy danh sách liên hệ của một nhóm

Method: GET

URL: /groups/:id/contacts

Middleware: verifyToken

Response:

- 200 OK: Trả về danh sách liên hệ của nhóm.
- 500 Internal Server Error: Lỗi hệ thống.

7. Mail

7.1. Lấy danh sách email

Method: GET

URL: /mails/get

Middleware: verifyToken

Query Parameters:

- email_id (*optional*): ID của email cần lấy. Nếu không có, trả về tất cả email.

Response:

- 200 OK: Trả về danh sách email hoặc email theo ID.
- 404 Not Found: Không tìm thấy email.
- 500 Internal Server Error: Lỗi hệ thống.

7.2. Thêm email mới

Method: POST

URL: /mails/add

Middleware: verifyToken

Body:

- name (*required*): Tên người gửi.
- subject (*required*): Chủ đề email.
- content (*required*): Nội dung email.

Response:

- 201 Created: Thêm email thành công.
- 500 Internal Server Error: Lỗi hệ thống.

7.3. Xóa email

Method: DELETE

URL: /mails/delete/:id

Middleware: verifyToken

Parameters:

- id (*required*): ID của email cần xóa.

Response:

- 200 OK: Xóa email thành công.
- 404 Not Found: Không tìm thấy email để xóa.
- 500 Internal Server Error: Lỗi hệ thống.

7.4. Cập nhật email

Method: PUT

URL: /mails/update/:id

Middleware: verifyToken

Parameters:

- id (*required*): ID của email cần cập nhật.

Body:

- name (*required*): Tên người gửi.
- subject (*required*): Chủ đề email.
- content (*required*): Nội dung email.

Response:

- 200 OK: Cập nhật email thành công.
- 404 Not Found: Không tìm thấy email để cập nhật.
- 500 Internal Server Error: Lỗi hệ thống.

Web Application: Mở giao diện, giải thích các chức năng.

CHƯƠNG IV XÂY DỰNG ỨNG DỤNG

1. Kết quả thực hiện



```
PS C:\Users\ADMIN\Downloads\emkt-main> cd api
PS C:\Users\ADMIN\Downloads\emkt-main\api> node src/server.js
Server is running on port 5800
Connected to the database
[]

PS C:\Users\ADMIN\Downloads\emkt-main> cd web
PS C:\Users\ADMIN\Downloads\emkt-main\web> npm start
```

4

Hình 4 : cách chạy terminal

Đăng Nhập

Email

2251330006@ut.edu.vn

Mật khẩu

.....

Đăng Nhập



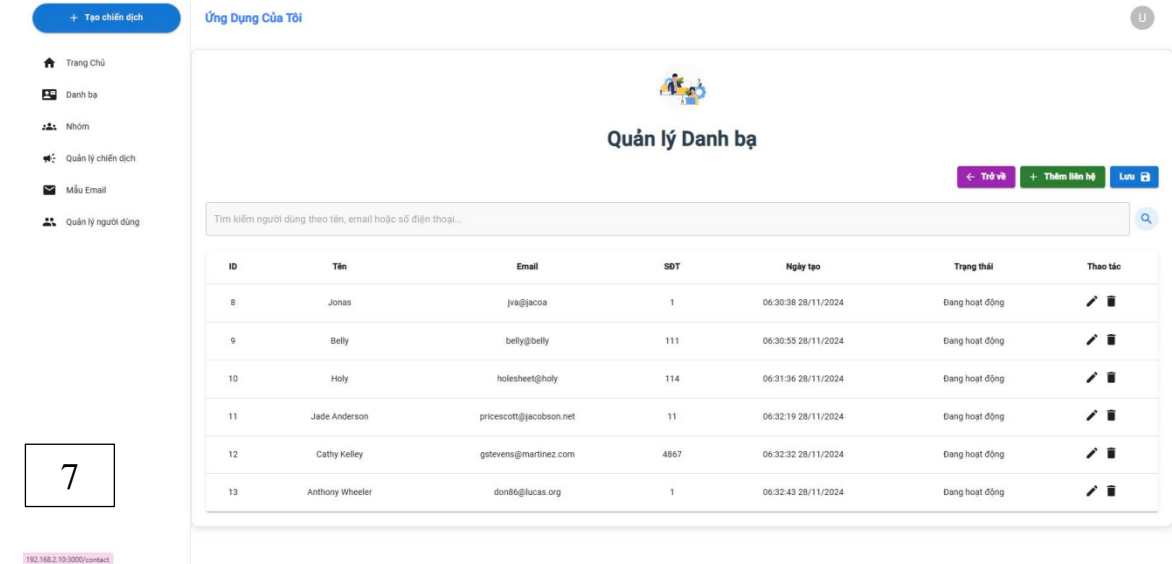
5

Chưa có tài khoản? [Đăng ký ngay](#)

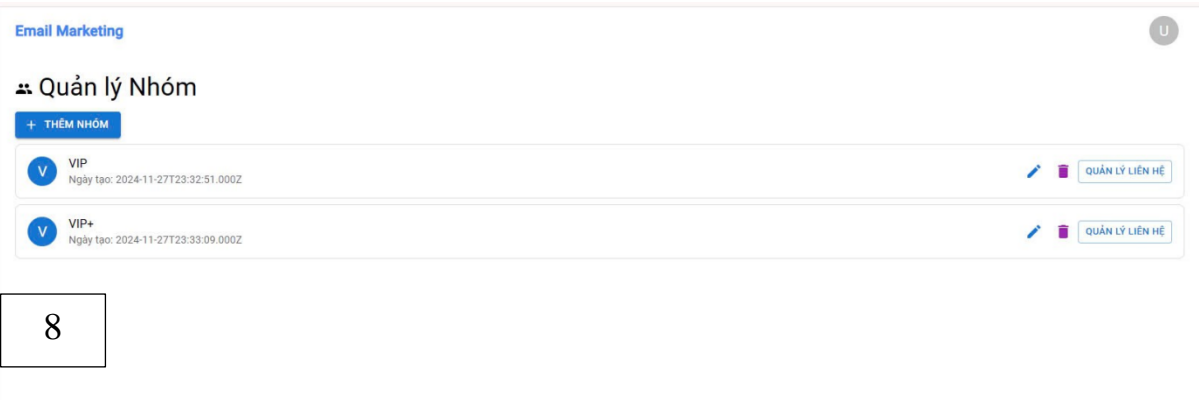
Hình 5 :Trang đăng nhập đăng ký



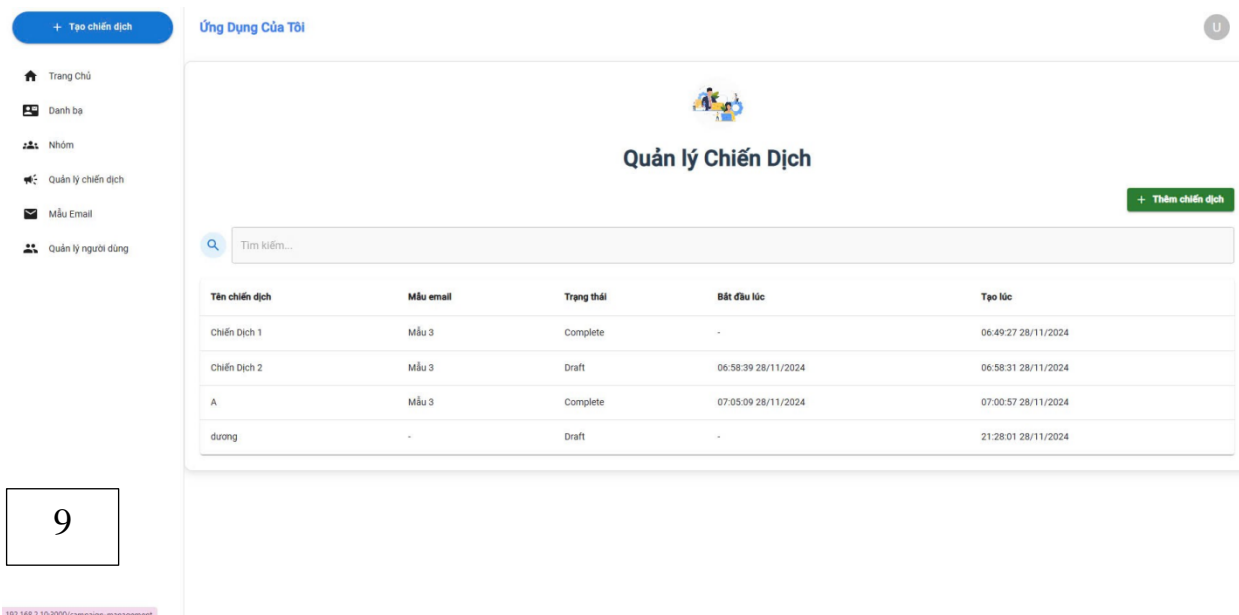
Hình 6 :Trang chủ



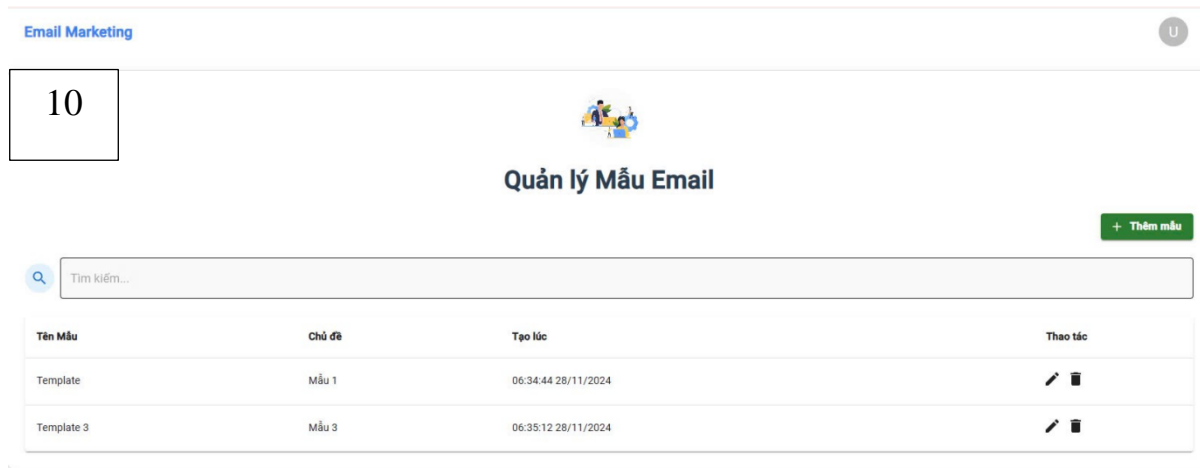
Hình 7 :Danh bạ



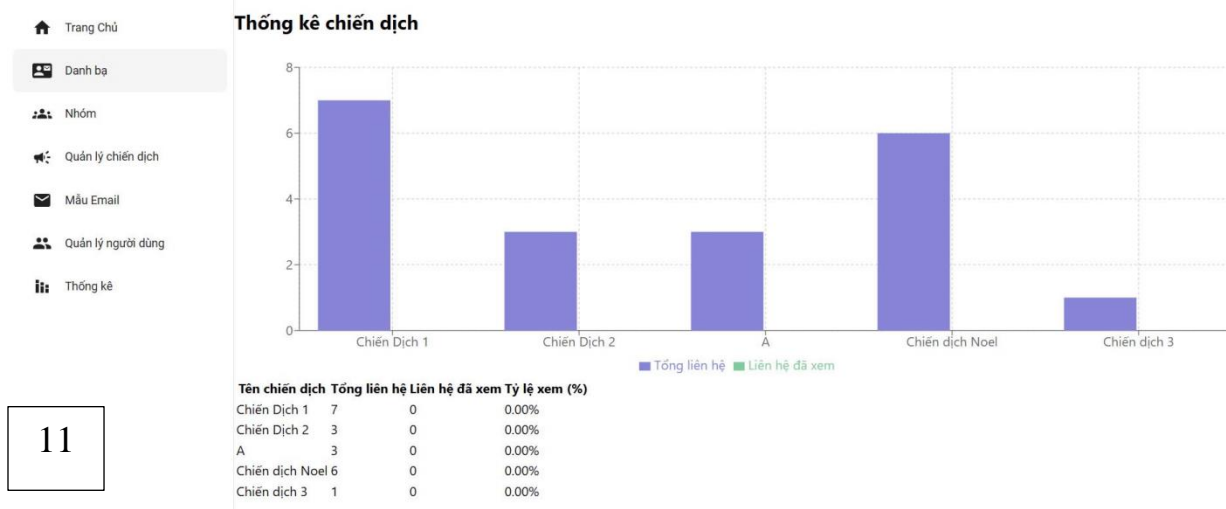
Hình 8: Nhóm



Hình 9 :Quản lí chiến dịch



Hình 10 : Mẫu email



Hình 11 :Thống kê

Kết luận

Trong thời đại công nghệ thông tin phát triển nhanh chóng, việc ứng dụng công nghệ vào quản lý chiến dịch marketing qua email trở nên ngày càng quan trọng. Dự án Email Marketing mà chúng tôi đã thực hiện không chỉ là một sản phẩm phần mềm, mà còn là một giải pháp toàn diện nhằm nâng cao hiệu quả trong việc tiếp cận và tương tác với khách hàng.

Mã nguồn của ứng dụng được xây dựng trên nền tảng React cho phần frontend và Node.js cho phần backend, cho phép tạo ra một trải nghiệm người dùng mượt mà và hiệu quả. Các thành phần giao diện người dùng được thiết kế với sự chú ý đến tính thân thiện và dễ sử dụng. Chúng tôi đã sử dụng các thư viện như Material-UI để cung cấp cho người dùng một giao diện hiện đại, trực quan và dễ dàng điều hướng.

Các chức năng chính của ứng dụng bao gồm quản lý người dùng, quản lý chiến dịch, và gửi email marketing, tất cả đều được tích hợp một cách mạch lạc. Giao diện dashboard được thiết kế để cung cấp cái nhìn tổng quan về các chiến dịch, cho phép người dùng dễ dàng theo dõi và quản lý các hoạt động của mình. Các thành phần như bảng điều khiển, biểu mẫu nhập liệu và thông báo đều được tối ưu hóa để người dùng có thể thao tác một cách nhanh chóng và hiệu quả.

Chúng tôi cũng đã chú trọng đến việc bảo mật thông tin người dùng thông qua việc sử dụng xác thực JWT và mã hóa dữ liệu. Điều này không chỉ giúp bảo vệ thông tin cá nhân mà còn tạo niềm tin cho người dùng khi sử dụng ứng dụng.

Cuối cùng, dự án Email Marketing không chỉ là một sản phẩm phần mềm đơn thuần, mà còn là một công cụ mạnh mẽ hỗ trợ doanh nghiệp trong việc xây dựng và duy trì mối quan hệ với khách hàng. Với những tính năng và giao diện thân thiện, ứng dụng hứa hẹn sẽ mang lại trải nghiệm tốt nhất cho người dùng, đồng thời góp phần nâng cao hiệu quả trong các chiến dịch marketing.