|  |
| --- |
| **TRƯỜNG ĐẠI HỌC KHOA HỌC – ĐẠI HỌC HUẾ**  **BẢN THUYẾT MINH SẢN PHẨM DỰ THI**  **CUỘC THI LẬP TRÌNH DÀNH CHO HỌC SINH TRUNG HỌC PHỔ THÔNG VÀ SẢN PHẨM SÁNG TẠO CÔNG NGHỆ THÔNG TIN DÀNH CHO SINH VIÊN CAO ĐẲNG, ĐẠI HỌC NĂM 2024 (HUE-ICT CHALLENGE-2024)**    **Tên sản phẩm:**  **XÂY DỰNG WEBSITE ĐỌC VÀ CHIA SẺ BLOG**  **Lĩnh vực: Phần mềm**    **Tác giả/nhóm tác giả:** Phạm Hùng Xuân Sang  ***Huế, ngày 07 tháng 06 năm 2024*** |

**I. NỘI DUNG**

**1. Tên sản phẩm**

Xây dựng website đọc và chia sẻ blog.

**2. Ý tưởng của người dự thi**

Kỷ nguyên công nghệ số bùng nổ đã mang đến những thay đổi to lớn trong cách con người kết nối và chia sẻ thông tin. Nhu cầu cập nhật tin tức, học hỏi những điều mới mẻ và chia sẻ ý tưởng của mỗi cá nhân ngày càng tăng cao. Trong bối cảnh đó blog trở thành một nền tảng lý tưởng để đáp ứng nhu cầu này, blog cho phép mọi người đăng tải bài viết, chia sẻ khoảnh khắc, kiến thức, trải nghiệm và kết nối với những người có chung niềm đam mê, sở thích.

**II. MÔ TẢ VỀ SẢN PHẨM**

**1. Trình bày tính mới, tính sáng tạo của sản phẩm**

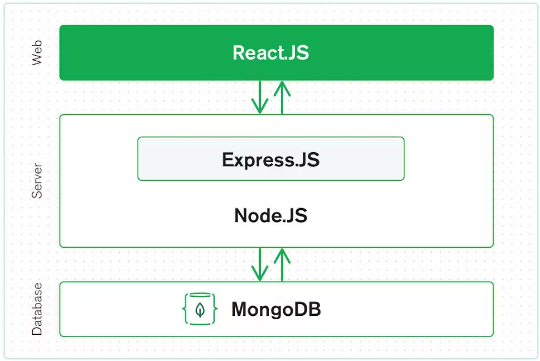
Sản phẩm phần mềm đọc và chia sẻ blog đem đến một loạt các tính năng độc đáo và tiên tiến, mang lại trải nghiệm người dùng vượt trội. Với trình soạn thảo văn bản chuẩn WYSIWYG (What You See Is What You Get), người dùng có thể trực quan hóa kết quả bài viết ngay trong quá trình soạn thảo. Đặc biệt, tính năng chuyển đổi giọng nói thành văn bản cho phép người dùng viết bài bằng giọng nói, giúp tiết kiệm thời gian và nâng cao hiệu suất. Ngoài ra, người dùng có thể cá nhân hóa danh sách bài viết lưu trữ bằng cách tạo các bộ sưu tập riêng biệt, tạo ra một không gian lưu trữ khoa học và tiện lợi. Tính năng chat online, hỗ trợ giao tiếp thời gian thực, kết nối người dùng với nhau, tạo nên một cộng đồng sôi động và tích cực.

**2. Các nguyên vật liệu làm ra mô hình, sản phẩm (nếu có)**

**2.1. Công nghệ lập trình**

Phần mềm áp dụng mô hình lập trình MERN Stack, bao gồm bốn công nghệ chủ chốt: MongoDB, ExpressJS, ReactJS, và NodeJS. Đây là bộ công nghệ mạnh mẽ dựa trên ngôn ngữ lập trình JavaScript, chuyên dùng để phát triển các ứng dụng web full-stack. Tích hợp nhiều thư viện và công nghệ bên thứ ba để tăng cường khả năng xây dựng và vận hành website, chẳng hạn như Firebase, Mongoose, Editor.js, Bcryptjs, JWT, Node-cron, Nodemailer, Socket.io, TailwindCSS, Redux, Flowbite-react, React-chartjs-2, React-google-recaptcha, và React-hot-toast. Những công nghệ này không chỉ giúp nâng cao hiệu suất mà còn mở rộng khả năng và tính năng của sản phẩm.

Kiến trúc MERN cho phép chúng ta dễ dàng xây dựng ứng dụng web theo kiến trúc 3 tầng front-end, back-end và database hoàn toàn bằng cách sử dụng JavaScript và JSON.



Hình 1: Kiến trúc ứng dụng web bằng MERN Stack

**2.2. Các nền tảng tích hợp**

Trong quá trình phát triển sản phẩm, em đã tích hợp một loạt các nền tảng công nghệ hàng đầu nhằm tối ưu hóa chức năng và trải nghiệm người dùng. Dưới đây là các nền tảng đã được sử dụng:

2.2.1. Firebase

Firebase không chỉ là một nền tảng mạnh mẽ cho việc phát triển ứng dụng di động và web mà còn là một công cụ tích hợp quan trọng cho việc xây dựng các tính năng như đăng nhập và lưu trữ dữ liệu. Em sử dụng Firebase Authentication để cho phép người dùng đăng nhập bằng tài khoản Google một cách dễ dàng và an toàn. Đồng thời, Firebase Storage giúp lưu trữ các hình ảnh được tải lên một cách hiệu quả và linh hoạt.

2.2.2. PayOS

Đối với việc xây dựng các tính năng thanh toán và quản lý giao dịch trực tuyến, em đã tích hợp PayOS. Đây là một nền tảng mạnh mẽ và linh hoạt giúp thực hiện các giao dịch thanh toán một cách an toàn và hiệu quả.

2.2.3. Nodemailer

Nodemailer là một công cụ mạnh mẽ để gửi email tự động từ ứng dụng, giúp tích hợp chức năng gửi email thông qua các giao thức email phổ biến như SMTP và IMAP.

2.2.4. Editor.js

Editor.js là một trình soạn thảo văn bản linh hoạt và dễ sử dụng, cho phép người dùng tạo ra nội dung đa dạng và phong phú trên ứng dụng. Việc tích hợp Editor.js giúp tăng cường tính năng và sự linh hoạt cho người dùng trong việc tạo nội dung.

2.2.5. Dom-speech-recognition

Dom-speech-recognition là một công nghệ tiên tiến được tích hợp để cung cấp khả năng nhận dạng giọng nói trong trình duyệt. Điều này tạo ra một trải nghiệm tương tác âm thanh độc đáo và thuận tiện.

**2.3. Lưu trữ dữ liệu**

Dữ liệu của website được tổ chức và quản lý một cách hiệu quả thông qua hệ quản trị cơ sở dữ liệu phi quan hệ MongoDB, sử dụng nền tảng MongoDB Atlas - một dịch vụ cơ sở dữ liệu đa đám mây tiên tiến. MongoDB Atlas không chỉ cung cấp khả năng mở rộng linh hoạt mà còn đảm bảo tính sẵn sàng và bảo mật cao, giúp dễ dàng quản lý và truy xuất dữ liệu trên nhiều môi trường khác nhau.

Để lưu trữ và quản lý hình ảnh mà người dùng tải lên, dự án sử dụng phương pháp tích hợp thêm Firebase Storage. Firebase Storage không chỉ cung cấp giải pháp lưu trữ an toàn và đáng tin cậy mà còn trả về đường dẫn URL của các hình ảnh, cho phép lưu trữ các đường dẫn này trong MongoDB. Sự kết hợp này đảm bảo rằng dữ liệu và hình ảnh được tổ chức một cách khoa học, dễ dàng truy cập và quản lý, đồng thời mang lại trải nghiệm người dùng mượt mà và liền mạch.

**3. Cách lắp ráp, cài đặt sản phẩm**

+ Bước 1: Truy cập github tại đường dẫn <https://github.com/PhamHungXuanSang/MERN-blog>

+ Bước 2: Mở terminal máy tính và thực thi dòng lệnh:

git clone https://github.com/PhamHungXuanSang/MERN-blog.git

+ Bước 3: Cài đặt các gói (package)

**Cài đặt các gói tại thư mục gốc của dự án:**

* Mở terminal và điều hướng đến thư mục gốc của dự án
* Gõ lệnh sau để cài đặt các gói cần thiết: npm install

**Cài đặt các gói cho phần client:**

* Từ thư mục gốc của dự án, thay đổi thư mục làm việc sang thư mục chứa mã nguồn client bằng lệnh sau: cd client
* Sau đó, tiếp tục cài đặt các gói cho phần client bằng lệnh: npm install

+ Bước 4: Chạy dự án tại máy tính cá nhân

**Khởi chạy server ứng dụng:**

* Mở terminal và điều hướng đến thư mục gốc của dự án
* Gõ lệnh chạy server: npm run dev

**Khởi chạy client ứng dụng:**

* Mở new terminal, thay đổi thư mục làm việc sang thư mục chứa mã nguồn client: cd client
* Gõ lệnh chạy client: npm run dev

**Truy cập trang web tại địa chỉ:** localhost:5173

**4. Nguyên tắc hoạt động, vận hành của sản phẩm dự thi**

Sau đây là giới thiệu về các chức năng chính, workflow của ứng dụng:

**4.1. Chức năng Đăng nhập/đăng ký bằng tài khoản google**

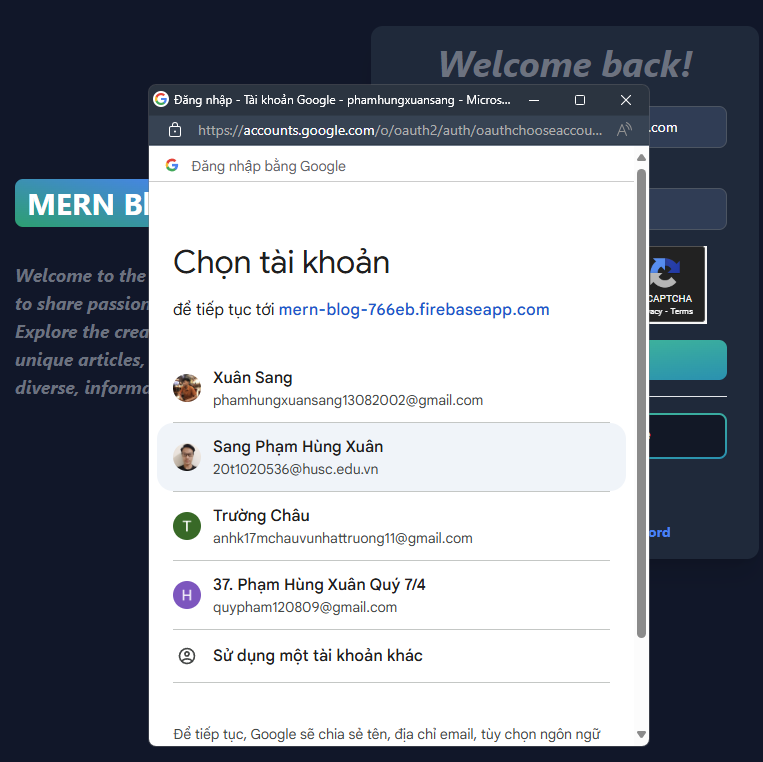
Chức năng này cho phép người dùng đăng ký tài khoản mới và sử dụng các tính năng nâng cao nhanh chóng mà không cần tạo tài khoản mới bằng cách thủ công, nâng cấp trải nghiệm và khả năng bảo mật nhờ xác thực của Google.

Firebase Authentication được em lựa chọn để tích hợp vì nó cung cấp hệ thống xác thực mạnh mẽ và dễ dàng tích hợp với ứng dụng web.

Mã hóa mật khẩu sử dụng bcryptjs: Trong quá trình đăng ký, mật khẩu người dùng được mã hóa bằng bcryptjs trước khi lưu vào cơ sở dữ liệu, bảo vệ mật khẩu khỏi các cuộc tấn công brute-force và đảm bảo an toàn ngay cả khi cơ sở dữ liệu bị lộ.

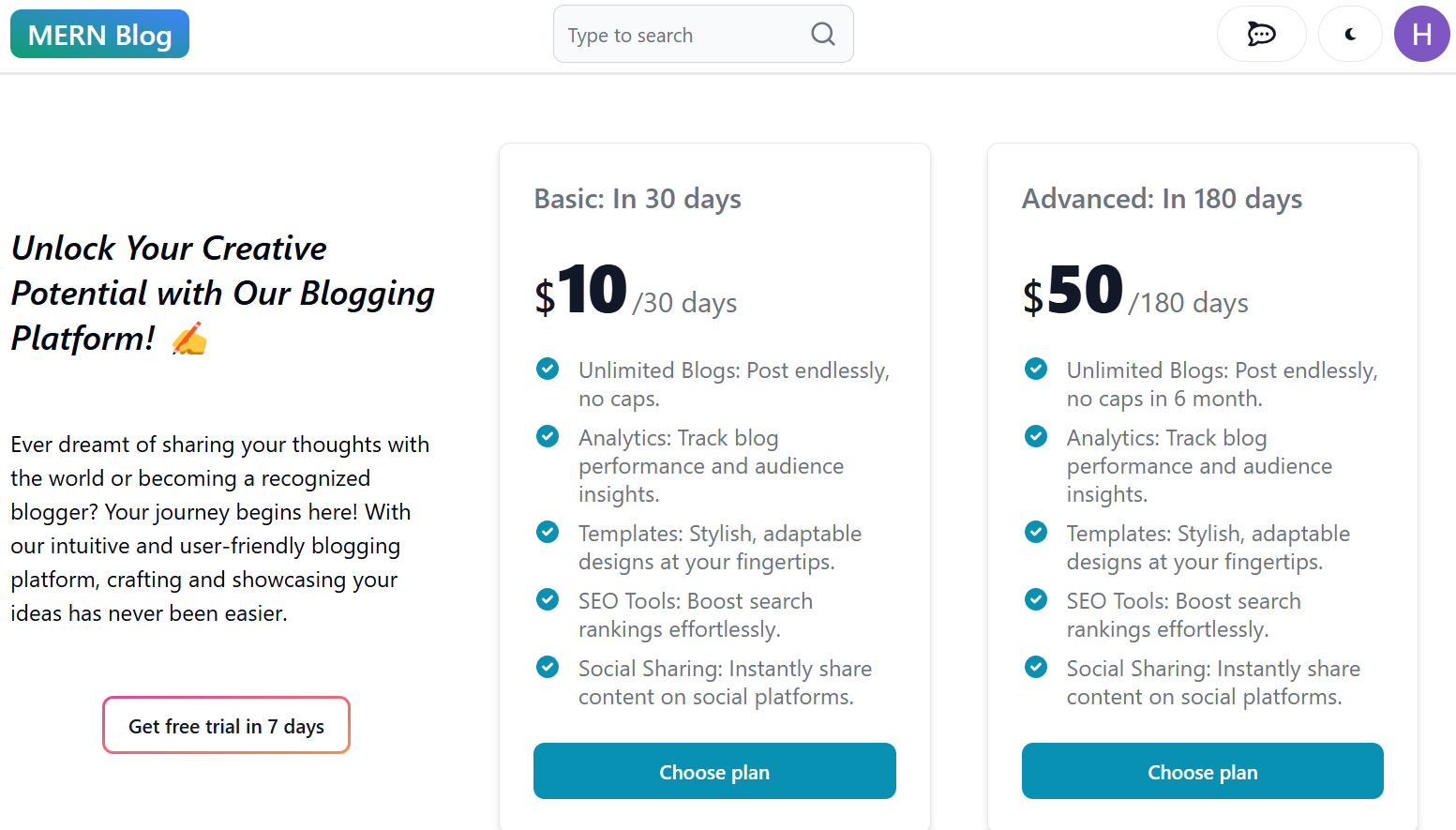
Quản lý phiên làm việc với access\_token: Sau khi đăng nhập thành công, hệ thống tạo một access\_token bằng JWT, chứa thông tin xác thực của người dùng với thời hạn 1 ngày. Token này lưu trong cookies với thuộc tính httpOnly: true, ngăn chặn truy cập từ JavaScript phía client.

Quản lý phiên làm việc và điều hướng: Hệ thống kiểm tra tính hợp lệ của access\_token mỗi khi người dùng truy cập các chức năng đòi hỏi đăng nhập. Nếu token hết hạn hoặc không hợp lệ, người dùng sẽ bị điều hướng về trang đăng nhập, đảm bảo tính bảo mật tuyệt đối cho các chức năng nâng cao và có tính phí.



Hình 2: Chức năng đăng nhập/đăng ký bằng tài khoản google

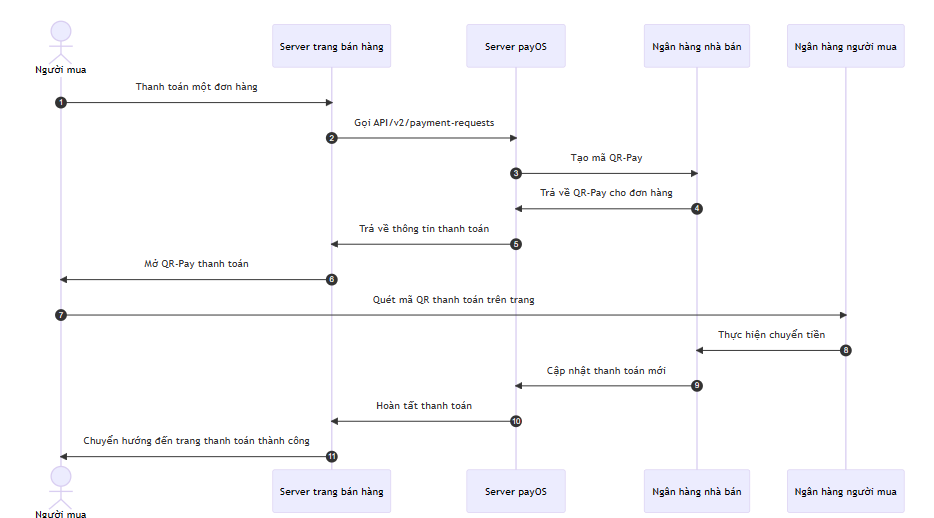
**4.2. Chức năng Thanh toán mua gói cước đăng bài viết**



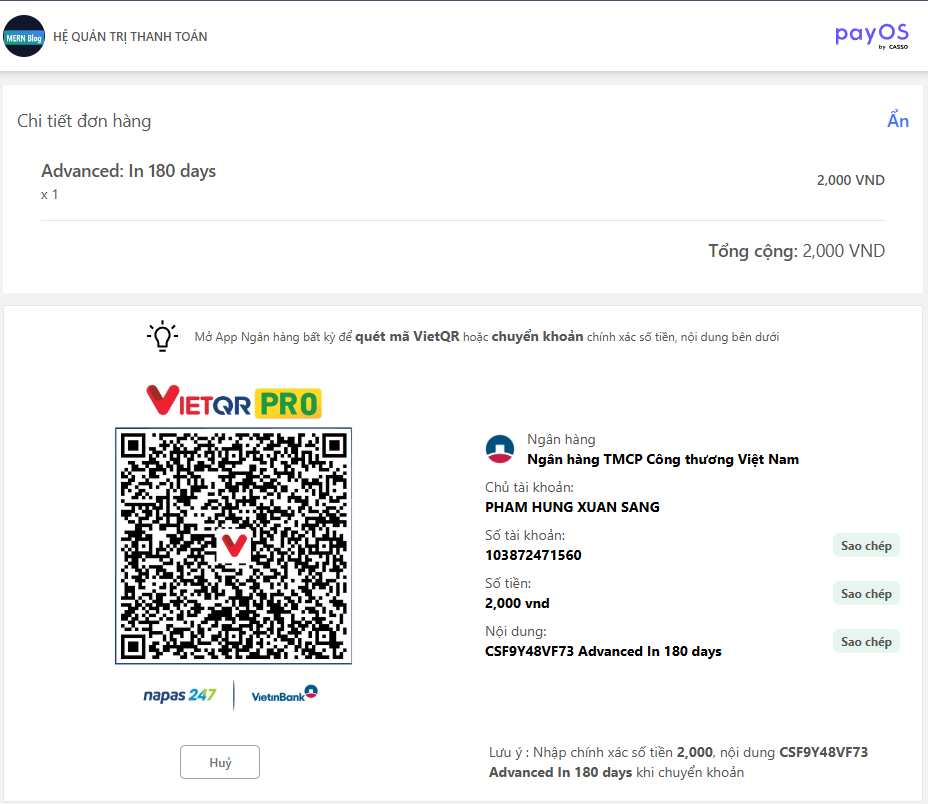
Hình 3: Giao diện chọn gói cước muốn mua

4.2.1. Thanh toán chuyển khoản ngân hàng quét mã QR Code

Đã tích hợp sử dụng thành công cổng thanh toán payOS liên kết với tài khoản ngân hàng cá nhân để khách hàng thanh toán QR Code. PayOS by CASSO là cổng thanh toán ứng dụng mô hình A2A Payment đầu tiên tại Việt Nam. A2A Payment (Account to Account Payment) cho phép tiền chảy trực tiếp từ tài khoản người mua đến tài khoản người bán mà không chịu sự can thiệp từ bên thứ ba (trung gian thanh toán), giúp quá trình thanh toán nhanh chóng, tiện lợi và an toàn hơn.



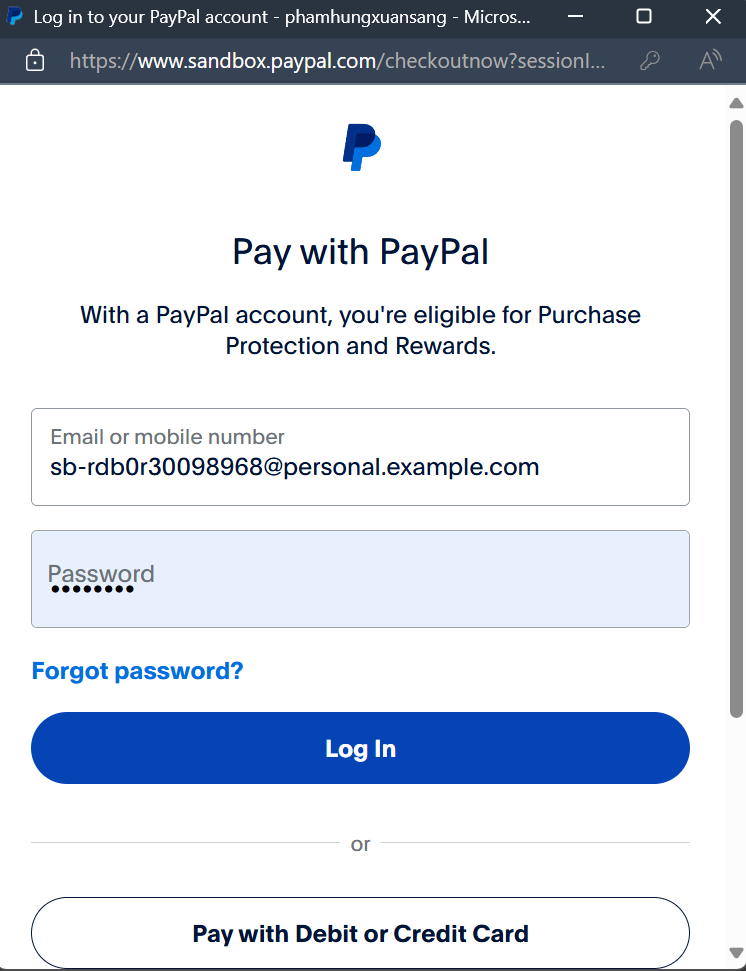
Hình 4: Sequence diagram thanh toán gói cước quét mã QR Code bằng payOS



Hình 5: Giao diện quét mã QR Code khi mua gói cước đăng bài viết

4.2.2. Thanh toán quốc tế bằng dịch vụ thanh toán PayPal

Đã thử nghiệm tích hợp dịch vụ thanh toán PayPal thành công trên môi trường sandbox để mô phỏng người dùng quốc tế có thể thực hiện thanh toán gói cước đăng bài viết.



Hình 6: Giao diện thanh toán gói cước bằng PayPal

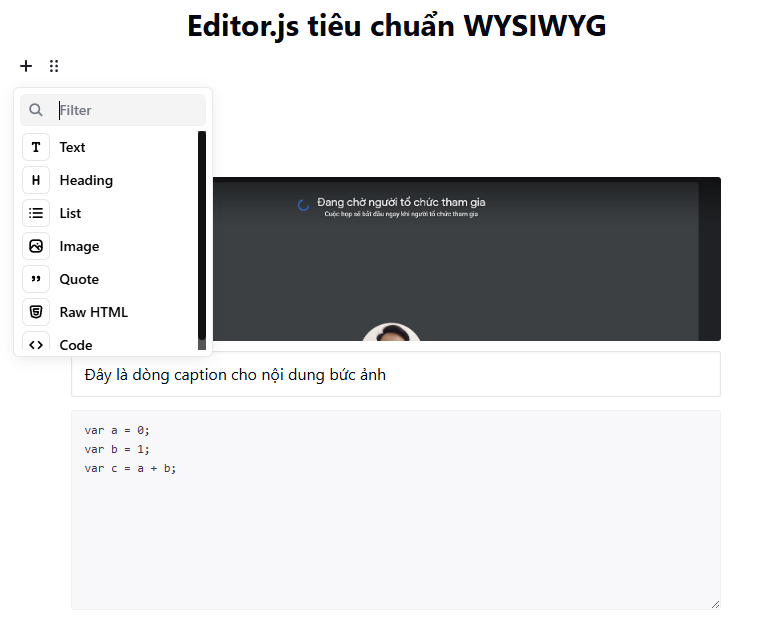
Thông tin account để test thanh toán trên môi trường sandbox:

* Email: [sb-rdb0r30098968@personal.example.com](mailto:sb-rdb0r30098968@personal.example.com)
* Password: EB))2xyD

**4.3. Chức năng Đăng bài viết**

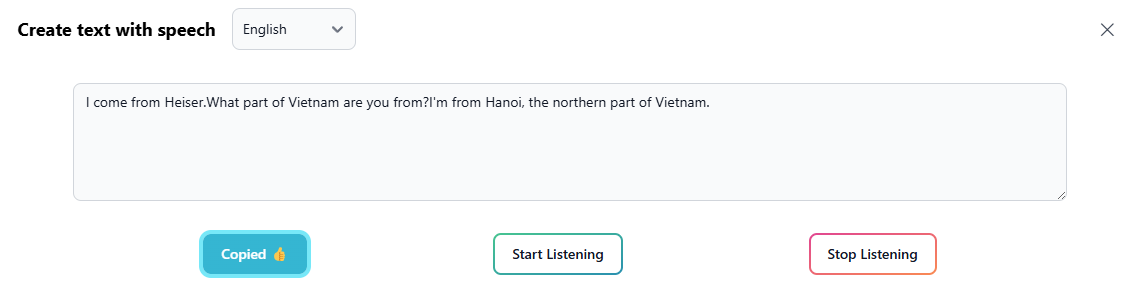
Chức năng đăng bài viết được tích hợp trình soạn thảo Editor.js theo tiêu chuẩn WYSIWYG (what you see is what you get) tạo nên trải nghiệm mới mẻ chuyên nghiệp cho người dùng. Bên cạnh khả năng viết nội dung, người dùng có thể trực tiếp thấy được kết quả nội dung mình tạo ra ngay trong quá trình soạn thảo giảm thiểu thời gian chỉnh sửa và đảm bảo bài viết đầu ra hiển thị đúng như ý muốn. Editor.js cung cấp rất nhiều plugin và công cụ khác nhau giúp người dùng có thể thêm các định dạng cần thiết như tiêu đề, đoạn văn, danh sách đánh số, danh sách bullet, trích dẫn, hình ảnh, video, bảng biểu, liên kết, và nhiều hơn nữa giúp bài viết trở nên phong phú và hấp dẫn hơn rất nhiều.

Được cài đặt và tối ưu hóa để hoạt động mượt mà trên cả máy tính để bàn và thiết bị di động thuận tiện cho người sử dụng có thể viết và đăng bài mọi lúc, mọi nơi mà không gặp bất kỳ trở ngại nào.



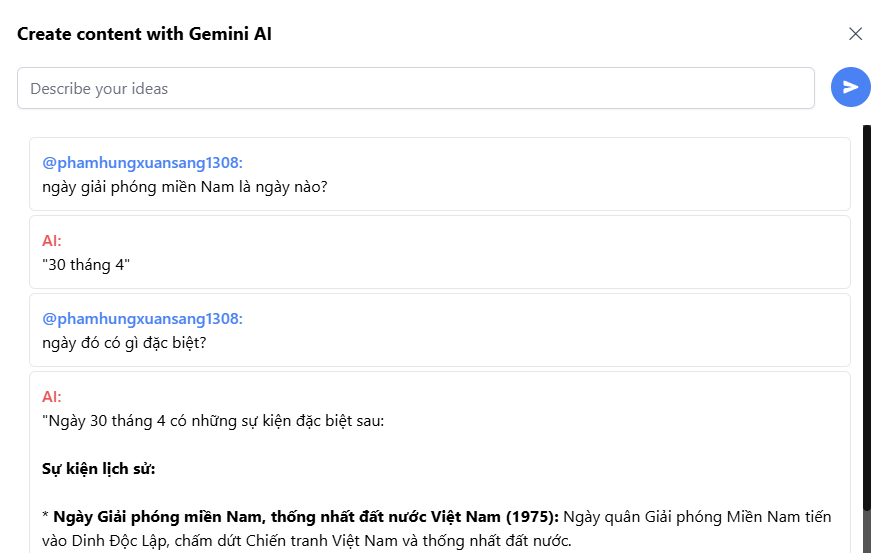
Hình 7: Giao diện soạn thảo nội dung bài viết bằng editor.js

Tích hợp soạn thảo văn bản bằng chức năng speech to text (nhận diện giọng nói chuyển thành văn bản) hỗ trợ hai ngôn ngữ tiếng Anh và tiếng Việt.



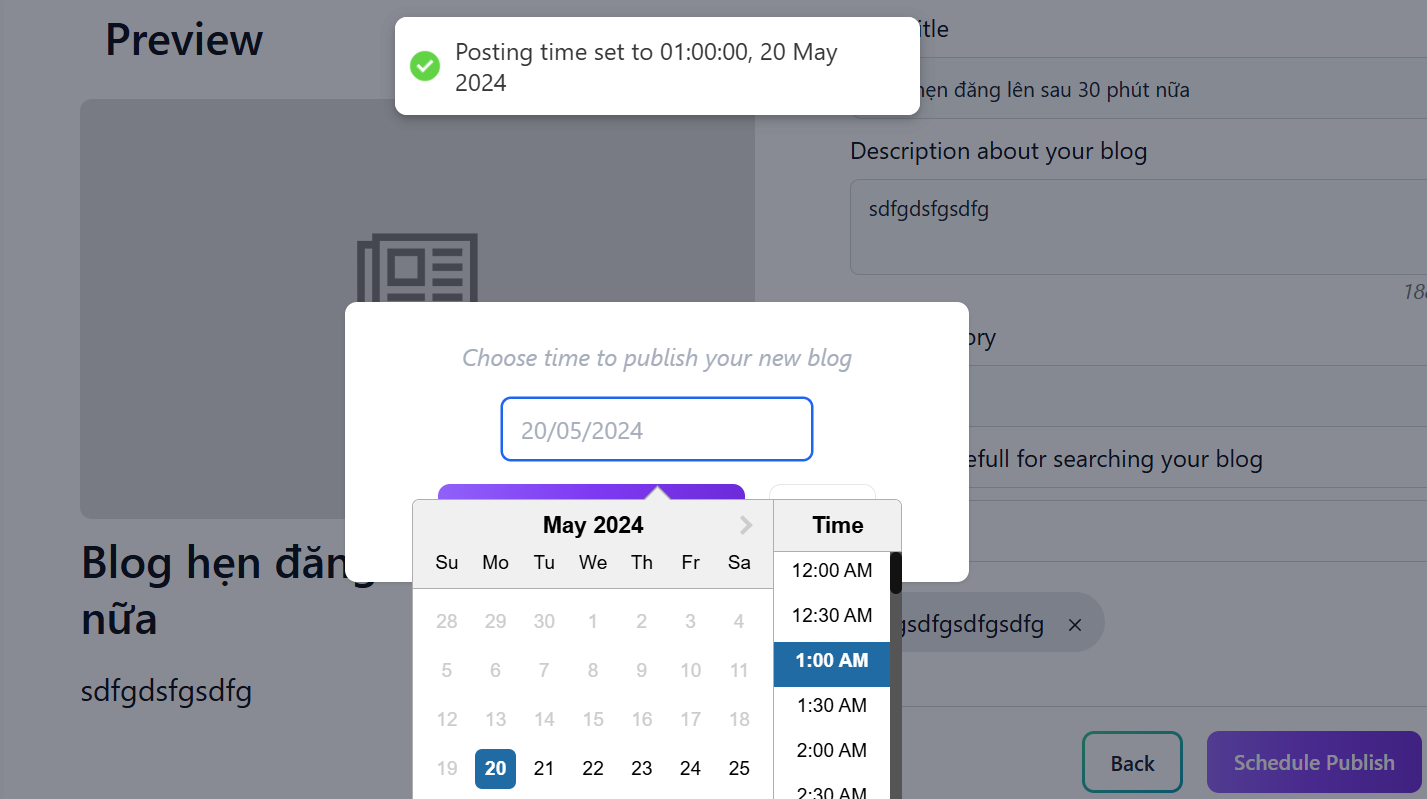
Hình 8: Giao diện chức năng chuyển giọng nói thành văn bản

Tích hợp chatbot AI hỗ trợ sáng tạo nội dung sử dụng API của chatbot đến từ Google Gemini.



Hình 9: Giao diện tích hợp chatbot Gemini AI hỗ trợ sáng tạo nội dung

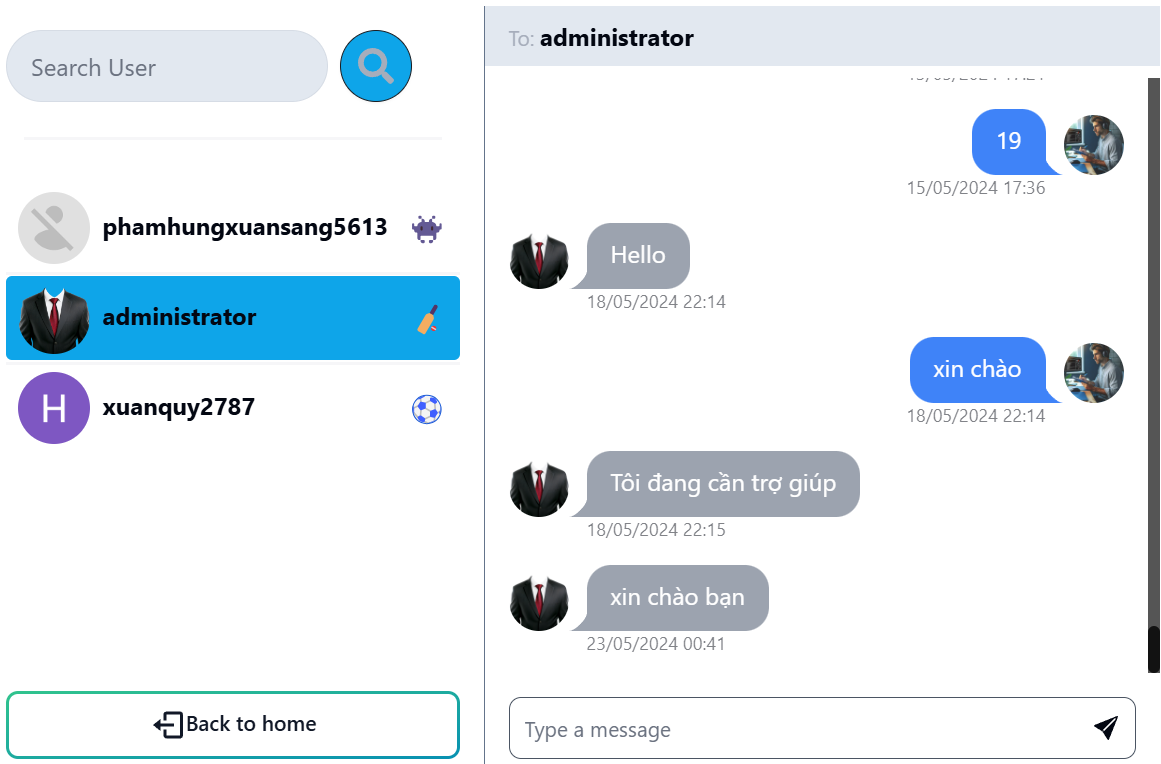
**4.4. Chức năng Hẹn giờ đăng bài viết** Chức năng này được thiết kế sử dụng một cron job chạy ở server NodeJS thông qua thư viện node-cron thực hiện một công việc kiểm tra các bài viết đang chờ đăng đã đến thời điểm đăng hay chưa để tiến hành đăng bài tự động.



Hình 10: Giao diện cài đặt thời gian tự động đăng bài viết

**4.5. Chức năng Chat online**

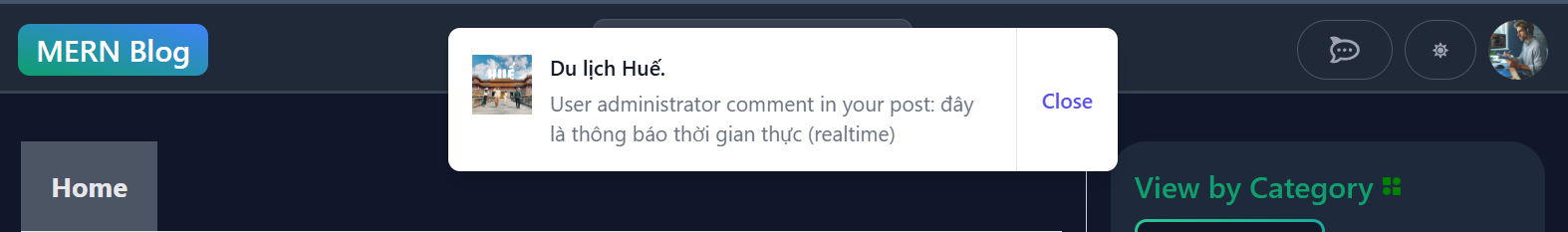
Chức năng này cho phép người dùng gửi nhận tin nhắn realtime với một người dùng khác bằng cách sử dụng giao thức websocket và thư viện socket-io.



Hình 11: Giao diện chức năng chat online

**4.6. Tính năng Gửi nhận thông báo về hoạt động và tin nhắn thời gian thực**

Tính năng này cho phép người dùng nhận mọi loại thông báo về hoạt động của tài khoản theo thời gian thực. Ví dụ: Thông báo bài viết mới, lượt like mới, lượt subscribe mới, comment, reply comment, tin nhắn mới, …



Hình 12: Ví dụ về tính năng gửi nhận thông báo thời gian thực

**5. Khả năng ứng dụng của sản phẩm**

**Giới thiệu về render.com**

Sau quá trình thực hiện phát triển dự án. Em đã tìm hiểu và sử dụng nền tảng render.com – một dịch vụ cung cấp khả năng hosting ứng dụng web và dịch vụ API một cách dễ dàng. Đối với việc deploy dự án sử dụng MERN Stack, render.com thật sự là một lựa chọn vượt trội với sự hỗ trợ mạnh mẽ cho cả MongoDB, ExpressJS, ReactJS và NodeJS.

**Đường dẫn truy cập ứng dụng**

Sản phẩm phần mềm đọc và chia sẻ blog đã được triển khai và thử nghiệm thực tế trên nền tảng mạng internet thông qua sử dụng render.com, cho phép người dùng truy cập và sử dụng một cách dễ dàng. Người dùng có thể trải nghiệm sản phẩm thông qua đường link: <https://mern-blog-csov.onrender.com>.

**6. Hiệu quả đạt được của sản phẩm**

Sau chặng đường dài học tập, tìm hiểu, lên ý tưởng và xây dựng hệ thống, tuy chưa thể hoàn thiện một cách chỉnh chu nhất nhưng về cơ bản thì em đã cố gắng hoàn thành đúng tiến độ các đầu mục công việc có trong dự án.

Sử dụng thành công mô hình MERN Stack kết hợp các công nghệ hỗ trợ khác để xây dựng trang web sử dụng RESTful API đáp ứng tốt về yêu cầu chức năng và giao tiếp truyền nhận dữ liệu.

Giao diện viết bằng ReactJS kết hợp sử dụng các thư viện nâng cao hơn là Redux, Flowbite React, Editor.js hoàn thành giao diện có tính thẩm mỹ và có thể thực hiện các thao tác tương tác của người dùng. Có hỗ trợ responsive trên nhiều kích cỡ màn hình thiết bị sử dụng khác nhau.

Thành công tích hợp websocket để tạo thông báo popup và chức năng gửi nhận tin nhắn realtime.

Trang web được thiết kế sử dụng JWT và access token để bảo mật và phân quyền các chức năng khác nhau dành cho cả 3 loại đối tượng người sử dụng (người dùng chưa đăng nhập, người dùng đã đăng nhập và quản trị viên) tạo ra trải nghiệm cá nhân hóa, an toàn và bảo mật cho mọi đối tượng sử dụng.

Thành công deploy source code để vận hành trang web trong điều kiện thực tế trên nền tảng mạng internet.

**7. Các tài liệu khác và địa chỉ đăng tải sản phẩm**

Toàn bộ mã nguồn của dự án đã được đăng tải lên github.com tại: <https://github.com/PhamHungXuanSang/MERN-blog>

Truy cập trải nghiệm website tại địa chỉ:

<https://mern-blog-csov.onrender.com>

**8. Cam kết về bản quyền sản phẩm**

Sản phẩm chưa từng được công bố hoặc tham gia trong bất kỳ cuộc thi nào.

Sản phẩm đúng bản quyền của sinh viên dự thi, trường hợp có sử dụng mã nguồn mở phải tuân thủ các yêu cầu của giấy phép mã nguồn mở của các tổ chức, cá nhân phát hành mã nguồn mở.