TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO GIỮA KỲ**

**MÔN** PHÁT TRIỂN ỨNG DỤNG WEB VỚI NODEJS

**Real-time Chat Application**

*Người hướng dẫn*: **GV NGUYỄN THANH QUÂN**

*Người thực hiện*: **VÕ THỊ PHƯƠNG THUỲ – 51900693**

**TRẦN MINH TRÍ – 52000815**

*Khoá****:* 23-24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ**

**MÔN** PHÁT TRIỂN ỨNG DỤNG WEB VỚI NODEJS

**Real-time Chat Application**

*Người hướng dẫn*: **GV NGUYỄN THANH QUÂN**

*Người thực hiện*: **VÕ THỊ PHƯƠNG THUỲ – 51900693**

**TRẦN MINH TRÍ – 52000815**

*Khoá****:* 23-24**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2023**

LỜI CẢM ƠN

Đầu tiên, nhóm chúng em muốn gửi lời cảm ơn chân thành nhất đến Gv Nguyễn Thanh Quân. Thầy đã bỏ thời gian công sức hết sức quý báu của bản thân rất nhiều để hỗ trợ và vạch ra những phương án, hướng đi tốt nhất cho bài báo cáo của chúng em trong suốt quá trình chúng em nghiên cứu, học tập thông qua bộ môn của thầy. Cùng với những giáo trình chi tiết và bài giảng dễ hiệu cùng với sự truyền tải kiến thức nhiệt huyết của thầy để chúng em có được một bài báo cáo hoàn thiện với đề tài “PHÁT TRIỂN ỨNG DỤNG WEB VỚI NODEJS ” như ngày hôm nay.

Tiếp theo, nhóm chúng em xin được gửi lời cảm ơn đến khoa Công Nghệ Thông Tin trường Đại học Tôn Đức Thắng vì đã tạo điều kiện tốt nhất cho chúng em được học tập và tìm hiểu kiến thức cũng như áp dụng thực tế đối với bộ môn ‘Phát triển ứng dụng web với nodeis’ này. Khoa CNTT đã luôn không ngừng cung cấp những tài liệu giúp nhóm chúng em hoàn thành đề tài nghiên một cách hiệu quả nhất tại trường Đại học Tôn Đức Thắng nói chung và bộ môn nói riêng.

Cuối cùng, sau quá trình tìm hiệu và có những buổi học từ bộ môn cũng những tài liệu, giáo trình đầy đủ, chi tiết của môn học từ sự chỉ dạy, giúp đỡ của Quý thầy cô, khoa và trường Đại học Tôn Đức Thắng mà chúng em đã hoàn thiện bài báo cáo tốt nhất hết mức có thể. Bên cạnh lí do giới hạn còn nhiều mặt kiến thức của bản thân chúng em còn chưa trọn vẹn và khả năng lập luận chưa thực sự hoàn hảo nên nhóm vẫn còn nhiều thiếu sót và hạn chế không đáng có, kính mong sự xem xét, chỉ dẫn và đóng góp của Quý thầy cô giáo,Khoa cũng như Trường để bài báo cáo của chúng em sẽ được chỉnh chu hơn. Chính những sự góp ý và đóng góp từ thầy cô,khoa cũng như trường sẽ giúp chúng em hoàn thiện tốt hơn về những bài báo cáo tương lai. Nhóm chúng em kính chúc Quý thầy cô, Khoa Công Nghệ Thông Tin cũng như Trường Đại học Tôn Đức Thắng và các bạn bè – những người luôn hỗ trợ, quan tâm chúng em luôn tràn đầy sức khỏe, thành công,…

NHÓM CHÚNG EM XIN CHÂN THÀNH CẢM ƠN!

**ĐỒ ÁN ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Nhóm chúng em xin cam đoan đây báo cáo của riêng chúng em và được sự hướng dẫn của **GV NGUYỄN THANH QUÂN**. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những tài liệu, mô hình được sử dụng trong bài báo được thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các bài báo, tác giả, cơ quan khác tổ chức khác đều có trích dẫn ở tài liệu tham khảo.

**Nếu phát hiện có bất kỳ sự gian lận nào nhóm chúng em xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện.

*TP. Hồ Chí Minh, ngày 21 tháng 04 năm 2023*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Võ Thị Phương Thuỳ*

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

**Phần xác nhận của GV hướng dẫn**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

**Phần đánh giá của GV chấm bài**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tp. Hồ Chí Minh, ngày tháng năm

(kí và ghi họ tên)

TÓM TẮT

Hiện nay có rất nhiều hệ thống sử dụng thời gian thực, mọi thứ hầu như là phản hồi ngay lập tức, không có độ trễ hoặc độ trễ thấp. Điều này giúp tăng trải nghiệm người dùng, hỗ trợ các công việc từ xa, làm giảm được các chi phí thời gian di chuyển, tăng hiệu suất làm việc. Nhóm chúng em tìm hiểu về công nghệ websocket đây là công nghệ hỗ trợ giao tiếp 2 chiều giữa máy khách và máy chủ. Từ công nghệ này thì nhóm chúng em áp dụng vào đề tài và thiết kế ứng dụng nhắn tin thời gian thực. Ứng dụng nhắn tin này có thể sử dụng để nhắn tin mọi lúc mọi nơi miễn là có kết nối internet. Mọi người có thể dùng nó để trò chuyện với nhau ngay cả khi không cần gặp mặt trực tiếp. Giúp ích cho đời sống rất nhiều.

PHẦN 1 – TỔNG QUAN ĐỀ TÀI: Tìm hiểu các kiến trúc, nguyên tắc, cách xây dựng, lập trình một ứng dụng nhắn tin thời gian thực.

PHẦN 2 – PHẦN LÝ THUYẾT: Tìm hiểu về front-end và back-end, nodejs, expressjs,… và những thứ liên quan giúp chúng ta xây dựng một ứng dụng web cho đề tài đang xây dựng.

CHƯƠNG 3 – HIỆN THỰC HOÁ: Giới thiệu về các đoạn code đã được viết sẵn, cấu thành nên một bộ khung và các thư viện lập trình được đóng gói. Đồng thời hiểu rõ các giao thức tương ứng với mỗi kiến trúc hệ thống cũng như các thông tin chi tiết về các function (hàm), class (lớp), return type (kiểu dữ liệu trả về), các argument (tham số),... được hỗ trợ bởi các bài hướng dẫn để tạo được website socket cho đề tài được nghiên cứu.

CHƯƠNG 4 - KẾT QUẢ DEMO: Các giao diện cũng như cách thức hiện thực bởi người dùng.

CHƯƠNG 5 - KẾT LUẬN

MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc133008371)

[PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN 3](#_Toc133008372)

[TÓM TẮT 4](#_Toc133008373)

[MỤC LỤC 5](#_Toc133008374)

[DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT 6](#_Toc133008375)

[DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ 7](#_Toc133008376)

[CHƯƠNG 1 – TỔNG QUAN ĐỀ TÀI 9](#_Toc133008377)

[1.1 Giới thiệu đề tài 9](#_Toc133008378)

[CHƯƠNG 5 – KẾT LUẬN 26](#_Toc133008379)

[TÀI LIỆU THAM KHẢO 27](#_Toc133008380)

DANH MỤC KÍ HIỆU VÀ CHỮ VIẾT TẮT

**CÁC CHỮ VIẾT TẮT**

CNTT Công nghệ thông tin

DANH MỤC CÁC BẢNG BIỂU, HÌNH VẼ, ĐỒ THỊ

**DANH MỤC HÌNH**

**DANH MỤC BẢNG**

[Bảng 1. Đăng ký tài khoản 15](#_Toc120917744)

[Bảng 2. Đăng nhập 16](#_Toc120917745)

[Bảng 3. Xem sản phẩm 16](#_Toc120917746)

[Bảng 4. Xem sản phẩm theo loại hàng 17](#_Toc120917747)

[Bảng 5. Xem sản phẩm theo màu sắc 17](#_Toc120917748)

[Bảng 6. Thêm sản phẩm vào giỏ hàng 18](#_Toc120917749)

[Bảng 7. Xem đánh giá sản phẩm 18](#_Toc120917750)

[Bảng 8. Đánh giá sản phẩm 18](#_Toc120917751)

[Bảng 9. Đặt hàng 19](#_Toc120917752)

[Bảng 10. Sử dụng khuyến mãi 20](#_Toc120917753)

[Bảng 11. Xem lịch sử đơn hàng 20](#_Toc120917754)

[Bảng 12. Lưu khuyến mãi 20](#_Toc120917755)

[Bảng 13. Đăng xuất 21](#_Toc120917756)

[Bảng 14. CRUD Nhân viên cửa hàng 21](#_Toc120917757)

[Bảng 15. CRUD Khuyến mãi 22](#_Toc120917758)

[Bảng 16. Xem hàng tồn kho 22](#_Toc120917759)

[Bảng 17. CRUD Sản phẩm 23](#_Toc120917760)

[Bảng 18. Xem doanh thu 23](#_Toc120917761)

[Bảng 19. Quản lý đơn trực tuyến 24](#_Toc120917762)

[Bảng 20. Tạo hóa đơn bán hàng 24](#_Toc120917763)

[Bảng 21. Quản lý đổi trả 25](#_Toc120917764)

CHƯƠNG 1 – TỔNG QUAN ĐỀ TÀI

* 1. Giới thiệu đề tài

Trong cuộc sống, giao tiếp là một phần không thể thiếu đối với mỗi người. Với công nghệ ngày càng phát triển thì việc trao đổi giữa ta và mọi người xung quanh cũng trở nên dễ dàng, thuận tiện hơn. Chỉ cần ngồi một chổ, sử dụng thiết bị điện tử (smartphone, pc, laptop,...) có kết nối internet là ta đã có thể kết nối được với thế giới, có thể trò chuyện, chơi game,... cùng nhau thông qua các ứng dụng, hệ thống hỗ trợ phản hồi thời gian thực (real-time). Vì thế để khám phá, tìm hiểu về nguyên lý hoạt động đằng sau những ứng dụng real-time điển hình mà chúng ta sử dụng hằng ngày như zalo, messenger, facebook,... nên nhóm em quyết định chọn đề tài này.

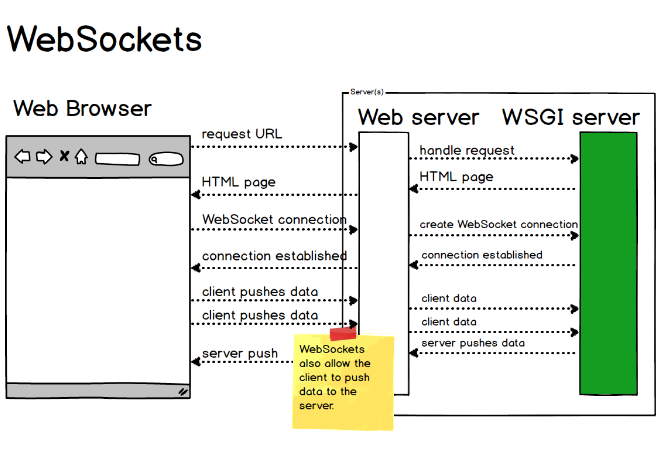
Trong quá trình học NodeJS, tụi em nhận thấy đây là một ngôn ngữ có kiến trúc event-driven, non-blocking I/O model giúp nó có thể xử lý một lượng lớn request đồng thời với độ trễ thấp mà không làm gián đoạn main-thread. Do đó, nó cực kỳ phù hợp với đề tài này và cũng là cơ hội để bọn em có thể tìm hiểu giao thức Web Socket cũng như các biến thể cải tiến nâng cao của nó trong lập trình xử lý real-time.

* 1. Công nghệ sử dụng

**1.2.1 WebSocket**

Trước khi đi vào dự án, ta sẽ cùng tìm hiểu về WebSocket để xem liệu nó khác gì so với HTTP request thông thường. WebSocket là một giao thức cho phép giao tiếp song song, hai chiều giữa client và server thông qua một kết nối TCP, cung cấp độ trễ thấp, cơ chế nhẹ trong việc gửi và nhận ở thời gian thực. Hầu hết các trình duyệt ngày nay đều hỗ trợ giao thức này.

Cách hoạt động của WebSocket cơ bản là nó sẽ cung cấp một cầu nối để server có thể gửi content đến client mà không cần được request bởi client. Đồng thời cho phép các thông báo được truyền qua lại trong khi vẫn giữ cho kết nối được mở. Từ đó tạo nên một giao tiếp hai chiều giữa client và server.



Hình: Minh họa các hoạt động của WebSocket. (nguồn: [Hoạt động của WebSocket](https://topdev.vn/blog/socket-la-gi-websocket-la-gi/#socket-la-gi))

Ưu điểm:

* Kết nối thời gian thực: Cho phép giao tiếp thời gian thực giữa client và server. Cực kỳ hữu dụng trong hầu hết các ứng dụng liên quan đến game online, nhắn tin trực tuyến, sàn giao dịch chứng khoán,... Hay nói chung là các ứng dụng, nền tảng cần tốc độ phản hồi nhanh, độ chính xác cao.
* Độ trễ thấp: Do chỉ sử dụng một chu trình (thread) nên cho độ trễ thấp hơn các HTTP request thông thường (Các http request thường dễ bị chậm, tắc nghẽn vì phải tốn nhiều chi phí trong việc thiết lập một kết nối mới cho mỗi request).
* Giao tiếp hai chiều: Cho phép client và server gửi data cho nhau bất cứ khi nào trong khi HTTP request chỉ là giao thức một chiều.
* Khả năng mở rộng: Có thể xử lý một lượng lớn kết nối và truyền tải dữ liệu giữa nhiều server và client.

Nhược điểm:

* Độ phức tạp: Cực kỳ phức tạp khi so với HTTP request thông thường. Có thể gây nhiều khó khăn cho lập trình viên khi triển khai và debug sản phẩm
* Vấn đề bảo mật: Dễ bị tấn công cross-site scripting (XSS) và denial-of-service (DoS). Do đó, đòi hỏi người lập trình phải càng cẩn trọng trong việc bảo vệ các kết nối.

Vậy tóm lại, WebSocket là một giao thức cực kỳ hữu ích cho phép giao tiếp thời gian thực, hai chiều giữa client và server phù hợp xây dựng những ứng dụng đòi hỏi độ phản hồi nhanh và chính xác cao. Tiếp theo đây, ta sẽ cùng tìm hiểu về một thư viện phổ biến được xây dựng, cải tiến dựa trên giao thức này đó là Socket.IO và lí do tại sao nhóm chọn sử dụng nó cho dự án của mình.

* + 1. **Socket.io**

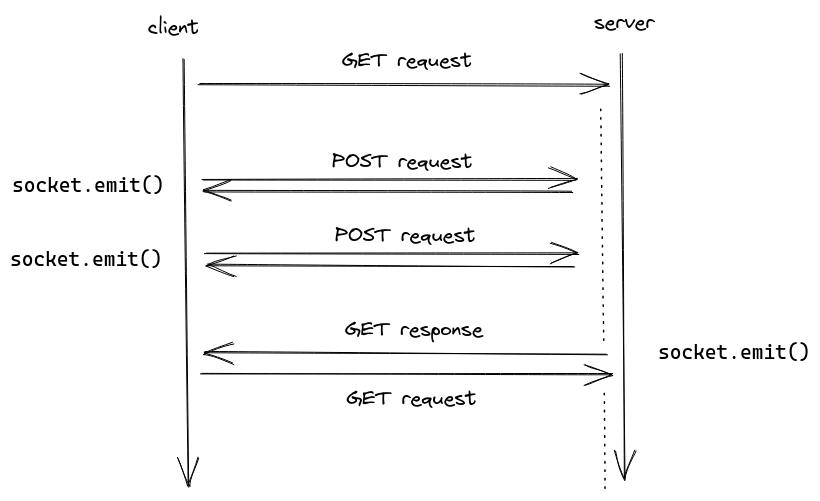
Vì được xây dựng dựa trên WebSocket, nên socket.io kế thừa những đặc điểm, tính năng từ tiền thân của nó là điều hiển nhiên. Được đóng gói thành một thư viện, nó cung cấp phương thức kết nối real-time, hai chiều, dài hạn và được phát triển theo hướng dễ tiếp cận đối với các ứng dụng, nền tảng đòi hỏi sự phản hồi ở thời gian thực như chat applications (messenger, facebook, ...), online gaming (League of Legends, PUBG, ...), online editing tools (excel, word,...), ....

Ngoài ra, socket.io được phát triển theo hướng khắc phục những hạn chế mà giao thức WebSocket tồn đọng. Sau đây là một số lợi ích mà socket.io mang lại vượt trội hơn so với WebSocket:

* Tự động khôi phục kết nối: Nếu kết nối giữa client và server bị ngắt, socket.io sẽ cố gắng khôi phục kết nối lại từ client đến server. Điều này đảm bảo việc kết nối ổn định kể cả khi tình trạng internet bị gián đoạn tạm thời.
* WebSocket fallback: Đây là một cơ chế cho phép socket.io sử dụng các phương thức truyền tải khác như AJAX long-polling hoặc Adobe Flash Socket nếu như trình duyệt không hỗ trợ WebSocket (thường là các trình duyệt cũ). Điều này giúp tăng khả năng tương thích của socket.io
* Custom events và data handling: Cho phép lập trình viên tự định nghĩa các hàm sự kiện và data handling theo ý muốn. Làm tăng khả năng linh hoạt trong xử lý dữ liệu real-time trong khi WebSocket bình thường thì chỉ có thể sử dụng các hàm định nghĩa sẵn.
* Rooms và namespaces: Cơ chế này cho phép gom nhóm người dùng vào các phòng khác nhau và broadcase messages đến từng nhóm cụ thể. Đây là tính năng cực kỳ hữu dụng đối với các ứng dụng cần chat room đặc biệt là trong online games

Sau đây là cách hoạt động của socket.io dựa theo từng tính năng:

* Connection: Về mặt kết nối, thư viện socket.io phía client sẽ kết nối với thư viện socket.io tại phía server thông qua WebSocket, AJAX long-polling hoặc các cơ chế khác.
* Events: Một khi kết nối được thiết lập, phía client có thể emit các sự kiện đến phía server và ngược lại. Dữ liệu được emit có thể thuộc bất kỳ loại nào (ví dụ: string, JSON object, hoặc binary, ...)
* Acknowledgements: Đây là tính năng cho phép người nhận sự kiện có thể gửi lại một phản hồi/ghi nhận (để xác nhận là người nhận đã thực sự nhận được rồi hay chưa).
* Rooms và namespaces: Cho phép người dùng tham gia và nhận các sự kiện tại một phòng cụ thể.
* Disconnection: Tính năng phát hiện người dùng ngắt kết nối với server và tự động cố gắng khôi phục kết nối từ client.



Hình: Minh họa quá trình hoạt động của socket.io. (nguồn: [Hoạt động của socket.io](https://socket.io/docs/v3/how-it-works/))

CHƯƠNG 2 – PHẦN LÝ THUYẾT

1. Tìm hiểu về front-end và back-end.
2. Front-end là gì ?

Là giao diện đồ hoạ của một ứng dụng, kết hợp với giao diện người dùng bên phía client. Nơi mà người sử dụng có thể nhìn, nghe, tương tác,… với ứng dụng như: chữ, màu sắc, hình ảnh, danh mục sản phẩm, âm thanh, các đường liên kết, nút nhấn,…

Front-end là nơi sẽ nhận các dữ liệu được nhập, thao tác,… từ người dùng sau đó sẽ chuyển cho phần back-end xử lý, sau đó sẽ trả về kết quả về lại front-end thông báo cho người dùng biết.

Các yếu tố tạo nên front-end bao gồm:

* Ứng dụng hoặc bố cục trang
* Đồ hoạ
* Âm thanh và video
* Nội dung văn bản
* Giao diện người dùng (nút nhấn, đường liên kết, thanh điều hướng,..)
* Vùng nhập liệu (hộp thoại, checkbox,...)
* Tuỳ chọn chủ đề, tuỳ chỉnh,..
* Luồng hoạt động của người dùng, cách dẫn người dùng tới bước tiếp theo khi dùng ứng dụng.

Ví dụ: Khi truy cập 1 trang web thì đầu tiên trang web sẽ hiển thị giao diện gồm những thứ đã thiết kế lập trình sẵn trước đó như: hình ảnh, văn bản,...các chức năng mà bạn có thể thao tác được để dùng trang web đó. Từ đó bạn có thể sử dụng trang web đó như: nhấn, nhập, xem,… các nội dung cũng như chức năng của trang web đó.

1. Back-end là gì ?

Back-end là phần mà người dùng không thể nhìn thấy được như front-end. Tất cả những yêu cầu, câu lệnh, lựa chọn, thông tin mà người dùng nhập từ front-end sẽ được back-end xử lý và hiển thị lên lại front-end. Thông thường phía back-end sẽ bao gồm: máy chủ, cơ sở dữ liệu, chương trình. Các phần back-end là một bộ “khung xương” cho tất cả những ứng dụng front-end. Các chương trình ở back-end thường là web server, API hoặc dịch vụ lưu trữ hay xử lý các dữ liệu.

Back-end là nơi lưu trữ tất cả các dữ liệu được tạo ra bởi người dùng phía front-end, nơi xử lý các nghiệp vụ. Nếu như back-end không còn làm được những việc trên thì ứng dụng front-end không thể hoạt động được.

Ví dụ: Khi người dùng sử dụng chức năng đăng nhập. Thì người dùng sẽ nhập thông tin bên front-end như là tài khoản, mật khẩu và một số yêu cầu khác liên quan. Sau khi nhập xong thì nhấn nút đăng nhập. Front-end sẽ gửi dữ liệu cho back-end, tại phần back-end sẽ tiếp nhận thông tin đó và xử lý, kiểm tra, kết nối đến server, truy vấn cơ sở dữ liệu,… và sau đó trả về kết quả sau khi xử lý, nếu thông tin đó đúng thì chuyển hướng người dùng vào trong ứng dụng, còn nếu sai thì trả về kết quả thông báo cho người dùng biết là mình nhập sai.

1. Tìm hiểu về NodeJS ?

NodeJS là một môi trường chạy JavaScript được sử dụng để triển khai trên các hệ điều hành máy tính mà chúng ta đang sử dụng, thay vì nhúng vào trình duyệt web thì chúng ta có thể viết code JavaScript để tạo ra các phần mềm, có thể sử dụng các công cụ tài nguyên mà hệ điều hành có thể cung cấp:

* Tạo, lưu trữ, xoá các tệp trên máy
* Tương tác với các thiết bị khác trong cùng mạng hoặc các thiết bị khác qua mạng.
* Điều khiển được các thiết bị phần cứng liên kết với máy tính như: bàn phím, chuột, thiết bị ngoại vi,… hỗ trợ cung cấp giao diện lập trình.

NodeJS không phải là một ngôn ngữ lập trình mà là một môi trường runtime dùng để chạy JavaScript bên ngoài trình duyệt. NodeJS có thể sử dụng ở front-end lẫn back-end. Một số lý do NodeJS phù hợp để sử dụng lẫn cả hai:

* Khả năng tái sử dụng.
* Năng suất và hiệu quả cho nhà phát triển.
* Cộng đồng hỗ trợ lớn.
* Các framework.

1. ExpressJS được dùng cho phần back-end.

ExpressJS được sử dụng cho phần back-end của ứng dụng web vì có những tính năng sau:

* Phát triển nhanh bên phía máy chủ.
* Là phần trung gian: sẽ xử lý các yêu cầu có quyền truy cập vào chu kỳ phản hồi yêu cầu của ứng dụng.
* Điều hướng: Điều hướng các yêu cầu của máy khách đến điểm cuối của URL ứng dụng
* Các khuôn mẫu: Nó cung cấp các công cụ tạo khuôn mẫu để xây dụng nội dung trên các trang web bằng cách tạo ra các mẫu HTML trên máy chủ.
* Gỡ lỗi: Làm cho việc gỡ lỗi dễ dàng hơn vì nó xác định chính xác thành phần có lỗi.

1. ExpressJS là framework

* ExpressJS là một framework linh hoạt được xây dựng trên nền tảng của NodeJS, phần lớn các mã đã được viết sẵn để các lập trình viên làm việc.

Nó cung cấp những tính năng để phát triển ứng dụng web hoặc di động, linh hoạt, tối giản và có nhiều khả năng mở rộng, nhiều các package hỗ trợ. ExpressJS chỉ yêu cầu JavaScript nên các lập trình viên và nhà phát triển sẽ dễ dàng xây dụng các ứng dụng web và API.

CHƯƠNG 3 – HIỆN THỰC HOÁ

3.1 Phân tích hệ thống và chọn framework sử dụng

Về phía client: Nhóm quyết định sử dụng React kết hợp Vite và một số npm module để thúc đẩy việc tạo UI cũng như việc xử lý các trạng thái của từng component trở nên dễ dàng hơn, nhằm mục đích tiết kiệm thời gian và công sức dành cho phần backend.

Về phía backend: Tất nhiên là chỉ sử dụng NodeJS kết hợp npm module. Đây là phần khó nhằn nhất vì cần phải xác định cấu trúc dự án như có bao nhiêu model, các thuộc tính phải tính đến là gì, mối quan hệ giữa các model là như thế nào.

Ta phân tích, đầu tiên model không thể thiếu là User. Tiếp theo, nhóm em quyết định là làm theo hướng hai người chung một phòng chat sẽ chat được với nhau (không làm phần chat group). Do đó, hai model tiếp theo lần lượt là Chat, Message. Mối quan hệ của chúng như sau:

* Chat – User: one to many (cụ thể là một phòng chat, hai user)
* Chat – Message: one to many
* User – Message: one to many

Database sẽ được phát triển theo hướng Referenced documents:

* Chat Model sẽ chứa danh sách định danh của hai người thực hiện chat.
* Message Model sẽ chứa định danh của người gửi.

3.2 Xây dựng API (Postman snapshots)

Vì thời gian có hạn nên nhóm chỉ cung cấp một số hình ảnh chạy kết quả trả về khi thực hiện request tới các API tương ứng:

A screenshot of a computer

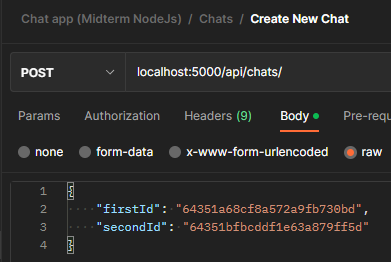
Description automatically generated with medium confidence

Hình: API đăng ký người dùng

A screenshot of a computer

Description automatically generated with medium confidence

Hình: API đăng nhập

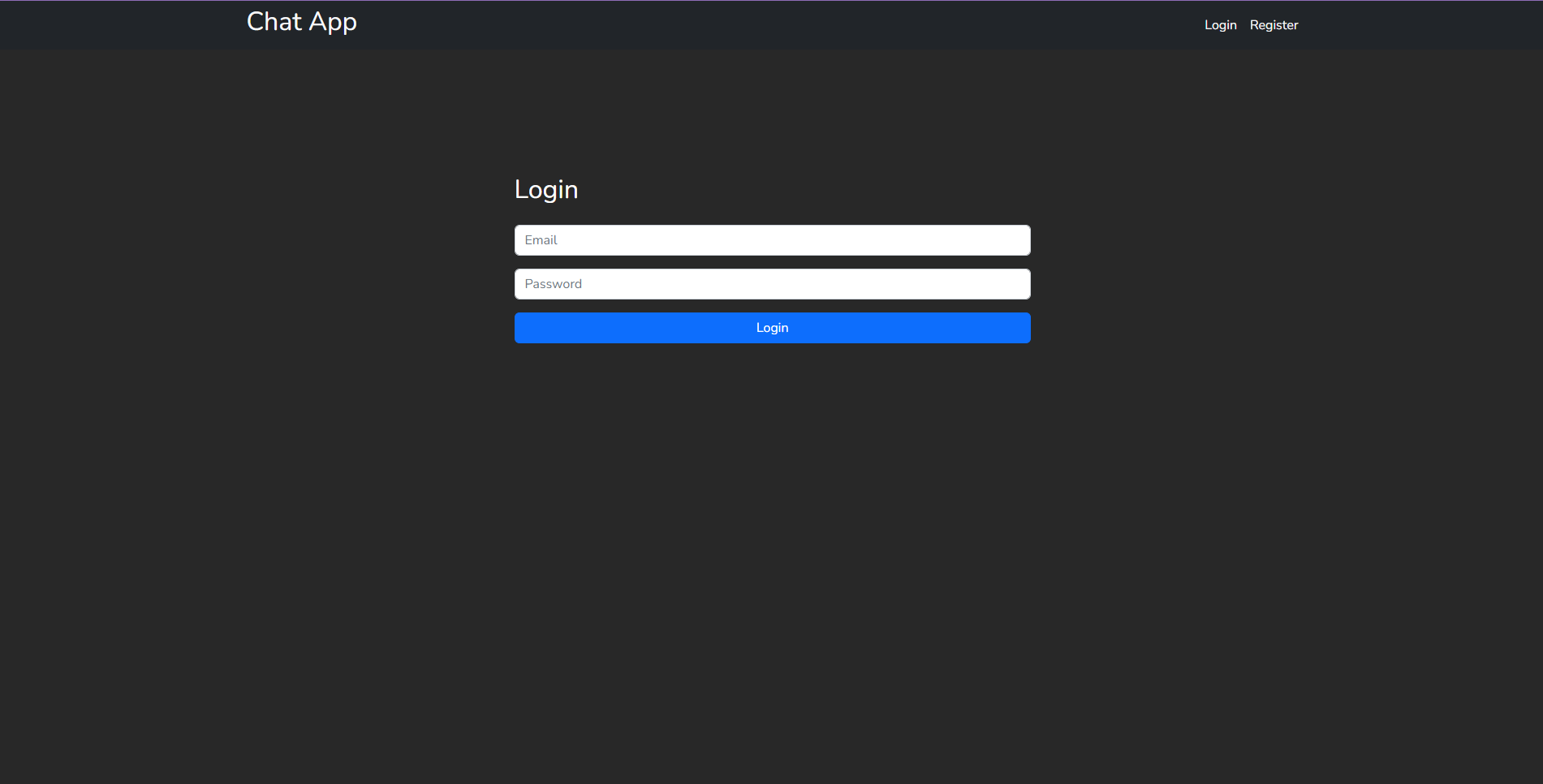


Hình: API tạo chat

Graphical user interface, text, application

Description automatically generated

Hình: API tạo message.

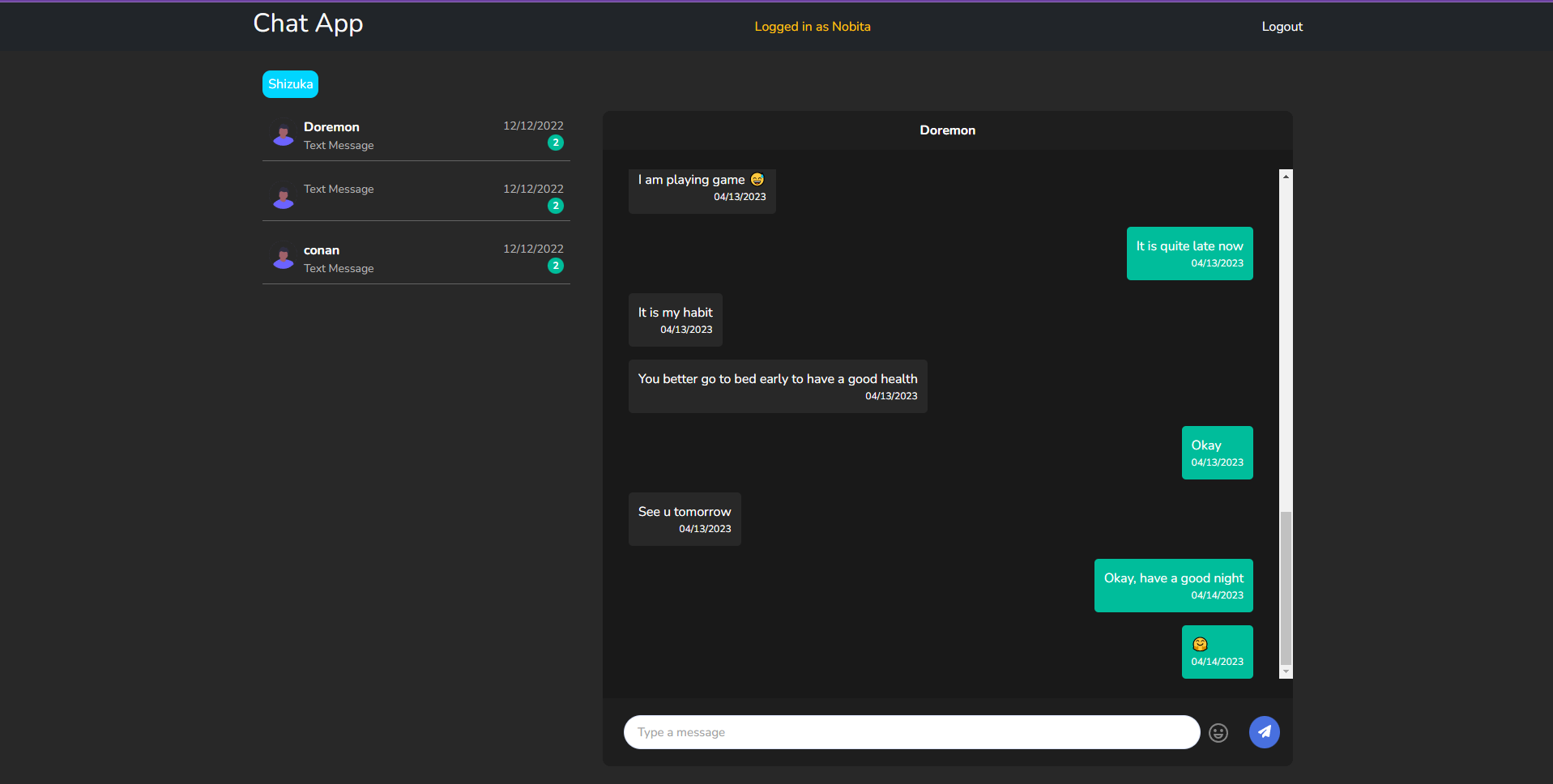
CHƯƠNG 4 – KẾT QUẢ DEMO 

Hình: Login Account Page

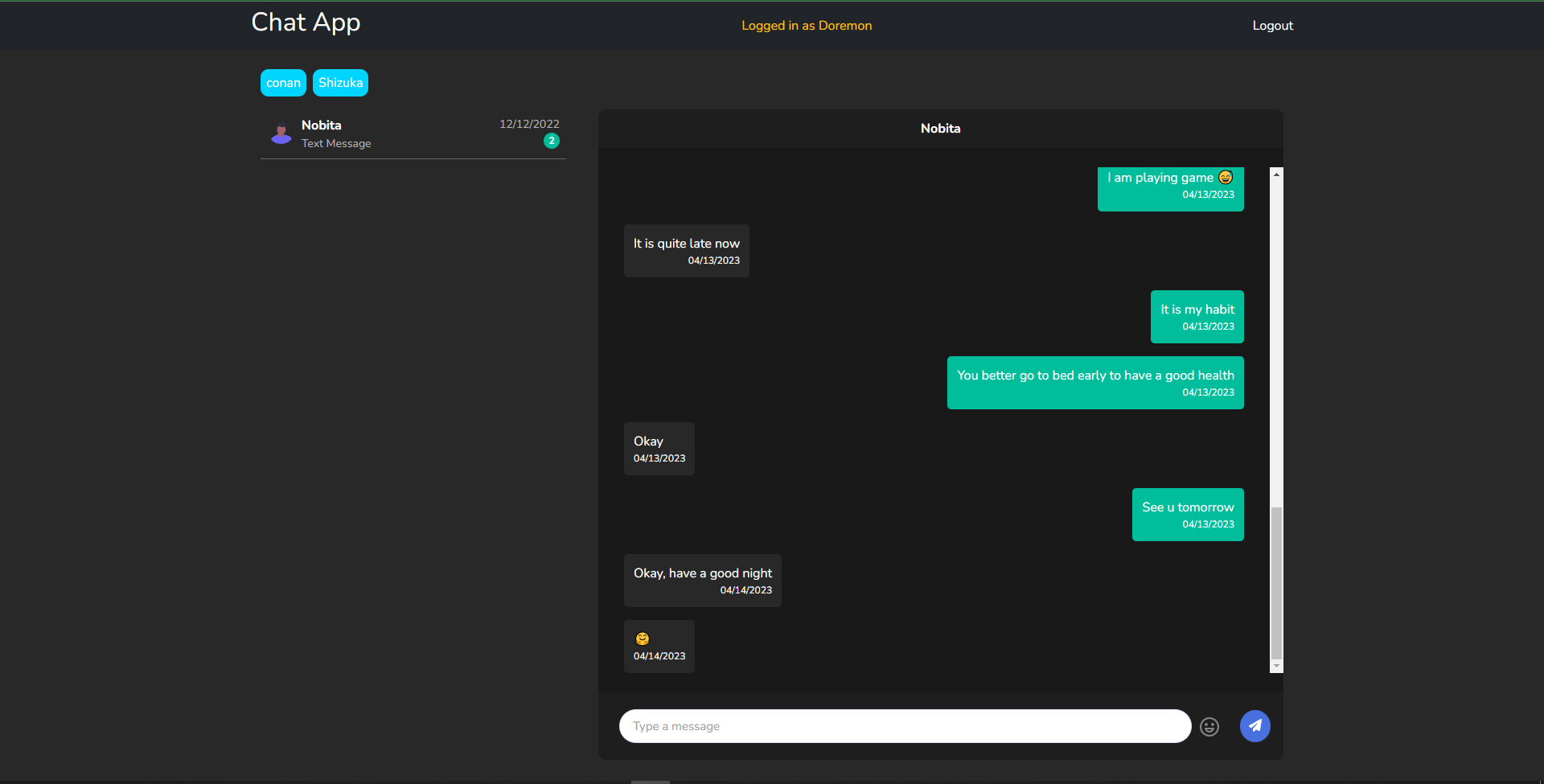
Graphical user interface

Description automatically generated

Hình: Register Account Page



Hình: Chat Page (Sender)



Hình: Chat Page (Receiver)

CHƯƠNG 5 – KẾT LUẬN

Qua bài báo cáo này chúng em lĩnh hội được thêm nhiều kiến thức từ quý thầy cô, Khoa Công Nghệ Thông Tin, cũng như Trường Đại Học Tôn Đức Thắng tốt hơn về môn phát triển ứng dụng web với nodejs. Ứng dụng của nhóm em vẫn còn nhiều những chức năng cần cập nhập phát triển trong tương lai.

.

TÀI LIỆU THAM KHẢO

**Tiếng Việt:**

* 1. [Socket-WebSocket là gì? - Top Dev Blog](https://viblo.asia/p/socket-io-va-chat-app-LzD5dDbW5jY)
  2. [Sơ lược về WebSocket – Nguyễn Hưng](https://vietnix.vn/websocket-la-gi/)
  3. [Xây dựng chat app với socket.io – Thanh Manci – 3/2/2018](https://viblo.asia/p/socket-io-va-chat-app-LzD5dDbW5jY)

**Tiếng Anh:**

* 1. [Socket documentation - Socket.io - Last updated 27/3/2023](https://socket.io/docs/v4/)