**Báo cáo bài tập lớn Công nghệ phần mềm**

**ĐỀ TÀI: WEBSITE HẸN HÒ ONLINE**

**(Web Anonymous)**

**Giảng viên:** TS. Vũ Thị Hồng Nhạn

**Nhóm thực hiện:** Nhóm 5

**Thành viên:**

* Tống Minh Trí: 21020249
* Nguyễn Văn Thuyên: 21020247
* Trần Quang Phúc: 21020659
* Lê Văn Bảo: 21020171
* Nguyễn Văn Trường: 21020250
* Nguyễn Ngọc Tuấn: 21020237

Mục lục

[Mục lục 1](#_Toc2046247620)

[I. Giới thiệu chung 2](#_Toc1560416404)

[a. Thông tin chung 2](#_Toc1172345512)

[c. Giới thiệu 2](#_Toc2090579118)

[i. Mục đích 2](#_Toc1889742660)

[ii. Cơ cấu tổ chức 2](#_Toc1204983550)

[iii. Mô tả 3](#_Toc2062593492)

[1. Đưa ra dữ liệu cần thiết trên trang web khi người dùng truy cập 3](#_Toc1039103580)

[2. Tương tác và tiếp nhận thông tin từ người sử dụng 3](#_Toc829404013)

[3. Lưu trữ và quản lý CSDL 3](#_Toc1951876265)

[4. Phân tích và xử lý yêu cầu của người dùng vào cơ sở dữ liệu người dùng 3](#_Toc2054202150)

[iv. Chức năng của trang web Anonymous 3](#_Toc1779558754)

[1. Yêu cầu chức năng 3](#_Toc735294970)

[2. Yêu cầu phi chức năng 4](#_Toc842910600)

[v. Đối tượng hướng đến 4](#_Toc1534357541)

[vi. Ràng buộc chung 5](#_Toc1968566022)

[II. Tổng quan về các công nghệ, tài nguyên được sử dụng 6](#_Toc526404523)

[a. Về ngôn ngữ 6](#_Toc762229415)

[i. Ngôn ngữ HTML 6](#_Toc1466605847)

[ii. Ngôn ngữ CSS 7](#_Toc1300628190)

[iii. Ngôn ngữ lập trình JavaScript 8](#_Toc604271524)

[b. Về platfrom NodeJS và các framework, thư viện sử dụng 9](#_Toc73215514)

[i. Platfrom NodeJS 9](#_Toc1751953806)

[ii. Các framework và thư viện sử dụng trong khi lập trình 9](#_Toc975893996)

[1. ReactJS 9](#_Toc184389927)

[2. Socket.IO 10](#_Toc1720464002)

[c. Về Cơ sở dữ liệu 10](#_Toc693855961)

[i. Cơ sở dữ liệu MongoDB 10](#_Toc114168206)

[1. Tìm hiểu về NoSQL 10](#_Toc549545793)

[2. Tìm hiểu về MongoDB 10](#_Toc1554005663)

[ii. Cơ sở dữ liệu Mongoose 11](#_Toc999893735)

[1. Khái niệm 12](#_Toc181428815)

[2. Chức năng 12](#_Toc8839221)

[d. Về AI tương tác với người dùng 12](#_Toc63672340)

[III. Quá trình phát triển phần mềm 14](#_Toc1061752965)

[a. Phân tích và đặc tả 14](#_Toc1882956525)

[i. Yêu cầu chi tiết cho phần mềm 14](#_Toc1572073755)

[ii. Chi tiết yêu cầu 15](#_Toc1392119493)

[1. Website 15](#_Toc1618370581)

[2. Hiệu năng 17](#_Toc226674211)

[iii. Yêu cầu về giao diện 18](#_Toc1183137058)

[1. Trên trang web 18](#_Toc1843045225)

[2. Trên Cơ sở dữ liệu: 23](#_Toc2062795043)

[iv. Ràng buộc trong quy trình phát triển và sử dụng phần mềm 24](#_Toc844734247)

[b. Thiết kế phần mềm 24](#_Toc1697036556)

[i. Thiết kế hệ thống 24](#_Toc1443552889)

[1. Kiến trúc hệ thống: 24](#_Toc719622597)

[- Gồm danh sách các use-case, Lược đồ use-case, Đặc tả. 24](#_Toc549197533)

[Danh sách các use-case 24](#_Toc45830901)

[Lược đồ use-case 24](#_Toc1821055833)

[Đặc tả 25](#_Toc875429213)

[2. Các biểu đồ 30](#_Toc400142054)

[Biểu đồ trạng thái 30](#_Toc1125485258)

[Biểu đồ tuần tự 31](#_Toc310091475)

[ii. Thiết kế cơ sở dữ liệu 32](#_Toc166317995)

[1. Mô hình chung: 32](#_Toc486855314)

[2. Các bảng dữ liệu: 32](#_Toc2038452714)

[c. Cài đặt phần mềm 33](#_Toc116943676)

[d. Kiểm thử phần mềm đối với trang web 34](#_Toc880822940)

[IV. Kết luận 35](#_Toc514210237)

[a. Ưu điểm 35](#_Toc393350558)

[b. Nhược điểm 35](#_Toc825700621)

[c. Hướng nâng cấp phần mềm 36](#_Toc276528545)

1. **Giới thiệu chung**
   1. Thông tin chung
      * Link github: <https://github.com/jeikai/anonymous.git>
   2. Vấnđề

Khoảng thời gian công nghệ chưa phát triển, đa phần chúng ta đều tìm hiểu người mình thích hay tâm sự với người khác qua các buổi gặp mặt và trò chuyện trực tiếp. Tuy nhiên không phải ai cũng có đủ điều kiện, sự tự tin và can đảm để điều đó. Với thời đại công nghệ 4.0 ngày nay, mọi người mong muốn có một giải pháp có thể giúp họ vừa có thể tâm sự với người khác, tìm hiểu người họ thích trức tuyến lại vừa có thể tiết kiệm thời gian, công sức gặp gỡ và đặc biệt phù hợp với những người hướng nội, ngại giao tiếp

* 1. Giới thiệu
     1. *Mục đích*
        + Đây là tài liệu cung cấp thông tin chi tiết toàn bộ quá trình phát triển Web Anonymous gồm các giai đoạn: Thiết kế, Lập trình, Kiểm thử, Nâng Cấp, Bảo trì, Nâng cấp
     2. *Cơ cấu tổ chức*

Hệ thống bao gồm: Trang web Anonymous

* + - * Trang web Anonymous: Truy cập thông qua Internet, trang web giúp người dùng có thể tìm kiếm và kết nối với những người khác. Hoặc nếu người dùng muốn cần một nơi để chia sẻ về bản thân hay hỏi đáp về một vấn đề nào đó thì trang web đã tích hợp công cụ tương tự chat GPT để đáp ứng cho yêu cầu này
    1. *Mô tả*

Hệ thống gồm 4 phần:

* + - * Đưa ra dữ liệu cần thiết trên trang web khi người dùng truy cập
      * Tương tác và tiếp nhận thông tin từ người sử dụng
      * Lưu trữ và quản lý CSDL
      * Phân tích và xử lý yêu cầu của người dùng vào cơ sở dữ liệu người dùng
      1. Đưa ra dữ liệu cần thiết trên trang web khi người dùng truy cập
      * Khi đăng nhập vào tài khoản sẽ hiện ra dữ liệu được cập nhật tùy theo người dùng như hiện thị những tài khoản theo dõi hay tin nhắn với những người dùng khác
      1. Tương tác và tiếp nhận thông tin từ người sử dụng
      * Có thể đăng ký và đăng nhập tài khoản
      * Có thể truy cập các hoạt động trên trang web để hiển thị thông tin chi tiết
      1. Lưu trữ và quản lý CSDL
      * CSDL xử lý dữ liệu đầu vào và lưu trữ theo định dạng
      * CSDL có khả năng truy vấn và trả về kết quả theo mong muốn của người sử dụng
      * CSDL có thể đổi mới dữ liệu
      1. Phân tích và xử lý yêu cầu của người dùng vào cơ sở dữ liệu người dùng
      * Tiếp nhận thông tin từ CSDL và điều chỉnh theo yêu cầu
      * Tìm kiếm và chọn lọc thông tin trong CSDL
    1. *Chức năng của trang web Anonymous*
       1. Yêu cầu chức năng

1. Chức năng quản lý và cài đặt:

Các tính năng để quản lý tài khoản bao gồm đăng ký, đăng nhập, đăng xuất, quản lý thông tin cá nhân và mật khẩu

Tính năng cài đặt bảo mật

1. Chức năng cập nhật hồ sơ:

Người dùng có thể cập nhật thông tin cá nhân như tên, số điện thoại,…

Ngoải ra người dùng có thể đăng, gỡ, chỉnh sửa chế độ hiển thị của ảnh và bài viết trên trang cá nhân

1. Chức năng trò chuyện

* Người dùng có thể nhắn tin trò chuyện với nhau trên trang web
* Người dùng có thể hỏi đáp hay chia sẻ với trợ lý ảo của hệ thống

1. Chức năng tìm kiếm người dùng
   * Người dùng có thể tìm kiếm và lọc người dùng khác dựa trên tên
2. Chức năng mở rộng khác
   * Dark mode: Khi mỏi mắt, người dùng có thể đổi theme của màn hình sang chế độ Dark mode để có trải nghiệm tốt hơn
   * Chuyển tiền: Người dùng sẽ chuyển tiền với một người dùng khác thông qua ví điện tử
     + 1. Yêu cầu phi chức năng
          - Bảo mật và tin cậy: phần mềm có tính bảo mật cao, đảm bảo thông tin cá nhân của người dùng được bảo vệ an toàn, tránh nguy cơ đánh cắp thông tin, truy cập trái phép.
          - Dễ sử dụng: Người dùng dễ dàng thao tác, sử dụng. Không tốn thời gian để người dùng tìm hiểu và sử dụng. Các chức năng liên quan sắp xếp gần nhau, thuận tiện cho truy cập
          - Giao diện thân thiện: phần mềm có giao diện đơn giản, thân thiện nhằm tạo sự thoải mái, thu hút với người dùng
          - Tính ổn định: Hệ thống ít lỗi, không có lỗi nghiêm trọng. Nếu xảy ra lỗi hệ thống vẫn hoạt động chứ không bị ngừng, đảm bảo số lượng người truy cập lớn mà không bị chậm, quá tải
     1. *Đối tượng hướng đến*
        + Dành cho người từ 18 tuổi trở lên, chủ yếu là từ 18-35 tuổi đang còn độc thân ở mọi nơi muốn tìm kiếm mối quan hệ tình cảm hoặc tìm người tâm sự.
     2. *Ràng buộc chung*
        + Khi đăng ký người dùng cần nhập đủ thông tin cần thiết và tuân thủ theo quy tắc mới đăng ký thành công
3. **Tổng quan về các công nghệ, tài nguyên được sử dụng**
   1. Về ngôn ngữ
      1. *Ngôn ngữ HTML*
         * HTML (HyperText Markup Language) – Ngôn ngữ đánh dấu siêu văn bản được sử dụng để tạo các tài liệu có thể truy cập trên mạng. Tài liệu HTML được tạo nhờ dùng các thẻ và các phần tử của HTML. File được lưu trên máy chủ dịch vụ web với phần mở rộng “.htm” hoặc “.html”. Các trình duyệt sẽ đọc tập tin HTML và hiển thị chúng dưới dạng trang web. Các thẻ HTML sẽ được ẩn đi, chỉ hiển thị nội dung văn bản và các đối tượng khác: hình ảnh, media. Với các trình duyệt khác nhau đều hiển thị một tập HTML với 1 kết quả nhất định. Các trang HTML được gửi đi qua mạng internet theo giao thức HTTP. HTML không những cho phép nhúng thêm các đối tượng hình ảnh, âm thanh mà còn cho phép nhúng các kịch bản vào trong đó như các ngôn ngữ kịch bản như Javascript để tạo hiệu ứng động cho trang web. Để trình bày trang web hiệu quả hơn thì HTML cho phép sử dụng kết hợp với CSS. HTML không những cho phép nhúng thêm các đối tượng hình ảnh, âm thanh mà còn cho phép nhúng các kịch bản vào trong đó như các ngôn ngữ kịch bản như Javascript để tạo hiệu ứng động cho trang web.
         * HTML là một chuẩn ngôn ngữ internet được tạo ra và phát triển bởi tổ chức World Wide Web Consortium còn được viết tắc là W3C. Trước đó thì HTML xuất bản theo chuẩn của RFC. HTML được tương thích với mọi hệ điều hành cùng các trình duyệt của nó. Khả năng dễ học, dễ viết là một ưu điểm của HTML không những vậy việc soạn thảo đòi hỏi hết sức đơn giản và thông thường sử dụng notepad là đã đủ. Hiện nay, phiên bản mới nhất của HTML là HTML5 với nhiều tính năng ưu việt so với các phiên bản cũ HTML cải tiến khá nhiều đặc biệt hỗ trợ mạnh mẽ các phần tử multimedia mà không cần các plugin. Một tập tin HTML bao gồm trong đó là các đoạn văn bản HTML, được tạo lên bởi các thẻ HTML. HTML5 nói chung mạnh mẽ hơn nhiều không chỉ về tốc độ và độ thích ứng cao mà chính là khả năng hỗ trợ API (Application Programming Interface - giao diện lập trình ứng dụng) và DOM (Document Object Model – các đối tượng thao tác văn bản).
      2. *Ngôn ngữ CSS*
         * CSS (Cascading Style Sheets) là một ngôn ngữ quy định cách trình bày cho các tài liệu viết bằng HTML, XHTML, XML, SVG, hay UML…CSS quy định cách hiển thị của các thẻ HTML bằng cách quy định các thuộc tính của các thẻ đó (font chữ, kích thước, màu sắc...). Các đặc điểm kỹ thuật của CSS được duy trì bởi tổ chức W3C.
         * CSS có cấu trúc đơn giản và sử dụng các từ tiếng anh để đặt tên cho các thuộc tính. CSS khi sử dụng có thể viết trực tiếp xen lẫn vào mã HTML hoặc tham chiếu từ một file css riêng biệt. Và hiện nay CSS thường được viết riêng thành một tập tin với mở rộng là “.css”. Chính vì vậy mà các trang web có sử dụng CSS thì mã HTML sẽ trở nên ngắn gọn hơn. Ngoài ra có thể sử dụng 1 tập tin CSS đó cho nhiều website tiết kiệm rất nhiều thời gian và công sức. Một đặc điểm quan trọng đó là tính kế thừa của CSS do đó sẽ giảm được số lượng dòng code mà vẫn đạt được yêu cầu.
         * Tuy nhiên, đối với CSS thì các trình duyệt hiểu theo kiểu riêng của nó. Do vậy, việc trình bày 1 nội dung trên các trình duyệt khác nhau là không thống nhất. CSS cung cấp hàng trăm thuộc tính trình bày dành cho các đối tượng với sự sáng tạo cao trong kết hợp các thuộc tính giúp mang lại hiệu quả.
      3. *Ngôn ngữ lập trình JavaScript*
         * JavaScript là 1 ngôn ngữ dạng script thường được sử dụng cho việc lập trình web ở phía client, nó tuân theo chuẩn ECMAScript. Là 1 ngôn ngữ linh động, cú pháp dễ sử dụng như các ngôn ngữ khác và dễ dàng lập trình. JavaScript không hề liên quan tới ngôn ngữ lập trình java, được hầu hết các trình duyệt ngày nay hỗ trợ.
         * Với Javascript, ứng dụng web của bạn sẽ trở nên vô cùng sinh động, mang tính trực quan và tương tác cao. JavaScript theo phiên bản hiện hành là một ngôn ngữ lập trình kịch bản dựa trên đối tượng được phát triển từ các ý niệm nguyên mẫu. Ngôn ngữ này được dùng rộng rãi cho các trang web, nhưng cũng được dùng để tạo khả năng viết script sử dụng các đối tượng nằm sẵn trong các ứng dụng. Giống Java, JavaScript có cú pháp tương tự C, nhưng nó gần với Self hơn Java. “JavaScript” là phần mở rộng thường được dùng cho tập tin mã nguồn JavaScript.
      4. Ngôn ngữ lập trình Solidity
         * Solidity là một ngôn ngữ lập trình hướng đối tượng được sử dụng để phát triển các hợp đồng thông minh trên nền tảng Ethereum. Nó được phát triển bởi Ethereum Foundation và được thiết kế để đơn giản hóa quá trình phát triển hợp đồng thông minh.
         * Solidity cung cấp các tính năng như kế thừa, đa hình, kiểu dữ liệu phức tạp và các phương thức để xử lý các giao dịch trên blockchain Ethereum. Nó cũng cung cấp một tập hợp các thư viện tiêu chuẩn để giúp lập trình viên phát triển các hợp đồng thông minh một cách dễ dàng và hiệu quả hơn.
         * Các hợp đồng thông minh được viết bằng Solidity có thể được triển khai trên blockchain Ethereum và được thực thi bởi các nút trên mạng. Khi được triển khai, các hợp đồng thông minh này có thể tự động thực hiện các giao dịch và tương tác với các ứng dụng khác trên blockchain Ethereum.
   2. Về platfrom NodeJS và các framework, thư viện sử dụng
      1. *Platfrom NodeJS*

* NodeJS là một nền tảng (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng
* NodeJS được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ
* Phần Core bên dưới của NodeJS được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.
* NodeJS tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực. Đáp ứng tốt thời gian thực và chạy đa nền tảng, đa thiết bị
* NodeJS áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể
* Nodejs chạy đa nền tảng phía Server, sử dụng kiến trúc hướng sự kiện Event-driven, cơ chế non-blocking I/O làm cho nó nhẹ và hiệu quả.
* *ReactJS* là một thư viện viết bằng javascript, dùng để xây dựng giao diện người dùng (UI)
* React sử dụng JavaScript Syntax Extension (JSX) cho phép người dùng viết HTML động
* Nó sử dụng Virtual Document Object Model (VDOM), cho phép nhà phát triển cập nhật các phần cụ thể của trang web mà không cần phải load lại trang
  + 1. *Các framework và thư viện sử dụng trong khi lập trình*
       1. *ReactJS*
* *ReactJS* là một thư viện viết bằng javascript, dùng để xây dựng giao diện người dùng (UI)
* React sử dụng JavaScript Syntax Extension (JSX) cho phép người dùng viết HTML động
* Nó sử dụng Virtual Document Object Model (VDOM), cho phép nhà phát triển cập nhật các phần cụ thể của trang web mà không cần phải load lại trang
  + - 1. *Socket.IO*
* *Socket.IO* là một thư viện Javascript cho phép giao tiếp hai chiều và độ trễ thấp giữa máy chủ và máy khách
* *Socket.IO* có khả năng tạo kết nối thuận lợi với máy chủ server, dễ dàng khôi phục kết nối giữa client-server khi gặp gián đoạn
* *Socket.IO* cho phép người dùng tạo mỗi liên kết giữa từng module riêng lẻ để tạo ra nhiều kênh riêng biệt
  1. Về Cơ sở dữ liệu
     1. *Cơ sở dữ liệu MongoDB*
        1. Tìm hiểu về NoSQL
        + NoSQL là 1 dạng CSDL mã nguồn mở và được viết tắt bởi: None-Relational SQL hay có nơi thường gọi là Not-Only SQL
        + NoSQL được phát triển trên Javascript Framework với kiểu dữ liệu là JAVASCRIPTON và dạng dữ liệu theo kiểu key và value
        + NoSQL ra đời như là 1 mảnh vá cho những khuyết điểm và thiếu xót cũng như hạn chế của mô hình dữ liệu quan hệ *RDBMS (Relational Database Management System - Hệ quản trị cơ sở dữ liệu quan hệ)* về tốc độ, tính năng, khả năng mở rộng...
        + Với NoSQL bạn có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc...
        + NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suất nhanh và khả năng mở rộng
        1. Tìm hiểu về MongoDB
        + MongoDB là một hệ quản trị cơ sở dữ liệu mã nguồn mở, là CSDL thuộc NoSQL và được hàng triệu người sử dụng
        + MongoDB là một database hướng tài liệu (document), các dữ liệu được lưu trữ trong document kiểu JAVASCRIPTON thay vì dạng bảng như CSDL quan hệ nên truy vấn sẽ rất nhanh
        + Với CSDL quan hệ chúng ta có khái niệm bảng, các cơ sở dữ liệu quan hệ (như MongoDB hay SQL Server...) sử dụng các bảng để lưu dữ liệu thì với MongoDB chúng ta sẽ dùng khái niệm là collection thay vì bảng
        + So với *RDBMS* thì trong MongoDB collection ứng với table, còn document sẽ ứng với row, MongoDB sẽ dùng các document thay cho row trong *RDBMS*
        + Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định
        + Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB
        1. Các thành phần trong MongoDB
        + Kết nối đến cơ sở dữ liệu MongoDB: tìm hiểu làm thế nào để sử dụng JavaScript để mở và đóng một kết nối cơ sở dữ liệu MongoDB
        + Tạo cơ sở dữ liệu sử dụng JavaScript: làm thế nào để tạo ra cơ sở dữ liệu và bảng MongoDB bằng cách sử dụng JavaScript
        + Xóa cơ sở dữ liệu MongoDB Sử dụng JavaScript: làm thế nào để xóa cơ sở dữ liệu và bảng MongoDB bằng cách sử dụng JavaScript
        + Chèn dữ liệu Cơ sở dữ liệu MongoDB: khi đã có cơ sở dữ liệu và muốn chèn dữ liệu vào CSDL thông qua JavaScript
        + Lấy dữ liệu từ cơ sở dữ liệu MongoDB: tìm hiểu làm thế nào để lấy dữ liệu từ cơ sở dữ liệu MongoDB bằng cách sử dụng JavaScript
        + Sử dụng Paging qua JAVASCRIPT: điều này giải thích làm thế nào để hiển thị kết quả truy vấn của bạn thành nhiều trang và làm thế nào để tạo ra các liên kết điều hướng.
        + Đang cập nhật dữ liệu vào cơ sở dữ liệu MongoDB: phần này giải thích làm thế nào để cập nhật hồ sơ hiện có vào cơ sở dữ liệu MongoDB bằng cách sử dụng JavaScript
        + Xóa dữ liệu từ cơ sở dữ liệu MongoDB: làm thế nào để xóa dữ liệu từ cơ sở dữ liệu MongoDB bằng JavaScript
        + Sử dụng JavaScript để sao lưu cơ sở dữ liệu MongoDB: tìm hiểu những cách khác nhau để sao lưu cơ sở dữ liệu
     2. *Cơ sở dữ liệu Mongoose*
        1. Khái niệm

Mongoose là một thư viện mô hình hóa đối tượng (Object Data Model - ODM) cho MongoDB và Node.js. Nó quản lý mối quan hệ giữa dữ liệu, cung cấp sự xác nhận giản đồ và được sử dụng để dịch giữa các đối tượng trong mã và biểu diễn các đối tượng trong MongoDB

* + - 1. Chức năng

Mongoose là một thư viện JavaScript cho phép bạn định nghĩa các schema với dữ liệu được định kiểu rõ ràng. Khi một schema được định nghĩa, Mongoose cho phép bạn tạo một Model dựa trên một schema cụ thể. Model của Mongoose sau đó được ánh xạ tới một MongoDB document thông qua định nghĩa schema của Model.

Khi bạn đã xác định các schema và model của mình, Mongoose chứa nhiều hàm khác nhau cho phép bạn xác thực tính hợp lệ, lưu, xóa và truy vấn dữ liệu của bạn bằng các hàm MongoDB phổ biến.

* 1. Về AI tương tác với người dùng
     + - Là chat AI GPT
       - Chat GPT có tên đầy đủ là Chat Generative Pre-training Transformer, đây là công cụ hỏi đáp tự động do công ty công nghệ OpenAI phát triển thông qua tích hợp trí tuệ nhân tạo (AI)
       - Bên cạnh đó, Chat AI GPT còn có khả năng tự học hỏi, nghĩa nếu người dùng phản hồi rằng nội dung Chat GPT trả lời là sai thì nó tự sửa chữa ngay lỗ hổng này để đưa ra câu trả lời đúng cho các lần giao tiếp sau đó.
       - Nguyên lý hoạt động của Chat GPT là dựa trên công cụ GPT-3 – sở hữu một hệ thống văn bản bao gồm 8 triệu tài liệu và hơn 10 tỷ từ. Nhờ “kho” văn bản này, Chat GPT sẽ xử lý một lượng lớn văn bản và thực hiện các thao tác xử lý, sắp xếp ngôn ngữ tự nhiên. Từ đó tạo ra văn bản mới mạch lạc và trôi chảy.
       - Vì vậy nó sẽ có những câu trả lời thông minh và có thể đưa ra những câu trả lời giải đáp những thắc mắc đối với người dùng
  2. Ứng dụng công nghệ Smart contract

1. *Metamask*

* Metamask là một ví tiền điện tử dựa trên trình duyệt web, cho phép người dùng lưu trữ và quản lý các tài sản kỹ thuật số như Ethereum và các token dựa trên chuẩn ERC-20. Nó được sử dụng để tương tác với các ứng dụng phi tập trung (dApps) trên blockchain Ethereum, cung cấp cho người dùng một giao diện đơn giản và trực quan để thực hiện các giao dịch và chức năng trong hệ thống blockchain.
* Metamask cho phép người dùng tạo ví mới hoặc nhập các ví đã có để quản lý các tài sản kỹ thuật số của họ. Nó cũng cung cấp cho người dùng một khối thông tin chi tiết về các tài sản của họ, ví dụ như số dư, lịch sử giao dịch và phí gas hiện tại. Metamask cũng cung cấp tính năng để ký và gửi giao dịch một cách an toàn và đáng tin cậy.

1. *Hardhat*

* Hardhat là một bộ công cụ phát triển hợp đồng thông minh mã nguồn mở cho Ethereum, được thiết kế để hỗ trợ quá trình phát triển và kiểm thử các ứng dụng trên blockchain Ethereum. Hardhat cung cấp một tập hợp các công cụ để lập trình, biên dịch, triển khai, kiểm thử và gỡ lỗi các hợp đồng thông minh.
* Hardhat cung cấp một môi trường phát triển ổn định và linh hoạt, cho phép lập trình viên phát triển hợp đồng thông minh bằng các ngôn ngữ như Solidity, Vyper và Serpent. Nó cung cấp một số tính năng hữu ích như kiểm tra tương tác hợp đồng thông minh với các ứng dụng khác, mô phỏng mạng lưới Ethereum và đưa ra các báo cáo về lỗi và cải thiện hiệu suất.
* Hardhat cũng hỗ trợ tích hợp với các công cụ khác như Truffle, Remix và Ganache, cung cấp cho lập trình viên một môi trường phát triển đầy đủ chức năng để phát triển các ứng dụng và hợp đồng thông minh trên nền tảng Ethereum.
  1. Về kiến trúc phầm mềm (Software Architecture)
* Để sử dụng cho Backend (Nodejs) và Frontend (ReactJS), kiến trúc được áp dụng trong sản phầm là MVVM (Model - View - ViewModel).
* Kiến trúc MVVM được chia thành ba phần:
  + Model: Model đại diện cho dữ liệu và logic kinh doanh của ứng dụng. Nó xác định cấu trúc dữ liệu và cung cấp các phương thức để tương tác với dữ liệu, chẳng hạn như truy xuất, tạo, cập nhật và xóa dữ liệu. Model cũng bao gồm các quy tắc xác nhận và logic kinh doanh quản lý hành vi của dữ liệu.
  + View: View đại diện cho giao diện người dùng của ứng dụng. Nó có trách nhiệm hiển thị dữ liệu cho người dùng và thu thập đầu vào từ người dùng. View được thiết kế để đơn giản nhất có thể, nghĩa là nó không nên chứa bất kỳ logic nào và tập trung chỉ vào hiển thị dữ liệu.
  + ViewModel: ViewModel hoạt động như một trung gian giữa view và model. Nó chứa logic hiển thị và hiển thị dữ liệu từ model theo cách dễ dàng cho view tiêu thụ. ViewModel chịu trách nhiệm cập nhật view khi dữ liệu cơ sở thay đổi và cập nhật model khi người dùng tương tác với view.
* MVVM là một mô hình kiến trúc phổ biến trong phát triển ứng dụng hiện đại, đặc biệt là trong phát triển ứng dụng di động và web. Bằng cách chia ứng dụng thành ba phần này, MVVM cho phép mã dễ bảo trì hơn và có thể kiểm thử, cũng như kiến trúc linh hoạt và có khả năng mở rộng hơn.



1. **Quá trình phát triển phần mềm**
2. Phân tích và đặc tả
   * 1. *Yêu cầu chi tiết cho phần mềm*

|  |  |  |
| --- | --- | --- |
|  | ID | Mục đích |
| Yêu cầu đối với website | W01 | Đăng ký |
| W02 | Đăng nhập |
| W03 | Hiện thị các tài khoản theo dõi |
| W04 | Nhắn tin |
| W05 | Tìm kiếm người dùng |
| W06 | Chuyển tiền |
| W07 | Hiện thị lịch sử giao dịch |
| Yêu cầu hiệu năng | F01 | Bảo mật và tin cậy |
| F02 | Dễ sử dụng |
| F03 | Giao diện thân thiện |
| F04 | Tính ổn định |

* + 1. *Chi tiết yêu cầu*
       1. Website

|  |  |
| --- | --- |
| ID | W01 |
| Tiêu đề | Đăng ký |
| Mô tả | Đăng ký tạo tài khoản |
| Hành động | Người dùng tạo tài khoản |
| Yêu cầu | Điền đủ thông tin, tên đăng nhập phải là duy nhất và các thông tin khác phải tuân thủ theo các quy tắc của hệ thống |

|  |  |
| --- | --- |
| ID | W02 |
| Tiêu đề | Đăng nhập |
| Mô tả | Đăng nhập vào tài khoản đã đăng ký |
| Hành động | Người dùng đăng nhập vào tài khoản |
| Yêu cầu | Điền đúng và đủ thông tin như đã đăng ký |

|  |  |
| --- | --- |
| ID | W03 |
| Tiêu đề | Hiện thị các tài khoản theo dõi |
| Mô tả | Hiện thị thông tin các tìa khoản mà người dùng đã theo dõi |
| Hành động | Người dùng bấm vào trang chủ |
| Yêu cầu | Hiện thị đầy đủ dữ liệu |

|  |  |
| --- | --- |
| ID | W04 |
| Tiêu đề | Nhắn tin |
| Mô tả | Người dùng ngắn tin với người dùng khác hoặc với trợ lý ảo của web |
| Hành động | Người dùng gõ nội dung tin nhắn rồi gửi |
| Yêu cầu | Tin nhắn phải có ký tự và thiết bị đang kết nối với internet |

|  |  |
| --- | --- |
| ID | W05 |
| Tiêu đề | Tìm kiếm người dùng |
| Mô tả | Hiện thị những người dùng khác dựa theo tên tìm kiếm |
| Hành động | Người dùng nhập nội dung tên trên khung tìm kiếm |
| Yêu cầu | Nội dung thông tin phải có ký tự và thiết bị của người dùng đang kết nối với internet |

|  |  |
| --- | --- |
| ID | W06 |
| Tiêu đề | Chuyển tiền |
| Mô tả | Người dùng chuyển tiền với người dùng khác thông qua ví điện tử |
| Hành động | Đầu tiên người dùng cài đặt ví điện tử nếu chưa có. Tiếp theo người dùng sẽ kết nối với ví điện tử trên trang web thông qua nút connect wallet sau đó nhập thông tin cần thiết rồi bấm xác nhận. Ví điện tử sẽ hiện lên để xác nhận hóa đơn. |
| Yêu cầu | Phải cài đặt ví điện tử |

|  |  |
| --- | --- |
| ID | W07 |
| Tiêu đề | Hiện thị lịch sử giao dịch |
| Mô tả | Người dùng xem thông tin về các giao dịch đã thực hiện |
| Hành động | Người dùng bấm vào nút lịch sử giao dịch trên web |
| Yêu cầu | Phải nhập vào một tài khoản người dùng |

* + - 1. Hiệu năng

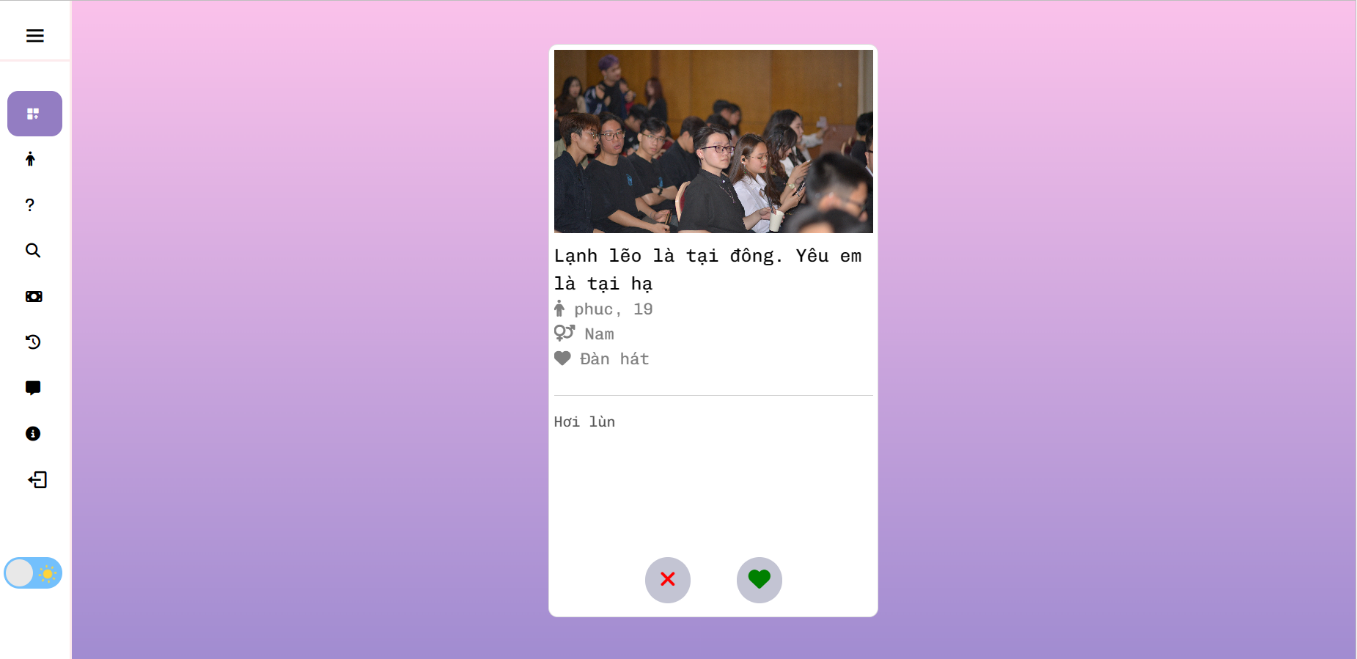
|  |  |
| --- | --- |
| ID | F01 |
| Tiêu đề | Bảo mật và tin cậy |
| Mô tả | Bảo mật dữ liệu người dùng, tránh bị lấy cắp, để lộ |

|  |  |
| --- | --- |
| ID | F02 |
| Tiêu đề | Dễ sử dụng |
| Mô tả | Người dùng dễ dàng thao tác, sử dụng  Không tốn thời gian để người dùng tìm hiểu và sử dụng  Các chức năng liên quan sắp xếp gần nhau, thuận tiện cho truy cập |

|  |  |
| --- | --- |
| ID | F03 |
| Tiêu đề | Giao diện thân thiện |
| Mô tả | Màu sắc, sắp xếp phối hợp hài hòa |

|  |  |
| --- | --- |
| ID | F04 |
| Tiêu đề | Tính ổn định |
| Mô tả | Hệ thống ít lỗi, không có lỗi nghiêm trọng  Nếu xảy ra lỗi hệ thống vẫn hoạt động chứ không bị ngừng, đảm bảo số lượng người truy cập lớn mà không bị chậm, quá tải |

* + 1. *Yêu cầu về giao diện*
       1. Trên trang web
       - Trang chủ:



* + - * Đăng ký:



* + - * Đăng ký thành công:



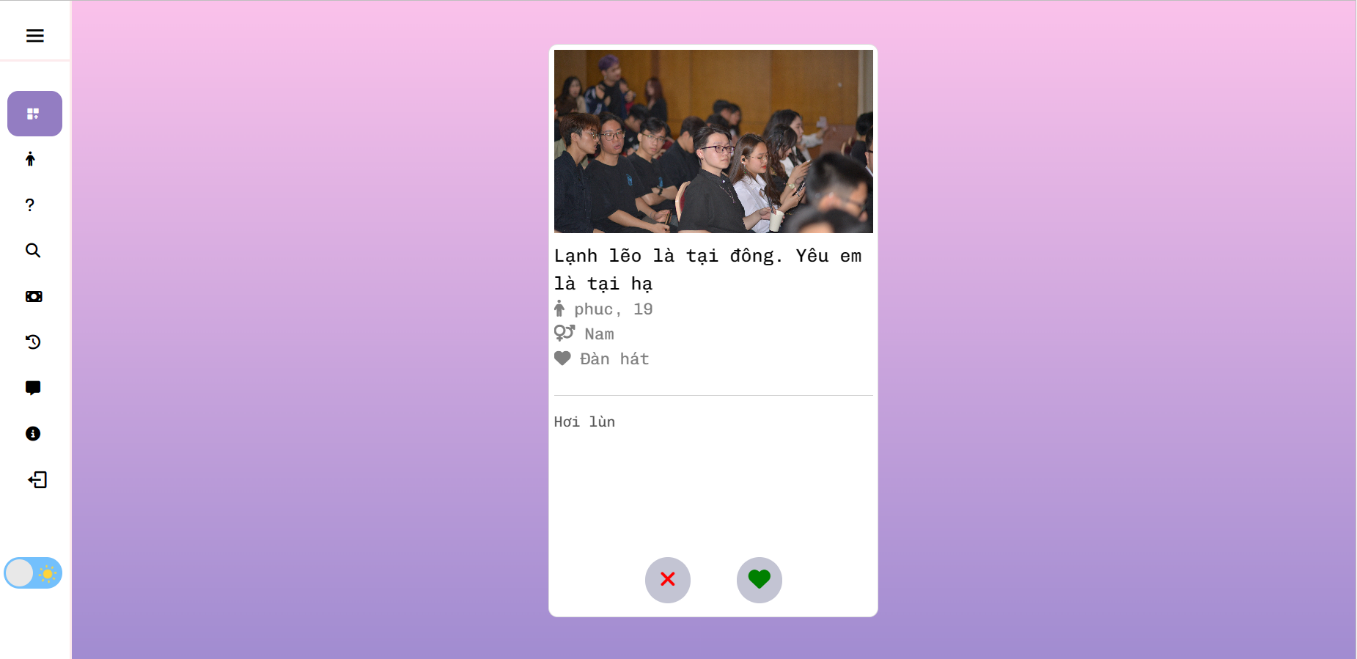
* + - * Đăng ký không thành công:



* + - * Đăng nhập:



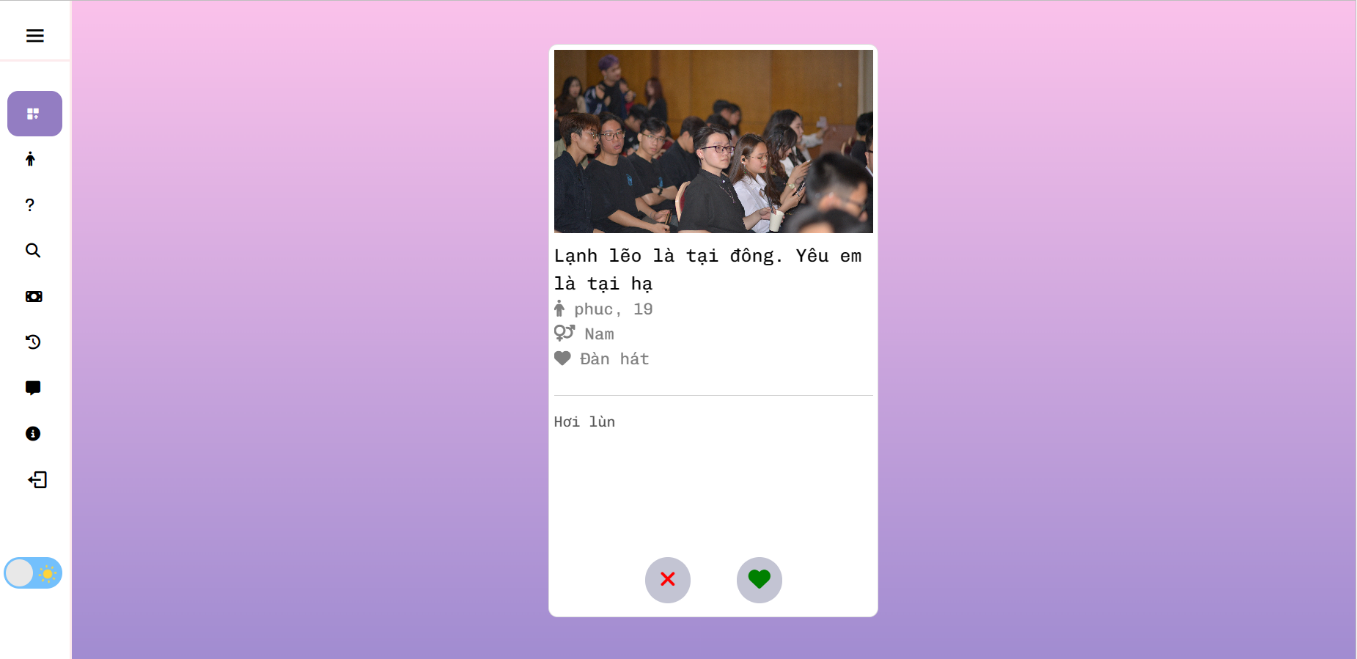
* + - * Đăng nhập thành công:



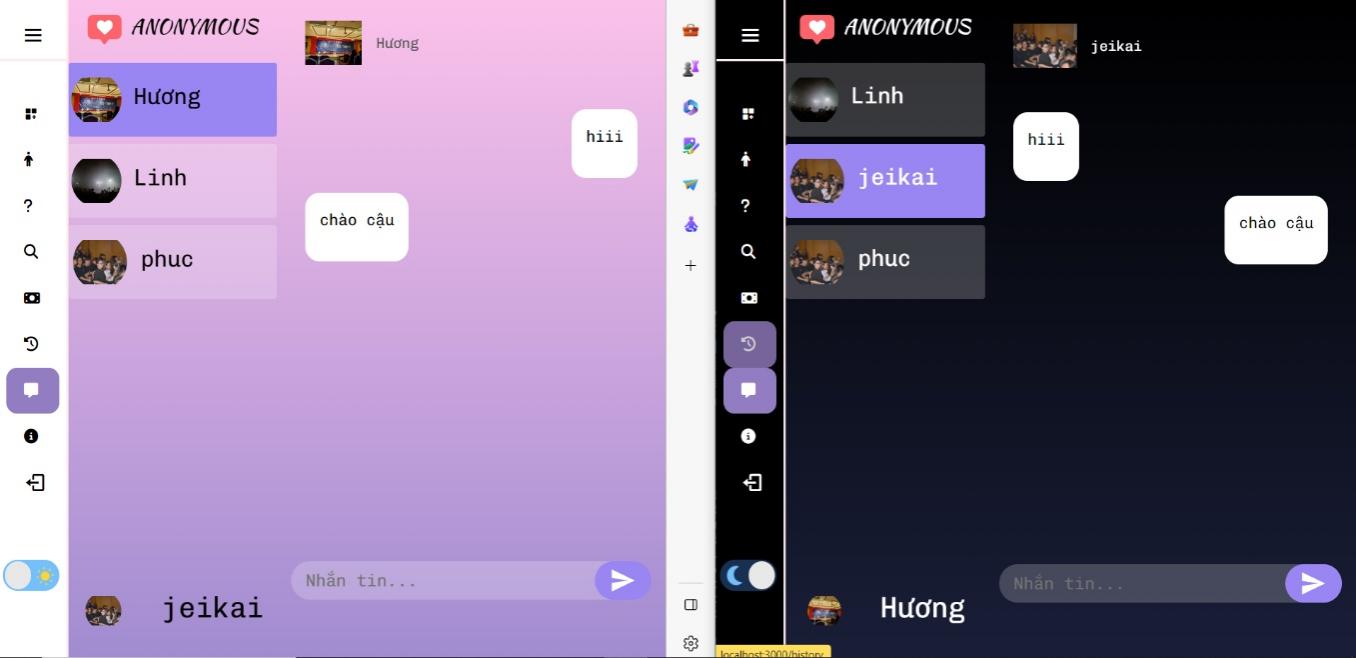
* + - * Đăng xuất:



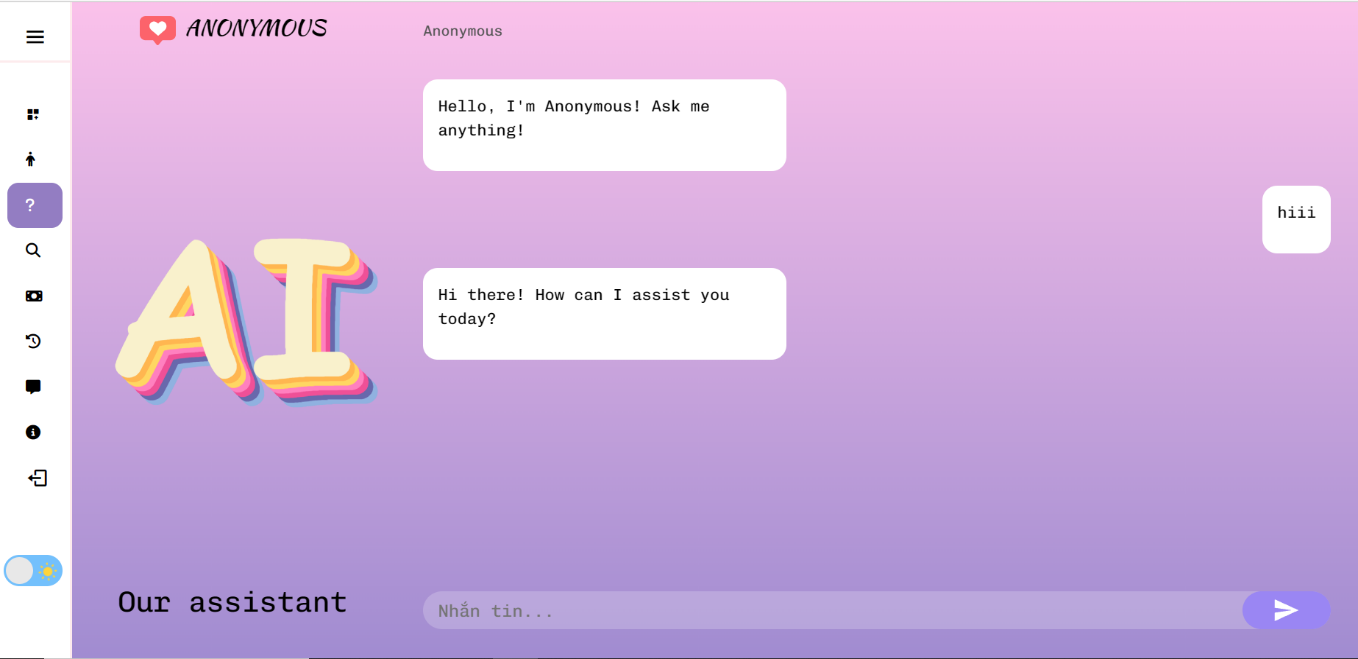
* + - * Trang chủ người dùng:



* + - * Nhắn tin với người dùng:

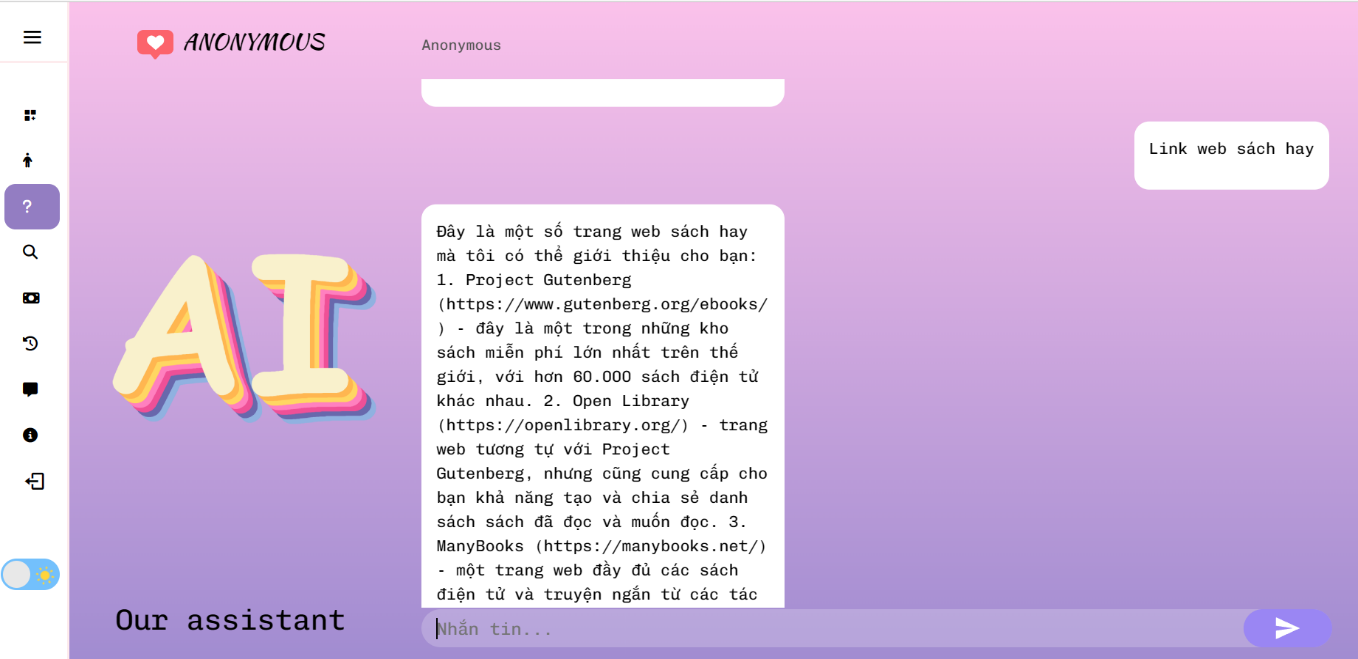


* + - * Nhắn tin với trợ lý ảo:

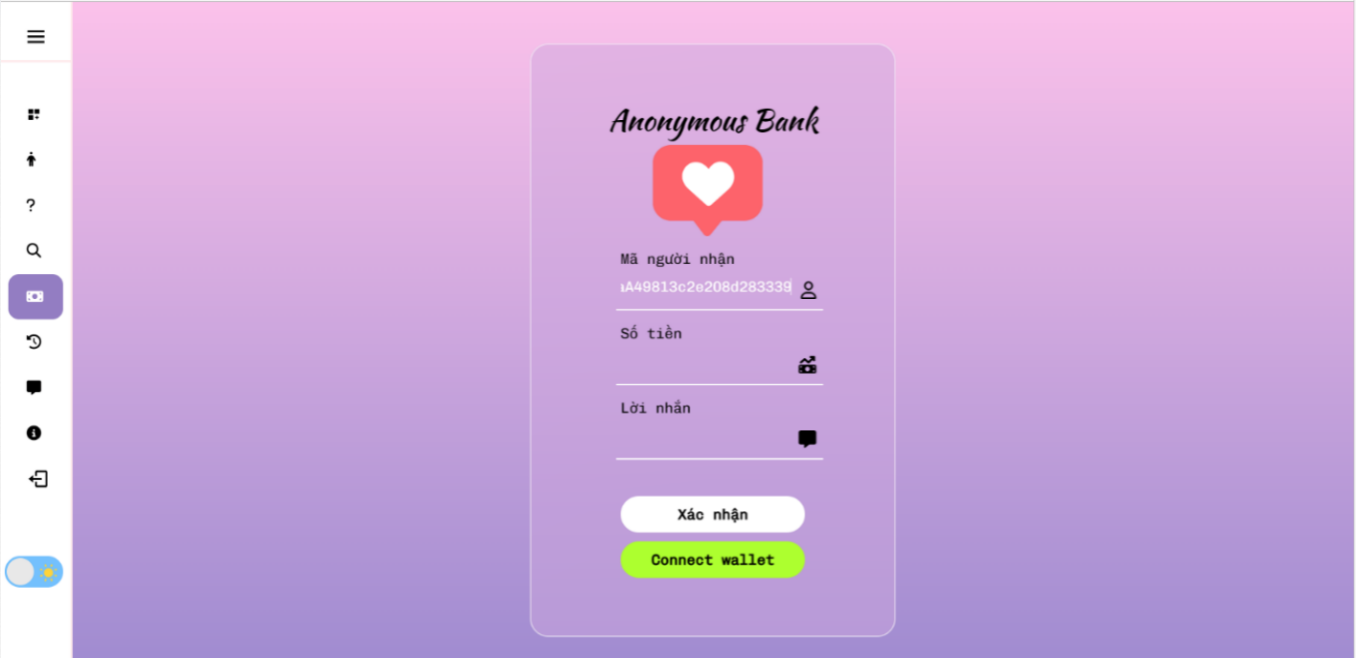


\

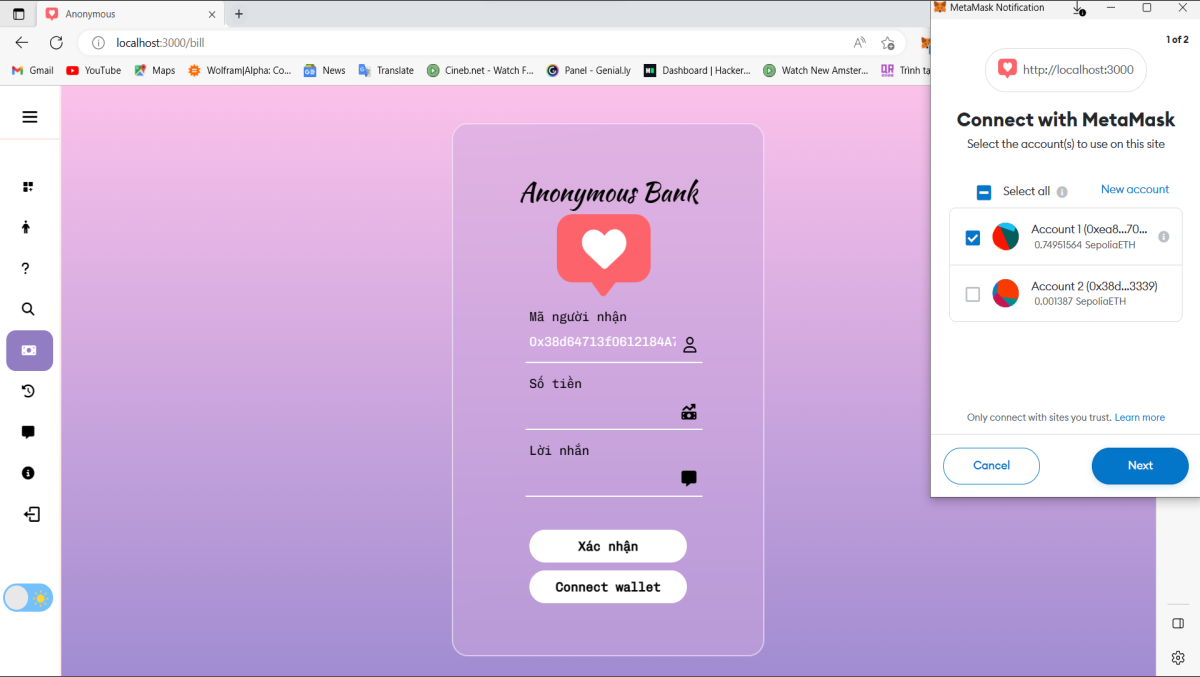
* + - * Đường liên kết ngoài:



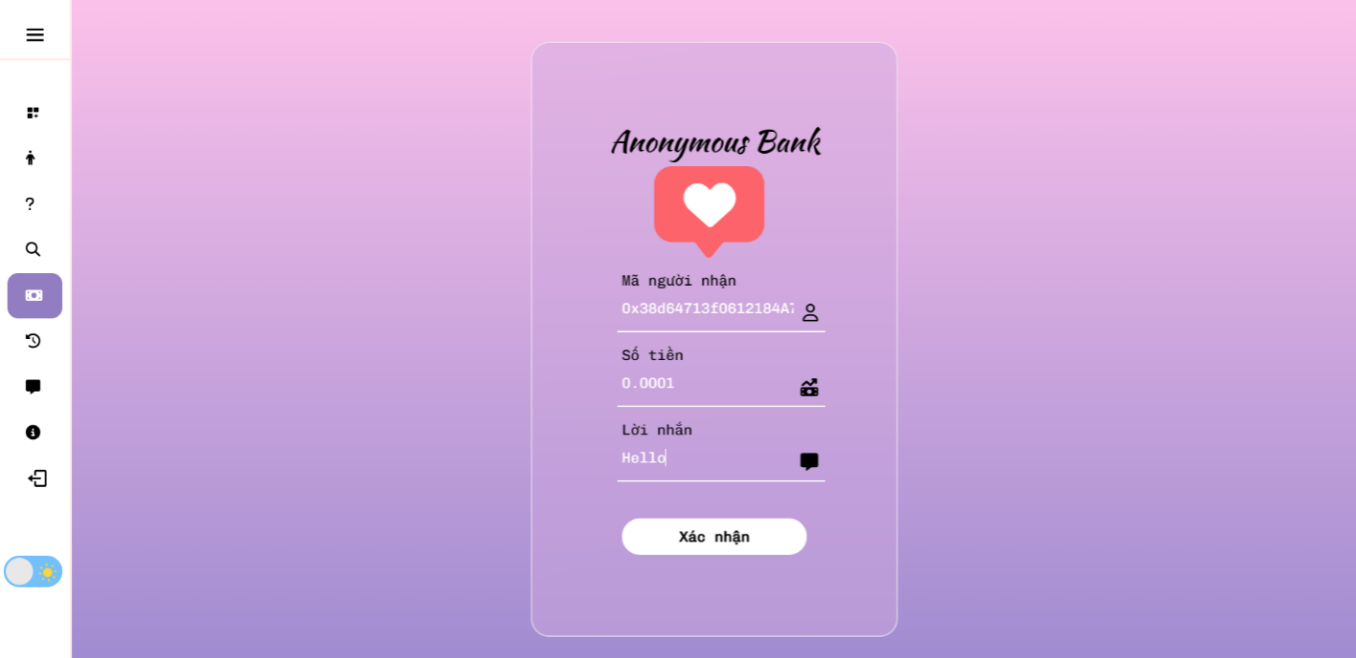
* + - * Chuyển tiền:
* Kết nối với ví điện tử



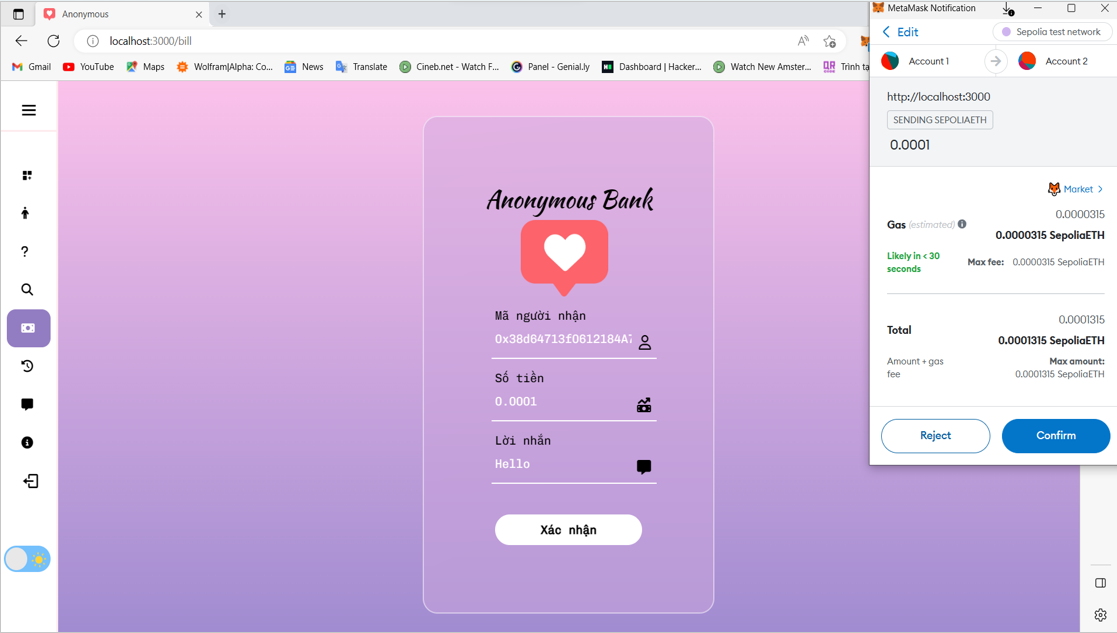
* Xác nhận kết nối với ví điện tử



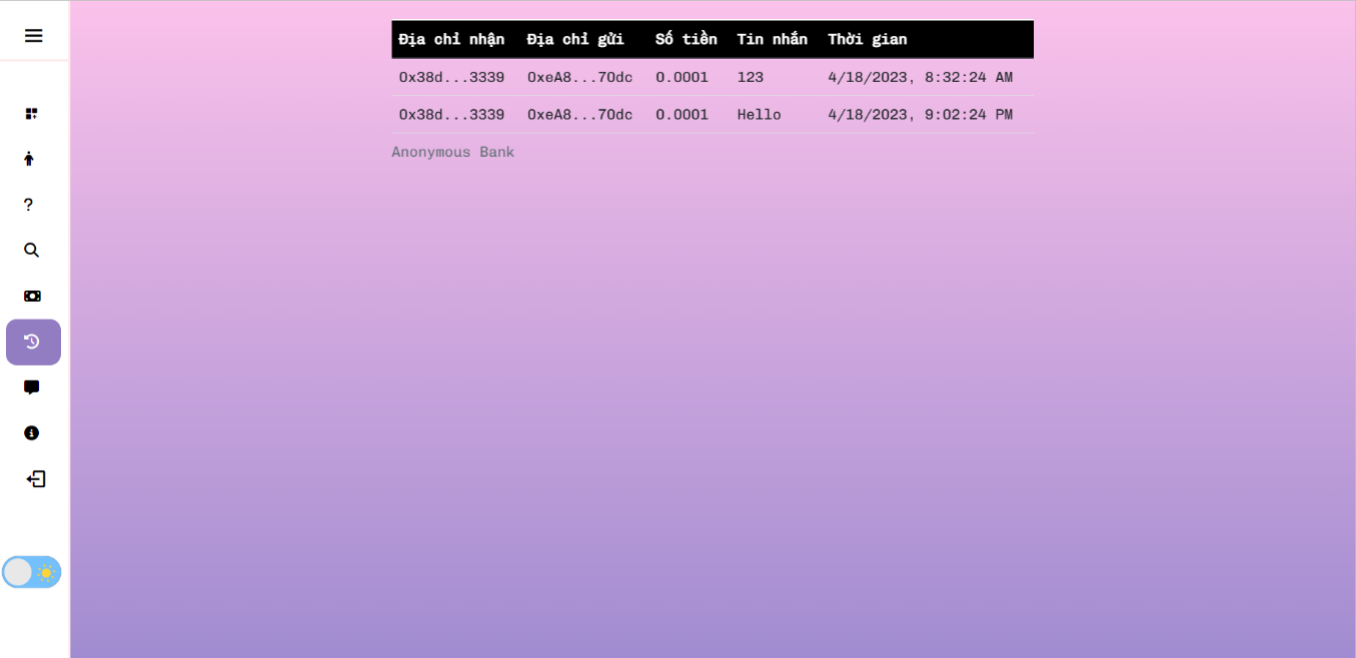
* Điền thông tin chuyển tiền:



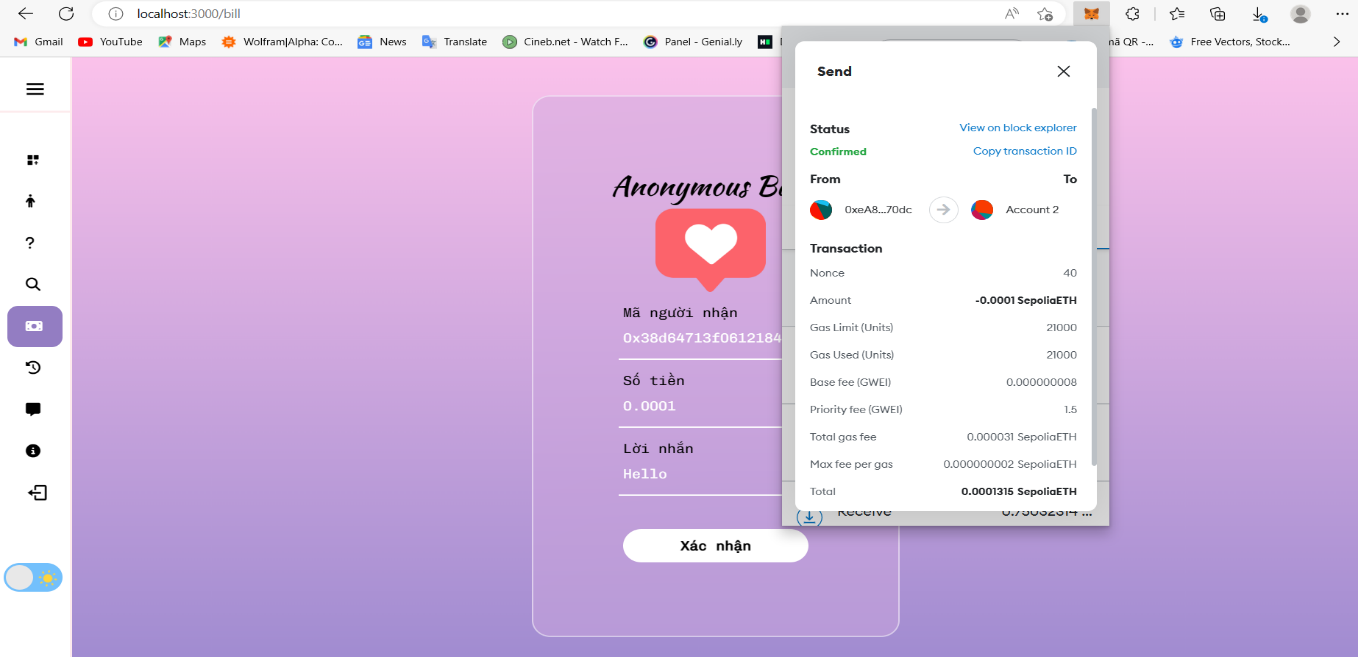
* Ví hiện tử hiện lên và xác nhận hóa đơn



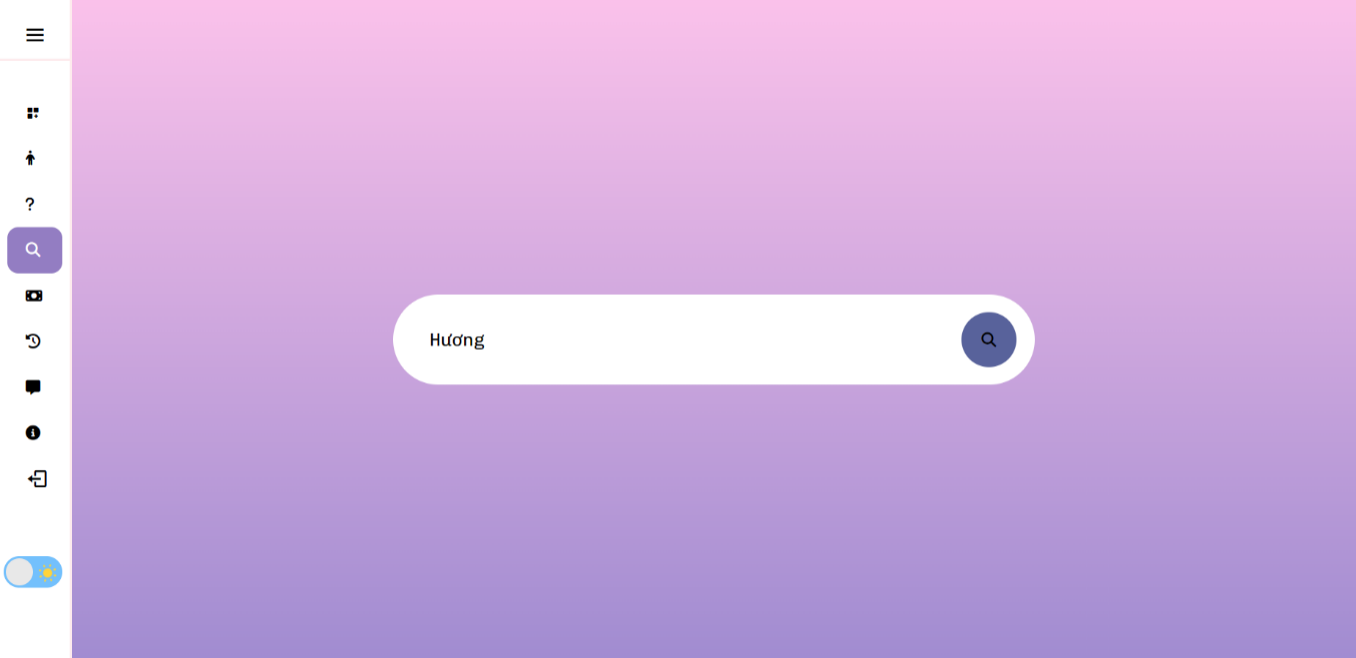
* Kết quả giao dịch



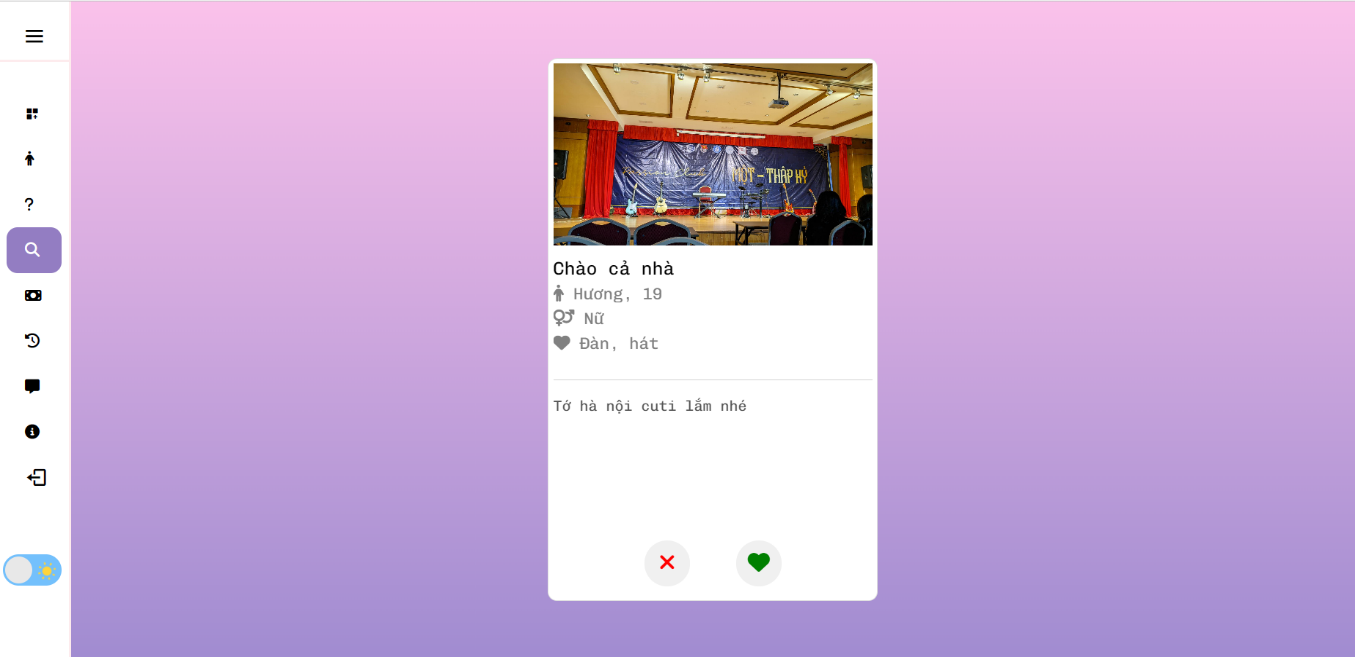
* + - * Lịch sử giao dịch:



* + - * Tìm kiếm người dùng:



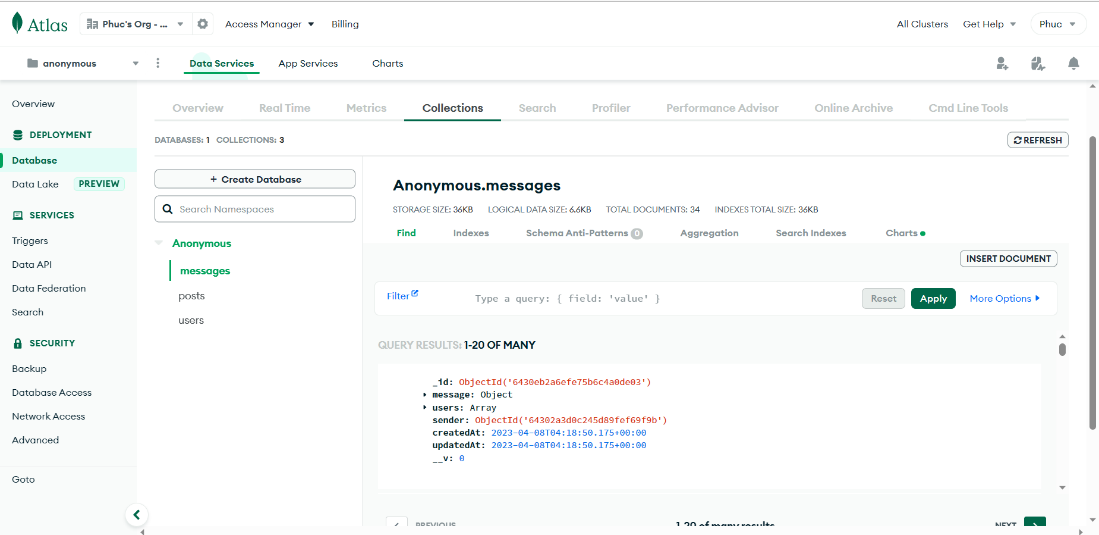
* + - * Kết quả tìm kiếm:



* + - * Chuyển sang Dark mode:



* + - 1. Trên Cơ sở dữ liệu:

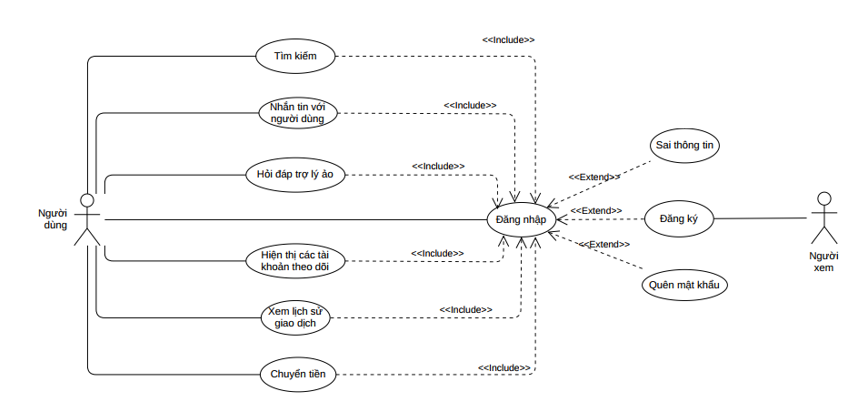


* + 1. *Ràng buộc trong quy trình phát triển và sử dụng phần mềm*
       - Trang web sử dụng ngôn ngữ JavaScript, Framework Bootstrap. Sử dụng HTML và CSS để căn chỉnh dữ liệu hiển thị
       - Cơ sở dữ liệu: MongoDB và Mongoose
       - Yêu cầu comment các hàm, biến quan trọng
       - Yêu cầu clean-code, coding convention
  1. Thiết kế phần mềm
     1. *Thiết kế hệ thống*
        1. Kiến trúc hệ thống:
* Gồm danh sách các use-case, Lược đồ use-case, Đặc tả.

Danh sách các use-case

|  |  |  |
| --- | --- | --- |
| STT | Tên use-case | Actor |
| 1 | Đăng ký tài khoản | Người xem |
| 2 | Đăng nhập hệ thống | Người dùng |
| 3 | Truy cập trang chủ | Người dùng |
| 4 | Tìm kiếm người dùng | Người dùng |
| 5 | Nhắn tin với người dùng khác | Người dùng |
| 6 | Hỏi đáp với trợ lý ảo | Người dùng |
| 7 | Chuyển tiền | Người dùng |
| 8 | Xem lịch sử giao dịch | Người dùng |

Lược đồ use-case



**Đặc tả**

* + - * Người xem

|  |  |
| --- | --- |
| Use-Case | Nội dung |
| Tên use-case | Đăng ký tài khoản |
| Mô tả | Cho phép người xem tạo tài khoản đăng nhập vào hệ thống và nhận được những đặc quyền khi có tài khoản |
| Actor | Người xem |
| Kích hoạt | Khi bấm chọn Đăng ký |
| Điều kiện | Cần truy cập vào trang web |
| Kết quả | Đăng ký tài khoản thành công |
| Các sự kiện chính | 1. Hệ thống hiển thị màn hình đăng ký tài khoản 2. Người xem nhập đủ các thông tin 3. Hệ thống hiển thị kiểm tra thông tin đăng ký 4. Nếu thành công thì hiện thông báo đăng ký thành công 5. Kết thúc use-case |
| Các sự kiện phụ | 1. Nếu nhập thiếu hoặc sai thông tin (username đã tồn tại) thông báo lỗi 2. Nếu mật khẩu không đầy đủ các dạng ký tự theo yêu cầu thì hiện thông báo 3. Làm mới trang đăng ký và lặp lại các sự kiện chính |

* + - * Người dùng

|  |  |
| --- | --- |
| Use-Case | Nội dung |
| Tên use-case | Đăng nhập hệ thống |
| Mô tả | Cho phép người xem đăng nhập vào hệ thống để nhận những đặc quyền của người dùng |
| Actor | Người xem |
| Kích hoạt | Khi người xem bấm chọn đăng nhập từ trang web |
| Điều kiện | Người xem phải có tài khoản đã đăng ký thành công |
| Kết quả | Người dùng đăng nhập thành công |
| Các sự kiện chính | 1. Hệ thống hiển thị màn hình đăng nhập 2. Người dùng nhập tên đăng nhập và mật khẩu 3. Hệ thống hiển thị kiểm tra thông tin đăng nhập 4. Nếu thành công chuyển trực tiếp vào trang tài khoản 5. Kết thúc use-case |
| Các sự kiện phụ | Trường hợp mật khẩu không hợp lệ thì hiện thông báo lỗi cho người xem và trở về trang đăng nhập, lặp lại luồng sự kiện chính |

|  |  |
| --- | --- |
| Use-Case | Nội dung |
| Tên use-case | Hiện thị các tài khoản theo dõi |
| Mô tả | Người dùng bấm vào trang chủ và hệ thống sẽ hiển thị từng tài khoản mà người đó theo dõi. |
| Actor | Người dùng |
| Kích hoạt | Người dùng bấm vào trang chủ. Và hệ thống sẽ hiện thị 1 tài khoản theo dõi |
| Điều kiện | Phải đăng nhập vào tài khoản trên hệ thống |
| Kết quả | Hiện thị những người theo dõi thành công |
| Các sự kiện chính | 1. Bấm vào trang chủ 2. Kết thúc use case |
| Các sự kiện phụ | 1. Kéo thả ảnh của tài khoản theo dõi để hiện thị tài khoản theo dõi khác |

|  |  |
| --- | --- |
| Use-Case | Nội dung |
| Tên use-case | Tìm kiếm người dùng |
| Mô tả | Hiện thị những tài khoản người dùng khác dựa theo thông tin tìm kiếm |
| Actor | Người dùng |
| Kích hoạt | Bấm vào nút tìm kiếm trên trang web |
| Điều kiện | Truy cập vào tài khoản trên hệ thống |
| Kết quả | Hiện thị ài khoản dựa theo tìm kiếm thành công |
| Các sự kiện chính | 1. Nhập thông tin của tài khoản tài khoản tìm kiếm 2. Bấm vào tìm kiếm 3. Hiện thị tài khoản dựa theo thông tin tìm kiếm 4. Kết thúc use case |

|  |  |
| --- | --- |
| Use-Case | Nội dung |
| Tên use-case | Nhắn tin với người dùng khác |
| Mô tả | Người dùng nhập thông tin muốn gửi cho người dùng khác |
| Actor | Người dùng |
| Kích hoạt | Bấm vào nút gửi trên trang web hoặc ấn phím Enter sau khi nhập tin nhắn |
| Điều kiện | Đăng nhập vào tài khoản trên hệ thống , tin nhắn phải có ký tự |
| Kết quả | Hiện thị tin nhắn muốn gửi thành công |
| Các sự kiện chính | 1. Nhập tin nhắn muốn gửi 2. Gửi tin nhắn 3. Kết thúc use case |

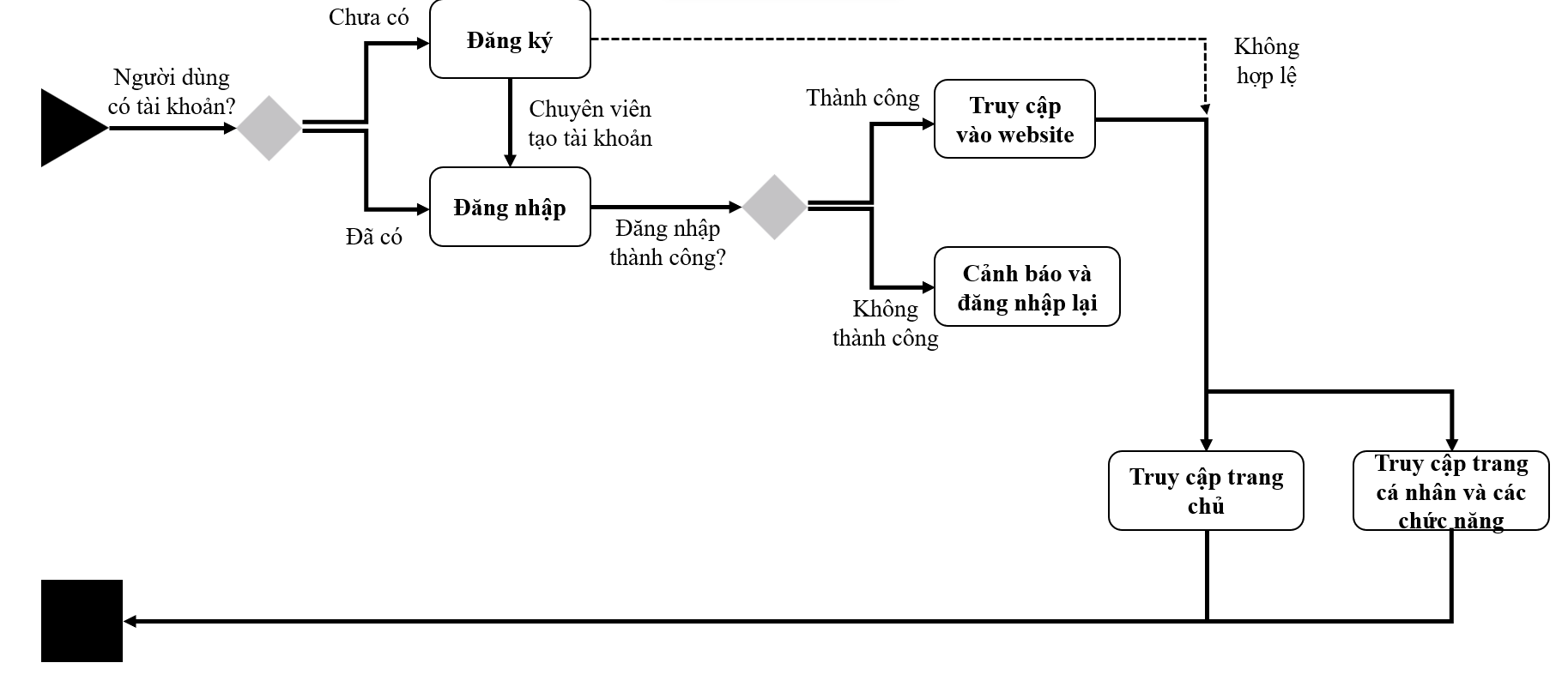
|  |  |
| --- | --- |
| Use-Case | Nội dung |
| Tên use-case | Hỏi đáp với trợ lý ảo |
| Mô tả | Người dùng nhập nhập câu hỏi và gửi cho trợ lý ảo. Và sau đó trợ lý ảo sẽ gửi trả lại câu trả lời cho người dùng tham khảo |
| Actor | Người dùng |
| Kích hoạt | Nhập câu hỏi trên và ấn phím Enter hoặc Bấm nút gửi |
| Điều kiện | Cần có tài khoản trên hệ thống |
| Kết quả | Hiện thị câu trả lời thành công |
| Các sự kiện chính | 1. Nhập câu hỏi muốn giải đáp 2. Gửi câu hỏi 3. Trợ lý ảo gửi câu trả lời 4. Kết thúc use case |

|  |  |
| --- | --- |
| Use-Case | Nội dung |
| Tên use-case | Chuyển tiền |
| Mô tả | Người dùng chuyển tiền cho một người dùng khác qua ví điện tử |
| Actor | Người dùng |
| Kích hoạt | Người dùng sẽ kết nối với ví điện tử trên trang web thông qua nút connect wallet sau đó nhập thông tin cần thiết rồi bấm xác nhận. Ví điện tử sẽ hiện lên để xác nhận hóa đơn |
| Điều kiện | Phải cài đặt ví điện tử và đăng nhập vào tài khoản trên hệ thống |
| Kết quả | Chuyển tiền thành công |
| Các sự kiện chính | 1. Kết nối ví điện tử 2. Nhập thông tin giao dịch 3. Xác nhận hóa đơn 4. Kết thúc use case |

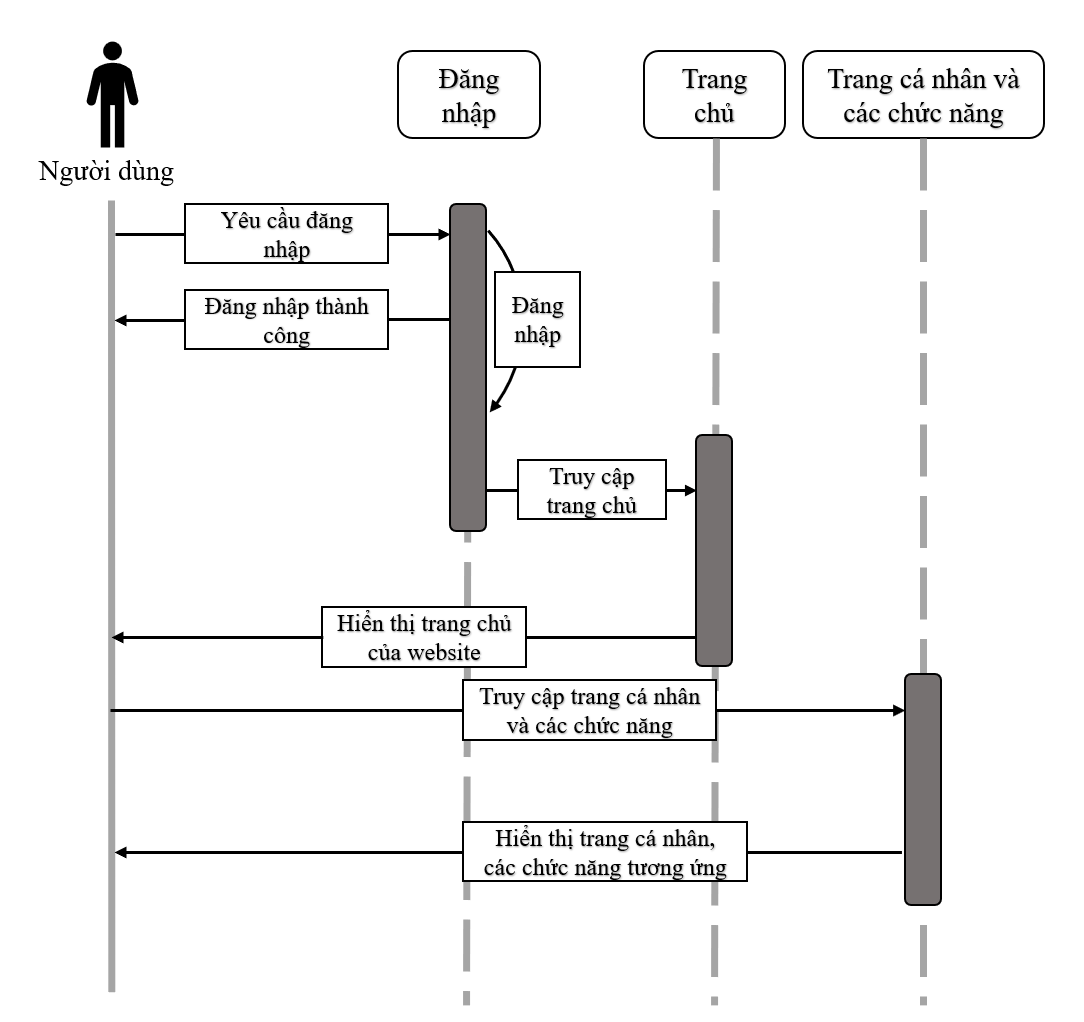
|  |  |
| --- | --- |
| Use-Case | Nội dung |
| Tên use-case | Xem lịch sử giao dịch |
| Mô tả | Người dùng xem lại các giao dịch đã thực hiện |
| Actor | Người dùng |
| Kích hoạt | Người dùng bấm vào nút lịch sử giao dịch trên trang web |
| Điều kiện | Phải đăng nhập vào tài khoản người dùng |
| Kết quả | Hiện thị thành công |
| Các sự kiện chính | 1. Người dùng bấm vào nút lịch sử giao dịch 2. Hệ thống hiển thị thông tin cần thiết 3. Kết thúc use case |

* + - 1. Các biểu đồ

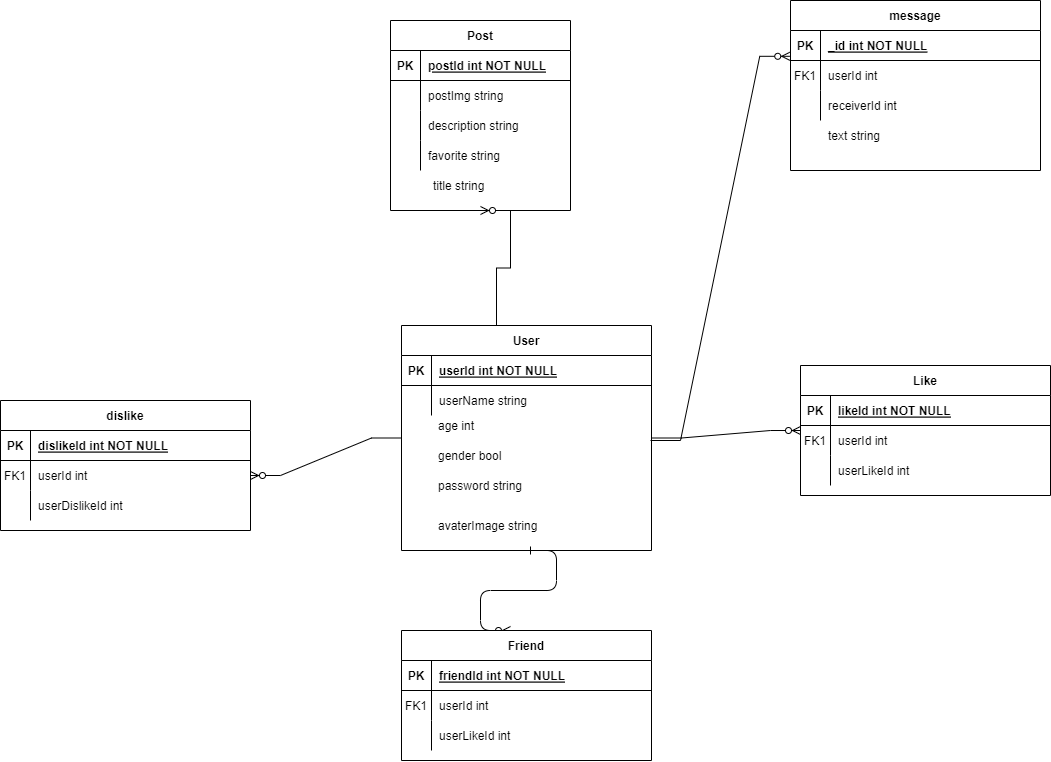
Biểu đồ trạng thái



Biểu đồ tuần tự



* + 1. *Thiết kế cơ sở dữ liệu*
       1. Mô hình chung:



* + - 1. Các bảng dữ liệu:

|  |  |  |  |
| --- | --- | --- | --- |
| **User** | | | |
| **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Null** |
| userId | Id của riêng mỗi người dùng | ObjectId | X |
| userName | Tên hiển thị của người dùng | string | X |
| age | Tuổi hiển thị của người dùng | int | X |
| gender | Giới tính hiện thị của người dùng | bool | X |
| password | Mật khẩu tài khoản người dùng | string | X |
| avatarImage | Ảnh hiển thị của người dùng | string | O |
|  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Post** | | | |
| **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Null** |
| postId | Id riêng của mỗi bài đăng | ObjectId | X |
| title | Tiêu đề của mỗi bài đăng | string | X |
| favorite | Sở thích của người dùng | string | X |
| description | Mô tả chi tiết về người dùng hiển thị trên bài đăng | string | X |
| postImg | Ảnh bài đăng | string | O |

|  |  |  |  |
| --- | --- | --- | --- |
| **message** | | | |
| **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Null** |
| \_id | Id riêng của mỗi hội thoại | ObjectId | X |
| userId | Id riêng của người gửi | ObjectId | X |
| receiverId | Id riêng của người nhận tin nhắn | ObjectId | X |
| text | Nội dung tin nhắn | string | X |

|  |  |  |  |
| --- | --- | --- | --- |
| **Like** | | | |
| **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Null** |
| likeId | Id riêng của bảng | ObjectId | X |
| userId | Id của người dùng và là người thích | ObjectId | X |
| userLikedId | Id của người được thích | ObjectId | X |

|  |  |  |  |
| --- | --- | --- | --- |
| **Dislike** | | | |
| **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Null** |
| dislikeId | Id riêng của bảng | ObjectId | X |
| userId | Id của người dùng và là người không thích | ObjectId | X |
| userDislikedId | Id của người không được thích | ObjectId | X |

|  |  |  |  |
| --- | --- | --- | --- |
| **friend** | | | |
| **Tên** | **Mô tả** | **Kiểu dữ liệu** | **Null** |
| friendId | Id riêng của bảng | ObjectId | X |
| userId | Id của người dùng | ObjectId | X |
| userLikedId | Id của bạn bè | ObjectId | X |

* 1. Cài đặt phần mềm
* Yêu cầu: Cần tải IDE tương thích (Visual Studio, Visual Studio Code), NodeJS, MongoDB, MongoDB Compass, các npm cần thiết
* Cài đặt:
  + git clone <https://github.com/jeikai/anonymous.git>
  + cd anonymous
  + npm install
  + cd server
  + npm start
  + Thêm các database vào MongoDB: 127.0.0.1:27017
* Chạy chương trình:
  + cd anonymous
  + npm start
  + cd server
  + npm start
  + Truy cập vào <http://localhost:3000/>
  1. Kiểm thử phần mềm đối với trang web

|  |  |  |
| --- | --- | --- |
| Test | Yêu cầu | Đánh giá |
| Đăng ký | Truy cập vào trang đăng ký | Đạt |
| Nhập tên đăng nhập đã tồn tại sẽ báo lỗi | Chưa đạt |
| Tạo tên đăng nhập chưa tồn tại thành công | Đạt |
| Yêu cầu về tính bảo mật cho mật khẩu | Đạt |
| Yêu cầu nhập đúng mật khẩu hai lần | Đạt |
| Yêu cầu nhập đủ thông tin thì đăng ký thành công và hiện thông báo | Đạt |
| Đăng nhập | Yêu cầu nhập đủ hai mục username và password | Đạt |
| Nhập tên tài khoản không tồn tại hiện thông báo nhập sai | Đạt |
| Nhập sai mật khẩu hiện thông báo nhập sai | Đạt |
| Nhập đúng tài khoản và mật khẩu đã đăng ký thì hiện đăng nhập thành công | Đạt |
| Cài đặt lại mật khẩu khi quên | Chưa đạt |
| Hiển thị các sự kiện, đề mục | Hiển thị theo đúng chuẩn CSS và HTML như trong dữ liệu | Đạt |
| Hiển thị thông tin người dùng đầy đủ như trong Cơ sở dữ liệu | Đạt |
| Chuyển trang, chuyển đến liên kết chính xác | Đạt |
| Các chức năng | Cho phép người dùng nhắn tin với người khác | Đạt |
| Tìm kiếm người dùng | Đạt |
| Hỏi đáp với trợ lý ảo | Đạt |
| Ví điện tử cho phép người dùng chuyển tiền trực tuyến | Đạt |
| Chức năng cập nhập hồ sơ người dùng | Đạt |

1. **Kết luận**
   1. Ưu điểm

* Phần mềm được xây dựng khá sát với yêu cầu đã đề ra
* Giao diện màu sắc của trang web dễ nhìn, thu hút được người dùng
* Thao tác đơn giản, thân thiện
  1. Nhược điểm
* Hệ thống còn đơn giản nên còn nhiều thiếu sót cần được khắc phục
* Website mởi chỉ chạy được trên localhost, người dùng chỉ có thể truy cập vào website thông qua máy tính đang chạy web server. Người dùng từ xa sẽ không thể truy cập được vào website
* Website chưa có nút bấm để lựa chọn người mình thích mà phải vuốt con trỏ chuột có thể gây bất tiện với người dùng
  1. Hướng nâng cấp phần mềm
* Nâng cao chức năng tìm kiếm theo thông tin như tên, sở thích, tuổi
* Triển khai website lên máy chủ để website trực tiếp đi vào hoạt động
* Tích hợp tính năng đăng nhập bằng tài khoản Facebook hoặc Google: Tích hợp tính năng đăng nhập bằng tài khoản Facebook hoặc Google giúp cho người dùng có thể đăng nhập và đăng ký nhanh chóng hơn, tạo sự thuận tiện cho người dùng.
* Phát triển thêm tính năng gợi ý người dùng trong một phạm vi nhất định
* Tạo thêm khung đánh giá và nhận xét về website cho người dùng để từ đó hoàn thiện và nâng cao mức độ hài lòng của người dùng khi sử dụng website.
* Dựa trên tính năng ví điện tử có thể phát triển thêm các tính năng VIP cho người dùng khi nạp tiền vào web