ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG

BÁO CÁO

MÔN HỌC: ĐỒ ÁN CHUYÊN NGÀNH

Đề tài:

THIẾT KẾ CÔNG CỤ QUẢN LÝ DỰ ÁN CÁ NHÂN VÀ NHÓM PTOLLO

Designing individual and team project management tools Ptollo

Tp. Hồ Chí Minh, 12/2024

LỜI NÓI ĐẦU

Trong thời đại công nghệ phát triển hiện nay, việc quản lý các công việc hay dự án một cách hiệu quả là một trong những yếu tố quan trong để một các nhân hay tập thể thành công. Bởi vì sự phức tạp và tốc độ làm việc nhanh nên các công cụ quản lý dự án trước đây như giấy tờ hay bảng tính không còn đáp ứng được nhu cầu của người dùng.

Xuất phát từ nhu cầu tối ưu hóa quá trình làm việc của con người hiện nay nhóm em đã tiến hành nghiên cứu, thiết kế và phát triển một ứng dụng web quản lý dự án cá nhân và nhóm có tên là PTollo. Sản phẩm của nhóm không chỉ dùng để chia nhỏ, sắp xếp thứ tự công việc giúp người dùng quản lý công việc một cách hiệu quả mà còn tích hợp các tính năng như chatbox và video call nhằm tăng sự tương tác giữa các thành viên trong một nhóm từ đó hiệu suất và tính chính xác của công việc được cải thiện, tạo ra một môi trường làm việc chuyên nghiệp, hiệu quả.

Chúng em sẽ trình bày chi tiết để tài mà nhóm thực hiện qua năm chương như sau:

- Chương 1: Tổng quan đề tài.
- Chương 2: Cơ sở lý thuyết.
- Chương 3: Thiết kế và phân tích sơ đồ hệ thống.
- Chương 4: Hiện thực hệ thống.
- Chương 5: Đánh giá kết quả và hướng phát triển.

MỤC LỤC

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1.1 Giới thiệu đề tài

Trong thời đại 4.0 hiện nay, con người đang phải đối mặt với mật độ công việc dày đặt, tầng suất các dự án liên tục và thời gian thực hiện vô cùng gấp rút. Vì vậy việc sắp xếp và quản lý các công việc hay dự án và một kỹ năng bắt buộc đối với mỗi cá nhân hay tập thể nào đó. Các công cụ quản lý công việc từ trước đến nay gần như là giấy hay các bảng tính cơ bản khiến bản thân người dùng gặp khó khăn khi phải đối mặt với quá nhiều dự án mà không có thời gian quản lý.

Các khó khăn mà người dùng phải đối mặt khi quản lý dự án theo cách truyền thống xưa nay có thể kể ra là là:

- Các thông tin có thể bị phân tán, không tập trung lại cũng như dễ bị thất lạc hoặc nhầm lẫn khi quản lý nhiều tài liệu cùng một lúc.
- Khó khăn trong việc cập nhật tiến độ công việc.
- Khi xảy ra một sự cố hay một thay đổi trong công việc, người dùng thường gặp khó khăn khi thay đổi tiến độ.
- Mất rất nhiều thời gian để tìm kiếm thông tin dự án.

- Thụ động trong việc phổ cập thông tin đến các thành viên trong team.
- Thiếu sự tương tác trực tiếp giữa các thành viên.
- Khó bảo quản, lưu trữ, dễ hỏng hóc các tài liệu quan trọng.
- Các thông tin dễ bị thiếu sót trong quá trình đồng bộ hóa.

Vì vậy một ứng dụng giúp cá nhân và nhóm quản lý, phân công cũng như theo giỏi tiến độ công việc là một nhu cầu thiết yếu hiện nay.

Đề tài Thiết kế công cụ quản lý dự án cá nhân và nhóm PTollo được nhóm chúng em thực hiện với mục tiêu là đơn giản hóa quá trình quản lý công việc của không chỉ cá nhân hay nhóm nhỏ mà còn có thể phát triển cho các mô hình công ty lớn hơn.

1.2 Mục tiêu đề tài

- Nghiên cứu, tìm hiểu và thiết kế một ứng dụng có chức năng quản lý các dự án từ nhỏ đến lớn.
- Tìm hiểu các cơ sở lý thuyết cấu thành nên một ứng dụng web.
- Xây dựng giao diện website PTollo trực quan dễ nhìn và có thể hiển thị trên nhiều thiết bị.
- Hiện thực hóa các chức năng chính và chức năng mở rộng của ứng dụng.
- Phân tích và đánh giá kết quả của ứng dụng trong môi trường thực tiễn.

1.3 Phương pháp thực hiện đề tài

- Tìm hiểu, nghiên cứu cơ sở lý thuyết: Các thành phần của Website, cách thức một website vận hành, các công nghệ cần dùng trong đồ án và tài liệu chuyên sâu về các công nghệ đó, phân tích các sơ đồ như usecase, sơ đồ phân rã chức năng, sơ đồ hoạt động cũng như cơ sở dữ liệu mà Website trong đề tài sử dụng.
- Tiến hành thực nghiệm: Xây dựng giao diện người dùng và các chức năng phía backend cùng với việc liên kết cơ sở dữ liệu.
- Phân tích và đánh giá ứng dụng: Nêu các ưu nhược điểm của ứng dụng, đánh giá trải nghiệm người dùng và đề xuất hướng phát triển sau này.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1 Ngôn ngữ lập trình Javascript

2.1.1 Giới thiệu về Javascript

- Javascript là một ngôn ngữ lập trình cấp cao được sử dụng để tạo ra các tương tác cho website. Ngôn ngữ này đóng vai trò quan trọng trong việc đem lại những trải nghiệm cho người dùng. Ngày nay javascript đã phát triển mạnh mẽ để phát triển thành ứng dụng server-side thay vì chỉ là ứng dụng client-side như trước đây, nhờ đó các lập trình viên có thể sử dụng một ngôn ngữ cho cả frontend và backend. JavaScript đóng vai trò như là một thành phần quan trọng của trang web, giúp thực thi cho phép Client-side script từ phía người dùng cũng như phía máy chủ (Nodejs) tạo ra các trang web động (hay dynamic website).
- Trước đây các website thường sẽ ở dạng tĩnh, gần giống như việc chúng ta đọc sách, các thông tin của trang web được bố trí theo một bố cục cố định nhàm chán. Sau này Javascript được sử dụng để làm cho việc duyệt web diễn ra một cách linh hoạt vả hấp dẫn hơn. Khi dùng Javascript trình duyệt sẽ phản hồi người dùng đồng thời thay đổi bố cục nội dung của trang web. JavaScript không chỉ được dùng ở phía trình duyệt (frontend) mà ngôn ngữ này còn được triển khai ở phía máy chủ thông qua Node.js. Khi ở phía máy chủ, nhờ JavaScript người dùng có thể thực hiện các nhiệm vụ phức

tạp như xử lý logic backend, quản lý cơ sở dữ liệu, và tương tác với các dịch vụ web phía client.

2.1.2 Lịch sử hình thành và phát triển

- Từ lúc được tạo ra cho đến nay Javascript đã trải qua rất nhiều giai đoạn. Đầu tiên Javascript được phát triển vào những năm 1990 nhằm đáp ứng nhu cầu phát triển web động trong thời kỳ bùng nổ Internet.
- Vào năm 1995, Brendan Eich - một lập trình viên của Netscape Communications đã tạo ra phiên bản đầu tiên của ngôn ngữ này chỉ vỏn vẹn trong vòng 10 ngày trong bối cảnh mà Netscape muốn tạo ra một ngôn ngữ lập trình dễ học và sử dụng trên trình duyệt web. Mục tiêu của Netscape là làm sao cho các trang web trở nên tương tác hơn, thay vì chỉ hiển thị nội dung tĩnh. Đầu tiên ngôn ngữ này có tên là Mocha sau đó đổi tên thành LiveScript và cuối cùng là JavaScript cho đến hiện nay. Cái tên Javascript được đặt là để tận dụng sức nóng của Java mặc dù 2 ngôn ngữ này không liên quan trực tiếp đến nhau.
- Tiếp theo là từ năm 1996 đến năm 1999, đây gọi là giai đoạn của sự chấp nhận và tiêu chuẩn hóa. Năm 1996, Microsoft tung ra một phiên bản JavaScript riêng có tên là JScript, phiên bản này được tích hợp vào trong trình duyệt Internet Explorer 3. Tuy nhiên điều này lại gây ra sự xung đột, không tương thích vì vậy cần phải có một tiêu chuẩn chung nào đó. Năm 1997, Netscape đã gửi JavaScript cho ECMA International để tiêu chuẩn hóa, dẫn đến sự ra đời của chuẩn ECMAScript (ES). Cho đến năm 1999, phiên bản ECMAScript 3 (ES3) ra đời, bổ sung và hoàn thiện nhiều tính năng quan trọng.

- Cho đến những năm 2000 đến 2004, đây là thời kỳ mà Javascript phát triển chậm và gặp rất nhiều chỉ trích. Tuy được sử dụng rộng rãi nhưng javascript vẫn mắc phải các vấn đề như chạy không ổn định, thường gây ra lỗi hoặc thiếu nhiều tính năng quan trọng.
- Đến giai đoạn bùng nổ Web 2.0, giai đoạn này bắt đầu từ năm 2005 đến năm 2009. Năm 2005 Google đã giới thiệu một ứng dụng có tên là Google Maps, ứng dụng web đầu tiên sử dụng JavaScript và AJAX để tạo trải nghiệm tương tác cao. Đến năm 2006, một loạt các framework của javascript được ra đời như jQuery và Prototype.js.
- Và cuối cùng từ năm 2009 cho đến nay chính là giai đoạn đại tăng trưởng của Javascript. Vào năm 2009, Node.js chính thức được ra mắt giúp cho javascript mở rộng ra cả máy chủ. Năm 2015, ES6 ra đời đem lại cho người dùng nhiều tính năng và cải tiến mới. Và từ năm 2016 cho đến nay, hàng loạt các thư viện và framework mạnh mẽ ra đời có thể kể tên như React.js, Vue.js và Angular.
- Hiện nay JavaScript vẫn tiếp tục phát triển với sự ra mắt của các ECMAScript mới hàng năm, cùng với sự cải thiện đáng kể về hiệu năng và khả năng tương thích cao, ổn định trên các nền tảng.

2.1.3 Cách hoạt động của Javascript

Hầu hết các ngôn ngữ lập trình hiện nay đều hoạt động dựa trên cách dịch cú pháp thành mã máy, hệ điều hành sau đó sẽ chạy mã được dịch. JavaScript được phân thành một ngôn ngữ viết tập lệnh hay còn gọi là một ngôn ngữ diễn giải. Đoạn mã JavaScript sẽ được diễn giải, có nghĩa là JavaScript sẽ được dịch trực tiếp sang mã ngôn ngữ máy cơ bản. Đối với các ngôn ngữ lập trình khác, trình biên dịch sẽ biên dịch toàn bộ mã thành mã máy ở một bước riêng biệt. Vì vậy, tất cả các ngôn ngữ viết tập lệnh đều là ngôn ngữ lập trình, tuy vậy không phải tất cả các ngôn ngữ lập trình đều là ngôn ngữ viết tập lệnh.

Các công cụ JavaScript đầu tiên chỉ là bộ diễn giải, nhưng sau này tất cả các công cụ hiện đại đều sử dụng biên dịch tại chỗ hoặc thời gian chạy để cải thiện hiệu suất.

- JavaScript ở phía máy khách sẽ hoạt động trực tiếp trong trình duyệt, sử dụng các công cụ JavaScript được tích hợp trong trình duyệt để xử lý mã. Khi người dùng truy cập một trang web, trình duyệt sẽ tải nội dung và chuyển đổi các phần tử của trang web như botton, tags, box thành một cấu trúc dữ liệu gọi là DOM. Mã JavaScript sẽ được chuyển đổi thành bytecode để cho máy có thể hiểu được và thực

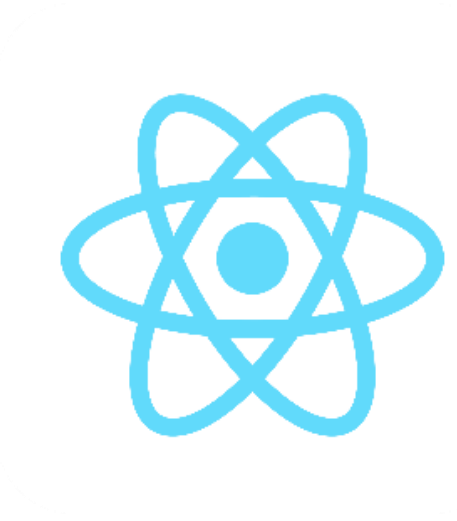
thi. Các sự kiện như nhấp chuột hoặc di chuột kích hoạt việc thực thi mã JavaScript, dẫn đến thay đổi DOM và cập nhật giao diện người dùng.

- JavaScript phía máy chủ hoạt động trên máy chủ, lập trình viên sử dụng công cụ JavaScript để xử lý các logic ở phía backend. Nó có thể được dùng để truy cập cơ sở dữ liệu, hay còn thực hiện được các phép toán logic và phản hồi các sự kiện từ hệ điều hành của máy chủ. Ưu điểm chính là khả năng tùy chỉnh phản hồi của trang web dựa trên yêu cầu của người dùng hoặc dữ liệu nhận được.

2.1.4 Ưu điểm của Javascript

- Javascript được xem như là một ngôn ngữ có thể tương thích đa nền tảng, bởi vì ngôn ngữ này có thể chạy trên hầu hết các trình duyệt hiện nay như cốc cốc, chrome, brave, firefox mà không cần cài thêm các phần mềm bên thứ ba.
- Cú pháp của Javascript khá đơn giản, dễ tiếp cận và phù hợp với hầu hết các lập trình viên từ mới bắt đầu đến chuyên nghiệp.
- Javascript đem lại một tốc độ thực thi nhanh vì được thực thi trực tiếp trên trình duyệt mà không cần biên dịch trước đó nhờ đó giúp phản hồi nhanh chóng khi người dùng tác động.
- Khả năng hỗ trợ nhiều tác vụ cùng một lúc như gọi API hay truyền tải dữ liệu mà không tác động tới luồng thực thi chính.
- Ngoài ra Javascript còn cung cấp khả năng tương tác động với HTML và CSS nhờ vậy lập trình viên có thể thay đổi nội dung, kiểu dáng và tương tác với trang web theo thời gian thực.
- Với Node.js javaScript không chỉ giới hạn ở phía client mà còn có thể được sử dụng để xây dựng các ứng dụng server-side mạnh mẽ.
- Cuối cùng JavaScript còn có thể thực thi một số chức năng ngay cả khi không kết nối Internet, giúp cải thiện trải nghiệm người dùng trong môi trường offline.

2.1.5 Framework



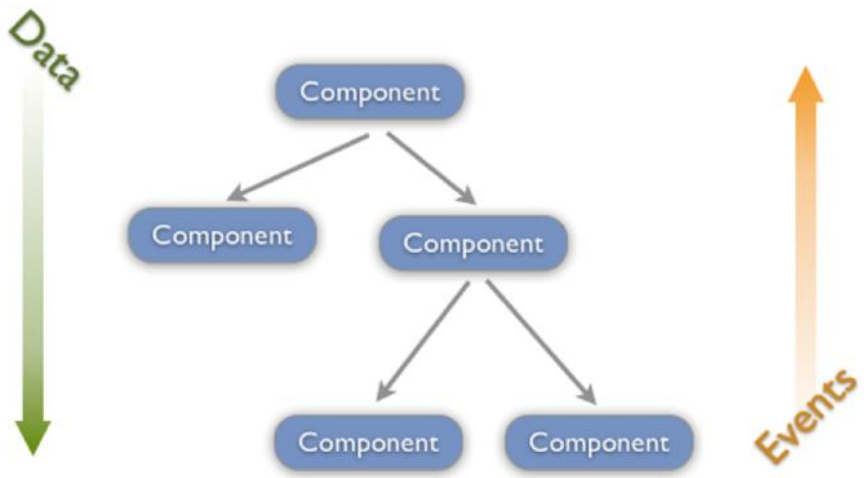
- **ReactJS** là một thư viện mã nguồn mở của javascript dùng để xây dựng giao diện phía người dùng dựa trên các thành phần UI đơn lẻ. ReactJS được phát triển và duy trì bởi Meta và cộng đồng các nhà phát triển và công ty cá nhân. React chỉ tập trung vào phần hiển thị giao diện (view), chứ không can thiệp vào cách sắp xếp logic nghiệp vụ hoặc cấu trúc của ứng dụng. Lập trình viên có thể tự do thiết kế giao diện so với các framework khác. Tuy nhiên vì vậy mà React trở nên khó học hơn đối với người mới bắt đầu lập trình hay khi phải quản lý các ứng dụng lớn.

  React được xây dựng trên kiến trúc component, có nghĩa là các thành phần có thể được tái sử dụng giúp ích cho việc tái sử dụng và duy trì hoạt động của ứng dụng. Lập trình viên có thể phát triển ứng dụng thành các quy mô lớn hơn bằng việc chia nhỏ các thành phần và tái sử dụng chúng mà không gặp vấn đề.

  Thành phần chính của ReactJS là JSX thường được hiểu là một cú pháp mở rộng cho phép lập trình viên viết các mã giống với HTML. Thay vì phải viết Javascript với HTML riêng thì với React ta có thể kết hợp chúng trong cùng một mã nguồn giúp lập trình viên dễ dàng quản lý. React xử lý các sự kiện bằng cách sử dụng cú pháp camelCase và truyền hàm sự kiện trực tiếp trong JSX.

  Virtual DOM cho phép người dùng không cần thao tác trực tiếp với DOM trên giao diện mà vẫn phản ánh được sự thay đổi đó. Vì Virtual-DOM vừa đóng vai trò là Model xử lý logic, vừa đóng vai trò là giao diện nên mọi sự thay đổi trên Model sẽ kéo theo sự thay đổi trên giao diện và ngược lại. Điều nay có nghĩa là mặc dù chúng ta không

tác động trực tiếp vào các phần tử DOM ở giao diện nhưng vẫn thực hiện được cơ chế Data - binding.



Redux là một thành phần quan trọng khác trong ReactJS. Đầu tiên nói đên khái niệm luồng dữ liệu một chiều. Trong ReacJS không có những module chuyên trách để xử lý dữ liệu vì vậy nên ReactJS sẽ chia nhỏ giao diện thành nhiều component nhỏ liên kết một cách chặt chẽ với nhau. Luồng dữ liệu này được quản lý và luân chuyển theo hướng component từ trên xuống hay còn gọi là từ component cha đến con. Các component cha có thể truyền dữ liệu cho component con thông qua các props, nhưng các component con không thể trực tiếp thay đổi ngược lại dữ liệu của component cha. Redux mang lại một số tính năng hữu ích cho lập trình viên như sau:

- Đầu tiên là khả năng lưu trữ trạng thái của ứng dụng trong một store duy nhất nhờ đó trạng thái sẽ dễ được quản lý, theo giỏi và đồng bộ giữa các thành phần với nhau.
- Cơ chế immutable state của Redux giúp mọi thay đổi đều phải được thông qua action để tạo ra một trạng thái mới.
- Redux tuân theo mô hình luồng dữ liệu một chiều: khi một sự kiện mô tả điều cần thay đổi, hàm thần sẽ xử lý hành động và trả về trạng thái mới sau đó lưu và store là nơi cung cấp dữ liệu cho giao diện.
- Không những thế thư viện này còn cho phép chia sẽ trạng thái giữa các thành phần mà không cần phải truyền dữ liệu thông qua nhiều cấp.
- Nó còn hỗ trợ công cụ cho phép ghi lại, kiểm tra và quay lại các trạng thái trước đó của ứng dụng.

- Và cuối cùng là Redux cung cấp khả năng mở rộng để xử lý các tác vụ bất đồng bộ như gọi API.

- **Express.js** là một framework quan trong được xây dựng trên nền tảng của Node.js, framework này cung cấp hỗ trợ giao thức HTTP, quản lý định tuyến (routing) và midleware tạo ra các APIs giúp ích cho việc phát triển ứng dụng web.

Nhờ có Express.js, các lập trình viên có thể xây dựng các API RESTful một cách dễ dàng và có thể quản lý cơ sở dữ liệu một cách hiệu quả mà không cần viết quá nhiều code phức tạp. Express.js giúp người dùng tập trung vào xử logic ứng dụng thay vì chỉ tập trung vào việc xử lý chi tiết các yêu cầu HTTP.

Các tính năng chính của Express.js có thể kể ra như sau:

- Tính năng Templating engines của Express.js cho phép người dùng tạo ra các templates chứa các placeholder sau đó có thể thay thế bằng việc chèn dữ liệu thực tế vào khi trang được hiển thị giúp dễ dàng tạo ra các trang HTML linh hoạt khi có các dữ liệu thường xuyên được thay đổi. Các template phổ biến thường xuyên xuất hiện là EJS, Pug hay Handlebars và Mustache.
- Việc sử dụng hàm express.static() làm cho việc xử lý các tệp tĩnh trong dự án được thực hiện dễ dàng. Express.js sẽ tự động cung cấp các tệp tĩnh như hình ảnh, CSS, JavaScript khi được người dùng yêu cầu. Các tệp này sẽ được gửi trực tiếp đến người dùng.

- Chức năng định tuyến sinh ra để giúp người dùng giải quyết các yêu cầu HTTP từ người dùng đến đúng nơi trong ứng dụng. Mỗi một URL cụ thể sẽ được gắn riêng với một route để xử lý các yêu cầu khác nhau GET, POST, PUT, DELETE.
- Sử dụng các hàm đặc biệt middleware giúp người dùng thực hiện các tác vụ như ghi lại các thông tin được yêu cầu, xác thực danh tính người dùng hay xử lý cơ sở dữ liệu.
- Cuối cùng là tính năng hỗ trợ các lập trình viên xây dựng RESTful API một cách dễ dàng thông qua việc thiết lập các route tương ứng với từng yêu cầu cụ thể.

- **Tailwindcss** là một Css framework utility-first hay còn được hiểu là ưu tiên tiện ích nhằm mục tiêu cung cấp các tiện ích cấp thấp giúp tối ưu cho việc xây dựng hay thiết kế giao diện một cách hiệu quả. Tailwindcss có nhiều class với nhiều thuộc tính Css khác nhau giúp việc mở rộng class từ chính những class này được thực hiện một cách dễ dàng.

Khi dùng Tailwindcss ta không cần thiết phải viết css lại từ đầu mà chỉ cần áp dụng các lớp tiện ích trực tiếp vào HTML. Cơ chế JIT cho phép Tailwind chỉ tạo css mà lập trình viên đang sử dụng điều này làm bỏ đi những file không cần thiết giúp cho Css nhỏ hơn đáng kể. Tailwindcss cung cấp tính tùy biến cao, người dùng có thể mở rộng hoặc thay đổi các giá trị mặc định trong file có tên là tailwind.config.js, tạo các theme riêng hoặc thêm các tiện ích khác nhau.

2.2 So sánh Html, Css và Javascript

- **HTML** hay ngôn ngữ đánh giá siêu văn bản (Hypertext Markup Language) không được xem là một ngôn ngữ lập trình chính thức mà thường được dùng cho việc tạo và cấu

trúc lại các thành phần như văn bản, hình ảnh, liên kết hay các yếu tố khác của một trang web.

Các thành phần chính của HTML thường bao gồm các thẻ hay tags đi theo từng cặp và phần nội dung được chèn chính giữa các cặp thẻ. Mỗi thẻ sẽ có những tác dụng riêng biệt, tất cả các thể gộp lại sẽ giúp xây dựng nên một cấu trúc hoàn chỉnh cho Website. Các thẻ tập hợp lại tạo nên một cấu trúc hình cây bao gồm section, paragraph, heading, và các khối nội dung khác nữa.

- **Css** (hay Cascading Style Sheets) là một ngôn ngữ được dùng để tìm kiếm và định dạng lại các thành phần được HTML tạo ra. Ngôn ngữ này sinh ra để hình thành phong cách riêng biệt cho một trang HTML giúp thay đổi bố cục, màu sắc, font chữ, …

Phương thức hoạt động của Css thường dựa vào việc tìm và áp dụng các thuộc tính riêng biệt lên các vùng được chọn của một trang HTML. Trong khi HTML định hình một nền tảng của trang web thì Css sẽ tạo nên một phong cách riêng biệt.

Cấu trúc của một đoạn Css sẽ thường là các thành phần nối tiếp như bộ chọn, khai báo, thuộc tính và giá trị thuộc tính. Mỗi thành phần thực hiện một chức năng riêng biệt.

- Cuối cùng **JavaScript** là một ngôn ngữ lập trình đa năng và hướng đối tượng (OEP), thường được sử dụng để thêm tính tương tác và động vào các trang web. Nó cùng với HTML và CSS là ba công nghệ chính yếu và cốt lõi trong việc phát triển một trang web hoàn chỉnh web.

Trong khi HTML là nền tảng cấu thành, Css định hình phong cách 1 trang web thì Javascript làm cho trang web trở nên sinh động và có thể tương tác một cách trực quan với người dùng. Nó được xem là bộ não của một trang web bên cạnh khung xương (HTML) và quần áo (Css).

2.3 Môi trường phát triển ứng dụng

2.3.1 Node.js

Node.js là một runtime environment mã nguồn mở và đa nền tảng cho phép các lập trình viên chạy JavaScript trên phía server và được xây dựng dựa trên công nghệ V8 Javascript engine. Điều này cho phép các nhà phát triển có thể xây dựng các ứng dụng phía back-end bằng cách dùng cùng một ngôn ngữ lập trình là JavaScript.

Node.js cung cấp một thư viện built-in tức có nghĩa là nó cho phép các ứng dụng hoạt động như một Webserver mà không cần phần mềm hỗ trợ như Nginx, Apache HTTP Server hoặc IIS.

Các đặc điểm chính của Node.js có thể kể ra như sau:

- Node.js hoạt động dựa trên một luồng duy nhất và sử dụng các mô hình vòng lặp sự kiện dùng cho việc xử lý các tác vụ không đồng bộ. Có nghĩa là khi thực hiện một tác vụ như gọi API, Node.js sẽ tiếp tục thực hiện các tác vụ khác nhau mà không nhất thiết phải đợi các tác vụ trước đó hoàn thành.
- Node.js sử dụng cơ chế nhập/xuất không đồng bộ hay còn gọi là Non-blocking Input/Output. Cơ chế này cho phép lập trình viên thực hiện gọi API hay truy xuất dữ liệu mà không cần buộc dừng chương trình đang thực thi.
- Cơ chế Cross-platform cho phép Node.js có thể hoạt động trên nhiều nền tảng khác nhau kể cả windows, macos hay linux.
- Node.js thường sẽ đi kèm với NPM, đây là một hệ thống quản lý các gói JavaScript lớn nhất thế giới với hàng triệu thư viện hỗ trợ.

- Ưu điểm của Node.js:
  - Node.js hỗ trợ lập trình viên xử lý tốt các tác vụ không đồng bộ, đặc biệt là các tác vụ thời gian thực như Online game, streaming hay thực hiện video Call. Dữ liệu gần như được cập nhật ngay lập tức.
  - Lập trình viên không cần tạo thêm các luồng mới cho mỗi kết nối nhờ đó tiết kiệm được khá nhiều tài nguyên hệ thống.

- Môi trường này còn cung cấp thêm khả năng mở rộng ứng dụng bằng việc hỗ trợ thêm các instance hoặc sử dụng một kỹ thuật gọi là cluster hay còn được biết đến là việc chạy nhiều luồng trên nhiều CPU.

- Ngoài các ưu điểm kể trên thì Node.js vẫn còn tồn tại một số nhược điểm như:
  - Môi trường này không phù hợp cho việc chạy các tác vụ nặng về CPU. Nguyên nhân là do Node.js chỉ chạy trên một luồng, vì vậy các tác vụ yêu cầu tính toán phức tạp như xử lý một lượng lớn dữ liệu sẽ làm chậm lại toàn bộ ứng dụng. Cách khắc phục nhược điểm này là sử dụng worker threads – một module cho phép các đoạn mã Javascipt có thể chạy song song với luồng chính. Mỗi worker có thể chạy độc lập nhưng vẫn giao tiếp được với nhau bằng cách gọi hàm postMessage (). Khi thực hiện, luồng chính sẽ gửi một yêu cầu đến workers, các workers sau khi thực hiện xong nhiệm vụ sẽ gửi dữ liệu đến luồng chính thông qua hàm postMessage () từ đó luồng chính tiếp tục xử lý đoạn dữ liệu được cung cấp.
  - Việc xử lý quá nhiều các tác vụ không đồng bộ cũng dẫn đến hiện tượng có tên là “callback hell” có nghĩa là có quá nhiều hàm chồng lấn lên nhau. Có hai cơ chế được sinh ra để giải quyết vấn đề này là Promise và Async/await. Đầu tiên về Promise, thì đây là cơ chế cho phép lập trình viên đính kèm các trình xử lý (.then(), .catch() và .finally()) để xử lý một hoạt động thường là 2 trạng thái resolve hay reject. Promise cho phép quản lý các tác vụ bất đồng bộ mà gặp tình trạng "callback hell". Còn về Async/await thì đây là một cú pháp được thiết kế dựa trên promise. Đầu tiên async sẽ được khai báo trước trong một hàm và hàm đó sẽ trả về promise, await dùng để chờ một Promise hoàn thành (resolve hoặc reject) trước khi tiếp tục các tác vụ khác.

2.3.2 MongoDB

MongoDB là một dạng cơ sở dữ liệu hướng tài liệu (hay document-oriented) hay một dạng NoSQL dùng để lưu trữ dữ liệu dưới dạng JSON-like hay BSON (Binary JSON). MongoDB được sử dụng phổ biến ngày nay nhờ vào việc nó đem lại cho lập trình viên tính linh hoạt, hiệu suất cao cùng khả năng mở rộng hiệu quả.

Không như những cơ sở dữ liệu khác thường lưu trữ dữ liệu dưới dạng bảng và hàng thì mongoDB lưu trữ dưới dạng tài liệu trong các file JSON, điều này giúp người dùng có thể lưu trữ các loại dữ liệu phi cấu trúc hay dữ liệu thường xuyên thay đổi.

MongoDB còn đem lại khả năng truy vấn dữ liệu một cách linh hoạt và nhanh chóng. Các toán tử như $eq, $gt, $regex thường được dùng để truy vấn dữ liệu theo nhiều điều kiện khác nhau do người dùng quyết định. Ngoài ra còn hỗ trợ nhiều tính năng khác như tìm kiếm toàn bộ văn bản, theo các nhu cầu phức tạp hoặc xử lý các dữ liệu lớn.

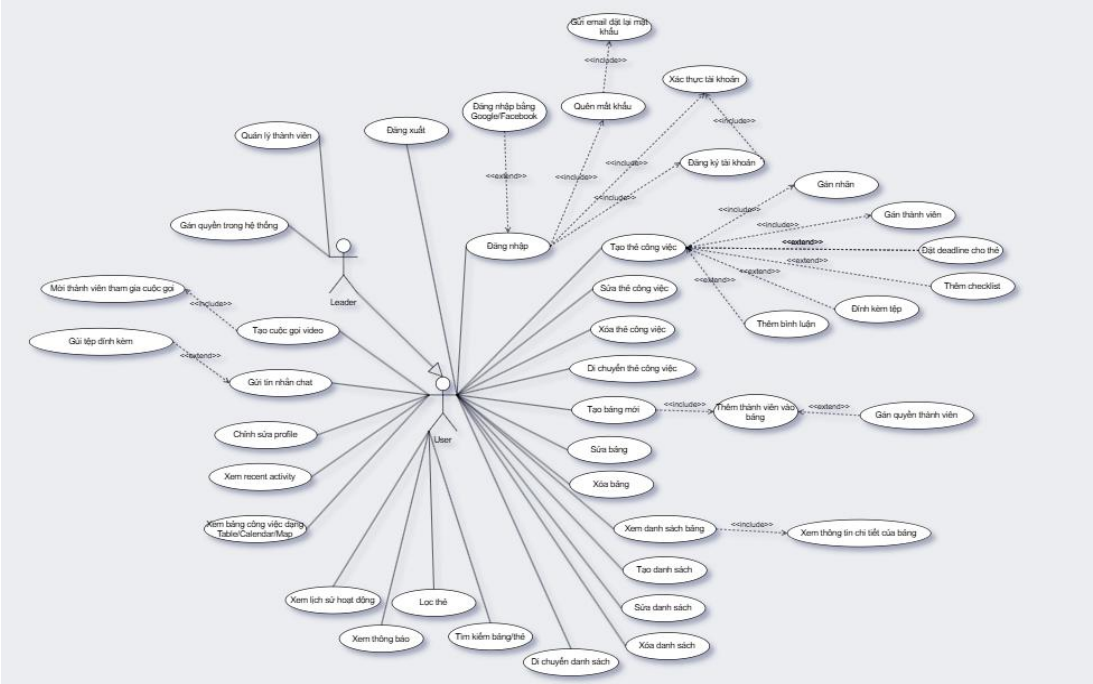
Một tính năng khác của mongoDB là khả năng mở rộng. Với Aggregation Framework hỗ trợ tính năng sharding hay còn gọi là phân mảnh dữ liệu, người dùng có thể chia nhỏ dữ liệu và lưu vào các máy chủ khác nhau nhờ đó hiệu suất sẽ được cải thiện. Ngoài ra còn một số chức năng khác như nhân bản, sao chép hay khôi phục dữ liệu.

Cuối cùng MongoDB tương thích tốt với hầu hết các ngôn ngữ lập trình hiện nay như Java, Python, C# hay Javascript. Người dùng còn có thể triển khai và quản lý cơ sở dữ liệu một cách dễ dàng với MongoDB Atlas với nhiều tính năng sao lưu, bảo mật và mở rộng linh hoạt.

CHƯƠNG 3: THIẾT KẾ VÀ PHÂN TÍCH SƠ ĐỒ HỆ THỐNG

3.1 Sơ đồ Usecase

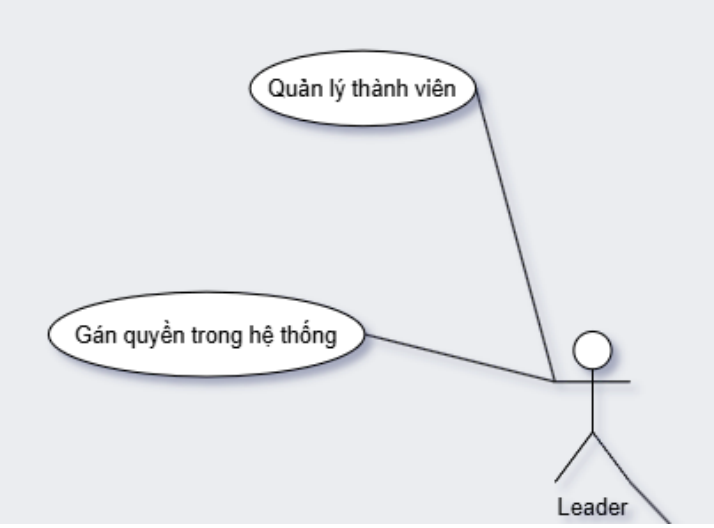
3.1.1 Sơ đồ Usecase tổng thể của hệ thống



Hình 1: Sơ đồ Usecase tổng thể của hệ thống

Sơ đồ Usecase mô tả chức năng tổng thể của toàn bộ hệ thống với 2 vai trò chính là Leader và User. Trong đó Leader có vai trò cao hơn, có quyền quản lý thành viên và kiểm soát cũng như tổ chức hệ thống. Trong khi đó User tức người dùng người dùng sẽ hoạt động dưới quyền chủ nhóm hay leader và được thực hiện các chức năng trong bảng và chỉnh sửa các công việc được phân công. Sơ đồ Usecase cung cấp cái nhìn tổng thể về toàn bộ hệ thống, các tác vụ và chức năng của một người dùng có thể thực hiện. Hệ thống hỗ trợ sự hợp tác, tổ chức công việc theo nhóm, và quản lý dự án một cách linh hoạt.

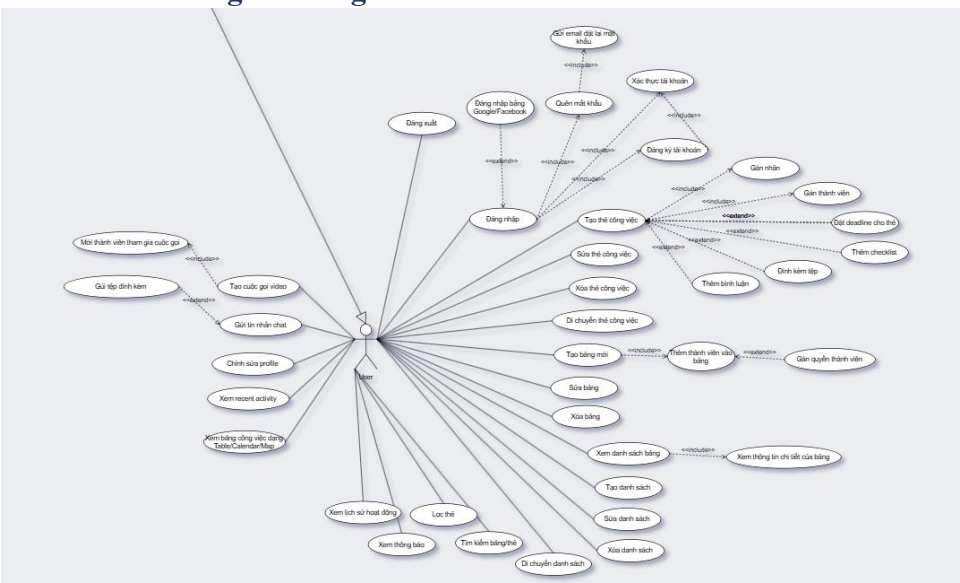
3.1.2 Sơ đồ Usercase của Leader



Hình 2: Sơ đồ Usecase của Leader

Leader là người có quyền cao nhất trong một nhóm hay một dự án, vai trò của leader không chỉ quản lý mà còn phải đảm bảo thành viên thực hiện đúng tiến độ công việc. Leader có 2 chức năng chính đó là quản lý danh sách các thành viên như thêm xóa sửa thành viên và chức năng thứ 2 là gán quyền xem hay sửa cho từng thành viên.

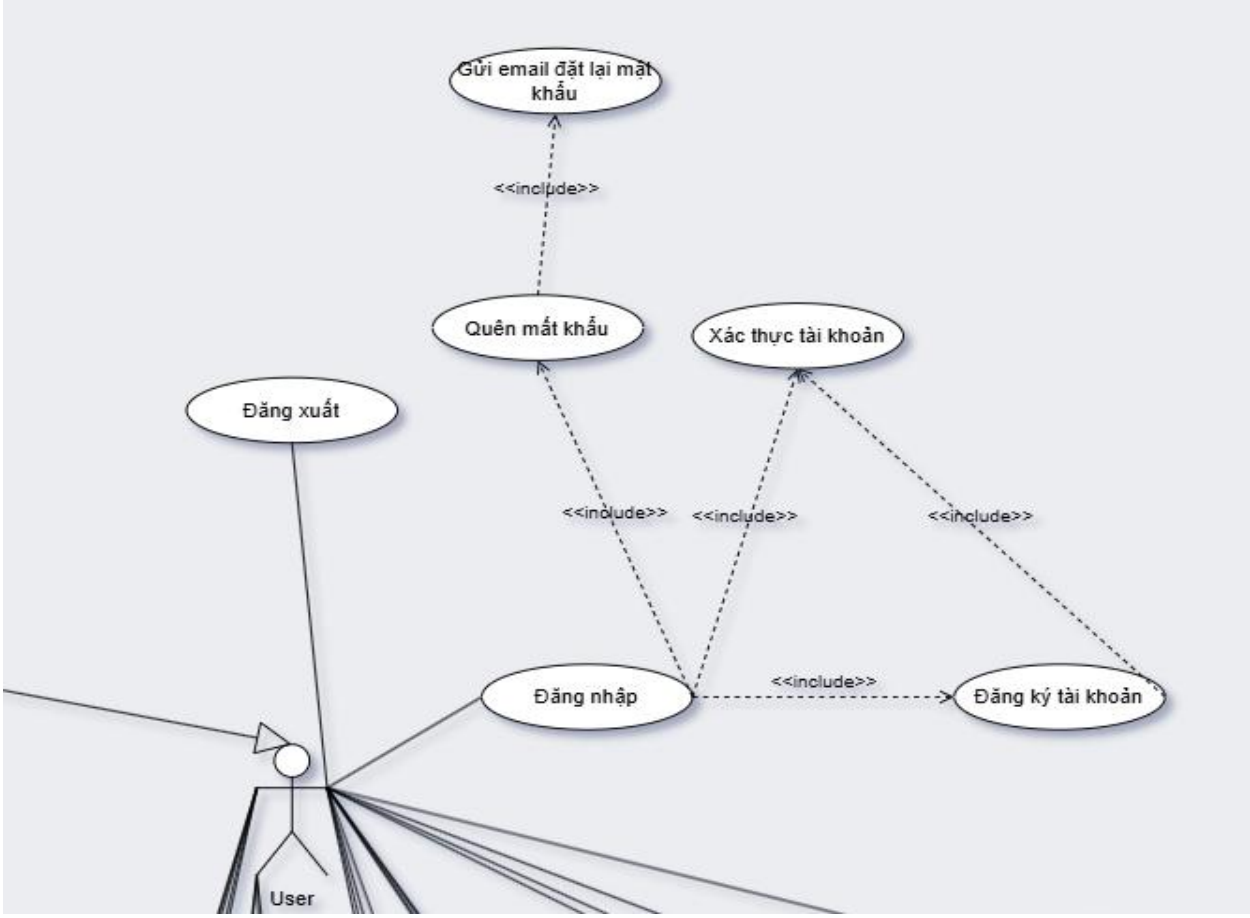
3.1.3 Sơ đồ Usecase của người dùng



Hình 3: Sơ đồ Usecase của User

Ở hệ thống này người dùng thực hiện hầu hết các thao tác mà web có thể thực hiện được khác với việc quản lý của Leader.

3.1.3.1 Chức năng đăng nhập, đăng xuất

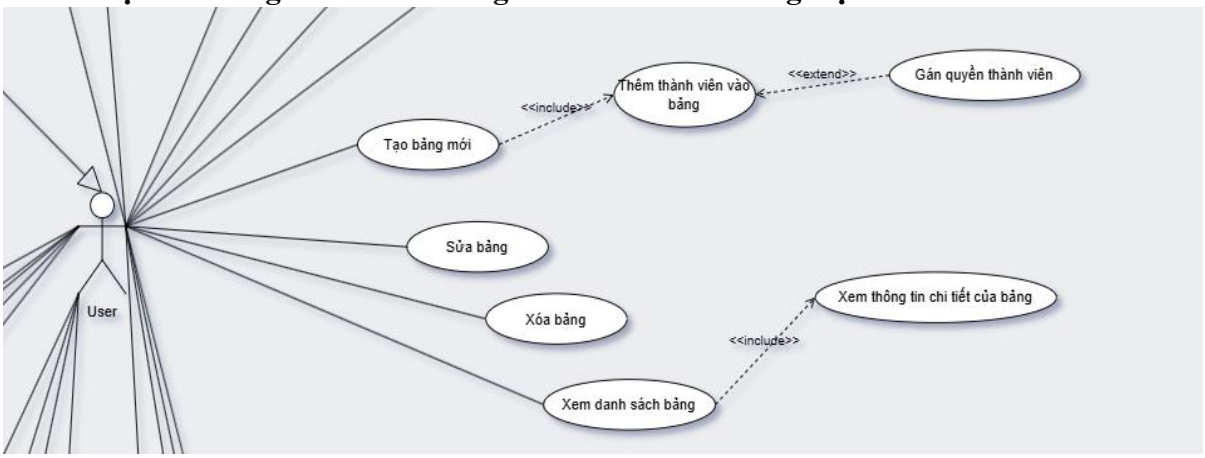


Hình 4: Sơ đồ Usecase chức năng đăng nhập đăng ký

Sơ đồ này mô tả chức năng đăng nhập và đăng xuất của User, cụ thể như sau:

- Người dùng được phép lựa chọn đăng nhập hoặc đăng ký khi tiến hành vào trang web. Khi lựa chọn chức năng đăng ký người dùng sẽ thực hiện nhập các thông tin như tên đăng nhập, số điện thoại, email, … Sau đó một đoạn mã xác nhận sẽ được gửi vào gmail. Sau khi người dùng xác nhận, tài khoảng đã được đăng ký thành công hệ thống sẽ trả về cửa sổ đăng nhập.
- Chức năng quên mật khẩu được sinh ra để người dùng có thể lấy đặt lại mật khẩu của mình trong trường hợp họ không nhớ.
- Sau khi hoàn thành công việc người dùng có thể chọn đăng suất để kết thúc công việc.

3.1.3.2 Tạo và tương tác với các bảng chứa danh sách công việc



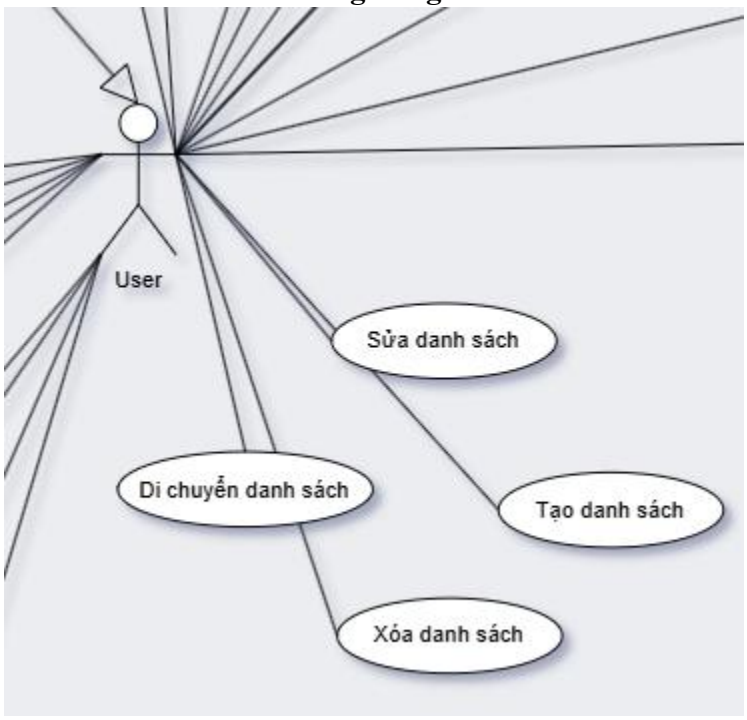
Hình 5: Sơ đồ Usecase board

Cấu tạo chủ yếu của website Ptollo là tập hợp gồm nhiều bảng khác nhau. Các bảng này đại diện cho một dự án mà Leader tổ chức hay phân công.

Trong các bảng này người dùng, leader hay thành viên của dự án được thực hiện một số thao tác liên quan đến bảng sau:

- Tạo mới các bảng, thường công việc này sẽ là leader thực hiện tuy nhiên ở các dự án có quy mô nhỏ thì người tạo bảng cũng chính là thành viên của dự án.
- Người tạo bảng được phép thêm thành viên cũng như gán quyền để cho phép các thành viên khác được phép thực hiện một số công việc cụ thể trong bảng hay trong dự án.
- Ngoài ra người dùng còn được phép chỉnh sửa các bảng, thêm background, đổi tên bảng cũng như xóa bỏ các bảng không dùng đến.
- Cuối cùng người dùng còn được phép xem danh sách cũng như thông tin chi tiết của các bảng đã tạo.

3.1.3.3 Tạo và tương tác với các danh sách trong bảng



Hình 6: Sơ đồ Usecase danh sách

List hay danh sách công việc là một dãy các công việc cụ thể và chi tiết mà người dùng hay leader thêm vào phân chia công việc cho các thành viên trong team hay công ty.

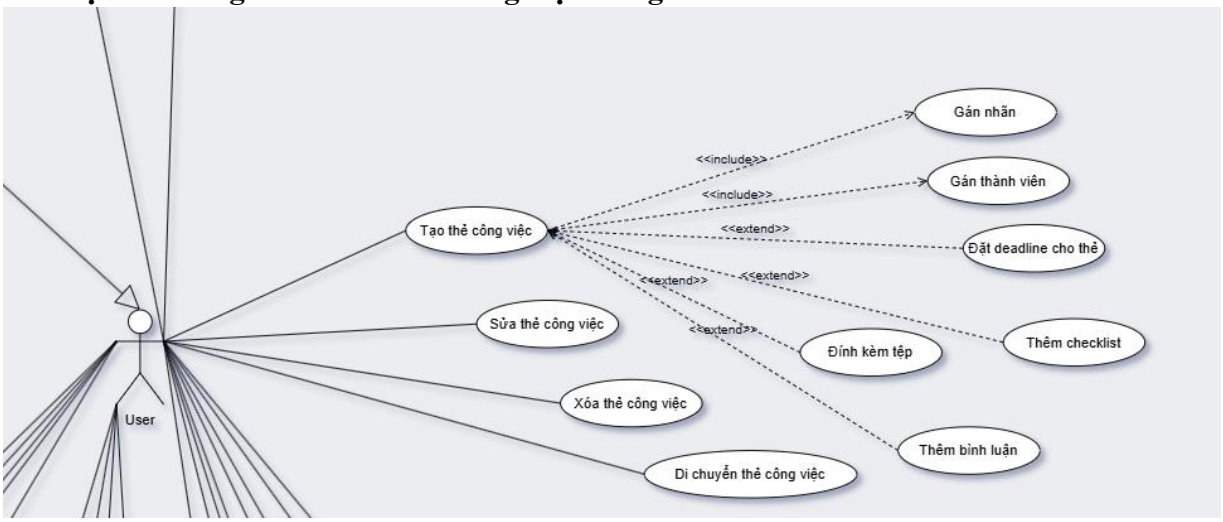
Việc tạo các danh sách giúp cụ thể hóa công việc và phân chia hạng mục chi tiết nhằm tối ưu hóa thời gian làm việc và dễ dàng quản lý cũng như phân chia.

Thực chất các danh sách này chính là các công việc hay hạng mục chính của dự án. Hệ thống sẽ dựa vào các danh sách này để tính tổng phần trăm dự án.

Trong các danh sách này, người dùng có thể thực hiện cách hành động như:

- Thêm, xóa hay đổi tên danh sách hay tên các công việc cụ thể.
- Hệ thống còn cho phép người dùng di chuyển các danh sách này để thay đổi thứ tự thực hiện các công việc.
- Ngoài các tính năng trên thì người dùng còn được phép để lại những nhận xét về công việc.
- Các danh sách công việc còn có thể được tùy biến bằng các thay đổi màu sắc khiến chúng trở nên sinh động.

3.1.3.4 Tạo và tương tác với các thẻ công việc trong danh sách



Hình 7: Sơ đồ Usecase thẻ công việc

Trong các danh sách, hệ thống còn cho phép người dùng tạo các thẻ công việc tương ứng với các công việc nhỏ hơn được chia trong mỗi hạng mục.

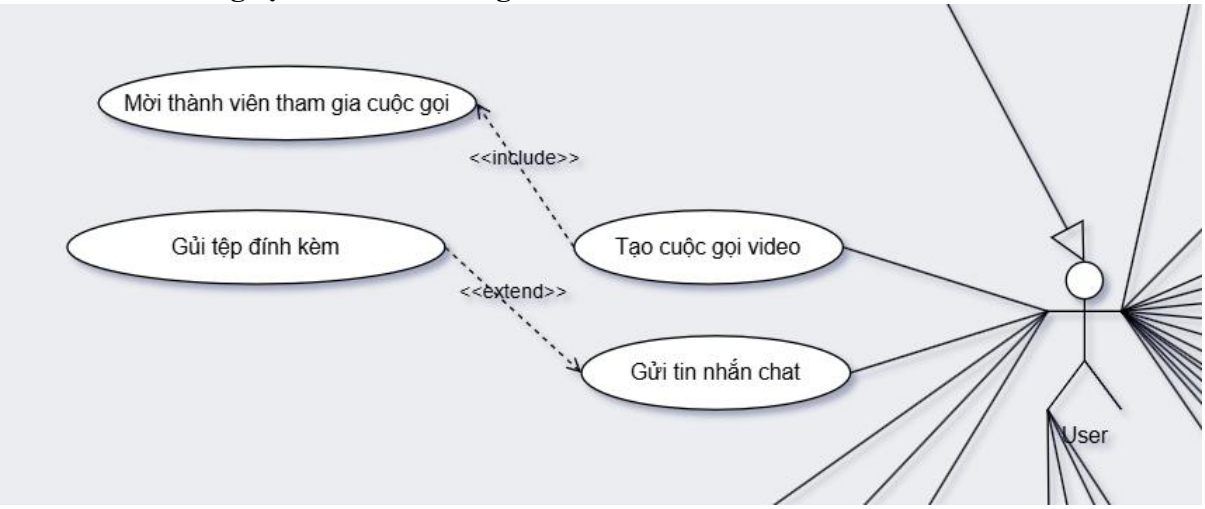
Việc chia nhỏ các hạng mục lớn trong một dự án thành các công việc nhỏ sẽ giúp ích rất nhiều trong việc phân công và quản lý công việc của từng nhân sự, đặc biệt là đối với các dự án lớn nhiều thành viên thực hiện.

Ở phần này, người dùng hầu như sẽ tương tác nhiều nhất, các tương tác mà người dùng có thể thực hiện với thẻ công việc sẽ là:

- Trong mỗi danh sách, người dùng đươc phép thêm, xóa, sửa hay di chuyển các thẻ công việc. Các thẻ công việc còn có thể được di chuyển từ danh sách này sang danh sách khác bằng cách kéo thả.
- Bên cạnh đó người dùng còn được phép gắn cho từng công việc một số tags để có thể dễ dàng nhận biết.
- Tất nhiên để có thể hoạt động như một nhóm hay công ty thì người dùng được phép mời thành viên để cùng thực hiện công việc. Chức năng thêm thành viên sẽ dựa vào việc mời các thành viên thông qua địa chỉ email.
- Deadline là một phần thiết yếu của một dự án. Leader hay người dùng sẽ có thể tạo và thay đổi deadline của từng công việc trong danh sách. Khi hoàn thành một việc nào đó người dùng sẽ đánh dấu hoàn thành. Hệ thống sẽ tổng hợp lại và đưa ra bảng phân tích số liệu liên quan đến phần trăm hoàn thành công việc.

- Ngoài ra các thành viên trong team sẽ được phép để lại comment để thông báo hay bày tỏ ý kiến về công việc được giao.
- Chức năng đính kèm tệp sẽ cho phép người dùng đăng tải và nộp các file như tài liệu, hình ảnh của dự án vào từng danh sách.

3.1.3.5 Chức năng tạo Video Call và gửi tin nhắn



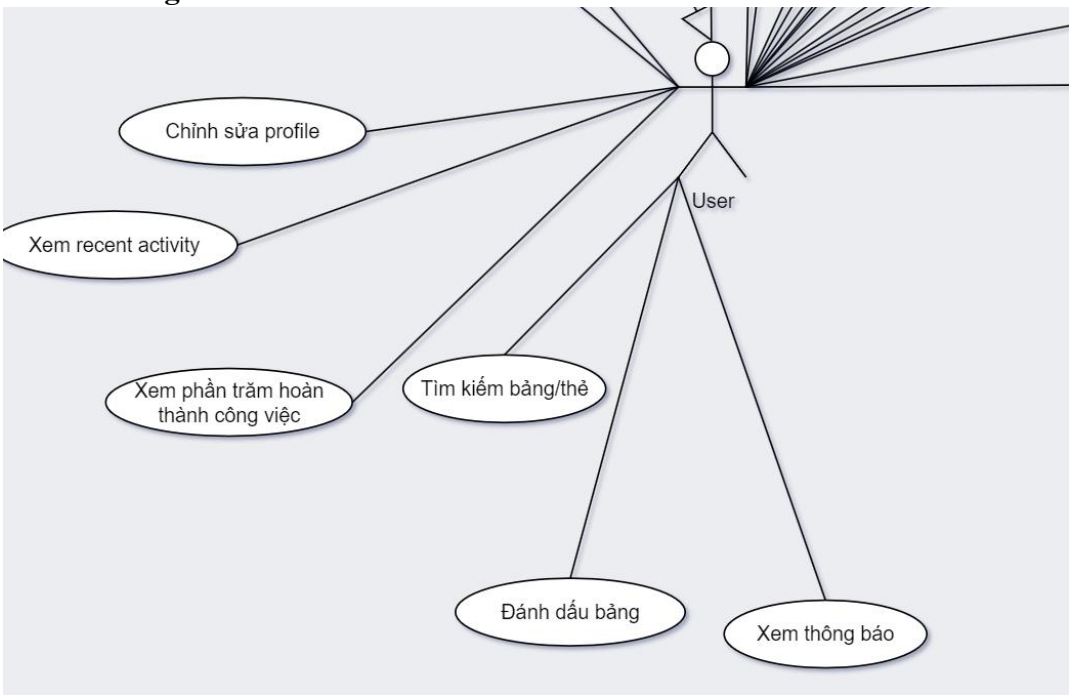
Hình 8: Sơ đồ Usecase cuộc gọi và tin nhắn

Khi hoạt động hay tham gia trong một nhóm hoặc một công ty thì khả năng liên lạc và trao đổi giữa các thành viên là hết sức quan trọng.

Vì vậy chức năng tạo video call hay một chatbox trong một bảng để team có thể liên lạc với nhau được tạo ra:

- Đầu tiên là chức năng video call, khi người dùng tiến hành tạo một cuộc gọi nhóm thì một đoạn mã sẽ được gửi đến các thành viên trong team, những người đồng ý tham gia cuộc gọi sẽ nhập đoạn mã được gửi vào ô sau khi nhập đoạn mã vào thì các thành viên sẽ tham gia vào được cuộc gọi.
- Chức năng chat box cho phép các thành viên giao tiếp với nhau bằng cách gửi các đoạn tin nhắn văn bản trong trường hợp không tiện thực hiện videocall. Tính năng này tương tự như ứng dụng Messenger hay Zalo.

3.1.3.6 Các chức năng khác



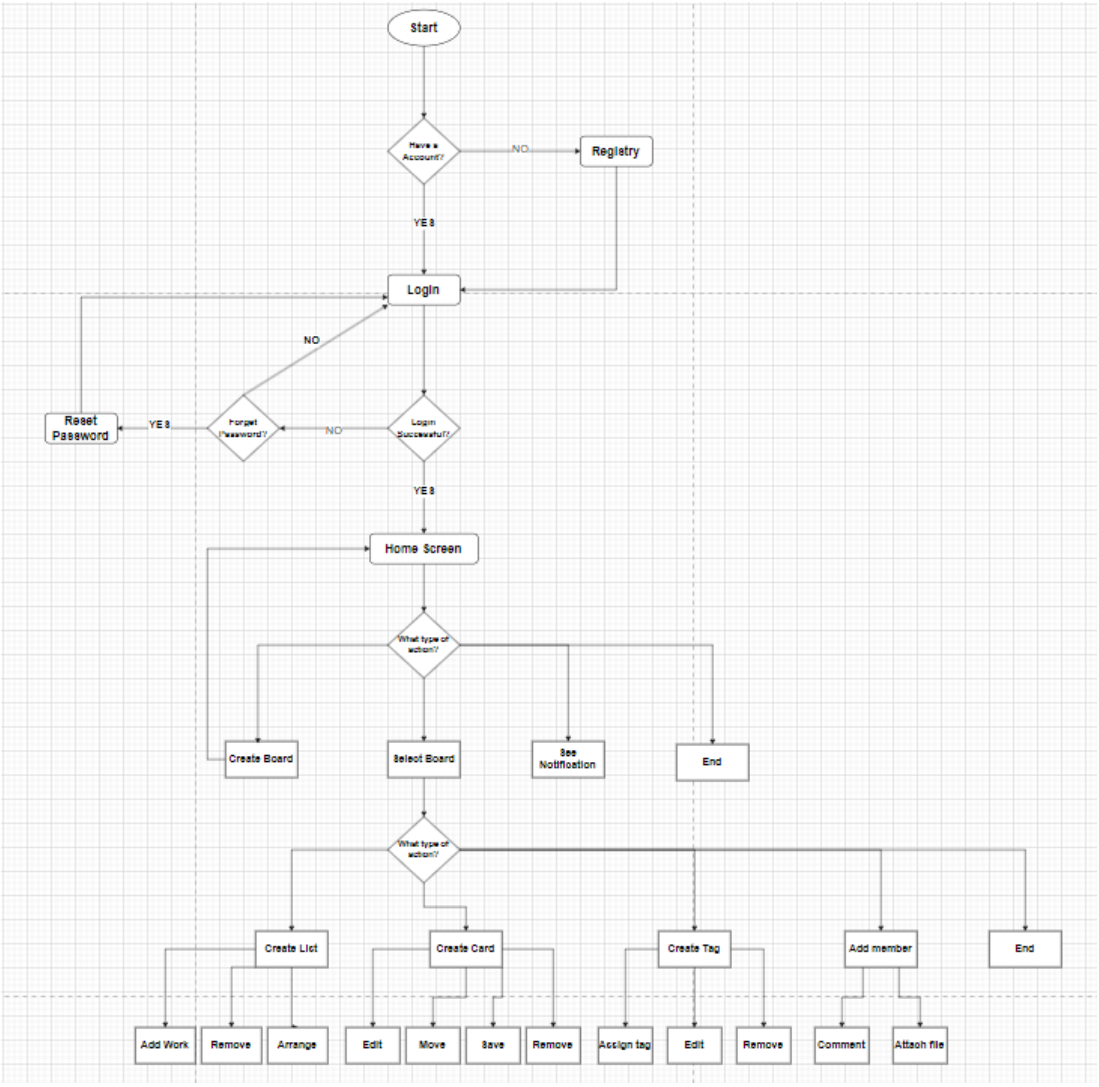
Hình 9: Sơ đồ Usecase các chức năng khác

Bên cạnh các tính năng cốt lõi của ứng dụng người dùng còn có thể xem thực hiện được một số tính năng khác có thể kể đến như:

- Tính năng chỉnh sửa một số thông số trong profile như tên, địa chỉ email hay ảnh đại diện.
- Xem các board thường được truy cập gần đây, tính năng này cho phép người dùng truy cập một cách nhanh chóng các dự án thường xuyên được thực hiện.
- Tính năng tìm kiếm cho phép người dùng tìm được các dự án hay công việc cụ thể trong dự án một cách nhanh chóng, chức năng này thực sự hữu ích với những người dùng tham gia nhiều dự án khác nhau.
- Xem thông báo là một tính năng hữu ích để người dùng không bị trễ hay bỏ lỡ các thông tin quan trọng trong quá trình thực hiện dự dán, chức nay này cũng cần thiết khi thực hiện các cuộc gọi nhóm. Khi tiến hành gọi thì các thông báo sẽ được gửi đến từng người trong dự án.
- Cuối cùng một tính năng quan trọng khác của website là xem được phần trăm công việc. Tính năng này giúp người dùng đánh giá mức độ hoàn thành dự án hay các công việc nhỏ trong dự án.

3.2 Sơ đồ hoạt động

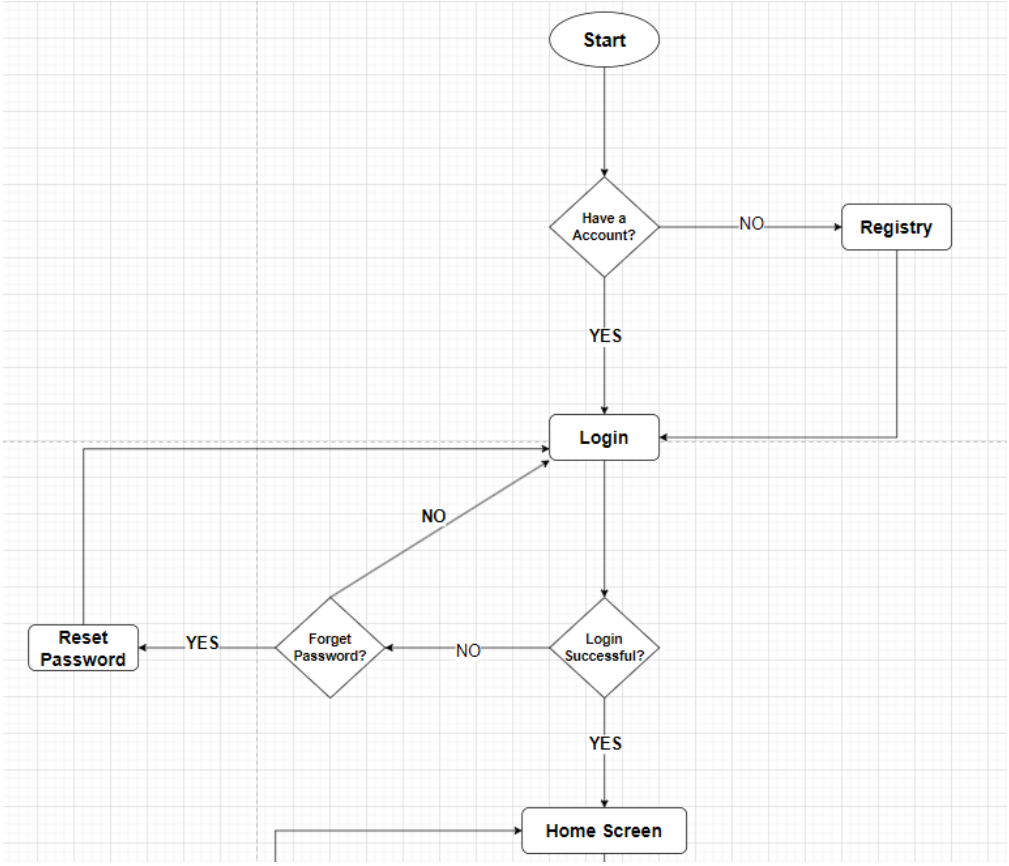
3.2.1 Sơ đồ hoạt động của tổng thể hệ thống



Hình 10: Sơ đồ hoạt động toàn bộ hệ thống

Sơ đồ hoạt động này mô tả khái quát quá trình mà người dùng có thể thực hiện tương tác với trang web bằng cách đưa ra các hoạt động mà một người dùng có thể đưa ra trong quá trình sử dụng ứng dụng.

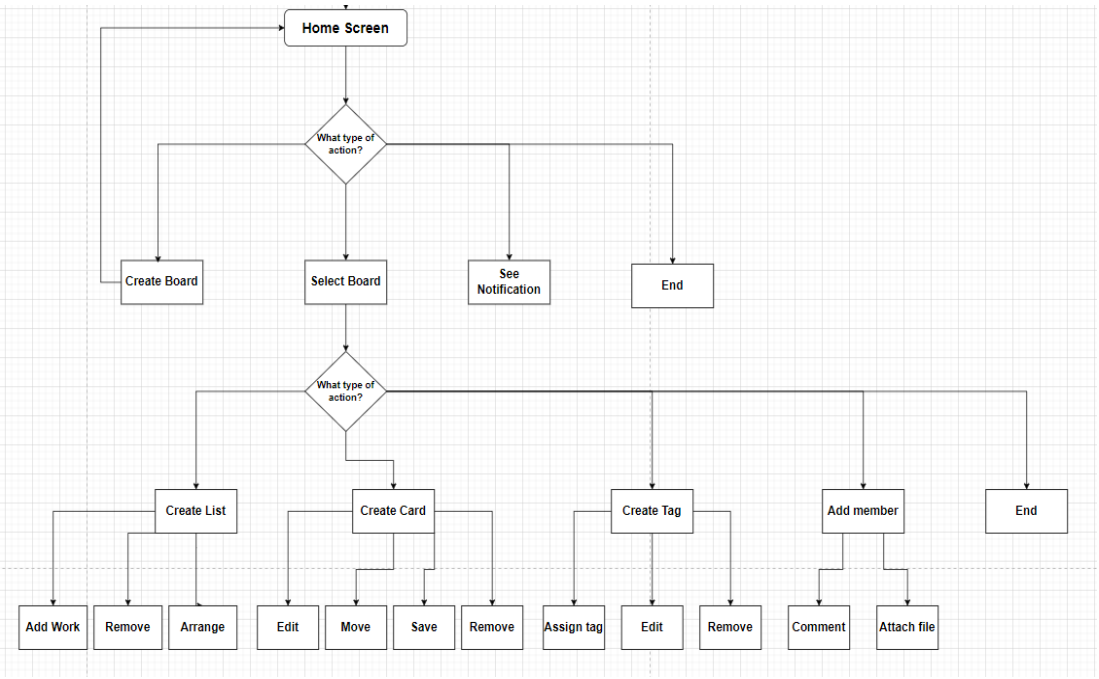
3.2.2 Phân tích chi tiết sơ đồ



Hình 11: Sơ đồ hoạt động chức năng đăng nhập, đăng ký

Phần đầu của sơ đồ mô tả quá trình xác thực người dùng cơ bản của hệ thống với thứ tự các bước thực hiện như sau:

- Đầu tiên khi bắt đầu chương trình sẽ xác nhận xem người dùng đã có tài khoản hay chưa, nếu đã có tài khoản người dùng sẽ tiến hành đăng nhập, ngược lại người dùng sẽ được di chuyển đến trang đăng ký tài khoản.
- Tại trang đăng nhập, hệ thống sẽ xác nhận đăng nhập đã thành công hay chưa, nếu quá trình thành công, người dùng sẽ di chuyển thẳng đến màn hình chính của website.
- Ngược lại hệ thống sẽ cho người dùng nhập lại hoặc chọn quên mật khẩu, nếu người dùng đồng ý họ sẽ được dẫn tới trang khôi phục mật khẩu, sau khi khôi phục sau họ sẽ tiến hành đến trang đăng nhập để thực hiện lại việc đăng nhập vào hệ thống.



Hình 12: Sơ đồ hoạt động sau khi đăng nhập

Tại trang màn hình chính, người dùng sẽ tiến hành đưa ra lựa chọn hoạt động của mình, các hoạt động chính người dùng có thể chọn là tạo bảng hay dự án mới, chọn các bảng đã có sẵn, xem các thông báo hoặc chọn kết thúc mà không làm gì cả.

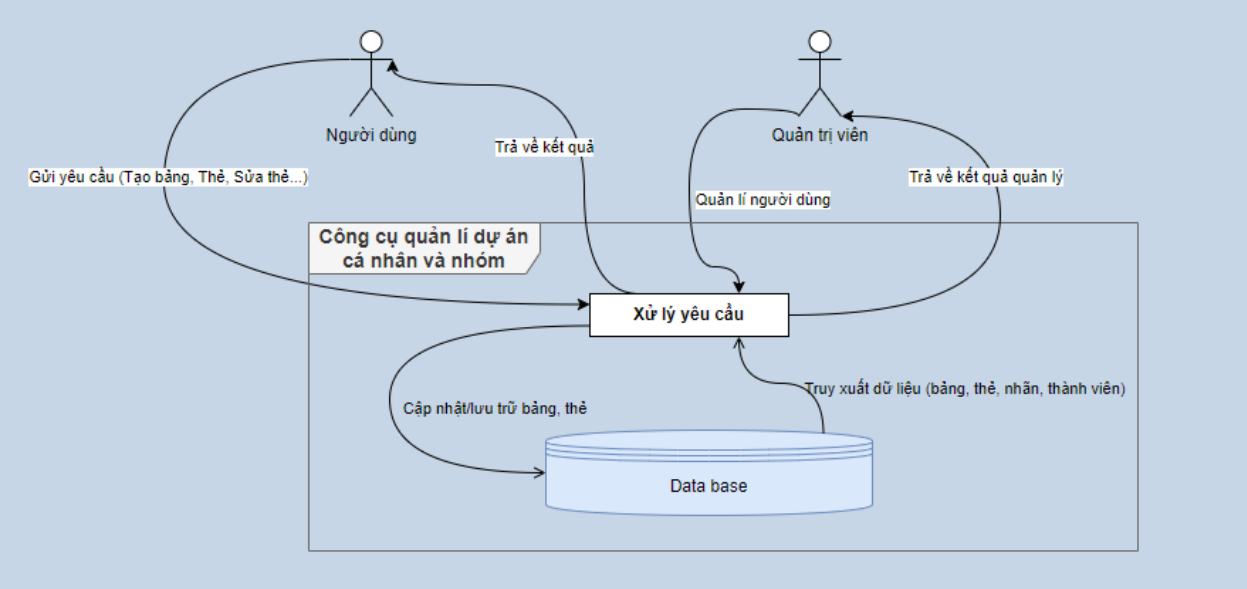
Khi thực hiện chức năng chọn bảng, người dùng có thể thực hiện các chức năng nhỏ khác có trong bảng:

- Tạo danh sách công việc: chức năng này cho phép người dùng tạo và quản lý danh sách công việc có trong dự án. Các chức năng có thể kể đến là thêm, xóa hay sắp xếp các công việc.
- Tạo các thẻ: đây đại diện cho các công việc nhỏ trong từng danh sách, tương tự người dùng có thể chỉnh sửa, di chuyển, lưu hay xóa các thẻ công việc này.
- Các nhãn giúp cho việc nhận diện các thẻ công việc được thực hiện một cách dễ dàng hơn, cũng như danh sách và thẻ, người dùng có thể chọn giữa việc gắn thẻ, chỉnh sửa hay xóa chúng đi.
- Cuối cùng người dùng còn có thể lựa chọn thêm thành viên cho dự án, thêm bình luận hay dính kèm các file cần thiết của dự án hay đơn giản không làm gì cả và kết thúc tiến trình.

3.3 Sơ đồ luồng dữ liệu (DFD – Data Flow Diagram)

Sơ đồ DFD (Data Flow Diagram) biểu diễn cách mà dữ liệu được sử dụng hay hướng đi của dữ liệu để phân tích và thiết kế hệ thống. Data Flow Diagram mô tả cách dữ liệu được truyền, xử lý hay lưu trữ trong một hệ thống cụ thể. Sơ đồ DFD thường được sử dụng trong các giai đoạn ban đầu của việc phát triển một hệ thống, ứng dụng hay website để hiểu rõ các luồng dữ liệu.

3.3.1 Sơ đồ luồng dữ liệu cấp 0 (DFD level 0)

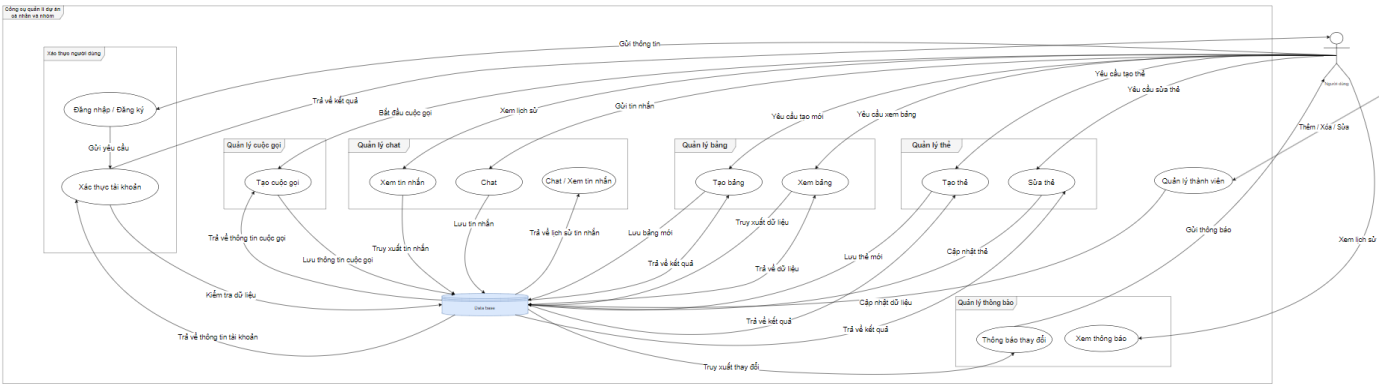


Hình 13: Sơ đồ luồng dữ liệu cấp 0

Có thể thấy sơ đồ này đem lại một cái nhìn tổng quát về hướng đi của ứng dụng như sau:

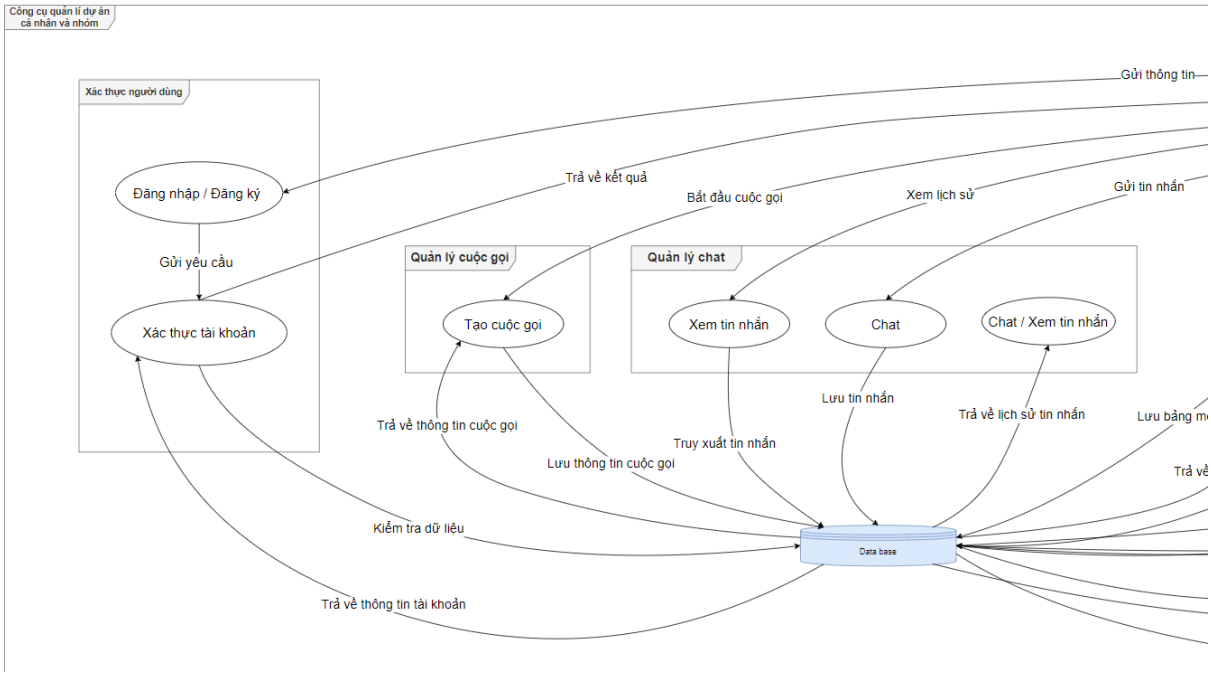
- Đầu tiên phía quản trị viên, khi một yêu cầu quản lý người dùng của quản trị viên được gửi đến, hệ thống sẽ xử lý yêu cầu và truy xuất thông tin về người dùng và gửi lại cho quản trị viên.
- Ở phía người dùng, họ sẽ gửi các yêu cầu trong quá trình sử dụng trang web như tạo bảng, thẻ hay chỉnh sửa các danh sách, … Sau đó hệ thống cũng sẽ xử lý các yêu cầu đó và cập nhật lại trên cơ sở dữ liệu. Kết quả sau khi được cập nhật sẽ được trả về phía người dùng.

3.3.2 Sơ đồ luồng dữ liệu cấp 1 (DFD level 1)



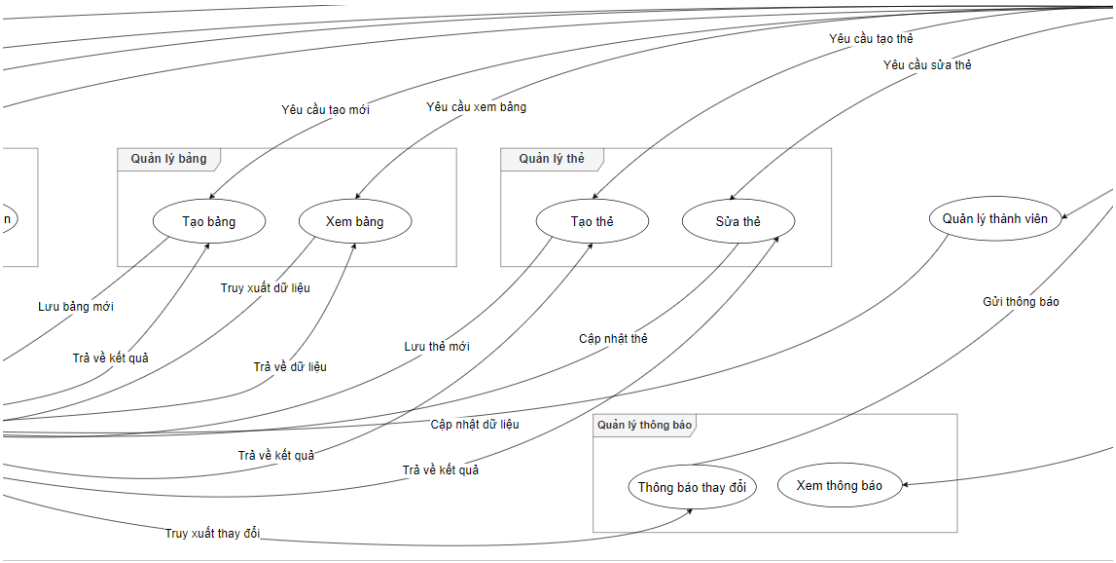
Hình 14: Sơ đồ luồng dữ liệu cấp 1

Sơ đồ này mô tả chi tiết hơn về đường đi của dữ liệu trong hệ thống, cụ thể như sau:



- Khi người dùng gửi thông tin yêu cầu ứng dụng xử lý các tác vụ đăng nhập đăng ký, hệ thống tiến hành xác thực tài khoản sau đó kiểm tra dữ liệu phía data base và trả về thông tin tài khoản cho người dùng.
- Tương tự khi người dùng muốn tạo một cuộc gọi, thông tin được gọi sẽ được hệ thống lưu vào database sau đó trả về hệ thống.
- Cơ chế quản lý chat cũng đưa ra đường đi của dữ liệu tương tự, dữ liệu từ các yêu cầu của người dùng như lưu tin nhắn sẽ được lưu vào database, khi người

dùng muốn xem thì dữ liệu lại được hệ thống truy suất ngược từ database đến người dùng.



Hình 15: Sơ đồ chi tiết luồng dữ liệu cấp 1

- Cơ chế di chuyển của dữ liệu của các tác vụ như quản lý thẻ, bảng hay quản lý thông báo của người dùng với hệ thống cũng cơ phần giống với các chức năng trên.
- Người dùng tiến hành yêu cầu tạo bảng, danh sách hay thẻ mới, hệ thống tiến hành lưu các thành phần được tạo vào cơ sở dữ liệu, sau đó dữ liệu được gửi về lại hệ thống và hiển thị cho phía người dùng.
- Các tác vụ như chỉnh sữa hay xem bảng thì hệ thống sẽ cập nhật các dữ liệu đó vào database và rồi cũng lại hiển thị về cho người dùng.

3.4 Thiết kế cơ sở dữ liệu

3.4.1 Các bảng chính trong cơ sở dữ liệu

3.4.1.1 Bảng board

Bảng board là bảng lưu trữ các thông tin chính cấu thành nên thành phần của một bảng hay dự án.

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	INT	Khoá chính
2	user_id	INT	ID người dùng (khoá ngoại)
3	name	VARCHAR	Tên của từng bảng
4	created_date	DATETIME	Thời gian khởi tạo
5	is_public	BOOLEAN	Trạng thái công khai của bảng

Bảng 1: Cấu trúc dữ liệu bảng board

3.4.1.2 Bảng danh sách (List)

Bảng này lưu trữ các thông tin của danh sách công việc trong từng board.

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	INT	Khoá chính
2	board_id	INT	ID bảng (khoá ngoại)
3	name	VARCHAR	Tên danh sách công việc
4	position	INT	Vị trí danh sách trong bảng

Bảng 2: Cấu trúc dữ liệu bảng list

3.4.1.3 Bảng board_label

Bảng này lưu thông tin các nhãn hay tags của board.

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	INT	Khoá chính
2	board_id	INT	ID bảng (khoá ngoại)
3	name	VARCHAR	Tên nhãn
4	colour	VARCHAR	Vị trí danh sách trong bảng

Bảng 3: Cấu trúc dữ liệu bảng board_label

3.4.1.4 Bảng card

Bảng này chứa thông tin của các card hay thẻ công việc cụ thể có trong danh sách.

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	INT	Khoá chính
2	list_id	INT	ID danh sách (khoá ngoại)
3	name	VARCHAR	Tên của từng thẻ
4	description	TEXT	Mô tả chi tiết của thẻ
5	created_date	DATETIME	Thời gian khởi tạo
6	is_active	BOOLEAN	Trạng thái hoạt động của thẻ
7	due_date	DATETIME	Ngày hết hạn deadline

Bảng 4: Cấu trúc dữ liệu bảng card

3.4.1.5 Bảng đính kèm tệp trong card

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	INT	Khoá chính
2	card_id	INT	ID thẻ (khoá ngoại)
3	uploaded_date	DATETIME	Thời gian đăng tải tệp đính kèm
4	filename	VARCHAR	Tên file đính kèm
5	location	VARCHAR	Đường dẫn, vị trí file đính kèm

Bảng 5: Cấu trúc dữ liệu bảng card_attachment

3.4.1.6 Bảng hoạt động của thẻ

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	INT	Khoá chính
2	card_id	INT	ID thẻ (khoá ngoại)
3	user_id	DATETIME	ID người dùng (khóa ngoại)
4	activity	VARCHAR	Mô tả hoạt động cụ thể trên thẻ
5	created_date	VARCHAR	Thời gian hoạt động được thực hiện

Bảng 6: Cấu trúc dữ liệu bảng card_activities

3.4.1.7 Bảng thành viên (Member)

Lưu trữ các thông tin về những thành viên được thêm vào dự án

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	user_id	INT	ID người dùng (khóa ngoại)
2	board_id	INT	ID bảng (khoá ngoại)

Bảng 7: Cấu trúc dữ liệu bảng Member

3.4.1.8 Bảng bình luận (Comment)

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	INT	Khoá chính
2	comment	TEXT	Nội dung bình luận
3	created_date	DATETIME	Ngày giờ bình luận được tạo
4	card_id	INT	ID thẻ (khóa ngoại)
5	user_id	INT	ID người dùng (khóa ngoại)

Bảng 8: Cấu trúc dữ liệu bảng comment

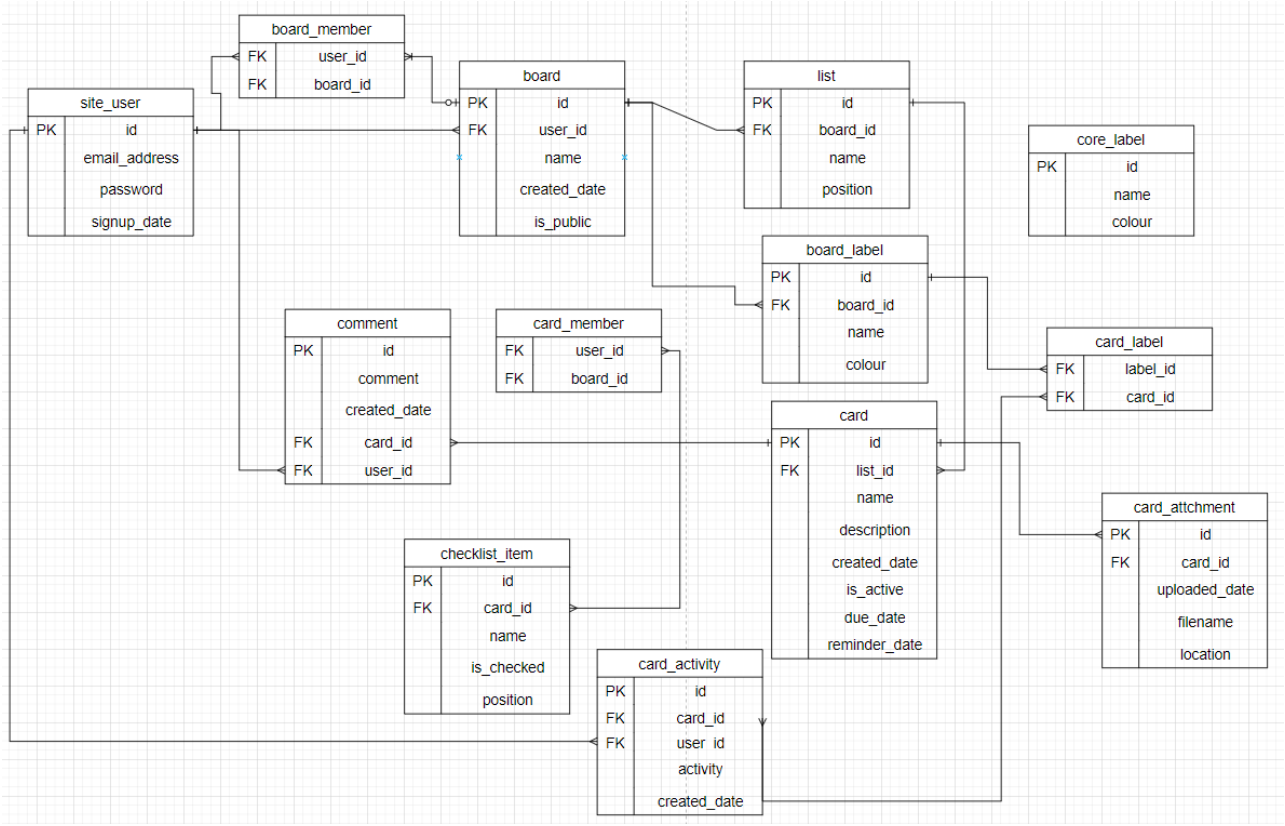
3.4.1.9 Bảng thông tin người dùng (Site_User)

Bảng này lưu trữ tất cả thông tin cần thiết về người dùng

STT	Tên trường	Kiểu dữ liệu	Mô tả
1	id	INT	Khoá chính
2	email_address	VARCHAR	Địa chỉ email
3	password	VARCHAR	Mật khẩu người dùng
4	signup_date	DATETIME	Thời gian đăng ký tài khoản

Bảng 9: Cấu trúc dữ liệu bảng Site_User

3.4.2 Lược đồ quan hệ của các bảng

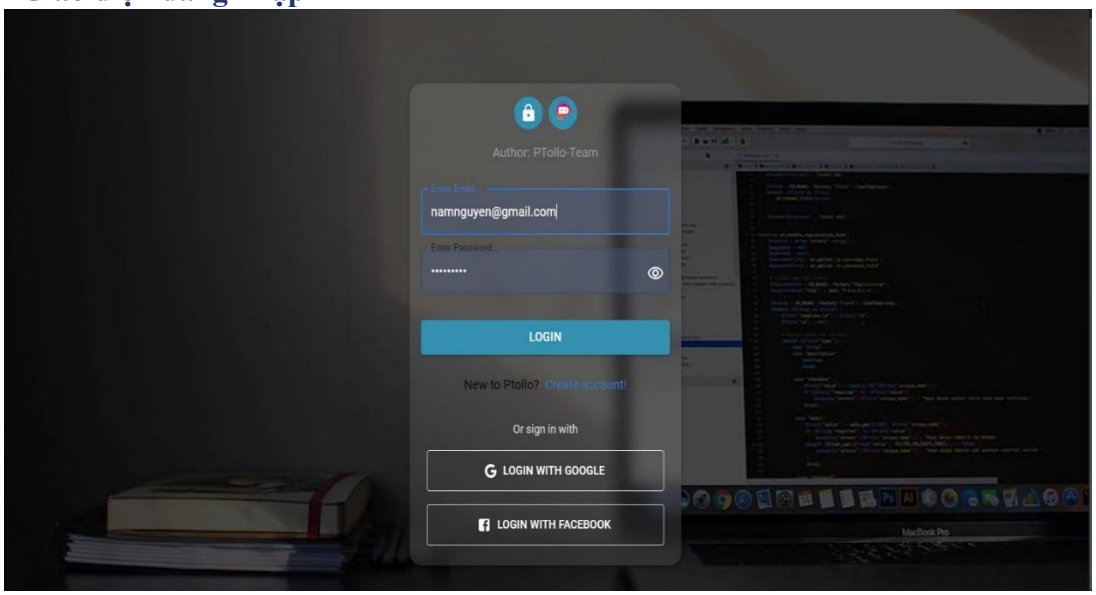


Hình 16: Lược đồ quan hệ của các bảng

CHƯƠNG 4: HIỆN THỰC HỆ THỐNG

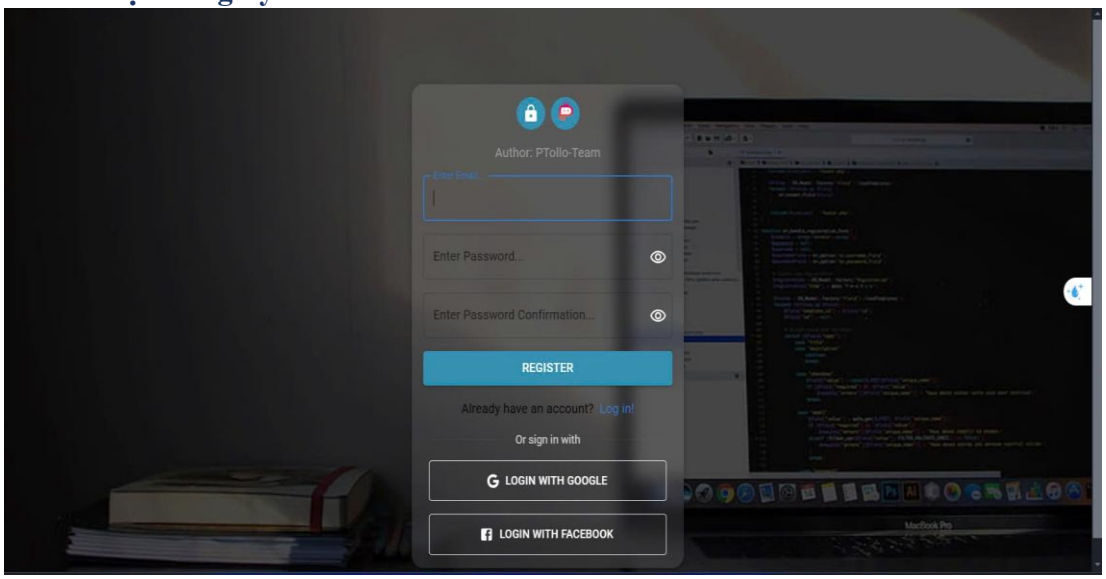
4.1 Giao diện một số trang chính

4.1.1 Giao diện đăng nhập



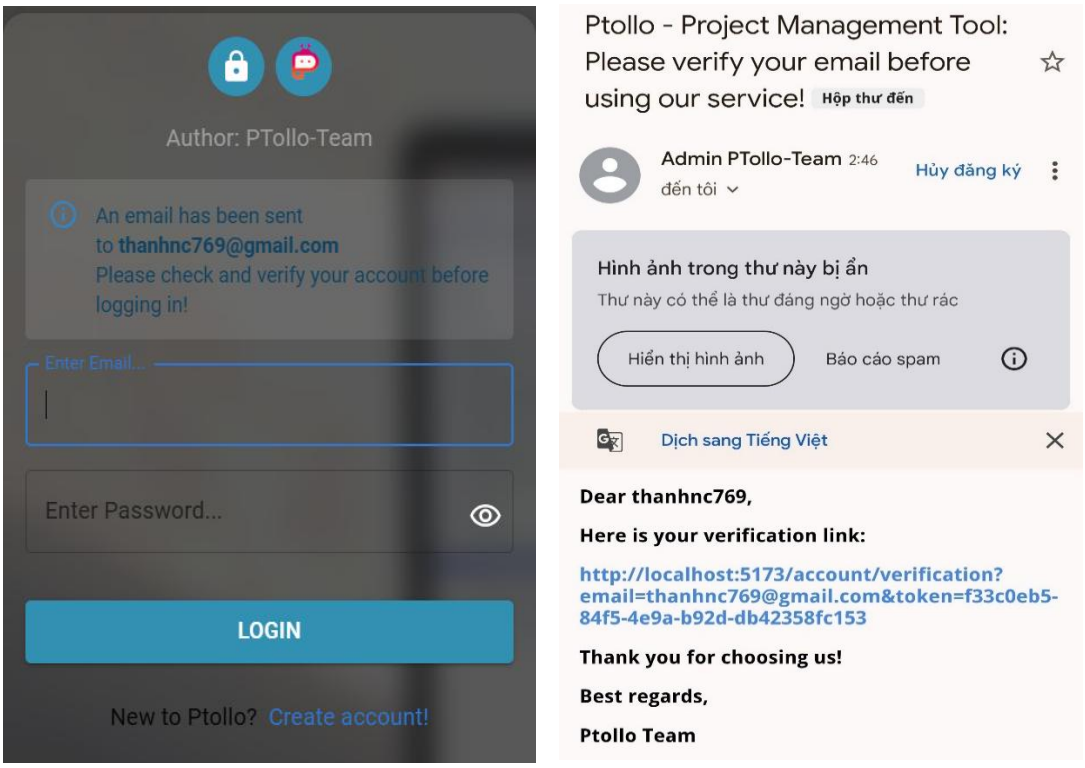
Hình 17: Giao diện đăng nhập

4.1.2 Giao diện đăng ký



Hình 18: Giao diện đăng ký

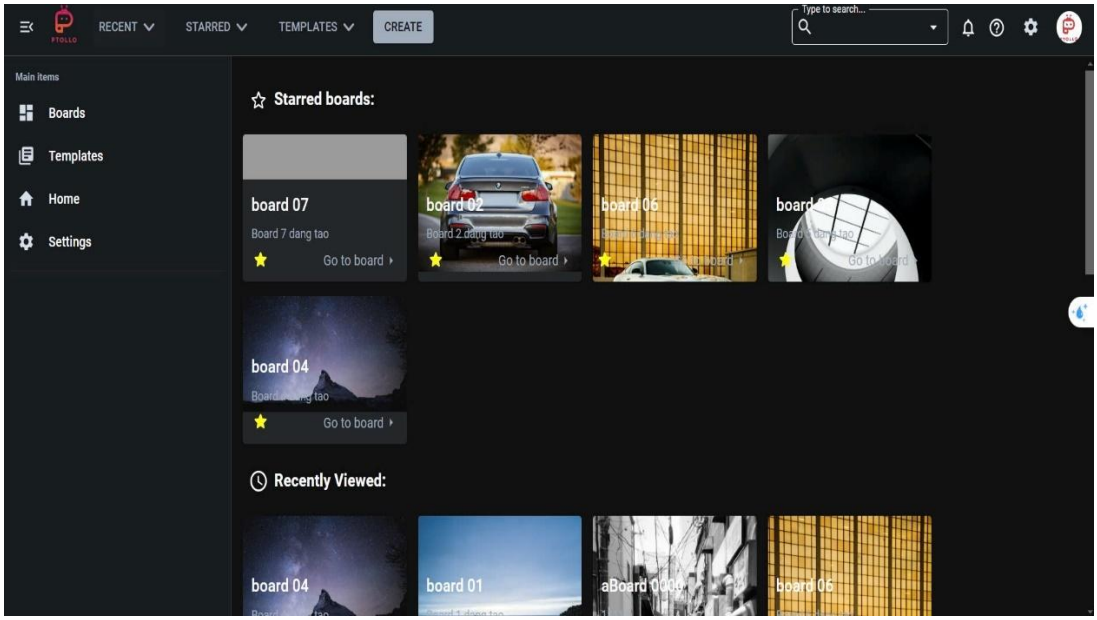
Sau khi nhập địa chỉ email, password và confirm password một email xác nhận được gửi đến người dùng, sau đó người dùng tiến hành xác nhận và đăng nhập vào hệ thống.



Hình 19: Xác nhận đăng ký

4.1.3 Màn hình chính của ứng dụng

Sau khi đăng nhập người dùng được đưa đến màn hình chính của ứng dụng.

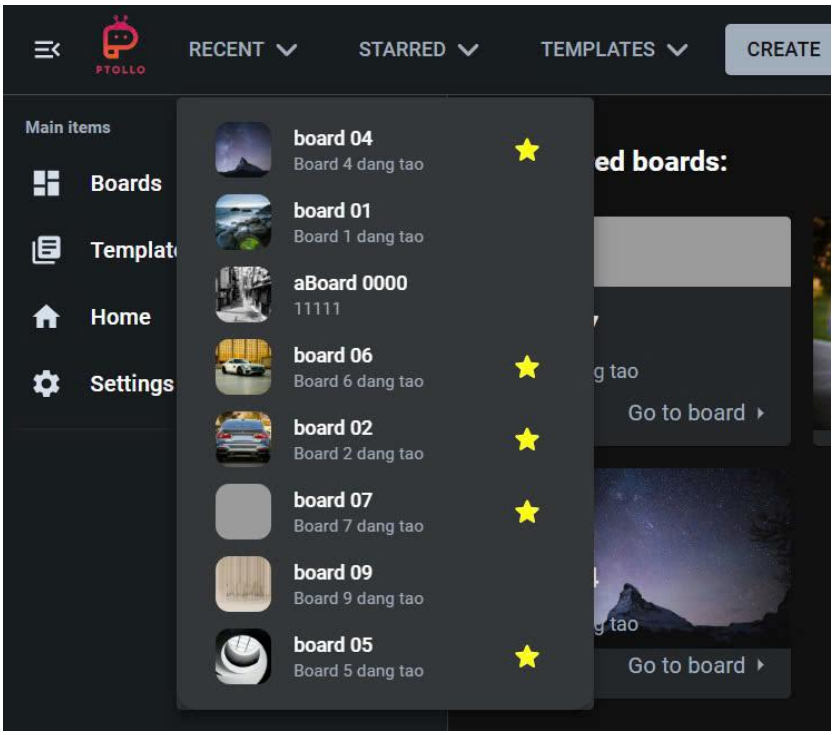


Hình 20: Màn hình chính website

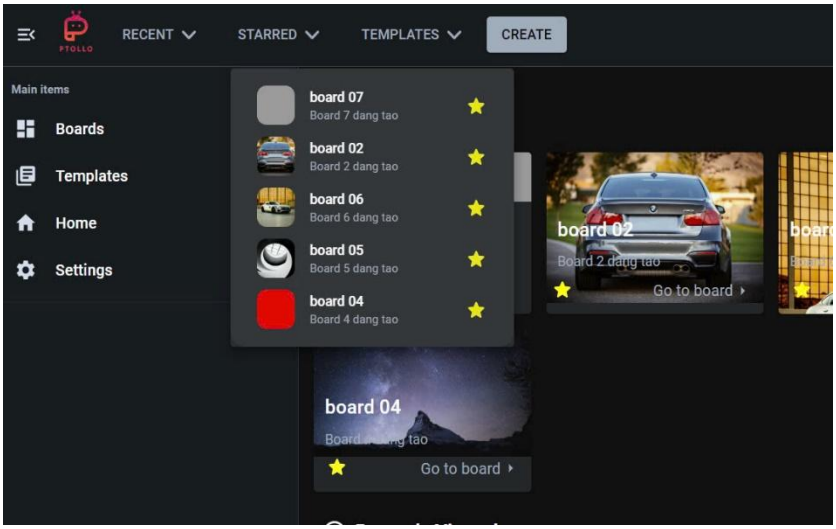
Màn hình chính sẽ hiển thị các bảng được đánh dấu hay được xem gần đây, bên trên là các thanh công cụ để xem thông tin chi tiết các trang riêng biệt và nút tạo bảng.

Góc phải trên có thanh công cụ giúp tìm kiếm nhanh các bảng, icon thông báo, cài đặt và xem profile. Bên phải là chức năng xem bảng, các template có sẵn, home và cài đặt.

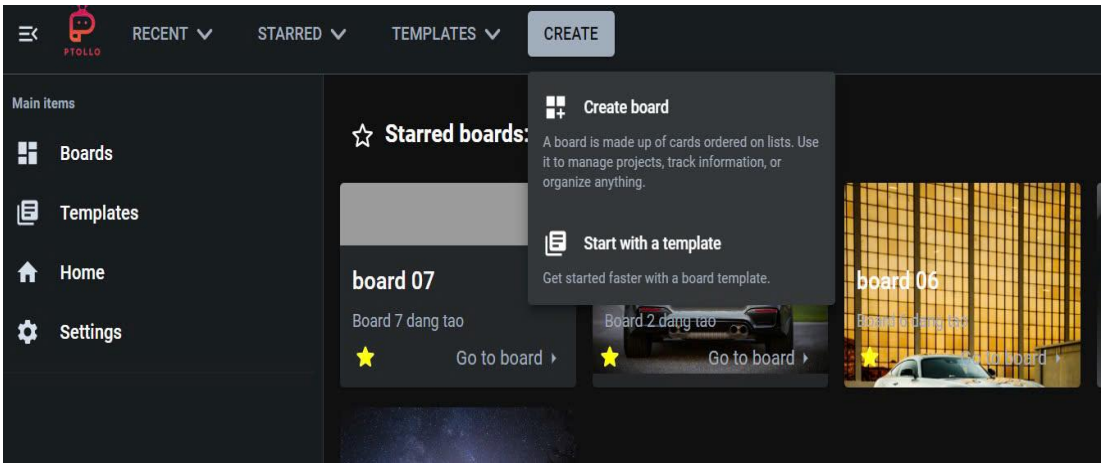
Khi nhấn vào Recent, một danh sách hiển thị nhanh các bảng được truy cập gần đây.



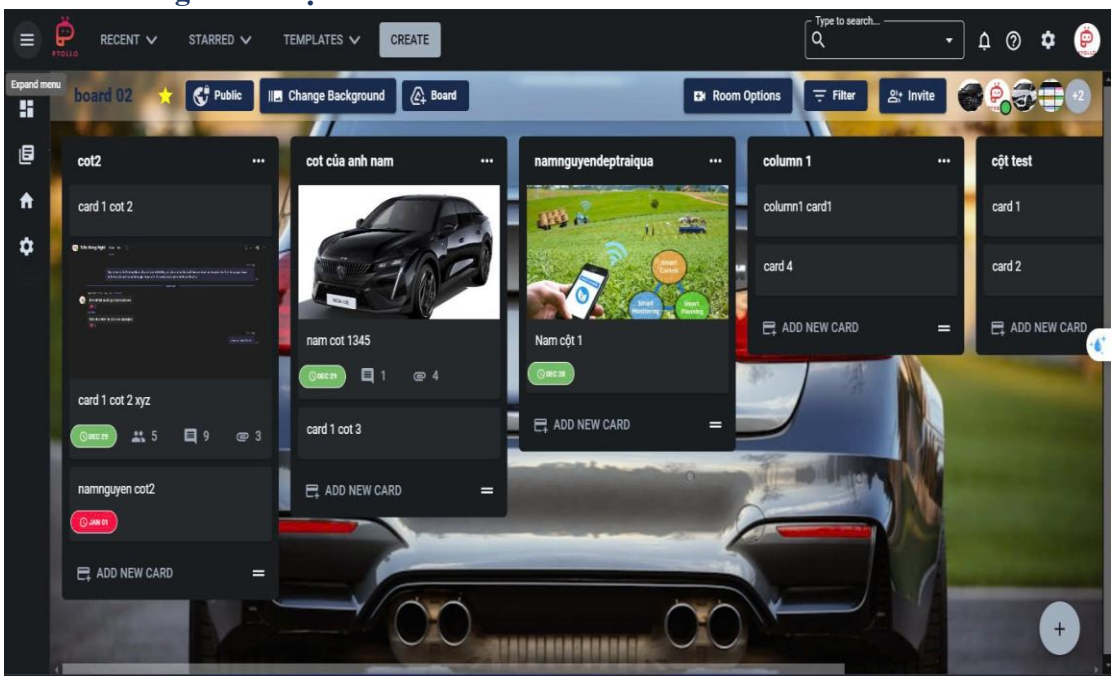
Khi nhấn vào Starred sẽ là danh sách các bảng quan trọng được đánh dấu sao.



Nhấn vào Create sẽ cho người dùng 2 lựa chọn là tạo bảng mới hoặc bắt đầu với template có sẵn.



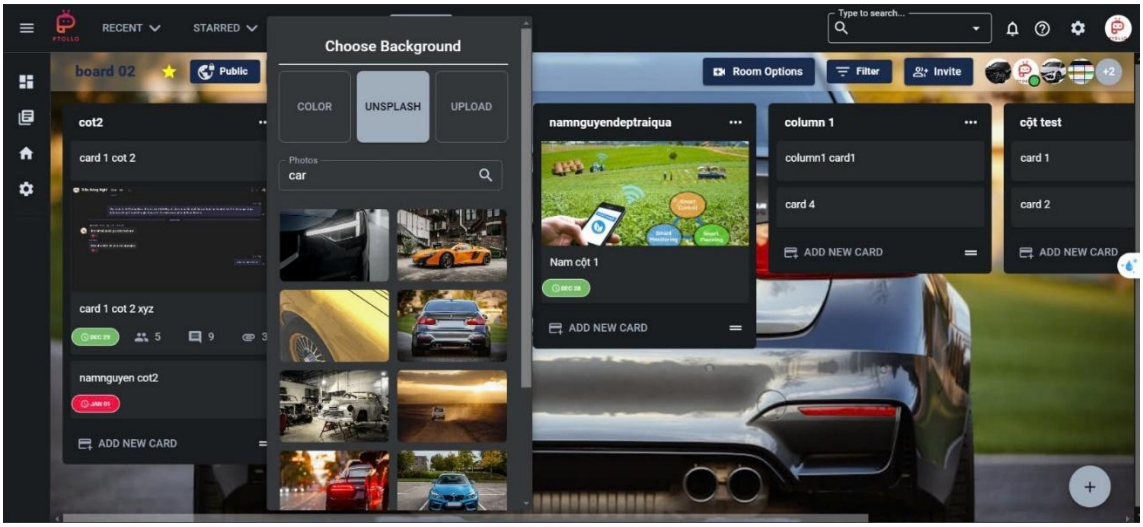
4.1.4 Màn hình từng board cụ thể



Hình 21: Giao diện trong board

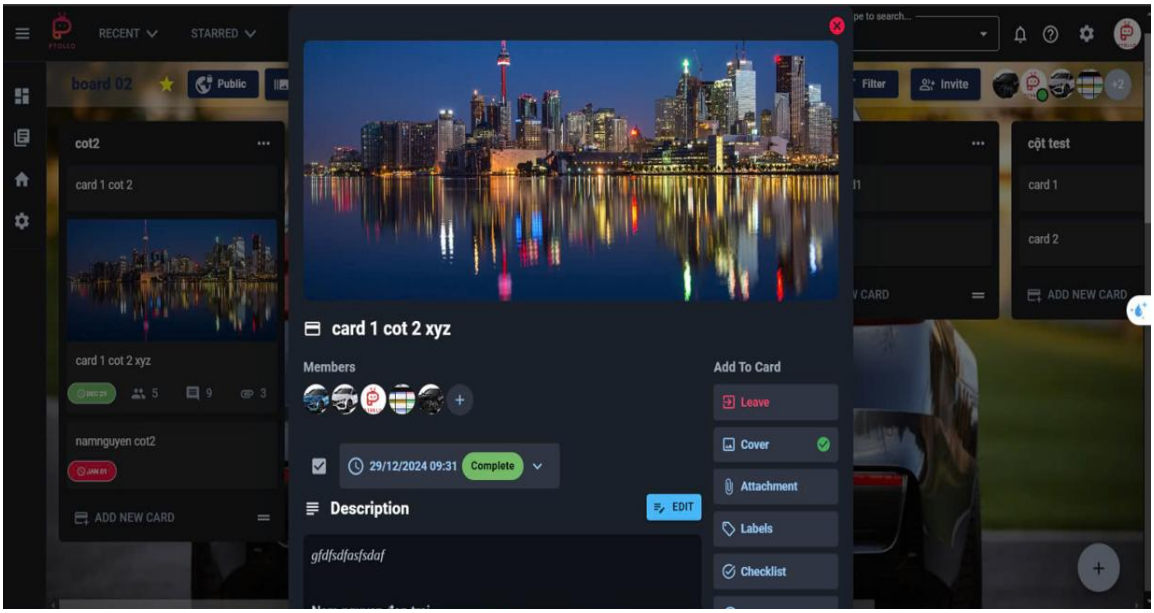
Màn hình được cấu tạo từ những list công việc chia theo dạng cột, trong các list công việc chứa các thẻ công việc, ta có thể thêm hình ảnh vào các thẻ, set deadline từng thẻ,…

Ngoài ra trên thanh công cụ của từng board, ta còn có thể set trạng thái của board, thay đổi background, tạo bảng mới, tạo cuộc họp video call hay mời thành viên khác vào bảng.



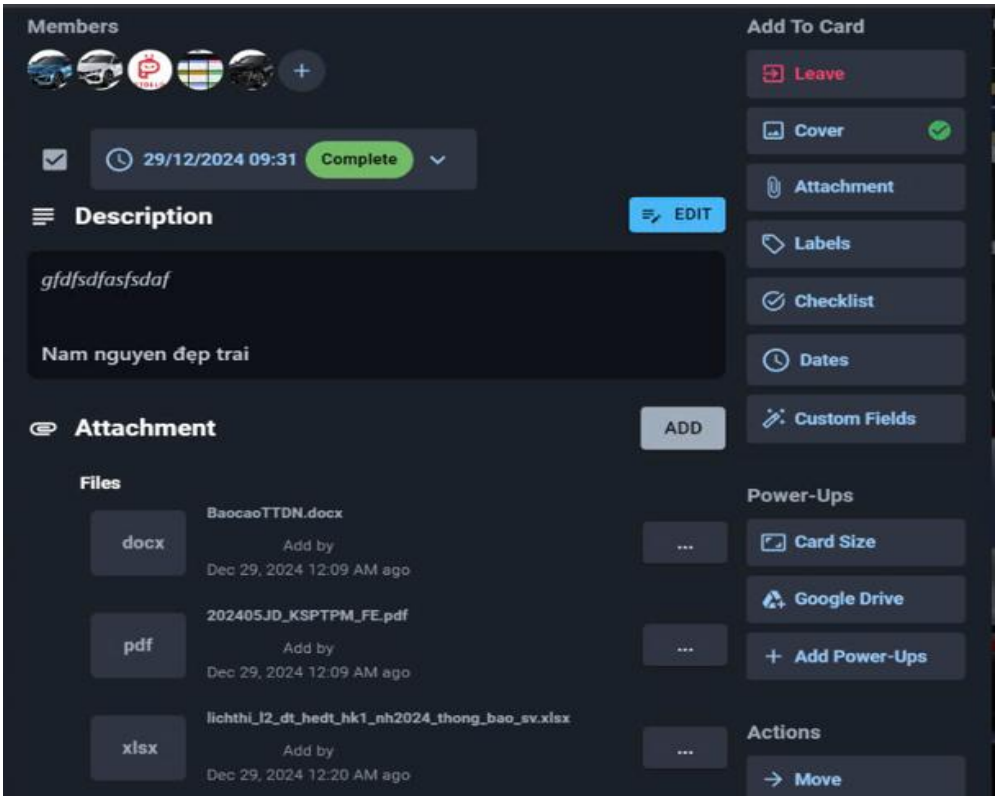
Chức năng thay đổi background theo màu, chủ đề tìm kiếm hay chọn ảnh từ thiết bị.

Tiếp theo là giao diện khi truy cập vào các thẻ công việc cụ thể.

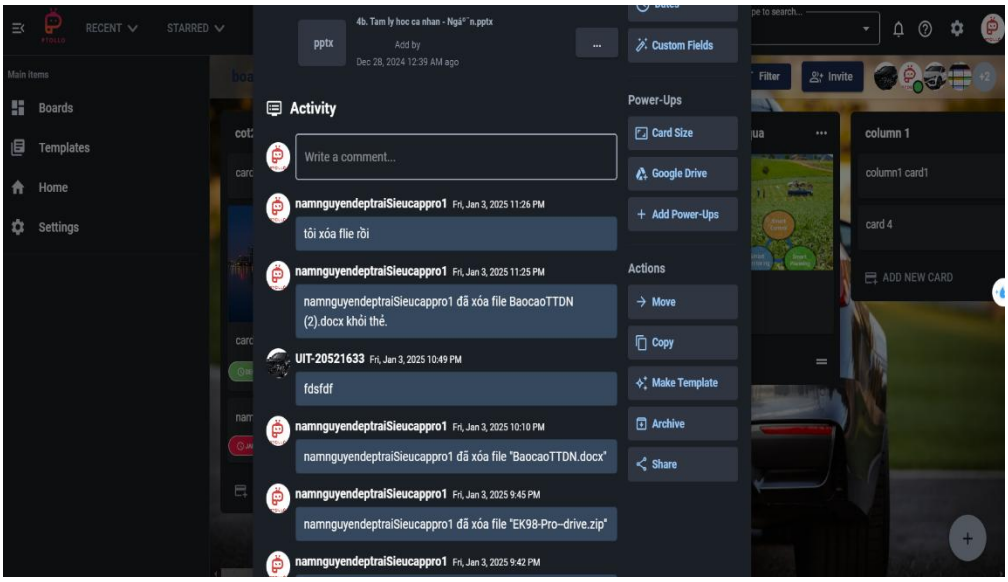


Hình 22: Giao diện trong thẻ

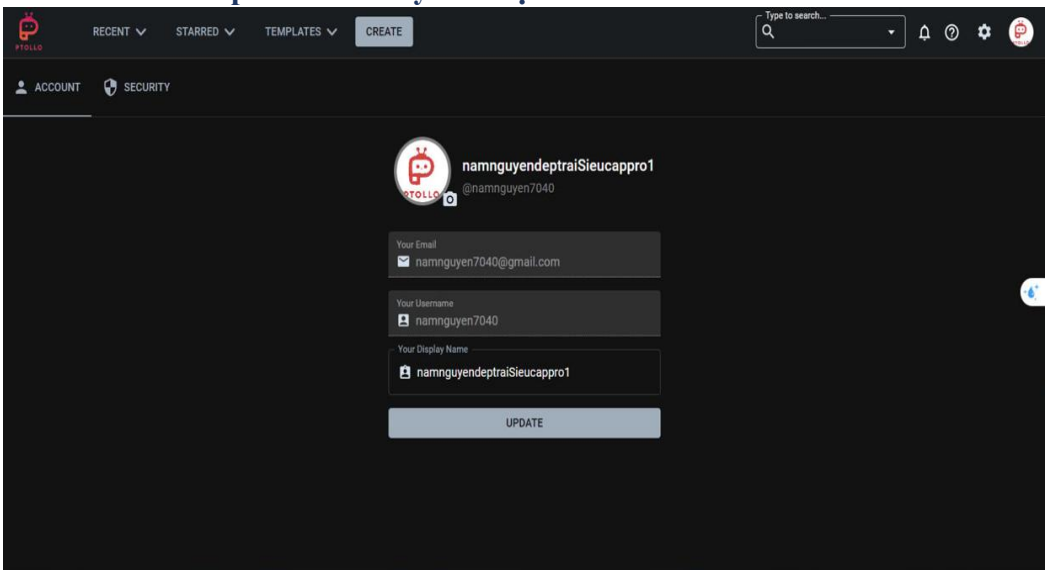
Ta có thể thêm background thẻ, thêm mô tả từng thẻ, set deadline công việc, gắn nhãn thẻ, đính kèm tập tin



Ngoài ra ta còn có thể tùy chỉnh kích cỡ của các thẻ, di chuyển hay thêm comment để đánh giá công hay trao đổi công việc.

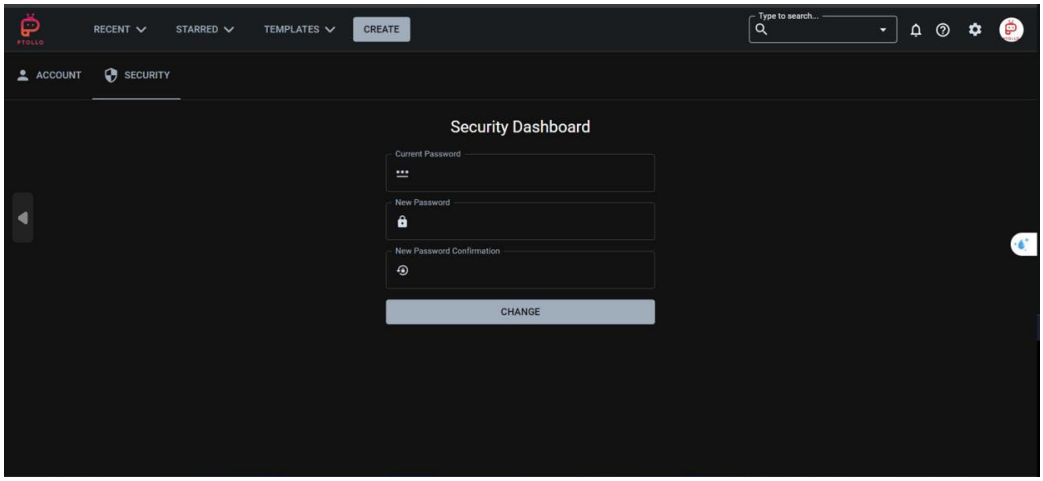


4.1.5 Màn hình chỉnh sửa profile và thay đổi mật khẩu



Hình 23: Giao diện profile

Hiển thị các thông tin về người dùng như email, tên người dùng, tên hiển thị, người dùng có thể cập nhật các thông tin trên.

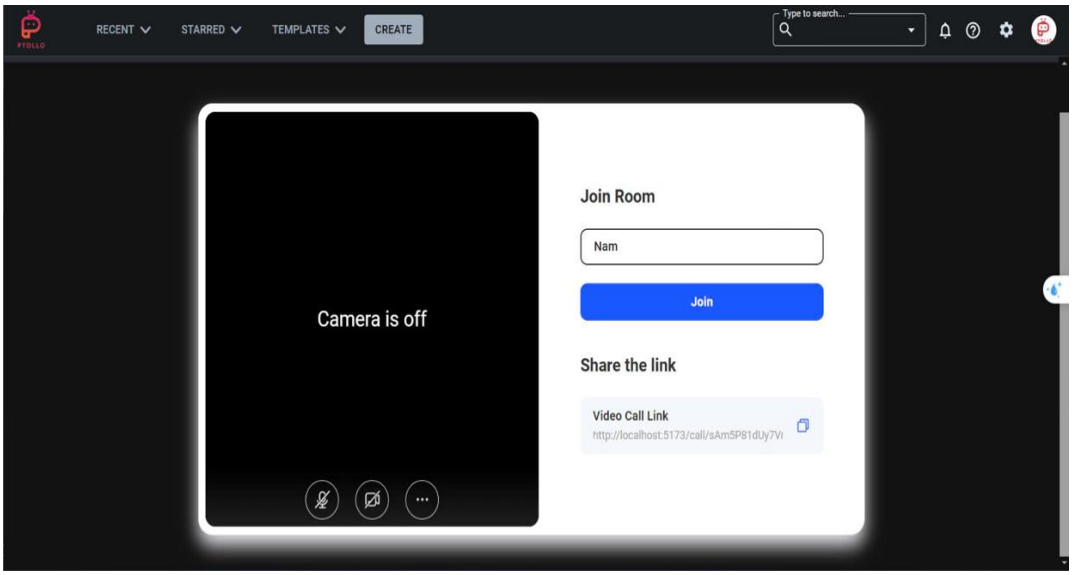


Người dùng còn có thể thay đổi password nhằm tăng tính bảo mật.

4.2 Màn hình chức năng khác

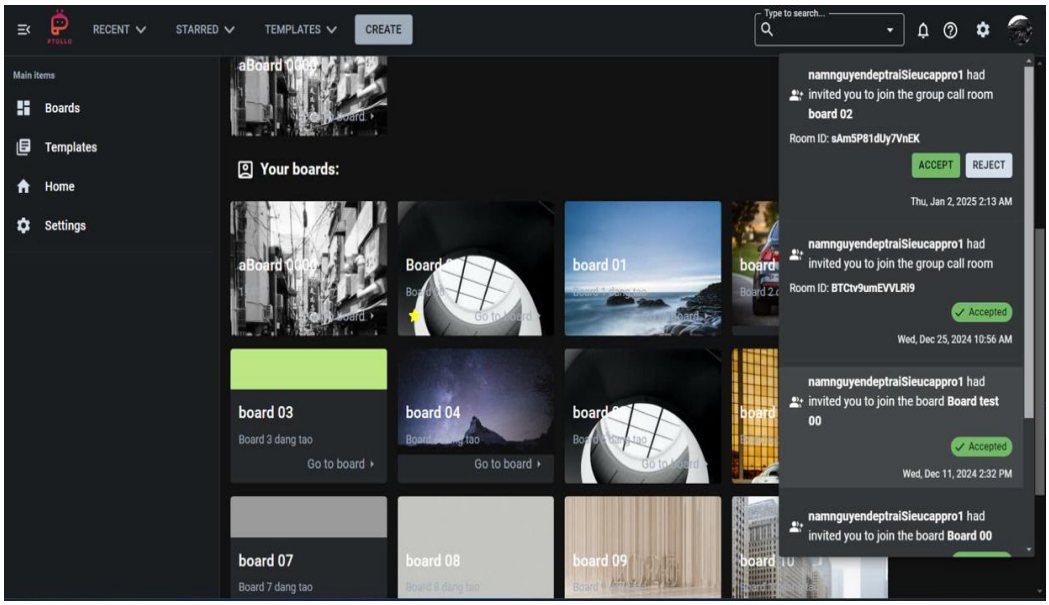
4.2.1 Chức năng tạo cuộc gọi video

Người dùng có thể thực hiện tạo các cuộc gọi video giữa các thành viên trong dự án

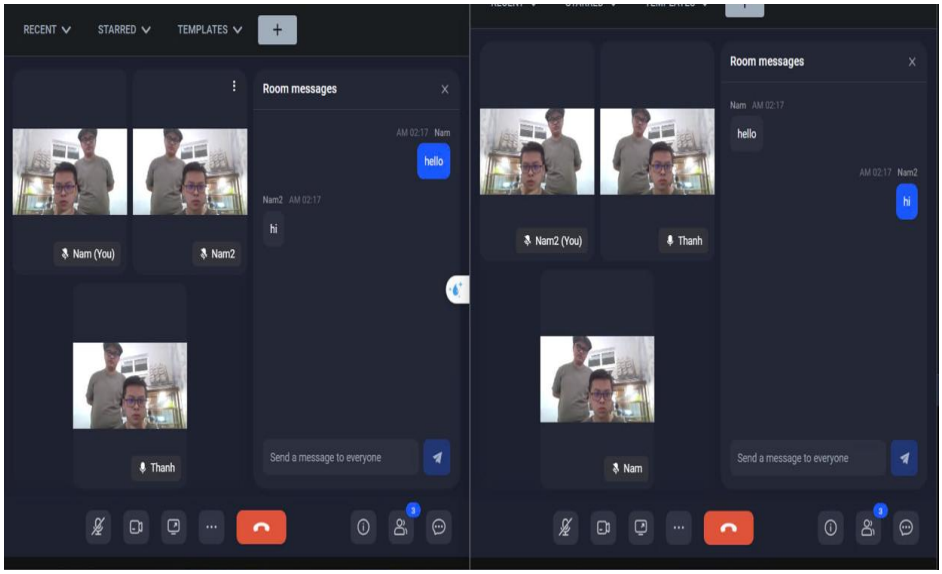


Hình 24: Giao diện tạo cuộc gọi Video

Sau đó một thông báo được gửi đến các thành viên có mặt trong bảng. Ngoài ra còn có thể chia sẽ link cuộc gọi

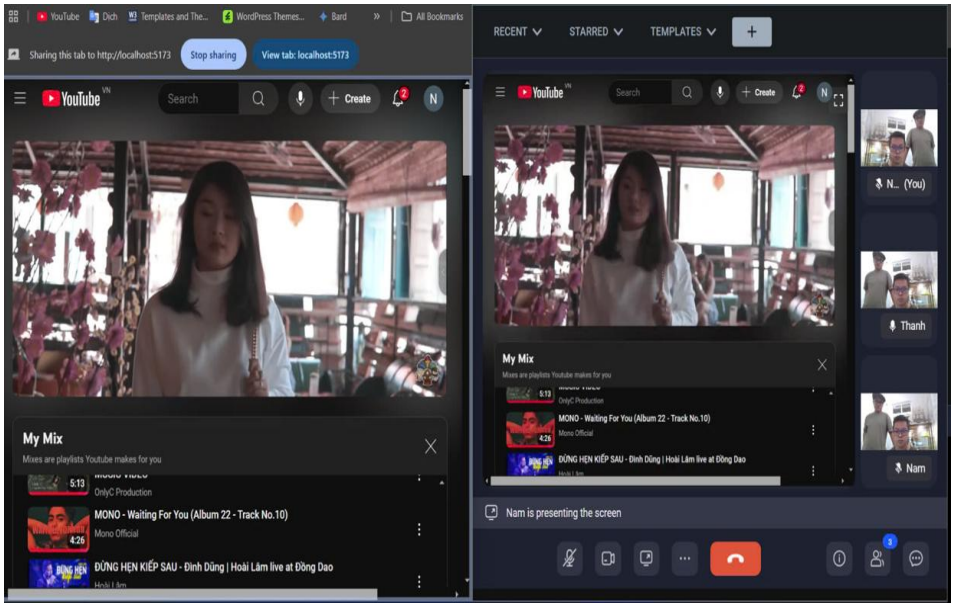


Sau khi nhấn Accept thì người dùng có thể tham gia vào cuộc gọi



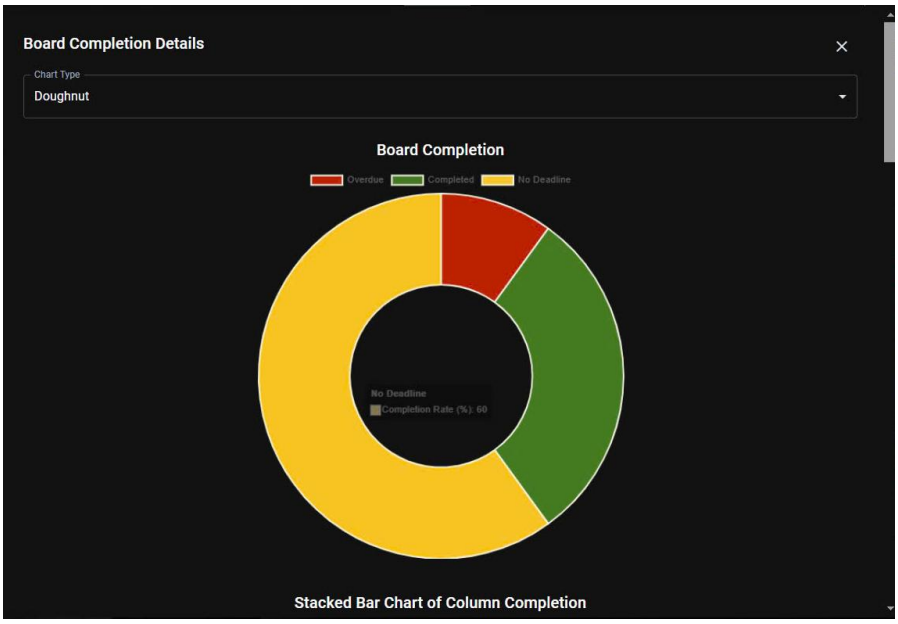
Hình 25: Giao diện cuộc gọi

Trong quá trình tham gia cuộc gọi người dùng có thể chat với nhau, ngoài ra còn có thể chia sẽ màn hình tương tự như messenger



4.2.2 Chức năng xem phần trăm hoàn thành công việc

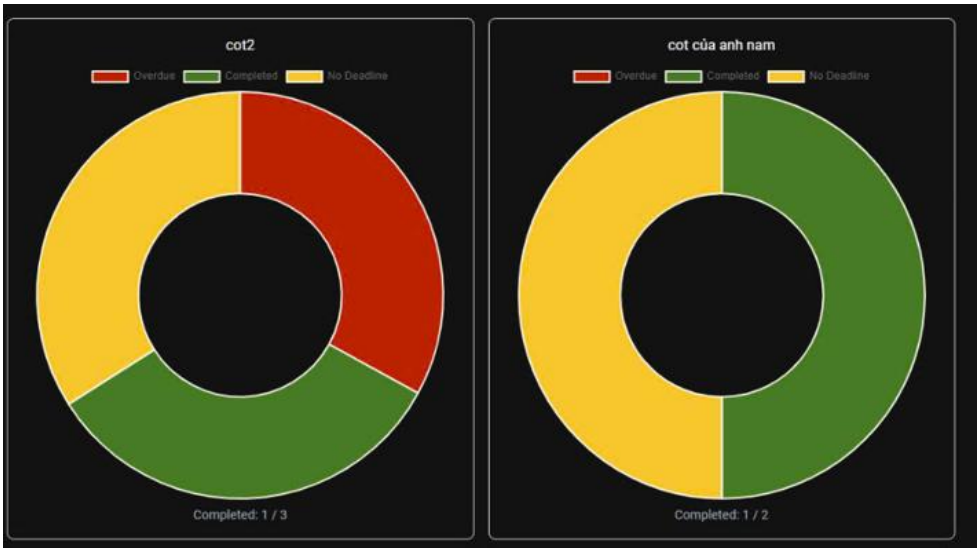
Đây là chức năng quan trọng của ứng dụng, mỗi một dự án đều phải có sự hoàn thành deadline và đánh giá phần trăm hoàn thành của từng thành viên.



Hình 26: Giao diện phần trăm công việc

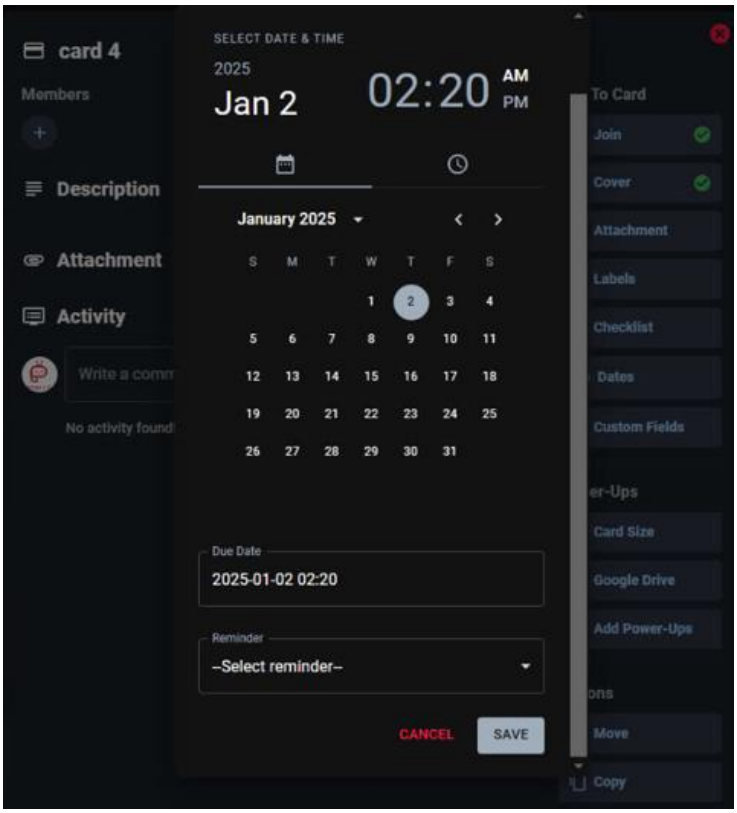
Giao diện tổng thể của phần trăm công việc là một vòng tròn dạng hình bánh doughnut với màu đỏ biểu trưng cho công việc quá thời gian mà chưa hoàn thành, màu xanh cho công việc đã xong và màu vàng là các thẻ công việc không được set deadline.

Ngoài ra ứng dụng còn đưa ra phần trăm hoàn thành của từng list công việc thông qua biểu đồ cột hay vòng tròn tương tự nhưng dành riêng cho từng cột.



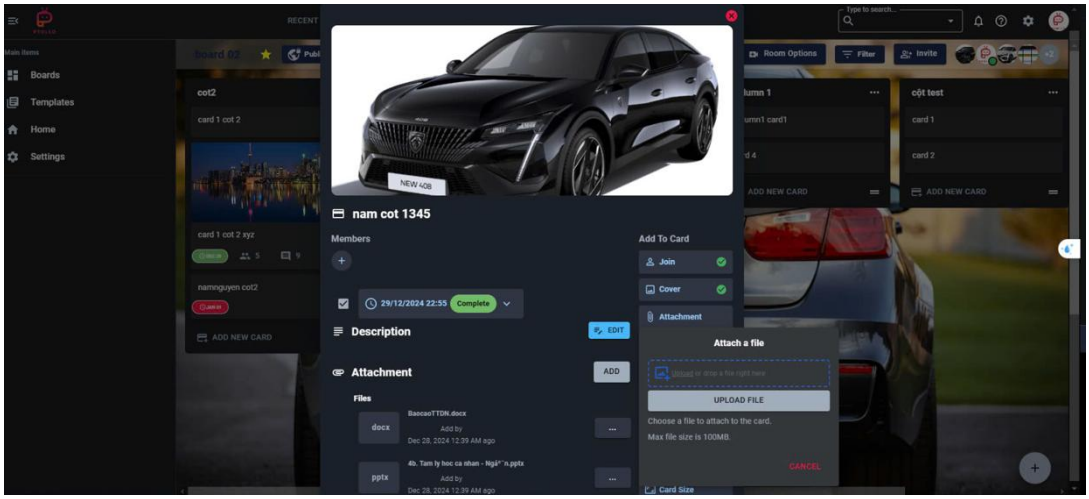
4.2.3 Chức năng set timeline và đính kèm tệp

Công cụ cho phép đặt các mốc thời gian cho các thành viên trong team hoàn thành đối với từng thẻ công việc cụ thể.



Hình 27: Giao diện set deadline

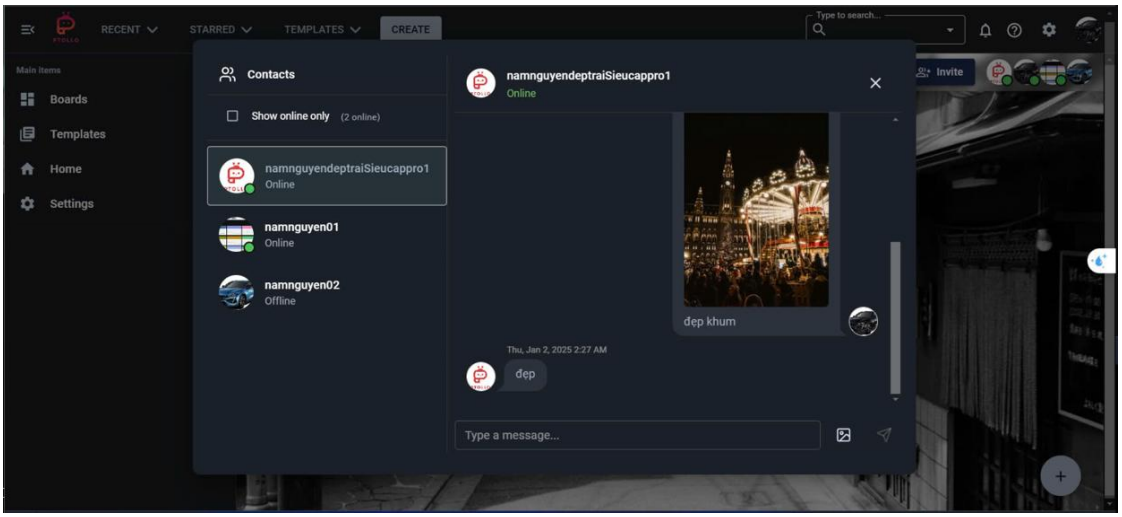
Còn đây là giao diện đính kèm file vào các thẻ công việc



Hình 28: Giao diện upload file

4.2.4 Chức năng chatbox

Một chatbox để các thành viên có thể nhắn tin với từng người khác nhau là cần thiết khi các thành viên không tiện để tham gia một cuộc họp video call hay muốn nhắn riêng với từng người.



Hình 29: Giao diện chatbox

Ứng dụng sẽ hiển thị những thành viên online hay offline, đồng thời còn cho phép gửi các file hình ảnh.

CHƯƠNG 5: ĐÁNH GIÁ KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

5.1 Đánh giá kết quả

- Thông qua quá trình thực hiện đồ án, nhóm đã xây dựng được một trang web có thể giúp cho người dùng cá nhân, nhóm làm việc hay một công ty có thể quản lý công việc một cách nhanh chóng hiệu quả chỉ bằng cách kéo thả.
- Ứng dụng giúp người dùng chia được các công việc ra thành từng phần nhỏ, sắp xếp chúng sao cho đúng thứ tự nhờ đó tiết kiệm được khá nhiều thời gian cho việc quản lý công việc bản thân.
- Các chức năng như chatbox và video call hoạt động ổn định, giúp các thành viên trong team có thể tương tác với nhau thuận lợi trao đổi trong quá trình làm việc.
- Chức năng upload đính kèm tệp tin hay gửi hình ảnh đem lại cho người dùng khả năng chia sẽ các file liên quan đến dự án.
- Website Ptollo hoàn thành được 100% mục tiêu cơ bản đề ra. Giao diện ứng dụng đúng như thiết kế trước đó. Các chứng năng cơ bản của ứng dụng đang hoạt động rất tốt.

5.2 Hạn chế

- Do thời gian gấp rút các chức năng mở rộng với AI như thu thập thông tin người dùng cùng dự án để gợi ý dự án cho người dùng vẫn chưa được thực hiện.
- Giao diện vẫn chưa thật sự tối ưu đối với đại đa số người dùng, có thể cải thiện hơn trong tương lai. Các template có sẵn còn hạn chế số lượng, chưa đa dạng.
- Cần bổ sung một số chức năng khác cung cấp cho người dùng nhiều tiện ích hơn như lọc các bảng theo các thông số như tên, kích thước hay những thông tin khác.

5.3 Hướng phát triển

- Thiết kế lại giao diện sao cho tối ưu hơn với đại đa số người dùng, làm giao diện trở nên đẹp và dễ tiếp cận hơn.
- Tạo thêm chức năng bộ lọc bảng, thiết kế hệ thống cho phép người dùng khác đăng tải các template khác nhau để người dùng khác có thể áp dụng.
- Mở rộng tính năng AI cho phép hệ thống gợi ý những người dùng phù hợp cho từng loại dự án khác nhau, từ đó chủ công ty hay leader các nhóm làm việc có thể chọn được các nhân viên phù hợp với dự án của mình.

TÀI LIỆU THAM KHẢO

1. W3Schools.com. (n.d.). https://www.w3schools.com/REACT/DEFAULT.ASP
2. ReactJS là gì? Tất tần tật những điều căn bản về ReactJS. (n.d.). https://stringee.com/vi/blog/post/reactJS-la-gi
3. Hướng dẫn Cài đặt và Sử dụng Tailwind CSS cơ bản. (n.d.). 200Lab Blog. https://200lab.io/blog/huong-dan-cai-dat-tailwind-css-co-ban
4. W3Schools.com. (n.d.-b). https://www.w3schools.com/mongodb/
5. Dương S. (2022, October 22). React là gì? Làm quen với ReactJS trong 5 phút. VNTALKING. https://vntalking.com/huong-dan-hoc-reactjs-co-ban.html
6. CodeGym. (2023, December 6). Web động và web tĩnh khác nhau như thế nào? CodeGym. https://codegym.vn/blog/web-dong-va-web-tinh-khac-nhau-nhu-the-nao/

HẾT.