BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**Logo

Description automatically generated**

**NIÊN LUẬN**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**XÂY DỰNG MODULE QUẢN LÝ NGƯỜI DÙNG VỚI SUPERTOKENS**

**BUILDING A USER MANAGEMENT MODULE WITH SUPERTOKENS**

**Sinh viên: Huỳnh Thiên Phú**

**Mã số: B2111859**

**Khóa: K47**

***Cần Thơ, 24/2025***

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**Logo

Description automatically generated**

**NIÊN LUẬN**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**XÂY DỰNG MODULE QUẢN LÝ NGƯỜI DÙNG VỚI SUPERTOKENS**

**BUILDING A USER MANAGEMENT MODULE WITH SUPERTOKENS**

|  |  |
| --- | --- |
| **Người hướng dẫn**  **TS. Bùi Võ Quốc Bảo** | **Sinh viên thực hiện**  **Huỳnh Thiên Phú**  **Mã số: B2111859**  **Khóa: K47** |
|  |  |

***Cần Thơ, 24/2025***

# LỜI CẢM ƠN

Để hoàn thành tốt niên luận ngành Công nghệ thông tin này em xin trân thành cảm ơn Thầy TS.Bùi Võ Quốc Bảo đã tận tình hướng dẫn, hỗ trợ em trong suốt quá trình thực hiện đề tài.

Đồng thời em cũng xin chân thành cảm ơn quý Thầy, Cô ở Khoa Công nghệ thông tin và Truyền thông đã tận tình giảng dạy, truyền đạt những kiến thức quí giá để em tích luỹ thực hiện đề tài trong các học kì vừa qua.

Mặc dù đã cố gắng hoàn thành đề tài một cách tốt nhất có thể, nhưng vẫn không thể tránh khỏi những thiếu sót. Rất mong sự góp ý của Thầy. Em xin chân thành cảm ơn

**MỤC LỤC**

[LỜI CẢM ƠN iii](#_Toc183384930)

[DANH MỤC HÌNH ẢNH vi](#_Toc183384931)

[DANH MỤC BẢNG vii](#_Toc183384932)

[TÓM LƯỢC viii](#_Toc183384933)

[ABSTRACT ix](#_Toc183384934)

[CHƯƠNG 1. GIỚI THIỆU 1](#_Toc183384935)

[1.1. Đặt vấn đề 1](#_Toc183384936)

[1.2. Mục tiêu đề tài 1](#_Toc183384937)

[1.3. Đối tượng và phạm vi nghiên cứu 1](#_Toc183384938)

[1.3.1. Đối tượng nghiên cứu 1](#_Toc183384939)

[1.4. Phương pháp nghiên cứu 1](#_Toc183384940)

[1.5. Nội dung nghiên cứu 2](#_Toc183384941)

[1.6. Bố cục của niên luận 2](#_Toc183384942)

[1.7. Tổng kết chương 1 2](#_Toc183384943)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 3](#_Toc183384944)

[2.1. Nodejs 3](#_Toc183384945)

[2.2. React 4](#_Toc183384946)

[2.3. Hệ quản trị cơ sở dữ liệu MySQL 5](#_Toc183384947)

[2.4. JSON Web Tokens (JWTs) 7](#_Toc183384948)

[2.5. Supertokens 9](#_Toc183384949)

[2.5.1. Cơ chế hoạt động của SuperTokens 10](#_Toc183384950)

[CHƯƠNG 3. CÀI ĐẶT GIẢI PHÁP 15](#_Toc183384951)

[1. Mô tả chung 15](#_Toc183384952)

[1.1. Thiết kế cơ sở dữ liệu của ứng dụng 15](#_Toc183384953)

[1.2. Sơ đồ use case 16](#_Toc183384954)

[1.3. Mô tả chi tiết use case 17](#_Toc183384955)

[2. Giao diện ứng dụng 22](#_Toc183384956)

[2.1. Giao diện đăng nhập 22](#_Toc183384957)

[2.2. Giao diện đăng ký 22](#_Toc183384958)

[2.3. Giao diện Forgot password 23](#_Toc183384959)

[2.4. Giao diện trang chủ 25](#_Toc183384960)

[2.5. Giao diện Profile 25](#_Toc183384961)

[2.6. Giao diện Change Password 25](#_Toc183384962)

[CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 27](#_Toc183384963)

[1. Về kết quả đạt được 27](#_Toc183384964)

[1.1. Về thực tiễn 27](#_Toc183384965)

[1.2. Về giao diện 27](#_Toc183384966)

[1.3. Về chức năng 27](#_Toc183384967)

[2. Hạn chế 27](#_Toc183384968)

[3. Hướng phát triển 27](#_Toc183384969)

[TÀI LIỆU THAM KHẢO 29](#_Toc183384970)

# DANH MỤC HÌNH ẢNH

[Hình 1.1. Nodejs 11](#_Toc183383407)

[Hình 1.2. React logo 12](#_Toc183383408)

[Hình 1.3 MySQL 14](#_Toc183383409)

[Hình 1.4 Cơ chế JWT trong xác thực người dùng 15](#_Toc183383410)

[Hình 1.5 Cấu trúc của JWT 16](#_Toc183383411)

[Hình 1.6 SuperTokens 17](#_Toc183383412)

[Hình 1.7 Sơ đồ kiến trúc hoạt động sefl-hosted của SuperTokens 18](#_Toc183383413)

[Hình 1.8 Sơ đồ tổng quan về luồng quản lý session 19](#_Toc183383414)

[Hình 1.9. Sơ đồ luồng đăng nhập của SuperTokens 21](#_Toc183383415)

[HÌnh 3.1 Sơ đồ thiết kế cơ sở dữ liệu 23](#_Toc183383416)

[Hình 3.2. Sơ đồ use case tổng quát 24](#_Toc183383417)

[Hình 3.3 Sơ đồ use case của người dùng 24](#_Toc183383418)

[Hình 3.4 Sơ đồ use case quản lý 25](#_Toc183383419)

[Hình 3.5 Giao diện đăng nhập của SuperTokens 30](#_Toc183383420)

[Hình 3.6 Giao diện đăng ký 31](#_Toc183383421)

[Hình 3.7 Giao diện Forgot password 31](#_Toc183383422)

[Hình 3.8 Giao diện email reset password 32](#_Toc183383423)

[Hình 3.9 Giao diện thay đổi password 32](#_Toc183383424)

[Hình 3.10 Giao diện trang chủ của ứng dụng 33](#_Toc183383425)

[Hình 3.11 Giao diện trang Profile 33](#_Toc183383426)

[Hình 3.12 Giao diện đổi password 34](#_Toc183383427)

# DANH MỤC BẢNG

[Bảng 3.1 Mô tả use case “Đăng ký” 17](#_Toc183384847)

[Bảng 3.2 Mô tả use case “Đăng nhập” 18](#_Toc183384848)

[Bảng 3.3 Mô tả use case “thêm thông tin tiêu dùng cá nhân” 19](#_Toc183384849)

[Bảng 3.4 Mô tả use case “Xóa thông tin tiêu dùng cá nhân” 19](#_Toc183384850)

[Bảng 3.5 Mô tả use case “Cập nhật email” 20](#_Toc183384851)

[Bảng 3.6 Mô tả use case “Đổi mật khẩu” 20](#_Toc183384852)

[Bảng 3.7 Mô tả use case “Thêm ngân sách” 21](#_Toc183384853)

[Bảng 3.8 Mô tả use case “Sửa ngân sách” 21](#_Toc183384854)

# TÓM LƯỢC

Việc tích hợp SuperTokens vào một website mang đến giải pháp xác thực mạnh mẽ, an toàn và thân thiện với nhà phát triển. SuperTokens đơn giản hóa việc xác thực người dùng và quản lý phiên làm việc, đồng thời duy trì các tiêu chuẩn bảo mật cao, bao gồm bảo vệ khỏi các lỗ hổng phổ biến như CSRF và XSS. Với kiến trúc mô-đun, nhà phát triển dễ dàng triển khai các tính năng như đăng nhập một lần (SSO), xác thực không cần mật khẩu, và đăng nhập qua mạng xã hội. SuperTokens còn có khả năng mở rộng cao, đảm bảo hiệu suất tối ưu ngay cả khi lưu lượng truy cập lớn. Sự tích hợp này cải thiện trải nghiệm người dùng bằng cách cung cấp quy trình đăng nhập và đăng ký liền mạch, đồng thời giảm thiểu độ phức tạp trong phát triển và bảo trì. Với mã nguồn mở và sự hỗ trợ từ cộng đồng tích cực, SuperTokens là lựa chọn lý tưởng cho các ứng dụng web hiện đại cần một hệ thống xác thực đáng tin cậy.

# ABSTRACT

Integrating SuperTokens into a website offers a robust, secure, and developer-friendly authentication solution. SuperTokens simplifies user authentication and session management while maintaining high standards of security, including protection against common vulnerabilities such as CSRF and XSS. By leveraging its modular architecture, developers can easily implement features like single sign-on (SSO), passwordless authentication, and social login. Furthermore, SuperTokens is highly scalable, ensuring optimal performance even under heavy traffic. This integration enhances the user experience by providing seamless login and registration flows while reducing development complexity and maintenance overhead. Its open-source nature and active community support make it an ideal choice for modern web applications seeking a reliable authentication system.

# CHƯƠNG 1. GIỚI THIỆU

## 1.1. Đặt vấn đề

Trong bối cảnh phát triển nhanh chóng của công nghệ thông tin, vấn đề bảo mật luôn là mối quan tâm hàng đầu của các tổ chức và doanh nghiệp. Việc đảm bảo an toàn dữ liệu, đặc biệt là trong quá trình xác thực và quản lý phiên đăng nhập của người dùng, trở nên vô cùng quan trọng. SuperTokens, một giải pháp mã nguồn mở, đã nổi lên như một công cụ hữu ích trong việc cung cấp cơ chế xác thực mạnh mẽ và an toàn. SuperTokens cho phép các nhà phát triển dễ dàng tích hợp hệ thống xác thực hiện đại vào ứng dụng, giảm thiểu rủi ro về bảo mật, đồng thời tối ưu hóa trải nghiệm người dùng.

## 1.2. Mục tiêu đề tài

Mục tiêu chính của đề tài “**Tìm hiểu và ứng dụng SuperTokens vào website**” là tìm hiểu về kiến trúc, cách cài đặt và cách tích hợp các chức năng của SuperTokens.

Mục tiêu cụ thể là tìm hiểu cách cài đặt, sử dụng dạng chứng thực Email Passsword with Social/Enterprise (OAuth 2.0, SAML) login, quản lý phiên và phân quyền quyền người dùng của SuperTokens, tích hợp vào website tiktok clone để chứng thực người dùng.

## 1.3. Đối tượng và phạm vi nghiên cứu

### 1.3.1. Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài bao gồm:

* Công nghệ SuperTokens: Nghiên cứu về kiến trúc, cơ chế hoạt động và quy trình xác thực mà SuperTokens cung cấp. Tìm hiểu về cách SuperTokens giải quyết các vấn đề liên quan đến bảo mật và quản lý phiên đăng nhập của người dùng.
* Tính ứng dụng thực tế: Phân tích các tình huống ứng dụng cụ thể của SuperTokens trong các dự án, bao gồm cách thức triển khai, hiệu quả đạt được, và những thách thức gặp phải trong quá trình sử dụng.

## 1.4. Phương pháp nghiên cứu

Phương pháp nghiên cứu tài liệu: Tìm hiểu và tham khảo các loại tài liệu online về công nghệ SuperTokens.

Phương pháp nghiên cứu thực nghiệm: Đọc tài liệu tham khảo, tìm hiểu và nắm bắt các kiến thức có liên quan. Tiến hành tích hợp công nghệ SuperTokens vào website.

## 1.5. Nội dung nghiên cứu

Đề tài thực hiện gồm các nội dung chính như sau:

* Tìm hiểu kiến trúc, cơ chế hoạt động của SuperTokens, nghiên cứu cách cấu hình, cài đặt và tích hợp SuperTokens vào website.
  + Nghiên cứu các tài liệu có liên quan đến công nghệ SuperTokens
  + Xây dựng website
* Tìm hiểu và sử dụng các công cụ, công nghệ, ngôn ngữ và thư viện hỗ trợ
  + Tìm hiểu về Nodejs, ReactJS

## 1.6. Bố cục của niên luận

Bố cục của niên luận gồm 6 phần chính như sau:

**Chương 1. Giới thiệu**

**Chương 2. Cơ sở lý thuyết**

**Chương 3. Cài đặt giải pháp**

**Chương 4. Kiểm thử và đánh giá**

**Chương 5. Kết luận và hướng phát triển**

## 1.7. Tổng kết chương 1

Trong chương này, niên luận cơ sở đã giới thiệu về các vấn đề, mục tiêu của đề tài, đối tượng và các phạm vi nghiên cứu, phương pháp nghiên cứu, nội dung nghiên cứu và bố cục của niên luận ngành.

# CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

## 2.1. Nodejs



Hình 1.1. Nodejs

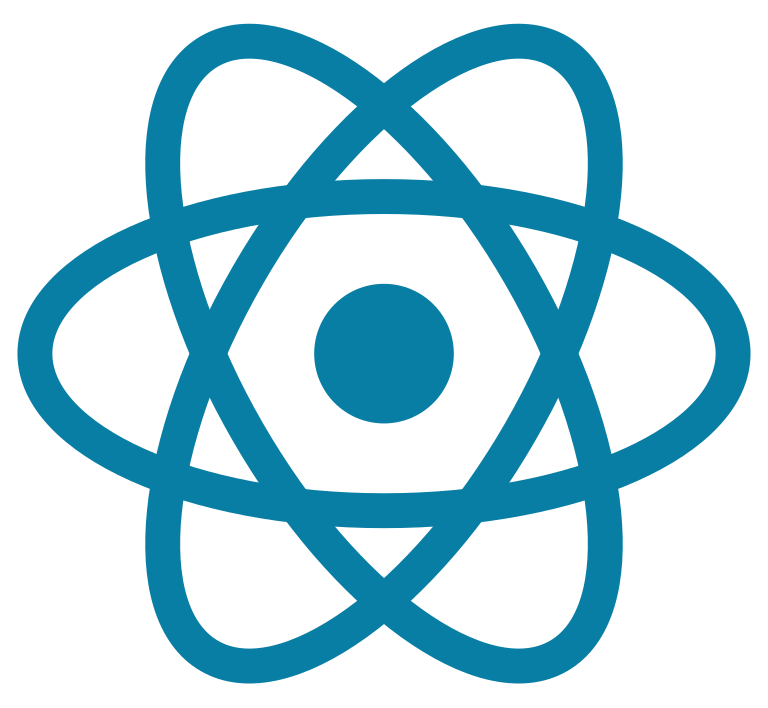
Nodejs là một môi trường JavaScript runtime mã nguồn mở hỗ trợ đa nền tảng được viết bằng C++, Nodejs có thể chạy trên Window, Linux, Unix, MacOS, …Nodejs được chạy trên công cụ V8 JavaScript.

Nodejs cho phép các nhà phát triển sử dụng JavaScript để viết các công cụ dòng lệnh và các script ở phía server-side. Khả năng để chạy JavaScript trên server thường được sử dụng để tạo nội dung trang web động trước khi trang được gửi đến web browser của người dùng.

Một số đặc điểm của Nodejs:

* Bất đồng bộ và hướng sự kiện: Node.js hoạt động dựa trên mô hình bất đồng bộ, tức là các tác vụ không cần phải chờ nhau hoàn thành. Điều này giúp xử lý nhiều tác vụ cùng lúc mà không làm gián đoạn tiến trình chính, thích hợp với các ứng dụng yêu cầu hiệu suất cao như các ứng dụng thời gian thực.
* Đơn luồng: Mặc dù Node.js là môi trường đơn luồng, nó sử dụng mô hình bất đồng bộ với vòng lặp sự kiện (event loop) để xử lý nhiều kết nối đồng thời mà không cần tạo ra nhiều luồng (threads), giúp tiết kiệm tài nguyên hệ thống.
* NPM (**Node Package Manager)**: Node.js đi kèm với NPM, hệ thống quản lý gói mạnh mẽ. NPM chứa hàng triệu gói phần mềm mã nguồn mở, giúp lập trình viên dễ dàng tích hợp các chức năng đã có sẵn vào ứng dụng của mình.
* Đa nền tảng: Node.js có thể chạy trên nhiều hệ điều hành như Windows, Linux, macOS, giúp phát triển ứng dụng đa nền tảng một cách dễ dàng.
* **Non-blocking I/O:** Một trong những lý do giúp Node.js trở nên mạnh mẽ là mô hình I/O không chặn, cho phép xử lý các yêu cầu đọc/ghi dữ liệu một cách hiệu quả mà không phải chờ hoàn tất một tác vụ I/O nào đó.

## 2.2. React



Hình 1.2. React logo

React ( ReactJS hay React.js) là một thư viện JavaScript front-end mã nguồn mở miễn phí được sử dụng để xây dựng giao diện người dùng dựa trên các components được phát triển bởi Facebook, ra mắt năm 2013. Nó được duy trì bởi Meta ( tiền thân là Facebook ) và một cộng đồng các công ty, các nhà phát triển cá nhân.

Các đặt điểm của React:

* Component-based: React cho phép xây dựng giao diện người dùng bằng cách chia nhỏ thành các thành phần (components) độc lập, có thể tái sử dụng. Mỗi component quản lý trạng thái và giao diện của riêng nó, giúp dễ dàng quản lý và bảo trì mã nguồn.
* JSX (JavaScript XML): JSX là một cú pháp mở rộng cho JavaScript, cho phép viết mã giống như HTML trong JavaScript. JSX giúp mã nguồn trở nên dễ đọc và dễ viết hơn, đồng thời tận dụng sức mạnh của JavaScript để xây dựng giao diện động.
* Virtual DOM: React sử dụng Virtual DOM để tối ưu hóa hiệu suất. Khi trạng thái của một component thay đổi, React sẽ tạo một cây DOM ảo mới và so sánh nó với cây DOM hiện tại để xác định các thay đổi cần thiết, sau đó chỉ cập nhật các phần bị thay đổi trên DOM thật. Điều này giúp giảm thiểu thao tác trực tiếp lên DOM, tăng tốc độ render và cải thiện hiệu suất.
* One-way Data Binding: Dữ liệu trong React chảy theo một chiều từ component cha xuống component con thông qua props. Điều này giúp dễ dàng theo dõi và quản lý dữ liệu, giảm thiểu các lỗi liên quan đến việc thay đổi dữ liệu ngẫu nhiên.
* State Management (Quản lý trạng thái): React cung cấp cơ chế quản lý trạng thái nội bộ thông qua state cho mỗi component. Ngoài ra, các thư viện quản lý trạng thái như Redux, MobX cũng thường được sử dụng kết hợp với React để quản lý trạng thái ứng dụng phức tạp.

Tóm lại, React là một công cụ mạnh mẽ và linh hoạt cho việc phát triển giao diện người dùng hiện đại. Với khả năng tái sử dụng component, hiệu suất cao và cộng đồng hỗ trợ rộng lớn, React tiếp tục là lựa chọn hàng đầu cho nhiều nhà phát triển trong việc xây dựng các ứng dụng web và di động chất lượng cao.

## 2.3. Hệ quản trị cơ sở dữ liệu MySQL

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System – RDBMS) mã nguồn mở, được sử dụng rộng rãi để lưu trữ, quản lý và truy xuất dữ liệu.



Hình 1.3 MySQL

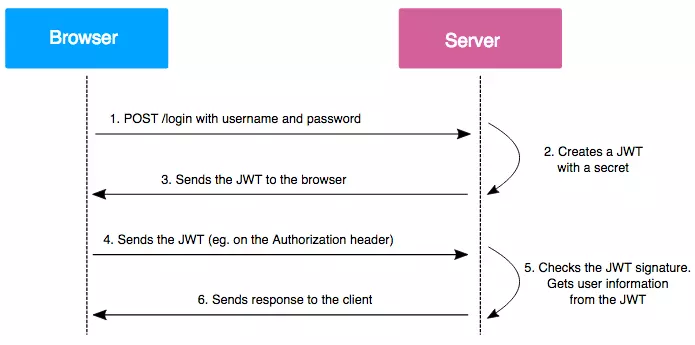
Một số đặc điểm của MySQL:

* Hiệu suất cao và khả năng mở rộng: MySQL được tối ưu hóa để xử lý một lượng lớn dữ liệu một cách nhanh chóng và hiệu quả. Hệ thống có thể dễ dàng mở rộng để đáp ứng nhu cầu lưu trữ và xử lý dữ liệu ngày càng tăng.
* Bảo mật và tính toàn vẹn dữ liệu: MySQL cung cấp các tính năng bảo mật mạnh mẽ như xác thực người dùng, phân quyền, mã hóa dữ liệu và phát hiện xâm nhập, giúp bảo vệ dữ liệu khỏi các mối đe dọa an ninh.
* Tính linh hoạt và tùy biến cao: MySQL hỗ trợ nhiều nền tảng, nhiều loại engine lưu trữ và cho phép người dùng tùy chỉnh cấu hình để phù hợp với nhu cầu cụ thể của ứng dụng.
* Cộng đồng lớn và hỗ trợ mạnh mẽ: MySQL có một cộng đồng người dùng rất lớn, sẵn sàng chia sẻ kinh nghiệm và giải đáp thắc mắc.
* Mã nguồn mở: MySQL là một phần mềm mã nguồn mở, người dùng có thể tự do sử dụng, sửa đổi và phân phối.
* Cơ chế quản lý dữ liệu: MySQL quản lý dữ liệu thông qua các cơ sở dữ liệu (database). Mỗi cơ sở dữ liệu có thể chứa nhiều bảng (table), mỗi bảng lưu trữ một loại dữ liệu cụ thể. Các bảng được liên kết với nhau thông qua các khóa (key) để tạo thành các mối quan hệ.
* Cơ chế phân quyền: MySQL có cơ chế phân quyền người dùng rất chi tiết. Mỗi người dùng có thể được cấp quyền truy cập đến một hoặc nhiều cơ sở dữ liệu, với các quyền hạn khác nhau như đọc, ghi, sửa đổi cấu trúc bảng, v.v.
* Ngôn ngữ SQL: MySQL sử dụng ngôn ngữ SQL (Structured Query Language) để tương tác với dữ liệu. SQL là ngôn ngữ chuẩn để truy vấn, cập nhật, xóa và quản lý dữ liệu trong các hệ quản trị cơ sở dữ liệu quan hệ.

## 2.4. JSON Web Tokens (JWTs)

JWT (JSON Web Token) là một chuẩn mở (RFC 7519) định nghĩa một cách nhỏ gọn và khép kín để truyền một cách an toàn thông tin giữa các bên dưới dạng đối tượng JSON. Thông tin này có thể được xác minh và đáng tin cậy vì nó có chứa chữ ký số. JWTs có thể được ký bằng một thuật toán bí mật (với thuật toán HMAC) hoặc một public / private key sử dụng mã hoá RSA.

Cụ thể, JWT (JSON Web Token) là các token tự chứa được sử dụng trong hệ thống xác thực để truyền thông tin một cách an toàn giữa các bên. Các token này bao gồm một payload mang các tuyên bố (claims), chẳng hạn như ID người dùng, vai trò hoặc các thông tin liên quan khác về dữ liệu người dùng, và được ký bằng một khóa bí mật.



Hình 1.4 Cơ chế JWT trong xác thực người dùng

JWT sau khi được ký sẽ được gửi đến client và sử dụng trong các yêu cầu tiếp theo, thường được truyền trong header Authorization. Tính chất không trạng thái của JWT làm cho chúng trở thành một lựa chọn phổ biến trong các hệ thống phân tán, vì không cần phải duy trì trạng thái phiên (session state) phía server.

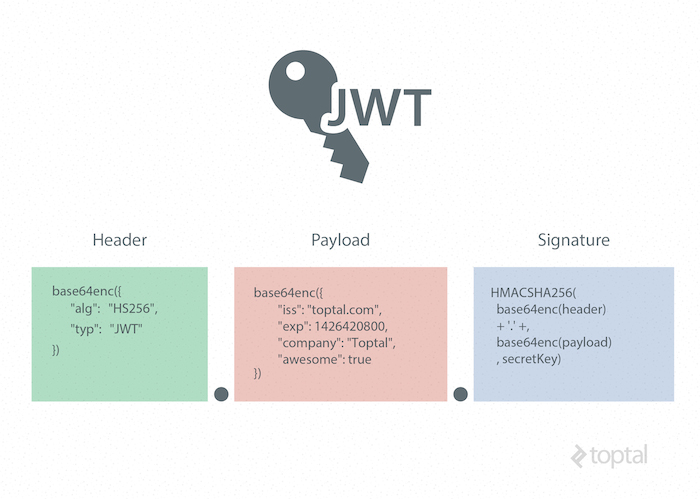
Cấu trúc của một JWT như sau:

<base64-encoded header>.<base64-encoded payload>.<base64-encoded signature>

Nói một cách khác, JWT là sự kết hợp (bởi dấu .) một Object Header dưới định dạng JSON được mã hóa base64, một payload object dưới định dạng JSON được mã hóa base64 và một Signature cũng được mã hóa bằng base64.

Đặc điểm 3 thành phần của JWT:

* Header: Header bao gồm hai phần chính: loại token (mặc định là JWT - Thông tin này cho biết đây là một Token JWT) và thuật toán đã dùng để mã hóa (HMAC SHA256 - HS256 hoặc RSA).
* Payload: Payload chứa các claims. Claims là một các biểu thức về một thực thể (chẳng hạn user) và một số metadata phụ trợ. Có 3 loại claims thường gặp trong Payload: reserved, public và private claims.
* Signature: Chữ ký Signature trong JWT là một chuỗi được mã hóa bởi header, payload cùng với một chuỗi bí mật theo nguyên tắc sau: HMACSHA256( base64UrlEncode(header) + "." + base64UrlEncode(payload),secret)



Hình 1.5 Cấu trúc của JWT

## 2.5. Supertokens

SuperTokens là một thư viện mã nguồn mở được thiết kế để cung cấp giải pháp xác thực (authentication) nhanh chóng, bảo mật và dễ sử dụng cho các ứng dụng web và di động. Nó giúp giảm bớt các công việc phức tạp liên quan đến xác thực người dùng, như quản lý token, lưu trữ phiên (session management), và xử lý đăng nhập/đăng xuất an toàn.



Hình 1.6 SuperTokens

Một số tính năng nổi bật của Supertokens:

* Quản lý phiên an toàn (Secure Session Management): SuperTokens cung cấp giải pháp lưu trữ phiên với bảo mật cao, sử dụng cả cookie HTTP-only và token xác thực.
* Hỗ trợ các phương thức xác thực hiện đại: Đăng nhập bằng email/password, Passwordless Login, Single Sign-On (SSO).
* Tùy biến cao: SuperTokens cho phép bạn tùy chỉnh giao diện người dùng (UI) và luồng xác thực để phù hợp với yêu cầu của ứng dụng. Hỗ trợ viết logic tùy chỉnh thông qua các API hooks hoặc mở rộng chức năng sẵn có.
* Triển khai dễ dàng: Có thể được triển khai dưới dạng một dịch vụ tự lưu trữ (self-hosted) hoặc sử dụng phiên bản đám mây (hosted service). Tích hợp nhanh chóng với các framework phổ biến như React, Vue, Angular, Node.js, hoặc các API REST/GraphQL.
* Tính năng mở rộng: Quản lý phân quyền và vai trò người dùng (User Roles and Permissions). Hỗ trợ đa thiết bị (multi-device session management) và khả năng đăng xuất từ xa (remote logout).

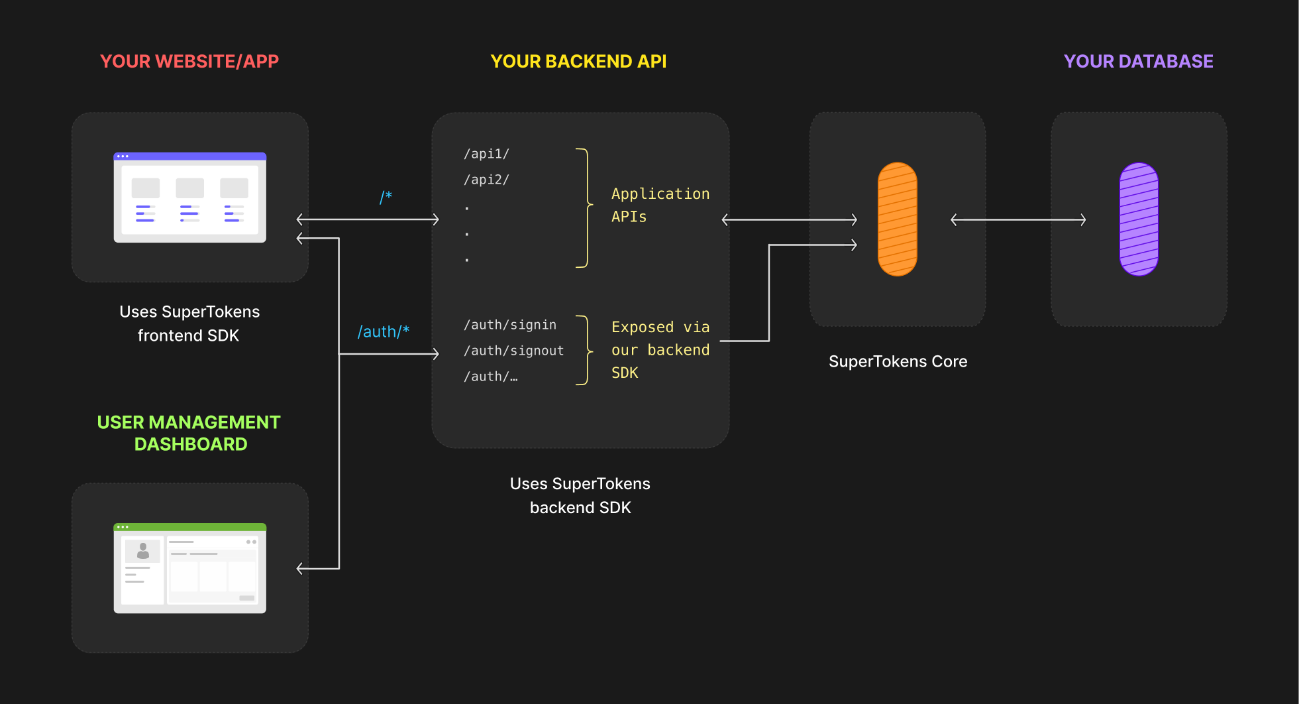
### 2.5.1. Cơ chế hoạt động của SuperTokens

#### 2.5.1.1. Kiến trúc SuperTokens và SDKs

[1] SuperTokens cung cấp 2 lựa chọn hoặc là tự triển khai và kết nối đến cơ sở dữ liệu của bản thân hoặc là sử dụng dịch vụ quản lý của SuperTokens. Trong đề tài này sử dụng lựa chọn tự triển khai với database.

SuperTokens gồm có 3 thành phần chính:

* Frontend SDK: Chịu trách nhiệm hiển thị các widget giao diện đăng nhập và tự động quản lý các token phiên làm việc.
* Backend SDK: Cung cấp các API cho việc đăng ký, đăng nhập, đăng xuất, làm mới phiên, v.v. Frontend sẽ giao tiếp với các API này. Các API này được triển khai trên cùng một domain với các API của ứng dụng.
* SuperTokens Core: Đây là một dịch vụ HTTP chứa logic cốt lõi cho xác thực. Nó chịu trách nhiệm tương tác với cơ sở dữ liệu và được Backend SDK truy vấn để thực hiện các thao tác yêu cầu truy cập cơ sở dữ liệu.

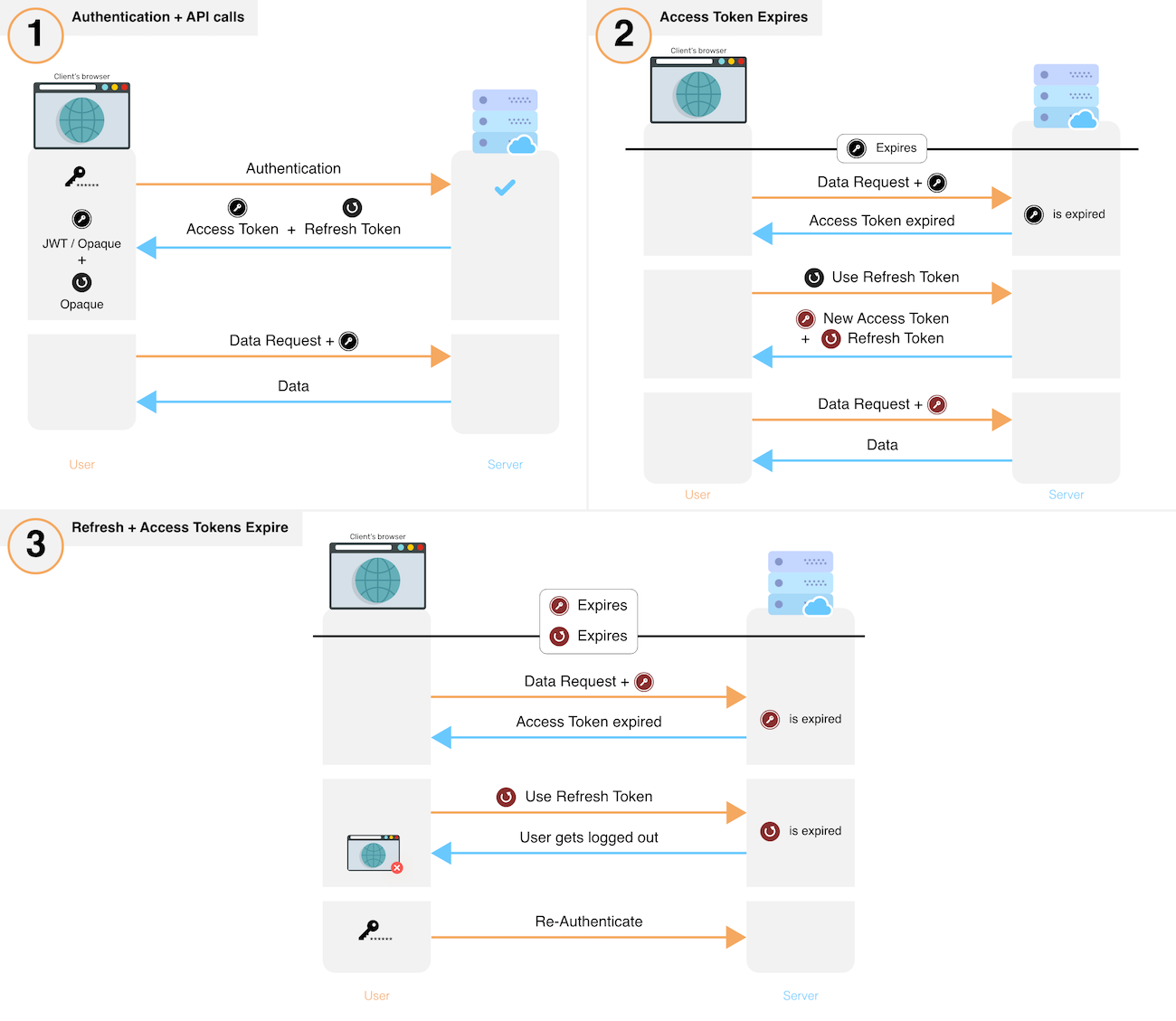


Hình 1.7 Sơ đồ kiến trúc hoạt động sefl-hosted của SuperTokens

Dựa vào sơ đồ trên, có thể thấy:

* SuperTokens Frontend SDK không bao giờ giao tiếp trực tiếp với Dịch vụ Xác thực (Authentication Service). Tất cả các yêu cầu đều được gửi đến Backend của ứng dụng . Tại đây, các Backend SDK được sử dụng để tạo các route xác thực mới. Những route này sau đó sẽ giao tiếp với SuperTokens Core.

#### 2.5.1.2. Quản lý Session



Hình 1.8 Sơ đồ tổng quan về luồng quản lý session

[2] Sau khi đăng nhập, một phiên làm việc mới (session) được tạo bằng cách cấp một refresh token và một access token cho frontend.  
Frontend sẽ gửi access token này kèm theo mỗi yêu cầu API yêu cầu xác thực phiên làm việc.

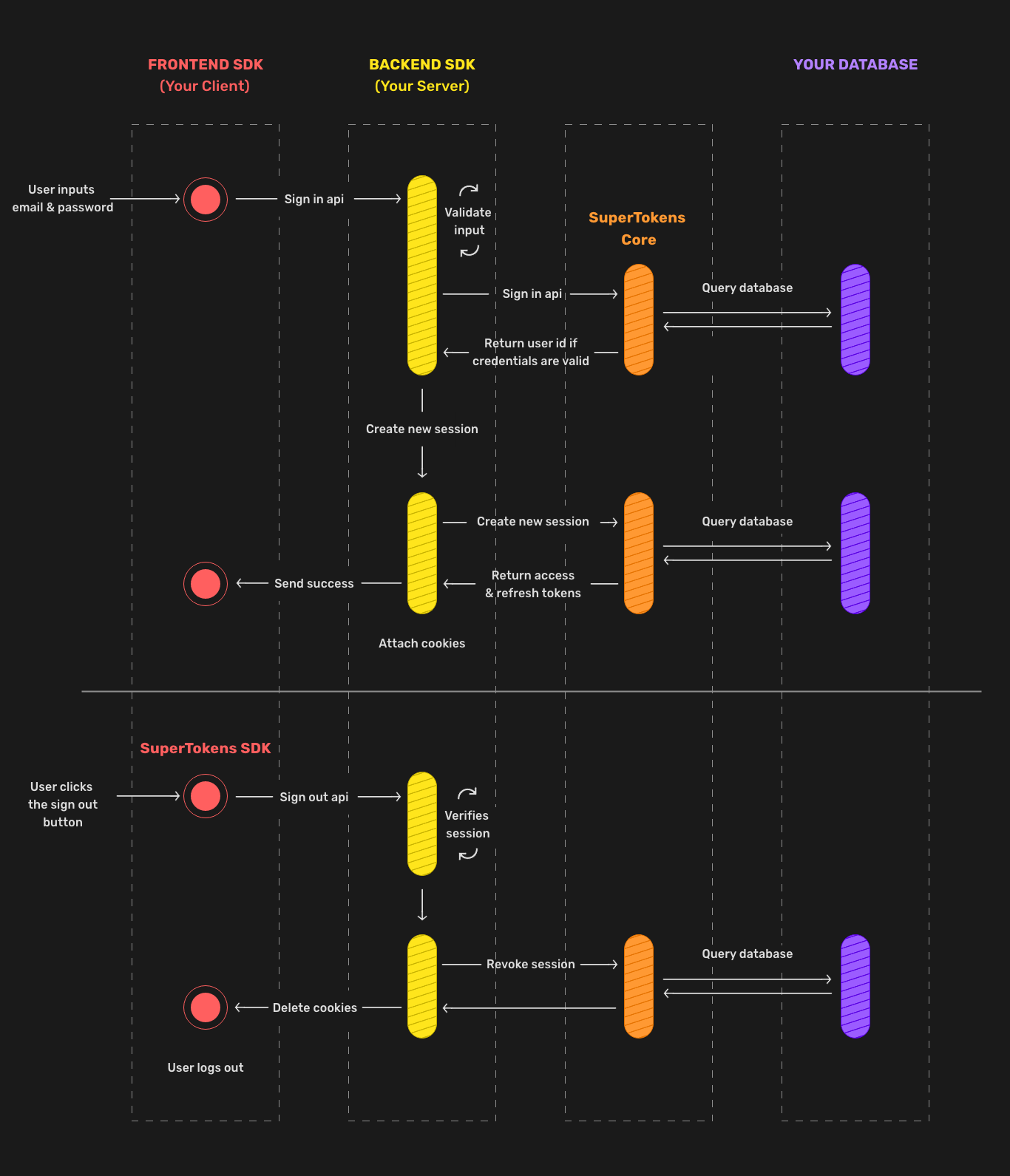
Các yêu cầu API này sẽ kiểm tra tính hợp lệ của access token và thời gian hết hạn của nó. Nếu quá trình xác minh thất bại, API sẽ trả về lỗi phiên làm việc đã hết hạn (session expired error), ngược lại, yêu cầu sẽ tiếp tục được thực thi.

Nếu một API trả về lỗi phiên làm việc đã hết hạn, frontend sẽ sử dụng refresh token để lấy một refresh token mới và một access token mới. Điều này được thực hiện thông qua một API đặc biệt trên backend. Nếu phiên làm việc đã bị thu hồi, API này cũng sẽ trả về lỗi phiên làm việc đã hết hạn, buộc người dùng phải đăng nhập lại.

Sau khi nhận được bộ token mới, frontend sẽ thực hiện lại yêu cầu API ban đầu và nhận được kết quả mong muốn.

Để thu hồi một phiên làm việc, backend sẽ xóa refresh token và thông tin phiên làm việc khỏi cơ sở dữ liệu của mình.

#### 2.5.1.3. Quản lý luồng đăng nhập



Hình 1.9. Sơ đồ luồng đăng nhập của SuperTokens

Các bước thực hiện luồng:

* Quy trình bắt đầu khi frontend gửi yêu cầu đến API của Backend SDK để thực hiện đăng nhập (/auth/signin - phương thức POST), kèm theo thông tin đăng nhập của người dùng.
* Backend SDK thực hiện kiểm tra cú pháp đầu vào và gửi thông tin đăng nhập tới SuperTokens Core.
* Core phản hồi với trạng thái: hoặc là OK (kèm theo userId), hoặc là chuỗi trạng thái WRONG\_CREDENTIALS\_ERROR
* Nếu trạng thái là WRONG\_CREDENTIALS\_ERROR, Backend SDK gửi thông báo này về frontend để hiển thị thông điệp phù hợp cho người dùng.
* Nếu trạng thái là OK, Backend sẽ gửi yêu cầu tới Core với userId để tạo các token phiên làm việc mới (Access token và Refresh token).
* Sau khi Core phản hồi với các token phiên mới, Backend SDK sẽ đính kèm chúng vào phản hồi dưới dạng cookie hoặc header và gửi về cho frontend. Theo mặc định, cookie được sử dụng cho các ứng dụng web, còn header được sử dụng cho các loại client khác.
* Người dùng hiện đã đăng nhập thành công.

# CHƯƠNG 3. CÀI ĐẶT GIẢI PHÁP

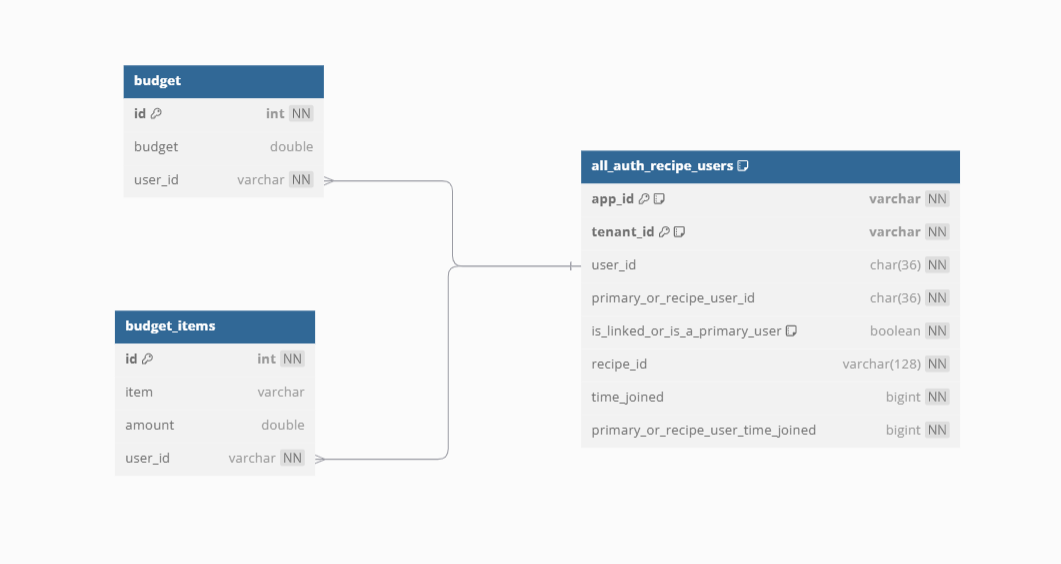
## 1. Mô tả chung

Trong đề tài này sẽ thực hiện tích hợp SuperTokens vào một website quản lý chi tiêu gia đình với cơ sở dữ liệu được thiết kế sẵn bởi SuperTokens core và được bổ sung thêm một số bảng dữ liệu riêng của ứng dụng.

Ứng dụng sẽ được tích hợp với các chức năng:

* Xác thực người dùng với Email/Password và Google.
* Reset password
* Xác minh email
* Cập nhật lại email
* Phân quyền người dùng

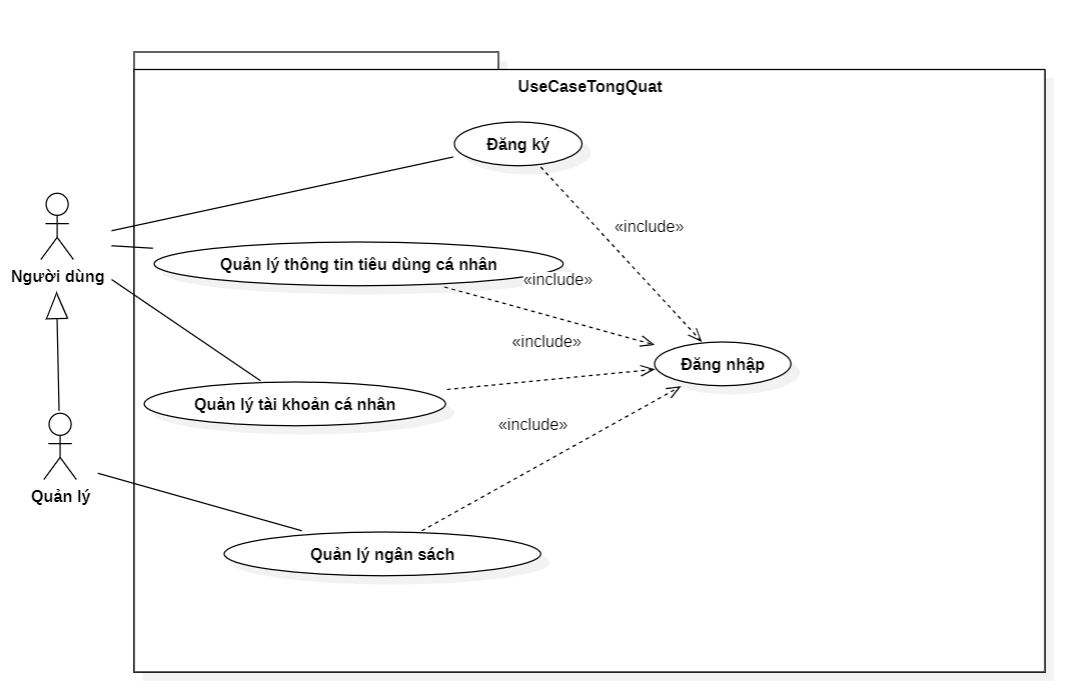
### 1.1. Thiết kế cơ sở dữ liệu của ứng dụng



HÌnh 3.1 Sơ đồ thiết kế cơ sở dữ liệu

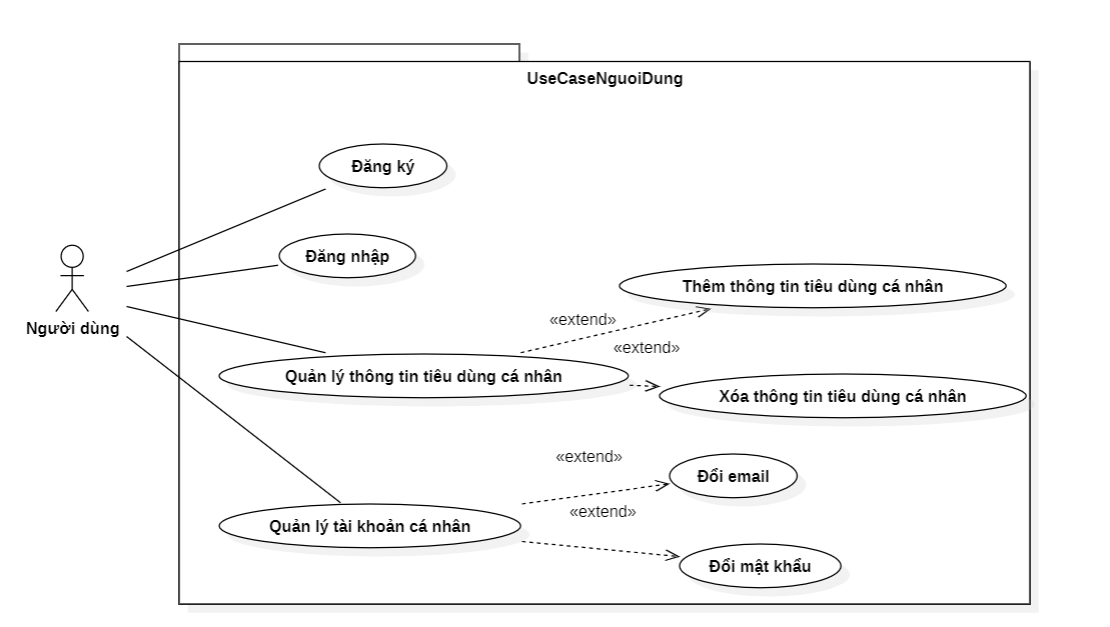
Trong sơ đồ trên các bảng dữ liệu được bổ sung bao gồm: budget, budget\_items. Bảng dữ liệu all\_auth\_recipe\_users được thiết kế sẵn bởi SuperTokens.

### 1.2. Sơ đồ use case



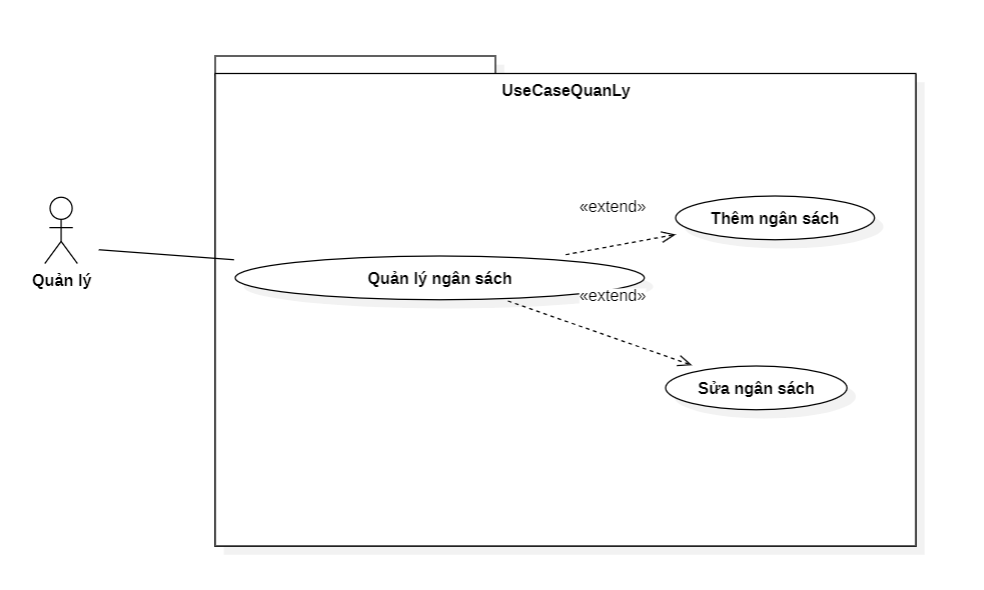
Hình 3.2. Sơ đồ use case tổng quát

#### 1.2.1. Sơ đồ use case người dùng



Hình 3.3 Sơ đồ use case của người dùng

#### 1.2.2. Sơ đồ use case của quản lý



Hình 3.4 Sơ đồ use case quản lý

### 1.3. Mô tả chi tiết use case

Bảng 3.1 Mô tả use case “Đăng ký”

|  |  |
| --- | --- |
| **Tác nhân** | **Người dùng** |
| **Mô tả** | Cho phép đăng ký tài khoản. |
| **Điều kiện** |  |
| **Luồng sự kiện chính** | 1. Người dùng truy cập vào trang web và chọn Sign up trên bảng đăng nhập. 2. Người dùng thực hiện nhập thông tin. 3. Người dùng nhấn nút Sign up 4. Hệ thống thực hiện gửi email xác minh. 5. Người dùng xác minh email. 6. Người dùng xác minh email thành công và tài khoản được tạo thành công.   Kết thúc đăng ký tài khoản. |
| **Luồng sự kiện rẽ nhánh** |  |
| **Kết quả** | Người dùng thực hiện đăng ký thành công. |

Bảng 3.2 Mô tả use case “Đăng nhập”

|  |  |
| --- | --- |
| **Tác nhân** | **Người dùng** |
| **Mô tả** | Cho phép đăng nhập tài khoản. |
| **Điều kiện** | Người dùng đã thực hiện đăng ký tài khoản |
| **Luồng sự kiện chính** | 1. Người dùng truy cập vào trang web và thực hiện nhập thông tin đăng nhập trên bảng đăng nhập. 2. Người dùng thực hiện nhấn nút Sign in 3. Người dùng đăng nhập thành công.   Kết thúc đăng nhập hệ thống. |
| **Luồng sự kiện rẽ nhánh** | Người dùng nhập thông tin sai, hệ thống sẽ thông báo dữ liệu không hợp lệ và cho phép người dùng thực hiện đăng nhập lại |
| **Kết quả** | Người dùng thực hiện đăng nhập thành công. |

Bảng 3.3 Mô tả use case “thêm thông tin tiêu dùng cá nhân”

|  |  |
| --- | --- |
| **Tác nhân** | **Người dùng** |
| **Mô tả** | Cho phép quản lý thông tin tiêu dùng cá nhân. |
| **Điều kiện** | Người dùng đã thực hiện đăng nhập thành công vào hệ thống |
| **Luồng sự kiện chính** | 1. Người dùng truy cập vào trang web, thực hiện thêm mới thông tin tiêu dùng. 2. Người dùng thực hiện nhấn nút Add 3. Người dùng thực hiện thêm thành công.   Kết thúc thêm thông tin tiêu dùng vào hệ thống. |
| **Luồng sự kiện rẽ nhánh** | Người dùng nhập thông tin không hợp lệ, hệ thống sẽ thông báo dữ liệu không hợp lệ và cho phép người dùng thực hiện nhập lại |
| **Kết quả** | Người dùng thực hiện thêm thành công thành công. |

Bảng 3.4 Mô tả use case “Xóa thông tin tiêu dùng cá nhân”

|  |  |
| --- | --- |
| **Tác nhân** | **Người dùng** |
| **Mô tả** | Cho phép xóa thông tin tiêu dùng cá nhân. |
| **Điều kiện** | Người dùng đã thực hiện đăng nhập thành công vào hệ thống |
| **Luồng sự kiện chính** | 1. Người dùng truy cập vào trang web, thực hiện xóa thông tin tiêu dùng. 2. Người dùng thực hiện nhấn nút Delete bên phải của thông tin cần xóa. 3. Người dùng thực hiện xóa thành công.   Kết thúc xóa thông tin tiêu dùng vào hệ thống. |
| **Luồng sự kiện rẽ nhánh** | Người dùng xóa thông tin không hợp lệ, hệ thống sẽ thông báo không hợp lệ. |
| **Kết quả** | Người dùng thực hiện xóa thành công thành công. |

Bảng 3.5 Mô tả use case “Cập nhật email”

|  |  |
| --- | --- |
| **Tác nhân** | **Người dùng** |
| **Mô tả** | Cho phép đổi email. |
| **Điều kiện** | Người dùng đã thực hiện đăng nhập thành công vào hệ thống |
| **Luồng sự kiện chính** | 1. Người dùng truy cập vào trang web, nhân vào biểu tượng user. 2. Người dùng thực hiện thay đổi email hiển thị trên màn hình. 3. Người dùng thực hiện nhấn nút Update để cập nhật. 4. Hệ thống thực hiện gửi email xác minh. 5. Người dùng thực hiện xác minh email và cập nhật lại email.   Kết thúc cập nhật email. |
| **Luồng sự kiện rẽ nhánh** | Người dùng nhập email đã được sử dụng, hệ thống hiển thị thông báo và cho phép người dùng thực hiện nhập lại. |
| **Kết quả** | Người dùng thực hiện cập nhật email thành công thành công. |

Bảng 3.6 Mô tả use case “Đổi mật khẩu”

|  |  |
| --- | --- |
| **Tác nhân** | **Người dùng** |
| **Mô tả** | Cho phép đổi mật khẩu. |
| **Điều kiện** | Người dùng đã thực hiện đăng nhập thành công vào hệ thống |
| **Luồng sự kiện chính** | 1. Người dùng truy cập vào trang web, nhân vào biểu tượng user, sau đó nhấn vào change password. 2. Người dùng thực hiện nhập mật khẩu hiện thời và mật khẩu mới. 3. Người dùng thực hiện nhấn nút Update để cập nhật.   Kết thúc đổi mật khẩu. |
| **Luồng sự kiện rẽ nhánh** |  |
| **Kết quả** | Người dùng thực hiện đổi mật khẩu thành công thành công. |

Bảng 3.7 Mô tả use case “Thêm ngân sách”

|  |  |
| --- | --- |
| **Tác nhân** | **Quản lý** |
| **Mô tả** | Cho phép đổi thêm ngân sách. |
| **Điều kiện** | Người dùng đã thực hiện đăng nhập thành công vào hệ thống. |
| **Luồng sự kiện chính** | 1. Người dùng truy cập vào trang web. 2. Người dùng thực hiện nhập ngân sách ở đầu trang web. 3. Người dùng thực hiện nhấn nút Save để cập nhật.   Kết thúc thêm ngân sách. |
| **Luồng sự kiện rẽ nhánh** |  |
| **Kết quả** | Người dùng thực hiện thêm ngân sách thành công. |

Bảng 3.8 Mô tả use case “Sửa ngân sách”

|  |  |
| --- | --- |
| **Tác nhân** | **Quản lý** |
| **Mô tả** | Cho phép đổi sửa ngân sách. |
| **Điều kiện** | Người dùng đã thực hiện đăng nhập thành công vào hệ thống. |
| **Luồng sự kiện chính** | 1. Người dùng truy cập vào trang web. 2. Người dùng thực hiện nhập ngân sách ở đầu trang web. 3. Người dùng thực hiện nhấn nút Save để cập nhật.   Kết thúc sửa ngân sách. |
| **Luồng sự kiện rẽ nhánh** |  |
| **Kết quả** | Người dùng thực hiện sửa ngân sách thành công. |

Ngoài ra, ứng dụng còn sử dụng Admin Dashboard được xây dựng sẵn của SuperTokens nhằm quản lý tài khoản, vai trò của người dùng.

## 2. Giao diện ứng dụng

## 2.1. Giao diện đăng nhập

Giao diện đăng nhập được xây dựng dựa trên giao diện được xây dựng sẵn của SuperTokens, từ đó có thể tùy chỉnh bổ sung.

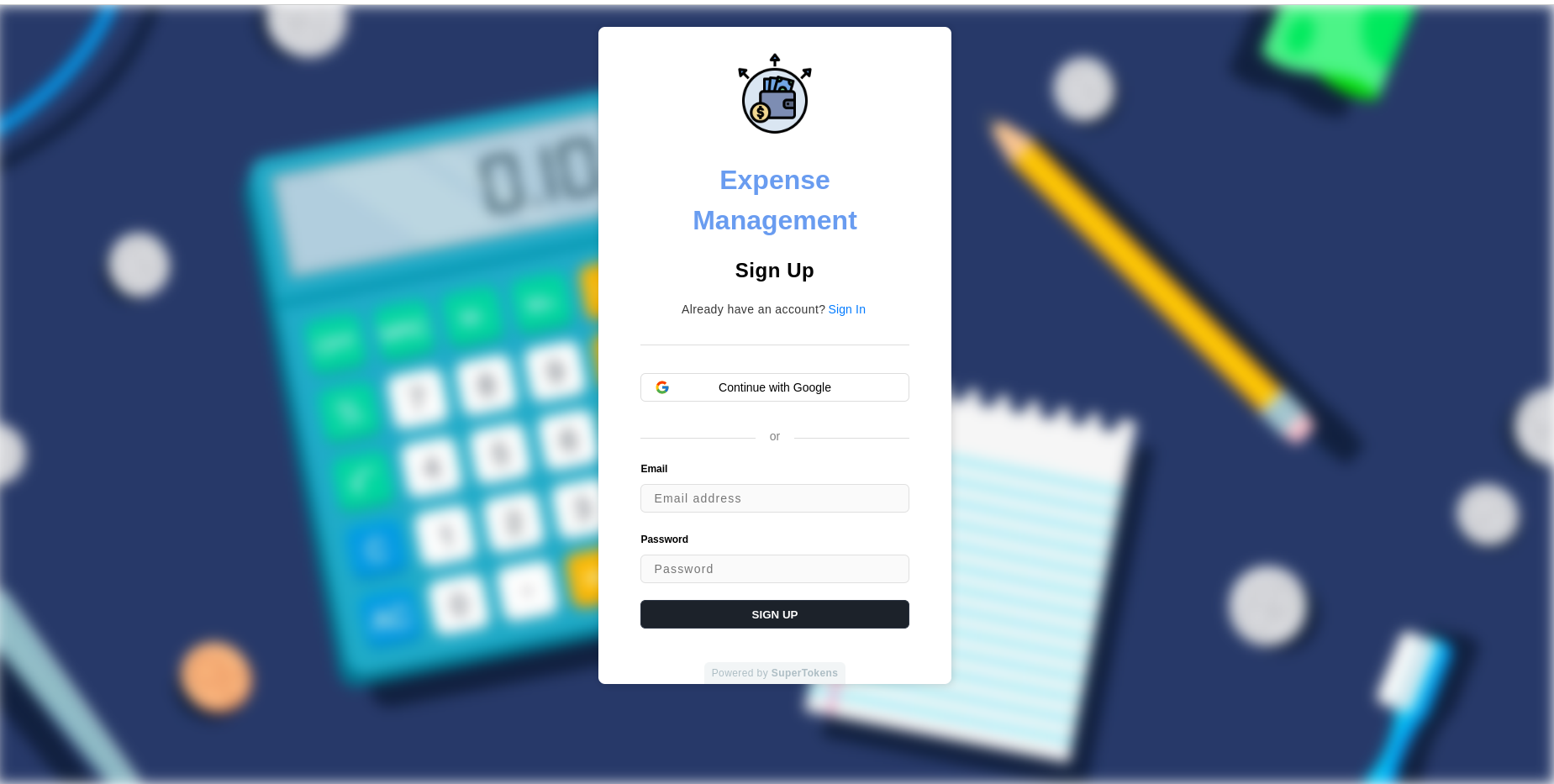
Trước khi sử dụng các chức năng của ứng dụng, người dùng và người quản lý phải đăng nhập vào hệ thống bằng tài khoản của mình. Khi đăng nhập thành công, hệ thống sẽ chuyển sang giao diện trang chủ.



Hình 3.5 Giao diện đăng nhập của SuperTokens

## 2.2. Giao diện đăng ký

Giao diện đăng ký tài khoản được xây dựng dựa trên giao diện được xây dựng sẵn của SuperTokens, từ đó có thể tùy chỉnh bổ sung.



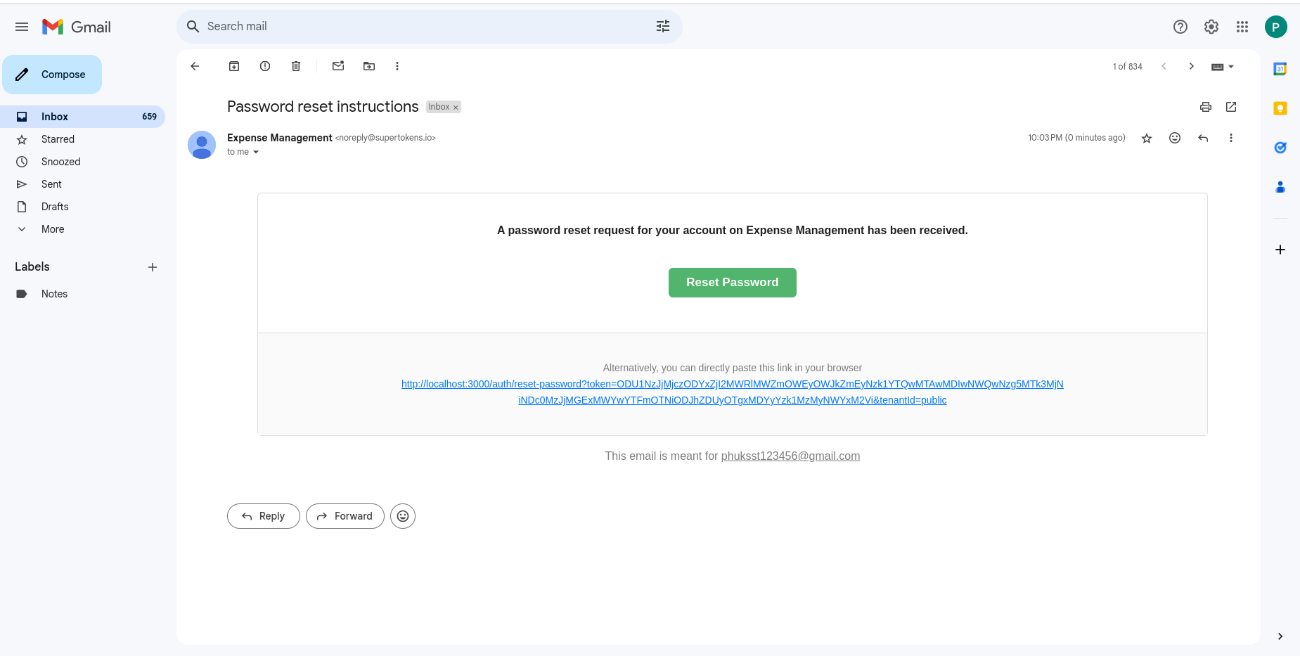
Hình 3.6 Giao diện đăng ký

## 2.3. Giao diện Forgot password

Các giao diện này được xây dựng sẵn của SuperTokens.



Hình 3.7 Giao diện Forgot password

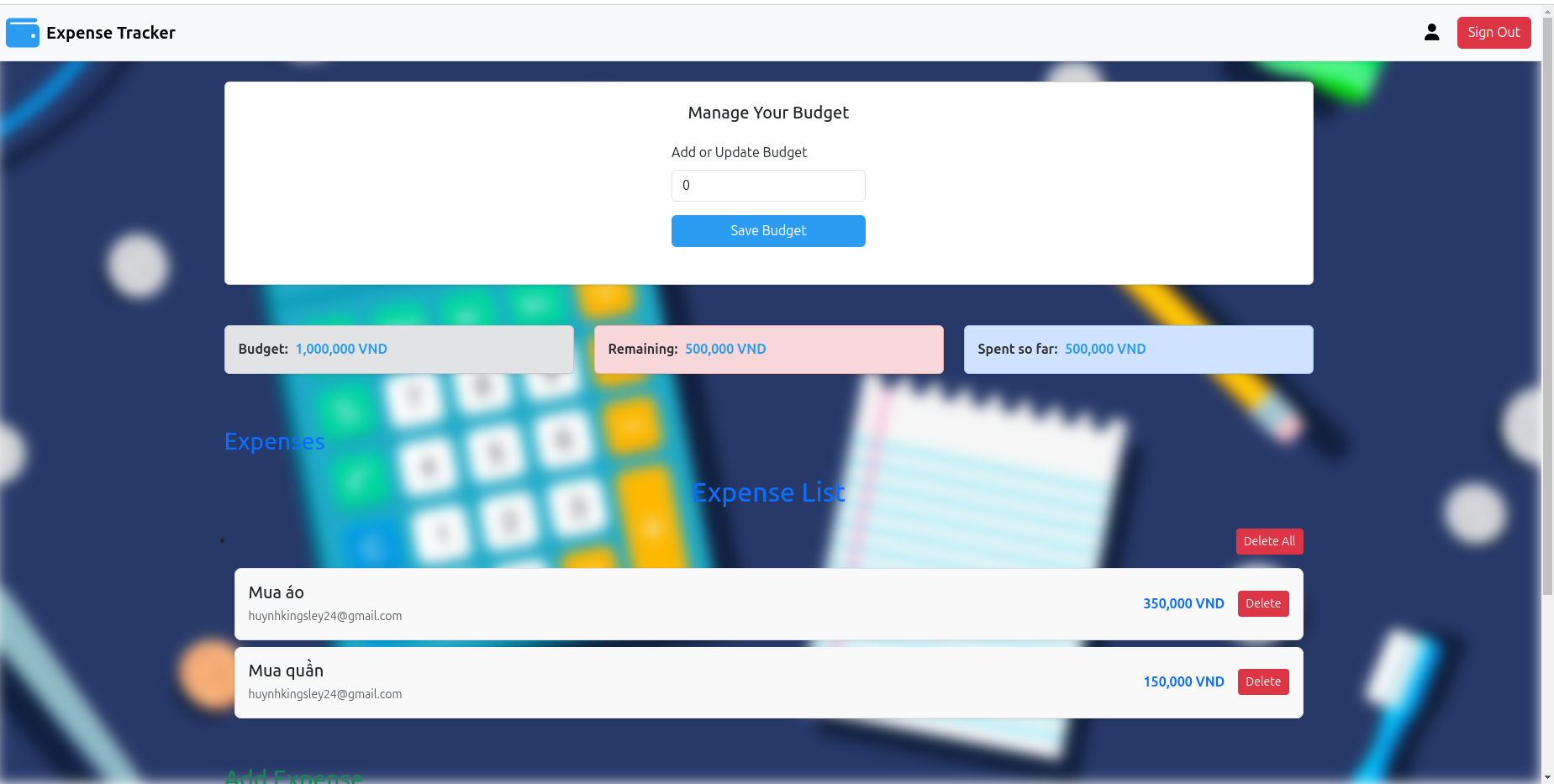


Hình 3.8 Giao diện email reset password



Hình 3.9 Giao diện thay đổi password

## 2.4. Giao diện trang chủ



Hình 3.10 Giao diện trang chủ của ứng dụng

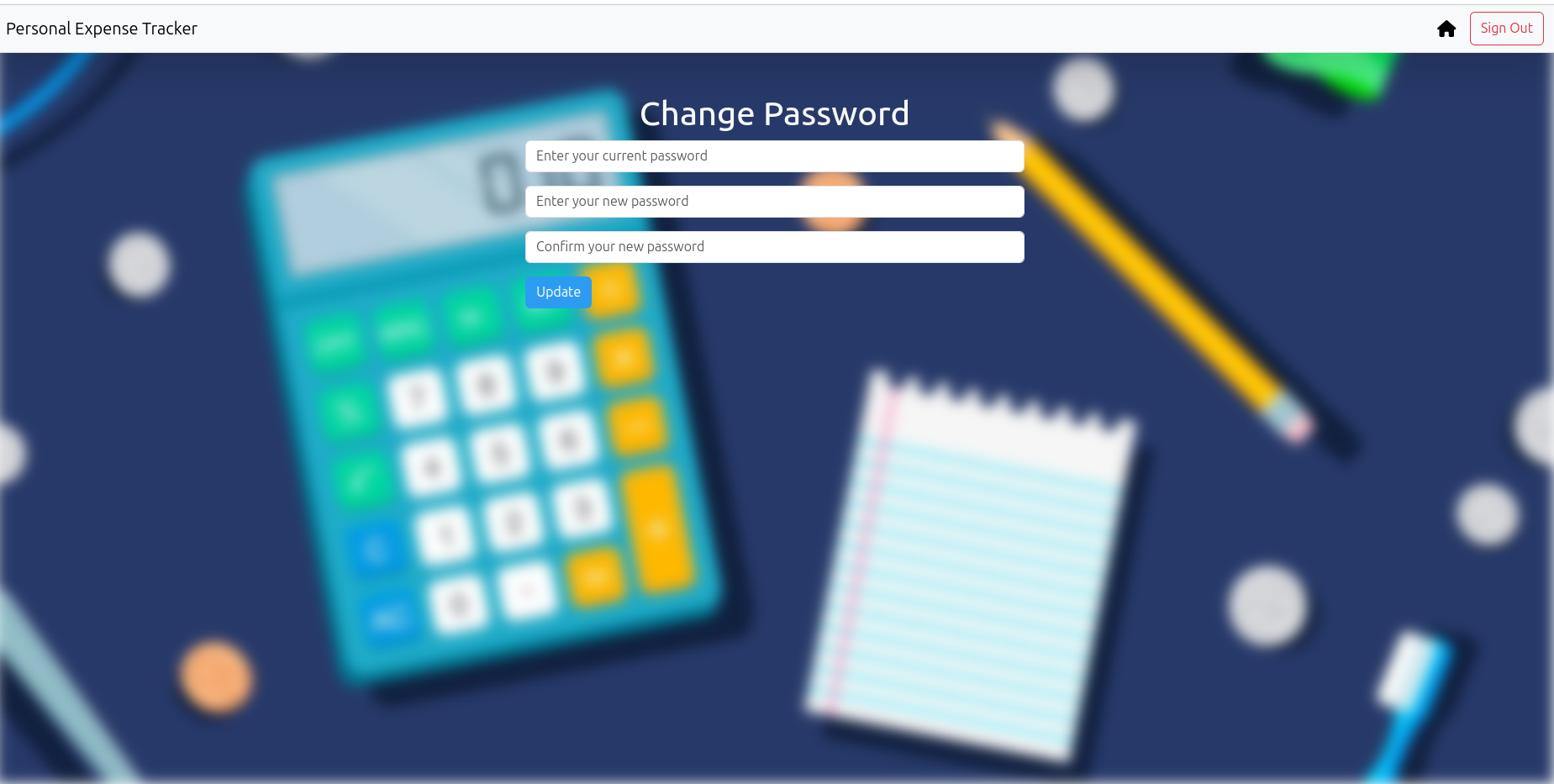
## 2.5. Giao diện Profile



Hình 3.11 Giao diện trang Profile

## 2.6. Giao diện Change Password

Đối với thay đổi password chỉ thực hiện được khi người dùng thực hiện đăng ký tài khoản trên ứng dụng, đối với người dùng đăng nhập bằng google sẽ được chuyển tiếp tới giao diện cài đặt thông tin của google.



Hình 3.12 Giao diện đổi password

# CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 1. Về kết quả đạt được

### 1.1. Về thực tiễn

Việc tích hợp SuperTokens vào hệ thống quản lý chi tiêu đã mang lại những cải tiến đáng kể trong việc xác thực và bảo mật người dùng. SuperTokens giúp đảm bảo quy trình đăng nhập, đăng ký diễn ra nhanh chóng, an toàn và dễ dàng mở rộng. Nhờ đó, hệ thống không chỉ bảo vệ dữ liệu người dùng mà còn cung cấp trải nghiệm liền mạch, đáng tin cậy cho người sử dụng. Điều này đóng vai trò quan trọng trong việc xây dựng lòng tin và tối ưu hóa hiệu suất vận hành của ứng dụng.

### 1.2. Về giao diện

Giao diện của hệ thống được thiết kế với sự sáng tạo, sử dụng màu sắc cổ điển để tạo ra không gian thân thiện và hấp dẫn cho người sử dụng. Màu sắc không chỉ nâng cao tính thẩm mỹ mà còn tăng cường trải nghiệm người dùng.

### 1.3. Về chức năng

Đồ án đã hoàn thiện được các chức năng cơ bản cần có của một websitequản lý chi tiêu.

## 2. Hạn chế

Do đề tài được thực hiện trong khoảng thời gian tương đối hạn chế và bản thân chưa có đủ kinh nghiệm về việc phân tích và thiết kế đối với một hệ thống toàn diện nên sẽ không thể tránh khỏi những thiếu sót và nhược điểm nhất định.

* Cơ sở dữ liệu còn đơn giản.
* Chưa thể phân tích và thiết kế các chức năng thống kê dành cho quản lý.
* Website chưa thích hợp hoạt động trên nhiều màn hình thiết bị.
* Chưa thực hiện thêm các loại chức năng đăng nhập khác

## 3. Hướng phát triển

Dưới đây là một số chức năng có thể được phát triển bổ sung vào hệ thống nhằm nâng cao chất lượng trải nghiệm của người dùng:

* Bổ sung các chức năng thống kê số liệu cho quản lý.
* Thiết kế giao diện tương thích với nhiều loại thiết bị và màn hình khác nhau.
* Bổ sung thêm nhiều lựa chọn đăng ký, đăng nhập cho người dùng

# TÀI LIỆU THAM KHẢO

[1] “How SuperTokens works | SuperTokens Docs.” Accessed: Nov. 24, 2024. [Online]. Available: https://supertokens.com/docs/thirdpartyemailpassword/architecture

[2] “How it works (flow diagram) | SuperTokens Docs.” Accessed: Nov. 24, 2024. [Online]. Available: https://supertokens.com/docs/thirdpartyemailpassword/common-customizations/sessions/about