BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC ĐIỆN LỰC**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO MÔN HỌC**

**ĐỒ ÁN LẬP TRÌNH WEB**

**ĐỀ TÀI:**

**XÂY DỰNG ỨNG DỤNG WEB BIÊN DỊCH MÃ NGUỒN TRÊN NỀN TẢNG NODEJS**

|  |  |
| --- | --- |
|  |  |
| |  |  | | --- | --- | | **Sinh viên thực hiện** | **: TRẦN THỊ DIỆU NINH** | | **Giảng viên hướng dẫn** | **: PHƯƠNG VĂN CẢNH** | | **Ngành** | **: CÔNG NGHỆ THÔNG TIN** | | **Chuyên ngành** | **: CÔNG NGHỆ PHẦN MỀM** | | **Lớp** | **: D10CNPM** | | **Khóa** | **:** | |  |

***Hà Nội, tháng 06 năm 2019***

**LỜI MỞ ĐẦU**

Hiện nay, công nghệ phát triển Web đang có sự phát triển vượt bậc và dần phủ song rộng khắp nơi. Điều này đã thực sự mang đến một kỷ nguyên mới cho thế giới, mang mọi người xích lại gần nhau hơn, đồng thời giúp cho việc giao lưu, hợp tác và trao đổi học hỏi kiến thức cũng trở nên thuận tiện hơn rất nhiều. Cùng với đó là sự ra đời và phát triển nhanh chóng của NodeJS đã tạo một cú hích rất lớn đối với các công nghệ phát triển Web. Nó là một nền tảng cho phép những nhà phát triển tạo ra website tương tác thời gian thực một cách nhanh chóng và hiệu quả. Bên cạnh đó, học lập trình trực tuyến đang trở thành phương pháp học phổ biến. Thay vì phải cài đặt các ide, editor cần thiết cho việc lập trình tại máy tính cá nhân, người học có thể lập trình trực tuyến ngay trên website, chia sẻ source code tới bạn bè, tăng khả năng tương tác, tiết kiệm tài nguyên máy tính. Do đó, trong đề tài đồ án môn học này, em muốn xây dựng một ứng dụng Web hỗ trợ việc biên dịch mã nguồn trên nền tảng NodeJS. Nội dung của đồ án sẽ tập trung trình bày những đặc điểm cơ bản của NodeJS, JavaScript, MySQL, Express và một số công nghệ phát triển web khác, các bước cơ bản để xây dựng một ứng dụng web trên nền tảng NodeJS và điểm mạnh của nó.

**LỜI CẢM ƠN**

Để hoàn thành quá trình nghiên cứu và hoàn thiện đồ án này, lời đầu tiên em xin gửi lời cảm ơn chân thành sâu sắc tới toàn thể thầy cô giáo khoa công nghệ thông tin trường Đại học Điện lực đã tận tình giảng dạy, giúp đỡ, truyền đạt những kiến thức, kinh nghiệm quý báu trong suốt những năm học vừa qua.

Đặc biệt, em xin gửi lời cảm ơn chân thành đến thầy giáo Phương Văn Cảnh đã tận tình chỉ bảo, giúp đỡ, định hướng cho em để hoàn thành tốt đồ án này. Trong thời gian qua, em đã học được rất nhiều những kiến thức bổ ích cùng tinh thần, thái độ làm việc nghiêm túc, hiệu quả từ thầy cô. Đây là những kinh nghiệm vô cùng quí báu cho em trong quá trình học tập và công tác sau này.

Tuy nhiên vì kiến thức chuyên môn còn hạn chế và bản thân còn thiếu nhiều kinh nghiệm thực tiễn nên nội dung của đồ án còn một số thiếu sót, em rất mogn nhận được sự góp ý, chỉ bảo thêm của các thầy cô.

Xin chân thành cảm ơn!

Sinh viên thực hiện

Trần Thị Diệu Ninh

**CHƯƠNG 1: PHẦN MỞ ĐẦU**

* 1. **Đặt vấn đề**

Những năm gần đây chúng ta đã thấy sự phát triển vượt bậc và độ phủ song rộng khắp của công nghệ phát triển Web. Chưa khi nào thế giới trở nên gần với nhau như vậy thông qua các trang Web được tạo lập hàng ngày, hang giờ. Sự ra đời và phát triển của công nghệ Web đã thực sự mang đến một kỷ nguyên mới cho thế giới, mang mọi người xích lại nhau hơn, đồng thời việc giao lưu, hợp tác và trao đổi học hỏi kiến thức cũng trở nên thuận tiện hơn bất cứ lúc nào.

Cùng với đó là sự phát triển nhanh chóng của NodeJS, một nền tảng dựa vào Chrome JavaScript runtime để xây dựng các ứng dụng Web. NodeJS rất hiệu quả khi xây dựng các ứng dụng Web thời gian thực có tính tương tác cao. Trong các framework của NodeJS, Express được xem là framework quan trọng nhất.

Bên cạnh đó, học tập luôn là vấn đề nóng nhận được sự quan tâm lớn của xã hội. Phương pháp học luôn là điều quan trọng với người học. Chính vì thế các phương pháp học luôn được tìm tòi và đổi mới. Trong những năm trở lại đây, học trực tuyến đang dần trở thành phương pháp phổ biến. Học trực tuyến có thể học mọi nơi, mọi lúc, mở rộng việc thảo luận, chia sẻ bài vở qua các nhóm, diễn đàn học tập.

Trong bối cảnh Internet phát triển, các thiết bị như máy tính, điện thoại thông minh trở nên phổ biến, học sinh dễ dàng tiếp cận với cách học mới. Việc học trực tuyến đang bộc lộ những ưu thế vượt trội hơn phương pháp cũ, giúp người học chủ động học mọi lúc, mọi nơi, tiết kiệm thời gian. Ngoài ra cách học này giúp người học dễ dàng tự do trao đổi vài vở, tương tác bạn bè nhiều hơn.

Từ những nhu cầu thiết thực cho cuộc sống, em đã chọn đề tài “**Xây dựng ứng dụng Web hỗ trợ biên dịch mã nguồn trên nền tảng NodeJS**” làm đề tài cho đồ án môn lập trình Web. Web cung cấp trình soạn thảo và hỗ trợ biên dịch nhiều ngôn ngữ lập trình, chia sẻ mã nguồn nhanh chóng.

* 1. **Mục tiêu đồ án**

Mục tiêu của đồ án là vừa nghiên cứu lý thuyết học các kiến thức mới, vừa triển khai xây dựng ứng dụng Web trên thực tế. Sau khi hoàn thành đồ án, một là nắm được quy trình phát triển một ứng dụng Web. Hai là hiểu và vận dụng được các công nghệ phát triển Web mới hiện nay để áp dụng vào dự án Web của mình. Ba là triển khai ứng dụng Web trên thực tế để hỗ trợ việc lập trình trực tuyến dễ dàng, thuận tiện cho mọi người.

* 1. **Cấu trúc đồ án**

Nội dung đồ án xoay quanh các vấn đề được trình bày ở các chương như sau:

**Chương 1: Phần mở đầu:** Trình bày tổng quan về tình hình phát triển Web, sự phát triển vượt bậc và nhanh chóng của NodeJS. Giới thiệu lý do chọn đề tài, mục tiêu, nội dung và ý nghĩa thực tiễn của đồ án.

**Chương 2: Cơ sở lý thuyết và công nghệ:** Giới thiệu về ứng dụng Web, các khái niệm, đặc điểm, điểm mạnh, các kiểu ứng dụng web phổ biến và cấu trúc cơ bản của một ứng dụng web. Giới thiệu về NodeJS, JavaScript, Express và MySQL.

**Chương 3: Phân tích yêu cầu bài toán:** Phân tích bài toán đặt ra gồm mô tả bài toán, thu thập và phân tích yêu cầu.

**Chương 4: Giải quyết bài toán:** Thiết kế và triển khai ứng dụng web trên thực tế.

**Chương 5: Kết luận và hướng phát triển cho đề tài:** Tóm tắt lại toàn bộ nội dung, các kết quả đạt được và hướng nghiên cứu tiếp theo.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ**

**2.1 Tổng quan về ứng dụng Web**

* + 1. **Khái niệm ứng dụng Web**

Một ứng dụng web (hay còn gọi là web app) là một ứng dụng chạy trên web thực hiện một số chức năng cơ bản như chia sẻ hình ảnh, tính toán, mua sắm mà có thể tiếp cận thông qua Internet.

Ứng dụng web sử dụng mô hình kiến trúc *client-server.* Kiến trúc này bao gồm máy chủ (server) chia sẻ tài nguyên với máy khách (client) qua mạng. Server có thể chia sẻ những tài nguyên như file, máy in, website, cơ sở dữ liệu và email. Server web là máy chủ chuyên chia sẻ trang web, còn trình duyệt web là ứng dụng phía người dùng để kết nối tới server web.

* + 1. **Đặc điểm của ứng dụng Web**

Ứng dụng web có những đặc điểm khác biệt so với website thông thường, một số những đặc điểm chính là: tính tương tác cao, nhiều chức năng. Ứng dụng web được dùng để thực hiện công việc, chức năng của một ứng dụng nhất định, có yêu cầu khó và tính quản trị lớn. Bên cạnh đó có thể phân tích hệ thống và xây dựng cơ sở dữ liệu đặc thù riêng biệt.

* + 1. **Ưu điểm của ứng dụng Web**

Hiện nay công nghệ điện toán đám mây phát triển mạnh mẽ được nhiều người lựa chọn sử dụng, nó chính là tiền đề phát triển ứng dụng web của các dịch vụ thiết kế website, có thể thấy việc phát triển ứng dụng web trên nền tảng website mang đến nhiều ưu điểm tích cực. Đó là người dung không phải cài đặt ứng dụng web, vì các thao tác kỹ thuật đều có thể truy cập và sử dụng bất kể lúc nào và ở đâu, chỉ cần kết nối Internet và kết nối với website có ứng dụng web cần thao tác là được. Thêm một điểm tích cực nữa là việc nâng cao hiệu quả làm việc nhóm, bảo mật dữ liệu rất tốt nhờ công nghệ lưu đám mây, từ đó giúp người sử dụng phá vỡ mọi khoảng cách địa lý khi học tập hay làm việc.

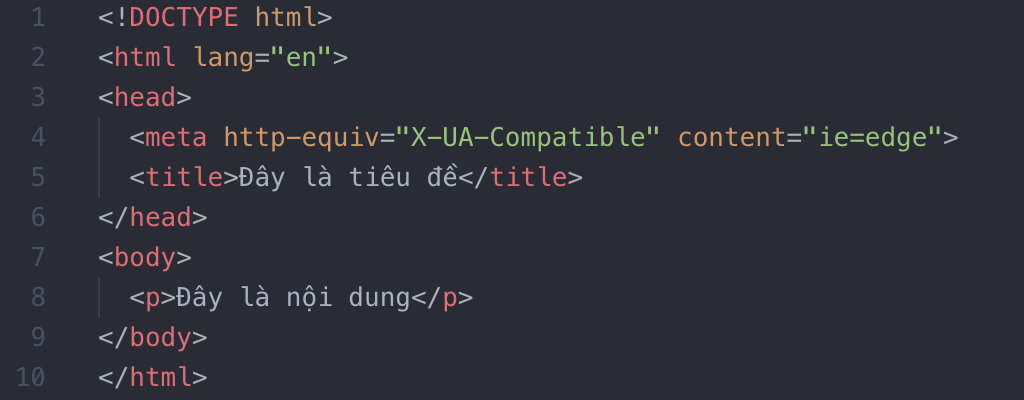
* + 1. **Các kiểu ứng dụng Web phổ biến**

Sự phân loại các ứng dụng web dựa trên cách các ứng dụng web thể hiện nội dung. Từ đó, ứng dụng web được chia thành 6 kiểu chính sau:

* Ứng dụng web tĩnh (static web application)
* Ứng dụng web động (dynamic web application)
* Cửa hàng online hoặc trang thương mại điện tử
* Cổng thông tin (Portal web app)
* Ứng dụng web hoạt hình (animated web application)
* Ứng dụng web với hệ thống quản lý nội dung CMS
  + 1. **Cấu trúc cơ bản của một ứng dụng Web**
  1. **Tổng quan về HTML**
     1. **Giới thiệu về HTML**

HTML là chữ viết tắt của cụm từ HyperText Markup Language (Ngôn ngữ đánh máy siêu văn bản) được sử dụng để tạo một trang web, trên một website có thể sẽ chứa nhiều trang và mỗi trang được quy ra là một tài liệu HTML. Cha đẻ của HTML là Tim Berners-Lee, cũng là người sinh ra World Wide Web và là chủ tịch của World Wide Web Consortium (W3C- tổ chức thiết lập ra các chuẩn trên môi trường Internet). Mỗi tài liệu HTML được hình thành bởi các phần tử HTML (HTML Element) được quy định bằng cặp thẻ (tag), các cặp thẻ này được bao bọc bởi một dấu ngoặc nhọn (ví dụ <html>) và thường sẽ được khai báo thành một cặp, bao gồm thẻ mở và thẻ đóng (ví dụ <strong></strong>). Các văn bản muốn được đánh dấu bằng HTML sẽ khai báo bên trong cặp thẻ. Một tập tin HTML sẽ bao gồm các phần tử HTML và được lưu lại dưới đuôi mở rộng là .html hoặc .htm.

* + 1. **Cấu trúc cơ bản của HTML**



Trong đó có các thành phần chính sau:

|  |  |
| --- | --- |
| **Thẻ** | **Miêu tả** |
| <!DOCTYPE…> | Thẻ khai báo một tài liệu HTML. Thẻ này xác định loại tài liệu và phiên bản HTML |
| <html> | Thẻ này bao phủ các tài liệu HTML đầy đủ |
| <head> | Thẻ đại diện cho đầu trang tài liệu mà có thể giữ các thẻ HTML như <title>, <link>... |
| <body> | Thẻ đại diện cho thân tài liệu và giữ các thẻ như <h1>, <p>... |
| <p> | Thẻ khai báo một đoạn văn bản |

Bảng 2.1 miêu tả một số thành phần chính của HTML gồm thẻ <!DOCTYPE> giúp khai báo một tài liệu HTML. Thẻ <html> là thẻ bao phủ các tài liệu HTML. Thẻ <head> đại diện cho đầu trang tài liệu chứa các thẻ thư viện cho HTML. Thẻ <body> đại diện cho thân tài liệu và giữ các thẻ định dạng văn bản. Các thẻ <p>, <div>, <h1> là các thẻ định dạng văn bản.

* + 1. **HTML được xử lý như thế nào?**

Khi một tập tin HTML được hình thành, việc xử lý nó sẽ do trình duyệt web đảm nhận. Trình duyệt sẽ đóng vai trò đọc hiểu nội dung HTML từ các thẻ bên trong và sẽ chuyển sang dạng văn bản đã được đánh dấu để đọc, nghe hoặc hiểu.

* 1. **Tổng quan về NodeJS**
     1. **Giới thiệu về JavaScript**

JavaScript là một ngôn ngữ lập trình của HTML và web được sử dụng phổ biến nhất như là một phần của các trang web giúp tạo ra các web động. JavaScript là một chương trình thông dịch dựa trên hướng đối tượng.

Ban đầu, JavaScript được phát triển bởi Brenda Eich, một lập trình viên của Netscape vào khoảng thời gian 1995-1996. Thời điểm đó, ngôn ngữ này được gọi là LiveScript. Sau đó các chuyên gia marketing của công ty đã đặt lại tên ngôn ngữ này thành JavaScript bởi vì sự phổ biến của Java lúc bấy giờ.

* + 1. **Ưu điểm của JavaScript**

Các lợi thế của việc sử dụng JavaScript là:

* **Sự tương tác Server ít hơn:** Ta có thể xác nhận đầu vào người sử dụng trước khi gửi trang tới Server. Điều này làm tiết kiệm lưu lượng tải ở Server, nghĩa là Server của bạn tải ít hơn.
* **Phản hồi ngay lập tức tới khách truy cập:** Họ không phải chờ một trang web tải lại để thấy xem nếu họ quên nhập một thông tin.
* **Tăng khả năng tương tác:** Có thể tạo các giao diện mà có thể phản ứng lại khi người dùng di chuyển chuột qua hoặc kích hoạt thông qua bàn phím.
* **Giao diện phong phú hơn:** Có thể sử dụng JavaScript để tạo các thành phần kéo, thả và các con trượt (Slider) để cung cấp một giao diện giàu tính năng hơn tới người dùng.
  + 1. **Hạn chế của JavaScript**

Ta không thể coi JavaScript như một ngôn ngữ chương trình chính thức (full-fieldged) do thiếu các tính năng quan trọng sau:

* Client-side JavaScript không cho phép đọc và ghi file vì lý do bảo mật.
* JavaScript không được sử dụng cho việc kết nối mạng các ứng dụng vì không có những hỗ trợ có sẵn.
* JavaScript không có khả năng xử lý đa luồng hoặc đa xử lý.
  + 1. **Các công cụ phát triển JavaScript**

Một trong những điểm mạnh lớn nhất của JavaScript là không yêu cầu các công cụ phát triển tốn kém. Ta có thể bắt đầu với một bộ biên soạn (Editor) văn bản đơn giản như Notepad hoặc sử dụng ngay trên trình duyệt.

Bản thân JavaScript là một ngôn ngữ linh động. Các nhà phát triển đã viết ra một số lượng lớn các công cụ của JavaScript, mở ra một lượng lớn tính năng bổ sung như sau:

* Giao diện lập trình ứng dụng trên trình duyệt (API) - Các API được xây dựng bên trong các trình duyệt web, cung cấp tính năng như tạo HTML động, cài đặt CSS, thu thập và điều khiển video trực tiếp từ webcam của người dùng hoặc sinh ra đồ hoạ 3D và các mẫu audio.
* Các API bên thứ ba cho phép nhà phát triển kết hợp tính năng trong website của họ từ người cung cấp nội dung khác như Twitter, Facebook.
* Từ các framework và thư viện bên thứ ba ta có thể áp dụng tới tài liệu HTML của mình, cho phép nhanh chóng xây dựng các trang web và ứng dụng.
  + 1. **Khái niệm về NodeJS**

NodeJS là một nền tảng Server side được xây dựng dựa trên JavaScript Engine (V8 Engine) được phát triển bởi Ryan Dahl năm 2009 và phiên bản cuối cùng là v0.10.36. Định nghĩa NodeJS bởi tài liệu chính thức như sau:

*Node.js là một nền tảng dựa vào Chrome JavaScript runtime để xây dựng các ứng dụng nhanh, có độ lớn. Node.js sử dụng các phần phát sinh các sự kiện (event-drivent), mô hình non-blocking I/O để tạo các ứng dụng nhẹ và hiệu quả cho các ứng dụng về dữ liệu thời gian thực chạy trên các thiết bị phân tán.*

NodeJS là một mã nguồn mở, đa nền tảng cho phát triển các ứng dụng phía Server và các ứng dụng liên quan đến mạng. Ứng dụng NodeJS được viết bằng JavaScript và có thể chạy trong môi trường NodeJS trên hệ điều hành Window, Linux.

NodeJS cũng cung cấp cho chúng ta các module JavaScript đa dạng, có thể đơn giản hoá sự phát triển của các ứng dụng web sử dụng NodeJS với các phần mở rộng.

* + 1. **Đặc điểm của NodeJS**

Dưới đây là một số đặc điểm quan trọng khiến NodeJS trở thành sự lựa chọn hàng đầu trong phát triển phần mềm:

**Không đồng bộ và phát sinh sự kiện (Event Drivent):** Tất cả các APIs của thư viện NodeJS đều đồng bộ, nghĩa là không khoá (blocking). Điều này rất cần thiết vì NodeJS không bao giờ đợi một API trả về dữ liệu. Server chuyển sang một API sau khi gọi và có cơ chế thông báo về sự kiện của NodeJS giúp Server nhận được phản hồi từ các API gọi trước đó.

**Tốc độ rất nhanh:** Dựa trên V8 JavaScript Engine của Google Chrome khiến NodeJS xử lý rất nhanh trong các quá trình thực hiện code. Đó là một yêu cầu khá quan trọng khi phát triển các startup muốn tạo ra một sản phẩm lớn và muốn đảm bảo có thể mở rộng nhanh chóng, đáp ứng được một lượng lớn người dùng khi trang Web đang phát triển.

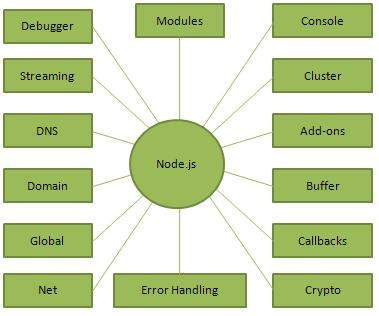
**Các tiến trình đơn giản nhưng hiệu năng cao:** NodeJS sử dụng một mô hình đơn luồng (single thread) với các sự kiện lặp. Các cơ chế sự kiện giúp Server trả lại các phản hồi một cách không khoá và tạo cho Server hiệu quả cao ngược lại với các cách truyền thống tạo ra một số lượng luồng hữu hạn để quản lý các request. NodeJS sử dụng các chương trình đơn luồng và các chương trình này cung cấp các dịch vụ cho số lượng request nhiều hơn so với các Server truyền thống như Apache HTTP Server.

**Không đệm:**  Ứng dụng NodeJS không lưu trữ các dữ liệu buffer.

**Có giấy phép:** NodeJS được phát hành dựa vào MIT License.

* + 1. **Các thành phần quan trọng trong NodeJS**

Lược đồ dưới đây mô tả các thành phần quan trọng của NodeJS



* + 1. **Ứng dụng của NodeJS**

Các lĩnh vực trong thực tế nên sử dụng NodeJS

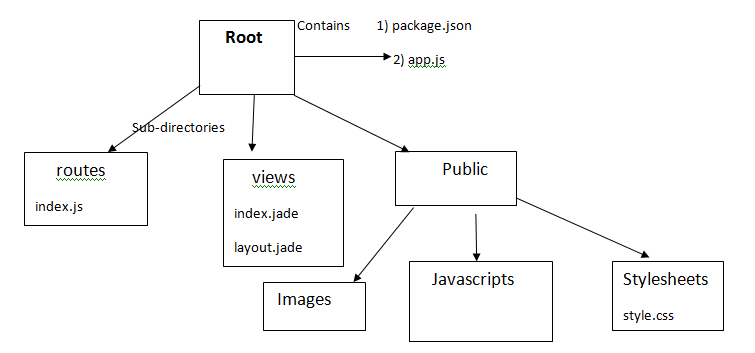
* Các ứng dụng về I/O
* Các ứng dụng về luồng dữ liệu
* Các ứng dụng về dữ liệu hướng đến thời gian thực
* Các ứng dụng dựa vào JSON APIs
* Các ứng dụng Single Page Application
  + 1. **Nhược điểm của NodeJS**

NodeJS không nên sử dụng trong các ứng dụng đòi hỏi về CPU.

* 1. **Tổng quan về ExpressJS**
     1. **Giới thiệu về ExpressJS**

Express là một framework nhỏ nhưng linh hoạt được xây dựng trên nền tảng của NodeJS. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Express có vô số các package hỗ trợ. Về performance, Express cung cấp thêm về các tính năng để lập trình tốt hơn chứ không làm giảm tốc độ của NodeJS. Hơn hết, các framework nổi tiếng của NodeJS hiện nay đều sử dụng ExpressJS như một core function, chẳng hạn: SailsJS, MEAN…

* + 1. **Cấu trúc của ExpressJS**



Cấu trúc của ExpressJS vô cùng đơn giản

* Root
  + app.js chứa các thông tin về cấu hình, khai báo, các định nghĩa… để ứng dụng của chúng ta chạy được.
  + package.json chứa các package cho ứng dụng chạy (tương tự như composer.json hoặc Gemfile của PHP hoặc RoR)
* Folder routes: chứa các route có trong ứng dụng
* Folder view: chứa các view/template cho ứng dụng
* Folder public: chứa các file css, js, images… cho ứng dụng
  1. **Tổng quan về MySQL**
     1. **Giới thiệu về MySQL**

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập cơ sở dữ liệu trên Internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, MacOSX, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS…

MySQL là một trong những ví dụ rất cơ bản về Hệ quản trị cơ sở dữ liệu quan hệ sử dụng ngôn ngữ truy vấn có cấu trúc (SQL).

MySQL được sử dụng cho việc bổ trợ PHP, Perl và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl…

* + 1. **Ưu điểm của MySQL**
* Là hệ quản trị cơ sở dữ liệu quan hệ
* Hoạt động theo mô hình client/server, làm database server cho các client kết nối đến
* Hỗ trợ SQL tiêu chuẩn
* Có nhiều table format với đặc tính khác nhau, ví dụ MyISAM không hỗ trợ transaction, InnoDB hỗ trợ transaction…
* Có nhiều API, library hỗ trợ cho nhiều ngôn ngữ lập trình
* Hỗ trợ giao tiếp ODBC
* Có khả năng replication
* Có thể chạy trên nhiều platform: Mac OS X, Windows, Linux, Unix (HP-UX, Solaris…)
  + 1. **Nhược điểm của MySQL**

Do MySQL hỗ trợ các table format khác nhau, một số table format sẽ không hỗ trợ transaction (row locking), không hỗ trợ hot backup, không hỗ trợ định nghĩa kiểu dữ liệu, không hỗ trợ XML, không hỗ trợ các tính năng OLAP...

**CHƯƠNG 3: PHÂN TÍCH YÊU CẦU BÀI TOÁN**

**3.1 Mô tả bài toán**

Hệ thống ứng dụng biên dịch mã nguồn trực tuyến cho phép người sử dụng có thể chọn các ngôn ngữ lập trình để lập trình và biên dịch mã nguồn. Hệ thống còn cung cấp tính năng lưu trữ và chia sẻ mã nguồn.

**3.2 Thu thập yêu cầu**

Hệ thống cần đảm bảo các yêu cầu sau:

* **Yêu cầu chức năng:**

Hệ thống có chức năng đăng ký, đăng nhập. Khi thực hiện việc đăng ký, hệ thống sẽ hiển thị giao diện cho người dùng cung cấp các thông tin cần thiết để đăng ký tài khoản thành viên. Đăng ký thành công hệ thống sẽ báo kết quả ngay trên trình duyệt và người dùng cần đăng nhập lại để sử dụng các tài nguyên của trang web.

Khi thực hiện việc đăng nhập, hệ thống cung cấp ba phương thức đăng nhập chính là đăng nhập bằng tài khoản Facebook, Google hoặc đăng nhập thủ công bằng tài khoản đã tạo khi đăng ký. Khi đăng nhập thành công, người dùng sẽ nhận được thông báo sau đó sẽ được chuyển hướng đến trang web chính.

Tất cả các tài nguyên của trang web chỉ có thể sử dụng sau khi đã đăng nhập.

* **Yêu cầu phi chức năng:**

Hệ thống đảm bảo khả năng đáp ứng các nhu cầu, thao tác của người dùng. Thời gian phản hồi nhanh chóng, kịp thời.

Hệ thống có giao diện trực quan, dễ sử dụng, dễ quản lý. Hạn chế tối đa thao tác của người dùng.

Hệ thống có tính bảo mật dữ liệu tốt.

Hệ thống có độ tin cậy cao, dữ liệu trả về chính xác.

**3.3 Phân tích yêu cầu**

**3.3.1 Phân tích biểu đồ ca sử dụng**

**3.3.1.1 Đăng kí**

**Mô tả:** Là người dùng truy cập vào web, chọn phần đăng ký tài khoản.

**Luồng sự kiện:**

* **Luồng cơ bản:** 
  + Khi người dùng muốn sử dụng các tài nguyên của hệ thống thì cần đăng ký tài khoản.
  + Người dùng vào phần đăng ký và điền các thông tin đăng ký.
  + Hệ thống tiến hành việc xác minh thông tin đăng ký và thông báo kết quả cho người dùng.
* **Luồng thay thế:**
  + Khi người dùng nhập sai thông tin đăng ký thì thông báo lỗi và cho người dùng nhập lại thông tin.

**Yêu cầu đặc biệt:** Không

**Điều kiện trước:**

Hệ thống có phần click chọn để hiện popup đăng ký.

**Điều kiện sau:**

Khi người dùng đăng ký thành công có thể đăng nhập ngay để truy cập vào các tài nguyên của hệ thống.

**Mở rộng thêm:** Không

**3.3.1.2 Đăng nhập**

**Mô tả:** Là người dùng truy cập vào web, sử dụng tài khoản Facebook, Google hoặc tài khoản và mật khẩu cá nhân đã đăng ký để đăng nhập vào web.

**Luồng sự kiện**

* **Luồng cơ bản**
  + Khi người dùng muốn sử dụng tài nguyên của hệ thống thì cần đăng nhập trước
  + Người dùng vào phần đăng nhập, chọn phương thức đăng nhập
  + Hệ thống tiến hành việc xác minh thông tin đăng nhập và thông báo kết quả cho người dùng
  + Đăng nhập thành công hệ thống sẽ chuyển hướng trang sang màn hình chính.
* **Luồng thay thế**
  + Khi người dùng nhập sai thông tin đăng nhập thì thông báo lỗi và cho người dùng nhập lại thông tin.

**Yêu cầu đặc biệt:** Không

**Điều kiện trước**

Hệ thống có phần click để chọn popup màn hình đăng nhập

**Điều kiện sau**

Khi người dùng đăng nhập thành công có thể sử dụng và truy cập vào các tài nguyên cho phép.

**Mở rộng thêm:** Không

**3.3.1.3 Xem danh sách mã nguồn đã biên dịch**

**Mô tả:** Là người dùng sau khi đã đăng nhập, truy cập vào trang danh sách mã nguồn đã biên dịch.

**Luồng sự kiện:**

* **Luồng cơ bản:**
  + Người dùng đăng nhập vào hệ thống
  + Sau khi đăng nhập người dùng click chọn phần my source
* **Luồng thay thế:** Không

**Yêu cầu đặc biệt:** Không

**Điều kiện trước**

Hệ thống có phần click chọn để vào trang danh sách mã nguồn đã biên dịch

**Điều kiện sau**

Khi truy cập thành công vào trang danh sách mã nguồn đã biên dịch, người dùng có thể xem được tất cả các mã nguồn đã biên dịch của mình.

**Mở rộng thêm:** Không

**3.3.1.4 Biên dịch mã nguồn**

**Mô tả:** Là người dùng truy cập vào trang biên dịch mã nguồn, lựa chọn một ngôn ngữ lập trình để tiến hành lập trình.

**Luồng sự kiện:**

* **Luồng cơ bản:**
  + Người dùng click chọn một ngôn ngữ lập trình
  + Sau khi chọn xong, người dùng sẽ nhận được mẫu code trên trình soạn thảo.
* **Luồng thay thế:**
  + Khi người dùng chuyển trang hoặc thoát khỏi hệ thống thì bài lập trình sẽ bị huỷ nếu như người dùng chưa compile mã nguồn. Ngược lại, bài sẽ được lưu sau khi đã compile mã nguồn.

**Yêu cầu đặc biệt:** Không

**Điều kiện trước**

Hệ thống có phần click chọn ngôn ngữ lập trình để tiến hành lập trình.

**Điều kiện sau**

Sau khi biên dịch, người dùng sẽ nhận được kết quả biên dịch mã nguồn. Danh sách mã nguồn đã biên dịch sẽ được cập nhật.

**Mở rộng thêm:** Không

**3.3.1.5 Tạo mới mã nguồn**

**Mô tả:** Là người dùng muốn tạo mới mã nguồn.

**Luồng sự kiện:**

* **Luồng cơ bản:**
  + Người dùng click chọn phần tạo mới mã nguồn.
  + Sau khi chọn xong, hệ thống hiển thị trang tạo mới mã nguồn.
  + Người dùng tiến hành code
  + Sau khi người dùng click chọn compile, hệ thống sẽ biên dịch mã nguồn mới
* **Luồng thay thế:** Không

**Yêu cầu đặc biệt:** Không

**Điều kiện trước**

Hệ thống có phần click chọn tạo mới mã nguồn.

**Điều kiện sau**

Sau khi người dùng tạo mới và biên dịch thành công, danh sách mã nguồn đã biên dịch sẽ được cập nhật.

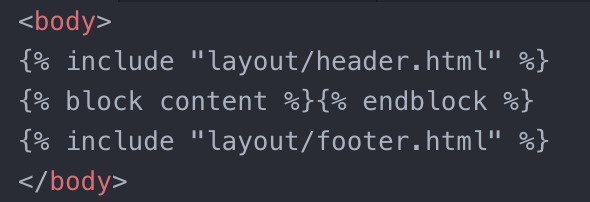
**Mở rộng thêm:** Không

**CHƯƠNG 4: GIẢI QUYẾT BÀI TOÁN**

**4.1 Thiết kế phần mềm**

**4.1.1 Thiết kế giao diện**

Giao diện chính của ứng dụng gồm các thành phần chính sau:



Trong đó:

* Header: thanh danh sách các button đến các trang cụ thể như: new code, my sources, about, login, register, profile
* Content: giao diện chi tiết các trang riêng biệt
* Footer: thông tin chính của website, các thông tin liên hệ

Giao diện chi tiết từng trang:

* Giao diện trang chủ:
* Hiển thị danh sách các button: new code, my sources, about, login, register, compile & run, profile
* Hiển thị editor, khung input, output, selector ngôn ngữ lập trình
* Giao diện trang mã nguồn đã biên dịch:
* Hiển thị danh sách mã nguồn, ngôn ngữ, ngày biên dịch
* Hiển thị khung search
* Giao diện trang cá nhân:
* Hiển thị thông tin cá nhân: tên người dùng, tài khoản đăng nhập.
* Hiển thị nút thay đổi thông tin cá nhân
* Giao diện trang chỉnh sửa thông tin cá nhân
* Hiển thị form cho người dùng thay đổi tên hiển thị, mật khẩu.

**4.1.2 Thiết kế cơ sở dữ liệu**

Ứng dụng sử dụng MySQL làm cơ sở dữ liệu. Danh sách các bảng trong cơ sở dữ liệu dưới dạng Schema trong NodeJS:

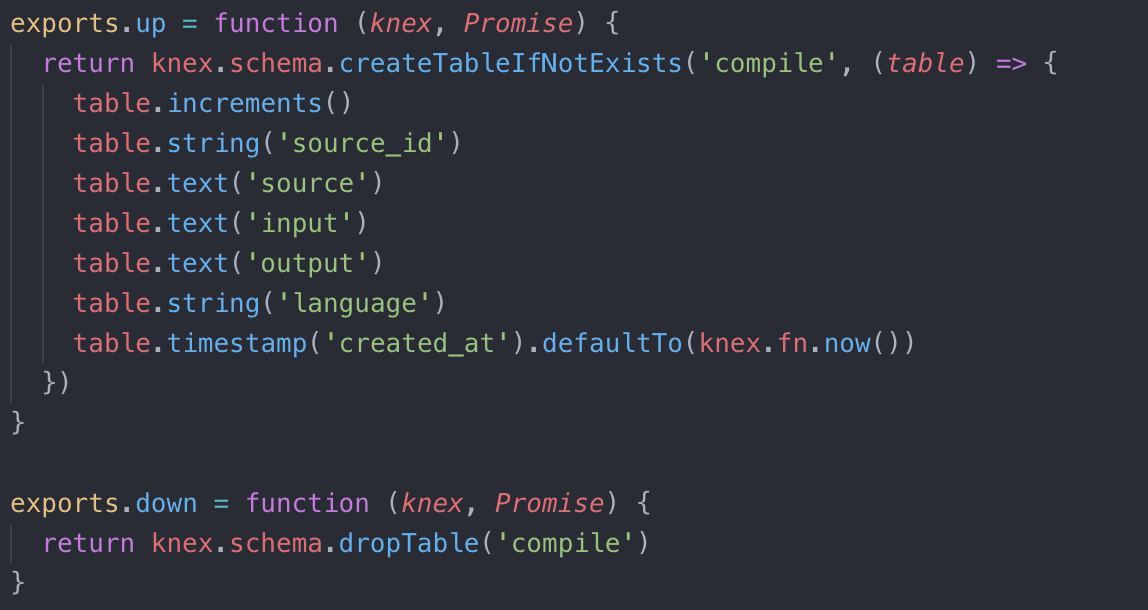
**4.1.2.1 Bảng users**: Lưu thông tin người dùng



Trong đó:

* id: là id của của người dùng được tạo, trường này bắt buộc và duy nhất, được tạo tăng tự động.
* email: địa chỉ email hay tên tài khoản của người dùng
* name: tên người dùng
* password: mật khẩu đăng nhập của người dùng
* token: token của tài khoản người dùng khi đăng nhập bằng phương thức khác như facebook, google account.
* created\_at: thời gian tạo tài khoản người dùng

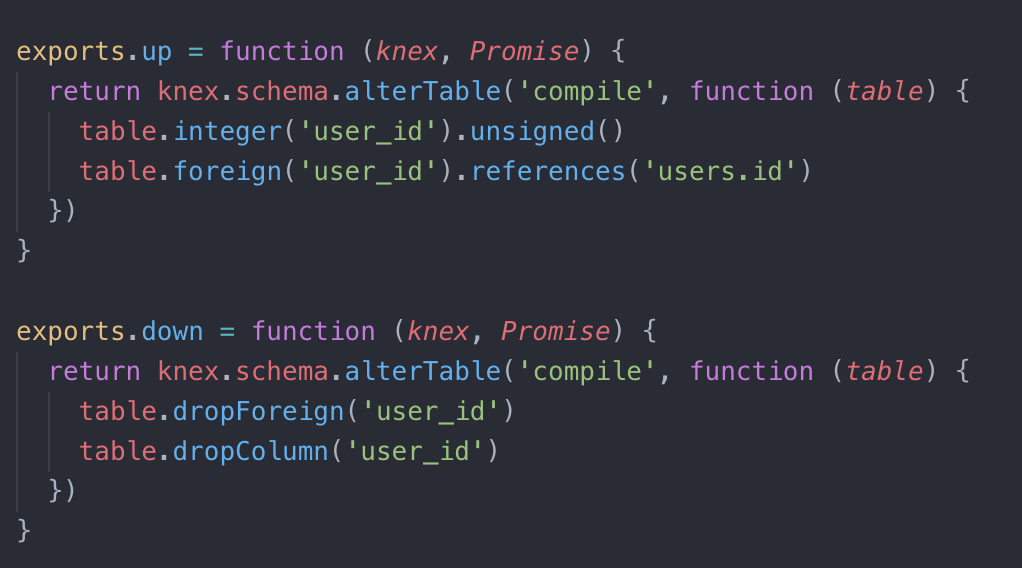
**4.1.2.2 Bảng compile**: Lưu thông tin mã nguồn đã biên dịch



Trong đó:

* id: là id của mã nguồn được biên dịch, trường này bắt buộc và duy nhất, được tạo tăng tự động.
* source\_id: id của mã nguồn được tạo ra từ API
* source: nội dung mã nguồn được viết bằng ngôn ngữ lập trình cụ thể
* input: dữ liệu đầu vào của mã nguồn
* output: kết quả biên dịch mã nguồn với dữ liệu đầu vào tương ứng
* language: ngôn ngữ lập trình của mã nguồn
* created\_at: thời gian biên dịch mã nguồn

**4.1.2.3 Bảng khoá ngoại của users và compiler**



**4.2 Triển khai phần mềm**

**4.2.1 Sử dụng template Nunjuck xây dựng giao diện**

Nunjuck là template engine đẹp và mạnh mẽ cho JavaScript, được lấy cảm hứng từ jinja2 nên nó có các tính năng về kế thừa các khối, điều khiển bất đồng bộ. Kích thước sau khi nén của nó khoảng 8KB nên load rất nhanh và hiệu suất rất cao. Nó có thể mở rộng rất tốt nhờ các bộ lọc và tiện ích mở rộng. Điểm mạnh lớn của nó là tương thích với mọi trình duyệt hiện nay.

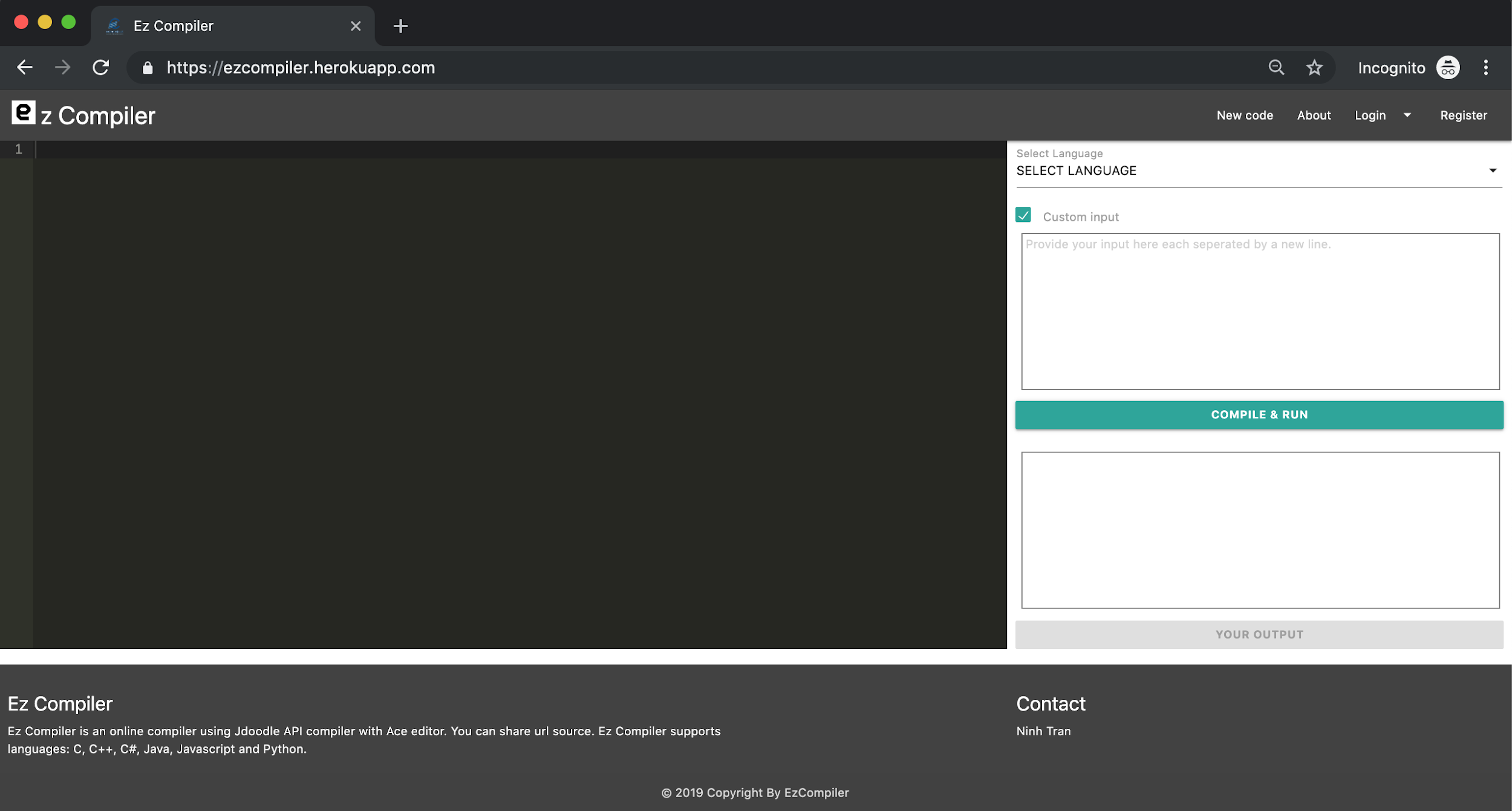
**4.2.1.1 Cấu trúc giao diện chính**



Trong đó:

* webTitle: tiêu đề trang hiện tại, giá trị này sẽ thay đổi theo trang hiện đang truy cập.
* block header: chứa nội dung hiển thị header của mỗi trang
* block content: chứa nội dụng chính hiển thị của mỗi trang
* block script: chứa các file thư viện javascript được viết trong quá trình tạo ứng dụng

Giao diện khi triển khai thực tế:



**4.2.1.2 Cấu trúc một trang giao diện cụ thể:**



Cấu trúc giao diện trang compile

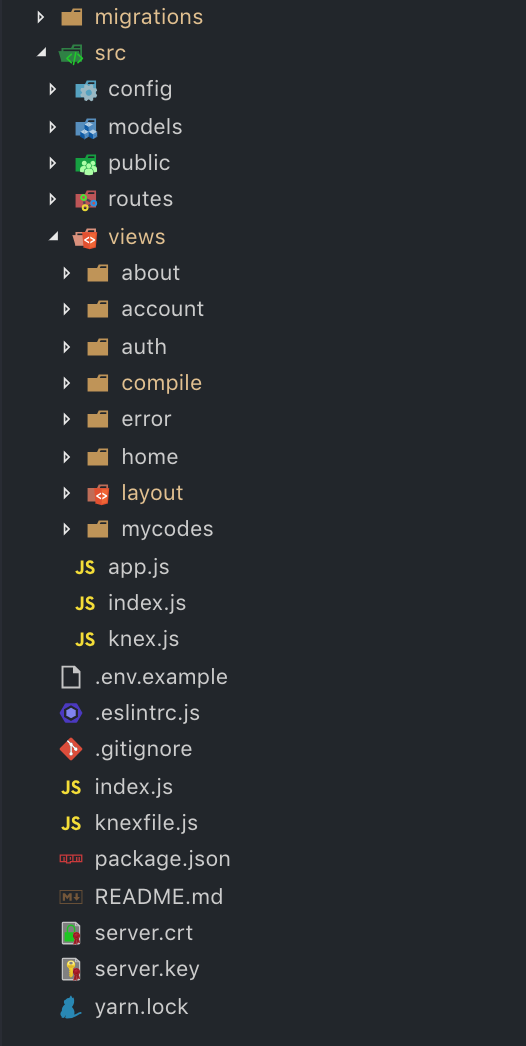
Trong đó:

* {% extend “layout/layout.html" %}: các trang cụ thể kế thừa lại trang layout.html chính
* Các block được định nghĩa lại

**4.2.2 Xây dựng Server bằng NodeJS**

Server được xây dựng sử dụng Express framework.

**4.2.2.1 Cấu trúc cây thư mục chính của ứng dụng**

****

Cấu trúc chính của ứng dụng gồm:

* Thư mục “node\_modules": thư mục chứa các thư viện mặc định và được cài thêm của ứng dụng
* Thư mục “src/views": thư mục chứa source code chính của ứng dụng. Thư mục này được phân chia thành các thư mục con được nhóm lại theo tính năng liên quan thành module. Việc nhóm các chức năng liên quan thành các khối liên kết này giúp cô lập code, dễ dàng cho việc test, tái sử dụng và bảo trì sau này. Thư mục này gồm các thành phần con chính sau:
* layout: Thư mục chứa các file quản lý layout chính và kiểm soát các hoạt động trên layout chính này.
* about: Thư mục chứa các file giới thiệu và hướng dẫn sử dụng ứng dụng
* account: Thư mục chứa các file quản lý tài khoản người dùng
* auth: Thư mục chứa các file quản lý việc đăng ký, đăng nhập
* compile: Thư mục chứa các file xử lý việc biên dịch mã nguồn
* error: Thư mục chứa các file xử lý ngoại lệ
* home: Thư mục chứa các file xử lý trang chủ
* mycodes: Thư mục chứa các file xử lý việc tạo lập và truy xuất lịch sử biên dịch mã nguồn.
* Thư mục “src/config": Thư mục này chứa các file cấu hình chính của ứng dụng
* Thư mục “src/public": Thư mục này chứa các thư mục và tài nguyên được công khai

**4.2.2.2 Các thư viện NodeJS được sử dụng**



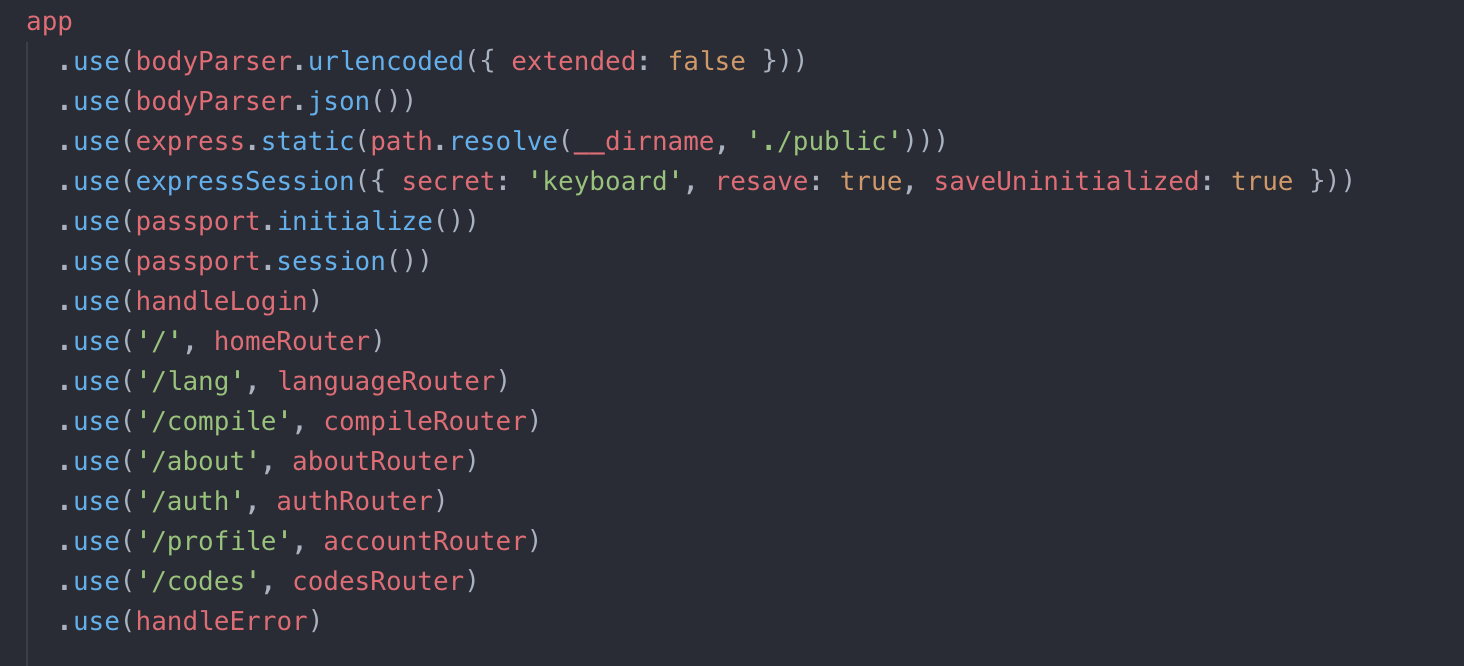
**4.2.2.3 Tạo ứng dụng chính**

Ứng dụng chính được đặt trong file “src/views/app.js". File gồm các thành phần:

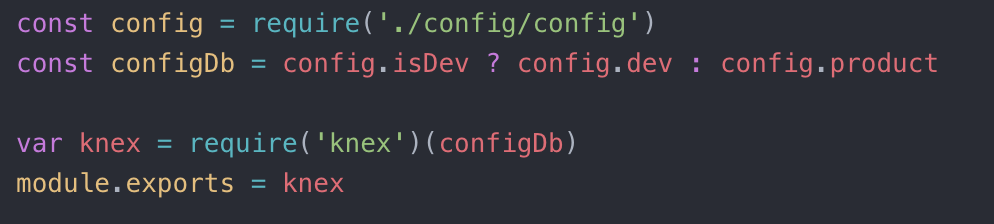
* Các module được cài đặt và tạo mới:



* Các middleware mới được thêm vào đối tượng app chính



* Sử dụng knex để kết nối tới database MySQL



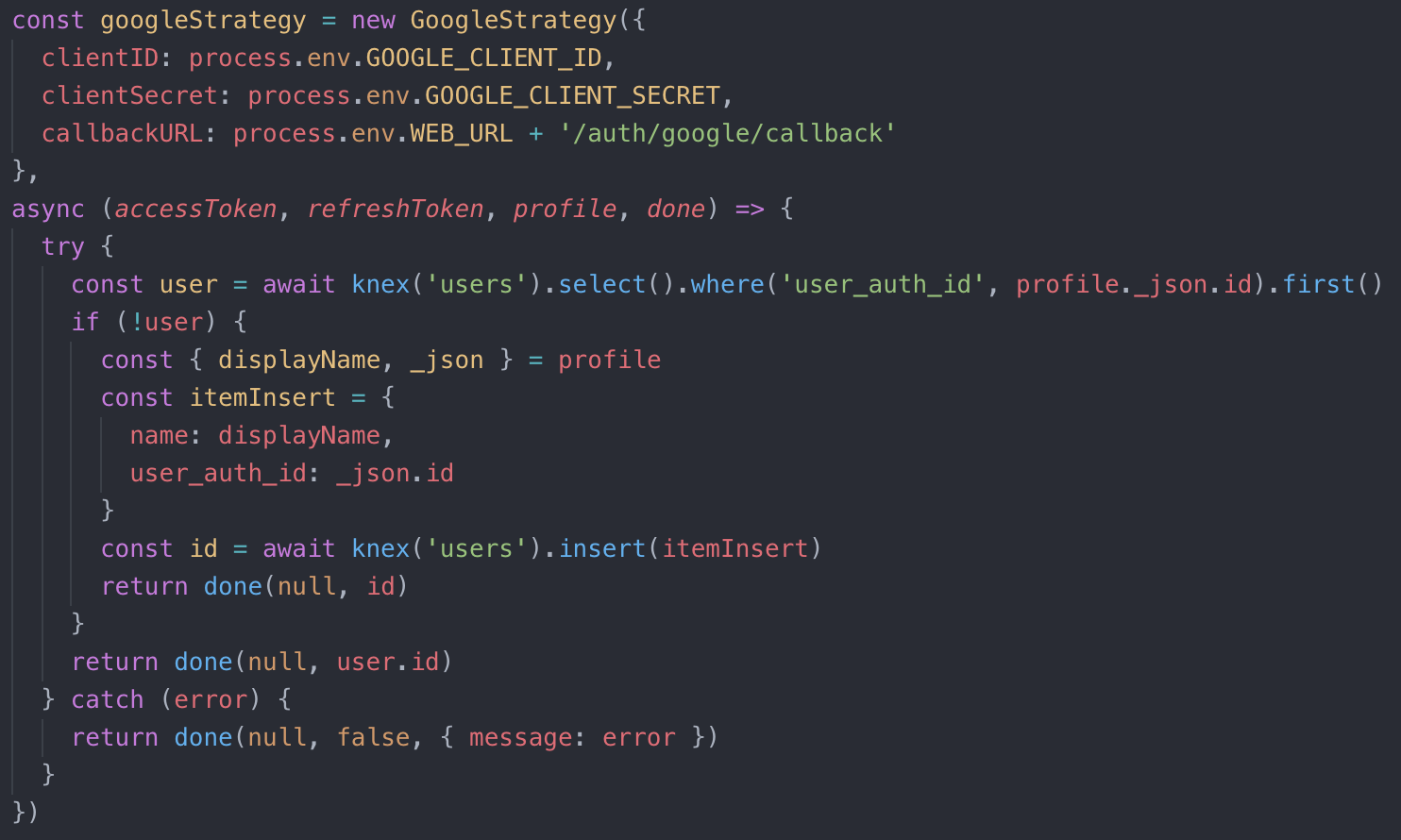


**4.2.2.4 Xử lý đăng ký, đăng xuất**

Ứng dụng sử dụng thư viện “passportjs” để xử lý việc xác thực thông tin người dùng. Đây là một trong những thư viện phổ biến nhất của NodeJS hỗ trợ xác thực người dùng. Nó được thiết kế là một middleware hết sức linh hoạt cho khả năng tuỳ biến cao với rất nhiều kịch bản xác thực như sử dụng Twitter, Facebook, Google và username-password trong database. Ứng dụng sử dụng ba phương thức xác thực đăng nhập là sử dụng tài khoản Facebook, Google và sử dụng username-password.

**Đăng nhập**

Đầu tiên, ta cần tạo một ứng dụng Google theo địa chỉ [https://console.developers.google.com](https://console.developers.google.com/). Sau khi tạo xong ứng dụng ra sẽ lấy được thông tin về Client ID và Client secret. Đây là hai thông tin quan trọng trong việc xác thực người dùng bằng tài khoản Google. Thông tin này được lưu lại trong file “config/passport.js" như sau:



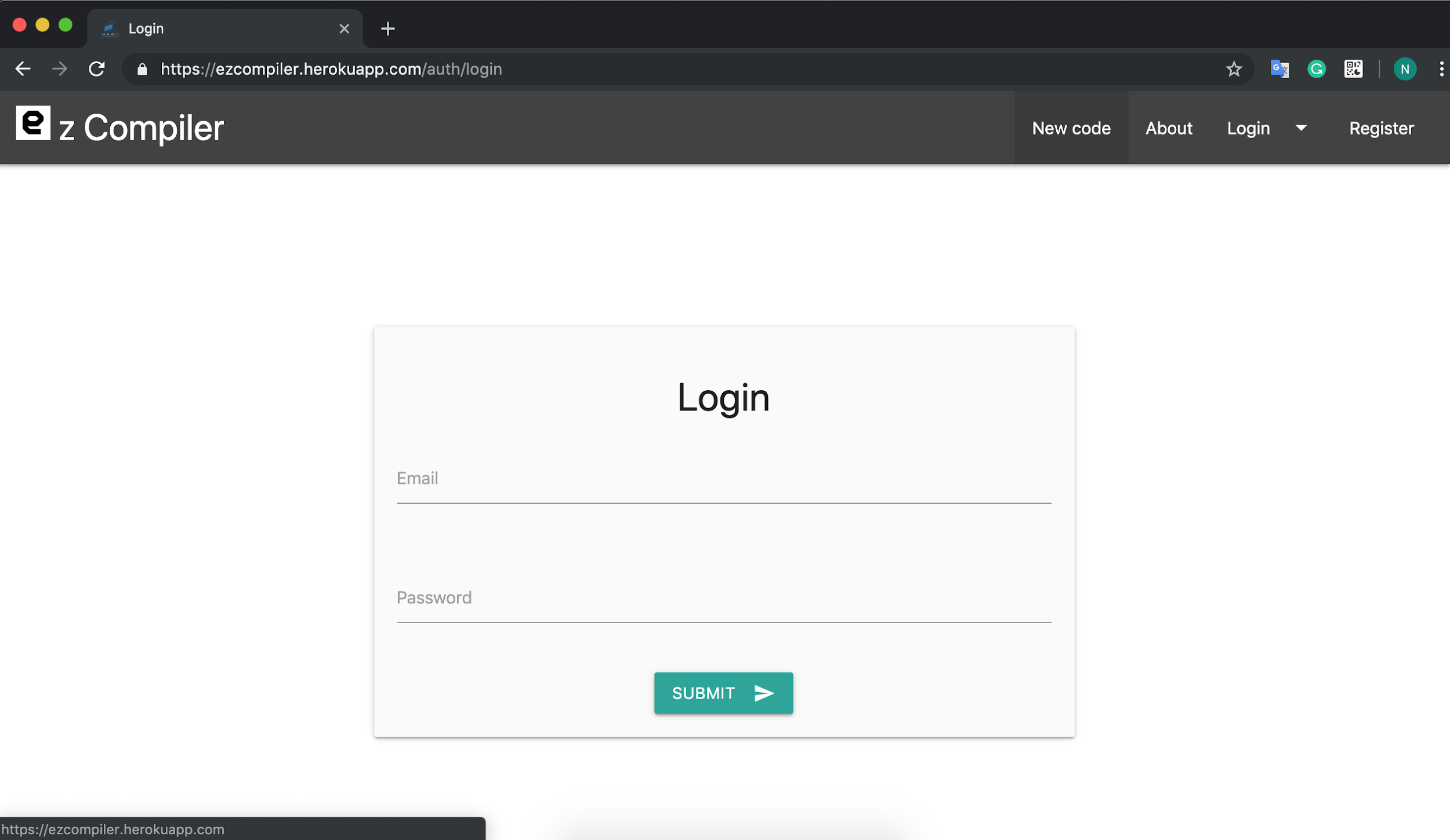
Các thông tin này được lấy từ biến môi trường *process.env.* Biến này được khởi tạo khi bắt đầu khởi chạy server và không thay đổi trong suốt quá trình chạy của ứng dụng. Sau khi có được thông tin, ta tiến hành khởi tạo các đối tượng GoogleStrategy và LocalStrategy. Tiếp đến ta thêm đối tượng này vào passport để thực hiện việc xác thực thông tin. Cuối cùng ta tạo các router và giao diện để thực hiện đăng nhập. Các router và hàm xử lý được đặt trong thư mục “src/routes/auth".

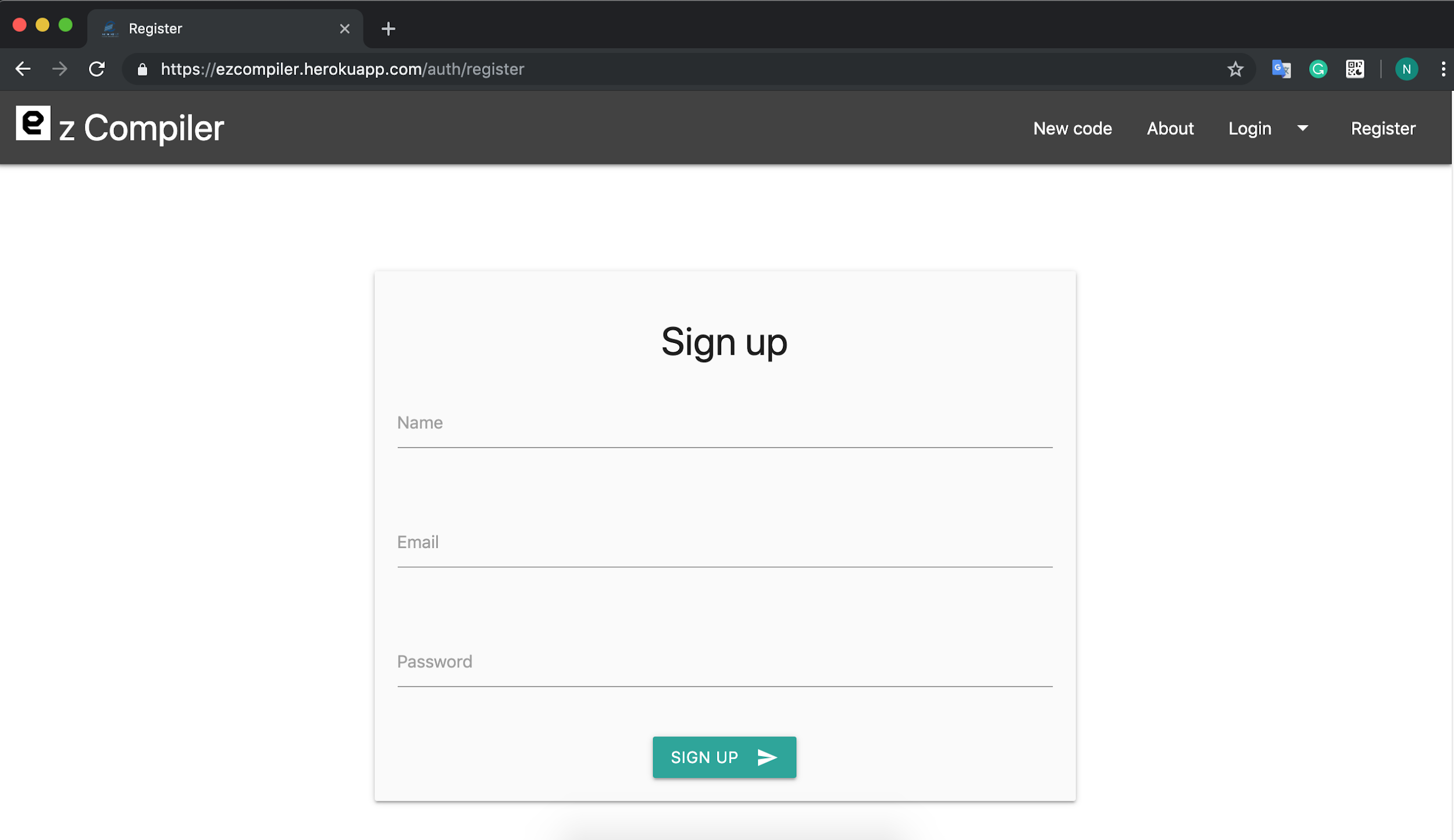
**Đăng ký**

Việc tạo đăng ký được thực hiện qua các bước sau:

* Tạo module users.
* Tạo form đăng ký người dùng.
* Xử lý thông tin đăng ký của người dùng. Nếu thông tin hợp lệ sẽ tạo người dùng mới và thông báo lại về phía người dùng. Nếu thông tin không hợp lệ sẽ báo lỗi.

Sau khi hoàn thành các trang đăng ký, đăng nhập có giao diện hiển thị như sau:

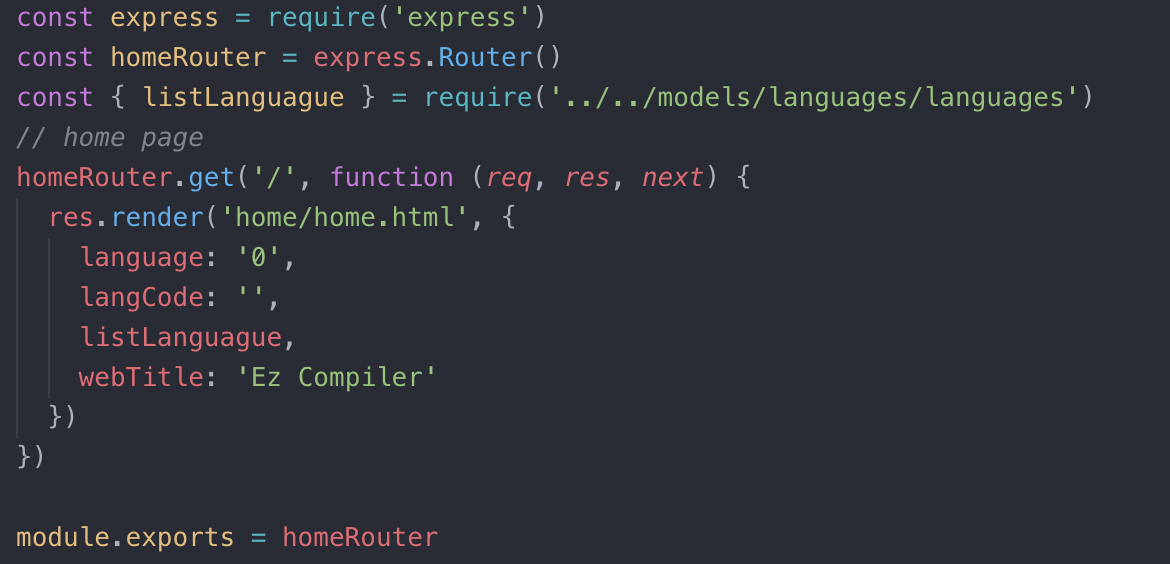




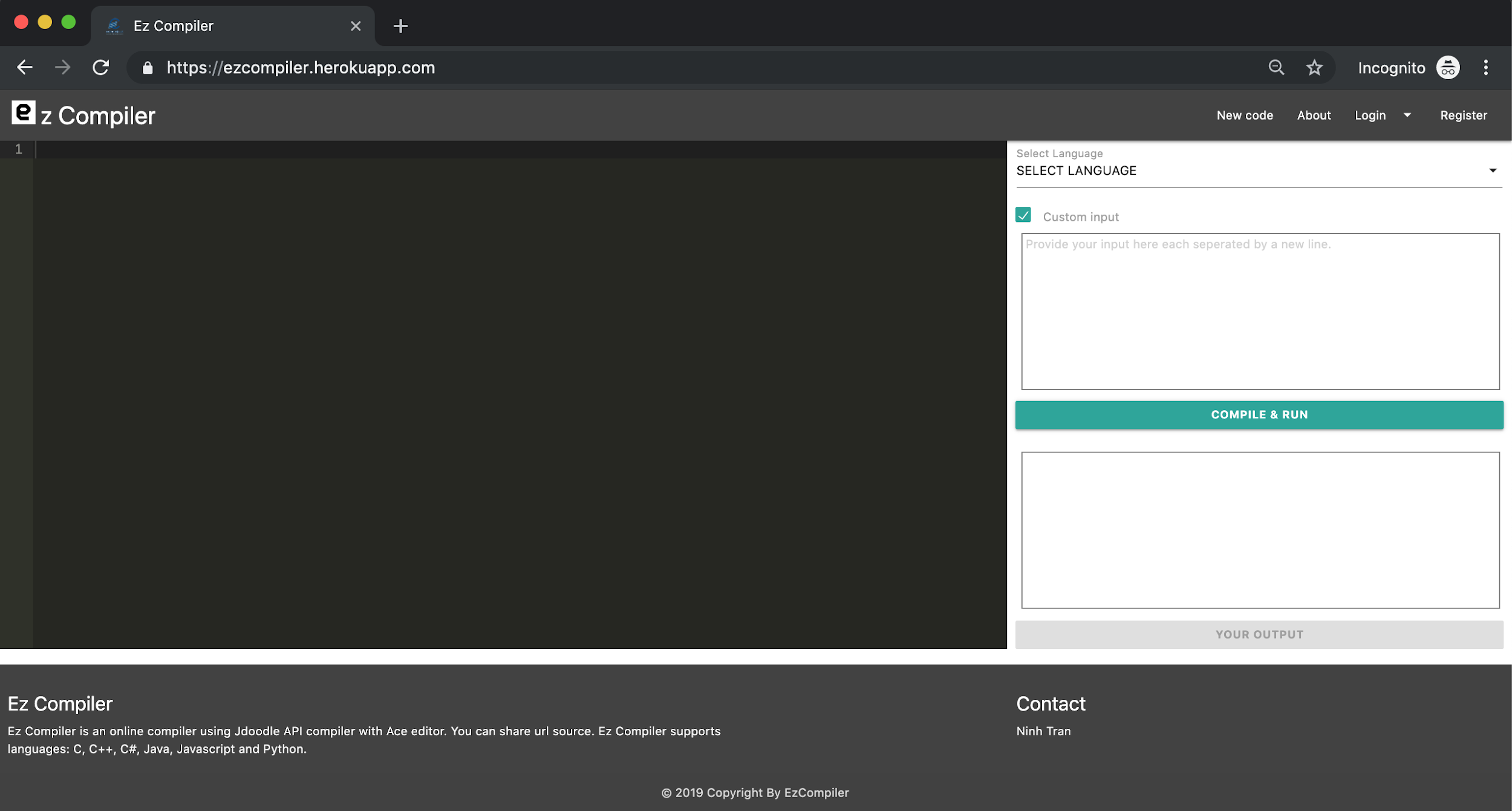
**4.2.2.5 Xây dựng trang chủ**

Việc xây dựng trang chủ được xử lý trong 2 file gồm:

* File “src/views/home/home.html”: File chứa mã HTML và JavaScript hiển thị giao diện và xử lý thông báo về phía người dùng.
* File “src/routes/home/index.js”: File chứa router trang chủ.

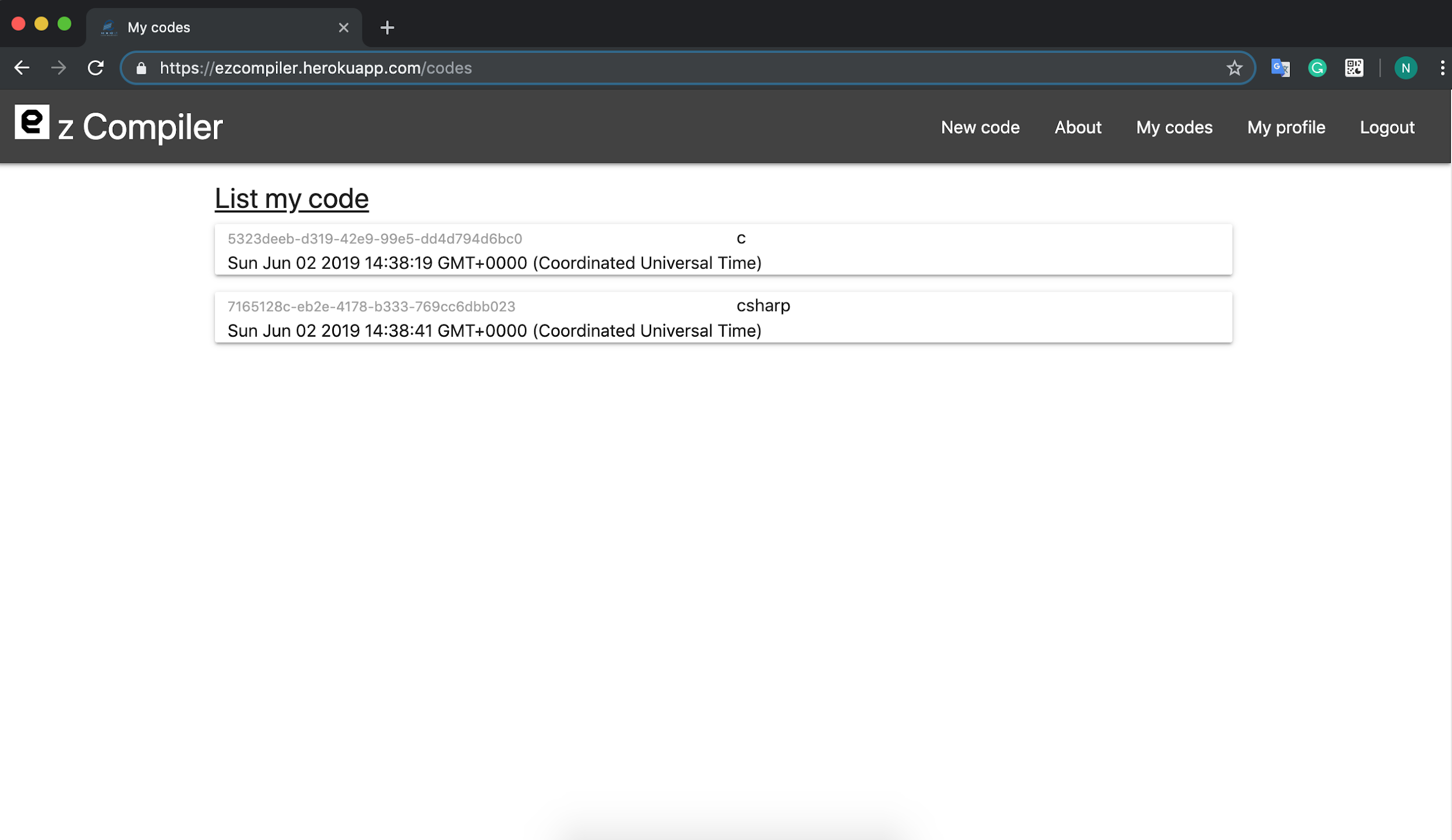


Giao diện trang chủ sau khi xây dựng:



**4.2.2.6 Xây dựng trang About, My codes, My profile**

Việc xây dựng các trang này cũng được xử lý tương tự phần trang chủ. Sau khi xây dựng, giao diện các trang About, My codes, My profile như sau:



**4.2.2.7 Đưa ứng dụng lên Heroku Cloud Application Platform**

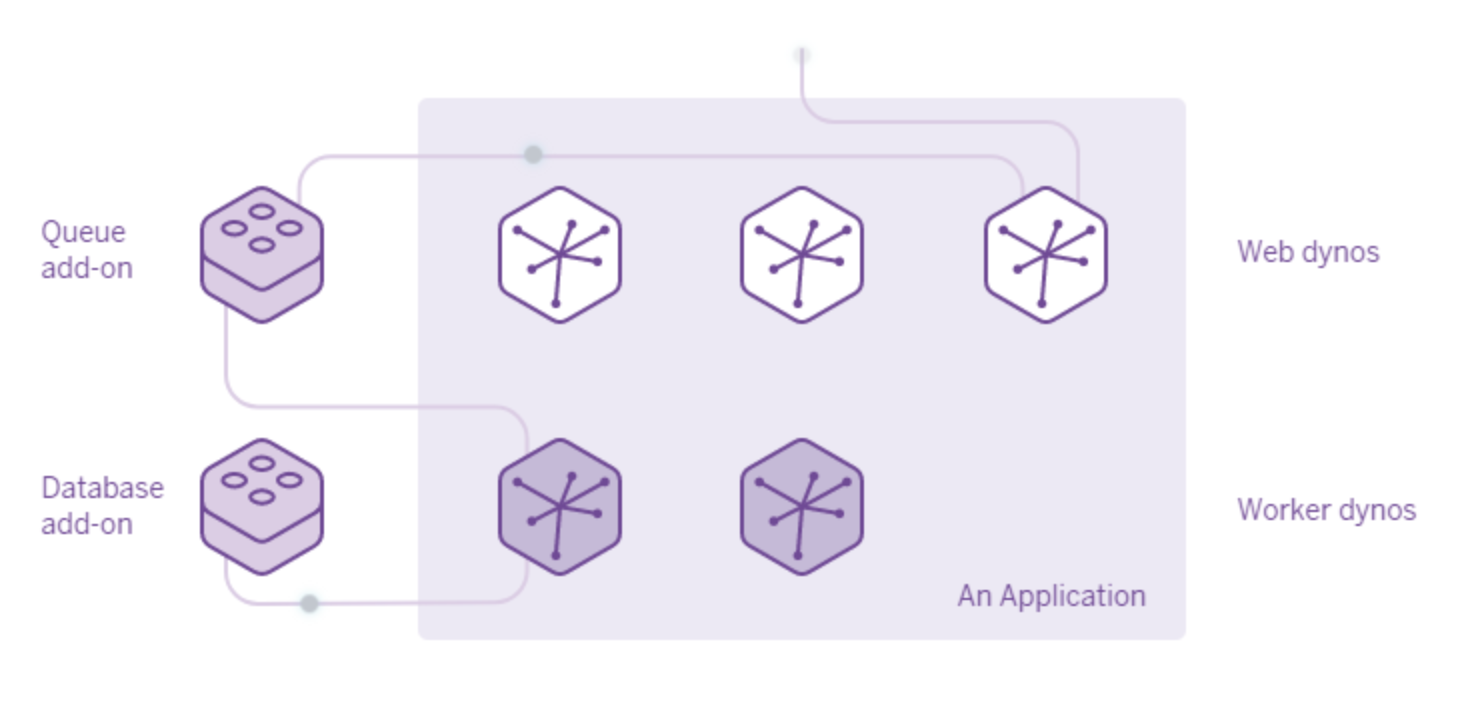
Heroku là một nền tảng đám mây dựa trên ứng dụng container dưới dạng dịch vụ (PaaS). Các nhà phát triển sử dụng Heroku để triển khai, quản lý và mở rộng các ứng dụng hiện tại. Nền tảng của họ rất linh hoạt và dễ sử dụng, cung cấp cho các nhà phát triển con đường đơn giản nhất để đưa ứng dụng của họ ra thị trường.

Heroku được quản lý hoàn toàn bởi Heroku, cho phép các nhà phát triển tự do tập trung vào sản phẩm cốt lõi của họ mà không bị phân tâm trong việc duy trì máy chủ, phần cứng hoặc cơ sở hạ tầng. Trải nghiệm Heroku cung cấp các dịch vụ, công cụ, quy trình làm việc và hỗ trợ tất cả được thiết kế để nâng cao năng suất của nhà phát triển ứng dụng.

Heroku được biết đến như một nơi để triển khai các ứng dụng trong dyno (Khi bạn triển khai một ứng dụng Heroku, bạn phải thiết lập một máy ảo được gọi là slug, sử dụng một hoặc nhiều buildpacks. Khi một máy ảo được khởi chạy từ slug, nó được gọi là dyno).

Trong trường hợp phải xử lý khối lượng công việc nhiều, phức tạp, bạn sẽ cần thêm nhiều block (scale chiều ngang) hoặc tăng kích thước các block (scale chiều dọc). Phí sử dụng Heroku được tính dựa trên số lượng dyno và kích thước mỗi dyno mà bạn dùng.

Mặc dùng bạn có thể phải trả phí để sử dụng Heroku, nhưng Heroku không thật sự lưu trữ ứng dụng của bạn. Thực tế thì cả nền tảng Heroku lẫn mọi ứng dụng phát triển trên đó đều được triển khai trên Amazon Web Services (AWS).



Việc triển khai thực tế ứng dụng được thực hiện qua các bước sau:

* Đưa source code của ứng dụng lên github: Source code của ứng dụng được lưu trữ tại địa chỉ

<https://github.com/dieuninh1997/CompilerOnline>

* Tạo tài khoản Heroku, tạo project trên Heroku.
* Dùng git để push source code local lên project Heroku.
* Sau khi kết nối Heroku, cài đặt các gói phần mềm cần thiết để khởi chạy Server NodeJS.

Sau khi hoàn thành, ứng dụng web đã được triển khai trên thực tế thông qua địa chỉ web: [https://ezcompiler.herokuapp.com](https://ezcompiler.herokuapp.com/)

**4.2.2.8 Thử nghiệm hệ thống với người dùng thật và lấy ý kiến đánh giá**

Sau khi triển khai ứng dụng trên thực tế thông qua địa chỉ web [https://ezcompiler.herokuapp.com](https://ezcompiler.herokuapp.com/), tiếp tục giới thiệu ứng dụng đến người dùng và tiếp nhận ý kiến phản hồi. Đối tượng tiếp cận ứng dụng chính là các em học sinh sinh viên theo học lập trình, việc giới thiệu được thực hiện thông qua các group trên facebook và giới thiệu qua bạn bè, người thân. Các ý kiến đánh giá của người dùng được tiếp nhận ngay, đánh giá và nêu ý kiến phù hợp tiến hành điều chỉnh lại ứng dụng.

**CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN CHO ĐỀ TÀI**

Qua quá trình tìm hiểu các ngôn ngữ lập trình, các thư viện, công cụ lập trình web để thực hiện đồ án: “Xây dựng ứng dụng web biên dịch mã nguồn trên nền tảng NodeJS", em đã đạt được những kết quả sau:

* Nghiên cứu học hỏi được thêm nhiều ngôn ngữ, công cụ, nền tảng lập trình web mới như JavaScript, NodeJS, MySQL. Biết thêm về những điểm mạnh, điểm yếu của từng công cụ, nền tảng trên.
* Nghiên cứu và nắm được quy trình xây dựng một ứng dụng web hoàn chỉnh.
* Tích luỹ thêm nhiều kinh nghiệm trong việc tìm hiểu tài liệu và xử lý khi gặp khó khăn.
* Tăng khả năng quản lý lỗi và debug được hoàn toàn một ứng dụng web.
* Tìm hiểu được thêm nhiều kiến thức về deploy ứng dụng trong thực tế.
* Tích luỹ thêm kinh nghiệm trong việc viết báo cáo và báo cáo tiến độ thường xuyên.
* Học hỏi được nhiều kiến thức, kinh nghiệm quý báu từ thầy cô hướng dẫn.

Kết quả đạt được sau khi thực hiện đồ án này là những kinh nghiệm kiến thức vô cùng quý báu với bản thân em. Mặc dù vậy ứng dụng vẫn còn một số phần có thể cải tiến sau này để trở nên hoàn chỉnh hơn như: giao diện người dùng, tích hợp thêm thông báo, cải thiện phần quản lý my codes, compile code của người dùng.

**Tài liệu tham khảo**

**Tiếng Việt**

[1] l Murach, Ray Harris, *Lập trình cơ bản PHP và MySQL*, Nhà xuất bản Bách Khoa Hà Nội, 2016, tr.4.

[2] Steve Suehring, *JavaScript - Hướng dẫn học qua ví dụ*, Nhà xuất bản Khoa học và Kỹ thuật, 2015, tr.6.