BỘ GIÁO DỤC VÀ ĐÀO TẠO

TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH

**KHOA CÔNG NGHỆ THÔNG TIN**

🙢⬥🙠



ĐỀ TÀI MÔN HỌC

**CÔNG NGHỆ WEB**

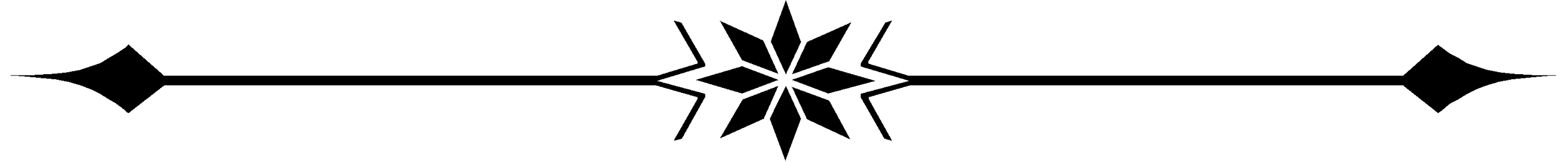
**TÌM HIỂU ANGULAR**

Giảng viên hướng dẫn: Lương Trần Hy Hiến

Lớp học phần:

Sinh viên thực hiện:

1. Lê Chu Báu – 4501104017
2. Lê Chí Hải - 4501104066
3. Trương Quốc Hiệp - 4501104081
4. Nguyễn Tuấn Phong - 4501104174
5. Phạm Nguyễn Hồng Nguyên - 4501104163

TP.HCM, ngày 10 tháng 09 năm 2021

**MỤC LỤC**

[LỜI CẢM ƠN 3](#_Toc86786952)

[a. Nội dung 3](#_Toc86786953)

[b. Yêu cầu cần giải quyết 3](#_Toc86786954)

[1. Khái niệm Angular 4](#_Toc86786955)

[2. Lịch sử phát triển Angular 5](#_Toc86786956)

[3. Kiến trúc của Angular 5](#_Toc86786957)

[1. Modules 6](#_Toc86786958)

[2. Components 6](#_Toc86786959)

[3. Templates 6](#_Toc86786960)

[4. Metadata 6](#_Toc86786961)

[5. Services 6](#_Toc86786962)

[6. Dependency Injection 7](#_Toc86786963)

[4. Ưu điểm và nhược điểm của AngularJS 7](#_Toc86786964)

[4.1. Ưu điểm 7](#_Toc86786965)

[4.2. Nhược điểm 8](#_Toc86786966)

[5. Tại sao chúng ta nên chọn Angular 8](#_Toc86786967)

[6. DEMO XÂY DỰNG ỨNG DỤNG CALCULATOR APP 11](#_Toc86786968)

# LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường trường Đại học đến nay, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô, gia đình, bạn bè. Với lòng biết ơn sâu sắc nhất, chúng em xin gửi đến quý thầy cô ở Khoa Công nghệ thông tin – Trường ĐH Sư Phạm TP.HCM với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường.

Sau cùng, chúng em xin kính chúc quý Thầy/Cô trong khoa Công nghệ thông tin thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

**NHIỆM VỤ ĐỀ TÀI**

Nội dung và các yêu cầu cần giải quyết trong nhiệm vụ đề tài tốt nghiệp

## Nội dung

Tìm hiểu Angular

## Yêu cầu cần giải quyết

Các yêu cầu chung:

* + Khái niệm về Angular
  + Ưu điểm của Angular
  + Demo ứng dụng

1. Khái niệm Angular

Angular được xem là một open source (mã nguồn mở) hay frameworks miễn phí chuyên dụng cho công việc thiết kế web. Angular được phát triển từ những năm 2009 và được duy trì bởi Google. Frameworks này được xem là frameworks front end mạnh mẽ nhất chuyên dụng với các lập trình viên cắt HTML cao cấp.



Angular được ứng dụng rộng rãi với mục đích xây dựng project Single Page Application (SPA). Hiện tại, Version stable của Angular là Angular 9 với TypeScript 3.6 và 3.7.

Trên thực tế, Angular sở hữu hai phiên bản hoàn toàn khác nhau như sau:

**Phiên bản 1 (AngularJS):** Phiên bản này có tên chính thức là AngularJS đến hiện nay nó vẫn đang được phát triển hoàn thiện bởi các chuyên gia. Vào năm 2018, thì phiên bản 1.7.3 đã được nhà sản xuất phát hành.

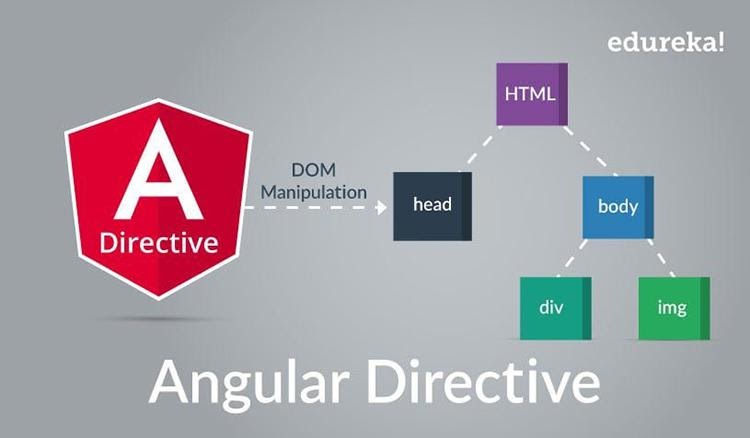
Đây là phiên bản được hoàn thành thực sự dựa trên Javascript bởi những lập trình viên viết mã theo ngôn ngữ Javascript thì nó thường dễ làm và dễ tìm kiếm Tuy nhiên, hiệu năng khi so sánh với phiên bản ReactJS thì vẫn còn nhiều công ty xây dựng phải cân nhắc sử dụng Angular với phiên bản khác 1.

**Phiên bản 2 (Angular):** Phiên bản này được gọi với cái tên chính thức là Angular. Nó sở hữu tư duy thay đổi hoàn toàn với phiên bản thứ 1 và mang lại tham vọng giúp cho Angular có thể đánh bại ReactJS. AngularJS được xem là một trong những framework sở hữu cấu  trúc ứng dụng dạng động và cho phép bạn có thể sử dụng HTML như là ngôn ngữ mẫu cũng như mở việc mở rộng cú pháp của HTML.

Nhờ vậy, việc diễn đạt các thành phần ứng dụng của người lập trình sẽ trở nên súc tích và rõ ràng hơn rất nhiều. Bạn có thể tham khảo thêm: 2 tính năng nổi bật là Databinding và Dependency injection của AngularJS có thể loại bỏ phần lớn code mà người lập trình thường xuyên phải viết. Điều này xảy ra trong mọi trình duyệt và giúp nó trở thành đối tác lý tưởng của mọi công nghệ Server.

1. Lịch sử phát triển Angular

AngularJS được phát triển từ năm 2009 bởi Misko Hevery cùng 1 người bạn khác là Adam Abrons. Nó được xem là một dự án riêng cho đến khi Misko Hevery tham gia vào dự án Google Feedback dưới tư cách lập trình viên bán thời gian. Khi đó, Misko Hevery và 2 lập trình viên khác đã viết lên 17.000 các dòng mã khác dành cho dự án Google Feedback trong khoảng 6 tháng.

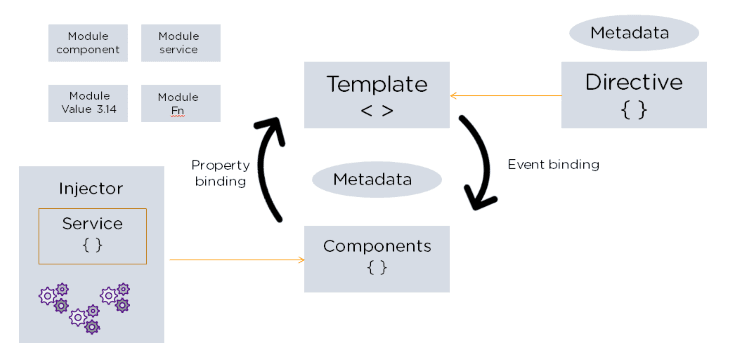


Số lượng mã ngày càng nhiều và việc cần phải sửa lỗi kiểm soát phát sinh. Ông đã mạnh dạn đánh cược với quản lý là có thể viết lại toàn bộ mã nhờ sử dụng dự án Get Angular của mình trong 2 tuần. Mặc dù đã thua cuộc nhưng mã dự án đã giảm nhanh từ 17.000 dòng xuống còn 1.500 dòng. Nhờ sự thành công này mà AngularJS đã được phát triển nhân rộng và tăng tốc nhanh từ đó.

1. Kiến trúc của Angular

Angular là một Framework dựa trên mô hình Model-View-Controller (MVC). Nó cung cấp hướng dẫn rõ ràng về cách ứng dụng nên được cấu trúc và cung cấp luồng dữ liệu hai chiều trong khi cung cấp DOM thực.

Sau đây là các thành phần của một ứng dụng Angular:



**1. Modules**

Một ứng dụng Angular có một module gốc, tên là AppModule, cung cấp cơ chế bootstrap để khởi chạy ứng dụng.

**2. Components**

Mỗi component trong ứng dụng xác định một lớp chứa dữ liệu và logic của ứng dụng. Một thành phần thường xác định một phần của giao diện người dùng (UI).



**3. Templates**

Angular template sẽ kết hợp các Angular markup với HTML để sửa đổi các phần tử HTML trước khi chúng được hiển thị. Có hai loại liên kết dữ liệu:

Liên kết sự kiện: Cho phép ứng dụng của bạn phản hồi thông tin đầu vào của người dùng trong môi trường đích bằng cách cập nhật dữ liệu ứng dụng của bạn.

Ràng buộc thuộc tính: Cho phép người dùng thêm các giá trị được tính toán từ dữ liệu ứng dụng của bạn vào HTML.

**4. Metadata**

Metadata cho Angular biết cách xử lý một lớp. Nó được sử dụng để trang trí lớp để nó có thể cấu hình hành vi mong đợi của một lớp.

**5. Services**

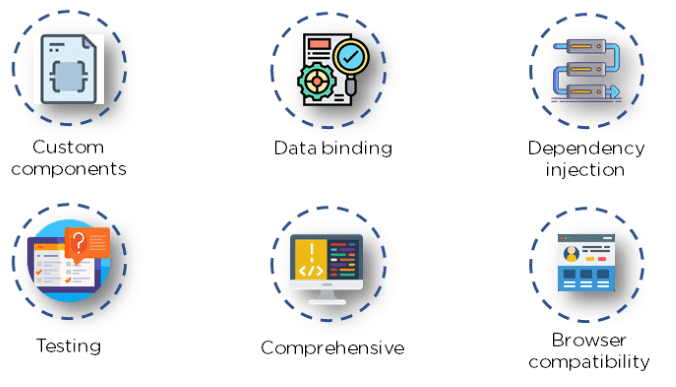
Khi bạn có dữ liệu hoặc logic không được liên kết với view nhưng phải được chia sẻ giữa các thành phần, một lớp service sẽ được tạo. Lớp luôn được liên kết với decorator @Injectible.

**6. Dependency Injection**

Tính năng này cho phép bạn giữ cho các lớp component của mình hoạt động rõ ràng và hiệu quả. Nó không tìm nạp dữ liệu từ máy chủ, xác thực đầu vào của người dùng hoặc đăng nhập trực tiếp vào bảng điều khiển. Thay vào đó, nó ủy nhiệm các nhiệm vụ như vậy cho các service.

1. Ưu điểm và nhược điểm của AngularJS

**4.1. Ưu điểm**

****

* **Custom Components**

**·**  Angular cho phép người dùng xây dựng các component của riêng họ và có thể đóng gói chức năng cùng với logic kết xuất thành các phần có thể tái sử dụng. Nó cũng hoạt động tốt với các component của web.

* **Data Binding**

·       Angular cho phép người dùng dễ dàng di chuyển dữ liệu từ mã JavaScript sang view và phản hồi với các sự kiện của người dùng mà không cần phải viết bất kỳ mã nào theo cách thủ công.

* **Dependency Injection**

 Angular cho phép người dùng viết các service mô-đun và đưa chúng vào bất cứ nơi nào họ cần. Điều này cải thiện khả năng kiểm tra và khả năng tái sử dụng của các service giống nhau.

* **Testing**

·       Bạn có thể dễ dàng Unit test mọi phần trong ứng dụng của mình.

* **Comprehensive**

·       Angular là một full-fledged Framework và có thể cung cấp các giải pháp tiện ích cho giao tiếp máy chủ, định tuyến trong ứng dụng của bạn và hơn thế nữa.

* **Browser Compatibility**

·       Angular cung cấp đa nền tảng và tương thích với nhiều trình duyệt. Một ứng dụng Angular thường có thể chạy trên tất cả các trình duyệt (Ví dụ: Chrome, Firefox) và hệ điều hành, chẳng hạn như Windows, macOS và Linux.

**4.2. Nhược điểm**

**·** Không an toàn: Thông thường, bản chất của AngularJS là một trong những các Front End, mà front end này thường vốn không thể bảo mật bằng Back end. Chính vì vậy, khi sử dụng API thì bạn cần xây dựng cho một hệ thống kiểm tra dữ liệu sao cho việc trả về được tốt nhất.

·        Với một số trình duyệt sở hữu tính năng Disable Javascript nên có nghĩa là website sẽ không hoàn toàn có thể sử dụng được dựa trên những trình duyệt đó nữa.

1. Tại sao chúng ta nên chọn Angular

* **Angular giúp nâng cao năng suất của các lập trình viên.**

Việc phát triển Web đã có bước thay đổi đáng kể trong vài năm qua. Với phiên bản ECMAScript (ES) 2015 – chúng ta quen thuộc với cái tên ES6, với những class hay arrow function. Angular 2+ ứng dụng những tính năng mới này giúp việc code với Angular trở nên rõ ràng và dễ học hơn rất nhiều.

Thêm vào đó, với việc ứng dụng Typescript – một ngôn ngữ – hay là một bản nâng cấp đáng giá của Javascript, Angular kết hợp với Typescript, chúng ta có một công cụ tuyệt vời giúp xử lý các vấn đề hạn chế của JS như kiểm tra kiểu dữ liệu, refactor code an toàn hơn,… từ đó cũng hỗ trợ tốt hơn cho việc Debug cũng như giúp các Dev thực sự hiểu rõ mã nguồn của họ hơn.

* **Cấu trúc phát triển rõ ràng.**

Điều quan trọng của một Frameworks đối với lập trình viên đó là cấu trúc phát triển ứng dụng của nó, và Angular mang đến một kiến trúc rất rõ ràng, dựa trên ba yếu tố chính: class, các dependency được thêm vào và mô hình MVVM (model-view-view/model).

Angular sử dụng class trong ES6 với một loạt các thuộc tính để xây dựng toàn bộ các cấu trúc chủ chốt, giả sử bạn muốn tạo một Angular component – Tạo một class và thêm vào các thuộc tính cần thiết. Hay bạn muốn tạo một Angular module – Hãy tạo một class và thêm vào đó các thuộc tính cần thiết. Về cơ bản sẽ là như vậy, Angular cung cấp một cấu trúc rõ ràng để xây dựng từng tính năng cho ứng dụng của bạn.

Các dependency mạnh mẽ được sử dụng trong ứng dụng khi cần thiết, và khi cần tích hợp bất kì dependency nào, như HTTP hay Router, chúng ta chỉ cần thêm nó vào bên trong constructor của class.

Mô hình MVVM cũng giúp Angular chiếm lợi thế trong xây dựng ứng dụng client-side, thường ta sẽ có 3 điều cần quan tâm chính: đó là giao diện người dùng, mã nguồn điều khiển giao diện và mô hình dữ liệu (data) cho giao diện. Angular với MVVM phân biệt hoàn toàn rõ ràng các yếu tố trên nhờ mô hình MVVM:

Phần giao diện (view) được định nghĩa trong một template bao hàm HTML dành cho một component nhất định. Template có thể là toàn bộ Layout hoặc bất cứ mảnh ghép nào trong Layout đó.

Model được định nghĩa như là các thuộc tính của component class. Có thể hiểu là dữ liệu, dựa vào đó để phần View sử dụng để thực thi.

view/model là class quản lý cả view cũng như model. Là phần code sẽ xử lý việc truy xuất dữ liệu, đồng thời thực thi các tương tác của người dùng trên view.

Với việc ứng dụng các điểm tích cực của các thành phần trên, Angular khiến việc phát triển ứng dụng trở nên dễ dàng và hiệu quả hơn.

* **Extensive binding**

Rất nhiều ứng dụng Web làm việc với dữ liệu (data). App sẽ truy xuất dữ liệu từ Server và hiển thị dữ liệu đó tới người dùng trên view, sử dụng template. Và các tương tác của người dùng sẽ được khiến dữ liệu thay đổi, được view ghi nhận và lưu lại trên server. Data Binding trong Angular giúp bạn thực thi tiến trình trên rất dễ dàng. Đơn thuần từ việc ràng buộc thành phần HTML trong template với các thuộc tính trong class và dữ liệu sẽ tự động xuất hiện trên màn hình. Với các tương tác của người dùng đòi hỏi thay đổi dữ liệu, Angular sử dụng phương pháp two-way binding. Bất kỳ thay đổi dữ liệu đến từ view sẽ tự động cập nhật thuộc tính “model” bên trong class.

Thêm vào đó, Angular cũng hỗ trợ property binding – cho phép chúng ta điều khiển DOM bằng cách ràng buộc thuộc tính HTML với thuộc tính của component class, data sẽ tự động xuất hiện bên trong view. Ví dụ, chúng ta ràng buộc thuộc tính hidden đối với một thẻ img với thuộc tính hideImg bên trong class. Khi thuộc tính hideImg nhận giá trị true, img sẽ tự động hidden và ngược lại khi hideImg nhận giá trị false, thẻ img sẽ tự động hiển thị tới người dùng.

Cuối cùng, Angular hỗ trợ event binding, có nghĩa là chúng ta có thể xử lí bất kì event nào từ phía view, như HTML event. Về cơ bản chúng ta sẽ gắt event với một method bên trong class. Mỗi khi event xuất hiện, method tương ứng sẽ được thực thi.

Extensive binding giúp quá trình hiển thị dữ liệu, điều khiển DOM, thực thi các event một cách trơn tru và dễ dàng.

* **Hỗ trợ đầy đủ tính năng điều hướng (routing)**

Đa số các ứng dụng Web không chỉ có 1 view hay một page duy nhất, mà sẽ cung cấp nhiều view khác nhau tương ứng với với các chức năng chính. Ví dụ như một trang web với các trang giới thiệu, trang nội dung, trang chi tiết, trang đăng nhập, đăng ký,… Chúng ta sẽ cần hiển thị đúng view vào đúng thời điểm. Đó là mục đích của điều hướng (routing). Và Angular cung cấp đầy đủ tính năng cho việc này, chúng ta định nghĩa các đường dẫn (route) cho mỗi page view của ứng dụng. Và chúng ta sẽ kích hoạt route dựa trên tương tác của người dùng (user). Chúng ta có thể truyền thêm dữ liệu vào các route, giúp view hiển thị nội dung một cách dynamic, có thể bảo vệ route để người dùng chỉ có thể truy cập sau khi đã đăng nhập hoặc có quyền truy cập, có thể ngăn chặn việc người dùng ngay lập tức rời một trang khi các thao tác còn dang dở cho đến khi họ thực sự xác nhận việc rời đi hoặc lưu lại tiến trình sử dụng,…Angular đồng thời cũng hỗ trợ child-route cho việc điều hướng bên trong một route. Việc điều hướng giữa các view bên trong ứng dụng Angular thực sự rất linh hoạt và mạnh mẽ.

* **Angular giúp giảm tối đa kích thước và tăng tối đa hiệu suất của ứng dụng.**

Kích thước và hiệu năng có mối liên quan mật thiết khi chúng ta làm việc trên nền tảng Web. Một component nhỏ hơn sẽ giúp nâng cao hiệu suất khởi động – giảm cả thời gian download cũng như thời gian cũng như thời gian compile trên trình duyệt. Giảm kích thước component và giúp tăng hiệu suất là một ưu điểm cũng như mục tiêu mà Angular mong muốn mang đến cho các lập trình viên.

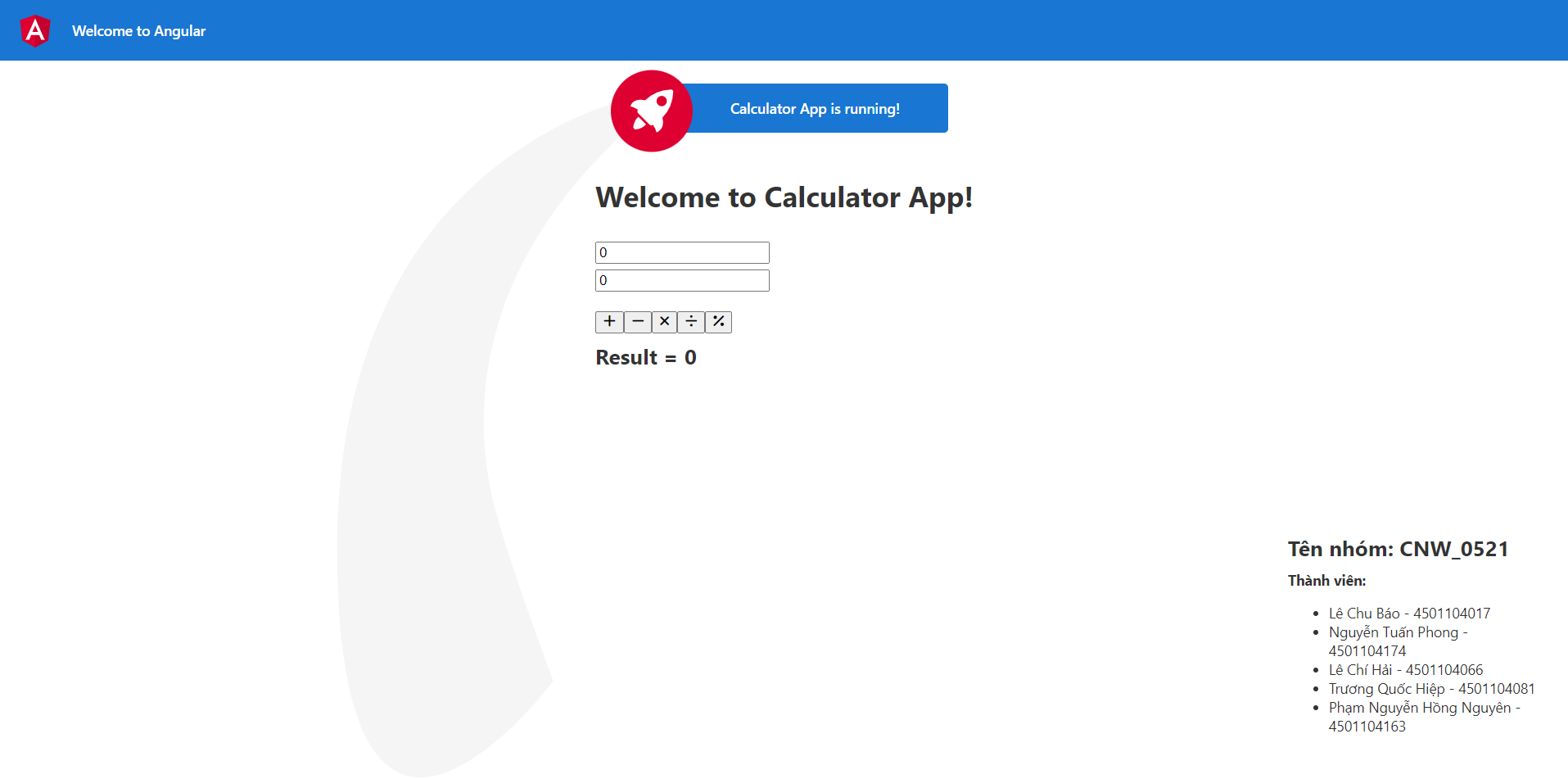
Giảm kích thước ứng dụng có thể thực hiện bằng nhiều cách. Đầu tiên chúng ta có thể giảm tối đa kích thước của từng component tới mức tối thiểu có thể. Tiếp theo các component sẽ được sắp xếp bên trong Angular Module bằng 1 cách đề cho các nhóm logic có liên quan đến nhau sẽ được download cùng với nhau. Và bước thứ ba, lazy loading bên trong các route sẽ chỉ download những module cần thiết cho việc hiển thị nội dung cần thiết tới người dùng, và sẽ không bao giờ download những nội dung không cần thiết.

Chúng ta có một trình biên dịch tên là AOT, trình biên dịch này sẽ chạy một lần trong thời gian build ứng dụng. Trình duyệt sau đó sẽ download phiên bản chưa được biên dịch của ứng dụng và render ứng dụng tới người dùng ngay lập tức mà không cần biên dịch nó lần đầu trong trình duyệt. Thêm nữa là sẽ không cần download trình biên dịch Angular, giúp làm giảm đáng kể kích thước (size) của ứng dụng cần tải về.

* **Document và cộng đồng (community)**

Document cho Angular 2+ – angular.io – rất đầy đủ và chi tiết, bao hàm giới thiệu cơ bản giúp bạn làm quen nhanh chóng với Angular, giới thiệu chi tiết, từ cơ bản đến nâng cao các API của Angular, cũng như có hẳn một Tutorial Basic được xây dựng nên bởi Angular team, cung cấp cho bạn nhanh chóng nắm bắt các thuộc tính cơ bản của Framework.

6. DEMO XÂY DỰNG ỨNG DỤNG CALCULATOR APP



Đây Màn hình chính sau khi chạy code

Gồm chức năng: Tính toán cộng trừ nhân chia đơn giản

Yêu cầu người dùng nhập vào hai giá trị trên hai ô input, sau đó nhấn vào các tác vụ cộng, trừ, nhân, chia hết, chia dư.

Kết quả thu được sẽ hiện thị trên biến Result

Graphical user interface, text

Description automatically generated with medium confidence

*Thực hiện phép tính cộng*

Graphical user interface

Description automatically generated

*Thực hiện phép tính trừ*

Graphical user interface, text

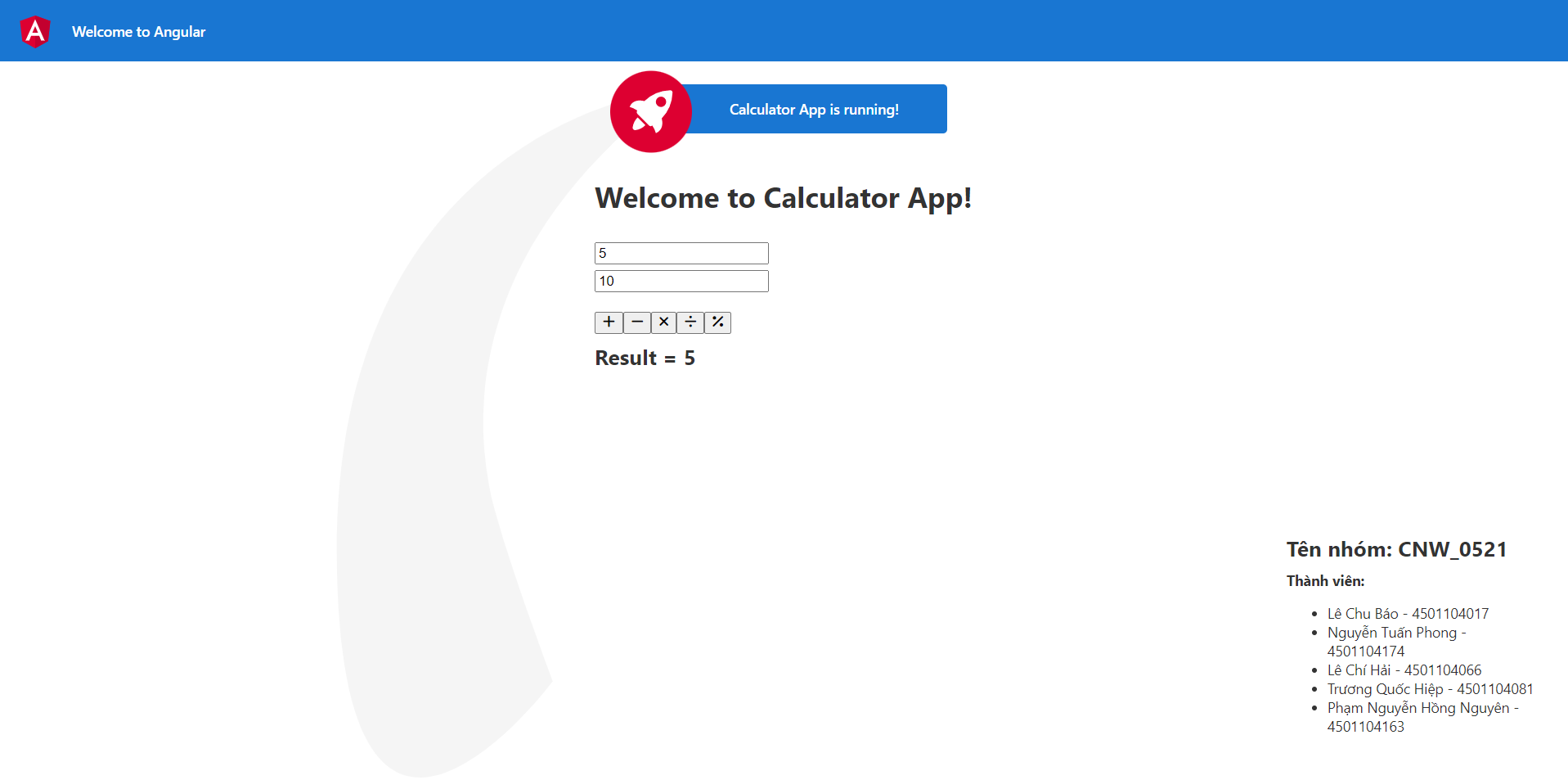
Description automatically generated

*Thực hiện phép tính nhân*

Graphical user interface, text

Description automatically generated

*Thực hiện phép tính chia hết*

**

*Thực hiện phép tính chia dư*

**-Về phần Code:**

Link github: https://github.com/lehai4/angular-demo

END

🙡---🕮---🙣