

**LUẬN VĂN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**XÂY DỰNG ỨNG DỤNG DI ĐỘNG SCANNER TRÊN FRAMEWORK IONIC**

**Sinh viên thực hiện: Nguyễn Phan Hoàng Oanh**

**MSSV: B1401075**

**Khóa: 40**



**LUẬN VĂN TỐT NGHIỆP**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**Đề tài**

**XÂY DỰNG ỨNG DỤNG DI ĐỘNG SCANNER TRÊN FRAMEWORK IONIC**

**Giáo viên hướng dẫn: Sinh viên thực hiện:**

**Ts. Ngô Bá Hùng Nguyễn Phan Hoàng Oanh**

**MSSV: B1401075**

**Khóa: 40**

**NHẬN XÉT CỦA GIẢNG VIÊN**

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

.......................................................................................................................................

**Tên giảng viên nhận xét**

# LỜI CẢM ƠN

Trong suốt thời gian theo học tại Trường Đại Học Cần Thơ em đã được các thầy cô chỉ bảo tận tình, nhận được nhiều sự giúp đỡ từ các anh chị khóa trước cũng như các bạn cùng lớp. Đó không chỉ là những kiến thức chuyên môn mà còn là những chia sẻ về kinh nghiệm, kỹ năng học tập và kỹ năng sống. Tất cả những điều đó sẽ là hành trang quý báu, là nền tảng góp phần giúp em sẵn sàng đón nhận những thử thách trong môi trường mới đầy cạnh tranh, trong công việc và cuộc sống ngoài xã hội sắp tới.

Em xin gửi lời cảm ơn chân thành đến thầy Ngô Bá Hùng, người đã tận tình hướng dẫn, giúp đỡ em trong quá trình thực hiện đề tài.

Em xin cám ơn quý thầy cô Khoa Công Nghệ Thông Tin và Truyền Thông trường Đại Học Cần Thơ đã sẵn sàng giúp đỡ em trong suốt thời gian em theo học tại trường.

Xin gửi lời cám ơn đến bạn bè, các anh, các chị đã tận tình giúp đỡ em trong suốt quá trình học tập và nghiên cứu đề tài.

Trong thời gian qua, mặc dù đã em rất cố gắng để hoàn thành đề tài luận văn, nhưng chắc chắn sẽ không thể tránh khỏi những thiếu sót. Vì vậy em rất mong nhận được sự chỉ bảo, ý kiến từ quý thầy cô và các bạn.

Cần Thơ, ngày 30 tháng 10 năm 2017

Chân thành cảm ơn

**Nguyễn Phan Hoàng Oanh**

# MỤC LỤC

[LỜI CẢM ƠN 1](#_Toc529036215)

[MỤC LỤC 2](#_Toc529036216)

[DANH MỤC BẢNG 5](#_Toc529036217)

[DANH MỤC HÌNH 6](#_Toc529036218)

[DANH MỤC TỪ VIẾT TẮT 7](#_Toc529036219)

[TÓM TẮT 8](#_Toc529036220)

[ABSTRACT 9](#_Toc529036221)

[CHƯƠNG 1: TỔNG QUAN 10](#_Toc529036222)

[**1.1** **Đặt vấn đề** 10](#_Toc529036223)

[**1.2** **Phạm vi đề tài** 10](#_Toc529036224)

[**1.3** **Phương pháp nghiên cứu** 10](#_Toc529036225)

[**1.3.1** **Về lý thuyết** 10](#_Toc529036226)

[**1.3.2** **Về kỹ thuật** 10](#_Toc529036227)

[**1.4** **Bố cục luận văn** 10](#_Toc529036228)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 12](#_Toc529036229)

[**2.1**  **ANGULARJS** 12](#_Toc529036230)

[**2.1.1** **Giới thiệu** 12](#_Toc529036231)

[**2.1.2** **Các tính năng cơ bản** 12](#_Toc529036232)

[**2.1.3** **Ưu điểm** 13](#_Toc529036233)

[**2.2** **TypeScript** 13](#_Toc529036234)

[**2.3** **NODE.JS** 14](#_Toc529036235)

[**2.3.1** **Giới thiệu Node.js** 14](#_Toc529036236)

[**2.3.2** **Giới thiệu NPM** 14](#_Toc529036237)

[**2.4** **CORDOVA** 14](#_Toc529036238)

[**2.5** **FIREBASE** 16](#_Toc529036239)

[**2.5.1** **Giới thiệu Firebase** 16](#_Toc529036240)

[**2.5.2** **Firebase Authentication** 16](#_Toc529036241)

[**2.6** **OPENCV** 16](#_Toc529036242)

[**2.6.1** **Lịch sử** 16](#_Toc529036243)

[**2.6.2** **Giới thiệu** 16](#_Toc529036244)

[**2.6.3** **OpenCV.js** 17](#_Toc529036245)

[**2.7** **Git** 17](#_Toc529036246)

[**2.7.1** **Giới thiệu Git** 17](#_Toc529036247)

[**2.7.2** **Giới thiệu Github** 17](#_Toc529036248)

[**2.7.3** **Sử dụng Git và Github** 18](#_Toc529036249)

[**2.8** **IONIC** 19](#_Toc529036250)

[**2.8.1** **Giới thiệu** 19](#_Toc529036251)

[**2.8.2** **Các thành phần cơ bản** 20](#_Toc529036252)

[**2.8.3** **Cài đặt và sử dụng** 21](#_Toc529036253)

[CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG 26](#_Toc529036254)

[**3.1** **Các yêu cầu và chức năng cần có** 26](#_Toc529036255)

[**3.2** **Mô hình chức năng, đặc tả chức năng** 26](#_Toc529036256)

[**3.3** **Mô hình dữ liệu** 26](#_Toc529036257)

[CHƯƠNG 4: MÔ TẢ GIAO DIỆN 27](#_Toc529036258)

[**4.1** **Giao diện** 27](#_Toc529036259)

[**4.2** **Giao diện** 27](#_Toc529036260)

[**4.3** **Giao diện** 27](#_Toc529036261)

[CHƯƠNG 5: TỔNG KẾT 28](#_Toc529036262)

[THAM KHẢO 29](#_Toc529036263)

# DANH MỤC BẢNG

# DANH MỤC HÌNH

Hình 1. Tải Git cho Windows 18

Hình 2. Menu Item của Git sau khi cài đặt xong 18

Hình 3. Tải Node.js phiên bản LTS 22

Hình 4. Command Prompt của Node.js 22

Hình 5. Cài đặt Ionic và Cordova CLI 22

Hình 6. Khởi tạo ứng dụng Ionic 23

Hình 7. Chạy thử ứng dụng trên trình duyệt 24

# DANH MỤC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| HyperText Markup Language | HTML |
| JavaScript | JS |
| Cascading Style Sheets | CSS |
| Cơ sở dữ liệu | CSDL |
| Node Package Manager | NPM |
| Open Computer Vision | OpenCV |
| User Interface | UI |
| Application Programming Interface | API |
| Global Positioning System | GPS |
| Command Line Interface | CLI |
| Distributed Version Control System | DVCS |
| Software development kit | SDK |
| Java Development Kit | JDK |
|  |  |

# TÓM TẮT

Ứng dụng Scanner là một ứng dụng được xây dựng cho điện thoại di động dựa trên framework Ionic dùng để quét ảnh chứng minh thư, hộ chiếu hoặc tài liệu. Ứng dụng có chức năng chụp ảnh cần quét, sau đó ảnh sẽ được trải qua các bước xử lý như cắt, xoay trái, xoay phải, chỉnh sáng bằng thư viện xử lý ảnh OpenCV để thu được một bức ảnh như khi dùng máy quét và lưu vào bộ nhớ của điện thoại. Từ kết quả trên, người dùng có thể sử dụng ứng dụng để chia sẻ file dưới định dạng ảnh hoặc PDF.

# ABSTRACT

The Scanner application is an application built for mobile phones based on the Ionic framework used to scan ID cards, passports or documents. The application has the function to capture the image to be processed, then the image will be undergo the processing steps such as crop, rotate left, rotate right, adjust the brightness by using the image processing library OpenCV to obtain an image as when using the scanner and save it to phone’s memory. From the above results, users can use the application to share files in image or PDF format.

# CHƯƠNG 1: TỔNG QUAN

## **Đặt vấn đề**

Đồng hành với sự phát triển không ngừng của thế giới, số lượng các loại tài liệu, văn bản cũng tăng lên nhanh chóng với một tốc độ đáng kể. Để tiện cho việc quản lý, chỉnh sửa, lưu trữ cũng như chia sẻ tài liệu một cách dễ dàng nhất không cần tốn nhiều thời gian của người sử dụng, chúng ta sử dụng scanner (máy quét) để chuyển các loại tài liệu từ dạng vật lý sang dữ liệu số. Nhưng scanner vẫn còn hạn chế ở điểm chỉ có thể sử dụng được tại một địa điểm nhất định, nếu đầu tư nhiều scanner để thuận tiện cho việc sử dụng thì chi phí bỏ ra lại quá cao.

Ở thời đại công nghệ phát triển như hiện nay, các thiết bị công nghệ di động nói chung và điển hình là điện thoại di động đang ngày càng được ưa chuộng bởi những lợi ích mà nó mang đến. Vậy tại sao chúng ta không tạo ra một ứng dụng scanner có thể sử dụng cho trên điện thoại di động?

Với một chiếc điện thoại di động được cài đặt ứng dụng scanner, việc quét ảnh ở bất kỳ đâu và vào bất kỳ lúc nào chưa bao giờ là chuyện dễ dàng đến vậy. Không những thế, sự ra đời của ứng dụng scanner trên điện thoại cũng giúp tiết kiệm chi phí hơn khi phải đầu tư vào những máy scanner.

Từ đó, ý tưởng xây dựng một ứng dụng di động scanner đã được hình thành và phát triển thành kết quả của luận văn này.

## **Phạm vi đề tài**

Ứng dụng di động Scanner có khả năng phục vụ cho tất cả mọi người có nhu cầu quét, xử lý, quản lý và lưu trữ tài liệu ở bất cứ đâu và bất kỳ thời điểm nào chỉ cần bạn sở hữu một chiếc điện thoại di động trong tay. Khi điện thoại được kết nối mạng, ứng dụng còn có khả năng chia sẻ ảnh đến những mạng xã hội hoặc bất kỳ ứng dụng nào dùng để liên lạc đang có trong điện thoại.

## **Phương pháp nghiên cứu**

### **1.3.1 Về lý thuyết**

* + - Tìm hiểu phân tích, thiết kế hệ thống thông tin.
    - Phương pháp phân tích thiết kế cơ sỡ dữ liệu.
    - Nắm vững các ngôn ngữ HTML, CSS, JavaScript.
    - Tìm hiểu về các kiến thức TypeScript, Angular, Cordova, NodeJS.
    - Các kiến thức nền tảng cho lập trình web và ứng dụng di động lai.
    - Nắm vững kiến thức về Ionic Framework.

### **1.3.2 Về kỹ thuật**

* + - Cài đặt Node để sử dụng Cordova và Ionic Framework.
    - Sử dụng Ionic Framework để tạo cấu trúc và phát triển ứng dụng.
    - Sử dụng và phát triển ứng dụng di động với Ionic Framework dựa trên ngôn ngữ lập trình web và các Component sẵn có của nó.
    - Tạo các chức năng cần thiết từ các API và Native của Ionic: Camera, File, SQLite, SocialSharing,...

## **Bố cục luận văn**

Chương 1 - Tổng quan: Giới thiệu chung về đề tài, lý do chọn đề tài, phạm vi của đề tài và phương pháp, kĩ thuật cần tìm hiểu để xây dựng được ứng dụng.

Chương 2 - Cơ sở lý thuyết: Giới thiệu các thuật ngữ, kỹ thuật, Framework, API và Native được sử dụng trong ứng dụng.

Chương 3 - Phân tích thiết kế hệ thống: Kết quả của quá trình phân tích, thiết kế hệ thống bao gồm: mô hình use case, đặc tả chức năng, mô hình CDM, PDM và mô tả các bảng dữ liệu.

Chương 4 – Mô tả giao diện: Giới thiệu và mô tả các giao diện chính của ứng dụng khi hoạt động.

Chương 5 - Tổng kết: Kết quả đạt được, các kinh nghiệm, ưu, nhược điểm và hướng phát triển của đề tài.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

## **2.1 ANGULARJS**

AngularJS là một một dự án mã nguồn mở được phát triển đầu tiên bởi Miško Hevery, một nhân viên của Google. Hevery bắt đầu nghiên cứu và phát triển dự án vào năm 2009 và phiên bản 1.0 được cho ra mắt vào năm 2012. Do sự hữu ích của dự án này nên Google quyết định trở thành công ty chính thức đứng đằng sau hỗ trợ cho sự phát triển của AngularJS. Phiên bản mới nhất của AngularJS tính đến thời điểm này là 7.0.3.

AngularJS là một framework được viết bằng JavaScript có cấu trúc cho các ứng dụng web động. Nó cho phép sử dụng HTML như là ngôn ngữ mẫu và cho phép mở rộng cú pháp của HTML để diễn đạt các thành phần ứng dụng một cách rõ ràng và súc tích. Hai tính năng cốt lõi: Data binding và Dependency injection của AngularJS loại bỏ phần lớn code thường phải viết. Nó xuất hiện trong tất cả các trình duyệt, làm cho nó trở thành đối tác lý tưởng của bất kỳ công nghệ Server nào.

Ngoài ra, AngularJs còn có các thành phần bổ sung cho Cordova, framework thường được sử dụng để viết các ứng dụng di động. Nó nhằm mục đích để đơn giản hóa cả việc phát triển và [thử nghiệm](https://vi.wikipedia.org/wiki/Ki%E1%BB%83m_th%E1%BB%AD_ph%E1%BA%A7n_m%E1%BB%81m) của các ứng dụng bằng cách cung cấp một framework với kiến trúc cho client side đó là MVC (Model View Controller) và MVVM (Model – View – ViewModel).

Ưu điểm của AngularJS:

* + - Cho phép tạo ra các ứng dụng một cách đơn giản, code sạch.
    - Sử dụng data binding giống .NET với tính năng liên kết với HTML nên giúp người dùng cảm thấy dễ sử dụng.
    - Có thể chạy trên hầu hết các trình duyệt điện thoại thông minh.
    - Cung cấp khả năng tái sử dụng các component.
    - Code dễ dàng khi unit test (kiểm tra từng đơn vị nhỏ như thuộc tính, sự kiện, thủ tục và hàm).

## **2.2 TypeScript**

Typescript là một dự án mã nguồn mở được Microsoft phát triển,  nó có thể được coi là một phiên bản nâng cao của JavaScript bởi việc bổ sung tùy chọn kiểu tĩnh và lớp hướng đối tượng mà điều này không có ở JavaScript. TypeScript có thể sử dụng để phát triển các ứng dụng chạy ở client-side (Angular2) và server-side (NodeJS). TypeScript là một ngôn ngữ giúp cung cấp quy mô lớn hơn so với JavaScript.

Typescript đang được sử dụng ở các framework front-end phổ biến như Angular 2, Ionic... cũng như nền tảng cho back-end như Node.js bởi những ưu điểm của nó.

Ưu điểm của Typescript:

* + - Dễ dàng hơn trong việc phát triển các dự án lớn, được hỗ trợ bởi các Framework JavaScript lớn.
    - Hỗ trợ hướng đối tượng mạnh: Hầu hết các cú pháp hướng đối tượng đều được hỗ trợ bởi Typescript như kế thừa, đóng gói, constructor, abstract, interface, implement, override...v.v
    - Cách tổ chức code rõ ràng hơn bởi được hỗ trợ các kỹ thuật mới nhất và hỗ trợ lập trình hướng đối tượng: Hỗ trợ cơ chế giúp kiến trúc hệ thống code hướng module, hỗ trợ namespace, giúp xây dựng các hệ thống lớn nơi mà nhiều lập trình viên có thể làm việc cùng nhau một cách dễ dàng hơn.
    - Hỗ trợ các tính năng mới nhất của JavaScript.
    - Một lợi thế của Typescript nữa là mã nguồn mở vì vậy nó miễn phí và có cộng đồng hỗ trợ rất lớn.

## **2.3 NODE.JS**

### **2.3.1 Giới thiệu Node.js**

Node.js được tạo bởi Ryan Dahl từ năm 2009 và phát triển dưới sự bảo trợ của [Joyent](https://vi.wikipedia.org/w/index.php?title=Joyent&action=edit&redlink=1). Mục tiêu ban đầu của Dahl là làm cho trang web có [khả năng push](https://vi.wikipedia.org/w/index.php?title=C%C3%B4ng_ngh%E1%BB%87_push&action=edit&redlink=1) như trong một số ứng dụng web như [Gmail](https://vi.wikipedia.org/wiki/Gmail). Sau khi thử với vài ngôn ngữ Dahl chọn JavaScript vì một API Nhập/Xuất không đầy đủ. Điều này cho phép anh có thể định nghĩa một quy ước Nhập/Xuất điểu khiển theo sự kiện, non-blocking.

Node.js là một [hệ thống phần mềm](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng_ph%E1%BA%A7n_m%E1%BB%81m) được thiết kế để viết các ứng dụng internet có khả năng mở rộng, đặc biệt là cho phép lập trình JavaScript phía server. Trước đây JavaScript chỉ được coi là một ngôn ngữ đơn giản tạo hiệu ứng hoạt hình, tương tác phía trình duyệt. Cho đến khi Google Chrome sử dụng V8 engine - một bộ dịch – để thực thi mã JavaScript nhanh nhất thế giới tính đến hiện nay. V8 được viết bằng C/C++, mã nguồn mở, cho phép bên thứ 3 toàn quyền sử dụng, triển khai trên các hệ điều hành khác nhau: Windows, Linux, Unix, MacOSX. Node.js được xây dựng trên V8 và thư viện LibUV cho phép lệnh thực thi hiện tại không ngăn các lệnh kế tiếp chạy luôn: cơ chế Non-Blocking I/O.

### **2.3.2 Giới thiệu NPM**

NPM (Node Package Manager) là một công cụ (chương trình) quản lý các thư viện lập trình JavaScript cho Node.js, được tích hợp sẵn trong Node.js, vì vậy khi cài đặt xong Node.js chúng ta sẽ có thể sử dụng được cả NPM.

## **2.4 CORDOVA**

Apache Cordova (trước kia được gọi là PhoneGap) là một framework mã nguồn mở phát triển ứng dụng di động nổi tiếng được tạo ra bởi Nitobi. Adobe Systems đã mua Nitobi vào năm 2011, đổi tên nó thành PhoneGap, và sau đó đã phát hành một phiên bản mã nguồn mở được gọi là Apache Cordova. Cordova là một nền tảng để xây dựng những ứng dụng di động lai (hybrid mobile applications) sử dụng HTML, CSS và JavaScript. Những ứng dụng này có thể chạy trên nhiều nền tảng khác nhau như Android, iOS, Windows Phone…. Ứng dụng lai đem lại nhiều lợi thế như khả năng hiển thị nội dung trên tất cả các thiết bị di động, cũng như tận dụng tối đa các tính năng khác của thiết bị di động như GPS, camera, danh sách liên lạc,… Bên cạnh đó, thời gian và chi phí dùng để tạo nên một ứng dụng lai cũng thấp hơn so với các ứng dụng di động thông thường.

Cordova bao gồm một tập hợp các API thiết bị cho phép người lập trình di động truy cập, sử dụng các chức năng native của thiết bị như là camera hay cảm biến gia tốc bằng JavaScript. Kết hợp với một bộ khung phát triển giao diện như jQuery Mobile, Dojo Mobile hoặc Ionic, nó cho phép có thể phát triển ứng dụng di động chỉ dựa trên HTML, CSS và JavaScript. Sử dụng Cordova sẽ nhanh hơn rất nhiều so với phát triển các ứng dụng thuần Android hay iOS. Chỉ cần sử dụng JavaScript khi làm việc với Cordova nên ta không cần thiết phải biết ngôn ngữ lập trình của mỗi loại hệ điều hành.

Khi sử dụng Cordova API, một ứng dụng có thể được xây dựng mà không phải sử dụng bất kỳ một đoạn code native nào. Thay vào đó, công nghệ web sẽ được sử dụng, và chúng sẽ được tổ chức trên chính ứng dụng đấy chứ không cần thông qua bất kỳ server nào. Và bởi vì những API JavaScript này là thống nhất trên tất cả các nền tảng thiết bị và được xây dựng trên chuẩn web nên những ứng dụng được viết trên nền tảng này có thể được sử dụng trên các nền tảng khác mà không cần có bất cứ sự thay đổi nào.

Các tính năng của Cordova:

* + - Command Line Interface (Cordova CLI): đây là một công cụ dùng để khởi tạo một dự án , xây dựng ứng dụng trên nhiều nên tảng khác nhau cũng như là thêm nhiều plugin có ích để giúp việc phát triển trở nên dễ dàng hơn.
    - Cordova Core Components: Cordova đưa ra một tập các thành phần mà mọi ứng dụng di động cần đến.
    - Cordova Plugins: Cordova đưa ra các API để sử dụng các chức năng của thiết bị di động như cảm biến, camera, GPS…

## **2.5 FRAMEWORK IONIC**

### **2.5.1 Giới thiệu**

Framework Ionic là một SDK (bộ công cụ phát triển phần mềm) mã nguồn mở cho phép các nhà phát triển xây dựng các ứng dụng di động lai chất lượng cao, hiệu quả bằng các công nghệ web quen thuộc (HTML, CSS và JavaScript). Sau khi phát hành phiên bản alpha của Ionic vào tháng 11 năm 2013, chúng tôi đã phát hành bản beta 1.0 vào tháng 3 năm 2014 và trận chung kết 1.0 vào tháng 5 năm 2015. Phiên bản mới nhất của Ionic tính đến thời điểm hiện tại là 4.3.1.

Có thể coi Ionic như là một bộ khung front-end giúp người dùng kiểm soát hình ảnh và trải nghiệm trên ứng dụng. Giống như “Bootstrap for Native”, nhưng với sự hỗ trợ của một lượng lớn các thành phần di động, hiệu ứng chuyển động mượt mà và thiết kế đẹp mắt. Không như các framework khác, Ionic mang đến cho chúng ta những thành phần giao diện và cách bố trí mang phong cách rất giống với phong cách thiết kế mặc định trên thiết bị di động.

Vì Ionic là một framework HTML5 nên nó cần bộ đóng gói (wrapper) như Cordova hay PhoneGap để có thể chạy được như một ứng dụng di động. Cordova là bộ đóng gói mặc định trong Ionic framework.

Ionic sử dụng AngularJS để tạo ra một framework tốt nhất cho việc xây dựng các ứng dụng di động đa nền tảng, nó cung cấp một bộ giao diện người dùng (UI) mã nguồn mở miễn phí đi cùng với các tính năng của AngularJS. Việc xây dựng ứng dụng dựa trên AngularJS đòi hỏi mã nguồn phải có khả năng mở rộng cao để bổ sung các tính năng mới. Tuy nhiên với Ionic, người ta có thể tái sử dụng các chức năng trong ứng dụng trên các nền tảng khác nhau đồng thời vẫn có thể tùy chỉnh giao diện người dùng cho mỗi nền tảng riêng biệt. Các thành phần trong Ionic như danh sách, slide,.. chính là các directive (các thuộc tính của thẻ HTML dùng trong Angular) của AngularJS. Đó là lí do khiến cho Ionic và AngularJS kết hợp rất tốt với nhau.

Ưu điểm:

* + - Các ứng dụng lai có nhiều ưu điểm như có thể hiển thị cho nhiều nền tảng.
    - Tận dụng các tính năng sẵn có của thiết bị di động như GPS, camera....
    - Thời gian và chi phí dùng để phát triển ứng dụng thấp hơn nhiều so với native.
    - Chỉ cần biết JavaScript mà không cần phải biết từng ngôn ngữ lập trình.
    - Có thể sử dụng các kỹ năng từ lập trình web.
    - Đa nền tảng.
    - Dễ dàng thiết kế giao diện cho các thiết bị có kích cỡ khác nhau.
    - Việc sử dụng AngularJS làm core giúp phần xử lý UI linh động hơn so với JavaScript hay thư viện jQuery, đồn thời cũng mang lại lợi thế lớn so với các framework cho ứng dụng lai khác.
    - Ionic cung cấp đầy đủ các thành phần trong giao diện người dùng như Pull-to-Refresh, Infinite-loader, tabs, ..

Nhược điểm:

* + - Tốc độ chậm với một số tính năng chuyển trang trên di động.
    - Các plugin có thể không tương thích với một số thiết bị và nền tảng. Một số API cũng chưa được hỗ trợ để giao tiếp với thiết bị.
    - Vẫn còn trong giai đoạn phát triển.
    - Hiệu năng vẫn chưa cao và ổn định.
    - Cộng đồng phát triển ứng dụng vẫn còn chưa đông.

### **2.5.2 Các thành phần cơ bản**

* + - UI components (Thành phần giao diện người dùng): Ứng dụng Ionic được tạo thành từ các khối xây dựng cao cấp được gọi là các Component. Component cho phép bạn xây dựng giao diện cho ứng dụng một cách nhanh chóng. Component gồm có Action Sheets, Alert, Badges, Button,...
    - API (Giao diện lập trình ứng dụng): Gồm hai loại Component và Service
      * API Component: bao gồm các lớp như Checkbox, Toggle hoặc Item. Component chỉ cho bạn thấy cách sử dụng chúng, ngoài ra còn liệt kê các bộ chọn, các thuộc tính và sự kiện có sẵn của chúng.
      * API Service: bao gồm các lớp như MenuController, Config hoặc Platform. Đây là những dịch vụ được cung cấp bởi Ionic có thể được tiêm vào các lớp của bạn.
    - Native: Là một trình bao bọc TypeScript cho các plugin Cordova/PhoneGap giúp bổ sung thêm bất kỳ chức năng gốc nào bạn cần cho ứng dụng di động Ionic của mình một cách dễ dàng như tương tác với camera, GPS,...
    - Storage: là một cách dễ dàng để lưu trữ các cặp key/value và các đối tượng JSON. Storage sử dụng nhiều công cụ lưu trữ bên dưới điện thoại, chọn công cụ lưu trữ tốt nhất có sẵn tùy thuộc vào nền tảng. Khi chạy trong ngữ cảnh ứng dụng Native, Storage sẽ ưu tiên sử dụng SQLite, vì nó là một trong những CSDL dựa trên tệp ổn định, được sử dụng rộng rãi nhất và tránh được một số cạm bẫy của những thứ như localstorage và IndexedDB, chẳng hạn như hệ điều hành quyết định xóa dữ liệu đó trong các tình huống không gian đĩa thấp.
    - CLI (Giao diện dòng lệnh Ionic): là công cụ giúp phát triển ứng dụng Ionic.

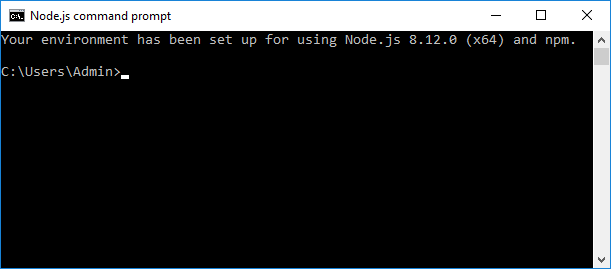
### **2.5.3 Cài đặt và sử dụng**

Ứng dụng Ionic được xây dựng và phát triển chủ yếu thông qua giao diện dòng lệnh Ionic (“CLI”) và sử dụng Cordova để xây dựng/triển khai dưới dạng ứng dụng Native. Điều này có nghĩa là chúng ta cần phải cài đặt một vài tiện ích để phát triển.

**2.5.3.1 Cài đặt Node.js và NPM**

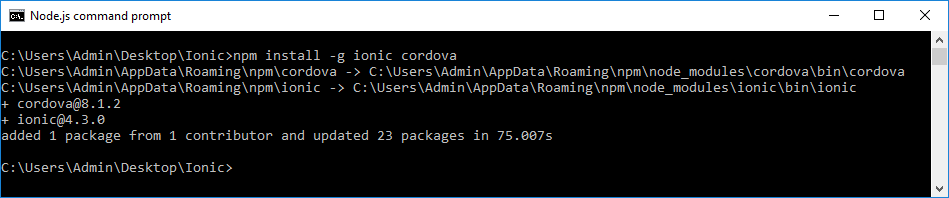
Hầu hết các công cụ trong CLI đều dựa trên Node và được quản lý thông qua NPM. Cách nhanh nhất để cài đặt Node và NPM là thông qua trình cài đặt NodeJS. Tải Node.js phiên bản LTS mới nhất tại trang chủ về máy và cài đặt.

Sau khi cài đặt hoàn tất, chúng ta sẽ có Command Prompt của Node.js



**2.5.3.2 Ionic CLI và Cordova**

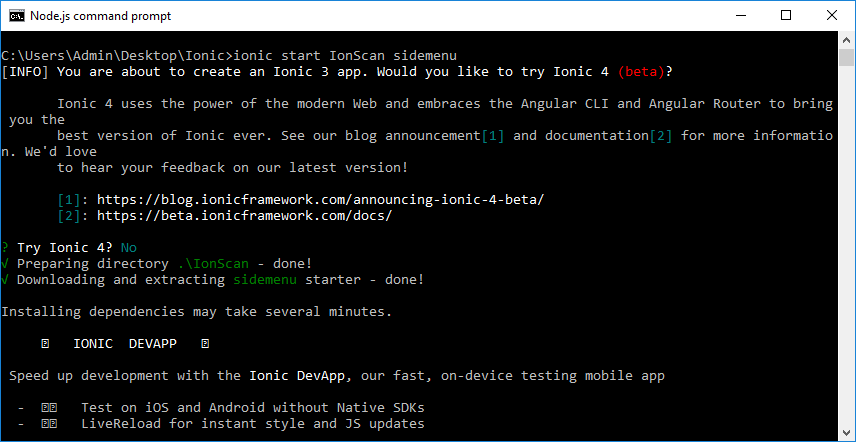
Với Command Prompt đang có, tiếp tục cài đặt Ionic và Cordova CLI bằng câu lệnh sau: $ npm install -g ionic cordova



**2.5.3.3 Khởi tạo một ứng dụng**

Khi đã xong, bắt đầu tạo ứng dụng Ionic, với IonScan là tên ứng dụng và sidemenu là template. Ionic có 3 template để lựa chọn: blank, sidemenu và tabs

$ ionic start IonScan sidemenu



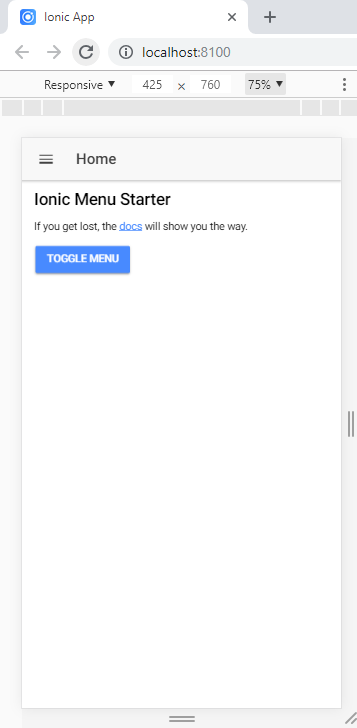
Trong lúc cài đặt, Ionic sẽ cho phép liên kết dự án với Ionic Pro SDK hoặc Git để lưu trữ dự án. Nếu bỏ qua bước này, chúng ta vẫn có thể liên kết trong quá trình phát triển ứng dụng.

**2.5.3.4 Chạy ứng dụng**

Để chạy ứng dụng, di chuyển vào thư mục đã được tạo bằng câu lệnh cd và sau đó chạy lệnh ion serve để kiểm tra ứng dụng ngay trên trình duyệt:

$ cd IonScan

$ ionic serve



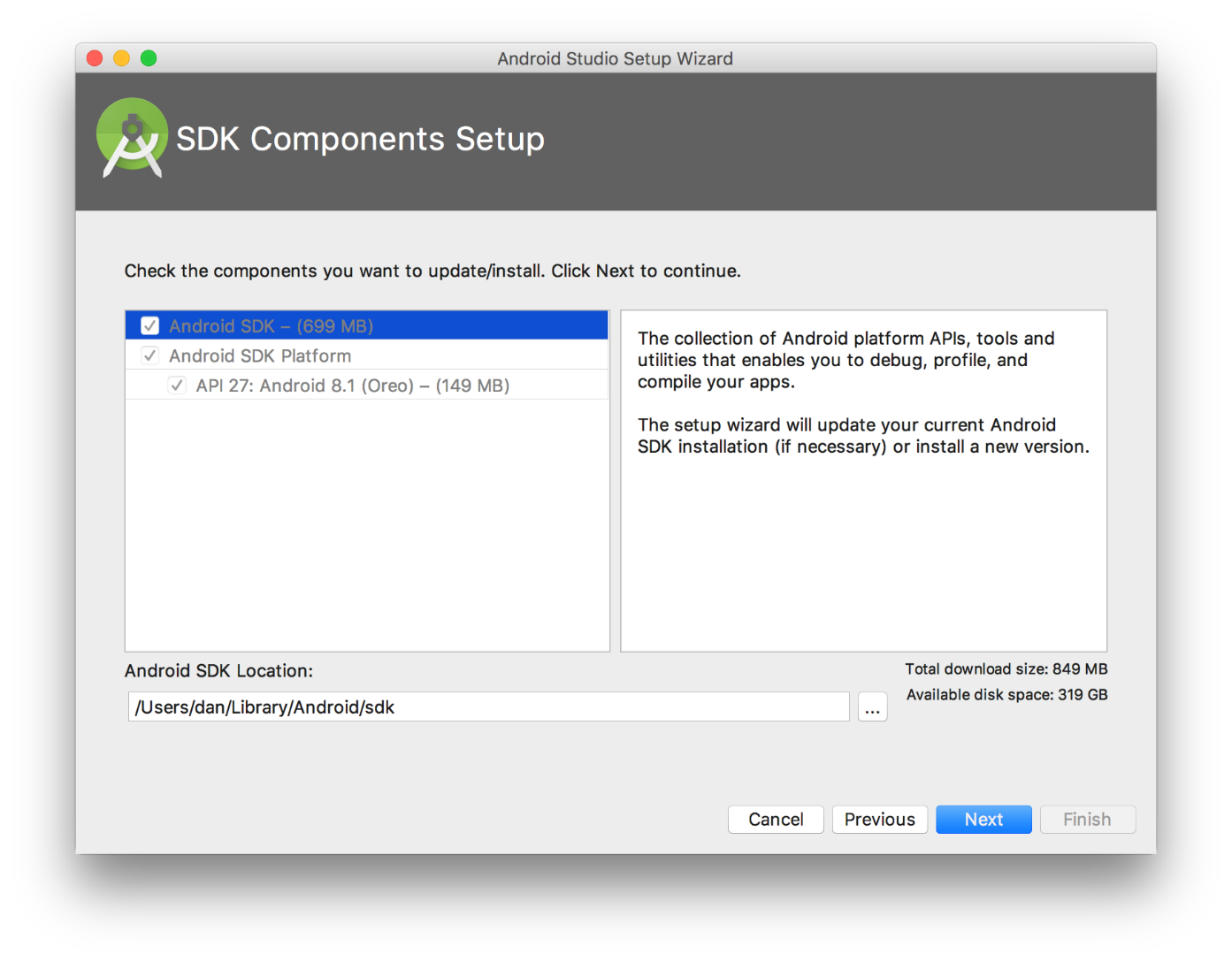
**2.5.3.5 Cài đặt Java**

Các ứng dụng Android Native được biên dịch bằng ngôn ngữ lập trình Java. Để tạo được ứng dụng Android Native chúng ta cần tải JDK8 (bộ công cụ phát triển Java) từ trang Java xuống, cài đặt và cấu hình.

**2.5.3.6 Cài đặt Android Studio**

Android Studio là IDE để tạo ứng dụng Android Native. Nó bao gồm Android SDK, cần được cấu hình để sử dụng trong dòng lệnh. Android Studio cũng được sử dụng để tạo thiết bị Android ảo, được yêu cầu cho trình giả lập Android. Ứng dụng Ionic cũng có thể được chạy trên một thiết bị.

Tải Android Studio về máy và cài đặt. Sau đó mở Android Studio lên, IDE sẽ phát hiện ra cần phải cài Android SDK. Trong màn hình SDK Components Setup, hoàn tất việc cài đặt SDK. Hãy lưu ý đến vị trí của Android SDK. Sau khi Android SDK được cài đặt xong, có thể đóng Android Studio lại.



**2.5.3.7 Tạo file apk**

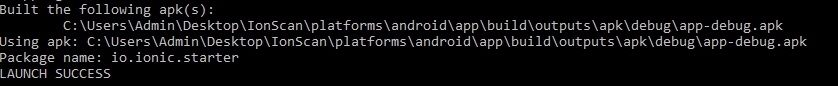
Kích hoạt nền tảng mà chúng ta muốn xây dựng (Android/iOS):

$ ionic cordova platform add android

Xây dựng dự án Ionic cho nền tảng vừa tạo:

$ ionic cordova build android

Sau khi xây dựng thành công, Ionic sẽ trả về đường dẫn chứa file apk



**2.5.3.8 Cài đặt Native plugin hoặc NPM**

Để sử dụng các Native plugin của Ionic hoặc NPM, chúng ta cần cài đặt Plugin như sau (ví dụ như Plugin Camera):

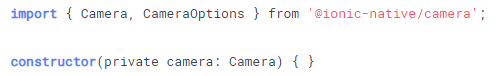
$ ionic cordova plugin add cordova-plugin-camera

$ npm install --save @ionic-native/camera

Tiếp theo, thêm nó vào NgModule trong app/app.module.ts:



Sau đó khai báo Plugin trong file TypeScript mà chúng ta cần sử dụng:



## **2.5 FIREBASE**

### **2.5.1 Giới thiệu Firebase**

Firebase là một dịch vụ CSDL thời gian thực hoạt động trên nền tảng đám mây được cung cấp bởi Google nhằm giúp các lập trình phát triển nhanh các ứng dụng bằng cách đơn giản hóa các thao tác với CSDL.

Các chức năng chính của Firebase:

* + - Realtime Database – Cơ sở dữ liệu thời gian thực: Firebase lưu trữ CSDL dưới dạng JSON và thực hiện đồng bộ CSDL tới tất cả các client theo thời gian thực. Điều đó có nghĩa là khi tất cả các client cùng sử dụng chung một CSDL đến từ Firebase, ứng dụng có thể tự động cập nhật mỗi khi dữ liệu trong CSDL được thêm mới hoặc sửa đổi. Ngoài ra Firebase còn cho phép phân quyền một cách đơn giản bằng cú pháp tương tự như javascript.
    - Firebase Hosting: Người dùng có thể triển khai một ứng dụng nền web chỉ với vài giây với hệ thống Firebase, và các dữ liệu sẽ được lưu trữ đám mây đồng thời được bảo mật thông qua giao thức truy cập SSL. Các ứng dụng sẽ được cấp một tên miền dạng \*.firebaseapp.com hoặc người dùng có thể trả tiền để sử dụng tên miền của riêng mình.
    - Firebase Authentication – Hệ thống xác thực người dùng.

Ưu điểm của Firebase:

* + - Triển khai ứng dụng cực nhanh: Tiết kiệm thời gian quản lý và đồng bộ CSDL và xây dựng ứng dụng đa nền tảng. Không chỉ vậy, Firebase còn giúp đơn giản hóa quá trình đăng kí và đăng nhập vào ứng dụng.
    - Bảo mật: Firebase hoạt động dựa trên nền tảng cloud và thực hiện kết nối thông qua giao thức bảo mật SSL, người dùng có thể yên tâm về việc bảo mật của dữ liệu cũng như đường truyền giữa client và server. Không chỉ vậy, việc cho phép phân quyền người dùng CSDL bằng cú pháp javascipt cũng nâng cao độ bảo mật cho ứng dụng hơn, bởi vì chỉ những người dùng được cho phép mới có thể có quyền chỉnh sửa cơ sở dữ liệu.
    - Tính linh hoạt và khả năng mở rộng: Sử dụng Firebase sẽ giúp người dễ dàng nâng cấp hoặc mở rộng dịch vụ. Ngoài ra firebase còn cho phép người dùng tự xây dựng server của riêng mình để thuận tiện cho việc quản lý hơn. Việc Firebase sử dụng NoSQL, giúp cho CSDL của người dùng sẽ không bị bó buộc trong các bảng và các trường mà có thể tùy ý xây dựng CSDL theo cấu trúc riêng.
    - Sự ổn định: Firebase hoạt động dựa trên nền tảng đám mây đến từ Google vì vậy gần như không phải lo lắng về việc sập server, tấn công mạng như DDOS, tốc độ kết nối lúc nhanh lúc chậm,…, bởi vì đơn giản là Firebase hoạt động trên hệ thống server của Google. Hơn nữa nhờ hoạt động trên nền tảng đám mây nên việc nâng cấp, bảo trì server cũng diễn ra rất đơn giản  mà không cần phải dừng server để nâng cấp như truyền thống.
    - Giá thành: Google Firebase có rất nhiều gói dịch vụ với các mức dung lượng lưu trữ cũng như băng thông khác nhau với mức giá dao động từ miễn phí đến $1500 đủ để đáp ứng được nhu cầu của tất cả các đối tượng. Điều này giúp bạn tới ưu hóa được vốn đầu tư và vận hành tùy theo số lượng người sử dụng. Ngoài ra người dùng còn không mất chi phí để bảo trì, nâng cấp, khắc phục các sự cố bởi vì những điều này đã có Firebase lo.

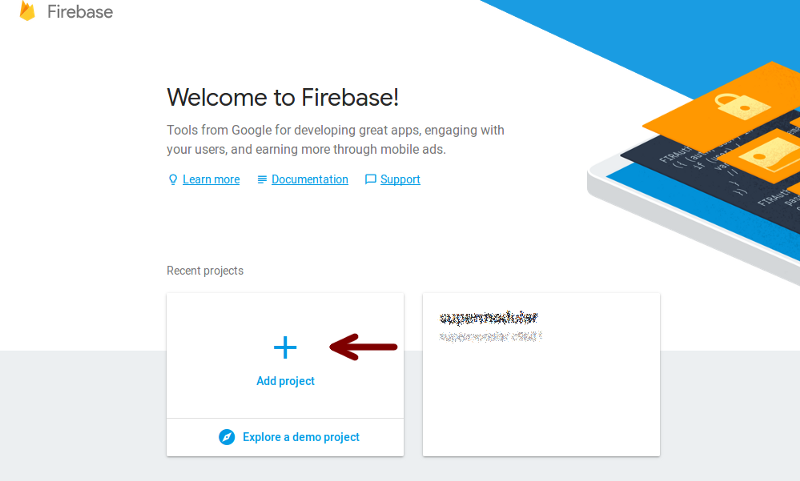
### **2.5.2 Giới thiệu Firebase Authentication**

Hầu hết các ứng dụng cần biết danh tính của người dùng. Việc biết danh tính của người dùng cho phép ứng dụng lưu dữ liệu người dùng trên đám mây một cách an toàn và cung cấp trải nghiệm được cá nhân hóa giống nhau trên tất cả các thiết bị của người dùng.

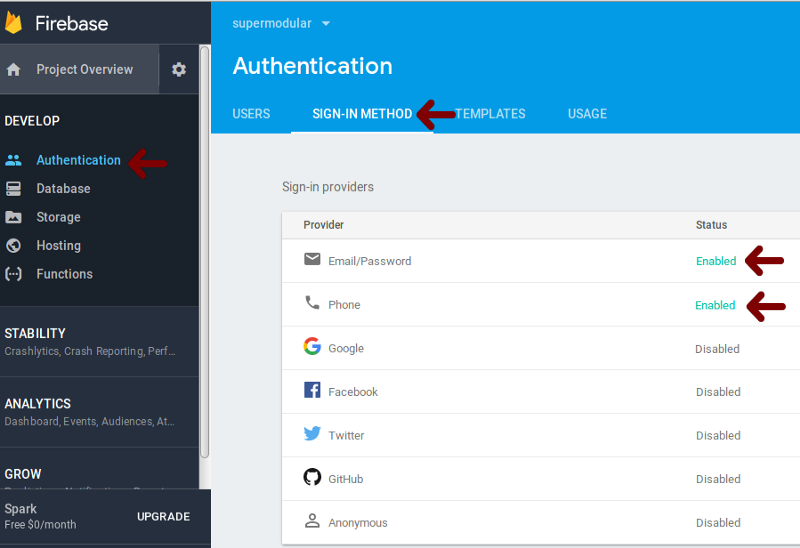
Firebase Authentication cung cấp dịch vụ backend, SDK dễ sử dụng và thư viện giao diện người dùng được tạo sẵn để xác thực người dùng cho ứng dụng của bạn. Nó hỗ trợ xác thực bằng mật khẩu, số điện thoại, tích hợp các công nghệ xác thực của các nhà cung cấp nhận dạng liên kết phổ biến như Google, Facebook và Twitter, v.v.

### **2.5.3 Sử dụng Firebase Authentication trong dự án Ionic**

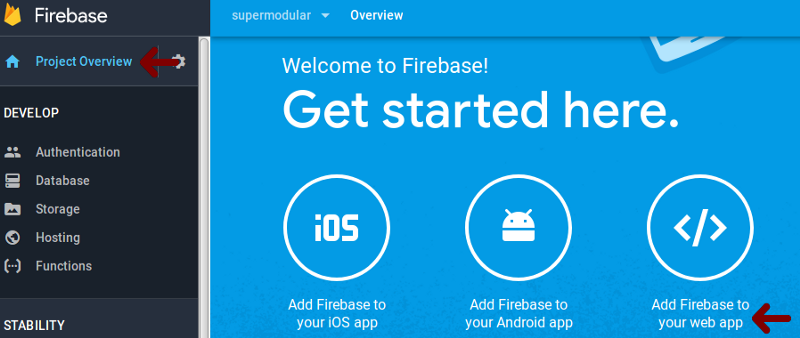
Truy cập <https://console.firebase.google.com/> và tạo dự án mới.

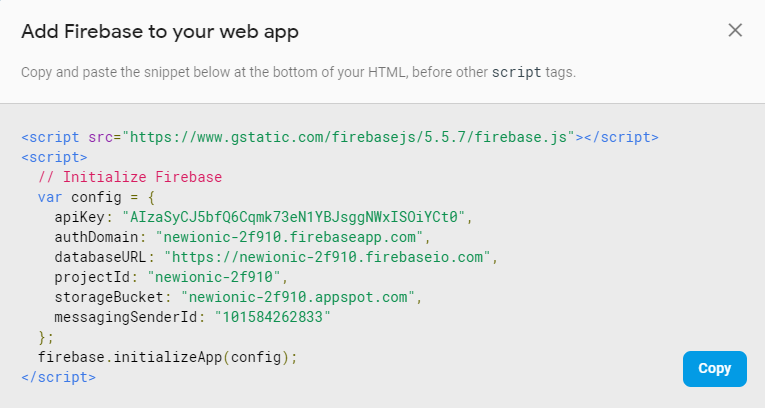


Sau đó, bật phương thức xác thực cần dùng lên, trong dự án này là Email và Phone.

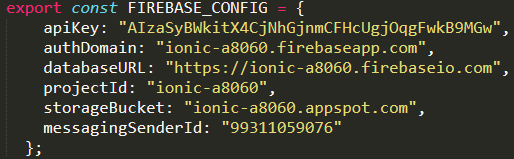


Truy cập vào Project Overview và chọn Add Firebase to your web app, chúng ta sẽ thấy được đoạn code bên dưới:





Tạo file app.firebase.config.ts trong src/app và copy đoạn code ở bên trên vào:



Tiếp theo chúng ta cài đặt AngularFire2 thông qua CLI

$ npm install firebase angularfire2 --save

AngularFire2 là thư viện AngularJS được hỗ trợ chính thức cho Firebase. Nó cho phép chúng ta liên kết các tham chiếu Firebase với các mô hình Angular để chúng được đồng bộ với cơ sở dữ liệu và với tất cả các ứng dụng khách khác hiện đang sử dụng ứng dụng.

Sau đó chúng ta khai báo chúng trong file app.module.ts





Tạo một folder service chứa file auth.service.ts để sử dụng các lệnh đăng nhập, đăng ký của AngularFireAuth

## **2.6 OPENCV**

### **2.6.1 Lịch sử**

* + - Năm 1999, OpenCV được tạo ra ở Intel vào bởi Gary Bradski.
    - Năm 2000, bản phát hành đầu tiên ra mắt. Vadim Pisarevsky gia nhập Gary Bradski để quản lý đội OpenCV phần mềm của Intel.
    - Năm 2005, OpenCV đã được sử dụng trên Stanley; chiếc xe giành được giải thưởng DARPA Grand Challenge năm 2005.
    - Sau đó, sự phát triển tích cực của nó được tiếp tục dưới sự hỗ trợ của Willow Garage, với Gary Bradski và Vadim Pisarevsky dẫn đầu dự án.
    - OpenCV hiện hỗ trợ vô số các thuật toán liên quan đến Computer Vision và Machine Learning và đang mở rộng từng ngày.

### **2.6.2 Giới thiệu**

OpenCV (Open Computer Vision) là một thư viện được cấp phép BSD mã nguồn mở hàng đầu với hàng trăm thuật toán đã được tối ưu cho xử lý về thị giác máy tính, máy học và xử lý ảnh. OpenCV đươc viết bằng C/C++, vì vậy có tốc độ tính toán rất nhanh, có thể sử dụng với các ứng dụng liên quan đến thời gian thực. OpenCV hỗ trợ nhiều ngôn ngữ lập trình khác nhau như C ++, Python và Java. OpenCV có sẵn trên các nền tảng khác nhau bao gồm Windows, Linux, MacOS, Android và iOS. Giao diện cho các hoạt động GPU tốc độ cao dựa trên CUDA và OpenCL cũng đang được phát triển tích cực. OpenCV.js mang OpenCV đến nền tảng web mở và tạo cơ hội cho lập trình viên JavaScript sử dụng.

### **2.6.3 OpenCV.js**

Web là nền tảng điện toán mở phổ biến nhất. Với các tiêu chuẩn HTML5 được triển khai trong mọi trình duyệt, các ứng dụng web có thể hiển thị video trực tuyến bằng thẻ video HTML5, quay video webcam qua API WebRTC và truy cập từng pixel của khung video thông qua API canvas. Với sự phong phú của nội dung đa phương tiện có sẵn, các nhà phát triển web đang cần một loạt các thuật toán xử lý hình ảnh và hình ảnh trong JavaScript để xây dựng các ứng dụng sáng tạo.

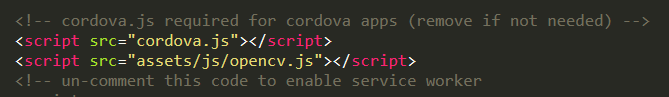
OpenCV.js là một ràng buộc JavaScript cho tập hợp con của các hàm OpenCV được chọn cho nền tảng web. Nó cho phép các ứng dụng web mới nổi với xử lý đa phương tiện hưởng lợi từ nhiều chức năng nhìn khác nhau có sẵn trong OpenCV. OpenCV.js sử dụng Emscripten để biên dịch các hàm OpenCV thành các mục tiêu asm.js hoặc WebAssembly, và cung cấp một API JavaScript cho ứng dụng web để truy cập chúng.

### **2.6.4 Sử dụng OpenCV.js trong dự án Ionic**

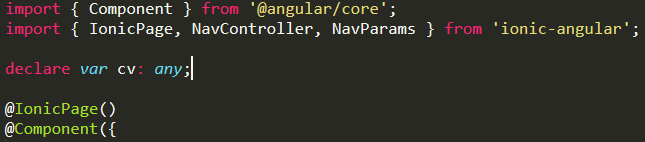
Tải file OpenCV.js tại trang <https://github.com/shkrwnd/openCV-js>

Chuyển file OpenCV.js vừa tải vào đường dẫn sau: src/assets/js.

Khai báo file OpenCV.js trong file index.html nằm trong thư mục src:



Như vậy, khi cần sử dụng OpenCV, chúng ta chỉ cần khai báo biến cv trong mỗi file TypeScript cần dùng:



## **2.7 Git và Github**

### **2.7.1 Giới thiệu Git**

Git là Hệ thống quản lý phiên bản phân tán (Distributed Version Control System *–*DVCS) mã nguồn mở được phát triển bởi [Linus Torvalds](https://vi.wikipedia.org/wiki/Linus_Torvalds) vào năm [2005](https://vi.wikipedia.org/wiki/2005), ban đầu dành cho việc phát triển [nhân Linux](https://vi.wikipedia.org/wiki/H%E1%BA%A1t_nh%C3%A2n_Linux). DVCS là hệ thống giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (commit) rồi đưa lên máy chủ nơi đặt kho chứa chính. Và một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa hoặc clone lại một tập hợp các thay đổi mới nhất trên máy tính kia.

Hiện nay, Git trở thành một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất. Trên Git, ta có thể lưu trạng thái của file dưới dạng lịch sử cập nhật. Vì thế, có thể đưa file đã chỉnh sửa một lần về trạng thái cũ hay có thể biết được file đã được chỉnh sửa chỗ nào. Nó có thể giúp các lập trình viên quản lý mã nguồn tốt hơn với cơ chế quản lý phiên bản.

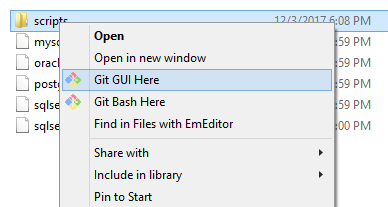
### **2.7.2 Giới thiệu Github**

GitHub là một nền tảng lưu trữ mã nguồn trên web để kiểm soát phiên bản và cộng tác dành cho các dự án có sử dụng hệ thống kiểm soát Git revision. Nó cho phép bạn và những người khác làm việc cùng nhau trên các dự án từ bất cứ đâu.

### **2.7.3 Sử dụng Git và GitHub trong dự án Ionic**

Truy cập vào trang <https://git-scm.com/download/win> để tải Git cho Windows.

Sau khi cài đặt xong, nhấn phải chuột vào một thư mục bất kỳ, một Context-Menu sẽ hiển thị, chúng ta có thể nhìn thấy các Menu-Item của Git, điều này chứng tỏ rằng chúng ta đã cài đặt Git thành công.

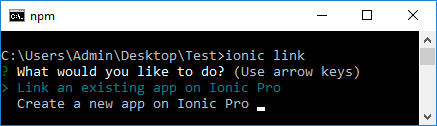


Hình 1. Menu Item của Git sau khi cài đặt xong

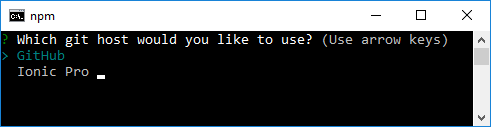
Tiếp đó, chúng ta truy cập vào trang chủ của Github <https://github.com/>, đăng ký tài khoản và tạo Repository để lưu trữ mã nguồn. Qua đó, chúng ta có thể push code từ dự án lên Repository và clone về để sử dụng thông qua giao diện của Git.

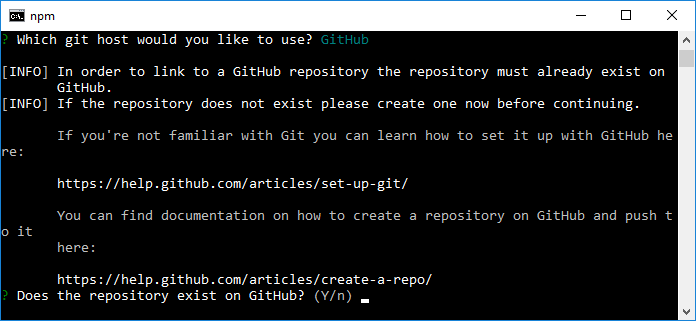
Đối với dự án Ionic, để sử dụng Git, chúng ta phải liên kết dự án Ionic, Ionic sẽ cho chúng ta 2 lựa chọn là kết nối với ứng dụng đã tồn tại trên Ionic Pro hoặc tạo một ứng dụng mới

$ ionic link

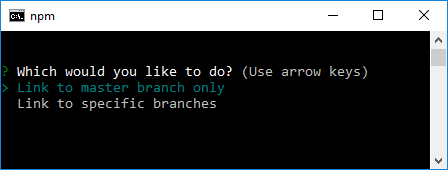


Sau đó Ionic sẽ hỏi chúng ta chọn git host của Ionic Pro hay Github, ứng dụng scanner này sử dụng Github.

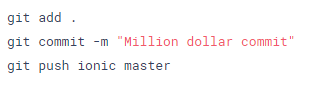


Tiếp theo Ionic sẽ hỏi Repository đó đã tồn tại hay chưa:

Nếu chưa Ionic sẽ thoát khỏi quá trình liên kết. Nếu có Ionic sẽ hiển thị danh sách Repository có trên Github. Sau khi chọn Repository, chúng ta tiếp tục bước chọn brach:



Như vậy quá trình liên kết với Github đã hoàn tất. Tiếp theo chúng ta có thể thao tác trực tiếp với Github trong Command Prompt của Node.js bằng các câu lệnh như:



# CHƯƠNG 3: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

## **Các yêu cầu và chức năng cần có**

## **Mô hình chức năng, đặc tả chức năng**

## **Mô hình dữ liệu**

# CHƯƠNG 4: MÔ TẢ GIAO DIỆN

## **4.1 Giao diện**

## **4.2 Giao diện**

## **4.3 Giao diện**

# CHƯƠNG 5: TỔNG KẾT

* 1. **Kết quả đạt được**
  2. **Hạn chế**
  3. **Hướng phát triển**

# THAM KHẢO

[] AngularJS: <https://docs.angularjs.org/guide/introduction>

[] Node.js: <https://www.tutorialspoint.com/nodejs/nodejs_introduction.html>

[] OpenCV: <https://opencv.org/>

[] OpenCV.js: <https://docs.opencv.org/3.4/df/d0a/tutorial_js_intro.html>

[] Apache Cordova: <https://cordova.apache.org/docs/en/latest/guide/overview/>

[] Firebase: <https://firebase.google.com/docs/auth/>

[] Ionic: <https://ionicframework.com/docs/intro/concepts>

[] TypeScript: <https://www.typescriptlang.org/docs/handbook/declaration-files/introduction.html>

[] Git: [https://vi.wikipedia.org/wiki/Git\_(phần mềm)](https://vi.wikipedia.org/wiki/Git_(ph%E1%BA%A7n_m%E1%BB%81m))

[] Github: <https://guides.github.com/activities/hello-world/>

[]

[]