**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**KHOA KỸ THUẬT – CÔNG NGHỆ**



**BÁO CÁO MÔN HỌC  
THỰC TẬP DOANH NGHIỆP**

***ĐỀ TÀI:***

**TÌM HIỂU VỀ ANGULAR FRAMEWORK**

**GVHD: ThS. Nguyễn Kim Duy**

**SVTH: LÊ THANH HIỆP**

**MSSV: 1824801030217**

**LỚP: D18PM04**

**BÌNH DƯƠNG – 12/2020**

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**KHOA KỸ THUẬT – CÔNG NGHỆ**



**ĐỒ ÁN MÔN HỌC  
THỰC TẬP DOANH NGHIỆP**

***ĐỀ TÀI:***

**TÌM HIỂU VỀ ANGULAR FRAMEWORK**

**GVHD: ThS. Nguyễn Kim Duy**

**SVTH: LÊ THANH HIỆP**

**MSSV: 1824801030217**

**LỚP: D18PM04**

**BÌNH DƯƠNG – 12/2020**

**TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT**

**KHOA KỸ THUẬT – CÔNG NGHỆ**

**NHẬN XÉT VÀ CHẤM ĐIỂM CỦA GIẢNG VIÊN**

Họ và tên giảng viên: **Ths. Nguyễn Kim Duy**

Tên báo cáo: **TÌM HIỂU TỔNG QUAN VỀ ....**

Nội dung nhận xét:

**Điểm:**

Bằng số:

Bằng chữ:

|  |  |
| --- | --- |
|  | **GIẢNG VIÊN CHẤM**  *(Ký, ghi rõ họ tên)* |

**ThS. Nguyễn Kim Duy**

DANH MỤC

[LỜI MỞ ĐẦU 6](#_Toc58677530)

[I. Tổng quan về hệ điều hành Ubuntu 7](#_Toc58677531)

[1. Hệ điều hành Ubuntu 7](#_Toc58677532)

[2. Các phiên bản của Ubuntu 8](#_Toc58677533)

[3. Các bản phân phối của Ubuntu 8](#_Toc58677534)

[4. Cài đặt Ubuntu song song Windows 9](#_Toc58677535)

[II. Tổng quan về Angular Framework 20](#_Toc58677536)

[1. Giới thiệu về Angular 20](#_Toc58677537)

[2. Cách cài đặt Angular 22](#_Toc58677538)

[3. Khởi tạo dự án đầu tiên 27](#_Toc58677539)

[I. Kiến trúc mức cao của ứng dụng 30](#_Toc58677540)

[II. Các thành phần cơ bản của Angular 31](#_Toc58677541)

[1. Module 31](#_Toc58677542)

[2. Component 35](#_Toc58677543)

[3. Thư viện Angular 36](#_Toc58677544)

[4. Structure Directive 37](#_Toc58677545)

[5. Data Binding 38](#_Toc58677546)

[**6.** **Demo Front-End với Angular** 40](#_Toc58677547)

[CHƯƠNG 3 KẾT LUẬT VÀ ĐỊNH HƯỚNG PHÁT TRIỂN 42](#_Toc58677548)

[**I.** **Kết quả đạt được** 42](#_Toc58677549)

[**II.** **Hướng phát triển đề tài** 42](#_Toc58677550)

[TÀI LIỆU THAM KHẢO 43](#_Toc58677551)

MỞ ĐẦU

LỜI MỞ ĐẦU

Ngày nay, chúng ta đang được sống trong kỷ nguyên của tin học nhờ sự vượt bậc, sự bùng nổ mạnh mẽ của công nghệ thông tin. Công nghệ thông tin không chỉ dừng lại ở mục đích phục vụ cho khoa học kỹ thuật mà đi sâu vào đời sống, chính trị, kinh tế, xã hội, trở nên thân thiện, gần gũi, mang lại nhiều lợi ích cho con người. Công nghệ thông tin ngày càng khẳng định dược tính hữu dụng và sức mạnh trong mọi phương diện, mọi ngành nghề của cuộc sống, nhất là trong thời đại kinh tế thị trường như bây giờ.

Đi kèm theo đó, các hệ thống website cũng phát triển không ngừng với đủ các lĩnh vực: văn hóa, thời sự, khoa học công nghệ, làm đẹp, nấu ăn, thể thao, ca nhạc, phim ảnh,... Tuy nhiên, để tạo ra một trang web có sức sống bền lâu thì bản thân nó phải mang lại lợi ích cho nhiều người. Giao diện bắt mắt là yếu tố quan trọng để người xem click chuột đến website của bạn. Nhưng yếu tố quyết định giữ chân độc giả lại là nội dung mà website cung cấp.

Trong thời gian kiến tập doanh nghiệp, được sự giúp đỡ của ThS. Nguyễn Kim Duy, em đã hoàn thành bài báo cáo thực tập của mình với đề tài Tìm hiểu về Angular Framework. Tuy nhiên với tầm hiểu biết còn hạn hẹp, kiến thức về công nghệ thiết kế web bằng Angular chưa nhiều nên không thể tránh khỏi những thiếu sót. Rất mong được sự góp ý của các giảng viên để bài báo cáo được hoàn chỉnh.

1. TỔNG QUAN MÔI TRƯỜNG VÀ CÔNG CỤ

1. Tổng quan về hệ điều hành Ubuntu

Hệ điều hành Ubuntu

* 1. Nguồn gốc của Ubuntu
* Ubuntu là tên của bản phân phối, đại thể bắt nguồn từ quan niệm "Ubuntu" của Nam Phi - "con người hướng đến con người". Bản phân phối Ubuntu mang tinh thần của quan niệm đó vào thế giới phần mềm.
  1. Giới thiệu về Ubuntu
  + Ubuntu là một hệ điều hành mã nguồn mở được phát triển bởi cộng đồng chung dựa trên nền tảng Debian GNU/Linux, nó được tài trợ bởi Canonical Ltd (chủ sở hữu là Mark Shuttleworth), rất phù hợp cho máy tính để bàn, máy tính xách tay và máy chủ. Dù bạn dùng cho máy tính ở nhà, ở trường hay trong công sở, Ubuntu có đầy đủ các chường trình bạn cần, từ phần mềm soạn thảo văn bản và gửi nhận thư, đến các phần mềm máy chủ web và các công cụ lập trình. Ubuntu hoàn toàn miễn phí. Bạn không phải trả tiền bản quyền cho bất cứ ai. Bạn có thể tải về, sử dụng và chia sẻ với bạn bè, gia đình hoặc đồng nghiệp mà không mất một khoản phí nào cả. Mỗi phiên bản Ubuntu đều được cập nhật 6 tháng một lần, điều đó có nghĩa là bạn luôn luôn có các ứng dụng mới nhất trong thế giới phần mềm mã nguồn mở. Ubuntu được thiết kế với tiêu chí chuyên về bảo mật. Bạn có thể lấy về các bản cập nhật về bảo mật ít nhất là 18 tháng trên máy để bàn và máy chủ. Với phiên bản Hỗ trợ dài hạn (Long Term Support - LTS), bạn sẽ có 3 năm hỗ trợ với máy để bàn và 5 năm hỗ trợ đối với máy chủ. Bạn không phải trả thêm một khoản phí nào cho phiên bản LTS. Việc nâng cấp lên phiên bản mới hơn của Ubuntu hoàn toàn miễn phí.
  + Mọi thứ bạn cần đều nằm trong một đĩa CD, đã cung cấp cho bạn một môi trường làm việc đầy đủ. Những phần mềm mở bổ sung, bạn có thể lấy về trực tiếp từ kho phần mềm miễn phí trên Internet. Ubuntu hướng đến người dùng phổ thông nên được bản địa hóa với sự giúp đỡ của cộng đồng người dùng mã mở các loại ngôn ngữ trên thế giới (trong đó có tiếng Việt). Bạn có thể tùy chỉnh ngôn ngữ sử dụng trong giao diện hiển thị bất kì lúc nào. Giao diện cài đặt cho phép bạn thực hiện các thao tác cài đặt một cách nhanh chóng và dễ dàng. Quá trình cài đặt tiêu chuẩn thường mất không quá 25 phút. Sau khi bạn cài đặt xong, hệ thống của bạn hoàn toàn có thể dùng được ngay mà không cần phải cấu hình, bổ sung thêm gì cả. Một loạt các ứng dụng cần thiết đã được cài đặt kèm theo trong quá trình cài đặt Ubuntu.
  1. Yêu cầu phần cứng
  + Ubuntu có thể cài đặt trên các loại máy tính để bàn, máy tính mini có cấu trúc dựa trên nền tảng Intel-based Mac architectures (dành cho chíp 64bit có phiên bản riêng). Yêu cầu tối thiểu để chạy Ubuntu là 256MB RAM (384MB RAM để chạy trực tiếp Ubuntu từ đĩa CD), nhưng chúng tôi khuyến cáo nên cài đặt trên máy có 512MB RAM trở lên. Ubuntu khi cài đặt vào ổ đĩa cứng cần ít nhất 4GB trống (bao gồm cả phân vùng trao đổi). Nên có card đồ họa mạnh để sử dụng các hiệu ứng trên giao diện đồ họa.

Các phiên bản của Ubuntu

* Ubuntu hướng đến nhiều đối tượng khách hàng và cho những nhu cầu khác nhau, do đó, ubuntu cung cấp nhiều phiên bản cho khách hàng có sự lựa chọn thoải mái.
* Ubuntu Desktop
* Ubuntu Server
* Ubuntu Cloud
* Ubuntu flavours
* Ubuntu for IoT

Các bản phân phối của Ubuntu

3.1. Các bản phân phối chính thức

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên bản phân phối | Phiên bản mới nhất | Trang web chính thức | Giao diện sử dụng |
| 1 | Kubuntu | 18.10 | https://kubuntu.org/ | KDE |
| 2 | Lubuntu | 18.10 | https://lubuntu.me/ | LXDE |
| 3 | Xubuntu | 18.04 | https://xubuntu.org/ | XFCE |
| 4 | Mythubuntu | 18.04 | http://www.mythbuntu.org/ |  |
| 5 | Ubuntu Studio | 18.10 | https://ubuntustudio.org/ |  |

3.2. Các bản phân phối không chính thức

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Tên bản phân phối | Phiên bản mới nhất | Trang web chính thức |
| 1 | Elementary OS | 5 Juno | https://elementary.io/ |
| 2 | Netrunner | 2018.08 | https://www.netrunner.com/ |
| 3 | Moon OS |  | *(không có)* |
| 4 | Peppermint OS | 9 | https://peppermintos.com/ |
| 5 | Trisquel | 8.0 | https://trisquel.info/ |
| 6 | Linux Mint | 19.1 | https://linuxmint.com/ |

Cài đặt Ubuntu song song Windows

* Chúng ta có thể thấy được sự tuyệt vời của Ubuntu dành cho dân lập trình, tuy nhiên Windows cũng không kém cạnh khi có nhiều ứng dụng hỗ trợ tốt về đồ hoạ hay các công việc liên quan đến đồ hoạ. Do đó, việc cài đặt song song Ubuntu với Windows luôn là lựa chọn hàng đầu để có thể vừa sử dụng những tính năng tuyệt vời của Ubuntu lại vừa có thể sử dụng những chức năng hay ho của Windows.

4.1. Chuẩn bị công cụ

* Để có thể cài đặt Ubuntu, chúng ta cần chuẩn bị USB có dung lượng tối thiểu 2GB và tệp tin cài đặt Ubuntu .iso.
* Để tải Ubuntu cần truy cập vào địa chỉ:

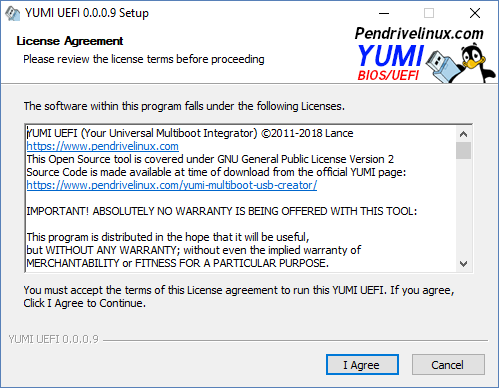
<https://www.ubuntu.com/download/desktop>

* Tiếp đó, cần tải phần mềm YUMI – Multiboot USB Creator để sao chép bộ cài đặt vào USB theo địa chỉ:

<https://www.pendrivelinux.com/yumi-multiboot-usb-creator/>

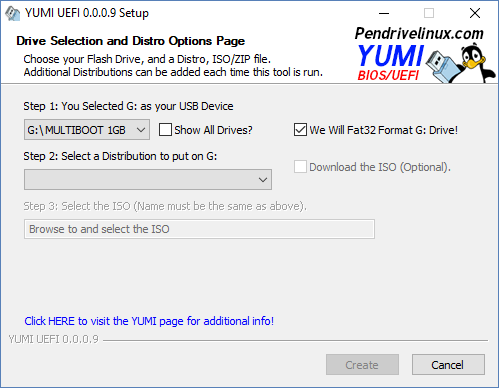
* Cài đặt phần mềm vừa tải về xong, lưu ý cần tắt tường lửa và các chương trình antivirus khi có lỗi cài đặt.

*Bước 1:* Khởi động phần mềm YUMI – Multiboot USB Creator trên máy tính, nhấn vào nút “I Agree” để đồng ý các điều khoản cài đặt.



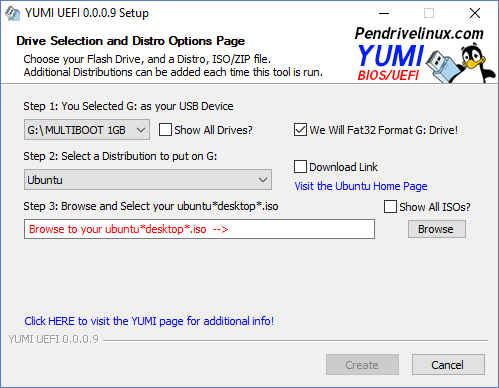
Hình 3. Bước 1. Tạo USB cài đặt Ubuntu

*Bước 2:* Tại phần “Step 1: Select the Drive Letter of your USB Device” chúng ta chọn đúng ổ USB ta cần tạo và chọn vào Format G: Drive (Erase Content) (Ở đây USB của mình là ổ đĩa G).



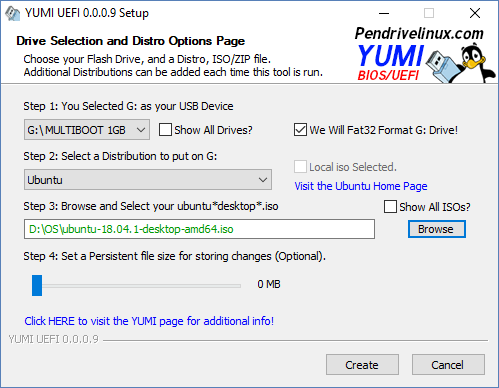
Hình 4. Bước 2. Tạo USB cài đặt Ubuntu

*Bước 3:* Tại phần “Step 2: Select a Distribution to put on G” chúng ta chọn phiên bản “Ubuntu”.



Hình 5. Bước 3. Tạo USB cài đặt Ubuntu

*Bước 4:* Tại phần “Step 3: Browse and Select your ubuntu\*desktop\*.iso”, chúng ta chọn “Browse” và trỏ đến bộ cài ubuntu .iso vừa tải ở trên và chọn “Create” để tiến hành tạo bộ cài.

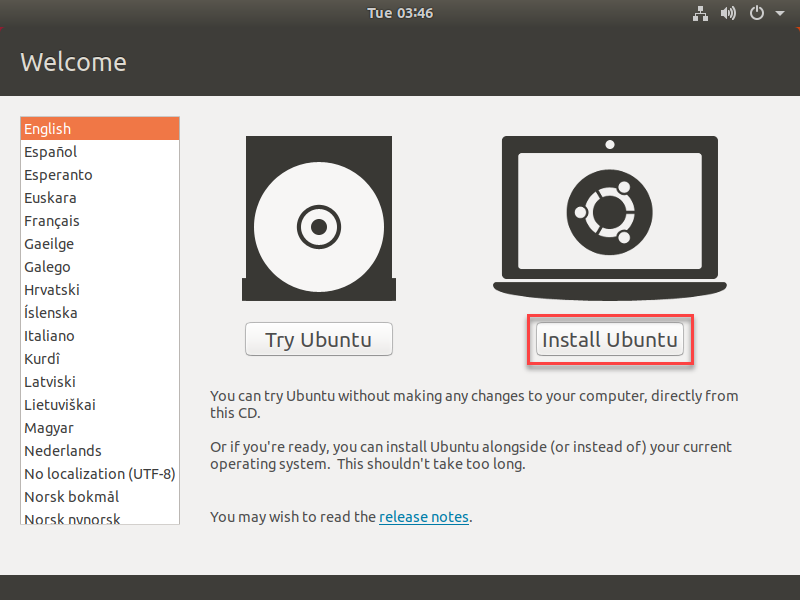


Hình 6. Bước 4. Tạo USB cài đặt Ubuntu

4.2. Cài đặt Ubuntu

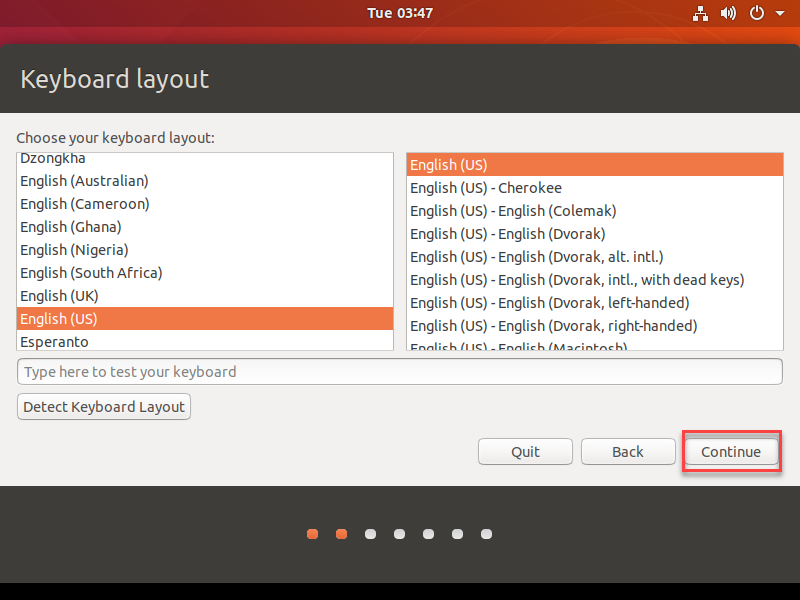
* Cắm USB vừa tạo vào máy tính cần cài, tiến hành khởi động máy tính và truy cập vào boot để khởi động phần cài đặt Ubuntu.

*Bước 1:* Tại giao diện chính, chúng ta có 2 sự lựa chọn: “Try Ubuntu” và “Install Ubuntu”. Chúng ta chọn “Install Ubuntu” để thực hiện việc cài đặt hoặc chọn “Try Ubuntu” để dùng thử và trải nghiệm mà không cần cài đặt.



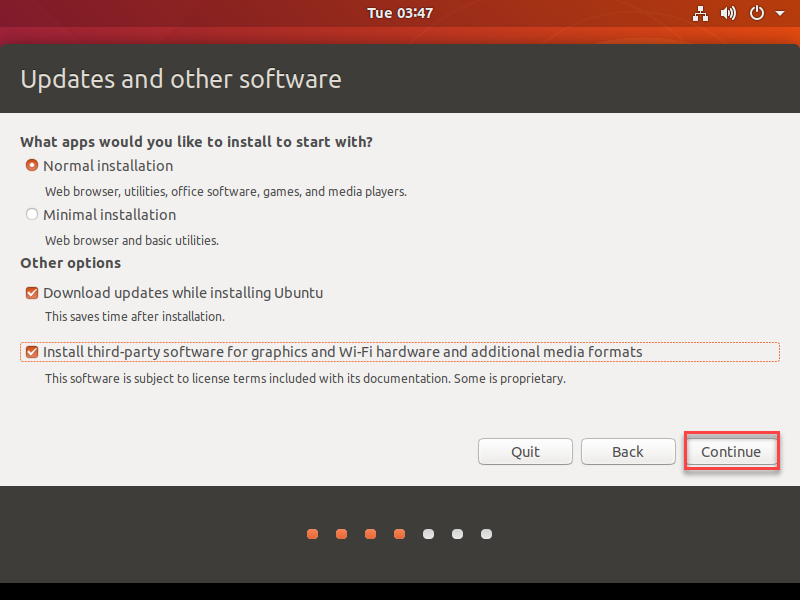
Hình 7. Lựa chọn cài đặt hoặc dùng thử Ubuntu

*Bước 2:* Bước này ta sẽ chọn bố cục bàn phím, ta sẽ để mặc định là “English (US)” và chọn “Continue”.



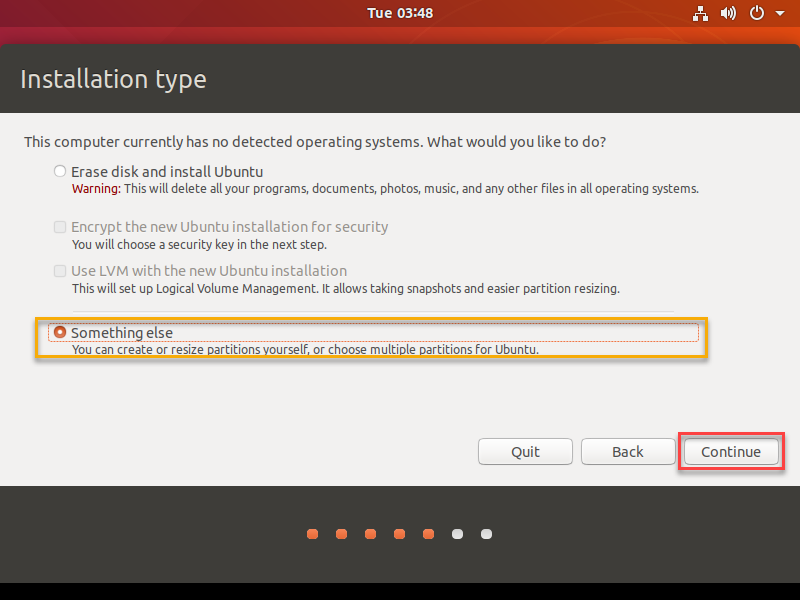
Hình 8. Chọn bố cục bàn phím

*Bước 3:* Chọn “Install third-party software for graphics and Wi-Fi hardware and additional media formats” để cài đặt thêm driver cho phần cứng và phần mềm hỗ trợ. Sau đó chọn “Continue”.



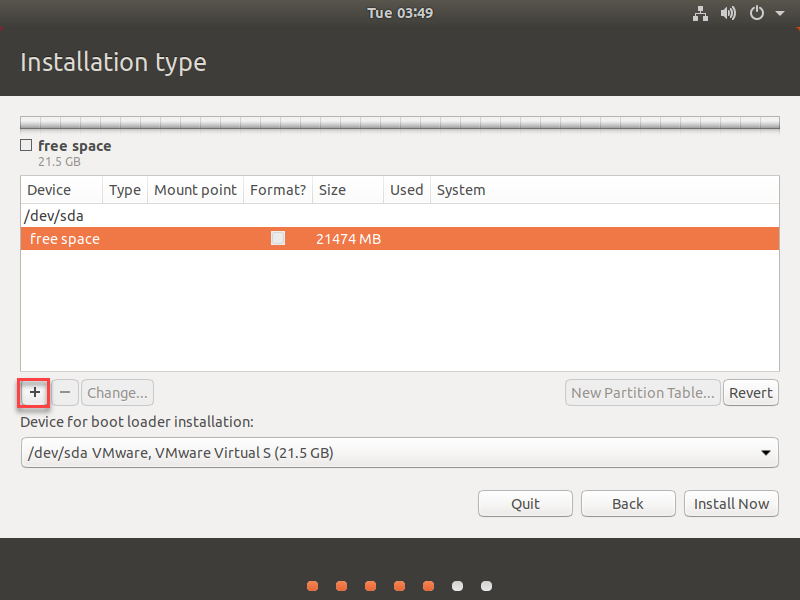
Hình 9. Tuỳ chọn các thuộc tính khi cài đặt Ubuntu

*Bước 4:* Tại bước này, đặc biệt chú ý chọn “Something else” để lựa chọn phân vùng cần cài đặt Ubuntu lên đó. Nếu chọn “Erase disk and install ubuntu”, ta sẽ xoá sạch tất cả mọi thứ, kể cả Windows. Do đó cần chọn “Something else” để không mất dữ liệu.



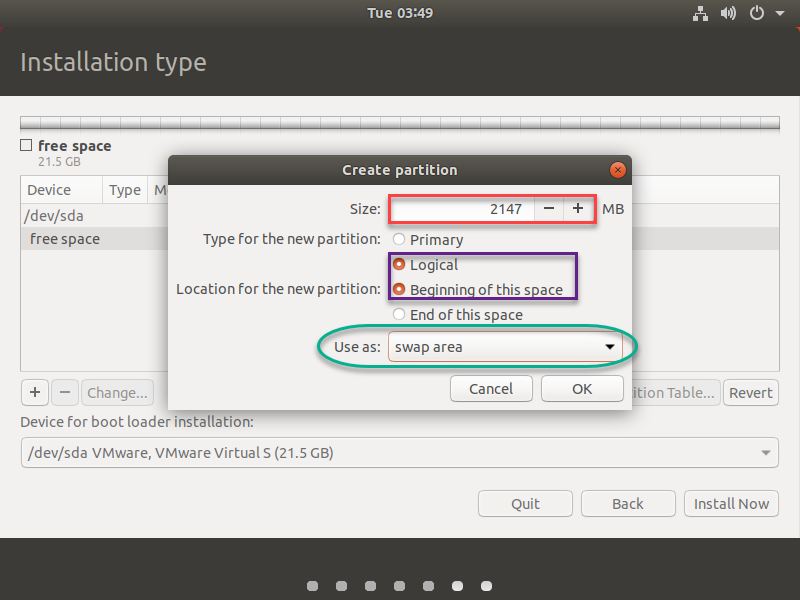
Hình 10. Lựa chọn kiểu cài đặt Ubuntu

*Bước 5:* Chọn vào phân vùng trống và chọn dấu cộng “+” để thêm phân vùng.



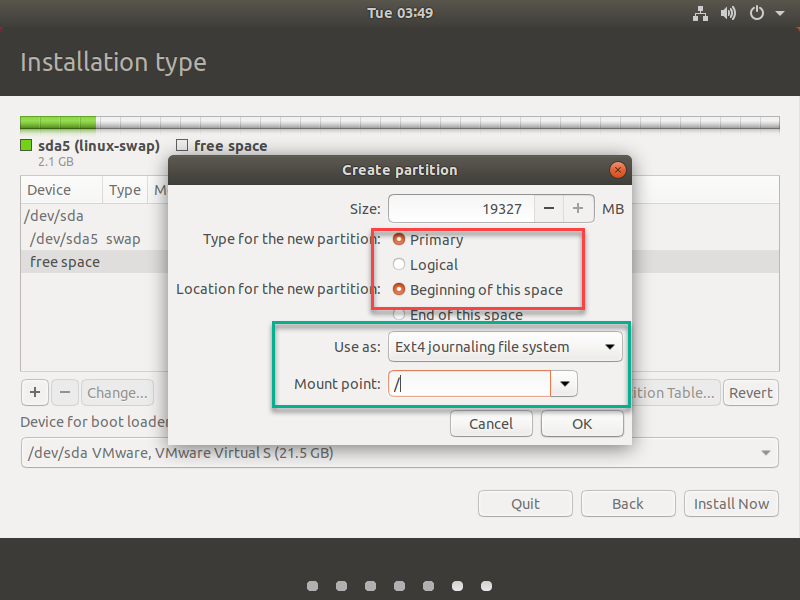
Hình 11. Chọn phân vùng để cài đặt

* Tại phần “Size”, ta chọn khoảng 1024MB trở lên để tạo RAM ảo cho hệ điều hành. Chọn “Logical” và “Begining of this space”. Tại mục “Use as” chọn “swap area”



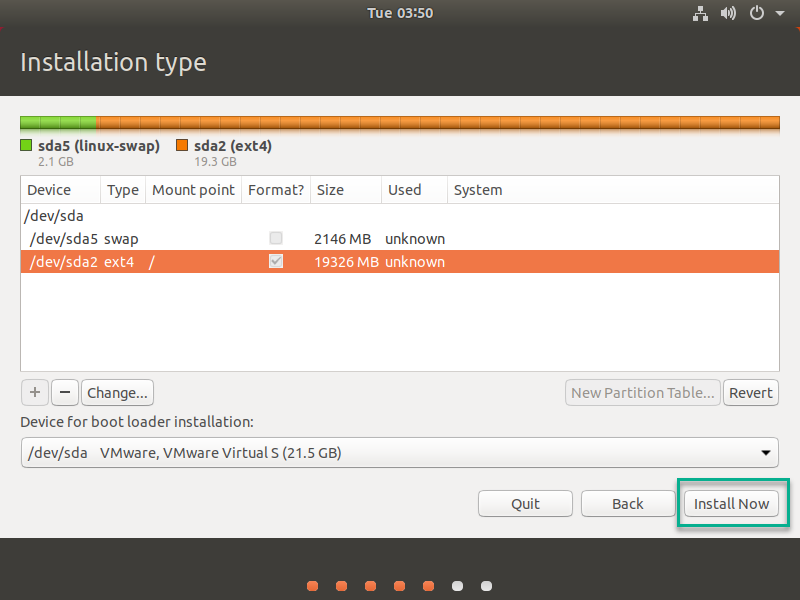
Hình 12. Tạo phân vùng swap

*Bước 6:* Tiếp tục chọn dấu cộng “+”. Phần “Size” chọn kích thước phân vùng mình cần để cài đặt Ubuntu. Chọn “Primary” và “Beginning of this space”. Tại mục “Use as” chọn “Ext4 journaling file system” và mục “Mount point” chọn “/”.



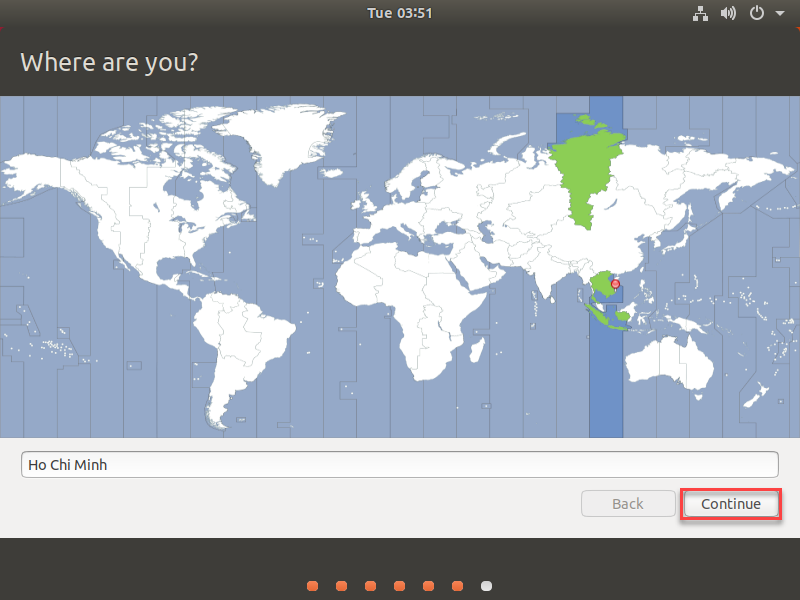
Hình 13. Tạo phân vùng Ext4

*Bước 7:* Nhấn chọn “Install Now” để cài đặt Ubuntu.



Hình 14. Cài đặt Ubuntu

*Bước 8:* Tại bước này, ta chọn vùng và lãnh thổ theo bản đồ và chọn “Continue”

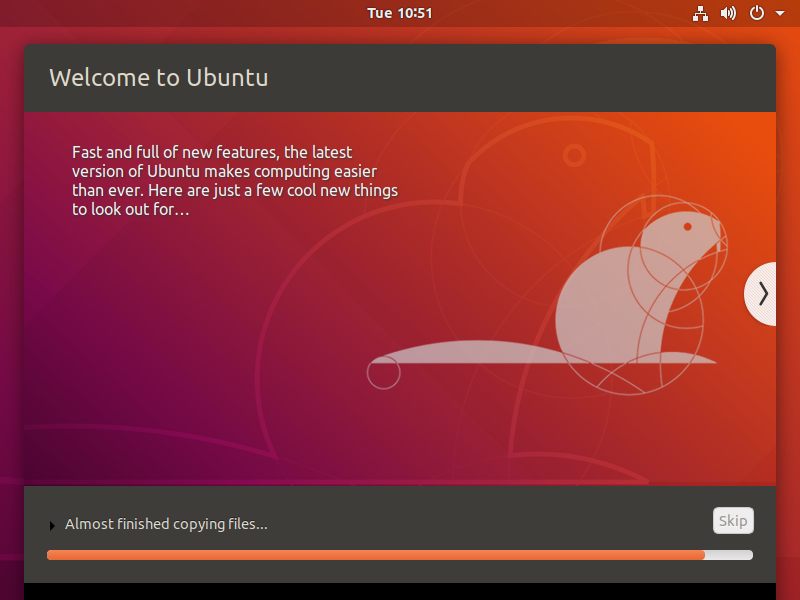


Hình 15. Lựa chọn vị trí quốc gia và vùng lãnh thổ

*Bước 9:* Bước này ta điền các thông tin cơ bản và chọn “Continue”, đặc biệt phải nhớ username và password để đăng nhập khi cài đặt xong.

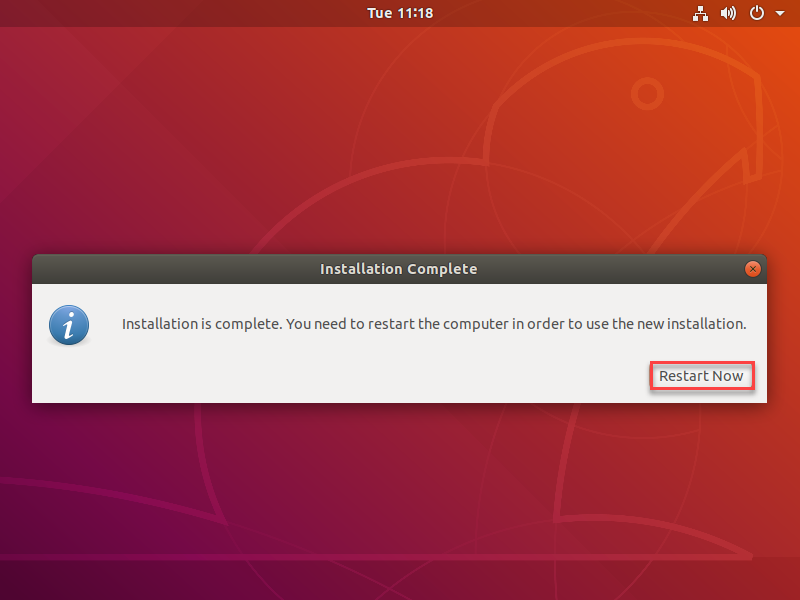


*Bước 10:* Quá trình cài đặt đang diễn ra.



Hình 17.Quá trình cài đặt đang được diễn ra

*Bước 11:* Chọn “Restart Now” để khởi động lại máy và áp dụng các cài đặt mới của hệ điều hành Ubuntu.



Hình 18. Quá trình cài đặt thành công và yêu cầu khởi động lại

1. Tổng quan về Angular Framework
2. Giới thiệu về Angular
   1. AngularJS

* AngularJs là một Javascript Framework được tạo ra để xây dựng các ứng dụng web động (dynamic web app), nó thường được sử dụng để tạo ra các ứng dụng một trang ( Single Page Application - SPA). Và hoạt động dựa trên việc mở rộng thêm các thuộc tính (attribute) cho các thẻ của HTML (Các thuộc tính theo quy tắc của Angular). AngularJS là một nền tảng miễn phí và được hàng ngàn lập trình viên trên thế giới ưa chuộng.
* Thời gian sau đó Misko Hevery tham gia vào một dự án có tên Google Feedback (Phản hồi Google) với tư cách một lập trình viên làm việc bán thời gian. Misko Hevery và 2 lập trình viên khác đã viết 17.000 dòng mã cho dự án Google Feedback trong vòng 6 tháng. Số lượng mã ngày càng lớn và Misko Hevery nhận thấy rằng nó ngày càng khó để sửa đổi và kiểm soát các lỗi phát sinh.
* Vì vậy Misko Hevery đã đánh cược với người quản lý của mình rằng anh có thể viết lại toàn bộ mã cho Google Feedback bằng cách sử dụng dự án GetAngular của mình trong 2 tuần. Hevery đã thua cuộc, thay vì 2 tuần anh đã mất 3 tuần để hoàn thành công việc, tuy nhiên mã của dự án đã giảm từ 17.000 dòng xuống còn 1.500 dòng.
* Nhờ sự thành công của Hevery, người quản lý của anh ấy Brad Green đã chú ý và phát triển AngularJS. AngularJS đã thực sự tăng tốc trong thời gian đó.
* Sau những thành công đó, nhóm Angular đã được hình thành bên trong Google và phát triển nó sử dụng từ nội bộ đến bên ngoài.
  1. Các phiên bản của Angular
* Thực tế Angular có 2 phiên bản hoàn toàn khác nhau. Đó là phiên bản Angular phiên bản 1 và khác 1.
* *Phiên bản 1:*
* Phiên bản 1 có tên chính thức là AngularJS và vẫn đang được phát triển.
* AngularJS dựa trên Javascripts và các lập trình viên sẽ viết theo ngôn ngữ Javascripts, vì vậy nó khá dễ học và dễ làm. Tuy nhiên nó thường được so sánh với ReactJS và hiện nay, các công ty thường cân nhắc khi sử dụng AngularJS.
* Dưới đây là các phiên bản và ngày phát hành của AngularJS:

|  |  |  |
| --- | --- | --- |
| STT | Phiên bản | Ngày phát hành |
| 1 | 1.7.3 | 09/08/2018 |
| 2 | 1.7.2 | 12/06/2018 |
| 3 | 1.7.1 | 08/06/2018 |
| 4 | 1.7.0 | 11/05/2018 |
| 5 | 1.6.10 | 18/04/2018 |
| 6 | 1.6.9 | 02/02/2018 |
| 7 | 1.6.8 | 22/12/2017 |
| 8 | 1.6.7 | 25/11/2017 |
| 9 | 1.6.6 | 18/08/2017 |

Bảng 1. Các phiên bảng và ngày phát hành của AngularJS

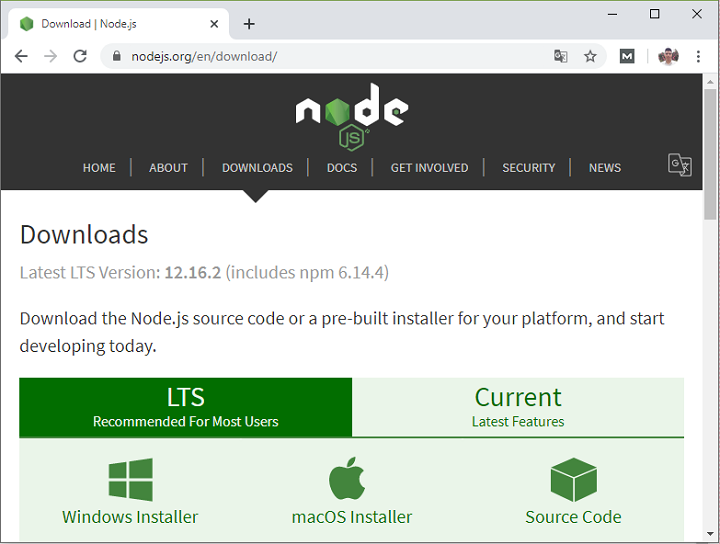
* *Phiên bản khác 1:*
* Bắt đầu từ phiên bản 2 trở đi, tên chính thức là Angular. Từ Angular 2 trở đi, nó thay đổi rất nhiều. Điển hình là Component và Typescripts.
* *Component:*
* Ý tưởng chủ đạo của Angular là Component (thành phần), component không quá khác biệt trong ReactJS nhưng Component là khái niệm hoàn toàn mới mà AngularJS không có. Trong Angular, Component được áp dụng để tạo cấu trúc code rõ ràng và có khả năng tái sử dụng các component có sẵn.
* *Typescripts:*
* Typescripts được Microsoft phát triển để viết mã Javascripts mang phong cách theo hướng đối tượng. Bản thân Javascripts nguyên thuỷ được thiết kế rất đơn giản và đi theo hướng lập trình chức năng nên sẽ không có tính năng kế thừa, đa hình,...
* Về bản chất, typescripts chứa nhiều cú pháp mà trình duyệt không hiểu được, vì vậy nó không thể chạy trực tiếp trên trình duyệt nên nó cần 1 bước biên dịch mã Typescripts thành Javascripts thông thường. Vai trò của Typescripts như là một ngôn ngữ trung gian, nó thường được biết tới như ngôn ngữ đặc tả (language specification).
* Qua nhiều năm phát triển, Angular đã có rất nhiều phiên bản:

|  |  |  |
| --- | --- | --- |
| STT | Phiên bản | Ngày phát hành |
| 1 | 7.0.0 | 11/2018 |
| 2 | 6.0.0 | 05/05/2018 |
| 3 | 5.0.0 | 01/11/2017 |
| 4 | 4.0.0 | 12/12/2016 |
| 5 | 2.0.0 | 23/10/2014 |

Bảng 2. Các phiên bản và ngày phát hành của Angular

1. Cách cài đặt Angular
   1. Cài đặt trên Windows
      1. Cài đặt Node.js

* Truy cập địa chỉ: <https://nodejs.org/en/>, tải Node.js về máy tính và tiến hành cài đặt.

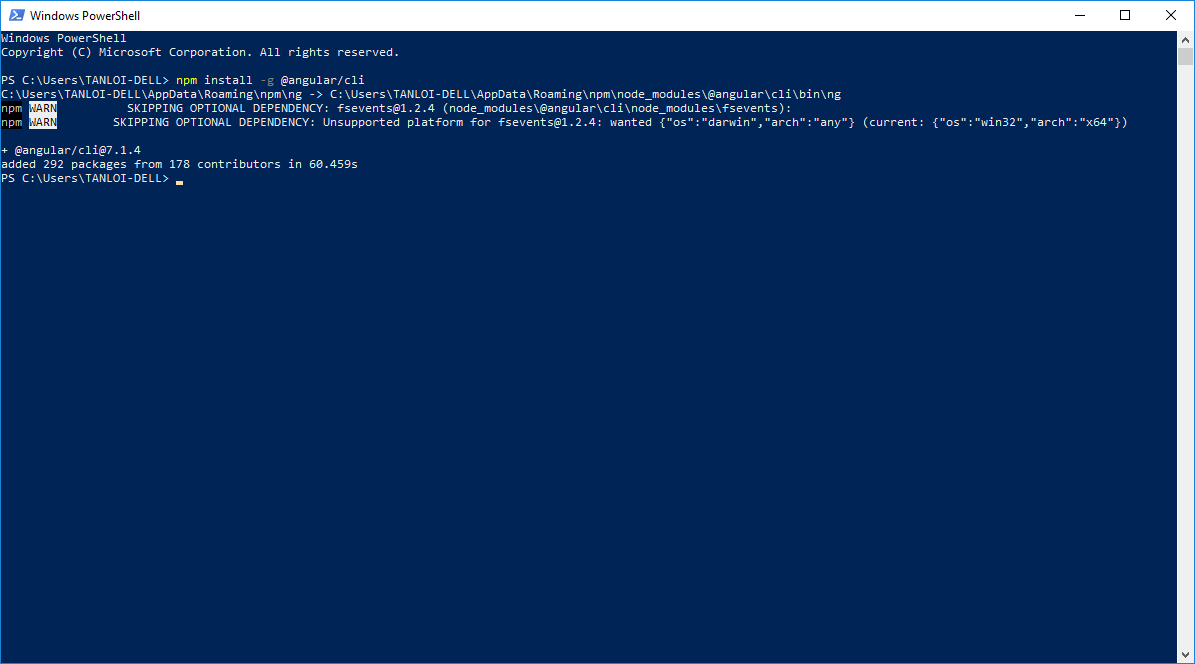


Hình 19. Trang chủ Node.js

* + 1. Cài đặt Angular
* Mở chương trình Windows PowerShell và thực hiện chạy đoạn lệnh:

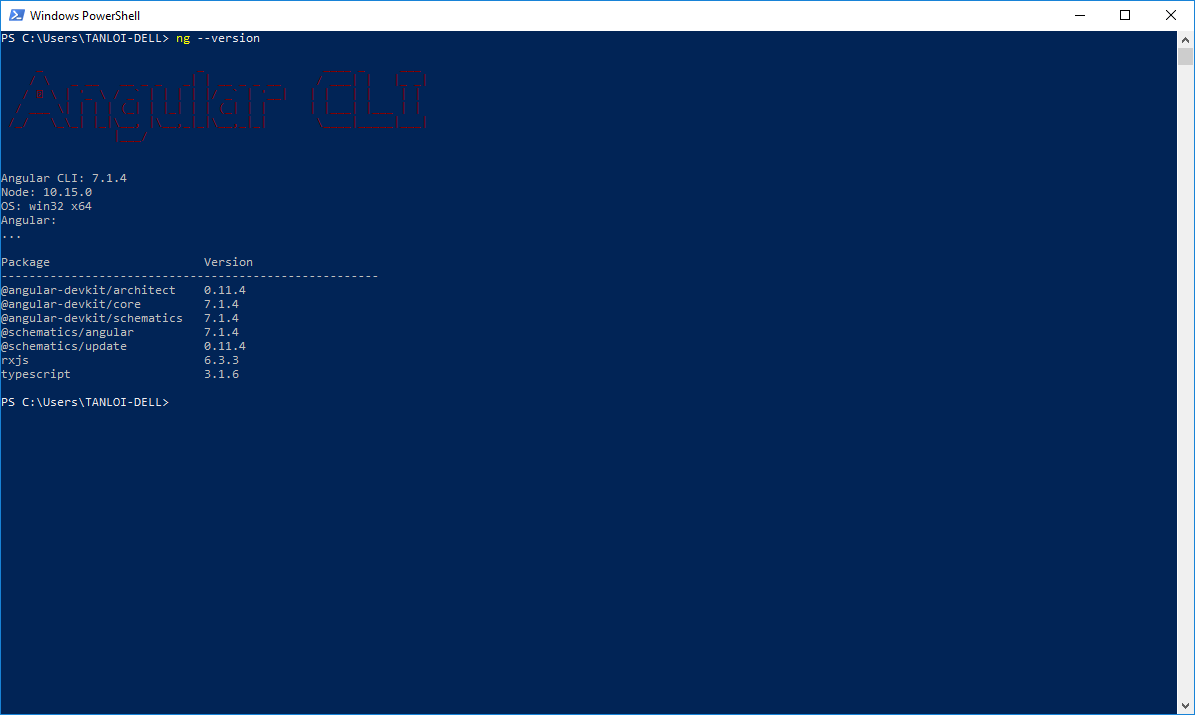
*npm install -g @angular/cli*

và hệ thống sẽ tự động tải và cài đặt Angular về máy cho chúng ta.



Hình 20. Cài đặt Angular bằng Windows PowerShell trên Windows

* Để điểm tra phiên bản Angular vừa cài đặt vào máy, ta thực hiện đoạn lệnh: *ng –version* hoặc *ng -v*

**

Hình 21. Kiểm tra phiên bản Angular hiện hành

* 1. Cài đặt trên Ubuntu
     1. Cài đặt Node.js
* Để thực hiện cài đặt Node.js, ta mở Terminal bằng tổ hợp phím Ctrl + Alt + T và chạy đoạn lệnh: sudo apt install nodejs.

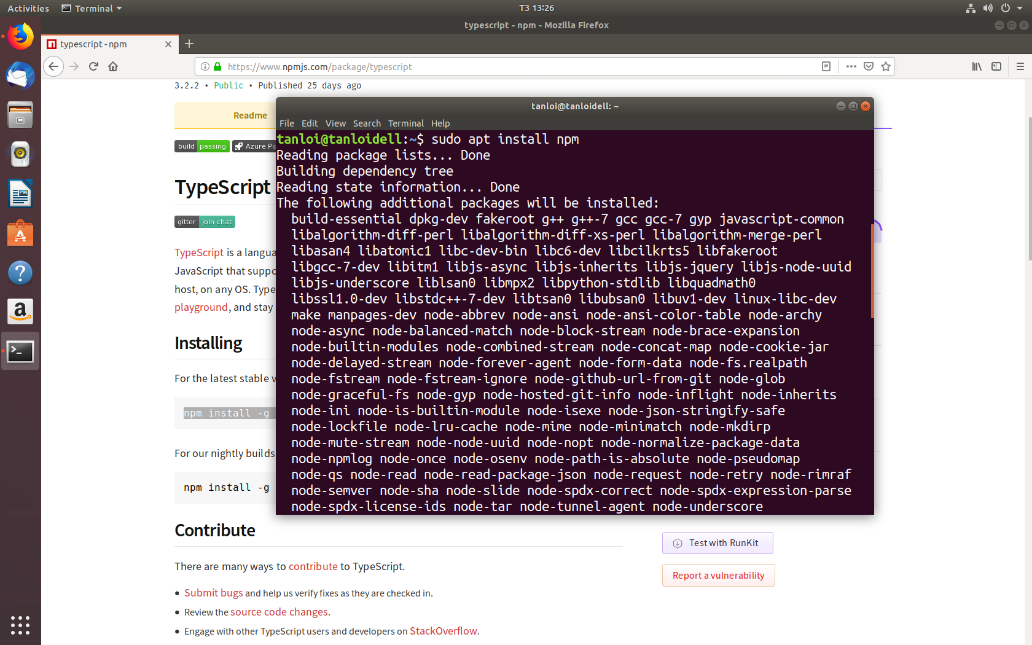


Hình 22. Cài đặt Node.js bằng Terminal trong Ubuntu

* Để kiểm tra phiên bản Node.JS hiện hành ta thực thi đoạn lệnh:

*node --version*

* + 1. Cài đặt Node Package Manager (npm)
* Ta thực thi đoạn lệnh: *sudo apt install npm*



Hình 23. Cài đặt Node Package Manager

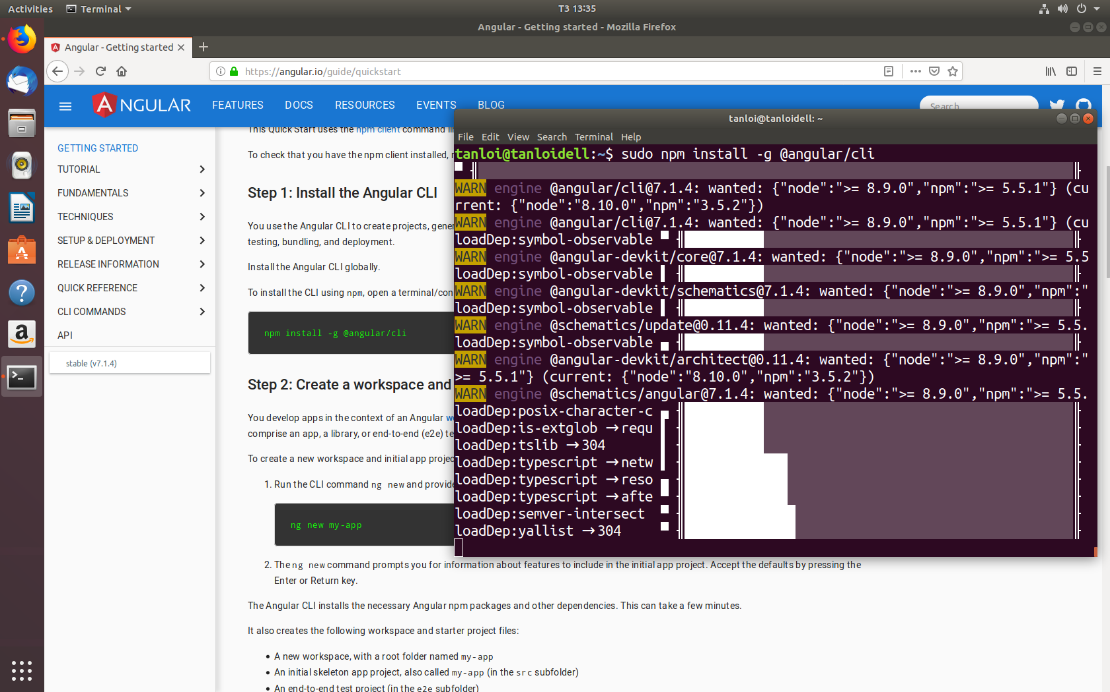
* + 1. Cài đặt TypeScript
* Ta thực thi đoạn lệnh: *npm install -g typescript*

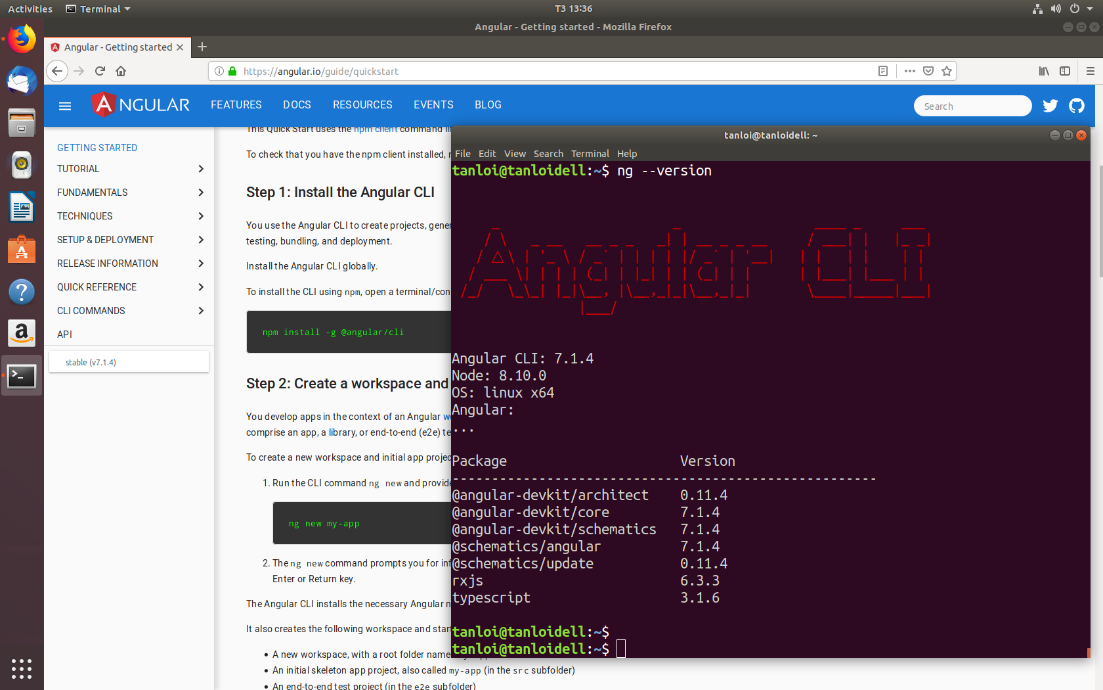


Hình 24. Cài đặt TypeScript

* Để kiểm tra phiên bản hiện hành: *tsc --version*
  + 1. Cài đặt Angular
* Để cài đặt Angular, ta thực thi đoạn lệnh sau:

*sudo npm install -g @angular/cli*



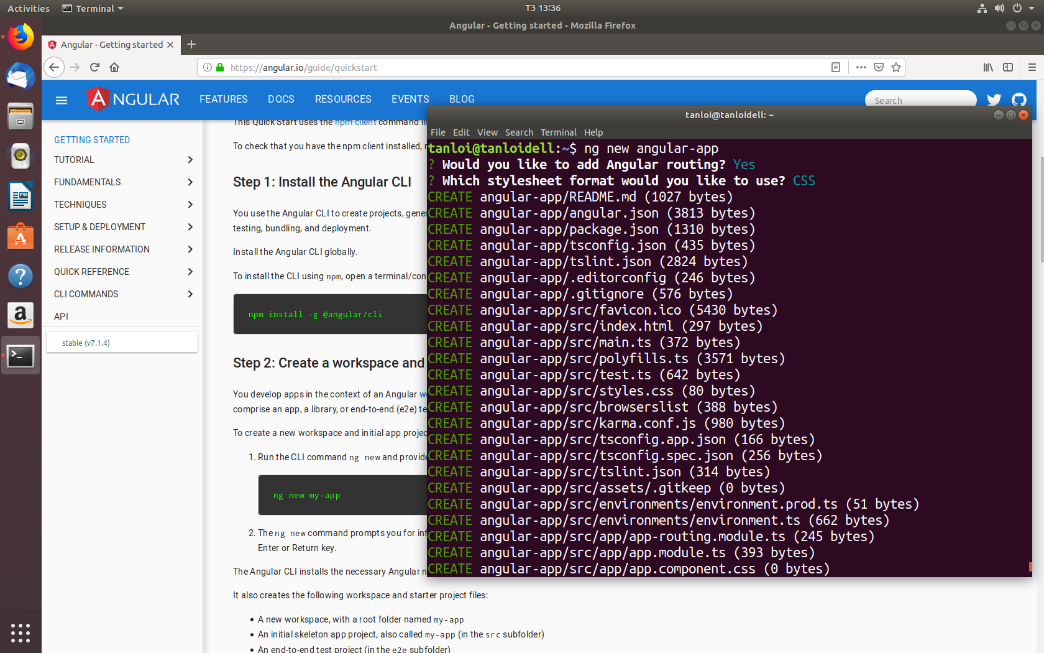
* Để kiểm tra phiên bản hiện hành: *ng –version*
* Trong ảnh, phiên bản hiện hành của ta là Angular 7.1.4, tức là Angular 7 (phiên bản mới nhất hiện thời).
* 

Hình 25. Kiểm tra phiên bản Angular hiện hành (Ubuntu)

1. Khởi tạo dự án đầu tiên
   1. Khởi tạo dự án đầu tiên

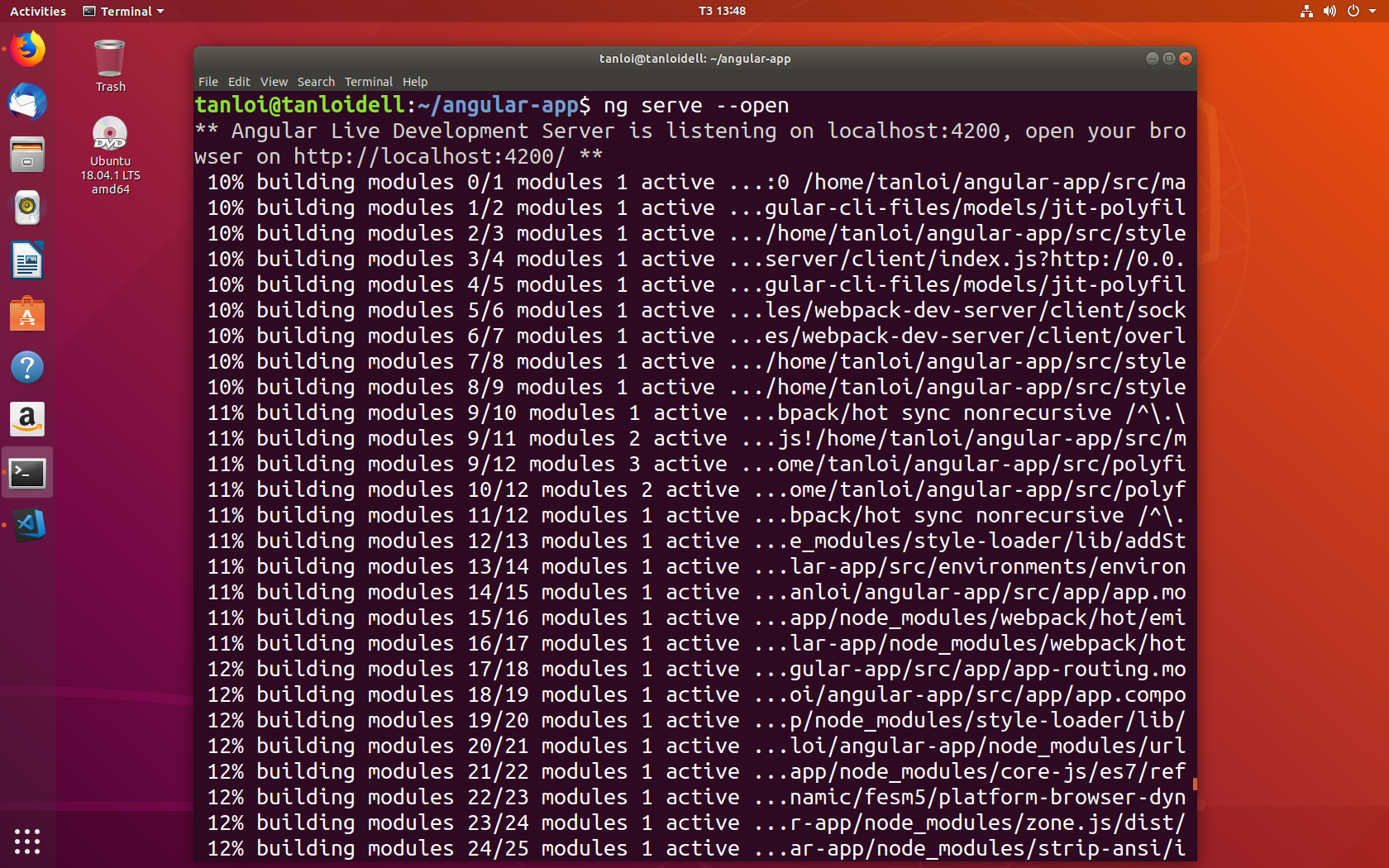
* Trong Angular, để khởi tạo được dự án đầu tiên, ta bắt buộc phải tạo thông qua dòng lệnh.
* Đầu tiên, ta cần trỏ đến vị trí ta cần tạo dự án bằng lệnh *cd (địa chỉ).*
* Thực thi dòng lệnh *ng new (tên dự án)*

*Ví dụ: ng new angular-app*

**

Hình 26. Khởi tạo dự án Angular đầu tiên trong Ubuntu

* Như vậy ta vừa tạo được dự án Angular đầu tiên với tên “*angular-app*”.
  1. Mở một dự án Angular đã tồn tại
* Để mở một dự án Angular đã tồn tại, ta thực hiện trỏ đến vị trí chứa dự án của mình. Trong ví dụ ở trên, ta vừa thực hiện tạo dự án tại thư mục “*Home*” (Thư mục User mình đang sử dụng.
* Tại đây, mình sẻ trỏ vào trong thư mục của dự án bằng cách thực thi đoạn lệnh: *cd angular-app (với angular-app chính là tên dự án)*
* Sau đó, ta cần thực thi đoạn lệnh sau để mở dự án: *ng serve --open*

**

Hình 27. Mở dự án Angular đã tồn tại

* Sau khi thực thi đoạn lệnh, dự án Angular sẽ được khởi chạy và ta có thể truy cập dự án theo địa chỉ local host: [*http://localhost:4200/*](http://localhost:4200/)

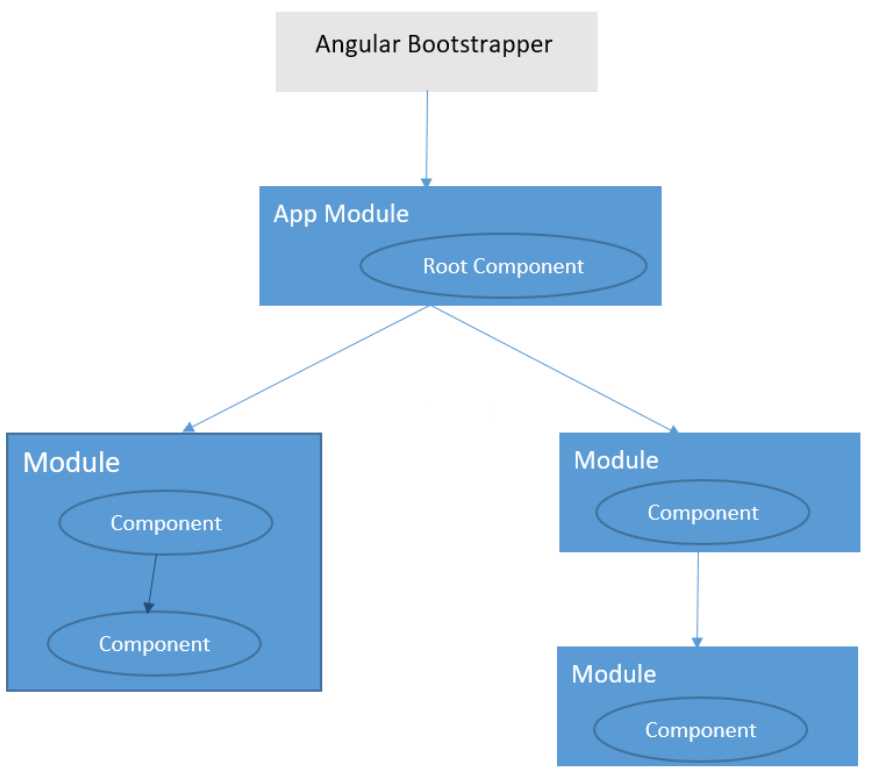


Hình 28. Giao diện mặc định của Angular

1. KIẾN TRÚC TỔNG QUÁT ANGULAR

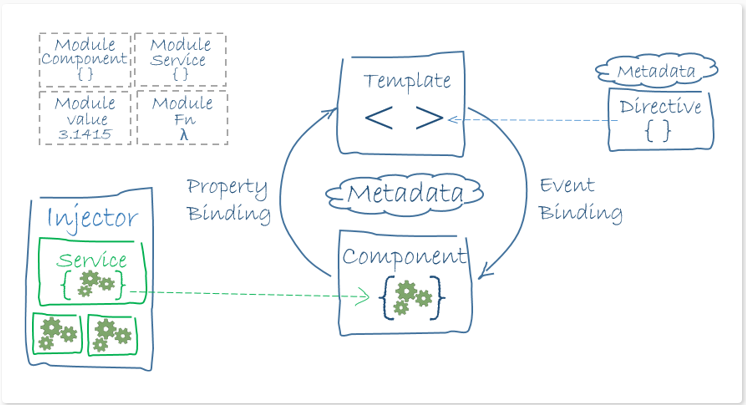
Kiến trúc mức cao của ứng dụng

* Ứng dụng được xây dựng và phát triển trên nền tảng Angular 4, 5, 6 và 7 có thể được mô tả theo sơ đồ cây tổng quát sau:



Hình 29. Kiến trúc Angular

* Components: Được dịch là thành phần, nó chức đựng các chức năng tương tác phía giao diện với người dùng. Component có chứa Html, css hoặc scss, và các đoạn mã được viết bằng ngôn ngữ Typescrip nhằm thực hiện gắn kết (Tương tác) giữa giao diện ứng dụng và các dịch vụ (service) ở phía Server.
* Trong hình vẽ trên thì các component trực thuộc Module. Và nó được nạp vào ứng dụng là nhờ Module.
* Component sử dụng Dịch vụ (Service) để lấy dữ liệu hoặc thao tác với dữ liệu và hiển thị chúng thông qua các đoạn mã chương trình được viết bằng TypeScript.
* Component có thể sử dụng các đối tượng là thể hiện của dịch vụ thông qua cơ chế Dependency Injection (Thể hiện của Dependency Injecttion - DI là rất dễ hiểu, lập trình viên chỉ cần khai báo đối tượng thông qua hàm khởi tạo của lớp).
* Directive được sử dụng trong template của Component nhằm để mở rộng các hành vi của Html.
* Dữ liệu có thể được hiển thị tương ứng với các file html trong component. Angular cung cấp cơ chế gắn kết dữ dữ liệu với giao diện (template) của component cơ chế đó được gọi là databinding.
* Thông qua cơ chế Databinding, bất kỳ thay đổi dữ liệu trong Angular, thì giao điện tương tác sẽ tự động cập nhật các thay đổi dữ liệu đó. Tương tác về sự thay đổi dữ liệu trong Angular với phía giao diện (UI) có thể theo hướng 1 chiều hoặc 2 chiều.
* Các thành phần cơ bản của Angular được mô tả qua hình ảnh dưới đây:

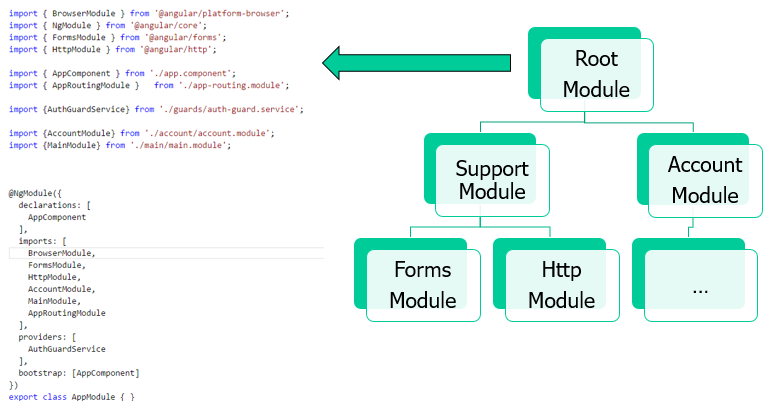


Hình 30. Các thành phần cơ bản của Angular

Các thành phần cơ bản của Angular

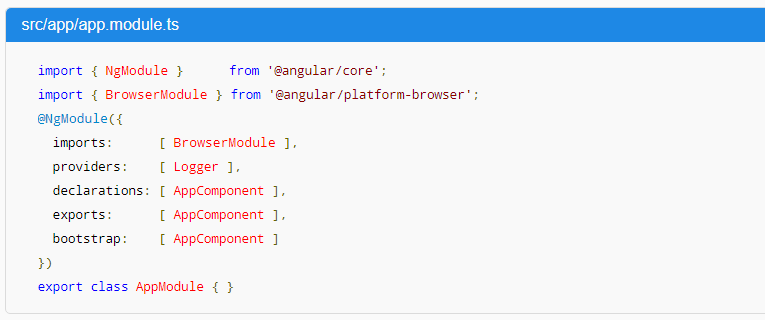
1. Module
   1. Giới thiệu về Module

* Angular chia sự phát triển ứng dụng thành các Module khác nhau. Tất cả các ứng dụng của Angular đều có 1 module gốc, module gốc được gọi là AppModule và có ít nhất một lớp NgModule.
* Tùy theo quy mô của ứng dụng, với ứng dụng nhỏ chỉ cần một module là đủ. Tuy nhiên với các ứng dụng lớn có thể có chứa rất nhiều các Module khác nhau.
* Chú ý trong một Module có thể chứa nhiều các Module con khác. Các module con được gọi là các SubModule. Ví dụ được mô tả ở hình dưới đây.



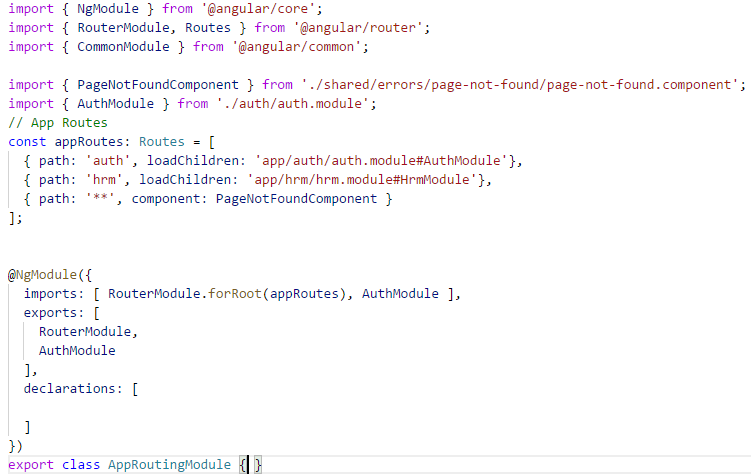
Hình 31. Module chứa các submodule

* Mỗi Module có chứa @ngModule. Với các đặc chưng có chứa Metadata nhằm để định nghĩa các thuộc tính được sử dụng như sau :
* Declarations: Khai báo các lớp View (Bao gồm Component, directive và Pipe nhằm thực hiện hiển thị thông tin do vậy người ta gọi là các lớp hiển thị (View class).
* Exports: Khi chúng ta sử dụng các Component tại Module, chúng ta muốn các Component này được sử dụng tiếp ở các Module khác thì chúng ta sẽ khai báo các Component này tại mảng đối tượng trực thuộc từ khóa Export này.
* Imports: Chúng tả muốn sử dụng các Module khác trong module này, thì chúng ta phải khai báo chúng tại mảng đối tượng trực thuộc từ khóa Imports.
* Providers : Khai báo các service được sử dụng trong module.
* Hình sau mô tả các thành phần của Module:



Hình 32. Mô tả các thành phần thuộc tính của module

* 1. Cách phân chia ứng dụng theo Module
* Khi xây dựng ứng dụng với quy mô từ trung bình trở lên, người ta thường hay sử dụng ngôn ngữ mô hình hóa ví dụ như UML để thực hình mô hình hóa hệ thống phần mềm. Trong ngôn ngữ UML có nhiều các biểu đồ khác nhau. Để mô hình hóa chức năng của hệ thống, các kỹ sư sẽ sử dụng biểu đồ Usecase để mô hình hóa 1 chức năng hoặc một nhóm chức năng của phần mềm, Khi muốn gom nhóm các chức năng lại với nhau thì người ta sẽ sử dụng biểu đồ biểu đồ thành phần (Component Diagram – Chứa nhiều biểu đồ Usecase) .
* Vởi các ứng dụng được viết bằng Angular 5 chúng ta có thể ánh xạ biểu đồ thành phần sang thành các Module ứng dụng. Với các SubModule tương ứng với các biểu đồ Usecase trực thuộc biểu đồ thành phần.
* Ví dụ ta xây dựng 1 chương trình quản lý nhân sự, chương trình này bao gồm có chứa ít nhất 2 biểu đồ thành phần (1 biểu đố Authortication – Dùng để kiểm tra truy cập hệ thống và 1 biểu đồ thành phần sử dụng để Quản lý nhân sự - Hrm). Tương ứng sang Angular ta cũng sẽ có 3 modules đó là: Application Module (Đóng vai trò là RootModule) và 2 module thành phần bao gồm AuthorModule và HrmModule.
* Hình dưới đây mô tả Module gốc của ứng dụng:



Hình 33. Module gốc của ứng dụng

* Hình mô tả Module xác thực thông tin (AuthModule) của ứng dụng:

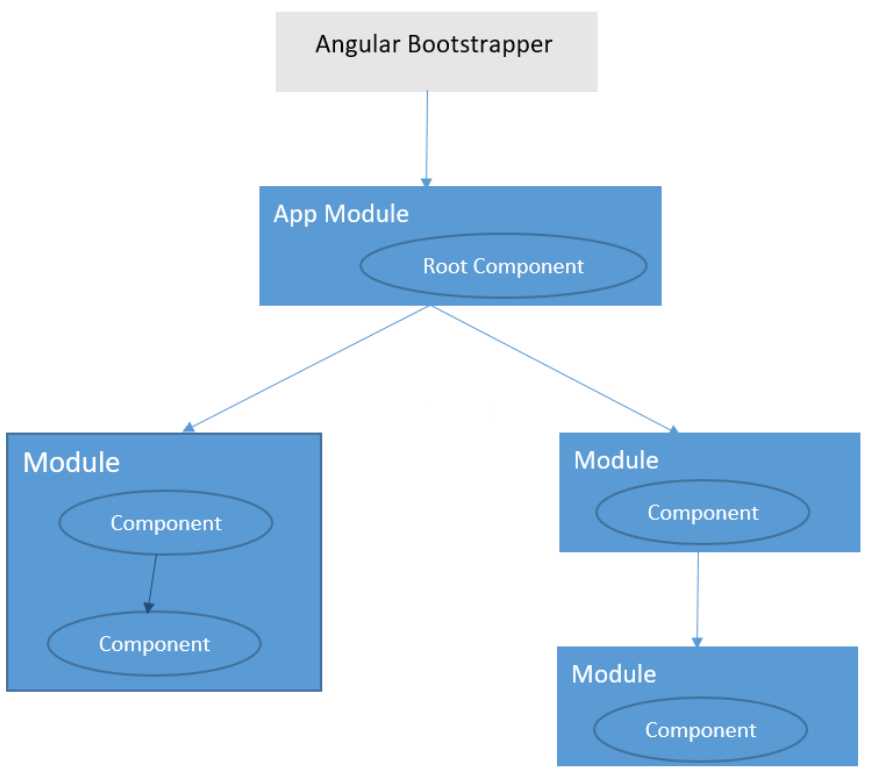


Hình 34. Module Auth ứng dụng

* Chúng ta dễ dàng nhận thấy trong hình trên là hình ảnh của Module có chứa các router nhằm định tuyến các đường dẫn của ứng dụng. Trong hình trên AuthorModule chứa các Component như : LoginComponent, ForgotPasswordComponent … Đó chính là các thành phần tương ứng với các Usecase như Login Usecase, ForgotPassword Usecase.

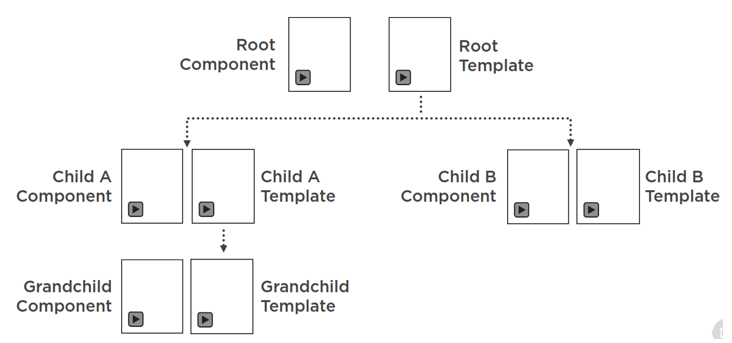
1. Component
   1. Tổng quát về Component

* Component là một thành phần không thể thiếu của ứng dụng được xây dựng bằng Angular nói chung và Angular 5 nói riêng. Một ứng dụng được viết bằng Angular chứa nhiều các Component khác nhau. Mỗi component thực thi một nhiệm vụ nào đó của ứng dụng.
* Với các ứng dụng lớn, các nhà thiết kế kiến trúc thường chia ứng dụng ra thành nhiều các Module khác nhau. Mỗi module khi đó sẽ chứa nhiều các component. Chúng ta xem lại kiến trúc Angular:



Hình 35. Kiến trúc Angular

* Ứng dụng Angular khi khởi tạo, nó sẽ tự động gọi Component gốc (root component – app.component), sau đó ứng dụng sẽ nạp các component khác có liên quan.
* Mô tả tổng quát theo sơ đồ hình cây như sau:

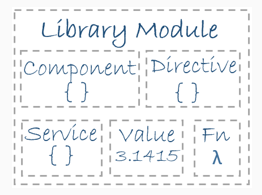


Hình 36. Kiến trúc hình cây của các Component

* Component có thể liên quan tới bất kỳ một phần tử hoặc một nội dung nào đó của một trang ví dụ quá trình sinh ra nội dung của biểu đồ khi hiển thị chi tiết một phần tử nào đó hay khi lập trình muốn phân tách nội dung của một trang ra thành nhiều các phần nhỏ khác nhau, khi đó mỗi phần nhỏ có thể là một component.
* Về cơ bản thì Component trong ứng dụng Angular được sử dụng để tạo ra giao diện từ tập hợp các HTML hoặc được sử dụng để tạo ra Control mới với một nghiệp vụ cụ thể của bài toán, chính vì vậy khi sử dụng Component lập trình viên thường kết hợp với Bootstrap hoặc với Angular Marterial … để tạo giao diện tương tác với người sử dụng.
  1. Tạo Component
* Lập trình viên thường hay sử dụng Angular Cli để tạo Component. Cú pháp tạo component như sau: ***ng g component my-new-component***
* Để hiểu chi tiết hơn về tạo Component thông qua Angular Cli chúng ta có thể vào url sau: ***https://github.com/angular/angular-cli***

1. Thư viện Angular

* Angular coi như là một con tầu chứ một tập hợp các Module Javascript, mỗi tên thư viện Angular được sử dụng với tiền tố *@angular*.



Hình 37. Thư viện Angular

* Khi chúng ta sử dụng các component, đôi khi chúng ta sẽ gặp lỗi khi thực hiện biên dịch đặc biệt khi chúng ta sử dụng với component với Angular marterial, lý do là chúng ta không import các thư viện cần thiết cho ứng dụng hoặc cho Module chứa component này.
* Để sử dụng các thư viện này ở mức độ Module chúng ta import nó vào trong *@NgModule.*

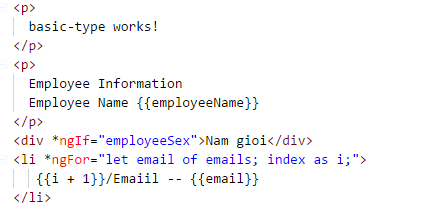


Hình 38. Sử dụng thư viện mức Module

1. Structure Directive

* Được sử dụng phổ biến trong Angular 5. Mục tiêu của Structure Directive là dùng để thay đổi cấu trúc DOM bằng việc thêm bớt các phần tử trong DOM. Structure Directive là các lớp được sử dụng với chỉ chỉ dẫn Directive. Khi sử dụng Structure directive, chúng đi liền với \* ở đầu.

*Ví dụ:* Directive NgForOf dùng để hiển thị danh sách các phần tử trong tập hợp dữ liệu.



Hình 39. Directive NgFor

* Ngoài ra Angular chứa một loạt các Directive khác gắn với các điều kiện để kiểm tra tính logic khi hiển thị và thao tác dữ liệu như NgIf, NgSwitch.

1. Data Binding

* Người sử dụng tương tác với chương trình với mục tiêu nhằm để tra cứu dữ liệu, chỉnh sửa dữ liệu và xóa dữ liệu. Để thực hiện các chức năng trên, người dùng sẽ phải thực hiện thông qua giao diện của chương trình.
* Với các lập trình viên có kinh nghiệm khi lập trình ứng dụng theo mô hình MVVM (Model, View và ViewModel) thì quá trình gắn kết xử lý dữ liệu của View và Model là thường xuyên trong quá trình phát triển ứng dụng.
* Với Angular, mô hình phát triển ứng dụng cũng chính là MVVM, trong đó Model chính là các cấu trúc dữ liệu (Thường là các Interface) và View chính là giao diện của ứng dụng. Quá trình gắn kết giữa dữ liệu với giao diện được gọi là Data binding.
* Vậy Data Binding trong Angular chính là quá trình gắn kết dữ liệu giữa Component và Template. Data Biding cung cấp quá trình liên kết giữa các thành phần HTML của Template và dữ liệu.



Hình 40. Data binding trong Angular

* Trong hình trên thể hiện Angular có 4 cách để thể hiện Data binding.
* **Interpolation** : Sử dụng cặp {{}} để thực hiện render ra giá trị khi thực hiện gắn kết giữa Data và template của component. Cách gắn kết này được thực hiện theo cơ chế một chiều (Chỉ đọc thông tin), nó tiện lợi khi lập trình viên muốn hiển thị dữ liệu đơn giản.
* **Property Biding**: Cơ chế gắn kết thông qua thuộc tính. Cách thể hiện của đặc tính này là thông qua []. Đây cũng là cơ chế gắn kết 1 chiều.
* **Even Binding**: Cơ chế gắn kết sự kiện. Các gắn kết này cho phép lập trình viên viết các phương thức trực thuộc Component sau đó gán cho các sự kiện của các phần tử ở phía template của component đó.
* **Gắn kết 2 chiều** [( )] : thực hiện gửi giá trị từ Component cho Template và gửi những giá trị thay đổi của Tempate lại cho Component. Điều đó có nghĩa là những giá trị thay đổi trên Template sẽ tự động được thay đổi trên Component.

*Ví dụ:* *Thực hiện gắn kết dữ liệu.*



Hình 41. Code của Component chứa dữ liệu

* Phần giao diện Template thực hiện cơ chế Binding 2 chiều [()], có nghĩa là khi người sử dụng thay đổi giá trị nhập vào 2 ô, khi đó giá trị NumberOne và NumberTwo sẽ tự động cập nhật.

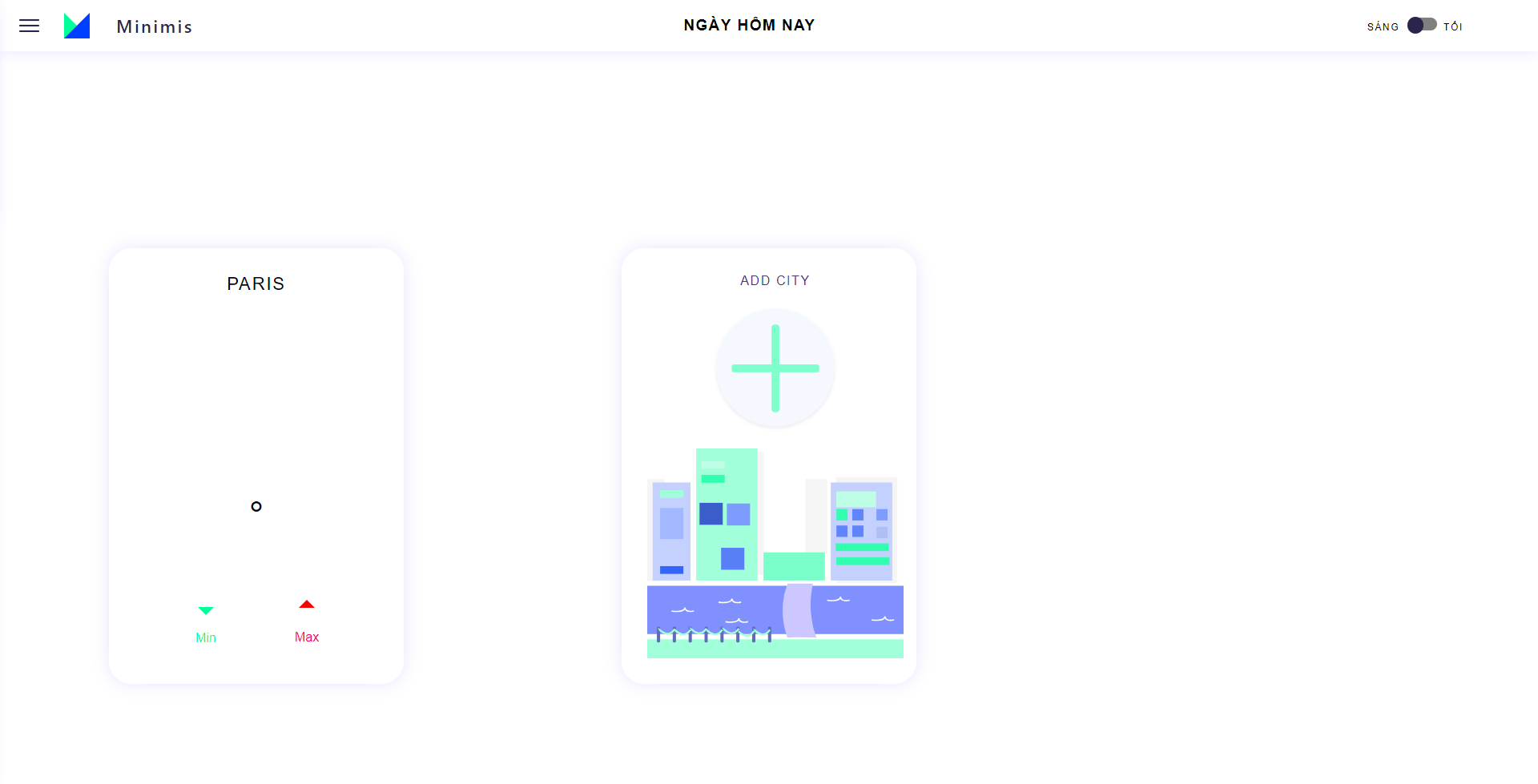


Hình 42. Giao diện ứng dụng

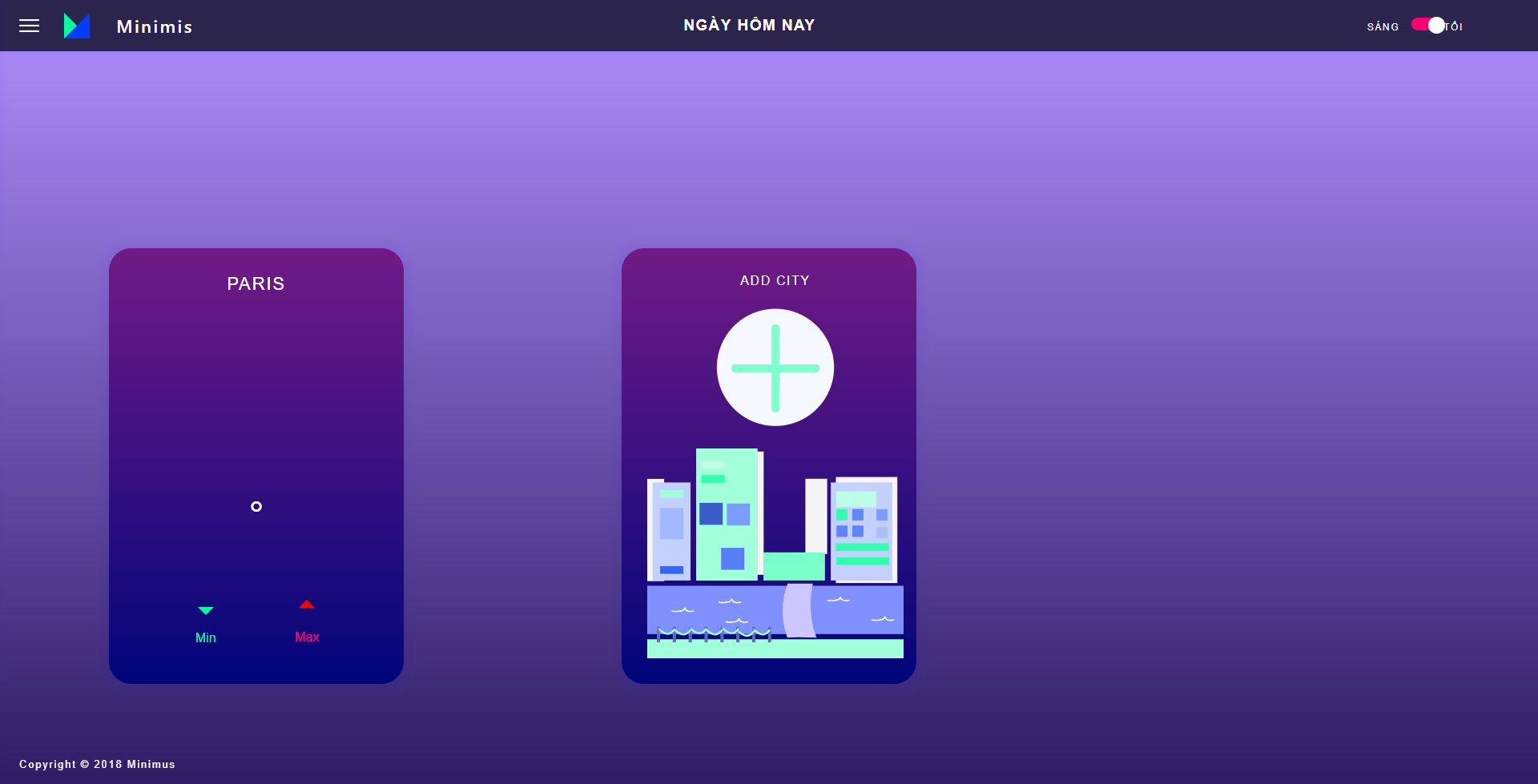
* Khi người dùng Click vào nút Submit khi đó hàm Add sẽ được gọi và totalNumber sẽ hiển thị thông qua cơ chế binding Interpolation.

1. **Demo Front-End với Angular**

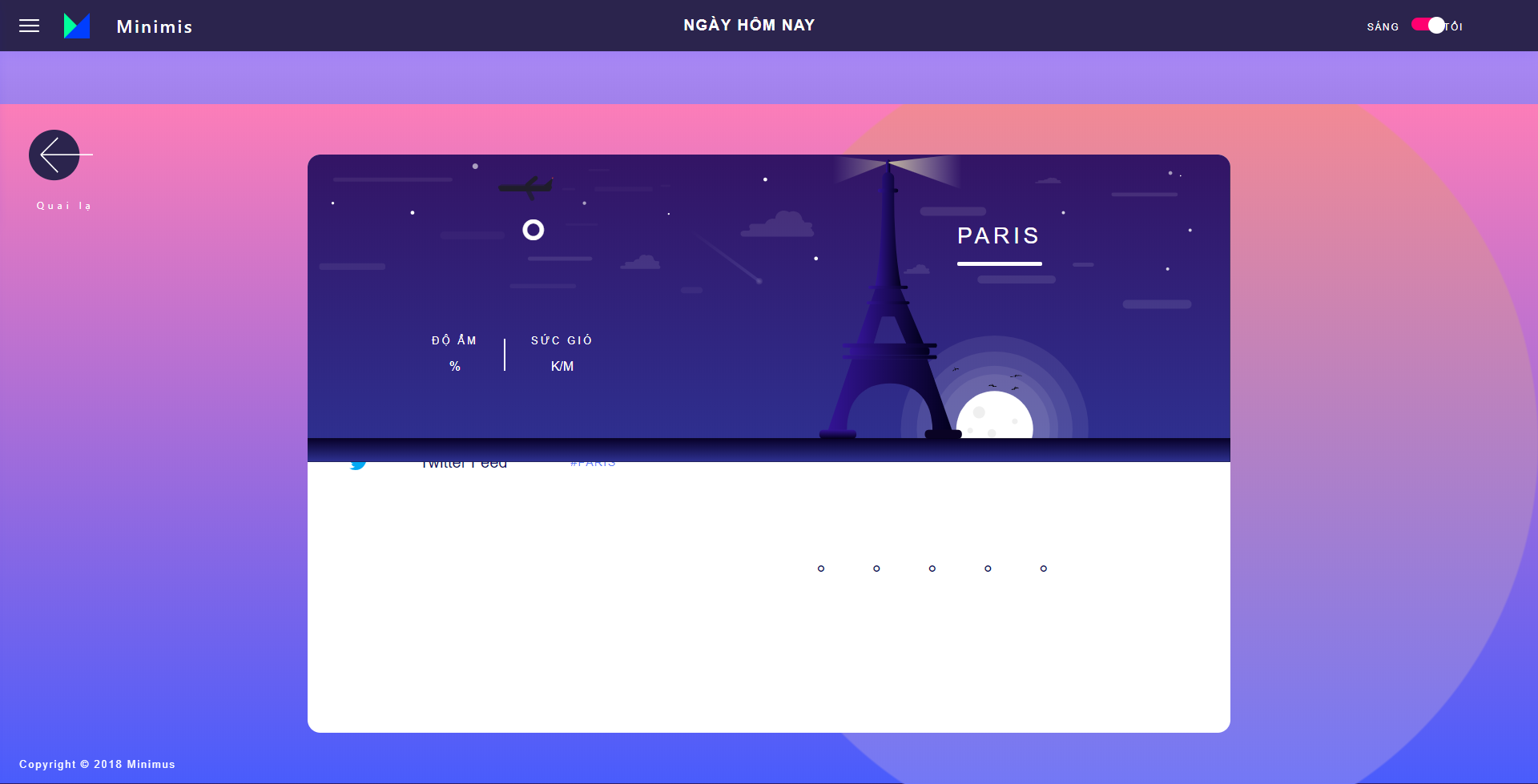
* Nhóm demo front-end website thời tiết với Angular từ hướng dẫn cụ thể thông qua từng component và module. Với giao diện còn đơn giản, có thể chỉnh chế độ ban ngày hoặc ban đêm.



Hình 43. Web app thời tiết với Angular (giao diện sáng)



Hình 44. Web app thời tiết với Angular (giao diện tối)



Hình 45. Chi tiết thời tiết theo khu vực

CHƯƠNG 3 KẾT LUẬT VÀ ĐỊNH HƯỚNG PHÁT TRIỂN

1. **Kết quả đạt được**

* Sau một thời gian tìm hiểu về đề tài, nhóm chúng em đã học thêm nhiều điều mới và bổ ích về hệ điều hành Ubuntu và Angular framework.
* ***Ubuntu:***
* Cài đặt được Ubuntu lên máy tính cá nhân.
* Làm quen với Terminal, các lệnh cơ bản.
* Cơ bản thành thạo với Ubuntu.
* Cài đặt được môi trường, IDE, phần mềm thông qua dòng lệnh.
* Hiểu và xử lý được các lỗi thường gặp trong quá trình sử dụng cũng như lập trình.
* ***Angular:***
* Cài đặt được Angular Framework lên hệ điều hành Ubuntu.
* Nâng cấp hoặc hạ cấp Angular theo đúng phiên bản mong muốn.
* Tạo và khởi động project Angular với Terminal.
* Hiểu về cách thức hoạt động của Angular với từng component,
* Nhúng được Boostrap hoặc code html thông thường vào Angular.
* Demo được Front-end website cơ bản.

1. **Hướng phát triển đề tài**

* Tuy nhiên, thời gian tìm hiểu còn ít nên nhóm chưa đạt được những kết quả như mong đợi. Thời gian tới, nhóm sẽ cố gắng tìm hiều thêm về Angular và phát triển đề tài theo hướng:
* Lập trình được Website sử dụng Angular Framework.
* Thành thạo TypeScript.
* Kết hợp được Boostrap và Firebase hoặc các dịch vụ Cloud Computing.
* Quản lý dự án với Github.
* Làm việc nhóm chuyên nghiệp hơn với Trello và Slack.

TÀI LIỆU THAM KHẢO

**Tiếng Việt:**

1. ***Khoá học Angular 4 Khoa Phạm***

*https://www.youtube.com/watch?v=q8RdFZw-gjo&list=PLzrVYRai0riTA1m7Dasg8eraBr6R9nFgC*

1. ***Khoá học Angular 5***

[*https://viblo.asia/p/thu-hoc-angular-5-Eb85oJzjl2G*](https://viblo.asia/p/thu-hoc-angular-5-Eb85oJzjl2G)

**Tiếng Anh:**

1. ***Ubuntu Tutorial***

[*https://tutorials.ubuntu.com/*](https://tutorials.ubuntu.com/)

1. ***Linux Mint Tutorial***

[*https://community.linuxmint.com/tutorial/*](https://community.linuxmint.com/tutorial/)

1. ***Angular Tutorial***

[*https://angular.io/tutorial*](https://angular.io/tutorial)

1. ***Angular CLI***

[*https://cli.angular.io/*](https://cli.angular.io/)

[*https://github.com/angular/angular-cli*](https://github.com/angular/angular-cli)

1. ***Build A Real World Beautiful Web APP with Angular 7 — A to Z Ultimate Guide (2018)***

[*https://medium.com/@hamedbaatour/build-a-real-world-beautiful-web-app-with-angular-6-a-to-z-ultimate-guide-2018-part-i-e121dd1d55e*](https://medium.com/@hamedbaatour/build-a-real-world-beautiful-web-app-with-angular-6-a-to-z-ultimate-guide-2018-part-i-e121dd1d55e)

**Phần mềm được sử dụng:**

***Microsoft Visual Code***

***Open Operation System: Ubuntu, Linux Mint***

***Microsoft Office 2019***

TÀI LIỆU THAM KHẢO

**Tiếng Việt**

1. Bryan Syverson, Joel Murach, *SQL Server hướng dẫn học qua ví dụ*, NXB Khoa học và Kỹ thuật, 2013.
2. Khoá học Angular 4 Khoa Phạm

*https://www.youtube.com/watch?v=q8RdFZw-gjo&list=PLzrVYRai0riTA1m7Dasg8eraBr6R9nFgC*

1. Khoá học Angular 5

[*https://viblo.asia/p/thu-hoc-angular-5-Eb85oJzjl2G*](https://viblo.asia/p/thu-hoc-angular-5-Eb85oJzjl2G)

**Tiếng Anh**

1. Floyd Sally, Jacobson Van (1993), *Random Early Detection gateways for Congestion Avoidance*, IEEE/ACM Transactions on Networking.
2. Ubuntu Tutorial

[*https://tutorials.ubuntu.com/*](https://tutorials.ubuntu.com/)

1. Linux Mint Tutorial

[*https://community.linuxmint.com/tutorial/*](https://community.linuxmint.com/tutorial/)

1. Angular Tutorial

[*https://angular.io/tutorial*](https://angular.io/tutorial)

1. Angular CLI

[*https://cli.angular.io/*](https://cli.angular.io/)

[*https://github.com/angular/angular-cli*](https://github.com/angular/angular-cli)

1. Build A Real World Beautiful Web APP with Angular 7 — A to Z Ultimate Guide (2018)

[*https://medium.com/@hamedbaatour/build-a-real-world-beautiful-web-app-with-angular-6-a-to-z-ultimate-guide-2018-part-i-e121dd1d55e*](https://medium.com/@hamedbaatour/build-a-real-world-beautiful-web-app-with-angular-6-a-to-z-ultimate-guide-2018-part-i-e121dd1d55e)

**Phần mềm được sử dụng:**

Microsoft Visual Code

Open Operation System: Ubuntu, Linux Mint

Microsoft Office 2019