**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

🙢 ❖ 🙠



**KHOÁ LUẬN TỐT NGHIỆP**

**IOT VÀ ỨNG DỤNG**

**(SMART HOME)**

Sinh viên thực hiện : 14127871 Phan Bảo Trinh

14026751 Vương Quốc Việt

Giáo viên hướng dẫn : ThS. Nguyễn Thành Thái

Lớp : ĐHCNTT10B

***TP. HỒ CHÍ MINH, THÁNG 05 NĂM 2018***

# LỜI NÓI ĐẦU

Lập trình máy tính là việc lập ra chương trình làm việc cho máy có bộ xử lý, để thực thi nhiệm vụ xử lý thông tin nào đó. Để viết được một chương trình người lập trình cần phải biết một ngôn ngữ lập trình nào đó. Ngôn ngữ lập trình là một tập con của ngôn ngữ máy tính, được thiết kế và chuẩn hóa để truyền các chỉ thị cho các máy có bộ xử lý. Ngôn ngữ lập trình dùng để tạo ra các chương trình máy nhằm mục đích điều khiển máy tính hoặc mô tả thuât toán để người khác có thể hiểu được.  
 Hiện nay có rất nhiều ngôn ngữ lập trình từ bậc thấp đến bậc cao hỗ trợ người lập trình một cách dễ dàng. Tuy nhiên tất cả đều cần có nền tảng về kỹ thuật lập trình, cấu trúc dữ liệu và giải thuật. Đây là những môn nền tảng được dạy trong các trường đại học, cao đẳng và cả trung học phổ thông. Đa số các môn học này đều tổ chức thi theo hình thức tự luận hoặc thực hành trên máy và giáo viên đi chấm bài theo từng máy.  
 Việc chấm bài thi lập trình mất rất nhiều thời gian và đôi khi còn sai sót. Nên nhóm em đã chọn đề tài xây dựng ứng dụng chấm bài lập trình để rút ngắn thời gian chấm và độ chính xác cao hơn. Phạm vi của đề tài là tạo môi trường trình biên dịch và chạy các đoạn mã của ngôn ngữ lập trình C/C++. Hướng phát triển sau này là tạo nên một hệ thống chấm bài được sử dụng rộng rãi trong môi trường dạy và học lập trình của các trường đại học, cao đẳng.

Ứng dụng được phát triển bằng ngôn ngữ Python sử dụng framework Flask để lập trình giao diện và các ngôn ngữ lập trình web như HTML, CSS, Ninja,…Sử dụng và áp dụng các dịch vụ điện toán đám mây của amazon để triển khai ứng dụng. Quá trình xây dựng và phát triển ứng dụng đã chạy ổn định, môi trường chạy thử các đoạn mã cho kết quả chính xác.

Trong quá trình xây dựng ứng dụng nhóm đã gặp rất nhiều khó khăn về việc sử dụng công nghệ mới. Nhưng nhờ sự giúp đỡ và hỗ trợ tận tình của thầy Đặng Thanh Bình và thầy Nguyễn Văn Quang nhóm em đã phát triển và hoàn thiện đề tài này. Nhóm em xin chân thành cảm ơn sự giúp đỡ tận tình của hai thầy.

# DANH MỤC TỪ VIẾT TẮT

ACM Association for Computing Machinery

AWS Amazon Web Service

CMS Content Management System

CSS Cascading Style Sheets

EC2 [Amazon Elastic Compute Cloud](https://aws.amazon.com/articles/Amazon-EC2?browse=1)

HTML Hypertext Markup Language

HTTP HyperText Transfer Protocol

HTTPS HyperText Transfer Protocol Secure

IAM Identity And Access Management

ICPC International Collegiate Programming Contest

ID Identification

IP Internet Protocol

SSD Solid State Drive

SSH Secure Socket Shell

# DANH MỤC HÌNH ẢNH

[Hình 1.1. Hệ thống chấm bài Themis. 1](#_Toc496047823)

[Hình 2.1. Use case tổng quát. 4](#_Toc496047824)

[Hình 2.2. Giao diện Mockup – Home. 14](#_Toc496047825)

[Hình 2.3. Giao diện Mockup – Customtest. 14](#_Toc496047826)

[Hình 2.4. Sơ đồ hoạt động của môi trường chấm bài. 15](#_Toc496047827)

[Hình 3.1. Cửa sổ dịch vụ của AWS. 17](#_Toc496047828)

[Hình 3.2. Thêm người dùng. 17](#_Toc496047829)

[Hình 3.3. Điền thông tin người dùng. 17](#_Toc496047830)

[Hình 3.4. Cấp quyền người dùng. 18](#_Toc496047831)

[Hình 3.5. Thông tin người dùng đã đăng ký. 18](#_Toc496047832)

[Hình 3.6. Cửa sổ dịch vụ của AWS. 19](#_Toc496047833)

[Hình 3.7. Bảng điều khiển EC2. 19](#_Toc496047834)

[Hình 3.8. Chọn hệ điều hành cho máy chủ EC2. 20](#_Toc496047835)

[Hình 3.9. Lựa chọn cấu hình máy chủ EC2. 20](#_Toc496047836)

[Hình 3.10. Lựa chọn cấu hình chi tiết. 20](#_Toc496047837)

[Hình 3.11. Thêm dung lương bộ nhớ. 21](#_Toc496047838)

[Hình 3.12. Gắn nhãn máy chủ. 21](#_Toc496047839)

[Hình 3.13. Cấu hình bảo mật. 21](#_Toc496047840)

[Hình 3.14. Xem lại thông tin máy chủ. 22](#_Toc496047841)

[Hình 3.15. Lựa chọn kerpair cho máy chủ. 22](#_Toc496047842)

[Hình 3.16. Thông báo tạo máy chủ thành công. 22](#_Toc496047843)

[Hình 3.17. Máy chủ EC2 đã tạo. 23](#_Toc496047844)

[Hình 3.18. Thông tin chi tiết của máy chủ EC2. 23](#_Toc496047845)

[Hình 3.19. Phần mềm putty và puttygen. 23](#_Toc496047846)

[Hình 3.20. Tạo khóa truy cập EC2 với Puttygen. 24](#_Toc496047847)

[Hình 3.21. Phần mềm Putty. 24](#_Toc496047848)

[Hình 3.22. Giao diện dòng lệnh làm việc với EC2 Instance. 25](#_Toc496047849)

[Hình 3.23. Bảng cấu hình AWS Command Line Interface. 27](#_Toc496047850)

[Hình 3.24. Cấu hình IP và port . 27](#_Toc496047851)

[Hình 3.25. Chương trình ở chế độ debug. 28](#_Toc496047852)

[Hình 3.26. Giao diện trang chủ. 28](#_Toc496047853)

[Hình 3.27. Giao diện đăng nhập. 29](#_Toc496047854)

[Hình 3.28. Đăng ký thành công tài khoản test01. 29](#_Toc496047855)

[Hình 3.29. Kết quả đoạn mã 1. 30](#_Toc496047856)

[Hình 3.30. Kết quả đoạn mã 2. 31](#_Toc496047857)

[Hình 3.31. Kết quả đoạn mã 3. 31](#_Toc496047858)

# DANH MỤC BẢNG BIỂU

[Bảng 2.1. Use case Submit. 5](#_Toc496047859)

[Bảng 2.2. Use case View Result (Student). 6](#_Toc496047860)

[Bảng 2.3. Use case View ScoreBoard. 7](#_Toc496047861)

[Bảng 2.4. Use case Login. 7](#_Toc496047862)

[Bảng 2.5. Use case Logout. 8](#_Toc496047863)

[Bảng 2.6. Use case Create Contest. 9](#_Toc496047864)

[Bảng 2.7. Use case Create Problem. 10](#_Toc496047865)

[Bảng 2.8. Use case View Result (Teacher). 11](#_Toc496047866)

[Bảng 2.9. Use case Get Result. 12](#_Toc496047867)

[Bảng 2.10. Use case View List Problem. 12](#_Toc496047868)

[Bảng 2.11. Use case Load Contest. 13](#_Toc496047869)

[Bảng 3.1. Các đoạn mã C/C++ mẫu. 30](#_Toc496047870)

# MỤC LỤC

[LỜI NÓI ĐẦU i](#_Toc8425460)

[DANH MỤC TỪ VIẾT TẮT ii](#_Toc8425461)

[DANH MỤC HÌNH ẢNH iii](#_Toc8425462)

[DANH MỤC BẢNG BIỂU v](#_Toc8425463)

[MỤC LỤC vi](#_Toc8425464)

[Chương 1: Tổng quan 1](#_Toc8425465)

[1.1. Internet of Thing. 1](#_Toc8425466)

[1.1.1. Giới thiệu. 1](#_Toc8425467)

[1.1.2. Mô hình. 1](#_Toc8425468)

[1.1.3. Ứng dụng. 2](#_Toc8425469)

[1.1.4. IoT và Cách Mạng Công Nghiệp 4.0. 2](#_Toc8425470)

[1.2. Smart Home. 3](#_Toc8425471)

[1.2.1. Giới thiệu. 3](#_Toc8425472)

[1.2.2. Mô tả đề tài. 3](#_Toc8425473)

[1.2.3. Công nghệ, thiết bị sử dụng. 4](#_Toc8425474)

[1.2.4. Mục tiêu, phạm vi đề tài. 4](#_Toc8425475)

[Chương 2: THIẾT BỊ VÀ GIAO THỨC 6](#_Toc8425476)

[2.1. Phân tích hệ thống. 6](#_Toc8425477)

[2.1.1. Thiết kế use case. 6](#_Toc8425478)

[2.1.2. Yêu cầu chức năng. 7](#_Toc8425479)

[2.1.3. Yêu cầu phi chức năng. 7](#_Toc8425480)

[2.1.4. Đặc tả use case. 7](#_Toc8425481)

[2.2. Thiết kế giao diện. 17](#_Toc8425482)

[2.2.1. Trang chủ. 17](#_Toc8425483)

[2.2.2. Customtest. 17](#_Toc8425484)

[2.3. Hoạt động của ứng dụng. 18](#_Toc8425485)

[Chương 3: TRIỂN KHAI ỨNG DỤNG TRÊN AMAZON WEB SERVICES 19](#_Toc8425486)

[3.1. Amazon Web Services (AWS). 19](#_Toc8425487)

[3.1.1. Giới thiệu. 19](#_Toc8425488)

[3.1.2. Các dịch vụ sử dụng. 19](#_Toc8425489)

[3.2. Triển khai trên EC2. 19](#_Toc8425490)

[3.2.1. Tạo tài khoản AWS. 19](#_Toc8425491)

[3.2.2. Tạo người dùng với IAM. 20](#_Toc8425492)

[3.2.3. Tạo máy chủ EC2. 21](#_Toc8425493)

[3.2.4. Kết nối EC2. 26](#_Toc8425494)

[3.2.5. Cài đặt môi trường. 28](#_Toc8425495)

[3.3. Đưa ứng dụng lên EC2. 29](#_Toc8425496)

[3.4. Kết nối với database (DynamoDB): 30](#_Toc8425497)

[3.5. Cấu hình ứng dụng. 30](#_Toc8425498)

[3.6. Kiểm thử. 30](#_Toc8425499)

[KẾT LUẬN ix](#_Toc8425500)

[TÀI LIỆU THAM KHẢO x](#_Toc8425501)

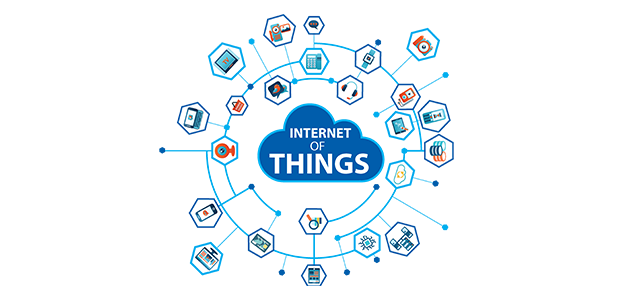
# Tổng quan

Giới thiệu về Internet of Thing và Smart Home. Mô tả đề tài công nghệ sử dung và mục tiêu, phạm vi thực hiện đề tài.

## Internet of Thing.

### Giới thiệu.

Internet of Things (IoT) là thuật ngữ dùng để chỉ các đối tượng có thể được nhận biết cũng như sự tồn tại của chúng trong một kiến trúc mang tính kết nối. Đây là một viễn cảnh trong đó mọi vật, mọi con vật hoặc con người được cung cấp các định danh và khả năng tự động truyền tải dữ liệu qua một mạng lưới mà không cần sự tương tác giữa con người với con người hoặc con người với máy tính. IoT tiến hoá từ sự hội tụ của các công nghệ không dây, hệ thống vi cơ điện tử (MEMS) và Internet. Là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet và với thế giới bên ngoài để thực hiện một công việc nào đó. Cụm từ IoT được đưa ra bởi Kevin Ashton vào năm 1999. Một nhà khoa học đã sáng lập ra Trung tâm Auto-ID ở đại học MIT.



IoT đã phát triển do sự hội tụ của nhiều công nghệ, phân tích thời gian thực, máy học, cảm biến hang hoá và hệ thống nhúng.

### Mô hình.

Mô hình cơ bản của IoT gồm 3 phần:

1. Cảm biến và thiết bị truyền động: có nhiệm vụ đọc giá trị từ các cảm biến như âm thanh, ánh sang, nhiệt độ, … và chuyển thành tín hiệu điện để giúp cho các thiết bị hiểu và đưa ra những hành động hợp lý.
2. Kết nối: các tín hiệu đọc được sẽ được truyền tải lên mạng lưới thông qua các phương thức giao tiếp khác nhau như Wifi, Bluetooth, ZipBee, Lora, …
3. Con người và quy trình: các đầu vào của mạng lưới IoT sẽ được tổng hợp thành một hệ thống bao gồm dữ liệu, con người và các quy trình với mục đính đưa ra quyết định tốt hơn.

Kết quả sau cùng sẽ được hiển thị trên trình duyệt web hoặc thiết bị di động của con người.

### Ứng dụng.

Trong thực tế, IoT có thể ứng dụng trong nhiều lĩnh vực khác nhau, đem lại hiệu quả cao cho các lĩnh vực ứng dụng. Một số ứng dụng phổ biến của IoT:

* Smart Home – Ngôi nhà thông minh: là ứng dụng phổ biển và là chủ đề IoT được tìm kiếm nhiều nhất trên Google. Là ứng dụng cho phép người dùng giám sát, điều khiển ngôi nhà của mình thông qua các thiết bị di động, ngoài ra còn tự động hóa một số công việc như đóng/mở cửa, bật/tắt đèn, … Đây là một ứng dụng tiện ích, nâng cao chất lượng cuộc sống của mọi gia đình.
* Smart Cart – Chiếc xe thông minh: với khả năng tự nhận biết vị trí, tốc độ và chướng ngại vật để có thể tự lái và cảnh báo cho người lái xe.
* Smart City – Thành phố thông minh: tập hợp gồm nhiều thiết bị IoT, với khả năng thu thập và đáp ứng mọi nhu cầu của người dân.

### IoT và Cách Mạng Công Nghiệp 4.0.

Công nghiệp 4.0 là thuật ngữ dùng để chỉ quá trình phát triển trong quản lý sản xuất và sản xuất dây chuyển, ngoài ra còn được dùng để nói đến cuộc Cách Mạng Công Nghiệp lần thứ tư. Mục tiêu của nền công nghiệp 4.0 là phát triển dây chuyển sản xuất tự động. Trong Cách Mang Công Nghiệp 4.0, con người có thể giao tiếp và giám sát thiết bị thay vì vận hành chúng.

Với sự phát triển của cuộc Cách Mạng Công Nghiệp 4.0, IoT được xem là một nhánh phát triển với việc đảm nhận nhiệm vụ kết nối mọi thiết bị lại với nhau, giúp các thiết bị có thể truyền, nhận dữ liệu thu thập từ các cảm biến, đồng thời kết nối con người với thiết bị.

## Smart Home.

### Giới thiệu.

Trong các ngôi nhà hiện đại ngày nay, số lượng trang thiết bị điện, điện tử đang không ngừng gia tăng. Tuy nhiên, do khác nhau về kiến trúc, việc điều khiển các thiết bị đôi khi bất cập. Thêm vào đó, việc điều khiển các thiết bị một cách thủ công với khoảng các địa lý lớn không dễ. Vì vậy, việc áp dụng các công nghệ điều khiển tự động nhằm giải quyết tương tác giữa môi trường và các thiết bị trong nhà một cách linh hoạt, dễ dàng là điều tất yếu, khái niệm nhà thông minh ra đời.

Nhiều công nghệ đã được áp dụng khi xây dựng nhà thông minh. Tuy nhiên, sự phức tạp nằm ở chỗ các hệ thống điều khiển phải cân bằng giữa sự phức tạp của hệ thống và tính tiện dụng cho người dùng, đặc biệt là có thể được điều khiển ở bất cứ đâu, từ trong chính ngôi nhà đó hay bất kỳ nơi nào trên thế giới thông qua điện thoại hoặc Internet.

Với IoT mỗi đồ vật, thiết bị, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính. Bên cạnh đó, IoT có thể triển khai một mạng lưới các thực thể thông minh, có khả năng tự tổ chức và hoạt động tùy theo tình huống, môi trường, đồng thời chúng cũng có thể liên lạc với nhau để trao đổi thông tin, dữ liệu. Với khả năng định danh cao, số lượng các thực thể trong hệ thống được định danh chính xác, duy nhất, đảm bảo tốt khả năng quản lý, điều khiển của hệ thống.

### Mô tả đề tài.

Với một ngôi nhà thông minh, con người sẽ quan tâm tới các thông tin về thiết bị trong nhà như nhiệt độ, độ ẩm, thiết bị chiếu sang, khí gas, … Và bên cạnh đó là điều khiển các thiết bị từ xa và tự động theo một quy tắc nào đó. Dựa vào những tiêu chí cơ bản đó, nhóm đã chọn sử dụng các thiết bị cảm biến đơn giản như nhiệt độ, độ ẩm, cảm biến khí gas, cảm biến ánh sáng để thu thập dữ liệu và thiết bị giả lập bóng đèn để điển khiển. Từ những dữ liệu đã thu thập sẽ được lưu trữ vào cơ sở dữ liệu và hiển thị và gửi cảnh báo qua mail cho người dùng theo thời gian thực. Người dùng cũng có thể theo dõi, giám sát ngôi nhả của mình thông qua web và điều khiển bật tắt đèn.

### Công nghệ, thiết bị sử dụng.

Công nghệ:

* Ngôn ngữ: Arduino, Python, Angular4, HTML, CSS, JQuery.
* Cơ sở dữ liệu: PostgresSQL.
* Web Server: Nginx.
* Giao thức: HTTP, MQTT, WebSocket.

Thiết bị: Raspberry PI, Arduino, Cảm biến.

### Mục tiêu, phạm vi đề tài.

Mục tiêu:

* Thu thập dữ liệu từ các thiết bị cảm biến.
* Lưu trữ trong cơ sở dữ liệu.
* Hiển thị thông tin cho người dùng thông qua web.
* Gửi cảnh báo khi có dữ liệu bất thường qua mail.
* Cho phép người dùng điểu khiển thiết bị.

Phạm vi đề tài: vấn đề về tài chính và thời gian thực hiện cũng như hiểu biết của nhóm nên đề tài thực hiện trong phạm vi các cảm biến đơn giản như cảm biến nhiệt độ, độ ẩm, cảm biến cường độ ánh sáng, cảm biến khí gas và thiết bị điều khiển là bóng đèn led. Độ chính xác của cảm biến là tương đối nên dữ liệu sẽ không đảm bảo tính tuyệt đối.

**Kết luận:** IoT đang là xu thế của tương lai và Smart Home được xem là một đề tài nổi bật nhất để nghiên cứu, phát triển và ứng dụng cho cuộc sống con người.

# THIẾT BỊ VÀ GIAO THỨC

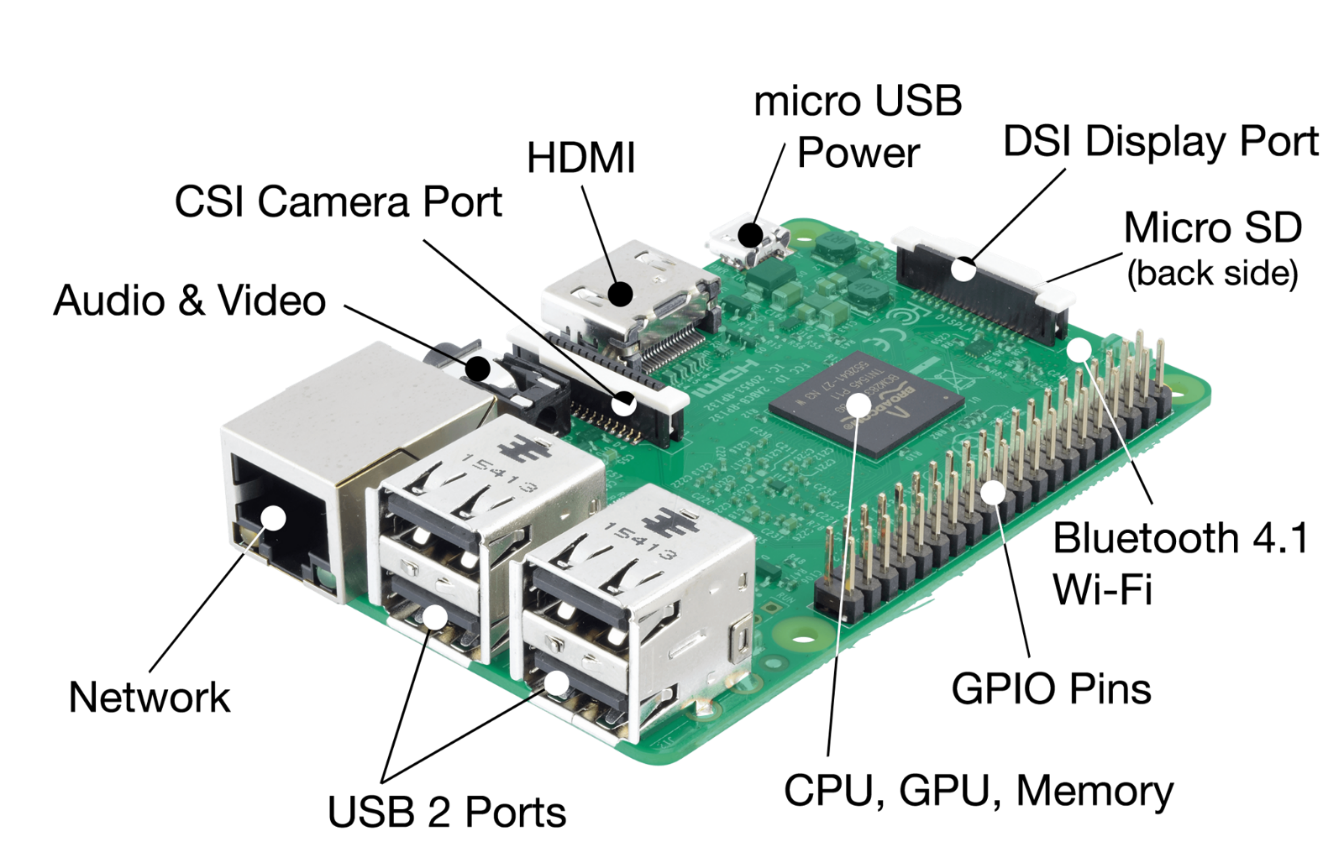
Giới thiệu cơ bản các thiết bị sử dụng: Raspberry, Arduino, cảm biến. Và giao thức kết nối giữa các thiết bị.

## Thiết bị.

### Raspberry.

Raspberry Pi là một máy tính rất nhỏ gọn, chỉ gồm có một board mạch (hay còn gọi là máy tính nhúng) kích thước chỉ bằng một thẻ tín dụng, được phát triển tại Anh với mục đích ban đầu là thúc đẩy việc giảng dạy về khoa học máy tính cơ bản trong các trường học và các nước đang phát triển.

Raspberry được sử dụng trong đề tài là Raspberry Pi 3 Model B+.



Thông số kĩ thuật:

* Vi xử lý: Broadcom BCM2837B0, quad-core A53 (ARMv8) 64-bit SoC @1.4GHz
* RAM: 1GB LPDDR2 SDRAM.
* Kết nối: 2.4GHz and 5GHz IEEE 802.11 b/g/n/ac wireless LAN, Bluetooth 4.2, BLE, Gigabit Ethernet over USB 2.0 (Tối đa 300Mbps).
* Hỗ trợ: 40-pin GPIO, 4 cổng USB2.0.
* Video và âm thanh: 1 cổng full-sized HDMI, Cổng MIPI DSI Display, cổng MIPI CSI Camera, cổng stereo output và composite video 4 chân.
* Multimedia: H.264, MPEG-4 decode (1080p30), H.264 encode (1080p30); OpenGL ES 1.1, 2.0 graphics.
* Lưu trữ: MicroSD.

Điện áp hoạt động: 5V/2.5A DC cổng microUSB, 5V DC trên chân GPIO, Power over Ethernet (PoE) (yêu cầu thêm PoE HAT).

### Arduino.

- Cung cấp giao diện đơn giản dễ sử dụng và có hướng dẫn sử dụng cho người dùng mới.

- Tương thích với mọi trình duyệt web.

- Chạy ổn định không bị treo khi nhiều người dùng sử dụng.

- Hệ thống phải đáp ứng môi trường thư viện liên tục cập nhật.

### Yêu cầu phi chức năng.

Ứng dụng không quá 8GB.

### Đặc tả use case.

Bảng 2.1. Use case Submit.

|  |  |
| --- | --- |
| **Tên Use case:** Submit | |
| **Actor:** Student | |
| **Mô tả:** Use case cho phép actor thực hiện chọn file và gửi file bài làm lên hệ thống chấm bài | |
| **Điều kiện trước:** Actor đã đăng nhập được vào hệ thống | |
| **Điều kiện sau:** Hiển thị thông tin kết quả các lần nộp bài trước đó | |
| **Luồng sự kiện chính:** | |
| **Actor** | **Hệ thống** |
| 1.Student chọn chức năng chọn file bài làm. | 2. Lấy danh sách thư mục và hiển thị giao diện cho chọn file. |
| 3. Studen vào thư mục lưu file và chọn file bài làm. | 4. Hệ thống kiểm tra file, lấy file về server và hiển thị đường dẫn của file ra giao diện |
| 5. Student nhấn chọn submit. | 6. Tiến hành compile file bài làm và chạy chương trình chấm bài, trả về kết quả và hiển thị kết quả chấm. |
| **Luồng sự kiện phụ:** | |
| 3.2. Student chọn lại file. | 3.1. Hệ thống kiểm tra file không hợp lệ, thông báo và hiển thị. |

Bảng 2.2. Use case View Result (Student).

|  |  |
| --- | --- |
| **Tên Use case:** View Result | |
| **Actor:** Student | |
| **Chức năng:** Use case cho phép actor xem thông tin kết quả các lần nộp bài. | |
| **Điều kiện trước:** Actor đã đăng nhập được vào hệ thống | |
| **Điều kiện sau:** Hiển thị thông tin kết quả các lần nộp bài. | |
| **Luồng sự kiện chính:** | |
| **Actor** | **Hệ thống** |
| 1. Student chọn chức năng View Result | 2. Lấy danh sách thông tin kết quả các lần nộp bài của Student và hiển thị. |

Bảng 2.3. Use case View ScoreBoard.

|  |  |
| --- | --- |
| **Tên Use case:** View ScoreBoard | |
| **Actor:** Student | |
| **Chức năng:** Use case cho phép actor xem scoreboard | |
| **Điều kiện trước:** Actor đã đăng nhập được vào hệ thống | |
| **Điều kiện sau:** Hiển thị thông tin scoreboard | |
| **Luồng sự kiện chính:** | |
| **Actor** | **Hệ thống** |
| 1. Student chọn chức năng View ScoreBoard | 2. Lấy danh sách kết quả của contest tại thời điểm đó và hiển thị |

Bảng 2.4. Use case Login.

|  |  |
| --- | --- |
| **Tên Use case:** Login | |
| **Actor:** Student | |
| **Chức năng:** Use case cho phép actor đăng nhập vào hệ thống | |
| **Điều kiện trước:** Actor truy cập vào web của hệ thống | |
| **Điều kiện sau:** Hiển thị thông tin contest, các chức năng View Result, View ScoreBoard và Submit, Logout | |
| **Luồng sự kiện chính:** | |
| **Actor** | **Hệ thống** |
| 1. Student dùng mã số sinh viên của mình để đăng nhập. | 2. Kiểm tra mã số sinh viên và hiển thị giao diện nộp bài. |
| **Luồng sự kiện phụ:** | |
| 1.2. Student nhập lại mã số sinh viên | 1.1. Kiểm tra mã số sinh viên không đúng thông báo và hiển thị |

Bảng 2.5. Use case Logout.

|  |  |
| --- | --- |
| **Tên Use case:** Logout | |
| **Actor:** Student | |
| **Chức năng:** Use case cho phép actor đăng xuất khỏi hệ thống | |
| **Điều kiện trước:** Actor đã đăng nhập được vào hệ thống | |
| **Điều kiện sau:** Hiển thị chức năng Login | |
| **Luồng sự kiện chính:** | |
| **Actor** | **Hệ thống** |
| 1. Student chọn chức năng Logout | 2. Hiển thị thông báo Logout thành công và cho phép Login. |

Bảng 2.6. Use case Create Contest.

|  |  |
| --- | --- |
| **Tên Use case:** Create Contest | |
| **Actor:** Teacher | |
| **Chức năng:** Use case cho phép actor tạo contest | |
| **Điều kiện trước:** Actor đã khởi động ứng dụng | |
| **Điều kiện sau:** Hiển thị contest đã được tạo | |
| **Luồng sự kiện chính:** | |
| **Actor** | **Hệ thống** |
| 1. Teacher chọn chức năng Create Contest | 2. Hiển thị form tạo contest |
| 3. Teacher điền thông tin contest, chọn problem hoặc chọn chức năng Create Problem và xác nhận. | 4. Hiển thị thông báo và thông tin contest vừa tạo. |
| **Luồng sự kiện phụ:** | |
| 3.1.1. Teacher chọn chức năng Create Problem. | 3.1.2. Hiển thị form tương ứng. |
| 3.1.3. Teacher thực hiện các thao tác yêu cầu của chức năng Create Problem và xác nhận. | 3.1.4. Hiển thị thông báo và form tạo contest ban đầu |
| 3.2.2. Teacher nhập lại thông tin. | 3.2.1. Hiển thị thông báo form chưa đủ thông tin hoặc không hợp lệ. |

Bảng 2.7. Use case Create Problem.

|  |  |
| --- | --- |
| **Tên Use case:** Create Problem | |
| **Actor:** Teacher | |
| **Chức năng:** Use case cho phép actor tạo problem | |
| **Điều kiện trước:** Actor đã khởi động ứng dụng | |
| **Điều kiện sau:** Hiển thị list problem | |
| **Luồng sự kiện chính:** | |
| **Actor** | **Hệ thống** |
| 1. Teacher chọn chức năng Create Problem | 2. Hiển thị form tương ứng |
| 3. Teacher điền thông tin của problem, chọn file đề bài, folder chứa file input, output và xác nhận | 4. Thêm problem vừa tạo, lấy danh sách problem, thông báo và hiển thị danh sách problem |
| **Luồng sự kiện phụ:** | |
| 3.1.1. Teacher chọn chức năng chọn file đề bài. | 3.1.2. Cho phép chọn file trong hệ thống thư mục, lưu file, hiển thị đường dẫn. |
| 3.2.1. Teacher chọn chức năng chọn folder input và output. | 3.2.2. Cho phép chọn folder, lấy các file input và output, hiển thị đường dẫn |

Bảng 2.8. Use case View Result (Teacher).

|  |  |
| --- | --- |
| **Tên Use case:** View Result | |
| **Actor:** Teacher | |
| **Chức năng:** Use case cho phép actor xem thông tin các lần nộp bài và thông tin chi tiết của lần nộp đó. | |
| **Điều kiện trước:** Actor đã khởi động ứng dụng và đã start contest. | |
| **Điều kiện sau:** | |
| **Luồng sự kiện chính:** | |
| **Actor** | **Hệ thống** |
| 1. Teacher chọn chức năng View Result. | 2. Lấy danh sách các lần submit trước đó và hiển thị. |
| **Luồng sự kiện phụ:** | |
| 1.1. Teacher chọn một hàng trong danh sách để xem thông tin chi tiết của lần nộp bài đó. | 1.2. Lấy thông tin và kết quả của lần nộp bài đó, hiển thị. |

Bảng 2.9. Use case Get Result.

|  |  |
| --- | --- |
| **Tên Use case:** Get Result | |
| **Actor:** Teacher | |
| **Chức năng:** Use case cho phép actor lấy danh sách kết quả contest tại thời điểm hiện tại. | |
| **Điều kiện trước:** Actor đã khởi động ứng dụng và đã start contest. | |
| **Điều kiện sau:** Hiển thị bảng kết quả | |
| **Luồng sự kiện chính:** | |
| **Actor** | **Hệ thống** |
| 1. Teacher chọn chức năng Get Result. | 2. Lấy danh sách kết quả của contest tại thời điểm đó và trả về bảng kết quả. |

Bảng 2.10. Use case View List Problem.

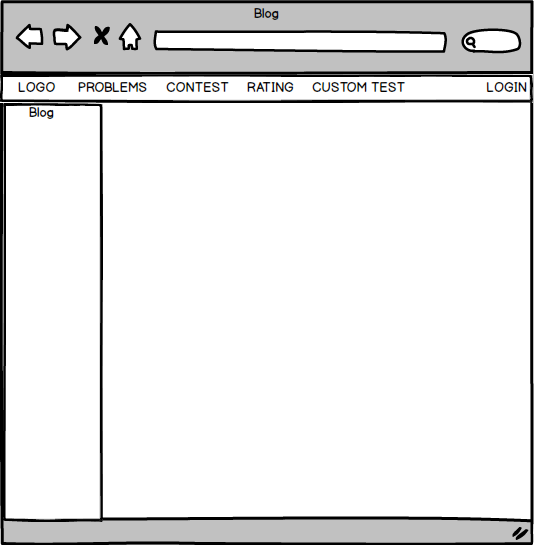
|  |  |
| --- | --- |
| **Tên Use case:** View List Problem | |
| **Actor:** Teacher | |
| **Chức năng:** Use case cho phép actor xem danh sách các problem. | |
| **Điều kiện trước:** Actor đã khởi động ứng dụng. | |
| **Điều kiện sau:** Hiển thị danh sách và thông tin problem. | |
| **Luồng sự kiện chính:** | |
| **Actor** | **Hệ thống** |
| 1. Teacher chọn chức năng View List Problem. | 2. Lấy danh sách problem và hiển thị. |
| **Luồng sự kiện phụ:** | |
| 1.1. Teacher chọn problem trong danh sách. | 1.2. Lấy thông tin problem và hiển thị. |

Bảng 2.11. Use case Load Contest.

|  |  |
| --- | --- |
| **Tên Use case:** Load Contest | |
| **Actor:** Teacher | |
| **Chức năng:** Use case cho phép actor start contet | |
| **Điều kiện trước:** Actor đã khởi động ứng dụng. | |
| **Điều kiện sau:** Contest bắt đầu. | |
| **Luồng sự kiện chính:** | |
| **Actor** | **Hệ thống** |
| 1. Teacher chọn chức năng Load Contest. | 2. Lấy thông tin contest và chạy contest, cho phép người dùng tham gia nộp bài. |

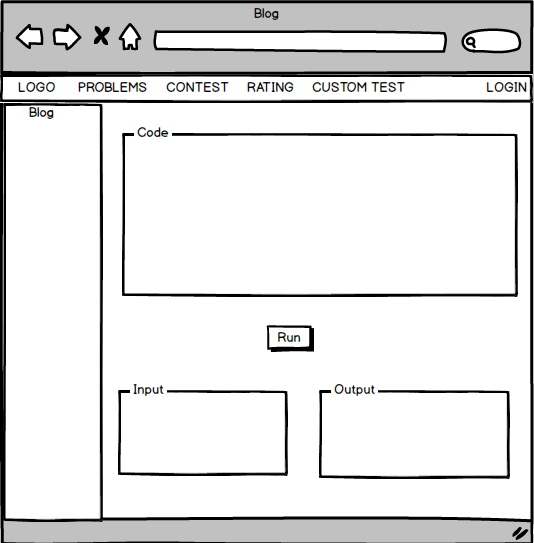
## Thiết kế giao diện.

### Trang chủ.



Hình 2.2. Giao diện Mockup – Home.

### Customtest.



Hình 2.3. Giao diện Mockup – Customtest.

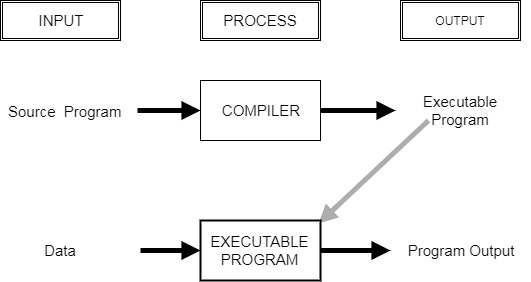
## Hoạt động của ứng dụng.

Ứng dụng sẽ lấy mã nguồn của người dùng và danh sách các dữ liệu đầu vào để ghi vào file trong thư mục lưu trữ của hệ thống. Sau đó ứng dụng thực thi file mã nguồn bằng lệnh hệ thống:

*$ sudo g++ -o fileclass sourcecode*

*$ sudo ./fileclass < fileinput > file output*

Hoạt động của chương trình được mô tả trong sơ đồ dưới:



Hình 2.4. Sơ đồ hoạt động của môi trường chấm bài.

Đầu vào là dữ liệu và mã nguồn qua tiến trình mã nguồn sẽ được chuyển thành file thực thi, file thực thi tiếp tục được chuyển thành tiến trình để chuyển đổi dữ liệu đầu vào thành dữ liệu đầu ra. Kết quả cần là dữ liệu đầu ra không bao gồm file thực thi. Nếu trong quá trình biên dịch bị lỗi hệ thống sẽ trả về lỗi và kết thúc quá trình chấm bài.

**Kết luận:** Quá trình xây dựng ứng dụng đảm bảo đầy đủ các bước xây dựng và phát triển một ứng dụng cơ bản. Hoạt động của ứng dụng được mô tả một cách đầy đủ việc chấm bài thi.

# TRIỂN KHAI ỨNG DỤNG TRÊN AMAZON WEB SERVICES

Giới thiệu về nền tảng điện toán đám mây của AWS và các dịch vụ sử dụng, các cài đặt và cấu hình để triển khai ứng dụng trên đám mây.

## Amazon Web Services (AWS).

### Giới thiệu.

AWS là dịch vụ điện toán đám mây được cung cấp bởi amazon.com. Cung cấp cho người dùng một nền tảng đám mây với cơ sở hạ tầng đáng tin cậy, có thể mở rộng và chi phí thấp, đặc biệt có thể đáp ứng được số lượng lớn các doanh nghiệp trên thế giới.

### Các dịch vụ sử dụng.

Elastic Compute Cloud (EC2): là nền tảng cơ sở cho môi trường điện toán đám mây. Giúp cho việc tạo ra, khởi động và dự phòng các ứng dụng ảo cho cá nhân hay doanh nghiệp một cách đơn giản và bất cứ khi nào.

Identity And Access Management(IAM): là dịch vụ cho phép tạo và quản lý, kiểm soát truy cập an toàn các dịch vụ và tài nguyên AWS của mình.

DynamoDB: dịch vụ cơ sỡ dữ liệu NoSQL.

## Triển khai trên EC2.

### Tạo tài khoản AWS.

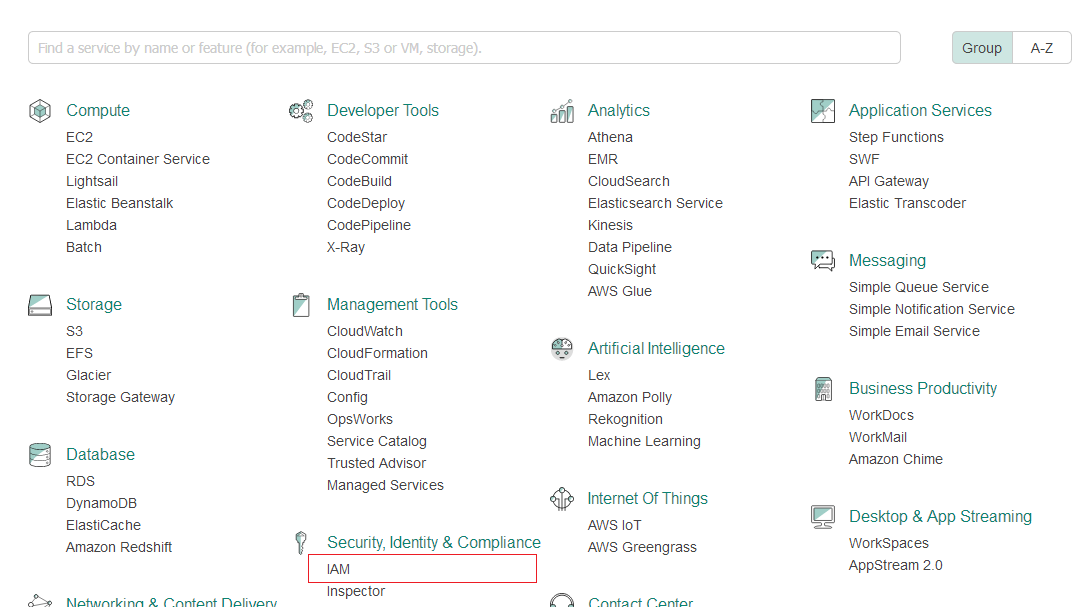
Để sử dụng các dịch vụ của AWS ta cần đăng ký tài khoản. Truy cập vào trang chủ <http://aws.amazon.com/> chọn Sign In to The Console để đăng ký tài khoản. Tài khoản đó được gọi là root account. Từ root account có thể tạo thêm nhiều người dùng khác với các quyền khác nhau.

Điều kiện đăng ký tài khoản AWS là người dùng phải có thẻ thanh toán quốc tế và số dư tài khoản lớn hơn 1$. Người dùng khi mới tạo tài khoản sẽ được hưởng các dịch vụ dùng thử miễn phí trải nghiệm các dịch vụ của AWS. Ứng dụng chấm bài của nhóm sử dụng các dịch vụ có sẵn miễn phí để triển khai.

### Tạo người dùng với IAM.

Để tăng độ bảo mật ta sử dụng IAM quản lý người dùng hệ thống.

Bước 1: Sử dụng root account để đăng nhập vào cửa sổ quản lý dịch vụ AWS. Và chọn dịch vụ IAM.



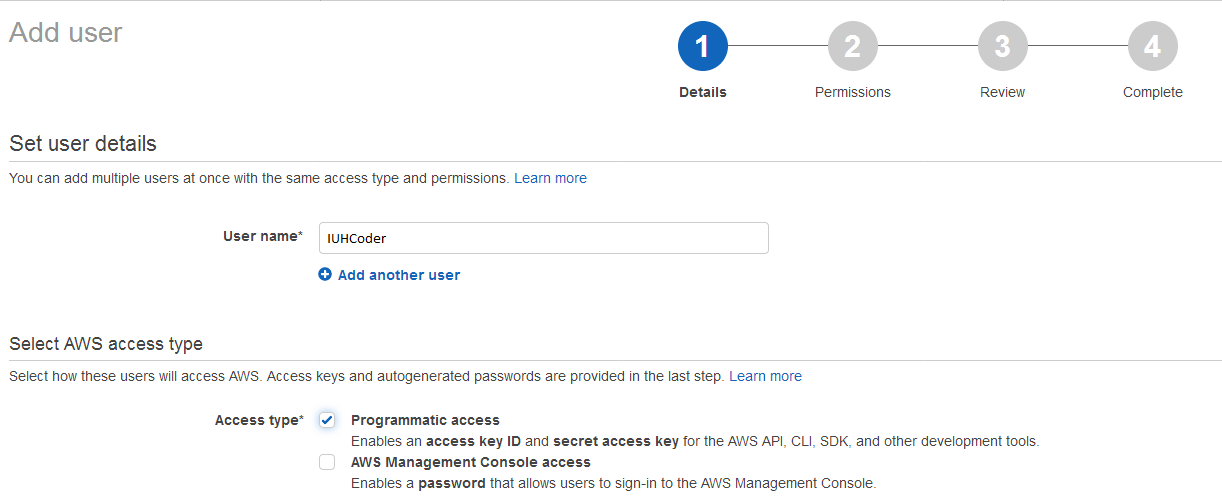
Hình 3.1. Cửa sổ dịch vụ của AWS.

Bước 2: Chọn User và Add user.



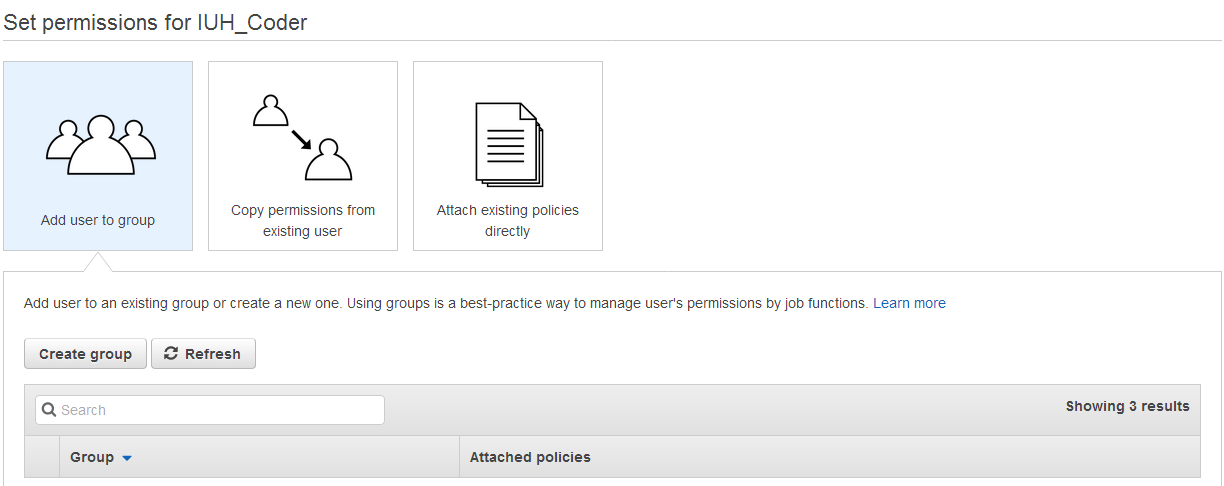
Hình 3.2. Thêm người dùng.

Bước 3: Đặt tên người dùng, chọn phương thức truy cập và chọn Next.



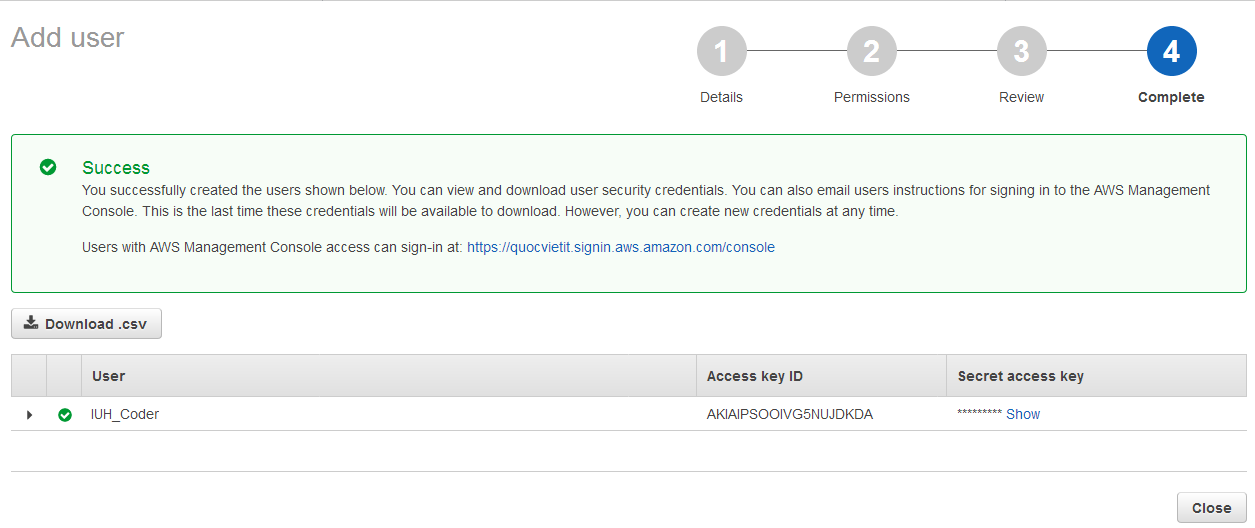
Hình 3.3. Điền thông tin người dùng.

Bước 4: Cấp quyền cho người dùng, ở đây ta có thể tạo nhóm người dùng hoặc cấp quyền trực tiếp. Ở đây sử dụng dịch vụ EC2 và DynamoDB nên ta cấp quyền [AmazonEC2FullAccess](https://console.aws.amazon.com/iam/home?region=us-east-1#policies/arn:aws:iam::aws:policy/AmazonEC2FullAccess) và AmazonDynamoDBFullAccess. Sau khi cấp quyền ta chọn Next để xem lại thông tin người dùng và chọn Next để đến tạo người dùng.



Hình 3.4. Cấp quyền người dùng.

Bước 5: Sau khi tạo người dùng ta sẽ có được Access Key ID và Secret Access Key dùng để sử dụng các dịch vụ đã được cấp quyền của AWS.



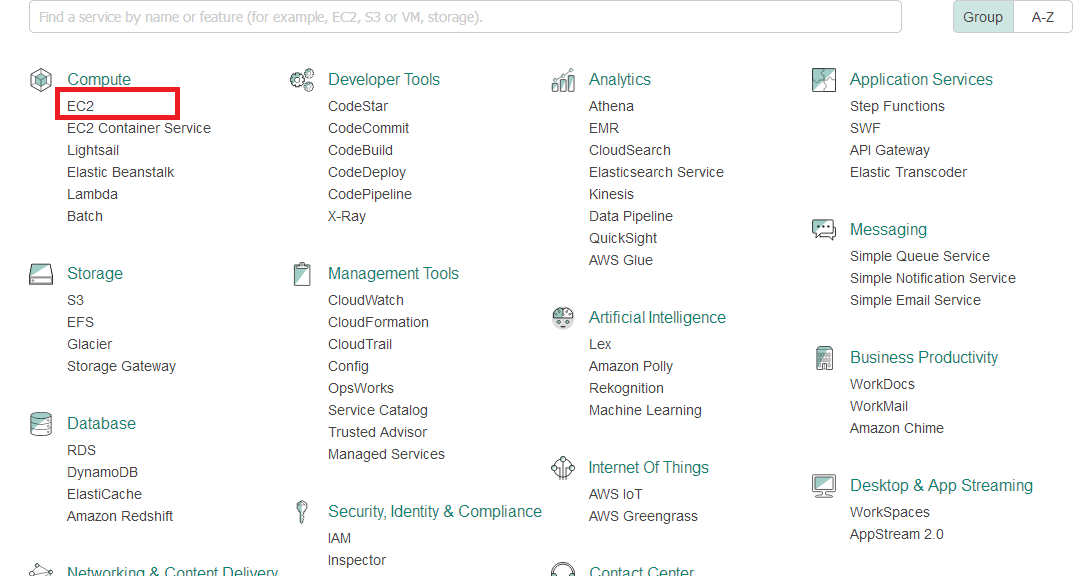
Hình 3.5. Thông tin người dùng đã đăng ký.

Secret Access Key chỉ cấp một lần khi tạo người dùng nên nếu không có Secret Access Key phải tạo lại một Access Key ID mới.

### Tạo máy chủ EC2.

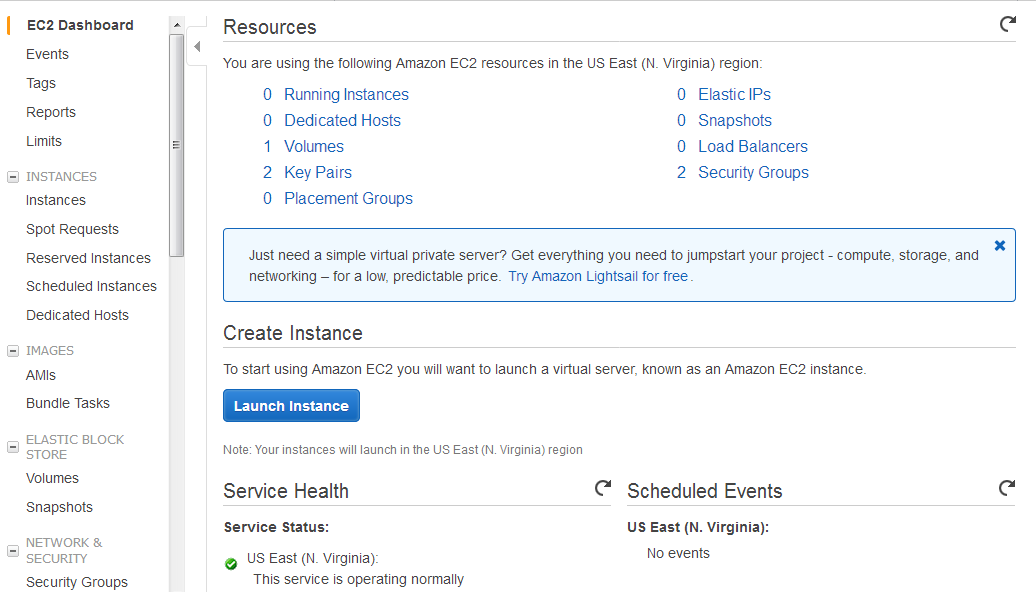
Sau khi có người dùng ta đăng nhập tài khoản người dùng để tạo máy chủ EC2. Máy chủ dùng để chạy ứng dụng sử dụng hệ điều hành Ubuntu Server 16.04 LTS.

Bước 1: Chọn dịch vụ EC2 tại cửa sổ quản lý dịch vụ của AWS.



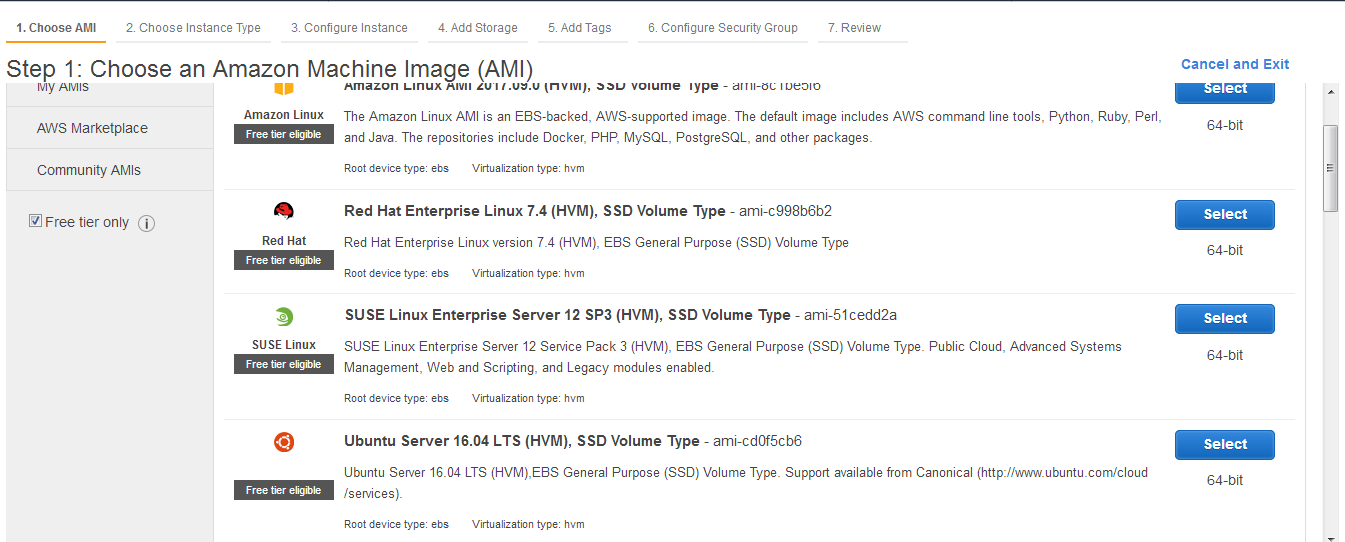
Hình 3.6. Cửa sổ dịch vụ của AWS.

Bước 2: Chọn Launch Instance để tạo một máy chủ dịch vụ EC2.



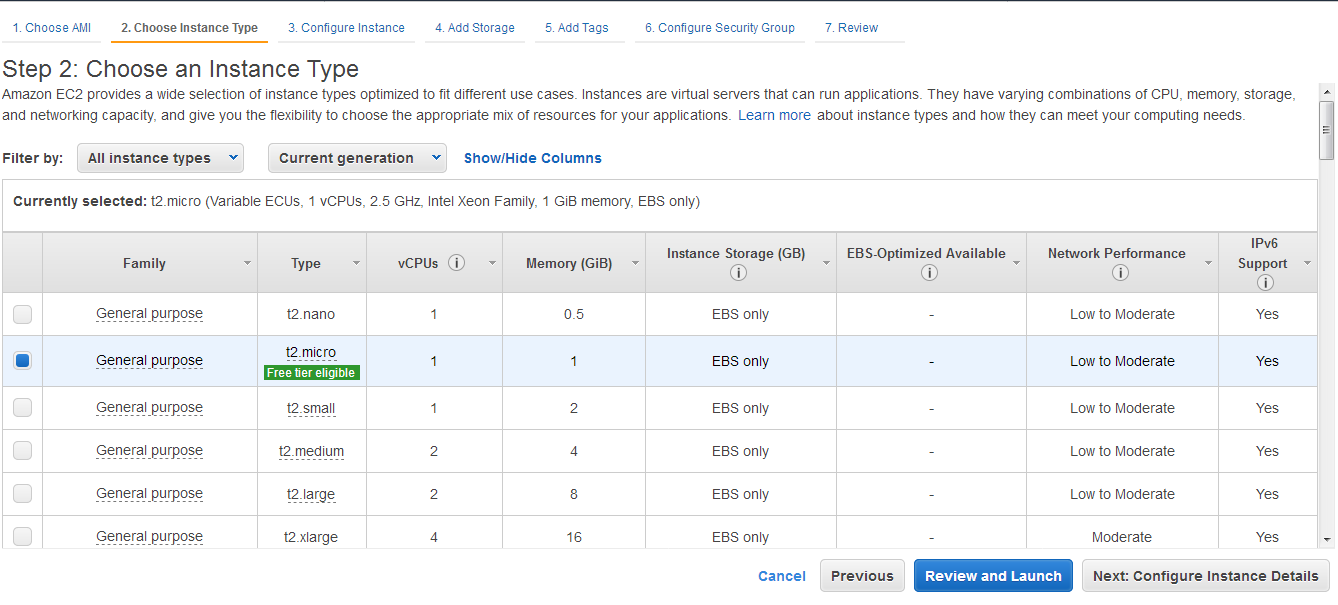
Hình 3.7. Bảng điều khiển EC2.

Bước 3: Lựa chọn hệ điều hành cho EC2. Ở đây ta sử dụng hệ điều hành Ubuntu Server 16.04 LTS.(Bản dùng thử miễn phí 750 giờ).



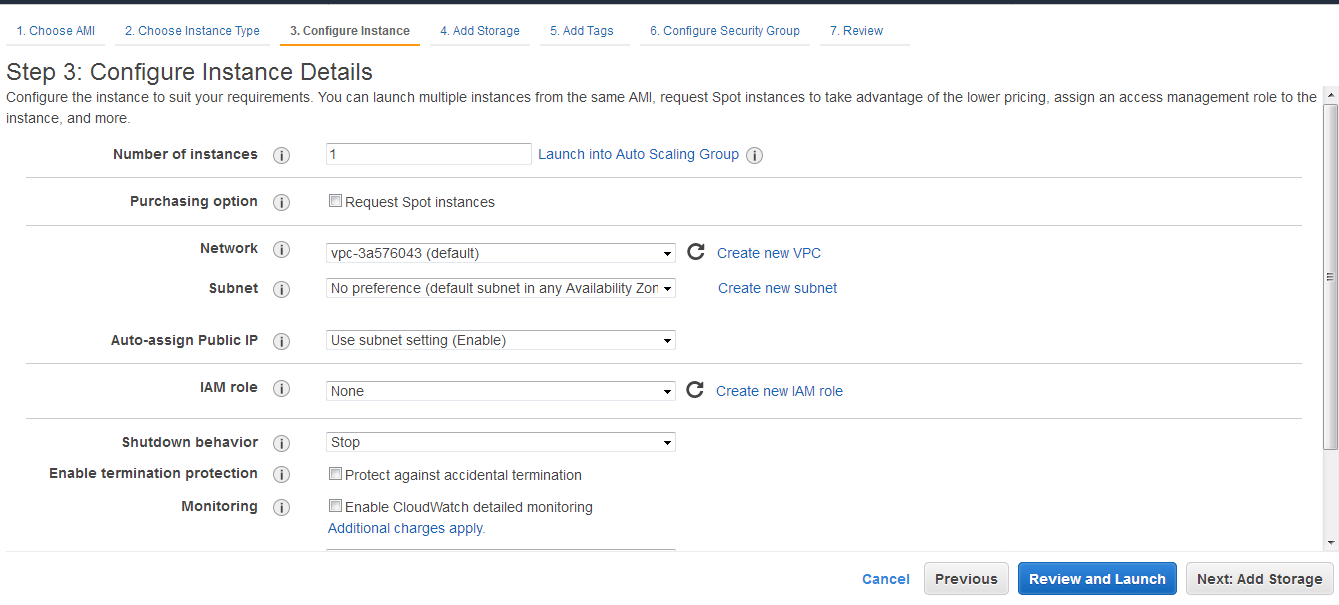
Hình 3.8. Chọn hệ điều hành cho máy chủ EC2.

Bước 4: Lựa chọn cấu hình máy chủ như type, CPU, memory… Tùy theo nhu cầu sử dụng của mỗi máy chủ mà ta có thể lựa chọn phù hợp. Cấu hình máy chủ càng cao thì giá thành lớn. Để đảm bảo sử dụng tối ưu ta nên chọn cấu hình vừa đủ để chạy khi có nhu cầu có thể mở rộng thêm, việc này chỉ mất vài phút.

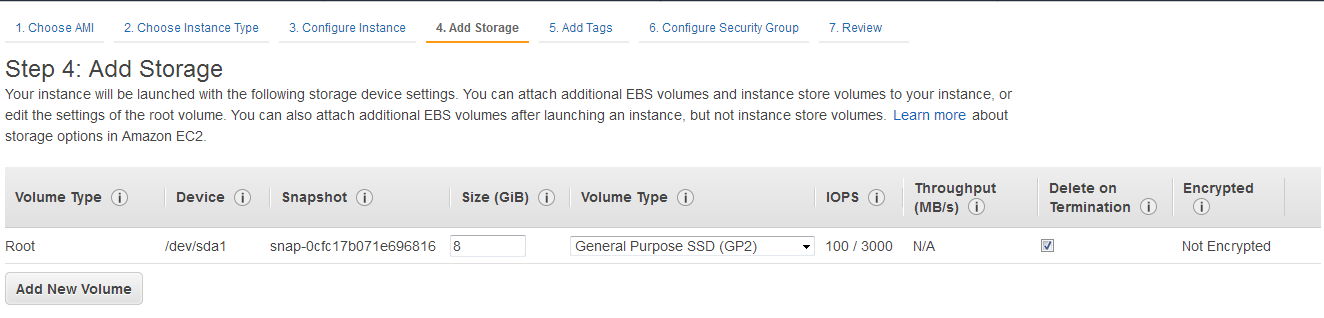


Hình 3.9. Lựa chọn cấu hình máy chủ EC2.

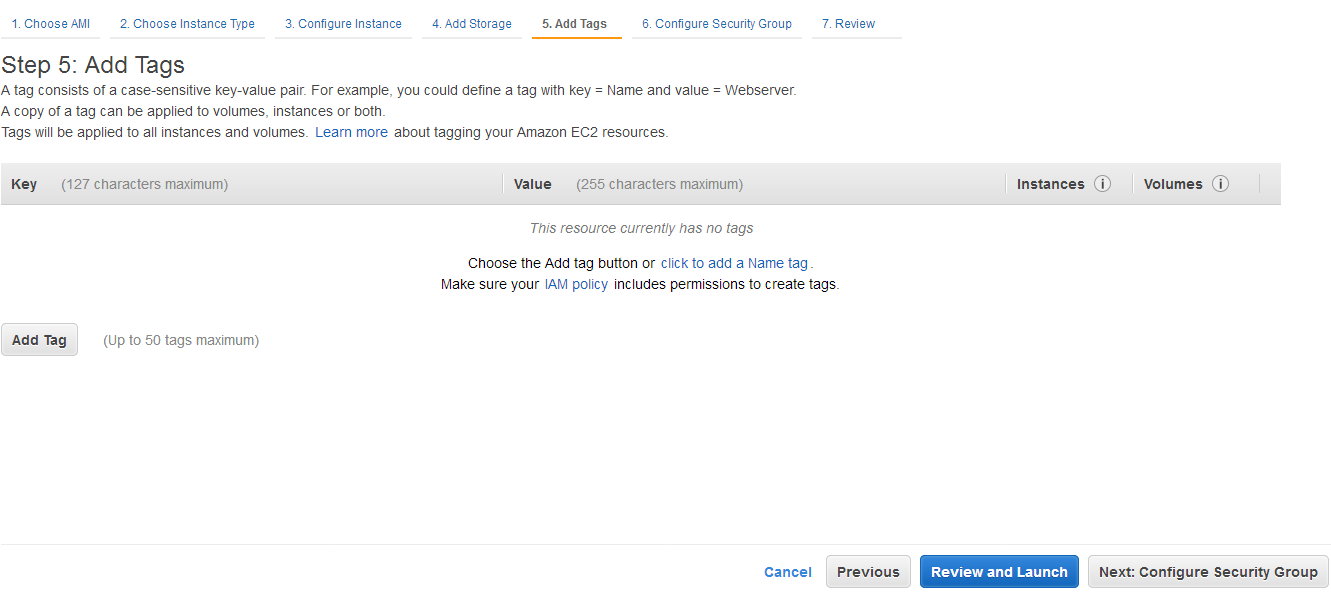
Bước 5: Lựa chọn cấu hình chi tiết. Lựa chọn số lượng máy chủ tạo ra, kết nối mạng của máy chủ, subnet,…



Hình 3.10. Lựa chọn cấu hình chi tiết.

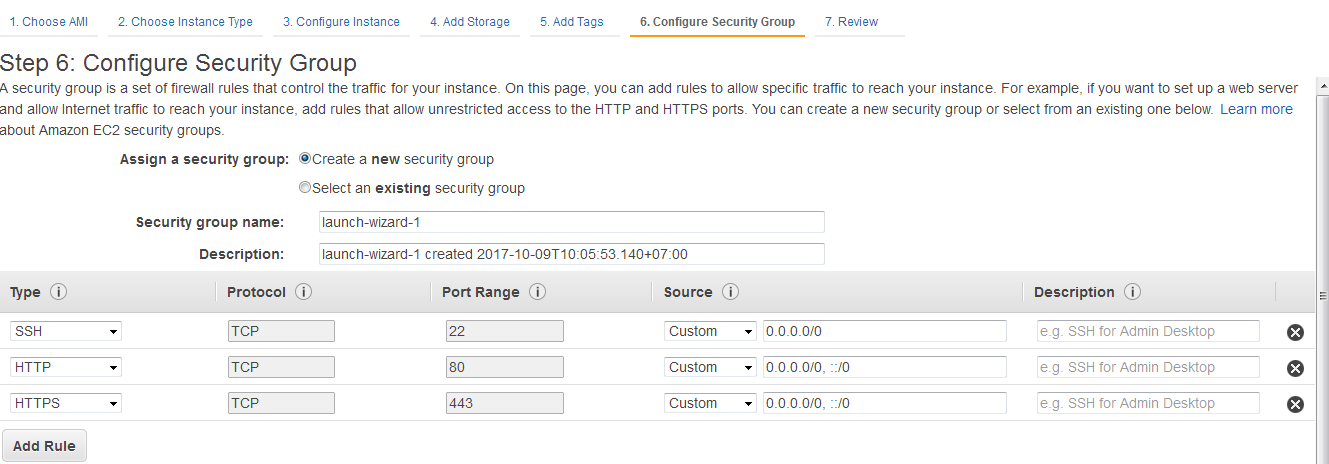
Bước 6: Thêm dung lượng bộ nhớ. Bộ nhớ mặc định được tạo là ổ cứng 8GB loại SSD.

Hình 3.11. Thêm dung lương bộ nhớ.

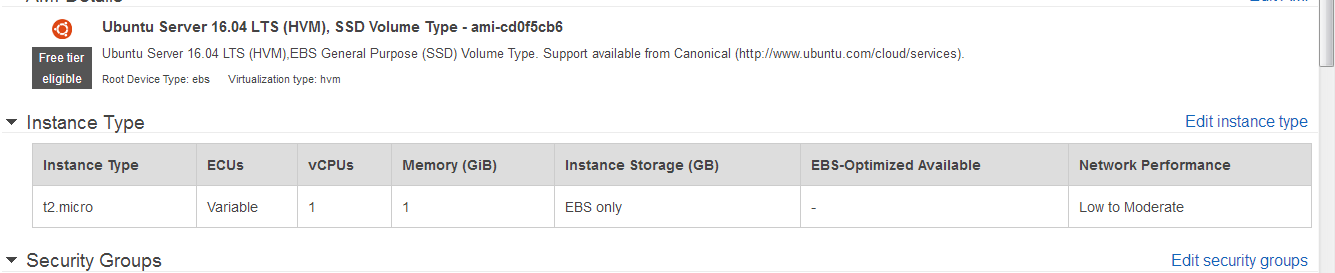
Bước 7: Gắn nhãn cho máy chủ.

Hình 3.12. Gắn nhãn máy chủ.

Bước 8: Cấu hình bảo mật. Ở đây ta mở cổng SSH, HTTP và HTTPS để có thể chạy ứng dụng web và làm việc trên máy chủ.

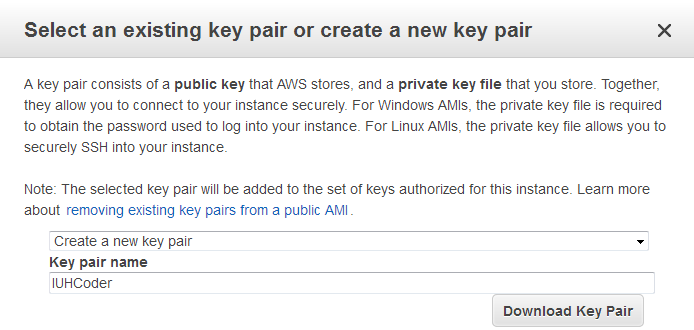


Hình 3.13. Cấu hình bảo mật.

Bước 9: Xem lại thông tin máy chủ và chọn Launch.

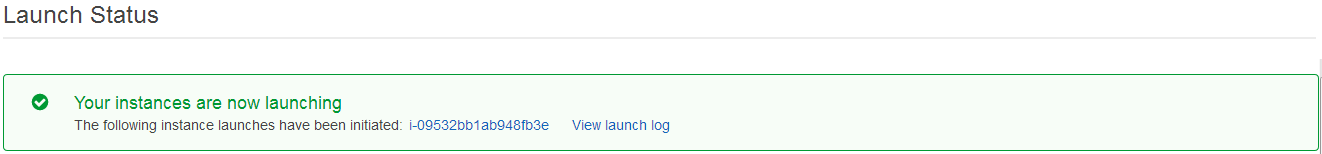
Hình 3.14. Xem lại thông tin máy chủ.

Bước 10: Lựa chọn keypair cho máy chủ. Nếu chưa có thì ta phải tạo mới và nhập tên cho keypair. Và chọn Launch Instance để khởi tạo máy chủ.



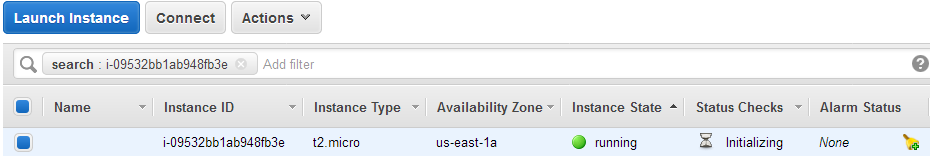
Hình 3.15. Lựa chọn kerpair cho máy chủ.

Bước 11: Quá trình khởi tạo máy chủ mất khoảng vài phút. Sau khi tạo thành công ta sẽ nhận được thông báo kèm id của máy chủ.



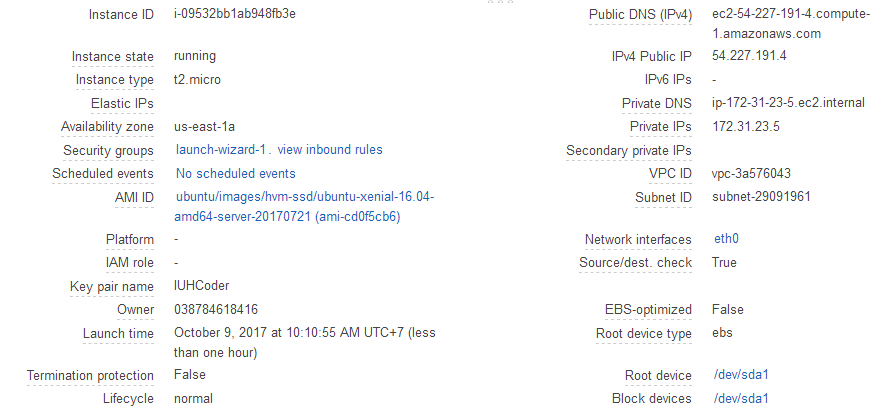
Hình 3.16. Thông báo tạo máy chủ thành công.

Máy chủ có ba trạng thái là Running, Stop, Terminal. Ta có thể chuyển đổi từ trạng thái Running sang Stop và ngược lại. Tuy nhiên chỉ có thể chuyển từ Stop sang Terminal. Nếu máy chủ ở trạng thái Terminal thì ta không thể chuyển trạng thái và máy chủ sẽ bị xóa sau vài giờ.



Hình 3.17. Máy chủ EC2 đã tạo.

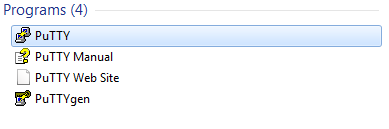
EC2 Instance được tạo thành công sẽ có Private IP và Public IP. Private IP là IP được cấp cho Instance đến khi nó bị xóa còn Public IP được cấp khi Instance chạy nếu như ta tắt đi và mở lại thì Public IP sẽ thay đổi. Một số thông tin khác như vùng, trạng thái, ổ cứng,… được mô tả trong hình bên dưới.



Hình 3.18. Thông tin chi tiết của máy chủ EC2.

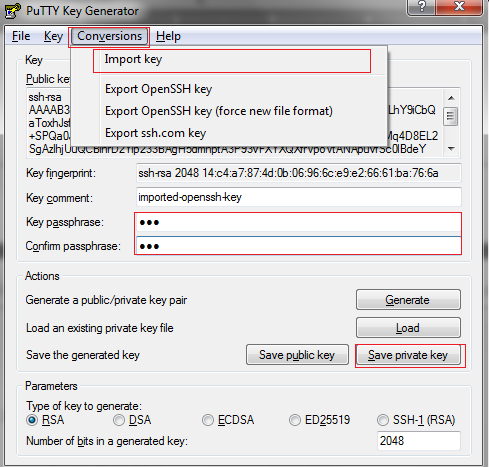
### Kết nối EC2.

Để kết nối tới EC2 Instance vừa tạo ta sử dụng phần mềm putty và puttygen, để có thể kết nối SSH ta đã mở port 22 và tạo keypair ở bước tạo EC2 Instance.



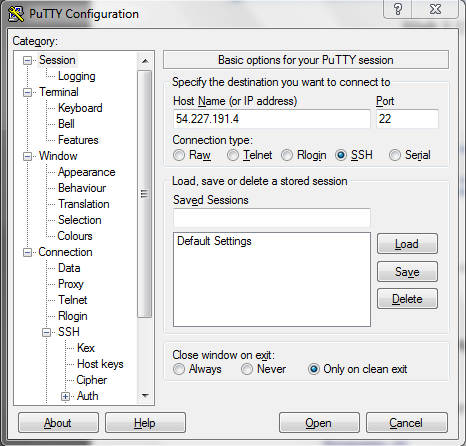
Hình 3.19. Phần mềm putty và puttygen.

Bước 1: Mở phần mềm puttygen, chọn Conversions Import key và chọn file keypair đã tạo ra khi tạo máy chủ EC2. Điền thông tin mật khẩu và chọn Save private key để tạo ra file .ppk.



Hình 3.20. Tạo khóa truy cập EC2 với Puttygen.

Bước 2: Mở phần mềm putty, điền Public IP và port của EC2 Instance.



Hình 3.21. Phần mềm Putty.

Bước 3: Chọn Connection SSH Auth Browse.. Ta lựa chọn file chứa khóa là file .ppk đã tạo ra ở bước 1.

Bước 4: Tại Data điền mục user name theo hệ điều hành đã cài đặt:

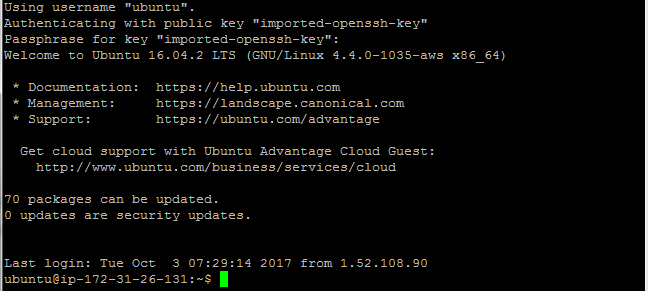
- Amazon Linux AMI**: ec2-user.** - RHEL5 AMI : **root** or **ec2-user.**

- Ubuntu AMI : **ubuntu.**

- Fedora AMI : **fedora** or **ec2-user.**

- SUSE Linux : **root** or **ec2-user.**

**Ở đây ta sử dụng hệ điều hành Ubuntu Server 16.04 LTS nên chọn user name là ubuntu.** Quay lại mục Session chọn Save và nhấn Open để kết nối tới EC2 Instance. Nhập mật khẩu đã tạo ở bước 1 để đăng nhập vào máy chủ.



Hình 3.22. Giao diện dòng lệnh làm việc với EC2 Instance.

### Cài đặt môi trường.

Cập nhật phiên bản mới nhất cho ubuntu:

*$ sudo apt-get install update*

Cài đặt Apache:  
 *$ sudo apt-get update && sudo apt-get upgrade  
 $ sudo apt-get install apache2 apache2-doc apache2-utils*

Cài đặt python 2.7:

*$ sudo apt-get apt-add-repository ppa:fkrull/deadsnakes-python2.7*

*$ sudo apt-get update*

*$ sudo apt-get install python2.7 python2.7-dev*

Cài đặt pip:

*$ sudo apt-get install python-pip*

*$ sudo apt-get --upgrade pip*

Cài đặt flask:

*$ sudo pip install Flask*

Cài đặt môi trường cho ứng dụng:

*$ sudo pip install virtualvenv*

*$ sudo apt-get install python-virtualenv*

Tạo thư mục và khởi tạo môi trường:

*$ sudo mkdir IUHCoder*

*$ sudo cd IUHCoder*

*$ sudo virtualenv venv*

Cài đặt môi trường chạy bài lập trình. Để chạy được chương trình C/C++ ta cần cài trình biên dịch GNU cho EC2 Instance. Hệ điều hành ubuntu đã tích hợp sẵn trình biên dịch cho ngôn ngữ C nên ta chỉ cần cái trình biên dịch cho C++ :

*$ sudo apt-get install g++*

## Đưa ứng dụng lên EC2.

Ứng dụng được lưu trữ trên github: <https://github.com/quocvietit/IUHCoder>.

Để đưa mã nguồn lên EC2 Instance ta cần cài git:  
 *$ sudo apt-get install git*

Lấy mã nguồn về EC2 Instance:

*$ sudo git clone* [*https://github.com/quocvietit/IUHCoder*](https://github.com/quocvietit/IUHCoder)

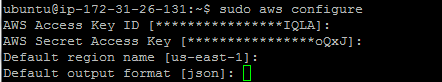
## Kết nối với database (DynamoDB):

Nhằm đảm bảo độ bảo mật của dữ liệu ngoài việc sử dụng IAM ta sử dụng AWS Command Line Interface :

*$ sudo pip install awscli*

Cấu hình AWS Command Line Interface:

*$ sudo aws configure*

**

Hình 3.23. Bảng cấu hình AWS Command Line Interface.

Sử dụng Access Key ID và Secret Access Key đã tạo ở bước tạo người dùng IAM để đăng nhập vào AWS Command Line Interface. Vùng ở đây ta sử dụng là us-east-1 (US East (N. Virginia)), định dạng của dữ liệu trả về là json.

Cài đặt boto3 để kết nối tới DynamoDB:

*$ sudo pip install botto3*

## Cấu hình ứng dụng.

Chương trình chính của ứng dụng là file \_\_main\_\_.py. Do việc thay đổi IP của EC2 Instance khi bật và tắt để giảm thiểu chi phí nên ta cần chỉnh file \_\_main\_\_.py để ứng dụng có thể chạy trên port 80 mà ta không cần cấu hình lại mỗi lần IP thay đổi.

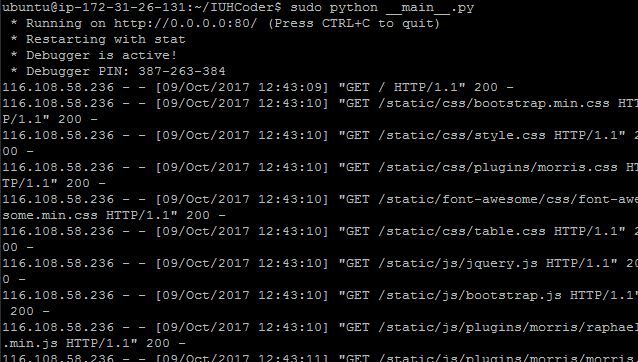
D:\IT\Project\IUHCoder\document\photo\AWSCLI\ip.PNG

Hình 3.24. Cấu hình IP và port .

## Kiểm thử.

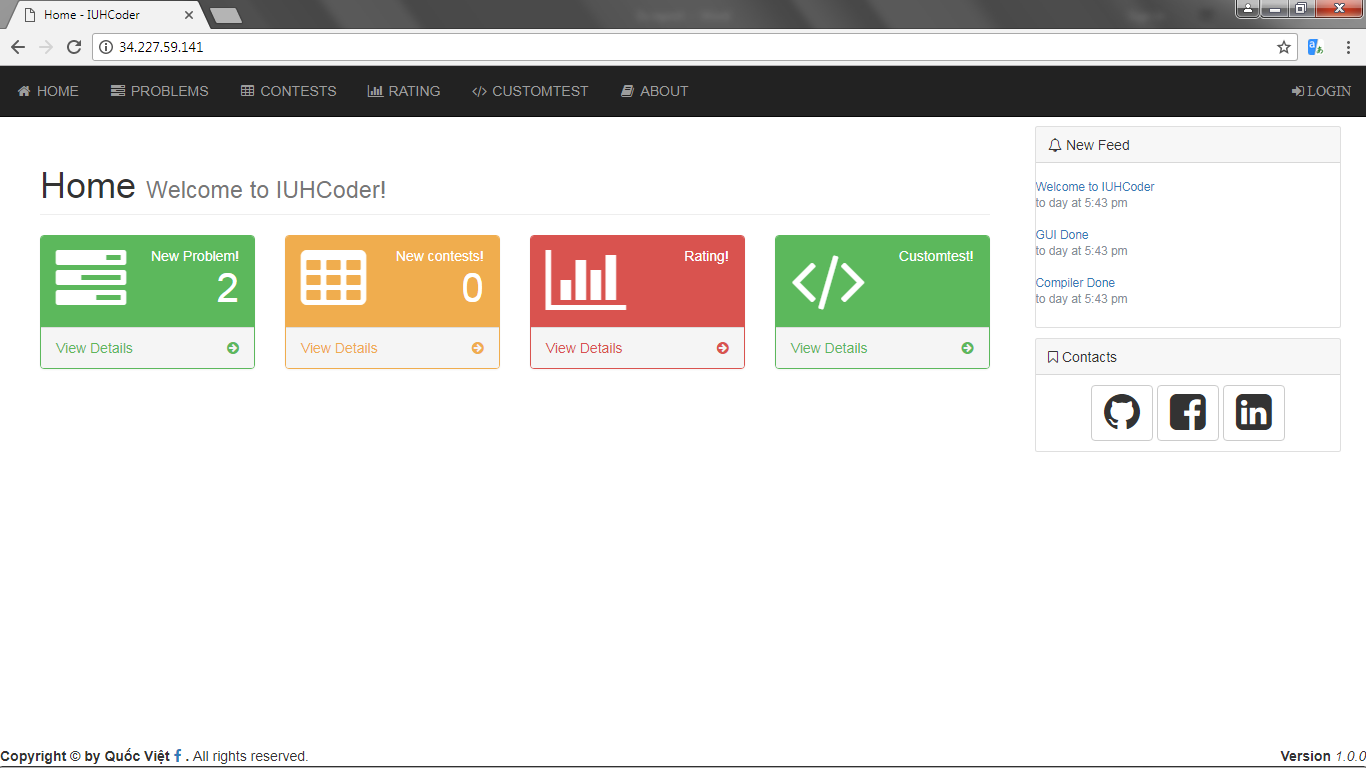
Chạy thử chương trình, ta vào thư mục chứa mã nguồn của ứng dụng và chạy lệnh:

*$ sudo python \_\_main\_\_.py*



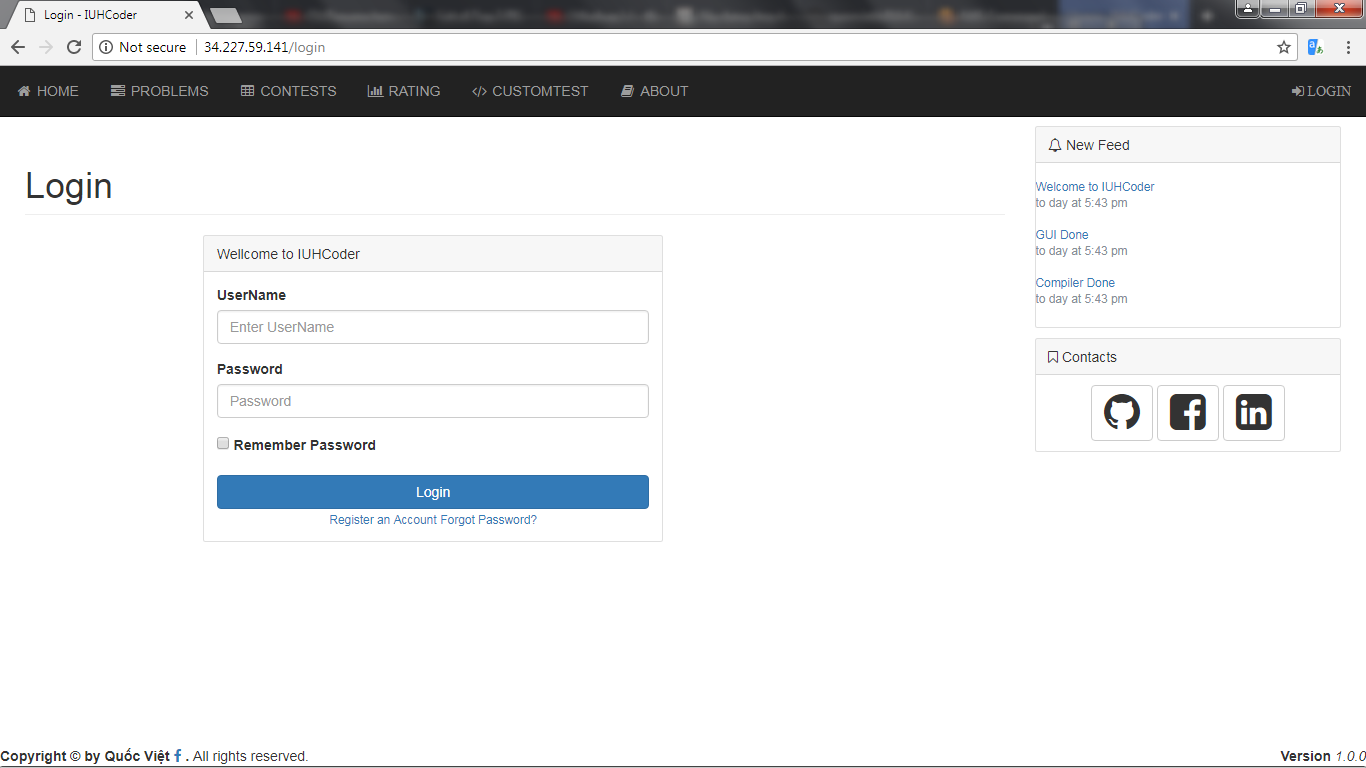
Hình 3.25. Chương trình ở chế độ debug.

Địa chỉ IP của EC2 Instance lúc này là: 34.227.59.14. Ta sử dụng trình duyệt web để truy cập ứng dụng.



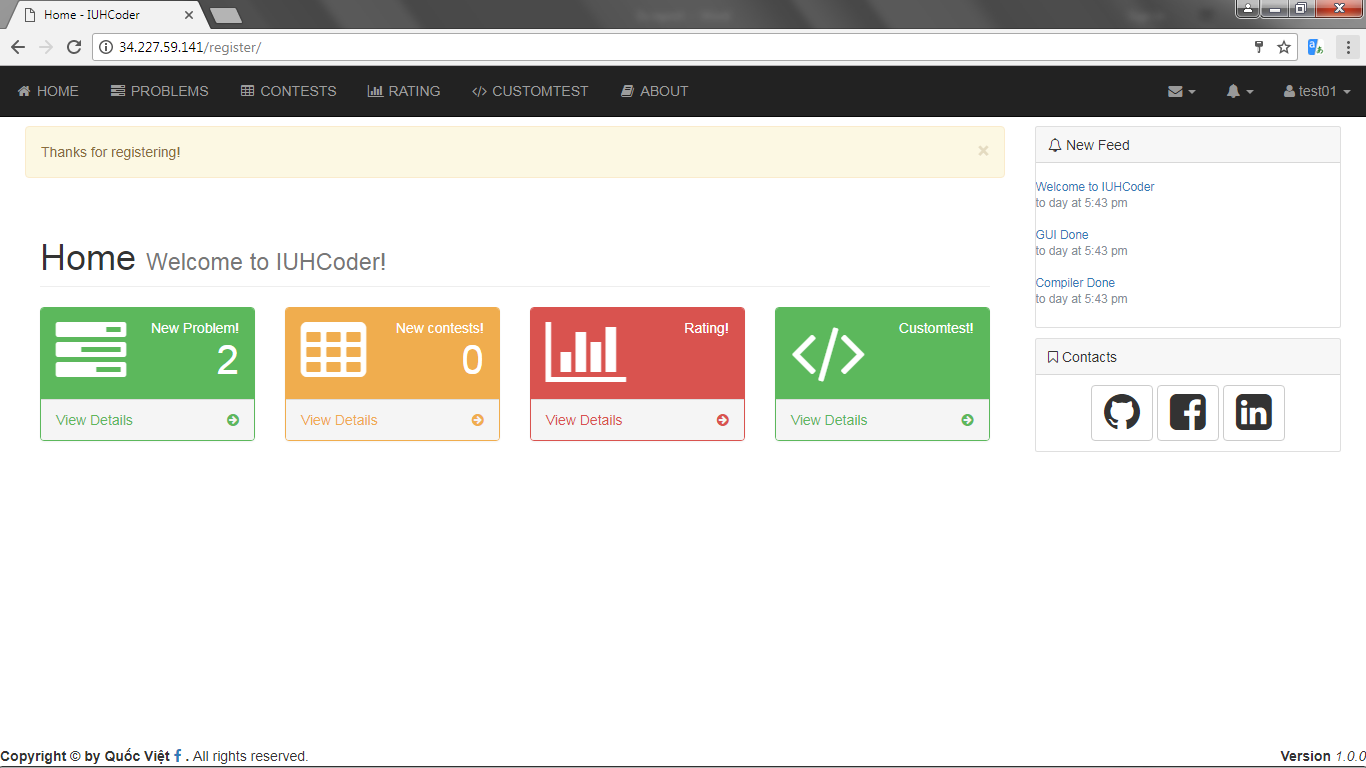
Hình 3.26. Giao diện trang chủ.

Hình 3.26 là giao diện khi truy cập đến ứng dụng gồm các danh mục của trang web, lời chào và thông tin liên hệ.



Hình 3.27. Giao diện đăng nhập.

Tại giao diện đăng nhập ta sẽ đăng ký một tài khoản tên là test01và mật khẩu là 123456789 để kiểm tra môi trường trình biên dịch của ngôn ngữ C/C++.



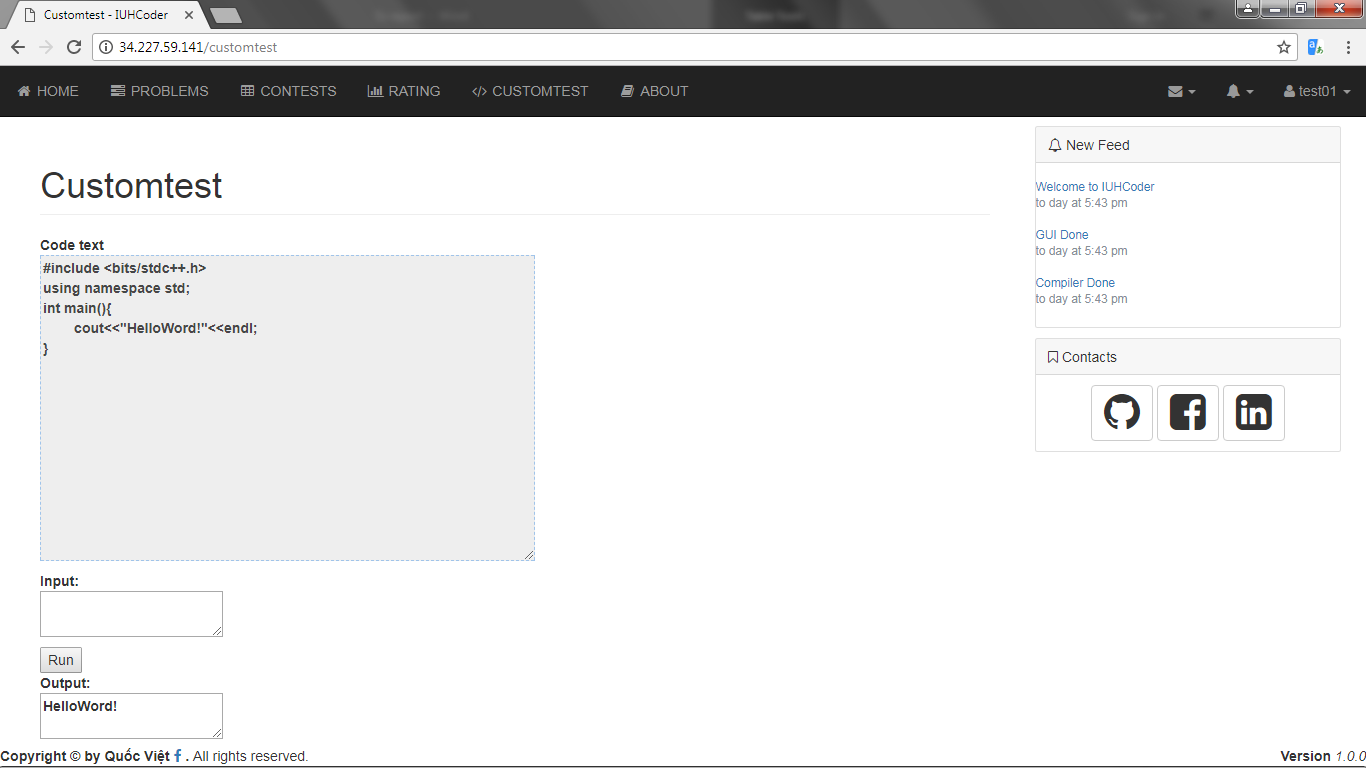
Hình 3.28. Đăng ký thành công tài khoản test01.

Tiến hành chạy thử các đoạn mã C/C++ đơn giản với kết quả đã được chạy trên máy tính.

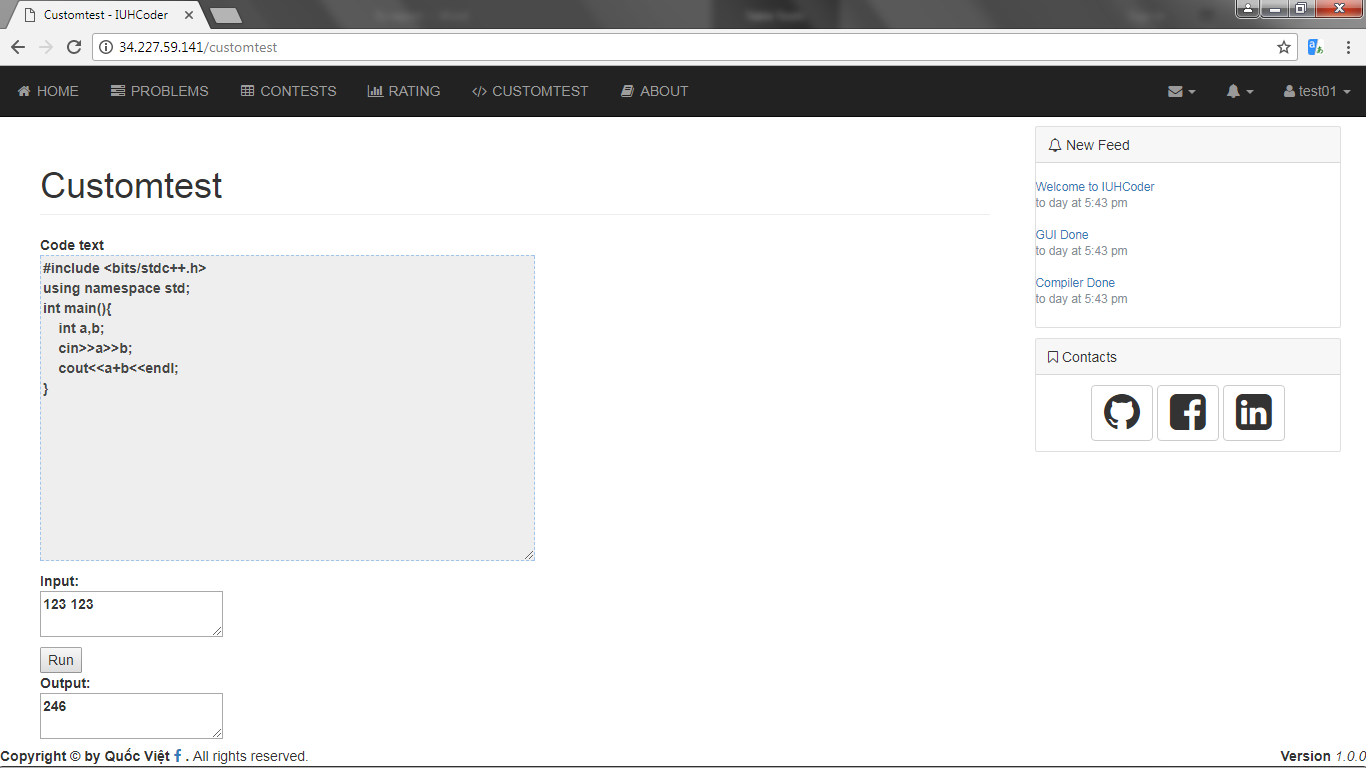
|  |  |  |  |
| --- | --- | --- | --- |
| STT | Mã nguồn | Đầu vào | Đầu ra |
| 1 | #include <bits/stdc++.h>  using namespace std;  **int** main(){  cout<<"HelloWord!"<<endl;  } |  | HelloWord! |
| 2 | #include <bits/stdc++.h>  using namespace std;  **int** main(){  **int** a,b;  cin>>a>>b;  cout<<a+b<<endl;  } | 123 123 | 246 |
| 3 | #include <bits/stdc++.h>  using namespace std;  **int** main(){  **int** a,b;  cin>>a>>b;  cout<<a+c<<endl;  } | 123 123 | Lỗi compiler |

Bảng 3.1. Các đoạn mã C/C++ mẫu.

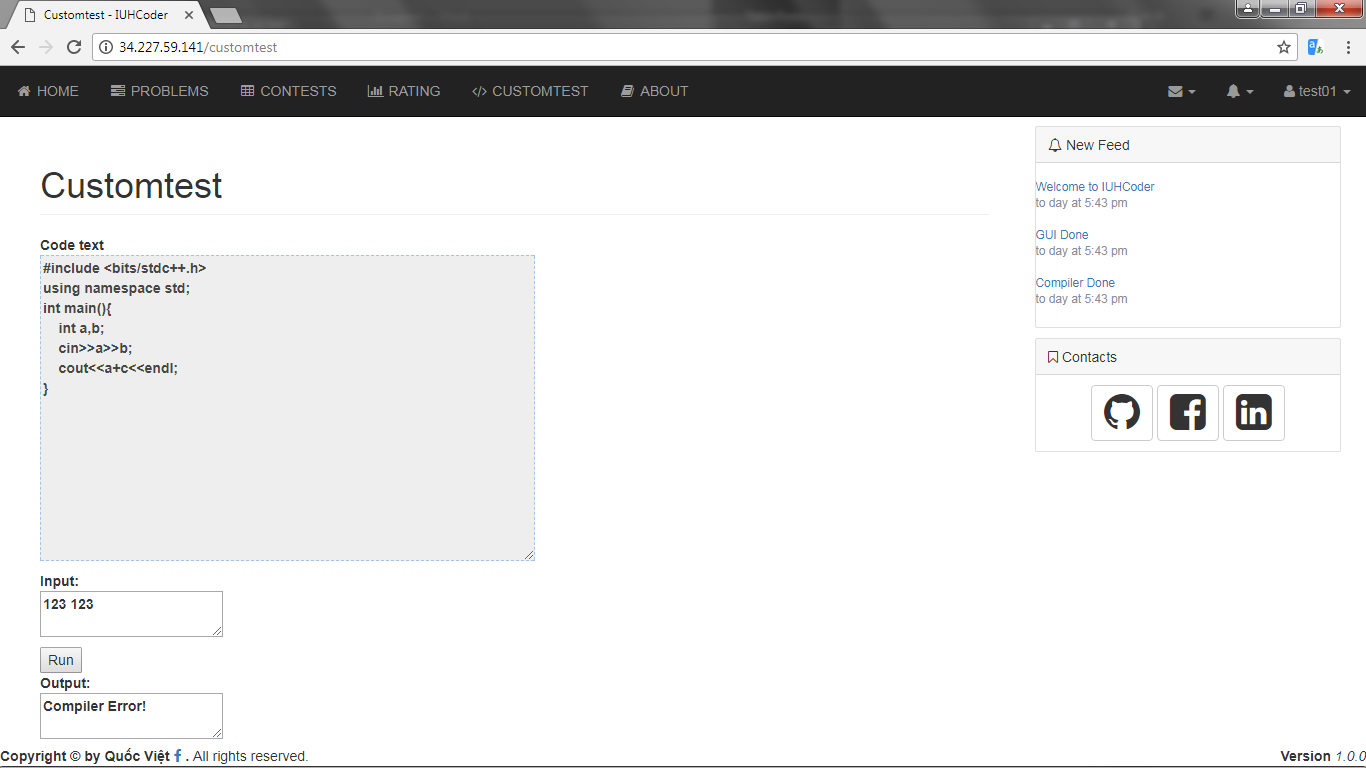
Chạy trên ứng dụng:



Hình 3.29. Kết quả đoạn mã 1.



Hình 3.30. Kết quả đoạn mã 2.



Hình 3.31. Kết quả đoạn mã 3.

**Kết luận:** Quá trình cài đặt và triển khai ứng dụng sử dụng các dịch vụ của AWS đều chạy ổn định. Kết quả của cả ba đoạn mã C/C++ so với đầu ra đều chính xác. Thời gian chạy so với chạy trên máy thật chênh lệch không lớn. Không xuất hiện lỗi trong quá trình chạy.

# KẾT LUẬN

Quá trình xây dựng ứng dụng chấm bài thi đã đảm bảo đầy đủ và được triển khai trên nền tảng công nghệ điện toán đám mây.

Ưu điểm:

- Giao diện đơn giản, dễ sử dụng.

- Ứng dụng đã tạo được môi trường chạy ngôn ngữ C/C++.

- Thời gian chấm bài nhanh, không xuất hiện lỗi.

- Kích thước mã nguồn của ứng dụng không lớn, có thể triển khai trên các máy chủ khác nhau một cách nhanh chóng.

Nhược điểm:

**-** Truy xuất dữ liệu người dùng chậm vì giới hạn request của DynamoDB.

**-** Mã nguồn và dữ liệu đầu vào của người dùng chỉ có thể sử dụng các ký tự trong bảng mã tiếng anh.

**-** Cơ sở dữ liệu sử dụng là DynamoDB nên nếu người dùng bị xóa thì cơ sở dữ liệu không thể truy cập.

Hướng phát triển của ứng là tạo trang quản lý cho phép người quản lý quản lý người dùng thành viên, tạo ra bài tập , các cuộc thi,…Người dùng khi đã là thành viên được tính điểm và xếp hạng. Ngoài ra còn có thể đóng góp ý kiến, xây dựng bài tâp. Xây dựng thêm các mục thảo luận, bản tin và tin nhắn,..

Để đáp ứng số lượng người truy cập lớn, máy chủ EC2 có thể mở rộng thêm bộ nhớ, CPU,…trong trường hợp cần thiết. Các dịch vụ đó đều được cung cấp một cách nhanh chóng.

# TÀI LIỆU THAM KHẢO

1. AWS tutorial: https://aws.amazon.com/articles/

2. Domjudge: <https://www.domjudge.org/>

3. PC^2: <https://pc2.ecs.csus.edu/>

4. Themis: <http://dsapblog.wordpress.com/2012/03/04/themis/>

5. Upcoder: <http://upcoder.hcmup.edu.vn/>