

**BỘ CÔNG THƯƠNG**  
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH**  
**KHOA CÔNG NGHỆ THÔNG TIN**  
❧ ❖ ❧

**ĐỒ ÁN MÔN HỌC**

**XÂY DỰNG ỨNG DỤNG CHĂM BÀI**

Giáo viên hướng dẫn : ThS. Đặng Thanh Bình  
ThS. Nguyễn Văn Quang  
Sinh viên thực hiện : 14127871 Phan Bảo Trinh  
14026751 Vương Quốc Việt  
Lớp : ĐHCNTT10B

***TP. HỒ CHÍ MINH, THÁNG 08 NĂM 2017***

## 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 thuật toán để người khác hiểu.

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.

## **DANH MỤC TỪ VIẾT TẮT**

AWS: Amazon Web Service

EC2: E

## DANH MỤC HÌNH ẢNH

Hình 2.1. Use case tổng quát.	2
Hình 2.2. Giao diện Mocup – Home.	12
Hình 2.3. Giao diện mocup – Customtest.	12
Hình 2.4. Sơ đồ hoạt động của môi trường chấm bài.	13
Hình 3.1. Cửa sổ dịch vụ của AWS.	15
Hình 3.2. Thêm người dùng.	15
Hình 3.3. Điền thông tin người dùng.	15
Hình 3.4. Cấp quyền người dùng.	16
Hình 3.5. Thông tin người dùng đã đăng ký.	16
Hình 3.6. Cửa sổ dịch vụ của AWS.	17
Hình 3.7. Bảng điều khiển EC2.	17
Hình 3.8. Chọn hệ điều hành cho máy chủ EC2.	18
Hình 3.9. Lựa chọn cấu hình máy chủ EC2.	18
Hình 3.10. Lựa chọn cấu hình chi tiết.	19
Hình 3.11. Thêm dung lượng bộ nhớ.	19
Hình 3.12. Gắn nhãn máy chủ.	19
Hình 3.13. Cấu hình bảo mật.	20
Hình 3.14. Xem lại thông tin máy chủ.	20
Hình 3.15. Lựa chọn keypair cho máy chủ.	20
Hình 3.16. Thông báo tạo máy chủ thành công.	21
Hình 3.17. Máy chủ EC2 đã tạo.	21
Hình 3.18. Thông tin chi tiết của máy chủ EC2.	21
Hình 3.19. Phần mềm putty và puttygen.	22
Hình 3.20. Tạo khóa truy cập EC2 với Puttygen.	22
Hình 3.21. Phần mềm Putty.	23

Hình 3.22. Giao diện dòng lệnh làm việc với EC2 instance.	24
Hình 3.23. Bảng cấu hình AWS Command Line Interface.	25
Hình 3.24. Cấu hình ip và port .	26
Hình 3.25. Chương trình ở chế độ debug.	26
Hình 3.26. Giao diện trang chủ.	27
Hình 3.27. Giao diện đăng nhập.	27
Hình 3.28. Đăng ký thành công tài khoản test01.	28
Hình 3.29. Kết quả đoạn mã 1.	29
Hình 3.30. Kết quả đoạn mã 2.	29
Hình 3.31. Kết quả đoạn mã 3.	30

## DANH MỤC BẢNG BIỂU

Bảng 2.1 Use case Submit. ....	3
Bảng 2.2 Use case View Result (Student).....	4
Bảng 2.3 Use case View ScoreBoard. ....	5
Bảng 2.4 Use case Login. ....	5
Bảng 2.5 Use case Logout. ....	6
Bảng 2.6 Use case Create Contest.....	7
Bảng 2.7 Use case Create Problem. ....	8
Bảng 2.8 Use case View Result (Teacher). ....	9
Bảng 2.9 Use case Get Result. ....	10
Bảng 2.10 Use case View List Problem. ....	10
Bảng 2.11 Use case Load Contest. ....	11
Bảng 3.1. Các đoạn mã C/C++ mẫu.....	28

## MỤC LỤC

LỜI NÓI ĐẦU .....	i
DANH MỤC TỪ VIẾT TẮT.....	ii
DANH MỤC HÌNH ẢNH .....	iii
DANH MỤC BẢNG BIỂU .....	v
MỤC LỤC .....	vi
Chương 1: GIỚI THIỆU .....	1
1.1. Mô tả đề tài.....	1
1.2. Mục tiêu, phạm vi đề tài.....	1
1.2.1. Mục tiêu.....	1
1.2.2. Phạm vi.....	1
Chương 2: XÂY DỰNG ỨNG DỤNG CHĂM BÀI.....	2
2.1. Phân tích hệ thống.....	2
2.1.1. Thiết kế use case.....	2
2.1.2. Yêu cầu chức năng.....	3
2.1.3. Yêu cầu phi chức năng.....	3
2.1.4. Đặc tả use case.....	3
2.2. Thiết kế giao diện.....	12
2.2.1. Trang chủ.....	12
2.2.2. Customtest.....	12
2.3. Hoạt động của ứng dụng.....	13
Chương 3: TRIỂN KHAI ỨNG DỤNG TRÊN AMAZON WEB SERVICES....	14
3.1. Amazon Web Services (AWS).....	14
3.1.1. Giới thiệu.....	14
3.1.2. Các dịch vụ sử dụng.....	14
3.2. Triển khai trên EC2.....	14

3.2.1.	Tạo tài khoản AWS. ....	14
3.2.2.	Tạo người dùng với IAM.....	14
3.2.3.	Tạo máy chủ EC2. ....	16
3.2.4.	Kết nối EC2. ....	22
3.2.5.	Cài đặt môi trường.....	24
3.3.	Đưa ứng dụng lên EC2. ....	25
3.4.	Kết nối với database (DynamoDB): .....	25
3.5.	Cấu hình ứng dụng. ....	26
3.6.	Kiểm thử.....	26
Chương 4:	KẾT LUẬN .....	31
4.1.	Kết quả triển khai ứng dụng. ....	31
4.1.1.	Ưu điểm. ....	31
4.1.2.	Nhược điểm. ....	31
4.1.3.	Lỗi.....	31
4.2.	Hướng phát triển.....	31
TÀI LIỆU THAM KHẢO.....		ix



## **Chương 1:      GIỚI THIỆU**

### **1.1. Mô tả đề tài.**

Ứng dụng chấm bài thi lập trình cho phép người quản lý quản lý người dùng, tạo ra các kỳ thi và các vấn đề cần giải quyết bằng lập trình. Người dùng sẽ dùng một ngôn ngữ lập trình nào đó để giải quyết vấn đề được đưa ra một cách tối ưu nhất. Ngoài ra người dùng còn được cung cấp một môi trường để chạy thử nghiệm các đoạn mã với dữ liệu đầu vào và trả về dữ liệu sau khi xử lý. Ứng dụng được xây dựng bằng ngôn ngữ Python.

### **1.2. Mục tiêu, phạm vi đề tài.**

#### **1.2.1. Mục tiêu.**

- Tạo ra môi trường chạy các ngôn ngữ lập trình phổ biến như C/C++, Java, Python, C#....
- Cung cấp cho người dùng môi trường chạy thử các đoạn mã và giải quyết các vấn đề được đưa ra.
- Tạo ra các kỳ thi với thời gian chấm bài nhanh và độ chính xác cao.
- Triển khai ứng dụng áp dụng các công nghệ mới.

#### **1.2.2. Phạm vi.**

- Ứng dụng được trong việc dạy và học lập trình trong các trường đại học, cao đẳng, trung học phổ thông.
- Sử dụng các dịch vụ điện toán đám mây của Amazon để triển khai ứng dụng.
- Tạo môi trường chạy ngôn ngữ lập trình C/C++.

## Chương 2: XÂY DỰNG ỨNG DỤNG CHẤM BÀI

### 2.1. Phân tích hệ thống.

#### 2.1.1. Thiết kế use case.



Hình 2.1. Use case tổng quát.

### 2.1.2. Yêu cầu chức năng.

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

### 2.1.3. Yêu cầu phi chức năng.

Ứng dụng không quá 8GB.

### 2.1.4. Đặc tả use case.

Bảng 2.1 Use case Submit.

<b>Tên Use case:</b> Submit	
<b>Actor:</b> Student	
<b>Mô tả:</b> Use case cho phép actor thực hiện chọn file và gửi file bài làm lên hệ thống chấm bài	
<b>Điều kiện trước:</b> Actor đã đăng nhập được vào hệ thống	
<b>Điều kiện sau:</b> Hiển thị thông tin kết quả các lần nộp bài trước đó	
<b>Luồng sự kiện chính:</b>	
<b>Actor</b>	<b>Hệ thống</b>
1.Student chọn chức năng chọn file bài làm.	2. Lấy danh sách thư mục và hiển thị giao diện cho chọn file.

3. Student 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.
<b>Luồng sự kiện phụ:</b>	
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).

<b>Tên Use case:</b> View Result	
<b>Actor:</b> Student	
<b>Chức năng:</b> Use case cho phép actor xem thông tin kết quả các lần nộp bài.	
<b>Điều kiện trước:</b> Actor đã đăng nhập được vào hệ thống	
<b>Điều kiện sau:</b> Hiển thị thông tin kết quả các lần nộp bài.	
<b>Luồng sự kiện chính:</b>	
<b>Actor</b>	<b>Hệ thống</b>
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.

<b>Tên Use case:</b> View ScoreBoard	
<b>Actor:</b> Student	
<b>Chức năng:</b> Use case cho phép actor xem scoreboard	
<b>Điều kiện trước:</b> Actor đã đăng nhập được vào hệ thống	
<b>Điều kiện sau:</b> Hiển thị thông tin scoreboard	
<b>Luồng sự kiện chính:</b>	
<b>Actor</b>	<b>Hệ thống</b>
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.

<b>Tên Use case:</b> Login
<b>Actor:</b> Student
<b>Chức năng:</b> Use case cho phép actor đăng nhập vào hệ thống
<b>Điều kiện trước:</b> Actor truy cập vào web của hệ thống
<b>Điều kiện sau:</b> Hiển thị thông tin contest, các chức năng View Result, View ScoreBoard và Submit, Logout

<b>Luồng sự kiện chính:</b>	
<b>Actor</b>	<b>Hệ thống</b>
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.
<b>Luồng sự kiện phụ:</b>	
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.

<b>Tên Use case:</b> Logout	
<b>Actor:</b> Student	
<b>Chức năng:</b> Use case cho phép actor đăng xuất khỏi hệ thống	
<b>Điều kiện trước:</b> Actor đã đăng nhập được vào hệ thống	
<b>Điều kiện sau:</b> Hiển thị chức năng Login	
<b>Luồng sự kiện chính:</b>	
<b>Actor</b>	<b>Hệ thống</b>
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.

<b>Tên Use case:</b> Create Contest	
<b>Actor:</b> Teacher	
<b>Chức năng:</b> Use case cho phép actor tạo contest	
<b>Điều kiện trước:</b> Actor đã khởi động ứng dụng	
<b>Điều kiện sau:</b> Hiện thị contest đã được tạo	
<b>Luồng sự kiện chính:</b>	
<b>Actor</b>	<b>Hệ thống</b>
1. Teacher chọn chức năng Create Contest	2. Hiện thị form tạo contest
3. Teacher điền thông tin contest, chọn problem hoặc chọn chức năng Create Problem và xác nhận.	4. Hiện thị thông báo và thông tin contest vừa tạo.
<b>Luồng sự kiện phụ:</b>	
3.1.1. Teacher chọn chức năng Create Problem.	3.1.2. Hiện 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. Hiện 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.

<b>Tên Use case:</b> Create Problem	
<b>Actor:</b> Teacher	
<b>Chức năng:</b> Use case cho phép actor tạo problem	
<b>Điều kiện trước:</b> Actor đã khởi động ứng dụng	
<b>Điều kiện sau:</b> Hiển thị list problem	
<b>Luồng sự kiện chính:</b>	
<b>Actor</b>	<b>Hệ thống</b>
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
<b>Luồng sự kiện phụ:</b>	
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).

<b>Tên Use case:</b> View Result	
<b>Actor:</b> Teacher	
<b>Chức năng:</b> Use case cho phép actor xem thông tin các lần nộp bài và thông tin chi tiết của lần nộp đó.	
<b>Điều kiện trước:</b> Actor đã khởi động ứng dụng và đã start contest.	
<b>Điều kiện sau:</b>	
<b>Luồng sự kiện chính:</b>	
<b>Actor</b>	<b>Hệ thống</b>
1. Teacher chọn chức năng View Result.	2. Lấy danh sách các lần submit trước đó và hiển thị.
<b>Luồng sự kiện phụ:</b>	
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.

<b>Tên Use case:</b> Get Result	
<b>Actor:</b> Teacher	
<b>Chức năng:</b> Use case cho phép actor lấy danh sách kết quả contest tại thời điểm hiện tại.	
<b>Điều kiện trước:</b> Actor đã khởi động ứng dụng và đã start contest.	
<b>Điều kiện sau:</b> Hiển thị bảng kết quả	
<b>Luồng sự kiện chính:</b>	
<b>Actor</b>	<b>Hệ thống</b>
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.

<b>Tên Use case:</b> View List Problem
<b>Actor:</b> Teacher
<b>Chức năng:</b> Use case cho phép actor xem danh sách các problem.
<b>Điều kiện trước:</b> Actor đã khởi động ứng dụng.
<b>Điều kiện sau:</b> Hiển thị danh sách và thông tin problem.

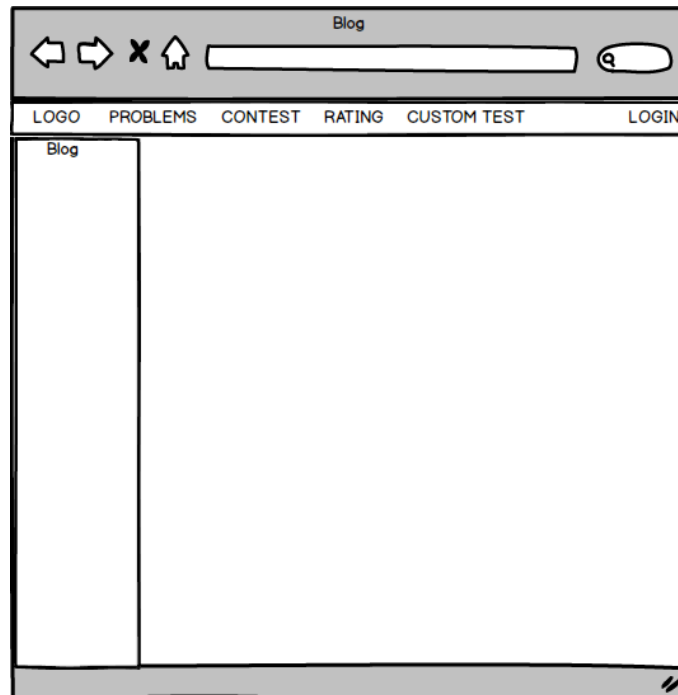
<b>Luồng sự kiện chính:</b>	
<b>Actor</b>	<b>Hệ thống</b>
1. Teacher chọn chức năng View List Problem.	2. Lấy danh sách problem và hiển thị.
<b>Luồng sự kiện phụ:</b>	
1.1. Teacher chọn problem trong danh sách.	1.2. Lấy thông tin problem và hiển thị.

Bảng 2.11 Use case Load Contest.

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

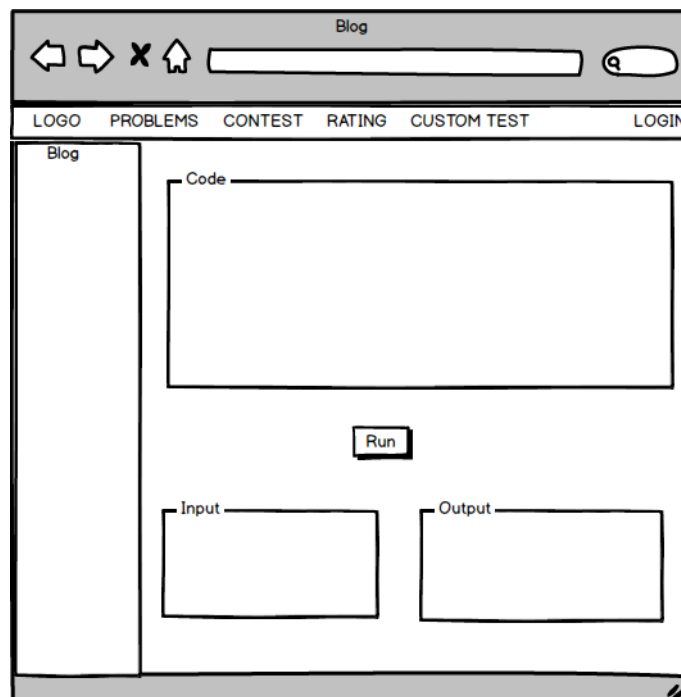
## 2.2. Thiết kế giao diện.

### 2.2.1. Trang chủ.



Hình 2.2. Giao diện Mocup – Home.

### 2.2.2. Customtest.



Hình 2.3. Giao diện mocup – Customtest.

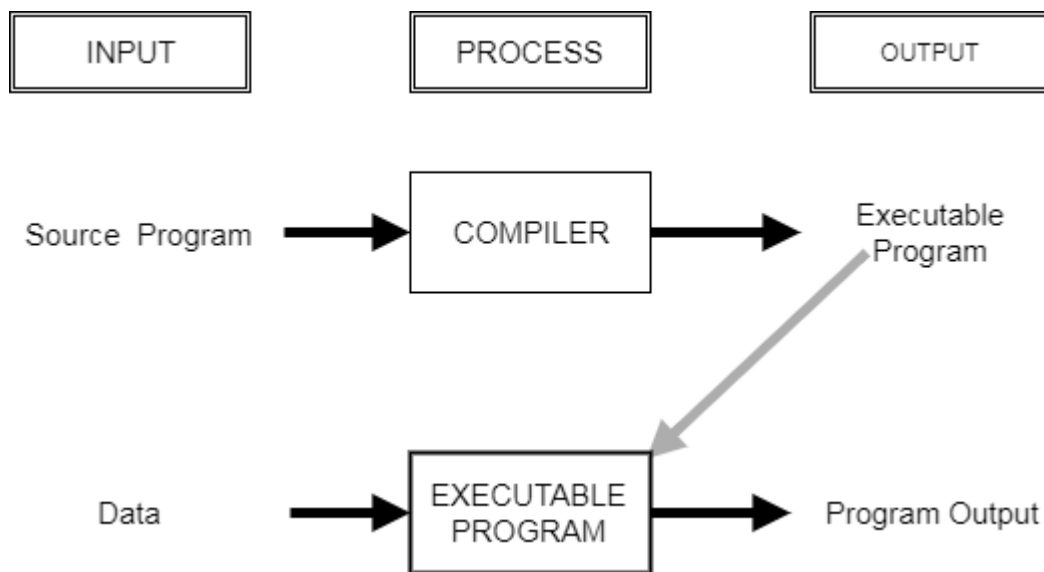
### 2.3. Hoạt động của ứng dụng.

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.

### **Chương 3: TRIỂN KHAI ỨNG DỤNG TRÊN AMAZON WEB SERVICES**

#### **3.1. Amazon Web Services (AWS).**

##### **3.1.1. 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.

##### **3.1.2. 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 sở dữ liệu NoSQL.

#### **3.2. Triển khai trên EC2.**

##### **3.2.1. Tạo tài khoản AWS.**

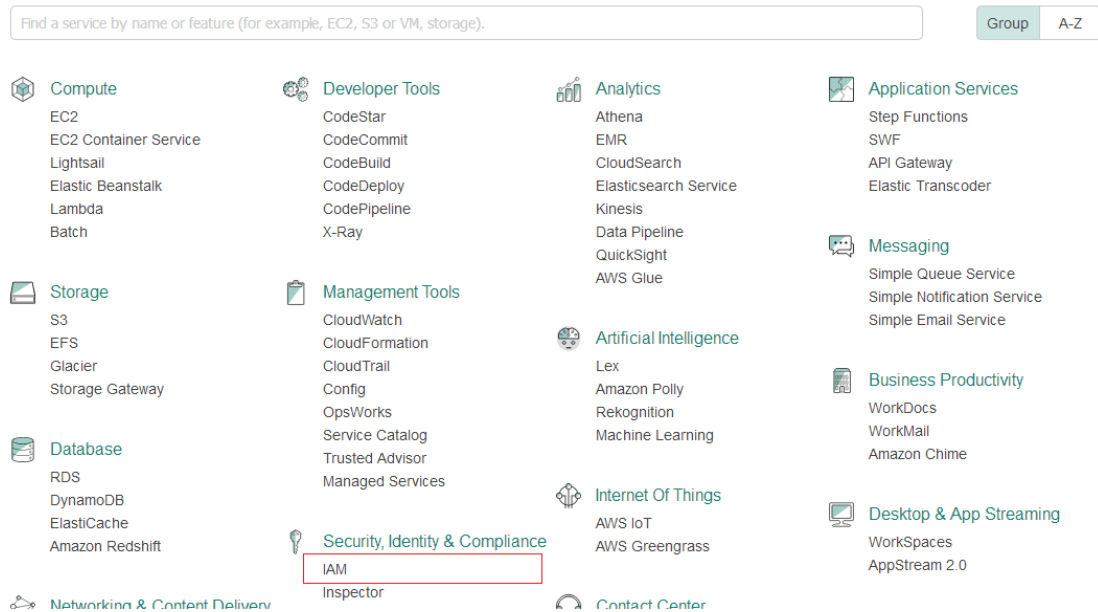
Để sử dụng các dịch vụ của AWS ta cần đăng ký tài khoản. Truy cập vào trang chủ <http://aws.amazon.com/> chọn Sign In to The Console để đăng ký tài khoản. Tài khoản đó được gọi là root account. Từ root account có thể tạo thêm nhiều người dùng khác với các quyền khác nhau.

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 chăm bài của nhóm sử dụng các dịch vụ có sẵn miễn phí để triển khai.

##### **3.2.2. 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.



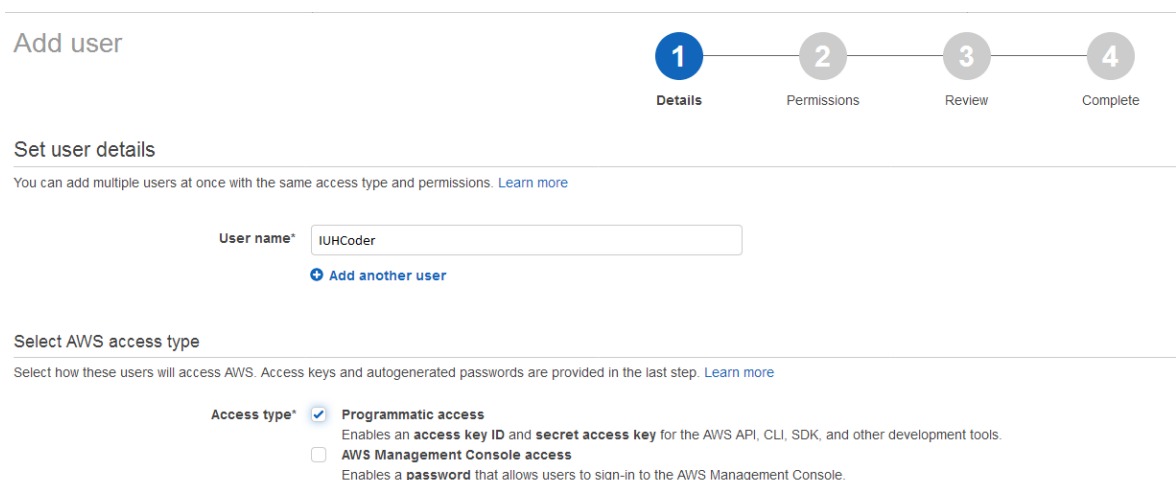
Hình 3.1. Cửa sổ dịch vụ của AWS.

Bước 2: Chọn User và Add user.



Hình 3.2. Thêm người dùng.

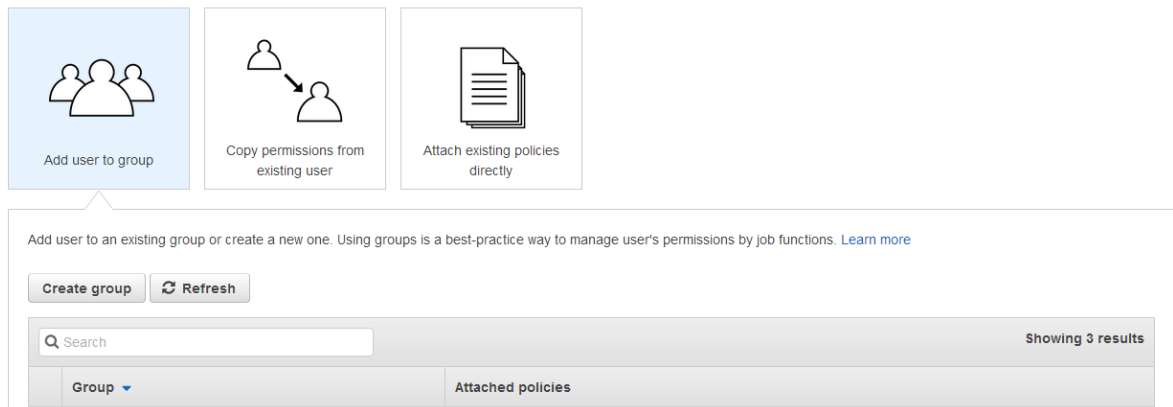
Bước 3: Đặt tên người dùng, chọn phương thức truy cập và chọn Next.



Hình 3.3. Điền thông tin người dùng.

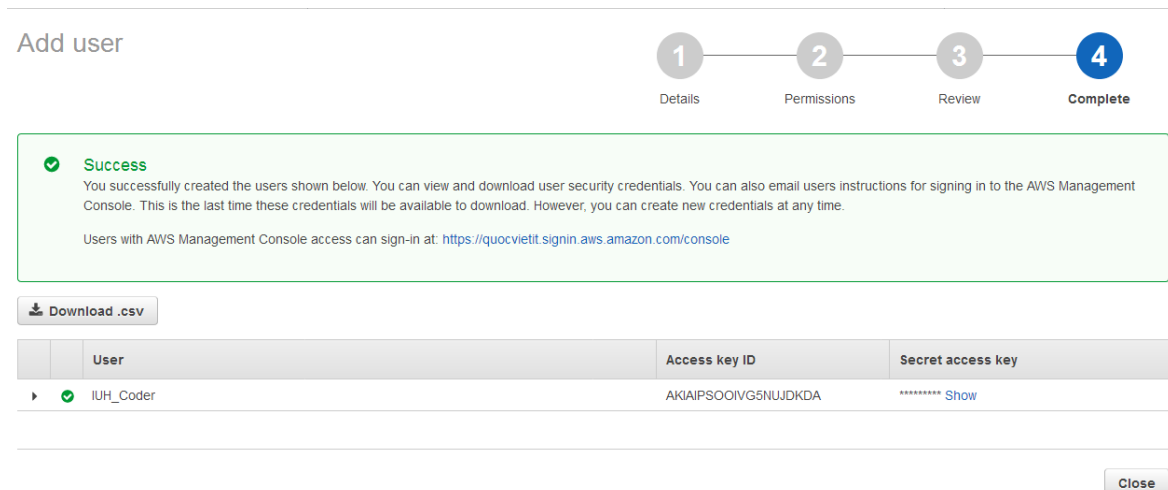
Bước 4: Cấp quyền cho người dùng, ở đây ta có thể tạo nhóm người dùng hoặc cấp quyền trực tiếp. Ở đây sử dụng dịch vụ EC2 và DynamoDB nên ta cấp quyền AmazonEC2FullAccess và AmazonDynamoDBFullAccess. Sau khi cấp quyền ta chọn Next để xem lại thông tin người dùng và chọn Next để đến tạo người dùng.

Set permissions for IUH\_Coder



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.

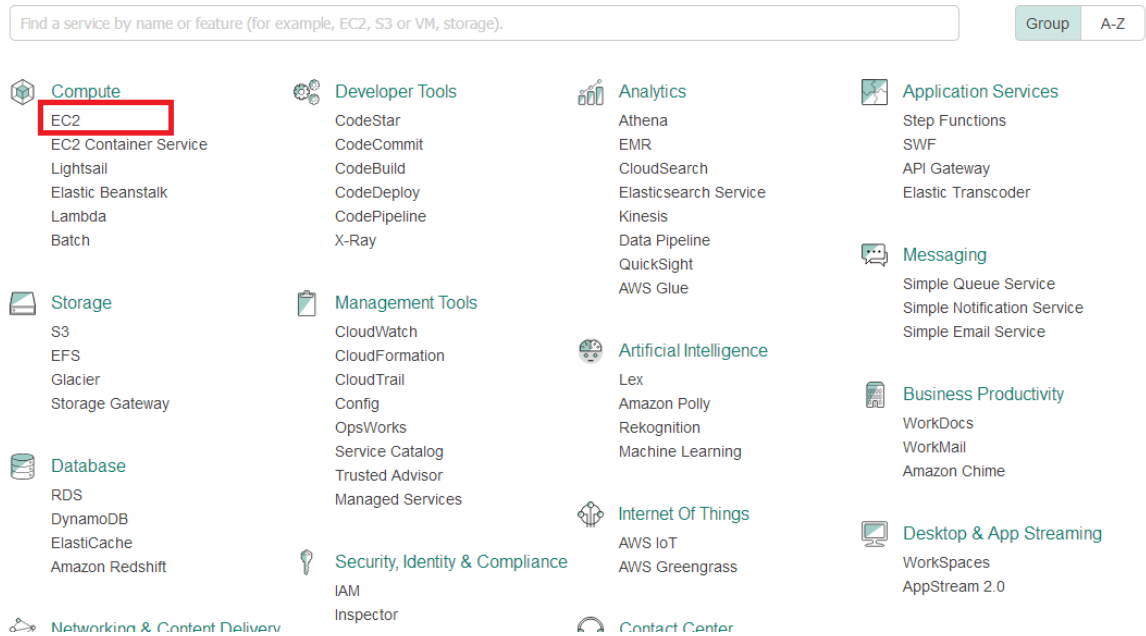
Secret Access Key chỉ cấp một lần khi tạo người dùng nên nếu không có Secret Access Key phải tạo lại một Access Key ID mới.

### 3.2.3. Tạo máy chủ EC2.



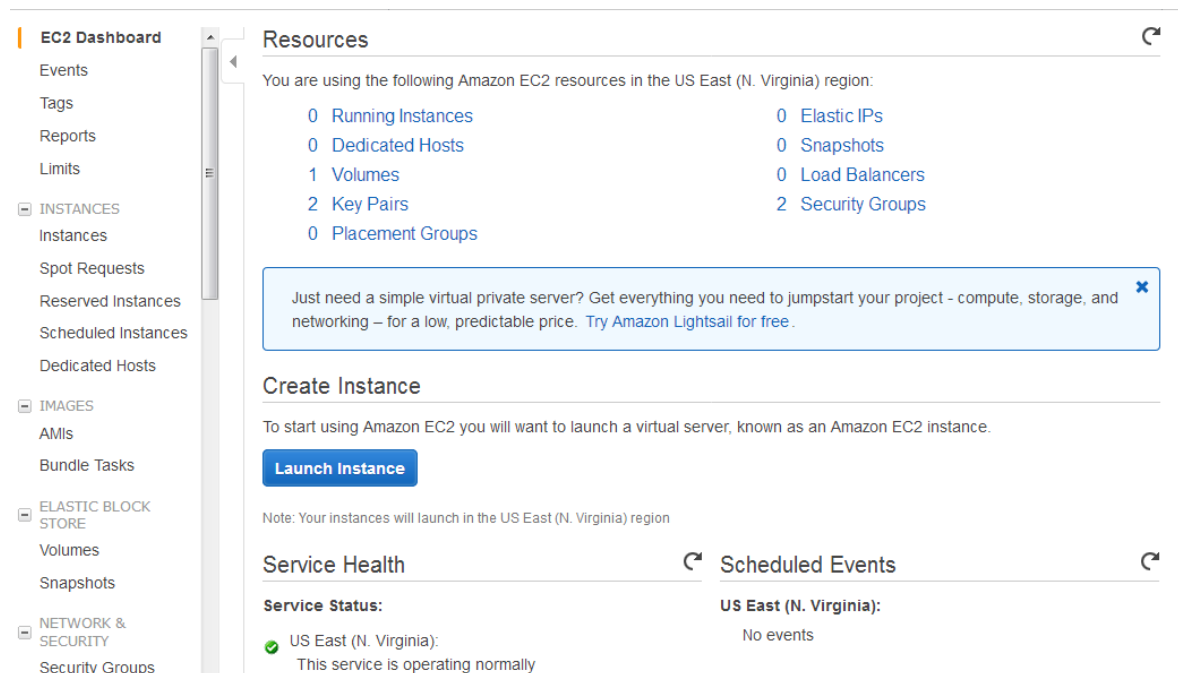
Sau khi có người dùng ta đăng nhập tài khoản người dùng để tạo máy chủ EC2. Máy chủ dùng để chạy ứng dụng sử dụng hệ điều hành Ubuntu Server 16.04 LTS.

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



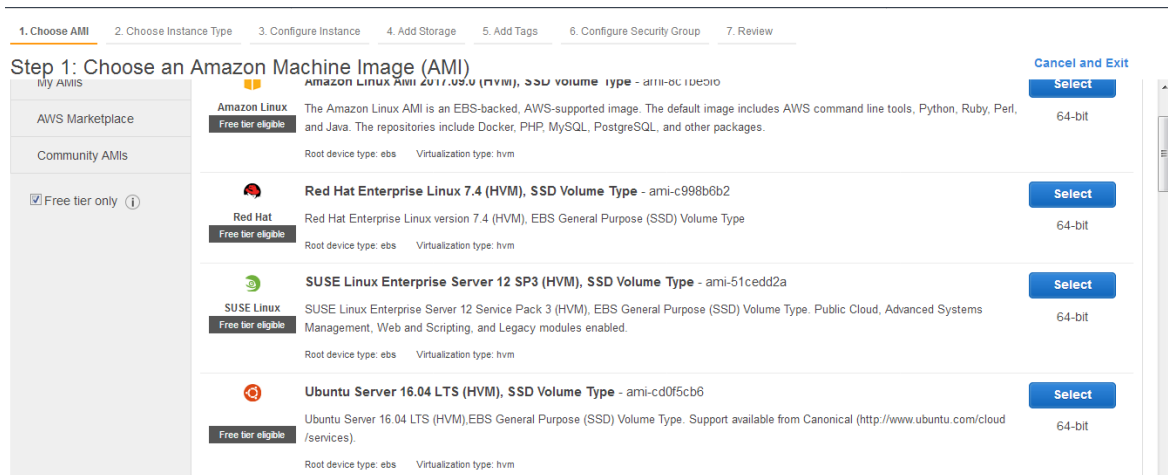
Hình 3.6. Cửa sổ dịch vụ của AWS.

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



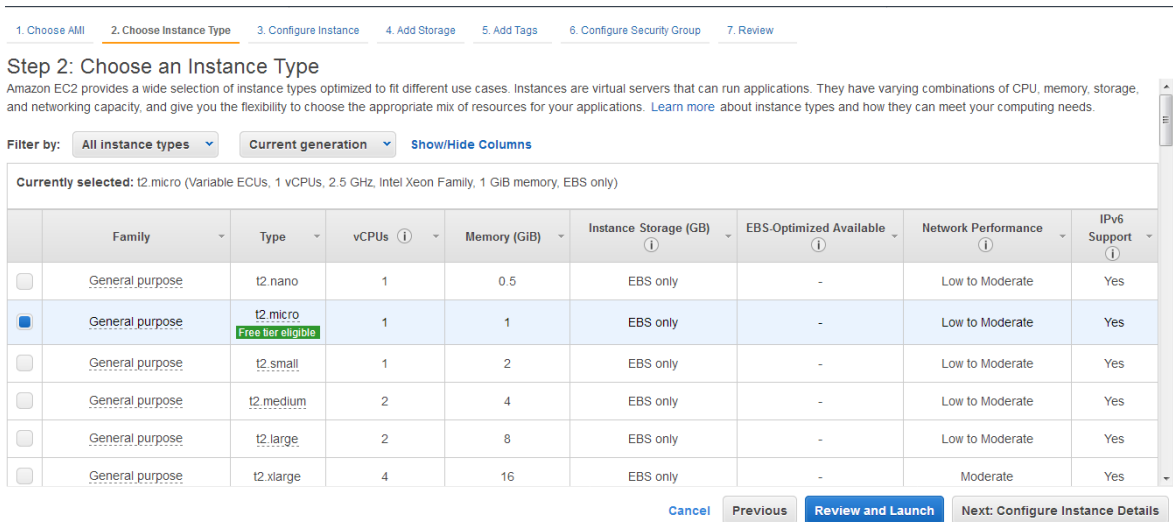
Hình 3.7. Bảng điều khiển EC2.

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,...

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances  Launch into Auto Scaling Group

Purchasing option ☐ Request Spot instances

Network  Create new VPC

Subnet  Create new subnet

Auto-assign Public IP

IAM role  Create new IAM role

Shutdown behavior

Enable termination protection ☐ Protect against accidental termination

Monitoring ☐ Enable CloudWatch detailed monitoring  
Additional charges apply.

Cancel Previous Review and Launch Next: Add Storage

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.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0cfc17b071e696816	8	General Purpose SSD (GP2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

Add New Volume

Hình 3.11. Thêm dung lượng bộ nhớ.

Bước 7: Gắn nhãn cho máy chủ.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

### Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes
This resource currently has no tags			

Choose the Add tag button or click to add a Name tag.  
Make sure your IAM policy includes permissions to create tags.

Add Tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

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.

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

**Step 6: Configure Security Group**

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
HTTP	TCP	80	Custom 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop
HTTPS	TCP	443	Custom 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop

[Add Rule](#)

Hình 3.13. Cấu hình bảo mật.

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

**Ubuntu Server 16.04 LTS (HVM), SSD Volume Type - ami-cd0f5cb6**

Free tier eligible

Root Device Type: ebs Virtualization type: hvm

▼ Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

▼ Security Groups [Edit security groups](#)

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.

**Select an existing key pair or create a new key pair** ✕

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. [Learn more about removing existing key pairs from a public AMI.](#)

Create a new key pair ▼

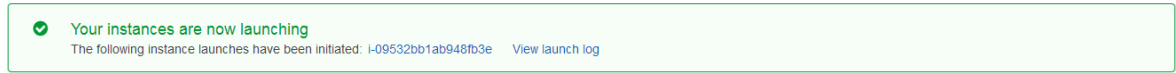
**Key pair name**

[Download Key Pair](#)

Hình 3.15. Lựa chọn keypair cho máy chủ.

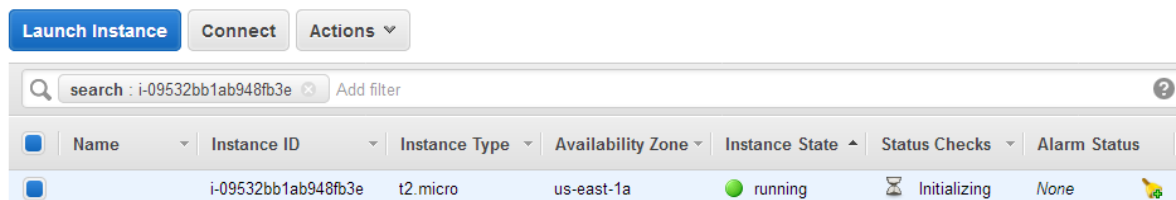
Bước 11: Quá trình khởi tạo máy chủ mất khoảng vài phút. Sau khi tạo thành công ta sẽ nhận được thông báo kèm id của máy chủ.

Launch Status



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.

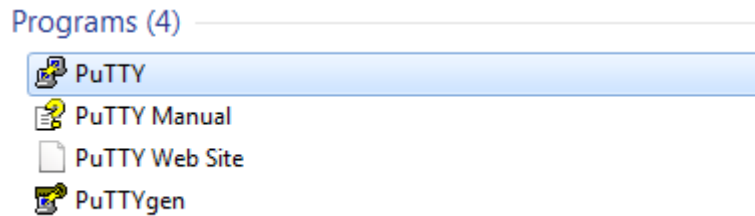
EC2 Instance được tạo thành công sẽ có private ip và public ip. Private ip là ip được cấp cho instance đến khi nó bị xóa còn public ip được cấp khi instance chạy nếu như ta tắt đi và mở lại thì public ip sẽ thay đổi. Một số thông tin khác như vùng, trạng thái, ổ cứng,... được mô tả trong hình bên dưới.

Instance ID	i-09532bb1ab948fb3e	Public DNS (IPv4)	ec2-54-227-191-4.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	54.227.191.4
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-23-5.ec2.internal
Availability zone	us-east-1a	Private IPs	172.31.23.5
Security groups	launch-wizard-1. view inbound rules	Secondary private IPs	
Scheduled events	No scheduled events	VPC ID	vpc-3a576043
AMI ID	ubuntu/images/hvm-ssd/ubuntu-xenial-16.04-amd64-server-20170721 (ami-cd0f5cb6)	Subnet ID	subnet-29091961
Platform	-	Network interfaces	eth0
IAM role	-	Source/dest. check	True
Key pair name	IUHCoder	EBS-optimized	False
Owner	038784618416	Root device type	ebs
Launch time	October 9, 2017 at 10:10:55 AM UTC+7 (less than one hour)	Root device	/dev/sda1
Termination protection	False	Block devices	/dev/sda1
Lifecycle	normal		

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

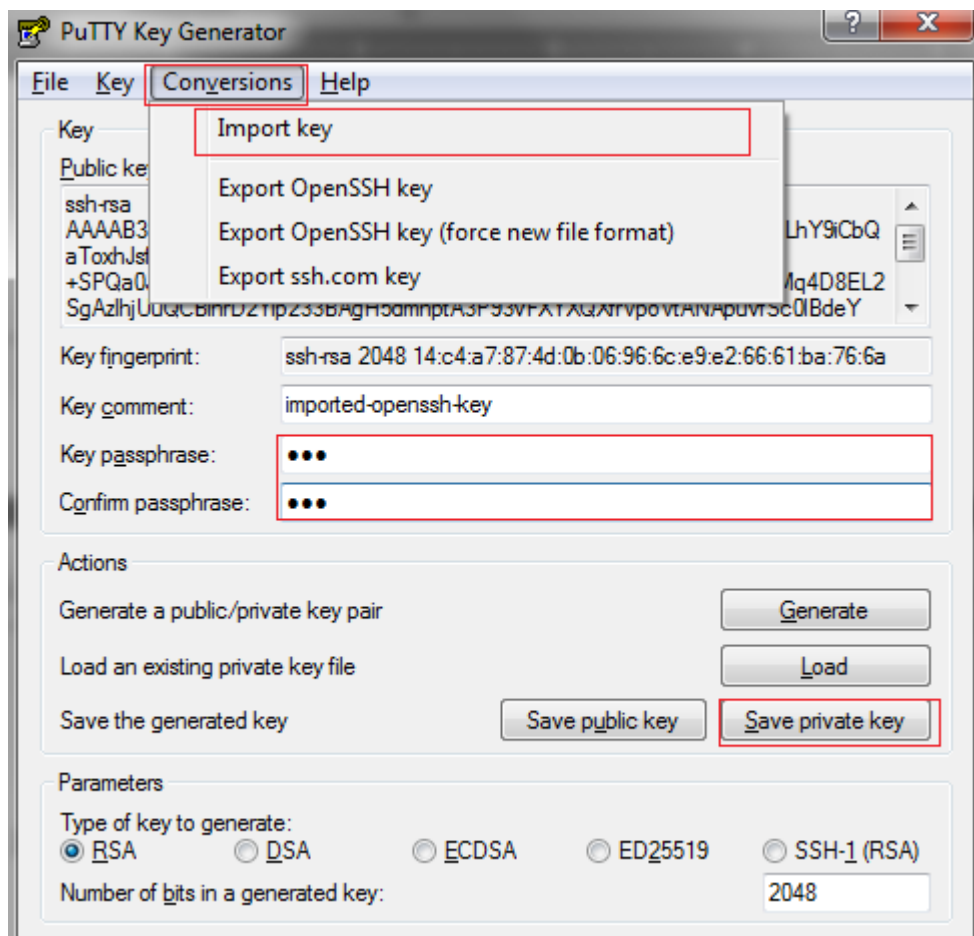
### 3.2.4. Kết nối EC2.

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.



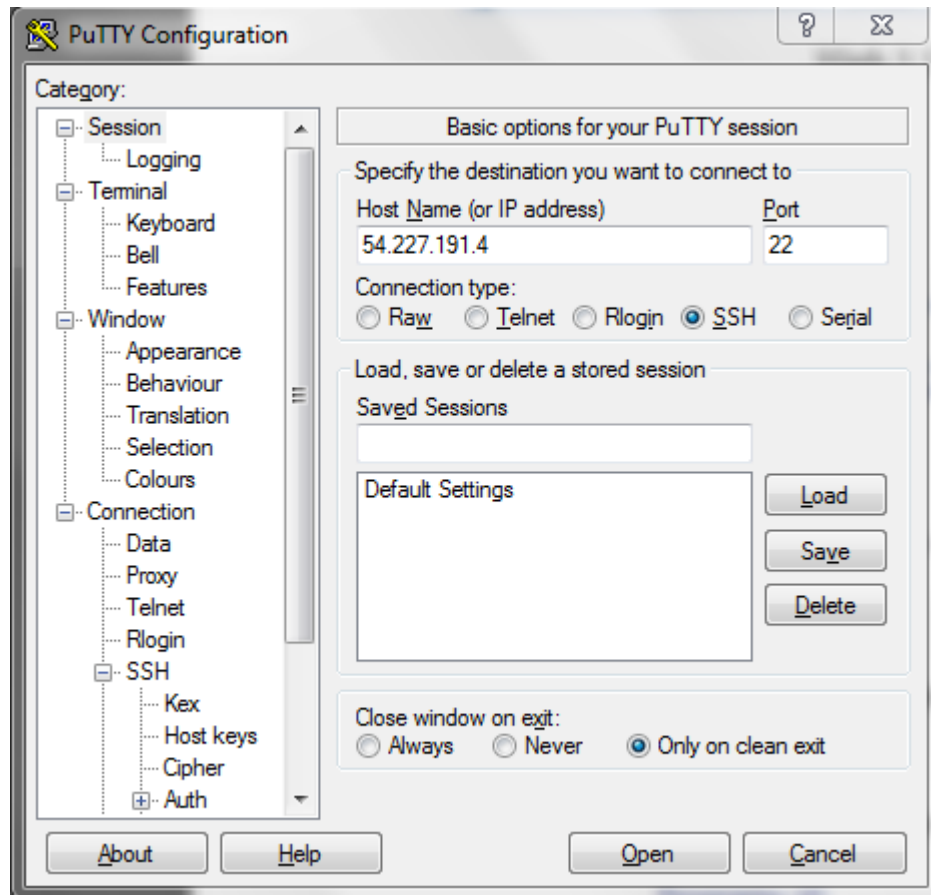
Hình 3.19. Phần mềm putty và puttygen.

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 của EC2 Instance và port của instance.



Hình 3.21. Phần mềm Putty.

Bước 3: Chọn Connection → SSH → Auth → Browse.. chọn 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.

```
Using username "ubuntu".
Authenticating with public key "imported-openssh-key"
Passphrase for key "imported-openssh-key":
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-1035-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

70 packages can be updated.
0 updates are security updates.

Last login: Tue Oct  3 07:29:14 2017 from 1.52.108.90
ubuntu@ip-172-31-26-131:~$
```

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

### 3.2.5. 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 virtualenv
```

```
$ 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 cài trình biên dịch cho C++ :

```
$ sudo apt-get install g++
```

### 3.3. 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
```

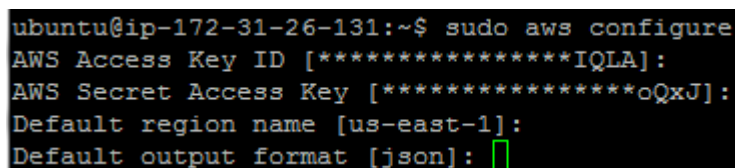
### 3.4. Kết nối với database (DynamoDB):

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
```



```
ubuntu@ip-172-31-26-131:~$ sudo aws configure
AWS Access Key ID [*****IQLA]:
AWS Secret Access Key [*****oQxJ]:
Default region name [us-east-1]:
Default output format [json]:
```

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.

### 3.5. 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.

```
app.run(debug = True, host=os.getenv('IP', '0.0.0.0'),port=int(os.getenv('PORT', 80)))
```

Hình 3.24. Cấu hình ip và port .

### 3.6. Kiểm thử.

Chạy thử chương trình:

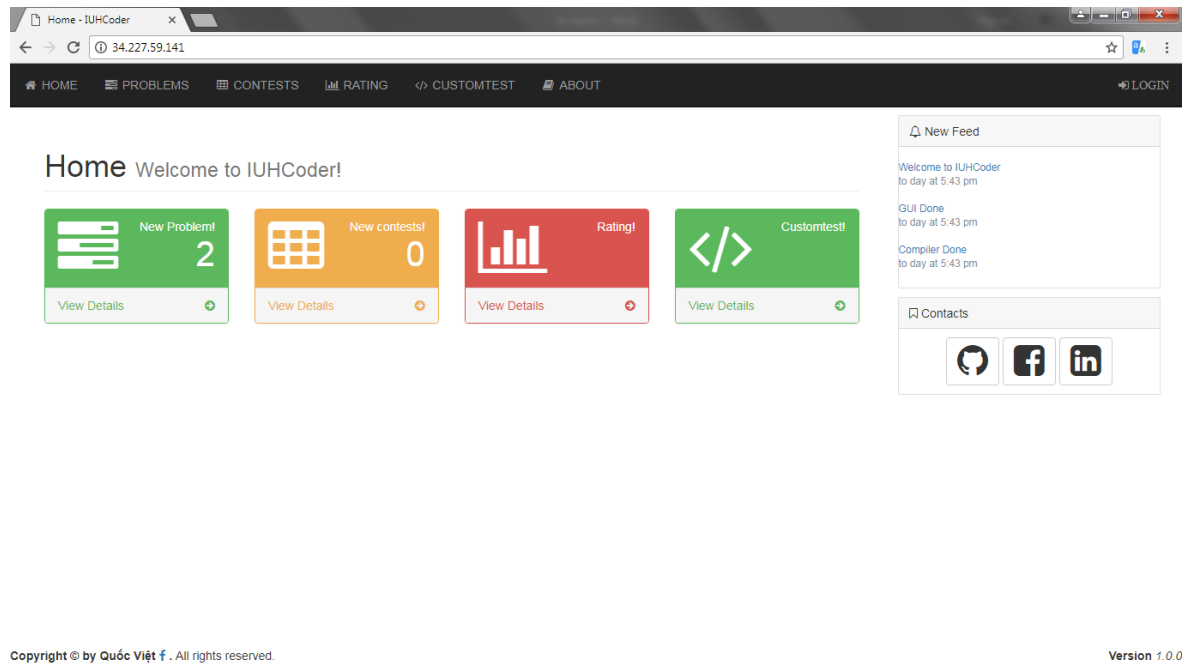
```
$ sudo cd IUHCoder
```

```
$ sudo python __main__.py
```

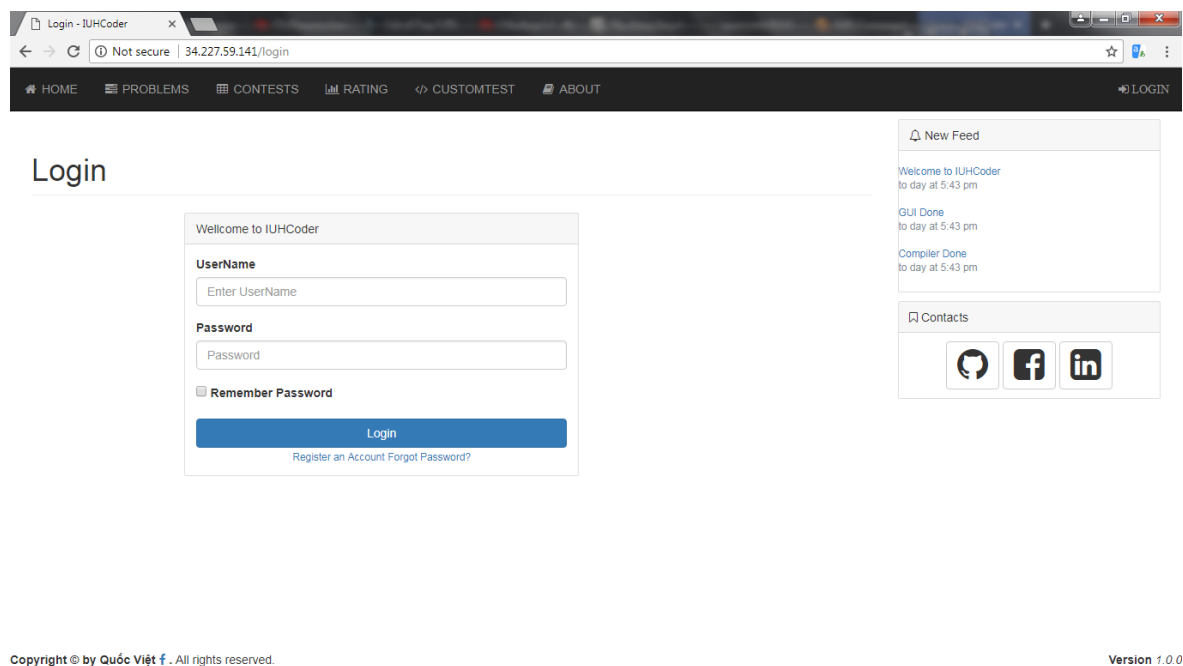
```
ubuntu@ip-172-31-26-131:~/IUHCoder$ sudo python __main__.py
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 387-263-384
116.108.58.236 - - [09/Oct/2017 12:43:09] "GET / HTTP/1.1" 200 -
116.108.58.236 - - [09/Oct/2017 12:43:10] "GET /static/css/bootstrap.min.css HTTP/1.1" 200 -
116.108.58.236 - - [09/Oct/2017 12:43:10] "GET /static/css/style.css HTTP/1.1" 200 -
116.108.58.236 - - [09/Oct/2017 12:43:10] "GET /static/css/plugins/morris.css HTTP/1.1" 200 -
116.108.58.236 - - [09/Oct/2017 12:43:10] "GET /static/font-awesome/css/font-awesome.min.css HTTP/1.1" 200 -
116.108.58.236 - - [09/Oct/2017 12:43:10] "GET /static/css/table.css HTTP/1.1" 200 -
116.108.58.236 - - [09/Oct/2017 12:43:10] "GET /static/js/jquery.js HTTP/1.1" 200 -
116.108.58.236 - - [09/Oct/2017 12:43:10] "GET /static/js/bootstrap.js HTTP/1.1" 200 -
116.108.58.236 - - [09/Oct/2017 12:43:10] "GET /static/js/plugins/morris/raphael.min.js HTTP/1.1" 200 -
116.108.58.236 - - [09/Oct/2017 12:43:11] "GET /static/js/plugins/morris/morris.js HTTP/1.1" 200 -
```

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.

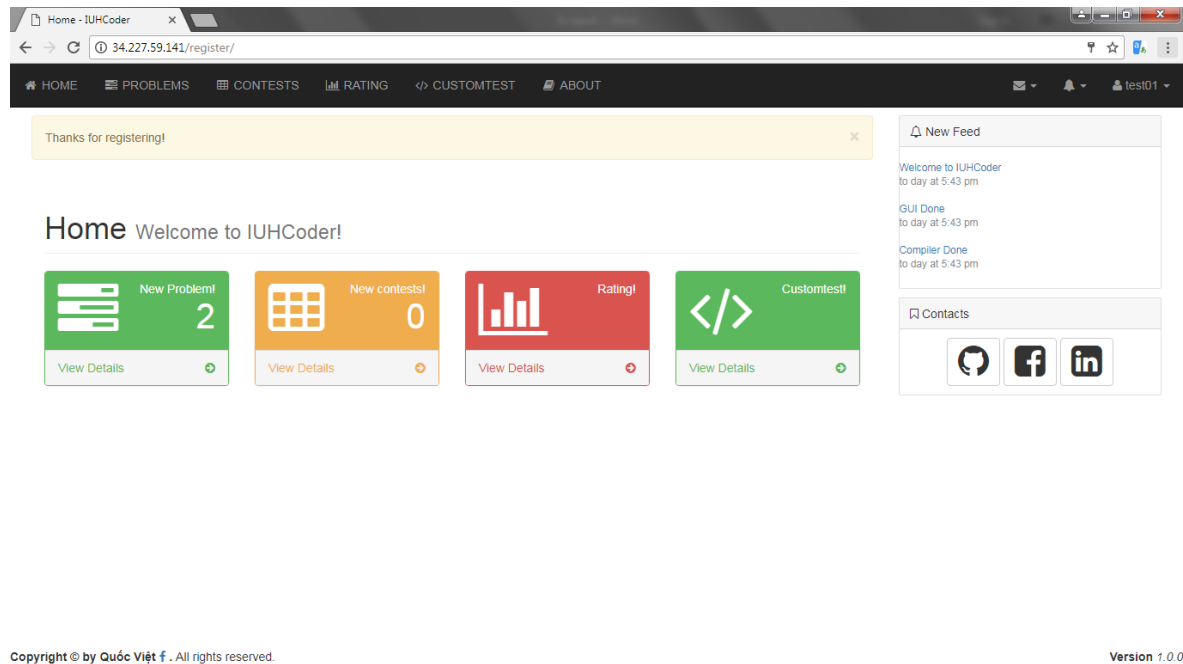


Hình 3.26. Giao diện trang chủ.



Hình 3.27. Giao diện đăng nhập.

Tại giao diện đăng nhập ta sẽ đăng ký một tài khoản tên là test01 và mật khẩu là 123456789 để kiểm tra môi trường trình biên dịch của ngôn ngữ C/C++.



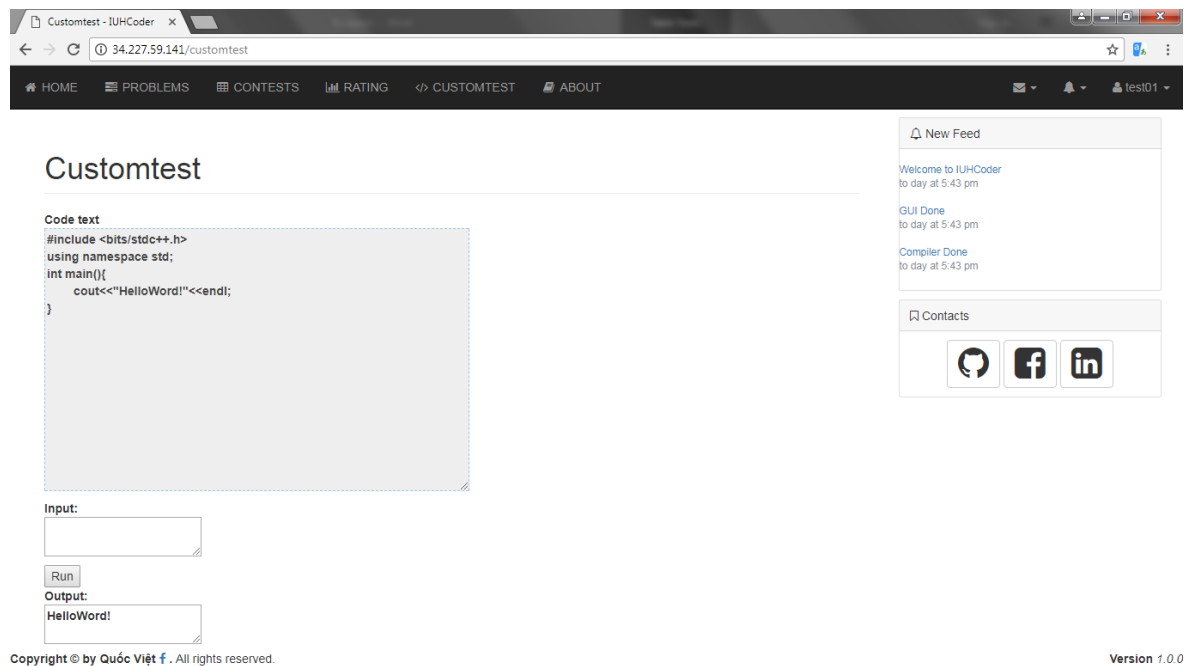
Hình 3.28. Đăng ký thành công tài khoản test01.

Chạy thử các đoạn mã C/C++ đơn giản với kết quả đã được chạy trên máy tính:

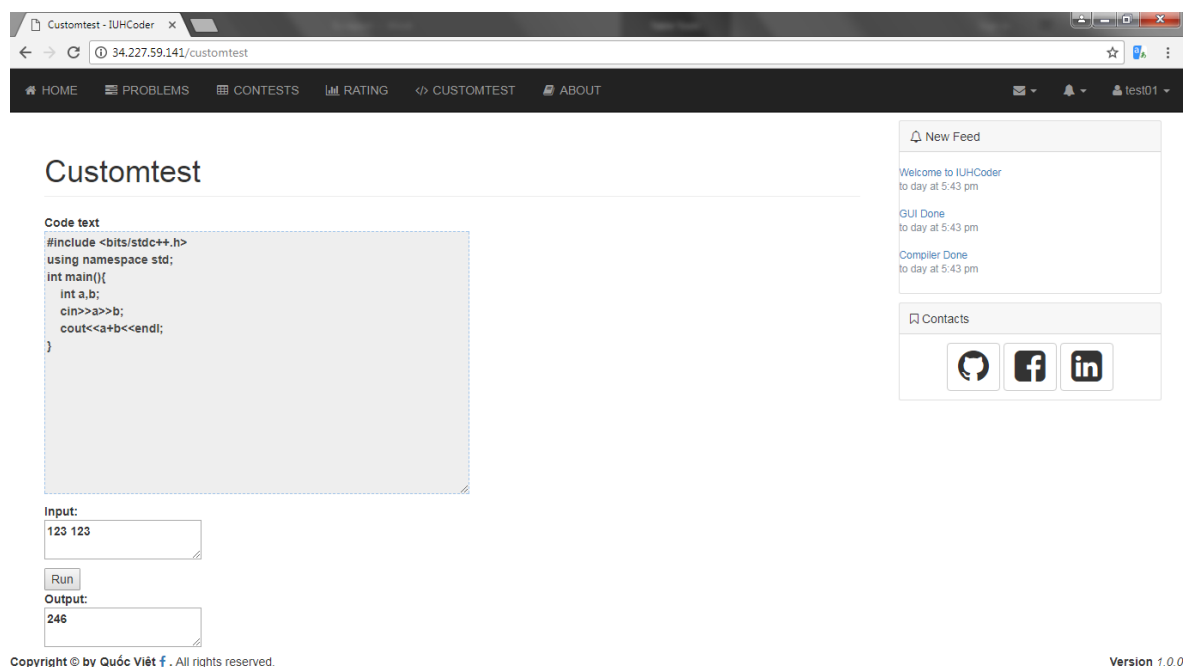
STT	Mã nguồn	Đầu vào	Đầu ra
1	<pre>#include &lt;bits/stdc++.h&gt; using namespace std; int main(){     cout&lt;&lt;"HelloWord!"&lt;&lt;endl; }</pre>		HelloWord!
2	<pre>#include &lt;bits/stdc++.h&gt; using namespace std; int main(){     int a,b;     cin&gt;&gt;a&gt;&gt;b;     cout&lt;&lt;a+b&lt;&lt;endl;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

Bảng 3.1. Các đoạn mã C/C++ mẫu.

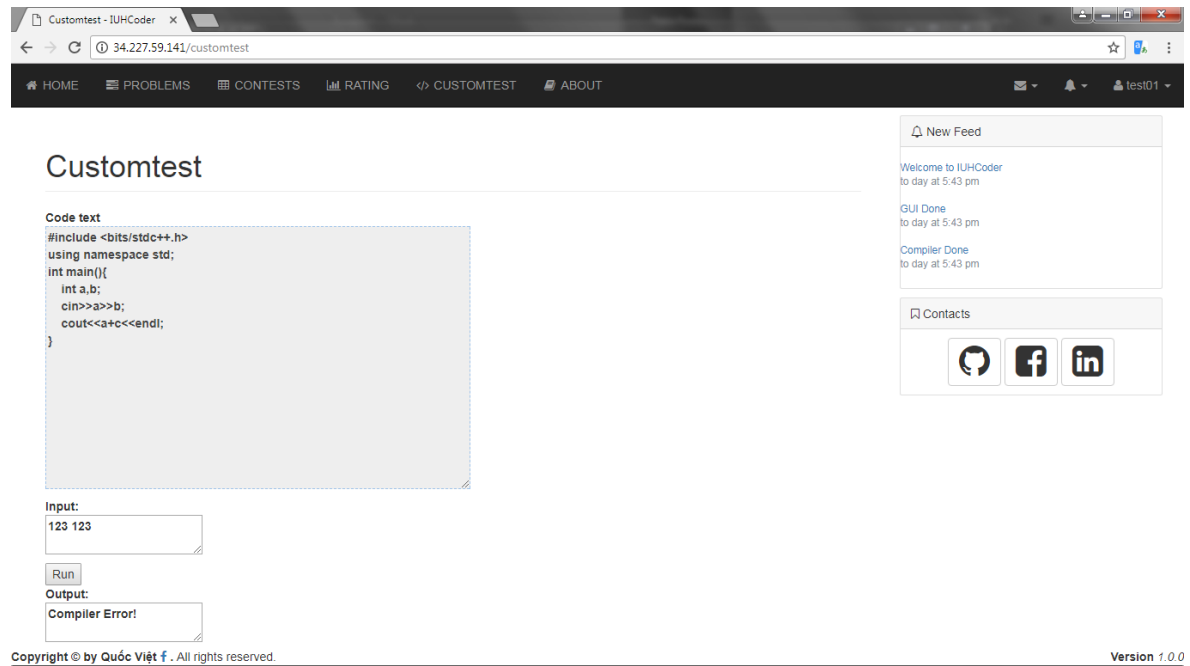
Chạy trên ứng dụng:



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 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.

## **Chương 4: KẾT LUẬN**

### **4.1. Kết quả triển khai ứng dụng.**

Sau khi kiểm thử ứng dụng đã chạy ổn định.

#### **4.1.1. Ưu điểm.**

- Ứng dụng đã tạo được môi trường chạy ngôn ngữ C/C++.
- Thời gian chấm bài nhanh.

#### **4.1.2. Nhược điểm.**

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

#### **4.1.3. Lỗi.**

(cần người test để tìm lỗi).

### **4.2. Hướng phát triển.**

Hướng phát triển của ứng dụng là tạo ra các vấn đề để các thành viên giải quyết, học tập thuật toán, kỹ năng lập trình cơ bản. Tạo trang quản lý cho phép giáo viên quản lý học sinh, tạo các kỳ thi thay thế cho các kỳ thi lập trình trước đây và trang thảo luận, góp ý,...

## TÀI LIỆU THAM KHẢO