TRƯỜNG ĐẠI HỌC ĐÀ LẠT KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỒ ÁN



Đề tài : Tìm hiểu ngôn ngữ lập trình Python

Đà Lạt, 18 tháng 11 năm 2018

Giáo viên hướng dẫn: Nguyễn Thị Lương

Sinh viên thực hiên:

* 1510289- Nguyễn Văn Vương
* 1512944- Ma Xuân Thoại

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

LỜI CẢM ƠN

Trong thời gian làm đồ án tốt , chúng em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô và bạn bè.

Chúng em xin gửi lời cảm ơn chân thành đến ThS Nguyễn Thị Lương, giảng viên Bộ môn Kỹ Thuật Phần Mềm - Trường ĐẠI HỌC ĐÀ LẠT người đã tận tình hướng dẫn, chỉ bảo chúng em trong suốt quá trình làm khoá luận.

Chúng em cũng xin chân thành cảm ơn các thầy cô giáo trong trường ĐẠI HỌC ĐÀ LẠT nói chung, các thầy cô trong Bộ môn Kỹ Thuật Phần Mềm nói riêng đã dạy dỗ kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp chúng em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ chúng em trong suốt quá trình học tập.

Cuối cùng, chúng em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên trong suốt quá trình học tập và hoàn thành đồ án.

**ĐỀ CƯƠNG THỰC HIỆN ĐỒ ÁN**

Tên đề tài: Tìm hiểu ngôn ngữ lập trình Python

Sinh viên thực hiện:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** |  | **MSSV** | **Lớp** | **Email liên hệ** |
| 1 | Nguyễn Văn Vương | | 1510289 | CTK39 | Nguyenvanvuong972@gmail.com |
| 2 | Ma Xuân Thoại | | 1512944 | CTK39 | thoaixuan97@gmail.com |

Giáo viên hướng dẫn: Nguyễn Thị Lương

Mục tiêu đề tài: Tìm hiểu ngôn ngữ lập trình Python

Nội dung đề tài: Tìm hiểu ngôn ngữ lập trình Python

Phần mềm và công cụ sử dung: Vistual Studio 2017

Dự kiến kết quả đạt được: Tìm hiểu và sử dụng thành thạo ngôn ngữ lập trình Python.

Tài liệu tham khảo: [www.howkteam.vn](http://www.howkteam.vn), <daynhauhoc.com>, <youtube.com> , <vi.wikipedia.org> …

*Đà Lạt, ngày ….. tháng ….. năm 2011*

Giáo viên hướng dẫn SV Thực hiện

(Ký tên) (Ký tên)

Mục lục  
  
[Chương 1: Tổng quan về đề tài 2](#_Toc531711034)

[**1.1** **Giới thiệu đề tài** 2](#_Toc531711035)

[**1.2 Mục tiêu đề tài** 3](#_Toc531711036)

[Chương 2: Ngôn ngữ lập trình Python và framework Django 3](#_Toc531711037)

[**2.1 Ngôn ngữ Python** 3](#_Toc531711038)

[**2.2 Tìm hiểu Framework Django** 5](#_Toc531711039)

[**2.3 Mục tiêu chính của Django** 6](#_Toc531711040)

[**2.4 Lập kế hoạch xây dựng website bằng Django** 7](#_Toc531711041)

[Chương 3: Tìm hiểu về cách sử dụngngôn ngữ lập trình Python 8](#_Toc531711042)

[**3.1 Khái niệm Python** 8](#_Toc531711043)

[**3.2 Cài đặt Python** 9](#_Toc531711044)

[**3.2.1 Download và cài đặt môi trường Python.** 9](#_Toc531711045)

[**3.2.2 Download và cài đặt Vistual Studio 2017** 10](#_Toc531711046)

[**3.3 Các kiến thức cơ bản về Python** 11](#_Toc531711047)

[**3.3.1 Chương trình đầu tiên Hello Word** 11](#_Toc531711048)

[**3.3.2 Chú thích trong Python** 12](#_Toc531711049)

[3.3.2.1 Cú pháp: 12](#_Toc531711050)

[Trường hợp đặc biệt 12](#_Toc531711051)

[**3.3.3 Kiểm tra kiểu dữ liệu của biến** 14](#_Toc531711052)

[**3.3.4 Một số kiểu dữ liệu cơ bản trong Python.** 15](#_Toc531711053)

[a) Kiểu số. 15](#_Toc531711054)

[**b) Kiểu chuỗi** 17](#_Toc531711055)

[**c) Kiểu danh sách ( list ).** 19](#_Toc531711056)

[**d) Kiểu dữ liệu Tuple.** 20](#_Toc531711057)

[**e) Kiểu dữ liệu Dic ( Dictionary )** 21](#_Toc531711058)

[**3.4 Các hàm xử lý file trong Python** 22](#_Toc531711059)

[**3.5 Hàm trong python** 24](#_Toc531711060)

[**3.6 Class trong python** 25](#_Toc531711061)

[**3.7 Câu lệnh rẽ nhánh trong python** 29](#_Toc531711062)

[**3.8 Các vòng lặp trong python** 0](#_Toc531711063)

[3.8.1 Vòng lặp while: 0](#_Toc531711064)

[3.8.2 Vòng lặp for 0](#_Toc531711065)

[**3.9 Nhập xuất trong python** 1](#_Toc531711066)

[3.9.1 Hàm xuất – print 1](#_Toc531711067)

[3.9.2 Hàm nhập – input: 2](#_Toc531711068)

[**3.10 Các kiến thức cơ bản về Framework Django: cú pháp, cài đặt môi trường, cấu trúc.** 3](#_Toc531711069)

[**3.10.1 Cài đặt Django** 3](#_Toc531711070)

[**3.10.2 Tạo dự án web django** 4](#_Toc531711071)

[**3.10.3 Khởi chạy dự án django** 4](#_Toc531711072)

[**3.10.4 Cấu trúc cơ bản của một website Django** 5](#_Toc531711073)

[Chương 4: Xây dựng các ứng dụng 7](#_Toc531711074)

[**4.1 Phân tích cơ sở dữ liệu** 7](#_Toc531711075)

[**4.2 Phân tích kiến trúc hệ thống** 11](#_Toc531711076)

[Giao diêṇ người dùng 11](#_Toc531711077)

[a) Sơ đồ Use case 11](#_Toc531711078)

[b) Tổng quan Use case 14](#_Toc531711079)

[**4.3** **Phân tích bố cục giao diện** 15](#_Toc531711080)

[**4.4** **Phân tích các chức năng của ứng dụng** 15](#_Toc531711081)

[4.4.1 Cách hoạt động của trang web. 15](#_Toc531711082)

[4.4.2 Danh sách actors. 15](#_Toc531711083)

[4.4.3 Danh sách Use case. 15](#_Toc531711084)

[**4.5** **Triển khai ứng dụng và kiểm thử** 17](#_Toc531711085)

[Chương 5: Tổng kết đề tài 18](#_Toc531711086)

[**5.1 Hướng phát triển** 18](#_Toc531711087)

[**5.2 Kết luận** 18](#_Toc531711088)

# Chương 1: Tổng quan về đề tài

* 1. **Giới thiệu đề tài**

Ngày nay, Công nghệ thông tin là một trong những ngành đang phát triển rất mạnh mẽ và có ảnh hưởng sâu rộng đến mọi mặt đời sống. Nó là nền tảng của nền kinh tế tri thức, là thước đo trình độ phát triển của một quốc gia.Vì vậy, việc đào tạo đội ngũ kỹ sư công nghệ thông tin có chất lượng đòi hỏi phải được chú trọng và đầu tư đúng mức.

Xã hội và kinh tế phát triển đòi hỏi công nghệ cũng phải phát triển. Công nghệ phát triển, con người ngày càng phát minh ra những thiết bị công nghệ số thông minh giúp đỡ con người về rất nhiều mặt trong cuộc sống. Trong thời đại ngày nay, đất nước ta đang hội nhập vào WTO cùng với sự phát triển nhanh chóng của công nghệ thông tin được áp dụng trên mọi lĩnh vực của cuộc sống như kinh tế, chính trị. Văn hóa xã hội, tất cả đều cần có công nghệ thông tin. Như hiện nay chúng ta đã biết lập trình và thiết kế Website có thể coi là ngành “hot” mà hiện nay chúng ta đang rất quan tâm.

Với việc xây dựng website áp dụng những ngôn ngữ lập trình khác nhau với những ưu nhược điểm khác nhau, tiếp cận người dùng nhanh chóng. Từ đó học hỏi và trau dồi thêm kiễn thức và kỹ năng của bản thân

Đó là lý do chúng em chọn đề tài “***Tìm hiểu ngôn ngữ Python***”.

## **Mục tiêu đề tài**

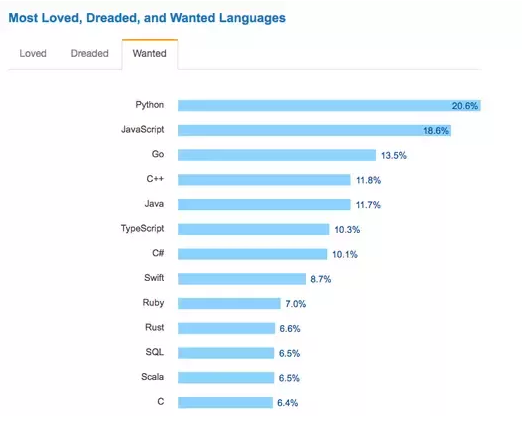
Đề tài giúp sinh viên tìm hiểu các kiến thức cơ bản về thiết kế và lập trình Web và các ngôn ngữ, công cụ, kỹ thuật như Python, Bootstrap, HTML/HTML5, CSS/CSS3, JavaScript và JQuery để xây dựng trang web cung cấp các tiện ích Internet. Thông qua đề tài, sinh viên nâng cao được các kỹ năng tìm kiếm, tổng hợp tài liệu; nghiên cứu các công nghệ, kỹ thuật mới; phát triển khả năng tư duy trong thiết kế Web; trau dồi và nâng cao khả năng trình bày, viết báo cáo, đọc hiểu tài liệu tiếng Anh.

# Chương 2: Ngôn ngữ lập trình Python và framework Django

* 1. **Ngôn ngữ Python**

Python là một ngôn ngữ lập trình thông dịch do Guido van Rossum tạo ra năm 1990. Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý. Python có rất nhiều đặc điểm nổi bật của Python như là:

* Hỗ trợ rất mạnh, thực hiện nhiều tính năng với ít code nhất.
* Python nghiêng mạnh về làm thư viện ( console,script …).
* Hỗ trợ mạnh internet of thing.
* Python khá giống ngôn ngữ lập tình Perl, Ruby, Schêm, Smalltalk, Tcl.
* Python được phát triển trong một dự án mã nguồn mở do một tổ chức phu lợi nhuận Python Software Foundation quản lý.
* Python được phát triển để chạy trên nền Unix. Nhưng theo thời gian đã có thể chạy tốt trên mọi mọi hệ điều hành từ MS- DOS đến MAC OS,OS/2, Window, Linux và một số hệ điều hành khác thuộc họ Unix.
* Python là ngôn ngữ cấp cao, hỗ trợ lập trình nhúng, sát ngôn ngữ tự nhiên, cấu trúc rõ ràng,thuận tiện cho người mới học lập trình, cho phép sử dụnng viết mã với số lần gõ tối thiểu.
* Python cũng là một trong những môn học phổ biến nhất thế giới, dễ đọc, dể hiểu.
* Thư viện phong phú ( <https://pypi.org/> ), cộng đồng sử dụng nhiều.
* Python hỗ trợ nhiều công việc:
  + - Đơn giản tốt hơn phức tạp.
    - Phức tạp tốt hơn rắc rối.
    - Dễ đọc, dễ học, dễ hiểu.
    - Hỗ trợ lập trình nhúng.
    - Hỗ trợ lập trình web: Youtube, Google, DropBox, Quora, Reddit, Intagram, Nasa, FireFox …..
    - Xử lý hình ảnh, video cực mạnh: nhận dạng, sửa đổi….
    - Hỗ trợ lập trình game, ứng dụng, Robot.
    - Bảo mật và mạng máy tính.



Hình 2. 1 Python đứng top 1 ngôn ngữ lập trình phổ biến 2017 ( nguồn: internet)

* 1. **Tìm hiểu Framework Django**

Django là một web framework miễn phí mã nguồn mở được viết bằng Python. Django sử dụng mô hình Model-View-Control (MVC). Django được phát triển bởi Django Software Foundation(DSF) – một tổ chức phi lợi nhuận độc lập.

Mục tiêu chính của Django là đơn giản hóa việc tạo các website phức tạp có sử dụng cơ sở dữ liệu. Django tập trung vào tính năng “có thể tái sử dụng” và “có thể tự chạy” của các component, tính năng phát triển nhanh, không làm lại những gì đã làm. Một số website phổ biến được xây dựng từ Django là Pinterest, Instagram, Mozilla, và Bitbucket.

**Ưu điểm của Django:**

* + Không còn các nỗi lo về các lỗi bảo mật thông thường như SQL Injection, cross-site scripting, cross-site request forgery hay clickjacking nữa. Django cũng cung cấp cả phương pháp để lưu mật khẩu an toàn.
  + Xây dựng CMS, hoặc Ecommerce Website, hay kể cả Social Network, Scientific Computing Platforms, tất cả đều có thể với Django.
  + Tự động tạo SQL tables và Admin Interface. Django sẽ làm thay lập trình viên khi bạn đã xác định được cấu trúc.
  1. **Mục tiêu chính của Django**

Đơn giản hóa việc tạo các website phức tạp có sử dụng cơ sở dữ liệu. Django tập trung vào tính năng “có thể tái sử dụng” và “có thể tự chạy” của các component, tính năng phát triển nhanh, không làm lại những gì đã làm. Một số website phổ biến được xây dựng từ Django là Pinterest, Instagram, Mozilla, và Bitbucket.

* + - Mặc dù có danh pháp riêng của nó, chẳng hạn như đặt tên các đối tượng có thể gọi tạo ra các phản hồi [HTTP](https://en.wikipedia.org/wiki/HTTP) khung cốt lõi Django có thể được xem như là một kiến trúc [MVC](https://en.wikipedia.org/wiki/Model-view-controller).  Nó bao gồm một trình ánh [xạ đối tượng](https://en.wikipedia.org/wiki/Object-relational_mapping) (ORM) trung gian giữa [các mô hình dữ liệu](https://en.wikipedia.org/wiki/Data_modeling) ( được định nghĩa là các lớp Python ) và một [cơ sở dữ liệu quan hệ](https://en.wikipedia.org/wiki/Relational_database) (“ Model”), một hệ thống xử lý các yêu cầu HTTP với [hệ thống khuôn mẫu web](https://en.wikipedia.org/wiki/Web_template_system) (“ View “), và một [biểu thức chính qui](https://en.wikipedia.org/wiki/Regular_expression) dựa trên [URL](https://en.wikipedia.org/wiki/Uniform_Resource_Locator) phối (“ Controller “).
* Một máy chủ web nhẹ và độc lập để phát triển và thử nghiệm
* Một hệ thống xác nhận và tuần tự hóa biểu mẫu có thể dịch giữa các form [HTML](https://en.wikipedia.org/wiki/HTML) và các giá trị phù hợp để lưu trữ trong cơ sở dữ liệu
* Một hệ thống form sử dụng khái niệm [thừa kế được](https://en.wikipedia.org/wiki/Inheritance_(object-oriented_programming)) mượn từ lập trình hướng đối tượng
* Một [bộ nhớ đệm](https://en.wikipedia.org/wiki/Web_cache) có thể sử dụng bất kỳ phương thức cache nào
* Hỗ trợ cho [các](https://en.wikipedia.org/wiki/Middleware) lớp [trung gian](https://en.wikipedia.org/wiki/Middleware) có thể can thiệp ở các giai đoạn khác nhau của xử lý yêu cầu và thực hiện các chức năng tùy chỉnh
* Một hệ thống điều phối nội bộ cho phép các thành phần của một ứng dụng truyền thông các sự kiện với nhau thông qua các tín hiệu được xác định trước
* Một hệ thống [quốc tế hóa](https://en.wikipedia.org/wiki/Internationalization_and_localization), bao gồm các bản dịch của các thành phần riêng của Django thành nhiều ngôn ngữ
* Một hệ thống [tuần tự hóa](https://en.wikipedia.org/wiki/Serialization) có thể sản xuất và đọc [các biểu diễn XML](https://en.wikipedia.org/wiki/XML) và / hoặc [JSON](https://en.wikipedia.org/wiki/JSON) của các cá thể mô hình Django
* Một hệ thống để mở rộng khả năng của công cụ mẫu
* Một giao diện cho khung [kiểm thử đơn vị tích](https://en.wikipedia.org/wiki/Unit_test) hợp của Python.
  1. **Lập kế hoạch xây dựng website bằng Django**
* Tìm hiểu cấu trúc của một Website viết bằng Django.
* Tìm hiểu tập lệnh, cách sử dụng mô hình MVC trong Django.
* Phát triển một Website học code viết bằng Django FrameWork.
* Củng cố khả năng phân tích hệ thống website và database website.
* Sử dụng django framework tạo ra một trang web học lập trình trực tuyến cho lập trình viên.

# Chương 3: Tìm hiểu về cách sử dụngngôn ngữ lập trình Python

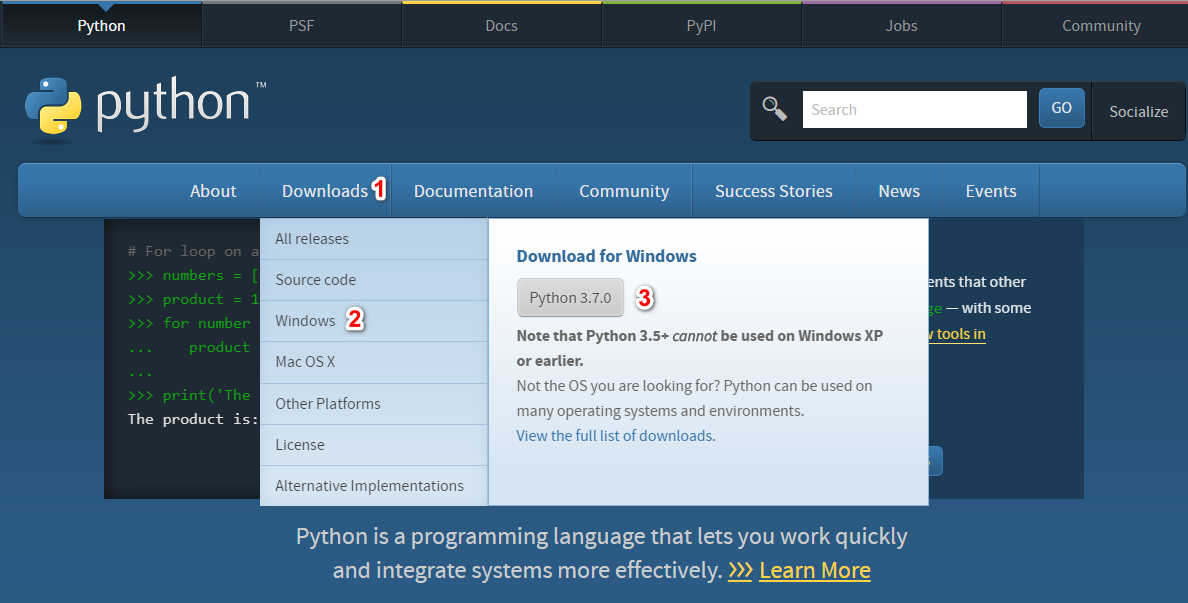
* 1. **Khái niệm Python**

Python là một ngôn ngữ lập trình thông dịch do Guido van Rossum tạo ra năm 1990. Python hoàn toàn tạo kiểu động và dùng cơ chế cấp phát bộ nhớ tự động; do vậy nó tương tự như Perl, Ruby, Scheme, Smalltalk, và Tcl. Python được phát triển trong một dự án mã mở, do tổ chức phi lợi nhuận Python Software Foundation quản lý.

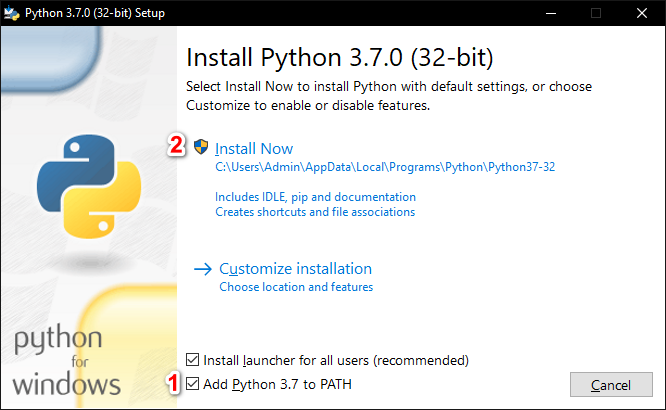
Theo đánh giá của Eric S. Raymond, Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình. Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu, như nhận định của chính Guido van Rossum trong một bài phỏng vấn ông.

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, nó đã “bành trướng” sang mọi hệ điều hành từ MS-DOS đến Mac OS, OS/2, Windows, Linux và các hệ điều hành khác thuộc họ Unix. Mặc dù sự phát triển của Python có sự đóng góp của rất nhiều cá nhân, nhưng Guido van Rossum hiện nay vẫn là tác giả chủ yếu của Python. Ông giữ vai trò chủ chốt trong việc quyết định hướng phát triển của Python.

* 1. **Cài đặt Python**
     1. **Download và cài đặt môi trường Python.**

**Bước 1:** Vào trang web <https://www.python.org>, di chuyển chuột vào phần Downloads, chọn hệ điều hành bạn đang dùng rồi chọn phiên bản cần tải   


Hình 3.2.1.1: Tải xuống bộ cài đặt Python

**Bước 2:** Mở bộ cài đặt vừa tải lên, tích chọn “Add Python 3.7 to Patch” để tự động cài đặt biến môi trường rồi nhấn “Install Now” để tiếp tục quá trình cài đặt   


Hình 3.2.1.2: Tiến hành cài đặt môi trường Python

**Bước 3:** Sau khi quá trình cài đặt diễn ra hoàn tất,hãy Kiểm tra xem đã cài đặt thành công hay chưa bằng cách dùng Command Promp ( cmd ). Tại cửa sổ **Command Prompt**, nhập python > **Enter**để kiểm tra, nếu hiện như dưới đây là bạn đã cài đặt thành công.

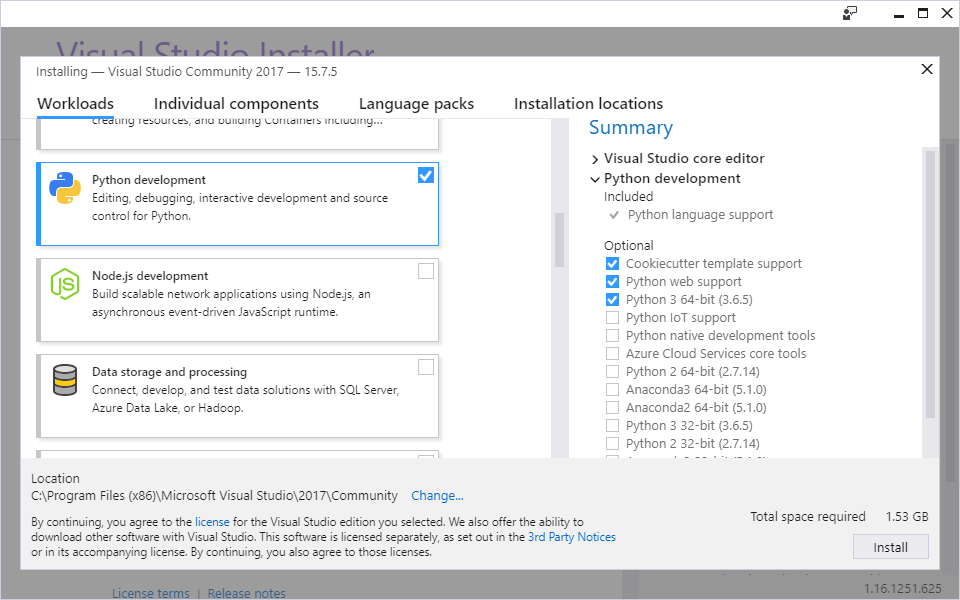


Hình 3.2.3: Cài đặt thành công Python

### **Download và cài đặt Vistual Studio 2017**

**Bước 1:** Truy cập trang web <https://visualstudio.microsoft.com/>, chọn phiên bản Community 2017 để sử dụng miễn phí

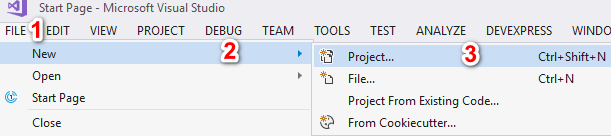
**Bước 2 :** Mở file vừa tải về lên, bộ cài đặt sẽ hiện lên giao diện chọn ngôn ngữ cần cài cài đặt, bạn tích chọn python ( bạn cũng có thể tích chọn thêm các ngôn ngữ lập trình khác nếu bạn muốn )



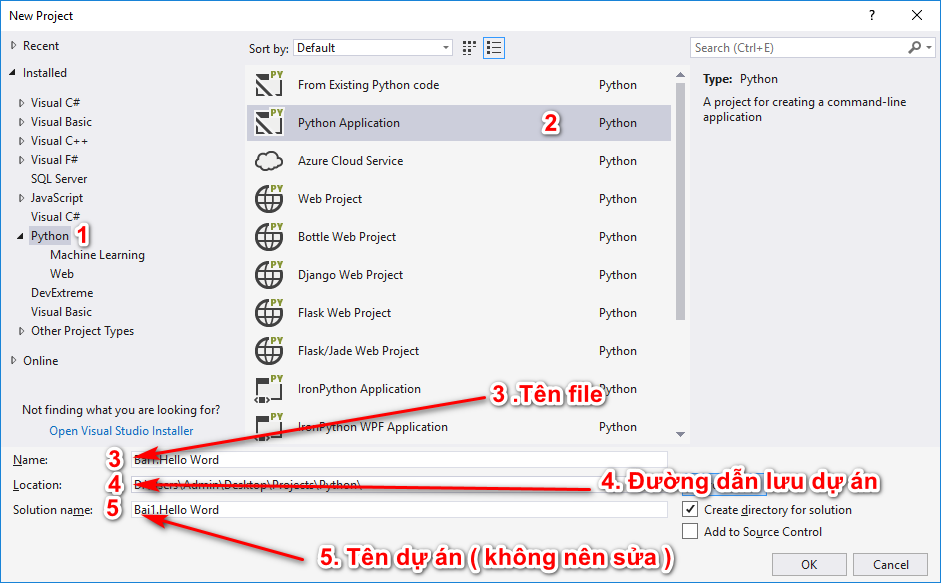
Hình 3.2.1.3: Chọn ngôn ngữ lập trình python

**Bước 3:** Sau khi click “install” quá trình cài đặt sẽ diễn ra một cách tự động, khi hoàn tất là đã chuẩn bị xong công cụ lập trình python.

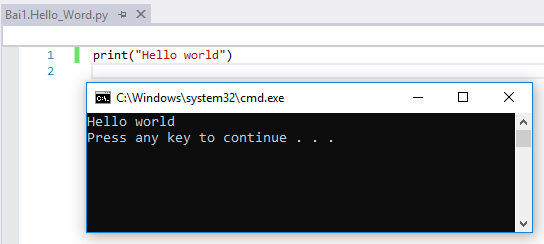
* 1. **Các kiến thức cơ bản về Python**
     1. **Chương trình đầu tiên Hello Word**

**Bước 1**: Mở chương trình VS lên tạo 1 dự án mới như hình dưới   


**Bước 2**: Chọn như hình dưới



**Bước 3**: Gõ code rồi nhấn tổ hợp phím Ctrl + F5 để chạy chương trình vừa code



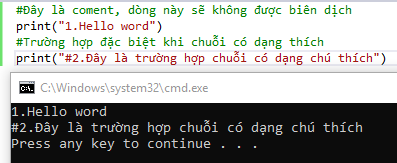
* + 1. **Chú thích trong Python**

#### Cú pháp.

# <Nội dung ghi chú>

#### Trường hợp đặc biệt.

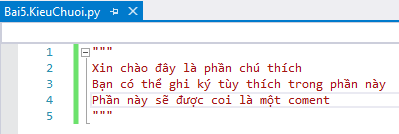
* Chuỗi có dạng là một comment: là coment nhưng lại nằm trong ngoặc kép



Hình 3.3.2. 1: Cách dùng chú thích trong Python

Để chú thích nhiều dòng, ta sử dụng khái niệm là Docstring:

* Một dạng chú thích nhiều dòng.
* Hay xuất hiện ở đầu một file Python, sau một dòng định nghĩa lớp, hàm.
* Và đây cũng là một trong những chuẩn quy ước về định dạng, trình bày code.



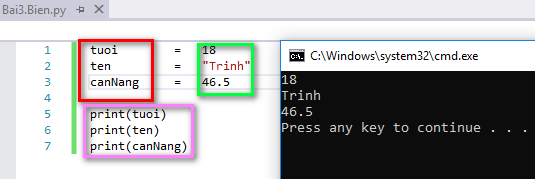
Hình 3.3.2. 2: Mội ví dụ về Docstring

* + 1. **Biến trong Python**

#### Cú pháp:

**TH1**: Khởi tạo 1 biến:

<tên biến> = <giá trị của biến>

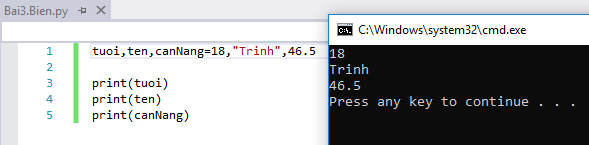


Hình 3.3.2. 3: Ví dụ về khởi tạo một biến

Giải thích ví dụ: Những dòng có khung đỏ đó chính là dòng lệnh dùng để khai báo tên biến. Còn dòng mũi tên màu xanh chính là kết quả của biến. “tuoi”, “ten”, “canNang” chính là những tên biến còn những thứ sau dấu bằng như là “18”, “Trinh”, “46.5” đó chính là giá trị của biến.

**TH2**: Khởi tạo nhiều biến:

<tên biến thứ nhất>, <tên biến thứ hai>,..,<tên tên biến thứ n> = <giá trị biến thứ nhất>, <giá trị biến thứ hai>,.., <giá trị biến thứ n>



Hình 3.3.2. 4: Một cách khởi tạo nhiều biến

Giải thích ví dụ: Ở ví dụ trên ‘tuoi’ là biến thứ nhất, ‘tuoi’ là biến thứ hai và ten’ là biến thứ ba là “canNang”. Và sau đó ‘18’ là giá trị biến thứ nhất, “Trinh” là giá trị biến thứ hai và “46.5” là giá trị biến thứ ba.

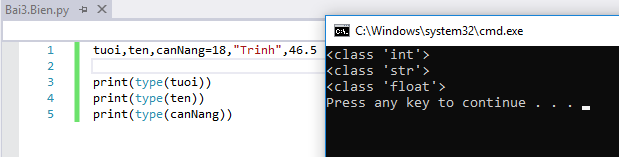
**Một số lưu ý khi đặt tên biến:**

* + Tên của biến không được bắt đầu bằng số.
  + Tên biến không được trùng với các từ khóa của Python.
  + Tên của biến chỉ chứa các chữ cái, số.
  + Tên biến trong Python có phân biệt chữ in hoa và in thường. Ví dụ: PI, Pi, pI, pi là 4 tên biến khác nhau.

**c)** **Kiểm tra kiểu dữ liệu của biến**

#### a) Cú pháp:

type(<tên biến>)



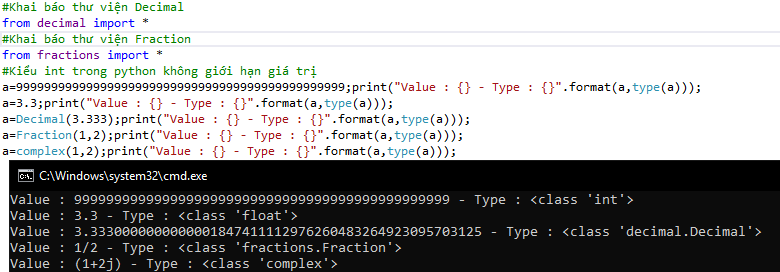
Hình 3.3.3. 1: Cách khởi tạo và gán giá trị cho biến

Kết quả: Như bạn thấy ở ví dụ trên kết quả tra ra một số kiểu dữ liệu đó là ‘int’, ‘str’, ‘float’. Đó là các kiểu dữ liệu phổ biến trong các ngôn ngữ lập trình hiện nay.

### **Một số kiểu dữ liệu cơ bản trong Python.**

#### Kiểu số.

Giống như các ngôn ngữ lập trình có trước, các kiểu dữ liệu số nguyên thủy vẫn được giữ lại trong python: số nguyên ( int) – Python hỗ trợ kiểu số nguyên vô hạn, số thực ( float, decimal,double …) … và bên cạnh đó, python còn có những kiểu dữ liệu mới ( fraction, complex )và cải tiến vượt bậc đi kèm những hàm xử lý hữu dụng.



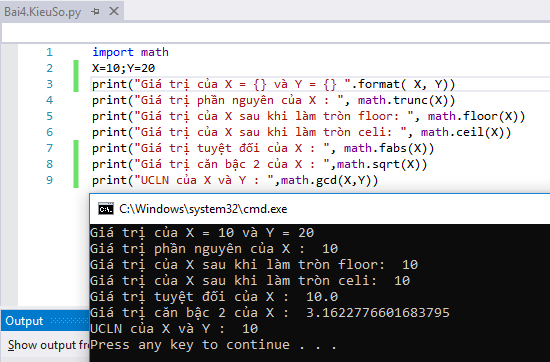
Hình 3.3.4. 1: Một ví dụ về kiểu dữ liệu số trong python

**Một số hàm toán học hay dùng.**

* Python hỗ trợ rất nhiều thư viện toán học và nổi bật là thư viện math với nhiều hàm hỗ trợ tính toán cực mạnh.
* Cú pháp sử dụng thư viện: **import** <tên\_thư\_viện>.
* Cú pháp sử dụng hàm: <tên\_thư\_viện>.<tên\_hàm>.
* Một số hàm hay dùng:

|  |  |
| --- | --- |
| **Tên hàm** | **Mô tả** |
| .trunc(x) | Trả về số nguyên là phần nguyên của x |
| .floor(x) | Trả về 1 số nguyên được làm tròn từ số x, kết quả luôn <= x |
| .celi(x) | Trả về 1 số nguyên được làm tròn từ số x, kết quả luôn >= x |
| .fabs(x) | Trả về giá trị tuyệt đối của x |
| .sqrt(x) | Trả về căn bậc hai của x |
| .gcd(x,y) | Trả về số nguyên là UCLN của x và y |

*Bảng 3.3.4.1: Những hàm thông dụng của thư viện math*



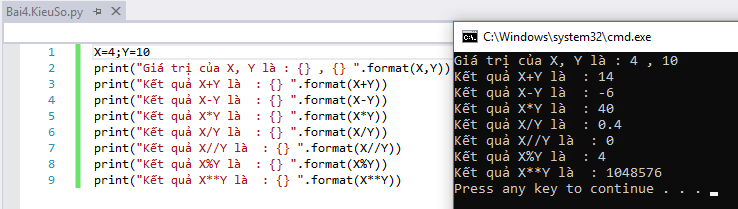
Hình 3.3.4. 2: Ví dụ về các hàm thông dụng của thư viện math

**Toán tử cơ bản với kiểu dữ liệu số**

* Biểu thức chính là một thực thể toán học. Nói cách khác, nó là một sự kết hợp giữa 2 thành phần:
  + - * Toán hạng: có thể là một hằng số, biến số (X, Y)
      * Toán tử: xác định cách thức làm việc giữa các toán hạng (+,-,\*,/)
* Dưới đây là một số biểu thức toán học của kiểu dữ liệu số trong Python

|  |  |
| --- | --- |
| **Biểu thức** | **Mô tả** |
| X+Y | Tổng X và Y |
| X-Y | Hiệu X và Y |
| X\*Y | Tích X và Y |
| X/Y | Thương X và Y ( kết quả luôn trả về 1 số thực) |
| X//Y | Thương X và Y ( trả về 1 số nguyên, kết quản luôn <=X/Y) |
| X%Y | Lấy phần dư của X chia Y |
| X\*\*Y | Lũy thừa mũ Y cơ số X |

*Bảng 3.3.4.2: Bảng toán tử cơ bản*



Hình 3.3.4. 3: Ví dụ các toán tử với 2 toán hạng X, Y

#### **b)** **Kiểu chuỗi**

Trong Python, chuỗi là những thứ được đặt trong cặp dấu ‘ ’ ( nháy đơn ), hoặc “ ” ( nháy kép ), có thể cũng là trong cặp ‘’’ ‘’’, “”” “””. Nhưng cơ bản và thường đường sử dụng nhất là cặp ‘ ’ và “ ”.

Một số toán tử cơ bản trong chuỗi:

|  |  |  |
| --- | --- | --- |
| **Biểu thức** | **Yêu cầu** | **Yêu cầu** |
| A+B | A và B là kiểu string | Nối chuỗi B vào sau chuỗi A |
| A\*n | A là kiểu string, n là kiểu Integer | Nhân n lần chuỗi A |
| A in B | A và B là kiểu string | Kiểm tra A có trong B hay không |
| A[s:e] | A là kiểu string, s và e là kiểu Integer | Lấy chuỗi con trong chuỗi A bắt đầu từ vị trí s kết thúc tại vị trí e |

**Lấy độ dài chuỗi:**Sử dụng hàm len(...) để trả về độ dài của chuỗi

count = len("Hello world")#(count có giá trị 11)

**Tìm & thay thế nội dung** Có thể tìm và thay thế trong chuỗi bằng cách gọi phươngthức replace(search, replace[, max]) của một chuỗi. Vídụ:

str = 'Hello world'

newstr = str.replace('Hello', 'Bye')

print (newstr)

#(Sẽ hiển thị chuỗi "Bye world" trên màn hình)

**Tìm vị trí chuỗi con**  
 Có thể tìm vị trí của một chuỗi con trong chuỗi lớn bằng cách gọi phương thức find(str, beg=0 end=len(string)). Bắt đầu là vị trí 0, nếu không tìm ra thì trả về -1 ( phương thức này trả về vị trí đầu tiên của chuỗi đang tìm ). Ví dụ:

str = 'Xin chào chúng tôi là CTK39'

print(str.find("chào",0,None))#Kết quả sẽ hiện là 4

print(str.find("DLU",0,None))#Kết quả sẽ hiện là -1

**Xóa khoảng trắng của chuỗi** Có thể loại bỏ các ký tự (mặc định là ký tự khoảng trắng) trước và sau một chuỗi, bằng cách gọi các phương thức sau:

* + - * strip([chuỗi]): loại bỏ trước và sau chuỗi
      * lstrip([chuỗi]): loại bỏ phía trước chuỗi
      * rstrip([chuỗi]): loại bỏ phía sau chuỗi

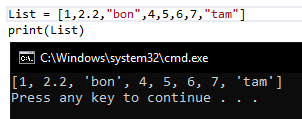
**Một số hàm xử lý chuỗi thông dụng khác:**

* isnumeric(): Kiểm tra một chuỗi có phải là chuỗi số.
* lower(): trả về một chuỗi viết thường.
* upper(): trả về một chuỗi viết hoa.
* capitalize(): Viết hoa ký tự đầu của chuỗi.
* title(): trả về chuỗi ký tự đầu viết hoa, ký tự sau viết thường.
* swapcase(): trả về chuỗi mà chữ thường thành chữ hoa và ngược lại.
* strip(‘char’): trả về chuỗi đã loại bỏ ký tự char
* split(‘char’,n): trả về một mảng được cắt bởi char ( mặc định là khoảng trắng ) số lần n ( mặt định là -1)

#### **c)** **Kiểu danh sách ( list ).**

**LIST** là một container được sử dụng rất nhiều trong các chương trình Python. Một List gồm các yếu tố sau:

* + - Được giới hạn bởi cặp ngoặc [ ], tất cả những gì nằm trong đó là những phần tử của List.
    - Các phần tử của List được phân cách nhau ra bởi dấu phẩy (,).
    - Một list có thể chứa nhiều phần tử thuộc nhiều kiểu dữ liệu khác nhau



Hình 3.3.4. 4: Một ví dụ về list

**Một số hàm xử lý cơ bản**

* count(X): Đếm số lần xuất hiện của phần tử X trong danh sách
* index(X): Trả về vị trí của phần tử X trong danh sách.
* copy(): Trả về một chuỗi tương tự với chuỗi nguồn.
* clear(): Xóa tất cả phần tử trong chuỗi.
* append(X): Thêm phần tử X vào cuối chuỗi.
* insert(index, X): Thêm một phần tử X vào danh sách tại vị trí index
* pop(index): Xóa phần tử thứ index trong danh sách, nếu index không được truyền vào 🡪 xóa phần tử cuối danh sách.
* remove(X): Xóa phần tử X đầu tiên trong danh sách.
* reverse(): trả về một chuỗi đảo ngược.
* sort(): Trả về một chuỗi được sắp xếp từ bé tới lớn

#### **d) Kiểu dữ liệu Tuple.**

Có thể coi Tuple là một dạng nâng cao của List, nó thừa hưởng tất cả những gì của List và còn có thêm 1 số cải tiến mới nhưng lại khó dùng hơn List.

**Thành phần của 1 tuple:**

* + Được giới hạn bởi cặp ngoặc (), tất cả những gì nằm trong đó là những phần tử của Tuple.
  + Các phần tử của Tuple được phân cách nhau ra bởi dấu phẩy (,).
  + Tuple có khả năng chứa mọi giá trị, đối tượng trong Python.

**Ưu điểm hơn List:**

* + - Tốc độ truy xuất của Tuple nhanh hơn List.
    - Dung lượng chiếm trong bộ nhớ nhỏ hơn List.
    - Bảo vệ dữ liệu không bị thay đổi.
    - Có thể dùng làm key của dictionary.

**Các phương thức cơ bản của Tuple**

|  |  |
| --- | --- |
| **Hàm** | **Chức năng** |
| [count(x)](https://www.programiz.com/python-programming/methods/tuple/count) | -Trả về số lần xuất hiện của phần tử x trong tuple |
| [index(x)](https://www.programiz.com/python-programming/methods/tuple/index) | -Trả về vị trí xuất hiệnđầu tiên của phần tử x trong tuple |
| [len()](https://www.programiz.com/python-programming/methods/built-in/len) | -Trả về số lượng phần tử của Tuple |
| [max()](https://www.programiz.com/python-programming/methods/built-in/max) | -Trả về phần tử lớn nhất trong Tuple |
| [min()](https://www.programiz.com/python-programming/methods/built-in/min) | -Trả về phần tử bé nhất trong Tuple |
| [sorted()](https://www.programiz.com/python-programming/methods/built-in/sorted) | -Trả về danh Tuple đã được sắp xếp |
| [sum()](https://www.programiz.com/python-programming/methods/built-in/sum) | -Trả tổng của Tuple |
| [tuple()](https://www.programiz.com/python-programming/methods/built-in/tuple) | -Ép kiểu list, string,set,dic sang kiểu tuple |

*Bảng 3.3.4.3: Bảng các phương thức cơ bản của Tuple*

#### **e) Kiểu dữ liệu Dic ( Dictionary )**

**Khái quát:**

Dictionary cũng là một cấu trúc mảng, nhưng các phần tử bao gồm key và value. Một Dictionary được khai báo bằng cặp dấu ngoặc {...}.

Ex = { key: value, key: value }

Ví dụ:

Ex = { “name”: “Vuong”, “class”: “CTK39”}

Để truy xuất giá trị dựa vào key của một đối tượng, ví dụ:

Ex = { “name”: “Vuong”, “class”: “CTK39”

print(Ex[“name”]) #Kết quả sẽ in ra chữ: Vuong

Để thêm một phần tử vào trong dic, ta sử dụng cú pháp:

<dic cần thêm>[key]=value

Ví dụ

Ex = { “name”: “Vuong”, “class”: “CTK39”}

Ex[“FullName”]=”Nguyen Van Vuong”

**Thành phần của 1 dic:**

* Được giới hạn bởi cặp ngoặc nhọn {}, tất cả những gì nằm trong đó là những phần tử của Dict.
* Các phần tử của Dict được phân cách nhau ra bởi dấu phẩy (,).
* Các phần tử của Dict phải là một cặp key-value.
* Cặp key-value của phần tử trong Dict được phân cách bởi dấu hai chấm (:).
* Các key buộc phải là một hash object.

**Các hàm thông dụng xử lý Dic**

* + - * copy(): Trả về một chuỗi giống với giống nguồn.
      * clear(): Xóa mọi phần tử.
      * get(key,default): lấy giá trị của khóa key, nếu không có khóa key trong mảng thì trả về default (mặc định là none ).
      * items(): Trả về một giá trị thuộc lớp dict\_items. Các giá trị của dict\_items sẽ là một tuple với giá trị thứ nhất là key, giá trị thứ hai là value.
      * keys(): Trả về một giá trị thuộc lớp dict\_keys (dict\_keys là một iterable ). Các giá trị của dict\_keys sẽ là các key trong Dict.
      * values(): Trả về một giá trị thuộc lớp dict\_values (Dict\_values là một iterable). Các giá trị của dict\_values sẽ là các value trong Dict.
      * pop( key,default ): Xóa đi phần tử có khóa key, nếu xóa thành công thì trả về value của key đó, còn không thì trả về default ( mặc định là None ).
      * popitem(): Xóa phần tử cuối trong danh sách, nếu xóa thành công thì sẽ trả về tuple với key và value tương ứng, trường hợp nếu Dic rỗng thì sẽ báo lỗi.
      * setdefault( keys, default ): Tìm kiếm phần tử có khóa key trong dic, nếu tìm thấy thì sẽ trả về value tương ứng, nếu không sẽ thêm vào dic một phần tử mới với key = keys và value= default ( mặc định là none ).

### **Các hàm xử lý file trong Python**

Hàm Open:

a) Cú pháp:

open(file\_patch, mode='r', -1, encoding=None,)

**Trong đó:**

* file\_patch: đường dẫn tới file
* mode: chế độ đọc viết ( xem bảng 3)
* -1: để kích thước bộ nhớ đệm là mặc định
* encoding: Chế độ mã hóa, tiếng việt là “utf-8-sig”

b) Công dụng: trả về một file object, tạo ra 1 phiên làm việc với file.

c) Mở rộng: hàm open cung cấp rất nhiều chế độ tương tác với file, dưới đây là bảng các mode thông dụng:

|  |  |
| --- | --- |
| **Tên** | **Chức năng** |
| **r** | Mở để đọc, là mode mặc định. |
| **r+** | Mở để đọc và ghi |
| **w** | Mở để ghi, nếu file có nội dung thì sẽ bị xóa trắng để ghi mới, nếu file chưa tồn tại sẽ tạo file mới để ghi. |
| **w+** | Mở để ghi và đọc, nếu file có nội dung thì sẽ bị xóa trắng để ghi mới, nếu file chưa tồn tại sẽ tạo file mới để ghi. |
| **a** | Mở để đọc, nếu file chưa tồn tại sẽ tạo file mới để ghi. |
| **a+** | Mở để đọc và ghi, nếu file chưa tồn tại sẽ tạo file mới để ghi. |

*Bảng 3: Các mode cơ bản của hàm open*

**Các hàm thông dụng khác:**

* close(): kết thúc phiên làm việc với file.
* read( n ): đọc n ký tự từ đầu từ file ( không có tham số n hoặc n <0 sẽ đọc hết file ).
* readlines( n ): đọc n dòng từ đầu từ file ( không có tham số n hoặc n <0 sẽ đọc hết file ).
* write( text ): ghi chuỗi text vào file, trả về số ký tự ghi vào file.
* seek( n ): chỉ định con trỏ tới vị trí n.

### **Hàm trong python**

Hàm nói một cách môn na là tập hợp của 1 hoặc nhiều dòng lệnh thực hiện một công việc cụ thể nhất định nào đó. Là một ngôn ngữ lập trình bậc cao không thể định nghĩa hàm. Python cũng không thể loại trừ. Để có thể sử dụng khái niệm “hàm” trong python, ta khai báo với cấu trúc như sau:

**def** <**Tên hàm**> ( **Tham\_số\_1**, **Tham\_Số\_2**…**Tham\_Số\_n** )

#Lệnh / Khối lệnh thực hiện

Trong đó:

* Tên hàm: bắt buộc – đây sẽ là định danh của hàm
* Các tham số: không bắt buộc – sử dụng cho mỗi mục đích riêng

Cách sử dụng:

Để gọi một hàm đã định nghĩa, ta có thể sử dụng cú pháp sau:

#1.Trường hợp có tham số

<**Tên hàm**> ( **Tham\_số\_1**, **Tham\_Số\_2**…**Tham\_Số\_n** )

#2.Trường hợp không có tham số

<**Tên hàm**> ()

#3.Ví dụ minh họa hàm không có tham số

**def Test():**

**print("Hello")**

**Test()**

#-->Kết quả xuất ra màn hình chuỗi “Hello”

#4.Ví dụ minh họa hàm có tham số

**def Test(name):**

**print("Hello",name)**

**Test("Vuong")**

#-->Kết quả xuất ra màn hình chuỗi “Hello Vuong”

Trong thực tế ta gặp phải rất nhiều trường hợp là yêu cầu người dùng nhập vào giá trị để ta lấy giá trị đó dùng cho một hàm đó nhưng vì lý do nào đó người dùng không nhập gì cả khiến cho chương trình bị lỗi hoặc ta phải tìm hướng xử lý khác. Để giải quyết điều này Python cung cấp cho người dùng một thuật ngữ là “default argument” cho phép lập trình viên gán giá trị mặc định cho tham số, hãy xem ví dụ dưới đây để làm rõ vấn đề:

#5 Ví dụ về default argument

**def Test(name=“Đây là default argument”):**

**print("Hello",name)**

**Test()**

# -->Kết quả xuất ra màn hình chuỗi “Hello Đây là default argument”

### **Class trong python**

Python là ngôn ngữ hướng đối tượng ngay từ đầu. Bởi vậy, việc tạo ra và sử dụng các lớp và các đối tượng là hết sức dễ dàng. Cũng như các ngôn ngữ lập trình khác, python cũng hỗ trợ lập trình hướng đối tượng bằng cách tạo các class ( gồm các thuộc tính, đặc trưng, phương thức …) giúp lập trình viên có thể lập trình hướng theo ý muốn.

Để khởi tạo class trong python:

**class** className:

‘Description’

#Code

Trong đó:

* className: là tên của lớp.
* Description: mô tả ngắn ngọn về class ( không bắt buộc ).
* Code: các thuộc tính, phương thức … của lớp.

**Cấu tạo cơ bản của lớp:**

* + - 1. Thuộc tính:

Thuộc tính là một thành viên của 1 lớp, là đặc tả của đối tượng mà lớp đó đang mô tả ( ví dụ như đối tượng hình vuông có thuộc tính là chiều dài cạnh ).

* + - 1. Phương thức:

- Phương thức của class giống như một hàm thông thường nhưng nó là một hàm của class được gọi thông qua đối tượng.

- Tham số đầu tiên của phương thức luôn là self (một từ khóa ám chỉ chính class đó).

* + - 1. Phương thức khởi tạo:

-Phương thức khởi tạo (Constructor) là một phương thức đặc biệt của lớp (class), nó luôn có tên là \_\_init\_\_.

-Tham số đầu tiên của constructor luôn là self (Một từ khóa ám chỉ chính class đó).

-Constructor được sử dụng để tạo ra một đối tượng.

-Constructor gán các giá trị từ tham số vào các thuộc tính của đối tượng sẽ được tạo ra.

-Bạn chỉ có thể định nghĩa nhiều nhất một phương thức khởi tạo (constructor) trong class.

-Nếu class không được định nghĩa constructor, Python mặc định coi rằng nó thừa kết từ constructor của lớp cha.

"Đây là class sinh viên"

class SinhVien(object):

#Khai báo thuộc tính

ID = ""

Name = ""

Class = ""

Point = ""

#Định nghĩa hàm khởi tạo

def \_\_init\_\_(self,ID,Name,Class,Point):

self.ID = ID

self.Name = Name

self.Class = Class

self.Point = Point

#Định nghĩa hàm trả về thông tin của Sinh viên

def toString(self):

return ("ID: {}, Name: {}, Class: {}, Point: {}".format(self.ID,self.Name,self.Class,self.Point))

*Một ví dụ về class trong python*

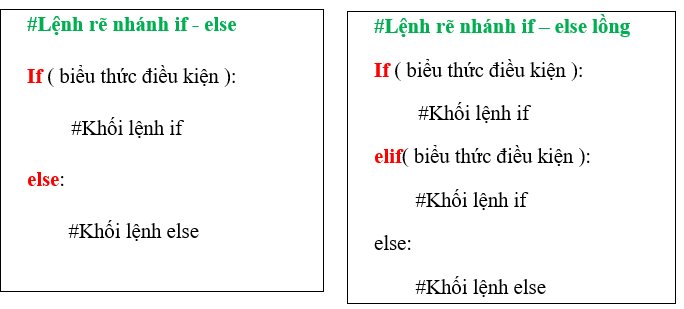
Các hàm hay dùng để xử lý class:

|  |  |
| --- | --- |
| **Hàm** | **Mô tả** |
| getattr(obj, name[, default]) | Trả về giá trị của thuộc tính, hoặc trả về giá trị mặc định nếu đối tượng không có thuộc tính này. |
| hasattr(obj,name) | Kiểm tra xem đối tượng này có thuộc tính cho bởi tham số 'name' hay không. |
| setattr(obj,name,value) | Sét giá trị vào thuộc tính. Nếu thuộc tính không tồn tại, thì nó sẽ được tạo ra. |
| delattr(obj, name) | Xóa bỏ thuộc tính. |

### **Câu lệnh rẽ nhánh trong python**

Trong python không có câu lệnh rẽ nhánh switch-case mà thay vào đó là lệnh rẽ nhánh if – else.

Về cú pháp cũng tương tự các ngôn ngữ khác:



Nếu biểu thức điều kiện trả về true, khối lệnh if sẽ được thực hiện và ngược lại khối lệnh else sẽ được thực thi.

### **Các vòng lặp trong python**

Trong các ngôn ngữ lập trình, vòng lặp rất quan trọng trong việc xử lý các lệnh / khối lệnh mang tính lặp lại, python hỗ trợ 2 loại vòng lặp chính là for và while.

* + 1. Vòng lặp while:

1. Khái quát :

Để sử dụng vòng lặp while, ta có thể khai báo như sau:

**while( biểu thức điều kiện ):**

#Lệnh / Khối lệnh thực thi

Vòng lặp while đầu tiên sẽ xét biểu thức điều kiện, nếu biểu thức này đúng thì sẽ thực hiện lệnh / khối lệnh tương ứng, việc này lặp đi lặp lại cho tới khi biểu thức điều kiện trở nên sai lệnh

* + 1. Vòng lặp for

1. Khái quát :

Để sử dụng vòng lặp for, ta có thể khai báo như sau:

**for <var> in <obj>**

#Lệnh / Khối lệnh thực thi

Vòng lặp for sẽ lấy các phần tử của **<obj>** ( **<obj>** phải cho phép indexing như chuỗi, tuple, list…) rồi gán gián trị phần tử đó vào **<var>**, sau đó thực hiện lệnh / khối lệnh tương ứng, vòng lặp dừng khi đã duyệt qua tất cả các phần tử trong **<obj>.**

### **Nhập xuất trong python**

* + 1. Hàm xuất – print

#### Cú pháp

print(\*objects, sep=' ', end='\n', file=sys.stdout)

Trong đó

* objects: là một / nhiều đối tượng muốn xuất ra ( có thể là kiểu int, string, decimal, tuple … ).
* sep: chuỗi ngăn cách các object ( mặc định là khoảng trắng ).
* end: chuỗi khi hết dòng ( mặc định là “\n”).
* file: sẽ nói rõ hơn trong phần tiếp

*b) Sử dụng*

Khi muốn xuất dữ liệu của nhiều obj, mỗi obj sẽ ngăn cách nhau bằng một sep, khi hết dòng sẽ kết thúc bằng end, hàm print còn có thể ghi dữ liệu xuống file thông qua tham số file, ví dụ dưới đây sẽ mô hình lại chức năng của các tham số :

#sep=" ", end="\n", file = sys.stdout ( mặc định )

print("Hello","CTK39")

# -->xuất ra chuỗi Hello CTK39

#sep="\_", end="\n", file = sys.stdout

print("Hello","CTK39",sep ="\_")

#--> xuất ra chuỗi Hello\_CTKK39

#sep="X", end=" Xuống Dòng", file = sys.stdout

print("Hello","CTK39",sep ="X",end="Xuống dòng")

#--> xuất ra chuỗi HelloXCTKK39XuốngDòng

#sep="X", end=" Xuống Dòng", file = file\_obj

with open("test.txt","a+",-1,"utf-8-sig") as file\_obj:

print("\nGhi xuống file",file = file\_obj)

#--> ghi xuống file chuỗi: “\nGhi xuống file”

* + 1. Hàm nhập – input:

*a) Cú pháp*

Input( text )

Trong đó:

- text: là chuỗi mà bạn sẽ xuất ra khi yêu cầu người dùng nhập ( làm rõ ở phần kế tiếp.

*b) Sử dụng*

Khi muốn người dùng nhập dữ liệu, nếu muốn xuất thông báo là người dùng nhập cái gì thì có thể truyền vào tham số text, mặc định text bằng None, ví dụ dưới đây sẽ làm rõ cú pháp này:

1.Trường hợp không truyền tham số text

Input1 = input()

#--> màn hình sẻ chỉ hiện con trỏ nhấp nháy yêu cầu nhập dữ liệu

2.Trường hợp truyền tham số text

Input2 = input("Vui lòng nhập vào tên bạn: ")

print(Input2)

#--> màn hình sẻ hiện chuỗi "Vui lòng nhập vào tên bạn: "cùng với con trỏ nhấp nháy yêu cầu nhập

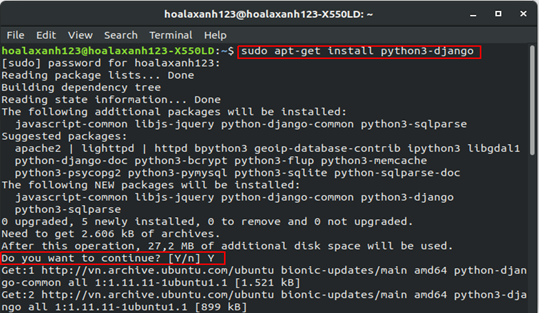
*Ví dụ về hàm input*

* 1. **Các kiến thức cơ bản về Framework Django:**

#### Cài đặt Django

Để có thể cài đặt Django , bạn cần cài phải cài cho máy tính của mình Python ( đã nêu ở phần trước ). Khi bạn đã cài được python, hãy mở terminal lên và gõ câu lệnh sau, nhập mật khẩu tài khoản hiện tại và chọn Yes khi được yêu cầu, Django sẽ tự động cài đặt mà bạn chẳng cần phải làm gì rườm rà.

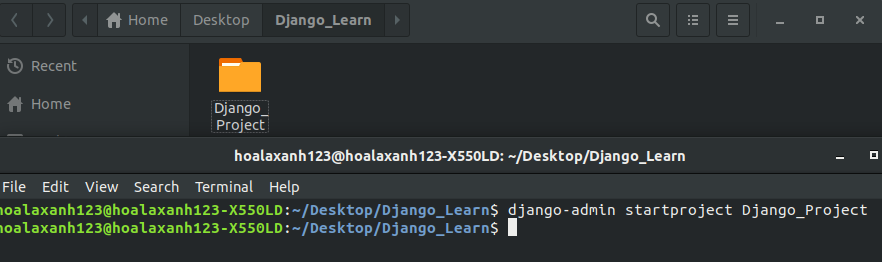
**Code: sudo apt-get install python3-django**



Hình 3.10. 1: Quá trình cài đặt django diễn ra hoàn toàn tự động

#### Tạo dự án web django

Để tạo một dự án web django, đầu tiên bạn cần vào thư m sẽ chứa dự án django, chuột phải vào vùng trống trong thư mục rồi chọn Terminal, khi terminal hiện lên, hãy nhập code sau, quá trình tạo dự án django cũng sẽ hoàn toàn tự động diễn ra.

Code: django-admin startproject tên\_dự\_án

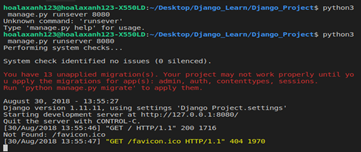
Hình 3.10.2. 1 Tạo dự án Django

* + 1. Khởi chạy dự án django

Để có thể khởi chạy django, ta sẽ dùng visual code + terminal, cách làm như sau:

Đầu tiên mở visual code ( VC ) lên, nhấn tổ hợp ctrl +O để chọn thư mục dự án mà ta vừa tạo ở bước trên, sau đó mở terminal của VC lên, gõ đoạn lệnh sau để khởi tạo server:

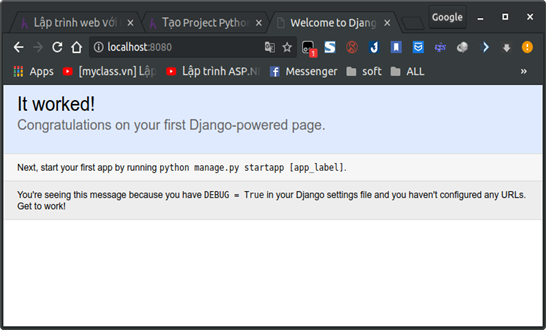
Code: python3 manage.py runserver số\_port



Hình 3.10.3. 1 : Khởi chạy dự án django

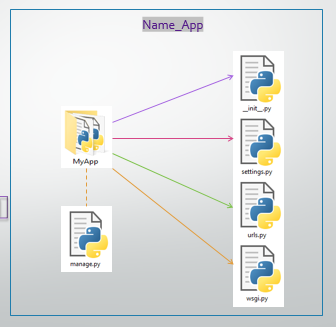
*Khởi chạy server django dễ dàng với terminal*

Sau khi khởi chạy server thành công, vào trình duyệt gõ vào thanh địa chỉ : localhost:số\_port mà bạn đã khởi tạo, nếu hiện như hình dưới là đã khởi tạo thành công.



Hình 3.10.3. 2: Khởi chạy server django

* + 1. Cấu trúc cơ bản của một website Django

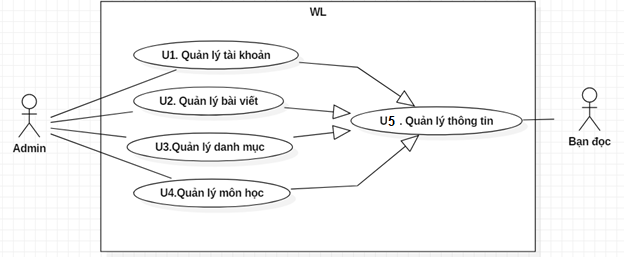


* **manage.py**: một CLI giúp bạn tương tác nhanh với Django.
* Folder mysite bên trong thực chất là một Python package, và tên của nó sẽ là tên package bạn dùng để import trong code.  
  VD: import mysite.urls.
* **mysite/\_\_init\_\_.py**: File rỗng, có mục đích biến folder này thành một Python package.
* **mysite/settings.py**: Các settings của project ở trong này.
* **mysite/urls.py**: Định nghĩa các URL của trang web.
* **mysite/wsgi.py**: Dùng khi deploy project của bạn.

# Chương 4: Xây dựng các ứng dụng

## **Phân tích cơ sở dữ liệu**

* + 1. **Xác định các lớp**
       1. **Tạo danh sách các lớp ứng viên**

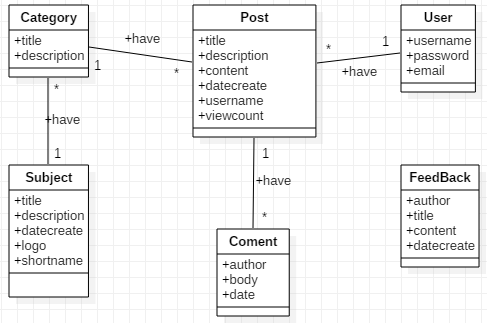
****

Hình 4 1 Sơ đồ tổng quan Use Case

* U1: Quản lý tài khoản (kế thừa U7,bao gồm U1.1,U1.2,U1.3,U1.5).
* Điều kiện tiên quyết: Đăng nhập.
* Admin chọn chức năng mà muốn quản lý tài khoản tùy vào phân quyền sẽ có thêm các chức năng.
* Admin điền vào các trường thông tin bắc buộc trên website (nếu có).
* Admin thực thi chức năng đã chọn.
  + Hâụ điều kiêṇ: Hệ thống lưu lại thông tin sau khi quản lý.
* U1.5: Đăng nhập (mở rộng của U1)
  + Điều kiện tiên quyết: Không có.
    - Người dùng hệ thống điền tên đăng nhập và mật khẩu của mình.
    - Chọn đăng nhập để đăng nhập vào hệ thống.
  + Hâụ điều kiêṇ: Người dùng đăng nhập tài khoản thành công.
* U1.1: Đổi mâṭ khẩu. (mở rộng của U1)
  + Điều kiện tiên quyết: Đăng nhập.
    - Thành viên lưạ choṇ thay đổi mật khẩu.
    - Thành viên nhập mật khẩu cũ (được che khuất trên màn hình).
    - Thành viên nhập mật khẩu mới (được ẩn).
    - Thành viên nhập mật khẩu mới một lần nữa.
    - Thành viên choṇ thay đổi.
    - WL yêu cầu xác nhâṇ (cảnh báo rằng mật khẩu mới phải được ghi nhớ).
    - Nếu thành viên xác nhâṇ, mật khẩu được thay đổi.
  + Hậu điều kiêṇ: Mật khẩu đã được thay đổi.
* U1.2: Đăng xuất (kế thừa U7, mở rộng của U1.5).
  + Điều kiện tiên quyết: Không có.
    - Thành viên lưạ choṇ đăng xuất.
    - WL kết thúc phiên hiện tại.
    - WL câp̣ nhâṭ không cho sử dụng chức năng.
  + Hâụ điều kiêṇ: Thành viên đăng xuất.
* U1.3: Phân quyền (mở rộng bởi U1)
  + Điều kiện tiên quyết: Đăng nhập.
    - Admin chọn chức năng phân quyền
    - Phân quyền các cấp bậc chức vụ cho các tài khoản.
    - Admin thực thi chức năng đã chọn.
  + Hâụ điều kiêṇ: Hệ thống lưu lại thông tin sau khi quản lý.
* U2: Quản lý bài viết. ( kế thừa U7 ).
  + Điều kiện tiên quyết: Đăng nhập.
    - Admin chọn chức năng mà muốn quản lý bài viết.
    - Admin điền vào các trường thông tin bắc buộc trên website (nếu có).
    - Admin thực thi chức năng đã chọn.
  + Hâu điều kiêṇ: Hệ thống lưu lại thông tin sau khi quản lý.
* U3: Quản lý danh mục. ( kế thừa U7 ).
  + Điều kiện tiên quyết: Đăng nhập.
    - Admin chọn chức năng mà muốn quản lý danh mục.
    - Admin điền vào các trường thông tin bắc buộc trên website (nếu có).
    - Admin thực thi chức năng đã chọn.
  + Hâu điều kiêṇ: Hệ thống lưu lại thông tin sau khi quản lý.
* U4: Quản lý môn học. ( kế thừa U7 ).
  + Điều kiện tiên quyết: Đăng nhập.
    - Admin chọn chức năng mà muốn quản lý môn học.
    - Admin điền vào các trường thông tin bắc buộc trên website (nếu có).
    - Admin thực thi chức năng đã chọn.
  + Hâu điều kiêṇ: Hệ thống lưu lại thông tin sau khi quản lý.
* U5: Quản lý thông tin. (mở rộng U5.2,U5.3,U5.5,U5.5,U5.6,U5.5)
  + Điều kiện tiên quyết: Đăng nhập.
    - Admin chọn chức năng mà muốn quản lý.
    - Admin điền vào các trường thông tin bắc buộc trên website (nếu có).
    - Admin thực thi chức năng đã chọn.
  + Hâụ điều kiêṇ: Hệ thống lưu lại thông tin sau khi quản lý.
* U5.2: Lấy tất cả. (mở rộng của U5)
  + Điều kiện tiên quyết: Đăng nhập.
    - Admin chọn chức năng Lấy tất cả.
    - Admin lấy tất cả các thông tin theo yêu cầu.
    - Admin hệ thống thực thi chức năng đã chọn.
  + Hâu điều kiêṇ: Hệ thống lưu lại thông tin.
* U5.3: Tìm kiếm. (mở rộng của U5 )
  + Điều kiện tiên quyết: Đăng nhập.
    - Admin hoặc bạn đọc chọn chức năng Tìm kiếm.
    - Admin hoặc bạn đọc tiến hành tìm kiếm theo các hạng mục như tên bài học, danh mục...
    - Admin hệ thống thực thi chức năng đã chọn.
  + Hâu điều kiêṇ: Hệ thống lưu lại thông tin.
* U5.5: Thêm. (mở rộng của U5)
  + Điều kiện tiên quyết: Đăng nhập.
    - Admin chọn chức năng Thêm.
    - Admin tiến hành thêm các thông tin theo yêu cầu.
    - Admin hệ thống thực thi chức năng đã chọn.
  + Hâu điều kiêṇ: Hệ thống lưu lại thông tin.
* U5.5: Xóa. ( mở rộng của U5)
  + Điều kiện tiên quyết: Đăng nhập.
    - Admin chọn chức năng Xóa.
    - Admin tiến hành xóa các thông tin theo yêu cầu.
    - Admin hệ thống thực thi chức năng đã chọn.
  + Hâu điều kiêṇ: Hệ thống lưu lại thông tin.
* U5.6: Sửa. ( mở rộng của U5)
  + Điều kiện tiên quyết: Đăng nhập.
    - Admin chọn chức năng Sửa.
    - Admin tiến hành sửa các thông tin theo yêu cầu.
    - Admin hệ thống thực thi chức năng đã chọn.
  + Hâu điều kiêṇ: Hệ thống lưu lại thông tin.
* U5.5: Hiển thị. ( mở rộng của U5)
  + Điều kiện tiên quyết: Đăng nhập.
    - Admin hoặc bạn đọc chọn chức năng Hiển thị thông tin.
    - Admin hoặc bạn đọc tiến hành thêm các thông tin theo yêu cầu.
    - Admin hoặc bạn đọc thực thi chức năng đã chọn.
  + Hâu điều kiêṇ: Hệ thống lưu lại thông tin.

**4.1.1.2 Danh sách các ứng viên:**

* User (Người dùng).
* Post (Bài viết).
* Category (Danh mục).
* Subject (Môn học).
* Coment (Bình luận).
* FeedBack(Phản hồi, đánh giá).
  + - 1. **Sơ đồ lớp mức phân tích.**



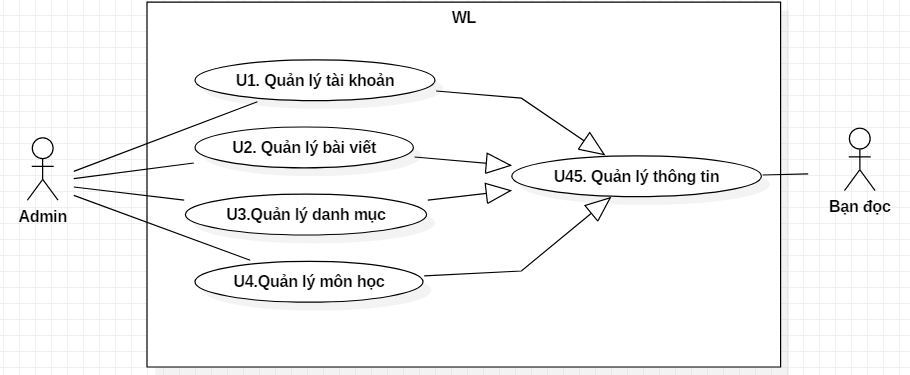
Hình 4 2 Sơ đồ lớp của website WeLearn

**4.2 Phân tích kiến trúc hệ thống**

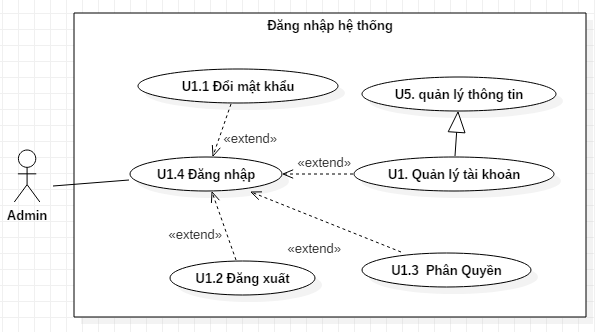
Giao diêṇ người dùng

Các phác thảo giao diện người dùng cho WeLearn, được đưa ra với sự giúp đỡ của khách hàng, được thể hiện như sau:

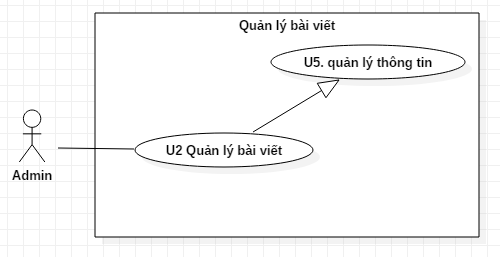
Sơ đồ Use case :



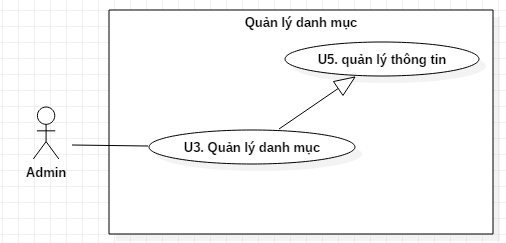
Hình 4 3 Use Case tổng quát hệ thống



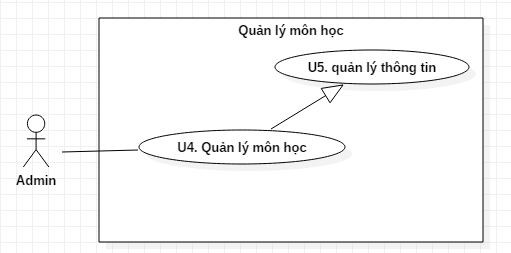
Hình 4 4 Use case quản lý tài khoản



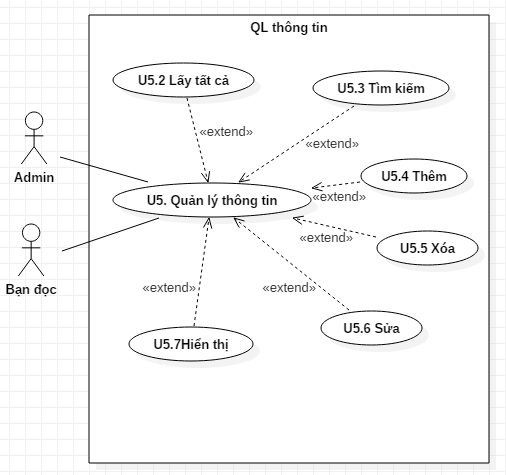
Hình 4 5 Use case quản lý bài viết



Hình 4 6 Use case quản lý danh mục



Hình 4 7 Use case quản lý môn học



Hình 4 8 Use case quản lý thông tin

**4.2 Tổng quan Use case**

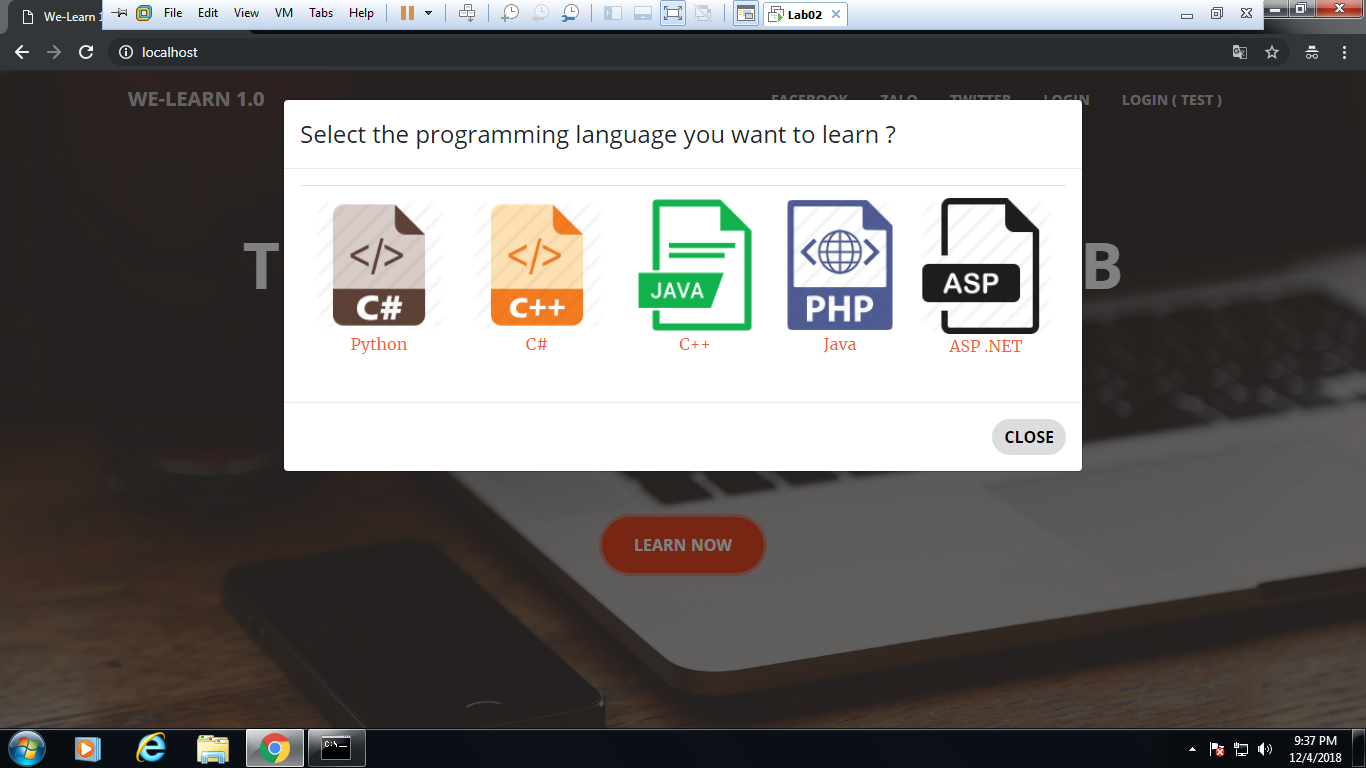
Khi truy cập hệ thống, người dùng có thể đăng nhập hệ thống hoặc không ( U1.5 ), sau đó tìm một bài viết ( U3.3 ) và mở lên để đọc ( U3.8 ). Trong trường hợp tìm kiếm ( U5.3 ), người dùng cụ thể hóa các hạng mục, tìm theo ID danh mục về ngôn ngữ lập trình mà họ quan tâm. Trong trường hợp tìm kiếm(U5.3), Dù bằng cách nào, sau mỗi lần đaṭ đươc̣ sẽ được hiển thị các kết quả tập các bài viết phù hợp (U3.8), cùng với các thông tin cơ bản như tên, danh mục,…v.v...

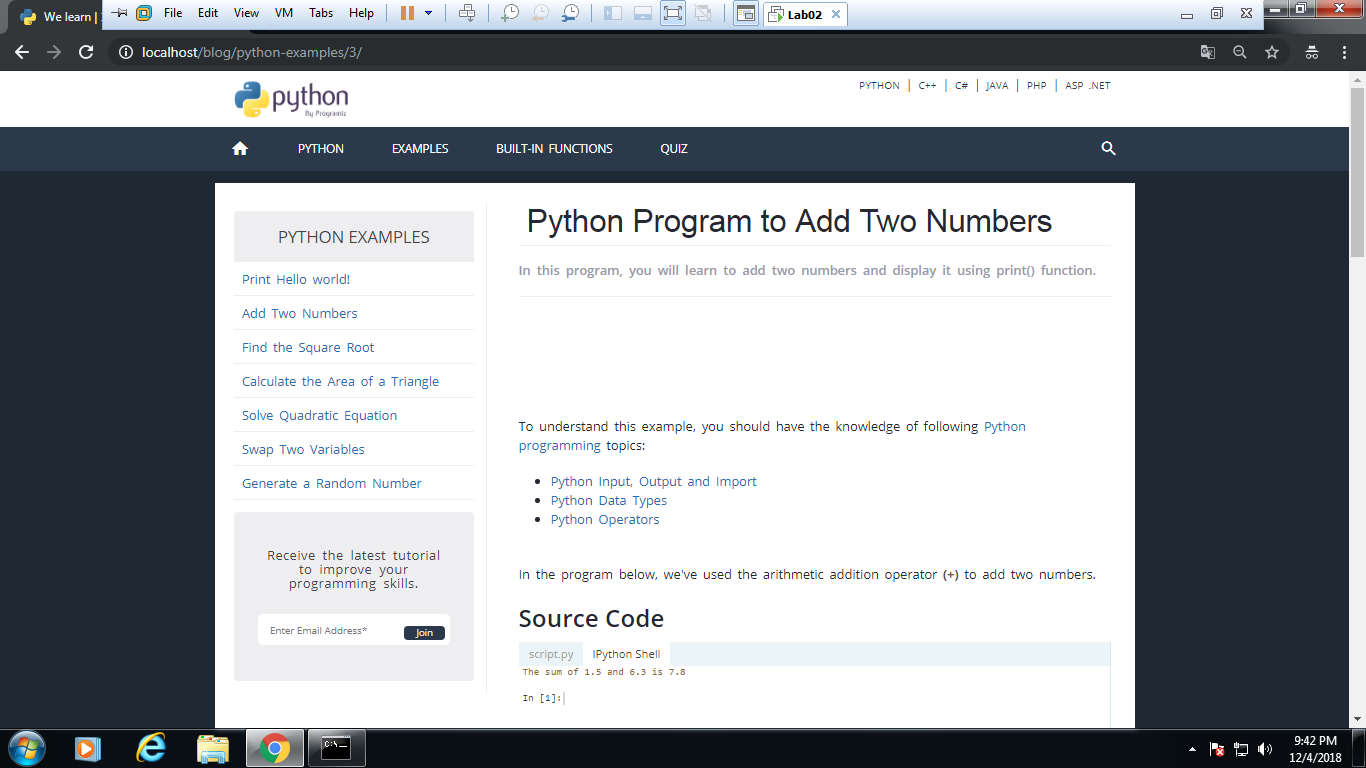
Người dùng có 2 loại: admin ( đăng nhập ) và bạn đọc ( không cần đăng nhập ). Admin sử dụng chức năng quản lý bài viết (U3) và quản lý danh mục (U5) để quản lý nội dung trang web.

**4.3 Phân tích bố cục giao diện**

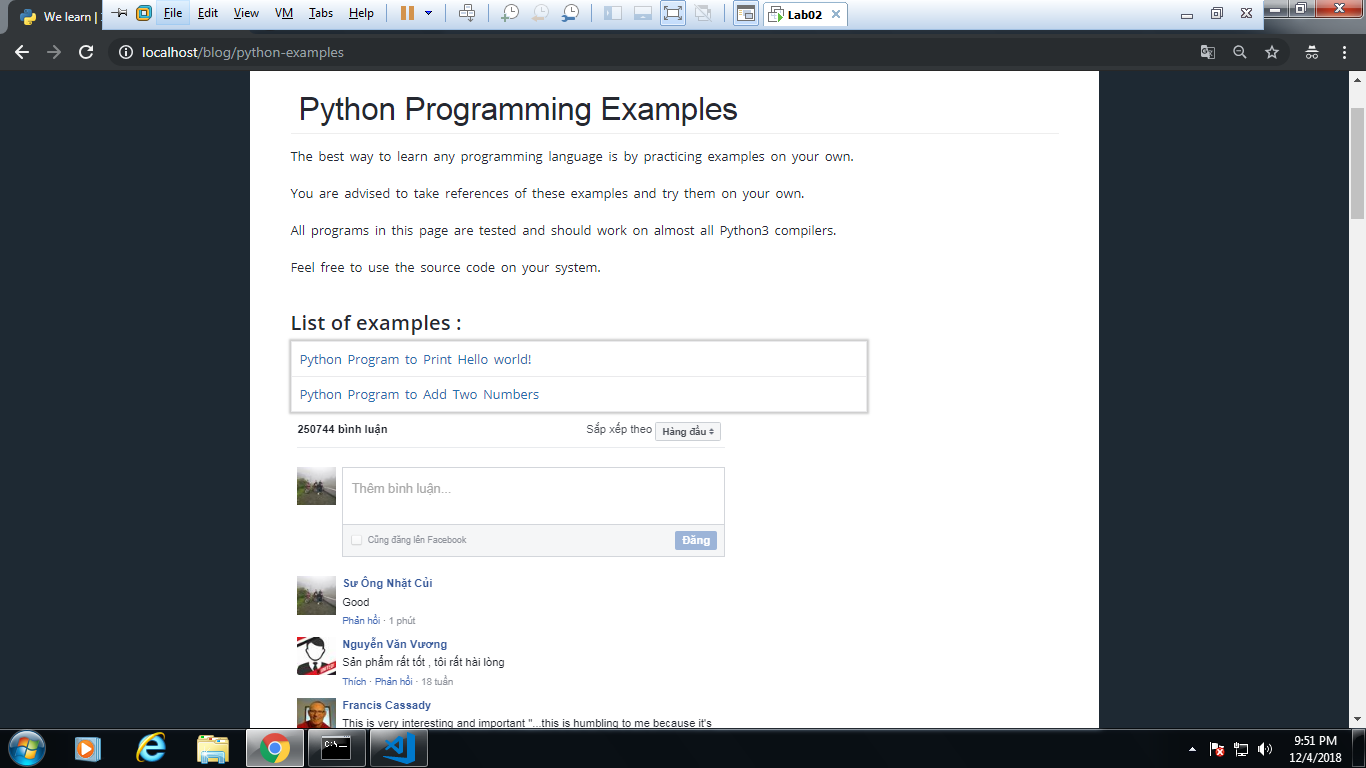
- Giao diện người dùng:

* Giao diện bài viết
* Học viên có thể bình luận trên website
* Editor Python mạnh mẽ
* Nhiều môn học, danh mục,… trên website



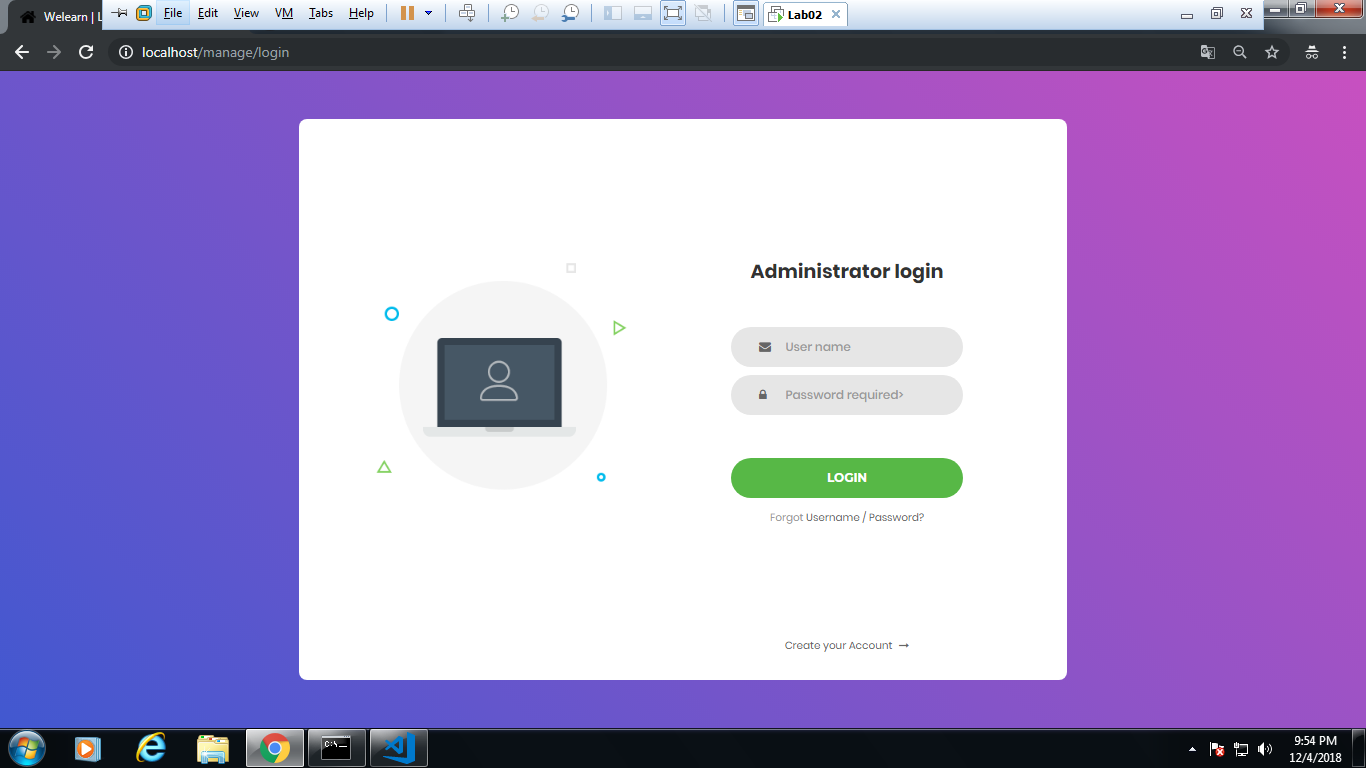
Hình 4 9 Giao diện người dùng khi vào trang

Hình 4 10 Giao diện bài viết

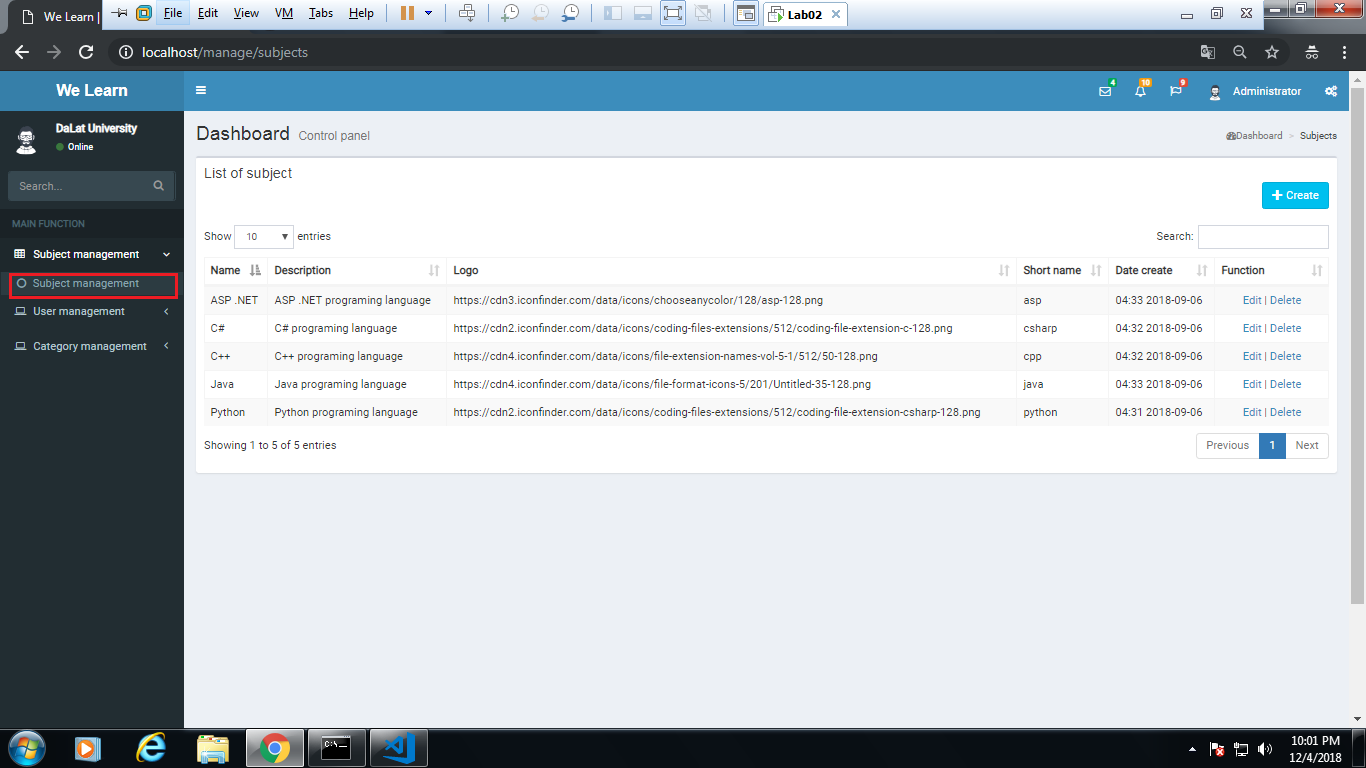


Hình 4 11 Giao diện bài viết và chức năng bình luận facebook

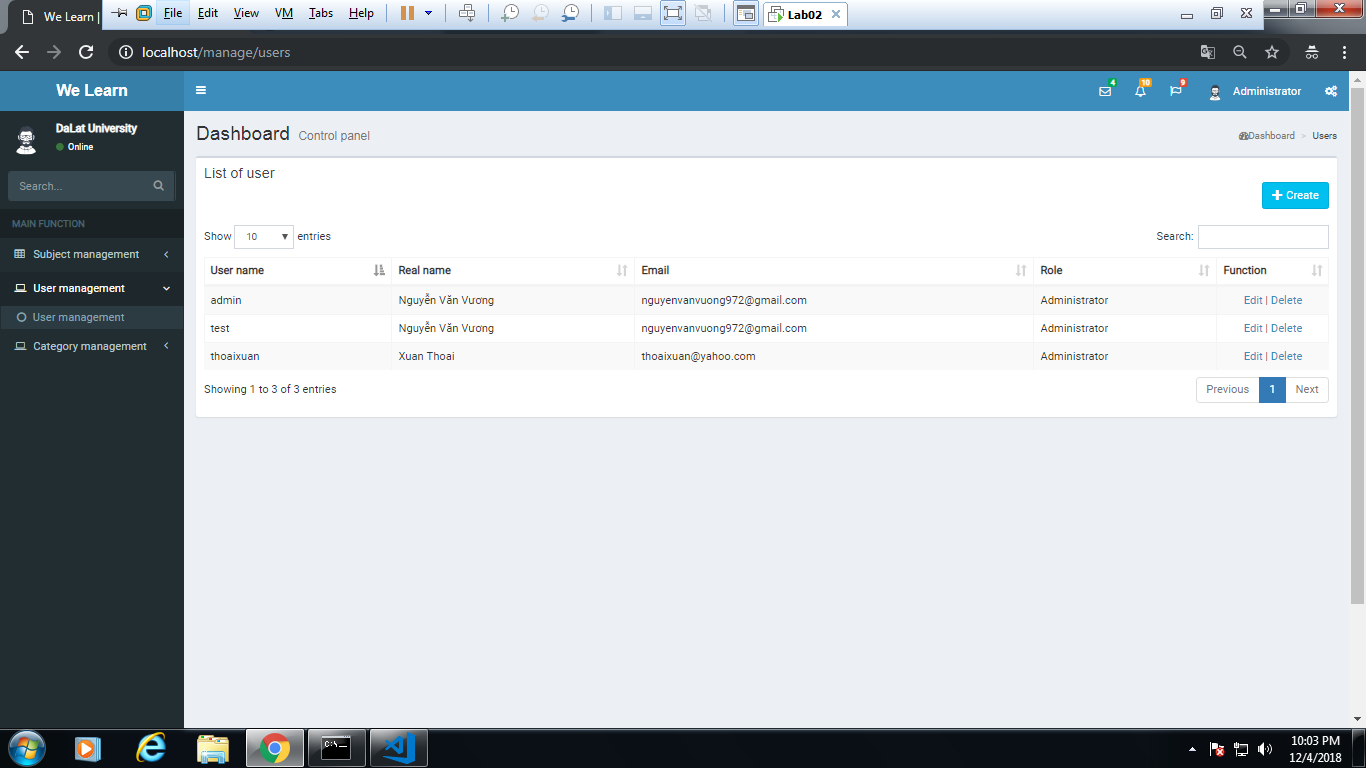
* Giao diện quản trị viên:
* Quản lý các bài viết
* Quản lý danh mục
* Công cụ soạn thảo bài viết mạnh mẽ
* Giao diện quản lý đẹp, tùy biến.



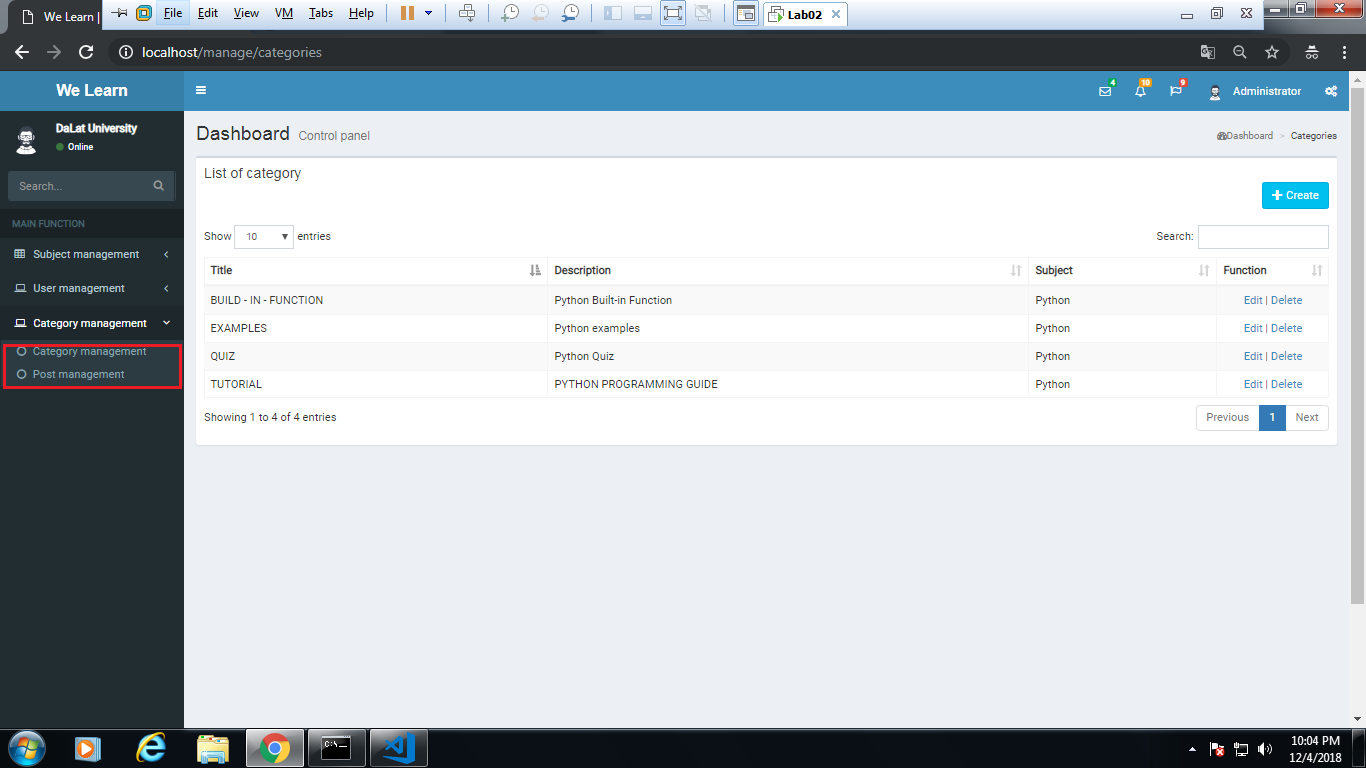
Hình 4 12 Giao diện trang đăng nhập của quản trị viên



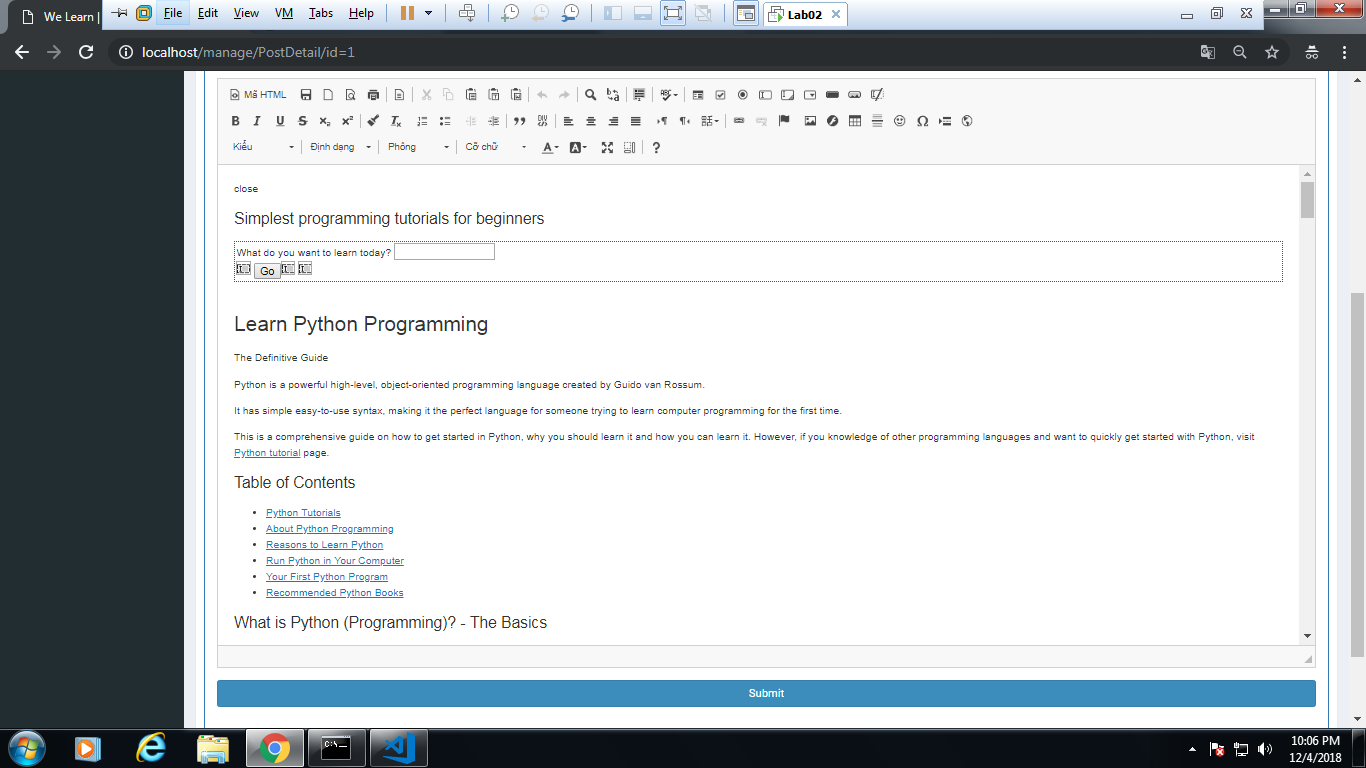
Hình 4 13 Giao diện quản lý các đối tượng môn học



Hình 4 14 Giao diện quản lý tài khoản Admin



Hình 4 15 Giao diện quản lý danh mục bài viết và bài viết



Hình 4 16 Giao diện CKEditor

* 1. **Phân tích các chức năng của ứng dụng**

### 4.4.1 Cách hoạt động của trang web.

* Người quản trị viên tạo các bài viết trong các danh mục, môn học.
* Học viên truy cập trang web, chọn môn học.
* Học viên chọn môn học và tìm kiếm / chọn bài viết để học.
* Học viên có thể chạy trực tiếp các ví dụ và xem kết quả ngay trên web, tăng sự hứng thú cho Học viên.

### 4.4.2 Danh sách actors.

* Admin: Người có quản trị hệ thống với hệ thống.
* Bạn đọc: Là những người học viên truy cập website để xem tài liệu.

### 4.4.3 Danh sách Use case.

* U1: Quản lý tài khoản : Admin sau khi đăng nhập vào hệ thống, có thể quản lý được tài khoản theo phân quyền của Admin (kế thừa U5,bao gồm U1.1,U1.2,U1.3,U1.4).
  + U1.4: Đăng nhập: Người dùng có tài khoản trong hệ thống có thể tiến hành đăng nhập vào hệ thống.
  + U1.1: Đổi mật khẩu: Những người dùng có tài khoản đăng nhập vào hệ thống có thể thay đổi mật khẩu của tài khoản mình nếu muốn.
  + U1.2: Đăng xuất:Người dùng có tài khoản trong hệ thống sau khi sử dụng xong có thể đăng xuất để thoát ra khởi hệ thống.
  + U1.3: Phân quyền: Admin có thể phân quyền truy cập và quản lý cho các tài khoản khác.
* U2: Quản lý bài viết: Admin có thể dùng để quản lý các các bàn (Kế thừa U5, mở rộng bởi U1.4).
* U3: Quản lý danh mục: Admin quản lý danh mục bài viết trên website ( kế thừa U5, mở rộng bởi U1.4)
* U4: Quản lý môn học: Admin quản lý môn học trên website ( kế thừa U5, mở rộng bởi U1.4)
* U5: Quản lý thông tin: Admin hoặc bạn đọc dùng để quản lý các thông tin tài khoản của mình trong quyền hạn ( bao gồm U5.2, U5.3, U5.5, U5.5, U5.6, U5.7, mở rộng bởi U1.5 )
* U5.2 Lấy tất cả: lấy tất cả thông tin các thông tin
* U5.3 Tìm kiếm: Tìm kiếm tất cả những thông tin
* U5.4 Thêm:Thêm một thông tin
* U5.5 Xóa: Xóa một thông tin
* U5.6 Sửa: Sửa một thông tin
* U5.7 Hiển thị: hiển thị những thông tin khi có yêu cầu

## **Triển khai ứng dụng và kiểm thử**

|  |  |
| --- | --- |
| Thời gian | Công việc |
| **5/7 - 5/8/2018** | **Tìm hiểu cách cài đặt môi trường Python. Tìm hiểu về Python, cú pháp, cấu trúc, đặc điểm nổi bật của Python.** |
| **15/8 – 8/9/2018** | **Tổng hợp và viết báo cáo.** |
| **17/9 – 26/9/2018** | **Tìm hiểu về Framework Django của Python** |
| **28/9 – 4/10/2018** | **Phân tích hệ thống website, database. Xây dựng template mẫu, database ban đầu.** |
| **7/10 -21/10/2018** | **Xây dựng website We-Learn bước đầu cơ bản các chức năng người dùng và admin** |
| **23/10- 26/10/2018** | **Tìm kiếm và fix lỗi. Hoàn thiện website và các chức năng còn thiếu.** |
| **28/10 – 31/11/2018** | **Tổng hợp, sửa lỗi, cải tiến và hoàn thiện sản phẩm** |

# Chương 5: Tổng kết đề tài

## **5.1 Hướng phát triển**

* Xây dựng hoàn thiện các chức năng cho Website
* Tạo ra một môi trường cho học sinh, sinh viên học tập và tìm hiểu nhiều hơn về CNTT.
* Xây dựng 1 cộng đồng cùng học tập, trao đổi, trau dồi kiến thức. (Ở Việt Nam có rất ít tài liệu, hướng dẫn về ngôn ngữ Python)
* Xây dựng thêm các chức năng tương tác người dùng như: nhập email để nhận thông báo về bài viết mới nhất của Website, lưu bài viết đã đọc, xây dựng chức năng đóng góp bài viết của người dùng.
* Củng cố và hoàn thiện website.
* Tìm hiểu cấu trúc của một Website viết bằng Django.
* Tìm hiểu tập lệnh, cách sử dụng mô hình MVC trong Django.
* Phát triển một Website học code viết bằng Django FrameWork.
* Củng cố khả năng phân tích hệ thống website và database website.
* Sử dụng django framework tạo ra một trang web học lập trình trực tuyến cho lập trình viên.

## **5.2 Kết luận**

* Python đã được ứng dụng trong IOT rất nhiều, từ các trang nổi tiếng còn khẳng định Python ( Stack Overflow ) là tương lai của IOT, đặc biệt là trí tuệ nhân tạo AI.
* Bước đầu hoàn thành xây dựng website bằng Framework Django
* Các framework Python còn tăng cường bảo mật khi loại bỏ các lỗ hổng bảo mật như: SQL injection, cross-site scripting, cross-site request forgery và clickjacking.
* Các công ty, tổ chức và chính phủ đã sử dụng Django để xây dựng mọi thứ - từ hệ thống quản lý nội dung đến mạng xã hội đến nền tảng điện toán khoa học.