TRƯỜNG ĐẠI HỌC ĐIỆN LỰC

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CHUYÊN ĐỀ HỌC PHẦN**

**NGÔN NGỮ LẬP TRÌNH PYTHON**

**ĐỀ TÀI:**

**XÂY DỰNG WEBSITE DIỄN ĐÀN HỌC TẬP**

|  |  |
| --- | --- |
| **Giảng viên hướng dẫn** | **: PHẠM ĐỨC HỒNG** |
| **Sinh viên thực hiện** | **: NGUYỄN MINH THÀNH**  **: TRẦN HOÀI NAM** |
| **Ngành** | **: CÔNG NGHỆ THÔNG TIN** | |
| **Lớp tín chỉ** | **: D13CNPM1** | |
| **Khóa** | **: 2018-2022** | |

***Hà Nội, tháng 6 năm 2021***

**PHIẾU CHẤM ĐIỂM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên sinh viên** | **Nội dung thực hiện** | **Điểm** | **Chữ ký** |
| 1 | Nguyễn Minh Thành  18810310345 | Báo cáo, phân tích, lập trình |  |  |
| 2 | Trần Hoài Nam  18810310310 | Kiểm thử |  |  |

|  |  |  |
| --- | --- | --- |
| **Họ và tên giảng viên** | **Chữ ký** | **Ghi chú** |
| Giảng viên chấm 1: |  |  |
| Giảng viên chấm 2: |  |  |

# 

MỤC LỤC

[Các hình vẽ, bảng biểu 5](#_Toc87177903)

[LỜI MỞ ĐẦU 6](#_Toc87177904)

[CHƯƠNG 1. 7](#_Toc87177905)

[GIỚI THIỆU CHUNG 7](#_Toc87177906)

[1.1. Ngôn ngữ lập trình Python 7](#_Toc87177907)

[1.2. Tìm hiểu cơ chế server-side rendering trong phát triển web 7](#_Toc87177908)

[*1.2.1.* *Khái niệm* 7](#_Toc87177909)

[*1.2.2. Ưu và nhược điểm của Server Side Rendering* 8](#_Toc87177910)

[1.3. Giới thiệu web framework Django và mô hình MVT 9](#_Toc87177911)

[*1.3.1.* *Giới thiệu về Django* 9](#_Toc87177912)

[*1.3.2.* *Mô hình MVT* 9](#_Toc87177913)

[CHƯƠNG 2. 11](#_Toc87177914)

[QUẢN LÝ DỰ ÁN 11](#_Toc87177915)

[2.1. Ước lượng dự án 11](#_Toc87177916)

[*2.1.1 Ước lượng chi phí dự án* 11](#_Toc87177917)

[*2.1.2. Ước lượng thời gian* 13](#_Toc87177918)

[*2.1.3.* *Ước lượng người tham gia* 14](#_Toc87177919)

[2.2. Lập lịch và theo dõi 14](#_Toc87177920)

[CHƯƠNG 3 17](#_Toc87177921)

[XÂY DỰNG WEBSITE DIỄN ĐÀN HỌC TẬP VỚI DJANGO 17](#_Toc87177922)

[3.1.Phân tích yêu cầu 17](#_Toc87177923)

[*3.1.1 Chức năng* 17](#_Toc87177924)

[*3.1.2. Yêu cầu đặt ra* 17](#_Toc87177925)

[3.2. Phân tích các chức năng cần thiết 19](#_Toc87177926)

[*3.2.1. Các chức năng chính của website* 19](#_Toc87177927)

[*3.2.2. Liệt kê một số chức năng chính* 20](#_Toc87177928)

[3.3. Lập trình trang web diễn đàn học tập 22](#_Toc87177929)

[*3.3.1. Các modules xử lý logic* 22](#_Toc87177930)

[*3.3.2. Một số hàm xử lý logic* 23](#_Toc87177931)

[3.4. Cơ sở dữ liệu của trang web 29](#_Toc87177932)

[*3.4.1. Các bảng trong CSDL và thông tin của chúng* 30](#_Toc87177933)

[*3.4.2. Các thực thể và thuộc tính:* 31](#_Toc87177934)

[CHƯƠNG 4. 32](#_Toc87177935)

[CHƯƠNG TRÌNH THỰC NGHIỆM 32](#_Toc87177936)

[4.1.Giới thiệu trang web diễn đàn học tập 32](#_Toc87177937)

[4.2. Giao diện hoàn thiện của trang web 32](#_Toc87177938)

[KẾT LUẬN 35](#_Toc87177939)

[Tài liệu tham khảo 36](#_Toc87177940)

# **Các hình vẽ, bảng biểu**

*Danh mục hình vẽ*

Hình 1.1. Logo của Python………………………………………………………...7

Hình 1.2. Mô hình đơn giản về Server Side Rendering……………………………8

Hình 1.3. Mô hình MVT của Django……………………………………………..10

Hình 3.1. Sơ đồ tương tác trang web……………………………………………...19

Hình 3.2. Sơ đồ khối chức năng đăng kí………………………………………….21

Hình 3.3. Sơ đồ khối của chức năng bình luận……………………………………22

Hình 4.1. Giao diện trang chủ……………………………………………………..32

Hình 4.2. Màn hình đăng nhập……………………………………………………33

Hình 4.3. Giao diện của một phòng:………………………………………………33

Hình 4.4. Giao diện tạo phòng…………………………………………………….34

*Danh mục bảng biểu*

Bảng 2.1. Ước lượng chi phí dự án……………………………………………….11

Bảng 2.2. Ước lượng thời gian……………………………………………………13

Bảng 2.3. Lập lịch và theo dõi…………………………………………………….15

# 

# **LỜI MỞ ĐẦU**

Chúng ta có thể nói rằng thế kỷ 21 đã và đang chứng kiến sự phát triển mạnh mẽ của ngành Công nghệ thông tin. Công nghệ thông tin đã và đang thay đổi thế giới một cách nhanh chóng và từng bước kéo nền tri thức của nhân loại xích lại gần nhau hơn. Đó là nhờ việc phát triển hệ thống website trên toàn thế giới.

Ngày nay, Website đã đóng một vai trò quan trọng đối với con người chúng ta từ giải trí cho đến quảng cáo, thương mại, quản lý…Website (thương mại điện tử) sẽ dần dần thay thế những phương thức kinh doanh cũ trong các doanh nghiệp bởi tính ưu việt mà Website mang lại như: nhanh hơn, rẽ hơn, tiện dụng hơn, hiệu quả hơn và không bị giới hạn không gian và thời gian.

Và Python là một ngôn ngữ lập trình rất phù hợp cho việc phát triển một website hoàn chỉnh, đây là một ngôn ngữ đa mục đích và được rộng rãi sử dụng, và rất dễ học, cùng với các thư viện và framework để phát triển một website hoàn chỉnh giúp cho việc lập trình, phát triển web trở lên dễ dàng hơn. Và Django là một trong số đó, Django là framework phát triển web phổ biến nhất của Python, với chúng ta có thể xây dựng bất kì trang web nào.

Chính vì vậy nhóm em quyết định “Xây dựng Website diễn đàn học tập” nhằm tạo ra một nơi để kết nối mọi người giúp mọi người trao đổi và đặt câu hỏi và trả lời từ đó giúp ích cho việc học tập và nghiên cứu không những của học sinh mà còn của giáo viên.

# **CHƯƠNG 1.**

# **GIỚI THIỆU CHUNG**

* 1. **Ngôn ngữ lập trình Python**

Python là một ngôn ngữ lập trình bậc cao cho các mục đích lập trình đa năng, do Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. 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. Vào tháng 7 năm 2018, Van Rossum đã từ chức Leader trong cộng đồng ngôn ngữ Python sau 30 năm lãnh đạo.

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.



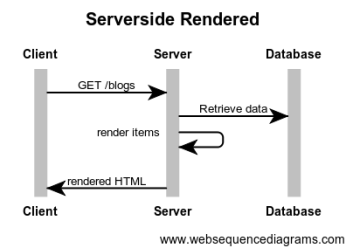
*Hình 1.1. Logo của Python*

Ban đầu, Python được phát triển để chạy trên nền Unix. Nhưng rồi theo thời gian, Python dần mở rộng 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.

* 1. **Tìm hiểu cơ chế server-side rendering trong phát triển web**
     1. ***Khái niệm***

Cơ chế server-side rendering đã được dùng từ rất lâu Gọi nó là server-side rendering là vì phần lớn logic sẽ được xử lý ở server:

* Khi người dùng vào một trang web, trình duyệt sẽ gửi GET request tới web server
* Web server sẽ nhận request, đọc dữ liệu từ database.
* Web server sẽ render HTML, trả về cho browser để hiển thị cho người dùng

[](https://toidicodedao.files.wordpress.com/2018/07/diagram-serversiderendered.png)

*Hình 1.2. Mô hình đơn giản về Server Side Rendering*

Một số tính chất của cơ chế server side rendering:

* Logic từ đơn giản (validation, đọc dữ liệu) cho đến phức tạp (phân quyền, thanh toán) đều nằm ở phía server
* Logic để routing – chuyển trang nằm ở server
* Logic để render – hiển thị trang web cũng nằm ở server nốt

***1.2.2. Ưu và nhược điểm của Server Side Rendering***

Ra đời đã lâu, server side rendering đã được sử dụng, cải tiến rất nhiều (Caching, CDN). Những ưu điểm của cơ chế này:

* Initial load nhanh, dễ otpimize, vì toàn bộ dữ liệu đã được xử lý ở server. Client chỉ việc hiển thị.
* Các web framework từ xưa đến nay đều hỗ trợ cơ chế này
* Dễ hiểu và dễ code hơn. Developer chỉ cần code 1 project web là được, không cần phải [tách ra front-end và back-end](https://toidicodedao.com/2017/08/17/lam-the-nao-de-tro-thanh-web-developer-phan-1/)
* SEO tốt vì khi bot của Google, Bing vào web sẽ thấy toàn bộ dữ liệu dưới dạng HTML.
* Chạy được trên phần lớn mọi trình duyệt, kể cả disable JavaScript vẫn chạy tốt

Thế nhưng, bên cạnh đó, server side rendering cũng có một số nhược điểm:

* Mỗi lần người dùng chuyển trang là site phải load lại nhiều lần, gây khó chịu
* Nặng server vì server phải xử lý nhiều logic và dữ liệu. Có thể sử dụng caching để giảm tải.
* Tốn băng thông vì server phải gửi nhiều dữ liệu thừa và trùng  (HTML, header, footer). Có thể [sử dụng CDN](https://toidicodedao.com/2018/05/29/cdn-la-gi-tang-toc-do-tai-he-thong/) để giảm tải.
* Tương tác không tốt như Client Side rendering vì trang phải refresh, load lại nhiều lần.

Cơ chế server side rendering vẫn còn được sử dụng ở **rất rất nhiều site.** Do vậy, chỉ cần hiểu rõ và nắm vững cơ chế này là bạn đã có thể làm một trang web đầy đủ tính năng rồi.

* 1. **Giới thiệu web framework Django và mô hình MVT** 
     1. ***Giới thiệu về Django***

Django là một khuôn khổ web mã nguồn mở và miễn phí dựa trên Python tuân theo mô hình kiến ​​trúc model – template – views. Nó được duy trì bởi Django Software Foundation, một tổ chức độc lập được thành lập tại Hoa Kỳ với tư cách là tổ chức phi lợi nhuận 501.

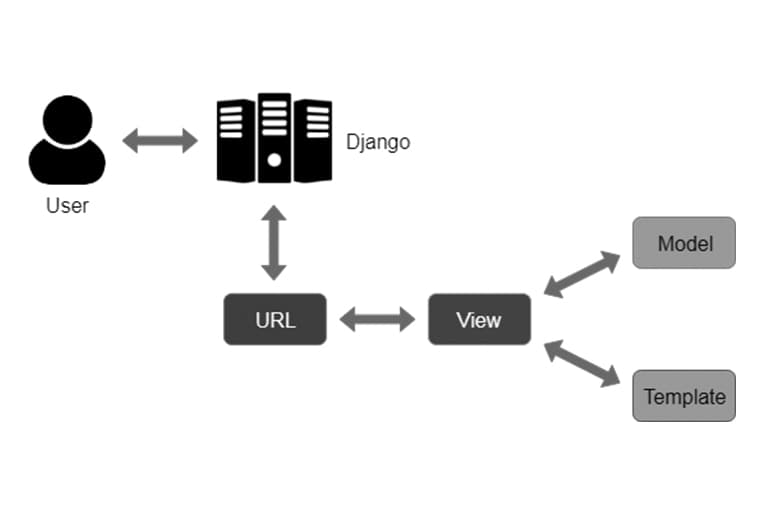
Mục tiêu chính của Django là dễ dàng tạo ra các trang web phức tạp, dựa trên cơ sở dữ liệu. Khuôn khổ nhấn mạnh khả năng tái sử dụng và "khả năng cắm" của các thành phần, ít mã hơn, ít khớp nối, phát triển nhanh và nguyên tắc không lặp lại chính mình. Python được sử dụng xuyên suốt, ngay cả cho cài đặt, tệp và mô hình dữ liệu. Django cũng cung cấp giao diện tạo, đọc, cập nhật và xóa quản trị tùy chọn được tạo động thông qua xem xét nội dung và được định cấu hình thông qua các mô hình quản trị viên.

Một số trang web nổi tiếng sử dụng Django bao gồm Instagram, Mozilla, Disqus, Bitbucket, Nextdoor và Clubhouse.

* + 1. ***Mô hình MVT***

Django sử dụng mô hình MVT (Model-View-Template) thay vì sử dụng [mô hình MVC](https://niithanoi.edu.vn/mo-hinh-mvc-la-gi.html) (Model-View-Controller).

Mô hình MVT được sử dụng trong khi tạo một ứng dụng với Tương tác người dùng.

  
  
*Hình 1.3. Mô hình MVT của Django*

Mô hình này thì bao gồm code HTML với Django Templage Language (DTL).  
Controller là mã được viết để kiểm soát sự tương tác giữa Model và View và Django dễ dàng chăm sóc nó.  
Bất cứ khi nào người dùng người request, nó xử lý request của người dùng đó bằng Model, View và Template.  
Nó hoạt động như một Controller để kiểm tra xem nó có khả dụng hay không bằng cách ánh xạ URL và nếu URL ánh xạ thành công thì View sẽ bắt đầu tương tác với Model và gửi lại Template cho người dùng dưới dạng respone.

# **CHƯƠNG 2.**

# **QUẢN LÝ DỰ ÁN**

## **2.1. Ước lượng dự án**

Công việc đầu tiên của người quản lý dự án là ước lượng - ước lượng về kích cỡ, chi phí, thời gian tiến hành dự án. Việc này thông thường được tiến hành bằng cách phân rã phần mềm cần phát triển thành các khối nhỏ và áp dụng các kinh nghiệm (các thông số như kích cỡ, chi phí, năng lực nhân viên...) đối với các phần mềm đã phát triển để ước lượng, đánh giá công việc.

### ***2.1.1 Ước lượng chi phí dự án***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Giai đoạn | Công việc | Mô tả | Thời gian | Chi phí |
| Quản lý dự án | Khảo sát yêu cầu dự án | Thu thập các yêu cầu tổng quát về dự án | 2 ngày | 200.000 vnđ |
| Khởi tạo dự án | Thông báo, triển khai dự án, lập project | 1 ngày | 1.000.000 vnđ |
| Lập kế hoạch phạm vi dự án | Lập bản kế hoạch phạm vi dự án | 5 ngày | 1.000.000 vnđ |
| Viết báo cáo tổng kết | Tổng kết lại toàn bộ công việc thành báo cáo cuối cùng. | Giai đoạn cuối của dự án | 0 |
| Rút kinh nghiệm | Rút kinh nghiệm cho đợt dự án sau | Giai đoạn cuối của dự án | 0 |
| Phân tích và thiết kế hệ thống | Đặc tả chi tiết các yêu cầu từ người dùng | Từ yêu cầu thu được từ phần khảo sát, mô tả cụ thể hơn về những yêu cầu cần thiết | 2 ngày | 200.000 vnđ |
| Mô tả kiến trúc hệ thống bằng các sơ đồ | Xây dựng kiến trúc phần mềm bằng các sơ đồ use case, trình tự, hoạt động (dùng Rational rose, astah) | 5 ngày | 500.000 vnđ |
| Thiết kế cơ sở dữ liệu hệ thống | Xây dựng cấu trúc cơ sở dữ liệu cho phần mềm (gồm các bảng cụ thể) | 3 ngày | 500.000 vnđ |
| Thiết kế giao diện | Xây dựng các form css, bootstrap, …. | 4 ngày | 1.500.000 vnđ |
| Viết bản phân tích hệ thống chi tiết | Tổng hợp các báo cáo để cho ra bản phân tích thiết kế hoàn chỉnh. | 2 ngày | 500.000 vnđ |
| Module thống kê sản phẩm | Phân tích yêu cầu cụ thể cho các form | Từ những thông tin đã thu thập ở trên lập kế hoạch xây dựng chi tiết những nhiệm vụ của quản lý cửa hàng | 2 ngày | 200.000 vnđ |
| Thiết kế form chung | - Xây dựng form đăng nhập  - Xây dựng form thống kê nhập xuất hàng, hình thức thống kê hàng hóa.  - Xây dựng các nút chức năng: thống kê, in phiếu TK | 3 ngày | 2.000.000 vnđ |
| Viết code | Xây dựng các chức năng chính cho form | 5 ngày | 4.000.000 vnđ |
| Cài đặt module | Thực hiện chạy thử module đã xây dựng | 1 ngày | 0 |
| Kiểm thử module | - kiểm thử giao diện, kiểm tra xem giao diện có đạt yêu cầu không, về hình thức bố trí giao diện của phần mềm có bố trí hợp lý hay không  - kiểm thử khả năng thống kê nhập xuất hàng… | 3 ngày | 1.000.000 vnđ |
| Viết báo cáo về module | Xây dựng báo cáo mô tả chi tiết về module | 1 ngày | 0 |
| Tích hợp và hoàn thiện sản phẩm | Tích hợp các form | Lắp ráp hoàn chỉnh các form , thư mục chức năng thành chương trình thống nhất để chạy thử | 2 ngày | 1.500.000 vnđ |
| Kiểm thử tích hợp toàn hệ thống | Test tổng quan toàn bộ hệ thống | 1 ngày | 1.000.000 vnđ |
| Thực hiện fix các lỗi tồn tại | Fix các lỗi phát sinh trong khi test | 2 ngày | 1.500.000 vnđ |

*Bảng 2.1. Ước lượng chi phí dự án*

### ***2.1.2. Ước lượng thời gian***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Giai đoạn | Công việc | Mô tả | Ngày bắt đầu | Ngày kết thúc |
| 1 | Ước lượng dự án | Đánh giá tính khả thi của dự án | 1-10-2021 | 3-10-2021 |
| Khảo sát dự án  Ưoc lượng dự án | 4-10-2021 | 7-10-2021 |
| 2 | Thiết kế , phân tích hệ thống | Vẽ các biểu đồ usecase, sơ đồ trình tự… | 8-10-2021 | 12-10-2021 |
| 3 | Lập trình | Xây dựng CSDL | 13-10-2021 | 14-10-2021 |
| Xây dựng chức năng đăng nhập | 14-10-2021 | 17-10-2021 |
| Xây dựng các chức năng hệ thống | 18-10-2021 | 30-10-2021 |
| 4 | Kiểm thử, đóng gói | Kiểm tra lỗi thiết kế  Đóng gói phần mềm | 1-11-2021 | 2-11-2021 |

*Bảng 2.2. Ước lượng thời gian*

* + 1. ***Ước lượng người tham gia***

Do tính chất đơn giản của chương trình web và chỉ gồm các chức năng cần thiết chính nên dự án này chỉ cần 1 người tham gia.

* 1. **. Lập lịch và theo dõi**

Trong khi dự án chưa hoàn thành hoặc chưa bị hủy bỏ thì thực hiện lặp đi lặp lại các công việc sau:

Lập lịch thực hiện dự án

Thực hiện các hoạt động theo lịch trình

Theo dõi sự tiến triển của dự án, so sánh với lịch trình

Đánh giá lại các tham số của dự án

Lập lại lịch thực hiện dự án cho các tham số mới

Thỏa thuận lại các ràng buộc và sản phẩm bàn giao của mỗi mốc thời gian

Nếu có vấn đề nảy sinh thì xem xét lại các kĩ thuật khởi đầu đưa ra các biện pháp cần thiết

Với dự án xây dựng phần mềm phát triển trong 2 tháng này , là khá nhiều cho việc phát triển ta có thể lập lịch làm việc không quá 3 buổi một tuần để làm dự án, và theo dõi tiến độ dự án một cách thường xuyên.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Danh mục | Hoạt động | Công việc | Mô tả | Kế thừa hoạt động | Ngày bắt đầu | Ngày kết thúc | Hoàn thành | Chưa hoàn thành |
| 1 | Xác lập, kháo sát dự án | 1.1 | Xác lập, kháo sát dự án |  | 1-10-2021 | 4-10-2021 | x |  |
| 1.2 | Xác định tính khả thi của dự án | 1.1 | 4-10-2021 | 5-10-2021 | x |  |
| 1.3 | Khảo sát thời gian | 1.2 | 5-10-2021 | 6-10-2021 | x |  |
| 1.4 | Khảo sát khối lương công việc | 1.3 | 7-10-2021 | 7-10-2021 | x |  |
| 1.5 | Khảo sát hệ thống | 1.4 | 7-10-2021 | 8-10-2021 | x |  |
| 2 | Phân tích thiết kế hệ thống | 2.1 | Phân tích hệ thống | 1.4 | 9-10-2021 | 10-10-2021 | x |  |
| 2.2, | Thiết kế hệ thống | 2.1 | 10-10-2021 | 11-10-2021 | x |  |
| 2.3, | Kiểm tra tính thực tế | 2.2 | 11-10-2021 | 11-10-2021 | x |  |
| 3 | Lập trình | 3.1 | Xây dựng cơ sở dữ liệu | 2.3 | 11-10-2021 | 13-10-2021 | x |  |
| 3.2 | Lập trình tính năng đăng nhập | 3.1 | 13-10-2021 | 14-10-2021 | x |  |
| 3.3 | Lập trình các chức năng hệ thống | 3.2 | 15-10-2021 | 30-10-2021 | x |  |
| 4 |  | 4.1 | Kiểm thử, đóng gói | 3.3 | 1-11-2021 | 2-11-2021 | x |  |

*Bảng 2.3. Lập lịch và theo dõi*

# **CHƯƠNG 3**

# **XÂY DỰNG WEBSITE DIỄN ĐÀN HỌC TẬP VỚI DJANGO**

## **3.1.Phân tích yêu cầu**

Ý tưởng là tạo trang web có giao diện giống trang web discord.com, dễ sứ dụng, đầy đủ chức năng. Thân thiện với người sử dụng.

### ***3.1.1 Chức năng***

Đây là một website bán và giới thiệu sản phẩm của cửa hàng, công ty đến người

tiêu dùng với các chi tiết mặt hàng với giá cả chính xác. Có các chứa năng sau:

- Cho phép cập nhật thông tin vào CSDL.

- Hiển thị danh sách các topic

- Hiển thị phòng user đã tạo

- Đăng kí, đăng nhập

- Hiển thị các người tham gia, các dòng bình luận trong phòng

- Danh mục topic, phòng để người dùng dễ chọn lựa

- Hiển thị thông tin người dùng

- Quản lý người dùng

- Cập nhật thông tin phòng

- Cập nhật topic

- Cập nhật thông tin người dùng

- Bình luận trong các phòng

### ***3.1.2. Yêu cầu đặt ra***

a. Thiết bị và phần mềm

- Máy tính có thể thiết kế được web

- Hệ quản trị cơ sở dữ liệu

- Máy chạy được các trang web

b. Yêu cầu trang Web

Hệ thống có 2 phần:

Thứ nhất : Phần Người dùng :

Người dùng là những người có nhu cầu sử dụng trang web cho việc học tập, họ sẽ tìm kiếm các topic, phòng cần thiết từ hệ thống và tìm câu trả lời hoặc đặt câu trả lời. Vì thế phải có

các chức năng sau:

- Hiển thị danh sách các topic, phòng để người dùng xem, lựa chọn

và thảo luận , hỏi hoặc trả lời.

- Người dùng cũng có thể tạo mới phòng để hỏi hoặc thảo luận những vấn để chưa được để cập trong trang web.

- Sau khi người dùng chọn và tạo phòng thì phải hiện lên

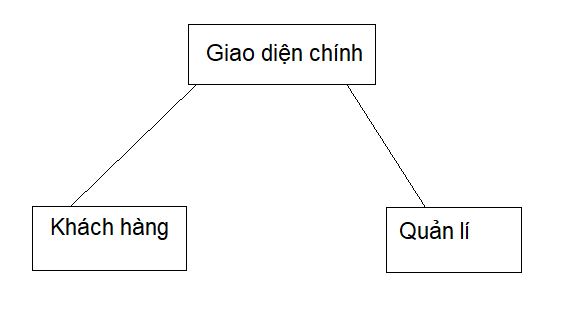
phòng được tạo trên trang chủ để người dùng có thể hỏi và tìm kiếm câu trả lời nhanh nhất

Thứ hai : Dành cho người quản trị: Người làm chủ ứng dụng có

quyền kiềm soát mọi hoạt động của hệ thống. Người này được cấp username

và password để đăng nhập hệ thống thực hiện chức năng của mình:

- Chức năng cập nhật, sửa, xoá các topic, phòng, Nó đòi hỏi sự chính xác.



*Hình 3.1. Sơ đồ tương tác trang web*

- Tiếp nhận kiểm tra các topic, room được tạo. Hiển thị thông tin.

- Thống kê theo ngày, khoảng thời gian.

Ngoài các chức năng trên thì trang Web phải được thiết kế sao cho dễ hiểu,

giao diện mang tính dễ dùng đẹp mắt và làm sao cho người dùng thấy được

thông tin cần tìm, cung cấp các chức năng, giao diện trực quan để người dùng dễ dàng sử dụng.

. Điều quan trọng là phải đảm bảo an toàn tuyệt đối thông tin người dùng trong quá sử dụng. Đồng thời trang web phải luôn đổi mới, hấp dẫn

## **3.2. Phân tích các chức năng cần thiết**

Sau khi khảo sát hiện trạng của hệ thống và xây dựng hệ thống thì em

đã đưa ra các phương hướng cho việc “Xây dựng website diễn đàn học tập”.

- Qua quá trình khảo sát, phân tích thực trạng và xây dựng hướng phát

triển cho hệ thống mới em đã xây dựng cấu trúc của hệ thống và xây dựng Website diễn đàn học tập

### ***3.2.1. Các******chức năng chính của website***

- Hiển thị danh sách các topic

- Tìm kiếm theo tên topic hoặc phòng

- Hiển thị phòng user đã tạo

- Đăng kí, đăng nhập

- Hiển thị các người tham gia, các dòng bình luận trong phòng

- Danh mục topic, phòng để người dùng dễ chọn lựa

- Hiển thị thông tin người dùng

- Quản lý người dùng

- Cập nhật thông tin phòng

- Cập nhật topic

- Cập nhật thông tin người dùng

- Bình luận trong các phòng

- Hiển thị các bình luận gần nhất trên trang chủ

### ***3.2.2. Liệt kê một số chức năng chính***

* Chức năng ĐĂNG KÝ THÀNH VIÊN

- Người dùng có thể đăng ký để trở thành người dùng thường xuyên (thành

viên) của công ty và được cấp một tài khoản người dùng ( Account ) để đăng

nhập khi cần đặt hàng ,thanh toán, hỗ trợ kỹ thuật …

- Đầu vào :

-Các thông tin cá nhân của người dùng

+ User name ( tài khoản ngưòi dùng)

+ Password(mật khẩu)

+ Họ tên đầy đủ ( Full name )

+ Email

+ Một số thông tin khác .

- Các quá trình thực hiện :

+ Module nhập thông tin cá nhân của người dùng .

+ Module xác nhận thông tin vừa nhập .

+ Module thông báo cho người dùng và yêu cầu nhập lại tài khoản

ngưòi dùng nếu tài khoản đó đã có người đăng ký .

+ Module lưu thông tin người dùng vừa đăng ký vào CSDL .

- Đầu ra :

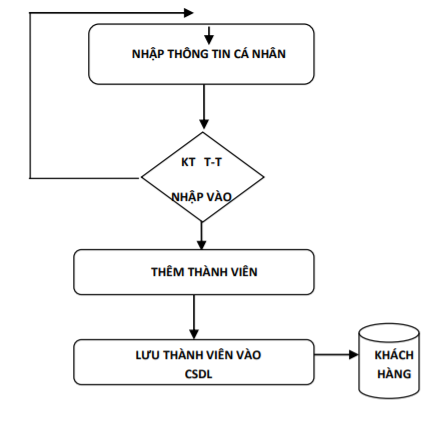
+ Thông báo cho người dùng và yêu cầu nhập lại tài khoản hoặc

Email nếu tài khoản hoặc Email đó đã tồn tại trong CSDL .

+ Thông báo và yêu cầu nhập lại thông tin trong trường hợp thông

tin nhập vào không chính xác , không phù hợp .

+ Thông báo kết qủa đăng ký.



*Hình 3.2. Sơ đồ khối chức năng đăng kí*

* Chức năng bình luận

- Cho phép tất cả mọi người (user đã có tài khoản trong database) bình luận trong một phòng của một người dùng.

- Đầu vào :

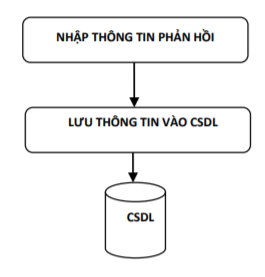
-các thông tin phản hồi trong phòng.

- Các quá trình thực hiện :

+ Module nhập bình luận.

+ Module lưu thông tin góp ý vào CSDL .

- Đầu ra : -Thông tin chi tiết về góp ý ,phản hồi chờ nhà quản trị trả lời .



*Hình 3.3. Sơ đồ khối của chức năng bình luận*

## **3.3. Lập trình trang web diễn đàn học tập**

Mục tiêu của em là trang web có thiết kế phù hợp với một trang bán hàng, có đầy đủ các tính năng với thiết kế đơn giản, thân thiện với người dùng.

### ***3.3.1. Các modules xử lý logic***

Project này có các thành phần(modules) cơ bản ở tầng logic, sử lý dữ liệu như sau:

**admin.py**: dùng để kết nối cơ sở dữ liệu lên trang quản trị của Django

**forms.py**: dùng để tạo form dựa vào các bảng trong csdl cho templates(phần client-side rendering)

**models.py**: tạo cơ sở dữ liệu ở đây.

**urls.py**: tạo các url router cho trang web

**views.py**: tạo các hàm xử lý logic để trao đổi dữ liệu giữa client-server, lấy dữ liệu tự cơ sở dữ liệu để nạp vào templates đồng thời xử lý dữ liệu người dùng nhập vào. Đây cũng là View trong mô hình MVT tương ứng với controller trong mô hình MVC.

Vì đây là báo cáo về ngôn ngữ lập trình Python, ta sẽ chỉ xem xét đến tầng View trong mô hình MVT

### ***3.3.2. Một số hàm xử lý logic***

Mỗi khi người dùng truy cập vào một url nhất định thì các hàm được gọi từ đó lấy dữ liệu tự cơ sở dữ liệu để nạp vào templates đồng thời xử lý dữ liệu người dùng nhập vào.

Một số hàm xư lý logic chính trong module views.py:

\*Các object, biến chuyền vào template được lưu ở dạng dictionary trong tham số thứ 3 của hàm trả về render(request, html\_file, **dictionary**)

*a, Trang đăng nhập*

+ Làm các công việc sau:

-Kiểm tra trang có phải trang login không bằng cách truyền vào một biến vào template

- Nếu user đã đăng nhập thì chuyển hướng về trang chủ

- Nếu người dùng nhấn vào button đăng nhập thì sẽ kiểm tra xem tài khoản có trong CSDL không

- Nếu người dùng có trong CSDL thì trả về một object

- Nếu đúng thông tin trong CSDL thì tạo một session cho phiên đăng nhập và chuyển hướng về trang chủ.

+code:

def loginPage(request):

page = 'login'

if request.user.is\_authenticated:

return redirect('home')

if request.method == 'POST':

email = request.POST.get('email').lower()

password = request.POST.get('password')

try:

user = User.objects.get(email=email)

except:

messages.error(request, 'User does not exist')

user = authenticate(request, email=email, password=password)

if user is not None:

login(request, user)

return redirect('home')

else:

messages.error(request, 'Username or password doesnt exist')

context = {'page': page}

return render(request, 'base/login\_register.html', context)

*b, Trang chủ*

+ Hàm có các chức năng sau:

* + Lấy dữ liệu tiềm kiếm của user và xử lý thông qua method GET
  + Lấy instance đã filter thông qua dữ liệu tìm kiểm từ client của bảng Room để truyền vào template
  + Lấy tất cả instance có sẵn của bảng Message để truyền vào template
  + Lấy 5 instance đầu tiên của bảng Topic để truyền vào template
  + Lấy số các instance trong bảng Room gán vào một biến để truyền vào template
  + Thông qua method GET, mỗi khi user bấm vào một topic thì sẽ hiển thì phòng cho các topic đó.

+code:

def home(request):

q = request.GET.get('q') if request.GET.get('q') != None else ''

rooms = Room.objects.filter(

Q(topic\_\_name\_\_icontains=q) |

Q(name\_\_icontains=q) |

Q(description\_\_icontains=q)

)

room\_messages = Message.objects.all()

topics = Topic.objects.all()[0:5]

room\_count = rooms.count()

room\_messages = Message.objects.filter(Q(room\_\_topic\_\_name\_\_icontains=q))

context = {'rooms': rooms, 'topics': topics,

'room\_count': room\_count, 'room\_messages': room\_messages}

return render(request, 'base/home.html', context)

*c, Trang của một phòng*

+ Hàm có các chức năng sau:

* + Truyền object của 1 instance trong bảng phòng vào 1 biến dựa vào tham số trong url
  + Truyền tất cả message của phòng đó
  + Truyền tất cả thành viên thạm gia của phòng đó
  + Nếu user nhập bình luận thông qua method POST thì một instance của bảng message được tạo.

+code:

def room(request, pk):

room = Room.objects.get(id=pk)#query method!

room\_messages = room.message\_set.all()

participants = room.participants.all()

if request.method == 'POST':

message = Message.objects.create(

user=request.user,

room=room,

body=request.POST.get('body')

)

room.participants.add(request.user)

return redirect('room', pk=room.id)

context = {'room': room, 'room\_messages': room\_messages, 'participants': participants}

return render(request, 'base/room.html', context)

*d, Trang thông tin người dùng*

+ Hàm có các chức năng sau:

* + Lấy object của 1 instance trong bảng phòng vào 1 biến để truyền vào template dựa vào tham số trong url.
  + Truyền vào template tất cả các phòng người đó đã tạo
  + Truyền tất cả các message của người đó
  + Truyền tất cả instance của topic

+code:

def userProfile(request, pk):

user = User.objects.get(id=pk)

rooms = user.room\_set.all()

room\_messages = user.message\_set.all()

topics = Topic.objects.all()

context = {'user': user, 'rooms': rooms,

'room\_messages': room\_messages, 'topics': topics}

return render(request, 'base/profile.html', context)

*e, Trang tạo phòng*

*\*Thêm, sửa, xóa phòng là quyền của người dùng đã đăng nhập, nên cần thêm hàm decorator cho các hàm để kiểm tra xem người dùng đã đăng nhập chưa, nếu chưa thì chuyển hướng về trang đăng nhập:*

@login\_required(login\_url='/login')

+ Hàm có các chức năng sau:

* + Truyền vào template RoomForm() để tự động generate một form dựa trên các thuộc tính của bảng
  + Truyền tất cả các object của topic đã tạo
  + Nếu người dùng bấm tạo phòng thông qua method POST thì sẽ tạo một object cho bảng Room, nếu người dùng nhập mới topic thì tạo mới một object cho Topic sau đó chuyển hướng về trang chủ.

+code:

def createRoom(request):

form = RoomForm()

topics = Topic.objects.all()

if request.method == 'POST':

topic\_name = request.POST.get('topic')

topic, created = Topic.objects.get\_or\_create(name=topic\_name)

Room.objects.create(

host= request.user,

topic=topic,

name=request.POST.get('name'),

description=request.POST.get('description')

)

return redirect('home')

context= {'form': form, 'topics': topics}

return render(request, 'base/room\_form.html', context)

*f, Trang sửa phòng*

+ Hàm có các chức năng sau:

* + Truyền một instance của Room nếu id bằng tham số trong url vào template room\_form.html
  + Truyền vào template RoomForm(instance=room) để tự động generate một form dựa trên các thuộc tính của bảng, trong đó tất cả các dữ liệu đã được nhập sẵn được lấy từ instance room ở trên
  + Nếu người dùng nhấn update thì instance đó sẽ được cập nhật

+code:

def updateRoom(request, pk):

room = Room.objects.get(id=pk)

form = RoomForm(instance=room)

topics = Topic.objects.all()

if request.user != room.host:

return HttpResponse('Your not allowed here!')

if request.method == 'POST':

topic\_name = request.POST.get('topic')

topic, created = Topic.objects.get\_or\_create(name=topic\_name)

room.name = request.POST.get('name')

room.topic = topic

room.description = request.POST.get('description')

room.save()

return redirect('home')

context = {'form': form, 'topics': topics, 'room': room}

return render(request, 'base/room\_form.html', context)

*g, Trang xóa phòng*

+ Hàm có các chức năng sau:

* + Truyền vào template delete.html một instance của Room nếu id bằng tham số trong url
  + Nếu user xác nhận xóa thì xóa instance đó trong CSDL.

+code:

def deleteRoom(request, pk):

room = Room.objects.get(id=pk)

if request.user != room.host:

return HttpResponse('Your not allowed here!')

if request.method == 'POST':

room.delete()

return redirect('home')

return render(request, 'base/delete.html', {'obj': room})

## **3.4. Cơ sở dữ liệu của trang web**

Sau khi tạo xong các bảng dữ liệu ta tiến hành thiết lập các mối quan hệ cho chúng. Một thực thể trong thực tế không tồn tại độc lập với các thực thể khác. Có sự liên kết qua lại giữa các thực thể khác nhau. Các thực thể trong mô hình này đều có loại liên kết là “Một” - “Nhiều”. Mỗi lần xuất của thực thể phĩa “Một” tương ứng với nhiều lần xuất của thực thể phĩa “Nhiều”. Nhưng mỗi lần xuất của thực thể phía “Nhiều” chỉ tương ứng duy nhất một lần xuất của thực thể phĩa “Một”. Khi muốn tạo liên kết giữa hai bảng dữ liệu ta dùng khoá chính của bảng này làm khoá ngoại lai cho bảng kia.

Quan hệ giữa các thực thể còn đảm bảo nguyên tắc toàn vẹn dữ liệu. Các bản ghi thuộc phĩa “Một” được cập nhật trước và chỉ có thể xoá khi đã xoá hết các bản ghi thuộc phĩa “Nhiều” liên kết với nó.

### ***3.4.1. Các bảng trong CSDL và thông tin của chúng***

#Bảng thông tin người dùng

class User(AbstractUser):

name = models.CharField(max\_length=200, null=True)

email = models.EmailField(unique=True, null=True)

bio = models.TextField(null=True)

avatar = models.ImageField(null=True, default="avatar.svg")

USERNAME\_FIELD = 'email'

REQUIRED\_FIELDS = []

#Bảng Topic

class Topic(models.Model):

name = models.CharField(max\_length=200)

def \_\_str\_\_(self):

return self.name

#Bảng Room

class Room(models.Model):

host = models.ForeignKey(User, on\_delete=models.CASCADE, null=True)

topic = models.ForeignKey(Topic, on\_delete=models.SET\_NULL, null=True)

name = models.CharField(max\_length=200)

description = models.TextField(null=True, blank=True)

participants = models.ManyToManyField(User, related\_name='participants', blank=True)

updated = models.DateTimeField(auto\_now=True)

created = models.DateTimeField(auto\_now\_add=True)

#Bảng Message

class Message(models.Model):

user = models.ForeignKey(User, on\_delete=models.CASCADE)

room = models.ForeignKey(Room, on\_delete=models.CASCADE)

body = models.TextField()

updated = models.DateTimeField(auto\_now=True)

created = models.DateTimeField(auto\_now\_add=True)

### ***3.4.2. Các thực thể và thuộc tính:***

* User: ID, name, email, bio ,avatar
* Topic: TopicID, name
* Room: RoomID , host, topic, name, description, participants, updated, created
* Message: MessageID, user ,room, body, updated, created

# **CHƯƠNG 4.**

# **CHƯƠNG TRÌNH THỰC NGHIỆM**

## **4.1.Giới thiệu trang web diễn đàn học tập**

Với việc áp dụng python , một ngôn ngữ lập trình mục đích chung(general-purpose language) vào để lập trình trang web, đã giúp cho việc quản lí, chỉnh sửa và thiết kế trang web trở lên dễ dàng hơn. Trang web có bố cục rõ ràng, dễ nhìn, dễ sử dụng cấp đầy đủ tính năng giúp cho việc bình luận, tìm kiếm topic,…

Đối với người sử dụng, chương trình đã đưa ra được các chức năng chính sau:

-Các cách tìm kiếm, tra cứu theo các chủ đề khác nhau.

-Giao diện dễ nhìn, trực quan.

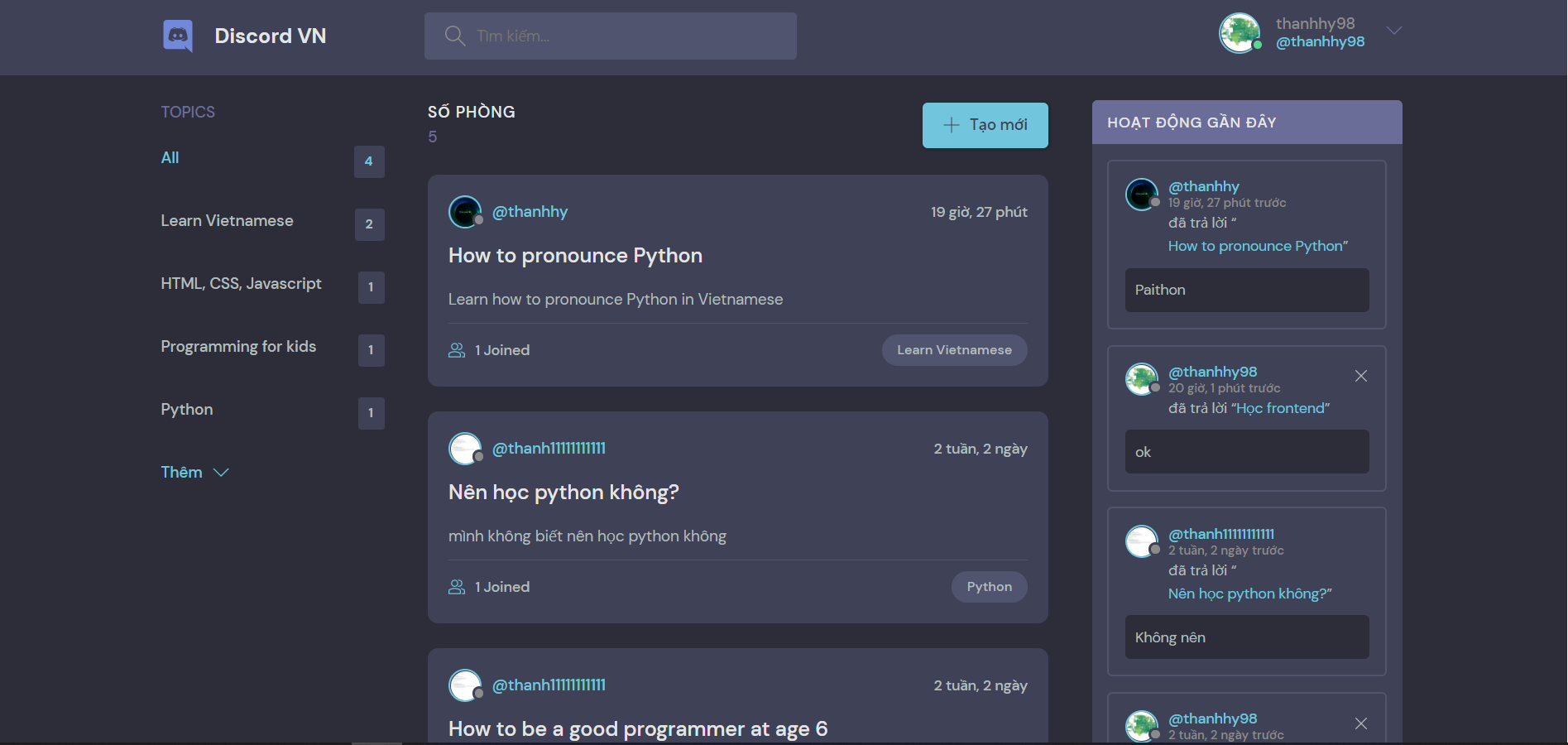
-Cung cấp đầy đủ các thông tin, chức năng của một phòng thảo luận.

-Có đầy đủ các tính năng như một chatapp thời gian thực để cho người dùng bình luận, giao tiếp với nhau trong một phòng.

-Có thể chạy trên nhiều hệ điều hành

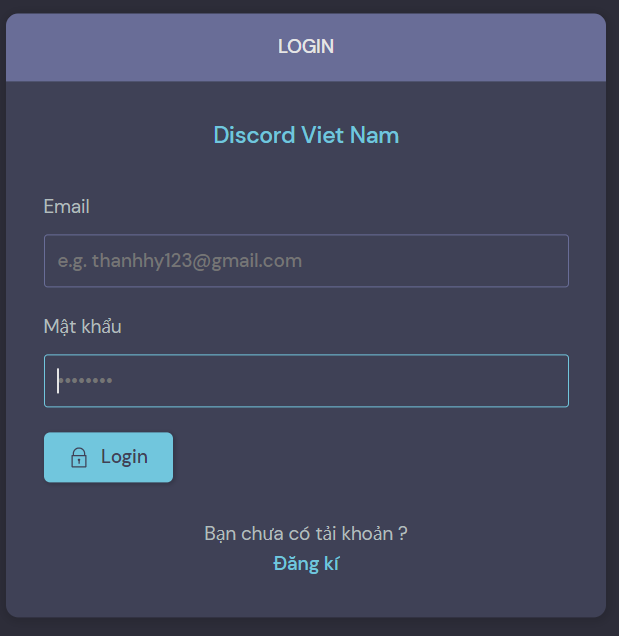
## **4.2. Giao diện hoàn thiện của trang web**

Dưới đây là một số hình ảnh về trang web bán đồ điện tử đã hoàn thiện.



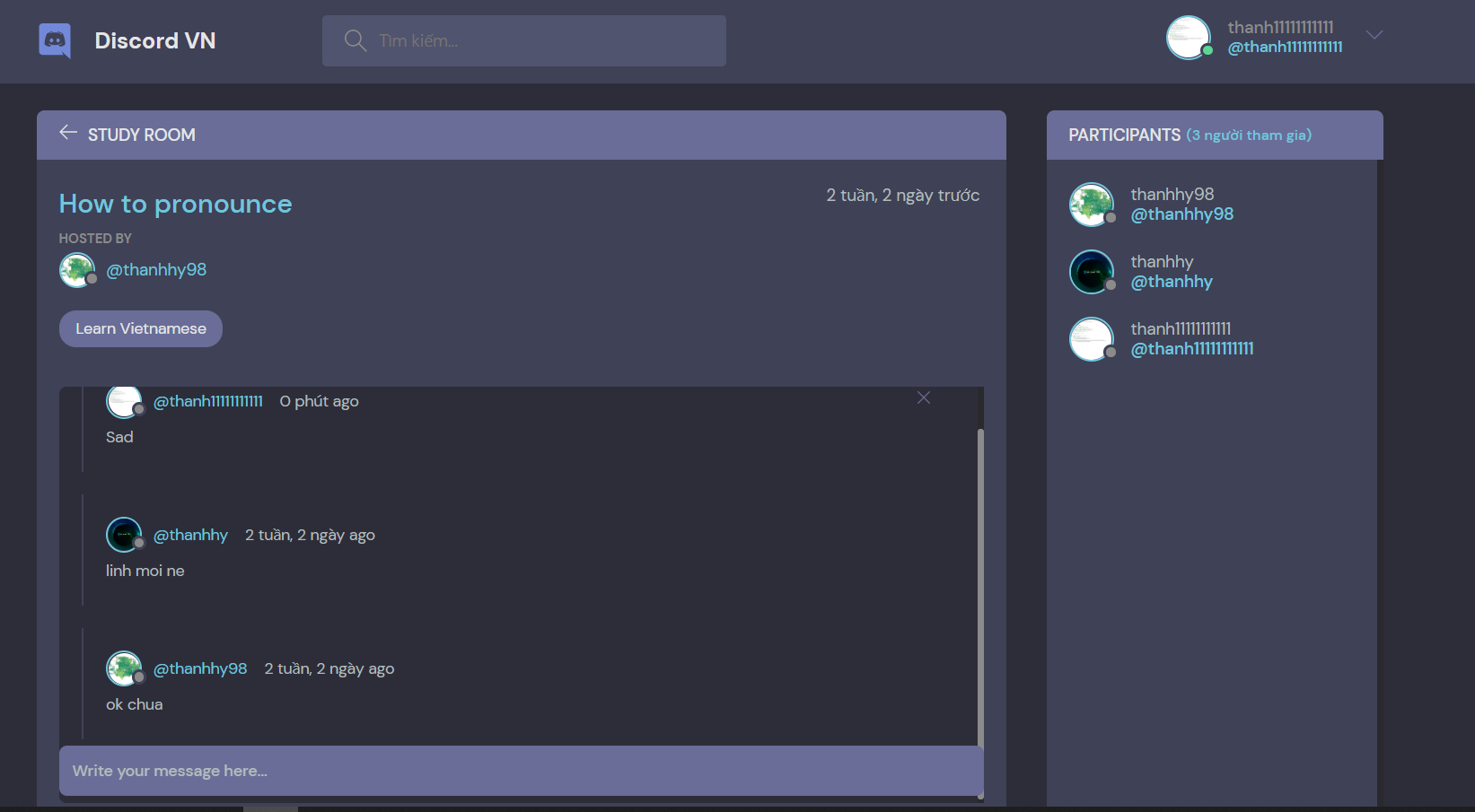
*Hình 4.1. Giao diện trang chủ*

*Màn hình đăng nhập:*



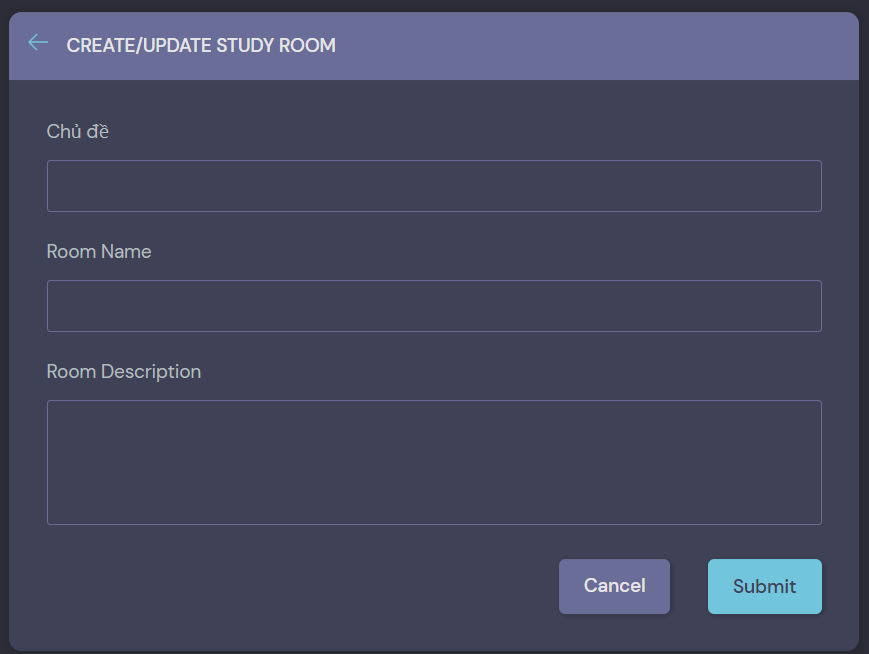
*Hình 4.2. Màn hình đăng nhập*

*Giao diện của một phòng:*



*Hình 4.3. Giao diện của một phòng:*

*Giao diện tạo phòng:*



*Hình 4.4. Giao diện tạo phòng*

# **KẾT LUẬN**

Ngày nay, ứng dụng công nghệ thông tin và việc tin học hóa được xem là một trong những yếu tố mang tính quyết định trong hoạt động của các chính phủ, tổ chức, cũng như của các công ty, nó đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ.

Đối với lập trình web cũng vậy, không chỉ ứng dụng công nghệ thông tin thông thường để làm ra các trang web mà chúng ta còn cần phải áp dụng những công nghệ tiến tiến nhất để dễ dàng thiết kế cũng như quản lí trang web có đầy đủ tiêu chí trên, em đã thấy được hiệu quả trong quản lí cũng như thiết kế trang web với mô hình Model- Views- Template này, chúng mang lại hiệu quả cao trong việc quản lí cơ sở dữ liệu cũng như xây dưng trang web.

Chính vì vậy, với đề tài ” Xây dựng trang web diễn đàn học tập” đã giúp chúng ta có thêm kinh nghiệm cũng như kiến thức nhất định trong việc lập trình, thiết kế một trang web có đầy đủ các chức năng cơ bản. Mong rằng qua đề tài này ta đã có thể xây dựng các website bán hàng hoàn chỉnh, cải thiện kĩ năng lập trình web qua đó tạo ra các trang web với nhiều tính năng mới lạ và có đầy đủ các tính năng cần cho một website diễn đàn điện tử giúp việc trao đổi kiến thức thuận tiện hơn.

**Tài liệu tham khảo**

1. Mark Lutz, Learning Python, 5th Edition, NXB O'Reilly Media, 2013
2. Nigel George, Mastering Django, NXB Packt, 2016
3. Ben Shaw, Web Development with Django, NXB Packt, 2021