|  |
| --- |
| Trường Đại Học Công Nghệ Sài Gòn  **Khoa Công Nghệ Thông Tin**  ---oOo---  **BÁO CÁO ĐỌC SÁCH**  **BEGINNING FLUTTER**  **&**  **DJANGO 4 BY EXAMPLE**  Người hướng dẫn : **Đoàn Trình Dục**  Sinh viên thực hiện :  **Nguyễn Thanh Tùng \_ DH52006150**  **Trần Nguyễn Gia Bảo \_ DH52005690**  TP. HỒ CHÍ MINH – NĂM 2023 |

**MỤC LỤC**

**Phần 1 : Beginning Flutter**

Chương 1 : Giới thiệu Flutter

Chương 2 : Tạo ứng dụng Hello World

Chương 3 : Học ngôn ngữ Dart cơ bản

Chương 4: Tạo một project mẫu

Chương 5: Tìm hiểu cây Widget

Chương 6: Sử dụng các widget thông dụng

Chương 7 : Thêm hiệu ứng và ứng dụng

Chương 8 : Tạo điều hướng ứng dụng

Chương 9 : Tạo danh sách và hiệu ứng cuộn

Chương 10 : Xây dựng Layouts

Chương 11 : Áp dụng tính tương tác

Chương 12 : Viết Platform-Native Code

Chương 13 : Lưu dữ liệu với Local Persistence

Chương 14 : Thêm Firestore và Firestore Backend

Chương 15 : Thêm quản lý trạng thái vào ứng dụng khách Firestore

Chương 16 : Thêm BLoC vào Firestore Trang ứng dụng khách

**Phần 2 : Django 4 By Example**

Chương 1: Xây dựng một ứng dụng Blog

Chương 2: Cải thiện Blog của bạn với các tính năng nâng cao

Chương 3: Mở rộng ứng dụng blog của bạn

Chương 4: Xây dựng một trang web xã hội

Chương 5: Thực hiện xác thực xã hội

Chương 6: Chia sẻ nội dung trên trang web của bạn

Chương 7: Theo dõi hành động của người dùng

Chương 8: Xây dựng cửa hàng trực tuyến

Chương 9: Quản lý thanh toán và đơn hàng

Chương 10: Mở rộng cửa hàng của bạn

Chương 11: Thêm quốc tế hóa vào cửa hàng của bạn

Chương 12: Xây dựng nền tảng học tập điện tử

Chương 13: Tạo hệ thống quản lý nội dung

Chương 14: Hiển thị và lưu trữ nội dung

Chương 15: Xây dựng API

Chương 16: Xây dựng máy chủ trò chuyện

Chương 17: Lên sóng

**Phần 1 : Beginning Flutter**

**Chương 1 : Giới thiệu Flutter**

Flutter là khung giao diện người dùng di động của Google để xây dựng các ứng dụng hiện đại, gốc và phản ứng cho iOS và Android. Google cũng đang nghiên cứu tính năng nhúng máy tính để bàn Flutter và Flutter cho Web (Hummingbird) cũng như các thiết bị nhúng (Raspberry Pi, gia đình, ô tô, v.v.). Flutter là một dự án nguồn mở được lưu trữ trên GitHub với sự đóng góp của Google và cộng đồng. Flutter sử dụng Dart, một ngôn ngữ hướng đối tượng hiện đại biên dịch thành mã ARM gốc và mã JavaScript sẵn sàng sản xuất. Flutter sử dụng công cụ kết xuất Skia 2D hoạt động với các loại nền tảng phần cứng và phần mềm khác nhau và cũng được sử dụng bởi Google Chrome, Chrome OS, Android, Mozilla Firefox, Firefox OS và các nền tảng khác. Skia được Google tài trợ và quản lý và có sẵn cho bất kỳ ai sử dụng theo Giấy phép Phần mềm Tự do BSD. Skia sử dụng kết xuất đường dẫn dựa trên CPU và cũng hỗ trợ phần phụ trợ được tăng tốc OpenGL ES2.

Dart là ngôn ngữ mà bạn sẽ sử dụng để phát triển các ứng dụng Flutter của mình và bạn sẽ tìm hiểu thêm về ngôn ngữ này trong Chương 3, “Học kiến thức cơ bản về Dart”. Dart được biên dịch trước thời hạn (AOT) thành mã gốc, giúp ứng dụng Flutter của bạn nhanh chóng. Dart cũng được biên dịch đúng lúc (JIT), giúp hiển thị các thay đổi mã của bạn một cách nhanh chóng, chẳng hạn như thông qua tính năng tải lại trạng thái nóng của Flutter.

Flutter sử dụng Dart để tạo giao diện người dùng của bạn, loại bỏ nhu cầu sử dụng các ngôn ngữ riêng biệt như Markup hoặc nhà thiết kế hình ảnh. Flutter mang tính khai báo; nói cách khác, Flutter xây dựng giao diện người dùng để phản ánh trạng thái của ứng dụng. Khi trạng thái (dữ liệu) thay đổi, giao diện người dùng sẽ được vẽ lại và Flutter sẽ tạo một phiên bản mới của tiện ích. Trong phần “Tìm hiểu về Cây tiện ích và Cây phần tử” của chương này, bạn sẽ tìm hiểu cách cấu hình và gắn các tiện ích (kết xuất) để tạo cây tiện ích và cây phần tử, nhưng dưới phần tóm tắt, cây kết xuất (cây thứ ba) ) sử dụng RenderObject để tính toán và triển khai các giao thức vẽ và bố cục cơ bản.

XÁC ĐỊNH WIDGET VÀ CÁC PHẦN TỬ

Các widget có thể được so sánh với các khối LEGO; bằng cách thêm các khối lại với nhau, bạn tạo ra một đối tượng và bằng cách thêm các loại khối khác nhau, bạn có thể thay đổi giao diện và hành vi của đối tượng.

Widget là các khối xây dựng của ứng dụng Flutter và mỗi widget là một khai báo bất biến của giao diện người dùng. Nói cách khác, widget là cấu hình (hướng dẫn) cho các phần khác nhau của giao diện người dùng. Việc đặt các widget lại với nhau sẽ tạo ra cây widget.

Ví dụ, giả sử một kiến trúc sư vẽ bản thiết kế của một ngôi nhà; tất cả các đồ vật như tường, cửa sổ và cửa ra vào trong ngôi nhà đều là các vật dụng và tất cả chúng phối hợp với nhau để tạo ra ngôi nhà hoặc trong trường hợp này là ứng dụng. Vì các widget là các phần cấu hình của giao diện người dùng và chúng cùng nhau tạo ra cây widget.

Flutter sử dụng widget làm cấu hình để xây dựng từng phần tử, nghĩa là phần tử đó là widget được gắn (kết xuất) trên màn hình. Các phần tử được gắn trên màn hình sẽ tạo thành cây phần tử.

Dưới đây là cái nhìn ngắn gọn về hàng loạt tiện ích mà bạn có thể tùy ý sử dụng:

* Widget với các thành phần cấu trúc như danh sách, lưới, văn bản và nút
* Widget có các thành phần đầu vào như biểu mẫu, trường biểu mẫu và trình nghe bàn phím
* Các tiện ích có các yếu tố tạo kiểu như loại phông chữ, kích thước, trọng lượng, màu sắc, đường viền và bóng
* Các widget để bố trí giao diện người dùng như hàng, cột, ngăn xếp, căn giữa và đệm
* Tiện ích có các yếu tốt tương tác phản hồi khi chạm, cử chỉ, kéo và loại bỏ
* Các tiện ích có các yếu tố hoạt hình và chuyển động như hoạt hình anh hùng, vùng chứa hoạt hình, hiệu ứng chuyển hoạt hình, chuyển tiếp mờ dần, xoay, tỷ lệ, kích thước, trang trình bày và độ mờ
* Widget có các thành phần như nội dung, hình ảnh và biểu tượng
* Các widget có thể được lồng vào nhau để tạo ra giao diện người dùng cần thiết
* Các vật dụng tùy chỉnh bạn có thể tự tạo

Để xây dựng giao diện người dùng, bạn sử dụng hai loại widget chính là StatelessWidget và StatefulWidget. Tiện ích không trạng thái được sử dụng khi các giá trị (trạng thái) không thay đổi và tiện ích có trạng thái được sử dụng khi giá trị (trạng thái) thay đổi.

Mỗi tiện ích không trạng thái hoặc có trạng thái đều có một phương thức xây dựng với BuildContext xử lý vị trí của tiện ích trong cây tiện ích. Bối cảnh xây dựng các đối tượng thực chất là các đối tượng Element , một bản sao của Widget tại một vị trí trên cây.

* Vòng đời của StatelessWidget

A diagram of a construction

Description automatically generated

class JournalList extends StatelessWidget {

@override Widget build(BuildContext context) {

return Container();

}

}

* Vòng đời của StatefulWidget

A diagram of a state

Description automatically generated

class JournalEdit extends StatefulWidget {

@override

\_ JournalEditState createState() => \_ JournalEditState();

StatelessWidget Constructor build FIGURE 1.1: StatelessWidget

}

class \_JournalEditState extends State {

@override

Widget build(BuildContext context) {

return Container();

}

}

|  |  |  |
| --- | --- | --- |
| **Phương pháp** | **Miêu tả** | **Mẫu code** |
| initState() | Được gọi một lần  Khi đối tượng được chèn vào cây | @override void initState() { super.initState();  print('initState');  } |
| dispose() | Được gọi khi đối tượng này bị xóa khỏi cây vĩnh viễn. | @override void dispose() { print('dispose');  super.dispose();  } |
| didChangeDependencies() | Được gọi khi State đối tượng thay đổi | @override void didChangeDependencies() { super.didChangeDependencies(); print('didChangeDependencies');  } |
| didUpdateWidget­  (Contacts oldWidget) | Được gọi khi cấu hình widget thay đổi. | @override  void didUpdateWidget  (Contacts oldWidget) { super.didUpdateWidget(oldWidget); print('didUpdateWidget: $oldWidget'); } |
| deactivate() | Được gọi khi đối tượng này được lấy ra khỏi cây | @override void deactivate() { print('deactivate');  super.deactivate();  } |
| build (BuildContext context) | Có thể được gọi nhiều lần để xây dựng giao diện người dùng và BuildContext xử lý vị trí của tiện ích này trong cây tiện ích. | @override  Widget build(BuildContext context) { print('build');  return Container();  } |
| setState() | Thông báo cho khung rằng trạng thái của đối tượng này đã thay đổi để lên lịch gọi bản dựng cho đối tượng Trạng thái này. | setState(() {  name = \_newValue;  }); |

* Cài đặt trên Window

Yêu cầu hệ thống:

Windows 7 SP1 trở lên ( 64 – bit )

400 MB dung lượng ổ đĩa

Các công cụ sau :

+ PowerShell 5.0 trở lên

+ Git for Windows

1. Tải xuống tệp cài đặt bản mới nhất.
2. Giải nén tệp vào vị trí mong muốn.
3. Tìm trong **C:\Users\WindowUserName\flutter** và nhấp chuột vào flutter\_con-sole.bat file.
4. Cập nhật đường dẫn
   * Vào Control Panel và vào Desktop App -> User Accounts -> Change My Environment Variables.
   * Vào User Variables for WindowsUserName, chọn biến Path và nhấn nút Edit. Nếu biến path bị thiếu thì chọn tạo mới

+ Nhập đường dẫn **C:\Users\WindowsUserName\flutter\bin.**

+ Nhấn nút Ok và đóng màn hình biến path.

* + Trong User Variables for WindowsUserName, nếu biến path bị thiếu, hãy nhấp vào nút New và nhập đường dẫn cho tên biến và **C:\Users\WindowsUserName\flutter\bin** cho giá trị biến.

1. Khởi động lại Windows để cập nhật thay đổi

**Chương 2 : Tạo ứng dụng Hello World**

1. Khởi động Android Studio.
2. Nhấn chọn Start a new Flutter project.
3. Chọn ứng dụng Flutter.
4. Đặt tên project theo quy ước đặt tên thông thường.
5. Nhập tên Company domain.
6. Trên cùng màn hình bước 5 , chọn cả 2 tùy chọn cho Kotlin và Swift.
7. Nhấp vào kết thúc.

**Chương 3 : Học ngôn ngữ Dart cơ bản**

Các kiểu biến :

Numbers, Strings, Booleans, Lists, Maps, Runes

3.1 Các toán tử số học

|  |  |  |
| --- | --- | --- |
| Kí hiệu | Tên | Ví dụ |
| + | Cộng | 7 + 3 = 10 |
| - | Trừ | 7 – 3 = 4 |
| \* | Nhân | 7 \* 3 = 21 |
| / | Chia | 7 / 3 = 2.33 |

3.2 Các toán tử đẳng thức và quan hệ

|  |  |  |
| --- | --- | --- |
| Kí hiệu | Miêu tả | Ví dụ |
| == | Bằng | 7 == 3 = false |
| != | Khác | 7 != 3 = true |
| > | Lớn hơn | 7 > 3 = true |
| < | Bé hơn | 7 < 3 = false |
| >= | Lớn hơn hoặc bằng | 7 >= 3 = true  4 >= 4 = true |
| <= | Bé hơn hoặc bằng | 7 <= 3 = false  4 <= 4 = true |

3.3 Các toán tử kiểm tra kiểu

|  |  |  |
| --- | --- | --- |
| Kí hiệu | Miêu tả | Ví dụ |
| as | Như là | import 'travelpoints .dart' as travel; |
| is | Nếu đối tượng chứa đó thì nó trả về true | if (points is Places) = true |
| is! | Nếu đối tượng chứa đó thì nó trả về false ( không thường sử dụng ) | if (points is! Places) = false |

3.4 Các toán tử gán

|  |  |  |
| --- | --- | --- |
| Kí hiệu | Miêu tả | Ví dụ |
| = | Gán giá trị | 7 = 3 = 3 |
| ??= | Chỉ gán giá trị nếu biến được gán có giá trị null | Null ??= 3 = 3  7 ??= 3 = 7 |
| += | Cộng thêm vào giá trị hiện tại | 7 += 3 = 10 |
| -= | Trừ đi vào giá trị hiện tại | 7 -= 3 = 4 |
| \*= | Nhân với giá trị hiện tại | 7 \*= 3 = 21 |
| /= | Chia cho giá trị hiện tại | 7 /= 3 = 2.33 |

3.5 Các toán tử logic

|  |  |  |
| --- | --- | --- |
| Kí hiệu | Miêu tả | Ví dụ |
| ! | Là một logic “ không “. Trả về giá trị ngược lại | if ( !(7 > 3) ) = false |
| && | Là một logic “ và “’. Trả về true nếu các giá trị của biến / biểu thức đều đúng. | if ( (7 > 3) && (3 < 7) ) = true  if ( (7 > 3) && (3 > 7) ) = false |
| !! | Là logic 'hoặc'. Trả về true nếu ít nhất một giá trị của biến/biểu thức là đúng. | if ( (7 > 3) || (3 > 7) ) = true  if ( (7 < 3) || (3 > 7) ) = false |

3.6 Biểu thức điều kiện

|  |  |  |
| --- | --- | --- |
| Cấu trúc | Miêu tả | Ví dụ |
| Điều kiện ? giá trị 1 : giá trị 2 | Nếu điều kiện đúng, kết quả trả về giá trị 1 ngược lại trả về giá trị 2 | (7 > 3) ? true : false = true  (7 < 3) ? true : false = false |

3.7 Kí hiệu ( .. )

|  |  |  |
| --- | --- | --- |
| Kí hiệu | Mô tả | Ví dụ |
| .. | Cho phép bạn thực hiện một chuỗi các thao tác trên cùng một đối tượng | Matrix4.identity()  ..scale(1.0, 1.0)  ..translate(30, 30); |

Các cấu trúc

is else, while and do-while, while and break, continue, switch and case

**Chương 4: Tạo một project mẫu**

1. Nhập gói/tập tin. Việc nhập mặc định là thư viện vật liệu.dart để cho phép sử dụng
2. Sau hai câu lệnh nhập , để lại một dòng trống và nhập hàm main() được liệt kê tiếp theo. Hàm main() là điểm vào ứng dụng và gọi lớp MyApp .
3. Nhập lớp MyApp mở rộng StatelessWidget.
4. Tạo một tệp Dart mới trong thư mục trang . Nhấp chuột phải vào thư mục trang , chọn MớiTệp phi tiêu, nhập home.dart và nhấp vào nút OK để lưu.
5. Giống như bước 1, nhập gói/tệp Material.dart . Xin nhắc lại, tôi sẽ sử dụng Material Design
6. Bắt đầu nhập st và –wow- trợ giúp tự động hoàn thành sẽ mở ra. Khi bạn gõ chữ viết tắt cho một
7. Bây giờ tất cả những gì bạn cần làm là đặt tên cho lớp StatefulWidget : Home. Vì đây là một lớp học nên quy ước đặt tên là bắt đầu từ bằng một chữ cái viết hoa.
8. Thay thế tiện ích Container() bằng tiện ích Giàn giáo . Tiện ích Scaffold triển khai bố cục trực quan của Material Design cơ bản, cho phép bổ sung đơn giản AppBar, BottomAppBar, FloatActionButton, Drawer, SnackBar, BottomSheet, v.v. (Nếu đây là CupertinoApp, bạn có thể sử dụng CupertinoPageScaffold hoặc CupertinoTabScaffold.)

**Chương 5: Tìm hiểu cây Widget**

Cây tiện ích là cách bạn tạo giao diện người dùng của mình; bạn đặt các vật dụng vào nhau để xây dựng các bố cục đơn giản và phức tạp. Vì hầu hết mọi thứ trong Flutter framework đều là một widget và khi bạn bắt đầu lồng chúng vào nhau, mã có thể trở nên khó theo dõi hơn. Một cách thực hành tốt là cố gắng giữ cây widget càng nông càng tốt. Để hiểu đầy đủ tác dụng của cây sâu, bạn sẽ xem xét cây tiện ích đầy đủ rồi cấu trúc lại nó thành cây tiện ích nông, giúp mã dễ quản lý hơn. Bạn sẽ tìm hiểu ba cách để tạo cây tiện ích nông bằng cách tái cấu trúc: bằng hằng số, bằng phương thức và bằng lớp tiện ích.

* Scaffold—Implements the Material Design visual layout, allowing the use of Flutter’s Material Components widgets
* AppBar—Implements the toolbar at the top of the screen
* CircleAvatar—Usually used to show a rounded user profile photo, but you can use it for any image
* Divider—Draws a horizontal line with padding above and below If the app you are creating is using Cupertino, you can use the following widgets instead. Note that with Cupertino you can use two different scaffolds, a page scaffold or a tab scaffold.
* CupertinoPageScaffold—Implements the iOS visual layout for a page. It works with CupertinoNavigationBar to provide the use of Flutter’s Cupertino iOS-style widgets.
* CupertinoTabScaffold—Implements the iOS visual layout. This is used to navigate mul-tiple pages, with the tabs at the bottom of the screen allowing you to use Flutter’s Cupertino iOS-style widgets.
* CupertinoNavigationBar—Implements the iOS visual layout toolbar at the top of the screen.

**Chương 6: Sử dụng các widget thông dụng**

* Cách sử dụng các widget cơ bản như Scaffold, AppBar, SafeArea, Container, Text, RichText, Column, and Row, cũng như các loại nút khác nhau
* Cách lồng các widget Cột và Hàng lại với nhau để tạo các bố cục UI khác nhau
* Các cách đưa hình ảnh, biểu tượng và trang trí vào
* Cách sử dụng tiện ích trường văn bản để truy xuất, xác thực và thao tác dữ liệu
* Cách kiểm tra hướng ứng dụng

Trong chương này, bạn sẽ học cách sử dụng các vật dụng phổ biến nhất. Tôi gọi chúng là các khối xây dựng cơ bản để tạo ra các giao diện người dùng và trải nghiệm người dùng đẹp mắt. Bạn sẽ tìm hiểu cách tải hình ảnh cục bộ hoặc qua Web thông qua bộ định vị tài nguyên thống nhất (URL), sử dụng các biểu tượng thành phần Vật liệu phong phú đi kèm và áp dụng các trình trang trí để nâng cao giao diện của các tiện ích hoặc sử dụng chúng làm hướng dẫn nhập liệu cho các trường nhập .

Bạn cũng sẽ khám phá cách tận dụng tiện ích Biểu mẫu để xác thực các tiện ích mục nhập trường văn bản dưới dạng một nhóm chứ không chỉ riêng lẻ. Ngoài ra, để tính đến sự đa dạng của kích thước thiết bị, bạn sẽ thấy cách sử dụng tiện ích MediaQuery hoặc OrientationBuilder là cách tuyệt vời để phát hiện hướng—vì việc sử dụng hướng thiết bị và tiện ích bố cục phù hợp dựa trên dọc hoặc ngang là cực kỳ quan trọng. Ví dụ: nếu thiết bị ở chế độ dọc, bạn có thể hiển thị một hàng gồm ba hình ảnh, nhưng khi thiết bị được chuyển sang chế độ ngang, bạn có thể hiển thị một hàng gồm năm hình ảnh vì chiều rộng là diện tích lớn hơn ở chế độ dọc.

**Chương 7 : Thêm hiệu ứng và ứng dụng**

* Cách sử dụng AnimatedContainer để thay đổi dần giá trị theo thời gian
* Cách dùng AnimatedCrossFade để làm mờ chéo giữa 2 widget con
* Cách sử dụng AnimatedOpacity để hiển thị hoặc ẩn khả năng hiển thị của widget bằng cách làm mờ hoạt ảnh theo thời gian
* Cách sử dụng AnimationController để tạo ảnh động tùy chỉnh
* Cách sử dụng AnimationController để điều khiển hoạt ảnh so le

Trong chương này, bạn sẽ tìm hiểu cách thêm hoạt ảnh vào ứng dụng để truyền tải hành động, điều này có thể cải thiện trải nghiệm người dùng (UX) nếu được sử dụng phù hợp. Quá nhiều hoạt ảnh mà không truyền tải hành động thích hợp có thể khiến trải nghiệm người dùng trở nên tồi tệ hơn. Flutter có hai loại hoạt ảnh: dựa trên vật lý và Tween. Chương này sẽ tập trung vào hoạt hình Tween.

Hoạt hình dựa trên vật lý được sử dụng để mô phỏng hành vi trong thế giới thực. Ví dụ, khi một vật được thả rơi và chạm đất, nó sẽ nảy lên và tiếp tục chuyển động về phía trước, nhưng với mỗi lần nảy, nó tiếp tục chậm lại với những lực bật lại nhỏ hơn và cuối cùng dừng lại. Khi vật tiến gần đến mặt đất hơn sau mỗi lần nảy, vận tốc tăng lên, nhưng độ cao của cú nảy giảm đi.

Tween là viết tắt của “in-between”, nghĩa là hoạt ảnh có điểm bắt đầu và kết thúc, dòng thời gian và đường cong chỉ định thời gian và tốc độ chuyển tiếp. Cái hay là khung này tự động tính toán quá trình chuyển đổi từ điểm đầu đến điểm cuối.

**Chương 8 : Tạo điều hướng ứng dụng**

* Cách sử dụng tiện ích Điều hướng để điều hướng giữa các trang
* Cách hoạt ảnh hero cho phép chuyển đổi tiện ích vào đúng vị trí từ một trang Tạo điều hướng của ứng dụng đến người khác
* Cách hiển thị danh sách ngang các BottomNavigationBarItems chứa biểu tượng và tiêu đề ở cuối trang
* Cách nâng cao giao diện của thanh điều hướng phía dưới bằng BottomAppBar widget và cho phép kích hoạt một notch
* Cách hiển thị dòng tab ngang bằng TabBar
* Cách sử dụng TabBarView kết hợp với TabBar để hiển thị trang của tab đã chọn
* Cách Drawer cho phép người dùng trượt một bảng từ trái hoặc phải
* Cách sử dụng hàm tạo ListView để xây dựng danh sách ngắn các mục một cách nhanh chóng
* Cách sử dụng hàm tạo ListView với tiện ích Drawer để hiển thị danh sách menu

Trong chương này, bạn sẽ biết rằng điều hướng là thành phần chính trong ứng dụng di động. Điều hướng tốt tạo ra trải nghiệm người dùng (UX) tuyệt vời bằng cách giúp bạn dễ dàng truy cập thông tin. Ví dụ: hãy tưởng tượng bạn đang tạo một mục nhật ký và khi cố gắng chọn một thẻ thì nó không có sẵn, vì vậy bạn cần tạo một mục mới. Bạn có đóng mục này và đi tới Cài đặt

Hoạt ảnh trong khi điều hướng đến các trang khác nhau cũng rất quan trọng nếu nó giúp truyền tải một hành động, thay vì chỉ đơn giản là gây xao lãng. Điều đó có nghĩa là gì? Chỉ vì bạn có thể hiển thị những hình ảnh động ưa thích không có nghĩa là bạn nên làm như vậy. Sử dụng hình ảnh động để nâng cao UX, không làm người dùng thất vọng.

**Chương 9 : Tạo danh sách và hiệu ứng cuộn**

Trong chương này, bạn sẽ học cách tạo danh sách cuộn giúp người dùng xem và chọn thông tin. Bạn sẽ bắt đầu với tiện ích Thẻ trong chương này vì nó thường được sử dụng cùng với các tiện ích có khả năng liệt kê để nâng cao giao diện người dùng (UI) và dữ liệu nhóm. Trong chương trước, bạn đã xem xét cách sử dụng hàm tạo cơ bản cho ListView và trong chương này, bạn sẽ sử dụng ListView.builder để tùy chỉnh dữ liệu. Tiện ích GridView là một tiện ích tuyệt vời hiển thị danh sách dữ liệu theo số ô (nhóm dữ liệu) cố định trên trục chéo.

Tiện ích Ngăn xếp thường được sử dụng để chồng chéo, định vị và căn chỉnh các tiện ích nhằm tạo giao diện tùy chỉnh. Một ví dụ điển hình là giỏ hàng có số lượng mặt hàng cần mua ở phía trên bên phải.

Tiện ích CustomScrollView cho phép bạn tạo các hiệu ứng cuộn tùy chỉnh bằng cách sử dụng danh sách các tiện ích con cúi.

Ví dụ: các phần nhỏ rất hữu ích nếu bạn có một mục nhật ký có hình ảnh ở đầu trang và mô tả nhật ký bên dưới. Khi người dùng vuốt để đọc thêm, thao tác cuộn mô tả sẽ nhanh hơn cuộn hình ảnh, tạo hiệu ứng thị sai.

Tiện ích Thẻ là một phần của Material Design và có các góc và bóng được bo tròn tối thiểu. Để nhóm và bố trí dữ liệu, Thẻ là một tiện ích hoàn hảo giúp cải thiện giao diện của giao diện người dùng. Tiện ích Thẻ có thể tùy chỉnh với các thuộc tính như độ cao, hình dạng, màu sắc, lề và các thuộc tính khác. Thuộc tính độ cao có giá trị gấp đôi và số càng cao thì bóng đổ càng lớn. Bạn đã học trong Chương 3, “Tìm hiểu kiến thức cơ bản về phi tiêu”, rằng double là một số yêu cầu độ chính xác của dấu thập phân, chẳng hạn như 8,50. Để tùy chỉnh hình dạng và đường viền của tiện ích Thẻ , bạn sửa đổi thuộc tính hình dạng . Một số thuộc tính hình dạng là StadiumBorder, Underline-InputBorder, OutlineInputBorder và các thuộc tính khác.

* Sử dụng GridView.count

Đối với children, bạn sử dụng List.generate để tạo dữ liệu mẫu, danh sách các giá trị. Trong đối số con , tôi đã thêm một câu lệnh in để cho thấy rằng toàn bộ danh sách các giá trị được tạo cùng lúc, không chỉ các hàng hiển thị như GridView.builder . Lưu ý đối với mã mẫu sau, 7.000 bản ghi được tạo để cho thấy GridView.count không hiển thị bất kỳ dữ liệu nào cho đến khi tất cả các bản ghi được xử lý trước.

* Sử dụng GridView.extent

GridView.extent yêu cầu bạn đặt đối số maxCrossAxisExtent và đối số trẻ em . Đối số maxCrossAxisExtent đặt kích thước tối đa của mỗi ô cho trục. Ví dụ, ở chế độ dọc, nó có thể vừa với hai đến ba ô, nhưng khi xoay sang ngang, nó có thể vừa với năm đến sáu ô tùy thuộc vào kích thước màn hình (Hình 9.5). Đối số ScrollDirection đặt hướng trục chính cho lưới để cuộn, Axis.vertical hoặc Axis.horizontal và mặc định là dọc.

* Sử dụng GridView.builder

GridView.builder được sử dụng khi bạn có một tập hợp dữ liệu lớn vì trình tạo chỉ được gọi cho các ô hiển thị, điều này rất tốt cho hiệu suất . Việc sử dụng hàm tạo GridView.builder sẽ tạo ra một danh sách các ô hiển thị một cách lười biếng và khi người dùng cuộn đến các ô hiển thị tiếp theo, chúng sẽ được tạo một cách lười biếng nếu cần.

**Chương 10 : Xây dựng Layouts**

Mục tiêu của chương này là tạo một trang nhật ký hiển thị chi tiết từ trên xuống dưới. Trang tạp chí hiển thị hình ảnh đầu trang, tiêu đề, chi tiết nhật ký, thời tiết, địa chỉ (vị trí tạp chí), thẻ và hình ảnh chân trang. Các phần thời tiết, thẻ và hình ảnh chân trang được xây dựng bằng cách lồng các widget để xây dựng bố cục tùy chỉnh

1. Vì thiết bị di động có nhiều kích cỡ khác nhau nên bố cục bắt đầu bằng cách thêm một ChildScrollView để tự động xử lý việc cuộn các phần màn hình bị các thiết bị nhỏ hơn cắt bỏ.
2. Tiếp theo, tiện ích Cột được sử dụng để căn chỉnh các tiện ích theo chiều dọc từ trên xuống dưới màn hình.
3. Đối với hình ảnh món quà được gói, tiện ích Hình ảnh được thêm làm con đầu tiên của Cột đầu tiên, cho phép hình ảnh lấp đầy toàn bộ chiều rộng của thiết bị.
4. Cột con đầu tiên là một tiện ích SafeArea để xử lý phần notch của thiết bị cho tạp chí nội dung đầu vào.
5. Thêm vào cột con SafeArea một Cột thứ hai với các tiện ích con bao gồm Văn bản tiện ích dành cho tiêu đề mục nhật ký và tiện ích Văn bản để biết chi tiết mục nhập tạp chí
6. Tiếp tục thêm vào cột con thứ hai một tiện ích Hàng sẽ chứa biểu tượng thời tiết, nhiệt độ thời tiết và địa chỉ vị trí mục nhật ký. Trong phần “Bố cục phần thời tiết” của chương này, bạn sẽ học cách thêm các tiện ích để tạo bố cục chi tiết.
7. Tiếp tục bằng cách thêm vào các cột con thứ hai một tiện ích Gói hiển thị các tiện ích Chip . Bạn sẽ học cách thêm tiện ích bố cục trong phần “Bố cục thẻ”.
8. Cuối cùng, bạn sẽ thêm tiện ích Cột một Hàng vào tiện ích Cột thứ hai để hiển thị hình ảnh và biểu tượng, đồng thời bạn sẽ tìm hiểu cách thêm tiện ích bố cục trong phần “Bố cục Hình ảnh Chân trang” của chương này.

**Chương 11 : Áp dụng tính tương tác**

* Cách sử dụng GestureDetector, tính năng nhận dạng các cử chỉ như chạm, chạm hai lần, nhấn và giữ, xoay, kéo dọc, kéo ngang và chia tỷ lệ.
* Cách sử dụng tiện ích Draggable được kéo vào DragTarget.
* Cách sử dụng tiện ích Draggable được kéo vào DragTarget.
* Cách sử dụng tiện ích InkWell và InkResponse . Bạn sẽ biết rằng InkWell là một khu vực hình chữ nhật phản hồi khi chạm và kẹp các tia bắn vào trong khu vực của nó. Bạn sẽ biết rằng InkResponse phản hồi khi chạm và các phần bắn tung tóe sẽ mở rộng ra ngoài khu vực của nó.
* Cách sử dụng tiện ích Dismissible được loại bỏ bằng cách kéo.

Trong chương này, bạn sẽ tìm hiểu cách thêm tính tương tác vào ứng dụng bằng cách sử dụng cử chỉ. Trong một ứng dụng di động, cử chỉ là trung tâm để lắng nghe sự tương tác của người dùng. Việc sử dụng cử chỉ có thể xác định một ứng dụng có UX tuyệt vời. Việc lạm dụng cử chỉ khi chúng không tăng thêm giá trị hoặc truyền tải một hành động sẽ tạo ra trải nghiệm người dùng kém. Bạn sẽ xem xét kỹ hơn cách tìm sự cân bằng bằng cách sử dụng cử chỉ chính xác cho nhiệm vụ trước mắt.

* THIẾT LẬP CỬ CHỈNH: CƠ BẢN

Tiện ích GestureDetector phát hiện các cử chỉ như chạm, chạm hai lần, nhấn lâu, xoay, kéo dọc, kéo ngang và chia tỷ lệ. Nó có một thuộc tính con tùy chọn và nếu một tiện ích con được chỉ định thì các cử chỉ chỉ áp dụng cho tiện ích con đó. Nếu tiện ích con bị bỏ qua thì GestureDetector sẽ lấp đầy toàn bộ tiện ích gốc. Nếu bạn cần thực hiện thao tác kéo dọc và kéo ngang cùng lúc, hãy sử dụng cử chỉ xoay. Nếu bạn cần bắt một thao tác kéo một trục, hãy sử dụng cử chỉ kéo dọc hoặc kéo ngang.

Sau đây là các cử chỉ GestureDetector mà bạn có thể lắng nghe và thực hiện hành động thích hợp:

* Tap

onTapDown

onTapUp

onTap

onTapCancel

* Double tap

onDoubleTap

* Long press

onLongPress

* Pan

onPanStart

onPanUpdate

onPanEnd

* Vertical drag

onVerticalDragStart

onVerticalDragUpdate

onVerticalDragEnd

* Horizontal drag

onHorizontalDragStart

onHorizontalDragUpdate

onHorizontalDragEnd

* Scale

onScaleStart

onScaleUpdate

onScaleEnd

**Chương 12 : Viết Platform-Native Code**

Các kênh nền tảng cung cấp cho bạn khả năng sử dụng các tính năng gốc như truy cập thông tin thiết bị, vị trí GPS, thông báo cục bộ, hệ thống tệp cục bộ, chia sẻ và nhiều tính năng khác. Trong phần “Gói bên ngoài” của Chương 2, “Tạo ứng dụng Hello World”, bạn đã học cách sử dụng các gói của bên thứ ba để thêm chức năng cho ứng dụng của mình. Trong chương này, thay vì dựa vào các gói của bên thứ ba, bạn sẽ tìm hiểu cách thêm chức năng tùy chỉnh vào ứng dụng của mình bằng cách sử dụng các kênh nền tảng và tự viết mã API. Bạn sẽ xây dựng một ứng dụng yêu cầu nền tảng iOS và Android trả về thông tin thiết bị.

Khi cần truy cập các API dành riêng cho nền tảng cho iOS và Android, bạn sử dụng các kênh nền tảng để gửi và nhận tin nhắn. Ứng dụng Flutter là máy khách và mã gốc nền tảng cho iOS và Android là máy chủ. Nếu cần, cũng có thể có mã gốc nền tảng để hoạt động như một ứng dụng khách để gọi các phương thức được viết bằng mã phi tiêu của ứng dụng Flutter.

Các tin nhắn giữa máy khách và máy chủ không đồng bộ, đảm bảo rằng giao diện người dùng vẫn phản hồi và không bị chặn. Trong Chương 3, “Tìm hiểu kiến thức cơ bản về Dart”, bạn đã biết rằng các hàm không đồng bộ thực hiện các thao tác tốn thời gian mà không cần đợi các thao tác đó hoàn tất.

Đối với phía máy khách (ứng dụng Flutter), bạn sử dụng MethodChannel từ một phương thức không đồng bộ để gửi tin nhắn chứa lệnh gọi phương thức sẽ được phía máy chủ thực thi (iOS và Android). Sau khi máy chủ gửi phản hồi lại, bạn có thể cập nhật giao diện người dùng để hiển thị thông tin nhận được.

Đối với phía máy chủ, bạn sử dụng FlutterMethodChannel trên iOS và MethodChannel trên Android.

Khi máy chủ nhận được lệnh gọi máy khách, mã nền tảng gốc sẽ thực thi phương thức được gọi và sau đó gửi lại kết quả

A diagram of a cell phone

Description automatically generated

**Chương 13 : Lưu dữ liệu với Local Persistence**

* Cách duy trì lưu và đọc dữ liệu cục bộ
* Cách cấu trúc dữ liệu bằng định dạng tệp JSON
* Cách tạo các lớp mô hình để xử lý tuần tự hóa JSON
* Cách truy cập các vị trí hệ thống tệp iOS và Android cục bộ bằng gói nhà cung cấp đường dẫn
* Cách định dạng ngày bằng gói quốc tế hóa
* Cách sử dụng lớp Future với showDatePicker để trình bày lịch chọn ngày
* Cách sử dụng lớp Future để lưu, đọc và phân tích tệp JSON
* Cách sử dụng hàm tạo ListView.separated để phân chia các bản ghi bằng Bộ chia
* Cách sử dụng List().sort để sắp xếp các mục nhật ký theo ngày
* Cách sử dụng textInputAction để tùy chỉnh thao tác trên bàn phím
* Cách sử dụng FocusNode và FocusScope bằng bàn phím onSubmit để di chuyển con trỏ đến TextField của mục tiếp theo
* Cách truyền và nhận dữ liệu trong một lớp bằng Bộ điều hướng

Trong chương này, bạn sẽ tìm hiểu cách lưu giữ dữ liệu—tức là lưu dữ liệu trên thư mục lưu trữ cục bộ của thiết bị—trong các lần khởi chạy ứng dụng bằng cách sử dụng định dạng tệp JSON và lưu tệp vào hệ thống tệp iOS và Android cục bộ. Ký hiệu đối tượng JavaScript (JSON) là một tiêu chuẩn mở phổ biến và định dạng dữ liệu tệp độc lập với ngôn ngữ với lợi ích là văn bản có thể đọc được. Việc lưu giữ dữ liệu là một quá trình gồm hai bước; trước tiên, bạn sử dụng lớp Tệp để lưu và đọc dữ liệu, sau đó, bạn phân tích cú pháp dữ liệu từ và sang định dạng JSON. Bạn sẽ tạo một lớp để xử lý việc lưu và đọc tệp dữ liệu sử dụng lớp Tệp . Bạn cũng sẽ tạo một lớp để phân tích danh sách dữ liệu đầy đủ bằng cách sử dụng json.encode và json.decode cùng một lớp để trích xuất từng bản ghi. Và bạn sẽ tạo một lớp khác để xử lý việc chuyển một hành động và một mục nhật ký riêng lẻ giữa các trang.

Bạn sẽ xây dựng một ứng dụng nhật ký để lưu và đọc dữ liệu JSON vào NSDocumentDirectory của iOS cục bộ và hệ thống tệp Android AppData . Ứng dụng sử dụng ListView để hiển thị danh sách các mục nhật ký được sắp xếp theo ngày và bạn sẽ tạo màn hình nhập dữ liệu để nhập ngày, tâm trạng và ghi chú.

**Chương 14 : Thêm Firestore và Firestore Backend**

Firebase bao gồm một nền tảng với vô số sản phẩm có thể chia sẻ và hoạt động cùng nhau. Firebase xử lý toàn bộ cơ sở hạ tầng máy chủ phụ trợ kết nối các ứng dụng iOS, Android và web.

Việc thêm bảo mật vào ứng dụng là cực kỳ quan trọng trong việc giữ thông tin riêng tư và an toàn. Xác thực Firebase cung cấp các dịch vụ phụ trợ tích hợp có thể truy cập được từ SDK của khách hàng để hỗ trợ xác thực đầy đủ

Sau đây là danh sách các nhà cung cấp dịch vụ đăng nhập xác thực hiện có:

Email/Password

Phone

Google

Play Games (Google)

Game Center (Apple)

Facebook

Twitter

GitHub

Anonymous

**Chương 15 : Thêm quản lý trạng thái vào ứng dụng khách Firestore**

Bạn sẽ tìm hiểu cách sử dụng mẫu Thành phần logic nghiệp vụ (BLoC) để tạo các lớp BLoC, chẳng hạn như quản lý quyền truy cập vào các lớp dịch vụ cơ sở dữ liệu Cloud Firestore và Xác thực Firebase.

Bạn sẽ tìm hiểu cách tạo một lớp dịch vụ để quản lý API xác thực Firebase bằng cách triển khai một lớp trừu tượng quản lý thông tin đăng nhập của người dùng. Bạn sẽ tạo một lớp dịch vụ riêng để xử lý API cơ sở dữ liệu Cloud Firestore. Bạn sẽ tìm hiểu cách tạo một lớp mô hình Nhật ký để xử lý việc ánh xạ Cloud Firestore QuerySnapshot tới các bản ghi riêng lẻ. Bạn sẽ tìm hiểu cách tạo một lớp biểu tượng tâm trạng để quản lý danh sách các biểu tượng tâm trạng, mô tả và vị trí xoay biểu tượng theo tâm trạng đã chọn. Bạn sẽ học cách tạo một lớp định dạng ngày tháng bằng gói intl .

A screenshot of a diagram

Description automatically generated

**Chương 16 : Thêm BLoC vào Firestore Trang ứng dụng khách**

Bạn sẽ tìm hiểu cách áp dụng các lớp BLoC, dịch vụ, nhà cung cấp, mô hình và tiện ích cho các trang tiện ích giao diện người dùng. Lợi ích của việc sử dụng mẫu BLoC cho phép tách các tiện ích giao diện người dùng khỏi logic nghiệp vụ. Bạn sẽ tìm hiểu cách sử dụng tính năng nội xạ phụ thuộc để đưa các lớp dịch vụ vào các lớp BLoC.

Bằng cách sử dụng tính năng chèn phụ thuộc, BLoC vẫn không phụ thuộc vào nền tảng.

Bạn cũng sẽ tìm hiểu cách áp dụng quản lý trạng thái trên toàn ứng dụng bằng cách triển khai lớp AuthenticationBlocProvider cho trang chính. Bạn sẽ tìm hiểu cách chuyển trạng thái giữa các trang và cây tiện ích bằng cách triển khai các lớp HomeBlocProvider và JournalEditBlocProvider .

Bạn sẽ tìm hiểu cách tạo trang Đăng nhập triển khai lớp Đăng nhậpBloc để xác thực email, mật khẩu và thông tin xác thực của người dùng. Bạn sẽ sửa đổi trang chủ và tìm hiểu cách triển khai lớp HomeBloc để xử lý danh sách các mục nhật ký cũng như thêm và xóa các mục riêng lẻ. Bạn sẽ tìm hiểu cách tạo trang chỉnh sửa tạp chí triển khai lớp JournalEditBloc để thêm, sửa đổi và lưu các mục nhập hiện có.

**Phần 1 : Django 4 By Example**

**Chương 1 : Xây dựng một ứng dụng Blog**

Trong cuốn sách này, bạn sẽ học cách xây dựng các dự án Django chuyên nghiệp. Chương này sẽ hướng dẫn bạn cách xây dựng một ứng dụng Django bằng cách sử dụng các thành phần chính của framework. Nếu bạn chưa cài đặt Django, bạn sẽ khám phá cách thực hiện điều đó trong phần đầu tiên của chương này.

Chương này sẽ phục vụ như một hướng dẫn để xây dựng một ứng dụng Django hoàn chỉnh và sẽ cung cấp cái nhìn sâu sắc về cách hoạt động của khung này. Đừng lo lắng nếu bạn không hiểu tất cả các khía cạnh của framework. Các thành phần khung khác nhau sẽ được khám phá chi tiết trong suốt cuốn sách này.

Chương này sẽ bao gồm các chủ đề sau:

* Cài đặt Python
* Tạo môi trường ảo Python
* Cài đặt Django
* Tạo và định cấu hình dự án Django
* Xây dựng ứng dụng Django
* Thiết kế mô hình dữ liệu
* Tạo và áp dụng di chuyển mô hình
* Tạo một trang quản trị cho mô hình của bạn
* Làm việc với Bộ truy vấn và trình quản lý mô hình
* Xây dựng chế độ xem, mẫu và URL
* Tìm hiểu chu trình yêu cầu/phản hồi của Django

**Chương 2 : Cải thiện Blog của bạn với các tính năng nâng cao**

Trong chương này, chúng ta sẽ mở rộng các chức năng của ứng dụng blog với các tính năng có thể tìm thấy trên nhiều nền tảng blog hiện nay. Trong chương này, bạn sẽ học các chủ đề sau:

* Sử dụng URL chuẩn cho mô hình

Django cho phép bạn triển khai phương thức get\_absolute\_url() trong mô hình của mình để trả về URL chuẩn cho đối tượng. Chúng tôi sẽ sử dụng URL post\_detail được xác định trong mẫu URL của ứng dụng để xây dựng URL chuẩn cho đối tượng Post . Django cung cấp các chức năng phân giải URL khác nhau cho phép bạn tạo URL động bằng cách sử dụng tên của chúng và mọi tham số bắt buộc. Chúng ta sẽ sử dụng hàm tiện ích Reverse() của module django.urls .

* Tạo URL thân thiện với SEO cho bài viết

URL chuẩn cho chế độ xem chi tiết bài đăng trên blog hiện có dạng /blog/1/. Chúng tôi sẽ thay đổi mẫu URL để tạo URL thân thiện với SEO cho bài đăng. Chúng tôi sẽ sử dụng cả giá trị ngày xuất bản và slug để tạo URL cho các bài đăng đơn lẻ. Bằng cách kết hợp ngày, chúng tôi sẽ tạo một URL chi tiết bài đăng có dạng /blog/2022/1/1/who-wasdjango-reinhardt/. Chúng tôi sẽ cung cấp cho các công cụ tìm kiếm các URL thân thiện để lập chỉ mục, chứa cả tiêu đề và ngày của bài đăng.

Bằng cách sử dụng Unique\_for\_date, trường slug hiện được yêu cầu phải là duy nhất cho ngày được lưu trữ trong trường xuất bản . Lưu ý rằng trường xuất bản là một phiên bản của DateTimeField, nhưng việc kiểm tra các giá trị duy nhất sẽ chỉ được thực hiện theo ngày (không phải thời gian). Django sẽ ngăn việc lưu một bài đăng mới có cùng slug với bài đăng hiện có trong một ngày xuất bản nhất định. Hiện tại, chúng tôi đã đảm bảo rằng các phần mở rộng là duy nhất cho ngày xuất bản, vì vậy giờ đây chúng tôi có thể truy xuất các bài đăng đơn lẻ theo trường xuất bản và phần mở rộng .

* Sửa đổi mẫu URL

Hãy sửa đổi mẫu URL để sử dụng ngày xuất bản và slug cho URL chi tiết bài đăng.

Mẫu URL cho chế độ xem post\_detail có các đối số sau:

* năm: Yêu cầu một số nguyên
* tháng: Yêu cầu số nguyên
* ngày: Yêu cầu số nguyên
* post: Yêu cầu một slug (một chuỗi chỉ chứa các chữ cái, số, dấu gạch dưới hoặc dấu gạch nối)

Bộ chuyển đổi đường dẫn int được sử dụng cho các tham số năm, tháng và ngày , trong khi bộ chuyển đổi đường dẫn slug được sử dụng cho tham số bài đăng . Bạn đã tìm hiểu về bộ chuyển đổi đường dẫn ở chương trước. Bạn có thể xem tất cả các trình chuyển đổi đường dẫn do Django cung cấp tại https://docs.djangoproject.com/en/4.1/topics/ http/urls/#path-converters.

* Sửa đổi chế độ xem

Bây giờ chúng ta phải thay đổi các tham số của chế độ xem post\_detail để khớp với các tham số URL mới và sử dụng chúng để truy xuất đối tượng Post tương ứng

Chỉnh sửa tệp view.py và chỉnh sửa chế độ xem post\_detail như thế này:

A screenshot of a computer screen

Description automatically generated

Sửa đổi chế độ xem post\_detail để lấy các đối số năm, tháng, ngày và bài đăng cũng như truy xuất một bài đăng đã xuất bản với slug và ngày xuất bản nhất định. Bằng cách thêm Unique\_for\_date='publish' vào trường slug của mô hình Bài đăng trước đó, chúng tôi đã đảm bảo rằng sẽ chỉ có một bài đăng có slug cho một ngày nhất định. Vì vậy, bạn có thể truy xuất các bài đăng đơn lẻ bằng cách sử dụng ngày tháng và slug.

* Sửa đổi URL chuẩn cho bài viết

Chúng tôi cũng phải sửa đổi các tham số của URL chuẩn cho các bài đăng trên blog để phù hợp với các tham số URL mới.

Hãy xem URL—nó sẽ có dạng /blog/2022/1/1/who-was-django-reinhardt/. Bạn đã thiết kế các URL thân thiện với SEO cho các bài đăng trên blog.

* Thêm phân trang

Khi bạn bắt đầu thêm nội dung vào blog của mình, bạn có thể dễ dàng lưu trữ hàng chục hoặc hàng trăm bài đăng trong cơ sở dữ liệu của mình. Thay vì hiển thị tất cả các bài đăng trên một trang, bạn có thể muốn chia danh sách các bài đăng trên nhiều trang và bao gồm các liên kết điều hướng đến các trang khác nhau. Chức năng này được gọi là phân trang và bạn có thể tìm thấy nó trong hầu hết mọi ứng dụng web hiển thị danh sách dài các mục.

* Tạo mẫu phân trang

Chúng ta cần tạo một điều hướng trang để người dùng duyệt qua các trang khác nhau. Chúng tôi sẽ tạo một mẫu để hiển thị các liên kết phân trang. Chúng tôi sẽ làm cho nó chung chung để có thể sử dụng lại mẫu cho bất kỳ phân trang đối tượng nào trên trang web của chúng tôi.

* Xử lý lỗi phân trang

Bây giờ tính năng phân trang đang hoạt động, chúng ta có thể thêm xử lý ngoại lệ cho các lỗi phân trang trong dạng xem. Tuyệt vời! Các liên kết phân trang đang hoạt động như mong đợi. Tham số trang được chế độ xem sử dụng để truy xuất trang đã cho có thể bị sử dụng với các giá trị sai, chẳng hạn như số trang không tồn tại hoặc giá trị chuỗi không thể được sử dụng làm số trang.

**Chương 3 : Mở rộng ứng dụng blog của bạn**

Trong chương này, bạn sẽ mở rộng ứng dụng blog của mình với các tính năng phổ biến khác được sử dụng trên nền tảng blog, chẳng hạn như gắn thẻ, đề xuất các bài đăng tương tự, cung cấp nguồn cấp RSS cho người đọc và cho phép họ tìm kiếm bài đăng. Bạn sẽ tìm hiểu về các thành phần và chức năng mới với Django bằng cách xây dựng các chức năng này.

* Thêm chức năng gắn thẻ

Một chức năng rất phổ biến trong blog là phân loại bài viết bằng cách sử dụng thẻ. Thẻ cho phép bạn phân loại nội dung theo cách không phân cấp, sử dụng các từ khóa đơn giản. Thẻ chỉ đơn giản là một nhãn hoặc từ khóa có thể được gán cho bài viết. Chúng tôi sẽ tạo một hệ thống gắn thẻ bằng cách tích hợp ứng dụng gắn thẻ Django của bên thứ ba vào dự án.

* Truy xuất bài viết bằng sự tương tự

Bây giờ chúng tôi đã triển khai tính năng gắn thẻ cho các bài đăng trên blog, bạn có thể thực hiện nhiều điều thú vị với thẻ. Thẻ cho phép bạn phân loại bài đăng theo cách không phân cấp. Các bài viết về chủ đề tương tự sẽ có một số thẻ chung. Chúng tôi sẽ xây dựng chức năng hiển thị các bài đăng tương tự theo số lượng thẻ họ chia sẻ. Bằng cách này, khi người dùng đọc một bài đăng, chúng tôi có thể gợi ý cho họ đọc các bài đăng khác có liên quan.

Để truy xuất các bài đăng tương tự cho một bài đăng cụ thể, bạn cần thực hiện các bước sau:

1. Truy xuất tất cả các thẻ cho bài viết hiện tại
2. Nhận tất cả các bài đăng được gắn thẻ bằng bất kỳ thẻ nào trong số đó
3. Loại trừ bài đăng hiện tại khỏi danh sách đó để tránh đề xuất bài đăng tương tự
4. Sắp xếp kết quả theo số tags chia sẻ với bài viết hiện tại
5. Trong trường hợp có hai bài đăng trở lên có cùng số lượng thẻ, hãy đề xuất bài đăng gần đây nhất
6. Giới hạn truy vấn ở số lượng bài đăng bạn muốn giới thiệu

Các bước này được dịch thành một Bộ truy vấn phức tạp mà bạn sẽ đưa vào chế độ xem post\_detail của mình .

* Tạo thẻ và bộ lọc mẫu tùy chỉnh

Django cung cấp nhiều loại thẻ mẫu tích hợp sẵn, chẳng hạn như {% if %} hoặc {% block %}. Bạn đã sử dụng các thẻ mẫu khác nhau trong Chương 1, Xây dựng ứng dụng Blog và Chương 2, Cải thiện Blog của bạn bằng các tính năng nâng cao. Bạn có thể tìm thấy tài liệu tham khảo đầy đủ về các thẻ mẫu và bộ lọc tích hợp tại https:// docs.djangoproject.com/en/4.1/ref/templates/buildins/.

Django cũng cho phép bạn tạo thẻ mẫu của riêng mình để thực hiện các hành động tùy chỉnh. Thẻ mẫu tùy chỉnh rất hữu ích khi bạn cần thêm một chức năng vào các mẫu không có trong bộ thẻ mẫu Django cốt lõi. Đây có thể là thẻ để thực thi Bộ truy vấn hoặc bất kỳ quá trình xử lý phía máy chủ nào mà bạn muốn sử dụng lại trên các mẫu. Ví dụ: chúng ta có thể tạo thẻ mẫu để hiển thị danh sách các bài đăng mới nhất được xuất bản trên blog. Chúng tôi có thể đưa danh sách này vào thanh bên để nó luôn hiển thị, bất kể chế độ xem xử lý yêu cầu như thế nào.

* Triển khai thẻ mẫu tùy chỉnh

Django cung cấp các chức năng trợ giúp sau cho phép bạn dễ dàng tạo thẻ mẫu:

* simple\_tag: Xử lý dữ liệu đã cho và trả về một chuỗi
* include\_tag: Xử lý dữ liệu đã cho và trả về mẫu được hiển thị

Thẻ mẫu phải nằm trong ứng dụng Django.

* Thêm tìm kiếm toàn văn vào blog

Tìm kiếm dữ liệu trong cơ sở dữ liệu với thông tin đầu vào của người dùng là một nhiệm vụ phổ biến đối với các ứng dụng web. Django ORM cho phép bạn thực hiện các thao tác khớp đơn giản bằng cách sử dụng bộ lọc chứa (hoặc phiên bản không phân biệt chữ hoa chữ thường, icontains). Bạn có thể sử dụng truy vấn sau để tìm các bài đăng có chứa khung từ trong nội dung của chúng:

**A close-up of words

Description automatically generated**

* Cài đặt PostgreSQL

Hỗ trợ SQLite cho tìm kiếm toàn văn bản bị hạn chế và Django không hỗ trợ nó ngay lập tức. Tuy nhiên, PostgreSQL phù hợp hơn nhiều cho tìm kiếm toàn văn bản và chúng ta có thể sử dụng mô-đun django.contrib.postgres để sử dụng khả năng tìm kiếm toàn văn bản của PostgreSQL. Chúng tôi sẽ di chuyển dữ liệu của mình từ SQLite sang PostgreSQL để hưởng lợi từ các tính năng tìm kiếm toàn văn bản của nó.

**Chương 4 : Xây dựng một trang web xã hội**

Trong chương này, bạn sẽ tìm hiểu cách phát triển các chức năng tài khoản người dùng để tạo một trang web xã hội, bao gồm đăng ký người dùng, quản lý mật khẩu, chỉnh sửa hồ sơ và xác thực. Chúng tôi sẽ triển khai các tính năng xã hội vào trang web này trong một số chương tiếp theo để cho phép người dùng chia sẻ hình ảnh và tương tác với nhau. Người dùng sẽ có thể đánh dấu bất kỳ hình ảnh nào trên internet và chia sẻ nó với những người dùng khác. Họ cũng sẽ có thể xem hoạt động trên nền tảng từ những người dùng mà họ theo dõi và thích/ không thích những hình ảnh được họ chia sẻ.

* Tạo dự án trang web xã hội

Chúng tôi sẽ tạo một ứng dụng xã hội cho phép người dùng chia sẻ hình ảnh mà họ tìm thấy trên internet.

Chúng ta sẽ cần xây dựng các yếu tố sau cho dự án này:

* Hệ thống xác thực để người dùng đăng ký, đăng nhập, chỉnh sửa hồ sơ và thay đổi hoặc đặt lại mật khẩu của họ.
* Hệ thống theo dõi cho phép người dùng theo dõi nhau trên website.
* Chức năng hiển thị hình ảnh được chia sẻ và hệ thống để người dùng chia sẻ hình ảnh từ bất kỳ trang web nào.
* Luồng hoạt động cho phép người dùng xem nội dung do những người mà họ theo dõi tải lên.
* Sử dụng khung xác thực Django

Django đi kèm với khung xác thực tích hợp có thể xử lý xác thực người dùng, phiên, quyền và nhóm người dùng. Hệ thống xác thực bao gồm các chế độ xem cho các hành động thông thường của người dùng như đăng nhập, đăng xuất, thay đổi mật khẩu và đặt lại mật khẩu.

* Đăng ký người dùng và hồ sơ người dùng

Người dùng trang web hiện có thể đăng nhập, đăng xuất, thay đổi mật khẩu và đặt lại mật khẩu của họ. Tuy nhiên, chúng ta cần xây dựng view để cho phép khách truy cập tạo tài khoản người dùng.

* Mở rộng mô hình người dùng

Khi xử lý tài khoản người dùng, bạn sẽ thấy rằng mô hình Người dùng của khung xác thực Django phù hợp với hầu hết các trường hợp phổ biến. Tuy nhiên, mô hình Người dùng tiêu chuẩn đi kèm với một số trường hạn chế. Bạn có thể muốn mở rộng nó với thông tin bổ sung có liên quan đến ứng dụng của bạn.

* Sử dụng khung tin nhắn

Khi người dùng tương tác với nền tảng, có nhiều trường hợp bạn có thể muốn thông báo cho họ về kết quả của các hành động cụ thể. Django có khung tin nhắn tích hợp cho phép bạn hiển thị thông báo một lần cho người dùng của mình.

Khung thông báo được đặt tại django.contrib.messages và được bao gồm trong danh sách INSTALLED\_APPS mặc định của tệp settings.py khi bạn tạo dự án mới bằng cách sử dụng python manager.py startproject. Tệp cài đặt cũng chứa phần mềm trung gian django.contrib.messages.middleware.

**Chương 5 : Thực hiện xã hội Xác thực**

Trong chương này, bạn sẽ thêm xác thực xã hội vào trang web của mình bằng Facebook, Google và Twitter. Bạn sẽ sử dụng Django Social Auth để triển khai xác thực xã hội bằng OAuth 2.0, giao thức ủy quyền tiêu chuẩn ngành . Bạn cũng sẽ sửa đổi quy trình xác thực xã hội để tự động tạo hồ sơ người dùng cho người dùng mới.

* Thêm xác thực xã hội vào trang web của bạn

Xác thực xã hội là một tính năng được sử dụng rộng rãi cho phép người dùng xác thực bằng tài khoản hiện có của nhà cung cấp dịch vụ bằng cách sử dụng Đăng nhập một lần (SSO). Quá trình xác thực cho phép người dùng xác thực vào trang web bằng tài khoản hiện có của họ từ các dịch vụ xã hội như Google. Trong phần này, chúng tôi sẽ thêm xác thực xã hội vào trang web bằng Facebook, Twitter và Google.

* Chạy máy chủ phát triển thông qua HTTPS

Một số phương thức xác thực xã hội mà chúng tôi sắp sử dụng yêu cầu kết nối HTTPS. Giao thức Bảo mật lớp vận chuyển (TLS) là tiêu chuẩn để phục vụ các trang web thông qua kết nối an toàn. Tiền thân của TLS là Lớp cổng bảo mật (SSL). Mặc dù SSL hiện không được dùng nữa nhưng trong nhiều thư viện và tài liệu trực tuyến, bạn sẽ tìm thấy tài liệu tham khảo cho cả hai điều khoản TLS và SSL. Máy chủ phát triển Django không thể phục vụ trang web của bạn thông qua HTTPS vì đó không phải là mục đích sử dụng của nó. Để kiểm tra chức năng xác thực xã hội phục vụ trang web thông qua HTTPS, chúng tôi sẽ sử dụng tiện ích mở rộng RunServerPlus của gói Django Exten-sions. Tiện ích mở rộng Django là bộ sưu tập các tiện ích mở rộng tùy chỉnh của bên thứ ba dành cho Django. Xin lưu ý rằng bạn không bao giờ nên sử dụng điều này để phục vụ trang web của mình trong môi trường thực; đây chỉ là một máy chủ phát triển.

* Xác thực bằng Facebook

Để sử dụng xác thực Facebook nhằm đăng nhập vào trang web của bạn, hãy thêm dòng in đậm sau đây vào cài đặt AUTHENTICATION\_BACKENDS trong tệp settings.py của dự án của bạn:

**A screenshot of a computer

Description automatically generated**

* Xác thực bằng Twitter

Để xác thực xã hội bằng Twitter, hãy thêm dòng in đậm sau đây vào AUTHENTICATION\_ Cài đặt BACKENDS trong tệp settings.py của dự án của bạn:

**A screenshot of a computer

Description automatically generated**

Bạn cần có tài khoản nhà phát triển Twitter. Mở https://developer.twitter.com/ trong trình duyệt của bạn và nhấp vào Đăng ký.

* Tạo hồ sơ cho người dùng đăng ký bằng xác thực xã hội

Khi người dùng xác thực bằng xác thực xã hội, một đối tượng Người dùng mới sẽ được tạo nếu không có người dùng hiện tại nào được liên kết với hồ sơ xã hội đó. Python Social Auth sử dụng một quy trình bao gồm một tập hợp các hàm được thực thi theo một thứ tự cụ thể được thực hiện trong luồng xác thực. Các chức năng này đảm nhiệm việc truy xuất mọi chi tiết người dùng, tạo hồ sơ xã hội trong cơ sở dữ liệu và liên kết nó với người dùng hiện có hoặc tạo một hồ sơ mới.

Thêm cài đặt SOCIAL\_AUTH\_PIPELINE sau vào tệp settings.py của dự án của bạn:

**A screenshot of a computer code

Description automatically generated**

Đây là quy trình xác thực mặc định được Python Social Auth sử dụng. Nó bao gồm một số chức năng thực hiện các tác vụ khác nhau khi xác thực người dùng. Bạn có thể tìm thêm thông tin chi tiết về quy trình xác thực mặc định tại <https://python-social-auth.readthedocs.io/en/latest/pipeline.html>.

**Chương 6 : Chia sẻ nội dung trên trang web của bạn**

Trong chương này, bạn sẽ tìm hiểu cách tạo một bookmarklet JavaScript để chia sẻ nội dung từ các trang web khác trên trang web của mình và bạn sẽ triển khai các tính năng AJAX trong dự án của mình bằng cách sử dụng JavaScript và Django.

Mã nguồn của chương này có thể được tìm thấy tại https://github.com/PacktPublishing/Django-4- by-example/tree/main/Chapter06.

* Tạo một trang web đánh dấu hình ảnh

Bây giờ chúng ta sẽ tìm hiểu cách cho phép người dùng đánh dấu các hình ảnh mà họ tìm thấy trên các trang web khác và chia sẻ chúng trên trang web của chúng tôi. Để xây dựng chức năng này, chúng ta sẽ cần các yếu tố sau:

1. Mô hình dữ liệu lưu trữ hình ảnh và thông tin liên quan.
2. Một biểu mẫu và dạng xem để xử lý việc tải lên hình ảnh.
3. Mã bookmarklet JavaScript có thể được thực thi trên bất kỳ trang web nào. Mã này sẽ tìm hình ảnh trên trang và cho phép người dùng chọn hình ảnh họ muốn đánh dấu.

* Tạo mối quan hệ nhiều-nhiều

Chúng ta sẽ thêm một trường khác vào mô hình Hình ảnh để lưu trữ những người dùng thích một hình ảnh. Chúng ta sẽ cần mối quan hệ nhiều-nhiều trong trường hợp này vì một người dùng có thể thích nhiều hình ảnh và mỗi hình ảnh có thể được nhiều người dùng thích.

Thêm trường sau vào mô hình Hình ảnh :

**A screenshot of a computer

Description automatically generated**

* Đăng ký model ảnh tại trang quản trị

Chỉnh sửa tệp admin.py của ứng dụng hình ảnh và đăng ký mô hình Hình ảnh vào trang quản trị, như sau:

**A screenshot of a computer

Description automatically generated**

* Đăng nội dung từ các trang web khác

Chúng tôi sẽ cho phép người dùng đánh dấu hình ảnh từ các trang web bên ngoài và chia sẻ chúng trên trang web của chúng tôi. Người dùng sẽ cung cấp URL của hình ảnh, tiêu đề và mô tả tùy chọn. Chúng ta sẽ tạo một biểu mẫu và dạng xem để tải hình ảnh xuống và tạo một đối tượng Hình ảnh mới trong cơ sở dữ liệu.

* Cài đặt thư viện Yêu cầu

Khi người dùng đánh dấu một hình ảnh, chúng ta sẽ cần tải file hình ảnh đó theo URL của nó. Chúng tôi sẽ sử dụng thư viện Yêu cầu Python cho mục đích này. Yêu cầu là thư viện HTTP phổ biến nhất cho Python. Nó tóm tắt sự phức tạp của việc xử lý các yêu cầu HTTP và cung cấp một giao diện rất đơn giản để sử dụng các dịch vụ HTTP. Bạn có thể tìm tài liệu về thư viện Yêu cầu tại https://requests. readthedocs.io/en/master/.

* Ghi đè phương thức save() của ModelForm

ModelForm cung cấp phương thức save() để lưu phiên bản mô hình hiện tại vào cơ sở dữ liệu và trả về đối tượng. Phương thức này nhận tham số cam kết Boolean , cho phép bạn chỉ định liệu đối tượng có được lưu giữ vào cơ sở dữ liệu hay không. Nếu cam kết là Sai, phương thức save() sẽ trả về một phiên bản mô hình nhưng sẽ không lưu nó vào cơ sở dữ liệu. Chúng ta sẽ ghi đè phương thức save() của biểu mẫu để truy xuất tệp hình ảnh theo URL đã cho và lưu nó vào hệ thống tệp.

* Xây dựng một bookmarklet bằng JavaScript

Bookmarklet là một bookmark được lưu trữ trong trình duyệt web có chứa mã JavaScript để mở rộng chức năng của trình duyệt. Khi bạn nhấp vào dấu trang trong dấu trang hoặc thanh yêu thích của trình duyệt, mã JavaScript sẽ được thực thi trên trang web đang được hiển thị trong trình duyệt. Điều này rất hữu ích cho việc xây dựng các công cụ tương tác với các trang web khác.

Một số dịch vụ trực tuyến, chẳng hạn như Pinterest, triển khai bookmarklet của riêng họ để cho phép người dùng chia sẻ nội dung từ các trang web khác lên nền tảng của họ. Bookmarklet Pinterest, nút trình duyệt được đặt tên, có sẵn tại https://about.pinterest.com/en/browser-button. Dấu trang Pinterest được cung cấp dưới dạng tiện ích mở rộng của Google Chrome, tiện ích bổ sung của Microsoft Edge hoặc dấu trang JavaScript đơn giản dành cho Safari và các trình duyệt khác mà bạn có thể kéo và thả vào thanh dấu trang của trình duyệt. Bookmarklet cho phép người dùng lưu hình ảnh hoặc trang web vào tài khoản Pinterest của họ.

* Tạo chế độ xem chi tiết cho hình ảnh

Bây giờ chúng ta hãy tạo một chế độ xem chi tiết đơn giản để hiển thị các hình ảnh đã được đánh dấu trên trang web. Mở tệp view.py của ứng dụng hình ảnh và thêm đoạn mã sau vào đó:

**A screen shot of a computer code

Description automatically generated**

Đây là một dạng xem đơn giản để hiển thị một hình ảnh. Chỉnh sửa tệp urls.py của ứng dụng hình ảnh và thêm mẫu URL sau được tô đậm:

**A screenshot of a computer code

Description automatically generated**

Chỉnh sửa tệp models.py của ứng dụng hình ảnh và thêm phương thức get\_absolute\_url() vào mô hình Hình ảnh , như sau:

**A screenshot of a computer

Description automatically generated**

Hãy nhớ rằng mẫu chung để cung cấp URL chuẩn cho các đối tượng là xác định get\_ phương thức tuyệt đối\_url() trong mô hình.

**Chương 7 : Theo dõi hành động của người dùng**

Trong chương này, bạn sẽ tìm hiểu cách xây dựng hệ thống theo dõi và tạo luồng hoạt động của người dùng. Bạn cũng sẽ khám phá cách tín hiệu Django hoạt động và tích hợp bộ lưu trữ I/O nhanh của Redis vào dự án của bạn để lưu trữ lượt xem mục.

Mã nguồn của chương này có thể được tìm thấy tại https://github.com/PacktPublishing/Django-4- by-example/tree/main/Chapter07.

* Xây dựng hệ thống theo dõi

Hãy xây dựng một hệ thống theo dõi trong dự án của bạn. Điều này có nghĩa là người dùng của bạn sẽ có thể theo dõi lẫn nhau và theo dõi những gì người dùng khác chia sẻ trên nền tảng. Mối quan hệ giữa những người dùng là mối quan hệ nhiều-nhiều: một người dùng có thể theo dõi nhiều người dùng và đến lượt họ, họ có thể được nhiều người dùng theo dõi.

* Tạo mối quan hệ nhiều-nhiều với mô hình trung gian

Trong các chương trước, bạn đã tạo mối quan hệ nhiều-nhiều bằng cách thêm ManyToManyField vào một trong các mô hình liên quan và để Django tạo bảng cơ sở dữ liệu cho mối quan hệ. Điều này phù hợp với hầu hết các trường hợp, nhưng đôi khi bạn có thể cần tạo ra một mô hình trung gian cho mối quan hệ.

* Xây dựng ứng dụng luồng hoạt động chung

Nhiều trang web xã hội hiển thị luồng hoạt động cho người dùng để họ có thể theo dõi những gì người dùng khác làm trên nền tảng. Luồng hoạt động là danh sách các hoạt động gần đây được thực hiện bởi một người dùng hoặc một nhóm người dùng.

Bạn sắp xây dựng một ứng dụng luồng hoạt động để mọi người dùng đều có thể thấy những tương tác gần đây của những người dùng mà họ theo dõi. Để làm như vậy, bạn sẽ cần một mô hình để lưu các hành động do người dùng thực hiện trên trang web và một cách đơn giản để thêm hành động vào nguồn cấp dữ liệu.

* Sử dụng khung nội dung

Django bao gồm một khung nội dung được đặt tại django.contrib.contenttypes. Ứng dụng này có thể theo dõi tất cả các mô hình được cài đặt trong dự án của bạn và cung cấp giao diện chung để tương tác với các mô hình của bạn. Theo mặc định, ứng dụng django.contrib.contenttypes được bao gồm trong cài đặt INSTALLED\_APPS khi bạn tạo một dự án mới bằng lệnh startproject . Nó được sử dụng bởi các gói đóng góp khác , chẳng hạn như khung xác thực và ứng dụng quản trị.

* Tránh các hành động trùng lặp trong luồng hoạt động

Đôi khi, người dùng của bạn có thể nhấp nhiều lần vào nút Thích hoặc Không thích hoặc thực hiện cùng một hành động nhiều lần trong một khoảng thời gian ngắn. Điều này sẽ dễ dẫn đến việc lưu trữ và hiển thị các hành động trùng lặp. Để tránh điều này, hãy cải thiện hàm create\_action() để bỏ qua các hành động trùng lặp rõ ràng.

* Tối ưu hóa các truy vấn liên quan đến các đối tượng liên quan

Mỗi khi bạn truy xuất một đối tượng Action , thông thường bạn sẽ truy cập vào đối tượng User liên quan của nó và đối tượng Profile liên quan của người dùng . Django ORM cung cấp một cách đơn giản để truy xuất các đối tượng liên quan cùng một lúc, do đó tránh được các truy vấn bổ sung vào cơ sở dữ liệu.

* Sử dụng select\_relative()

Django cung cấp một phương thức QuerySet có tên là select\_similar() cho phép bạn truy xuất các đối tượng liên quan cho mối quan hệ một-nhiều. Điều này chuyển thành một Bộ truy vấn phức tạp hơn, nhưng bạn tránh được các truy vấn có điều kiện bổ sung khi truy cập vào các đối tượng liên quan. Phương thức select\_relative dành cho ForeignKey và các trường OneToOne . Nó hoạt động bằng cách thực hiện THAM GIA SQL và bao gồm các trường của đối tượng liên quan trong câu lệnh SELECT.

* Các lớp cấu hình ứng dụng

Django cho phép bạn chỉ định các lớp cấu hình cho ứng dụng của mình. Khi bạn tạo một ứng dụng bằng lệnh startapp , Django sẽ thêm tệp apps.py vào thư mục ứng dụng, bao gồm cấu hình ứng dụng cơ bản kế thừa từ lớp AppConfig.

Lớp cấu hình ứng dụng cho phép bạn lưu trữ siêu dữ liệu và cấu hình cho ứng dụng , đồng thời nó cung cấp khả năng xem xét nội tâm cho ứng dụng. Bạn có thể tìm thêm thông tin về cấu hình ứng dụng tại <https://docs.djangoproject.com/en/4.1/ref/appluggest/>.

* Sử dụng Thanh công cụ gỡ lỗi Django

Tại thời điểm này, bạn đã quen với trang gỡ lỗi của Django. Trong suốt các chương trước, bạn đã thấy trang gỡ lỗi Django màu vàng và xám đặc biệt nhiều lần. Ví dụ: trong Chương 2, Cải thiện blog của bạn bằng các tính năng nâng cao, trong phần Xử lý lỗi phân trang, trang gỡ lỗi hiển thị thông tin liên quan đến các ngoại lệ chưa được xử lý khi triển khai phân trang đối tượng.

**Chương 8 : Xây dựng một cửa hàng trực tuyến**

Trong chương này, bạn sẽ bắt đầu một dự án Django mới bao gồm một cửa hàng trực tuyến có đầy đủ tính năng. Chương này và hai chương tiếp theo sẽ chỉ cho bạn cách xây dựng các chức năng thiết yếu của nền tảng thương mại điện tử. Cửa hàng trực tuyến của bạn sẽ cho phép khách hàng duyệt qua sản phẩm, thêm chúng vào giỏ hàng, áp dụng mã giảm giá, thực hiện quy trình thanh toán, thanh toán bằng thẻ tín dụng và nhận hóa đơn Bạn cũng sẽ triển khai một công cụ đề xuất để giới thiệu sản phẩm cho khách hàng của mình và bạn sẽ sử dụng tính năng quốc tế hóa để cung cấp trang web của mình bằng nhiều ngôn ngữ.

* Tạo dự án cửa hàng trực tuyến

Hãy bắt đầu với dự án Django mới để xây dựng một cửa hàng trực tuyến. Người dùng của bạn sẽ có thể duyệt qua danh mục sản phẩm và thêm sản phẩm vào giỏ hàng. Cuối cùng, họ sẽ có thể kiểm tra giỏ hàng và đặt hàng. Chương này sẽ đề cập đến các chức năng sau của một cửa hàng trực tuyến:

* Tạo các mô hình danh mục sản phẩm, thêm chúng vào trang quản trị và xây dựng các chế độ xem cơ bản để hiển thị danh mục.
* Xây dựng hệ thống giỏ hàng sử dụng phiên Django để cho phép người dùng giữ lại các sản phẩm đã chọn trong khi họ duyệt trang web.
* Tạo biểu mẫu và chức năng đặt hàng trên trang web.
* Gửi xác nhận email không đồng bộ cho người dùng khi họ đặt hàng.
* Tạo mô hình danh mục sản phẩm

Danh mục cửa hàng của bạn sẽ bao gồm các sản phẩm được sắp xếp thành các danh mục khác nhau. Mỗi sản phẩm sẽ có tên, mô tả tùy chọn, hình ảnh tùy chọn, giá cả và tình trạng còn hàng.

**A screenshot of a computer

Description automatically generated**

* Đăng ký mô hình catalog trên trang quản trị

Hãy thêm mô hình của bạn vào trang quản trị để bạn có thể dễ dàng quản lý danh mục và sản phẩm.

Chỉnh sửa tệp admin.py của ứng dụng cửa hàng và thêm mã sau vào đó:

**A screenshot of a computer

Description automatically generated**

Hãy nhớ rằng bạn sử dụng thuộc tính prepopulated\_fields để chỉ định các trường trong đó giá trị được đặt tự động bằng giá trị của các trường khác. Như bạn đã thấy trước đây, điều này rất thuận tiện cho việc tạo ra các phần mở rộng.

* Xây dựng giỏ hàng

Sau khi xây dựng danh mục sản phẩm, bước tiếp theo là tạo giỏ hàng để người dùng có thể chọn sản phẩm mình muốn mua. Giỏ hàng cho phép người dùng chọn sản phẩm và đặt số lượng họ muốn đặt hàng, sau đó lưu trữ thông tin này tạm thời trong khi họ duyệt trang web cho đến khi cuối cùng họ đặt hàng. Giỏ hàng phải được duy trì trong phiên để các mục trong giỏ hàng được duy trì trong suốt lượt truy cập của người dùng.

* Sử dụng phiên Django

Django cung cấp khung phiên hỗ trợ phiên ẩn danh và phiên người dùng. Khung phiên cho phép bạn lưu trữ dữ liệu tùy ý cho mỗi khách truy cập. Dữ liệu phiên được lưu trữ ở phía máy chủ và cookie chứa ID phiên trừ khi bạn sử dụng công cụ phiên dựa trên cookie. Phần mềm trung gian phiên quản lý việc gửi và nhận cookie. Công cụ phiên mặc định lưu trữ dữ liệu phiên trong cơ sở dữ liệu, nhưng bạn có thể chọn các công cụ phiên khác.

* Cài đặt phiên

Có một số cài đặt bạn có thể sử dụng để định cấu hình phiên cho dự án của mình. Điều quan trọng nhất là SESSION\_ENGINE. Cài đặt này cho phép bạn đặt nơi lưu trữ phiên. Theo mặc định, Django lưu trữ các phiên trong cơ sở dữ liệu bằng mô hình Phiên của ứng dụng django.contrib.sessions.

* Phiên hết hạn

Bạn có thể chọn sử dụng phiên kéo dài trên trình duyệt hoặc phiên liên tục bằng cách sử dụng SESSION\_EXPIRE\_AT\_ cài đặt BROWSER\_CLOSE . Điều này được đặt thành Sai theo mặc định, buộc thời lượng phiên thành giá trị được lưu trong cài đặt SESSION\_COOKIE\_AGE . Nếu bạn đặt SESSION\_EXPIRE\_AT\_BROWSER\_CLOSE thành True thì phiên sẽ hết hạn khi người dùng đóng trình duyệt và cài đặt SESSION\_COOKIE\_AGE sẽ không có bất kỳ ảnh hưởng nào.

* Tạo bộ xử lý ngữ cảnh cho giỏ hàng hiện tại

Bạn có thể nhận thấy thông báo Giỏ hàng của bạn trống được hiển thị trong tiêu đề của trang web, ngay cả khi giỏ hàng chứa các mặt hàng. Thay vào đó, bạn nên hiển thị tổng số mặt hàng trong giỏ hàng và tổng chi phí. Vì điều này phải được hiển thị trên tất cả các trang nên bạn cần xây dựng bộ xử lý ngữ cảnh để đưa giỏ hàng hiện tại vào ngữ cảnh yêu cầu, bất kể chế độ xem xử lý yêu cầu như thế nào.

* Bộ xử lý bối cảnh

Bộ xử lý ngữ cảnh là một hàm Python lấy đối tượng yêu cầu làm đối số và trả về một từ điển được thêm vào ngữ cảnh yêu cầu. Bộ xử lý bối cảnh sẽ hữu ích khi bạn cần cung cấp nội dung nào đó trên toàn cầu cho tất cả các mẫu.

* Đặt giỏ hàng vào ngữ cảnh yêu cầu

Hãy tạo một bộ xử lý ngữ cảnh để đặt giỏ hàng hiện tại vào ngữ cảnh yêu cầu. Với nó, bạn sẽ có thể truy cập vào giỏ hàng trong bất kỳ mẫu nào.

Tạo một tệp mới bên trong thư mục ứng dụng giỏ hàng và đặt tên là context\_processors.py. Bộ xử lý bối cảnh có thể nằm ở bất kỳ đâu trong mã của bạn nhưng việc tạo chúng ở đây sẽ giữ cho mã của bạn được tổ chức tốt. Thêm mã sau vào tập tin:

**A close-up of words

Description automatically generated**

* Đăng ký đơn hàng của khách hàng

Khi giỏ hàng được kiểm tra, bạn cần lưu đơn hàng vào cơ sở dữ liệu. Đơn hàng sẽ chứa thông tin về khách hàng và sản phẩm họ đang mua.

* Tạo mô hình đơn hàng

Mô hình Đơn hàng chứa một số trường để lưu trữ thông tin khách hàng và trường Boolean trả phí , mặc định là Sai. Sau này, bạn sẽ sử dụng trường này để phân biệt giữa các đơn hàng đã thanh toán và chưa thanh toán. Chúng ta cũng đã xác định phương thức get\_total\_cost() để tính tổng chi phí của các mặt hàng đã mua trong đơn hàng này. Model OrderItem cho phép bạn lưu trữ sản phẩm, số lượng và giá thanh toán cho từng mặt hàng. Chúng tôi đã xác định phương thức get\_cost() trả về giá của mặt hàng bằng cách nhân giá mặt hàng với số lượng.

* Bao gồm các mô hình đặt hàng trong trang quản trị

Tạo đơn hàng của khách hàng Bạn sẽ sử dụng các mô hình đơn hàng mà bạn đã tạo để duy trì các mặt hàng có trong giỏ hàng khi người dùng cuối cùng đặt hàng. Một đơn hàng mới sẽ được tạo theo các bước sau:

1. Trình bày cho người dùng một mẫu đơn đặt hàng để điền dữ liệu của họ.
2. Tạo một phiên bản Order mới với dữ liệu đã nhập và tạo một phiên bản OrderItem được liên kết cho từng món hàng trong giỏ hàng.
3. Xóa tất cả nội dung của giỏ hàng và chuyển hướng người dùng đến trang thành công.

* Nhiệm vụ không đồng bộ

Khi nhận được yêu cầu HTTP, bạn cần trả về phản hồi cho người dùng càng nhanh càng tốt.

Hãy nhớ rằng trong Chương 7, Theo dõi hành động của người dùng, bạn đã sử dụng Thanh công cụ gỡ lỗi Django để kiểm tra thời gian cho các giai đoạn khác nhau của chu kỳ yêu cầu/phản hồi và thời gian thực hiện cho các truy vấn SQL được thực hiện.

* Làm việc với các tác vụ không đồng bộ

Chúng tôi có thể giảm tải công việc khỏi chu kỳ yêu cầu/phản hồi bằng cách thực hiện một số tác vụ nhất định trong nền. Ví dụ: nền tảng chia sẻ video cho phép người dùng tải video lên nhưng cần nhiều thời gian để chuyển mã các video đã tải lên. Khi người dùng tải video lên, trang web có thể trả về phản hồi thông báo rằng quá trình chuyển mã sẽ sớm bắt đầu và bắt đầu chuyển mã video không đồng bộ. Một ví dụ khác là gửi email cho người dùng. Nếu trang web của bạn gửi thông báo qua email từ một chế độ xem, Giao thức truyền thư đơn giản.

* Công nhân, hàng đợi tin nhắn và người môi giới tin nhắn

Trong khi máy chủ web của bạn xử lý các yêu cầu và trả về phản hồi, bạn cần một máy chủ dựa trên tác vụ thứ hai, có tên là worker, để xử lý các tác vụ không đồng bộ. Một hoặc nhiều công nhân có thể chạy và thực thi các tác vụ ở chế độ nền. Những công nhân này có thể truy cập cơ sở dữ liệu, xử lý tệp, gửi e-mail, v.v. Công nhân thậm chí có thể xếp hàng các nhiệm vụ trong tương lai. Tất cả đều đồng thời giữ cho máy chủ web chính được rảnh tay để xử lý các yêu cầu HTTP.

**Chương 9 : Quản lý thanh toán và đơn đặt hàng**

Trong chương này, bạn sẽ tìm hiểu cách tích hợp cổng thanh toán vào trang web của mình để cho phép người dùng thanh toán bằng thẻ tín dụng. Bạn cũng sẽ mở rộng trang quản trị với các tính năng khác nhau.

Mã nguồn của chương này có thể được tìm thấy tại https://github.com/PacktPublishing/Django-4- by-example/tree/main/Chapter09.

* Tích hợp cổng thanh toán

Cổng thanh toán là công nghệ được người bán sử dụng để xử lý thanh toán trực tuyến từ khách hàng. Bằng cách sử dụng cổng thanh toán, bạn có thể quản lý đơn đặt hàng của khách hàng và ủy quyền xử lý thanh toán cho bên thứ ba an toàn, đáng tin cậy. Bằng cách sử dụng cổng thanh toán đáng tin cậy, bạn sẽ không phải lo lắng về sự phức tạp về kỹ thuật, bảo mật và quy định khi xử lý thẻ tín dụng trong hệ thống của riêng bạn.

Có một số nhà cung cấp cổng thanh toán để lựa chọn. Chúng tôi sẽ tích hợp Stripe, một cổng thanh toán rất phổ biến được sử dụng bởi các dịch vụ trực tuyến như Shopify, Uber, Twitch và GitHub, cùng nhiều dịch vụ khác.

* Tạo tài khoản Stripe

Bạn cần có tài khoản Stripe để tích hợp cổng thanh toán vào trang web của mình. Hãy tạo một tài khoản để kiểm tra API Stripe. Mở https://dashboard.stripe.com/register trong trình duyệt của bạn. Bạn sẽ thấy một biểu mẫu giống như sau:

**A screenshot of a login form

Description automatically generated**

Điền vào biểu mẫu dữ liệu của riêng bạn và nhấp vào Tạo tài khoản.

* Cài đặt thư viện Stripe Python

Stripe cung cấp thư viện Python giúp đơn giản hóa việc xử lý API của nó. Chúng tôi sẽ tích hợp cổng thanh toán vào dự án bằng thư viện sọc . Bạn có thể tìm mã nguồn của thư viện Stripe Python tại <https://github.com/stripe/stripe-python>.

* Thêm Stripe vào dự án của bạn

Mở https://dashboard.stripe.com/test/apikeys trong trình duyệt của bạn. Bạn cũng có thể truy cập trang này từ bảng điều khiển Stripe bằng cách nhấp vào Nhà phát triển và sau đó nhấp vào khóa API. Bạn sẽ thấy màn hình sau:

**A screenshot of a computer

Description automatically generated**

Hãy tích hợp cổng thanh toán vào quy trình thanh toán. Bạn có thể tìm tài liệu Python cho Stripe tại https:// stripe.com/docs/api?lang=python.

* Xây dựng quy trình thanh toán

Quá trình thanh toán sẽ diễn ra như sau:

1. Thêm sản phẩm vào giỏ hàng.
2. Kiểm tra giỏ hàng.
3. Nhập chi tiết thẻ tín dụng và thanh toán.

* Tích hợp Stripe Checkout

Tích hợp Stripe Checkout bao gồm một trang thanh toán do Stripe lưu trữ cho phép người dùng nhập chi tiết thanh toán, thường là thẻ tín dụng và thu khoản thanh toán. Nếu thanh toán thành công, Stripe sẽ chuyển hướng khách hàng đến trang thành công. Nếu khách hàng hủy khoản thanh toán, nó sẽ chuyển hướng khách hàng đến trang hủy.

Chúng tôi sẽ thực hiện ba quan điểm:

* Payment\_process: Tạo Phiên thanh toán Stripe và chuyển hướng khách hàng đến hình thức thanh toán được lưu trữ trên máy chủ Stripe. Phiên thanh toán là sự trình bày theo chương trình về những gì khách hàng nhìn thấy khi họ được chuyển hướng đến hình thức thanh toán, bao gồm các sản phẩm, số lượng, đơn vị tiền tệ và số tiền cần tính phí.
* Payment\_completed: Hiển thị thông báo thanh toán thành công. Người dùng được chuyển hướng đến chế độ xem này nếu thanh toán thành công.
* Payment\_canceled: Hiển thị thông báo thanh toán bị hủy. Người dùng được chuyển hướng đến chế độ xem này nếu thanh toán bị hủy.

**A screenshot of a computer program

Description automatically generated**

Quá trình thanh toán hoàn chỉnh sẽ diễn ra như sau:

1. Sau khi đơn hàng được tạo, người dùng sẽ được chuyển hướng đến chế độ xem Payment\_process . Người dùng sẽ được gửi trước bản tóm tắt đơn hàng và một nút để tiến hành thanh toán.
2. Khi người dùng tiến hành thanh toán, phiên thanh toán Stripe sẽ được tạo. Phiên thanh toán bao gồm danh sách các mặt hàng mà người dùng sẽ mua, URL để chuyển hướng người dùng đến sau khi thanh toán thành công và URL để chuyển hướng người dùng đến nếu thanh toán bị hủy.
3. Chế độ xem chuyển hướng người dùng đến trang thanh toán được lưu trữ trên máy chủ Stripe. Trang này bao gồm thanh toán hình thức. Khách hàng nhập chi tiết thẻ tín dụng của họ và gửi biểu mẫu.
4. Stripe xử lý thanh toán và chuyển hướng khách hàng đến chế độ xem Payment\_completed . Nếu khách hàng không hoàn tất thanh toán, thay vào đó, Stripe sẽ chuyển hướng khách hàng đến chế độ xem Payment\_canceled.

* Sử dụng webhooks để nhận thông báo thanh toán

Stripe có thể đẩy các sự kiện theo thời gian thực vào ứng dụng của chúng ta bằng cách sử dụng webhook. Webhook , còn được gọi là lệnh gọi lại, có thể được coi là API hướng sự kiện thay vì API hướng yêu cầu. Thay vì thường xuyên thăm dò API Stripe để biết khi nào một khoản thanh toán mới hoàn tất, Stripe có thể gửi yêu cầu HTTP tới URL trong ứng dụng của chúng tôi để thông báo về các khoản thanh toán thành công trong thời gian thực. Thông báo về những sự kiện này sẽ không đồng bộ khi sự kiện xảy ra, bất kể lệnh gọi đồng bộ của chúng tôi tới API Stripe.

* Tạo điểm cuối webhook

Bạn có thể thêm URL điểm cuối webhook vào tài khoản Stripe của mình để nhận sự kiện. Vì chúng tôi đang sử dụng webhook và không có trang web được lưu trữ có thể truy cập thông qua URL công khai nên chúng tôi sẽ sử dụng Giao diện dòng lệnh Stripe (CLI) để theo dõi các sự kiện và chuyển tiếp chúng đến môi trường cục bộ của chúng tôi.

Mở https://dashboard.stripe.com/test/webhooks trong trình duyệt của bạn.

* Xuất đơn hàng sang tệp CSV

Đôi khi, bạn có thể muốn xuất thông tin có trong một mô hình sang một tệp để có thể nhập thông tin đó vào hệ thống khác. Một trong những định dạng được sử dụng rộng rãi nhất để xuất/nhập dữ liệu là Giá trị được phân tách bằng dấu phẩy (CSV). Tệp CSV là một tệp văn bản thuần túy bao gồm một số bản ghi. Thường có một bản ghi trên mỗi dòng và một số ký tự phân cách, thường là dấu phẩy, ngăn cách các trường bản ghi. Chúng tôi sẽ tùy chỉnh trang quản trị để có thể xuất đơn hàng sang tệp CSV.

* Thêm hành động tùy chỉnh vào trang quản trị

Django cung cấp nhiều tùy chọn để tùy chỉnh trang quản trị. Bạn sẽ sửa đổi chế độ xem danh sách đối tượng để bao gồm hành động quản trị tùy chỉnh. Bạn có thể triển khai các hành động quản trị tùy chỉnh để cho phép người dùng nhân viên áp dụng các hành động cho nhiều thành phần cùng một lúc trong chế độ xem danh sách thay đổi.

Một hành động quản trị hoạt động như sau: người dùng chọn các đối tượng từ trang danh sách đối tượng quản trị bằng các hộp kiểm, sau đó họ chọn một hành động để thực hiện trên tất cả các mục đã chọn và thực hiện các hành động đó. Hình 9.26 cho thấy vị trí của các hành động trong trang quản trị:

**A screenshot of a computer

Description automatically generated**

Bạn có thể tạo một hành động tùy chỉnh bằng cách viết một hàm thông thường nhận các tham số sau:

* ModelAdmin hiện tại đang được hiển thị.
* Đối tượng yêu cầu hiện tại là một phiên bản HttpRequest.
* Bộ truy vấn cho các đối tượng được người dùng chọn.
* Mở rộng trang quản trị với chế độ xem tùy chỉnh

Đôi khi, bạn có thể muốn tùy chỉnh trang quản trị ngoài những gì có thể thông qua việc định cấu hình ModelAdmin, tạo các hành động quản trị và ghi đè các mẫu quản trị. Bạn có thể muốn triển khai các chức năng bổ sung không có sẵn trong các dạng xem hoặc mẫu quản trị hiện có. Trong trường hợp này, bạn cần tạo chế độ xem quản trị tùy chỉnh. Với chế độ xem tùy chỉnh, bạn có thể xây dựng bất kỳ chức năng nào bạn muốn; bạn chỉ cần đảm bảo rằng chỉ người dùng là nhân viên mới có thể truy cập vào chế độ xem của bạn và bạn duy trì giao diện quản trị bằng cách làm cho mẫu của bạn mở rộng mẫu quản trị.

* Tạo hóa đơn PDF một cách linh hoạt

Bây giờ bạn đã có hệ thống thanh toán và thanh toán hoàn chỉnh, bạn có thể tạo hóa đơn PDF cho mỗi đơn hàng. Có một số thư viện Python để tạo tệp PDF. Một thư viện phổ biến để tạo tệp PDF bằng mã Python là ReportLab. Bạn có thể tìm thông tin về cách xuất tệp PDF bằng ReportLab tại https://docs.djangoproject.com/en/4.1/howto/ outputting-pdf/.

Trong hầu hết các trường hợp, bạn sẽ phải thêm kiểu và định dạng tùy chỉnh vào tệp PDF của mình. Bạn sẽ thấy thuận tiện hơn khi hiển thị mẫu HTML và chuyển đổi nó thành tệp PDF, giúp Python tránh xa lớp trình bày. Bạn sẽ làm theo phương pháp này và sử dụng mô-đun để tạo tệp PDF bằng Django. Bạn sẽ sử dụng WeasyPrint, đây là thư viện Python có thể tạo tệp PDF từ các mẫu HTML.

* Tạo mẫu PDF

Bạn cần một tài liệu HTML làm đầu vào cho WeeasyPrint. Bạn sẽ tạo một mẫu HTML, hiển thị nó bằng Django và chuyển nó tới WeeasyPrint để tạo tệp PDF.

* Hiển thị tệp PDF

Bạn sẽ tạo chế độ xem để tạo hóa đơn PDF cho các đơn hàng hiện có bằng cách sử dụng trang quản trị. Chỉnh sửa tệp view.py bên trong thư mục ứng dụng đặt hàng và thêm đoạn mã sau vào đó:

**A computer screen shot of a program

Description automatically generated**

A close-up of a text

Description automatically generated

Đây là chế độ xem để tạo hóa đơn PDF cho một đơn đặt hàng. Bạn sử dụng trang trí Staff\_member\_required để đảm bảo chỉ người dùng nhân viên mới có thể truy cập chế độ xem này.

**Chương 10 : Mở rộng cửa hàng của bạn**

Trong chương này, bạn sẽ thêm hệ thống phiếu giảm giá vào cửa hàng của mình và tạo công cụ đề xuất sản phẩm.

* Tạo hệ thống phiếu giảm giá

Bạn sẽ tạo một hệ thống phiếu giảm giá cho cửa hàng của mình. Phiếu giảm giá của bạn sẽ có giá trị cho khách hàng trong một khung thời gian nhất định. Các phiếu giảm giá sẽ không có bất kỳ giới hạn nào về số lần có thể đổi và chúng sẽ được áp dụng cho tổng giá trị của giỏ hàng.

Đối với chức năng này, bạn sẽ cần tạo một mô hình để lưu trữ mã phiếu giảm giá, khung thời gian hợp lệ và mức giảm giá để áp dụng.

* Áp dụng phiếu giảm giá vào giỏ hàng

Bạn có thể lưu trữ các phiếu giảm giá mới và thực hiện truy vấn để truy xuất các phiếu giảm giá hiện có. Bây giờ bạn cần một cách để khách hàng áp dụng phiếu giảm giá khi mua hàng. Chức năng áp dụng phiếu giảm giá sẽ như sau:

1. Người dùng thêm sản phẩm vào giỏ hàng.
2. Người dùng có thể nhập mã phiếu giảm giá theo mẫu hiển thị trên trang chi tiết giỏ hàng.
3. Khi người dùng nhập mã phiếu giảm giá và gửi biểu mẫu, bạn sẽ tìm phiếu giảm giá hiện có với mã đã cho và hiện đang hợp lệ. Bạn phải kiểm tra xem mã phiếu giảm giá có khớp với mã do người dùng nhập hay không, thuộc tính hoạt động có đúng không và ngày giờ hiện tại có nằm giữa giá trị valid\_from và valid\_to hay không .
4. Nếu tìm thấy phiếu giảm giá, bạn lưu phiếu giảm giá đó trong phiên của người dùng và hiển thị giỏ hàng, bao gồm mức giảm giá áp dụng cho phiếu giảm giá đó và tổng số tiền được cập nhật.
5. Khi người dùng đặt hàng, bạn lưu phiếu giảm giá vào đơn hàng đã cho.

* Xây dựng công cụ đề xuất

Công cụ đề xuất là một hệ thống dự đoán mức độ ưu tiên hoặc xếp hạng mà người dùng sẽ dành cho một mặt hàng. Hệ thống chọn các mục có liên quan cho người dùng dựa trên hành vi của họ và kiến thức mà hệ thống có về họ. Ngày nay, hệ thống khuyến nghị được sử dụng trong nhiều dịch vụ trực tuyến. Họ giúp đỡ người dùng bằng cách chọn nội dung họ có thể quan tâm từ lượng dữ liệu khổng lồ có sẵn nhưng không liên quan đến họ. Đưa ra các đề xuất tốt giúp tăng cường sự tham gia của người dùng. Các trang web thương mại điện tử cũng được hưởng lợi từ việc đưa ra các đề xuất sản phẩm có liên quan bằng cách tăng doanh thu trung bình trên mỗi người dùng.

* Đề xuất sản phẩm dựa trên những lần mua hàng trước đó

Chúng ta sẽ giới thiệu sản phẩm cho người dùng dựa trên các mặt hàng thường xuyên mua cùng nhau. Để làm được điều đó, chúng tôi sẽ lưu trữ một khóa trong Redis cho mỗi sản phẩm được mua trên trang web. Khóa sản phẩm sẽ chứa tập hợp được sắp xếp Redis với điểm số. Mỗi lần mua hàng mới hoàn tất, chúng tôi sẽ cộng thêm 1 điểm cho mỗi sản phẩm được mua cùng nhau. Bộ được sắp xếp sẽ cho phép bạn chấm điểm cho các sản phẩm được mua cùng nhau. Chúng ta sẽ lấy số lần mua sản phẩm cùng với sản phẩm khác làm điểm cho mặt hàng đó.

**Chương 11 : Thêm quốc tế hóa vào cửa hàng của bạn**

Trong chương này , bạn sẽ tìm hiểu cách hoạt động của quốc tế hóa và bản địa hóa.

* Quốc tế hóa với Django

Django cung cấp hỗ trợ quốc tế hóa và bản địa hóa đầy đủ. Nó cho phép bạn dịch ứng dụng của mình sang nhiều ngôn ngữ và xử lý định dạng cụ thể theo địa phương cho ngày, giờ, số và múi giờ. Hãy làm rõ sự khác biệt giữa quốc tế hóa và nội địa hóa:

Quốc tế hóa (thường được viết tắt là i18n) là quá trình điều chỉnh phần mềm cho khả năng sử dụng các ngôn ngữ và địa phương khác nhau để phần mềm không bị gắn cứng vào một ngôn ngữ hoặc địa điểm cụ thể.

Bản địa hóa (viết tắt là l10n) là quá trình dịch phần mềm thực sự và điều chỉnh nó sang một ngôn ngữ cụ thể. Bản thân Django được dịch sang hơn 50 ngôn ngữ bằng cách sử dụng khuôn khổ quốc tế hóa của nó.

* Cài đặt quốc tế hóa và bản địa hóa

Django cung cấp một số cài đặt để quốc tế hóa. Các cài đặt sau đây có liên quan nhất:

* USE\_I18N: Boolean chỉ định xem hệ thống dịch thuật của Django có được bật hay không. Điều này là Đúng theo mặc định.
* USE\_L10N: Boolean cho biết định dạng bản địa hóa có được bật hay không. Khi hoạt động, các định dạng bản địa hóa được sử dụng để thể hiện ngày và số. Đây là Sai theo mặc định.
* USE\_TZ: Boolean chỉ định liệu ngày giờ có nhận biết được múi giờ hay không. Khi bạn tạo một dự án bằng lệnh startproject , điều này được đặt thành True.
* Lệnh quản lý quốc tế hóa

Django bao gồm các lệnh quản lý sau để quản lý bản dịch:

* makemessages: Lệnh này chạy trên cây nguồn để tìm tất cả các chuỗi được đánh dấu để dịch và tạo hoặc cập nhật các tệp tin nhắn .po trong thư mục miền địa phương . Một tệp .po duy nhất được tạo cho mỗi ngôn ngữ.
* biên dịch tin nhắn: Điều này biên dịch các tệp thông báo .po hiện có thành các tệp .mo , được sử dụng để truy xuất bản dịch..
* Cài đặt bộ công cụ gettext

Bạn sẽ cần bộ công cụ gettext để có thể tạo, cập nhật và biên dịch các tệp tin nhắn. Hầu hết các bản phân phối Linux đều có bộ công cụ gettext. Nếu bạn đang sử dụng macOS, cách đơn giản nhất để cài đặt nó là thông qua Homebrew, tại <https://brew.sh>.

Nếu bạn đang sử dụng Windows, hãy làm theo các bước tại https://docs.djangoproject.com/en/4.1/topics/ i18n/translation/#gettext-on-windows. Bạn có thể tải xuống trình cài đặt nhị phân gettext được biên dịch sẵn cho Windows từ <https://mlocati.github.io/articles/gettext-iconv-windows.html>.

* Cách thêm bản dịch vào dự án Django

Cách thêm bản dịch vào dự án Django:

1. Đánh dấu các chuỗi cần dịch trong mã Python và các mẫu của bạn.
2. Chạy lệnh makemessages để tạo hoặc cập nhật các tập tin tin nhắn bao gồm tất cả các nội dung truyền chuỗi quan hệ từ mã của bạn.
3. Dịch các chuỗi có trong tệp tin nhắn và biên dịch chúng bằng cách sử dụng các thông báo biên dịch lệnh quản lý.

* Dịch mã Python

Để dịch các chữ trong mã Python, bạn có thể đánh dấu các chuỗi cần dịch bằng cách sử dụng gettext() chức năng có trong django.utils.translation. Hàm này dịch tin nhắn và trả về một chuỗi. Quy ước là nhập hàm này dưới dạng bí danh ngắn hơn có tên \_ (ký tự gạch dưới).

**Chương 12 : Xây dựng nền tảng học tập điện tử**

Trong chương này, bạn sẽ bắt đầu một dự án Django mới bao gồm một nền tảng học tập điện tử với hệ thống quản lý nội dung (CMS) của riêng bạn . Nền tảng học tập trực tuyến là một ví dụ tuyệt vời về các ứng dụng mà bạn cần cung cấp các công cụ để tạo nội dung một cách linh hoạt.

* Thiết lập dự án e-learning

Dự án thực tế cuối cùng của bạn sẽ là một nền tảng học tập điện tử. Đầu tiên, tạo môi trường ảo cho dự án mới của bạn trong thư mục env/ bằng lệnh sau:

****

Nếu bạn đang sử dụng Windows, thay vào đó hãy sử dụng lệnh sau:



* Phục vụ các tập tin media

Trước khi tạo mô hình cho các khóa học và nội dung khóa học, chúng tôi sẽ chuẩn bị dự án để phục vụ các tệp phương tiện. Người hướng dẫn khóa học sẽ có thể tải các tệp phương tiện lên nội dung khóa học bằng cách sử dụng CMS mà chúng tôi sẽ xây dựng. Do đó, chúng tôi sẽ cấu hình dự án để phục vụ các tệp phương tiện.

* Tạo các mô hình Nội dung

Mô hình Nội dung của ứng dụng khóa học của bạn chứa một mối quan hệ chung để liên kết các loại nội dung khác nhau với nó. Bạn sẽ tạo một mô hình khác nhau cho từng loại nội dung. Tất cả các mô hình Nội dung sẽ có một số trường chung và các trường bổ sung để lưu trữ dữ liệu tùy chỉnh. Bạn sẽ tạo một mô hình trừu tượng cung cấp các trường chung cho tất cả các mô hình Nội dung.

* Thêm chế độ xem xác thực

Bây giờ bạn đã tạo mô hình dữ liệu đa hình, bạn sẽ xây dựng một CMS để quản lý các khóa học và nội dung của chúng. Bước đầu tiên là thêm hệ thống xác thực cho CMS.

* Thêm hệ thống xác thực

Bạn sẽ sử dụng khung xác thực của Django để người dùng xác thực với nền tảng học tập điện tử. Cả người hướng dẫn và sinh viên sẽ là phiên bản của mô hình Người dùng của Django , vì vậy họ sẽ có thể đăng nhập vào trang web bằng cách sử dụng chế độ xem xác thực của django.contrib.auth.

**Chương 13 : Tạo quản lý nội dung hệ thống**

Trong chương này, bạn sẽ tìm hiểu cách xây dựng chức năng dành cho người hướng dẫn để tạo các khóa học và quản lý nội dung của các khóa học đó một cách linh hoạt và hiệu quả.

* Tạo một CMS

Bây giờ bạn đã tạo một mô hình dữ liệu linh hoạt, bạn sẽ xây dựng CMS. CMS sẽ cho phép người hướng dẫn tạo các khóa học và quản lý nội dung của họ. Bạn cần cung cấp các chức năng sau:

* Liệt kê các khóa học được tạo bởi người hướng dẫn.
* Tạo, chỉnh sửa và xóa các khóa học.
* Thêm mô-đun vào khóa học và sắp xếp lại chúng.
* Thêm các loại nội dung khác nhau vào mỗi học phần.
* Sắp xếp lại các mô-đun và nội dung khóa học.
* Làm việc với các nhóm và quyền

Hiện tại, bất kỳ người dùng nào cũng có thể truy cập vào chế độ xem để quản lý các khóa học. Bạn muốn hạn chế những chế độ xem này để chỉ những người hướng dẫn mới có quyền tạo và quản lý các khóa học.

Khung xác thực của Django bao gồm một hệ thống cấp phép cho phép bạn gán quyền cho người dùng và nhóm. Bạn sẽ tạo một nhóm cho người dùng là người hướng dẫn và chỉ định quyền tạo, cập nhật và xóa các khóa học.

* Quản lý các mô-đun khóa học và nội dung của chúng

Bạn sắp xây dựng một hệ thống để quản lý các mô-đun khóa học và nội dung của chúng. Bạn sẽ cần xây dựng các biểu mẫu có thể được sử dụng để quản lý nhiều mô-đun cho mỗi khóa học và các loại nội dung khác nhau cho mỗi mô-đun. Cả mô-đun và nội dung của chúng sẽ phải tuân theo một thứ tự cụ thể và bạn có thể sắp xếp lại chúng bằng CMS.

* Thêm nội dung vào mô-đun khóa học

Bây giờ, bạn cần một cách để thêm nội dung vào các mô-đun khóa học. Bạn có bốn loại nội dung khác nhau: văn bản, video, hình ảnh và tệp. Bạn có thể cân nhắc việc tạo bốn chế độ xem khác nhau để tạo nội dung, mỗi chế độ xem cho mỗi mô hình. Tuy nhiên, bạn sẽ thực hiện một cách tiếp cận tổng quát hơn và tạo một khung nhìn xử lý việc tạo hoặc cập nhật các đối tượng của bất kỳ mô hình nội dung nào.

**Chương 14 : Hiển thị và lưu nội dung vào bộ nhớ đệm**

Trong chương này, bạn sẽ xây dựng chức năng truy cập nội dung khóa học, tạo hệ thống đăng ký sinh viên và quản lý việc đăng ký sinh viên vào các khóa học. Bạn cũng sẽ tìm hiểu cách lưu trữ dữ liệu vào bộ đệm bằng khung bộ đệm Django.

* Hiển thị các khóa học

Đối với danh mục khóa học của bạn, bạn phải xây dựng các chức năng sau:

* Liệt kê tất cả các khóa học có sẵn, tùy chọn lọc theo chủ đề
* Hiển thị tổng quan về khóa học
* Thêm đăng ký sinh viên

Tạo một ứng dụng mới bằng lệnh sau:

**A screenshot of a computer

Description automatically generated**

* Tạo chế độ xem đăng ký sinh viên

Đây là chế độ xem cho phép sinh viên đăng ký trên trang web của bạn. Bạn sử dụng CreateView chung , cung cấp chức năng tạo đối tượng mô hình. Chế độ xem này yêu cầu các thuộc tính sau:

* template\_name: Đường dẫn của mẫu để hiển thị chế độ xem này.
* form\_class: Biểu mẫu tạo đối tượng, phải là ModelForm. Bạn sử dụng Django.
* Success\_url: URL để chuyển hướng người dùng đến khi biểu mẫu được gửi thành công.
* Đăng ký các khóa học

Sau khi người dùng tạo tài khoản, họ sẽ có thể đăng ký các khóa học. Để lưu trữ thông tin đăng ký, bạn cần tạo mối quan hệ nhiều-nhiều giữa mô hình Khóa học và Người dùng.

Bạn sẽ sử dụng biểu mẫu này để sinh viên đăng ký vào các khóa học. Trường khóa học dành cho khóa học mà người dùng sẽ đăng ký; do đó, đó là ModelChoiceField. Bạn sử dụng tiện ích HiddenInput vì bạn sẽ không hiển thị trường này cho người dùng. Bạn sẽ sử dụng biểu mẫu này trong chế độ xem CourseDetailView để hiển thị nút đăng ký.

* Truy cập nội dung khóa học

Bạn cần một chế độ xem để hiển thị các khóa học mà sinh viên đã đăng ký và một chế độ xem để truy cập nội dung khóa học thực tế. Chỉnh sửa tệp view.py của ứng dụng dành cho sinh viên và thêm mã sau vào đó:

**A screenshot of a computer program

Description automatically generated**

* Sử dụng khung bộ đệm

Việc xử lý các yêu cầu HTTP tới ứng dụng web của bạn thường đòi hỏi phải truy cập cơ sở dữ liệu, thao tác dữ liệu và hiển thị mẫu. Về mặt xử lý, nó đắt hơn nhiều so với việc chỉ phục vụ một trang web tĩnh. Chi phí trong một số yêu cầu có thể đáng kể khi trang web của bạn bắt đầu nhận được ngày càng nhiều lưu lượng truy cập. Đây là nơi bộ nhớ đệm trở nên quý giá. Bằng cách lưu các truy vấn vào bộ đệm, kết quả tính toán hoặc hiển thị nội dung trong yêu cầu HTTP, bạn sẽ tránh được các thao tác tốn kém trong các yêu cầu cần trả về cùng một dữ liệu sau đây. Điều này chuyển thành thời gian phản hồi ngắn hơn và xử lý ít hơn ở phía máy chủ.

* Các đoạn mẫu trong bộ nhớ đệm

Các đoạn mẫu lưu vào bộ nhớ đệm là một cách tiếp cận cấp cao hơn. Bạn cần tải các thẻ mẫu bộ đệm trong mẫu của mình bằng cách sử dụng {% tải bộ đệm %}. Sau đó, bạn sẽ có thể sử dụng thẻ mẫu {% cache %} để lưu vào bộ đệm các đoạn mẫu cụ thể. Bạn thường sẽ sử dụng thẻ mẫu như sau:

**A white background with black text

Description automatically generated**

**Chương 15 : Xây dựng API**

Trong chương này, bạn sẽ tạo API RESTful cho nền tảng học tập trực tuyến của mình. API cho phép bạn xây dựng một lõi chung có thể được sử dụng trên nhiều nền tảng như trang web, ứng dụng di động, plugin, v.v. Ví dụ: bạn có thể tạo một API để ứng dụng di động sử dụng cho nền tảng học tập trực tuyến của mình. Nếu bạn cung cấp API cho bên thứ ba, họ sẽ có thể sử dụng thông tin và vận hành với ứng dụng của bạn theo chương trình. API cho phép nhà phát triển tự động hóa các hành động trên nền tảng của bạn và tích hợp dịch vụ của bạn với các ứng dụng hoặc dịch vụ trực tuyến khác. Bạn sẽ xây dựng một API đầy đủ tính năng cho nền tảng học tập trực tuyến của mình.

* Xây dựng API RESTful

Khi xây dựng một API, có một số cách bạn có thể cấu trúc các điểm cuối và hành động của nó, nhưng việc tuân theo các nguyên tắc REST được khuyến khích. Kiến trúc REST xuất phát từ Chuyển giao trạng thái đại diện.

API RESTful dựa trên tài nguyên; mô hình của bạn đại diện cho tài nguyên và các phương thức HTTP như GET, POST, PUT hoặc DELETE được sử dụng để truy xuất, tạo, cập nhật hoặc xóa đối tượng. Mã phản hồi HTTP cũng được sử dụng trong ngữ cảnh này. Các mã phản hồi HTTP khác nhau được trả về để cho biết kết quả của yêu cầu HTTP, ví dụ: mã phản hồi 2XX cho thành công, 4XX cho lỗi, v.v.

* Cài đặt khung Django REST

Khung Django REST cho phép bạn dễ dàng xây dựng API RESTful cho dự án của mình. Bạn có thể tìm thấy tất cả thông tin về khung REST tại <https://www.django-rest-framework.org/>.

* Hiểu trình phân tích cú pháp và trình kết xuất

Dữ liệu được tuần tự hóa phải được hiển thị ở định dạng cụ thể trước khi bạn trả về dữ liệu đó trong phản hồi HTTP. Tương tự như vậy, khi bạn nhận được một yêu cầu HTTP, bạn phải phân tích cú pháp dữ liệu đến và giải tuần tự hóa nó trước khi có thể thao tác với nó. Khung REST bao gồm trình kết xuất và trình phân tích cú pháp để xử lý việc đó.

* Thêm quyền vào chế độ xem

Khung REST bao gồm một hệ thống cấp phép để hạn chế quyền truy cập vào các chế độ xem. Một số nhiệm vụ tích hợp của khung REST là:

* AllowAny: Quyền truy cập không hạn chế, bất kể người dùng có được xác thực hay không.
* IsAuthenticated: Chỉ cho phép truy cập đối với người dùng đã được xác thực.
* IsAuthenticatedOrReadOnly: Truy cập đầy đủ vào người dùng đã được xác thực. Người dùng ẩn danh chỉ được phép thực thi các phương thức đọc như GET, HEAD hoặc OPTIONS..
* Tạo ViewSets và bộ định tuyến

ViewSets cho phép bạn xác định các tương tác của API và để khung REST xây dựng các URL một cách linh hoạt với đối tượng Bộ định tuyến . Bằng cách sử dụng ViewSets, bạn có thể tránh lặp lại logic cho nhiều chế độ xem.

ViewSets bao gồm các hành động cho các hoạt động tiêu chuẩn sau:

* Tạo thao tác: create()
* Thao tác truy xuất: list() và get()
* Thêm hành động bổ sung vào ViewSets

Bạn có thể thêm các hành động bổ sung vào ViewSets. Hãy thay đổi chế độ xem CourseEnrollView trước đó của bạn thành hành động ViewSet tùy chỉnh . Chỉnh sửa tệp api/views.py và sửa đổi lớp CourseViewSet để trông như sau:

1. Bạn sử dụng trình trang trí hành động của khung với tham số chi tiết=True để chỉ định rằng đây là một hành động được thực hiện trên một đối tượng duy nhất.
2. Trình trang trí cho phép bạn thêm các thuộc tính tùy chỉnh cho hành động. Bạn chỉ định rằng chỉ phương thức post() mới được phép cho chế độ xem này và đặt các lớp xác thực và quyền.
3. Bạn sử dụng self.get\_object() để truy xuất đối tượng Course.
4. Bạn thêm người dùng hiện tại vào mối quan hệ nhiều-nhiều của học sinh và trả về một tùy chỉnh phản hồi thành công.

* Tạo quyền tùy chỉnh

Bạn muốn sinh viên có thể truy cập vào nội dung của các khóa học mà họ đã đăng ký. Chỉ những sinh viên đã đăng ký vào một khóa học mới có thể truy cập nội dung của nó. Cách tốt nhất để làm điều này là sử dụng lớp quyền tùy chỉnh.

**Chương 16 : Xây dựng máy chủ trò chuyện**

* Tạo ứng dụng trò chuyện

Bằng cách sử dụng Kênh, bạn có thể dễ dàng triển khai các chức năng thời gian thực hoặc không đồng bộ vào dự án của mình ngoài các chế độ xem đồng bộ HTTP tiêu chuẩn. Bạn sẽ bắt đầu bằng cách thêm một ứng dụng mới vào dự án của mình. Ứng dụng mới sẽ chứa logic cho máy chủ trò chuyện.

Bạn có thể xem tài liệu về Kênh Django tại <https://channels.readthedocs.io/>.

* Triển khai chế độ xem phòng trò chuyện

Bạn sẽ cung cấp cho sinh viên một phòng trò chuyện khác nhau cho mỗi khóa học. Bạn cần tạo chế độ xem để học viên tham gia phòng trò chuyện của một khóa học nhất định. Chỉ những sinh viên đã đăng ký khóa học mới có thể truy cập vào phòng trò chuyện của khóa học.

* Kích hoạt lớp kênh

Các lớp kênh cho phép bạn giao tiếp giữa các phiên bản khác nhau của ứng dụng. Lớp kênh là cơ chế vận chuyển cho phép nhiều phiên bản người tiêu dùng giao tiếp với nhau và với các phần khác của Django.

Trong máy chủ trò chuyện của mình, bạn dự định có nhiều phiên bản của ứng dụng tiêu dùng ChatConsumer cho cùng một phòng trò chuyện của khóa học. Mỗi sinh viên tham gia phòng trò chuyện sẽ khởi tạo ứng dụng khách WebSocket trong trình duyệt của họ và điều đó sẽ mở ra một kết nối với một phiên bản của người tiêu dùng WebSocket. Bạn cần một lớp kênh chung để phân phối tin nhắn giữa những người tiêu dùng.

* Cập nhật người tiêu dùng để phát tin nhắn

Hãy chỉnh sửa người tiêu dùng ChatConsumer để sử dụng lớp kênh. Bạn sẽ sử dụng một nhóm kênh cho mỗi phòng trò chuyện của khóa học. Vì vậy, bạn sẽ sử dụng id khóa học để xây dựng tên nhóm. Các phiên bản ChatConsumer sẽ biết tên nhóm và có thể liên lạc với nhau.

**Chương 17 : Phát trực tiếp**

* Tạo môi trường sản xuất

Đã đến lúc triển khai dự án Django của bạn trong môi trường sản xuất. Bạn sẽ bắt đầu bằng cách định cấu hình cài đặt Django cho nhiều môi trường và sau đó bạn sẽ thiết lập môi trường sản xuất.

* Quản lý cài đặt cho nhiều môi trường

Trong các dự án thực tế, bạn sẽ phải làm việc với nhiều môi trường. Bạn thường sẽ có ít nhất một môi trường cục bộ để phát triển và một môi trường sản xuất để phục vụ ứng dụng của bạn. Bạn cũng có thể có các môi trường khác, chẳng hạn như môi trường thử nghiệm hoặc dàn dựng.

Một số cài đặt dự án sẽ chung cho tất cả các môi trường, nhưng một số cài đặt khác sẽ dành riêng cho từng môi trường . Thông thường, bạn sẽ sử dụng tệp cơ sở xác định các cài đặt chung và tệp cài đặt cho mỗi môi trường sẽ ghi đè mọi cài đặt cần thiết và xác định các cài đặt bổ sung.

* Định cấu hình dịch vụ PostgreSQL

Trong suốt cuốn sách này, bạn chủ yếu sử dụng cơ sở dữ liệu SQLite. SQLite thiết lập đơn giản và nhanh chóng, nhưng đối với môi trường sản xuất, bạn sẽ cần một cơ sở dữ liệu mạnh hơn, chẳng hạn như PostgreSQL, MySQL hoặc Oracle. Bạn đã học cách cài đặt PostgreSQL trong Chương 3, Mở rộng ứng dụng Blog của bạn. Đối với môi trường sản xuất, thay vào đó, chúng tôi sẽ sử dụng hình ảnh Docker PostgreSQL. Bạn có thể tìm thông tin về hình ảnh Docker PostgreSQL chính thức tại <https://hub.docker.com/_/postgres>.

* Bảo mật trang web của bạn bằng SSL/TLS

Giao thức Bảo mật lớp vận chuyển (TLS) là tiêu chuẩn để phục vụ các trang web thông qua kết nối an toàn. Tiền thân của TLS là Lớp cổng bảo mật (SSL). Mặc dù SSL hiện không còn được dùng nữa nhưng trong nhiều thư viện và tài liệu trực tuyến, bạn sẽ tìm thấy các tham chiếu đến cả thuật ngữ TLS và SSL.

* Kiểm tra dự án của bạn để sản xuất

Django bao gồm khung kiểm tra hệ thống để xác thực dự án của bạn bất kỳ lúc nào. Khung kiểm tra sẽ kiểm tra các ứng dụng được cài đặt trong dự án Django của bạn và phát hiện các sự cố thường gặp. Việc kiểm tra được kích hoạt ngầm khi bạn chạy các lệnh quản lý như máy chủ và di chuyển. Tuy nhiên, bạn có thể kích hoạt kiểm tra một cách rõ ràng bằng lệnh quản lý kiểm tra.

* Sử dụng Daphne cho kênh Django

Trong Chương 16, Xây dựng máy chủ trò chuyện, bạn đã sử dụng Kênh Django để xây dựng máy chủ trò chuyện bằng WebSockets. uWSGI phù hợp để chạy Django hoặc bất kỳ ứng dụng WSGI nào khác, nhưng nó không hỗ trợ giao tiếp không đồng bộ bằng Giao diện cổng máy chủ không đồng bộ (ASGI) hoặc WebSockets. Để chạy Kênh trong quá trình sản xuất, bạn cần có máy chủ web ASGI có khả năng quản lý WebSockets.

Daphne là máy chủ HTTP, HTTP2 và WebSocket dành cho ASGI được phát triển để phục vụ Kênh. Bạn có thể chạy Daphne cùng với uWSGI để phục vụ cả ứng dụng ASGI và WSGI một cách hiệu quả. Bạn có thể tìm thêm thông tin về Daphne tại <https://github.com/django/daphne>.

* Tạo một phần mềm trung gian tùy chỉnh

Bạn đã biết cài đặt MIDDLEWARE , chứa phần mềm trung gian cho dự án của bạn. Bạn có thể coi nó như một hệ thống plugin cấp thấp, cho phép bạn triển khai các hook được thực thi trong quy trình yêu cầu/phản hồi. Mỗi phần mềm trung gian chịu trách nhiệm về một số hành động cụ thể sẽ được thực thi cho tất cả các yêu cầu hoặc phản hồi HTTP.

* Tạo phần mềm trung gian tên miền phụ

Khi nhận được yêu cầu HTTP, bạn thực hiện các tác vụ sau:

1. Bạn lấy tên máy chủ đang được sử dụng trong yêu cầu và chia nó thành nhiều phần. Ví dụ : nếu người dùng đang truy cập mycourse.educaproject.com, bạn tạo danh sách ['mycourse', 'educaproject', 'com'].
2. Bạn kiểm tra xem tên máy chủ có bao gồm tên miền phụ hay không bằng cách kiểm tra xem phần tách có tạo ra nhiều hơn hai phần tử hay không. Nếu tên máy chủ bao gồm một tên miền phụ và đây không phải là www, bạn hãy thử truy cập khóa học bằng slug được cung cấp trong tên miền phụ.
3. Nếu không tìm thấy khóa học, bạn sẽ đưa ra ngoại lệ HTTP 404 . Nếu không, bạn chuyển hướng trình duyệt tới URL chi tiết khóa học.

A computer screen shot of text

Description automatically generated

* Thực hiện các lệnh quản lý tùy chỉnh

Lệnh quản lý bao gồm một mô-đun Python chứa lớp Lệnh kế thừa từ django.core.management.base.BaseCommand hoặc một trong các lớp con của nó. Bạn có thể tạo các lệnh đơn giản hoặc làm cho chúng lấy các đối số vị trí và tùy chọn làm đầu vào.

Django tìm kiếm các lệnh quản lý trong thư mục management/commands/ cho mỗi ứng dụng đang hoạt động trong cài đặt INSTALLED\_APPS . Mỗi mô-đun được tìm thấy sẽ được đăng ký dưới dạng lệnh quản lý được đặt tên theo nó.