

Tutorial 01

TO DO IN CLASS- REMEMBER TO UPLOAD THE REPO LINK TO TEAMS

FIRST COMPLETE THE ENTIRE TUTORIAL (IGNORE THE ACTIVITES) – AT THE END, COMPLETE THE PROPOSED ACTIVITIES

- **Before starting (create a new Django project):**

Open your Terminal, and in a location of your choice (it's recommended to create a folder for each django project in this course), execute the following:

Execute in Terminal

```
cd.djangocourse  
mkdir && cd helloworld
```

Execute in Terminal

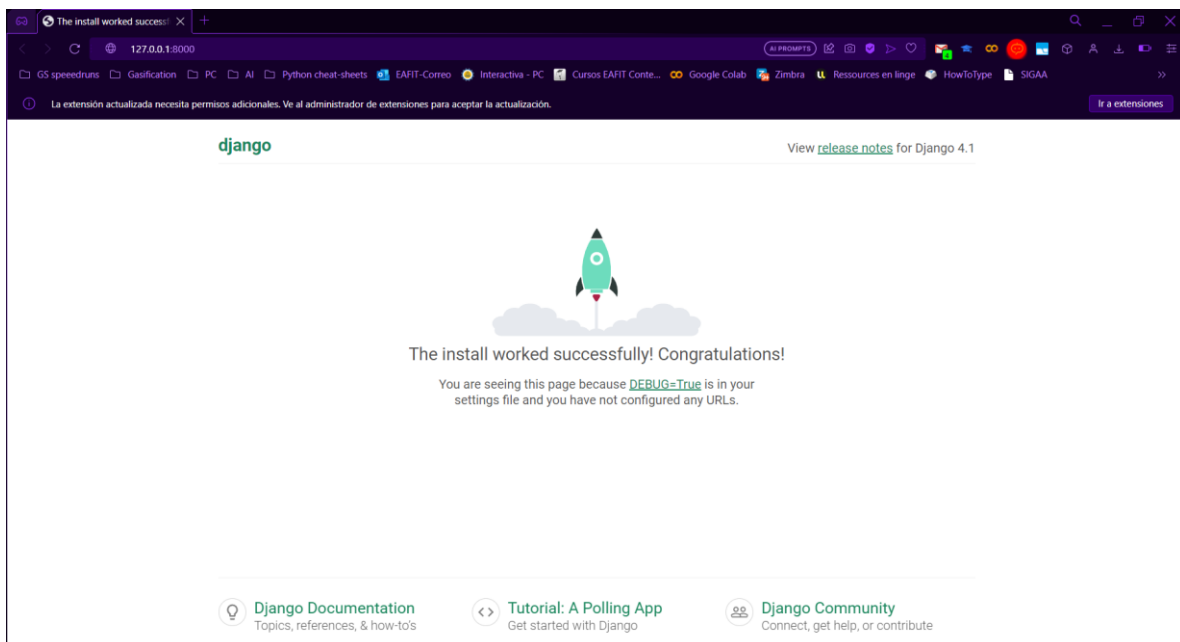
```
django-admin startproject helloworld_project .
```

Note: the dot (.) at the end of this command tells django to create the project in the current folder. The previous command creates a new django project inside the *djangolcourse/helloworld* folder. Next, in your Terminal, and run the application with the following:

Execute in Terminal

```
python manage.py runserver
```

If the installation and setup were successful, you could open the django development server link in your browser (<http://127.0.0.1:8000/>). You should see your django application (in django 4).



Django 4 default page.

You can press CTRL+C (on Windows, CMD+C on Mac) on your Terminal to stop running the server, as we are not going to run it from here in the future.

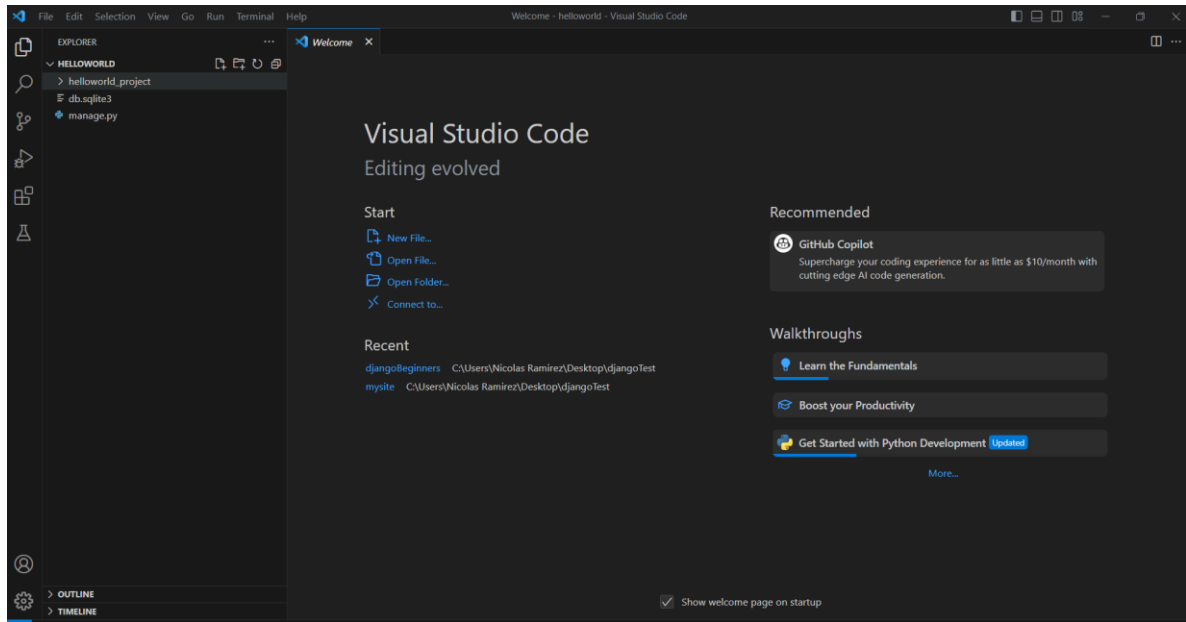
A. “Clean” hello world version

- In your Terminal run the following command:

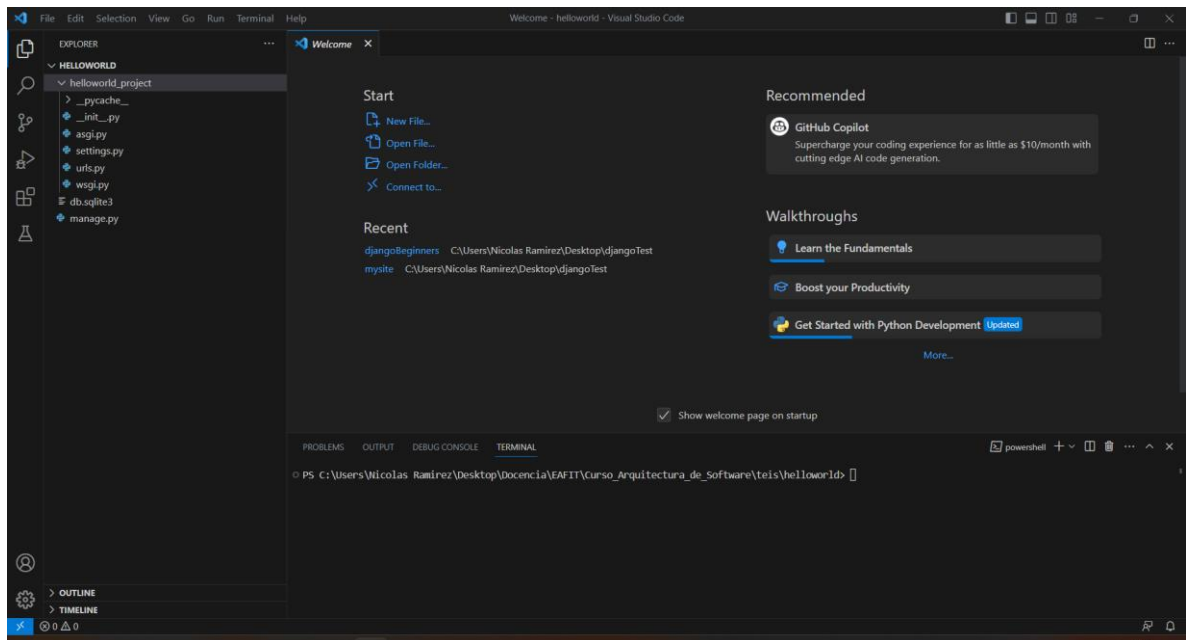
Execute in Terminal

- `code`

This command should open the VS Code editor with your project in the Explorer tab. If not, you can go to File->Open Folder -> select the helloworld folder.



Additionally, in the Terminal tab, click on New Terminal to open a Terminal in VS Code from which we will run terminal commands for this project.



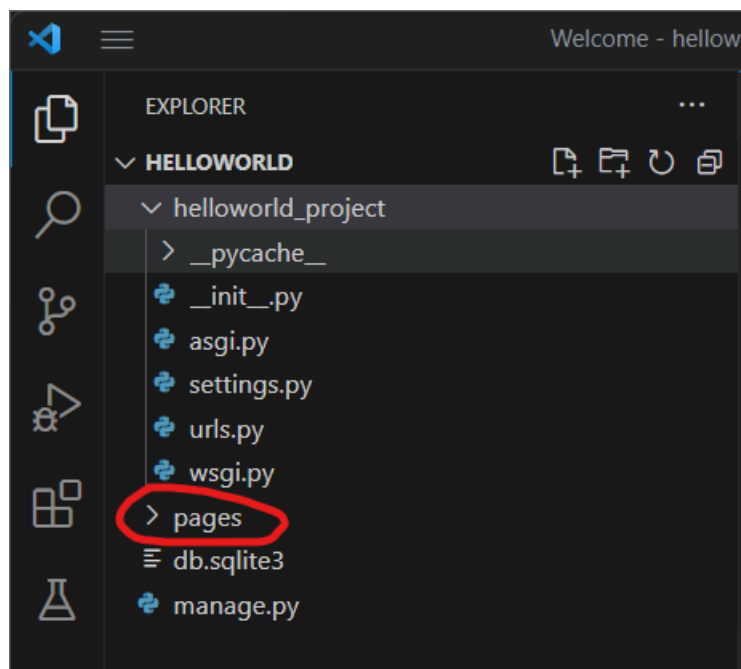
Create your first app

In your VS code Terminal run the following command:

Execute in VS Terminal

- `python manage.py startapp pages`

The previous command should add a new directory named *pages* in your project:



Go to *helloworld_project/settings.py* and add the following code in the `INSTALLED_APPS` list:

Add Bold Code

- # Application definition
-
- INSTALLED_APPS = [- 'django.contrib.admin', # here by default
- 'django.contrib.auth', # here by default
- 'django.contrib.contenttypes', # here by default
- 'django.contrib.sessions', # here by default
- 'django.contrib.messages', # here by default
- 'django.contrib.staticfiles', # here by default
- **'pages.apps.PagesConfig', # new**
-]

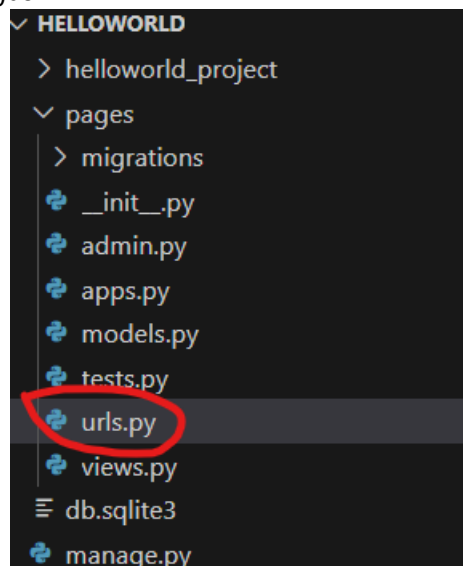
Controller/View

- Go to *pages/views.py* and add the following content to it (do not remove any content already there):

Add Bold Code

- from django.shortcuts import render # here by default
- **from django.http import HttpResponse # new**
-
- # Create your views here.
- **def homePageView(request): # new**
- **return HttpResponse('Hello World!') # new**

- Create a *urls.py* file in *pages*



- Go to *pages/urls.py* and add the following content to it:

Add Entire Code

```
• from django.urls import path
• from .views import homePageView
•
•
• urlpatterns = [
•     path("", homePageView, name='home')
• ]
```

- Go to *helloworld_project/urls.py* and add the following content to it:

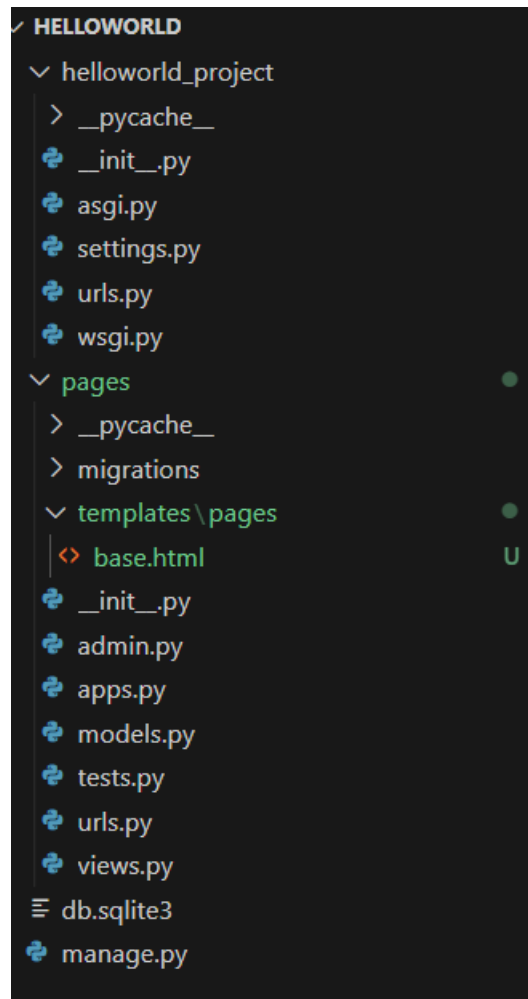
Add Bold Code

```
• from django.contrib import admin
• from django.urls import path, include
•
•
• urlpatterns = [
•     path('admin/', admin.site.urls),
•     path("", include('pages.urls')),
• ]
```

- **Note:** it is recommended that you create a Github repository for your projects and push changes as you make them.

Template/view

- Go to *pages* and create a *templates* directory; then, inside this new directory create a *pages* directory inside, then create a *base.html* file. The layout should look like this:



- Go to `pages/templates/pages/base.html` with the following content:

Add Entire Code

```
{% load static %}
<!doctype html>
<html lang="en">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet"
crossorigin="anonymous" />
  <link href="{% static 'pages/app.css' %}" rel="stylesheet" />
  <title>{% block title %}'Online Store'{% endblock %}</title>
</head>
<body>
  <!-- header -->
  <nav class="navbar navbar-expand-lg navbar-dark bg-secondary py-4">
    <div class="container">
      <a class="nav-bar-brand" href="#">Online Store</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup"
        aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
```

```

        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
        <div class="navbar-nav ms-auto">
            <a class="nav-link active" href="#">Home</a>
            <a class="nav-link active" href="#">About</a>
        </div>
    </div>
</div>
</nav>

<header class="masthead bg-primary text-white text-center py-4">
    <div class="container d-flex align-items-center flex-column">
        <h2>{% block header_title %}'A Laravel EAFIT App'{% endblock %}</h2>
    </div>
</header>
<!-- header -->

<div class="container my-4">
    {% block content %}
    {% endblock %}
</div>

<!-- footer -->
{% block footer %}
<div class="copyright py-4 text-center text-white">
    <div class="container">
        <small>
            Copyright - <a class="text-reset fw-bold text-decoration-none" target="_blank"
            href="https://twitter.com/danielgarax">
                Daniel Correa
            </a>
        </small>
    </div>
</div>
<!-- footer -->
{% endblock %}

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"
crossorigin="anonymous">
</script>
</body>
</html>

```

- Go to *templates/pages/home.html* file with the following content:

Add Entire Code

```

@extends('layouts.app')
{% extends 'pages/base.html' %}
{% block title %}'Home Page - Online Store'{% endblock %}
{% block content %}
<div class="text-center">
    Welcome to the application
</div>
{% endblock %}

```

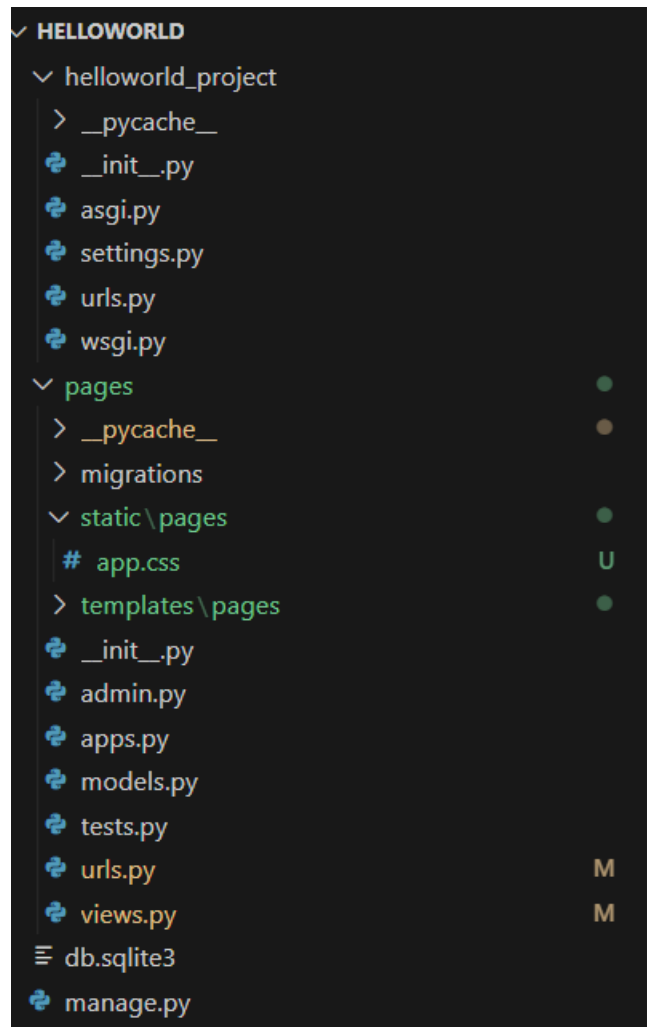
CSS

- Create a new directory *pages/static/pages*
- Go to *static/pages* and create a file *app.css* with the following content:

Add Entire Code

```
.bg-secondary {  
  background-color: #2c3e50 !important;  
}  
  
.copyright {  
  background-color: #1a252f;  
}  
  
.bg-primary {  
  background-color: #1abc9c !important;  
}  
  
nav{  
  font-weight: 700;  
}  
  
.img-card{  
  height: 18vw;  
  object-fit: cover;  
}
```

- The layout should look like this:



Update View

Note: in future projects you can create your Views here first rather than creating a function.

- In *pages/views.py*, make the following changes in **bold**.

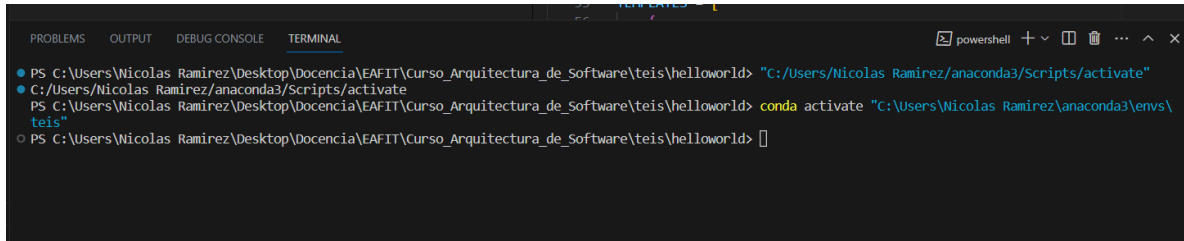
Modify Bold Code

- from django.shortcuts import render # here by default
- ~~from django.http import HttpResponse # new~~
- from django.views.generic import TemplateView
- # Create your views here.
- ~~def homePageView(request): # new~~
- ~~return HttpResponse("Hello World!") # new~~
- class HomePageView(TemplateView):

- `template_name = 'home.html'`

Run the application

- Open a VS Terminal by going to the *Terminal* tab and clicking *New Terminal*. The new Terminal should appear below your code :



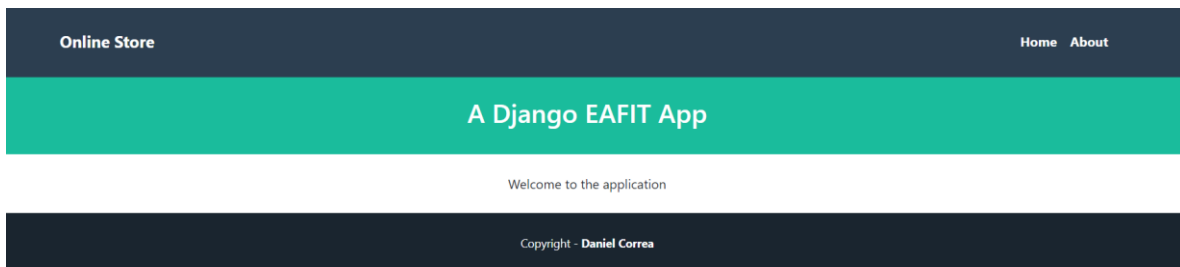
```
PS C:\Users\Nicolas Ramirez\Desktop\Docencia\EAFIT\Curso_Arquitectura_de_Software\teis\helloworld> "C:/Users/Nicolas Ramirez/anaconda3/Scripts/activate"
C:/Users/Nicolas Ramirez/anaconda3/Scripts/activate
PS C:\Users\Nicolas Ramirez\Desktop\Docencia\EAFIT\Curso_Arquitectura_de_Software\teis\helloworld> conda activate "C:/Users/Nicolas Ramirez/anaconda3/envs/teis"
PS C:\Users\Nicolas Ramirez\Desktop\Docencia\EAFIT\Curso_Arquitectura_de_Software\teis\helloworld> 
```

-

Execute in Terminal

```
python manage.py runserver
```

You should see the application running.



B. An About page

Views

- Go to `pages/templates/pages` and create a file `about.html` with the following content:

Add Entire Code

```
{% extends 'pages/base.html' %}

{% block title %} {{title}} {% endblock %}
{% block header_title %} {{subtitle}} {% endblock %}

{% block content %}
<div class="container">
  <div class="row">
    <div class="col-lg-4 ms-auto">
      <p class="lead">{{description}}</p>
    </div>
    <div class="col-lg-4 me-auto">
      <p class="lead">{{author}}</p>
    </div>
  </div>
</div>
{% endblock %}
```

Updating links in Header

- Now that we have the proper routes, controller, and views, let's include the links in the header. In `templates/pages/base.html`, make the following changes in **bold**.

Modify Bold Code

```
<!doctype html>
...
<body>
  <!-- header -->
  <nav class="navbar navbar-expand-lg navbar-dark bg-secondary py-4">
    <div class="container">
      <a class="navbar-brand" href="{% url 'home' %}">Online Store</a>
      <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNavAltMarkup"
        aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
        <div class="navbar-nav ms-auto">
          <a class="nav-link active" href="{% url 'home' %}">Home</a>
          <a class="nav-link active" href="{% url 'about' %}">About</a>
        </div>
      </div>
    </div>
  </nav>...
```

Routes

- In *pages/views.py*, make the following changes in **bold**.

Modify Bold Code

```
...
from django.shortcuts import render
from django.http import HttpResponse
from django.views.generic import TemplateView

# Create your views here.
class HomePageView(TemplateView):
    template_name = 'pages/home.html'

class AboutPageView(TemplateView):
    template_name = 'pages/about.html'

    def get_context_data(self, **kwargs):
        context = super().get_context_data(**kwargs)
        context.update({
            "title": "About us - Online Store",
            "subtitle": "About us",
            "description": "This is an about page ...",
            "author": "Developed by: Your Name",
        })

    return context
```

- Go to *pages/urls.py* and make the following changes in **bold**

Modify Bold Code

```
...
from django.urls import path
from .views import HomePageView, AboutPageView

urlpatterns = [
    path("", HomePageView.as_view(), name='home'),
    path('about/', AboutPageView.as_view(), name='about')
]
```

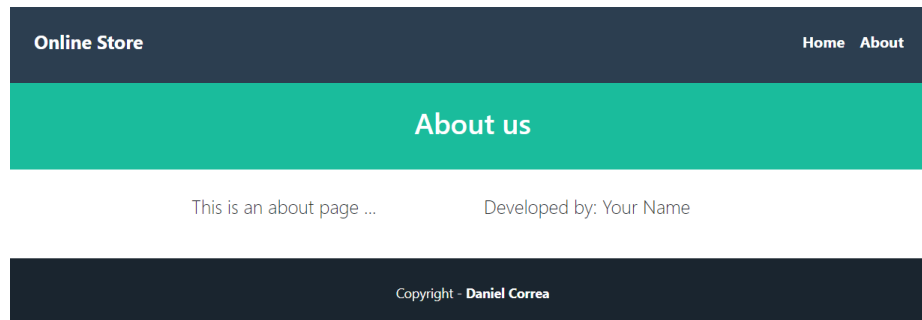
Run the application

- In the Terminal, execute the following:

Execute in Terminal

```
python manage.py runserver
```

Click the “about link” in the header, and you should see the application running.



Activity 1

- Create a “/contact” section in which you display the application email, address, and phone number. Use fake information.

C. Show products

View

- Go to *pages/views.py* and add the following content:

Add Entire Code

```
from django.views import View

class Product:
    products = [
        {"id": "1", "name": "TV", "description": "Best TV"},
        {"id": "2", "name": "iPhone", "description": "Best iPhone"},
        {"id": "3", "name": "Chromecast", "description": "Best Chromecast"},
        {"id": "4", "name": "Glasses", "description": "Best Glasses"}
    ]

class ProductIndexView(View):
    template_name = 'products/index.html'

    def get(self, request):
        viewData = {}
        viewData["title"] = "Products - Online Store"
        viewData["subtitle"] = "List of products"
        viewData["products"] = Product.products

        return render(request, self.template_name, viewData)

class ProductShowView(View):
    template_name = 'products/show.html'

    def get(self, request, id):
        viewData = {}
        product = Product.products[int(id)-1]
        viewData["title"] = product["name"] + " - Online Store"
        viewData["subtitle"] = product["name"] + " - Product information"
        viewData["product"] = product

        return render(request, self.template_name, viewData)
```

Product index template

In *pages/templates*, create a subfolder *products*. Then, in *pages/templates/products*, create a new file *index.html* and fill it with the following code.

Add Entire Code

```
@extends('layouts/app')
{% extends 'pages/base.html' %}
{% block title %} {{title}} {% endblock %}
{% block header_title %} {{subtitle}} {% endblock %}
```

```
{% block content %}
<div class="row">
  {% for product in products %}
  <div class="col-md-4 col-lg-3 mb-2">
    <div class="card">
      
      <div class="card-body text-center">
        <a href="{% url 'show' id=product.id %}"
          class="btn bg-primary text-white">{{product.name}}</a>
      </div>
    </div>
  </div>
  {% endfor %}
</div>
{% endblock %}
```

Product show view

In `pages/templates/products`, create a new file `show.html` and fill it with the following code.

Add Entire Code

```
{% extends 'pages/base.html' %}
{% block title %} {{title}} {% endblock %}
{% block header_title %} {{subtitle}} {% endblock %}

{% block content %}
<div class="card mb-3">
  <div class="row g-0">
    <div class="col-md-4">
      
    </div>
    <div class="col-md-8">
      <div class="card-body">
        <h5 class="card-title">
          {{product.name}}
        </h5>
        <p class="card-text">{{product.description}}</p>
      </div>
    </div>
  </div>
</div>
{% endblock %}
```

Modifying routes

In `pages/urls.py`, add the following changes in **bold**.

Modify Bold Code

```
...
urlpatterns = [
    path("", HomePageView.as_view(), name='home'),
    path('about/', AboutPageView.as_view(), name='about'),
    path('products/', ProductIndexView.as_view(), name='index'),
    path('products/<str:id>', ProductShowView.as_view(), name='show'),
]
```

```
] 
```

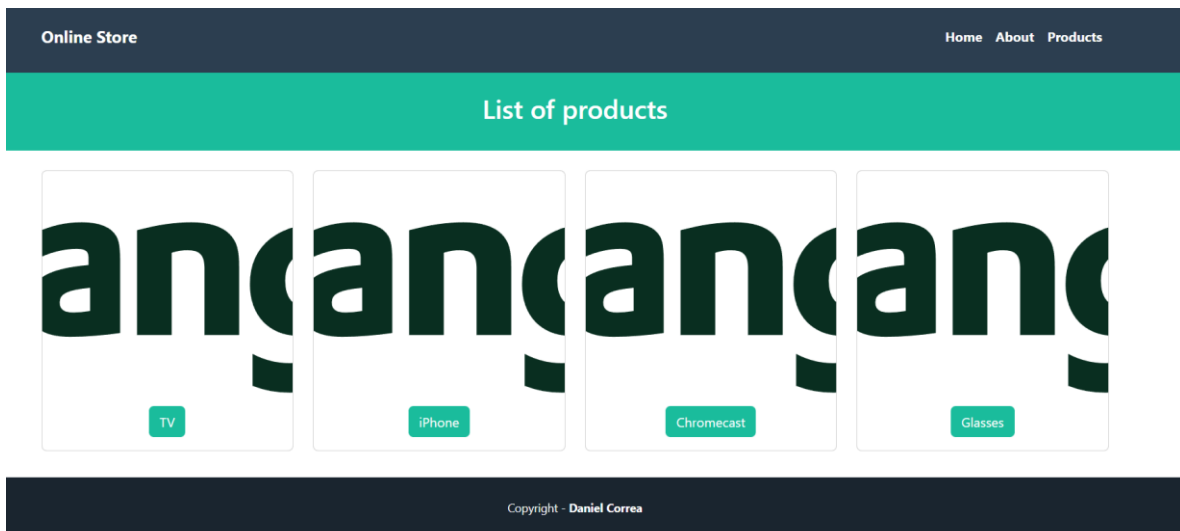
Run the application

In the Terminal execute the following:

```
python manage.py runserver
```

Execute in Terminal

Go to the (“/products”) route and you should see the application running.



You can also click a specific product to see its information.

Activity 2

- Add the (“/products”) route as a new menu option (in the header navbar).

Activity 3

- Add prices for each product and display the information in the product.show view.

Activity 4

- Modify the show method. If the product number entered by the URL is not valid, redirect the user to the home page (“home”) route: <https://docs.djangoproject.com/en/4.2/intro/tutorial04/> . Be careful, you should include this in the method response: `HttpResponseRedirect`;

Activity 5

- Add a conditional in the “show” template. If the price of a product is greater than 100, display the product name in red <https://docs.djangoproject.com/en/4.2/ref/templates/builtins/>

D. Product creation (simulation)

Views

- In `pages/views.py`, add the following content.

Modify Bold Code

```
...  
  
from django import forms  
from django.shortcuts import render, redirect  
  
class ProductForm(forms.Form):  
    name = forms.CharField(required=True)  
    price = forms.FloatField(required=True)  
  
class ProductCreateView(View):  
    template_name = 'products/create.html'  
  
    def get(self, request):  
        form = ProductForm()  
        viewData = {}  
        viewData["title"] = "Create product"  
        viewData["form"] = form  
        return render(request, self.template_name, viewData)  
  
    def post(self, request):  
        form = ProductForm(request.POST)  
        if form.is_valid():  
  
            return redirect(form)  
        else:  
            viewData = {}  
            viewData["title"] = "Create product"  
            viewData["form"] = form  
            return render(request, self.template_name, viewData)}
```

Routes

- In `pages/urls.py`, make the following changes in **bold**. (check the route order, it matters).

Modify Bold Code

```
...  
from django.urls import path  
from .views import HomePageView, AboutPageView, ProductIndexView, ProductShowView, ProductCreateView  
  
urlpatterns = [  
    path("", HomePageView.as_view(), name='home'),  
    path('about/', AboutPageView.as_view(), name='about'),  
    path('products/', ProductIndexView.as_view(), name='index'),  
    path('products/create', ProductCreateView.as_view(), name='form'),  
    path('products/<str:id>', ProductShowView.as_view(), name='show'),
```

Templates

- Go to `templates/products`, create a file `create.html` with the next content:

Add Entire Code

```
{% extends 'pages/base.html' %}
{% block title %} {{title}} {% endblock %}

{% block content %}
<div class="container">
  <div class="row justify-content-center">
    <div class="col-md-8">
      <div class="card">
        <div class="card-header">Create product</div>
        <div class="card-body">
          {% if form.errors %}
            <ul id="errors" class="alert alert-danger list-unstyled">
              {% for field, errors in form.errors.items %}
                {% for error in errors %}
                  <li>{{ error }}</li>
                {% endfor %}
              {% endfor %}
            </ul>
          {% endif %}

          <form method="POST" action="{% url 'form' %}">
            {% csrf_token %}
            <input type="text" class="form-control mb-2" placeholder="Enter name" name="name" value="{{form.data.name}}"/>
            <input type="text" class="form-control mb-2" placeholder="Enter price" name="price" value="{{form.data.price}}"/>
            <input type="submit" class="btn btn-primary" value="Send" />
          </form>
        </div>
      </div>
    </div>
  </div>
</div>
{% endblock %}
```

Run the application

In the Terminal, go to the project directory, and execute the following:

Execute in Terminal

```
python manage.py runserver
```

Go to the (`/products/create`) route and you should see the application running.

Online Store

HomeAboutProducts

A Django EAFIT App

Create product

Enter name

Enter price

Send

Copyright - Daniel Correa

Activity 6

- Try to understand the previous code. Add a new product but leave the name empty (and click send). Then, leave the price empty. Then, enter the two fields.

Activity 7

- Modify the previous code to only allow numbers greater than zero for the prices.
<https://docs.djangoproject.com/en/4.2/ref/forms/validation/> HINT: use `clean_<field_name>` methods.

Activity 8

- If the info entered by the form is valid. Then display a message saying, "Product created".
Note: create a new template (which uses the lbase template).

Activity 9

- Add a new menu option in the header (base.html), that links to the `"/products/create"` page.

Activity 10

- Create a Github repo with all this code (called DjangoTutorials) and upload the repo link to the corresponding Teams Assignment.