

## الفصل العاشر: Bootstrap

يتطلب تطوير الويب الكثير من المهارات. ليس عليك فقط برمجة موقع الويب ليعمل بشكل صحيح ، بل يتوقع المستخدمون أن يبدو جيدًا أيضًا. عند إنشاء كل شيء من البداية ، قد يكون من الصعب أيضًا إضافة كل HTML / CSS الضروري لموقع جميل. في حين أنه من الممكن كتابة جميع CSS وجافا سكريبت المطلوبة يدويًا لموقع ويب ذو مظهر حديث ، إلا أن معظم المطورين يستخدمون في الواقع إطار عمل مثل Bootstrap أو TailwindCSS . بالنسبة لمشروعنا ، سنستخدم Bootstrap الذي يمكن تمديده وتخصيصه حسب الحاجة.

### Pages App

في الفصل السابق ، قمنا بعرض صفحتنا الرئيسية من خلال تضمين منطق العرض في ملف `urls.py` الخاص بنا. على الرغم من أن هذا الأسلوب يعمل ، إلا أنه يبدو لي نوعًا ما اختراقًا ومن المؤكد أنه لا يتسع مع نمو موقع الويب بمرور الوقت. من المحتمل أيضًا أن يكون محيرًا إلى حد ما بالنسبة للوافدين الجدد على Django. بدلاً من ذلك ، يمكننا ويجب علينا إنشاء تطبيق صفحات مخصص لجميع صفحاتنا الثابتة ، مثل الصفحة الرئيسية ، ومستقبل حول الصفحة ، وما إلى ذلك. سيؤدي هذا إلى الحفاظ على رمزنا لطيفًا ومنظمًا للمضي قدمًا. في سطر الأوامر ، استخدم الأمر `startapp` لإنشاء تطبيق الصفحات الجديد. إذا كان الخادم لا يزال قيد التشغيل ، فقد تحتاج إلى كتابة `Control + c` أولاً لإنهاءه.

### Command Line

**(news) \$ python manage.py startapp pages**

ثم قم على الفور بتحديث ملف `config / settings.py` الخاص بنا. غالبًا ما أنسى القيام بذلك ، لذا فمن الجيد التفكير في إنشاء تطبيق جديد كعملية من خطوتين: قم بتشغيل الأمر `startapp` ثم تحديث `INSTALLED_APPS`.

### Code

```
# config/settings.py
INSTALLED_APPS = [
'django.contrib.admin',
'django.contrib.auth',
'django.contrib.contenttypes',
'django.contrib.sessions',
'django.contrib.messages',
'django.contrib.staticfiles',
'accounts',
'pages', # new
]
```

يمكننا الآن تحديث ملف `urls.py` الخاص بنا داخل دليل التكوين عن طريق إضافة تطبيق الصفحات ، وإزالة استيراد `TemplateView` ، وإزالة مسار `URL` السابق للصفحة الرئيسية القديمة.

#### Code

```
# config/urls.py

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

حان الوقت لإضافة صفحتنا الرئيسية مما يعني أن عناوين `urls/views/templates` لـ Django ترقص. سنبدأ بملف `pages` / `urls.py`. أولاً قم بإنشائه

#### Command Line

```
(news) $ touch pages/urls.py
```

ثم قم باستيراد `views` التي لم يتم إنشاؤها بعد ، وقم بتعيين مسارات المسار ، وتأكد من تسمية كل عنوان `url` أيضاً.

#### Code

```
# pages/urls.py

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

يجب أن يبدو رمز `views.py` مألوفاً في هذه المرحلة. نحن نستخدم طريقة العرض العامة المستندة إلى فئة `TemplateView` من Django مما يعني أننا نحتاج فقط إلى تحديد اسم القالب الخاص بنا لاستخدامه.

## Code

```
# pages/views.py
```

```
from django.views.generic import TemplateView
```

```
class HomePageView(TemplateView):
```

```
    template_name = 'home.html'
```

لدينا بالفعل نموذج home.html موجود. دعنا نؤكد أنه لا يزال يعمل كما هو متوقع من خلال عنوان URL الجديد والعرض. ابدأ تشغيل الخادم المحلي python manager.py runserver وانتقل إلى الصفحة الرئيسية على `/http://127.0.0.1:8000` لتأكيد أنه لم يتغير.

يجب أن يظهر اسم حساب المستخدم المتميز الذي تم تسجيل دخوله والذي استخدمناه في نهاية الفصل الأخير. ومن المثير للاهتمام ، نظرًا لأننا لم نقوم بإضافة حقن عمر لمستخدمنا المتميز ، فإن Django يتخلف عن عرض لا شيء.

## Bootstrap

إذا لم تكن قد استخدمت Bootstrap مطلقًا قبل أن تحصل على متعة حقيقية. مثل Django ، فإنه ينجز الكثير في القليل من التعليمات البرمجية. هناك طريقتان لإضافة Bootstrap إلى مشروع: يمكنك تنزيل جميع الملفات وتقديمها محليًا أو الاعتماد على شبكة توصيل المحتوى (CDN). الطريقة الثانية أسهل في التنفيذ بشرط أن يكون لديك اتصال إنترنت ثابت ، وهذا ما سنستخدمه هنا. يمكنك رؤية الأسطر الأربعة لإضافتها في مقدمة الصفحة Bootstrap: ( استبدل صفحة base.html ) بالكود التالي:

- Bootstrap.css
- jQuery.js
- Popper.js
- Bootstrap.js

## Code

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>{% block title %}Newspaper App{% endblock title %}</title>
```

```
<meta name="viewport" content="width=device-width,  
initial-scale=1, shrink-to-fit=no">
```

```
<!-- Bootstrap CSS -->
```

```
<link rel="stylesheet"
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/css/bootstrap.min.css"
```

```
integrity="sha384-zCbKRCUGaJDkqS1kPbPd7TveP5iyJE0EjAuZQTgFLD2ylzuqKfdKlfG/eSrtx
Ukn" crossorigin="anonymous">
```

```
</head>
```

```
<body>
```

```
<main>
```

```
{% block content %}
```

```
{% endblock content %}
```

```
</main>
```

```
<!-- Optional JavaScript -->
```

```
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"
```

```
integrity="sha512-894YE6QWD5I59HgZOGReFYm4dnWc1Qt5NtvYSaNcOP+u1T9qYdvdihz0
```

```
PPSiqn+/3e7Jo4EaG7TubfWGUrMQ==" crossorigin="anonymous"
```

```
referrerpolicy="no-referrer"></script>
```

```
<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/2.11.4/umd/popper.min.js"
```

```
integrity="sha512-+Tn2V/oN9O/kiaJg/1o5bETqyS35pMDJzkhkf8uvCzxmRox69AsWkSpBSME
```

```
GEE4EZp07Nunw712J3Xvh5Tti0w==" crossorigin="anonymous"
```

```
referrerpolicy="no-referrer"></script>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.1/dist/js/bootstrap.min.js"
```

```
integrity="sha384-VHvPCCyXqtD5DqJeNxl2dtTyhF78xXNXdkwX1CZeRusQfRKp+tA7hAShO
```

```
K/B/fQ2" crossorigin="anonymous"></script>
```

```
</body>
```

```
</html>
```

لا يتضمن مقتطف الشفرة هذا الارتباطات الكاملة لكل من Bootstrap CSS و JavaScript. يتم اختصارها. انسخ والصق  
الروابط الكاملة لـ Bootstrap 4.5 من مستند البدء السريع {docs\}. {docs\} إذا بدأت الخادم مرة أخرى باستخدام python

manager.py runserver وقم بتحديث الصفحة الرئيسية على http://127.0.0.1:8000/ ، فسترى أن حجم الخط قد تغير بالإضافة إلى ألوان الرابط.



#### Homepage with Bootstrap

Let's add a navigation bar at the top of the page which contains our links for the homepage, log in, log out, and sign up. Notably, we can use the `if/else` tags in the Django templating engine to add some basic logic. We want to show a "log in" and "sign up" button to users who are logged out, but a "log out" and "change password" button to users logged in.

دعنا نضيف شريط تنقل في الجزء العلوي من الصفحة يحتوي على روابطنا للصفحة الرئيسية ، وتسجيل الدخول ، وتسجيل الخروج ، والاشتراك. بشكل ملحوظ ، يمكننا استخدام علامات `if / else` في محرك قوالب Django لإضافة بعض المنطق الأساسي. نريد إظهار زر "تسجيل الدخول" و "التسجيل" للمستخدمين الذين قاموا بتسجيل الخروج ، ولكن زر "تسجيل الخروج" و "تغيير كلمة المرور" للمستخدمين الذين قاموا بتسجيل الدخول. مرة أخرى ، لا بأس من النسخ / اللصق هنا نظرًا لأن يركز هذا الكتاب على تعلم Django وليس HTML و CSS و Bootstrap. اصف وسم `:navbar`

#### Code

```
<!-- templates/base.html -->
...
<body>
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">
<a class="navbar-brand" href="{% url 'home' %}">Newspaper</a>
<button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarSupportedContent"
aria-controls="navbarSupportedContent" aria-expanded="false"
aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
{% if user.is_authenticated %}
<ul class="navbar-nav ml-auto">
<li class="nav-item">
```

```
<a class="nav-link dropdown-toggle" href="#"
id="navbarDropdown" role="button" data-toggle="dropdown"
aria-haspopup="true" aria-expanded="false">
{{ user.username }}
</a>

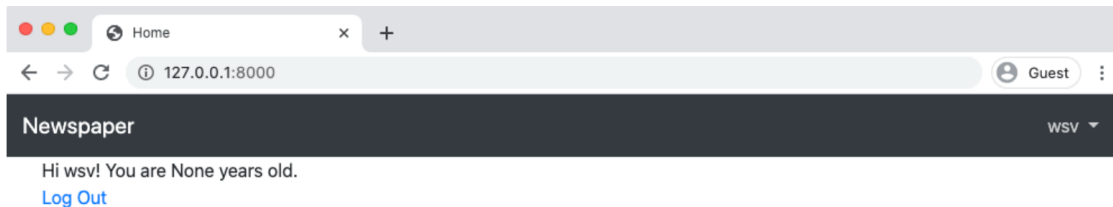
<div class="dropdown-menu dropdown-menu-lg-right"
aria-labelledby="navbarDropdown">
<a class="dropdown-item"
href="{% url 'password_change' %}">Change password</a>
<div class="dropdown-divider"></div>
<a class="dropdown-item" href="{% url 'logout' %}">
Log Out</a>
</div>
</li>
</ul>

{% else %}
<form class="form-inline ml-auto">
<a href="{% url 'login' %}" class="btn btn-outline-secondary">
Log In</a>
<a href="{% url 'signup' %}" class="btn btn-primary ml-2">
Sign up</a>
</form>
{% endif %}
</div>
</nav>

<main>
<div class="container">
{% block content %}
{% endblock content %}
</div>
</main>
```

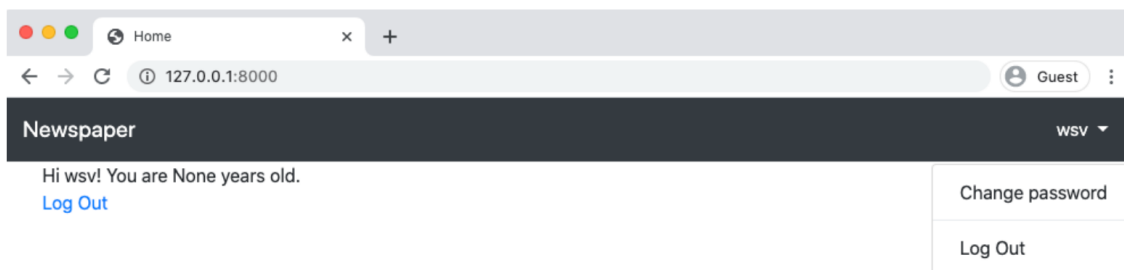
</body>

إذا قمت بتحديث الصفحة الرئيسية على <http://127.0.0.1:8000> فقد ظهر التنقل الجديد بطريقة سحرية! لقد أضفنا أيضًا عناصر div "الحاوية" حول المحتوى الخاص بنا ، وهي طريقة Bootstrap لتوفير حشوة جانبية تلقائية لموقع ما.

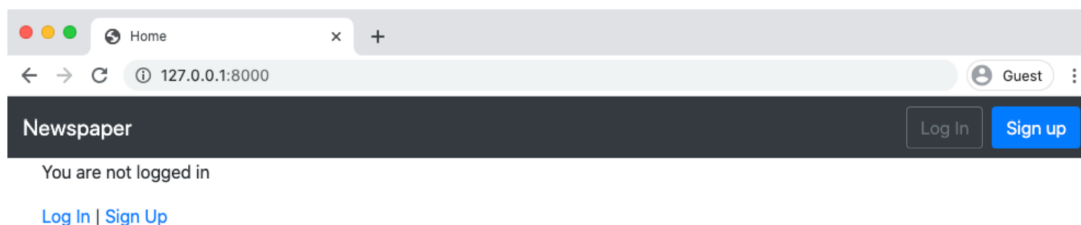


Homepage with Bootstrap nav logged in

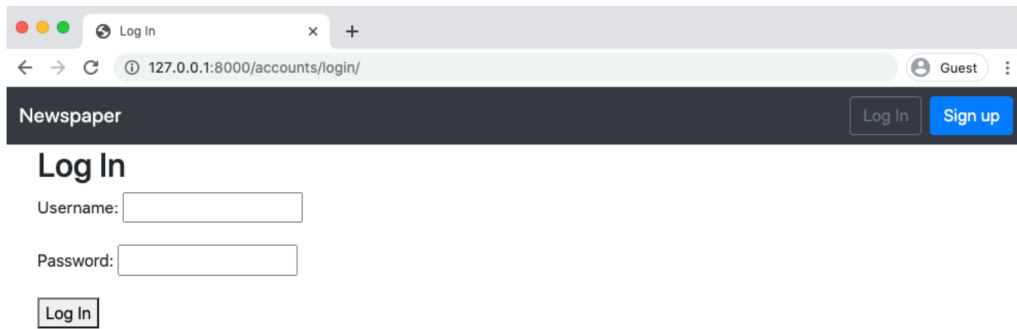
انقر فوق اسم المستخدم في الزاوية اليمنى العليا – في حالتي- لرؤية القائمة المنسدلة الرائعة التي يوفرها Bootstrap.



إذا قمت بالنقر فوق "تسجيل الخروج" ، يتغير شريط التنقل إلى روابط الأزرار إما لـ "تسجيل الدخول" أو "التسجيل".



والأفضل من ذلك ، إذا قمت بتقليل حجم نافذة المتصفح ، فإن Bootstrap يغير حجمه تلقائيًا ويقوم بإجراء تعديلات بحيث يبدو جيدًا على جهاز محمول أيضًا. يمكنك حتى تغيير عرض متصفح الويب لترى كيف تتغير الهوامش الجانبية مع زيادة حجم الشاشة وانخفاضه. إذا قمت بالنقر فوق الزر "تسجيل الخروج" ثم "تسجيل الدخول" من التنقل العلوي ، يمكنك أيضًا رؤية أن صفحة تسجيل الدخول <http://127.0.0.1:8000/accounts/login> تبدو أفضل أيضًا.



Log In

Username:

Password:

Log In

#### Bootstrap login

الشيء الوحيد الذي يلفت النظر هو زر "تسجيل الدخول" الخاص بنا. يمكننا استخدام Bootstrap لإضافة بعض التصميم الجميل مثل جعله أخضر وجذابًا. قم بتغيير سطر "الزر" في النموذج / login.html / Registration كما يلي.

#### Code

```
<!-- templates/registration/login.html -->
{% extends 'base.html' %}
{% block title %}Log In{% endblock title %}
{% block content %}
<h2>Log In</h2>
<form method="post">
{% csrf_token %}
{{ form.as_p }}
<button class="btn btn-success ml-2" type="submit">Log In</button>
</form>
{% endblock content %}
```

الآن قم بتحديث الصفحة لرؤية زرنا الجديد.



### Bootstrap log in with new button

## Sign Up Form

تحتوي صفحة التسجيل الخاصة بنا على `/http://127.0.0.1:8000/accounts/signup` على أساليب Bootstrap ولكن أيضاً نص مساعد مشتت للانتباه. على سبيل المثال بعد "اسم المستخدم" نقول "مطلوب. 150 حرفاً أو أقل.

Letters, digits and `@./+/_ only.`"

## Updated navbar logged out

من أين أتى هذا النص ، أليس كذلك؟ عندما يكون هناك شيء ما مثل "السحر" في Django كن مطمئناً أنه بالتأكيد ليس كذلك. من المحتمل أن الكود جاء من قطعة داخلية من Django. أسرع طريقة اكتشافها لمعرفة ما يحدث تحت الغطاء في Django هي ببساطة الانتقال إلى شفرة مصدر Django على Github ، واستخدام شريط البحث ومحاولة العثور على جزء معين من النص. على سبيل المثال ، إذا أجريت بحثاً عن "150 حرفاً أو أقل" ، فستجد نفسك في الصفحة الخاصة بـ `django / Contrib / auth` `Models.py`. حتى كتابة هذه السطور ، هذا هو `link141` والسطر المحدد موجود في السطر 334. يأتي النص كجزء من تطبيق `auth` ، في حقل اسم المستخدم لـ `AbstractUser`. لدينا ثلاثة خيارات الآن:

- تجاوز `help_text` الموجود

- إخفاء help\_text

- إعادة تصميم help\_text

سنختار الخيار الثالث لأنه طريقة جيدة لتقديم حزمة الجهات الخارجية الممتازة `django-crispy-Forms`. يُعد العمل باستخدام النماذج تحديًا ، كما أن أشكال `django-crispy` تجعل من السهل كتابة كود جاف (لا تكرر نفسك). أولاً ، قم بإيقاف الخادم المحلي باستخدام `Control + c`. ثم استخدم `Pipenv` لتنصيب الحزمة في مشروعنا.

## Command Line

```
(news) $ pip install django-crispy-forms
```

أضف التطبيق الجديد إلى قائمة `INSTALLED_APPS` في ملف `config / settings.py`. مع تزايد عدد التطبيقات ، أجد أنه من المفيد التمييز بين تطبيقات الجهات الخارجية والتطبيقات المحلية التي أضفتها بنفسني. هذا ما يبدو عليه الرمز الآن.

## Code

```
# config/settings.py
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    # 3rd Party
    'crispy_forms', # new
    # Local
    'accounts',
    'pages',
]
```

أضف إعدادًا جديدًا لـ `CRISPY_TEMPLATE_PACK` ، مضبوطًا على `bootstrap4` ، أسفل ملف `config / settings.py` الخاص بنا.

```
# config/settings.py
```

```
CRISPY_TEMPLATE_PACK = 'bootstrap4'
```

الآن في نموذج signup.html الخاص بنا ، يمكننا استخدام النماذج المقرمشة بسرعة. أولاً ، نحمل علامات `crispy_forms` في الأعلى ثم نستبدل `{% form.as_p %}` بـ `{% crispy %}`.

Code

```
<!-- templates/signup.html -->
{% extends 'base.html' %}
{% load crispy_forms_tags %}
{% block title %}Sign Up{% endblock title %}
{% block content %}
<h2>Sign Up</h2>
<form method="post">
{% csrf_token %}
{{ form|crispy }}
<button type="submit">Sign Up</button>
</form>
{% endblock content %}
```

إذا قمت بتشغيل الخادم مرة أخرى باستخدام `python manager.py runserver` وقم بتحديث صفحة التسجيل ، يمكننا رؤية التغييرات الجديدة.

Sign Up

Username\*

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Email address

Age

Password\*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation\*

Enter the same password as before, for verification.

Sign Up

أفضل بكثير. على الرغم من ماذا لو كان زر "التسجيل" أكثر جاذبية؟ ربما تجعلها خضراء؟ يحتوي Bootstrap على جميع أنواع خيارات تصميم الأزرار التي يمكننا الاختيار من بينها. دعونا نستخدم كلمة "نجاح" ذات الخلفية الخضراء والنص الأبيض. قم بتحديث ملف `signup.html` الموجود في السطر الخاص بزر التسجيل.

## Code

```
<!-- templates/registration/signup.html -->
{% extends 'base.html' %}
{% load crispy_forms_tags %}
{% block title %}Sign Up{% endblock title %}
{% block content %}
<h2>Sign Up</h2>
<form method="post">
{% csrf_token %}
{{ form|crispy }}
<button class="btn btn-success" type="submit">Sign Up</button>
</form>
{% endblock content %}
```

قم بتحديث الصفحة ويمكنك رؤية عملنا المحدث.

Sign Up

127.0.0.1:8000/accounts/signup/

Guest

Newspaper

Log InSign up

## Sign Up

Username\*

Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.

Email address

Age

Password\*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation\*

Enter the same password as before, for verification.

Sign Up

خاتمة بدأ تطبيق الصحف الخاص بنا في الظهور بشكل جيد. تتمثل الخطوة الأخيرة في تدفق مصادقة المستخدم في تكوين تغيير كلمة المرور وإعادة تعيينها. هنا مرة أخرى ، اعتنى Django بالحمل الثقيل بالنسبة لنا ، لذا فهو يتطلب قدرًا ضئيلاً من التعليمات البرمجية من جانبنا.