**Bootstrap**

Веб-разработка требует много навыков. Вам нужно не только программировать сайт для корректной работы, но также пользователи ожидают, что он будет хорошо выглядеть. Когда вы создаете все с нуля, может быть крайне тяжело добавить все необходимые HTML / CSS для красивого сайта.

К счастью, есть Bootstrap, самая популярная платформа для создания адаптивных проектов, ориентированных на мобильные устройства. Вместо того, чтобы писать все наши собственные CSS и JavaScript для общих функций макета веб-сайта, мы можем вместо этого положиться на Bootstrap, чтобы сделать эту тяжелую работу. Это означает, что с небольшим количеством кода с нашей стороны мы сможем быстро создать хорошо выглядящие сайты. И если мы хотим вносить изменения по мере развития проекта, то при необходимости Bootstrap легко переопределить.

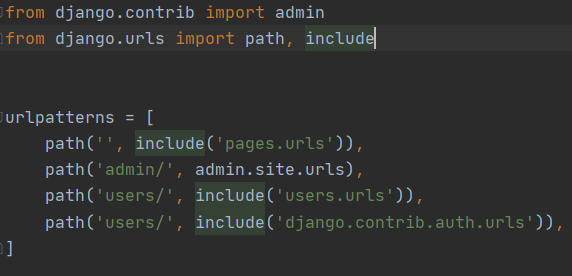
Если вы хотите сосредоточиться на функциональности проекта, а не на дизайне, Bootstrap - отличный выбор.

**Приложение Pages**

В предыдущей главе мы отображали нашу домашнюю страницу, включив логику просмотра в наш файл urls.py. Хотя этот подход работает, он кажется мне несколько хакерским, и, конечно, он не масштабируется, поскольку веб-сайт со временем растет. Это также, вероятно, несколько сбивает с толку новичков в Django. Вместо этого мы можем и должны создать специальное приложение для всех наших статических страниц. Это сделает наш код красивым и организованным далее. В командной строке используйте команду startapp для создания нашего нового приложения pages.

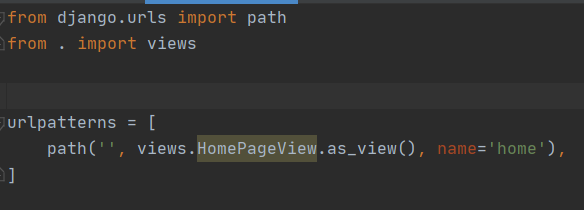
Затем сразу обновите наш файл settings.py.

Теперь мы можем обновить наш файл urls.py на уровне проекта. Идем дальше и удалим импорт TemplateView. Мы также обновим маршрут ' ', чтобы включить приложение pages. В итоговом виде файл должен выглядеть следующим образом:

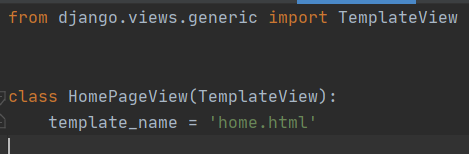


Пришло время добавить нашу домашнюю страницу, что означает стандартные urls/views/ templates. Начнем с файла pages/urls.py . Создайте его в папке приложения pages.

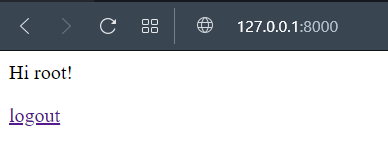
Затем импортируйте наши еще не созданные представления, задайте пути маршрута и убедитесь, что каждый url так же имеет имя(name) (Проще говоря мы перенесли url домашней страницы с уровня проекта на уровень приложения). Код должен выглядеть так:



Теперь отредактируем файл views.py в нашем новом приложении. На данном этапе код views.py должен выглядеть знакомым. Мы используем представление шаблонов на основе классов Django, что означает, что нам нужно только указать template\_name, чтобы использовать его.



У нас уже есть существующий шаблон home.html. Давайте проверим, что он по-прежнему работает должным образом с нашим новым url и представлением. Перейдите к странице http://127.0.0.1:8000/ что бы проверить что все остается неизменным.



**Bootstrap**

Если вы никогда не использовали Bootstrap, я надеюсь вы получите реальное удовольствие от его использования.

Существует два способа добавить Bootstrap в проект: можно загрузить все файлы и обслуживать их локально или использовать Сеть Доставки Контента (CDN). Второй подход проще реализовать, если у вас есть стабильное подключение к интернету, это мы и будем использовать здесь.

Bootstrap поставляется со стартовым шаблоном, который включает в себя основные необходимые файлы. В частности, есть четыре, которые мы подключаем:

* Bootstrap.css
* jQuery.js
* Popper.js
* Bootstrap.js

Обновите файл base.html. Вот как должен выглядеть обновленный файл base.html:



Ну и чтобы вам не пришлось так много писать, вот:

<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css"  
 integrity="sha384Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous"/>

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UENzmM7KCkRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"  
 crossorigin="anonymous"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.12.9/umd/popper.min.js"  
 integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>

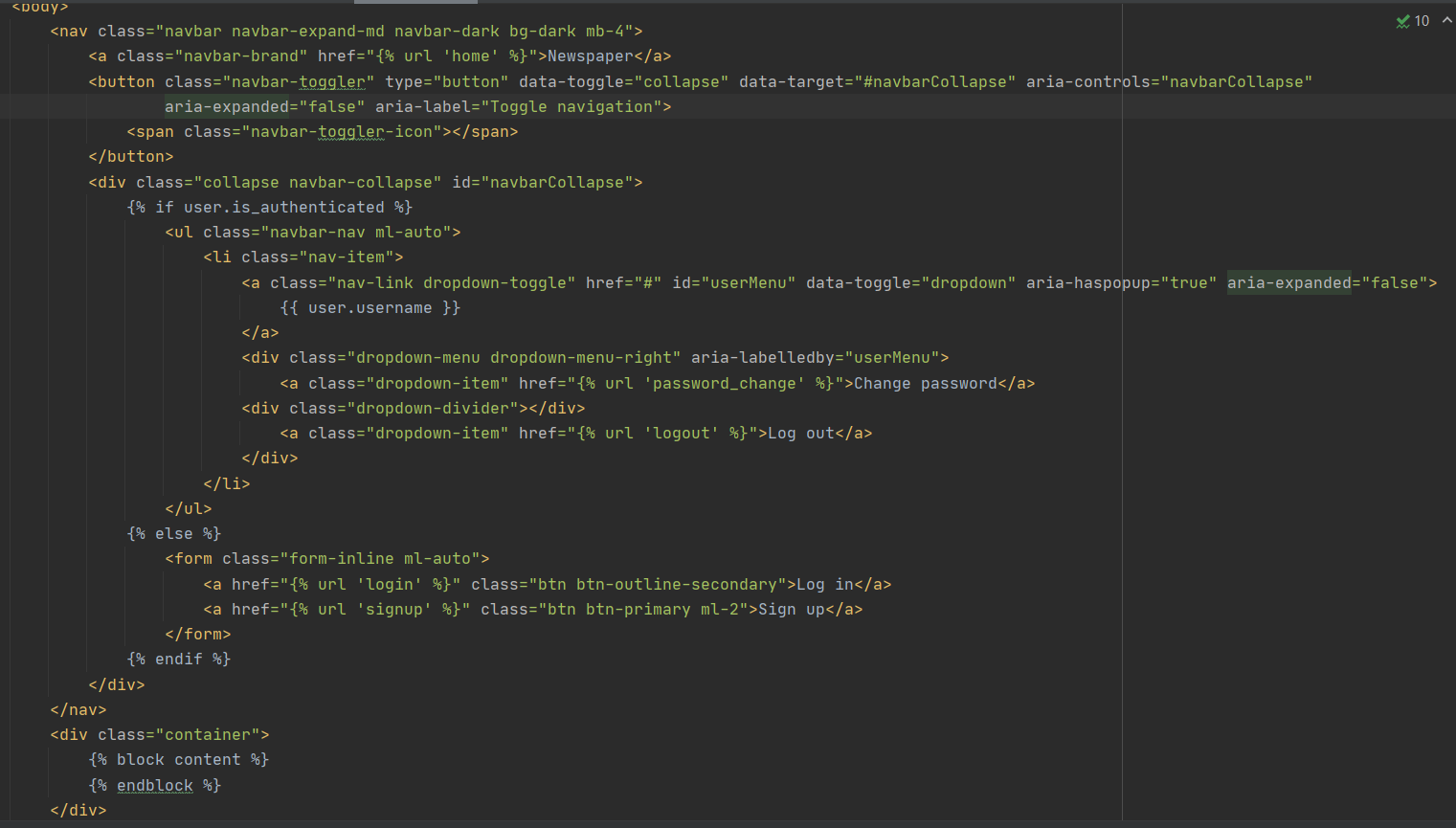
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/js/bootstrap.min.js"  
 integrity="sha384-JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl" crossorigin="anonymous"></script>

Если вы снова запустите сервер, вы увидите, что в данный момент практически ничего не изменилось.

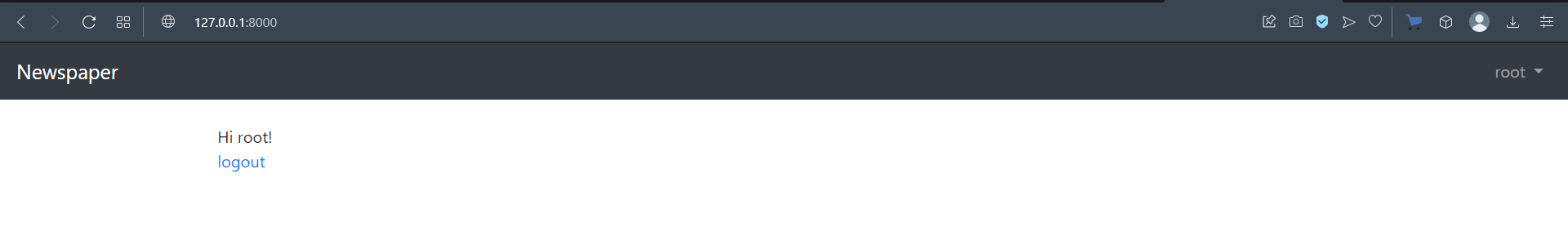
Давайте добавим панель навигации вверху страницы, которая содержит наши ссылки для домашней страницы, входа в систему, выхода из системы и регистрации. В частности, мы можем использовать теги if / else в шаблонизаторе Django для добавления некоторой базовой логики. Мы хотим показать кнопки «войти» и «зарегистрироваться» пользователям, которые вышли из системы, а кнопки «выйти» и «изменить пароль» пользователям, которые вошли в систему.

Вот как выглядит код:

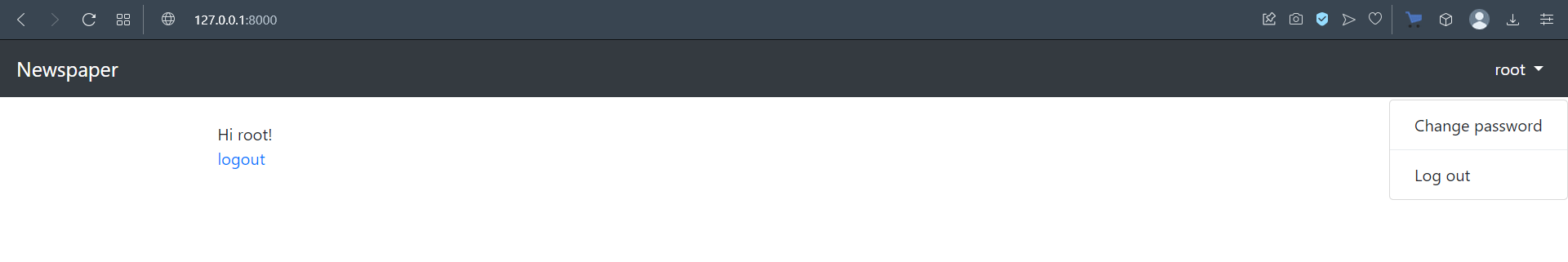




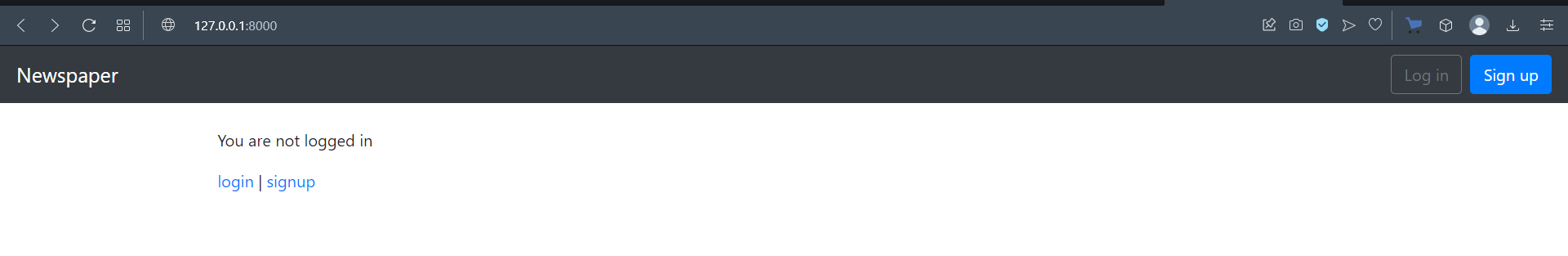
Если вы обновите домашнюю страницу на http://127.0.0.1:8000 наша новая навигационная панель появится волшебным образом!



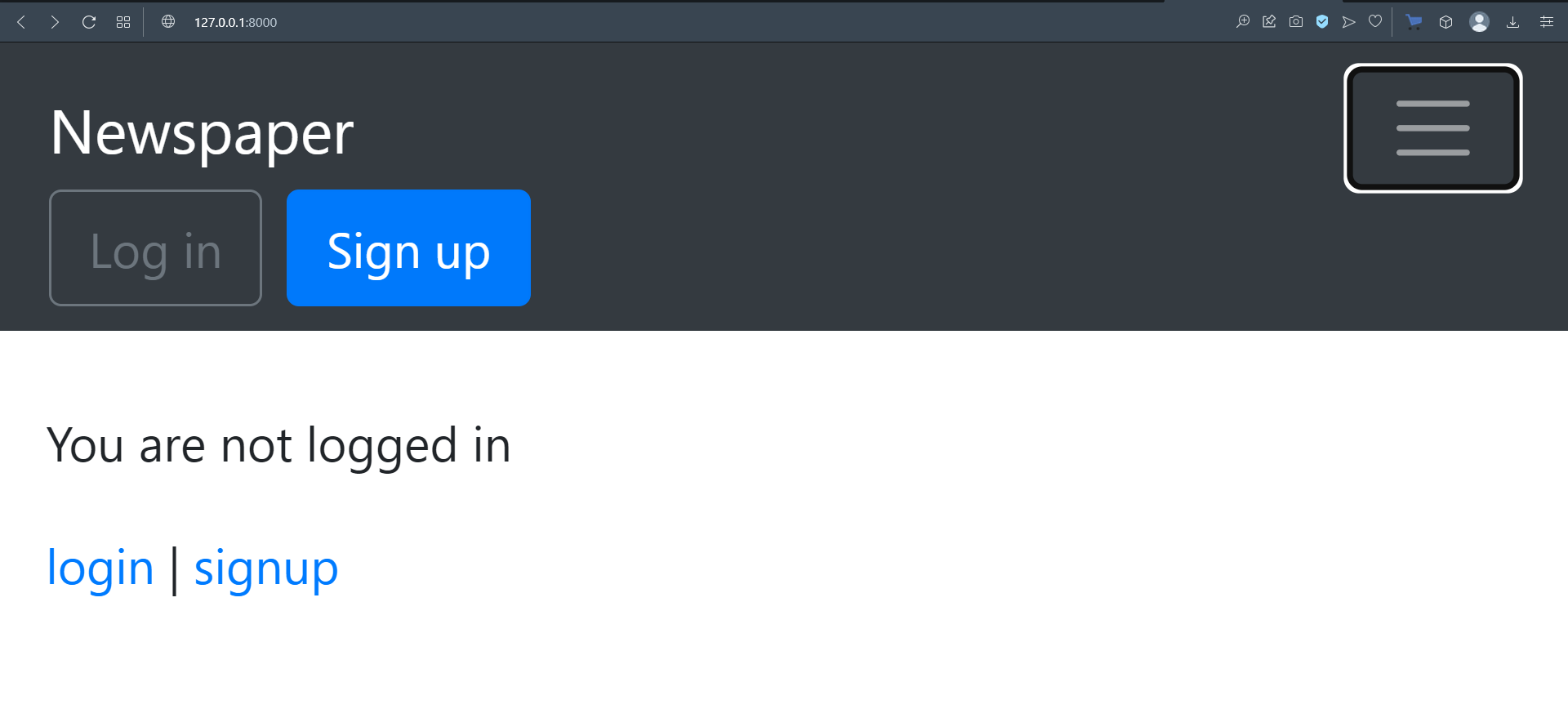
Нажмите на имя пользователя в верхнем правом углу - wsv в моем случае - чтобы увидеть хорошее выпадающее меню, которое предлагает Bootstrap.

****

Если вы нажмете на ссылку «logout», то наша навигационная панель изменится, предложив ссылки либо «log in», либо «sign up».

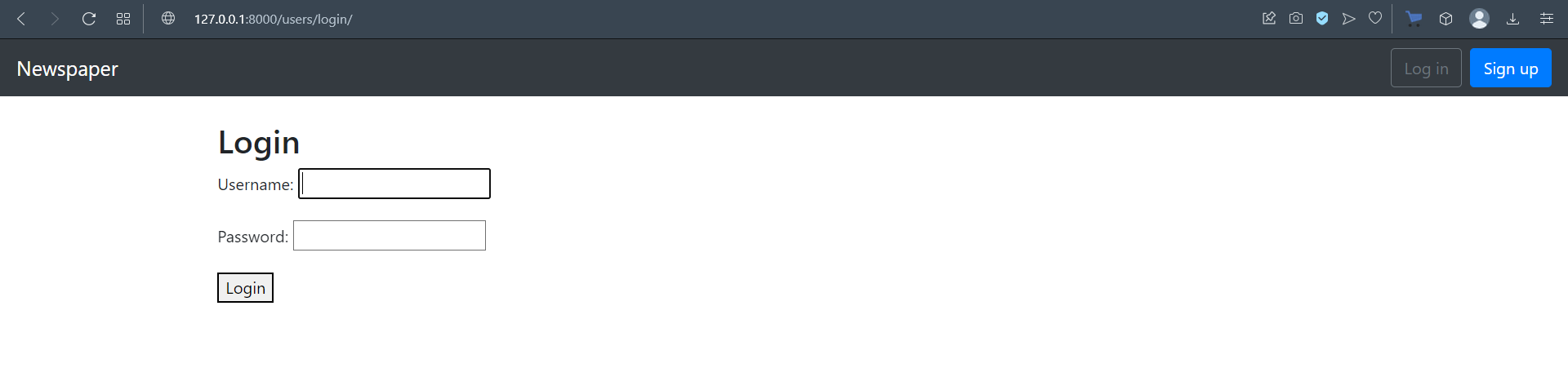


Еще лучше, если вы уменьшите размер окна вашего браузера. Bootstrap автоматически изменяет размеры и вносит изменения, чтобы он выглядел хорошо и на мобильном устройстве.



Обновите домашнюю страницу, и вы увидите ее в действии. Вы даже можете изменить ширину веб-браузера, чтобы увидеть, как изменяются боковые поля при увеличении и уменьшении размера экрана.

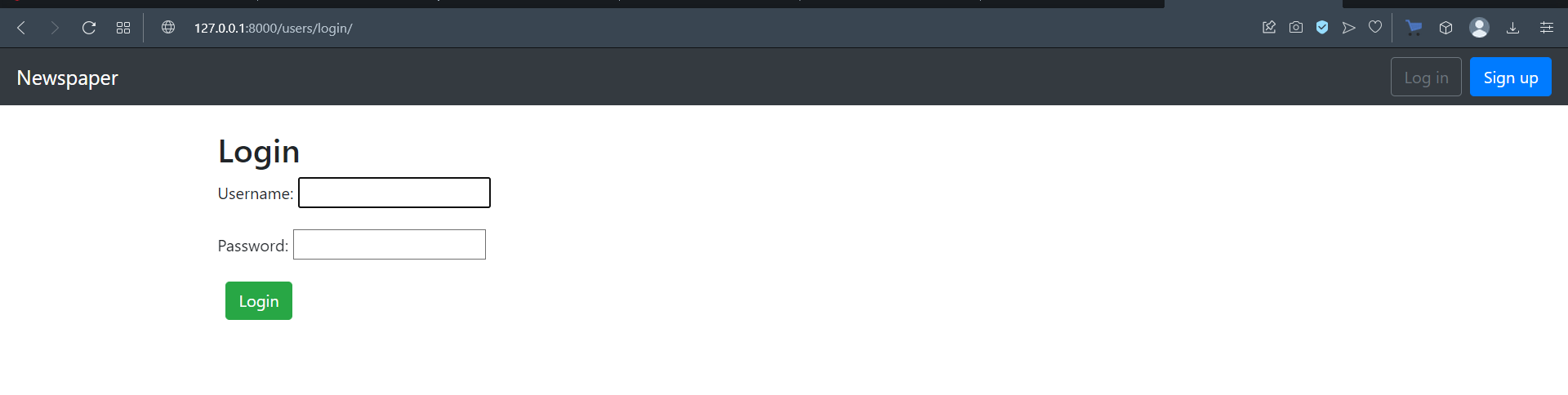
Если вы нажмете кнопку «logout», а затем «log in» в верхней панели навигации, вы также увидите, что наша страница входа в систему http://127.0.0.1:8000/users/login также выглядит лучше.



Единственное, что отвлекает - это кнопка «Login». Мы можем использовать Bootstrap, чтобы добавить приятный стиль, например, сделать его зеленым и привлекательным. Измените строку “button” в templates/registration/login.html следующим образом.

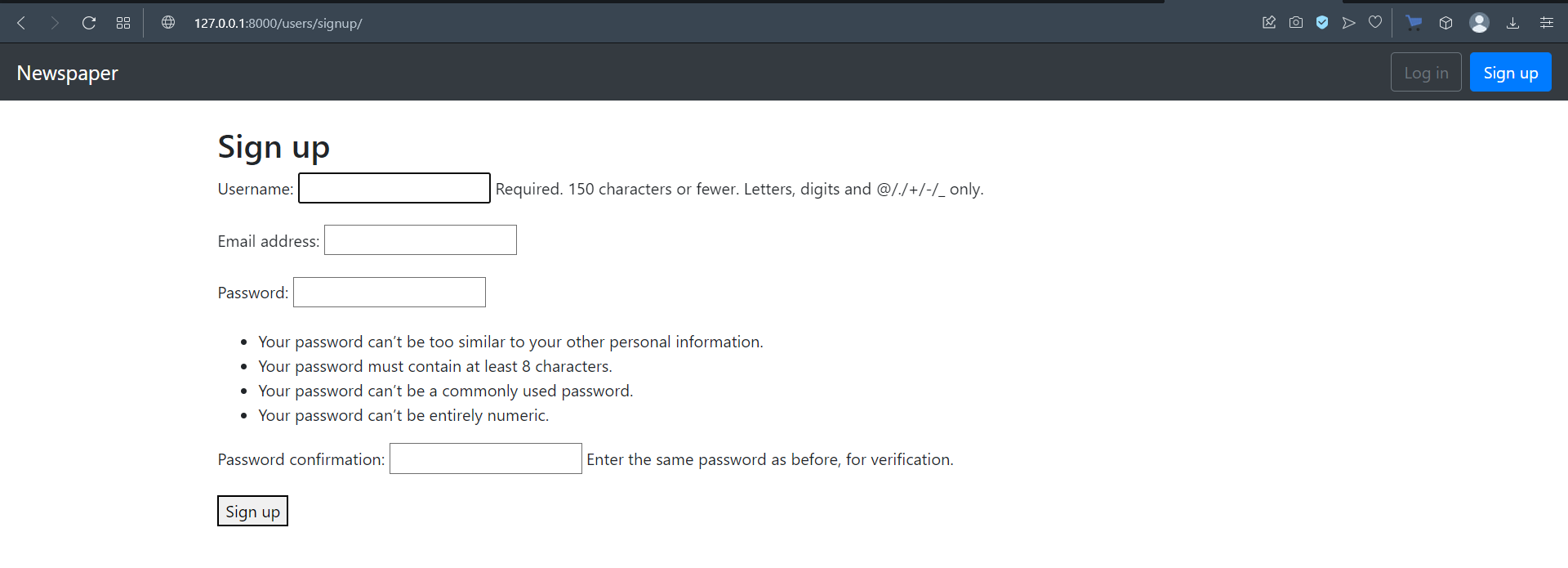


Теперь обновите страницу, чтобы увидеть нашу новую кнопку.



**Signup Form**

Наша страница регистрации по адресу http://127.0.0.1:8000/users/signup/ есть стили Bootstrap, а также отвлекающий вспомогательный текст. Например, после “Username” написано “Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.”



Откуда взялся этот текст? Всякий раз, когда что-то кажется «волшебным» в Django, будьте уверены, что это определенно не так. Вероятно, код пришел из внутреннего фрагмента Django.

Самый быстрый метод, который я нашел, чтобы выяснить, что происходит в Django, - просто перейти к исходному коду Django на Github, использовать панель поиска и попытаться найти конкретный фрагмент текста.

Например, если вы выполните поиск «150 characters or fewer», вы окажетесь на странице django / contrib / auth / models.py, расположенной, в строке 301.

Текст поставляется как часть приложения auth, в поле имени пользователя для AbstractUser. Сейчас у нас есть три варианта:

* переопределить существующий help\_text
* скрыть help\_text
* изменить стиль help\_text

Мы выберем третий вариант, так как это хороший способ представить отличный сторонний пакет django-crispy-forms.

Используйте Pip для установки пакета в нашем проекте.

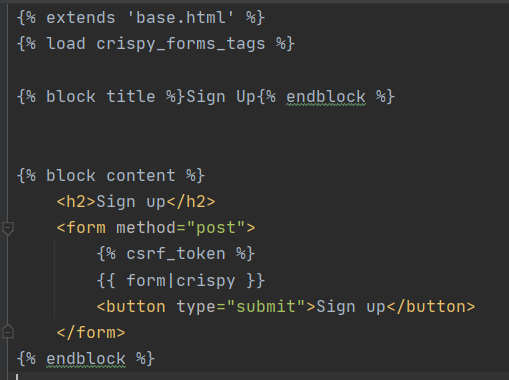
*pip install django-crispy-forms*

При загрузке пакета создается новое приложение с названием crispy\_forms. Добавьте его в список INSTALLED\_APPS в settings.py файл.

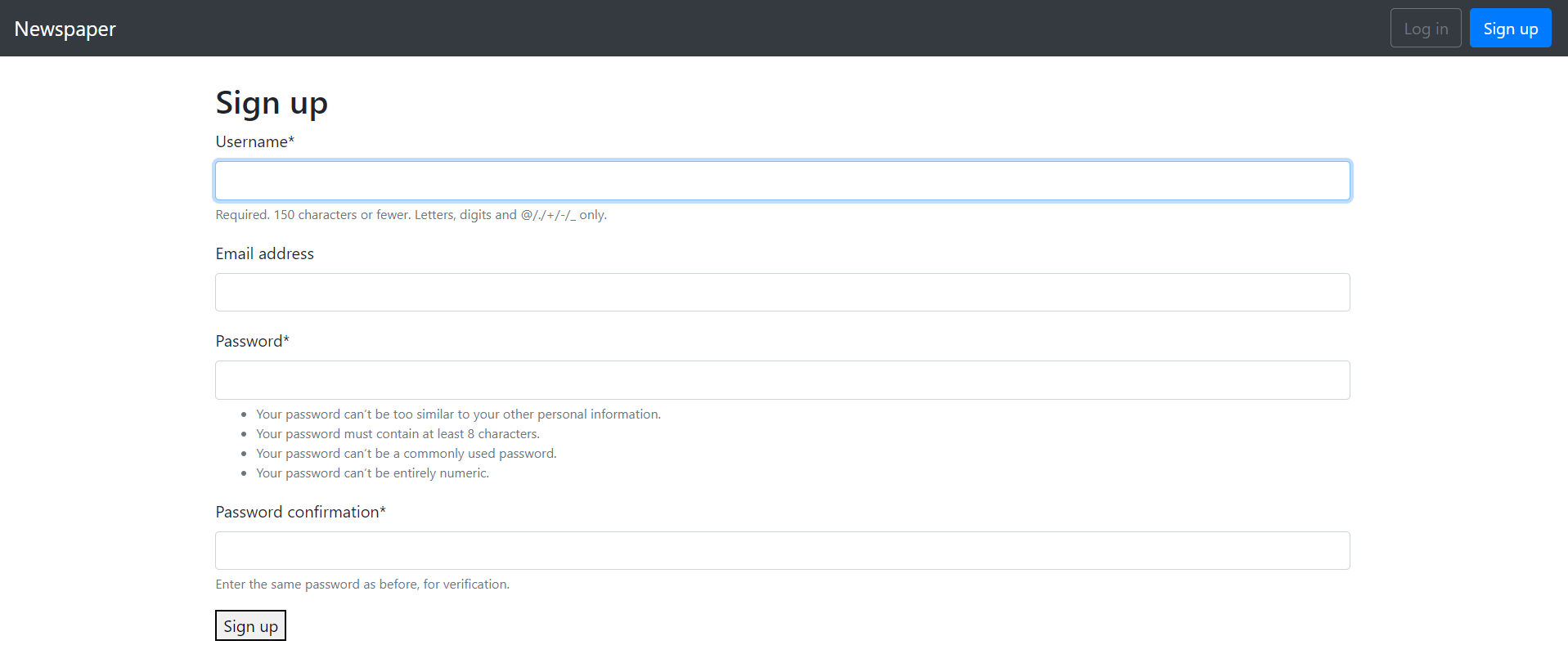
Поскольку мы используем Bootstrap 4, мы также должны добавить эту конфигурацию в settings.py файл. Это идет в нижней части файла.



Теперь в нашем шаблоне signup.html мы можем быстро использовать crispy формы. Сначала мы загружаем crispy\_forms\_tags сверху, а затем меняем {{ form.as\_p }} на {{ form|crispy }}.



Если вы снова запустите сервер с помощью python manage.py runserver и обновите страницу регистрации, мы увидим новые изменения.

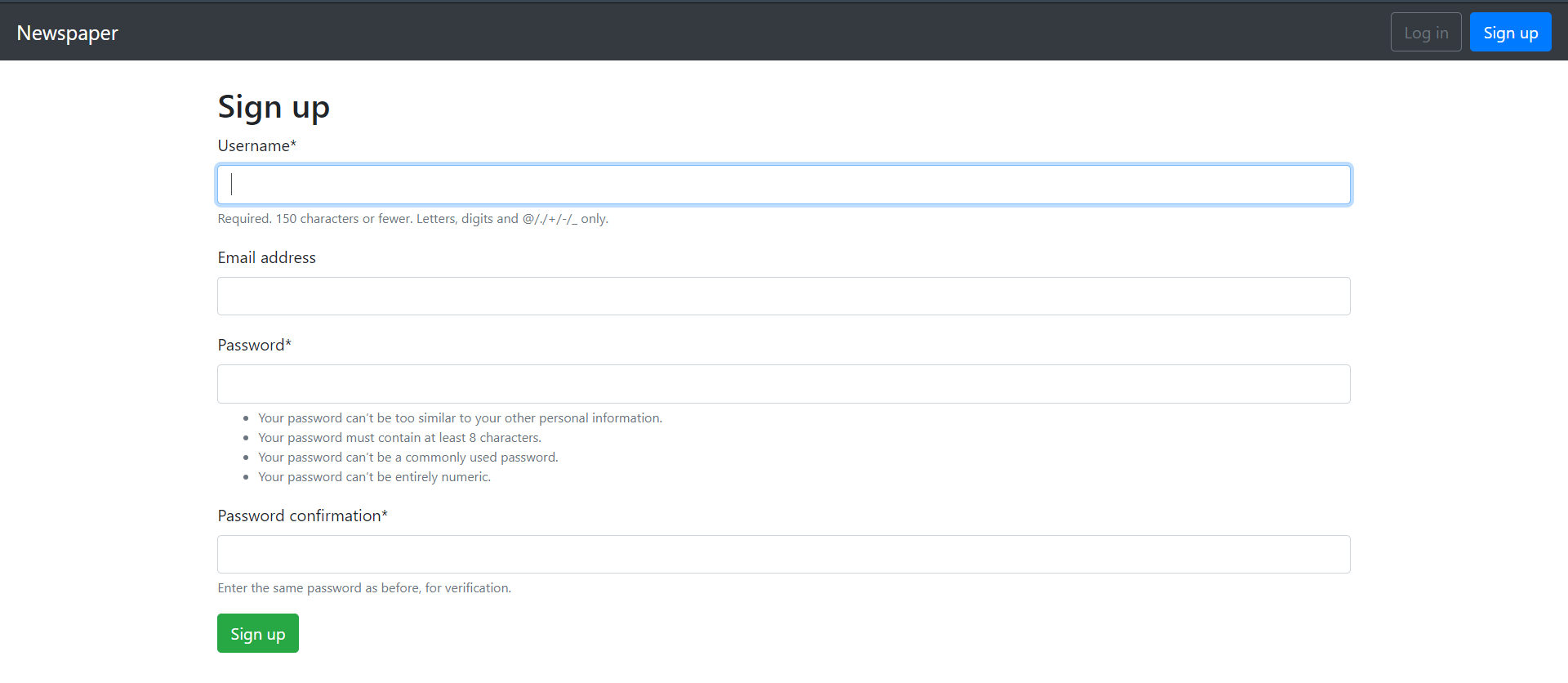


Намного лучше. Хотя, как насчет того, чтобы наша кнопка “Sign up” была немного более привлекательной? В Bootstrap есть все виды стилей кнопок, которые мы можем выбрать. Давайте использовать “success”, который имеет зеленый фон и белый текст.

Обновите файл signup.html в строке для кнопки регистрации.



Обновите страницу, и вы увидите нашу обновленную работу.



**Отправьте проект на GitHub.**