

## Lab 6 Part (3)

### Bootstrap

In this exercise we are going to improve the layout of our website. Instead of writing all the HTML and CSS code, we will use Bootstrap, the most popular framework for building responsive, mobile-first projects. Rather than write all our own code for common website layout features, we can instead rely on Bootstrap to do this for us. This means with only a small amount of code on our part we can quickly have great looking websites. And if we want to make custom changes as a project progresses, it is easy to override Bootstrap where needed, too. When you want to focus on the functionality of a project and not the design, Bootstrap is a great choice.

There are two ways to add Bootstrap to a project: you can download all the files and serve them locally or rely on a Content Delivery Network (CDN). The second approach is simpler to implement provided you have a consistent internet connection so that is what we will use here.

Bootstrap comes with a starter template that includes the basic files needed. Notably there are four that we incorporate:

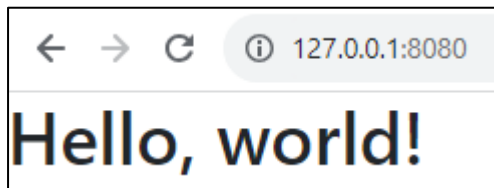
- Bootstrap.css
- jQuery.js
- Popper.js
- Bootstrap.js

Overwrite the **base.html** file with the code provided here.

## base.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <!-- Required meta tags -->
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1" />
    <!-- CSS only -->
    <link
      href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
      rel="stylesheet"
      integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdeRh002iuK6FUUVM"
      crossorigin="anonymous"
    />
    <title>{% block title %}{% endblock title %}</title>
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
      integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"
      crossorigin="anonymous"
    ></script>
  </body>
</html>
```

If you start the server again with **python manage.py runserver** and refresh the homepage at <http://127.0.0.1:8080/> you will see that only the font size has changed at the moment.



Next, we will 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.

Delete the `<h1>` tag with “Hello World” i.e., line 17 and copy paste all the code provided into the **base.html** file at this location.

```
templates > <> base.html > ...
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <!-- Required meta tags -->
5          <meta charset="utf-8" />
6          <meta name="viewport" content="width=device-width, initial-scale=1" />
7          <!-- CSS only -->
8          <link
9              href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
10             rel="stylesheet"
11             integrity="sha384-9ndCyUaIbzAi2FUVXJi0CjmCapSm07SnpJef0486qhLnuZ2cdeRhO02iuK6FUUVM"
12             crossorigin="anonymous"
13         />
14         <title>{% block title %}{% endblock title %}</title>
15     </head>
16     <body>
17         <h1>Hello World</h1>
18         <div class="container">{% block content %}{% endblock content %}</div>
19         <script
20             src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"
21             integrity="sha384-geWF76RCwLtnZ8qwWowPQNguL3RmwHVBC9FhGdlKrxdiJJigb/j/68SIy3Te4Bkz"
22             crossorigin="anonymous"
23         ></script>
24     </body>
25 </html>
```

## base.html

```
<nav class="navbar navbar-expand-lg navbar-dark bg-dark">

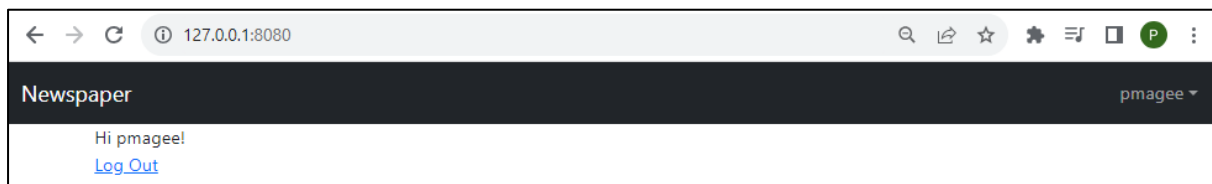
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Newspaper</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
target="#navbarSupportedContent"
      aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarSupportedContent">
      {% if user.is_authenticated %}
      <ul class="nav navbar-nav ms-auto">
        <li class="nav-item dropdown">
          <a href="#" class="nav-link dropdown-toggle" data-bs-toggle="dropdown">{{user.username}}</a>
          <div class="dropdown-menu dropdown-menu-end">
            <a href="{% url 'password_change' %}" class="dropdown-item">Change Password</a>
            <a href="{% url 'logout' %}" class="dropdown-item">Logout</a>
          </div>
        </li>
      </ul>
      </div>
      {% 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>
<div class="container">

    {% block content %}
    {% endblock content %}

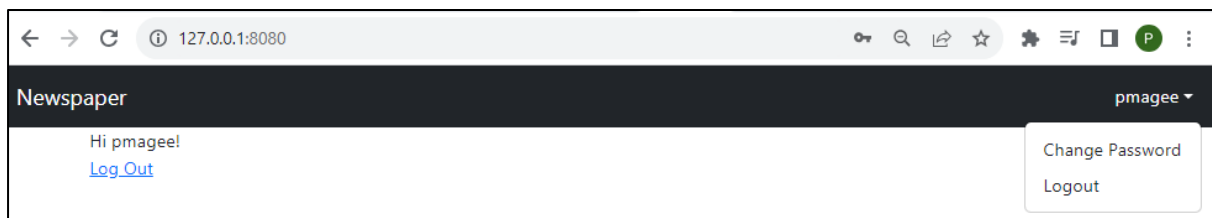
</div>
```

If you refresh the homepage at <http://127.0.0.1:8080/> our new nav has appeared (make sure you are logged in):

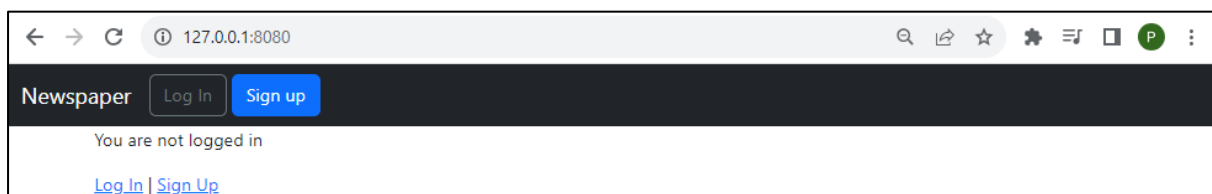


We have also added in our {% block content %} tags so the user greeting has returned, as has our “Newspaper App” in the title.

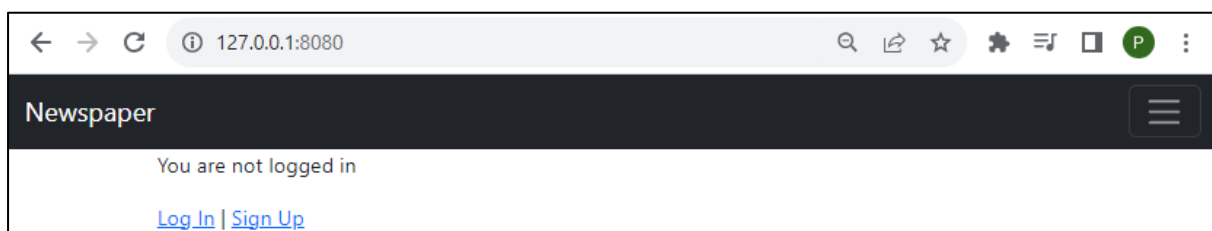
Click on the username in the dropdown menu provided by Bootstrap and you will see two menu items, one for Change Password and the other for Log Out.



If you click on the “Log Out” link, then our nav bar changes offering links to either “Log In” or “Sign Up.”



If you shrink the size of your browser window Bootstrap automatically resizes and adjusts so it looks good on a mobile device, too.



If you click on the “Log Out” button and then “Log In” from the top nav you can also see that our log in page <http://127.0.0.1:8080/users/login> looks better too.



Change the “button” line in **templates/registration/login.html** as follows to make it green.

```
templates > registration > <> login.html > ...
1   {% extends 'base.html' %}
2
3   {% block title %}Log In{% endblock title %}
4
5   {% block content %}
6   <h2>Log In</h2>
7   <form method="post">
8       {% csrf_token %}
9       {{ form.as_p }}
10  <button class="btn btn-success ml-2" type="submit">Log In</button>
11 </form>
12 {% endblock content %}
```

Now refresh the page to see the updated button.



## Sign Up Form

Our sign-up page at <http://127.0.0.1:8080/users/signup/> has Bootstrap stylings but also distracting helper text. For example, after “Username” it says “Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only.”

← → ↻ ⓘ 127.0.0.1:8080/users/signup/ 🔍 📄 ☆ ⚙️ ☰ 📱 P ⋮

Newspaper Log In Sign 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

This helper text comes from Django. We will now look at a way to improve the layout of this helper text in our form. To do this will use a 3<sup>rd</sup> party application **django-crispy-forms** to help us to manage django forms.

Stop the local server with **Control+c**. Then use pip to install the package in our project.

## Command Line

Move into the **django projects** folder and then type the following command:

---

```
pip install crispy-bootstrap5
```

---

Add the new app to our INSTALLED\_APPS list in the **settings.py** file. As the number of apps starts to grow, it is helpful to distinguish between 3rd party apps and local apps we have added. Add in the comment on line 40 below and the new code at lines 43-45. Here's what the code looks like now.



## Code

```
33 INSTALLED_APPS = [  
34     'django.contrib.admin',  
35     'django.contrib.auth',  
36     'django.contrib.contenttypes',  
37     'django.contrib.sessions',  
38     'django.contrib.messages',  
39     'django.contrib.staticfiles',  
40     # local  
41     'users',  
42     'pages',  
43     # 3rd party  
44     'crispy_forms',  
45     'crispy_bootstrap5',  
46 ]
```

Since we are using Bootstrap5 we must add that config to our **settings.py** file. This goes at the bottom of the file.

## Code

```
134 LOGIN_REDIRECT_URL = 'home'  
135 LOGOUT_REDIRECT_URL = 'home'  
136  
137 CRISPY_TEMPLATE_PACK = 'bootstrap5'
```

Now in our **signup.html** template we can quickly use crispy forms. First, we load `crispy_forms_tags` at the top and then replace `{{ form.as_p }}` with `{{ form|crispy }}`.

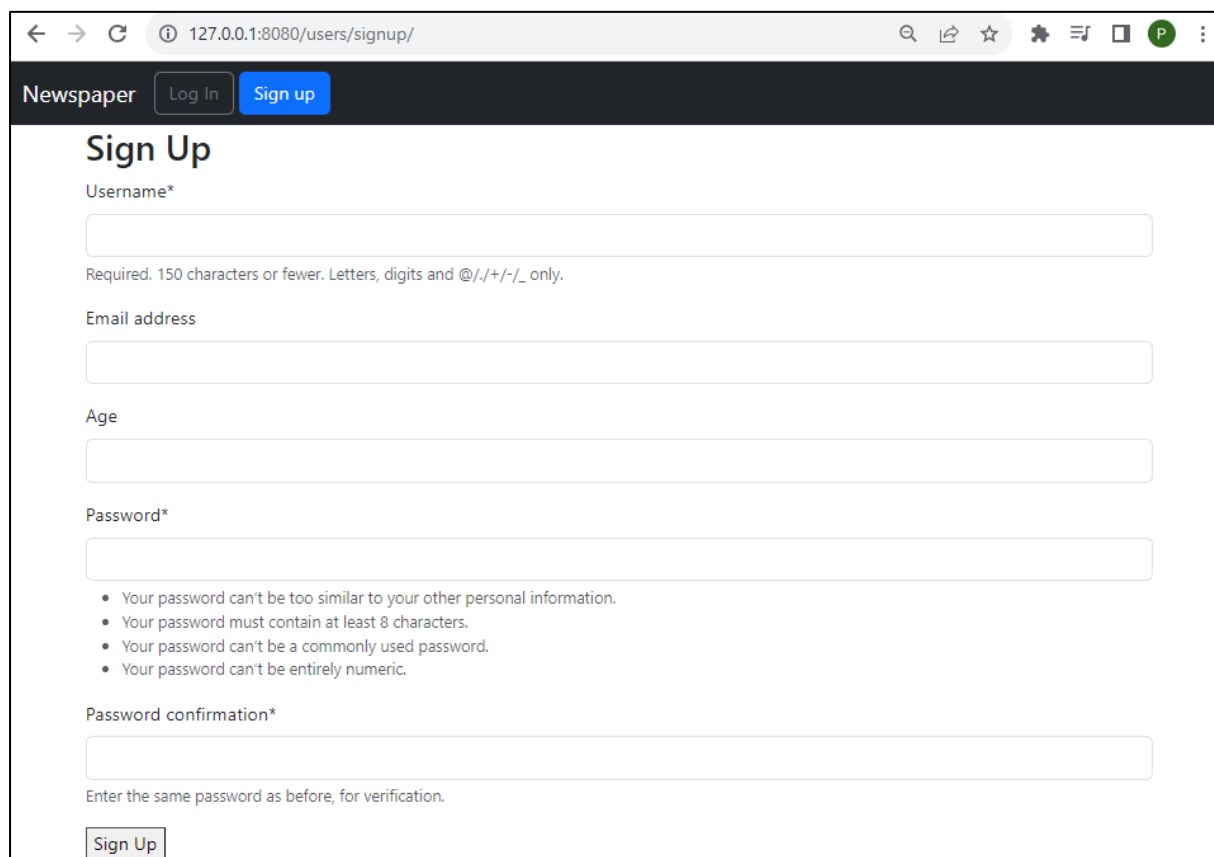
Implement the changes as shown below:

```

templates > registration > <> signup.html > ...
1  {% extends 'base.html' %}
2  {% load crispy_forms_tags %}
3
4  {% block title %}Sign Up{% endblock title %}
5
6  {% block content %}
7  <h2>Sign Up</h2>
8  <form method="post">
9      {% csrf_token %}
10     {{ form|crispy }}
11     <button type="submit">Sign Up</button>
12 </form>
13 {% endblock content %}

```

If you start up the server again with **python manage.py runserver** and refresh the signup page, we can see the new changes.



The screenshot shows a web browser at the URL `127.0.0.1:8080/users/signup/`. The page has a dark header with the word "Newspaper" and two buttons: "Log In" and "Sign up". The main content area is titled "Sign Up" and contains a form with the following fields and labels:

- Username\***: A text input field with a required message below it: "Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only."
- Email address**: A text input field.
- Age**: A text input field.
- Password\***: A text input field with a list of requirements below it:
  - 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\***: A text input field with a message below it: "Enter the same password as before, for verification."

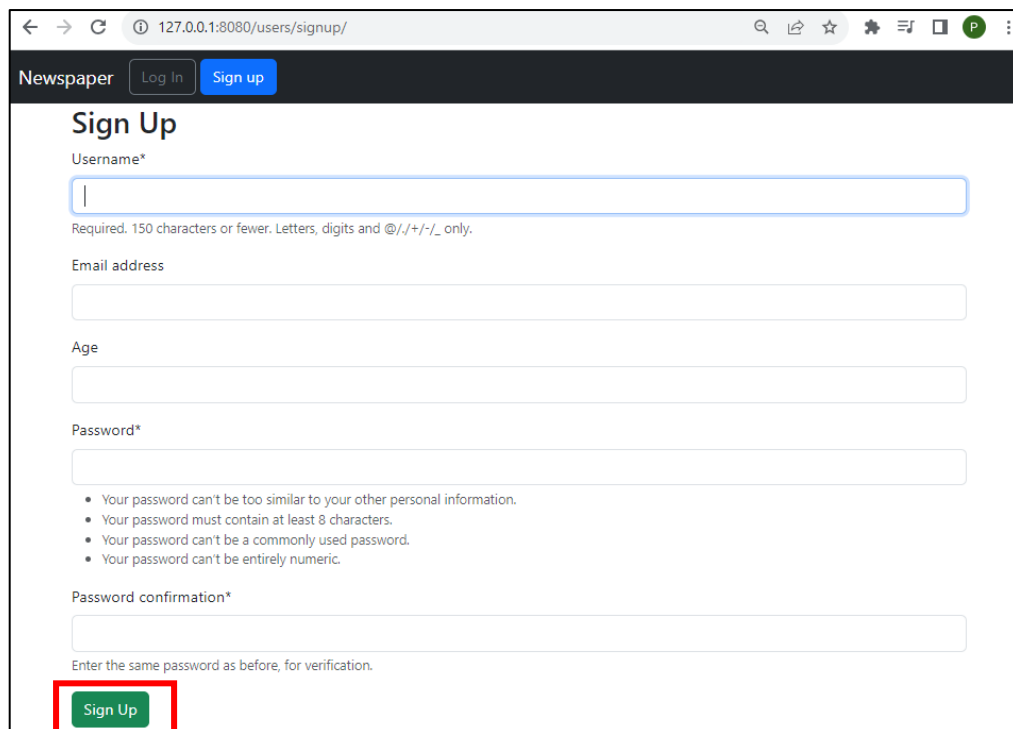
At the bottom of the form is a "Sign Up" button.

The form now looks much better. Next, we will change the colour of our “Sign Up” button to make it green? Bootstrap has all sorts of button styling options we can

choose from. Let's use the "success" one which has a green background and white text. Update the **signup.html** file on the line for the sign up button.

```
templates > registration > <> signup.html > ...
1  {% extends 'base.html' %}
2  {% load crispy_forms_tags %}
3
4  {% block title %}Sign Up{% endblock title %}
5
6  {% block content %}
7  <h2>Sign Up</h2>
8  <form method="post">
9      {% csrf_token %}
10     {{ form|crispy }}
11     <button class="btn btn-success" type="submit">Sign Up</button>
12 </form>
13 {% endblock content %}
```

Refresh the page and you can see our updated work.



The screenshot shows a web browser window with the URL `127.0.0.1:8080/users/signup/`. The page has a dark header with the text "Newspaper" and two buttons: "Log In" and "Sign up". The main content area is titled "Sign Up" and contains several form fields:

- Username\***: A text input field with a placeholder character. Below it, a note says "Required. 150 characters or fewer. Letters, digits and @/./+/-/\_ only."
- Email address**: A text input field.
- Age**: A text input field.
- Password\***: A text input field. Below it, a list of requirements is shown:
  - 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\***: A text input field. Below it, a note says "Enter the same password as before, for verification."

At the bottom of the form, there is a green "Sign Up" button, which is highlighted with a red rectangular box.

Run the following git commands to update the local and remote repositories:

**git add -A**

**git commit -m "lab 6 part 3"**

**git push -u origin main**