Lab 9 Part 3

Static Assets/Images

Static assets like CSS, JavaScript, and images are a core component of any website and Django provides us with a large degree of flexibility around their configuration and storage.

staticfiles app

Django relies on the **staticfiles** app to manage static files from across our entire project, make them accessible for rapid local development on the file system, and combine them into a single location that can be served in a better performing manner in production. This process and the distinction between local and production static files confuses many Django newcomers.

To start we will update the **staticfiles** app configuration in **settings.py**.

STATIC_URL

The first static file setting, STATIC_URL, is already included for us in the **bookstore_project/settings.py** file. If you scroll down to the end of **settings.py** you should see the following entry:

```
125   STATIC_URL = 'static/'
```

This sets the URL that we can use to reference static files. Note that it is important to include a trailing slash / at the end of the directory name.

STATICFILES_DIRS

Now we need to set the STATICFILES_DIRS which defines the location of static files in local development. In our project these will all live within a top-level static directory in our project.

It is often the case that there will be multiple directories with static files within a project so Python brackets [], which denote a list, are typically added here to accommodate future additions.

```
125   STATIC_URL = '/static/'
126   STATICFILES_DIRS = [str(BASE_DIR.joinpath('static'))] # new
```

STATIC_ROOT

STATIC_ROOT is the location of static files for production so it must be set to a different name, typically staticfiles. When it comes time to deploy a Django project, the collectstatic command will automatically compile all available static files throughout the entire project into a single directory. This is far faster than having static files sprinkled across the project as is the case in local development.

```
125   STATIC_URL = '/static/'
126   STATICFILES_DIRS = [str(BASE_DIR.joinpath('static'))] # new
127   STATIC_ROOT = str(BASE_DIR.joinpath('staticfiles')) # new
```

STATICFILES_FINDERS

The last setting is STATICFILES_FINDERS which tells Django how to look for static file directories. It is implicitly set for us and although this is an optional step, we will make it explicit in all projects.

```
128   STATICFILES_FINDERS = [
129       "django.contrib.staticfiles.finders.FileSystemFinder",
130       "django.contrib.staticfiles.finders.AppDirectoriesFinder",
131   ]
```

The FileSystemFinder looks within the STATICFILES_DIRS setting, which we set to static, for any static files. Then the AppDirectoriesFinder looks for any directories named static located within an app, as opposed to located at a project-level static directory. This setting is read from top-to-bottom meaning if a file called static/img.jpg is first found by FileSystemFinder it will be in place of an img.jpg file located within, say, the pages app at pages/static/img.jpg.

Static Directory

Let's now add some static files and incorporate them into our project. Even though we are referring to a static directory for our files we must create this directory along with new subdirectories for CSS, JavaScript, and images.

In VS Code create a folder called **static** at the project level. Inside this folder create three other folders called **css**, **js** and **images**.

Images

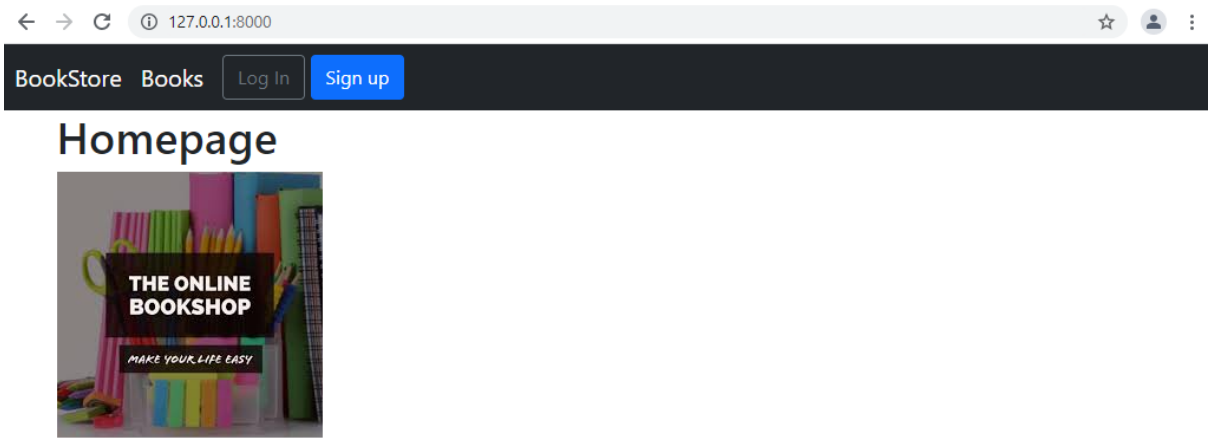
Now we will add an image to our homepage. You can download and extract a set of images from Moodle (projectimages.zip). Copy the **home.jpg** image from **projectimages** folder the into the folder **lab9/static/images**.

To display the image on the home page, open **templates/home.html** and check that the {% load static %} tag is at the top of the file as shown on line 2 below.

Add the line of code shown at line 6 to include the image on the home page:

```
templates > <> home.html > ...
1   {% extends 'base.html' %}
2   {% load static %}
3   {% block title %}Home{% endblock title %}
4   {% block content %}
5       <h1>Homepage</h1>
6       <img class="homecover" src="{% static 'images/home.jpg' %}">
7   {% endblock content %}
```

Open the homepage at http://http://127.0.0.1:8080/ & you will see the image.



JavaScript

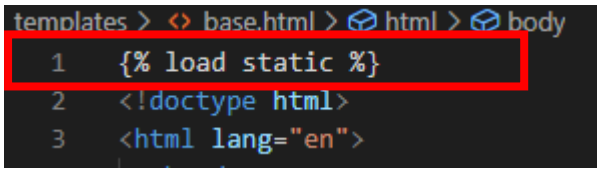
To add JavaScript create a new file in VS Code inside the **static/js** folder called **base.js**

We will add just a small piece of test code into our **base.js** file so we can confirm the JavaScript loaded correctly.

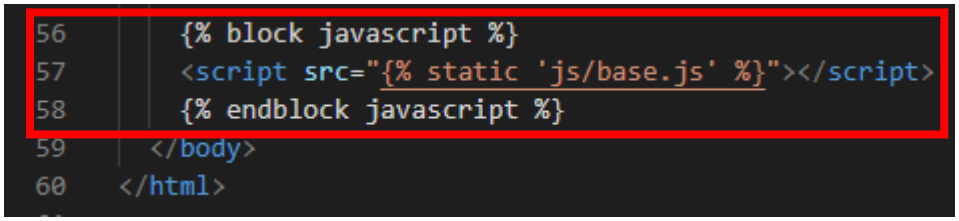
Enter the following line of code into **base.js**

Next, we need to add the JavaScript to our **base.html** template. JavaScript should be added at the bottom of the file, so it is loaded last, after the HTML, CSS, and other assets that appear first on the screen when rendered in the web browser. This gives the appearance of the complete webpage loading faster.

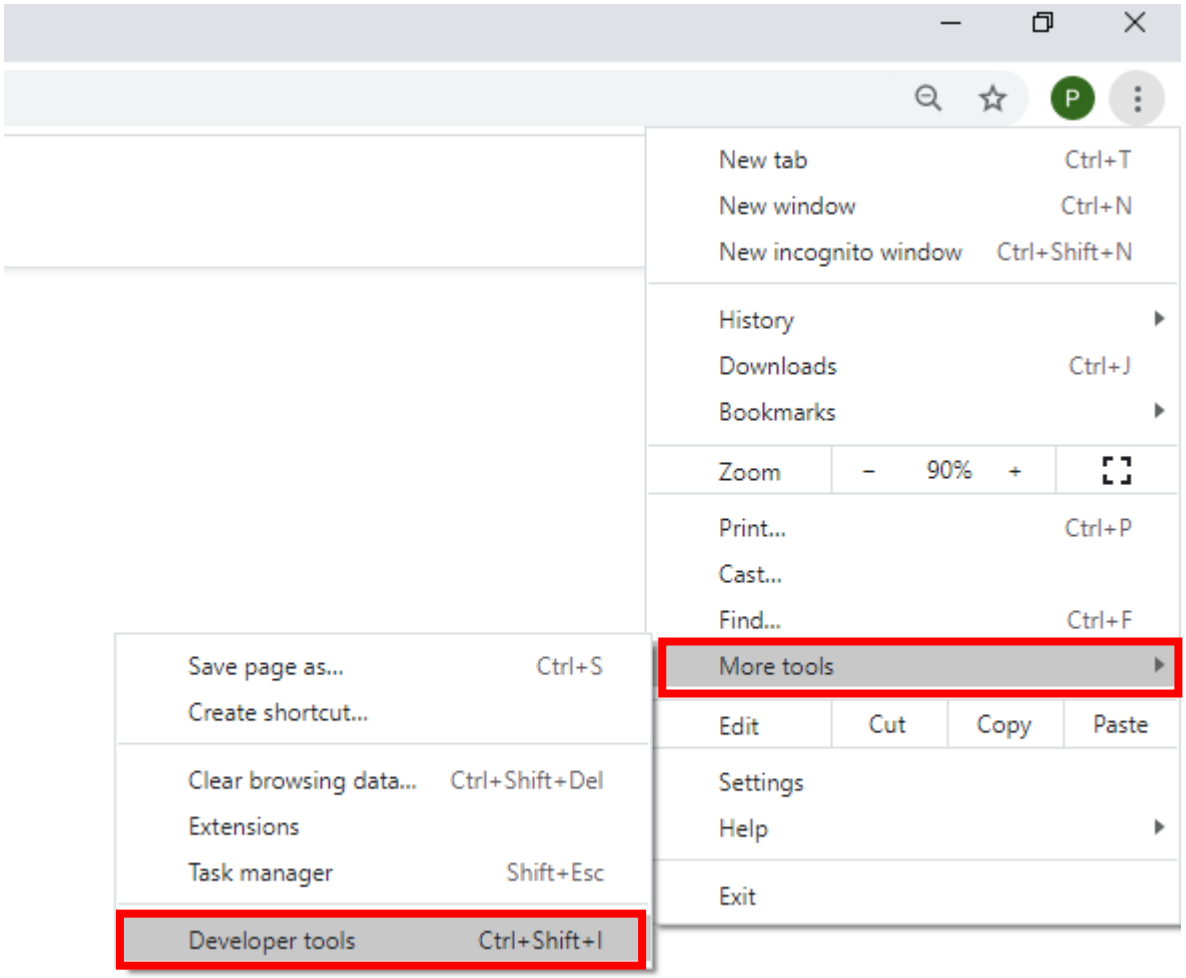
Add the following line of code to the top of **base.html**:



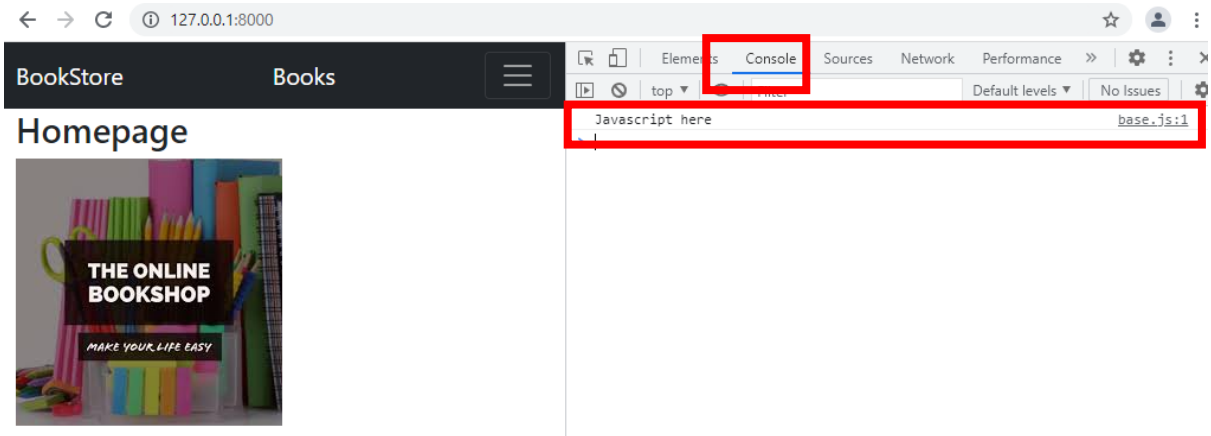
Add the following block of code to the end of **base.html** just before the closing body tag:



Go to the home page in your browser and make the JavaScript console available. To do this open Developer Tools and making sure you're on the "Console" section. In Chrome for example, go to **More tools**, then **Developer Tools** which will open a sidebar.



Make sure **Console** is selected from the options. If you refresh the page, you should see the following:



Stop the server and run the following git commands to update the local and remote repositories:

git add -A

git commit -m “lab 9 part 3 commit”

git push -u origin main