

Exercise 7: Creating the Project

🕒 15 to 25 minutes

1. Open `djangojokes.com` in the terminal and create the new project by running (**Don't forget the dot at the end**):

```
(.venv) ~/projects/djangojokes.com> django-admin startproject djangojokes .
```

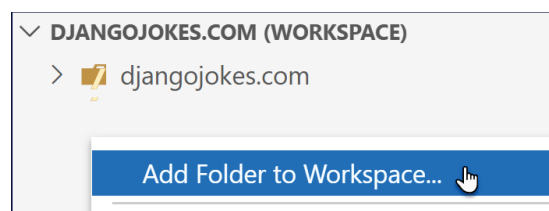
2. Add a `templates` folder to the `djangojokes.com` folder. The folder should now look like this:

```
📁 djangojokes.com
  📁 .venv
  📁 djangojokes
  📁 templates
  📄 .gitignore
  📄 LICENSE
  📄 manage.py
  📄 README.md
```

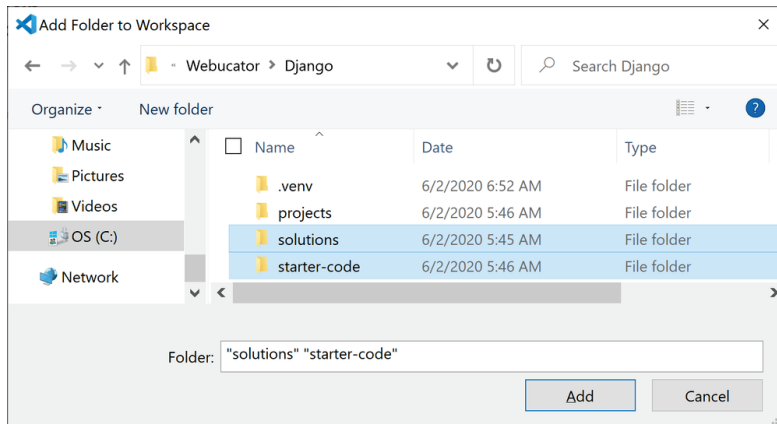
❖ E7.1. Starter Code

We have a lot of code pre-written for you in the `starter-code` folder. We also have the final solutions for each lesson in the `solutions` folder. It will be easier for you to access this code if those folders are in your workspace:

1. Right-click in the **Explorer** area and select **Add Folder to Workspace...**:



2. Select solutions and starter-code from Webucator/Django:



3. You don't want the solutions and starter-code folders to be in version control. Open .gitignore and add a "# MY ADDITIONS" section to the bottom of the file:








```
# MY ADDITIONS
solutions
starter-code
```

Git Commit

1. Click the **Source Control** icon:



The changes should include:

-  .gitignore
-  manage.py
-  __init__.py
-  asgi.py
-  settings.py
-  urls.py
-  wsgi.py

They should **not include** anything in your solutions or starter-code folders.

2. Add a short message (e.g., “Initial project”), click the “Commit” check, click the ellipsis, and select **Push**.

❖ E7.2. Adding a Base Template

Every project typically will have a base template, which we like to name `_base.html`. We have created one that you can customize as you see fit:

1. Directly in your `starter-code` folder, you will find a `_base.html` file. Open `_base.html` for editing:
 - A. The `_base.html` file includes Bootstrap classes. Bootstrap is a popular framework for designing web pages with CSS and JavaScript. You need to add references to the Bootstrap CSS and JavaScript files in `_base.html`:
 - i. Visit <https://getbootstrap.com>.
 - ii. Click the **Get Started** button.
 - iii. Copy the Bootstrap CSS `<link>` tag on that page and paste it over the “ADD BOOTSTRAP CSS” comment in the head of `_base.html`.
 - iv. Copy the Bootstrap JS `<script>` tags on that page and paste them over the “ADD BOOTSTRAP JS” comment immediately above the closing `</body>` tag at the bottom of `_base.html`.
 - B. You will also use some Font Awesome icons. Font Awesome provides a collection of free vector icons that you can use on your websites. You can get access to these icons through a free content delivery network (CDN). To do so, you will need get your own unique `<script>` tag with the latest version of Font Awesome:
 - i. Go to <https://fontawesome.com/start>.
 - ii. Enter your email address in the form and submit. You will be sent an email asking you to confirm your email address and create an account. After creating an account, you will be provided with a `<script>` tag that looks something like this:

```
<script src="https://kit.fontawesome.com/yoursecretcode.js"
    crossorigin="anonymous"></script>
```
 - iii. Paste this tag over the “ADD FONT AWESOME” comment in the head of `_base.html`.

2. Close `_base.html`. Then, copy the `_base.html` file (not the text of the file but the file itself) from the `starter-code` folder and paste it into the `djangojokes.com/templates` folder you just created.
3. Open `djangojokes.com/templates/_base.html` for editing.
4. Replace all instances of “SITENAME” in `_base.html` with “Django Jokes”.
5. Replace the link text that reads “NavLink” with “Jokes”.
6. We need to tell Django to look in this `templates` folder when searching for templates. Open `djangojokes/settings.py` and add the following to the `DIRS` value of the dictionary in `TEMPLATES`:

```
'DIRS': [BASE_DIR / 'templates'],
```

And that’s it: the project is created and ready to use.

Git Commit

Whenever you make significant changes, you should commit your code to Git. From here on out, we will remind you to commit (you can commit even more often if you wish), but we will not walk you through the process every time. If you need to, you can refer back to the initial instructions (see page 43). Note that you do not need to push your code every time, but it won’t be backed up on GitHub until you do.