# **Create Virtual Environment and Django Project**

## Windows <**Python 3.11.7 and PyCharm**> Edition

1. Begin by checking the version of Python you are using.

A black background with white text

Description automatically generated

1. Create a workspace folder using: “mkdir CS3300 | cd CS3300”
2. Create a portfolio folder; then, create a virtual environment inside said folder.

A computer screen shot of a computer

Description automatically generated

1. Activate virtual environment with: “djvenv\bin\activate.bat”
2. Begin installing Django in the virtual environment.

A screenshot of a computer program

Description automatically generated

1. Update pip using the given command: “…\python.exe -m pip install --upgrade pip”.
2. Create a Django project using: “py -m django startproject django\_project”.
3. For ease of use, reorder the directory structure to match the one shown below.

A screen shot of a computer

Description automatically generated

1. Run the server and ignore and migration warnings that appear.

A screen shot of a computer

Description automatically generated

1. See if the installation worked by going to: <http://localhost:8000/>
2. Open two more terminal windows; activate the virtual environment in one.

A screenshot of a computer

Description automatically generated

1. Check all installed Python packages using either pip’s “list” or “freeze” commands.

A computer screen with white text

Description automatically generated

1. Create a requirements file containing what packages are installed.

A screen shot of a computer

Description automatically generated

1. In PyCharm (IDE), go to ‘File -> Open Folder’ and select the folder/repository containing the newly-created Django project.

# **Create Local Git and Github Repository**

1. Create a private GitHub repository (separate from the team’s repository).
2. Initialize your local directory/folder; create your “.gitignore” file.

A screen shot of a computer

Description automatically generated

1. Commit changes and link your local repository to the GitHub repository.

**A screen shot of a computer

Description automatically generated**

1. Once finished, push the changes to the main branch.

**A screen shot of a computer

Description automatically generated**

# **Create Portfolio App**

1. Create and switch to a branch called “sprint01”.

A black screen with white text

Description automatically generated

1. Activate the virtual environment.
2. Create a portfolio application.

A screen shot of a computer

Description automatically generated

1. In PyCharm (IDE), go to ‘File -> Open Folder’ and select the folder/repository containing the newly-created portfolio application.
2. In PyCharm (IDE), open the django\_project’s setting.py file.
3. Add the app to the “INSTALLED\_APPS” list; append support for authenticating users.

A screenshot of a computer

Description automatically generated

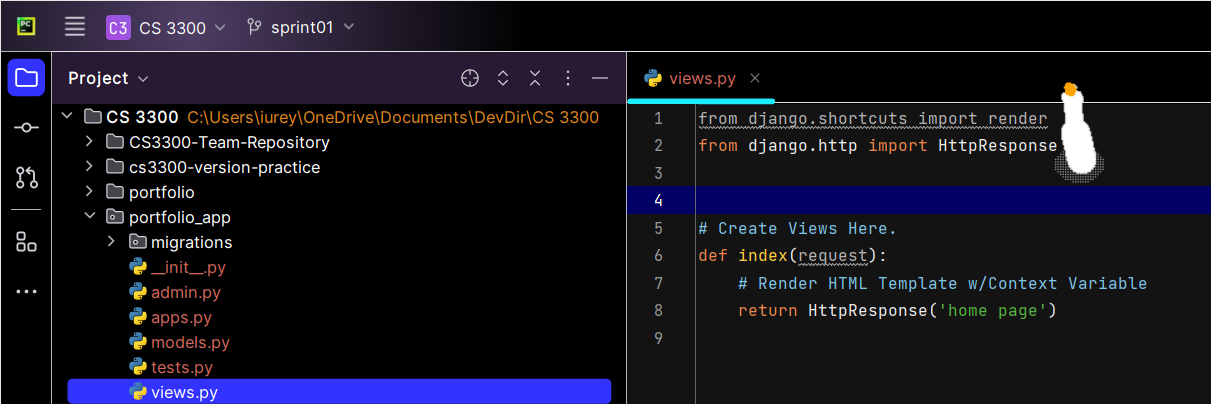
# **Define URI Path & View**

1. In PyCharm (IDE), go to ‘File -> Open Folder’ and select the folder/repository containing the newly-created portfolio application.
2. In PyCharm (IDE), open the django\_project’s urls.py file.
3. Add a path below the admin path to include the portfolio\_app urls.py specific URLs.

**A screenshot of a computer

Description automatically generated**

1. Update the portfolio\_app/views.py by defining the following homepage view.



1. Create a urls.py file in portfolio\_app that contains a path to the defined view.

A screenshot of a computer

Description automatically generated

1. Run the server and ignore and migration warnings that appear.

A screen shot of a computer

Description automatically generated

1. See if the updated view worked by going to: <http://127.0.0.1:8000>

A screenshot of a computer

Description automatically generated

1. Update the portfolio\_app/views.py by defining a new homepage view.

A screenshot of a computer

Description automatically generated