

## ADDITIONAL READING

# Django Project Setup

[Visit our website](#)

## WELCOME TO THE DJANGO PROJECT SETUP GUIDE!

In this guide, we will walk through the process of creating a new Django project, explaining the essential steps, from installing Django to setting up a new application within your project.

## CREATING A NEW DJANGO PROJECT

Ensure your **venv** is active. There will be an extension to your file path in your terminal such as “**(.venv)**” or “**(virtual\_environment\_name)**”. If this is not visible, you can select the environment by using Ctrl+Shift+P, and in the dropdown type “Python: Select Interpreter” and choose your virtual environment from the list.

Next, we will need to install the latest version of Django using **pip**, which is the package installer for Python. It's a tool that helps you install and manage Python packages, which are collections of code and resources that provide specific functionality. Do this by using the following code:

```
pip install django
```

Creating a Django project with the following:

```
django-admin startproject project_name
```

You will notice a new directory with your **project\_name** with a matching folder inside. This folder is the main site of your project that links all your site's functionality together. The folder structure looks like this:

```
project_name/ # This is your project directory
  manage.py
  project_name/ - # This is your main site
    __init__.py
    settings.py
    urls.py
    asgi.py
    wsgi.py
```

Please take note of the Python file called **manage.py**, which is the program that allows us to interact with the project. We can view our interaction options by running the following command (ensure you are inside the upper **project\_name** folder):

```
python manage.py
```

With this command, we are specifying the program (**python**) and the program we want to use to run (**manage.py**).

To create an application for our project we can use **manage.py**. First, we need to **cd** (change directory) into our project folder to access the **manage.py** program.

```
cd your_path/to/your/project/project_name
```

Then we will run the following command:

```
python manage.py startapp app_name
```

This will add the following folder and structure to your project:

```
app_name
  __init__.py
  migrations/
    __init__.py
    admin.py
    apps.py
    models.py
    tests.py
    views.py
```

Once your app is built, you need to modify some aspects to create a functioning site.

Navigate to the **settings.py** program (**project\_name\project\_name\settings.py**). Inside this program we will scroll to the **INSTALLED\_APPS = []** section, where we will append the list with the name of our application:

```
INSTALLED_APPS = [
    'polls.apps.PollsConfig',
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
```

```
'django.contrib.messages',  
'django.contrib.staticfiles',  
'app_name',  
)
```

Run the initial database migrations to set up the database tables:

```
python manage.py migrate
```

### Create a superuser (optional):

If you want to access the Django admin interface, you can create a superuser account with the following command:

```
python manage.py createsuperuser
```

**Note:** When creating a superuser the password field will remain blank while you are typing the password, but Django is in fact accepting your input.

Start the Django development server to see your project in action:

```
python manage.py runserver
```

By default, it will run on **<http://localhost:8000/>**.

### Access the admin interface (optional):

If you created a superuser, you can access the admin interface at **<http://localhost:8000/admin/>** and log in with the superuser credentials.

Now that your project is set up, you can start building your app by defining models, views, and templates within your app directory.