• Introduction to deploying Django projects to live servers like PythonAnywhere.

Deploying a Django project to live servers like **PythonAnywhere** involves several steps. PythonAnywhere is a cloud-based hosting service specifically designed for Python applications, and it offers an easy way to deploy Django projects:

**1. Set Up Your PythonAnywhere Account**

* Sign up for an account at [PythonAnywhere](https://www.pythonanywhere.com/).
* After signing in, you’ll be taken to the dashboard.

**2. Prepare Your Django Project**

* Make sure your Django project is ready for deployment:
  + **Settings Adjustments**: Update the settings.py file to handle production environments:
    - Set DEBUG = False.
    - Add the server domain name (e.g., yourusername.pythonanywhere.com) to the ALLOWED\_HOSTS list.
    - Configure the DATABASES setting to use the PythonAnywhere database (if using their free-tier database).
    - Set up static files (STATIC\_ROOT and STATIC\_URL) and media files (MEDIA\_ROOT and MEDIA\_URL).
    - Configure your SECRET\_KEY securely.

**3. Upload Your Code to PythonAnywhere**

* You can either:
  + **Git**: Push your project to a Git repository (e.g., GitHub) and clone it on PythonAnywhere.
  + **Manual Upload**: Upload your project files directly using PythonAnywhere’s file management interface or SFTP.
* After uploading, navigate to the folder containing your project.

**4. Set Up a Virtual Environment**

* PythonAnywhere allows you to create a virtual environment to manage dependencies.
  + Go to the **Virtualenvs** tab in the dashboard.
  + Create a new virtual environment (e.g., python3.8).
* Install the required packages (e.g., Django, database connectors, etc.) using pip:

pip install -r requirements.txt

**5. Set Up the Web App on PythonAnywhere**

* Navigate to the **Web** tab on your PythonAnywhere dashboard.
* Click **Add a new web app**.
  + Choose the **Django** option.
  + Select your virtual environment.
  + Set the working directory to your project folder.
  + Set the WSGI file to point to your project. This file will link PythonAnywhere's web server to your Django application.

**6. Configure the Database**

* If using PythonAnywhere's default SQLite database, ensure it’s configured correctly.
* For MySQL or PostgreSQL, configure the database connection settings (username, password, host) in your Django settings file.
* Run Django's migrations to set up the database schema:

python manage.py migrate

**7. Set Up Static and Media Files**

* PythonAnywhere requires you to set up static and media file directories manually.
* In the **Web** tab:
  + Set up **Static files**: Specify the URL (e.g., /static/) and the directory where static files will be served from (e.g., /home/yourusername/mysite/static/).
  + Set up **Media files**: Define the URL (e.g., /media/) and the directory for media files.
* Collect static files by running:

python manage.py collectstatic

**8. Update the WSGI Configuration**

* In the **Web** tab, under the **WSGI configuration file**, make sure it points to your project’s wsgi.py file. Typically, it will look like this:
* python

import os

import sys

path = '/home/yourusername/mysite'

if path not in sys.path:

sys.path.append(path)

os.environ['DJANGO\_SETTINGS\_MODULE'] = 'mysite.settings'

from django.core.wsgi import get\_wsgi\_application

application = get\_wsgi\_application()

* This connects PythonAnywhere’s web server to Django.

**9. Restart the Web App**

* Once everything is configured, click **Reload** in the Web tab to restart the web app and apply your changes.

**10. Testing and Debugging**

* Visit your Django site using the URL provided by PythonAnywhere (e.g., yourusername.pythonanywhere.com).
* If something isn’t working, check the **error logs** under the Web tab for troubleshooting.