

Tops Technology

Module 16)

Python DB and Framework

Presented By : Nandni Vala

Live Project Deployment (PythonAnywhere)

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

- Deploying a Django project to a live server like **PythonAnywhere** makes your application accessible to the public.
 - **Prepare Your Django Project for Deployment**
 - **Ensure Debug is Disabled:** Set `DEBUG = False` in your `settings.py` file for production.
 - `DEBUG = False`
 - `ALLOWED_HOSTS = ['your-username.pythonanywhere.com']`
 - **Collect Static Files:** Run the following command to gather all static files into a single directory:
 - `python manage.py collectstatic`
 - **Secure Secrets:** Store sensitive information, like secret keys and database credentials, in environment variables or a `.env` file (do not commit `.env` to Git).
-

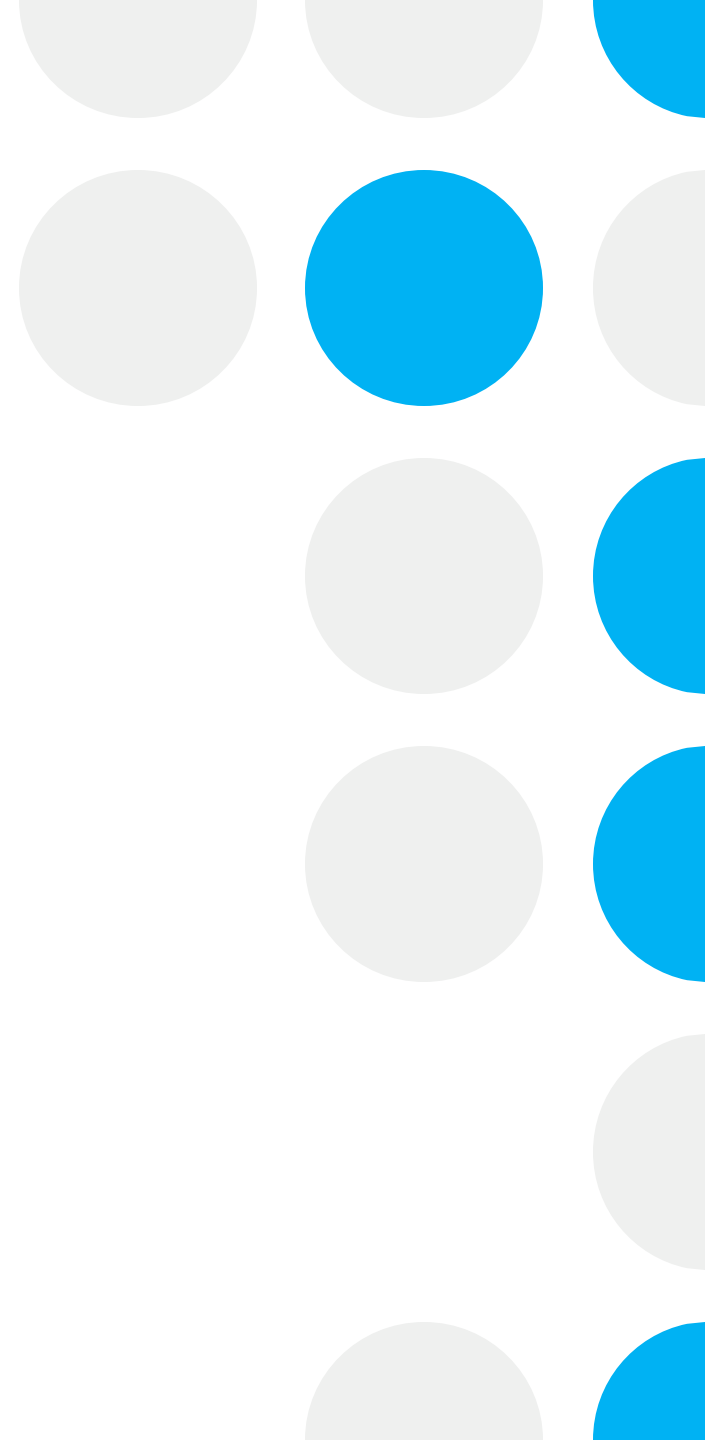
- **Create an Account on PythonAnywhere**
 - Sign up for a free or paid PythonAnywhere account at <https://www.pythonanywhere.com>.
 - After logging in, navigate to the **Dashboard**.
 - **Upload Your Django Project**
 - Go to the **Files** section on PythonAnywhere.
 - Create a directory for your project (e.g., my_project/).
 - Upload your project files (or use Git to clone your repository):
 - git clone <https://github.com/username/your-repo.git>
 - **Set Up a Virtual Environment**
 - Navigate to the **Consoles** section and open a new **Bash console**.
 - Create a virtual environment (if not already set up):
 - python3 -m venv venv
 - Activate the virtual environment:source
 - Venv/bin/activate
 - Install required dependencies:
 - pip install -r requirements.txt
-

➤ **Set Up the Web App**

- Go to the **Web** section in PythonAnywhere.
 - Click **Add a new web app**.
 - Choose **Manual configuration** and select the Python version for your project.
 - Configure the Django app:
 - Set the path to your project directory (e.g., /home/your-username/my_project).
 - Set the path to your WSGI file (e.g., /home/your-username/my_project/my_project/wsgi.py)
 - **Configure the WSGI File**
 - Edit the WSGI file (/var/www/your-username_pythonanywhere_com_wsgi.py) to point to your Django project.
-



- `import os`
 - `import sys`
 - `# Add your project directory to the system path`
 - `sys.path.append('/home/your-username/my_project')`
 - `sys.path.append('/home/your-username/my_project/my_project')`
 - `# Set the Django settings module`
 - `os.environ['DJANGO_SETTINGS_MODULE'] = 'my_project.settings'`
 - `# Activate the virtual environment`
 - `activate_env = '/home/your-username/venv/bin/activate_this.py'`
 - `with open(activate_env) as f:`
 - `exec(f.read(), dict(__file__=activate_env))`
 - `from django.core.wsgi import get_wsgi_application`
 - `application = get_wsgi_application()`
-



➤ **Configure the Database**

- **SQLite:** If you're using SQLite, ensure the database file (db.sqlite3) is in your project directory.
- **MySQL:** For MySQL, configure the database settings in settings.py and set up the database in PythonAnywhere using the **Databases** section.

➤ **8. Restart the Web App**

- After making changes, restart the web app from the **Web** section.

➤ **Test Your Application**

- Visit your site at <https://your-username.pythonanywhere.com> to test the deployment.
 - Debug issues using the error logs available in the **Web** section.
 - **Advantages of Using PythonAnywhere**
 - **Free Tier:** Suitable for small projects.
 - **Built-In Tools:** Simplifies database and virtual environment setup.
 - **Beginner-Friendly:** Easy to navigate and configure.
-

