

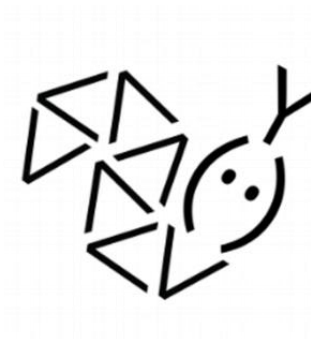
Python

Full stack Skills Bootcamp

Introduction to Deployment

■ Deploying Your Django Application

- **Why Deploy?** Deployment allows you to share your application with others online, making it accessible through a web browser. In this tutorial, we'll use PythonAnywhere for hosting and GitHub for version control.
- **Overview:** You'll learn to prepare your code, set up hosting on PythonAnywhere, push code to GitHub, and configure settings for a successful deployment.



Preparing for Deployment

- **Complete Your Project:** Ensure your Django project is fully functional locally. This means testing your code to confirm everything works.
- **Create Accounts:** Make sure you have accounts on both PythonAnywhere (for hosting) and GitHub (for code storage).
- **Check Dependencies:** Create a requirements.txt file listing all dependencies, so they can be installed easily in the deployment environment.

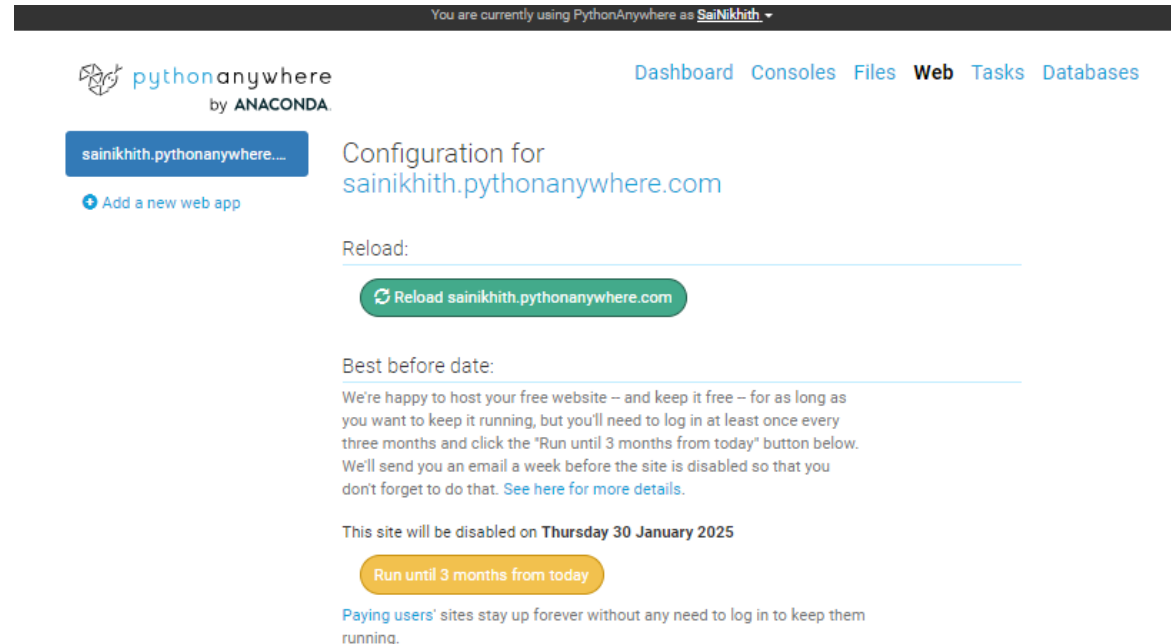


```
bash
```

```
pip freeze > requirements.txt
```

Setting Up PythonAnywhere

- **Sign Up or Log In:** Go to PythonAnywhere and sign up for a free account if you don't have one or log in if you already do.
- **Create a New Web App:** In the PythonAnywhere dashboard, select "Web" and then "Add a new web app." Choose Python as your framework and proceed. This step sets up a hosting environment for your Django project.



The screenshot shows the PythonAnywhere dashboard for a user named 'Sainikhith'. The top navigation bar includes links for Dashboard, Consoles, Files, Web, Tasks, and Databases. The 'Web' section is active, showing the configuration for 'sainikhith.pythonanywhere.com'. A button labeled 'Add a new web app' is visible. The 'Reload' section contains a button to 'Reload sainikhith.pythonanywhere.com'. The 'Best before date' section explains the free hosting policy and provides a 'Run until 3 months from today' button. At the bottom, it states the site will be disabled on Thursday 30 January 2025 and mentions that paying users can keep their sites running forever.

You are currently using PythonAnywhere as **Sainikhith**

pythonanywhere
by ANACONDA

Dashboard Consoles Files **Web** Tasks Databases

sainikhith.pythonanywhere.com

+ Add a new web app

Configuration for
sainikhith.pythonanywhere.com

Reload:

Reload sainikhith.pythonanywhere.com

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details.](#)

This site will be disabled on **Thursday 30 January 2025**

Run until 3 months from today

Paying users' sites stay up forever without any need to log in to keep them running.

Connecting with GitHub

- **Why GitHub?** GitHub allows you to store your code online and easily share it with PythonAnywhere for deployment. It also keeps track of all changes you make to your code.
- **Push Your Code to GitHub:** In your terminal, navigate to your Django project directory and run the following commands to add, commit, and push your code to GitHub.
- **Open a Bash Console:** Go to your PythonAnywhere dashboard, select "Consoles," and open a new Bash console. This is where you'll clone your GitHub repository.
- **Clone the Repository:** Run the following command in the Bash console to copy your code from GitHub to PythonAnywhere.
- After cloning, navigate to the directory where `manage.py` is located.

```
bash
```

```
git init  
git add .  
git commit -m "Initial commit"  
git remote add origin <repository_url>  
git push -u origin main
```

```
bash
```

```
git clone <repository_url>
```

Setting Up Virtual Environment and Dependencies

- **Create a Virtual Environment:** A virtual environment isolates dependencies for your project.
- **Activate the Virtual Environment:** Then, activate the virtual environment.
- **Install Dependencies:** Install all necessary packages listed in requirements.txt.

```
bash
```

```
python -m venv myenv
```

```
bash
```

```
source myenv/bin/activate
```

```
bash
```

```
pip install -r requirements.txt
```

Configuring Django Settings for Deployment

Update ALLOWED_HOSTS:

Open settings.py in your project directory and add your PythonAnywhere domain to ALLOWED_HOSTS so the server can access your app.

Configure Static Files: Set up STATIC_ROOT to collect static files for use in deployment.

```
python
```

```
ALLOWED_HOSTS = ['yourusername.pythonanywhere.com']
```

```
python
```

```
STATIC_ROOT = BASE_DIR / 'static'
```

Applying Database Migrations

Run Migrations: Migrations update your database schema to match your Django models. In the Bash console, navigate to your project directory and run :

```
bash  
  
python manage.py migrate
```

This command ensures that all tables and fields in your database are aligned with your Django app's data structure

Setting Up Static Files

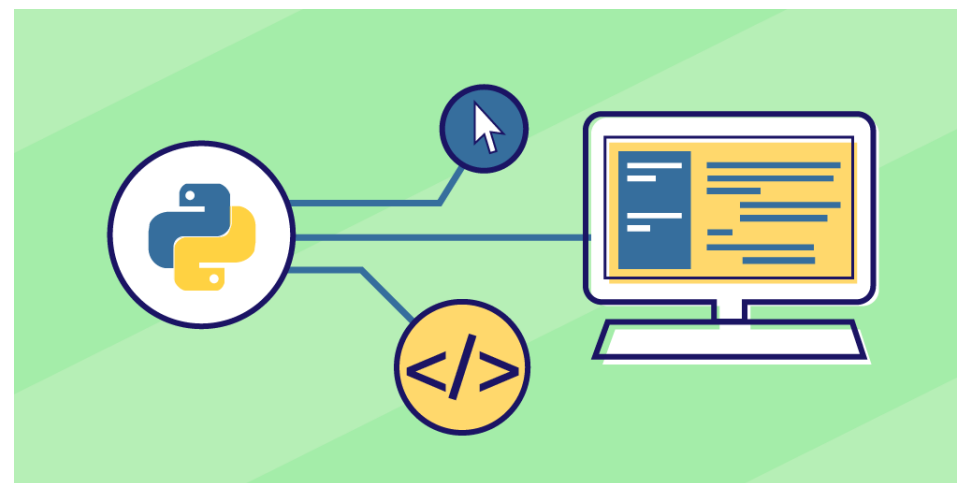
Collect Static Files: Static files, such as CSS and JavaScript, need to be prepared for deployment. Run:

```
bash  
  
python manage.py collectstatic
```

This command gathers all static files from your project into the STATIC_ROOT directory, so they are ready to be served by PythonAnywhere.

Configuring the Web App on PythonAnywhere

- Set Virtual Environment Path: In PythonAnywhere's "Web" tab, go to your web app settings. Specify the path to your virtual environment (e.g., /home/yourusername/myvenv).
- Set WSGI File Path: In the web app settings, link to your WSGI configuration file. This file tells PythonAnywhere how to interact with your Django app.
- Reload the App: After setting up, reload the web app to apply these configurations.



Testing and Finalizing Deployment

- Restart the Web App: Go to the "Web" tab in PythonAnywhere and click "Reload" to apply all changes.
- Visit Your Deployed Site: Open a browser and go to <http://yourusername.pythonanywhere.com>. Verify that everything works as expected and test the main features of your application.

