Okay, here's a simple Django application structure that achieves this.

This example assumes you have Django installed (pip install Django) and know how to start a project (django-admin startproject myproject) and an app within it (cd myproject then python manage.py startapp greeter).

1. Project Structure:

myproject/

├── manage.py

├── myproject/

│ ├── \_\_init\_\_.py

│ ├── settings.py

│ ├── urls.py

│ ├── wsgi.py

│ └── asgi.py

└── greeter/ <-- Your new app

├── \_\_init\_\_.py

├── admin.py

├── apps.py

├── forms.py <-- Create this file

├── migrations/

│ └── \_\_init\_\_.py

├── models.py

├── tests.py

├── urls.py <-- Create this file

├── views.py <-- Edit this file

└── templates/ <-- Create this directory

└── greeter/ <-- Create this directory

└── greet.html <-- Create this file

2. greeter/forms.py:

Create this file to define the input form.

Python

# greeter/forms.py

from django import forms

class NameForm(forms.Form):

name = forms.CharField(label='Your name', max\_length=100, required=True)

3. greeter/views.py:

Modify this file to handle the logic.

Python

# greeter/views.py

from django.shortcuts import render

from .forms import NameForm

def greet\_user(request):

user\_name = None # Variable to hold the name if submitted

if request.method == 'POST':

# Create a form instance and populate it with data from the request:

form = NameForm(request.POST)

# Check whether it's valid:

if form.is\_valid():

# Process the data in form.cleaned\_data as required

user\_name = form.cleaned\_data['name']

# We don't necessarily need to redirect here for this simple case.

# We'll just render the same template but pass the user\_name.

# Create a new empty form instance for the template after successful POST

form = NameForm() # Show a fresh form below the greeting

else:

# If it's a GET (or any other method), create a blank form

form = NameForm()

# Prepare the context to pass to the template

context = {

'form': form,

'user\_name': user\_name,

}

# Render the template with the context

return render(request, 'greeter/greet.html', context)

4. greeter/templates/greeter/greet.html:

Create the template file to display the form and the greeting.

HTML

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Greeter</title>

</head>

<body>

<h1>Enter Your Name</h1>

<form method="post">

{% csrf\_token %} {# Important for security! #}

{{ form.as\_p }} {# Renders the form fields as paragraphs #}

<button type="submit">Submit</button>

</form>

<hr>

{% if user\_name %}

<h2>Hello {{ user\_name }}!</h2>

{% else %}

<p>Please enter your name above.</p>

{% endif %}

</body>

</html>

5. greeter/urls.py:

Create this file to define the URL pattern for the greeter app.

Python

# greeter/urls.py

from django.urls import path

from . import views

urlpatterns = [

# Maps the root URL of the app ('') to the greet\_user view

path('', views.greet\_user, name='greet\_user'),

]

6. myproject/urls.py:

Modify your project's urls.py file to include the URLs from the greeter app.

Python

# myproject/urls.py

from django.contrib import admin

from django.urls import path, include # Make sure 'include' is imported

urlpatterns = [

path('admin/', admin.site.urls),

path('', include('greeter.urls')), # Include greeter app's URLs at the root

]

7. myproject/settings.py:

Add your greeter app to the INSTALLED\_APPS list.

Python

# myproject/settings.py (Inside the file)

INSTALLED\_APPS = [

'django.contrib.admin',

'django.contrib.auth',

'django.contrib.contenttypes',

'django.contrib.sessions',

'django.contrib.messages',

'django.contrib.staticfiles',

# Add your app here:

'greeter.apps.GreeterConfig', # Or just 'greeter' often works too

]

# Make sure TEMPLATES setting is configured to find app templates

# The default settings usually handle this if 'APP\_DIRS': True is set

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [], # You could add project-level template dirs here

'APP\_DIRS': True, # This tells Django to look inside app 'templates' dirs

'OPTIONS': {

'context\_processors': [

'django.template.context\_processors.debug',

'django.template.context\_processors.request',

'django.contrib.auth.context\_processors.auth',

'django.contrib.messages.context\_processors.messages',

],

},

},

]

8. Run the Application:

Navigate to your project directory (myproject/) in the terminal.

Apply migrations (although we haven't created models, it's good practice):

Bash

python manage.py makemigrations

python manage.py migrate

Start the development server:

Bash

python manage.py runserver

Open your web browser and go to http://127.0.0.1:8000/.

You should see a page asking for your name. Enter a name and click submit, and the page will refresh, displaying the "Hello <name>!" message below the form.