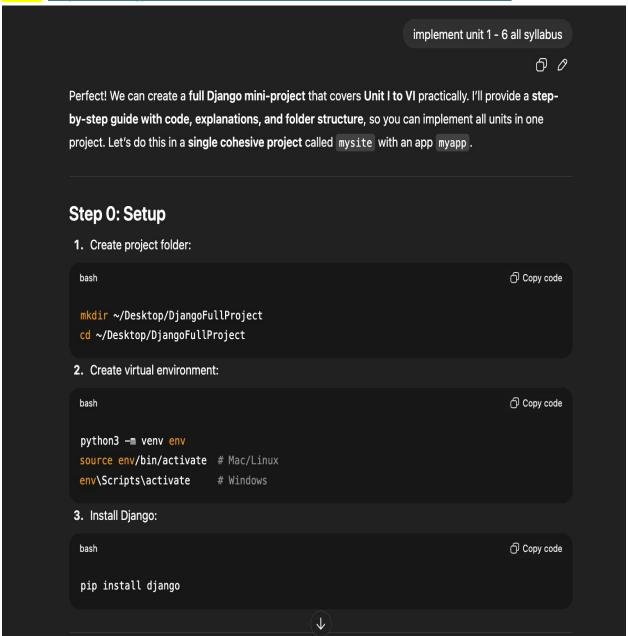
Viva: https://chatgpt.com/share/68c167a2-0ce0-800b-a776-86244e4bef12



Unit I - Project and App Setup 1. Create project bash django-admin startproject mysite cd mysite 2. Create app bash python manage.py startapp myapp 3. Add app to mysite/settings.py python installeD_APPS = ['myapp',]

Unit II - Views and URLs

1. Create simple views in myapp/views.py

```
python

from django.http import HttpResponse
from django.shortcuts import render

def home(request):
    return HttpResponse("Welcome to MySite Home Page!")

def about(request):
    return HttpResponse("This is the About Page.")

def greet_user(request, username):
    return HttpResponse(f"Hello, {username}!")
```

2. Map URLs in myapp/urls.py

```
python

from django.urls import path
from . import views

urlpatterns = [
   path('', views.home, name='home'),
   path('about/', views.about, name='about'),
   path('greet/<str:username>/', views.greet_user, name='greet_user'),
]
```

3. Include app URLs in mysite/urls.py



```
3. Include app URLs in mysite/urls.py
 python
                                                                             Copy code
  from django.contrib import admin
  from django.urls import path, include
 urlpatterns = [
     path('admin/', admin.site.urls),
     path('', include('myapp.urls')),
Unit III – Templates, Debugging and Testing
1. Create template folder
 arduino
                                                                             Copy code
 myapp/
  └─ templates/
     └─ myapp/
          - home.html
           — about.html
2. Sample home.html
```

2. Sample home.html html <!DOCTYPE html> <html> <head> <title>Home</title> </head> <body> <h1>Welcome, {{ username }}!</h1> {% for item in items %} {{ item }} {% endfor %} {% if show_footer %} <footer>MySite Footer</footer> {% endif %} </body> </html> 3. Update views.py to render templates

```
3. Update views.py to render templates
```

```
python

def home(request):
    items = ['Django', 'Python', 'Web Development']
    context = {'username': 'Sanuj', 'items': items, 'show_footer': True}
    return render(request, 'myapp/home.html', context)

def about(request):
    return render(request, 'myapp/about.html')
```

Unit IV - Forms in Django

1. Create form in myapp/forms.py

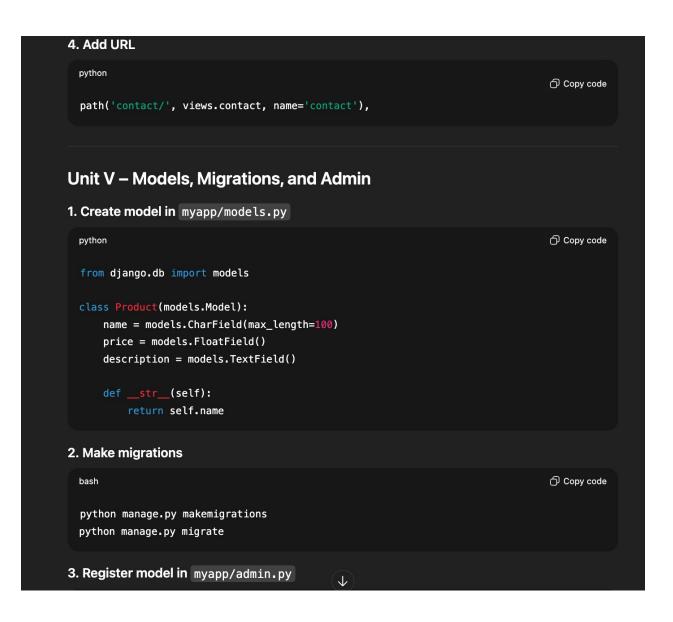
```
python

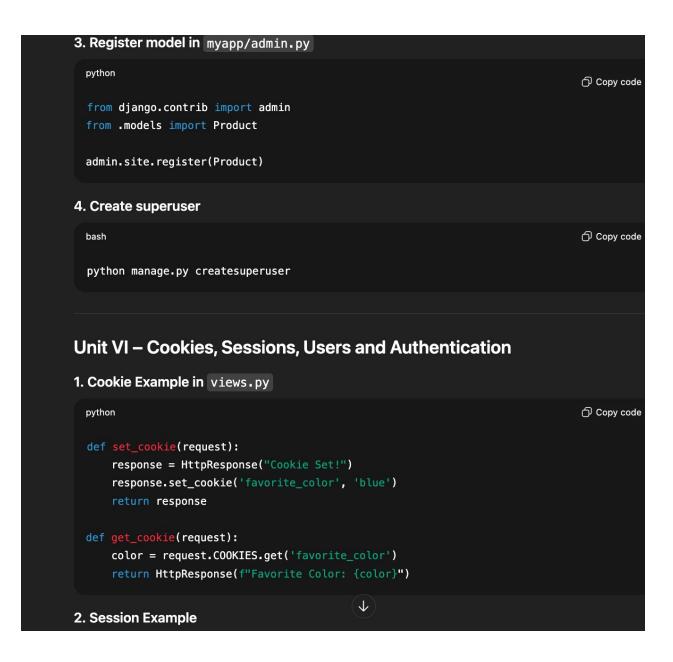
from django import forms

class ContactForm(forms.Form):
    name = forms.CharField(max_length=50)
    email = forms.EmailField()
    message = forms.CharField(widget=forms.Textarea)
```

2. Create form view in views.py

2. Create form view in views.py python Copy code from .forms import ContactForm from django.shortcuts import redirect def contact(request): if request.method == 'POST': form = ContactForm(request.POST) if form.is_valid(): print(form.cleaned_data) return redirect('home') form = ContactForm() return render(request, 'myapp/contact.html', {'form': form}) 3. contact.html template html Copy code <h1>Contact Us</h1> <form method="post"> {% csrf_token %} {{ form.as_p }} <button type="submit">Send</button> </form> 4. Add URL Copy code python



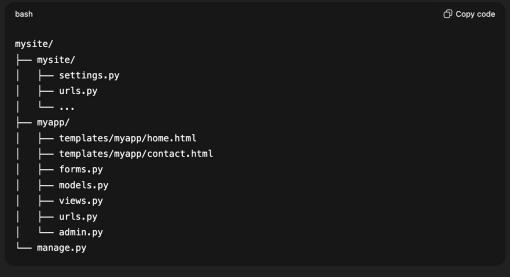


2. Session Example python def set_session(request): request.session['username'] = 'Sanuj' return HttpResponse("Session Set!") def get_session(request): username = request.session.get('username') return HttpResponse(f"Session Username: {username}") 3. User Authentication python from django.contrib.auth import authenticate, login, logout from django.contrib.auth.decorators import login_required def user_login(request): if request.method == 'POST': username = request.POST['username'] password = request.POST['password'] user = authenticate(request, username=username, password=password) if user is not None: login(request, user) return redirect('home') return render(request, 'myapp/login.html')

3. User Authentication python ○ Copy code from django.contrib.auth import authenticate, login, logout from django.contrib.auth.decorators import login_required def user_login(request): if request.method == 'POST': username = request.POST['username'] password = request.POST['password'] user = authenticate(request, username=username, password=password) if user is not None: login(request, user) return redirect('home') return render(request, 'myapp/login.html') @login_required def user_logout(request): logout(request) return redirect('home') **▼** Project Folder Structure ○ Copy code bash

> ├── settings.py ├── urls.py

Project Folder Structure



This one project covers all Units I-VI:

- Unit I: Project & App setup
- Unit II: Views & URLs, HTTP requests
- Unit III: Templates, loops, if-else, template inheritance
- Unit IV: Forms, GET/POST, CSRF
- Unit V: Models, Migrations, Admin
- Unit VI: Cookies, Sessions, User Auth

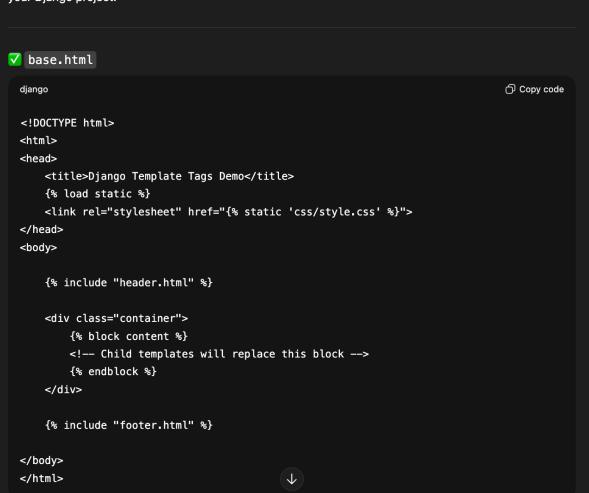
2. Template Tags

• **Definition**: Special instructions inside {% %} for logic, loops, conditions, inheritance, etc.

Common Template Tags

Тад	Purpose	Example
{% if %}	Conditional statements	{% if age >= 18 %} Adult {% else %} Minor {% endif %}
{% for %}	Loop over lists	<pre>{% for student in students %} {{ student.name }} {% endfor %}</pre>
{% block %}	Define blocks in base template	{% block content %} {% endblock %}
{% extends %}	Inherit from base template	{% extends "base.html" %}
{% include %}	Include another template	<pre>{% include "header.html" %}</pre>
{% url 'name' %}	Reverse URL mapping	Home
{% csrf_token %}	CSRF protection in forms	<form method="post">{% csrf_token %} </form>
{% empty %}	Executes if loop is empty	{% for i in items %}{{ i }}{% empty %}No Items{% endfor %}
{% comment %}	Add comments	{% comment %} This is a comment {% endcomment %}
{% static %}	Load static files	<pre></pre>

Perfect de You already listed the most common Django template tags with their purpose and examples. Let me now implement all of them together in a single working example so you can directly use it in your Django project.



```
√ index.html

                                                                          django
 {% extends "base.html" %}
 {% block content %}
 <h2>Common Template Tags Demo</h2>
 <!-- 1. if / else -->
 {% if age >= 18 %}
     You are an <strong>Adult</strong>.
     You are a <strong>Minor</strong>.
 {% endif %}
 <!-- 2. for loop + empty -->
 <h3>Student List:</h3>
 {% for student in students %}
     {{ forloop.counter }}. {{ student.name }}
 {% empty %}
     No students available.
 {% endfor %}
 <!-- 3. url tag -->
 Go back to <a href="{% url 'home' %}">Home</a>
 <!-- 4. csrf_token -->
 <form method="post">
     {% csrf_token %}
     <input type="text" name="username" placeholder="Enter Name">
     <button type="submit">Submit
 </form>
 <!-- 5. comment -->
 This is a comment and will not appear in the HTML output.
 {% endcomment %}
 <!-- 6. static tag -->
 <img src="{% static 'images/logo.png' %}" alt="Logo" width="150">
 {% endblock %}
```

```
<!-- 6. static tag -->
 <img src="{% static 'images/logo.png' %}" alt="Logo" width="150">
 {% endblock %}

√ header.html

 html
                                                                              Copy code
 <header>
     <h1>Welcome to My Django App</h1>
         <a href="{% url 'home' %}">Home</a> |
         <a href="{% url 'about' %}">About</a>
     </nav>
 </header>

√ footer.html

 html
                                                                              Copy code
 <footer>
     © 2025 My Django Project
 </footer>

√ views.py
```

```
√ views.py

 python

    ○ Copy code

 from django.shortcuts import render
 def home(request):
     students = [
         {"name": "Alice"},
         {"name": "Bob"},
         {"name": "Charlie"},
     context = {
         "age": 20,
         "students": students
     return render(request, "index.html", context)
 def about(request):
     return render(request, "about.html")
Covered Tags:
• {% if %} / {% else %} → Conditional rendering
• {% for %} / {% empty %} → Loop with empty fallback
• {% block %} → Content placeholder in base template
• {% extends %} → Inheriting base.html
• {% include %} → Reuse header/footer templates
• {% url %} → Reverse URL resolution
• {% csrf_token %} → Security in forms
• {% comment %} → Template comments
```

• {% static %} → Load static files

Django Project Folder Layout Explained

```
pgsql
                                                                   FirstProject/
 – venv/
                  ← Virtual Environment (optional but recommended)
 — first/
                ← Project folder (settings & configurations)
   — __init__.py ← Marks directory as a Python package
                ← Entry point for ASGI servers (async deployment)
   - asgi.py

— settings.py ← Project settings (database, apps, middleware, etc.)

   — urls.py
                  ← Project-level URL mappings (routes)
   └─ wsgi.py
                  ← Entry point for WSGI servers (deployment)
                  ← Application folder (your actual app logic)
  - myapp/
   ├─ __init__.py ← Marks directory as Python package
   ├─ admin.py ← Register models for Django Admin
   — apps.py
                  ← App configuration file
   └─ __init__.py
     — models.py
                  ← Database models (tables as Python classes)
                 ← Unit tests for your app
     — tests.py
                ← Business logic (functions/classes returning responses)
   └─ views.py
                  ← Command—line utility to runserver, migrations, etc.
  manage.py
```