**Task 4: Create a simple HTML template**

**Step 1:** Create a templates directory for bookmodule app, by doing the following:

> mkdir apps/templates/bookmodule

**Step 2:** Update main/settings.py with template location

**import os**

**...**

**BASE\_DIR = Path(\_\_file\_\_).resolve().parent.parent**

**TEMPLATE\_DIR = os.path.join(BASE\_DIR, "apps" + os.sep + "templates")**

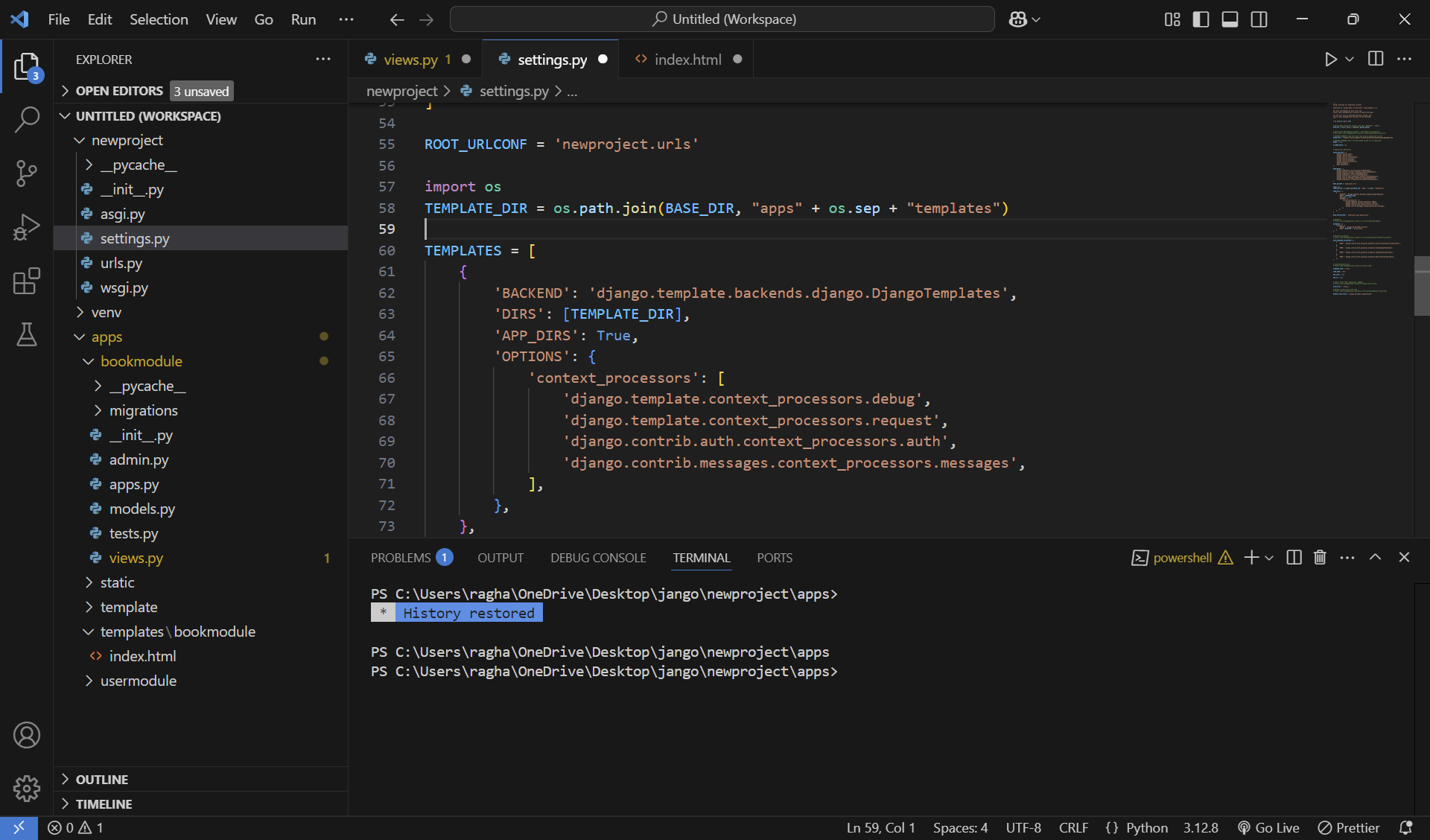
**TEMPLATES = [{**

**...**

**'DIRS': [TEMPLATE\_DIR],**

**...**

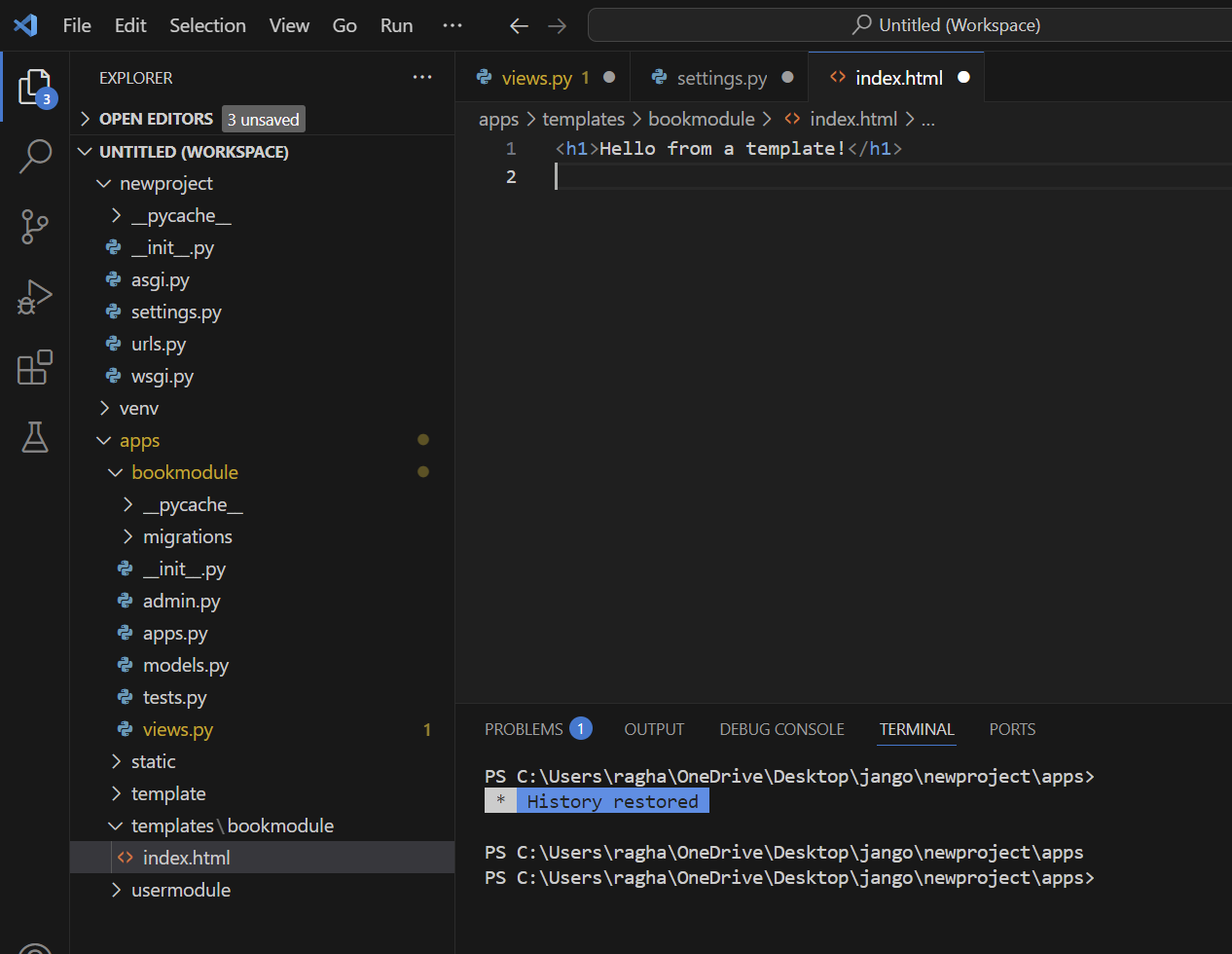
**}]**



Note that many options are available to tell Django how to find templates, which can be set in the TEMPLATES setting. The easiest one is to create a templates directory inside the bookmodule directory. Django will look in this (and in other apps’ templates directories) because APP\_DIRS is True in the settings.py file. Anyway, this task is intended to utilize the one template inside the apps folder and make it accessible to all apps.

**Step 3:** Create a file with the name “index.html” within the templates directory ‘apps/templates/bookmodule’, and add the following html content:

<h1>Hello from a template!</h1>



**Step 4:** Open views.py in the bookmodule directory. We no longer manually create HttpResponse, so you may remove the HttpResponse import line and add this line after the other existing imports:

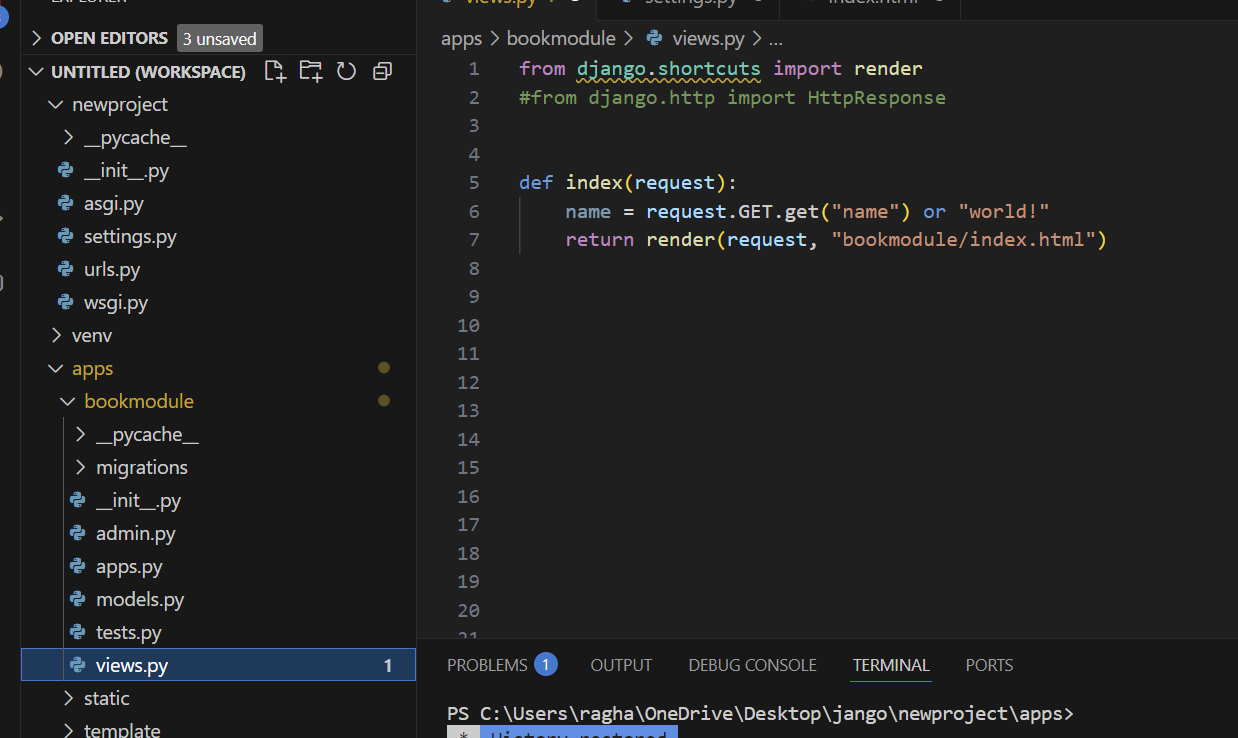
from django.shortcuts import render

**Step 5:** Update the index function, in apps/bookmodule/views.py, so that, instead of returning HttpResponse, it’s returning a call to render, passing in request and the template name:

def index(request):

name = request.GET.get("name") or "world!"

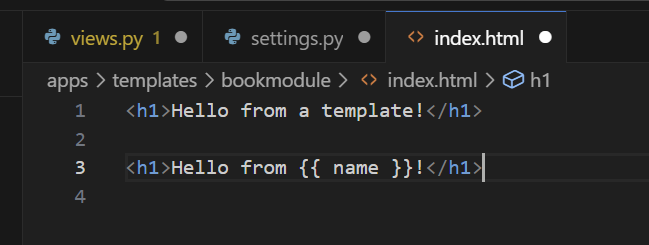
return render(request, "bookmodule/index.html") #Change HttpResponse to render function



**Task 5: Rendering variables in the HTML template that processes a context**

**Step 1:** open the file “index.html”, and update the file so that it contains a place to render the name variable:

<h1>Hello from {{ name }}!</h1>

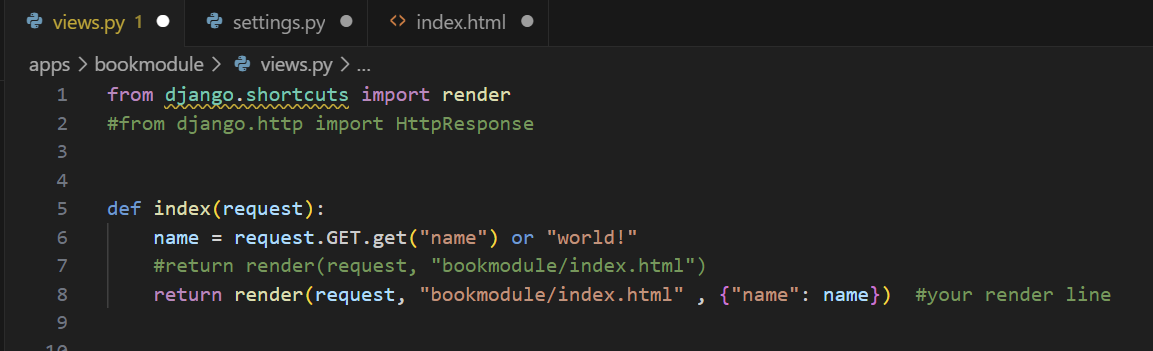


**Step 2:** Update the index function, in apps/bookmodule/views.py, and add the context dictionary as the third argument to the render function. Change your render line to this:

def index(request):

name = request.GET.get("name") or "world!"

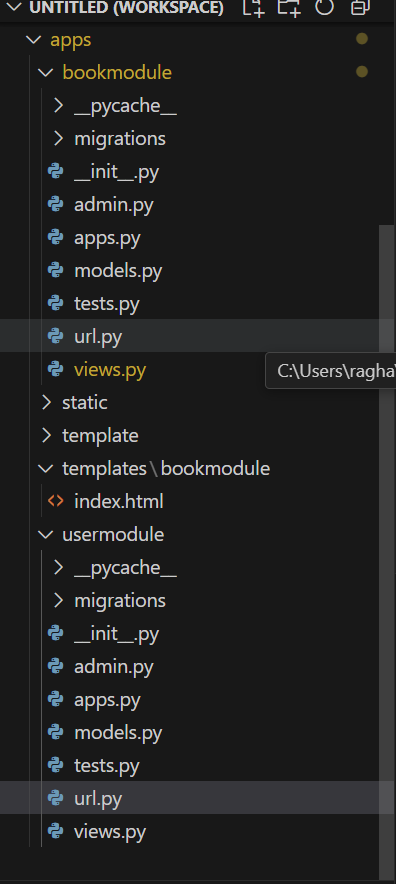
return render(request, "bookmodule/index.html" , {"name": name}) #your render line



**Task 6: Define patterns globally (DjangoProjects/urls.py) with specific urls.py file for each app/module**

Note that it is common to keep one URL configuration per application in your Django project. Here, we will create a separate URL configuration for each app and add it to our project-level URL configuration.

**Step 1:** Create a new urls.py file for each app module(bookmodule and usermodule) in the same folder as the views.py file.



**Step 2:** move urls of bookmodule to apps/bookmodule/urls.py, and include newly created urls.py in core/urls.py using include module.

|  |
| --- |
| DjangoProjects/urls.py |
| from django.contrib import admin  from django.urls import include, path  urlpatterns = [  path('admin/', admin.site.urls),  path('books/', include("apps.bookmodule.urls")), #include urls.py of bookmodule app  path('users/', include("apps.usermodule.urls")) #include urls.py of usermodule app  ] |
| apps/bookmodule/urls.py |
| from django.urls import path  from . import views  urlpatterns = [  path('', views.index),  path('index2/<int:val1>/', views.index2)  ] |
| apps/usermodule/urls.py |
| from django.urls import path  urlpatterns = [  ] |

**Step 3:** Now from the browser navigate to the following urls:

<http://127.0.0.1:8000/books/>. You should see “Hello, world!”

A screenshot of a computer

AI-generated content may be incorrect.

<http://127.0.0.1:8000/books/?name=world>!. You should see “Hello, world!”

A screenshot of a computer

AI-generated content may be incorrect.

<http://127.0.0.1:8000/books/?name=Dr>. Faisal. You should see “Hello, Dr. Faisal”

A screenshot of a computer

AI-generated content may be incorrect.

<http://127.0.0.1:8000/books/?name=Mousa>. You should see “Hello, Mousa”

A screenshot of a computer

AI-generated content may be incorrect.

http://127.0.0.1:8000/books/index2/3/.. You should see “value1 = 3”

A screenshot of a computer

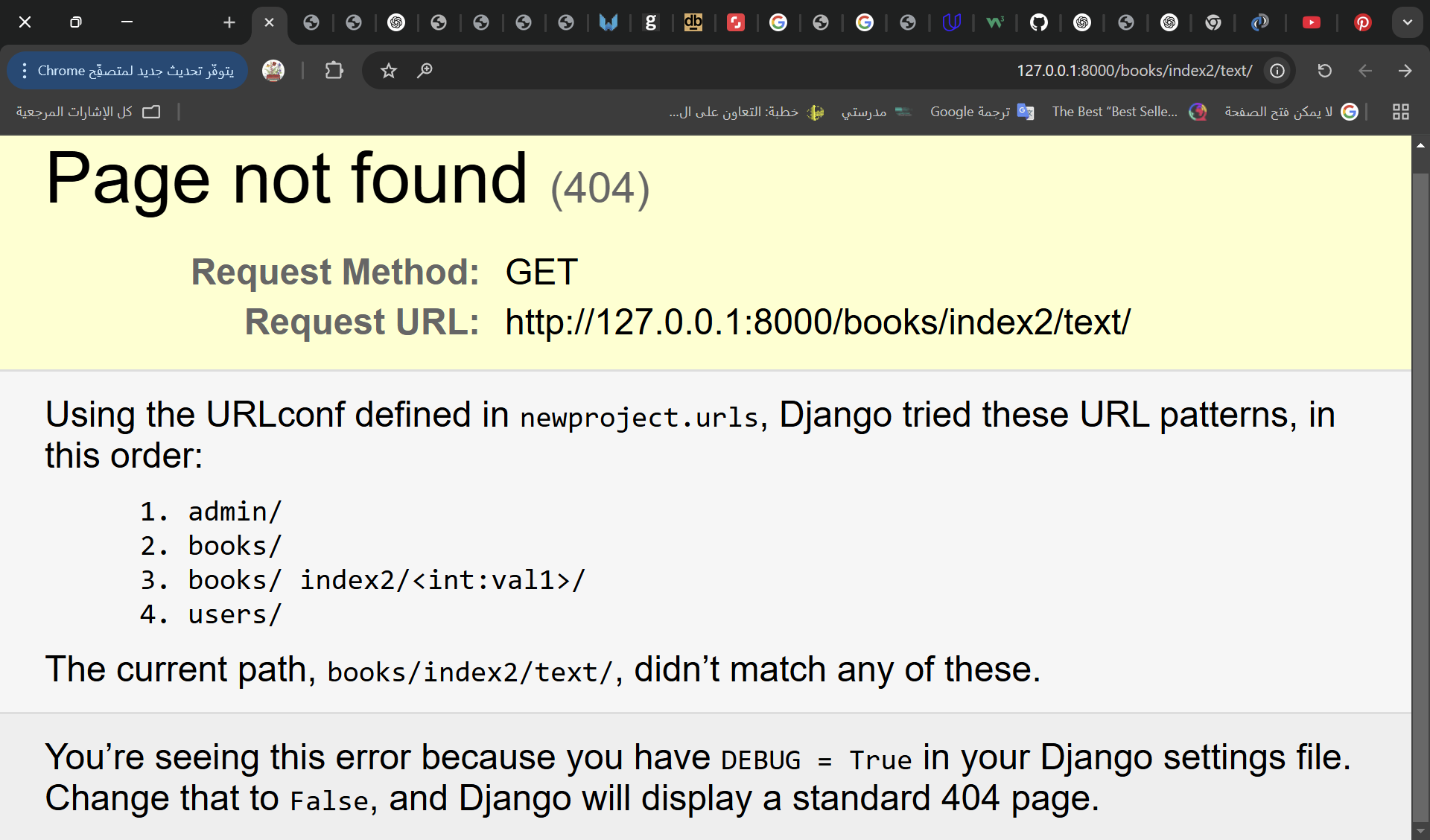
AI-generated content may be incorrect.

http://127.0.0.1:8000/books/index2/10/. You should see “value1 = 10”

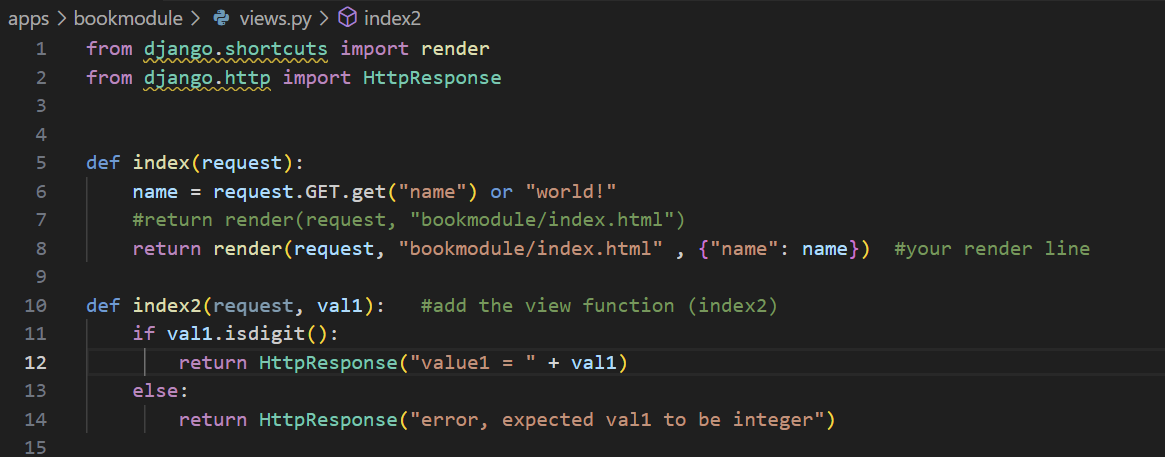
A screenshot of a computer

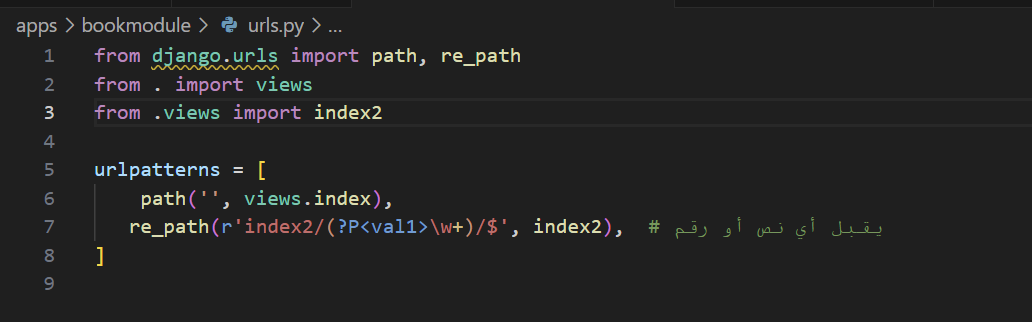
AI-generated content may be incorrect.

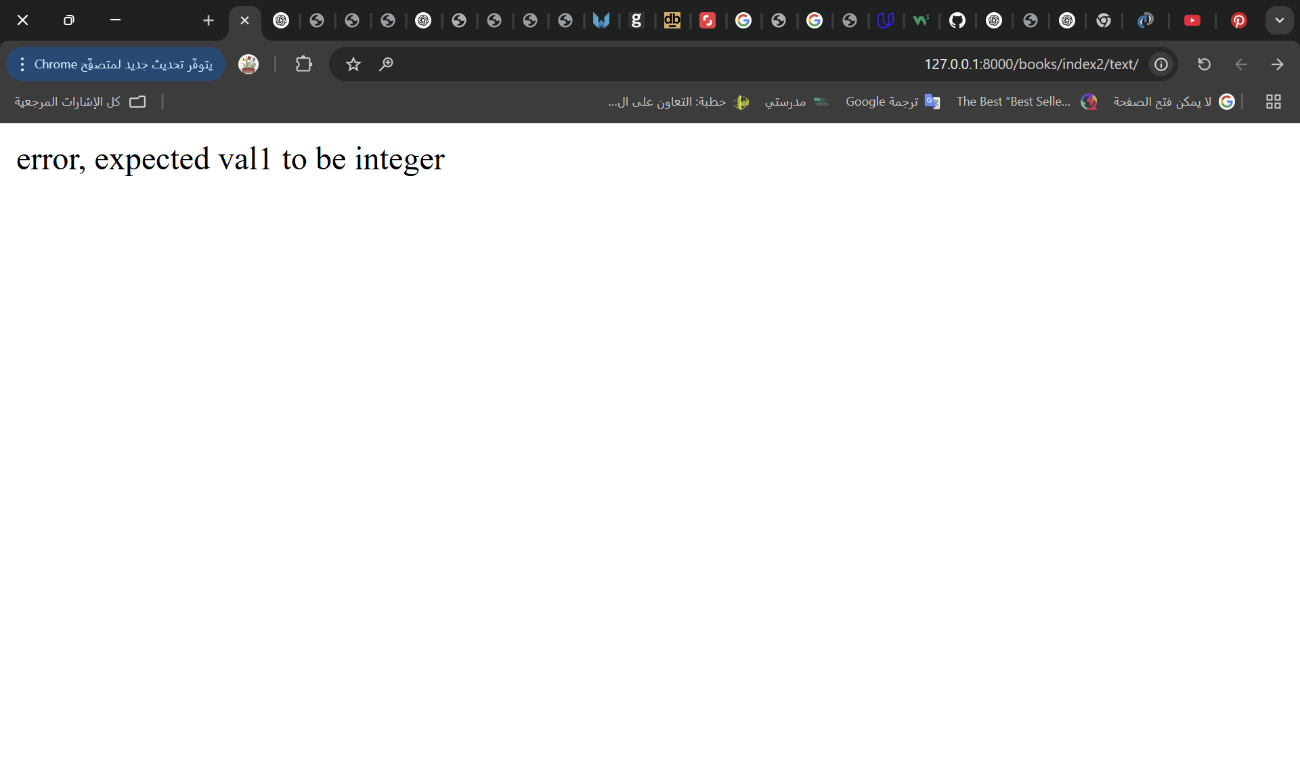
http://127.0.0.1:8000/books/index2/text/. You should see “error, expected val1 to be integer”



urls و views عدلت على







**Task7: Create a URL, view, and HTML to display one book details**

**Step 1:** Create a new function at the location: ’root/apps/bookmodule/views.py’ that accept an ID argument:

**def viewbook(request, bookId):**

**# assume that we have the following books somewhere (e.g. database)**

**book1 = {'id':123, 'title':'Continuous Delivery', 'author':'J. Humble and D. Farley'}**

**book2 = {'id':456, 'title':'Secrets of Reverse Engineering', 'author':'E. Eilam'}**

**targetBook = None**

**if book1['id'] == bookId: targetBook = book1**

**if book2['id'] == bookId: targetBook = book2**

**context = {'book':targetBook} # book is the variable name accessible by the template**

**return render(request, 'bookmodule/show.html', context)**

**Step 2:** In ’root/apps/bookmodule/urls.py’, add the following path function to urlpatterns:

**path('<int:bookId>', views.viewbook)**

**Step 3:** Create the template at the location: ‘root/apps/templates/bookmodule/show.html’ and put some code in it.

<h4>ID: {{ book.id }}</h4>

<h4>Title: {{ book.title }}</h4>

<h4>Authors</h4>

<p>{{ book.author }}</p>

<h4>Description</h4>

<p>Description here</p>

**Step 4:** Now from the browser navigate to the following urls:

<http://127.0.0.1:8000/books/123>

A screenshot of a computer

AI-generated content may be incorrect.

<http://127.0.0.1:8000/books/456>

A screenshot of a computer

AI-generated content may be incorrect.

**Additional content:** To redirect to another view or URL, use the function redirect, and for details and examples visit the following link: <https://docs.djangoproject.com/en/5.1/topics/http/shortcuts/#redirect>