WP Lab Week 6

Q1.Develop a simple web page to perform basic arithmetic operations. Take two

integer inputs from the user, select the operation to be performed using a

dropdown. Include a button “Calculate” to perform the selected operation, and

then display the result in the same web page.

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Calculator</title>

</head>

<body>

<h1>Simple Calculator</h1>

<form method="post">

{% csrf\_token %}

<label for="num1">Number 1:</label>

<input type="number" name="num1" required>

<label for="num2">Number 2:</label>

<input type="number" name="num2" required>

<label for="operation">Operation:</label>

<select name="operation">

<option value="add">Add</option>

<option value="subtract">Subtract</option>

<option value="multiply">Multiply</option>

<option value="divide">Divide</option>

</select>

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

</form>

{% if result is not none %}

<h2>Result: {{ result }}</h2>

{% endif %}

</body>

</html>

**views.py**

# calculator/views.py

from django.shortcuts import render

def calculator\_view(request):

if request.method == "POST":

num1 = float(request.POST['num1'])

num2 = float(request.POST['num2'])

operation = request.POST['operation']

if operation == 'add':

result = num1 + num2

elif operation == 'subtract':

result = num1 - num2

elif operation == 'multiply':

result = num1 \* num2

elif operation == 'divide':

if num2 != 0:

result = num1 / num2

else:

result = 'Cannot divide by zero'

return render(request, 'calculator/index.html', {'result': result})

return render(request, 'calculator/index.html')

**calculator/urls.py**

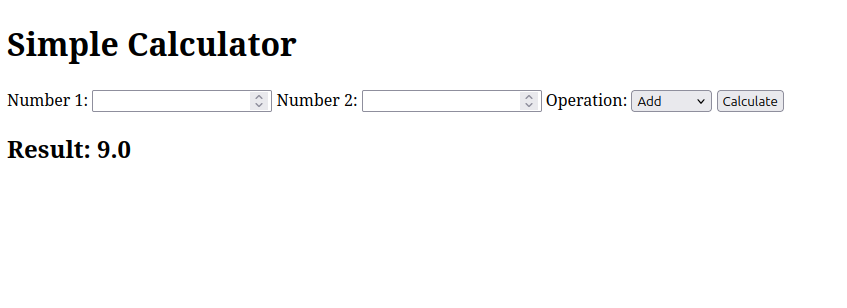
from django.urls import path

from . import views

urlpatterns = [

path('', views.calculator\_view, name='calculator'),

]



Q2. Develop a simple web form that generates the front cover for a magazine. The

form should provide the options for selecting the image, background color,

changing font size, color etc. Input messages must be taken from the user so

as to display it on the front cover with legible font family and font size. The

front cover developed should be proportionate to the web page size. Place the

css files inside static folder.

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Magazine Cover Generator</title>

<!-- Add this line to load the static files -->

{% load static %}

<link rel="stylesheet" href="{% static 'frontcover/styles.css' %}">

</head>

<body>

<h1>Generate Your Magazine Cover</h1>

<form method="post" enctype="multipart/form-data">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Generate Cover</button>

</form>

{% if form.is\_valid %}

<h2>Your Magazine Cover</h2>

<img src="{{ cover\_url }}" alt="Magazine Cover" />

{% endif %}

</body>

</html>

**styles.css**

body {

font-family: Arial, sans-serif;

background-color: #f4f4f4;

text-align: center;

padding: 20px;

}

h1 {

color: #333;

}

form {

background-color: #fff;

padding: 20px;

border-radius: 10px;

box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);

margin-top: 20px;

}

form input, form select {

margin: 10px 0;

}

button {

padding: 10px 20px;

background-color: #5cb85c;

border: none;

color: white;

font-size: 16px;

cursor: pointer;

border-radius: 5px;

}

button:hover {

background-color: #4cae4c;

}

**forms.py**

from django import forms

class MagazineForm(forms.Form):

title = forms.CharField(max\_length=100, label='Magazine Title')

subtitle = forms.CharField(max\_length=100, label='Subtitle')

image = forms.ImageField(label='Upload Image')

bg\_color = forms.CharField(max\_length=7, label='Background Color (Hex)', required=False)

font\_size = forms.IntegerField(label='Font Size', initial=30)

font\_color = forms.CharField(max\_length=7, label='Font Color (Hex)', required=False)

**views.py**

# frontcover/views.py

from django.shortcuts import render

from .forms import MagazineForm

from PIL import Image, ImageDraw, ImageFont

import io

from django.http import HttpResponse

def magazine\_cover(request):

if request.method == "POST":

form = MagazineForm(request.POST, request.FILES)

if form.is\_valid():

title = form.cleaned\_data['title']

subtitle = form.cleaned\_data['subtitle']

image = form.cleaned\_data['image']

bg\_color = form.cleaned\_data['bg\_color'] or '#FFFFFF' # Default white if not provided

font\_size = form.cleaned\_data['font\_size']

font\_color = form.cleaned\_data['font\_color'] or '#000000' # Default black if not provided

# Load the uploaded image

img = Image.open(image)

img = img.resize((800, 1200)) # Resize the image to fit the cover proportionally

# Create a drawing object

draw = ImageDraw.Draw(img)

# Load font (use a built-in font for simplicity)

font\_path = '/home/student/Desktop/wp\_lab5/lab6/frontcover/static/frontcover/FreeMono.ttf' # Replace with the actual font path

font = ImageFont.truetype(font\_path, font\_size)

# Add text to the image

draw.text((50, 50), title, font=font, fill=font\_color)

draw.text((50, 150), subtitle, font=font, fill=font\_color)

# Convert the image to a file-like object

img\_byte\_arr = io.BytesIO()

img.save(img\_byte\_arr, format='PNG')

img\_byte\_arr.seek(0)

# Return the image as a response

return HttpResponse(img\_byte\_arr, content\_type='image/png')

else:

form = MagazineForm()

return render(request, 'frontcover/index.html', {'form': form})

**frontcover/urls.py**

# frontcover/urls.py

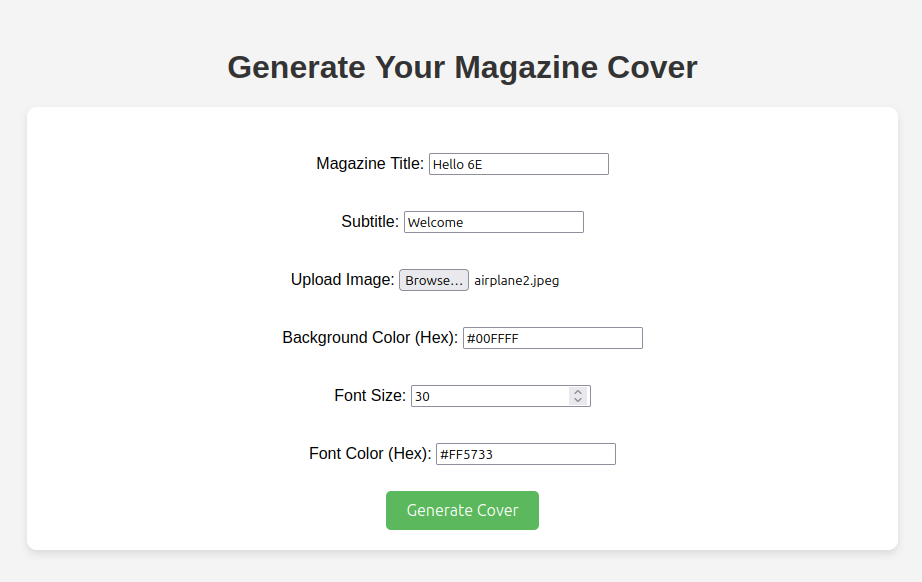
from django.urls import path

from . import views

urlpatterns = [

path('', views.magazine\_cover, name='magazine\_cover'), # This should render the cover

]





Q3.Design a simple web application to provide information about a book. The

home page of the application should display the cover page of the book along

with three hyperlinks: Metadata, Reviews, Publisher info. Give provision to

revert to home page from any other page.

**home.html**

<!-- book/templates/book/home.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>{{ book.title }}</title>

{% load static %}

<link rel="stylesheet" href="{% static 'book/styles.css' %}">

</head>

<body>

<div class="container">

<h1>{{ book.title }}</h1>

<h2>By {{ book.author }}</h2>

<!-- book/templates/book/home.html -->

<img src="{% static 'book/bookcover.jpeg' %}" alt="Cover Image" width="200px">

<div class="links">

<a href="{% url 'metadata' %}">Metadata</a> |

<a href="{% url 'reviews' %}">Reviews</a> |

<a href="{% url 'publisher\_info' %}">Publisher Info</a>

</div>

</div>

</body>

</html>

**metadata.html**

<!-- book/templates/book/metadata.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Metadata - {{ book.title }}</title>

{% load static %}

<link rel="stylesheet" href="{% static 'book/styles.css' %}">

</head>

<body>

<div class="container">

<h1>Metadata: {{ book.title }}</h1>

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

<a href="{% url 'home' %}">Back to Home</a>

</div>

</body>

</html>

**publisher\_info.html**

<!-- book/templates/book/publisher\_info.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Publisher Info - {{ book.title }}</title>

{% load static %}

<link rel="stylesheet" href="{% static 'book/styles.css' %}">

</head>

<body>

<div class="container">

<h1>Publisher Info for {{ book.title }}</h1>

<p>{{ book.publisher\_info }}</p>

<a href="{% url 'home' %}">Back to Home</a>

</div>

</body>

</html>

**reviews.html**

<!-- book/templates/book/reviews.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Reviews - {{ book.title }}</title>

{% load static %}

<link rel="stylesheet" href="{% static 'book/styles.css' %}">

</head>

<body>

<div class="container">

<h1>Reviews for {{ book.title }}</h1>

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

<a href="{% url 'home' %}">Back to Home</a>

</div>

</body>

</html>

**styles.css**

/\* book/static/book/styles.css \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

background-color: #f4f4f4;

}

.container {

max-width: 900px;

margin: 20px auto;

background-color: white;

padding: 20px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

}

h1, h2 {

text-align: center;

}

img {

display: block;

margin: 0 auto;

}

.links {

text-align: center;

margin-top: 20px;

}

.links a {

text-decoration: none;

color: #007BFF;

font-weight: bold;

margin: 0 15px;

}

.links a:hover {

text-decoration: underline;

}

**views.py**

from django.shortcuts import render

book\_info = {

'title': 'The Great Gatsby',

'author': 'F. Scott Fitzgerald',

'cover\_image': '/home/student/Downloads/bookcover.jpeg', # Use a real path here or database if storing

'metadata': 'Published in 1925, The Great Gatsby is widely regarded as one of the greatest American novels.',

'reviews': 'The Great Gatsby is an extraordinary work, examining the American Dream.',

'publisher\_info': 'Scribner is the publisher of The Great Gatsby, a division of Simon & Schuster.'

}

def home(request):

return render(request, 'book/home.html', {'book': book\_info})

def metadata(request):

return render(request, 'book/metadata.html', {'book': book\_info})

def reviews(request):

return render(request, 'book/reviews.html', {'book': book\_info})

def publisher\_info(request):

return render(request, 'book/publisher\_info.html', {'book': book\_info})

**book/urls.py**

from django.urls import path

from . import views

urlpatterns = [

path('', views.home, name='home'), # Home page

path('metadata/', views.metadata, name='metadata'),

path('reviews/', views.reviews, name='reviews'),

path('publisher/', views.publisher\_info, name='publisher\_info'),

]

