Lab 06

Q1.

views.py

from django.shortcuts import render

from .forms import ArithmeticForm

def calculate(request):

result = None

if request.method == 'POST':

form = ArithmeticForm(request.POST)

if form.is\_valid():

num1 = form.cleaned\_data['number1']

num2 = form.cleaned\_data['number2']

operation = form.cleaned\_data['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

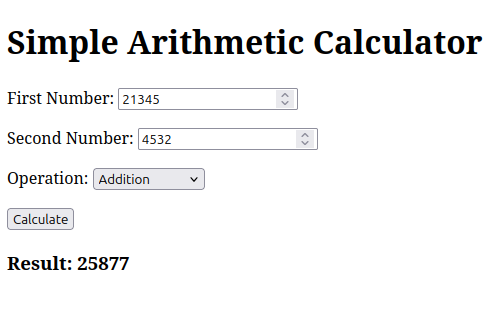
else:

result = 'Error: Division by zero'

else:

form = ArithmeticForm()

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



Q2.

views.py

from django.shortcuts import render

from .forms import MagazineCoverForm

from PIL import Image

from PIL import ImageDraw, ImageFont

import io

from django.core.files.storage import default\_storage

import base64

def create\_magazine\_cover(request):

result\_image = None # To hold the image data for displaying on the template

if request.method == 'POST':

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

if form.is\_valid():

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

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

text\_color = form.cleaned\_data['text\_color']

bg\_color = form.cleaned\_data['bg\_color']

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

image\_file = form.cleaned\_data['image'] # The uploaded image

# Create the base image (for simplicity, a 600x800 px cover)

base\_image = Image.new('RGB', (600, 800), color=bg\_color)

# Draw text on the image (simple method for demonstration)

draw = ImageDraw.Draw(base\_image)

try:

font = ImageFont.truetype("arial.ttf", font\_size)

except IOError:

font = ImageFont.load\_default()

title\_position = (30, 50)

subtitle\_position = (30, 150)

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

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

# Add the uploaded image

uploaded\_image = Image.open(image\_file)

uploaded\_image = uploaded\_image.resize((550, 400)) # Resize to fit cover

base\_image.paste(uploaded\_image, (25, 250))

# Save the final image to a BytesIO object for display

buffer = io.BytesIO()

base\_image.save(buffer, format="PNG")

buffer.seek(0)

# Create a URL or base64 encoded string of the image

result\_image = base64.b64encode(buffer.read()).decode('utf-8') # Convert image to base64 string

else:

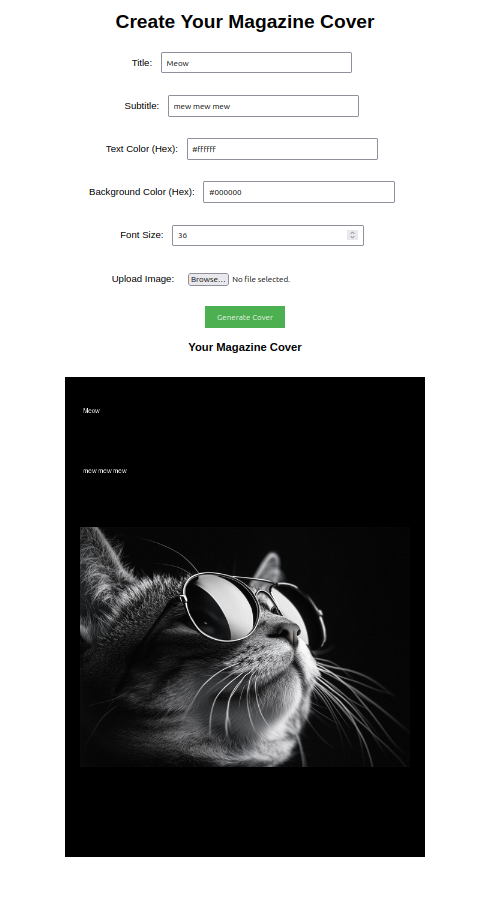
form = MagazineCoverForm()

return render(request, 'magazine/magazine\_cover.html', {

'form': form,

'result\_image': result\_image

})



Q3.

home.html

<html>

<head>

<title>Book Information</title>

{% load static %}

</head>

<body>

<h1>Book Title</h1>

<img src="{% static 'book\_cover.jpg' %}" alt="Book Cover" width="200">

<ul>

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

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

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

</ul>

</body>

</html>

metadata.html

<html>

<head>

<title>Book Information</title>

{% load static %}

</head>

<body>

<h1>Book Title</h1>

<img src="{% static 'book\_cover.jpg' %}" alt="Book Cover" width="200">

<ul>

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

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

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

</ul>

</body>

</html>

reviews.html

<html>

<head>

<title>Reviews</title>

{% load static %}

</head>

<body>

<h1>Book Reviews</h1>

<p>"A great book!" - Reviewer 1</p>

<p>"Really engaging story!" - Reviewer 2</p>

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

</body>

</html>

publisher\_info.html

<html>

<head>

<title>Publisher Info</title>

{% load static %}

</head>

<body>

<h1>Publisher Information</h1>

<p>Publisher: XYZ Publishers</p>

<p>Location: New York, USA</p>

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

</body>

</html>

