WP LAB 6 – Developing Web Application using Django - Part II

Name: Pranamya G Kulal
Class: CSE A
Reg no: 220905018
Roll no: 80

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.

i)arithmetic_app/urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
  path('admin/', admin.site.urls),
  path('', include('calculator.urls')),
]
```

ii) calculator/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
  path('', views.calculate, name='calculate'),
]
```

iii) calculator/forms.py

```
from django import forms

class ArithmeticForm(forms.Form):
  OPERATION_CHOICES = [
    ('add', 'Add'),
    ('subtract', 'Subtract'),
    ('multiply', 'Multiply'),
    ('divide', 'Divide'),
  ]

  num1 = forms.IntegerField(label="First Number")
  num2 = forms.IntegerField(label="Second Number")
  operation = forms.ChoiceField(choices=OPERATION_CHOICES, label="Operation")
```

iv) calculator/forms.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['num1']
            num2 = form.cleaned_data['num2']
            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 = "Cannot divide by zero"
    else:
        form = ArithmeticForm()

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

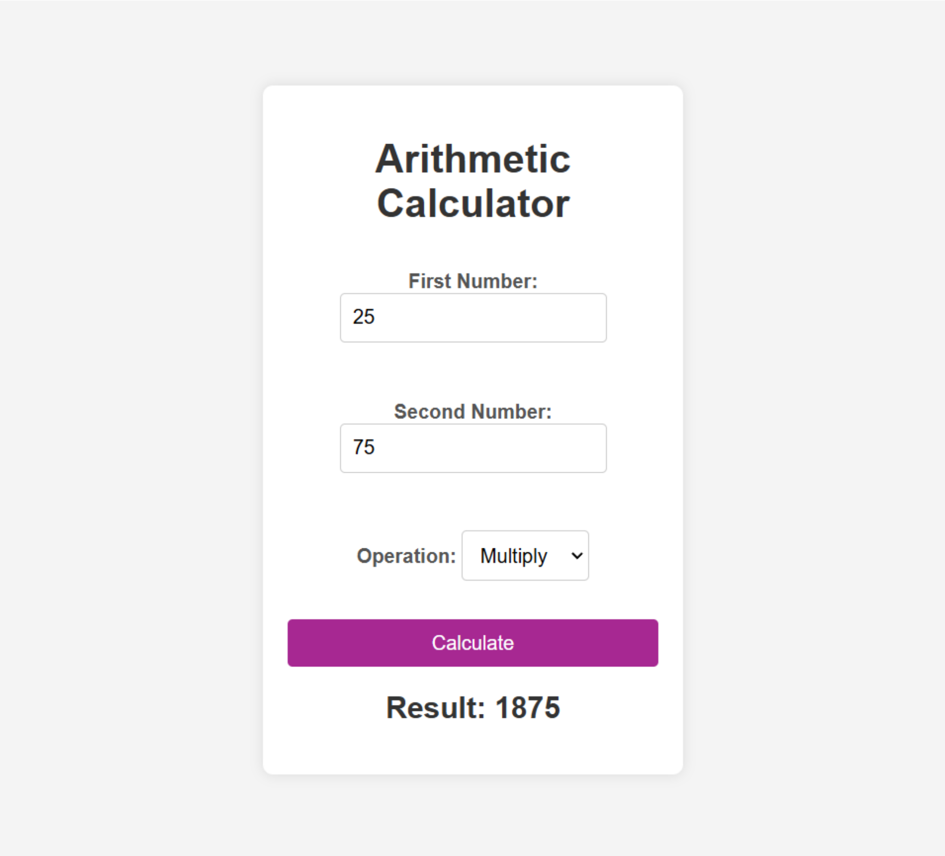
v) calculator/templates/calculator/calculate.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Arithmetic Calculator</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 0;
            display: flex;
            justify-content: center;
            align-items: center;
            height: 100vh;
        }

        .calculator-container {
            background-color: #fff;
```

```
  padding: 20px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
  width: 300px;
  text-align: center;
}

h1 {
  margin-bottom: 20px;
  color: #333;
}

form {
  display: flex;
  flex-direction: column;
}

label {
  margin-bottom: 5px;
  font-weight: bold;
  color: #555;
}

input[type="number"], select {
  padding: 10px;
  margin-bottom: 15px;
  border: 1px solid #ccc;
  border-radius: 4px;
  font-size: 16px;
}

button {
  padding: 10px;
  background-color: #7f1f6f;
  color: white;
  border: none;
  border-radius: 4px;
  font-size: 16px;
  cursor: pointer;
}

button:hover {
  background-color: #590d60;
}

h2 {
  margin-top: 20px;
```

```
      color: #333;
    }
  </style>
</head>
<body>
  <div class="calculator-container">
    <h1>Arithmetic Calculator</h1>
    <form method="post">
      {% csrf_token %}
      {{ form.as_p }}
      <button type="submit">Calculate</button>
    </form>

    {% if result is not None %}
      <h2>Result: {{ result }}</h2>
    {% endif %}
  </div>
</body>
</html>
```

<u>vi) Screenshot</u>



Arithmetic Calculator

First Number: 25

Second Number: 75

Operation: Multiply

Result: 1875

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.

i)magazine_cover/urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('cover.urls')),
]
```

ii)cover/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.cover_generator, name='cover_generator'),
]
```

iii) cover_forms.py

```
from django import forms

class CoverForm(forms.Form):
    BACKGROUND_COLORS = [
        ('#FFFFFF', 'White'),
        ('#000000', 'Black'),
        ('#FF0000', 'Red'),
        ('#00FF00', 'Green'),
        ('#0000FF', 'Blue'),
    ]

    FONT_FAMILIES = [
        ('Arial', 'Arial'),
        ('Times New Roman', 'Times New Roman'),
        ('Courier New', 'Courier New'),
        ('Verdana', 'Verdana'),
    ]
    image_choices = [
        ('magazine.jpg', 'Cover 1'),
        ('magazine.jpg', 'Cover 2'),
    ]
    image = forms.ChoiceField(choices=image_choices, label='Select Cover Image')
    background_color = forms.ChoiceField(choices=BACKGROUND_COLORS, label="Background Color")
    font_family = forms.ChoiceField(choices=FONT_FAMILIES, label="Font Family")
    font_size = forms.IntegerField(label="Font Size", min_value=10, max_value=100)
    font_color = forms.CharField(label="Font Color", widget=forms.TextInput(attrs={'type': 'color'}))
    message = forms.CharField(label="Message", widget=forms.Textarea)
```

iv) cover/views.py

```
from django.shortcuts import render
from .forms import CoverForm

def cover_generator(request):
    if request.method == 'POST':
        form = CoverForm(request.POST, request.FILES)
        if form.is_valid():
            # Process the form data
            context = {
                'image': form.cleaned_data['image'],
                'background_color': form.cleaned_data['background_color'],
                'font_family': form.cleaned_data['font_family'],
                'font_size': form.cleaned_data['font_size'],
                'font_color': form.cleaned_data['font_color'],
                'message': form.cleaned_data['message'],
            }
            return render(request, 'cover/cover.html', context)
    else:
        form = CoverForm()
    return render(request, 'cover/cover_generator.html', {'form': form})
```

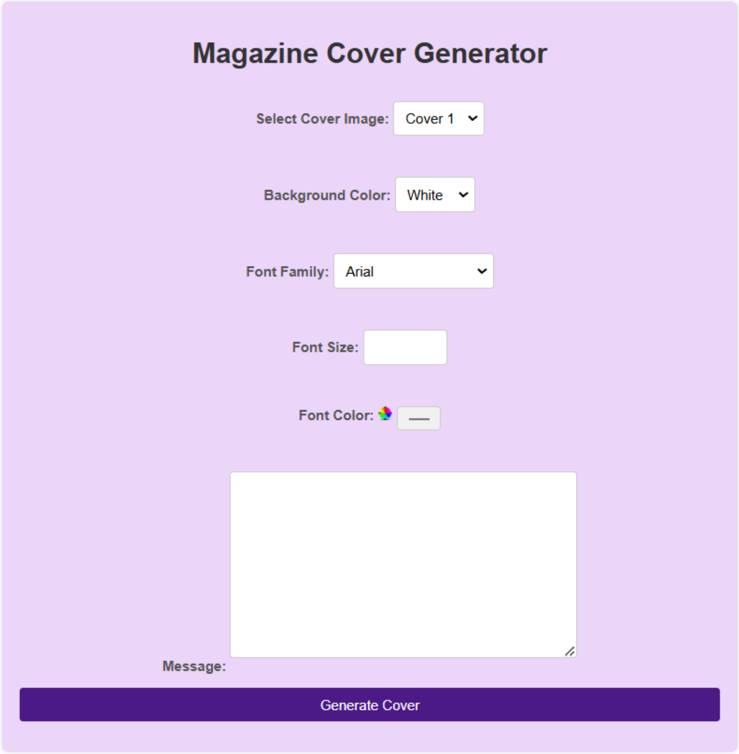
v) cover/templates/cover_generator.html

```
<!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>
    {% load static %}
    <link rel="stylesheet" href="{% static 'cover/styles.css' %}">
</head>
<body>
    <div class="container">
        <h1>Magazine Cover Generator</h1>
        <form method="post" enctype="multipart/form-data">
            {% csrf_token %}
            {{ form.as_p }}
            <button type="submit">Generate Cover</button>
        </form>
    </div>
</body>
</html>
```

vi) cover/templates/cover.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Generated Cover</title>
    {% load static %}
    <link rel="stylesheet" href="{% static 'cover/styles.css' %}">
    <style>
        .cover {
            width: 80%;
            height: 80vh;
            margin: 0 auto;
            background-color: {{ background_color }};
            background-image: url("{% static 'images/'|add:image %}");
            background-size: cover;
            background-position: center;
            display: flex;
            justify-content: center;
            align-items: center;
            position: relative~;
        }
        .message {
            font-family: {{ font_family }};
            font-size: {{ font_size }}px;
            color: {{ font_color }};
            text-align: center;
        }
    </style>
</head>
<body>
    <div class="container">
        <h1>Your Magazine Cover</h1>
        <div class="cover">
            <div class="message">{{ message }}</div>
        </div>
        <a href="{% url 'cover_generator' %}">Create Another Cover</a>
    </div>
</body>
</html>
```

vii) Screenshots



Magazine Cover Generator

Select Cover Image: Cover 1

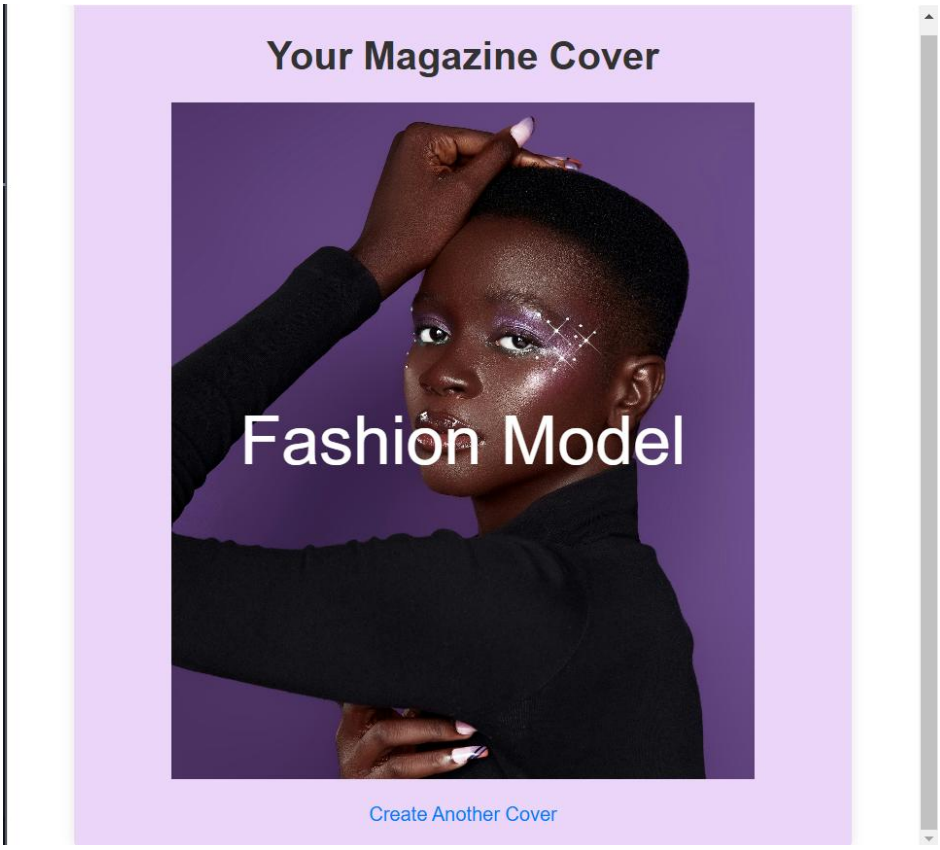
Background Color: White

Font Family: Arial

Font Size:

Font Color:

Message:



Your Magazine Cover

Create Another Cover