

LAB ASSIGNMENTS

NAME: AYANIKA PAUL

BATCH: D1

ROLL NO. 22

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.

CODE:

q1/settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'calculator',
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [
            BASE_DIR / 'templates',
        ],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]
```

q1/urls.py

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

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

calculator/urls.py

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

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

calculator/urls.py

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

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

calculator/views.py

```
from django.shortcuts import render
```

```
def calculate(request):
    result = None
    if request.method == "POST":
        try:
            num1 = int(request.POST['num1'])
            num2 = int(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'
except ValueError:
result = 'Invalid input'

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

calculator/templates/calculate.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Simple Calculator</title>
<style>
body {
font-family: Arial, sans-serif;
background-color: #f4f4f9;
margin: 0;
padding: 0;
display: flex;
justify-content: center;
align-items: center;
height: 100vh;
background-color: rgb(87, 124, 204);
}

.calculator-container {
background-color: #fff;
border-radius: 10px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
padding: 30px;
width: 479px;
text-align: center;
}

h1 {
font-size: 24px;
margin-bottom: 20px;
color: #333;
}

label {
font-size: 16px;
margin-top: 10px;
display: block;
text-align: left;
color: #555;
}
```

```
input[type="text"], select {
width: 100%;
padding: 10px;
margin-top: 5px;
border: 1px solid #ccc;
border-radius: 5px;
font-size: 16px;
}

button {
background-color: #4CAF50;
color: white;
border: none;
padding: 10px 20px;
border-radius: 5px;
cursor: pointer;
font-size: 16px;
margin-top: 20px;
}

button:hover {
background-color: #45a049;
}

h3 {
margin-top: 20px;
font-size: 18px;
color: #333;
}

.error {
color: red;
font-size: 16px;
}

</style>
</head>
<body>
<div class="calculator-container">
<h1>Simple Calculator</h1>

<form method="POST">
{% csrf_token %}
<label for="num1">Number 1:</label>
<input type="text" name="num1" id="num1" required>

<label for="num2">Number 2:</label>
<input type="text" name="num2" id="num2" required>

<label for="operation">Operation:</label>
<select name="operation" id="operation" required>
```

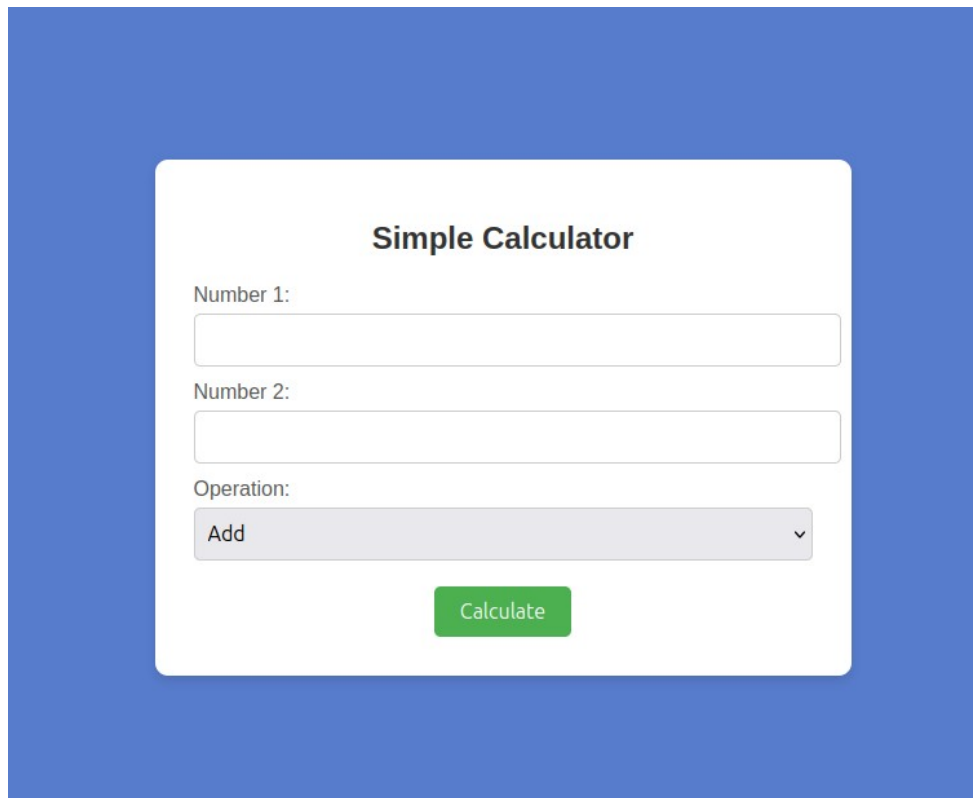
```
<option value="add">Add</option>
<option value="subtract">Subtract</option>
<option value="multiply">Multiply</option>
<option value="divide">Divide</option>
</select>

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

{% if result is not none %}
<h3>Result: {{ result }}</h3>
{% endif %}

{% if result == 'Invalid input' %}
<p class="error">Please enter valid numbers!</p>
{% endif %}
</div>
</body>
</html>
```

OUTPUT:

A screenshot of a web application titled "Simple Calculator". The interface is set against a solid blue background. A white, rounded rectangular form is centered on the page. At the top of the form, the title "Simple Calculator" is displayed in a bold, black font. Below the title, there are three input fields. The first is labeled "Number 1:" and the second is labeled "Number 2:", both in a standard black font. Each label is followed by a white rectangular input box with a thin grey border. Below these two input fields is a third field labeled "Operation:". This field is a dropdown menu with a light grey background, showing "Add" as the selected option and a small downward-pointing chevron icon on the right. At the bottom center of the form is a green rectangular button with the word "Calculate" written in white text.

Simple Calculator

Number 1:

2

Number 2:

4

Operation:

Add

Calculate

Result: 6

Simple Calculator

Number 1:

2

Number 2:

4

Operation:

Multiply

Calculate

Result: 8

Simple Calculator

Number 1:

8

Number 2:

4

Operation:

Divide

Calculate

Result: 2.0

Simple Calculator

Number 1:

10

Number 2:

4

Operation:

Subtract

Calculate

Result: 6

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.

CODE:

q2/settings.py

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'magazine_cover',
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [BASE_DIR / 'templates'],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]

STATIC_URL = '/static/'

STATICFILES_DIRS = [
    BASE_DIR / "static",
]

MEDIA_URL = '/media/'
MEDIA_ROOT = BASE_DIR / 'media'
```

q2/urls.py

```
from django.contrib import admin
from django.urls import path, include
from django.shortcuts import redirect
from django.conf import settings
from django.conf.urls.static import static

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

```
path('cover/', include('magazine_cover.urls')),
path('', lambda request: redirect('cover/')),
]
```

```
if settings.DEBUG:
urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

magazine_cover/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
path('', views.generate_cover, name='form'),
]
```

magazine_cover/views.py

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

```
def generate_cover(request):
if request.method == 'POST':
```

```
form = MagazineCoverForm(request.POST, request.FILES)
if form.is_valid():
```

```
title = form.cleaned_data['title']
image = form.cleaned_data['image']
bg_color = form.cleaned_data['bg_color']
font_color = form.cleaned_data['font_color']
```

```
return render(request, 'magazine_cover/result.html', {
'title': title,
'image': image,
'bg_color': bg_color,
'font_color': font_color,
})
```

```
else:
```

```
form = MagazineCoverForm()
```

```
return render(request, 'magazine_cover/form.html', {'form': form})
```

magazine_cover/templates/form.html

```
<!DOCTYPE html>
<html lang="en">
<head>
```



```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Form Page</title>
{% load static %}
<link rel="stylesheet" href="{% static 'css/form.css' %}">
</head>
<body>
<div class="form-container">
<h2>Magazine Cover Form</h2>
<form action="" method="POST" enctype="multipart/form-data">
{% csrf_token %}
<div class="form-group">
<label for="title">Title</label>
<input type="text" id="title" name="title" placeholder="Enter title"
required>
</div>
<div class="form-group">
<label for="image">Image</label>
<input type="file" id="image" name="image" accept="image/*" required>
</div>
<div class="form-group">
<label for="bg_color">Background Color</label>
<input type="color" id="bg_color" name="bg_color">
</div>
<div class="form-group">
<label for="font_color">Font Color</label>
<input type="color" id="font_color" name="font_color">
</div>
<button type="submit" class="submit-btn">Generate Cover</button>
</form>
</div>
</body>
</html>

```

magazine_cover/templates/result.html

```

{% load static %}
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>{{ title }} - Magazine Cover</title>
<link rel="stylesheet" href="{% static 'css/result.css' %}">
</head>
<body>
<div class="cover-container" style="background-color: {{ bg_color }};">
<div class="cover-content">
<h1 style="color: {{ font_color }};">{{ title }}</h1>

```

```


</div>
</div>
</body>
</html>
```

static/css/form.css

```
body {
font-family: Arial, sans-serif;
background-color: #f4f4f4;
margin: 0;
padding: 0;
}

.form-container {
width: 100%;
max-width: 600px;
margin: 50px auto;
background-color: #fff;
padding: 20px;
box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
border-radius: 8px;
}

h2 {
text-align: center;
font-size: 24px;
color: #333;
}

.form-group {
margin-bottom: 15px;
}

label {
font-size: 16px;
color: #333;
margin-bottom: 5px;
display: block;
}

input[type="text"],
input[type="file"],
input[type="color"] {
width: 100%;
padding: 10px;
font-size: 14px;
border: 1px solid #ccc;
```

```
border-radius: 4px;
}

button.submit-btn {
width: 100%;
padding: 12px;
background-color: #333;
color: white;
border: none;
border-radius: 4px;
font-size: 16px;
cursor: pointer;
}

button.submit-btn:hover {
background-color: #555;
}
```

static/css/result.css

```
/* static/css/result.css */
body {
font-family: Arial, sans-serif;
background-color: #fafafa;
margin: 0;
padding: 0;
}

.result-container {
width: 100%;
max-width: 800px;
margin: 50px auto;
background-color: #fff;
padding: 20px;
box-shadow: 0 0 15px rgba(0, 0, 0, 0.1);
border-radius: 8px;
}

.cover-container{
background-color: #9a9996;
height: 100vh;
width: 682px;
margin-left: 147px;
text-align: center;
}

.cover-image {
max-width: 100%;
height: 500px;
border-radius: 10px;
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
```

```
display: block;
margin: 0 auto;
max-height: 500px;
}
h2 {
text-align: center;
font-size: 28px;
color: #333;
}
```

```
.result-info {
margin-top: 20px;
font-size: 18px;
color: #555;
}
```

OUTPUT:

Magazine Cover Form

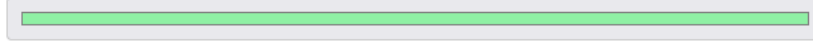
Title

EDUCATION

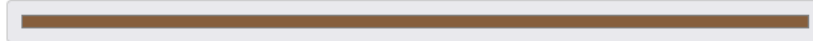
Image

Browse... coverpage.png

Background Color



Font Color



Generate Cover

EDUCATION

