

Lab 06 – Parallel Programming

Arhaan Girdhar | 220962050

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.

Views.py

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

def arithmetic_view(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':
                result = num1 / num2 if num2 != 0 else 'Error: Division by zero'
        else:
            form = ArithmeticForm()
    context = {'form': form, 'result': result}
    return render(request, 'calculator/index.html', context)
```

Forms.py

```
from django import forms

OPERATION_CHOICES = [
    ('add', 'Addition'),
    ('subtract', 'Subtraction'),
    ('multiply', 'Multiplication'),
    ('divide', 'Division'),
]

class ArithmeticForm(forms.Form):
    number1 = forms.IntegerField(label="Number 1")
    number2 = forms.IntegerField(label="Number 2")
    operation = forms.ChoiceField(choices=OPERATION_CHOICES, label="Operation")
```

Urls.py

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

urlpatterns = [
    path("", arithmetic_view, name='arithmetic'),
]
```

Apps.py

```
from django.apps import AppConfig

class CalculatorConfig(AppConfig):
    name = 'calculator'
```

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>
  <style>
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
      font-family: 'Inter', sans-serif;
    }

    body {
      background: bisque;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      padding: 20px;
    }

    .calculator-container {
      background: #fff;
      width: 360px;
      padding: 30px;
      border-radius: 20px;
      display: flex;
      flex-direction: column;
      justify-content: space-between;
      box-shadow: 0px 10px 30px rgba(0, 0, 0, 0.2);
      transition: 0.3s ease-in-out;
    }

    h2 {
```

```
    text-align: center;
    margin-bottom: 20px;
    color: #333;
    font-size: 1.8em;
    font-weight: bold;
}

.form-group {
    margin-bottom: 20px;
}

label {
    display: block;
    margin-bottom: 6px;
    color: #444;
    font-size: 1.1em;
    font-weight: 600;
}

input, select {
    width: 100%;
    padding: 14px;
    border: 1px solid #ccc;
    border-radius: 10px;
    font-size: 1.1em;
    outline: none;
    background: #f9f9f9;
    transition: all 0.3s ease-in-out;
}

input:focus, select:focus {
    border-color: #ff9800;
    background: #fff;
}

.btn-calculate {
    width: 100%;
    padding: 14px;
    background: #ff9800;
    color: rgb(12, 1, 1);
    border: none;
    border-radius: 10px;
    font-size: 1.2em;
    font-weight: bold;
    cursor: pointer;
    transition: all 0.3s ease-in-out;
    margin-top: 10px;
}

.btn-calculate:hover {
    background: #e68900;
}

.result {
    margin-top: 20px;
    font-size: 1.6em;
    text-align: center;
    color: #222;
    background: #fff3e0;
    padding: 18px;
    border-radius: 12px;
    font-weight: bold;
    border: 1px solid #ff9800;
```

```

    }

    @media (max-width: 400px) {
        .calculator-container {
            width: 100%;
            padding: 20px;
        }
    }
</style>
</head>
<body>
    <div class="calculator-container">
        <h2>Calculator</h2>
        <form method="post">
            {% csrf_token %}
            <div class="form-group">
                <label for="id_number1">Enter First Number</label>
                {{ form.number1 }}
            </div>
            <div class="form-group">
                <label for="id_number2">Enter Second Number</label>
                {{ form.number2 }}
            </div>
            <div class="form-group">
                <label for="id_operation">Select Operation</label>
                {{ form.operation }}
            </div>
            <button type="submit" class="btn-calculate">Calculate</button>
        </form>
        {% if result is not None %}
            <div class="result">Result: {{ result }}</div>
        {% endif %}
    </div>
</body>
</html>

```

Output

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000'. The page content is a calculator application. It features a white card with a light orange background. The card has a title 'Calculator' and three input fields: 'Enter First Number' (containing '99'), 'Enter Second Number' (containing '1'), and 'Select Operation' (a dropdown menu showing 'Addition'). Below these fields is an orange 'Calculate' button. At the bottom of the card, a box displays 'Result: 100'.

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.

Views.py

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

def index(request):
    form = MagazineCoverForm(request.POST or None, request.FILES or None)
    cover = None
    if request.method == 'POST' and form.is_valid():
        cover = form.cleaned_data
    return render(request, 'magazine_app/index.html', {'form': form, 'cover': cover})
```

Forms.py

```
from django import forms

class MagazineCoverForm(forms.Form):
    image = forms.ImageField(required=False, label="Select Cover Image")
    background_color = forms.CharField(max_length=7, required=True, label="Background Color",
    widget=forms.TextInput(attrs={'type': 'color'}))
    font_color = forms.CharField(max_length=7, required=True, label="Font Color",
    widget=forms.TextInput(attrs={'type': 'color'}))
    font_size = forms.IntegerField(required=True, min_value=12, max_value=72, label="Font Size (px)")
    font_family = forms.ChoiceField(choices=[('Arial', 'Arial'), ('Times New Roman', 'Times New Roman'), ('Helvetica',
    'Helvetica')], label="Font Family")
    title = forms.CharField(max_length=100, required=True, label="Magazine Title")
    tagline = forms.CharField(max_length=150, required=False, label="Tagline")
```

Tests.py

```
from django.test import TestCase
```

Urls.py

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

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

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Magazine Cover Generator</title>
  <link rel="stylesheet" href="css/styles.css">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <div class="container py-4">
    <div class="row">
      <div class="col-md-6">
        <form id="coverForm" method="post" enctype="multipart/form-data" class="card p-4 shadow-sm">
          <p>
            <label for="image">Select Cover Image</label><br>
            <input type="file" name="image" id="image">
          </p>
          <p>
            <label for="background_color">Background Color</label><br>
            <input type="color" name="background_color" id="background_color" value="#ffffff">
          </p>
          <p>
            <label for="font_color">Font Color</label><br>
            <input type="color" name="font_color" id="font_color" value="#000000">
          </p>
          <p>
            <label for="font_size">Font Size (px)</label><br>
            <input type="number" name="font_size" id="font_size" min="12" max="72" value="24">
          </p>
          <p>
            <label for="font_family">Font Family</label><br>
            <select name="font_family" id="font_family">
              <option value="Arial">Arial</option>
              <option value="Times New Roman">Times New Roman</option>
              <option value="Helvetica">Helvetica</option>
            </select>
          </p>
          <p>
            <label for="title">Magazine Title</label><br>
            <input type="text" name="title" id="title" placeholder="Enter magazine title">
          </p>
          <p>
            <label for="tagline">Tagline</label><br>
            <input type="text" name="tagline" id="tagline" placeholder="Enter tagline">
          </p>
          <button type="submit" class="btn btn-primary w-100">Generate Cover</button>
        </form>
      </div>
      <div class="col-md-6">
        <div class="card shadow-sm" id="coverPreview" style="background-color: #ffffff;">
          
          <div class="card-body text-center">
            <h2 id="coverTitle" style="font-size: 24px; color: #000000; font-family: Arial;">Magazine Title</h2>
            <p id="coverTagline" style="font-size: 19px; color: #000000;">Tagline goes here</p>
          </div>
        </div>
      </div>
    </div>
  </div>
</body>
</html>
```

```

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
<script>
const form = document.getElementById('coverForm');
const coverPreview = document.getElementById('coverPreview');
const coverImage = document.getElementById('coverImage');
const coverTitle = document.getElementById('coverTitle');
const coverTagline = document.getElementById('coverTagline');

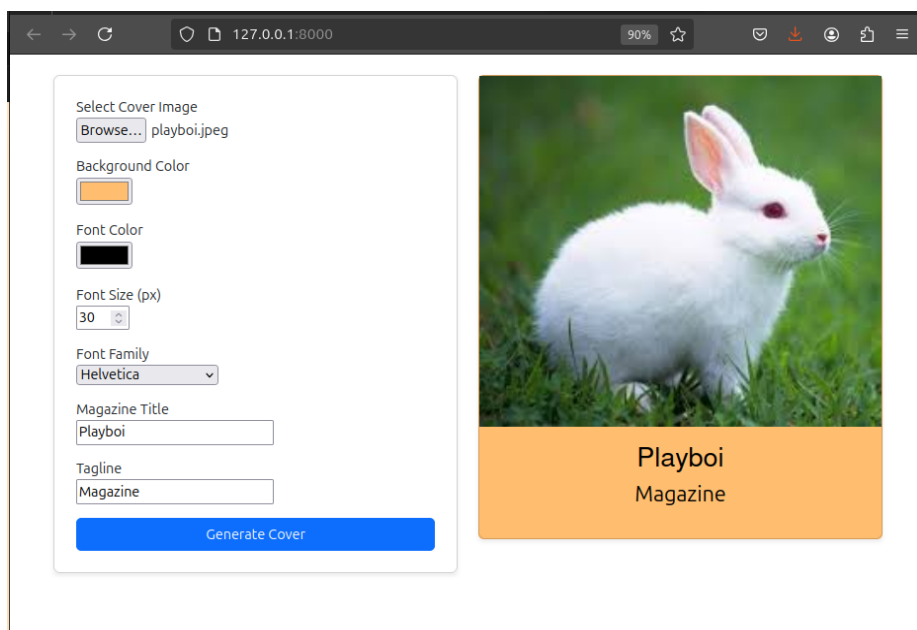
form.addEventListener('submit', function(e) {
  e.preventDefault();
  const bgColor = document.getElementById('background_color').value;
  const fontColor = document.getElementById('font_color').value;
  const fontSize = document.getElementById('font_size').value;
  const fontFamily = document.getElementById('font_family').value;
  const title = document.getElementById('title').value;
  const tagline = document.getElementById('tagline').value;

  coverPreview.style.backgroundColor = bgColor;
  coverTitle.style.fontSize = fontSize + 'px';
  coverTitle.style.color = fontColor;
  coverTitle.style.fontFamily = fontFamily;
  coverTitle.textContent = title;
  coverTagline.style.fontSize = (fontSize * 0.8) + 'px';
  coverTagline.style.color = fontColor;
  coverTagline.textContent = tagline;

  const imageInput = document.getElementById('image');
  if (imageInput.files && imageInput.files[0]) {
    const reader = new FileReader();
    reader.onload = function(e) {
      coverImage.src = e.target.result;
    }
    reader.readAsDataURL(imageInput.files[0]);
  } else {
    coverImage.src = "images/default.jpg";
  }
});
</script>
</body>
</html>

```

Output



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.

Views.py

```
from django.shortcuts import render

def index(request):
    return render(request, 'book_app/index.html')

def metadata(request):
    return render(request, 'book_app/metadata.html')

def reviews(request):
    return render(request, 'book_app/reviews.html')

def publisher(request):
    return render(request, 'book_app/publisher.html')
```

Urls.py

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

urlpatterns = [
    path("", views.index, name='home'),
    path('metadata/', views.metadata, name='metadata'),
    path('reviews/', views.reviews, name='reviews'),
    path('publisher/', views.publisher, name='publisher'),
]
```

Apps.py

```
from django.apps import AppConfig

class BookAppConfig(AppConfig):
    default_auto_field = 'django.db.models.BigAutoField'
    name = 'book_app'
```

Settings.py

```
import os
from pathlib import Path

BASE_DIR = Path(__file__).resolve().parent.parent
SECRET_KEY = 'replace-with-your-secret-key'
DEBUG = True
ALLOWED_HOSTS = []

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



```

'django.contrib.staticfiles',
'book_app',
]

MIDDLEWARE = [
'django.middleware.security.SecurityMiddleware',
'django.contrib.sessions.middleware.SessionMiddleware',
'django.middleware.common.CommonMiddleware',
'django.middleware.csrf.CsrfViewMiddleware',
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'book_project.urls'

TEMPLATES = [
{
'BACKEND': 'django.template.backends.django.DjangoTemplates',
'DIRS': [],
'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',
],
},
},
]

WSGI_APPLICATION = 'book_project.wsgi.application'

DATABASES = {
'default': {
'ENGINE': 'django.db.backends.sqlite3',
'NAME': BASE_DIR / 'db.sqlite3',
}
}

AUTH_PASSWORD_VALIDATORS = [
{'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator'},
{'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator'},
{'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator'},
{'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator'},
]

LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_L10N = True
USE_TZ = True

STATIC_URL = '/static/'
STATICFILES_DIRS = [BASE_DIR / 'book_app' / 'static']

```

Index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Book Home</title>
  <link rel="stylesheet" href="/static/css/style.css">
</head>
<body>
  <div class="container">
    <h1>Rich Dad Poor Dad</h1>
    
    <nav>
      <ul>
        <li><a href="/metadata/">Metadata</a></li>
        <li><a href="/reviews/">Reviews</a></li>
        <li><a href="/publisher/">Publisher Info</a></li>
      </ul>
    </nav>
  </div>
</body>
</html>
```

Output

