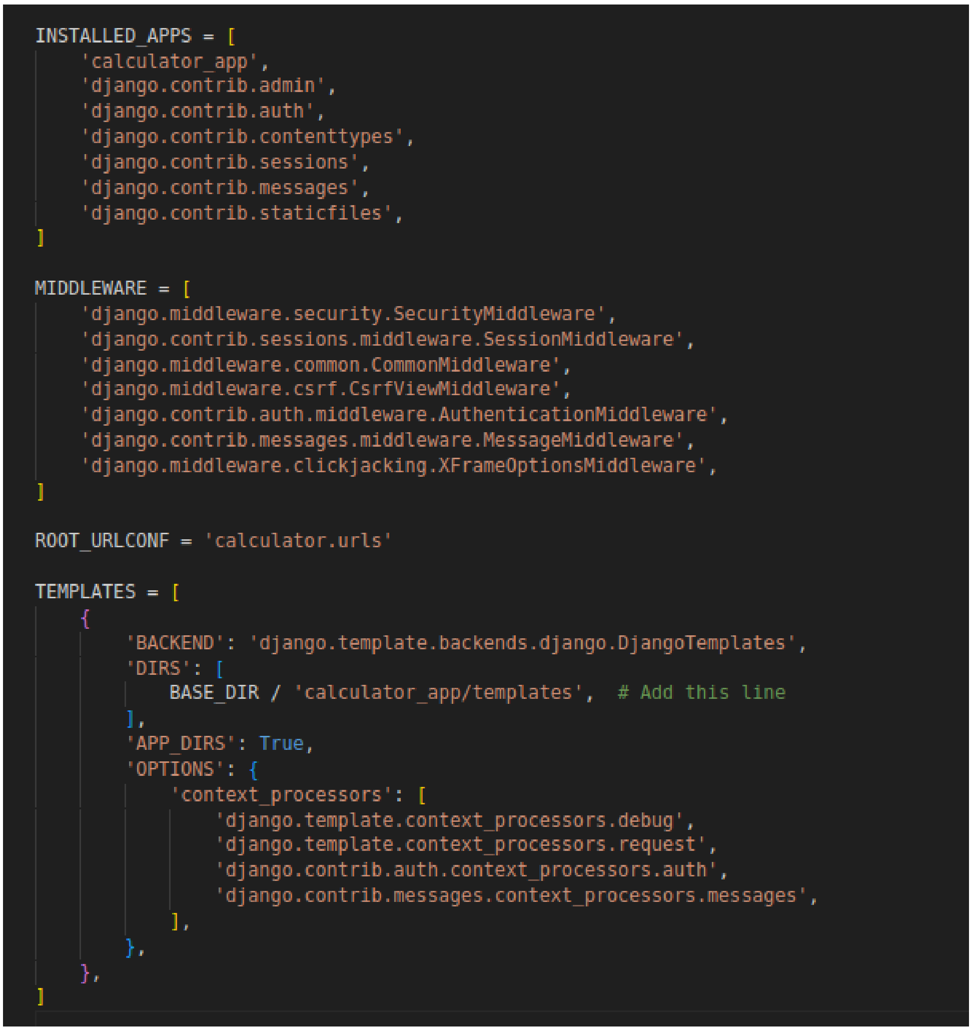
Lab No: 6 Developing Web Application using Django – Part II

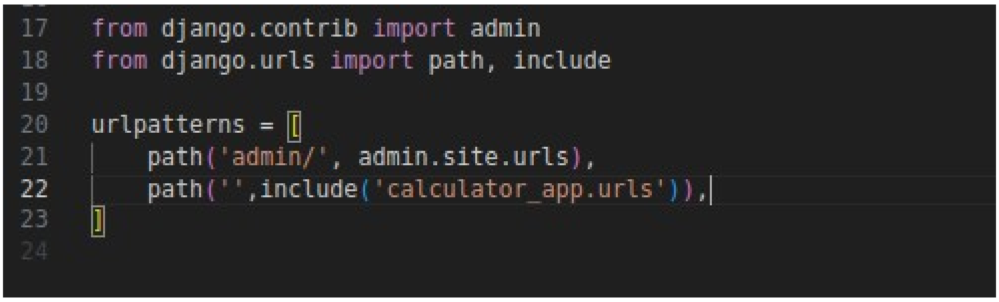
Name : Ketan Goud
Reg No: 220905260
Section: D D2
Roll No: 39

LAB ASSIGNMENTS:

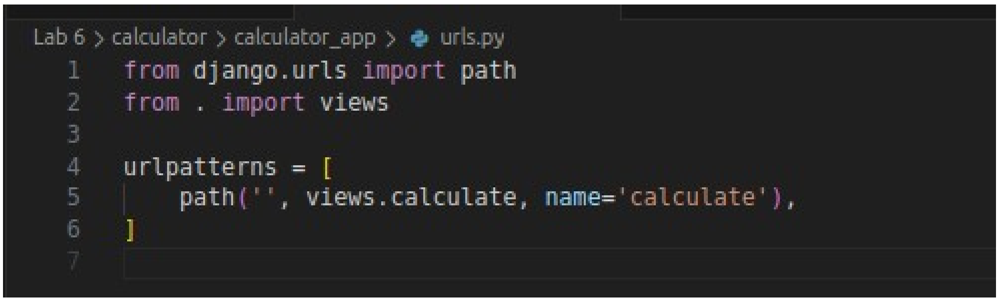
1) Develop a simple web page to perform basic arithmetic operations. Take two integer inputs from the user, select the operation to be performed using a drop down. Include a button "Calculate" to perform the selected operation, and then display the result in the same web page.



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

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 = 'calculator.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [
            BASE_DIR / 'calculator_app/templates',  # Add this line
        ],
        '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',
            ],
        },
    },
]
```

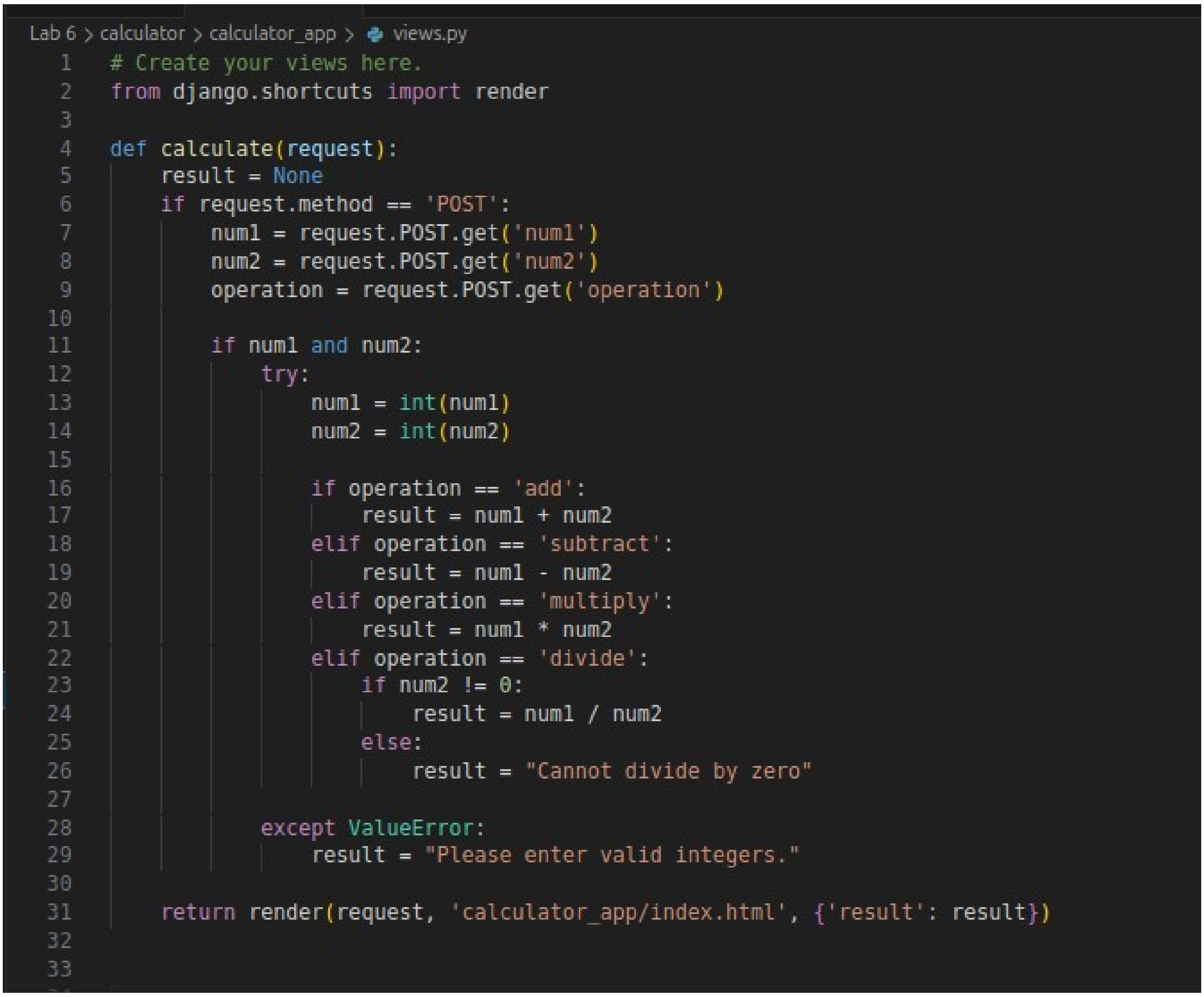


```
from django.contrib import admin
from django.urls import path, include

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



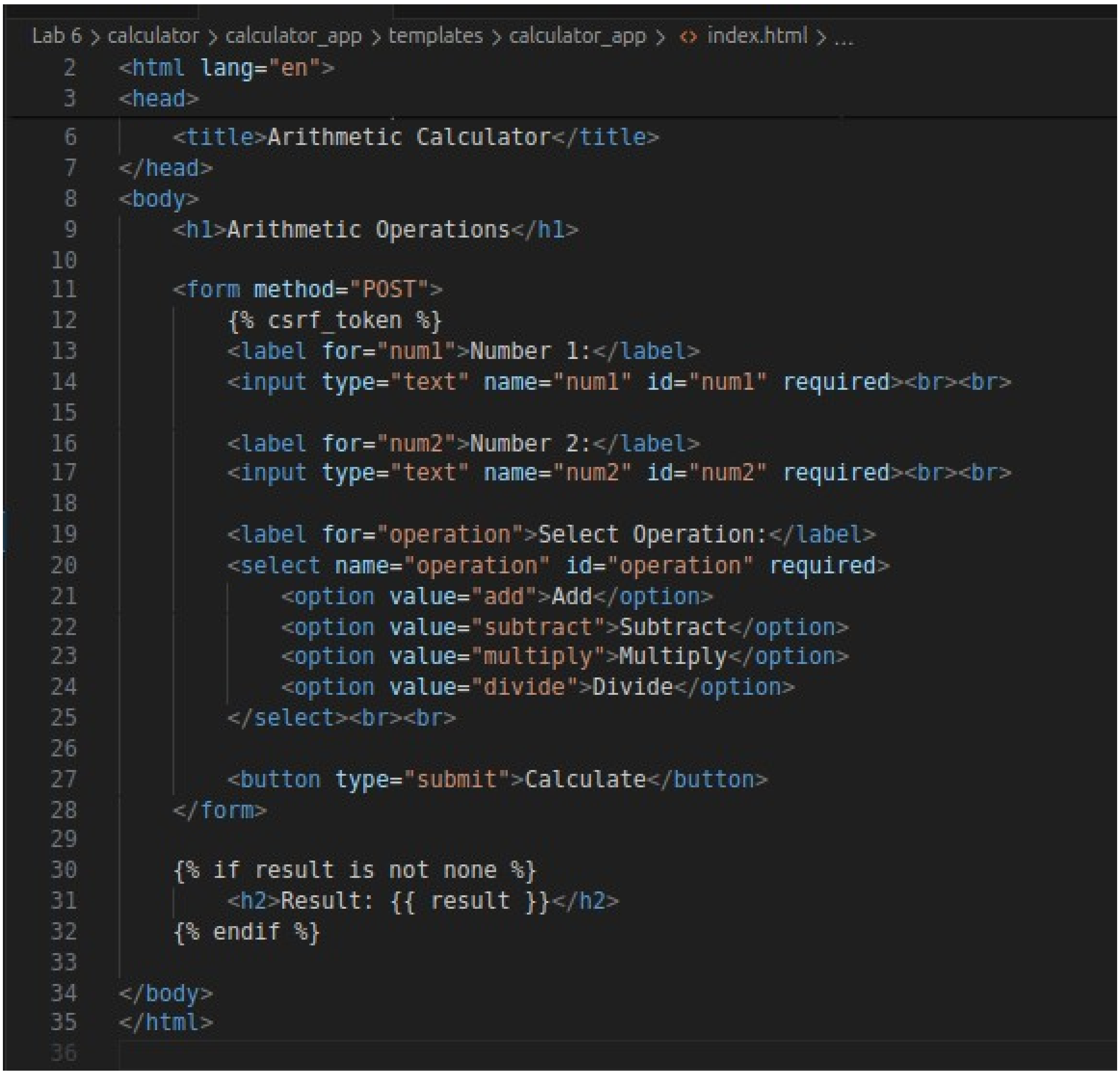
Lab 6 > calculator > calculator_app > urls.py

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

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



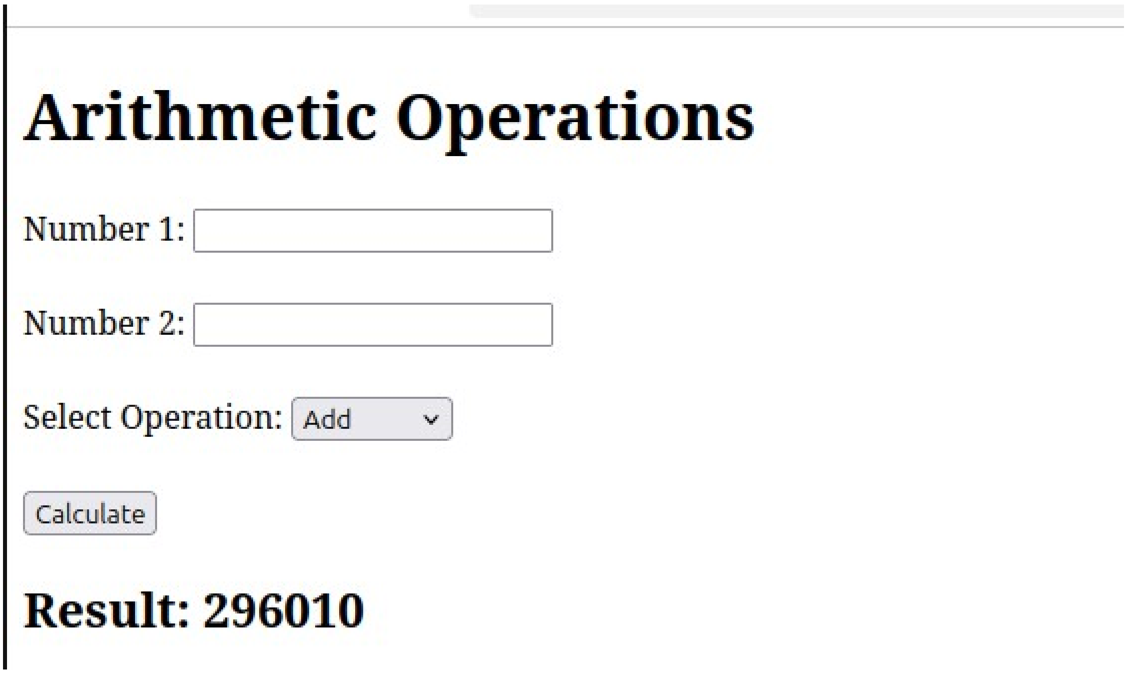
Lab 6 > calculator > calculator_app > views.py

```python
# Create your views here.
from django.shortcuts import render

def calculate(request):
    result = None
    if request.method == 'POST':
        num1 = request.POST.get('num1')
        num2 = request.POST.get('num2')
        operation = request.POST.get('operation')

        if num1 and num2:
            try:
                num1 = int(num1)
                num2 = int(num2)

                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 = "Please enter valid integers."

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



Lab 6 > calculator > calculator_app > templates > calculator_app > index.html > ...

```
<html lang="en">
<head>
    <title>Arithmetic Calculator</title>
</head>
<body>
    <h1>Arithmetic Operations</h1>

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

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

        <label for="operation">Select 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><br><br>

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

    {% if result is not none %}
        <h2>Result: {{ result }}</h2>
    {% endif %}

</body>
</html>
```



Arithmetic Operations

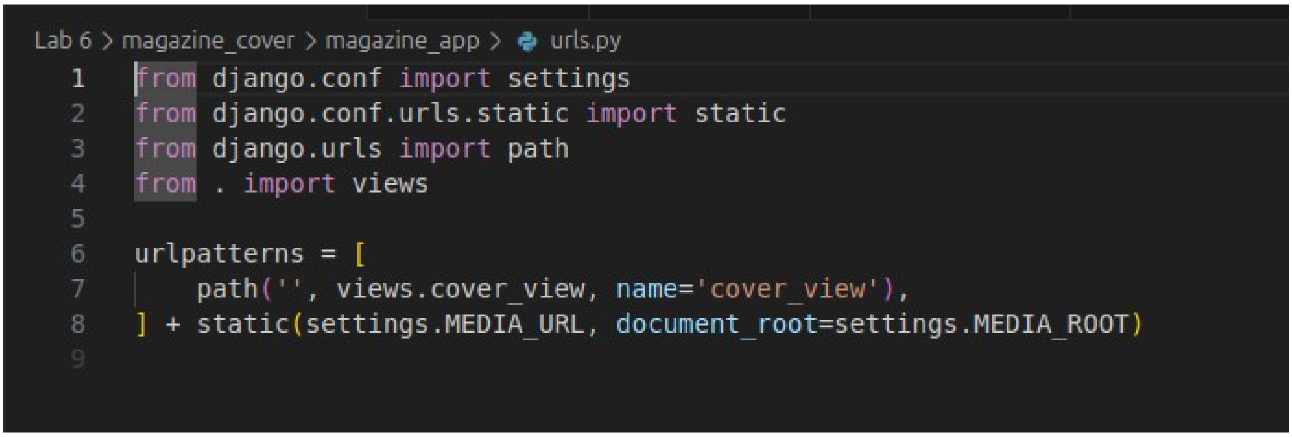
Number 1:

Number 2:

Select Operation: Add

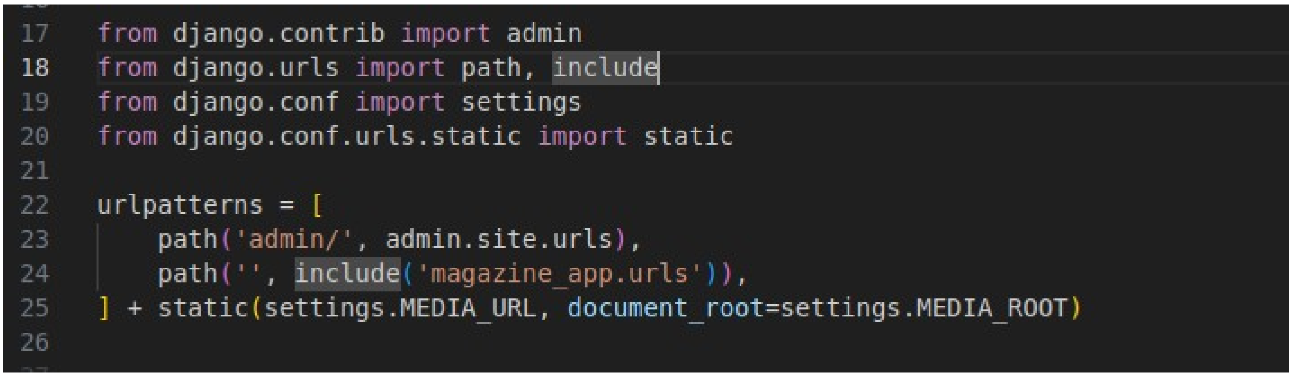
Result: 296010

2) 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.

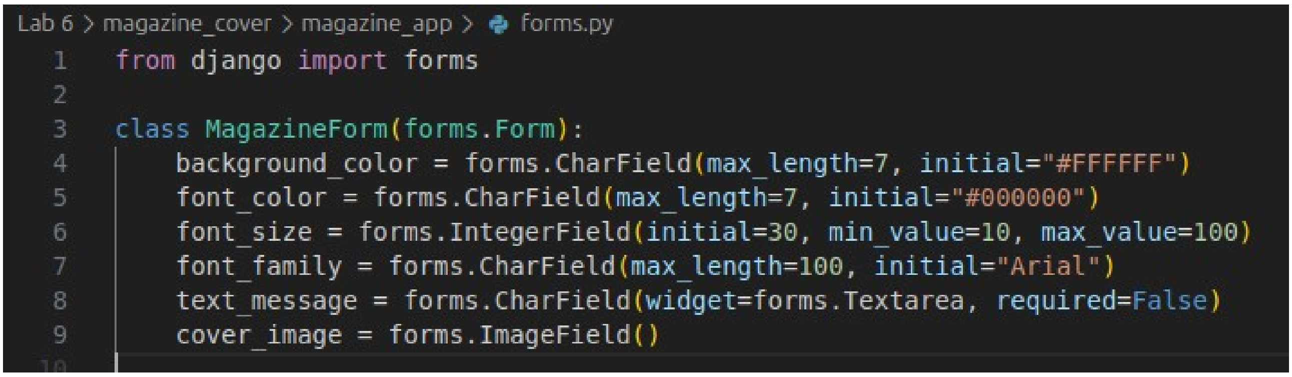


Lab 6 > magazine_cover > magazine_app > urls.py

```python
from django.conf import settings
from django.conf.urls.static import static
from django.urls import path
from . import views

urlpatterns = [
    path('', views.cover_view, name='cover_view'),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

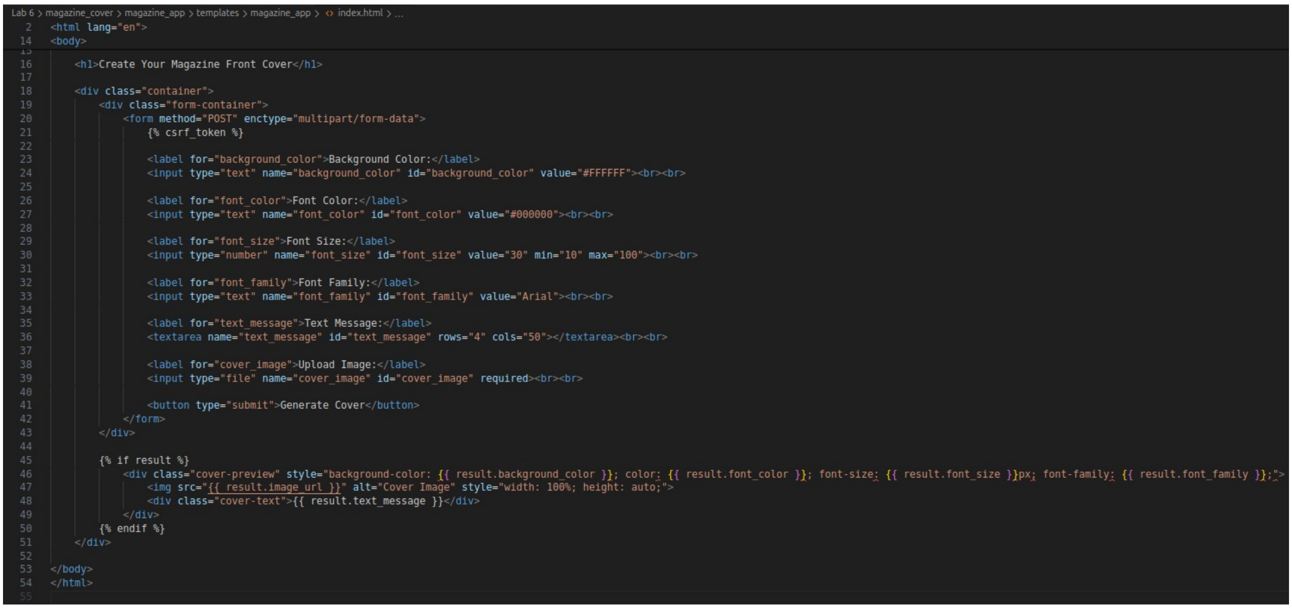


```python
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('magazine_app.urls')),
] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```



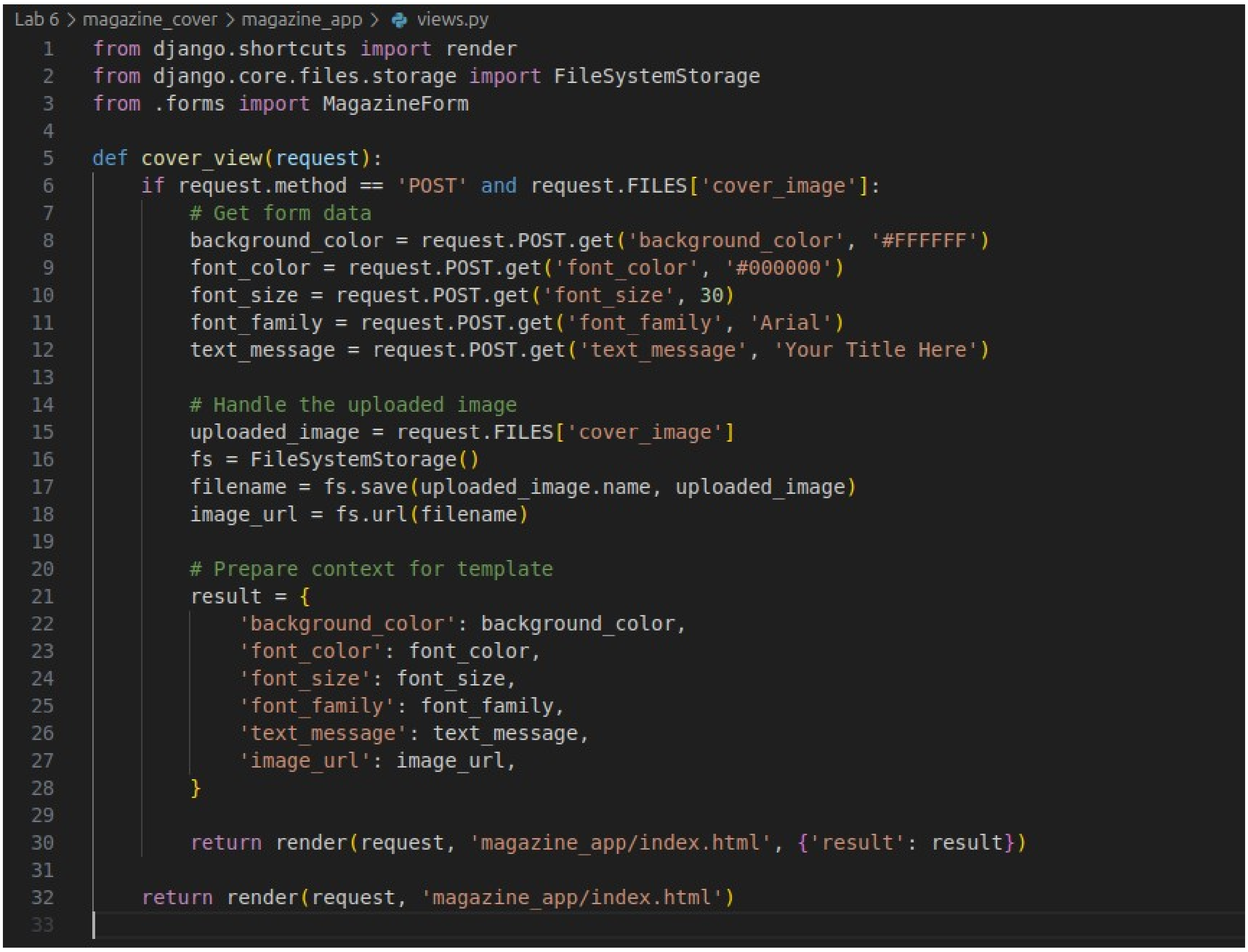
Lab 6 > magazine_cover > magazine_app > forms.py

```python
from django import forms

class MagazineForm(forms.Form):
    background_color = forms.CharField(max_length=7, initial="#FFFFFF")
    font_color = forms.CharField(max_length=7, initial="#000000")
    font_size = forms.IntegerField(initial=30, min_value=10, max_value=100)
    font_family = forms.CharField(max_length=100, initial="Arial")
    text_message = forms.CharField(widget=forms.Textarea, required=False)
    cover_image = forms.ImageField()
```



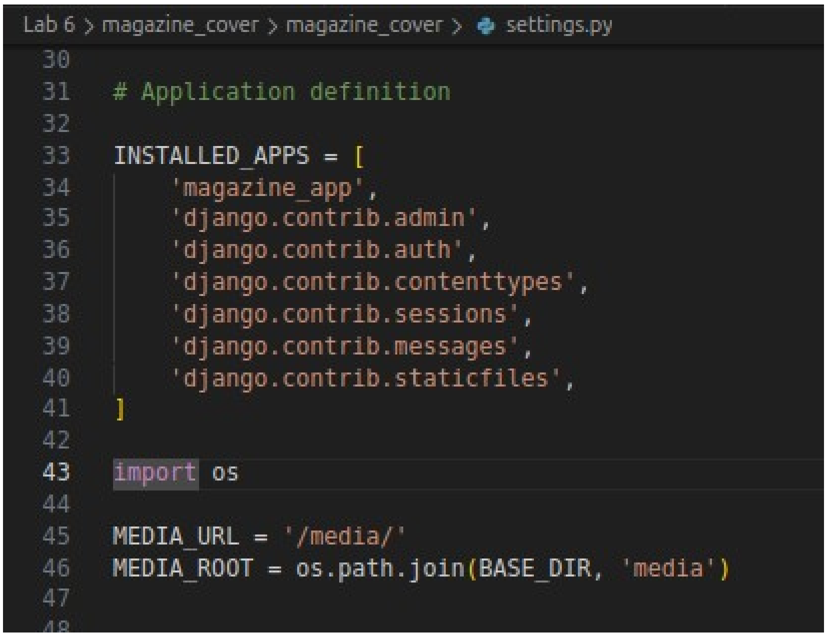
Lab 6 > magazine_cover > magazine_app > templates > magazine_app > <> index.html > ...

```html
<html lang="en">
<body>

    <h1>Create Your Magazine Front Cover</h1>

    <div class="container">
        <div class="form-container">
            <form method="POST" enctype="multipart/form-data">
                {% csrf_token %}

                <label for="background_color">Background Color:</label>
                <input type="text" name="background_color" id="background_color" value="#FFFFFF"><br><br>

                <label for="font_color">Font Color:</label>
                <input type="text" name="font_color" id="font_color" value="#000000"><br><br>

                <label for="font_size">Font Size:</label>
                <input type="number" name="font_size" id="font_size" value="30" min="10" max="100"><br><br>

                <label for="font_family">Font Family:</label>
                <input type="text" name="font_family" id="font_family" value="Arial"><br><br>

                <label for="text_message">Text Message:</label>
                <textarea name="text_message" id="text_message" rows="4" cols="50"></textarea><br><br>

                <label for="cover_image">Upload Image:</label>
                <input type="file" name="cover_image" id="cover_image" required><br><br>

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

        {% if result %}
            <div class="cover-preview" style="background-color: {{ result.background_color }}; color: {{ result.font_color }}; font-size: {{ result.font_size }}px; font-family: {{ result.font_family }};">
                <img src="{{ result.image_url }}" alt="Cover Image" style="width: 100%; height: auto;">
                <div class="cover-text">{{ result.text_message }}</div>
            </div>
        {% endif %}
    </div>

</body>
</html>
```

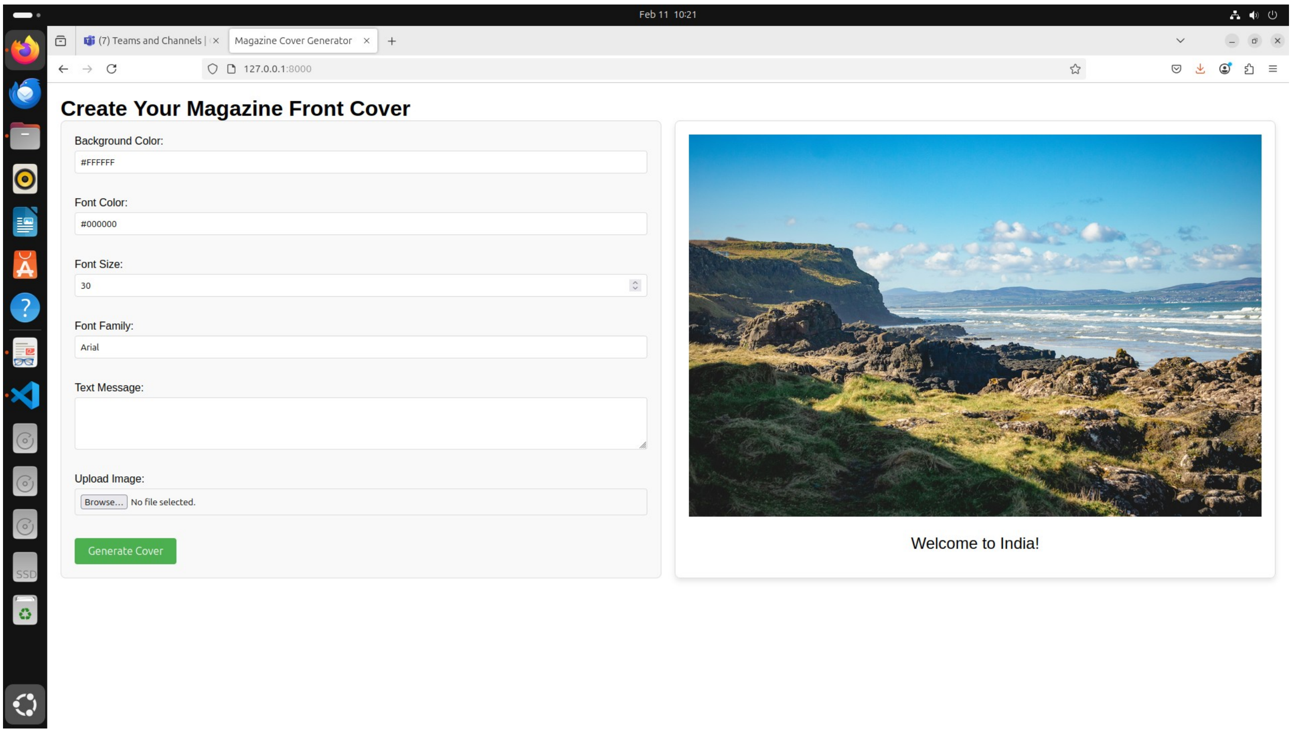


Lab 6 > magazine_cover > magazine_app > views.py

```python
from django.shortcuts import render
from django.core.files.storage import FileSystemStorage
from .forms import MagazineForm

def cover_view(request):
    if request.method == 'POST' and request.FILES['cover_image']:
        # Get form data
        background_color = request.POST.get('background_color', '#FFFFFF')
        font_color = request.POST.get('font_color', '#000000')
        font_size = request.POST.get('font_size', 30)
        font_family = request.POST.get('font_family', 'Arial')
        text_message = request.POST.get('text_message', 'Your Title Here')

        # Handle the uploaded image
        uploaded_image = request.FILES['cover_image']
        fs = FileSystemStorage()
        filename = fs.save(uploaded_image.name, uploaded_image)
        image_url = fs.url(filename)

        # Prepare context for template
        result = {
            'background_color': background_color,
            'font_color': font_color,
            'font_size': font_size,
            'font_family': font_family,
            'text_message': text_message,
            'image_url': image_url,
        }

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

    return render(request, 'magazine_app/index.html')
```



```python

# Application definition

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

import os

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')

```



Create Your Magazine Front Cover

Background Color: #FFFFFF

Font Color: #000000

Font Size: 30

Font Family: Arial

Text Message:

Upload Image: Browse... No file selected.

Welcome to India!