Week 5

Django

Hardik Garg 220962071

Q1 CODE

Settings.py

```
"""
Django settings for studentform project.

Generated by 'django-admin startproject' using Django 5.1.6.

For more information on this file, see
https://docs.djangoproject.com/en/5.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/5.1/ref/settings/
"""

from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent



# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/5.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-+-9v2=h^*^%kcy*0i9$4%vw805vg$y3_(pspvikf46)(odzbbq'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []
```

```
# Application definition

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

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 = 'studentform.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 = 'studentform.wsgi.application'
```

```
# Database
# https://docs.djangoproject.com/en/5.1/ref/settings/#databases

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



# Password validation
# https://docs.djangoproject.com/en/5.1/ref/settings/#auth-password-validators

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',
},
]



# Internationalization
# https://docs.djangoproject.com/en/5.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True
```

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.1/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/5.1/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Urls.py

```
"""
URL configuration for studentform project.

The `urlpatterns` list routes URLs to views. For more information please see:
https://docs.djangoproject.com/en/5.1/topics/http/urls/
Examples:
Function views
1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path('', views.home, name='home')
Class-based views
1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from django.urls import include, path
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include

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

Apps.py

```
from django.apps import AppConfig


class StudentinfoConfig(AppConfig):
default_auto_field = 'django.db.models.BigAutoField'
name = 'studentinfo'
```

Forms.py

```
from django import forms

class StudentForm(forms.Form):
name = forms.CharField(label='Student Name', max_length=100)
dob = forms.DateField(label='Date of Birth', widget=forms.DateInput(attrs={'type': 'date'}))
address = forms.CharField(label='Address', widget=forms.Textarea)
contact_number = forms.CharField(label='Contact Number', max_length=15)
email = forms.EmailField(label='Email')
english_marks = forms.IntegerField(label='English Marks')
physics_marks = forms.IntegerField(label='Physics Marks')
chemistry_marks = forms.IntegerField(label='Chemistry Marks')
```

Views.py

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

def student_form(request):
if request.method == 'POST':
form = StudentForm(request.POST)
if form.is_valid():
# Get form data
name = form.cleaned_data['name']
dob = form.cleaned_data['dob']
address = form.cleaned_data['address']
contact_number = form.cleaned_data['contact_number']
email = form.cleaned_data['email']
english_marks = form.cleaned_data['english_marks']
physics_marks = form.cleaned_data['physics_marks']
chemistry_marks = form.cleaned_data['chemistry_marks']
```

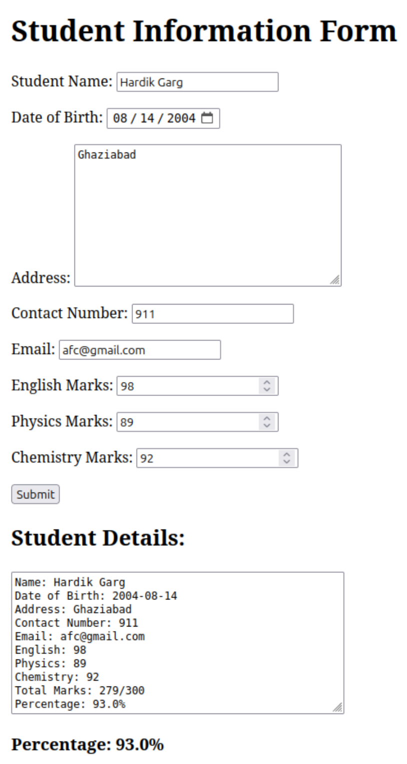
```
# Calculate total and percentage
total_marks = english_marks + physics_marks + chemistry_marks
percentage = (total_marks / 300) * 100

# Prepare the output
student_details = f"Name: {name}\nDate of Birth: {dob}\nAddress: {address}\nContact Number:
{contact_number}\nEmail: {email}\nEnglish: {english_marks}\nPhysics: {physics_marks}\nChemistry:
{chemistry_marks}\nTotal Marks: {total_marks}/300\nPercentage: {percentage}%"

# Pass data to template
return render(request, 'studentinfo/form.html', {'form': form, 'student_details': student_details,
'percentage': percentage})

else:
form = StudentForm()

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

.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Student Form</title>
</head>
<body>
<h1>Student Information Form</h1>
<form method="POST">
{% csrf_token %}
{{ form.as_p }}
<button type="submit">Submit</button>
</form>

{% if student_details %}
<h2>Student Details:</h2>
<textarea rows="10" cols="50" readonly>{{ student_details }}</textarea>
<h3>Percentage: {{ percentage }}%</h3>
{% endif %}
```

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

OUTPUT



Student Information Form

Student Name: Hardik Garg

Date of Birth: 08 / 14 / 2004

Address: Ghaziabad

Contact Number: 911

Email: afc@gmail.com

English Marks: 98

Physics Marks: 89

Chemistry Marks: 92

Student Details:

```
Name: Hardik Garg
Date of Birth: 2004-08-14
Address: Ghaziabad
Contact Number: 911
Email: afc@gmail.com
English: 98
Physics: 89
Chemistry: 92
Total Marks: 279/300
Percentage: 93.0%
```

Percentage: 93.0%

Q2 CODE

Settings.py

```
"""
Django settings for employeepromotion project.

Generated by 'django-admin startproject' using Django 5.1.6.

For more information on this file, see
https://docs.djangoproject.com/en/5.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/5.1/ref/settings/
"""
```

```python
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve().parent.parent


# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/5.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-8x%1i#dq)&@ymxt$uv-p(e=_=%vx*+^f0@xb2vu4wp2^347dnx'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []


# Application definition

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

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 = 'employeepromotion.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 = 'employeepromotion.wsgi.application'


# Database
# https://docs.djangoproject.com/en/5.1/ref/settings/#databases

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


# Password validation
# https://docs.djangoproject.com/en/5.1/ref/settings/#auth-password-validators

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',
},
]



# Internationalization
# https://docs.djangoproject.com/en/5.1/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_TZ = True



# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/5.1/howto/static-files/

STATIC_URL = 'static/'

# Default primary key field type
# https://docs.djangoproject.com/en/5.1/ref/settings/#default-auto-field

DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

Urls.py

```
"""
URL configuration for employeepromotion project.

The `urlpatterns` list routes URLs to views. For more information please see:
https://docs.djangoproject.com/en/5.1/topics/http/urls/
Examples:
Function views
1. Add an import: from my_app import views
2. Add a URL to urlpatterns: path('', views.home, name='home')
Class-based views
```

```
1. Add an import: from other_app.views import Home
2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
Including another URLconf
1. Import the include() function: from django.urls import include, path
2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
"""
from django.contrib import admin
from django.urls import path, include

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

Forms.py

```
from django import forms

class EmployeeForm(forms.Form):
employee_id = forms.ChoiceField(
label='Employee ID',
choices=[('1', 'E001'), ('2', 'E002'), ('3', 'E003'), ('4', 'E004'), ('5', 'E005')],
widget=forms.Select
)
date_of_joining = forms.DateField(
label='Date of Joining',
widget=forms.DateInput(attrs={'type': 'date'})
)
```

Views.py

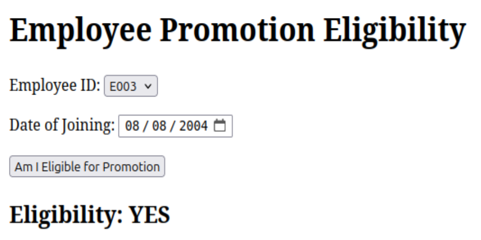
```
from django.shortcuts import render
from datetime import datetime
from .forms import EmployeeForm

def promotion_check(request):
eligibility = ""
if request.method == 'POST':
form = EmployeeForm(request.POST)
```

```
if form.is_valid():
# Get employee data from the form
employee_id = form.cleaned_data['employee_id']
date_of_joining = form.cleaned_data['date_of_joining']
# Calculate the years of experience
current_date = datetime.now().date()
years_of_experience = (current_date - date_of_joining).days // 365 # Approximate years

# Check if employee is eligible for promotion
if years_of_experience >= 5:
eligibility = "YES"
else:
eligibility = "NO"
return render(request, 'promotion/promotion_form.html', {'form': form, 'eligibility': eligibility})
else:
form = EmployeeForm()

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

.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Employee Promotion Check</title>
</head>
<body>
<h1>Employee Promotion Eligibility</h1>
<form method="POST">
{% csrf_token %}
{{ form.as_p }}
<button type="submit">Am I Eligible for Promotion</button>
</form>

{% if eligibility %}
<h2>Eligibility: {{ eligibility }}</h2>
{% endif %}
</body>
</html>
```

OUTPUT



Employee Promotion Eligibility

Employee ID: E003

Date of Joining: 08 / 08 / 2004

Eligibility: YES