Sriram Sunderrajan

220962444

WP Lab

Week-9

13th March 2025

**PROJECT FILES**

**settings.py**

"""

Django settings for myProj 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

import os

# 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-h0&nok12&=cc@-zp+$4\_-bq)-6cu9h$0tgo9owcn\*6\_rfih\*kn'

# 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',

]

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 = 'myProj.urls'

TEMPLATES = [

{

'BACKEND': 'django.template.backends.django.DjangoTemplates',

'DIRS': [

os.path.join(BASE\_DIR, "registerApp/templates"),

os.path.join(BASE\_DIR, "voteApp/templates"),

os.path.join(BASE\_DIR, "calcApp/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',

],

},

},

]

WSGI\_APPLICATION = 'myProj.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**

from django.contrib import admin

from django.urls import path, re\_path, include

urlpatterns = [

path('admin/', admin.site.urls),

re\_path(r'^registerApp/', include('registerApp.urls')),

re\_path(r'^voteApp/', include('voteApp.urls')),

re\_path(r'^calcApp/', include('calcApp.urls'))

]

**Question – 1**

**urls.py**

from django.urls import re\_path

from . import views

urlpatterns = [

re\_path(r'^$', views.index, name='index'),

]

**forms.py**

from django import forms

class RegrationForm(forms.Form):

username = forms.CharField(required=True)

password = forms.CharField(required=True, widget=forms.PasswordInput)

email\_id = forms.EmailField(required=False)

contact\_number = forms.IntegerField(required=False, min\_value=1000000000, max\_value=9999999999)

**views.py**

from django.shortcuts import render

from .forms import RegrationForm

def index(request):

username, password, email\_id, contact\_number = "None", "None", "None", "None"

if request.method == "POST":

myForm = RegrationForm(request.POST)

if myForm.is\_valid():

username, password, email\_id, contact\_number = myForm.cleaned\_data['username'], myForm.cleaned\_data['password'], myForm.cleaned\_data['email\_id'], myForm.cleaned\_data['contact\_number']

email\_id = "None" if email\_id == "" else email\_id

request.session['username'], request.session['password'], request.session['email\_id'], request.session['contact\_number'] = username, password, email\_id, contact\_number

return render(request, 'details.html', {"username" : username,

"password": password,

"email\_id": email\_id,

"contact\_number": contact\_number})

else:

return render(request, 'register.html', {"form": RegrationForm(request.POST or None)})

**CODE:**

**HTML**

**register.html**

<html>

<body style="height: 95%; display: flex; flex-direction: column; align-items: center; justify-content: center;">

<form action="" method="POST" style="background-color: bisque; padding: 30px 30px; border-radius: 5%;">

{% csrf\_token %}

{{ form.as\_p }}

<input type="submit" value="Submit">

</form>

</body>

</html>

**details.html**

<html>

<head>

<title>Student Details</title>

</head>

<body style="text-align: center; display: flex; flex-direction: column; justify-content: center; align-items: center; height: 100%;">

<div style="background-color: aquamarine; border: blueviolet; display: flex; flex-direction: column; gap: 30px; padding: 20 20;">

<label>

Welcome <strong> {{username}} </strong> <br>

Email ID: <strong> {{email\_id}} </strong> <br>

Contact Number: <strong> {{contact\_number}} </strong> <br>

</label>

<a href="/registerApp/">

<button>Go to Home</button>

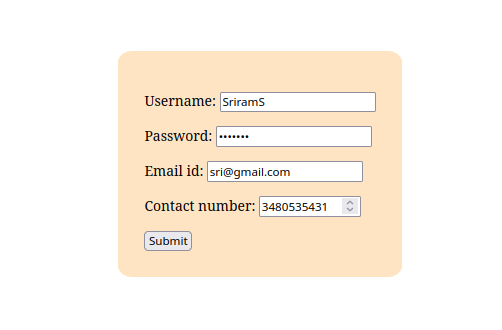
</a>

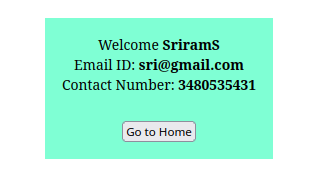
</div>

</body>

</html>

**OUTPUT**





**Question – 2**

**urls.py**

from django.urls import re\_path

from . import views

urlpatterns = [

re\_path(r'^$', views.index, name='index'),

]

**forms.py**

from django import forms

CHOICES = (("Good", "Good"), ("Satisfactory", "Satisfactory"), ("Bad", "Bad"))

class VoteForm(forms.Form):

vote = forms.ChoiceField(required=True, widget=forms.RadioSelect(), choices=CHOICES)

**views.py**

from django.shortcuts import render

from .forms import VoteForm

VOTES = {"Good": 0,

"Satisfactory": 0,

"Bad": 0}

def index(request):

if request.method == "POST":

myForm = VoteForm(request.POST)

if myForm.is\_valid():

vote = myForm.cleaned\_data['vote']

VOTES[vote] += 1

print(VOTES)

total\_votes = sum(list(VOTES.values()))

context = {"vote\_good": f"{VOTES["Good"]/total\_votes:.2%}",

"vote\_satisfactory": f"{VOTES["Satisfactory"]/total\_votes:.2%}",

"vote\_bad": f"{VOTES["Bad"]/total\_votes:.2%}"}

print(context)

return render(request, 'result.html', context)

else:

return render(request, 'vote.html', {"form": VoteForm(request.POST or None)})

**CODE:**

**HTML**

**vote.html**

<html>

<body style="height: 95%; display: flex; flex-direction: column; align-items: center; justify-content: center;">

<form action="" method="POST" style="background-color: bisque; padding: 30px 30px; border-radius: 5%;">

{% csrf\_token %}

<label>How is the book about C#?</label> <br>

{{ form.as\_p }}

<input type="submit" value="Vote">

</form>

</body>

</html>

**result.html**

<html>

<body style="height: 95%; display: flex; flex-direction: column; align-items: center; justify-content: center;">

<form action="" method="POST" style="background-color: bisque; padding: 30px 30px; border-radius: 5%;">

<label>How is the book about C#?</label> <br><br>

<div style="display: flex; flex-direction: row; gap: 30px;">

<div>

<input type="radio" value="Good" disabled="true">Good</input>

</div>

<label> {{ vote\_good }} </label>

</div> <br>

<div style="display: flex; flex-direction: row; gap: 30px;">

<div>

<input type="radio" value="Satisfactory" disabled="true">Satisfactory</input>

</div>

<label> {{ vote\_satisfactory }} </label>

</div> <br>

<div style="display: flex; flex-direction: row; gap: 30px;">

<div>

<input type="radio" value="Bad" disabled="true">Bad</input>

</div>

<label> {{ vote\_bad }} </label>

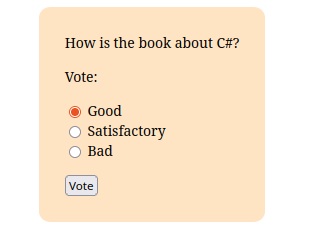
</div> <br>

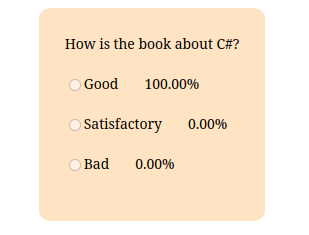
</form>

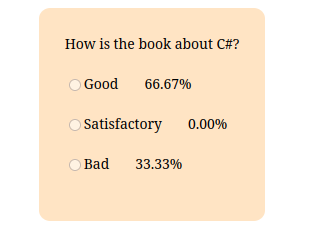
</body>

</html>

**OUTPUT:**





****

**Question – 3**

**urls.py**

from django.urls import re\_path

from . import views

urlpatterns = [

re\_path(r'^$', views.index, name='index'),

]

**forms.py**

from django import forms

class CGPAForm(forms.Form):

name = forms.CharField(required=True)

marks = forms.IntegerField(required=True, min\_value=0, max\_value=500)

**views.py**

from django.shortcuts import render

from .forms import CGPAForm

def index(request):

if request.method == "POST":

myForm = CGPAForm(request.POST)

if myForm.is\_valid():

name, marks = myForm.cleaned\_data['name'], myForm.cleaned\_data['marks']

cgpa = marks / 50

return render(request, 'cgpa\_result.html', {"CGPA": cgpa, "name": name})

else:

return render(request, 'calc.html', {"form": CGPAForm(request.POST or None)})

**CODE:**

**HTML**

**calc.html**

<html>

<body style="height: 95%; display: flex; flex-direction: column; align-items: center; justify-content: center;">

<form action="" method="POST" style="background-color: bisque; padding: 30px 30px; border-radius: 5%;">

{% csrf\_token %}

{{ form.as\_p }}

<input type="submit" value="Calculate">

</form>

</body>

</html>

**cgpa\_result.html**

<html>

<head>

<title>Student Details</title>

</head>

<body style="text-align: center; display: flex; flex-direction: column; justify-content: center; align-items: center; height: 100%;">

<div style="background-color: aquamarine; border: blueviolet; display: flex; flex-direction: column; gap: 30px; padding: 20 20;">

<strong> Welcome {{ name }} </strong> <br>

<strong> Your CGPA is {{ CGPA }} </strong> <br>

<a href="/calcApp/">

<button>Go to Home</button>

</a>

</div>

</body>

</html>

**OUTPUT:**

