Sriram Sunderrajan

220962444

WP Lab

Week-7

27th February 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-+sjzzgz2k)=mbwcl@w+\*)a\_p!gx@f@ymmdy9s^f$2r#7^+6nzd'

# 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, "carApp/templates"),

os.path.join(BASE\_DIR, "sessApp/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

"""

URL configuration for myProj 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, re\_path

urlpatterns = [

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

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

re\_path(r'^sessApp/', include('sessApp.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

CHOICES = (('BMW', 'BMW'), ('Toyota', 'Toyota'), ('Maruti Suzuki', 'Maruti Suzuki'), ('Lamborghini', 'Lamborghini'), ('Tesla', 'Tesla'))

class RegForm(forms.Form):

manufacturer = forms.ChoiceField(widget=forms.Select, choices=CHOICES)

model = forms.CharField(required=True)

views.py

from django.shortcuts import render

from .forms import RegForm

from django.views.decorators.csrf import csrf\_protect

@csrf\_protect

def index(request):

manufacturer, model = "None", "None"

if request.method == "POST":

# Get the posted form

myForm = RegForm(request.POST)

if myForm.is\_valid():

manufacturer, model = myForm.cleaned\_data['manufacturer'], myForm.cleaned\_data['model']

context = {'manufacturer': manufacturer, 'model': model}

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

else:

myForm = RegForm(request.POST or None)

context = {'form': myForm}

return render(request, "form.html", context)

CODE:

HTML

form.html

<html>

<body>

<form action="" method="POST">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

</body>

</html>

info.html

<html>

<head>

<title>Car 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; padding: 20px 20px;">

<h4>The Manufacturer is {{manufacturer}}</h4>

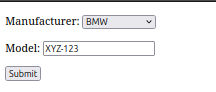
<p>The Model is {{model}}</p>

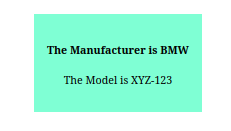
</div>

</body>

</html>

**OUTPUT**





Question – 2

urls.py

from django.urls import re\_path

from . import views

urlpatterns = [

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

re\_path(r'^info/$', views.studentInfo, name='studentInfo'),

]

forms.py

from django import forms

SUBJECTS = (('Mathematics', 'Mathematics'), ('English', 'English'), ('Computer Science', 'Computer Science'), ('Physics', 'Physics'), ('Chemistry', 'Chemistry'), ('Biology', 'Biology'))

class StudentForm(forms.Form):

name = forms.CharField(required=True)

rollno = forms.CharField(required=True)

subject = forms.ChoiceField(widget=forms.Select, choices=SUBJECTS)

views.py

from django.shortcuts import render

from .forms import StudentForm

from django.views.decorators.csrf import csrf\_protect

def index(request):

name, rollno, subject = "None", "None", "None"

if request.method == "POST":

# Get the posted form

myForm = StudentForm(request.POST)

if myForm.is\_valid():

name, rollno, subject = myForm.cleaned\_data['name'], myForm.cleaned\_data['rollno'], myForm.cleaned\_data['subject']

request.session['name'], request.session['rollno'] , request.session['subject'] = name, rollno, subject

return render(request, 'second.html', {"name" : request.session['name'],

"rollno": request.session['rollno'],

"subject": request.session['subject']})

else:

myForm = StudentForm(request.POST or None)

context = {'form': myForm}

return render(request, "first.html", context)

def studentInfo(request):

if request.session.has\_key('name'):

return render(request, 'second.html', {"name" : request.session['name'],

"rollno": request.session['rollno'],

"subject": request.session['subject']})

else:

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

CODE:

HTML

first.html

<html>

<body>

<form action="" method="POST">

{% csrf\_token %}

{{ form.as\_p }}

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

</form>

</body>

</html>

second.html

<html>

<head>

<title>Car 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>

<strong>Name:</strong> {{name}} <br>

<strong>Roll Number:</strong> {{rollno}} <br>

<strong>Subject:</strong> {{subject}} <br>

</label>

<a href="/sessApp/">

<button>Go to Home</button>

</a>

</div>

</body>

</html>

**OUTPUT:**

