ARUNIMA SINGH THAKUR

180905218

ROLL NO. 31

SECTION C

BRANCH CSE

IT LAB

24TH MAY 2021

(FORM PROCCESING USING DJANGO)

lab6/settings.py

```
import os
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/3.2/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'django-insecure-yq2&3@go@^-&_f@)_jcgyqjz69eh9t8(b67+196d6+7v(99^u6'

# 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',
    'q1_app',
    'q2_app',
    'q3_app',
    'q4_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 = 'LAB6.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 = 'LAB6.wsgi.application'


# Database
# https://docs.djangoproject.com/en/3.2/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        #'NAME': BASE_DIR / 'db.sqlite3',
        'NAME': str(os.path.join(BASE_DIR, "db.sqlite3"))
    }
}


# Password validation
# https://docs.djangoproject.com/en/3.2/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/3.2/topics/i18n/

LANGUAGE_CODE = 'en-us'

TIME_ZONE = 'UTC'

USE_I18N = True

USE_L10N = True

USE_TZ = True


# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.2/howto/static-files/

STATIC_URL = '/static/'

# Default primary key field type
# https://docs.djangoproject.com/en/3.2/ref/settings/#default-auto-field

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

lab6/urls.py

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

urlpatterns = [
    path('admin/', admin.site.urls),
    path('q1_app/', include('q1_app.urls')),
    path('q2_app/', include('q2_app.urls')),
    path('q3_app/', include('q3_app.urls')),
    path('q4_app/', include('q4_app.urls')),
]
```

QUESTION 1

Develop a web application using Django framework to demonstrate the transfer of multiple parameters between web pages. User should be presented with a dropdown list containing car manufacturers, a text box which takes model name of the manufacturer and a submit button. On submitting the web page, the user is forwarded to a new page. This new page should display the selected car manufacturer name and the model name.

Q1/forms.py

```
from django import forms
from django.core.management.color import Style

CHOICES=[('AUDI', 'AUDI'), ('Maruti', 'Maruti'), ('TATA', 'TATA'), (
'Ford', 'Ford'), ('Mercedes', 'Mercedes'), ('Toyota', 'Toyota')]

class ManuFactForm(forms.Form):
    Manufacturers= forms.ChoiceField(choices=CHOICES)
    Model_Name= forms.CharField()
```

Q1/views.py

```
from django.shortcuts import render
from django.http import HttpResponse
from .forms import ManuFactForm
# Create your views here.
def submit(request):
    form = ManuFactForm(request.POST or None)
    Manu_Name=''
    Model=''
    if request.method == 'POST':
        FinalForm= ManuFactForm(request.POST)
        if FinalForm.is_valid():
            Manu_Name=FinalForm.cleaned_data['Manufacturers']
            Model=FinalForm.cleaned_data['Model_Name']

    context = {'Manu_Name':Manu_Name,'Model':Model,'form':form}
    return render(request,'result.html', context)
```

```
def home(request):
    form = ManuFactForm(request.POST or None)
    context={'form':form}
    return render(request,'home.html',context)
def result(request):
    return render(request,'result.html')
```

Q1/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
path('',views.home,name="home"),
path('result',views.result,name='result'),
path('submit',views.submit,name='submit')
]
```

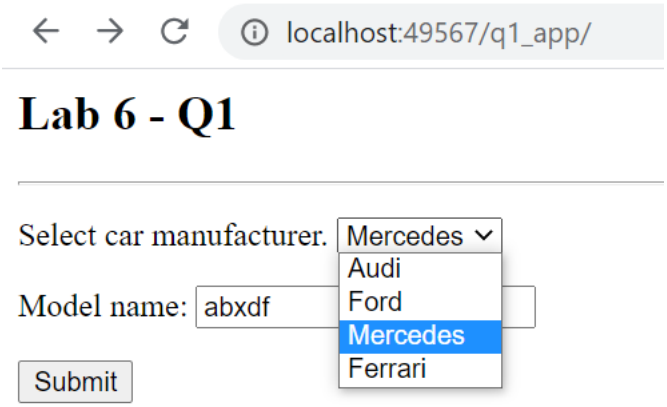
Q1/templates/home.html

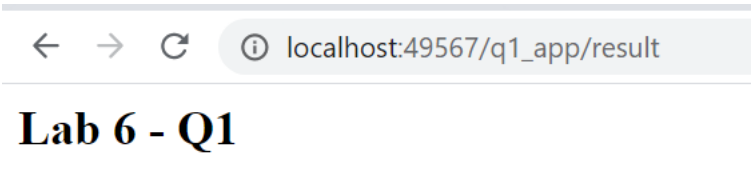
```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-
width, initialscale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <form name="form" action="{% url 'submit' %}" method="POST">
      {% csrf_token %}
      {{ form.as_p }}
      <input type="submit" value="Submit">
    </form>
  </body>
</html>
```

Q1/templates/result.html

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Result</title>
  </head>
  <body>
    <p style="font-size:30px">
    The Car Manufacturers name is : {{Manu_Name}}
    <br>
    The Model Name is : {{Model}}
    </p>
  </body>
</html>
```

OUTPUT:





QUESTION 2:

Create a page firstPage.html with two TextBoxes [Name, Roll], DropDownList [Subjects], and a button. Create another page secondPage.html with a label and a button. When the user clicks the button in first Page, he should be sent to the second page and display the contents passed from first page in the label. The button in second page should navigate the user back to firstPage. Use Django sessions to transfer information.

Q2/forms.py

```
from django import forms

class StudentForm(forms.Form):

    name = forms.CharField(max_length=100, required=False)
    rollno = forms.CharField(max_length=100, required = False)

    CHOICES = (
        ('1','Maths'),
        ('2','Physics'),
        ('3','Chemistry'),
        ('4','Biology')
    )

    sub = forms.ChoiceField(widget=forms.Select, choices = CHOICES, required = False)
```

Q2/views.py

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

def firstPage(request):

    context = {}
```

```
    form = StudentForm(request.POST or None)
    context['form'] = form
    return render(request, 'firstPage.html', context)

def secondPage(request):

    context = {}
    name = ""
    rollno = ""
    sub = ""

    if request.method == 'POST':

        form = StudentForm(request.POST)
        context['form'] = form

        if form.is_valid():

            name = form.cleaned_data['name']
            rollno = form.cleaned_data['rollno']

            sub = form.cleaned_data['sub']
            sub = dict(form.fields['sub'].choices)[sub]

            print(name,rollno,sub)

            request.session['name'] = name
            request.session['rollno'] = rollno
            request.session['sub'] = sub

    else:
        form = StudentForm()
        context['form'] = form

    return render(request,'secondPage.html',{'name' : name, 'rollno' : rollno, 'sub' : sub})
```

Q2/urls.py

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

app_name = 'q2_app'
urlpatterns = [
    path('firstPage',views.firstPage,name = 'firstPage'),
    path('secondPage',views.secondPage,name = 'secondPage')
]
```

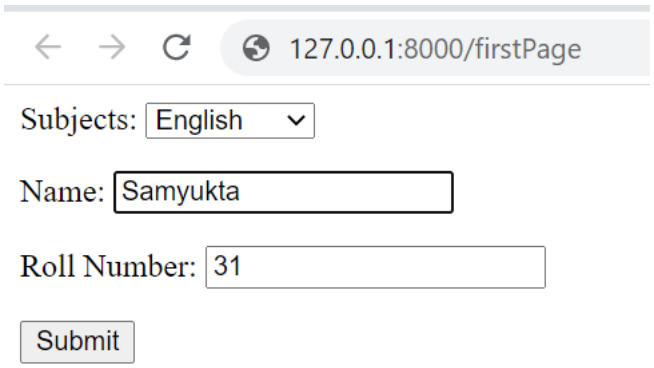
Q2/templates/firstpage.html

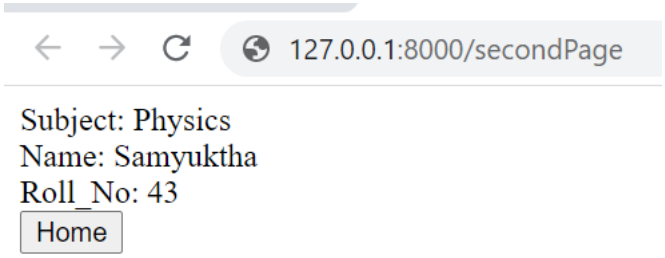
```
<html>

<head>

   <meta charset="utf-8">
   <title>Student Form</title>

</head>

<body>

   <h2>Fill Details</h2>

   <form action="{% url 'secondPage' %}" method='POST'>
      {% csrf_token %}

      {{form.as_p}}

      <input type="submit">

   </form>

</body>

</html>
```

Q2/templates/secondpage.html

```
<html>

<body>

   <label>Name : <strong>{{name}}</strong></label><br>
   <label></label>Roll No.: <strong>{{rollno}}</strong><label></label><br>
   <label>Subject : <strong>{{sub}}</strong></label><br>

   <form action="{% url 'firstPage' %}" method='POST'>

      {% csrf_token %}

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

   </form>

</body>

</html>
```

OUTPUT:





QUESTION 3:

Create a Register page and Success page with the following requirements:

i. Register page should contain four input TextBoxes for UserName, Password, Email id and Contact Number and also a button to submit. Make the username as compulsory field and other fields as optional.

ii. On button click, Success page is displayed with message "Welcome {UserName}" and also his Email and Contact Number has to be displayed.

iii. Use secure technique to send details to the Success page (Hint: use csrftoken)

Q3/forms.py

```
from django import forms
from django.core.management.color import Style
from django.forms.widgets import EmailInput, PasswordInput

class LoginForm(forms.Form):
    UserName=forms.CharField()
    PassWord=forms.CharField(widget=PasswordInput,required=False)
    Email=forms.CharField(widget=EmailInput,required=False,)
    Contact_Num=forms.IntegerField(required=False)
```

Q3/views.py

```
from django.forms.widgets import EmailInput
from django.shortcuts import render
from django.http import HttpResponse
from .forms import LoginForm
# Create your views here.
def submit(request):
    form = LoginForm(request.POST or None)

    if request.method =='POST':
        FinalForm= LoginForm(request.POST)
        if FinalForm.is_valid():
            UserName=FinalForm.cleaned_data['UserName']
            Email=FinalForm.cleaned_data['Email']
            Contact_Num=FinalForm.cleaned_data['Contact_Num']
    else:
        FinalForm=LoginForm()
    context={"UserName":UserName,"Email":Email,"Contact_Num":Contact
_Num}
    return render(request,'secondpage.html',context)

def firstpage(request):
    form = LoginForm(request.POST or None)
    context={'form':form}
    return render(request,'firstpage.html',context)
def secondpage(request):
    return render(request,'secondpage.html')
```

Q3/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
path('',views.firstpage,name="home"),
path('submit',views.submit,name='submit')
]
```

Q3/templates/firstpage.html

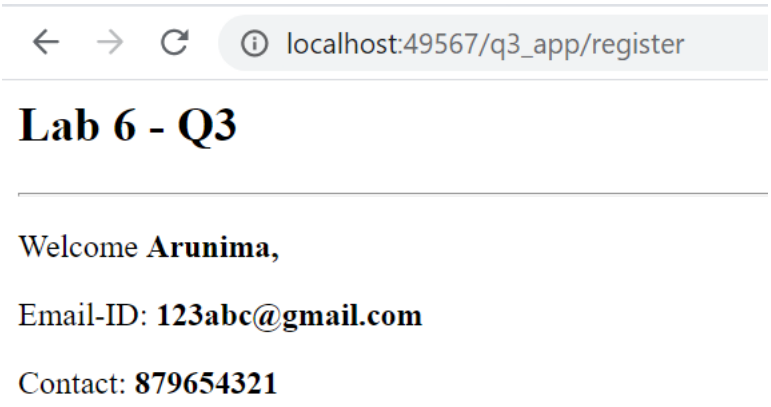
```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-
width, initialscale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <form name="form" action="{% url 'submit' %}" method="POST">
      {% csrf_token %}
      {{ form.as_p }}
      <input type="submit" value="Register">
    </form>
  </body>
</html>
```

Q3/templates/secondpage.html

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
```

```
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Result</title>
  </head>
  <body>
  <center>
  <img src="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcQSCCNnoi4cKmakYxg6icwlNGh9RJUMfM
2Mug&usqp=CAU">

  <br><br><br>
  <h2>Welcome {{UserName}}</h2>

  <pre style="font-size:20px;font-family:Helvetica">

  Email Id:{{Email}}

  Contact no:{{Contact_Num}}
  </pre>
  </center>

  </body>
</html>
```

OUTPUT:



QUESTION 4:

Design a website with two pages. First page contains:

RadioButton with HP, Nokia, Samsung, Motorola, Apple as options.

CheckBox with Mobile and Laptop as items.

TextBox to enter quantity. There is a button with text as "Produce Bill".

On Clicking Produce Bill button, item should be displayed with total amount on another page

Q4/forms.py

```
from django import forms
from django.core.management.color import Style

COMPANY =[('HP','HP'), ('Nokia','Nokia'), ('Samsung','Samsung'), ('M
otorola','Motorola'), ('Apple','Apple')]
class BillForm(forms.Form):
    Company_Name= forms.ChoiceField(widget=forms.RadioSelect,choices
=COMPANY)
    Mobile= forms.CharField(widget=forms.CheckboxInput)
    Laptop= forms.CharField(widget=forms.CheckboxInput)
    Quantity=forms.IntegerField()
```

Q4/views.py

```
from django.shortcuts import render
from django.http import HttpResponse
from .forms import BillForm
# Create your views here.
def submit(request):
    form = BillForm(request.POST or None)
    Mob_Cost={
        "HP":1000,
        "Nokia":2000,
        "Samsung":1200,
        "Motorola":1500,
        "Apple":500
    }
    Laptop_Cost={
        "HP":5000,
        "Nokia":8000,
        "Samsung":12000,
        "Motorola":7000,
        "Apple":9000
    }
    Company_Name=''
    Mobile=''
    Laptop=''
    Quantity=''
    Price=0
```

```
    if request.method =='POST':
        FinalForm= BillForm(request.POST)
        if FinalForm.is_valid():
            Company_Name=FinalForm.cleaned_data['Company_Name']
            Mobile=FinalForm.cleaned_data['Mobile']
            Laptop=FinalForm.cleaned_data['Laptop']
            Quantity=FinalForm.cleaned_data['Quantity']
    else:
        FinalForm=BillForm()

    if (Mobile)=='True':
        Price=Price+(Mob_Cost[Company_Name]*Quantity)
    if (Laptop)=='True':
        Price=Price+(Laptop_Cost[Company_Name]*Quantity)
    context={"Company_Name":Company_Name,"Mobile":Mobile,"Laptop":La
ptop,"Quantity":Quantity,"Price":Price}
    return render(request,'secondpage.html',context)

def firstpage(request):
    form = BillForm(request.POST or None)
    context={'form':form}
    return render(request,'firstpage.html',context)
def secondpage(request):
    return render(request,'secondpage.html')
```

Q4/urls.py

```
from django.urls import path
from . import views
urlpatterns = [
path('',views.firstpage,name="home"),
path('submit',views.submit,name='submit')
]
```

Q4/templates/firstpage.html

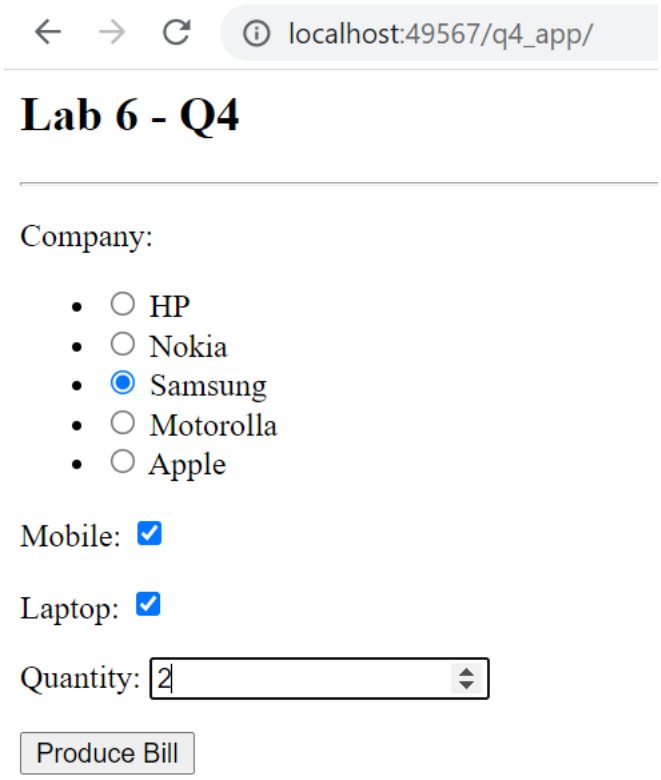
```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-
width, initialscale=1.0" />
    <title>Document</title>
  </head>
  <body>
    <form name="form" action="{% url 'submit' %}" method="POST">
      {% csrf_token %}
      {{ form.as_p }}
      <input type="submit" value="Produce Bill">
    </form>
  </body>
</html>
```

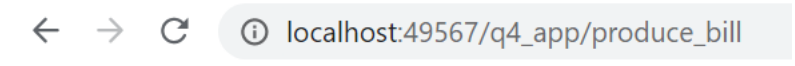
Q4/templates/secondpage.html

```
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-
scale=1.0" />
    <title>Result</title>
  </head>
  <body>
  <pre style="color:black;font-size:20px">
                Purchase Bill

  Product Company : {{Company_Name}}
  Mobile Purchase: {{Mobile}}
  Laptop Purchase: {{Laptop}}
  Quantity: {{Quantity}}

          Price In TOTAL : {{Price}}
  </pre>
```

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

OUTPUT:





Lab 6 - Q4

Company: **Samsung**

Mobile: **yes**

Laptop: **yes**

Quantity: **2**

Total Bill: **Rs. 24000**