WEEK 6 FORM PROCESSING USING DJANGO

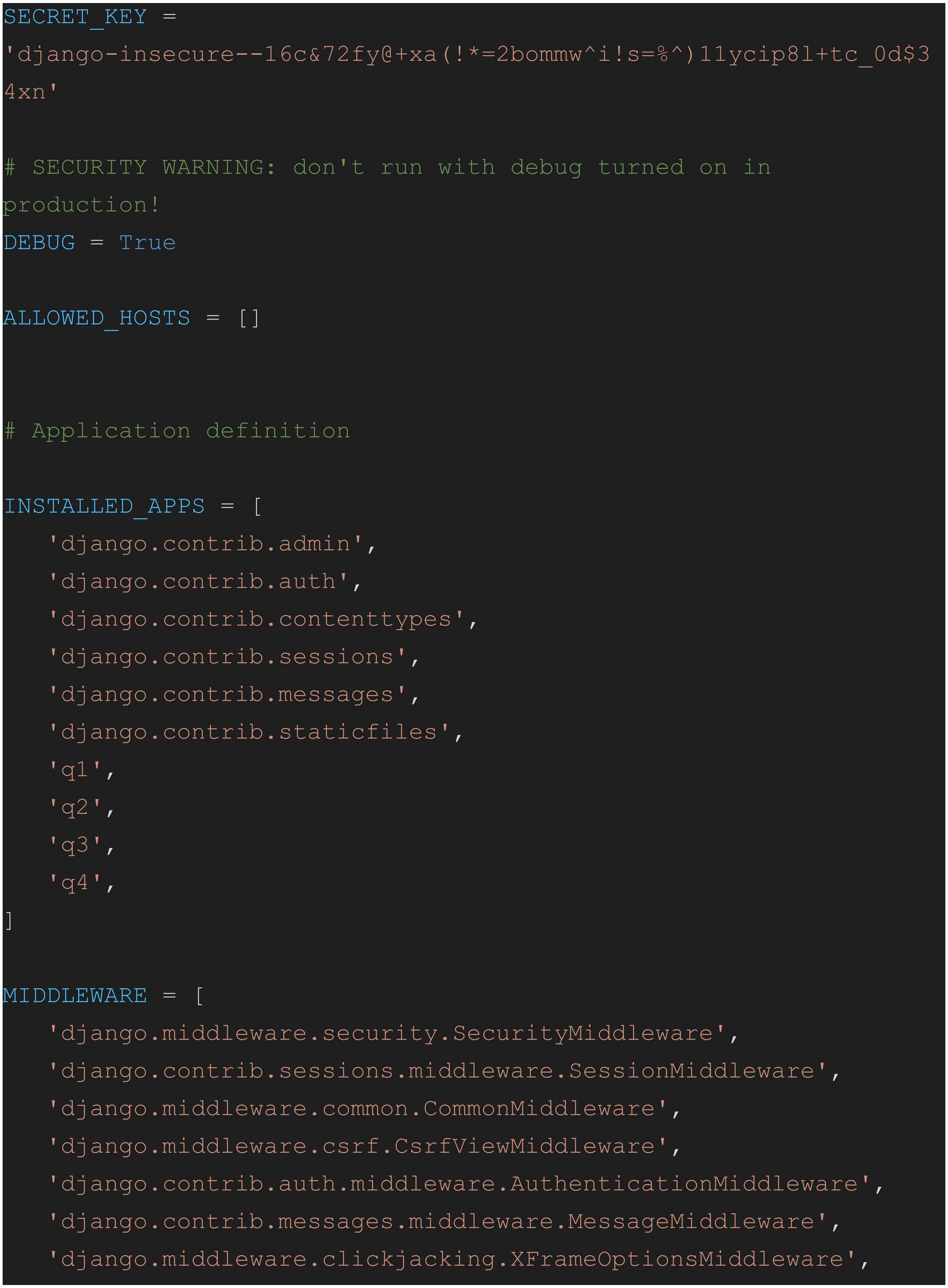
Name: Sagnik Chatterjee
Reg: 180905478
Roll No: 61
SEM: 6 B

Lab6: urls.py

```
"""lab6 URL Configuration

The `urlpatterns` list routes URLs to views. For more
information please see:
   https://docs.djangoproject.com/en/3.2/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
from django.urls.conf import include

urlpatterns = [
   path('admin/', admin.site.urls),
   path('q1/', include('q1.urls')),
   path('q2/', include('q2.urls')),
   path('q3/', include('q3.urls')),
   path('q4/', include('q4.urls')),
]
```

Lab6: settings.py

```
"""
Django settings for lab6 project.

Generated by 'django-admin startproject' using Django 3.2.1.

For more information on this file, see
https://docs.djangoproject.com/en/3.2/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.2/ref/settings/
"""
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--16c&72fy@+xa(!*=2bommw^i!s=%^)11ycip81+tc_0d$3
4xn'

# 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',
   'q2',
   'q3',
   'q4',
]

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',
   }
}
```

```
# Password validation
#
https://docs.djangoproject.com/en/3.2/ref/settings/#auth-passwor
d-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarity
Validator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator'
,
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator
',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidato
r',
    },
]


# 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'
```

Q1 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.forms import widgets

car_manufacturers = [
    ('Q2', 'Audi'),
    ('Aspire', 'Ford'),
    ('SL400', 'Mercedes'),
    ('F8', 'Ferrari'),
    ('Model S','Tesla'),
    ('Cerna', 'Toyota')
]

class BasicForm(forms.Form):
    car_manufacturer = forms.CharField(label='Select car manufacturer.', widget=forms.Select(choices=car_manufacturers))
    model_name = forms.CharField()
```

q1/urls.py

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

app_name = 'q1'
urlpatterns = [
    path('', views.index, name='index'),
    path('result', views.result, name='result')
]
```

q1/views.py

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


def index(request):
   form = BasicForm()
   return render(request, 'q1/index.html', {'form': form})

def result(request):
   if request.method == 'POST':
       form = BasicForm(request.POST)
       if form.is_valid():
           car_manufacturer =
form.cleaned_data['car_manufacturer']
           model_name = form.cleaned_data['model_name']

   context = {'car_manufacturer': car_manufacturer,
'model_name': model_name}
   return render(request, 'q1/result.html', context)
```

q1/templates/q1/index.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>Lab6 | Q1</title>
 </head>
 <body>
   <p>
     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
```

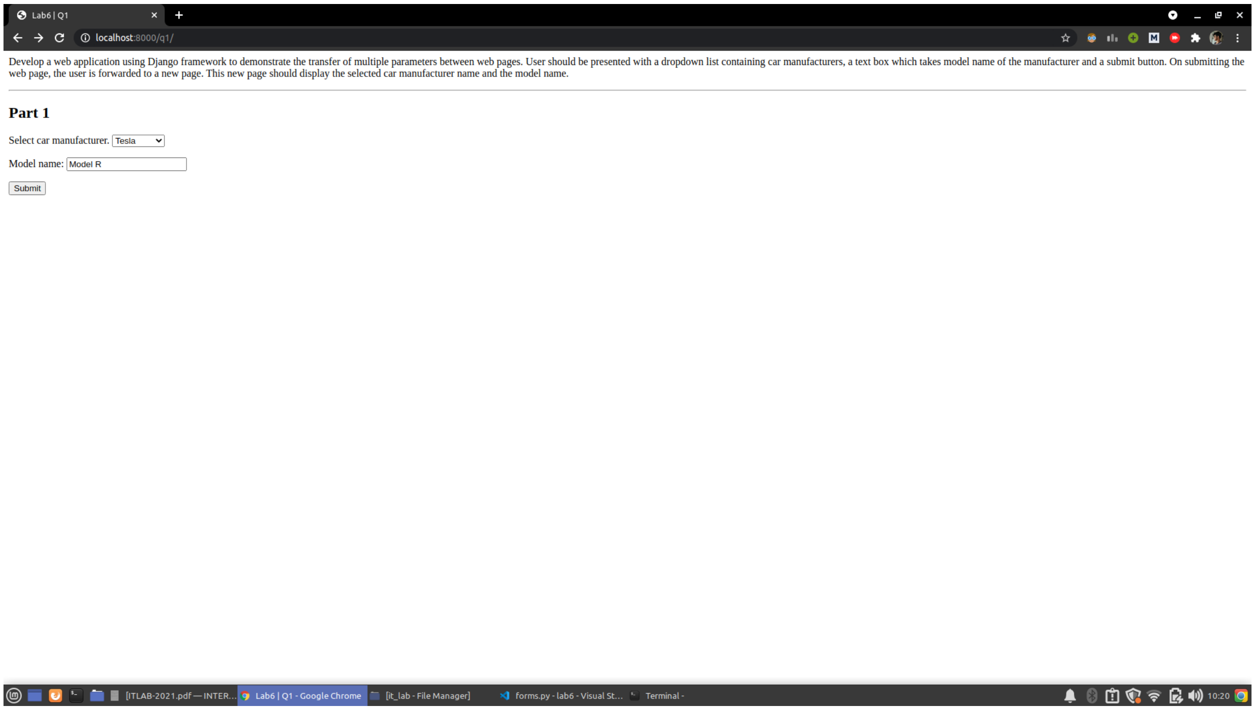
```
     which takes model name of the manufacturer and a submit
button. On
     submitting the web page, the user is forwarded to a new
page. This new
     page should display the selected car manufacturer name and
the model name.
   </p>
   <hr />
   <h2>
       Part 1
   </h2>
   <form method="POST" action="result">
     {% csrf_token %} {{ form.as_p }}
     <button type="submit">Submit</button>
   </form>

</body>
 </html>
```

q1/templates/q1/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>Lab 6 | Q1</title>
 </head>
 <body>
   <p>
     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.
   </p>
   <hr />
   <ul>
     <li>
       <p>Car Manufacturer :- <strong>{{ car_manufacturer
}}</strong></p>
     </li>
     <li>
       <p>Model Name :- <strong>{{ model_name }}</strong></p>
     </li>
   </ul>
 </body>
</html>
```

Img1 :


Lab6 | Q1
localhost:8000/q1/
[ITLAB-2021.pdf — INTER...
Lab6 | Q1 - Google Chrome
[it_lab - File Manager]
forms.py - lab6 - Visual St...
Terminal -
10:20

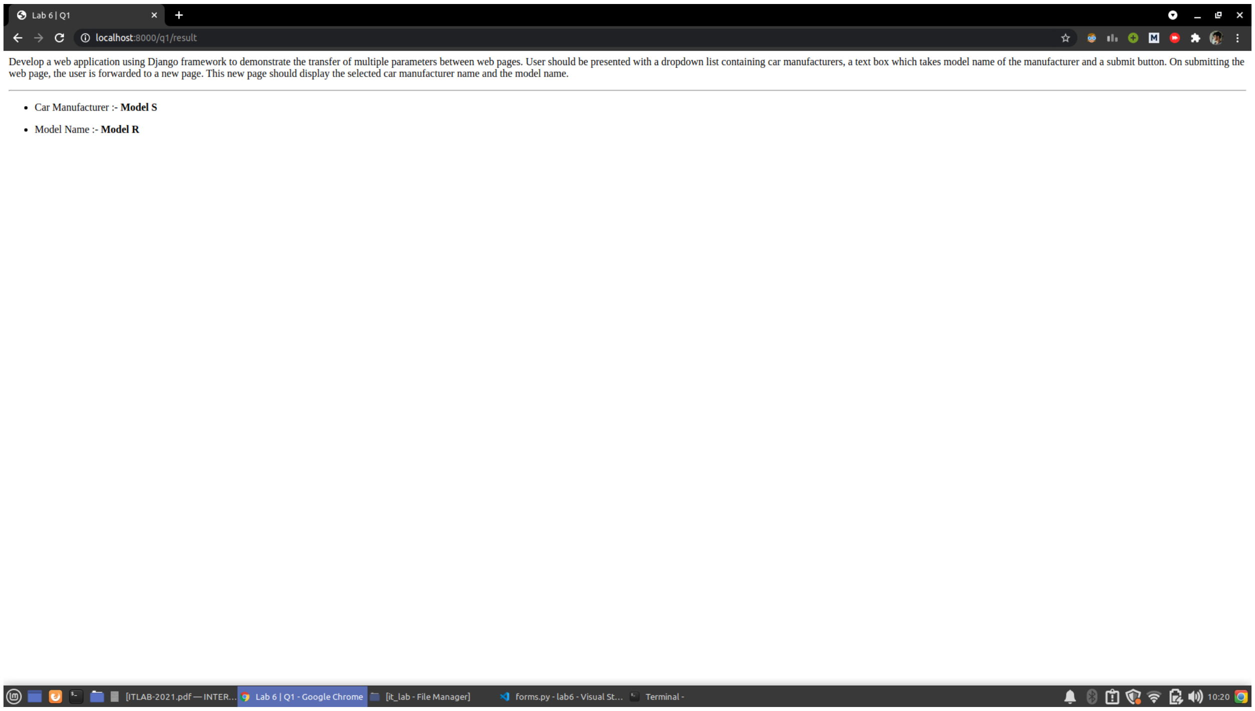
Develop a web application using Django framework to demonstrate the transfer of multiple parameters between web pages. User should be presented with a dropdown list containing car manufacturers, a text box which takes model name of the manufacturer and a submit button. On submitting the web page, the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

Part 1

Select car manufacturer. Tesla

Model name: Model R

Img2:


Lab 6 | Q1
localhost:8000/q1/result
[ITLAB-2021.pdf — INTER...
Lab 6 | Q1 - Google Chrome
[it_lab - File Manager]
forms.py - lab6 - Visual St...
Terminal -
10:20

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.

- Car Manufacturer :- **Model S**
- Model Name :- **Model R**

Q2 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/views.py

```
from django.forms.forms import Form
from django.shortcuts import render,redirect
from . import BasicForm

def index(request):
   form = BasicForm()
   request.session["name"] = ""
   request.session["roll"] = ""
   request.session["subject"] = ""
   return render(request, 'q2/firstPage.html', {'form': form})

def info(request):
   if "name" not in request.session:
       request.session["name"] = ""
   if "roll" not in request.session:
       request.session["rollno"] = ""
   if "subject" not in request.session:
       request.session["subject"] = ""

   if request.method == 'POST':
       form = BasicForm(request.POST)
       if form.is_valid():
           name = form.cleaned_data['name']
           roll = form.cleaned_data['roll']
           subject = form.cleaned_data['subject']
```

```
            request.session['name'] += name
            request.session['roll'] = roll
            request.session['subject'] += subject

            return redirect('/secondPage')
    else:
        form = BasicForm()
        request.session["name"] = ""
        request.session["roll"] = ""
        request.session["subject"] = ""

        return render(request,'q2/firstPage.html',{'form':form})

def second(request):
    name = request.session['name']
    roll = request.session['roll']
    subject = request.session['subject']

    context = {'name': name, 'roll': roll, 'subject': subject}
    return render(request, 'q2/secondPage.html', context)
```

q2/urls.py

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

app_name = 'q2'
urlpatterns = [
    path('index', views.index, name='index'),
    path('firstPage', views.first_page, name='first_page'),
    path('secondPage', views.second_page, name='second_page'),
]
```

q2/forms.py

```
from django import forms
from django.forms import widgets
from django.forms.fields import CharField, IntegerField

subjects = [
   ('Math', 'Math'),
   ('Phycics', 'Physics'),
   ('English', 'English'),
   ('Chemistry', 'Chemistry'),
   ('Web Dev','Web Dev'),
   ('Algorithms','Algorithms'),
]

class BasicForm(forms.Form):
   name = CharField()
   roll = IntegerField()
   subject = CharField(label='Select subject.',
widget=forms.Select(choices=subjects))
```

q1/templates/q2/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,
initial-scale=1.0" />
   <title>Lab6 | Q2</title>
 </head>
 <body>
   <p>
     Create a page firstPage.html with two TextBoxes [Name,
Roll], DropDownList
```

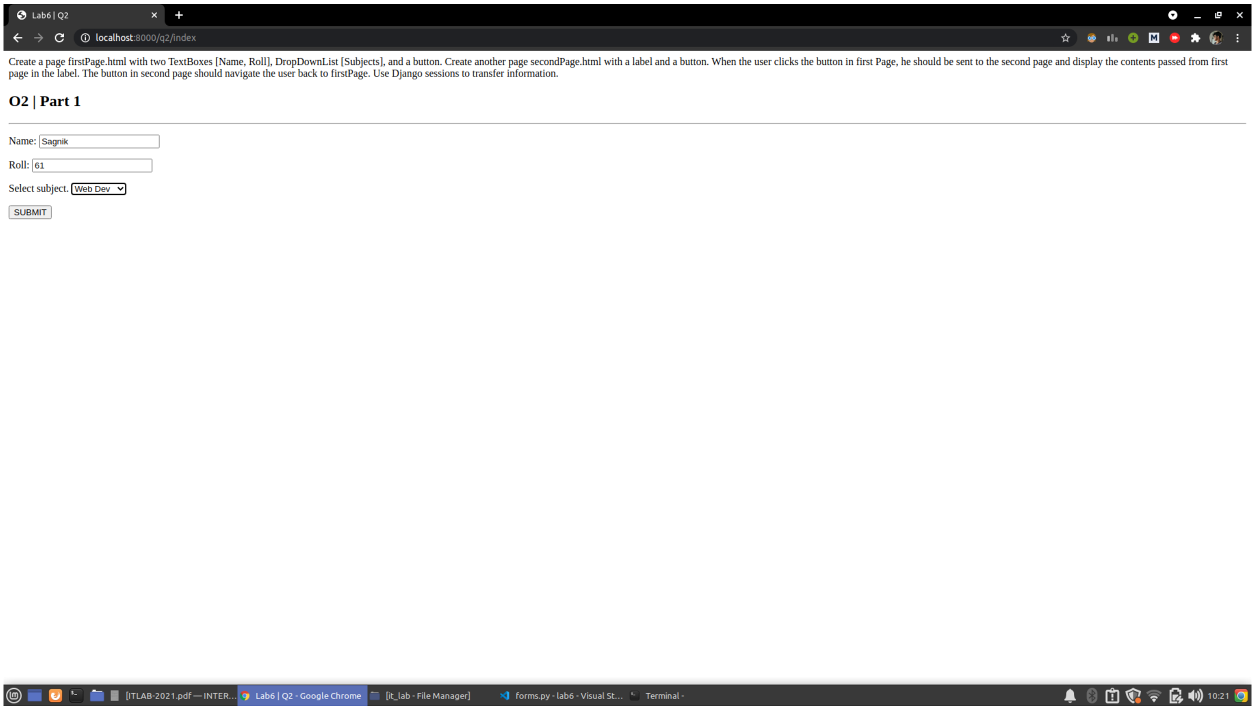
```
     [Subjects], and a button. Create another page
secondPage.html with a label
     and a button. When the user clicks the button in first
Page, he should be
     sent to the second page and display the contents passed
from first page in
     the label. The button in second page should navigate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

   <h2>Q2 | Part 1</h2>
   <hr />

   <form method="POST" action="firstPage">
     {% csrf_token %} {{ form.as_p }}
     <button type="submit">SUBMIT</button>
   </form>
 </body>
</html>
```

q2/templates/q2/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>Lab6 | Q2</title>
 </head>
 <body>
   <p>
     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.
   </p>
   <h2>Q2 | Part 2</h2>
   <hr />

   <p>Name: <strong>{{ name }}</strong></p>
   <p>Roll: <strong>{{ roll }}</strong></p>
   <p>Subject: <strong>{{ subject }}</strong></p>

   <form method="GET" action="index">
     <button type="submit">First Page</button>
   </form>
 </body>
</html>
```

Img1 :


Lab6 | Q2
localhost:8000/q2/index
[ITLAB-2021.pdf — INTER...
Lab6 | Q2 - Google Chrome
[it_lab - File Manager]
forms.py - lab6 - Visual St...
Terminal -
10:21

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.

O2 | Part 1

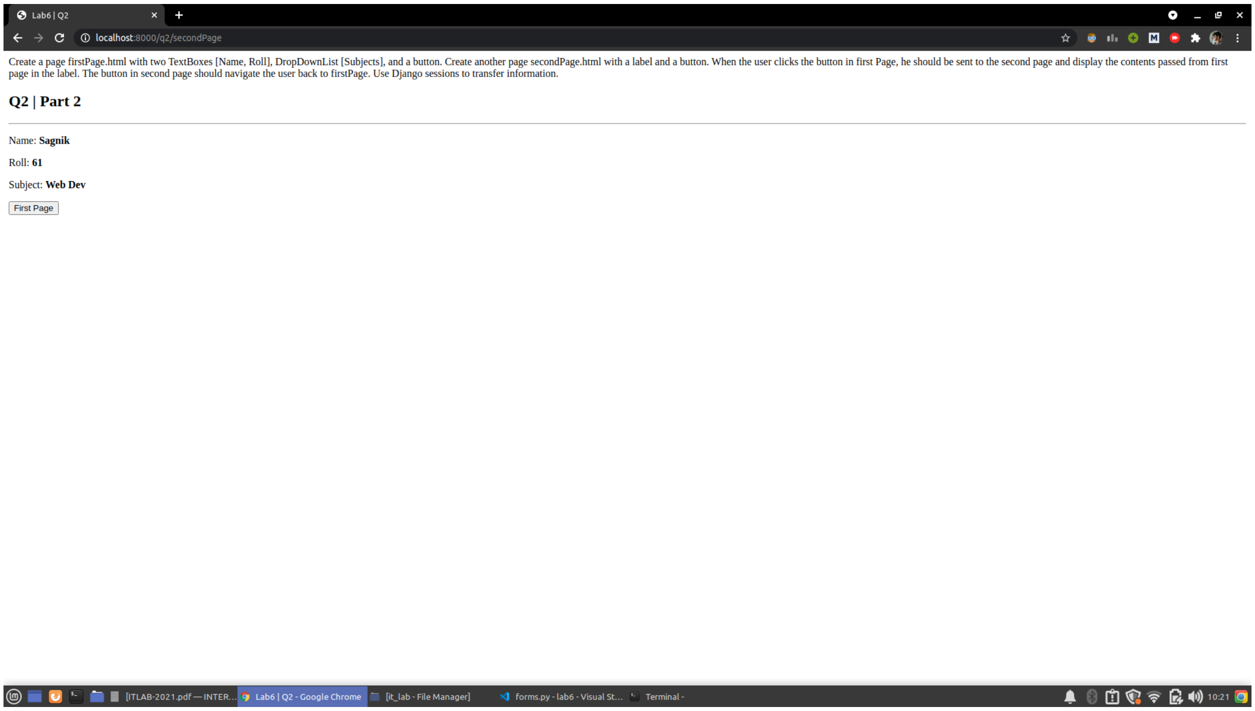
Name: Sagnik

Roll: 61

Select subject. Web Dev

SUBMIT

Img2 :


Lab6 | Q2
localhost:8000/q2/secondPage
[ITLAB-2021.pdf — INTER...
Lab6 | Q2 - Google Chrome
[it_lab - File Manager]
forms.py - lab6 - Visual St...
Terminal -
10:21

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 | Part 2

Name: **Sagnik**

Roll: **61**

Subject: **Web Dev**

First Page

Q3 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/views.py

```
from django.shortcuts import render, redirect
from .forms import BasicForm


def index(request):
    form = BasicForm()
    return render(request, 'q3/register.html', {'form': form})

def register(request):
    username=""
    password=""
    email=""
    contact=""
    if request.method == 'POST':
        form = BasicForm(request.POST)
        if form.is_valid():
            username = form.cleaned_data['username']
            password = form.cleaned_data['password']
            email = form.cleaned_data['email']
            contact = form.cleaned_data['contact']

        context = {'username': username, 'password': password,
'email': email, 'contact': contact}
    return render(request, 'q3/success.html', context)
```

q3/forms.py

```
from django import forms
from django.forms import widgets
from django.forms.fields import CharField, EmailField,
IntegerField


class BasicForm(forms.Form):
   username = CharField(required=True)
   password = CharField(widget=widgets.PasswordInput(),
required=False)
   email = CharField(widget=widgets.EmailInput(),
required=False)
   contact=
IntegerField(min_value=1000000000,max_value=9999999999,required=
False)
```

q3/urls.py

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

app_name = 'q3'
urlpatterns = [
   path('', views.index, name='index'),
   path('register', views.register, name='register'),
]
```

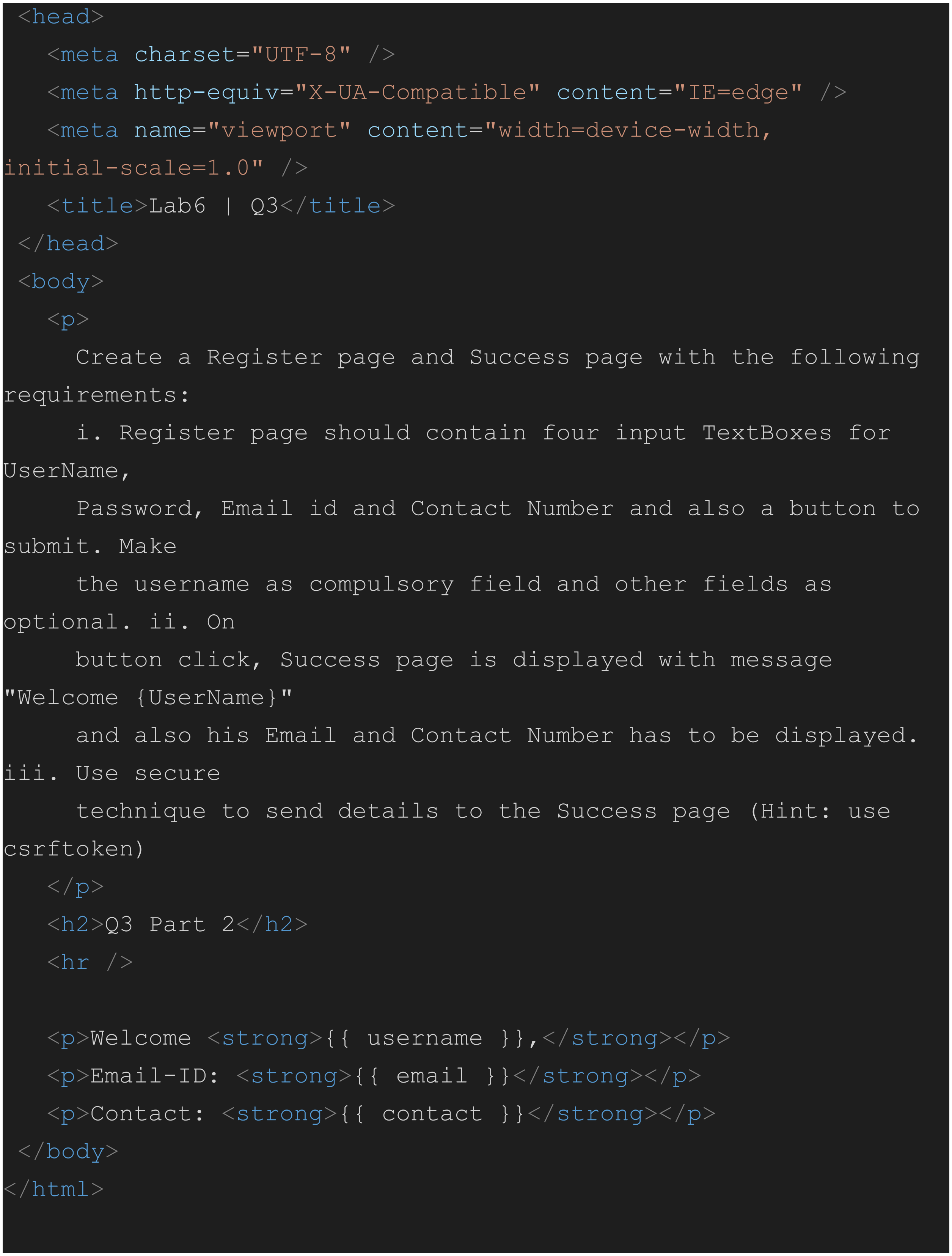
q3/templates/q3/register.html

```
<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>Lab6 | Q3</title>
 </head>
 <body>
   <p>
     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)
   </p>
   <h2>Q3 Part 1 </h2>
   <hr />

   <form method="POST" action="register">
     {% csrf_token %} {{ form.as_p }}
     <button type="submit">Submit</button>
   </form>
 </body>
</html>
```

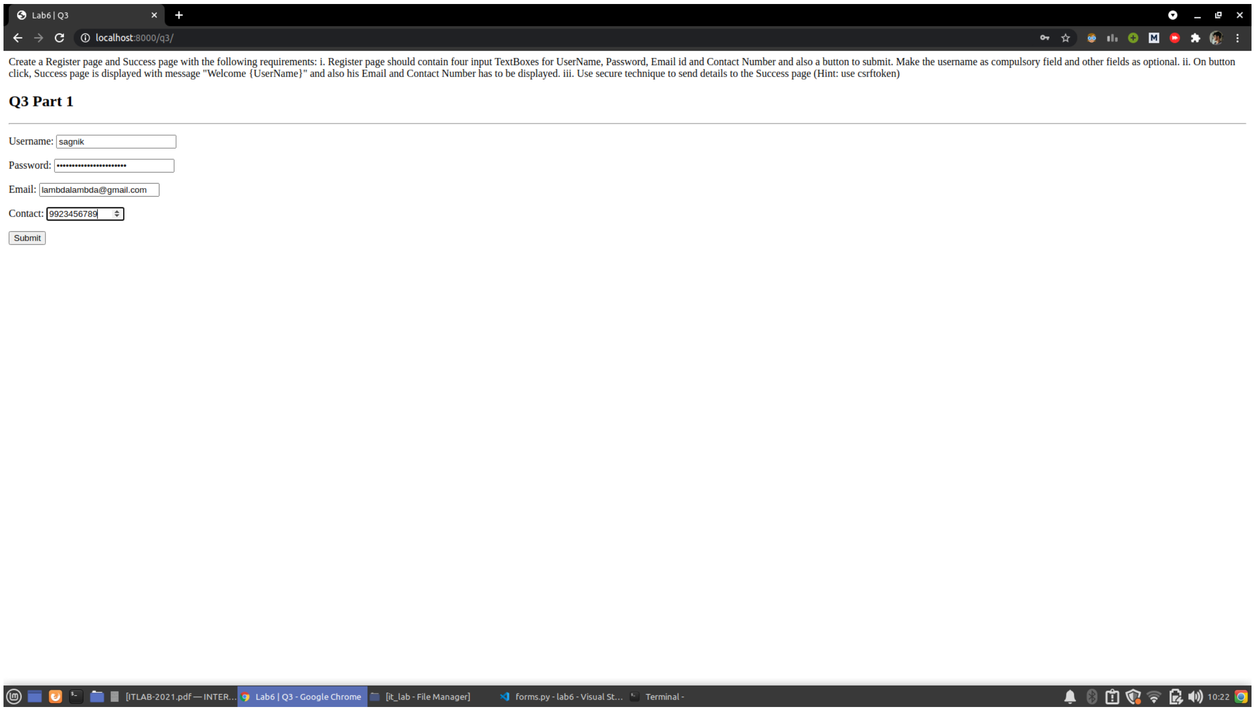
q3/templates/q3/success.html

```
<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>Lab6 | Q3</title>
 </head>
 <body>
   <p>
     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)
   </p>
   <h2>Q3 Part 2</h2>
   <hr />

   <p>Welcome <strong>{{ username }},</strong></p>
   <p>Email-ID: <strong>{{ email }}</strong></p>
   <p>Contact: <strong>{{ contact }}</strong></p>
 </body>
</html>
```

Img1 :


Lab6 | Q3
localhost:8000/q3/
[ITLAB-2021.pdf — INTER...
Lab6 | Q3 - Google Chrome
[it_lab - File Manager]
forms.py - lab6 - Visual St...
Terminal -
10:22

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 Part 1

Username: sagnik

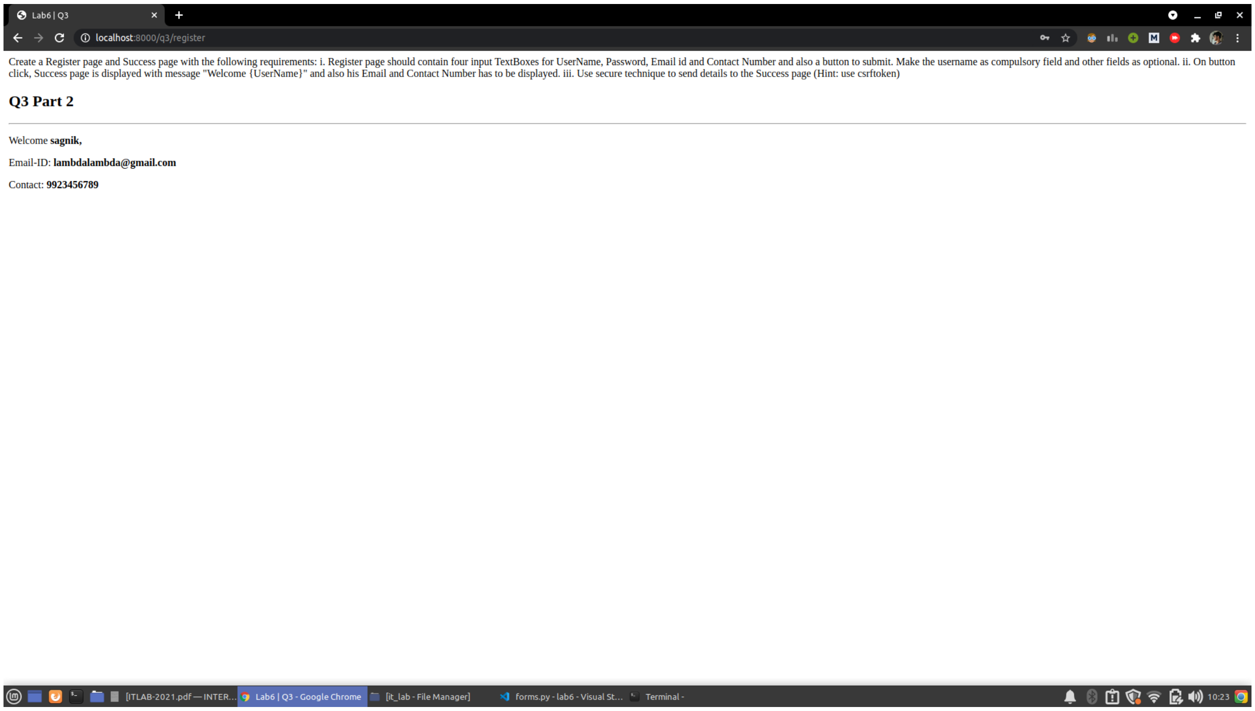
Password: ••••••••••••••••••••••••

Email: lambdalambda@gmail.com

Contact: 9923456789

Submit

Img2 :


Lab6 | Q3
localhost:8000/q3/register
[ITLAB-2021.pdf — INTER...
Lab6 | Q3 - Google Chrome
[it_lab - File Manager]
forms.py - lab6 - Visual St...
Terminal -
10:23

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)

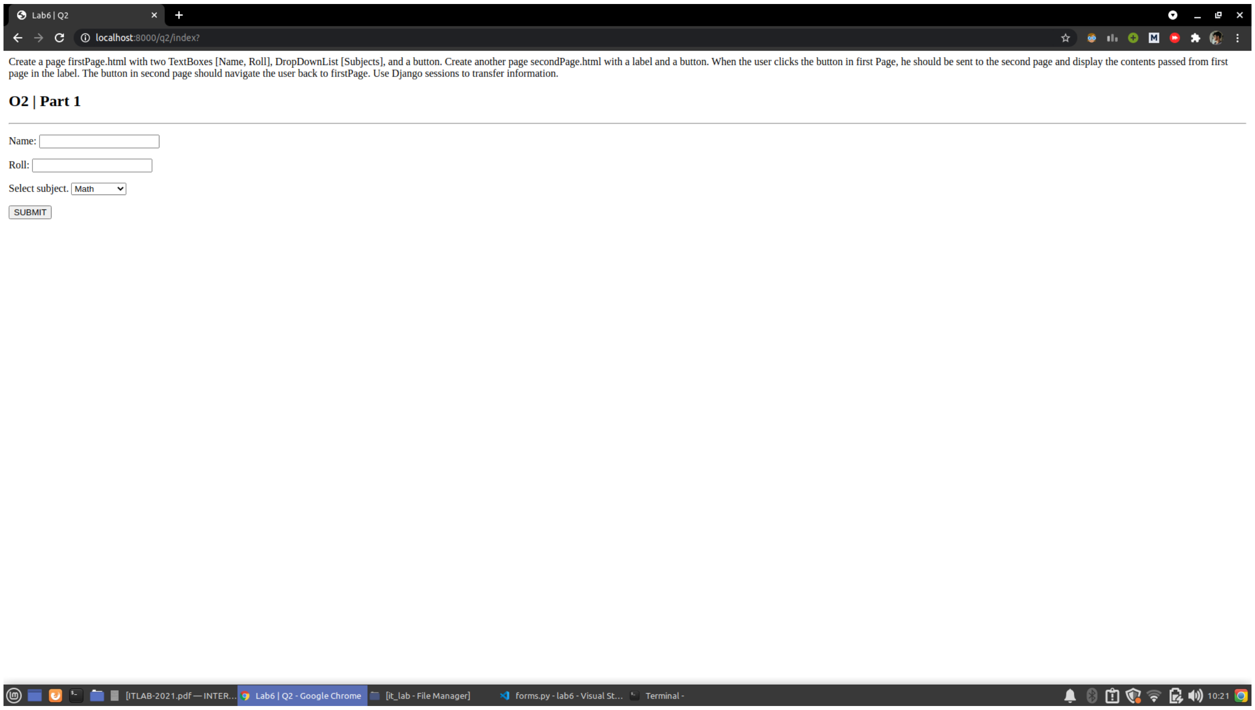
Q3 Part 2

Welcome **sagnik,**

Email-ID: **lambdalambda@gmail.com**

Contact: **9923456789**

Img3 :



Lab6 | Q2
localhost:8000/q2/index?
[ITLAB-2021.pdf — INTER...
Lab6 | Q2 - Google Chrome
[it_lab - File Manager]
forms.py - lab6 - Visual St...
Terminal -
10:21

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.

O2 | Part 1

Name:

Roll:

Select subject. Math

SUBMIT

Q4 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/views.py

```
from django.shortcuts import render, redirect
from .forms import BasicForm

prices = {
    'HP': {'mobile': 5000, 'laptop': 15000},
    'Nokia': {'mobile': 6000, 'laptop': 16000},
    'Samsung': {'mobile': 7000, 'laptop': 17000},
    'Motorolla': {'mobile': 8000, 'laptop': 18000},
    'Apple': {'mobile': 9000, 'laptop': 19000},
}
```

```
def index(request):
    form = BasicForm()
    return render(request, 'q4/first_page.html', {'form': form})

def produce_bill(request):
    if request.method == 'POST':
        form = BasicForm(request.POST)
        if form.is_valid():
            company = form.cleaned_data['company']
            mobile = form.cleaned_data['mobile']
            laptop = form.cleaned_data['laptop']
            quantity = form.cleaned_data['quantity']

            total_bill = 0

            if (mobile):
                total_bill = total_bill +
prices[company]['mobile']
            if (laptop):
                total_bill = total_bill +
prices[company]['laptop']


            context = {'company': company, 'mobile': mobile,
'laptop': laptop, 'quantity': quantity, 'total_bill':
total_bill}
        else:
            context = {}
    return render(request, 'q4/second_page.html', context)
```

q4/urls.py

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

app_name = 'q4'
urlpatterns = [
    path('', views.index, name='index'),
    path('produce_bill', views.produce_bill,
name='produce_bill'),
]
```

q4/forms.py

```
from django import forms
from django.forms import widgets
from django.forms.fields import CharField, EmailField,
IntegerField

companies = [
    ('HP', 'HP'),
    ('Nokia', 'Nokia'),
    ('Samsung', 'Samsung'),
    ('Motorolla', 'Motorolla'),
    ('Apple', 'Apple'),
]

class BasicForm(forms.Form):
    company = forms.ChoiceField(widget=forms.RadioSelect,
choices=companies)
    mobile = forms.BooleanField(required=False)
    laptop = forms.BooleanField(required=False)
    quantity = forms.IntegerField()
```

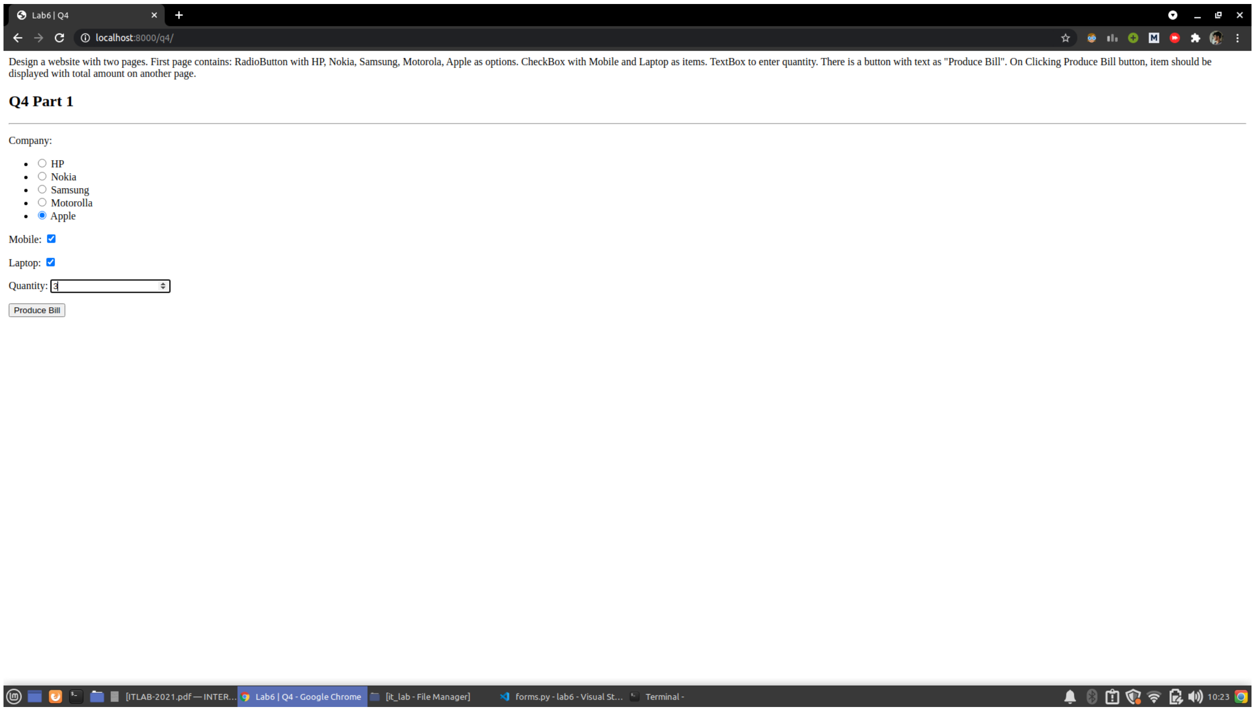
q4/templates/q4/first_page.html

```
<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>Lab6 | Q4</title>
 </head>
 <body>
   <p>
     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.
   </p>
   <h2>Q4 Part 1 </h2>
   <hr />

   <form method="POST" action="produce_bill">
     {% csrf_token %} {{ form.as_p }}
     <button type="submit">Produce Bill</button>
   </form>
 </body>
</html>
```

q4/templates/q4/second_page.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>Lab6 | Q4</title>
 </head>
 <body>
   <p>
     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.
   </p>
   <h2>Q4 Part 2</h2>
   <hr />

   <p>Company: <strong>{{ company }}</strong></p>

   <p>
     Mobile: {% if mobile %}
     <strong> yes </strong>
     {% else %}
     <strong> no </strong>
     {% endif %}
   </p>

   <p>
     Laptop: {% if laptop %}
     <strong> yes </strong>
     {% else %}
     <strong> no </strong>
     {% endif %}
   </p>

   <p>Quantity: <strong>{{ quantity }}</strong></p>
   <br />
   <p>Total Bill: <strong>Rs. {{ total_bill }}</strong></p>
 </body>
</html>
```

Img1:



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 Part 1

Company:

- ○ HP
- ○ Nokia
- ○ Samsung
- ○ Motorolla
- ● Apple

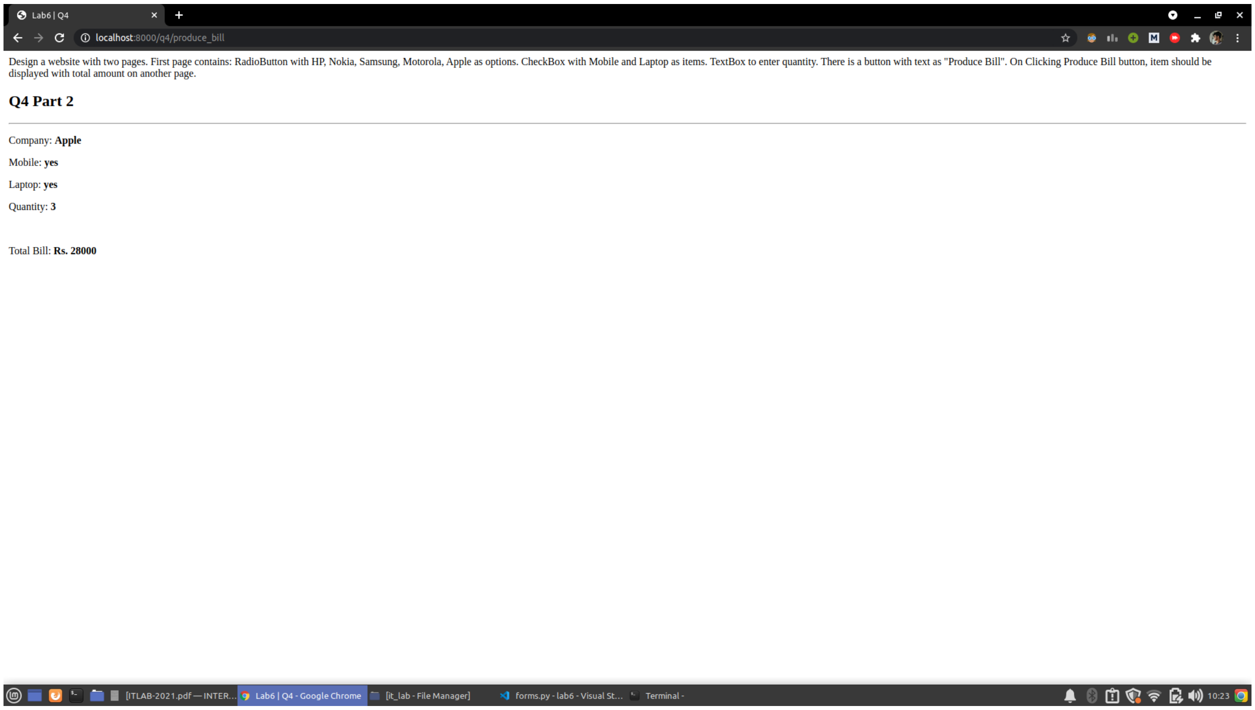
Mobile: ☑

Laptop: ☑

Quantity: 3

Produce Bill
[ITLAB-2021.pdf — INTER...
Lab6 | Q4 - Google Chrome
[it_lab - File Manager]
forms.py - lab6 - Visual St...
Terminal -
10:23

Img2:


Lab6 | Q4
localhost:8000/q4/produce_bill
[ITLAB-2021.pdf — INTER...
Lab6 | Q4 - Google Chrome
[it_lab - File Manager]
forms.py - lab6 - Visual St...
Terminal -
10:23

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 Part 2

Company: **Apple**

Mobile: **yes**

Laptop: **yes**

Quantity: **3**

Total Bill: **Rs. 28000**