

## LAB 8: Form Processing using Django — Part II

Aditya Agarwal – 220905106 – 14

- 1) 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)
  - iv. Design a website with two pages.

### q1/settings.py

```
import os
INSTALLED_APPS = [
    'usersapp',
    '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',
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'usersapp/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',
],
},
},
]
```

## q1/urls.py

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

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

## usersapp/urls.py

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

urlpatterns = [
    path('', views.register_view, name='register'),
    path('success/', views.success_view, name='success'),
]
```

## usersapp/forms.py

```
from django import forms

class RegistrationForm(forms.Form):
    username = forms.CharField(max_length=100, required=True)
    password = forms.CharField(widget=forms.PasswordInput, required=False)
    email = forms.EmailField(required=False)
    contact = forms.CharField(max_length=15, required=False)
```

## usersapp/views.py

```
from django.shortcuts import render, redirect
from django.contrib import messages
from .forms import RegistrationForm

# Register Page View
```

```

def register_view(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        if form.is_valid():
            # Get form data
            username = form.cleaned_data['username']
            password = form.cleaned_data['password']
            email = form.cleaned_data['email']
            contact = form.cleaned_data['contact']

            # Save data in session
            request.session['username'] = username
            request.session['password'] = password
            request.session['email'] = email
            request.session['contact'] = contact

            # Redirect to Success page
            return redirect('success')
        else:
            # If form is invalid, display error messages
            messages.error(request, 'Please correct the errors below.')
        else:
            form = RegistrationForm()

    return render(request, 'register.html', {'form': form})

# Success Page View
def success_view(request):
    username = request.session.get('username')
    email = request.session.get('email')
    contact = request.session.get('contact')

    if username:
        return render(request, 'success.html', {
            'username': username,
            'email': email,
            'contact': contact,
        })
    else:
        return redirect('register')

```

## usersapp/templates/register.html

```

<!DOCTYPE html>
<html>

```

```

<head>
<title>Register</title>
</head>
<body>
<h2>Register</h2>

<form method="POST">
{% csrf_token %}
{{ form.as_p }} <!-- This will automatically render all form fields -->
<button type="submit">Submit</button>
</form>
</body>
</html>

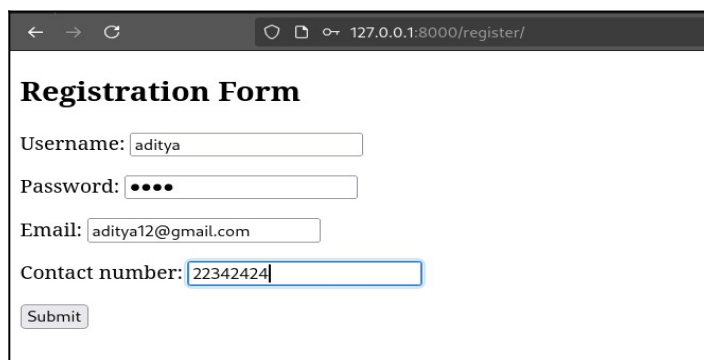
```

## usersapp/templates/success.html

```

<!DOCTYPE html>
<html>
<head>
<title>Success</title>
</head>
<body>
<h2>Welcome {{ username }}!</h2>
<p>Email: {{ email }}</p>
<p>Contact: {{ contact }}</p>
</body>
</html>

```



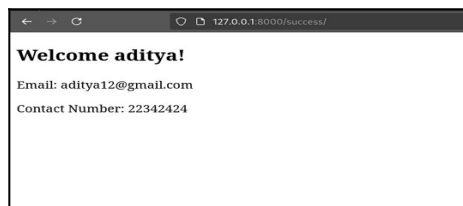
Registration Form

Username:

Password:

Email:

Contact number:



Welcome aditya!

Email: aditya12@gmail.com

Contact Number: 22342424

2) “How is the book ASP.NET with c# by Vipul Prakashan?” Give the user three

choice: i) Good ii) Satisfactory iii) Bad. Provide a VOTE button. After user votes, present the result in percentage using labels next to the choice

## q2/settings.py

```
import os
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',
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'voteapp/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',
            ],
        },
    },
]
```

## q2/urls.py

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

urlpatterns = [
```

```
path('admin/', admin.site.urls),
path('', include('voteapp.urls')), # Include the voteapp URLs here
]
```

## **voteapp/urls.py**

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

urlpatterns = [
    path('', views.vote_view, name='vote'),
]
```

## **voteapp/views.py**

```
from django.shortcuts import render
from django.http import HttpResponseRedirect
from .forms import VoteForm

def vote_view(request):
    # Initialize votes in session if not already present
    if 'votes' not in request.session:
        request.session['votes'] = {'Good': 0, 'Satisfactory': 0, 'Bad': 0}
    if request.method == "POST":
        form = VoteForm(request.POST)
        if form.is_valid():
            vote = form.cleaned_data['vote']
            # Increment the vote count in the session
            request.session['votes'][vote] += 1
            request.session.modified = True # Mark the session as modified to save it
        else:
            form = VoteForm() # Initialize an empty form when the page is first loaded
            # Calculate percentages based on the votes in the session
            total_votes = sum(request.session['votes'].values())
            percentages = {}
            if total_votes > 0:
                for choice, count in request.session['votes'].items():
                    percentages[choice] = (count / total_votes) * 100
            else:
                percentages = {choice: 0 for choice in request.session['votes']}
            return render(request, 'vote.html', {
                'form': form,
                'votes': request.session['votes'],
                'percentages': percentages
            })
```

## voteapp/forms.py

```
from django import forms

class VoteForm(forms.Form):
    VOTE_CHOICES = [
        ('Good', 'Good'),
        ('Satisfactory', 'Satisfactory'),
        ('Bad', 'Bad'),
    ]
    vote = forms.ChoiceField(choices=VOTE_CHOICES, widget=forms.RadioSelect)
```

## voteapp/templates/vote.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Vote for the Book</title>
</head>
<body>
<p>How is the book "ASP.NET with C#" by Vipul Prakashan?<p>
<form method="post">
{% csrf_token %}
<label>
<input type="radio" name="vote" value="Good"> Good
{% if percentages.Good %}
({{ percentages.Good|floatformat:2 }}%)
{% endif %}
</label><br>
<label>
<input type="radio" name="vote" value="Satisfactory"> Satisfactory
{% if percentages.Satisfactory %}
({{ percentages.Satisfactory|floatformat:2 }}%)
{% endif %}
</label><br>
<label>
<input type="radio" name="vote" value="Bad"> Bad
{% if percentages.Bad %}
({{ percentages.Bad|floatformat:2 }}%)
{% endif %}
</label><br>
<button type="submit">Vote</button>
</form>
```

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

How is the book "ASP.NET with C#" by Vipul Prakashan?

- ☐ Good (57.14%)
- ☐ Satisfactory (28.57%)
- ☐ Bad (14.29%)

Vote

- 3) Create a website with two pages. Page 1 has two TextBoxes (name and total marks) and one 'Calculate' Button as shown in the figure. On clicking the 'Calculate' Button, CGPA (total marks/50) along with the name should be displayed in the Page Use Django sessions to store the information.

### q3/settings.py

```
import os
INSTALLED_APPS = [
    'cgpa_app',
    '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',
]

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [os.path.join(BASE_DIR, 'cgpa_app/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',
],
},
},
]
```

### **q3/urls.py**

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

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

### **cgpa\_app/urls.py**

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

urlpatterns = [
    path('', views.input_page, name='input_page'),
    path('result/', views.result_page, name='result_page'),
]
```

### **cgpa\_app/views.py**

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

def input_page(request):
    if request.method == "POST":
        # Process the form data when the form is submitted
        form = CGPAForm(request.POST)
        if form.is_valid():
            # Get the cleaned data from the form
            name = form.cleaned_data['name']
            total_marks = form.cleaned_data['total_marks']

            # Calculate CGPA
            cgpa = total_marks / 50
```

```

# Store the name and CGPA in the session
request.session['name'] = name
request.session['cgpa'] = cgpa

# Redirect to the result page
return redirect('result_page')
else:
form = CGPAForm() # Create an empty form if the request is GET

return render(request, 'input_page.html', {'form': form})

def result_page(request):
# Retrieve the name and CGPA from the session
name = request.session.get('name')
cgpa = request.session.get('cgpa')

# If no data exists in the session, redirect back to the input page
if not name or not cgpa:
return redirect('input_page')

return render(request, 'result_page.html', {'name': name, 'cgpa': cgpa})

```

### **cgpa\_app/forms.py**

```

from django import forms

class CGPAForm(forms.Form):
name = forms.CharField(max_length=100, label='Name')
total_marks = forms.FloatField(label='Total Marks')

```

### **cgpa\_app/templates/input\_page.html**

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Enter Your Details</title>
</head>
<body>
<h1>Enter Your Name and Total Marks</h1>
<!-- Render the form using Django's form rendering -->
<form method="POST">
{% csrf token %}

```

```
{{ form.as_p }} <!-- This will render the form fields as paragraphs -->
<button type="submit">Calculate</button>
</form>
</body>
</html>
```

## cgpa\_app/templates/result\_page.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>CGPA Result</title>
</head>
<body>
<h1>CGPA Result</h1>
<p>Name: {{ name }}</p>
<p>Your CGPA: {{ cgpa|floatformat:2 }}</p>
</body>
</html>
```



A screenshot of a web browser showing a form titled "Enter Your Name and Total Marks". The form has two input fields: "Name:" with the value "adityaagarwal" and "Total Marks:" with the value "50". Below the fields is a "Calculate" button. The browser's address bar shows "127.0.0.1:8000".



A screenshot of a web browser showing the result page titled "CGPA Result". The page displays the name "Name: adityaagarwal" and the calculated CGPA "Your CGPA: 1.00". The browser's address bar shows "127.0.0.1:8000/result/".