Lab 09 – Web Programming Lab

Arhaan Girdhar – 220962050

Q1. 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) 4) Design a website with two pages.**

Views.py

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


def register(request):
  if request.method == "POST":
    form = RegisterForm(request.POST)
    if form.is_valid():
      username = form.cleaned_data.get('username')
      email = form.cleaned_data.get('email')
      contact = form.cleaned_data.get('contact')
      return render(request, 'registerapp/success.html', {
        'username': username,
        'email': email,
        'contact': contact,
      })
  else:
    form = RegisterForm()
  return render(request, 'registerapp/register.html', {'form': form})
```

Urls.py

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

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

Forms.py

```
from django import forms

class RegisterForm(forms.Form):
   username = forms.CharField(max_length=100, label="User Name", required=True)
   password = forms.CharField(widget=forms.PasswordInput, max_length=100, label="Password", required=False)
   email = forms.EmailField(label="Email", required=False)
   contact = forms.CharField(max_length=20, label="Contact Number", required=False)
```

Index.html

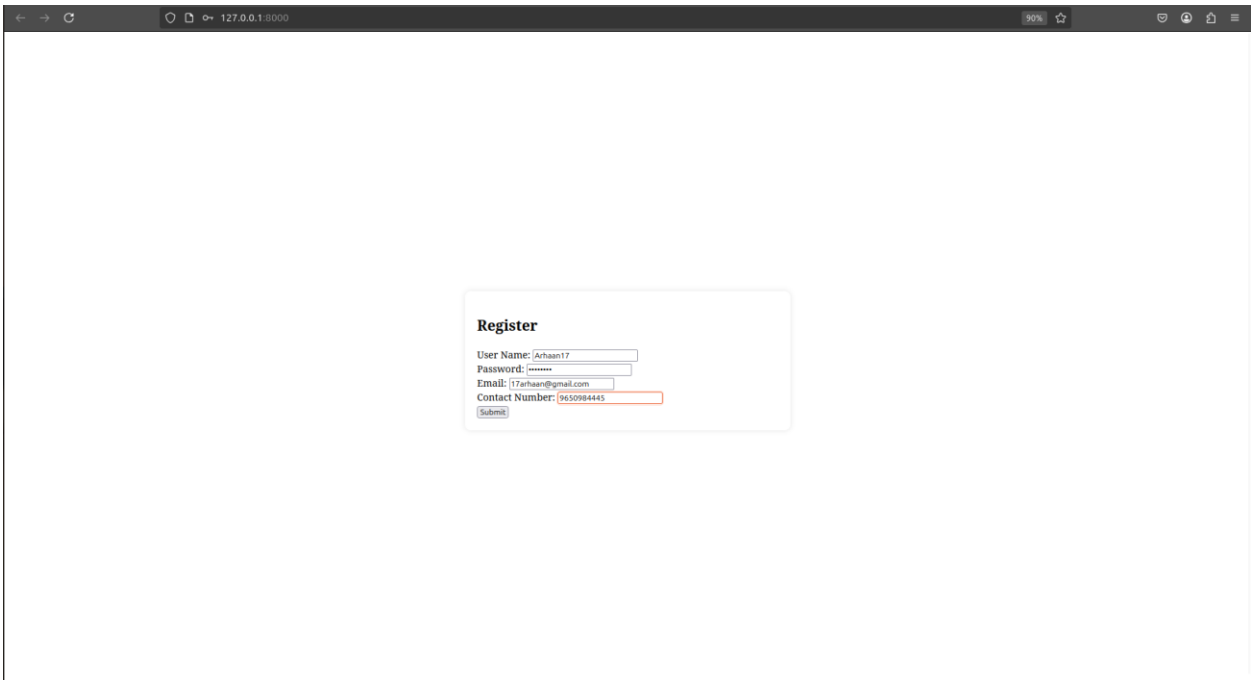
```
<!DOCTYPE html>
<html lang="en">
<head>
 <meta charset="UTF-8">
 <title>Register Page</title>
 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.6.0/css/bootstrap.min.css">
 <style>
   html, body { height: 100%; }
   .my-container { min-height: 100%; display: flex; align-items: center; justify-content: center; }
   .card { width: 100%; max-width: 500px; padding: 20px; box-shadow: 0 0 10px rgba(0,0,0,0.1); border-radius: 10px; }
 </style>
```

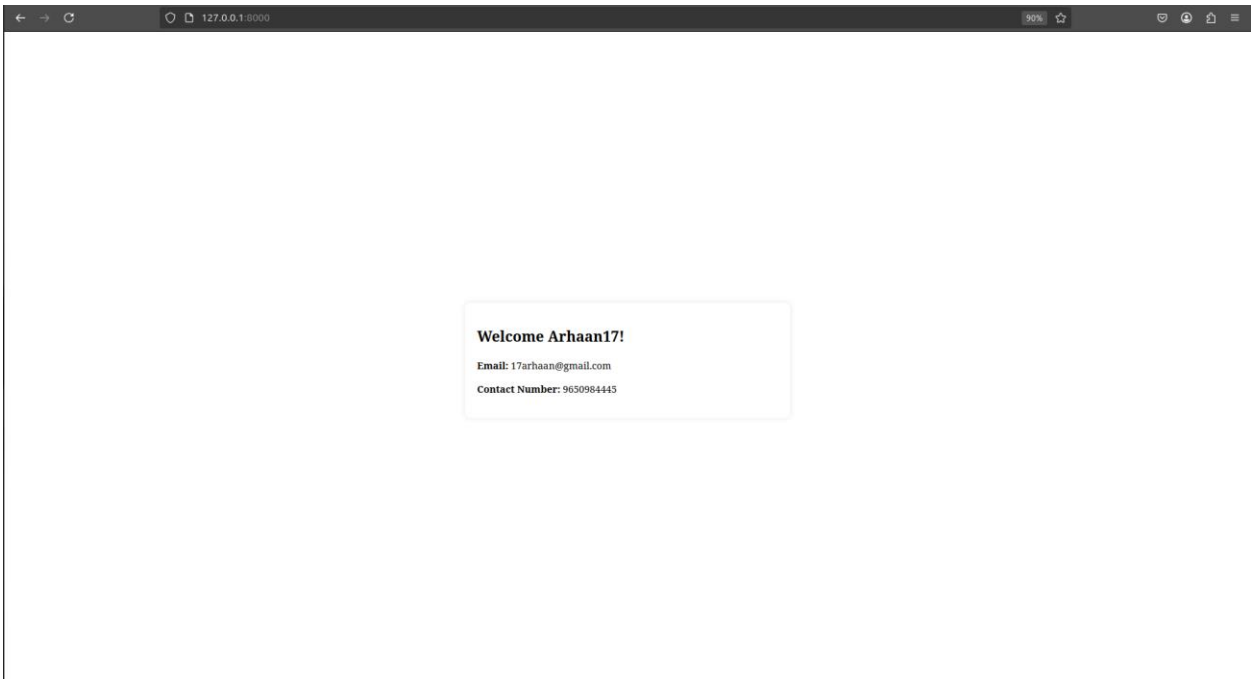
```
</head>
<body>
 <div class="container my-container">
  <div class="card">
   <h2 class="text-center mb-4">Register</h2>
   <form method="post">
    {% csrf_token %}
    <div class="form-group">
     {{ form.username.label_tag }}
     {{ form.username }}
    </div>
    <div class="form-group">
     {{ form.password.label_tag }}
     {{ form.password }}
    </div>
    <div class="form-group">
     {{ form.email.label_tag }}
     {{ form.email }}
    </div>
    <div class="form-group">
     {{ form.contact.label_tag }}
     {{ form.contact }}
    </div>
    <div class="text-center">
     <button type="submit" class="btn btn-primary">Submit</button>
    </div>
   </form>
  </div>
 </div>
```

```
 <script src="https://code.jquery.com/jquery-3.6.0.slim.min.js"></script>

 <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.0/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>
```

OUTPUT





Q2. 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 choices

Views.py

```
from django.shortcuts import render


VOTE_COUNTS = {'good': 0, 'satisfactory': 0, 'bad': 0}


def vote_view(request):
    total = sum(VOTE_COUNTS.values())
    percentages = {'good': 0, 'satisfactory': 0, 'bad': 0}
    if total > 0:
```

```
    percentages['good'] = round(VOTE_COUNTS['good'] * 100 / total)
    percentages['satisfactory'] = round(VOTE_COUNTS['satisfactory'] * 100 / total)
    percentages['bad'] = round(VOTE_COUNTS['bad'] * 100 / total)
  if request.method == 'POST':
    choice = request.POST.get('vote')
    if choice in VOTE_COUNTS:
      VOTE_COUNTS[choice] += 1
    total = sum(VOTE_COUNTS.values())
    if total > 0:
      percentages['good'] = round(VOTE_COUNTS['good'] * 100 / total)
      percentages['satisfactory'] = round(VOTE_COUNTS['satisfactory'] * 100 / total)
      percentages['bad'] = round(VOTE_COUNTS['bad'] * 100 / total)
  return render(request, 'voteapp/vote.html', {'percentages': percentages})
```

Urls.py

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

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

Apps.py

```
from django.apps import AppConfig
class VoteappConfig(AppConfig): default_auto_field = 'django.db.models.BigAutoField' name = 'voteapp'
```

Index.html

```
<!DOCTYPE html>
<html>
```

```
<head>
 <meta charset="UTF-8">
 <title>Vote</title>
 <style>
  html, body {
   margin: 0;
   padding: 0;
   height: 100%;
   display: flex;
   justify-content: center;
   align-items: center;
  }
 </style>
</head>
<body>
 <table border="0" cellpadding="5" cellspacing="5">
  <tr>
   <td>How is the book ASP.NET with c# by Vipul Prakashan?</td>
  </tr>
  <tr>
   <td>
    <form method="post">
     {% csrf_token %}
     <input type="radio" name="vote" value="good" id="good">
     <label for="good">Good</label>
     [{{ percentages.good }}%]
     <br>
     <input type="radio" name="vote" value="satisfactory" id="satisfactory">
     <label for="satisfactory">Satisfactory</label>
```

```
        [{{ percentages.satisfactory }}%]
        <br>
        <input type="radio" name="vote" value="bad" id="bad">
        <label for="bad">Bad</label>
        [{{ percentages.bad }}%]
        <br><br>
        <input type="submit" value="Vote">
      </form>
    </td>
  </tr>
 </table>
</body>
</html>
```

OUTPUT