TALENTSERVE SOFTWARE NANO PROJECT

Problem Statement:

Voting System. Its college representative election and you need to create a campaign page where the students can vote for their college representative depending upon campaign they suggest in their campaign page.

It should have two pages Campaign page where the College Participants would upload the problems they would be solving. Other page would be where the students can see the campaign and Vote for their representative. Also decide a page for unanimous selection of College Representative.

Programming Language: Python and Django for BackEnd and HTML CSS and Java Scrips for the Front End.

Software Functional Requirements:

Backend (Python and Django):

1.Set up a Django project and create a new Django app within the project.

2.Define the necessary models in the Django app's models.py file.

For this example, we'll have two models: Candidate and Vote.

3. Create the views in the Django app's views.py file. We'll have views for displaying the campaign page, the voting page, and the unanimous selection page.

4.Create the necessary URL patterns in the Django project's urls.py file.

Code Implementation:

Back End:

```
pip install Django

django-admin startproject talentserve_project

cd talentserve_project

python manage.py startapp timer
```

```
from django.db import models
from django.contrib.auth.models import User
# models.py
class Candidate(models.Model):
name = models.CharField(max_length=100)
campaign = models.TextField()
class Vote(models.Model):
candidate = models.ForeignKey(Candidate, on_delete=models.CASCADE)
voted_by = models.CharField(max_length=100)
# views.py
from django.shortcuts import render
from .models import Candidate, Vote
def campaign_page(request):
candidates = Candidate.objects.all()
return render(request, 'campaign.html', {'candidates': candidates})
def vote_page(request):
candidates = Candidate.objects.all()
return render(request, 'vote.html', {'candidates': candidates})
def unanimous_selection(request):
candidates = Candidate.objects.all()
return render(request, 'unanimous_selection.html', {'candidates': candidates})
```

```
# urls.py

from django.urls import path

from your_app_name import views

urlpatterns = [

path('campaign/', views.campaign_page, name='campaign'),

path('vote/', views.vote_page, name='vote'),

path('unanimous/', views.unanimous_selection, name='unanimous'),

]
```

Frontend (HTML, CSS, and JavaScript):

1.Create HTML templates for the campaign page (campaign.html), voting page (vote.html), and unanimous selection page (unanimous_selection.html) in the Django app's templates directory.

2.Customize the HTML templates based on your requirements.

3.Add CSS styling by creating a styles.css file in the static directory of Django app.

4.Repeat the HTML template and CSS styling process for the vote.html and unanimous_selection.html pages, customizing them according to your requirements.

Front End:

```
<!-- campaign.html -->

<!DOCTYPE html>

<html>

<head>

<title>Campaign Page</title>

<link rel="stylesheet" type="text/css" href="{% static 'styles.css' %}">
```

```
</head>
<body>
<h1>Campaign Page</h1>
<ul>
{% for candidate in candidates %}
<li>
<h2>{{ candidate.name }}</h2>
<p>{{ candidate.campaign }}</p>
</li>
{% endfor %}
</ul>
</body>
</html>
/* styles.css */
body {
font-family: Arial, sans-serif;
}
h1 {
color: #333;
}
h2 {
color: #666;
}
ul {
list-style-type: none;
}
li {
margin-bottom: 20px;
padding-bottom: 10px;
border-bottom: 1px solid #ccc;
}
```

@TalentServe

Intern Name: Gudiya.Divya

Intern ID: 379457

Intern College Name: RGUKT-SKLM