



NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Team Members

Student Name:

M.SATHIYAPRAKASH

Student ID :au912421104039

College Name

SHANMUGANATHAN
ENGINEERING COLLEGE

CAPSTONE PROJECT SHOWCASE

Project Title

**VOTING APPLICATION USING DJANGO
FRAMEWORK**

Abstract | Problem Statement | Project Overview | Proposed Solution |
Technology Used | Modelling & Results | Conclusion



Abstract

- 1.Objective:** Develop a user-friendly electronic voting (e-voting) application to modernize the voting process, ensuring accessibility and security.
- 2.Accessibility:** Prioritize user-centric design for ease of use, enabling voters of all demographics to cast their votes conveniently from any location with internet access.
- 3.Security Measures:** Implement robust encryption techniques, cryptographic protocols, and stringent authentication mechanisms to safeguard vote integrity and confidentiality, preventing tampering and unauthorized access.
- 4.Transparency:** Incorporate real-time result tracking and auditing features to enhance transparency and trust in the electoral process, enabling stakeholders to monitor the voting process and verify result accuracy.
- 5.Conclusion:** The proposed e-voting application aims to revolutionize democracy by making voting more accessible, efficient, and secure, contributing to the advancement of democratic practices in the digital age.

Problem Statement

- 1. Inefficiencies of Traditional Systems:** Traditional paper-based voting systems are plagued by logistical challenges, including long queues, time constraints, and high operational costs, leading to inefficiencies in the electoral process.
- 2. Security Vulnerabilities:** Existing voting systems are vulnerable to various forms of fraud, manipulation, and coercion, compromising the integrity of elections and eroding public trust in the electoral process.
- 3. Limited Accessibility:** Marginalized groups, such as the elderly, disabled, and geographically isolated individuals, face significant barriers to participation due to limited access to polling stations and cumbersome voting procedures.
- 4. Need for Innovation:** Addressing these challenges requires the development of an innovative electronic voting (e-voting) application that ensures the integrity, security, and accessibility of the voting process.
- 5. User-Friendly Interface:** The e-voting application must be user-friendly, enabling voters to cast their ballots conveniently from any location with internet access, thereby enhancing voter turnout and engagement.

Project Overview

1. Project Title: DjangoVote

2. Objective: Develop a robust electronic voting (e-voting) application using the Django framework to streamline the voting process, ensure security, and enhance accessibility for voters.

3. Key Features:

- **User Authentication:** Implement secure user authentication for voter registration and login.
- **Ballot Creation:** Allow election administrators to create and customize electronic ballots for different elections.
- **Vote Casting:** Enable registered voters to cast their ballots online securely.

5. Expected Outcomes:

- **Increased Efficiency:** By automating the voting process and reducing manual intervention.
- **Enhanced Security:** By leveraging Django's built-in security features to protect against vulnerabilities.

Proposed Solution

1.Overview:

1. The proposed solution involves developing an electronic voting (e-voting) application using the Django framework, a high-level Python web framework known for its simplicity, flexibility, and security features.

2.User Authentication:

1. Implement secure user authentication mechanisms using Django's built-in authentication system to ensure that only registered voters can access the application.

3.Ballot Creation and Customization:

1. Develop an intuitive interface for election administrators to create electronic ballots for different elections.
2. Allow administrators to customize ballot options, candidate profiles, and voting rules based on the specific requirements of each election.

1.Vote Casting:

1. Design a user-friendly interface for registered voters to cast their ballots securely online.
2. Ensure that the voting process is simple, intuitive, and accessible across different devices, including desktops, tablets, and smartphones.

2.Result Tabulation and Reporting:

1. Automate the tabulation of votes and calculation of election results based on the votes cast by registered voters.
2. Provide real-time reporting features to display election results to election officials, candidates, and voters as soon as voting closes.

3.Administration Panel:

1. Create an administration panel for election officials to manage voter registration, monitor voting activity, and verify election results.
2. Implement role-based access control to ensure that only authorized personnel can access sensitive election data.

1.Security Features:

1. Utilize Django's security features, including protection against common web vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF).
2. Implement encryption techniques and data validation measures to protect sensitive voter information and prevent unauthorized access to the application.

2.Scalability and Performance:

1. Design the application architecture to be scalable and capable of handling a large volume of users and concurrent voting sessions.
2. Optimize database queries, caching mechanisms, and server configurations to ensure optimal performance during peak voting periods.

3.Accessibility:

1. Adhere to web accessibility standards (WCAG) to ensure that the voting application is accessible to users with disabilities, including support for screen readers and keyboard navigation.

4.Deployment and Maintenance:

1. Deploy the e-voting application on a secure server infrastructure, ensuring high availability, reliability, and data integrity.
2. Provide ongoing maintenance and support to address any issues, perform updates, and implement enhancements based on user feedback and evolving requirements.

Technology Used

Front-end



Back-end



Modelling & Results

```
<form> {% csrf_token %}
{{ form.as_p }}
<Button>submit</button>
</form>

from django import forms
from django.contrib.auth.models import User
from django.contrib.auth.forms import AuthenticationForm

class RegistrationForm(forms.ModelForm):
    password1 = forms.CharField(label='Password', widget=forms.PasswordInput)
    password2 = forms.CharField(label='Password confirmation', widget=forms.PasswordInput)

    class Meta:
        model = User
        fields = ('username', 'first_name', 'last_name', 'email')
```

python

```
def clean_password2(self):
password1 = self.cleaned_data.get("password1")
password2 = self.cleaned_data.get("password2")
if password1 and password2 and password1!= password2:
raise forms.ValidationError("Passwords don't match")
return password2

def save(self, commit=True):
user = super(RegistrationForm, self).save(commit=False)
user.set_password(self.cleaned_data["password1"])
if commit:
user.save()
return user

class CustomAuthenticationForm(AuthenticationForm):
# You can customize the form fields here if needed
# For example, adding CSS classes or placeholders

username = forms.CharField(widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder':
'Username'}))
password = forms.CharField(widget=forms.PasswordInput(attrs={'class': 'form-control',
'placeholder': 'Password'}))
```

Modelling & Results

```
from django.shortcuts import render, redirect
from django.contrib.auth import authenticate, login
from .forms import RegistrationForm, CustomAuthenticationForm

def home(request):
    if request.method == 'POST':
        form = RegistrationForm(request.POST)
        if form.is_valid():
            form.save()
            return redirect('login')
        else:
            form = RegistrationForm()
            return render(request, 'pollingwebsite/signup.html', {'form': form})
    def home1(request):
        if request.method == 'POST':
            form = CustomAuthenticationForm(request, request.POST)
```

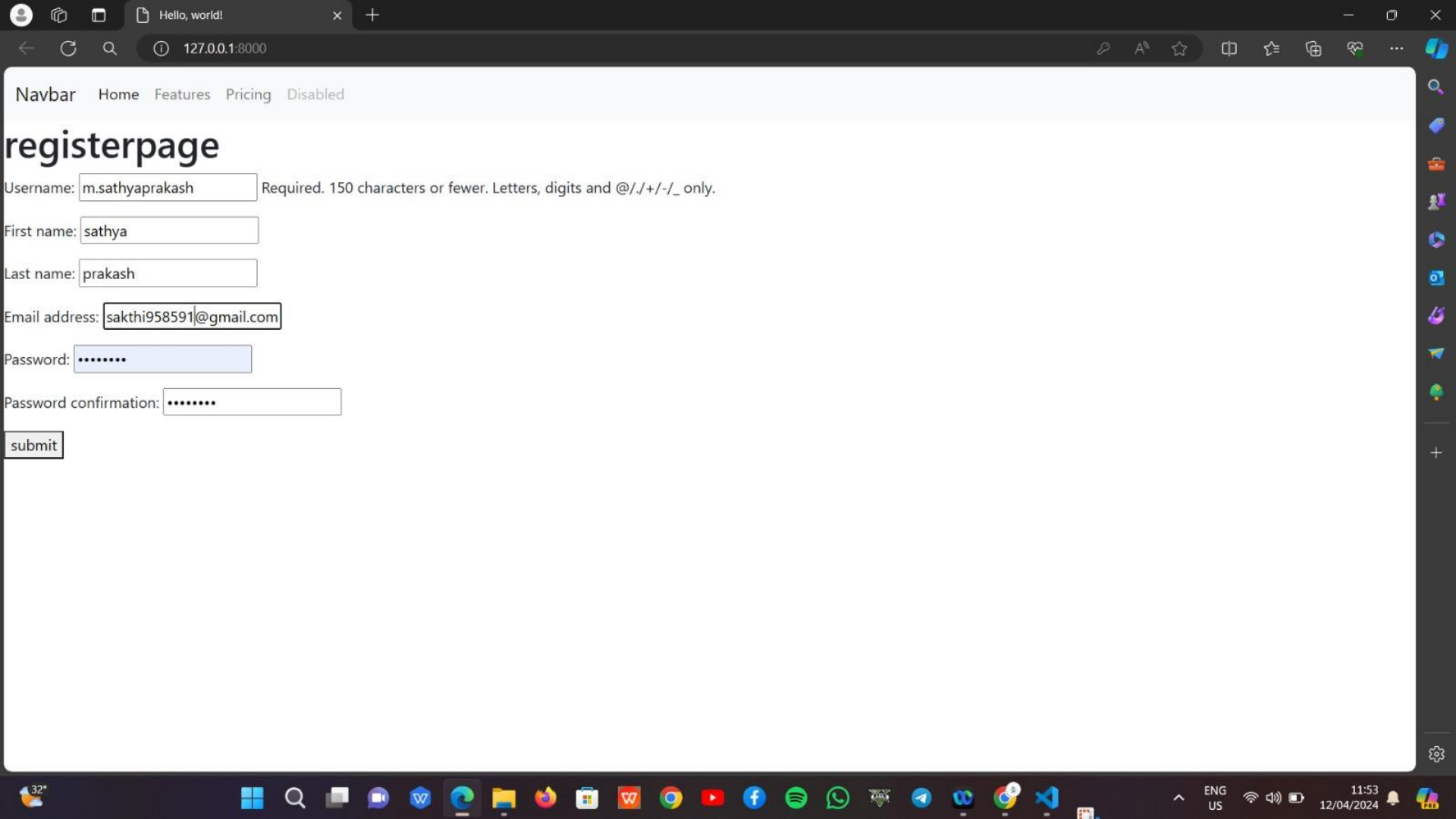
python

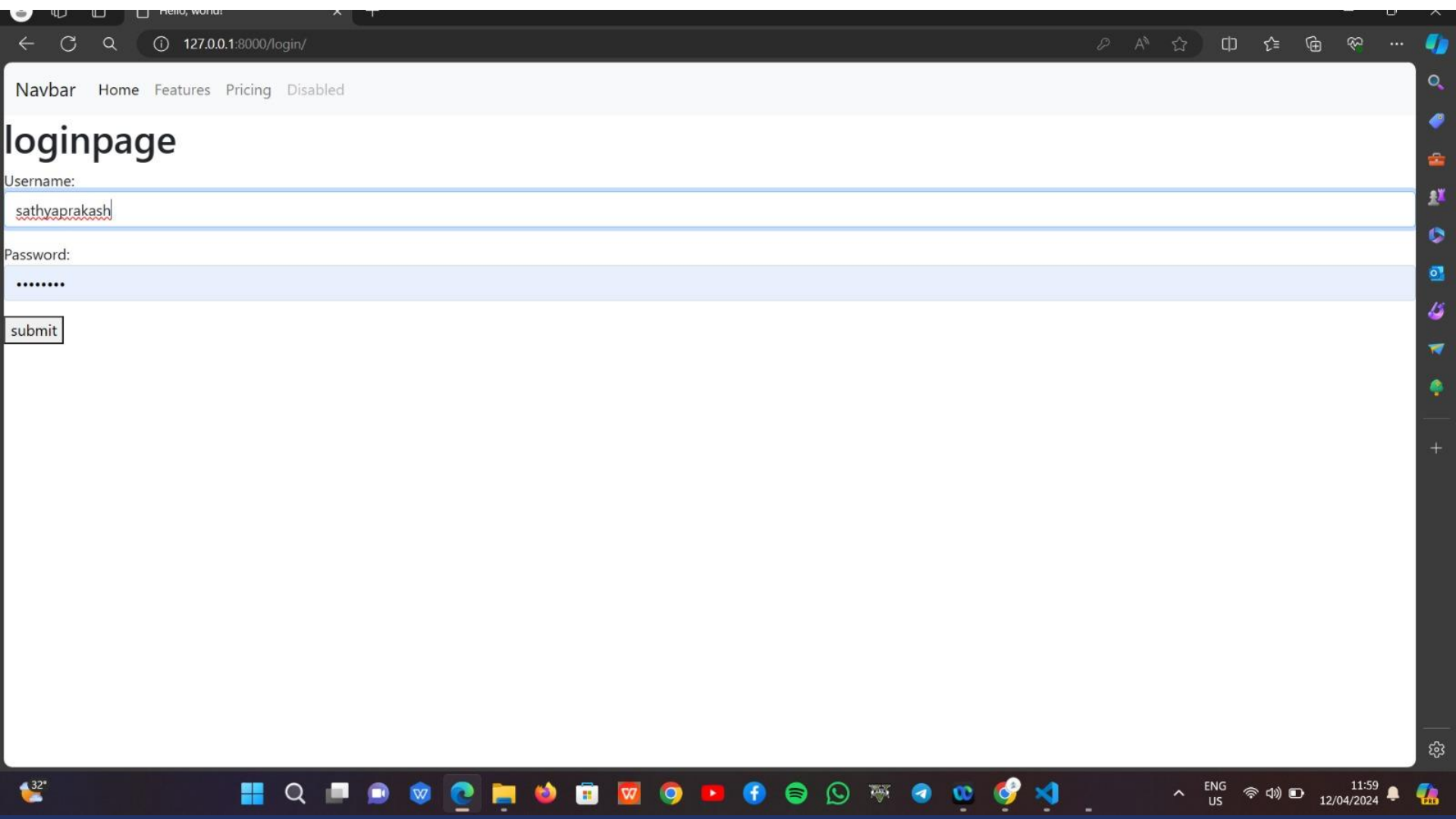
python

Modelling & Results

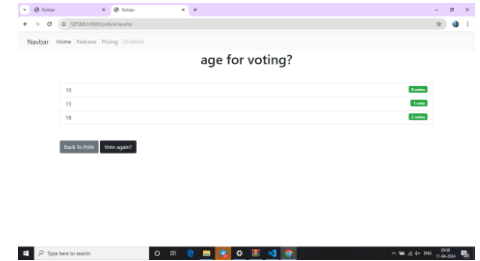
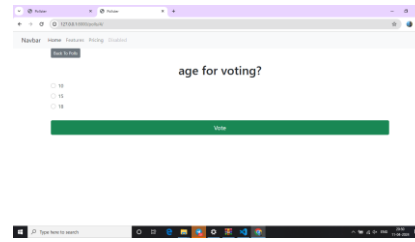
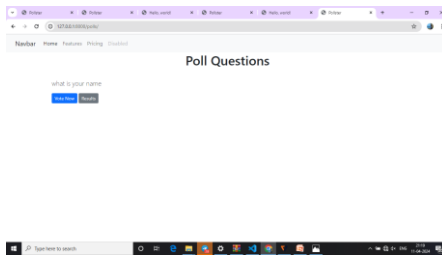
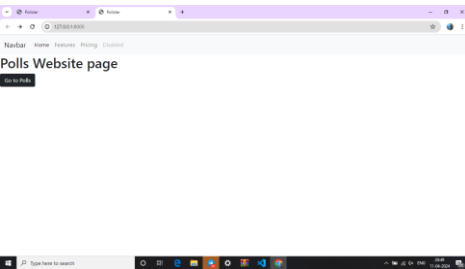
```
if form.is_valid():
    username = form.cleaned_data.get('username')
    password = form.cleaned_data.get('password')
    user = authenticate(request, username=username, password=password)
    if user is not None:
        login(request, user)
        return redirect('index')
    else:
        form = CustomAuthenticationForm()
        return render(request, 'pollingwebsite/login.html', {'form': form})
def home2(request):
    return render(request, 'pollingwebsite/index.html')
```

pyth

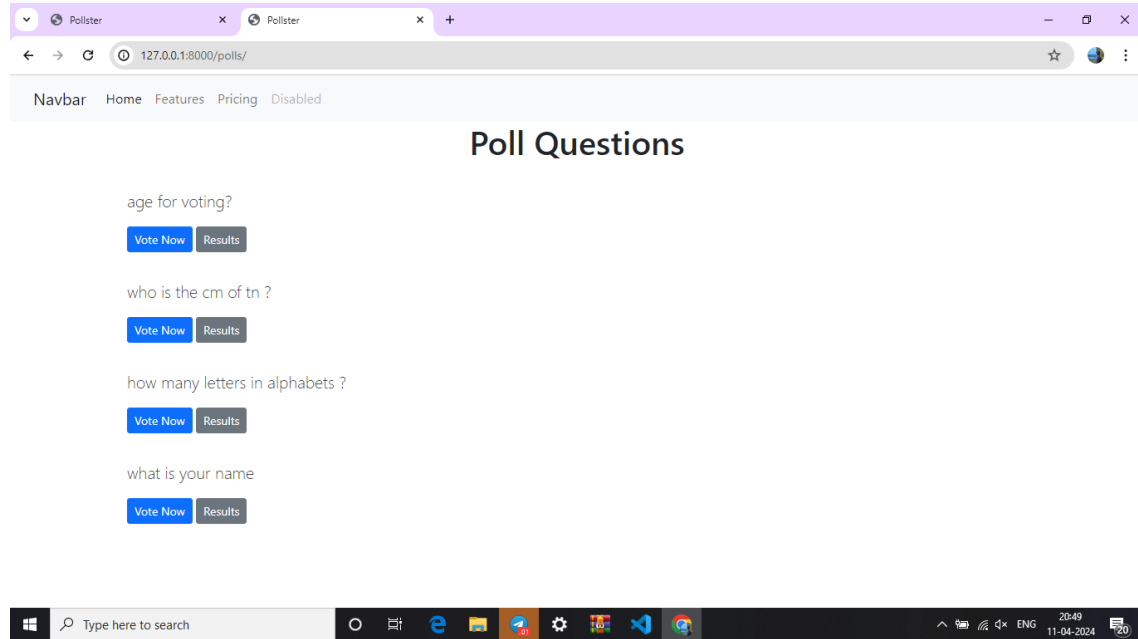
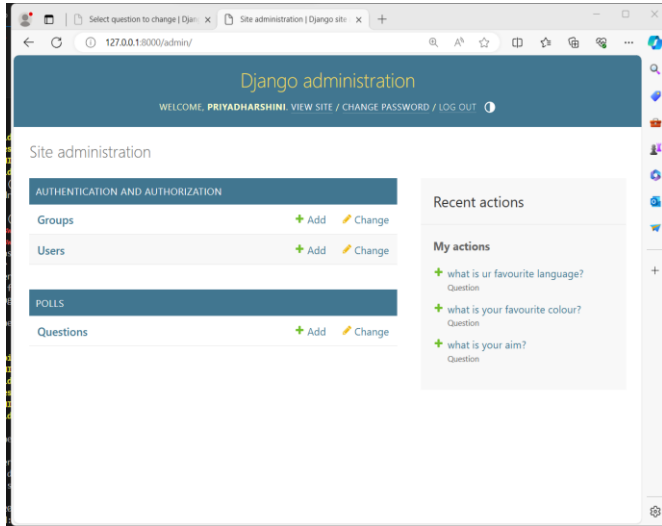




Service-Page



Departments-Page



Blog-Page

Title: "Empowering Democracy: Introducing Our Innovative Voting Application"

Introduction: In an age where technology is reshaping every aspect of our lives, it's time for democracy to catch up. Our voting application represents a leap forward in how we engage with the democratic process. In this blog post, we're thrilled to introduce you to our innovative voting application and explore how it's revolutionizing the way we participate in elections.

The Need for Change: Traditional voting methods are often cumbersome, time-consuming, and inaccessible to many. Long lines at polling stations, logistical challenges, and concerns about security and fraud have led to disenchantment with the electoral process. Our voting application seeks to address these issues by providing a modern, convenient, and secure way to cast your vote.

Future Enhancements:

User Authentication:

Implement user authentication to allow only registered users to vote.

Associate each vote with a user to prevent duplicate voting.

Multiple Choice Options:

Allow users to select multiple choices for a single poll.

Implement validation to ensure users do not exceed the allowed number of choices.

Conclusion

*The e-voting application developed using the Django framework represents a significant step towards modernizing democracy and promoting inclusive participation in electoral processes. As we continue to refine and enhance the application, we remain committed to upholding the principles of transparency, integrity, and accessibility in the democratic process.

Thank You!