



NEXT GEN EMPLOYABILITY PROGRAM

| Creating a future-ready workforce

Student Name :T.Raguvignesh
Student ID :au820621104065

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Voting Application using Django Framework T.Raguvignesh(4065,AEC)

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



Abstract

The proposed voting application is a web-based platform that allows users to create and participate in online votes. The application is built using the Django framework, a popular and well-supported Python-based web framework that provides a robust foundation for building scalable and secure web applications . The application is also designed to be flexible and scalable, with a modular architecture that allows for easy customization and extension. This makes it suitable for a wide range of use cases, from small-scale internal votes to large-scale public elections . Overall, the proposed voting application is a secure, user-friendly, and flexible platform for conducting online votes. Its use of the Django framework ensures a robust and scalable foundation, while its focus on security and user experience makes it an ideal choice for a wide range of voting scenarios.

Problem Statement

Online voting has become increasingly popular in recent years, with a growing number of organizations and governments turning to digital platforms to conduct elections and polls. However, online voting also presents a number of challenges, particularly in terms of security and integrity . Overall, the proposed voting application will address the challenges of security and integrity in online voting, while also providing a user-friendly platform for conducting online votes. Its use of the Django framework will ensure a robust and scalable foundation, while its focus on security and user experience will make it an ideal choice for a wide range of voting scenarios. In addition to its focus on security, the application will also prioritize user experience, with a clean and intuitive interface that makes it easy for users to create and participate in votes. The application will support multiple types of votes, including single-choice and multiple-choice votes, and will allow users to set deadlines and restrictions for each vote.

Project Overview

The project overview for a voting application using the Django framework involves creating a secure and user-friendly online voting system. The application allows users to register, vote, and view real-time results. Here is a steps involved in building the voting application:

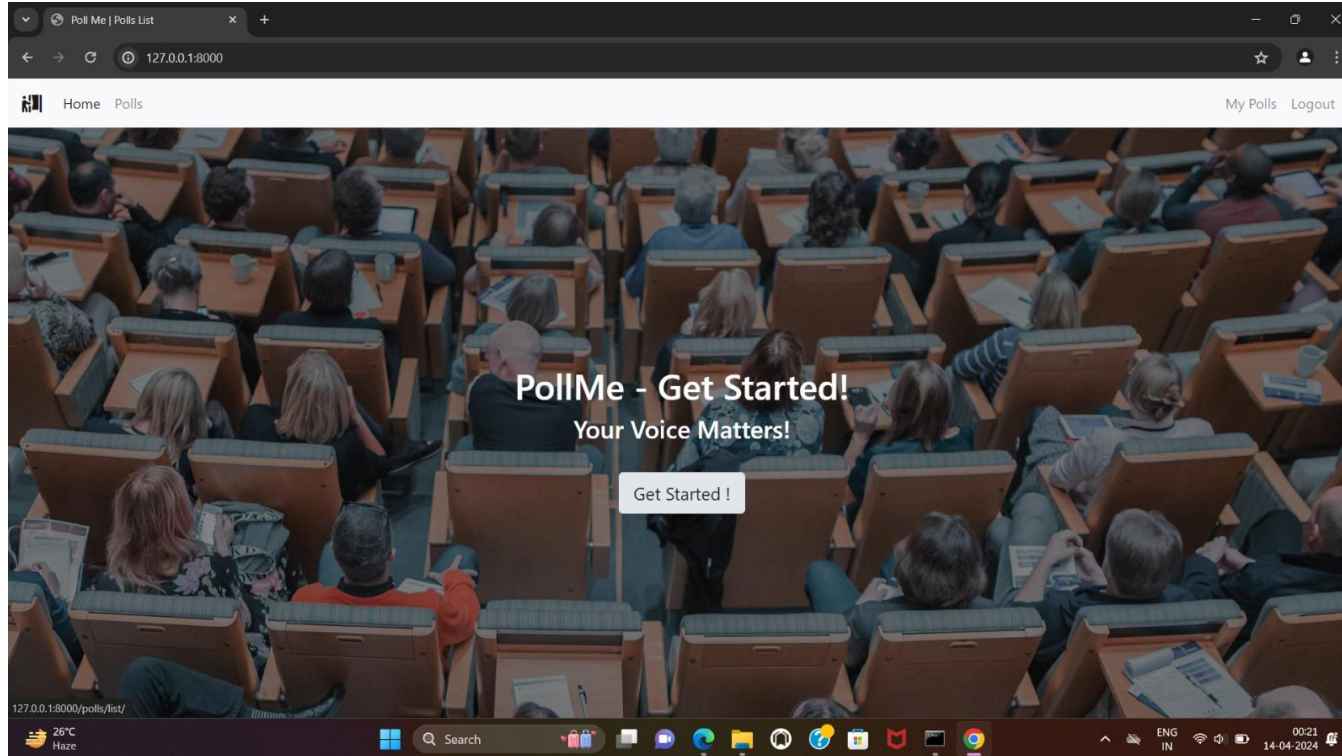
- 1.Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2.Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3.Creating User Authentication:** Implement user authentication to allow users to register, log in, and participate in voting.
- 4.Building the Voting Interface:** Develop the interface where users can view options, select their choices, and submit votes.
- 5.Implementing Real-time Results:** Display the voting results dynamically to provide instant feedback to users.
- 6.Developing an Admin Panel:** Build an admin panel to manage the voting process, candidates, and user accounts effectively.

Proposed Solution

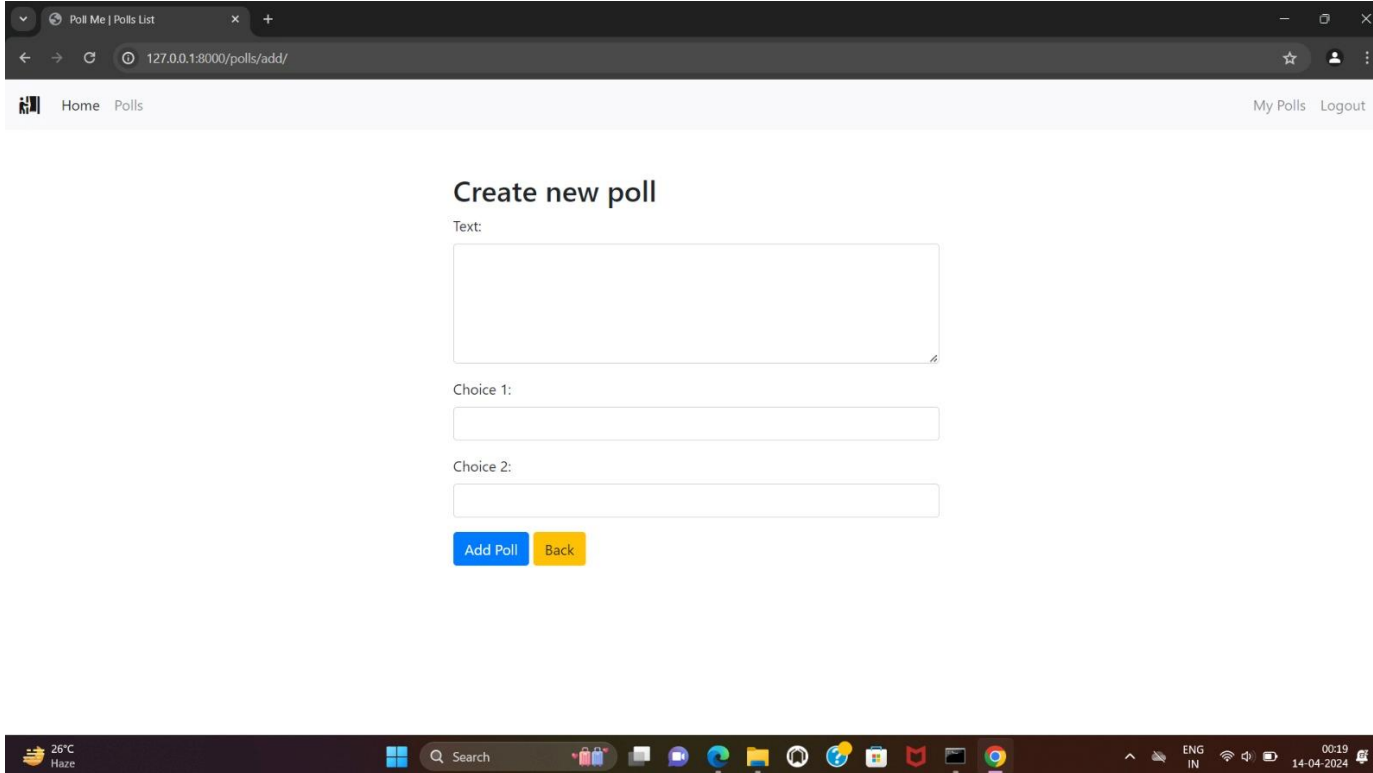
The proposed solution for a voting application using the Django framework is to create a secure and user-friendly online voting platform. The application will allow users to register, vote, and view real-time results. To build the application, the Django framework will be used as the foundation due to its robustness and scalability. The application will have a user-friendly interface, a secure database, real-time results, and an admin panel for efficient management of elections, candidates, and user accounts.

In summary, the proposed solution for a voting application using the Django framework is a secure, user-friendly, and flexible platform for conducting online votes. Its use of the Django framework ensures a robust and scalable foundation, while its focus on security and user experience makes it an ideal choice for a wide range of voting scenarios.

Home Page



Poll Page



Poll Me | Polls List

127.0.0.1:8000/polls/add/

Home Polls My Polls Logout

Create new poll

Text:

Choice 1:

Choice 2:

Add Poll Back

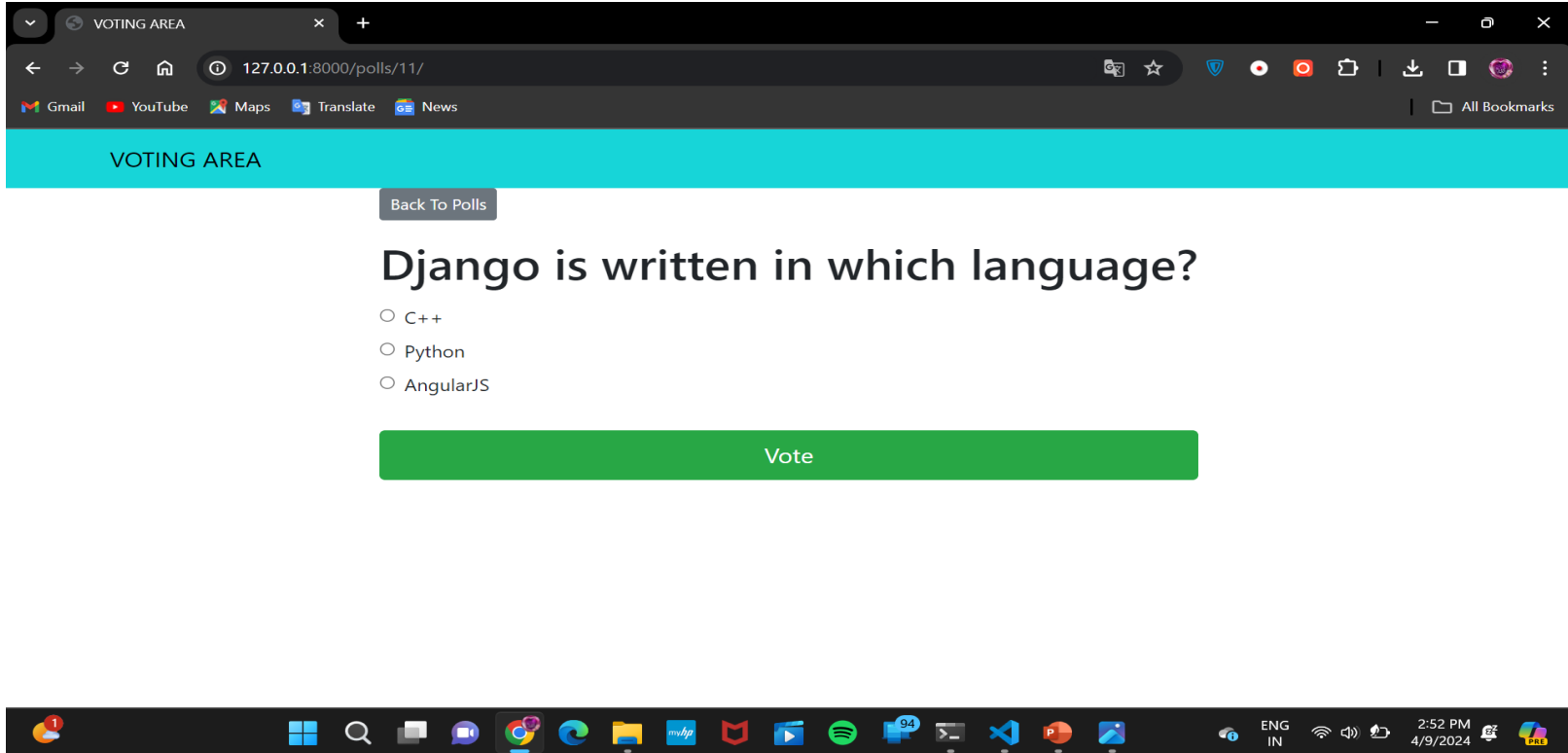
26°C Haze

Search

ENG IN

00:19 14-04-2024

Voting Page



The screenshot shows a web browser window with a single tab titled "VOTING AREA". The address bar displays the URL "127.0.0.1:8000/polls/11/". Below the address bar is a navigation bar with links to Gmail, YouTube, Maps, Translate, and News. The main content area has a light blue header with the text "VOTING AREA". Below this header is a button labeled "Back To Polls". The poll question is "Django is written in which language?". There are three radio button options: "C++", "Python", and "AngularJS". At the bottom of the poll is a large green button labeled "Vote". The Windows taskbar is visible at the bottom of the screen, showing various application icons and the system clock indicating 2:52 PM on 4/9/2024.

VOTING AREA

[Back To Polls](#)

Django is written in which language?

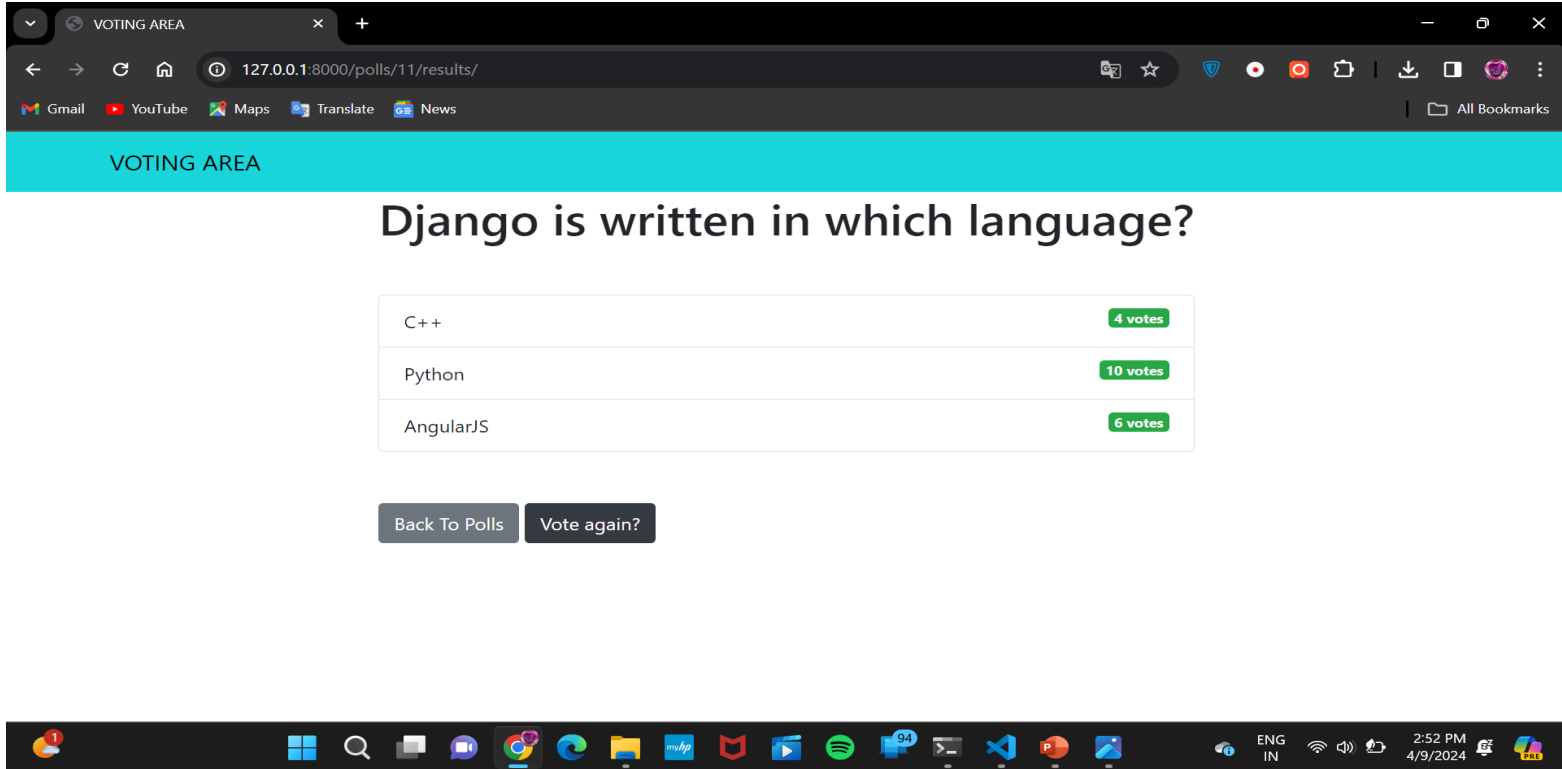
☐ C++

☐ Python

☐ AngularJS

[Vote](#)

Voting Details Page



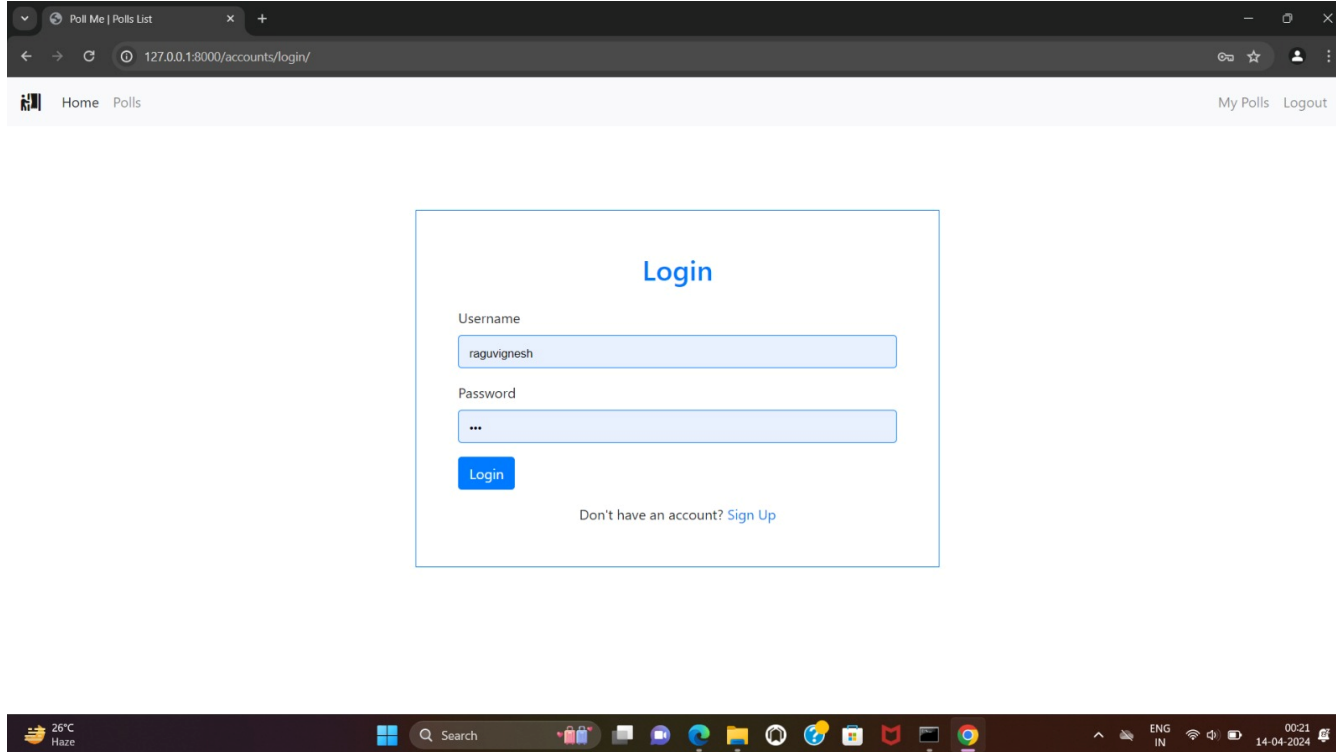
VOTING AREA

Django is written in which language?

C++	4 votes
Python	10 votes
AngularJS	6 votes

[Back To Polls](#) [Vote again?](#)

Admin Login Page



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/accounts/login/`. The browser's navigation bar includes links for Home and Polls, and user options for My Polls and Logout. The main content area features a login form with the title "Login". The form contains two input fields: "Username" with the text "raguvignesh" and "Password" with masked characters. Below the password field is a blue "Login" button. At the bottom of the form, there is a link that says "Don't have an account? Sign Up". The Windows taskbar at the bottom shows the system clock as 00:21 on 14-04-2024, along with various application icons and a search bar.

127.0.0.1:8000/accounts/login/

Home Polls My Polls Logout

Login

Username

raguvignesh

Password

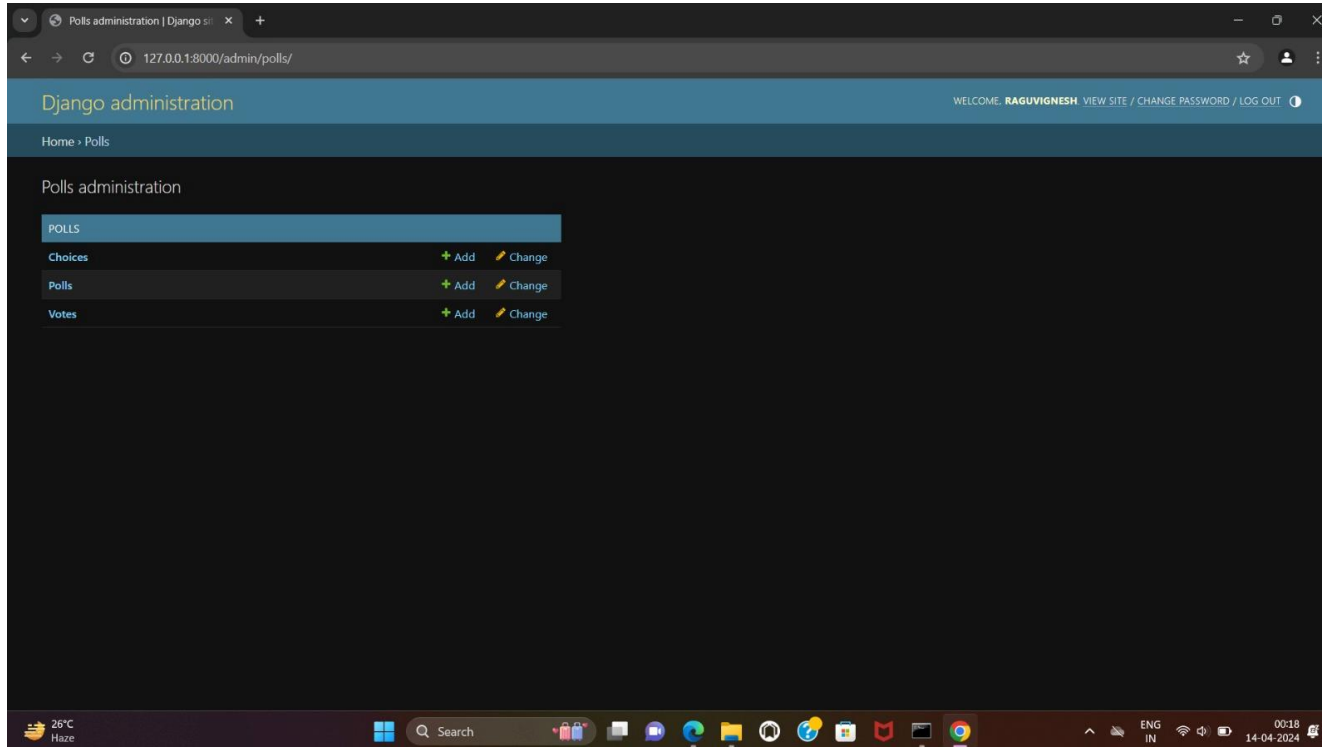
...

Login

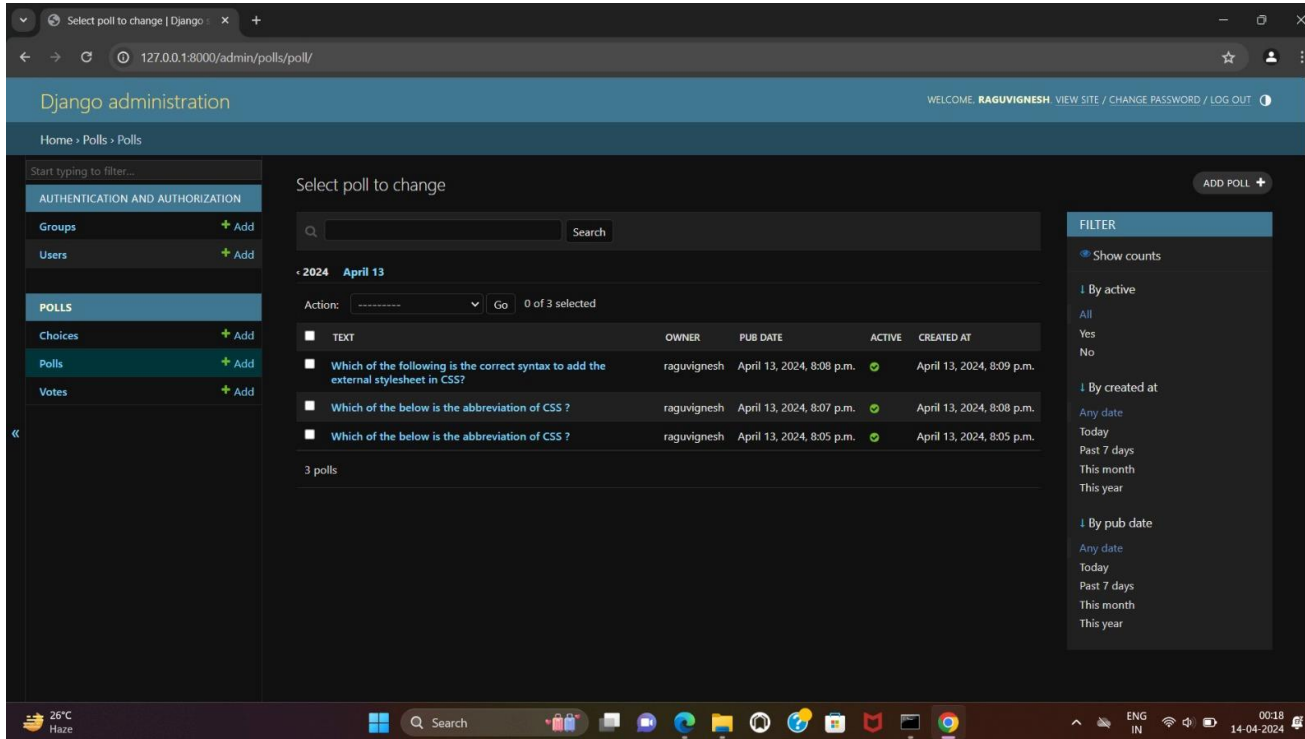
Don't have an account? [Sign Up](#)

26°C Haze Search ENG IN 00:21 14-04-2024

Admin Home Page



Authentication and Authorization Page



The screenshot displays the Django administration interface for a poll application. The browser address bar shows the URL `127.0.0.1:8000/admin/polls/poll/`. The page title is "Django administration". The user is logged in as "RAGUVIGNESH" and can view the site, change their password, or log out.

The left sidebar contains a navigation menu with the following sections:

- Start typing to filter...
- AUTHENTICATION AND AUTHORIZATION
 - Groups + Add
 - Users + Add
- POLLS
 - Choices + Add
 - Polls + Add
 - Votes + Add

The main content area is titled "Select poll to change". It features a search bar and a table of polls. The table has columns for TEXT, OWNER, PUB DATE, ACTIVE, and CREATED AT. There are 3 polls listed, all created on April 13, 2024.

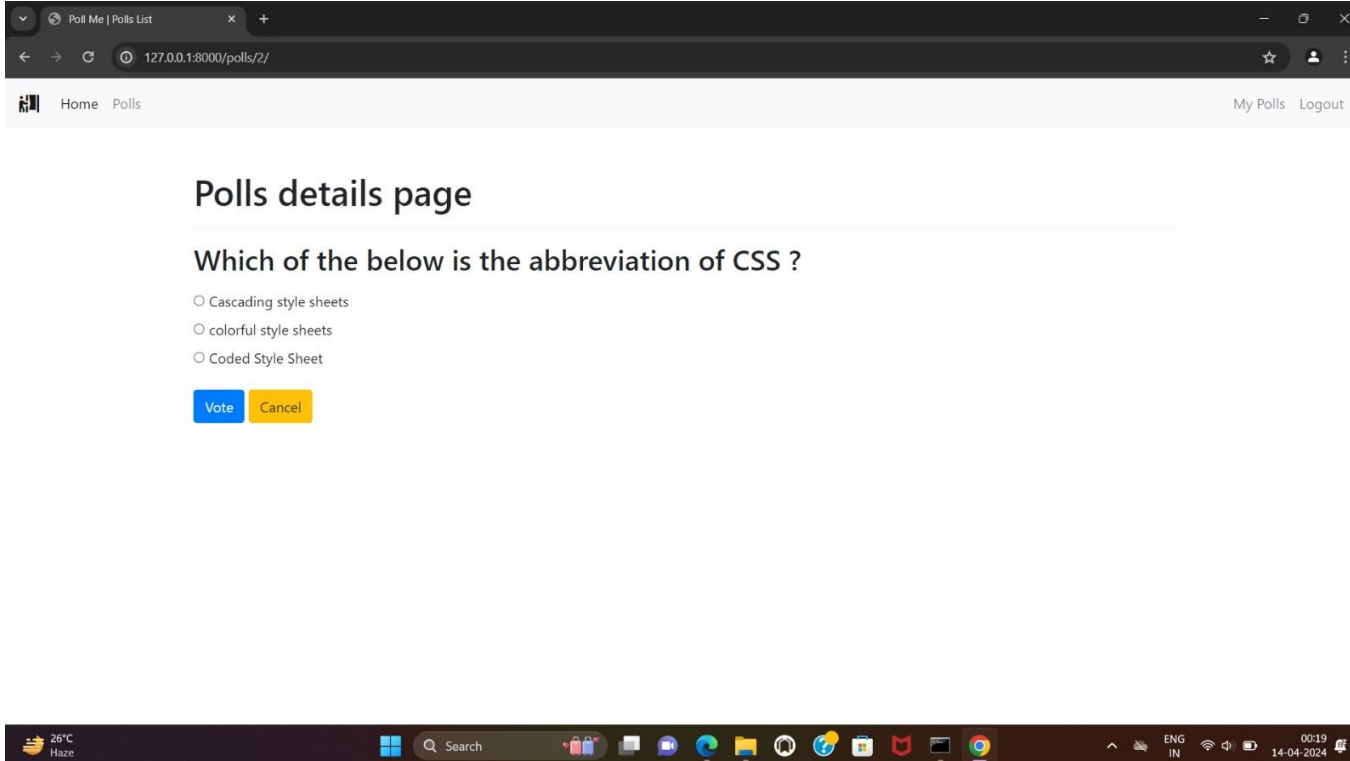
TEXT	OWNER	PUB DATE	ACTIVE	CREATED AT
Which of the following is the correct syntax to add the external stylesheet in CSS?	raguvignesh	April 13, 2024, 8:08 p.m.	Yes	April 13, 2024, 8:09 p.m.
Which of the below is the abbreviation of CSS ?	raguvignesh	April 13, 2024, 8:07 p.m.	Yes	April 13, 2024, 8:08 p.m.
Which of the below is the abbreviation of CSS ?	raguvignesh	April 13, 2024, 8:05 p.m.	Yes	April 13, 2024, 8:05 p.m.

Below the table, it indicates "3 polls".

The right sidebar contains a "FILTER" section with the following options:

- Show counts
- By active
 - All
 - Yes
 - No
- By created at
 - Any date
 - Today
 - Past 7 days
 - This month
 - This year
- By pub date
 - Any date
 - Today
 - Past 7 days
 - This month
 - This year

Questions Adding Section Page



Poll Me | Polls List

127.0.0.1:8000/polls/2/

Home Polls

My Polls Logout

Polls details page

Which of the below is the abbreviation of CSS ?

- ☐ Cascading style sheets
- ☐ colorful style sheets
- ☐ Coded Style Sheet

[Vote](#) [Cancel](#)

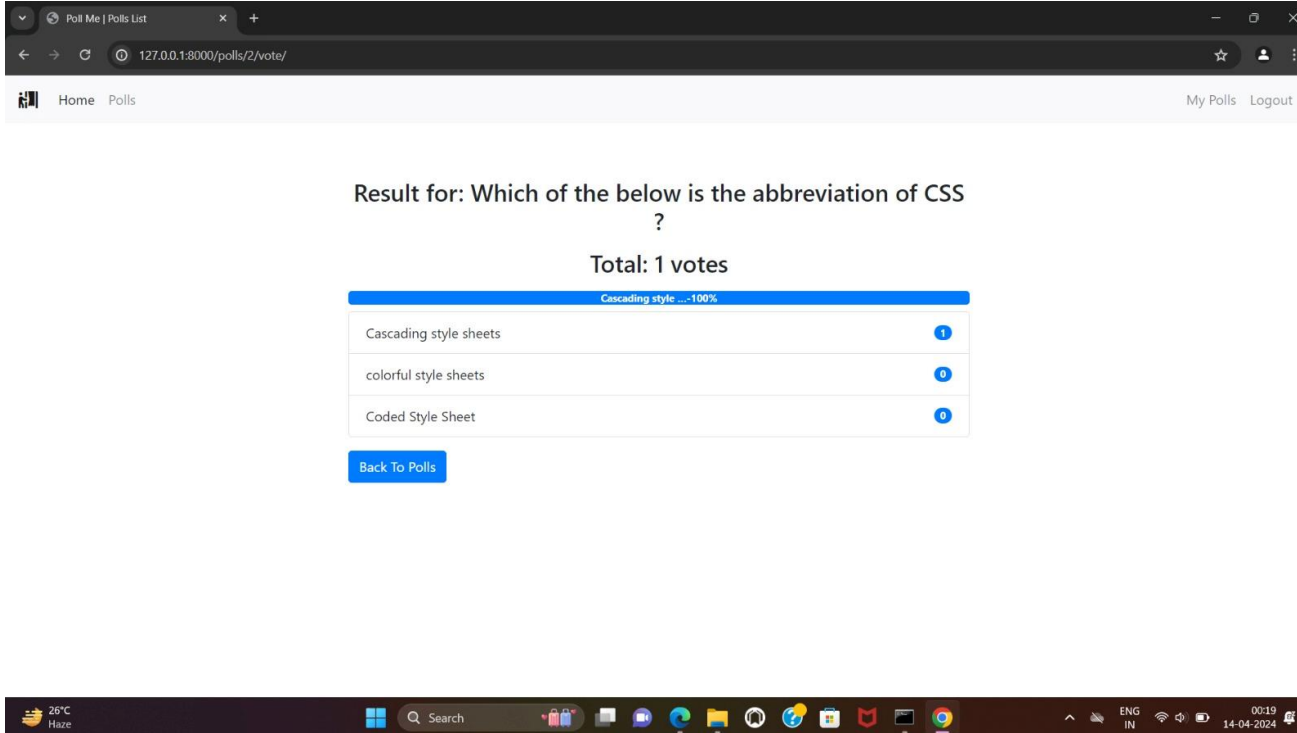
26°C Haze

Search

ENG IN

00:19 14-04-2024

Voting Details Page



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/polls/2/vote/". The page has a navigation bar with "Home" and "Polls" links, and "My Polls" and "Logout" links on the right. The main content area displays the poll results for the question "Which of the below is the abbreviation of CSS?". The results show "Total: 1 votes" and a table with three options: "Cascading style sheets" (1 vote), "colorful style sheets" (0 votes), and "Coded Style Sheet" (0 votes). A "Back To Polls" button is located at the bottom of the table.

Result for: Which of the below is the abbreviation of CSS ?

Total: 1 votes

Cascading style sheets ---100%	
Cascading style sheets	1
colorful style sheets	0
Coded Style Sheet	0

[Back To Polls](#)

28°C Haze 00:19 14-04-2024

Technology Used

Front-end



Back-end



Future Enhancements:

Future enhancements in a voting application using the Django framework, several key features and improvements can be considered based on the information from the provided sources,

1.Asynchronous Programming: Implementing asynchronous programming can enhance the performance of the application by allowing tasks to run concurrently, improving responsiveness and scalability.

2.Microservices Architecture: Adopting a microservices architecture can make the application more modular, easier to maintain, and scalable by breaking it into smaller, independent services that communicate with each other

3.Serverless Computing: Utilizing serverless computing can optimize resource utilization and reduce costs by enabling automatic scaling and only paying for actual usage, enhancing the application's efficiency and cost-effectiveness.

4.Client-Side Encryption: Enhancing security by implementing client-side encryption can protect sensitive data and ensure the confidentiality of votes, contributing to a more secure e-voting platform.

5.Blockchain Technology: Integrating blockchain technology can provide transparent and verifiable voting processes, ensuring the integrity of elections and promoting trust in the system

Conclusion

To create a voting application using Django, one should have a solid understanding of Python programming, Django framework, HTML, CSS, and Bootstrap. The development process involves creating a new Django project, creating a Django app, defining models, creating views, defining templates, and creating URLs. The application can be further enhanced with features such as real-time results, a user-friendly interface, and a secure database design. It can also include an admin panel for managing elections, candidates, and user accounts. Overall, a voting application using the Django framework is a powerful and flexible solution for creating online voting systems that can cater to various use cases and requirements.

Thank You!