



NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Student Name :Sermaraj.M
Student ID:autb21csl011

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Voting Application using Django Framework-Sermaraj(4313,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:

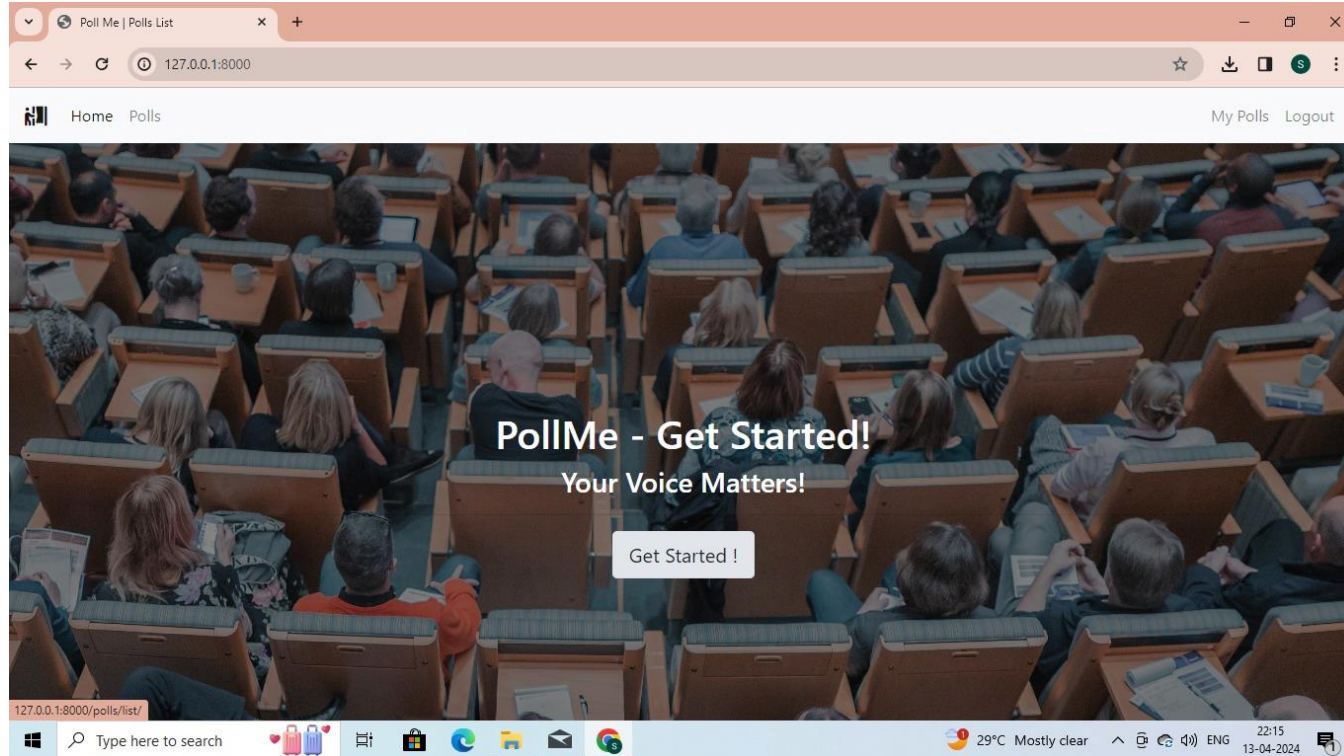
- 1.Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2.Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3.Creating User Authentication:** Implement user authentication 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

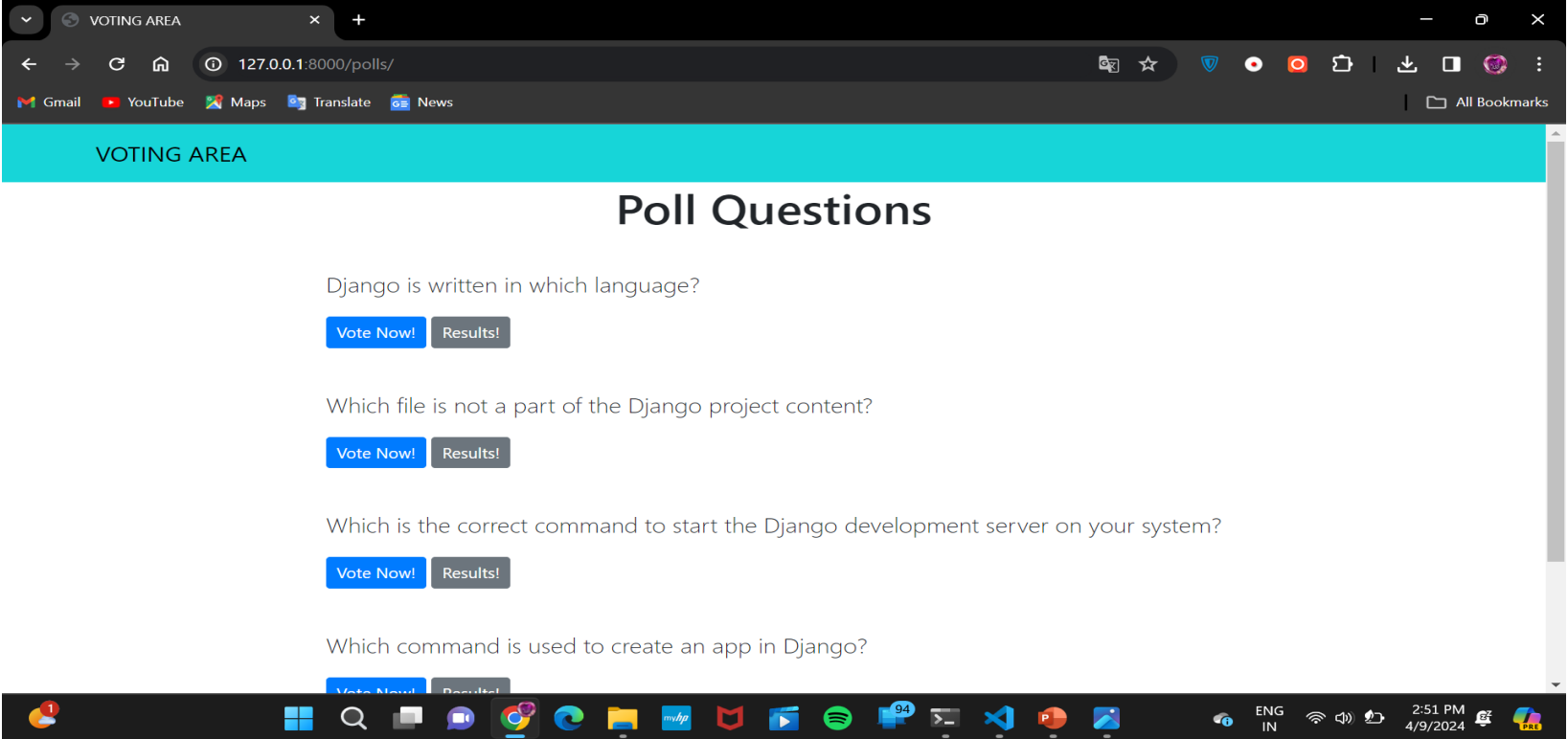
The proposed solution for a voting application using the Django framework is to create a secure and user-friendly online voting platform. The application will allow users to register, vote, and view real-time results. To build the application, the Django framework will be used as the foundation due to 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



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/polls/`. The page has a teal header with the text "VOTING AREA". Below the header, the main heading is "Poll Questions". There are four poll questions listed, each with a blue "Vote Now!" button and a grey "Results!" button.

VOTING AREA

Poll Questions

Django is written in which language?

[Vote Now!](#) [Results!](#)

Which file is not a part of the Django project content?

[Vote Now!](#) [Results!](#)

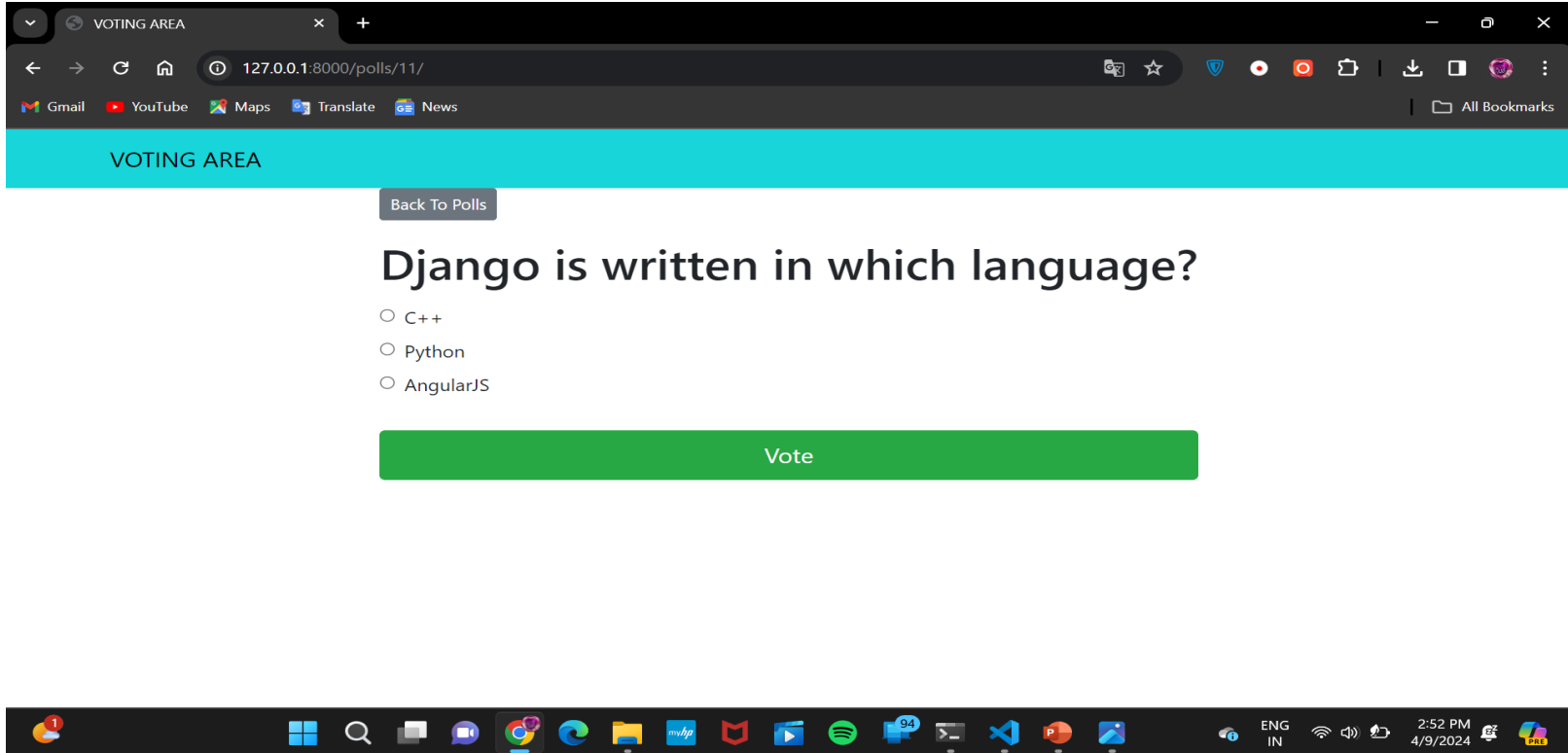
Which is the correct command to start the Django development server on your system?

[Vote Now!](#) [Results!](#)

Which command is used to create an app in Django?

[Vote Now!](#) [Results!](#)

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 a cyan background and the text "VOTING AREA". A button labeled "Back To Polls" is positioned above the main content area. The main content area features a large heading "Django is written in which language?" followed by three radio button options: "C++", "Python", and "AngularJS". A large green button labeled "Vote" is centered below the options. The browser's taskbar at the bottom shows various application icons, including the Start menu, search, and several open applications like Chrome, Edge, and File Explorer. The system tray on the right indicates the language is "ENG IN", the time is "2:52 PM", and the date is "4/9/2024".

VOTING AREA

Back To Polls

Django is written in which language?

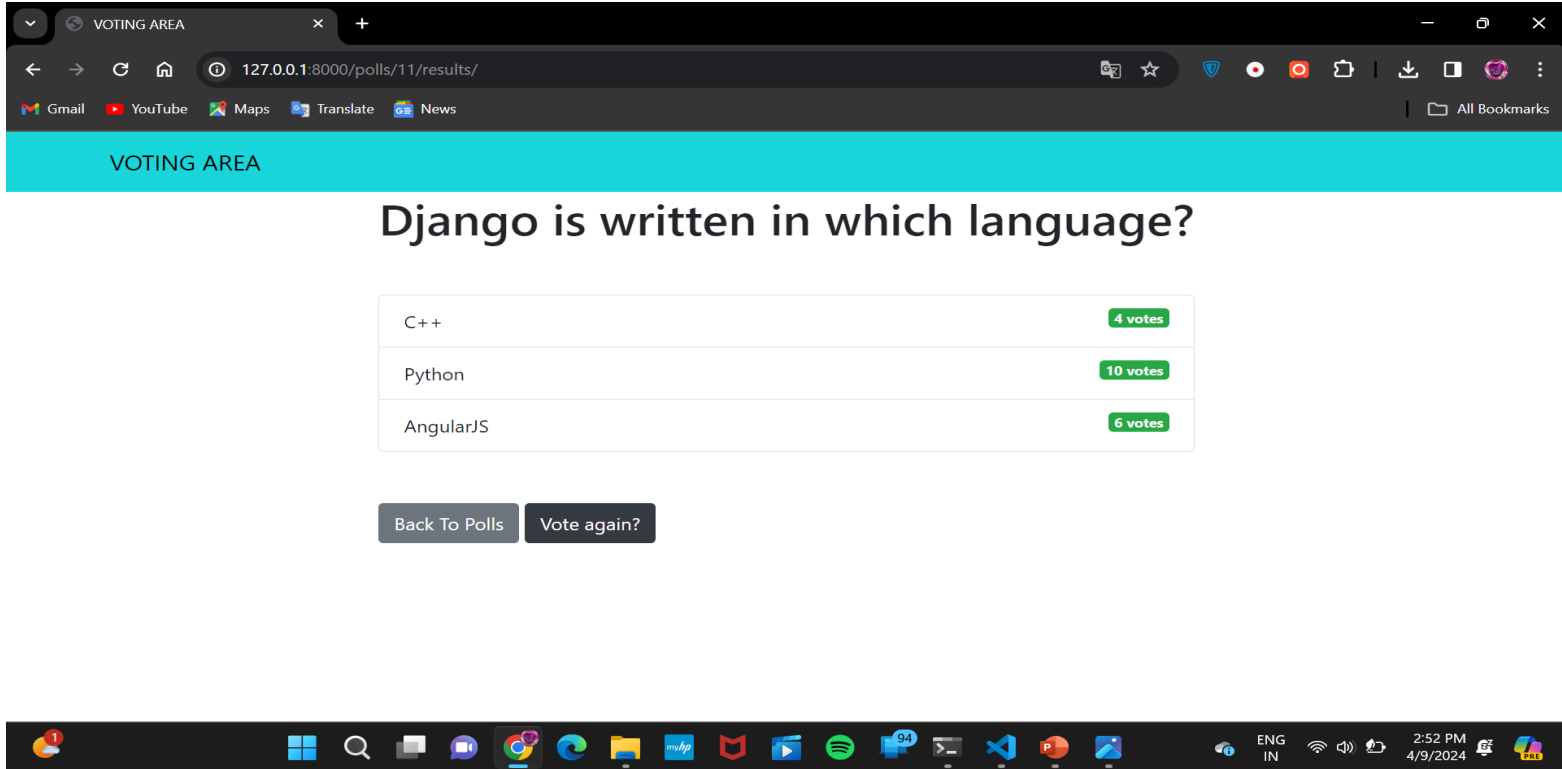
☐ C++

☐ Python

☐ AngularJS

Vote

Voting Details Page

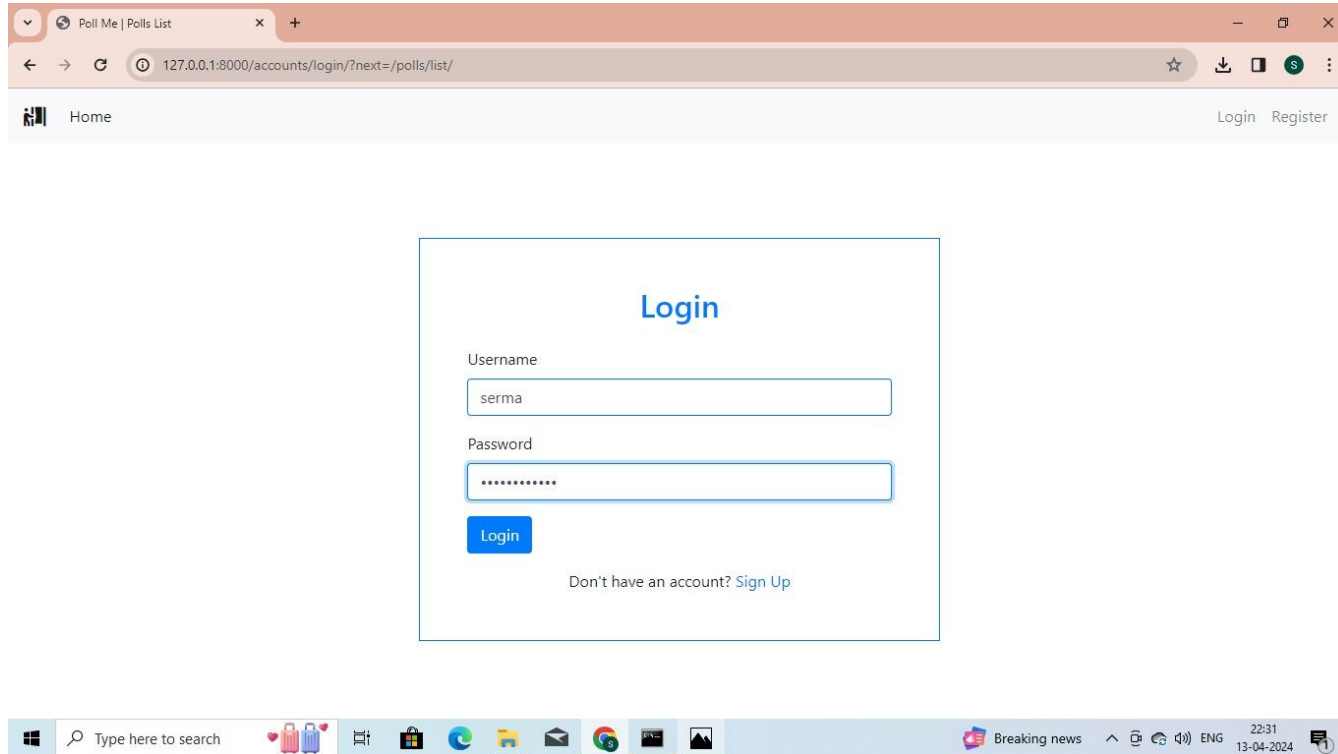


The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/polls/11/results/". The page has a teal header with the text "VOTING AREA". Below the header, the question "Django is written in which language?" is displayed. A table lists three options with their respective vote counts:

C++	4 votes
Python	10 votes
AngularJS	6 votes

At the bottom of the page, there are two buttons: "Back To Polls" and "Vote again?". The Windows taskbar at the bottom shows the time as 2:52 PM on 4/9/2024.

Admin Login Page



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/accounts/login/?next=/polls/list/`. The page title is "Poll Me | Polls List". The browser's address bar includes navigation buttons (back, forward, refresh) and icons for bookmarks, downloads, and extensions. The page content features a "Home" link and "Login" and "Register" links. The main login form is titled "Login" and contains two input fields: "Username" with the value "serma" and "Password" with masked characters. A blue "Login" button is positioned below the password field. At the bottom of the form, there is a link: "Don't have an account? [Sign Up](#)". The Windows taskbar at the bottom shows the search bar, task view button, and several application icons (Store, Edge, File Explorer, Mail, Chrome, etc.). The system tray on the right indicates the time as 22:31 on 13-04-2024, along with network and volume icons.

Home Login Register

Login

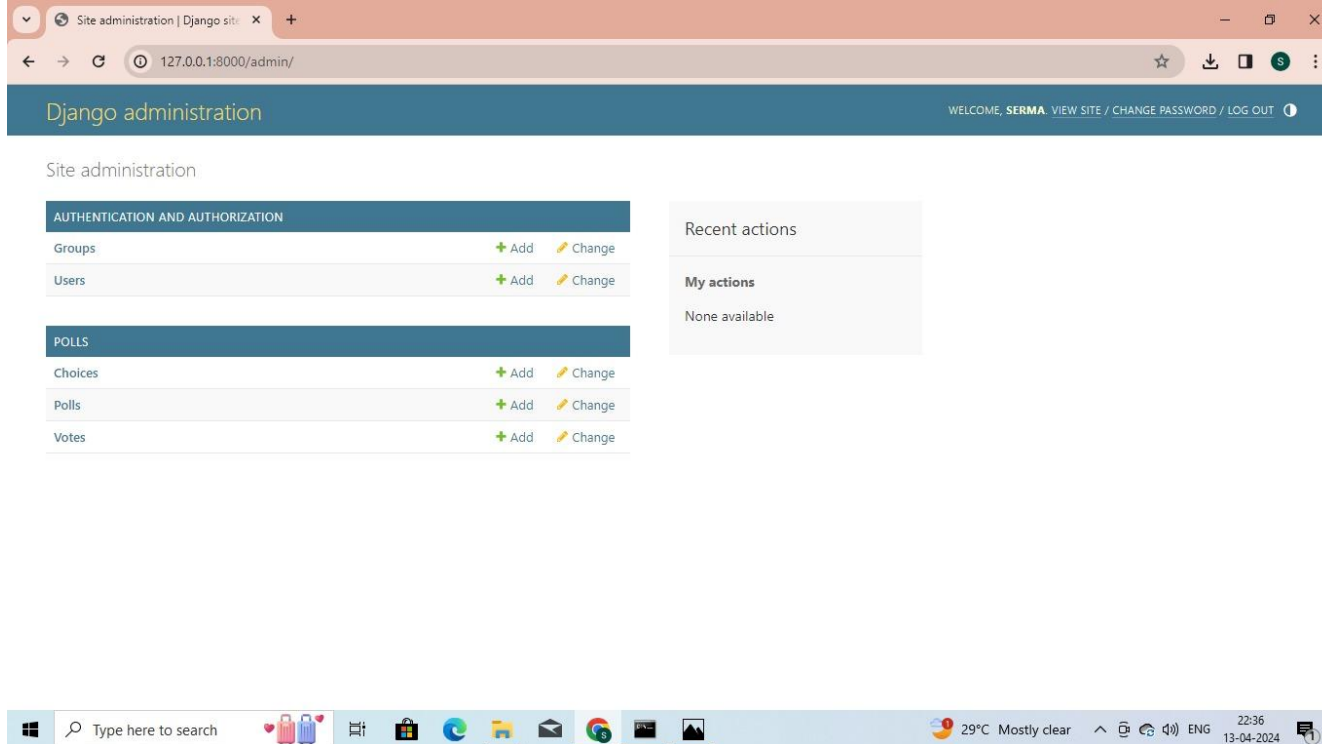
Username

Password

Login

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

Admin Home Page



The screenshot displays the Django administration interface in a web browser. The browser's address bar shows the URL `127.0.0.1:8000/admin/`. The page header includes the text "Django administration" and a welcome message for "SERMA" with links to "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT".

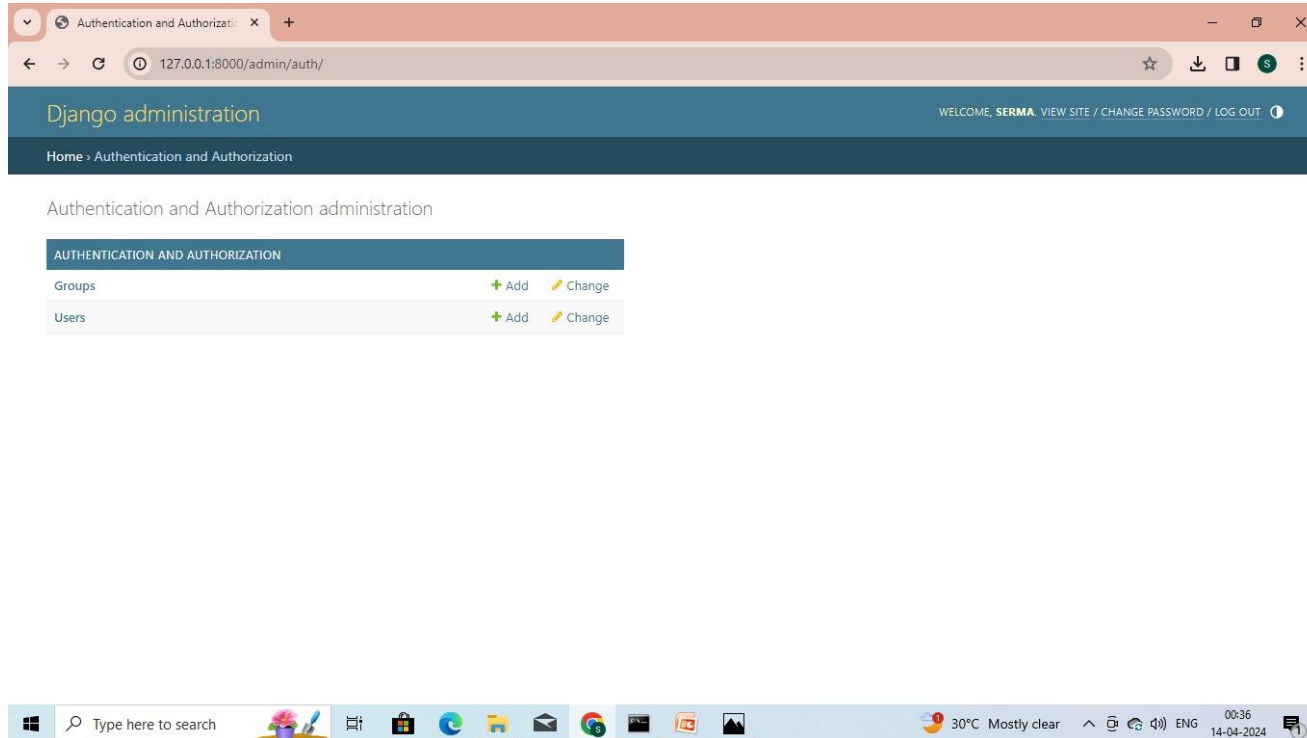
The main content area is titled "Site administration" and is divided into two sections:

- AUTHENTICATION AND AUTHORIZATION**
 - Groups**: Includes links for "+ Add" and "Change".
 - Users**: Includes links for "+ Add" and "Change".
- POLLS**
 - Choices**: Includes links for "+ Add" and "Change".
 - Polls**: Includes links for "+ Add" and "Change".
 - Votes**: Includes links for "+ Add" and "Change".

On the right side of the page, there is a "Recent actions" section and a "My actions" section, both of which currently display "None available".

The Windows taskbar at the bottom shows the search bar, several application icons, and system information including the temperature (29°C), weather (Mostly clear), and the date and time (22:36, 13-04-2024).

Authentication and Authorization Page



The screenshot shows a web browser window displaying the Django administration interface. The browser's address bar shows the URL `127.0.0.1:8000/admin/auth/`. The page title is "Django administration" and the user is logged in as "SERMA". The breadcrumb trail is "Home > Authentication and Authorization". The main heading is "Authentication and Authorization administration". Below this, there is a table with two rows: "Groups" and "Users". Each row has a green plus icon and the text "Add" followed by a yellow pencil icon and the text "Change".

AUTHENTICATION AND AUTHORIZATION	
Groups	+ Add Change
Users	+ Add Change

Questions Adding Section Page

Select vote to change | Django

127.0.0.1:8000/admin/polls/vote/

Django administration

WELCOME, SERMA. VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Polls > Votes

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups Add

Users Add

POLLS

Choices Add

Polls Add

Votes Add

Select vote to change

ADD VOTE

Q

Search

Action:

 Go 0 of 4 selected

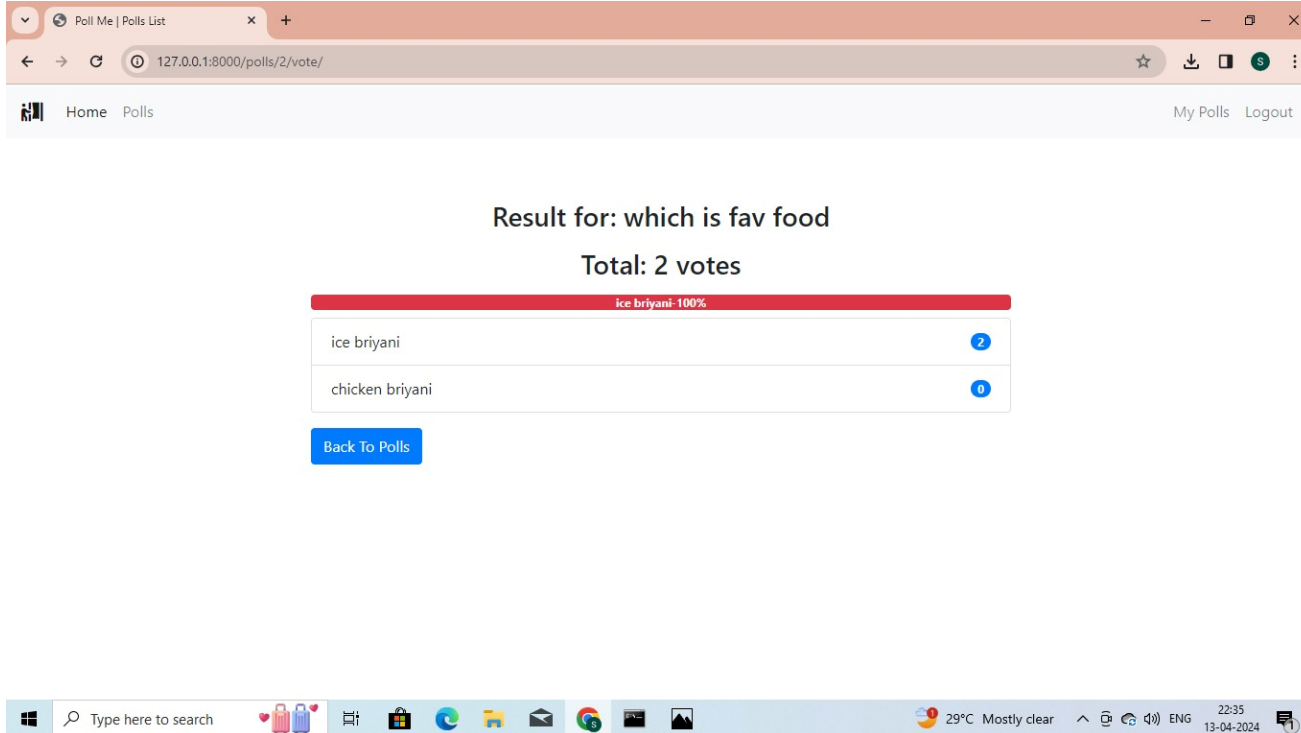
<input type="checkbox"/>	CHOICE	POLL	USER	CREATED AT
<input type="checkbox"/>	which is fav food - ice briyani	which is fav food	serma	April 13, 2024, 11:05 p.m.
<input type="checkbox"/>	which is fav actor - vijay	which is fav actor	sermaraj	April 13, 2024, 10:35 p.m.
<input type="checkbox"/>	which is fav food - ice briyani	which is fav food	sermaraj	April 13, 2024, 10:35 p.m.
<input type="checkbox"/>	what is best subject - english	what is best subject	sermaraj	April 13, 2024, 10:34 p.m.

4 votes

Windows Taskbar

29°C Mostly clear 22:38 13-04-2024

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 title is "Poll Me | Polls List". The navigation bar includes "Home" and "Polls" links, and "My Polls" and "Logout" buttons. The main content area displays the result for a poll titled "which is fav food". The total number of votes is 2. The results are as follows:

ice briyani- 100%	
ice briyani	2
chicken briyani	0

Below the table is a blue button labeled "Back To Polls". The Windows taskbar at the bottom shows the search bar, task view, and various application icons. The system tray displays the date and time as 22:35 on 13-04-2024, along with weather information (29°C Mostly clear) and system icons.

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!