



edunet
foundation



NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Student Name :Vengadeshwaran v
Student ID :au820621104088

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Voting Application using Django Framework-VENGADESHWARAN V(4088,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:

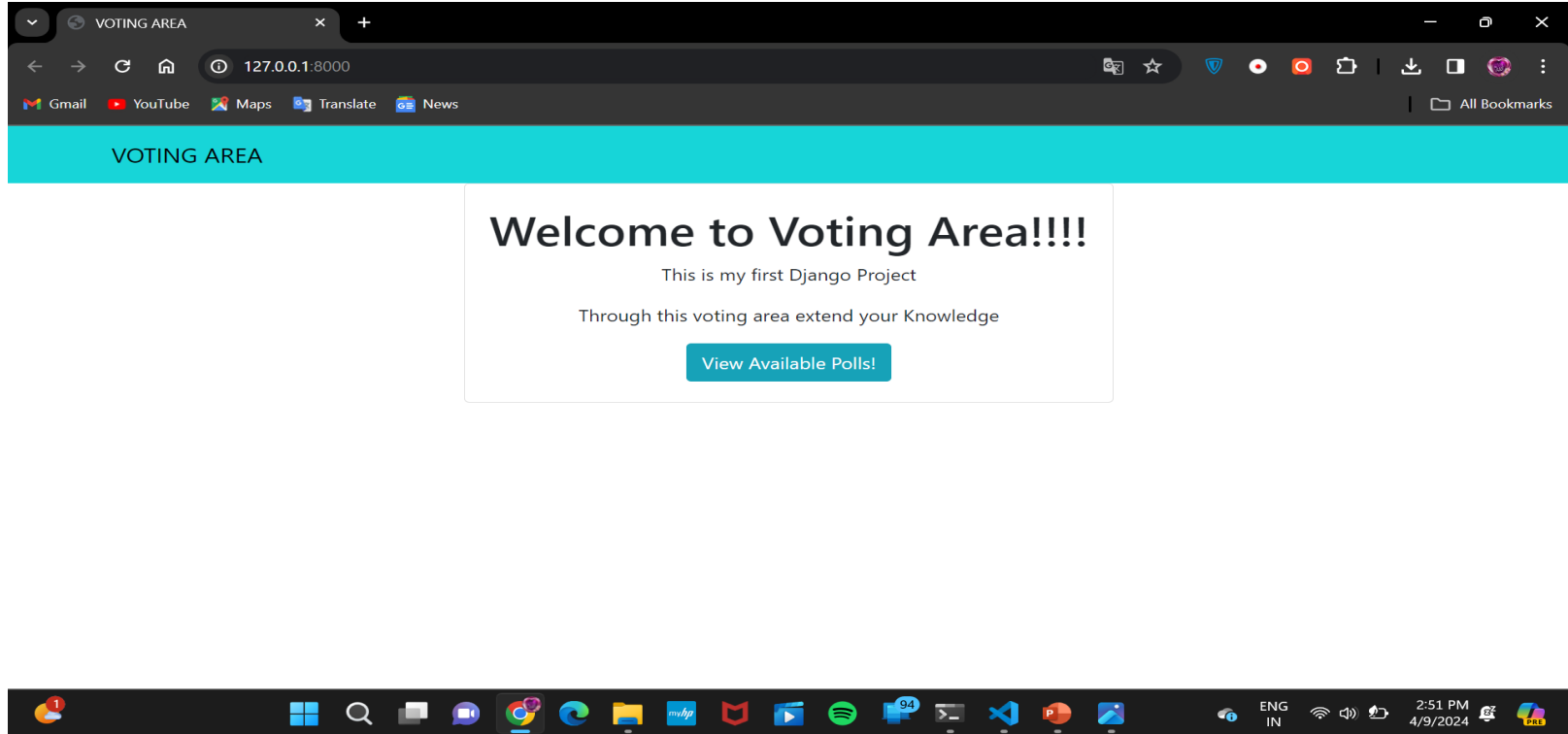
- 1.Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2.Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3.Creating User Authentication:** Implement user authentication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

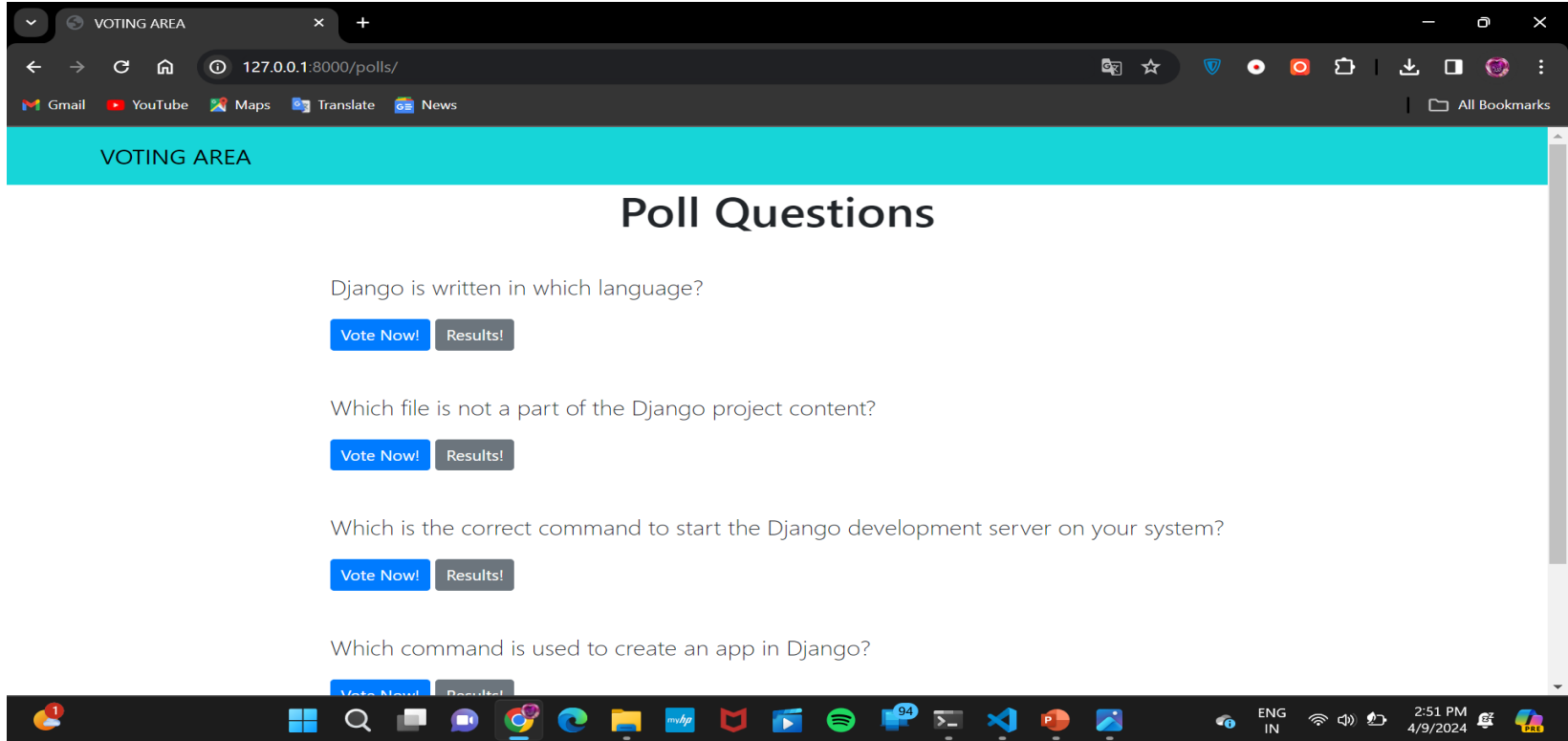
The proposed solution for a voting application using the Django framework is to create a secure and user-friendly online voting platform. The application will allow users to register, vote, and view real-time results. To build the application, the Django framework will be used as the foundation due to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

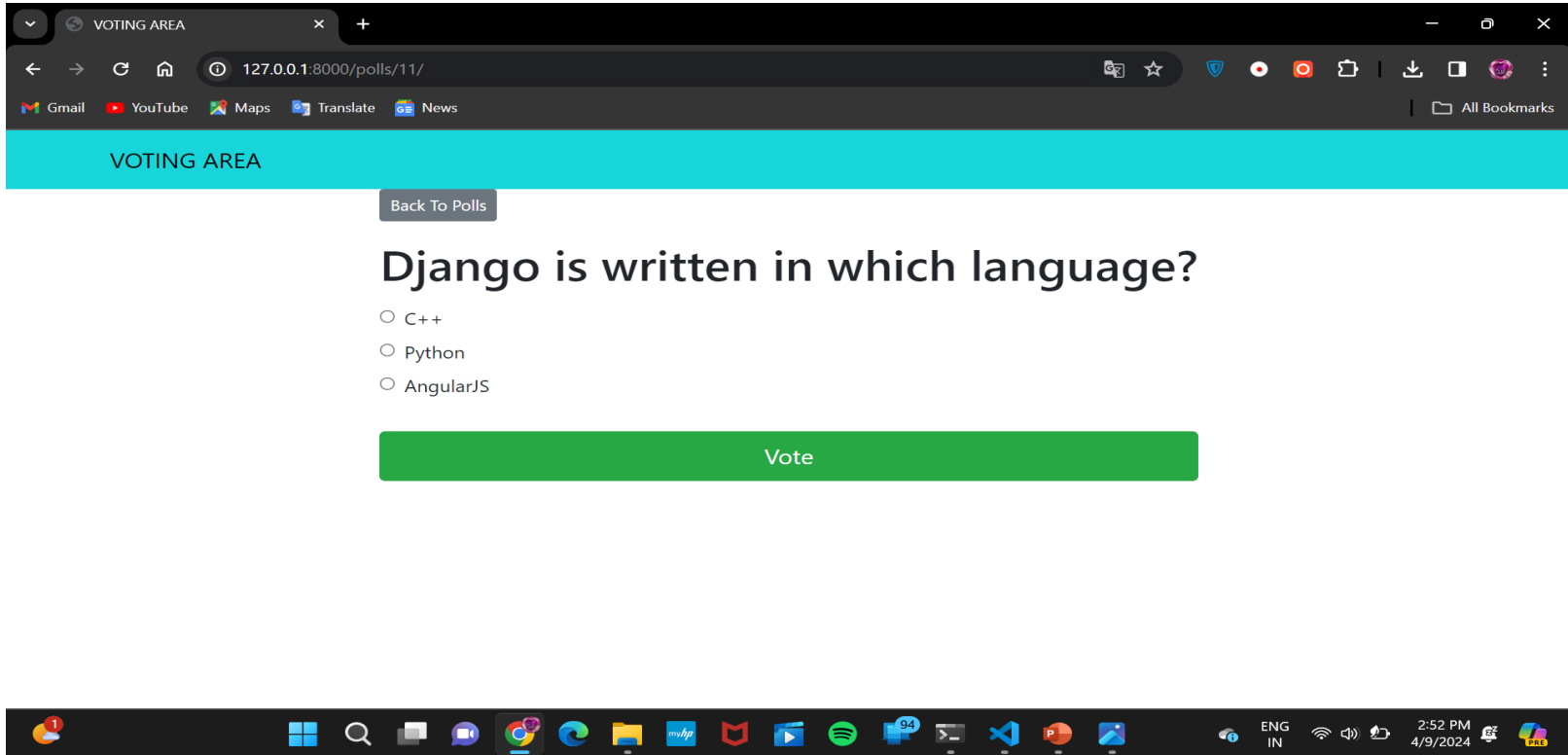
Which is the correct command to start the Django development server on your system?

[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, there 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, there is a button labeled "Back To Polls". The main question displayed is "Django is written in which language?". Below the question, there are three radio button options: "C++", "Python", and "AngularJS". At the bottom of the page, there is a large green button labeled "Vote". The browser's taskbar is visible at the bottom, 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?

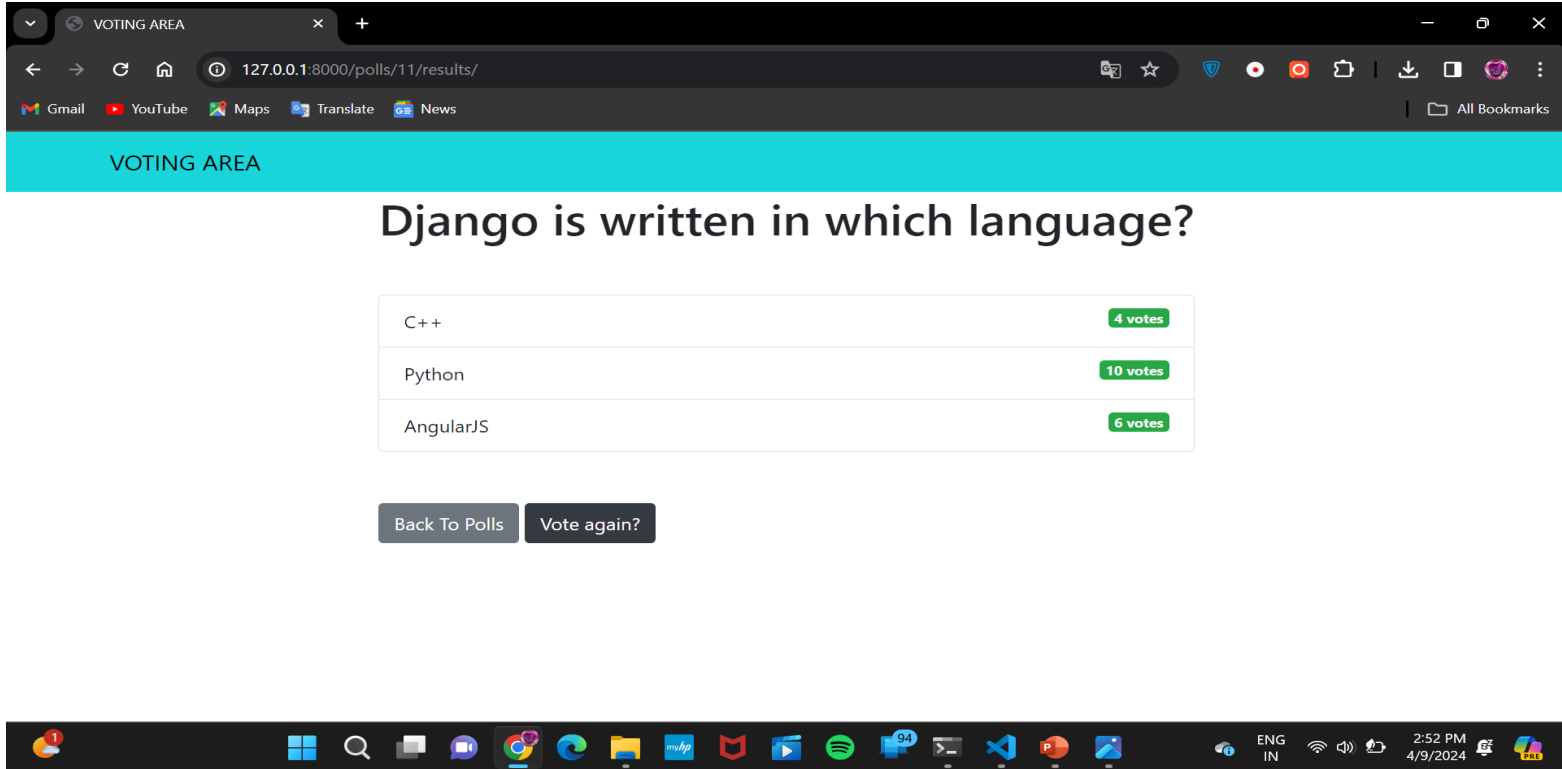
☐ C++

☐ Python

☐ AngularJS

Vote

Voting Details Page



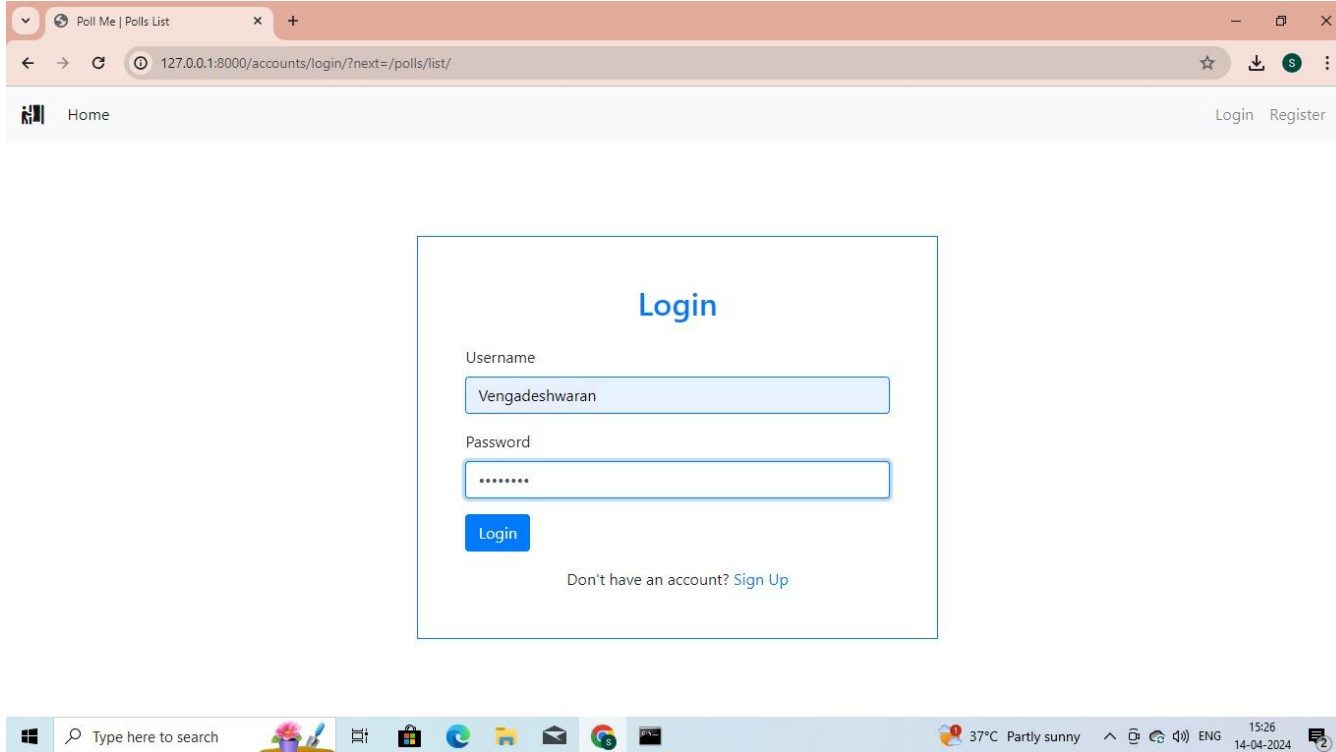
VOTING AREA

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/?next=/polls/list/`. The page has a header with a "Home" link and "Login" and "Register" buttons. The main content area is a login form titled "Login" with fields for "Username" (containing "Vengadeshwaran") and "Password" (masked with "*****"). A blue "Login" button is 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, and several application icons. The system tray on the right indicates a temperature of 37°C, "Partly sunny" weather, and the date/time "15:26 14-04-2024".

Home Login Register

Login

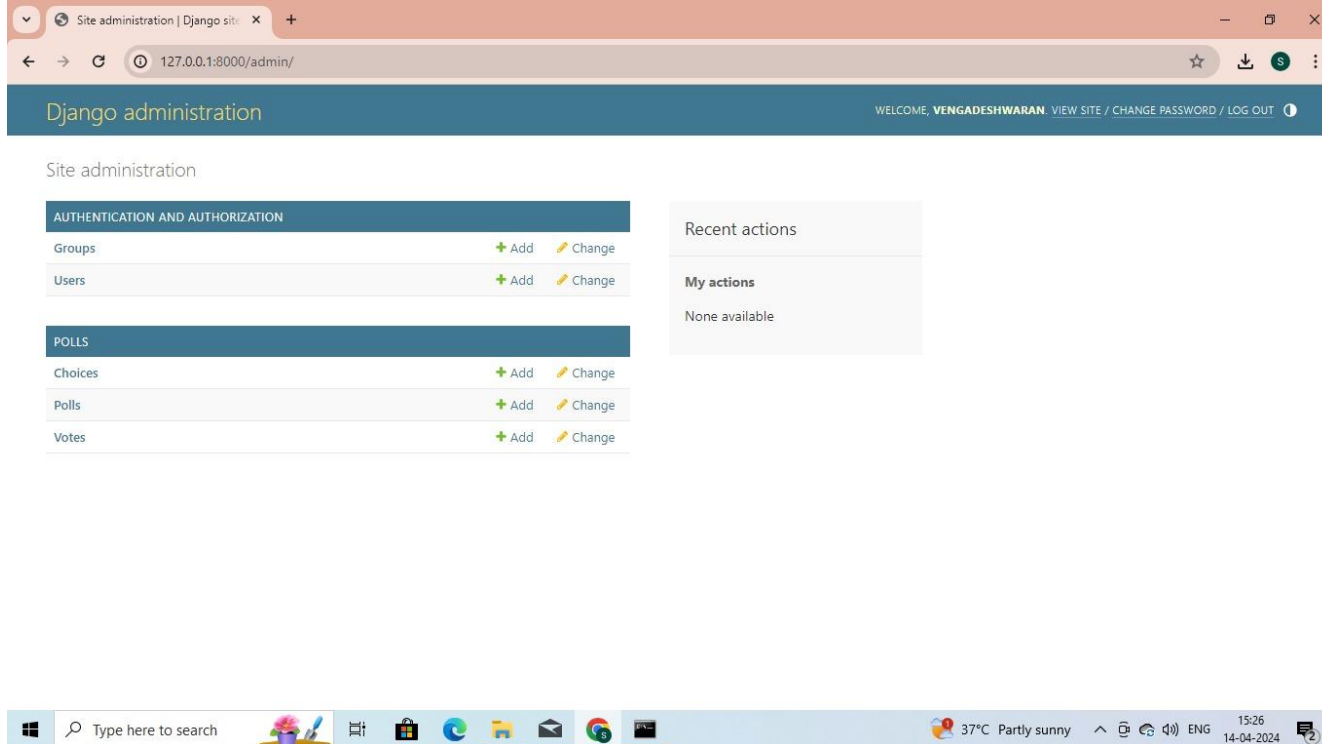
Username

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 "VENGADESHWARAN" with links for "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT".

The main content area is titled "Site administration" and is divided into two columns. The left column contains two sections: "AUTHENTICATION AND AUTHORIZATION" and "POLLS".

Authentication and Authorization:

Item	Actions
Groups	+ Add Change
Users	+ Add Change

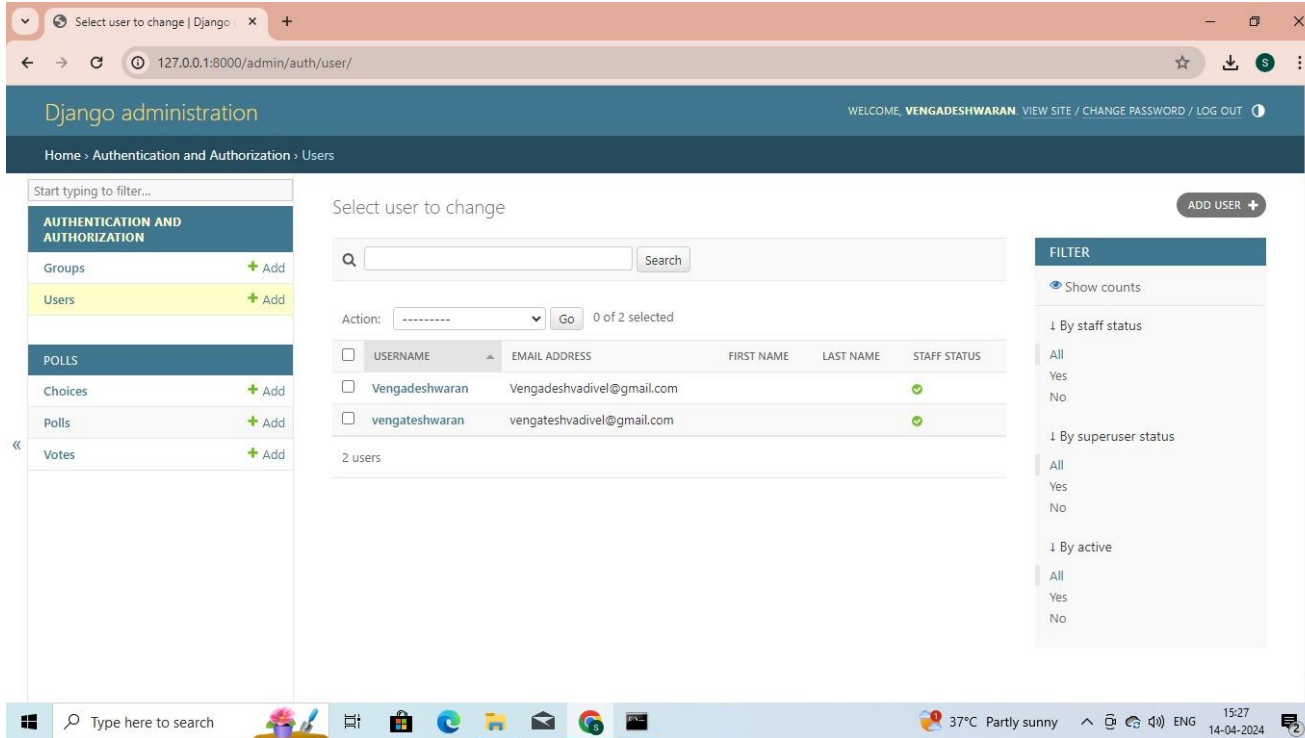
Polls:

Item	Actions
Choices	+ Add Change
Polls	+ Add Change
Votes	+ Add Change

The right column contains a "Recent actions" section, which is currently empty, and a "My actions" section, which also shows "None available".

The Windows taskbar at the bottom of the screen shows the search bar, task view icon, and several application icons. The system tray on the right indicates a temperature of 37°C, "Partly sunny" weather, and the date and time as 15:26 on 14-04-2024.

Authentication and Authorization Page



The screenshot displays the Django administration interface for the Authentication and Authorization section. The browser address bar shows the URL `127.0.0.1:8000/admin/auth/user/`. The page header includes the Django administration logo and a welcome message for the user **VENGADESHWARAN**.

The sidebar on the left contains the following navigation links:

- 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 user to change". It features a search bar and a table of users. The table has columns for USERNAME, EMAIL ADDRESS, FIRST NAME, LAST NAME, and STAFF STATUS. Two users are listed:

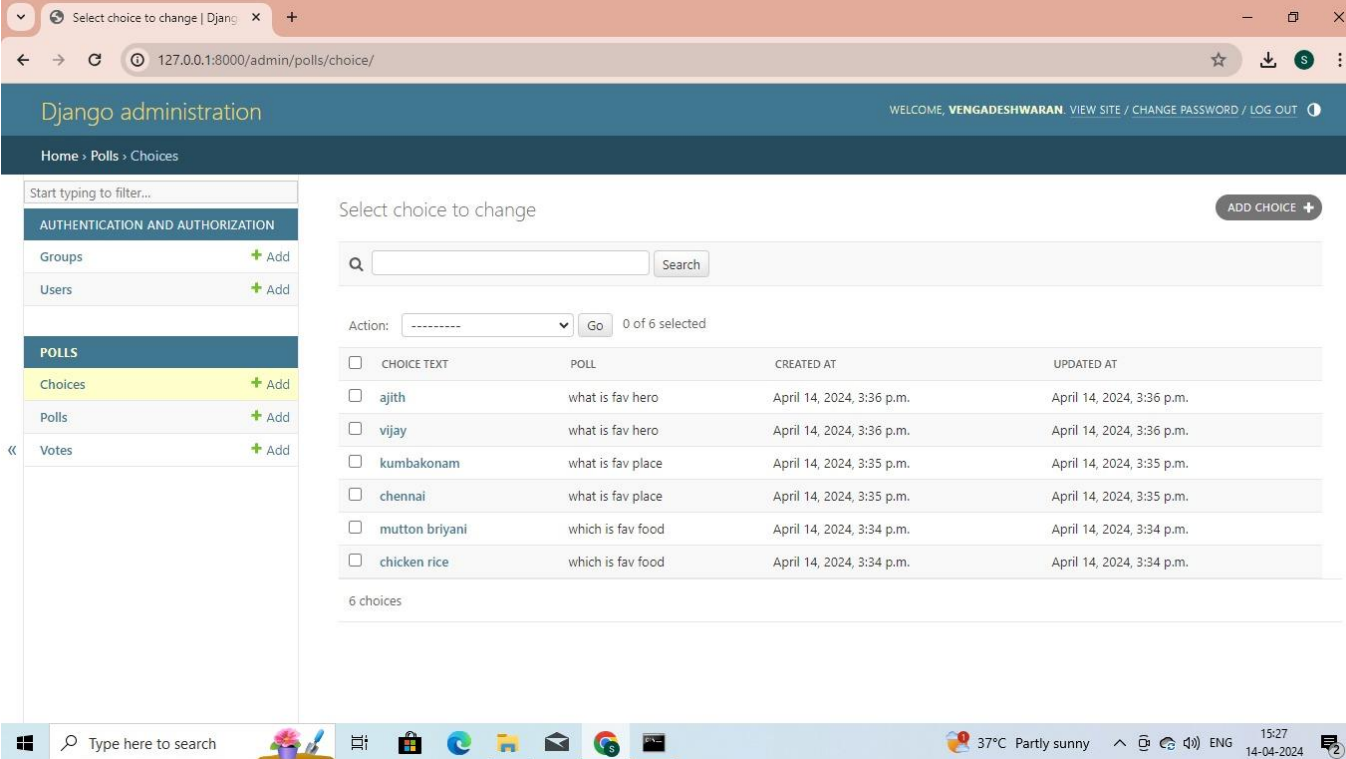
<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	Vengadeshwaran	Vengadeshvadel@gmail.com			✓
<input type="checkbox"/>	vengateshwaran	vengateshvadel@gmail.com			✓

Below the table, it indicates "2 users".

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

- Show counts
- By staff status
 - All
 - Yes
 - No
- By superuser status
 - All
 - Yes
 - No
- By active
 - All
 - Yes
 - No

Questions Adding Section Page



The screenshot shows the Django administration interface for managing poll choices. The browser address bar indicates the URL is `127.0.0.1:8000/admin/polls/choice/`. The page title is "Django administration" and the user is logged in as "VENGADESHWARAN". The breadcrumb trail shows "Home > Polls > Choices".

On the left sidebar, the "POLL" section is expanded, and "Choices" is highlighted. The main content area is titled "Select choice to change" and includes an "ADD CHOICE +" button. A search bar is present above a table of existing choices.

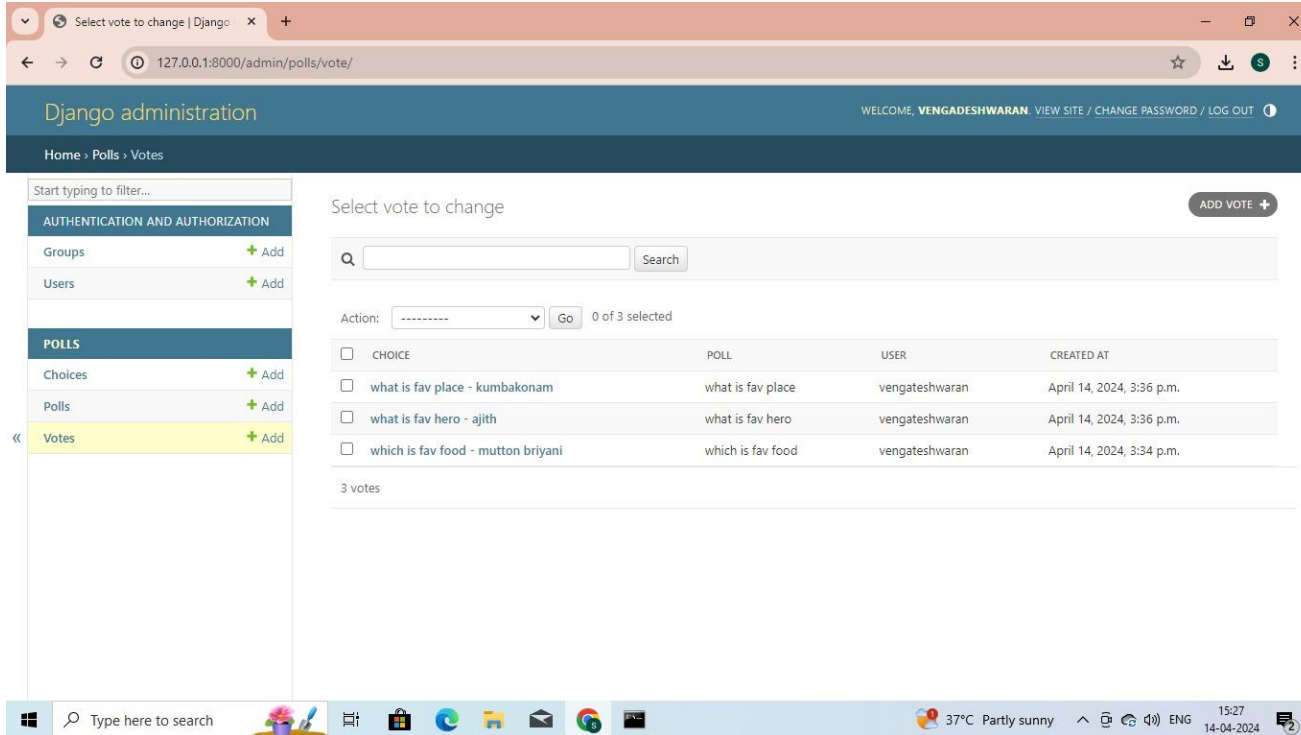
The table displays the following data:

CHOICE TEXT	POLL	CREATED AT	UPDATED AT
<input type="checkbox"/> ajith	what is fav hero	April 14, 2024, 3:36 p.m.	April 14, 2024, 3:36 p.m.
<input type="checkbox"/> vijay	what is fav hero	April 14, 2024, 3:36 p.m.	April 14, 2024, 3:36 p.m.
<input type="checkbox"/> kumbakonam	what is fav place	April 14, 2024, 3:35 p.m.	April 14, 2024, 3:35 p.m.
<input type="checkbox"/> chennai	what is fav place	April 14, 2024, 3:35 p.m.	April 14, 2024, 3:35 p.m.
<input type="checkbox"/> mutton briyani	which is fav food	April 14, 2024, 3:34 p.m.	April 14, 2024, 3:34 p.m.
<input type="checkbox"/> chicken rice	which is fav food	April 14, 2024, 3:34 p.m.	April 14, 2024, 3:34 p.m.

Below the table, it indicates "6 choices".

The Windows taskbar at the bottom shows the system clock as 15:27 on 14-04-2024, with a weather widget indicating 37°C and "Partly sunny".

Voting Details Page



The screenshot displays the Django administration interface for a voting system. The browser address bar shows the URL `127.0.0.1:8000/admin/polls/vote/`. The page title is "Django administration" with a welcome message for "VENGADESHWARAN" and links for "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT". The breadcrumb trail is "Home > Polls > Votes".

Left Sidebar:

- Start typing to filter...
- AUTHENTICATION AND AUTHORIZATION**
 - Groups [+ Add](#)
 - Users [+ Add](#)
- POLLS**
 - Choices [+ Add](#)
 - Polls [+ Add](#)
 - Votes** [+ Add](#)

Main Content Area:

Select vote to change [ADD VOTE +](#)

Search: [Search](#)

Action: ----- [Go](#) 0 of 3 selected

<input type="checkbox"/>	CHOICE	POLL	USER	CREATED AT
<input type="checkbox"/>	what is fav place - kumbakonam	what is fav place	vengateshwaran	April 14, 2024, 3:36 p.m.
<input type="checkbox"/>	what is fav hero - ajith	what is fav hero	vengateshwaran	April 14, 2024, 3:36 p.m.
<input type="checkbox"/>	which is fav food - mutton briyani	which is fav food	vengateshwaran	April 14, 2024, 3:34 p.m.

3 votes

Taskbar: Windows Start button, search bar (Type here to search), task view icon, pinned apps (File Explorer, Mail, Chrome, VS Code), system tray (37°C Partly sunny, 15:27, 14-04-2024, ENG, 2 notifications).

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!