



# NEXT GEN EMPLOYABILITY PROGRAM

| Creating a future-ready workforce

Student Name :Pasupathi R  
Student ID :au820621104501

College Name

Arasu Engineering College

# CAPSTONE PROJECT SHOWCASE

## Project Title

Voting Application using Django Framework-Pasupathi R(4501,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:

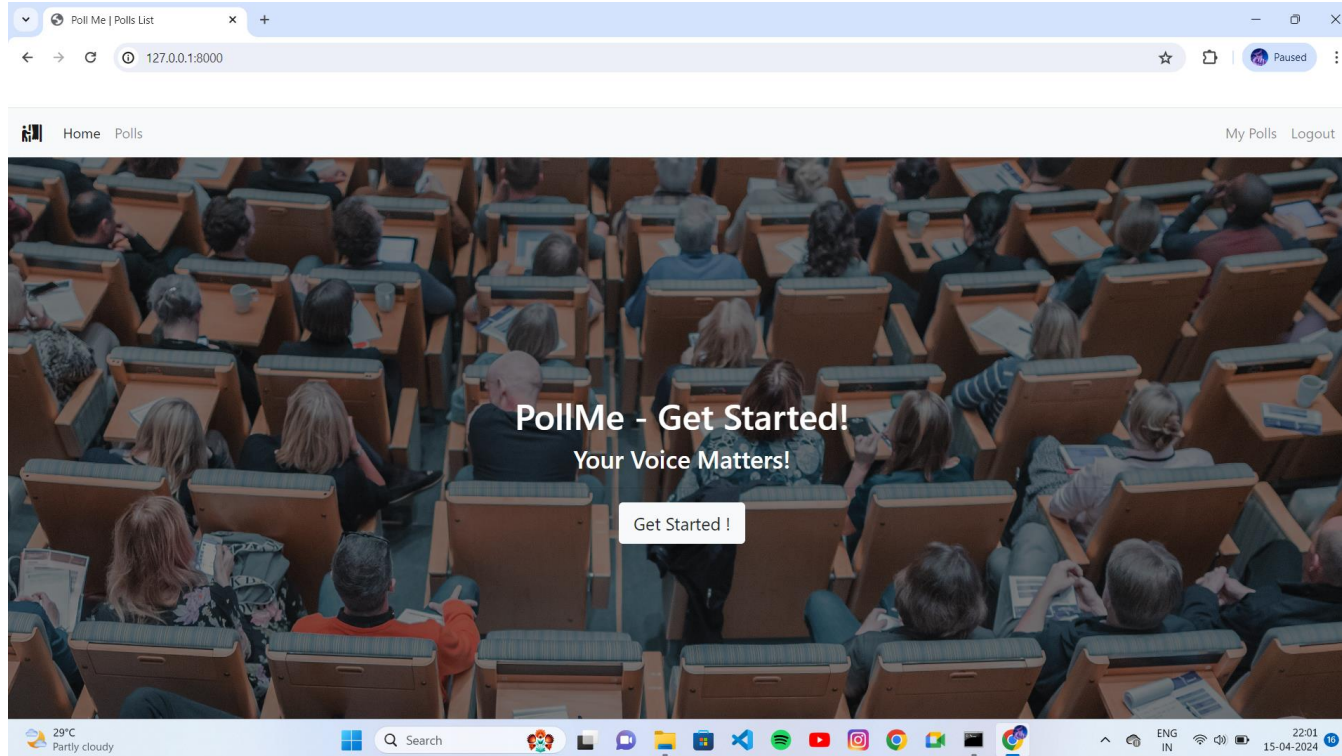
- 1.Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2.Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3.Creating User Authentication:** Implement user authentication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

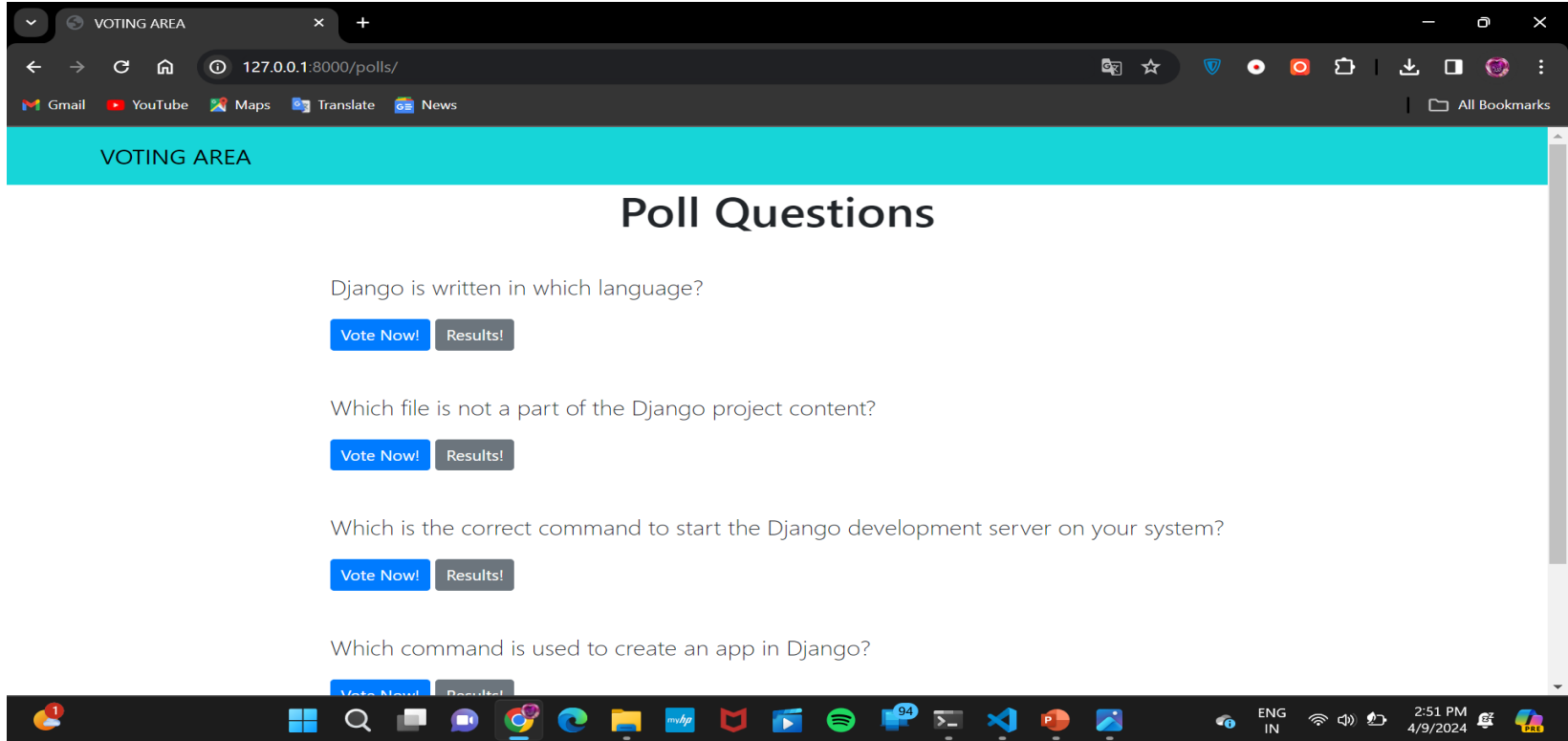


The screenshot shows a web browser window displaying the PollMe website. The browser's address bar shows the URL `127.0.0.1:8000`. The website has a navigation bar with "Home" and "Polls" links, and a user profile icon labeled "Paused". The main content area features a large background image of an audience in a lecture hall. Overlaid on this image is the text "PollMe - Get Started! Your Voice Matters!" and a "Get Started !" button. The bottom of the browser window shows a Windows taskbar with various application icons, the system clock displaying 22:01 on 15-04-2024, and weather information for 29°C Partly cloudy.

PollMe - Get Started!  
Your Voice Matters!

Get Started !

## 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?

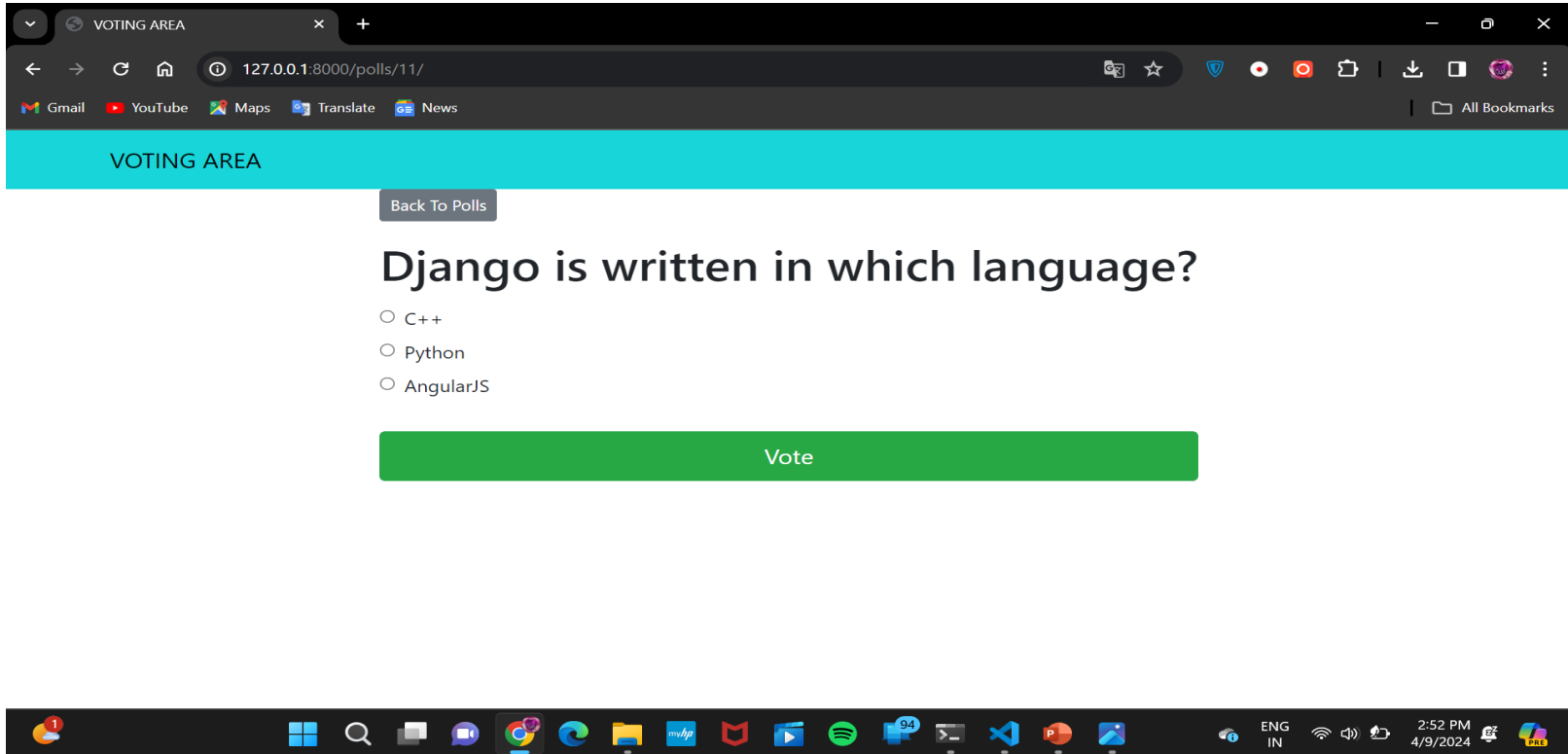
[Vote Now!](#) [Results!](#)

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 browser window, a cyan banner reads "VOTING AREA". A grey button labeled "Back To Polls" is positioned above the poll question. The question is "Django is written in which language?". Below the question are three radio button options: "C++", "Python", and "AngularJS". A large green button labeled "Vote" is centered below the options. The Windows taskbar at the bottom shows various application icons, including Chrome, Edge, File Explorer, Mail, and a system tray with the date "4/9/2024" and time "2:52 PM".

VOTING AREA

[Back To Polls](#)

Django is written in which language?

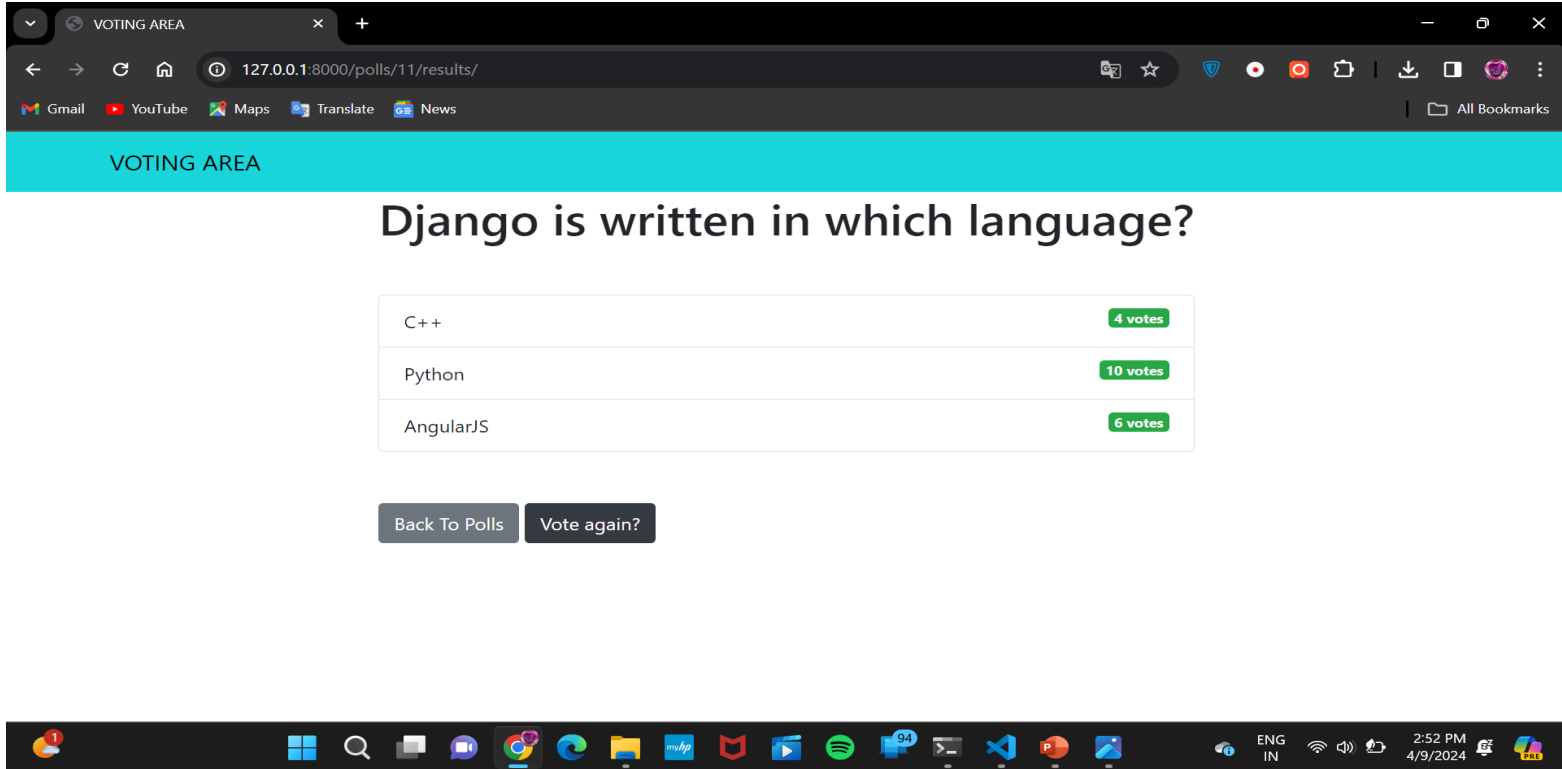
☐ C++

☐ Python

☐ AngularJS

[Vote](#)

## Voting Details Page

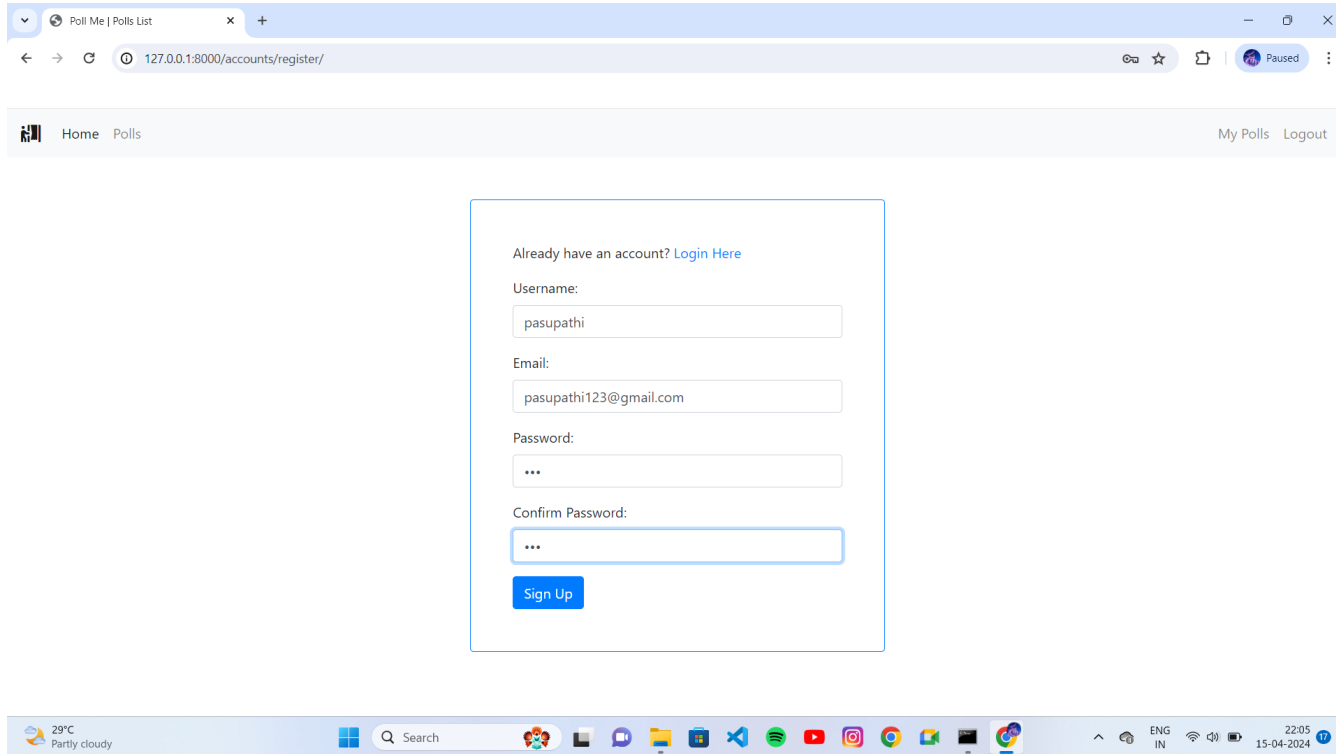


The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/polls/11/results/`. The page has a teal header with the text "VOTING AREA". Below the header, the question "Django is written in which language?" is displayed. A table lists three options with their respective vote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



Poll Me | Polls List

127.0.0.1:8000/accounts/register/

Home Polls

My Polls Logout

Already have an account? [Login Here](#)

Username:

pasupathi

Email:

pasupathi123@gmail.com

Password:

...

Confirm Password:

...

Sign Up

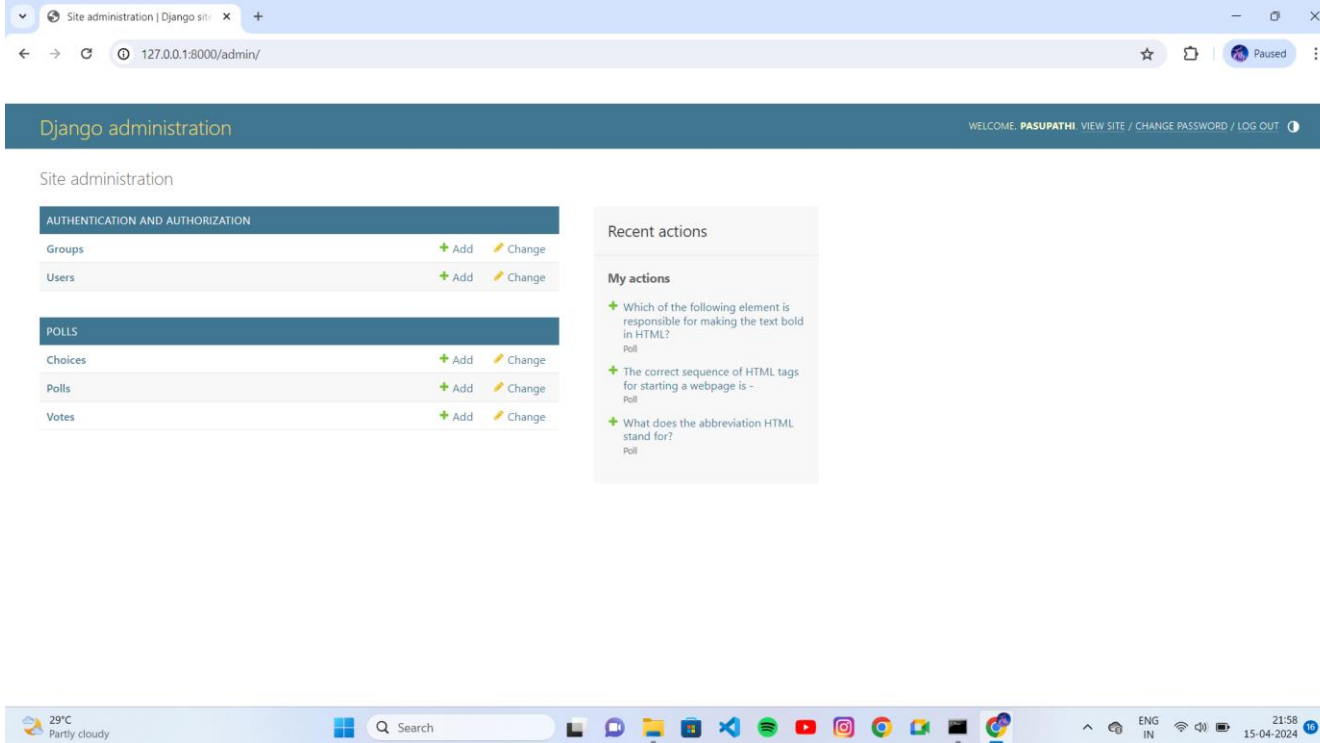
29°C Partly cloudy

Search

ENG IN

22:05 15-04-2024

## Admin Home Page



The screenshot shows the Django administration interface in a web browser. The browser tab is titled "Site administration | Django site" and the address bar shows "127.0.0.1:8000/admin/". The interface has a dark blue header with the text "Django administration" and a welcome message: "WELCOME, PASUPATHI. VIEW SITE / CHANGE PASSWORD / LOG OUT". Below the header, the main content area is titled "Site administration". It contains two main sections: "AUTHENTICATION AND AUTHORIZATION" and "POLLS". The "AUTHENTICATION AND AUTHORIZATION" section has two rows: "Groups" and "Users", each with "Add" and "Change" links. The "POLLS" section has three rows: "Choices", "Polls", and "Votes", each with "Add" and "Change" links. To the right of these sections is a "Recent actions" sidebar. It contains a "My actions" section with three entries, each starting with a green plus icon and followed by a question and the word "Poll". The first entry is "Which of the following element is responsible for making the text bold in HTML?". The second entry is "The correct sequence of HTML tags for starting a webpage is -". The third entry is "What does the abbreviation HTML stand for?". At the bottom of the screen is a Windows taskbar showing the date and time as "21:58 15-04-2024" and the weather as "29°C Partly cloudy".

Site administration | Django site

127.0.0.1:8000/admin/

Django administration

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

Site administration

**AUTHENTICATION AND AUTHORIZATION**

Groups	<a href="#">Add</a>	<a href="#">Change</a>
Users	<a href="#">Add</a>	<a href="#">Change</a>

**POLLS**

Choices	<a href="#">Add</a>	<a href="#">Change</a>
Polls	<a href="#">Add</a>	<a href="#">Change</a>
Votes	<a href="#">Add</a>	<a href="#">Change</a>

**Recent actions**

**My actions**

- Which of the following element is responsible for making the text bold in HTML?  
Poll
- The correct sequence of HTML tags for starting a webpage is -  
Poll
- What does the abbreviation HTML stand for?  
Poll

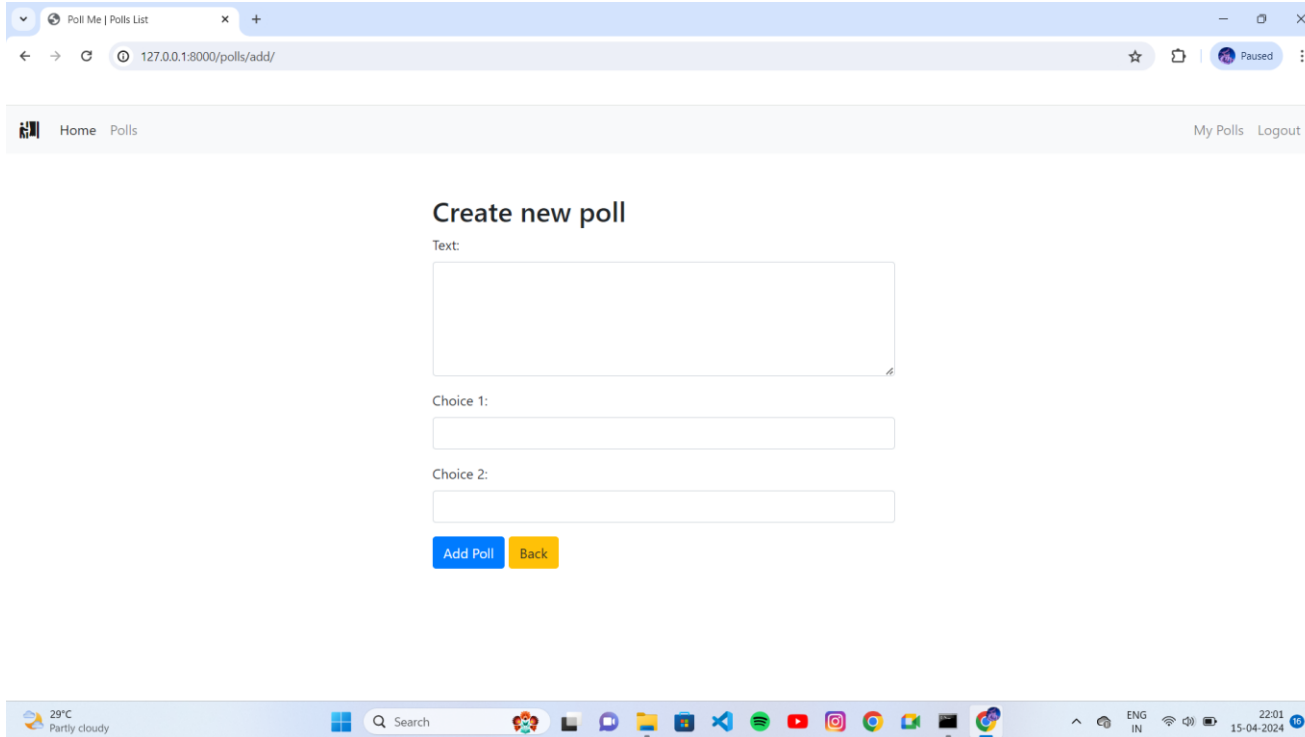
29°C Partly cloudy

Search

ENG IN

21:58 15-04-2024

## Authentication and Authorization Page



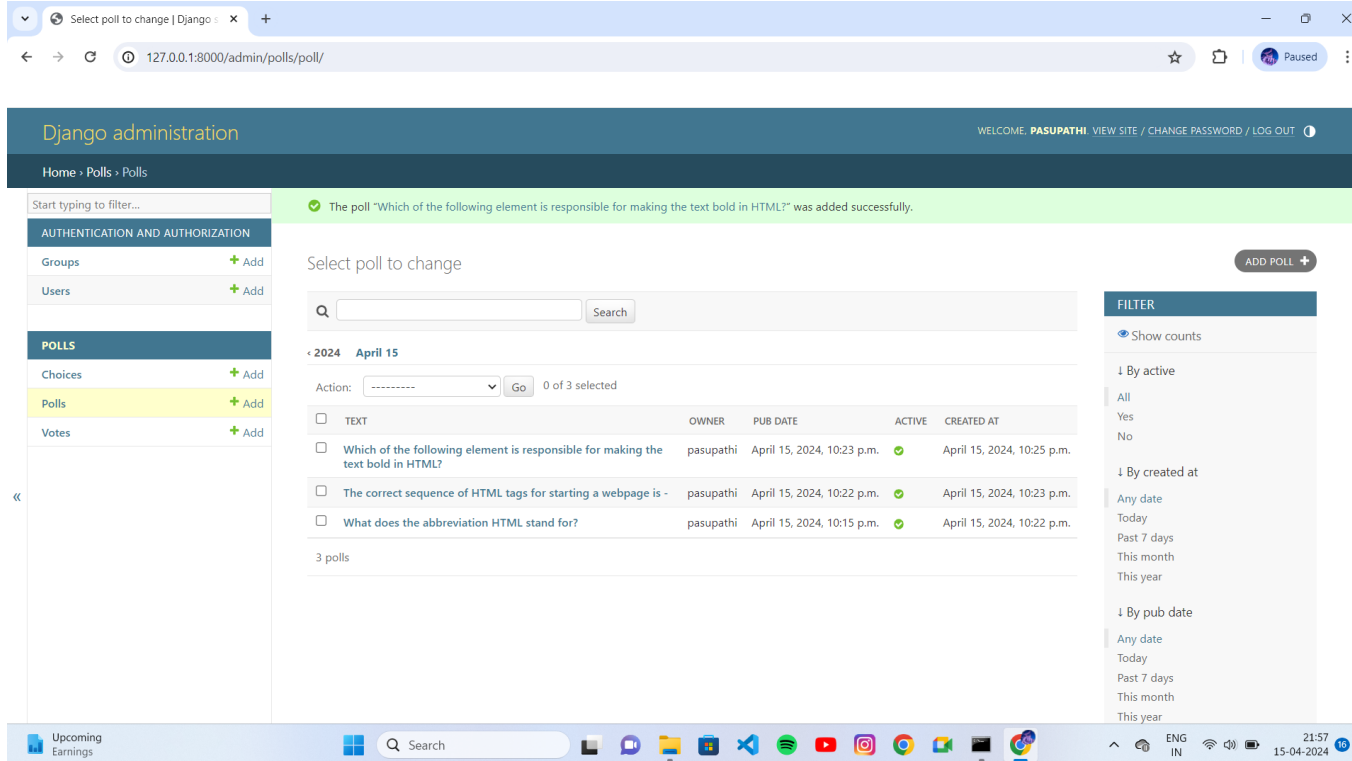
The screenshot shows a web browser window with the address bar displaying '127.0.0.1:8000/polls/add/'. The page has a navigation bar with 'Home' and 'Polls' links, and 'My Polls' and 'Logout' links on the right. The main content area is titled 'Create new poll' and contains a form with the following fields:

- Text:** A large text area for entering the poll question.
- Choice 1:** A text input field for the first choice.
- Choice 2:** A text input field for the second choice.

At the bottom of the form are two buttons: 'Add Poll' (blue) and 'Back' (yellow).

The Windows taskbar at the bottom shows the date and time as 22:01 on 15-04-2024, along with various system icons and application shortcuts.

## Questions Adding Section Page



Django administration

WELCOME, PASUPATHI [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Polls > Polls

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups [+ Add](#)

Users [+ Add](#)

POLL'S

Choices [+ Add](#)

Polls [+ Add](#)

Votes [+ Add](#)

✓ The poll "Which of the following element is responsible for making the text bold in HTML?" was added successfully.

Select poll to change

Q  Search

2024 April 15

Action:  Go 0 of 3 selected

<input type="checkbox"/>	TEXT	OWNER	PUB DATE	ACTIVE	CREATED AT
<input type="checkbox"/>	Which of the following element is responsible for making the text bold in HTML?	pasupathi	April 15, 2024, 10:23 p.m.	✓	April 15, 2024, 10:25 p.m.
<input type="checkbox"/>	The correct sequence of HTML tags for starting a webpage is -	pasupathi	April 15, 2024, 10:22 p.m.	✓	April 15, 2024, 10:23 p.m.
<input type="checkbox"/>	What does the abbreviation HTML stand for?	pasupathi	April 15, 2024, 10:15 p.m.	✓	April 15, 2024, 10:22 p.m.

3 polls

FILTER

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

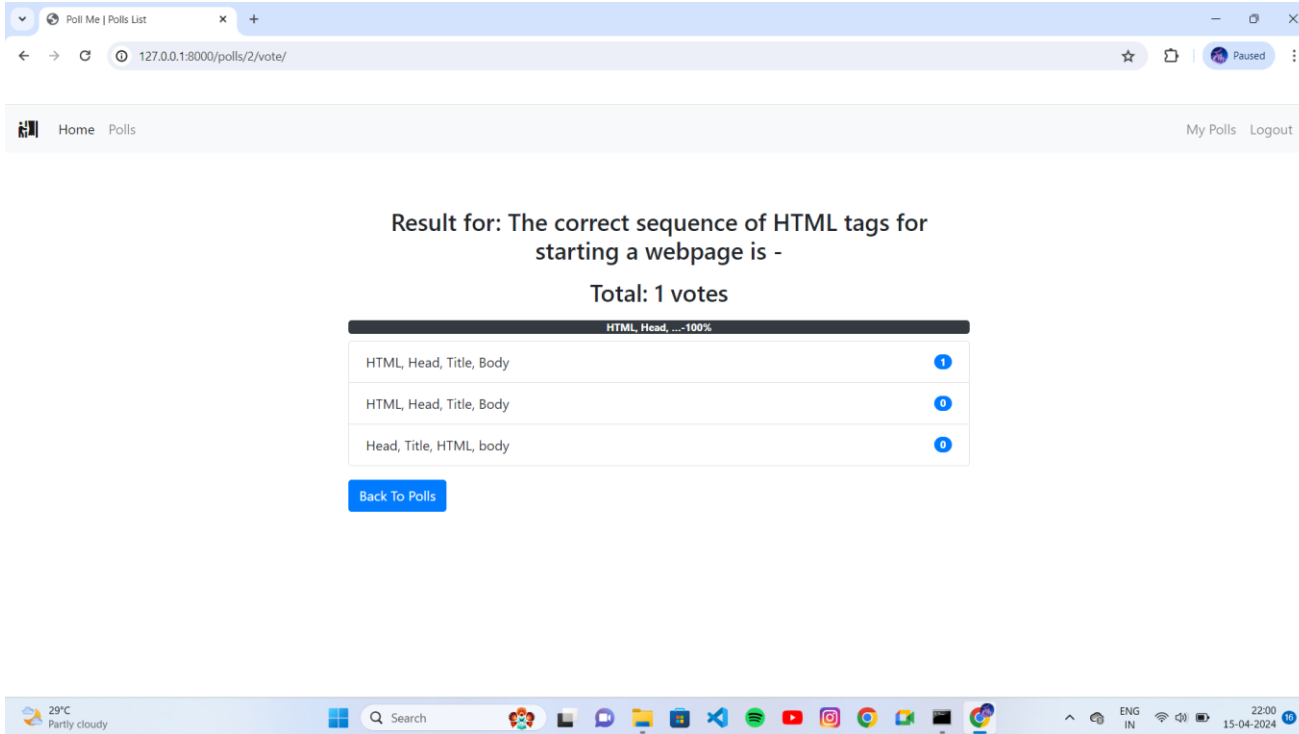
Upcoming Earnings

Search

ENG IN

21:57 15-04-2024

## 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 a user profile section with "My Polls" and "Logout" links. The main content area displays the result of a poll: "Result for: The correct sequence of HTML tags for starting a webpage is -". Below this, it states "Total: 1 votes". A table shows the poll results:

HTML, Head, ...-100%	
HTML, Head, Title, Body	1
HTML, Head, Title, Body	0
Head, Title, HTML, body	0

Below the table is a blue button labeled "Back To Polls". The Windows taskbar at the bottom shows the date as 15-04-2024 and the time as 22:00.

## 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!**