



# NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Student Name : VEDAVARSHINI G  
Student ID : au820621104087

College Name

Arasu Engineering College

## CAPSTONE PROJECT SHOWCASE

### Project Title

Django Poll App Voting Project- VEDAVARSHINI (4087,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:

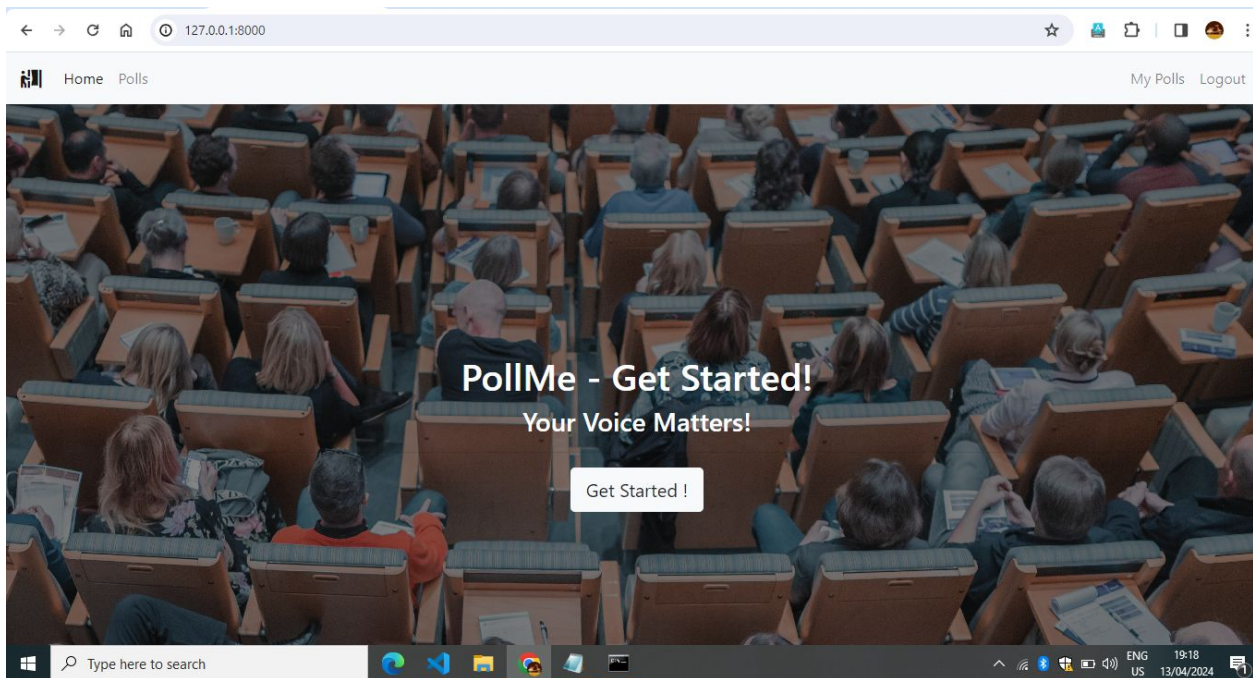
- 1. Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2. Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3. Creating User Authentication:** Implement user authentication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



## Welcome to polls List!

Poll & Choices added successfully. ✕

👤 Name 🕒 Date 🗳️ Vote Add +

🔍

What is the full form ... ✎ 🗳️

Which of the following language ... ✎ 🗳️



## Voting Page



### Polls details page

Which of the following language does the computer understand?

- ☒ Computer understands only Binary Language
- ☐ Computer understands only C Language

Vote

Cancel

## Voting Details Page



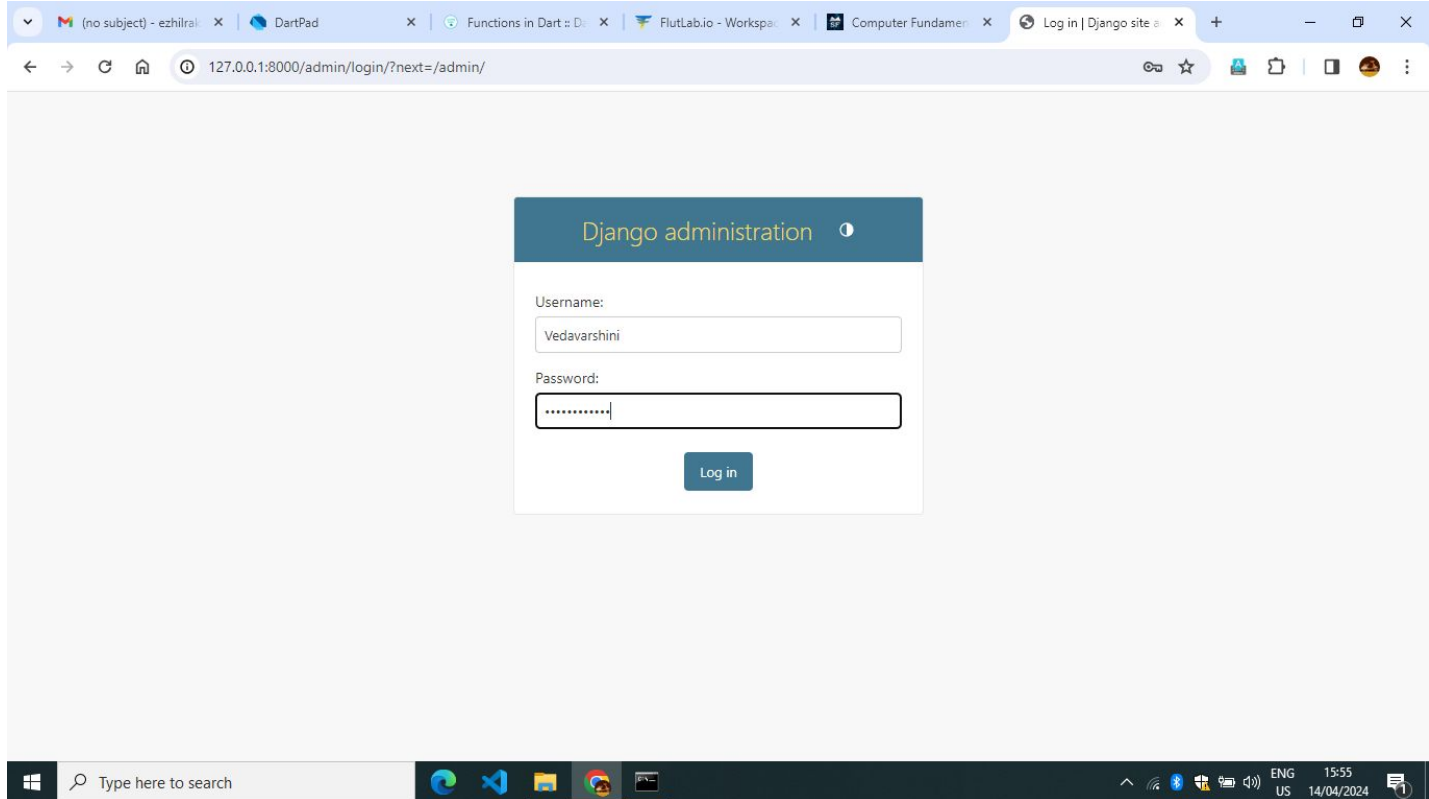
Result for: Which of the following language does the computer understand?

Total: 1 votes

Computer understands ...-100%	
Computer understands only Binary Language	1
Computer understands only C Language	0

[Back To Polls](#)

## Admin home



Browser tabs: (no subject) - ezhilra, DartPad, Functions in Dart, FlutLab.io - Workspa, Computer Fundamen, Log in | Django site

Address bar: 127.0.0.1:8000/admin/login/?next=/admin/

Django administration

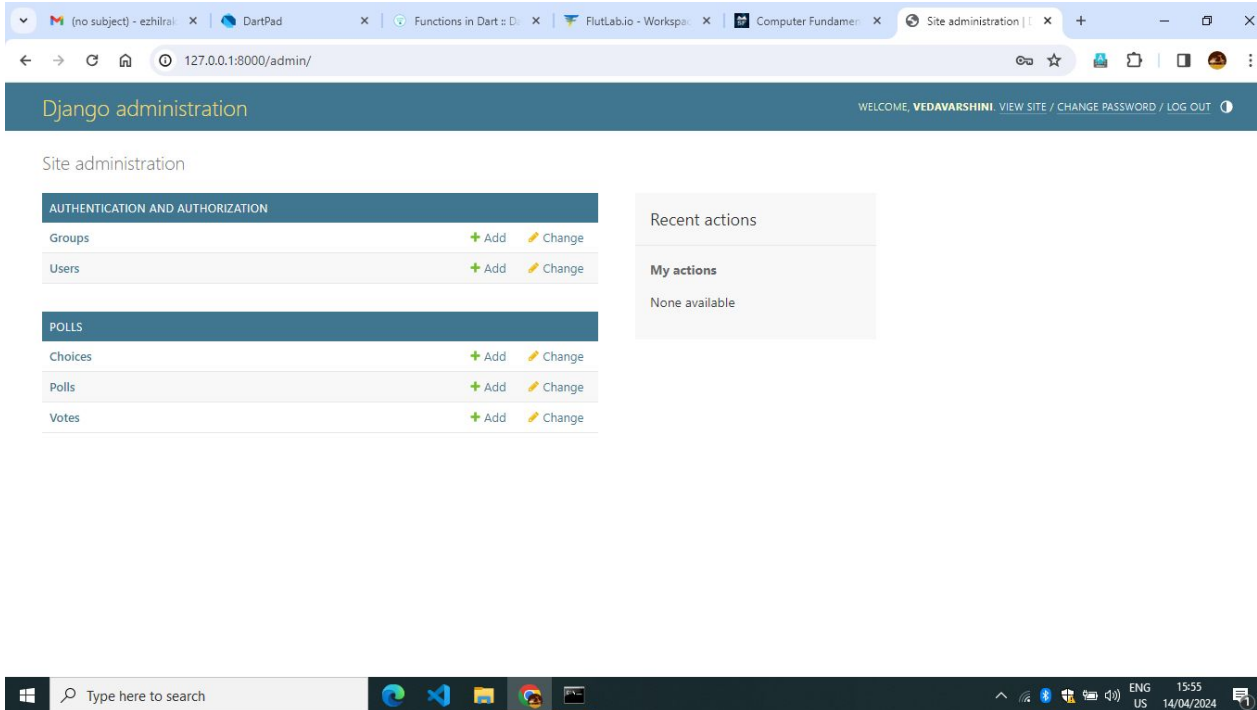
Username:

Password:

Log in

Windows taskbar: Type here to search, ENG US, 15:55, 14/04/2024

## Admin Home Page



The screenshot shows a web browser window with multiple tabs. The active tab is 'Site administration', showing the Django administration interface. The browser's address bar displays '127.0.0.1:8000/admin/'. The Django admin header is dark blue with the text 'Django administration' and a welcome message for 'VEDAVARSHINI'. The main content area is titled 'Site administration' and contains two primary sections: 'AUTHENTICATION AND AUTHORIZATION' and 'POLLS'. Each section has a table with links to 'Add' and 'Change' items. To the right, there are boxes for 'Recent actions' and 'My actions', both showing 'None available'. The Windows taskbar at the bottom includes a search bar, application icons, and system status information.

Browser tabs: (no subject) - ezhlra, DartPad, Functions in Dart, FlutterLab.io - Workspa, Computer Fundamen, Site administration |

Address bar: 127.0.0.1:8000/admin/

Django administration WELCOME, VEDAVARSHINI. 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>

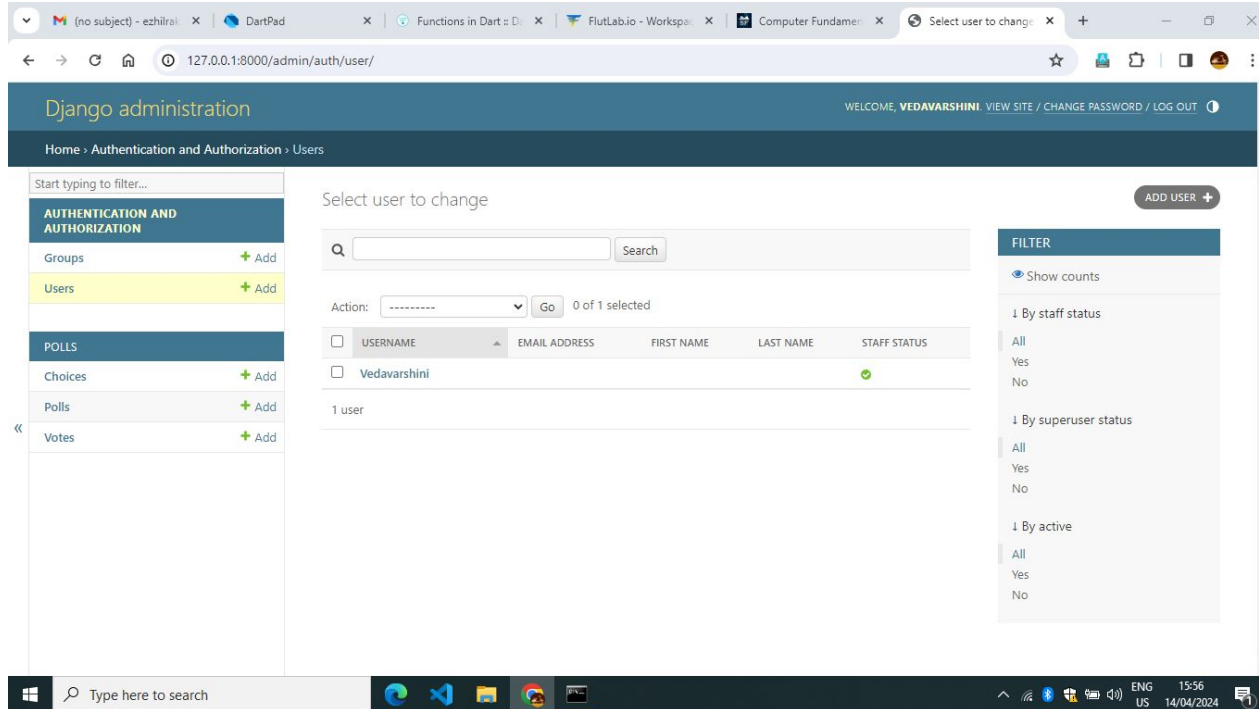
Recent actions

My actions

None available

Windows taskbar: Type here to search, 15:55, 14/04/2024, ENG US

## Authentication and Authorization Page



The screenshot displays the Django administration interface for the 'Authentication and Authorization' section, specifically the 'Users' page. The browser address bar shows the URL '127.0.0.1:8000/admin/auth/user/'. The page header includes the Django logo and the text 'Django administration', along with a welcome message for 'VEDAVARSHINI' and links for 'VIEW SITE', 'CHANGE PASSWORD', and 'LOG OUT'.

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

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

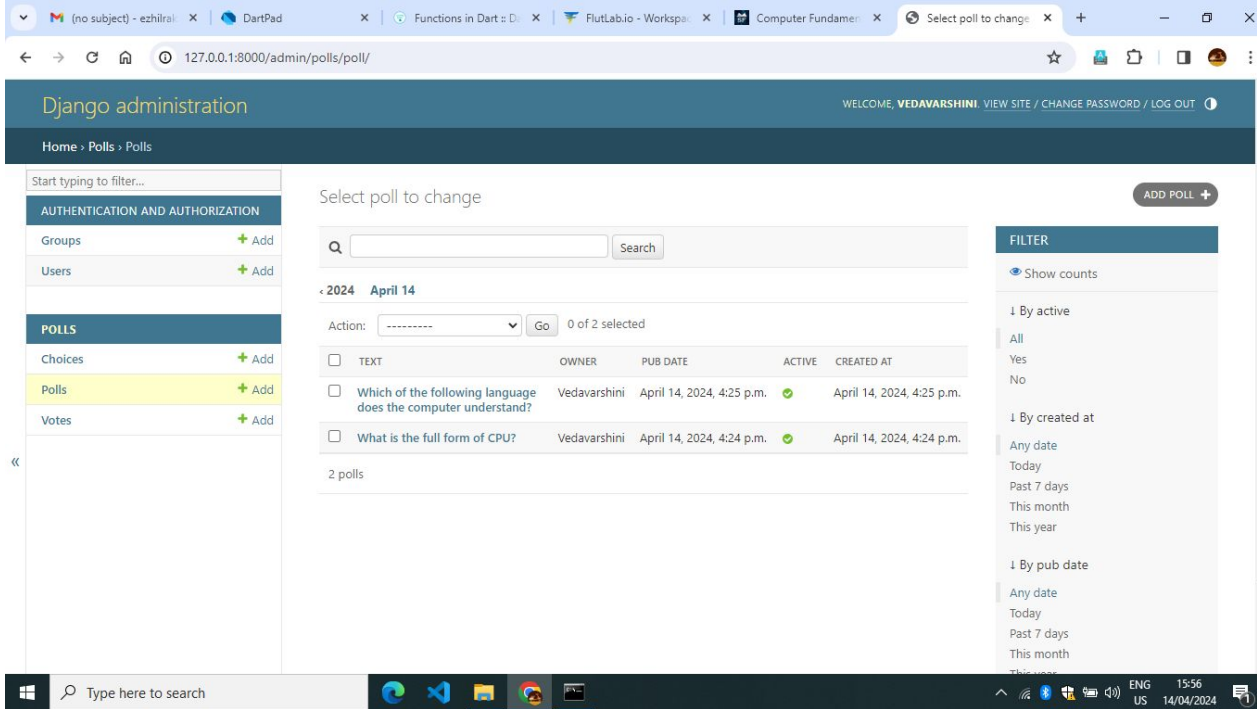
The main content area is titled 'Select user to change'. It features a search bar and a table of users. The table has columns for USERNAME, EMAIL ADDRESS, FIRST NAME, LAST NAME, and STAFF STATUS. The user 'Vedavarshini' is listed with a green checkmark in the STAFF STATUS column. Below the table, it indicates '1 user'.

On the right side, there is a 'FILTER' section with the following options:

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

The Windows taskbar at the bottom shows the search bar and various application icons, including the Start menu, Edge browser, and others. The system clock indicates the time is 15:56 on 14/04/2024.

## Questions Adding Section Page



The screenshot displays the Django administration interface for managing polls. The browser's address bar shows the URL `127.0.0.1:8000/admin/polls/poll/`. The page title is "Django administration" and the user is logged in as "VEDAVARSHINI".

The left sidebar 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 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".

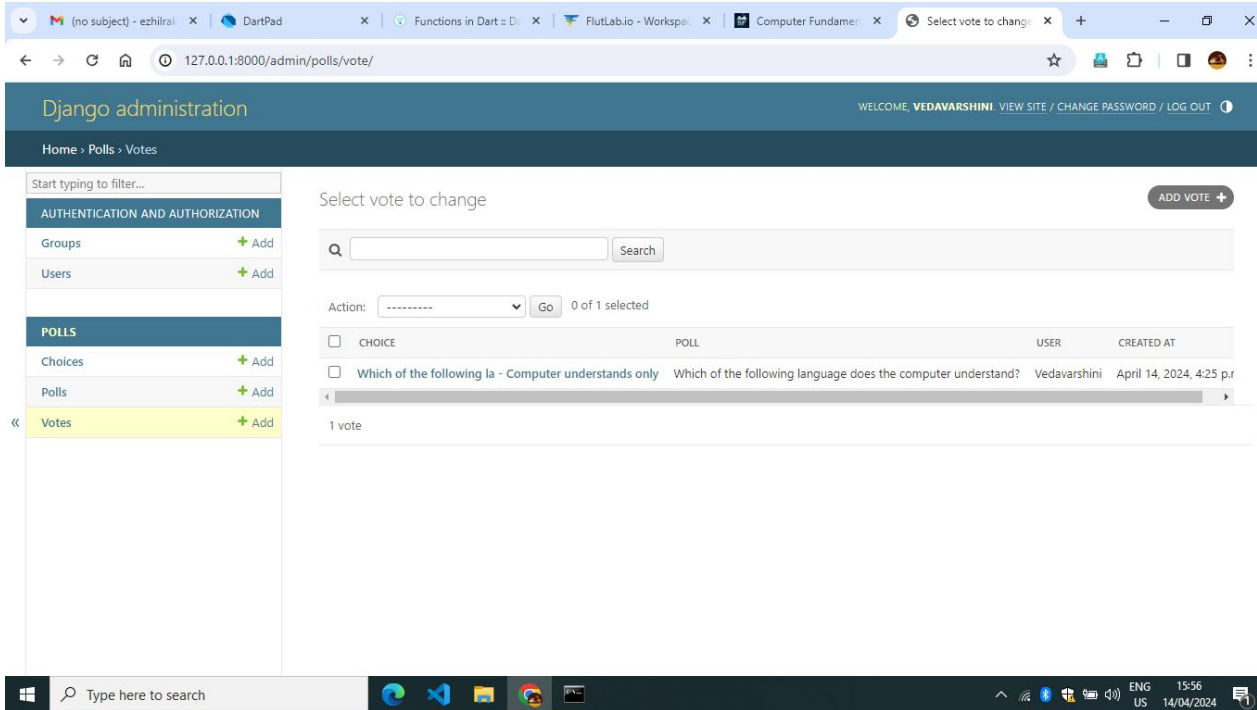
TEXT	OWNER	PUB DATE	ACTIVE	CREATED AT
<input type="checkbox"/> Which of the following language does the computer understand?	Vedavarshini	April 14, 2024, 4:25 p.m.	✓	April 14, 2024, 4:25 p.m.
<input type="checkbox"/> What is the full form of CPU?	Vedavarshini	April 14, 2024, 4:24 p.m.	✓	April 14, 2024, 4:24 p.m.

Below the table, it indicates "2 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

## Voting Details Page



The screenshot displays a web browser window with multiple tabs open. The active tab is titled "Select vote to change". The browser's address bar shows the URL "127.0.0.1:8000/admin/polls/vote/".

The Django administration interface is visible, featuring a dark blue header with the text "Django administration" and a welcome message for "VEDAVARSHINI". The breadcrumb trail indicates the current location: "Home > Polls > Votes".

On the left sidebar, the "VOTES" section is highlighted. The main content area is titled "Select vote to change" and includes a search bar and a table of votes.

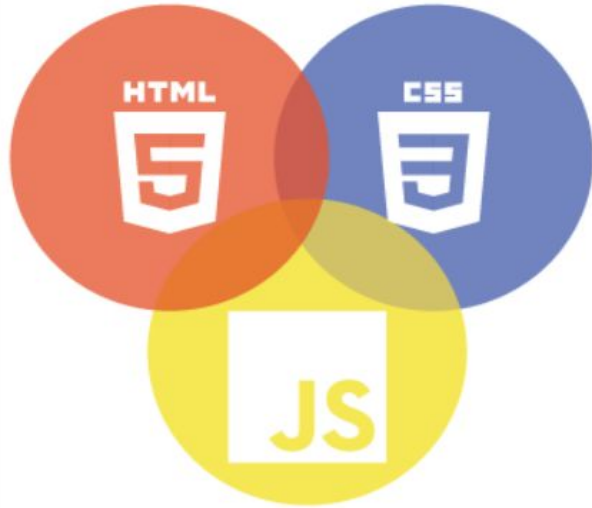
**Table: Select vote to change**

CHOICE	POLL	USER	CREATED AT
<input type="checkbox"/> Which of the following la - Computer understands only	Which of the following language does the computer understand?	Vedavarshini	April 14, 2024, 4:25 p.m.

Below the table, it indicates "1 vote".

## 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!**