



NEXT GEN EMPLOYABILITY PROGRAM

| Creating a future-ready workforce

Student Name : EZHILARASAN T
Student ID : autb21csl006

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Django Voting App-EZHILARASAN T(4307,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

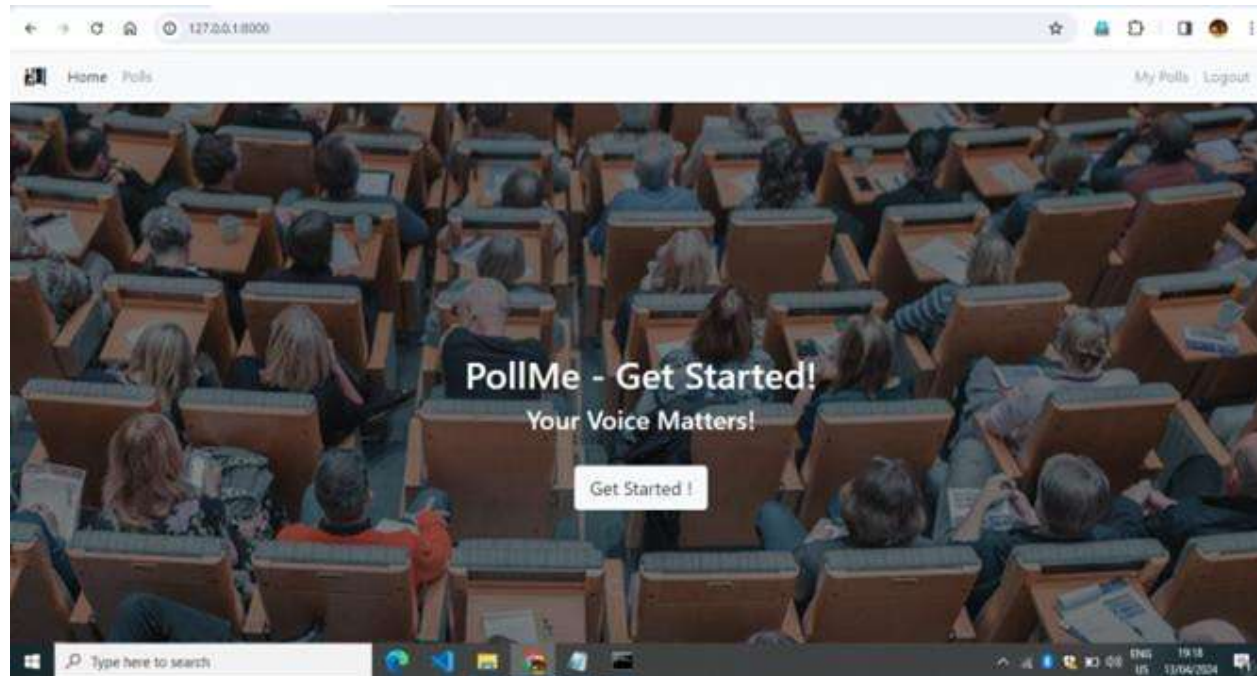
The project overview for a voting application using the Django framework involves creating a secure and user-friendly online voting system. The application allows users to register, vote, and view real-time results. Here is a steps involved in building the voting application:

- 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.

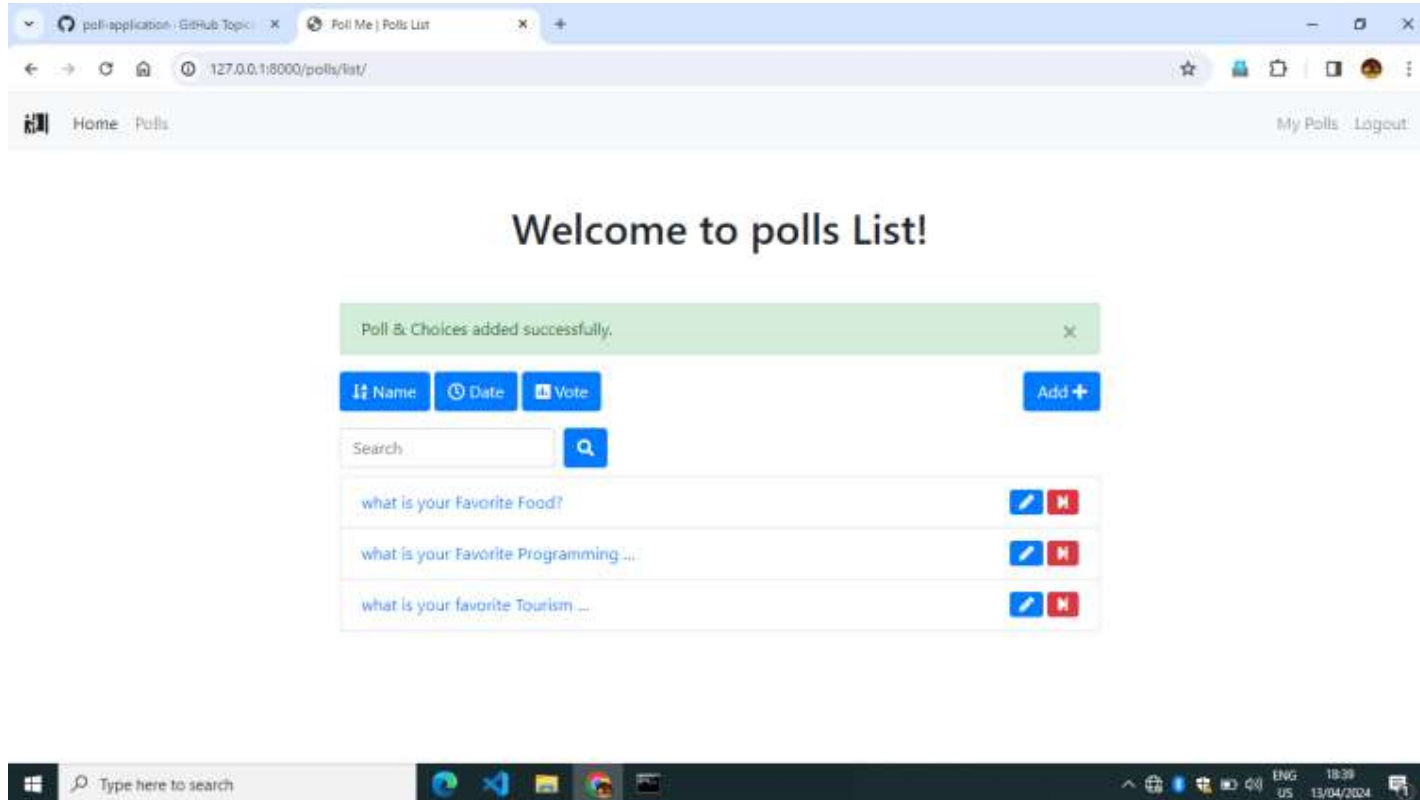
Proposed Solution

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 two tabs: 'poll-application - Github Topic' and 'Poll Me | Polls List'. The address bar shows the URL '127.0.0.1:8000/polls/list/'. The page has a navigation bar with 'Home' and 'Polls' links, and a user profile section with 'My Polls' and 'Logout' links.

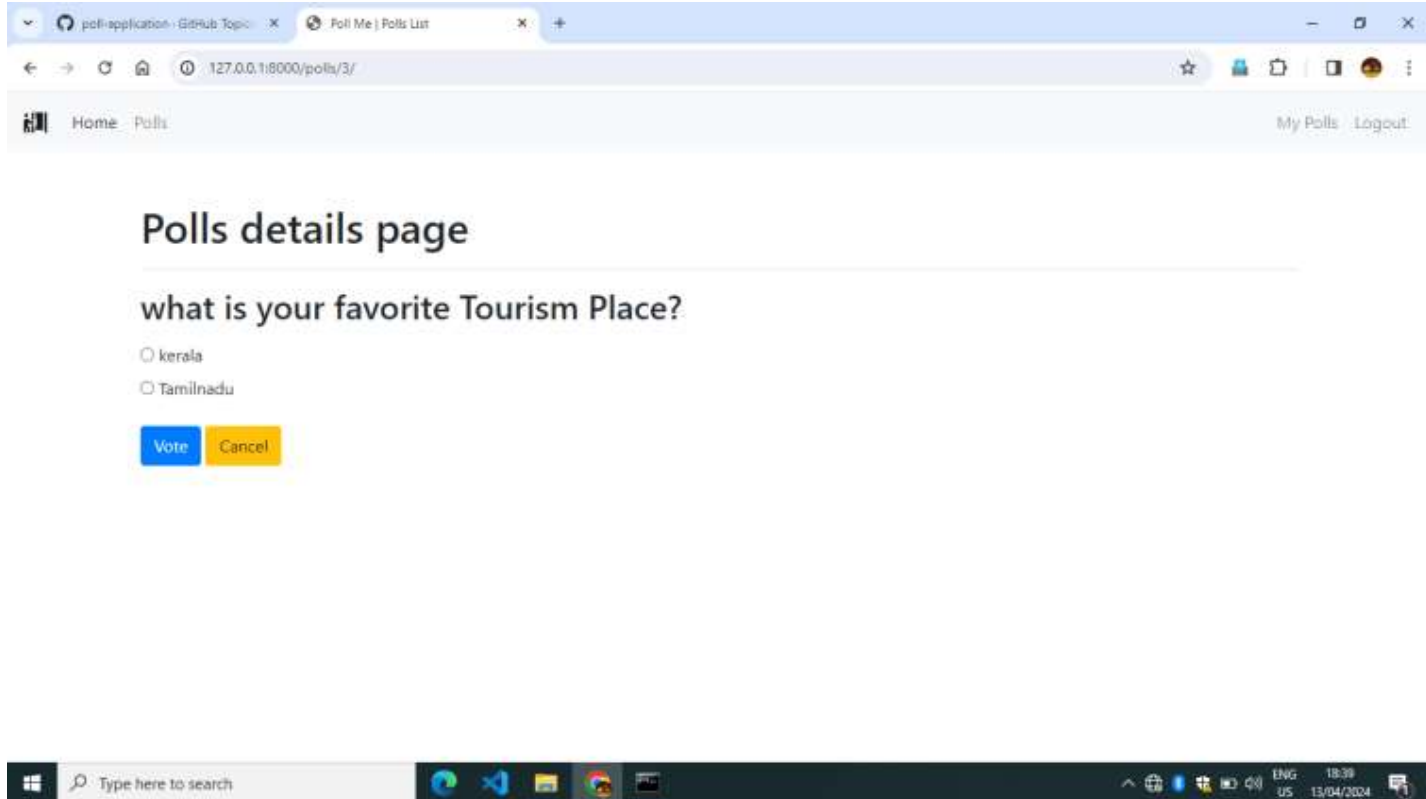
The main content area displays a large heading 'Welcome to polls List!'. Below this, a green success message states 'Poll & Choices added successfully.' with a close button. A toolbar contains buttons for 'Name', 'Date', 'Vote', and 'Add +'. A search bar is also present.

A list of three polls is shown, each with a title, a blue edit icon, and a red delete icon:

- what is your Favorite Food?
- what is your Favorite Programming ...
- what is your favorite Tourism ...

The Windows taskbar at the bottom shows the search bar, taskbar icons, and system tray with the date '13/04/2024' and time '18:39'.

Voting Page



The screenshot shows a web browser window with two tabs: 'poll-application - GitHub Topic' and 'Poll Me | Polls List'. The address bar shows the URL '127.0.0.1:8000/polls/3/'. The browser interface includes navigation buttons, a star icon, and a user profile icon. The page content features a navigation bar with 'Home' and 'Polls' links, and a 'My Polls | Logout' link on the right. The main heading is 'Polls details page'. Below it, the question 'what is your favorite Tourism Place?' is displayed. Two radio button options are listed: 'kerala' and 'Tamilnadu'. At the bottom of the form, there are two buttons: 'Vote' (blue) and 'Cancel' (yellow). The Windows taskbar at the bottom shows the search bar, task view button, and several application icons (Edge, VS Code, File Explorer, Chrome, and a terminal). The system tray on the right indicates the language is 'ENG US', the time is '18:39', and the date is '13/04/2024'.

poll-application - GitHub Topic | Poll Me | Polls List

127.0.0.1:8000/polls/3/

Home | Polls | My Polls | Logout

Polls details page

what is your favorite Tourism Place?

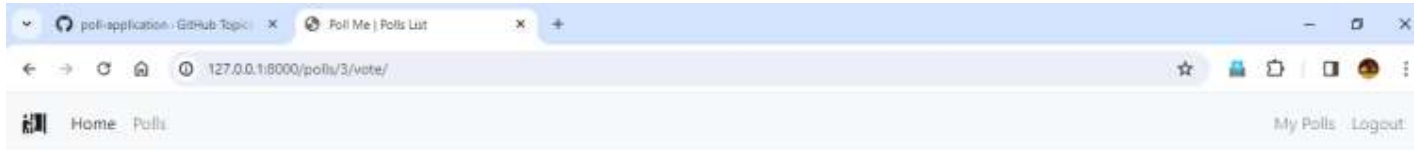
☐ kerala

☐ Tamilnadu

Type here to search

ENG US 18:39 13/04/2024

Voting Details Page

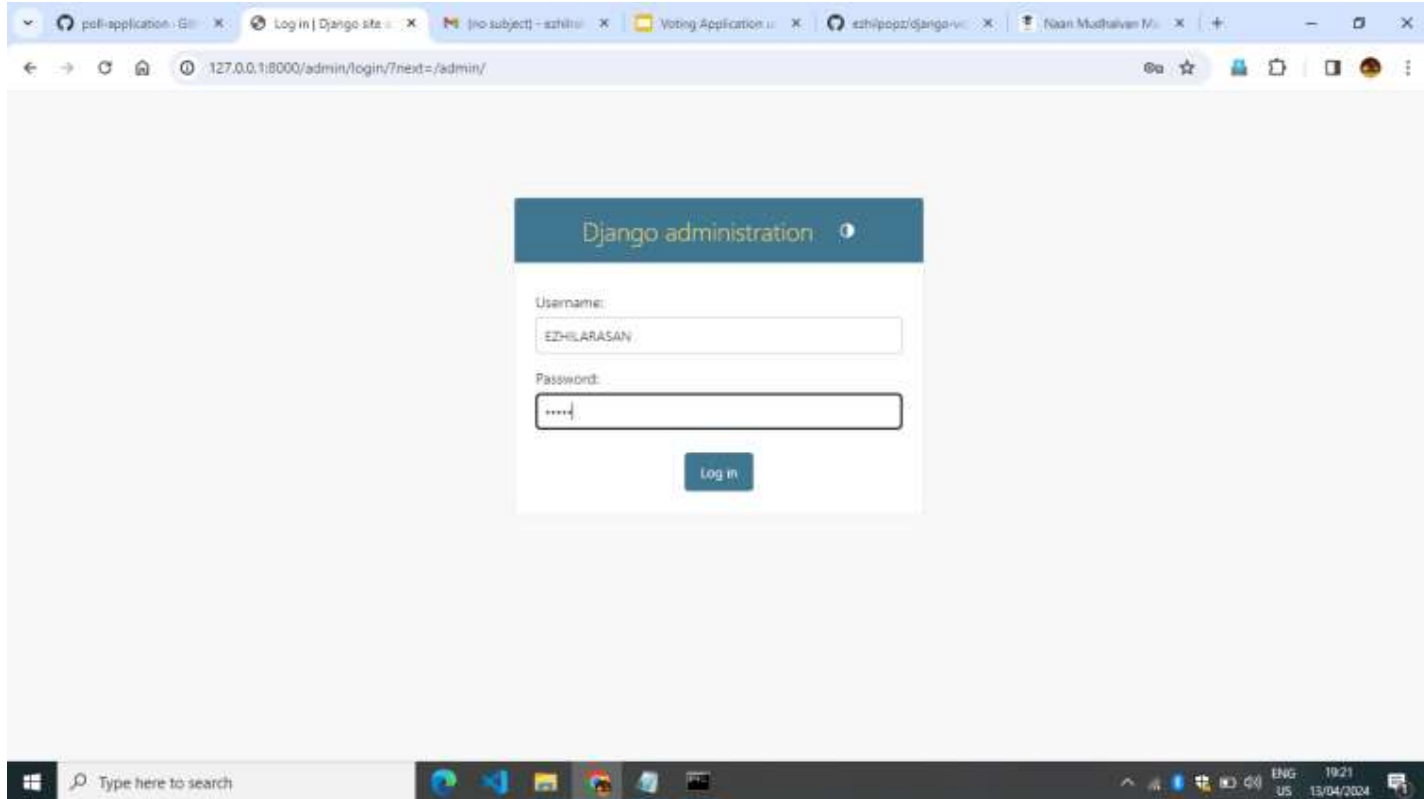


Result for: what is your favorite Tourism Place?

Total: 1 votes



Admin home



Log in | Django site

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

Django administration

Username:

EZHILARASAN

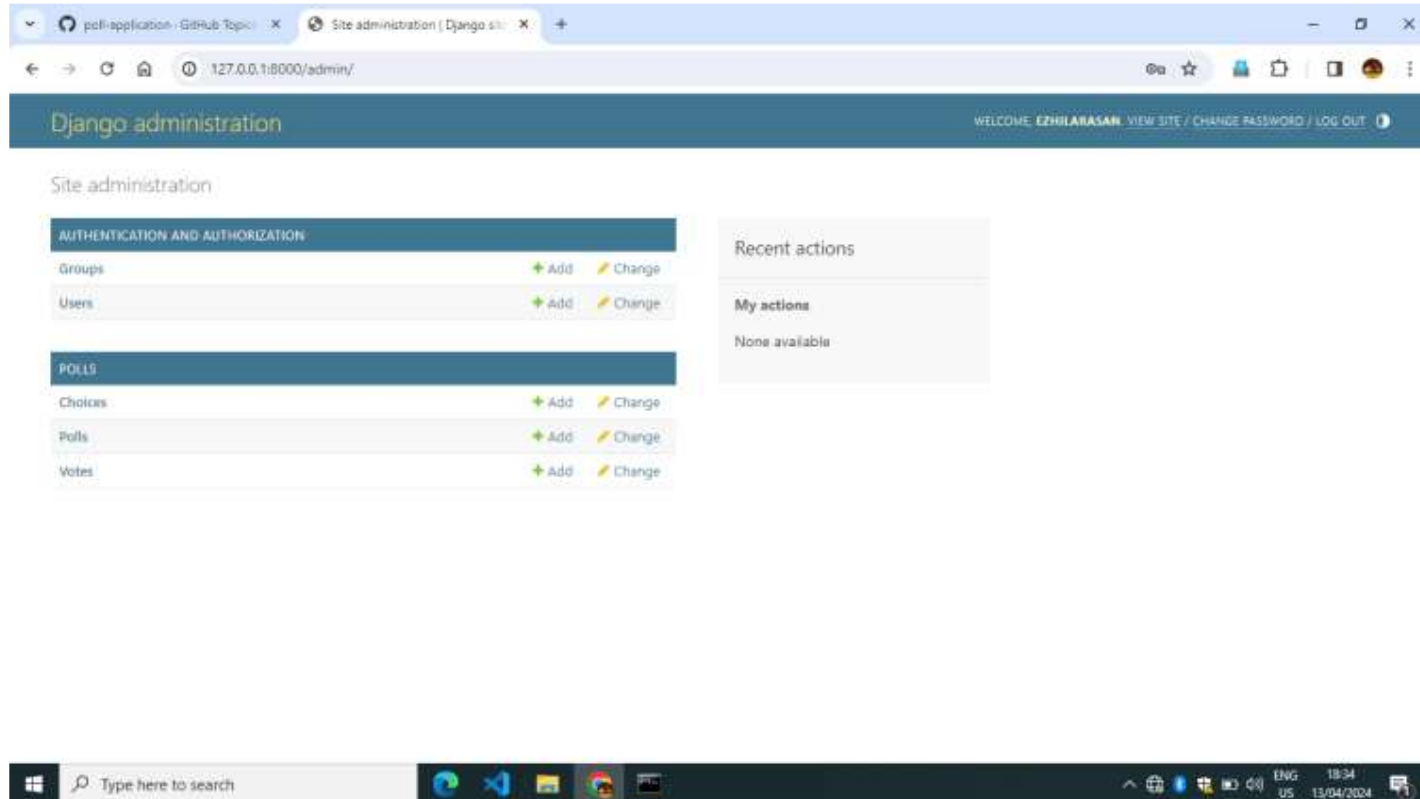
Password:

Log in

Type here to search

ENG US 19:21 13/04/2024

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

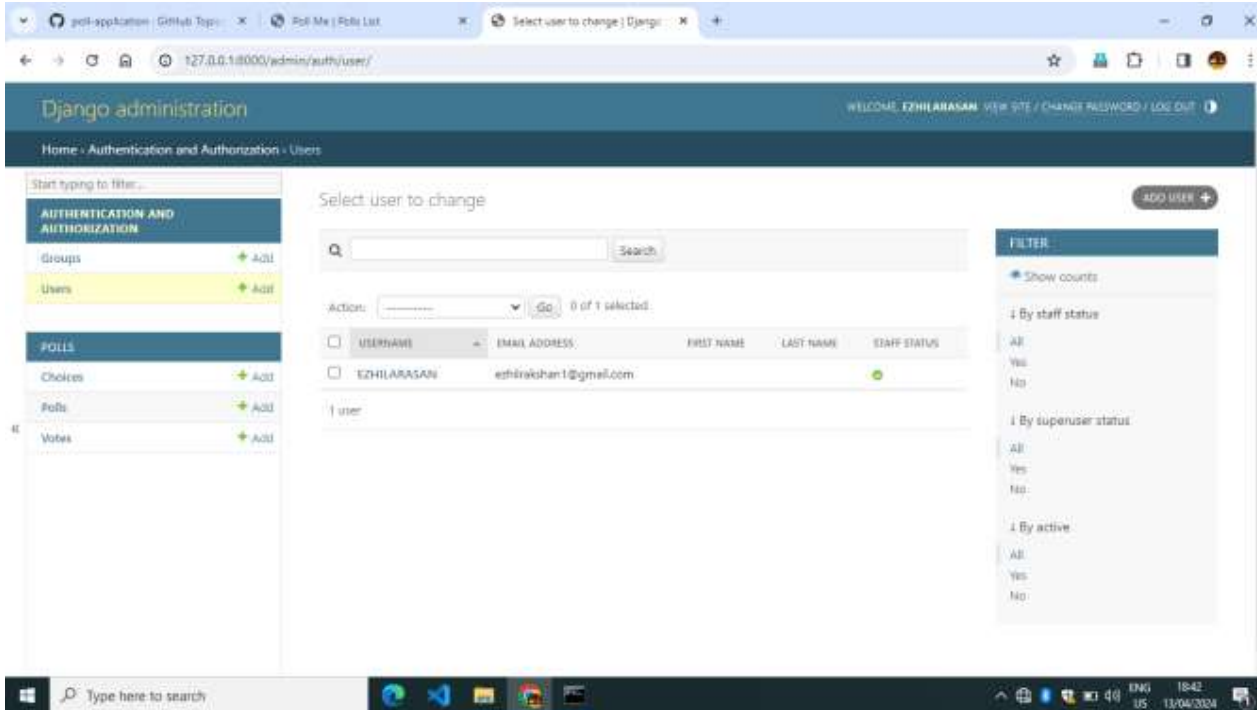
The main content area is titled "Site administration" and is divided into two columns. The left column contains two sections:

- AUTHENTICATION AND AUTHORIZATION**: This section includes links for "Groups" and "Users", each with "Add" and "Change" options.
- POLLS**: This section includes links for "Choices", "Polls", and "Votes", each with "Add" and "Change" options.

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 with the text "Type here to search" and several application icons. The system tray on the right indicates the language is "ENG US", the time is "18:34", and the date is "13/04/2024".

Authentication and Authorization Page



The screenshot displays the Django administration interface for a poll application. The browser tabs include 'poll-application: GitHub Top...', 'Poll Me | Poll List', and 'Select user to change | Django...'. The address bar shows the URL '127.0.0.1:8000/admin/auth/user/'. The page title is 'Django administration' with a welcome message for 'EZHILARASAN' and links for 'VIEW SITE', 'CHANGE PASSWORD', and '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](#)

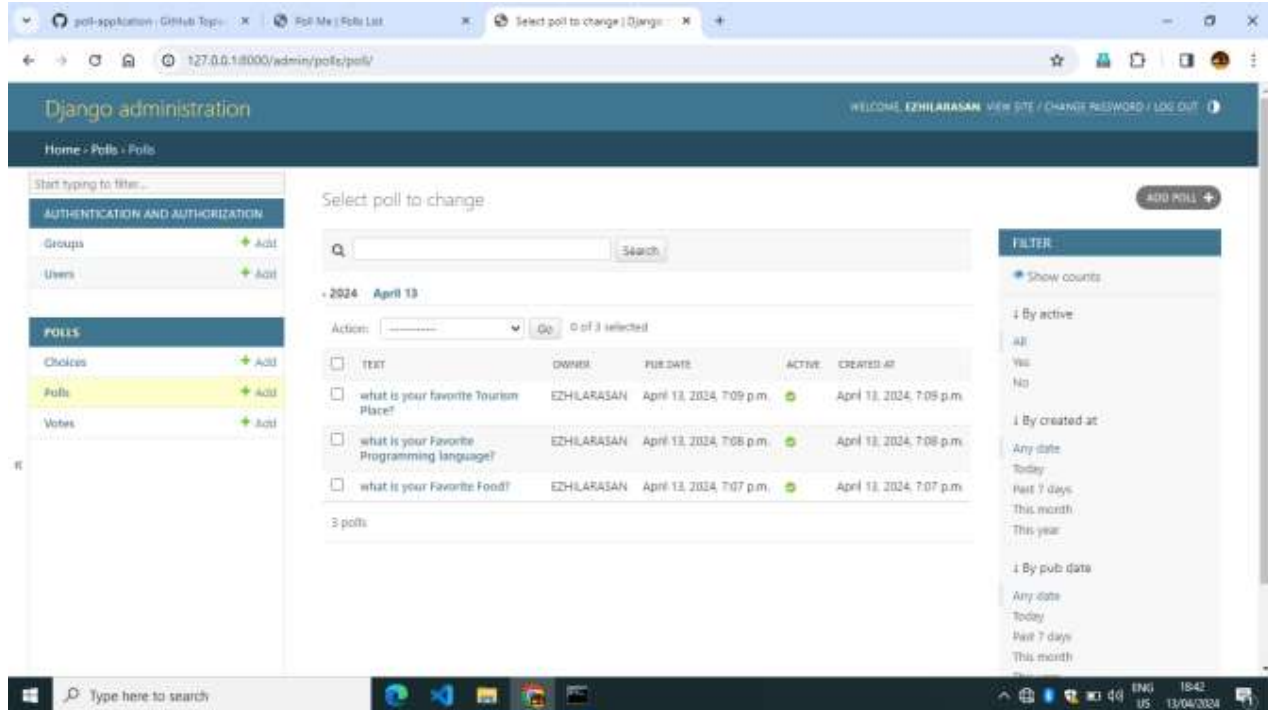
The main content area is titled 'Select user to change'. It features a search bar and a table of users. The table has columns for 'USERNAME', 'EMAIL ADDRESS', 'FIRST NAME', 'LAST NAME', and 'STAFF STATUS'. One user is listed: 'EZHILARASAN' with email 'ezhilakshan1@gmail.com' and a green status indicator. Below the table, it says '1 user'.

On the right side, there is 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

The Windows taskbar at the bottom shows the search bar, taskbar icons, and system clock indicating 18:42 on 13/04/2024.

Questions Adding Section Page



The screenshot displays the Django administration interface for the 'polls' app. The page title is 'Django administration' and the user is logged in as 'EZHILARASAN'. The left sidebar shows the navigation menu with 'POLLs' selected. The main content area is titled 'Select poll to change' and contains a search bar, a filter sidebar, and a table of polls.

Search: A search bar with the placeholder text 'Search' and a 'Search' button.

Filter: A sidebar with the title 'FILTER' and the following options:

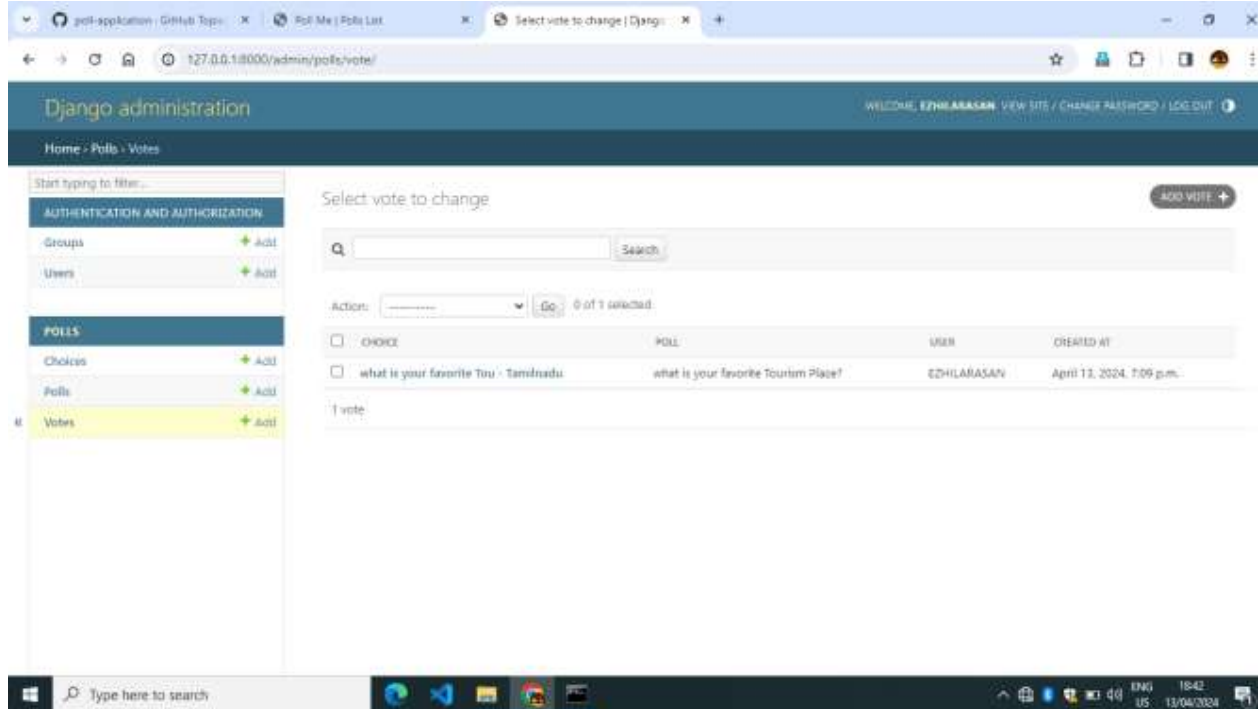
- Show counts
- By active: All, Yes, No
- By created at: Any date, Today, Past 7 days, This month, This year
- By pub date: Any date, Today, Past 7 days, This month

Polls Table: A table with the following columns: ☐ (checkbox), TEXT, OWNER, PUB DATE, ACTIVE, and CREATED AT. The table contains three rows of polls:

<input type="checkbox"/>	TEXT	OWNER	PUB DATE	ACTIVE	CREATED AT
<input type="checkbox"/>	what is your favorite Tourism Place?	EZHILARASAN	April 13, 2024, 7:09 p.m.	Yes	April 13, 2024, 7:09 p.m.
<input type="checkbox"/>	what is your Favorite Programming language?	EZHILARASAN	April 13, 2024, 7:08 p.m.	Yes	April 13, 2024, 7:08 p.m.
<input type="checkbox"/>	what is your Favorite Food?	EZHILARASAN	April 13, 2024, 7:07 p.m.	Yes	April 13, 2024, 7:07 p.m.

At the bottom of the table, it says '3 polls'.

Voting Details Page



The screenshot shows the Django administration interface for the 'polls' app. The browser address bar indicates the URL is `127.0.0.1:8000/admin/polls/vote/`. The page title is 'Django administration'. The user is logged in as 'EZHILARASAN' and can view the site, change their password, or log out.

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**

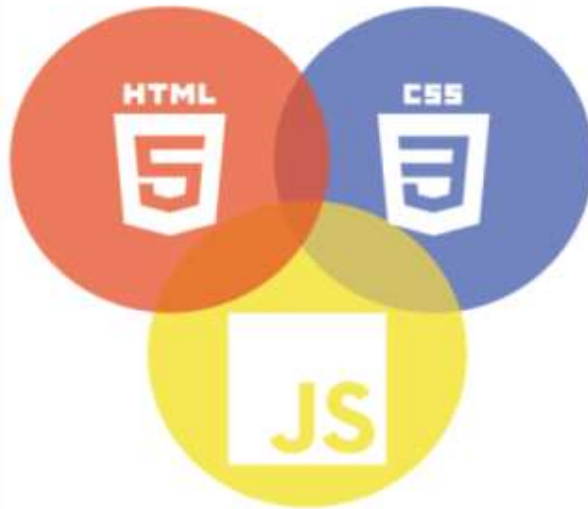
The main content area is titled 'Select vote to change' and includes an 'ADD VOTE +' button. Below this is a search bar and an action dropdown menu. The table below shows the available votes for editing:

CHOICE	POLL	USER	CREATED AT
<input type="checkbox"/> what is your favorite You - Tamilnadu	what is your favorite Tourism Place?	EZHILARASAN	April 13, 2024, 7:09 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!