



edunet  
foundation



## NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Student Name :Shahin Rose M  
Student ID :au820621104079

College Name

Arasu Engineering College

# CAPSTONE PROJECT SHOWCASE

## Project Title

Voting Application using Django Framework-Shahin rose M(4079,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:

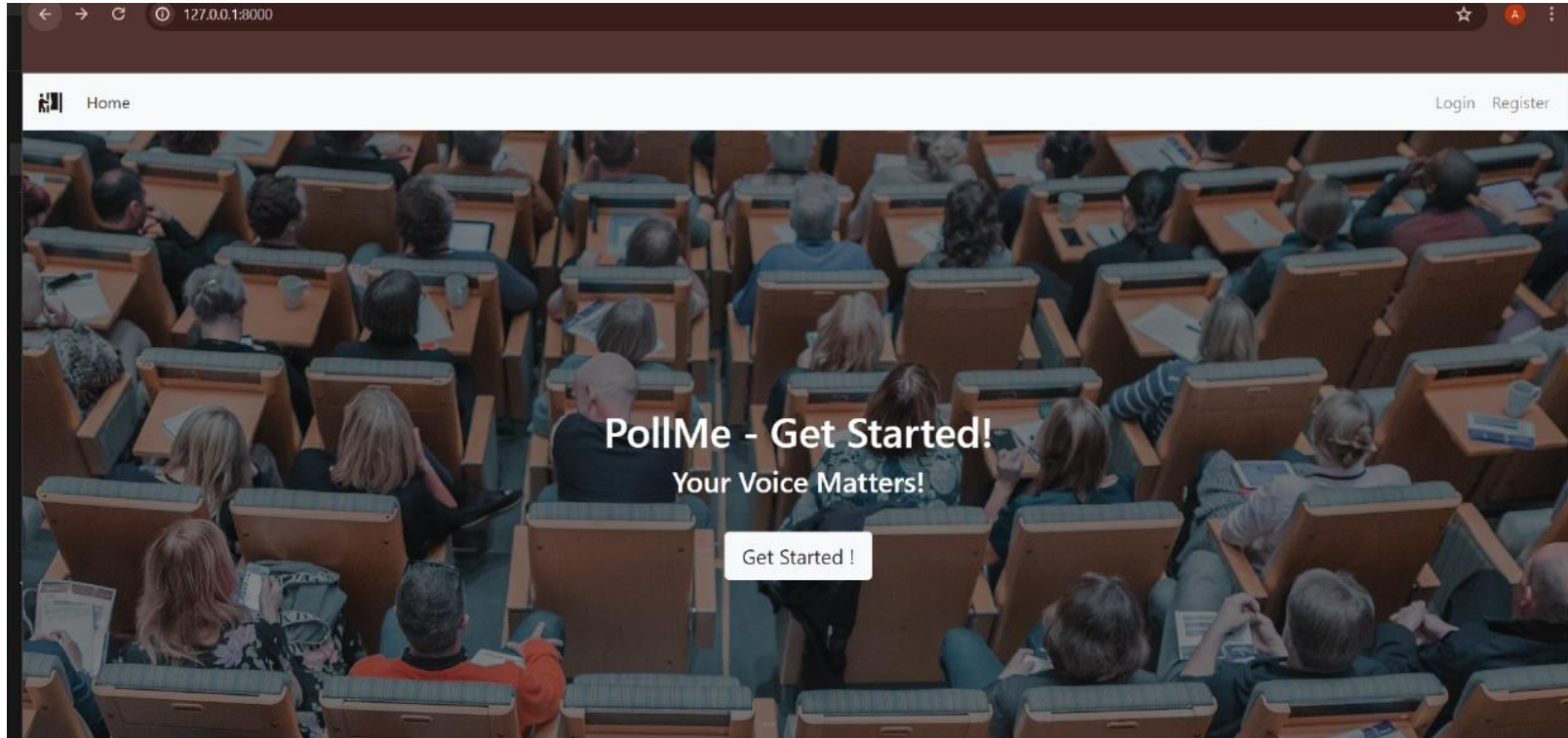
- 1.Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2.Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3.Creating User Authentication:** Implement user authentication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

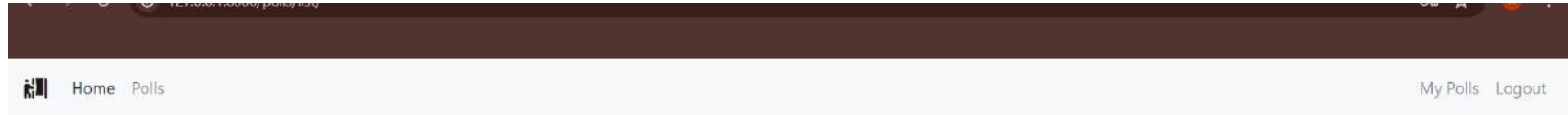
The proposed solution for a voting application using the Django framework is to create a secure and user-friendly online voting platform. The application will allow users to register, vote, and view real-time results. To build the application, the Django framework will be used as the foundation due to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

 Name

 Date

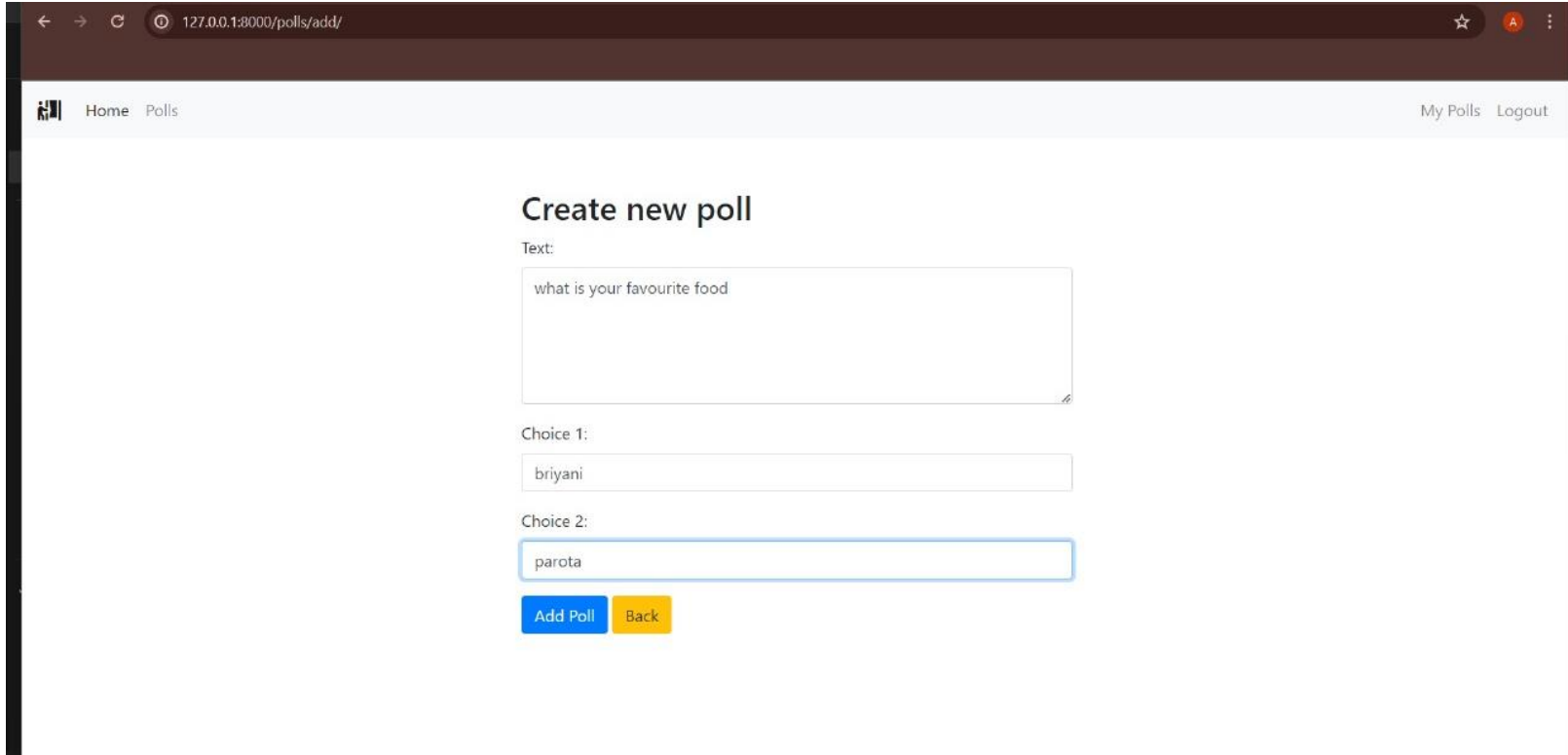
 Vote

Add 





## Voting Page



← → ↻ 127.0.0.1:8000/polls/add/ ☆ A ⋮

Home Polls My Polls Logout

### Create new poll

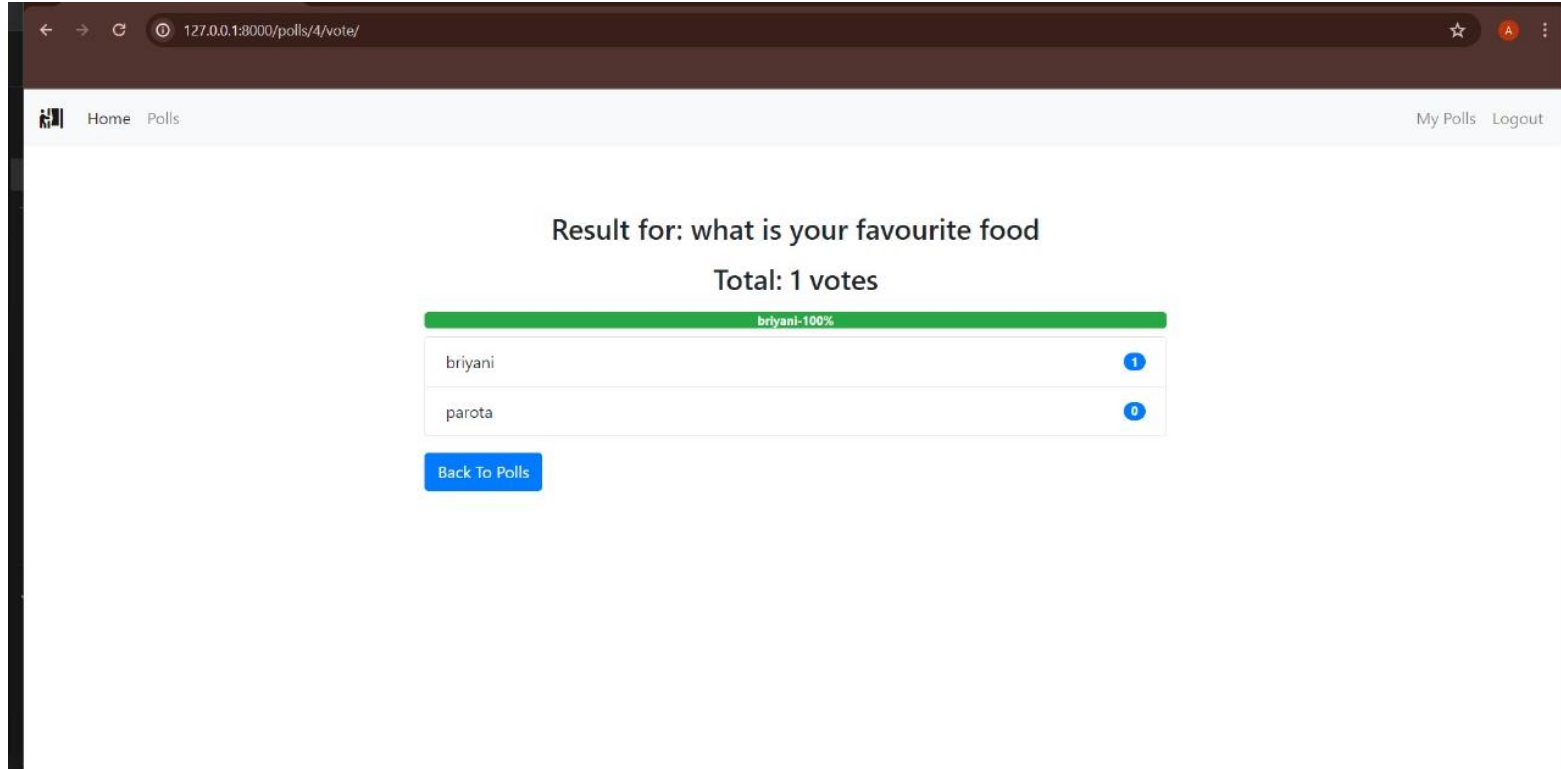
Text:

Choice 1:

Choice 2:

Add Poll Back

## Voting Details Page



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/polls/4/vote/". The page has a navigation bar with "Home" and "Polls" links, and a user menu with "My Polls" and "Logout". The main content area displays the poll results for the question "what is your favourite food". The results show "briyani" with 100% of the votes (1 vote) and "parota" with 0 votes. A "Back To Polls" button is located at the bottom of the results section.

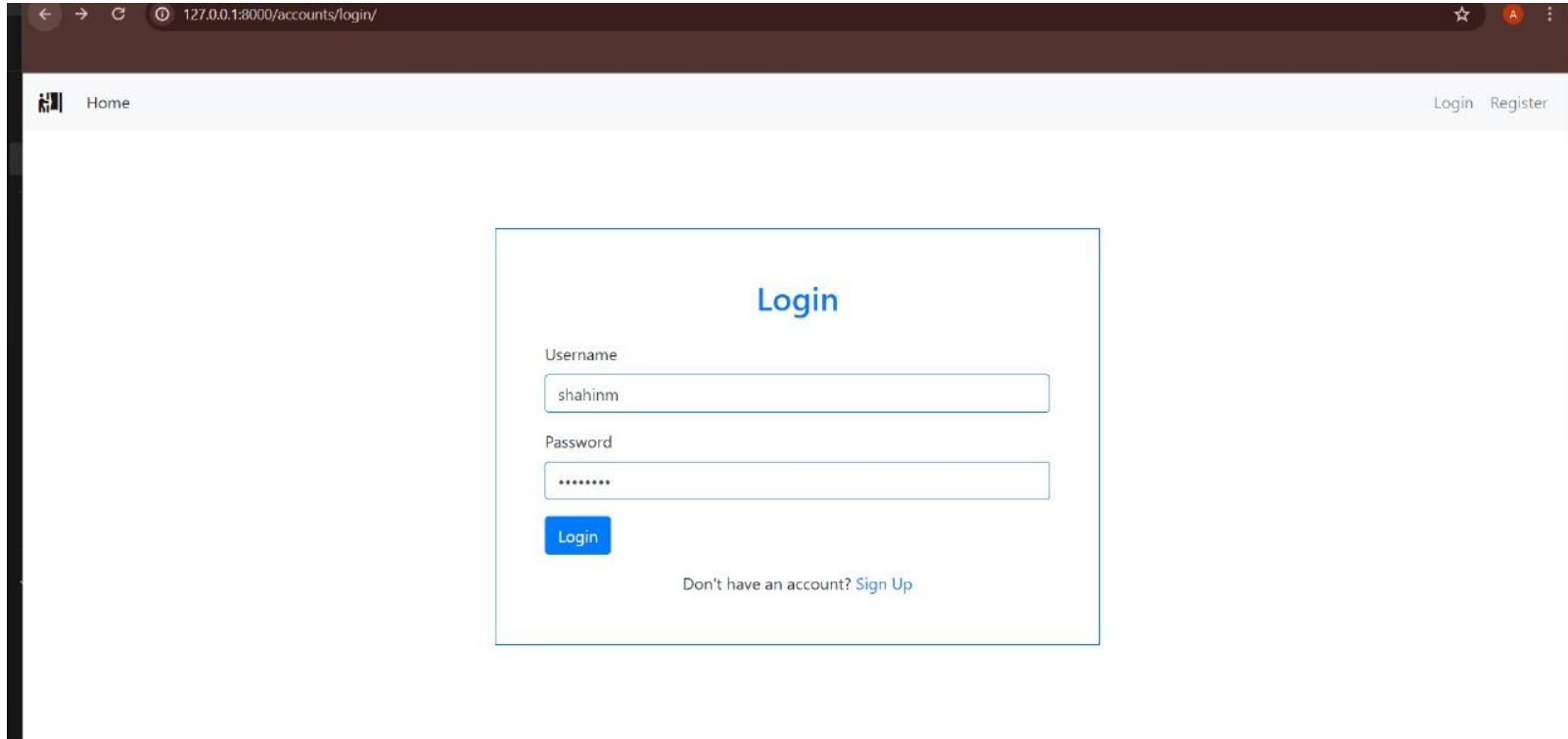
Result for: what is your favourite food

Total: 1 votes

briyani-100%	
briyani	1
parota	0

[Back To Polls](#)

## Admin Login Page



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/accounts/login/". The page has a dark blue header with a "Home" link and a "Login" link. The main content area is white and contains a "Login" form. The form has a title "Login" in blue. It includes a "Username" field with the text "shahinm" and a "Password" field with masked characters "\*\*\*\*\*". Below the password field is a blue "Login" button. At the bottom of the form, there is a link "Don't have an account? Sign Up".

127.0.0.1:8000/accounts/login/

Home Login Register

### Login

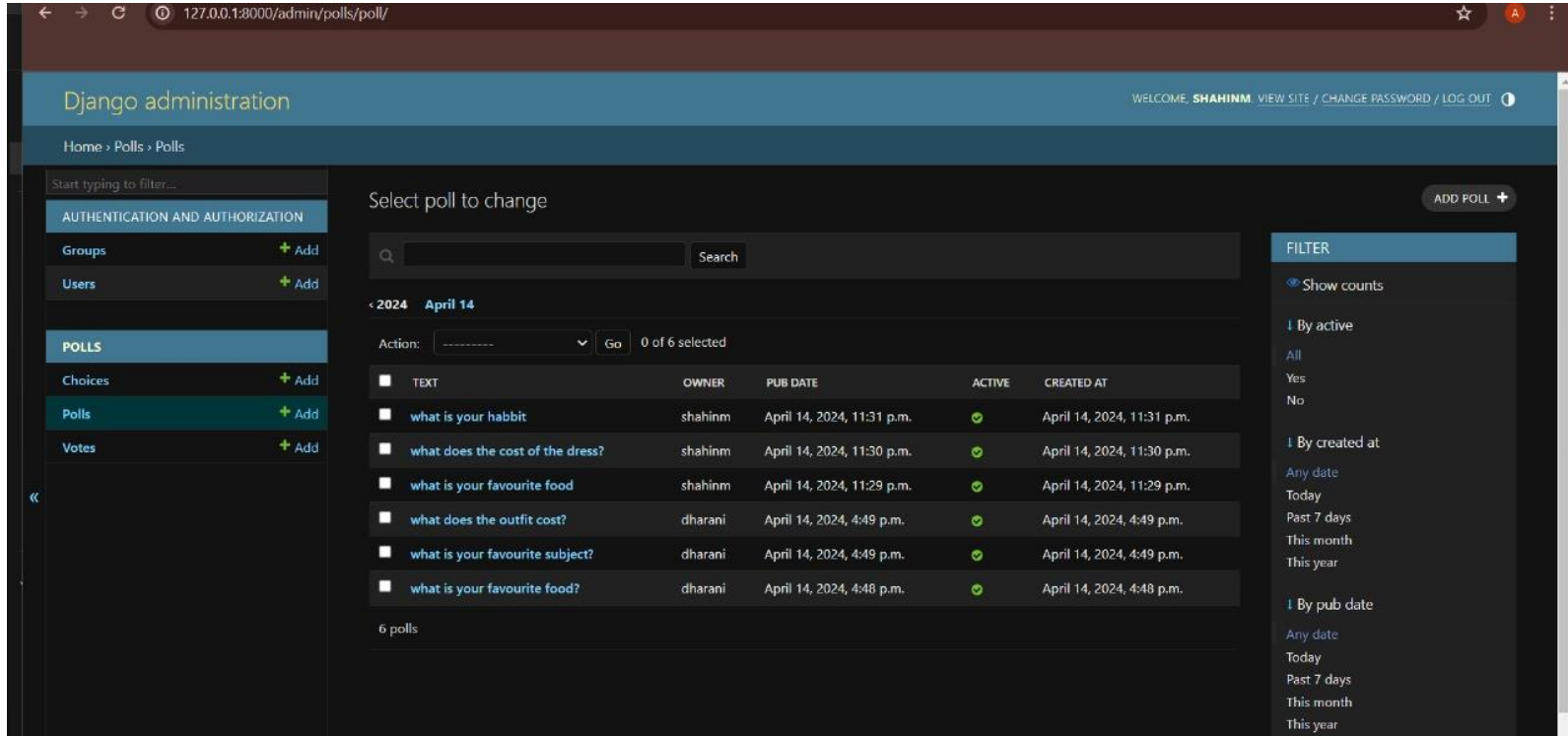
Username

Password

Login

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

## Admin Home 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/poll/. The page title is 'Django administration'. The user is logged in as 'SHAHINM' and can view the site, change their password, or log out.

The left sidebar contains the following navigation links:

- Home > Polls > Polls
- 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 includes a search bar and a filter section on the right.

The table displays the following data:

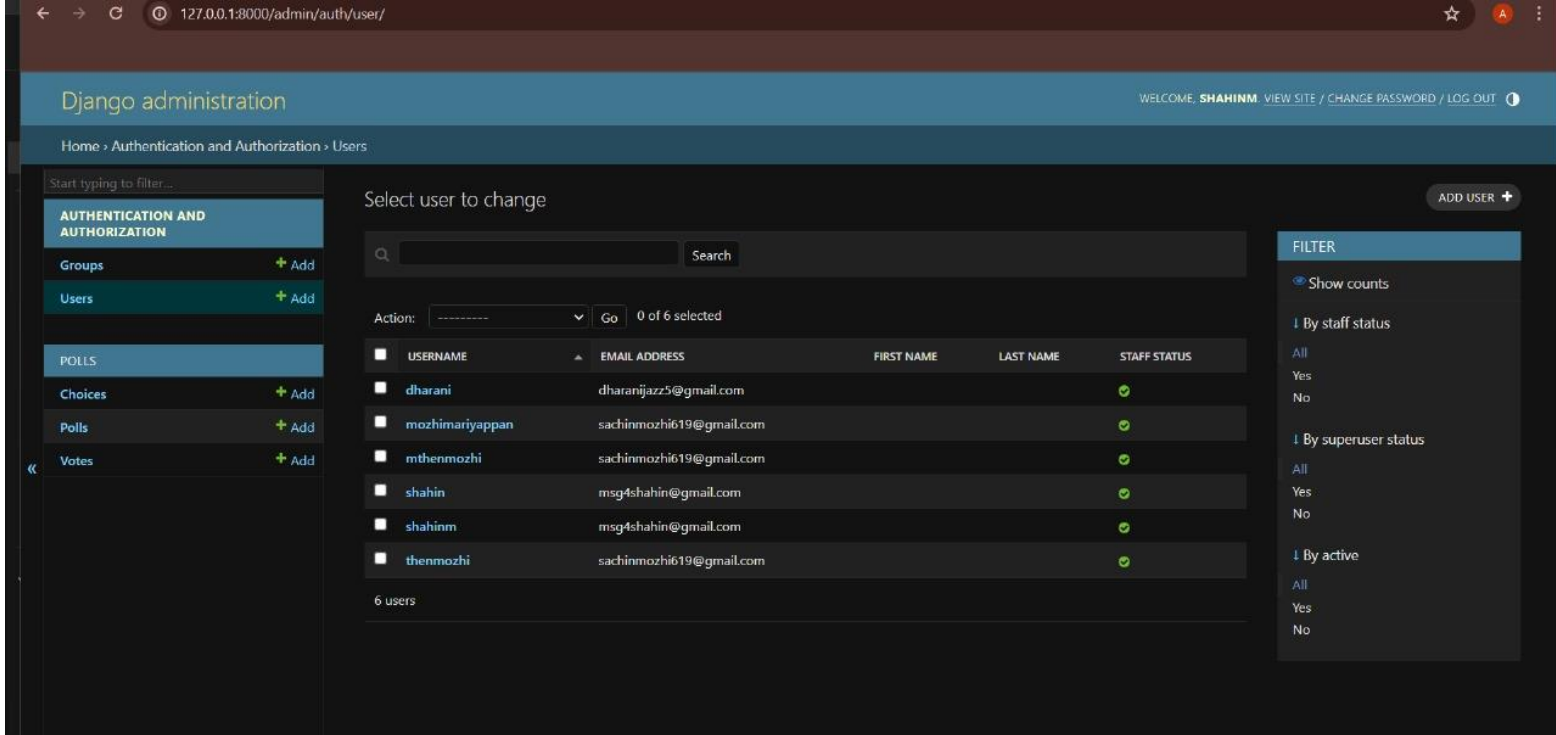
TEXT	OWNER	PUB DATE	ACTIVE	CREATED AT
<input type="checkbox"/> what is your habbit	shahinm	April 14, 2024, 11:31 p.m.	<input checked="" type="checkbox"/>	April 14, 2024, 11:31 p.m.
<input type="checkbox"/> what does the cost of the dress?	shahinm	April 14, 2024, 11:30 p.m.	<input checked="" type="checkbox"/>	April 14, 2024, 11:30 p.m.
<input type="checkbox"/> what is your favourite food	shahinm	April 14, 2024, 11:29 p.m.	<input checked="" type="checkbox"/>	April 14, 2024, 11:29 p.m.
<input type="checkbox"/> what does the outfit cost?	dharani	April 14, 2024, 4:49 p.m.	<input checked="" type="checkbox"/>	April 14, 2024, 4:49 p.m.
<input type="checkbox"/> what is your favourite subject?	dharani	April 14, 2024, 4:49 p.m.	<input checked="" type="checkbox"/>	April 14, 2024, 4:49 p.m.
<input type="checkbox"/> what is your favourite food?	dharani	April 14, 2024, 4:48 p.m.	<input checked="" type="checkbox"/>	April 14, 2024, 4:48 p.m.

6 polls

The right sidebar contains the following filter options:

- 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

## Authentication and Authorization Page



The screenshot shows the Django administration interface for the Authentication and Authorization section, specifically the Users page. The interface is dark-themed and includes a sidebar with navigation links, a main content area with a search bar and a table of users, and a right sidebar with filter options.

**Navigation Sidebar:**

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

**Main Content Area:**

Home > Authentication and Authorization > Users

Select user to change

Search

Action: ----- Go 0 of 6 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	dharani	dharanijazz5@gmail.com			✓
<input type="checkbox"/>	mozhimariyappan	sachinmozhi619@gmail.com			✓
<input type="checkbox"/>	mthenmozhi	sachinmozhi619@gmail.com			✓
<input type="checkbox"/>	shahin	msg4shahin@gmail.com			✓
<input type="checkbox"/>	shahinm	msg4shahin@gmail.com			✓
<input type="checkbox"/>	thenmozhi	sachinmozhi619@gmail.com			✓

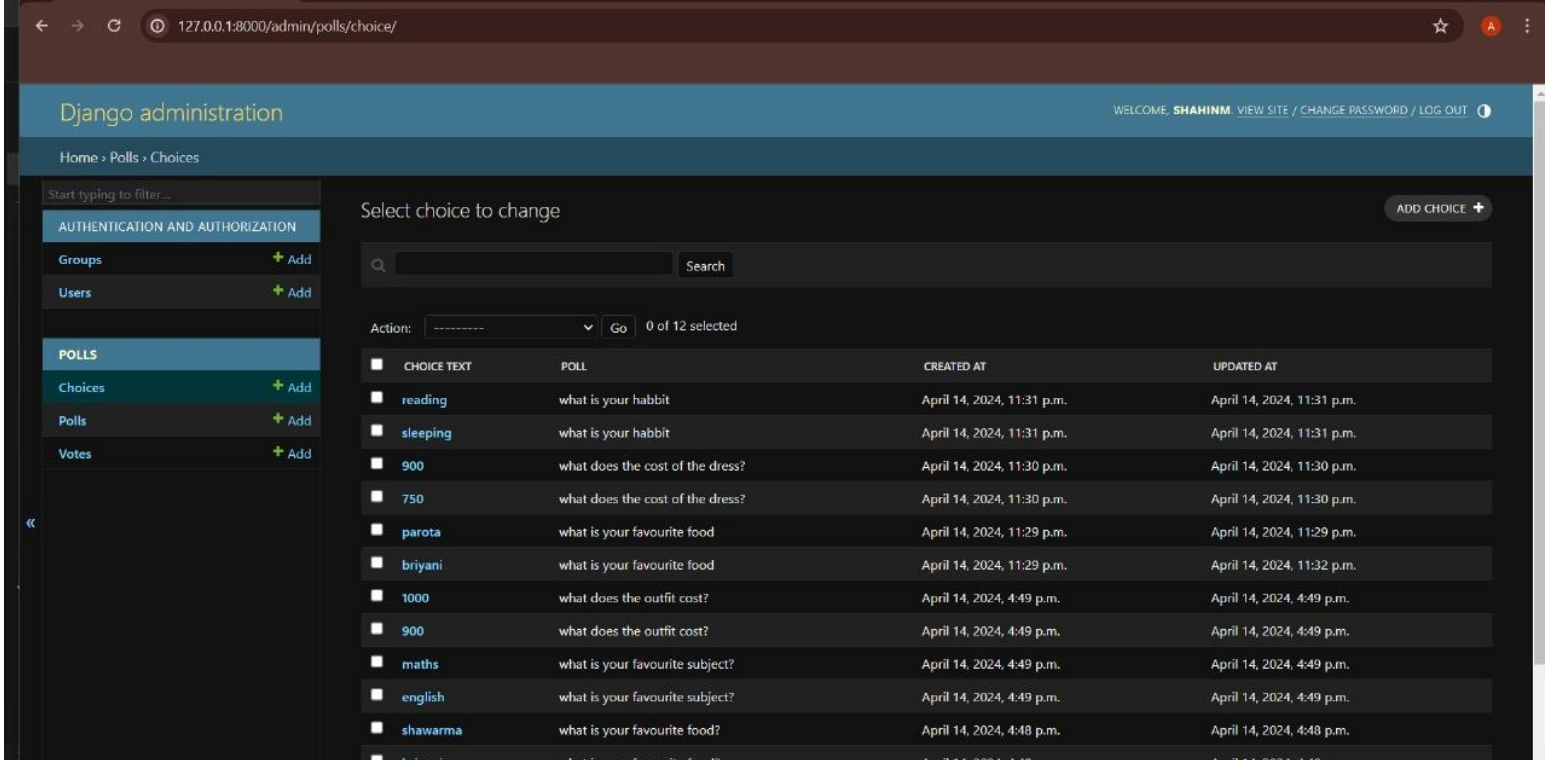
6 users

**Filter Sidebar:**

**FILTER**

- 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 the 'polls' app. The left sidebar contains a navigation menu with 'AUTHENTICATION AND AUTHORIZATION' (Groups, Users) and 'POLLS' (Choices, Polls, Votes). The main content area is titled 'Select choice to change' and features a table of existing choices. A search bar and an 'ADD CHOICE +' button are at the top right. The table has columns for selection, choice text, poll, created at, and updated at. The data shows multiple choices for different polls, such as 'reading' and 'sleeping' for 'what is your habbit', and '900', '750', and 'parota' for 'what does the cost of the dress?'.

Django administration

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

Home > Polls > Choices

Start typing to filter...

**AUTHENTICATION AND AUTHORIZATION**

- Groups + Add
- Users + Add

**POLLS**

- Choices + Add
- Polls + Add
- Votes + Add

Select choice to change

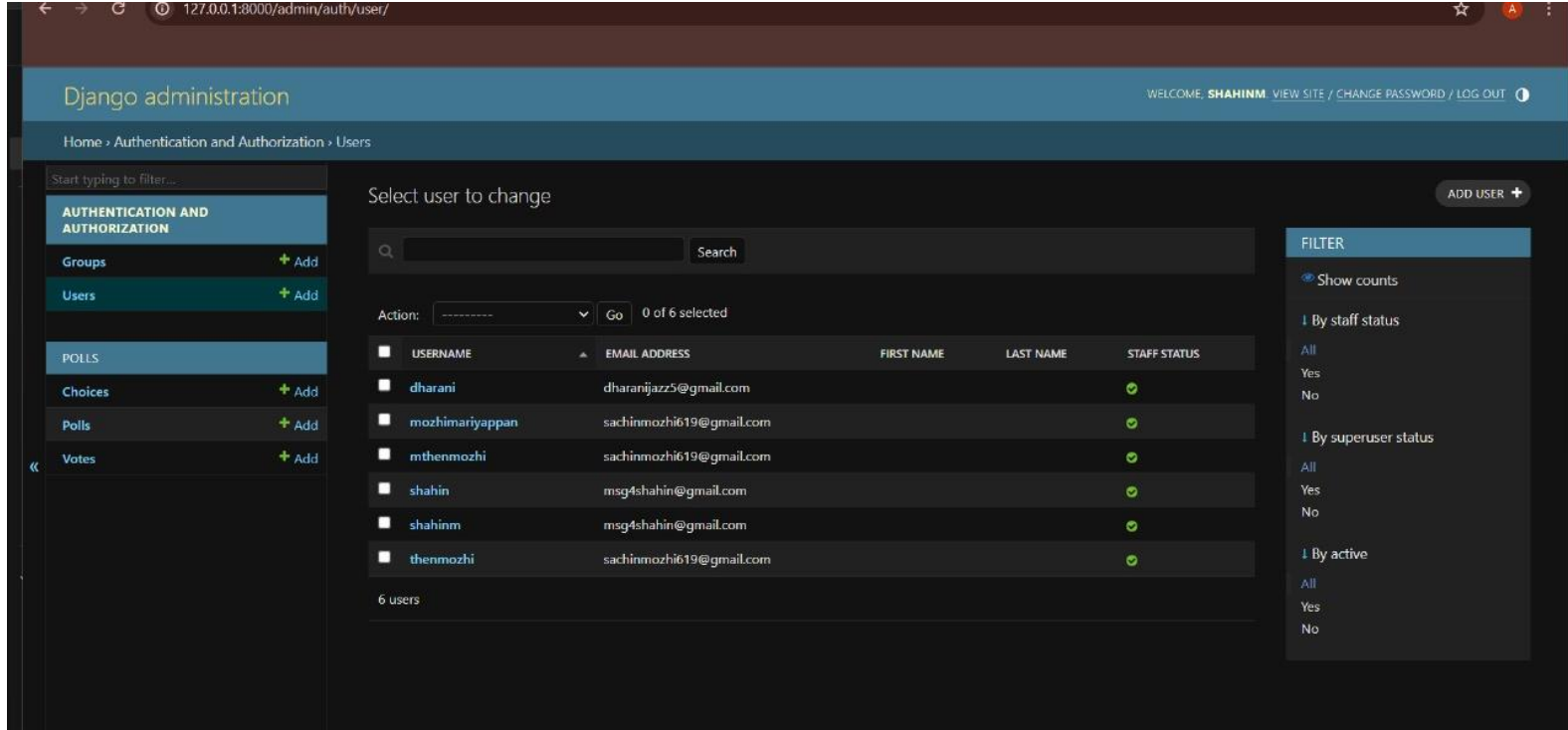
ADD CHOICE +

Search

Action: ----- Go 0 of 12 selected

<input type="checkbox"/>	CHOICE TEXT	POLL	CREATED AT	UPDATED AT
<input type="checkbox"/>	reading	what is your habbit	April 14, 2024, 11:31 p.m.	April 14, 2024, 11:31 p.m.
<input type="checkbox"/>	sleeping	what is your habbit	April 14, 2024, 11:31 p.m.	April 14, 2024, 11:31 p.m.
<input type="checkbox"/>	900	what does the cost of the dress?	April 14, 2024, 11:30 p.m.	April 14, 2024, 11:30 p.m.
<input type="checkbox"/>	750	what does the cost of the dress?	April 14, 2024, 11:30 p.m.	April 14, 2024, 11:30 p.m.
<input type="checkbox"/>	parota	what is your favourite food	April 14, 2024, 11:29 p.m.	April 14, 2024, 11:29 p.m.
<input type="checkbox"/>	briyani	what is your favourite food	April 14, 2024, 11:29 p.m.	April 14, 2024, 11:32 p.m.
<input type="checkbox"/>	1000	what does the outfit cost?	April 14, 2024, 4:49 p.m.	April 14, 2024, 4:49 p.m.
<input type="checkbox"/>	900	what does the outfit cost?	April 14, 2024, 4:49 p.m.	April 14, 2024, 4:49 p.m.
<input type="checkbox"/>	maths	what is your favourite subject?	April 14, 2024, 4:49 p.m.	April 14, 2024, 4:49 p.m.
<input type="checkbox"/>	english	what is your favourite subject?	April 14, 2024, 4:49 p.m.	April 14, 2024, 4:49 p.m.
<input type="checkbox"/>	shawarma	what is your favourite food?	April 14, 2024, 4:48 p.m.	April 14, 2024, 4:48 p.m.
<input type="checkbox"/>	biryani	what is your favourite food?	April 14, 2024, 4:49 p.m.	April 14, 2024, 4:49 p.m.

## Voting Details Page



The screenshot shows the Django administration interface for the 'Next Gen Employability Program'. The page is titled 'Django administration' and displays the 'Users' page under 'Authentication and Authorization'. The left sidebar contains a navigation menu with 'AUTHENTICATION AND AUTHORIZATION' (Groups, Users), 'POLLS' (Choices, Polls, Votes), and a search bar. The main content area is titled 'Select user to change' and features a search bar, an action dropdown, and a table of users. The table has columns for USERNAME, EMAIL ADDRESS, FIRST NAME, LAST NAME, and STAFF STATUS. There are 6 users listed, all with a 'Yes' status. A right sidebar contains a 'FILTER' section with options to show counts and filter by staff status, superuser status, and active status.

127.0.0.1:8000/admin/auth/user/

Django administration

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

Home > Authentication and Authorization > Users

Start typing to filter...

**AUTHENTICATION AND AUTHORIZATION**

- Groups + Add
- Users + Add

**POLLS**

- Choices + Add
- Polls + Add
- Votes + Add

«

Select user to change

ADD USER +

Search

Action: ----- Go 0 of 6 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	dharani	dharanijazz5@gmail.com			Yes
<input type="checkbox"/>	mozhimariyappan	sachinmozhi619@gmail.com			Yes
<input type="checkbox"/>	mthenmozhi	sachinmozhi619@gmail.com			Yes
<input type="checkbox"/>	shahin	msg4shahin@gmail.com			Yes
<input type="checkbox"/>	shahinm	msg4shahin@gmail.com			Yes
<input type="checkbox"/>	thenmozhi	sachinmozhi619@gmail.com			Yes

6 users

**FILTER**

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

## 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!**