



NEXT GEN EMPLOYABILITY PROGRAM

| Creating a future-ready workforce

Student Name :Noor alisha S
Student ID :au820621104060

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Voting Application using Django Framework-Noor Alisha S(4060,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:

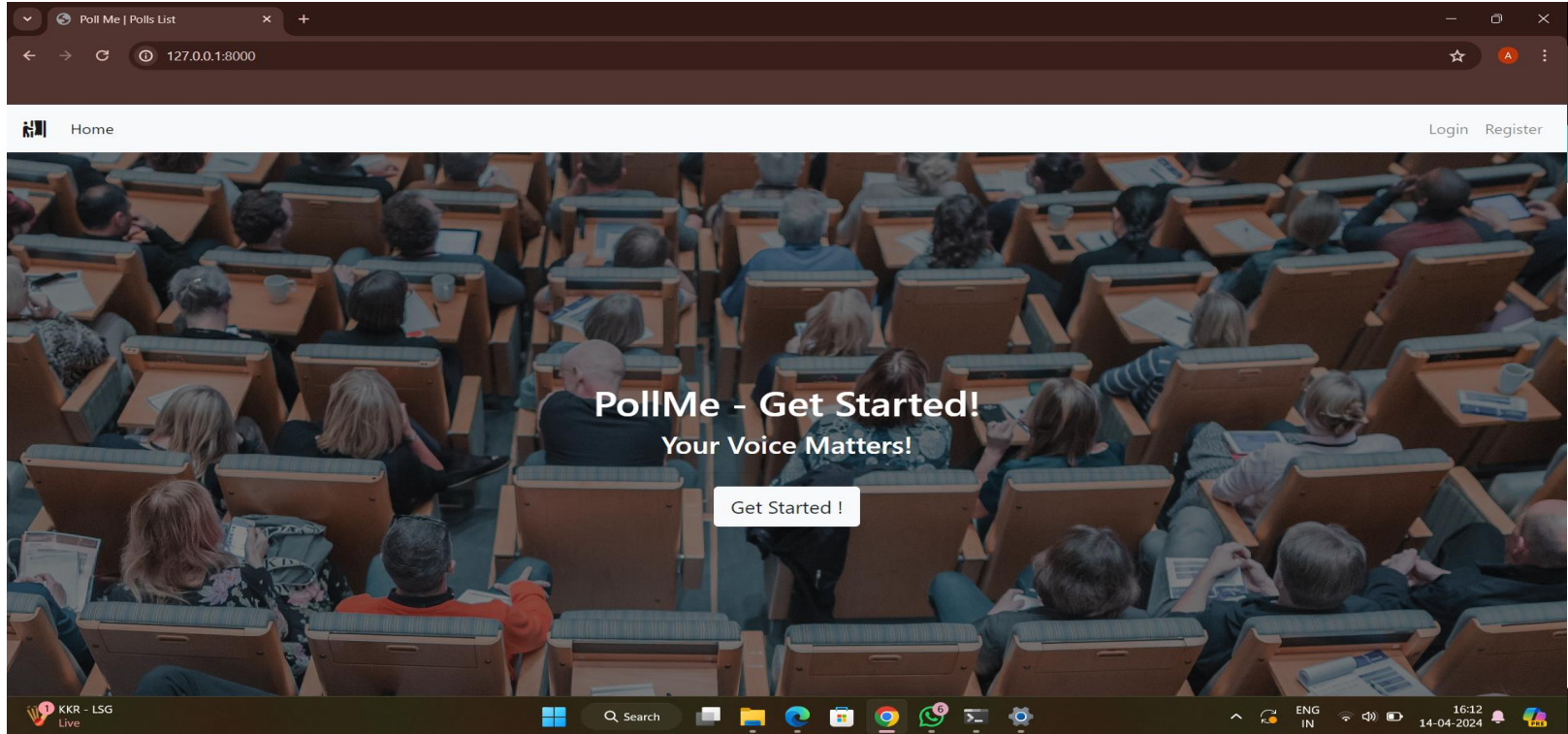
- 1.Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2.Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3.Creating User Authentication:** Implement user authentication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

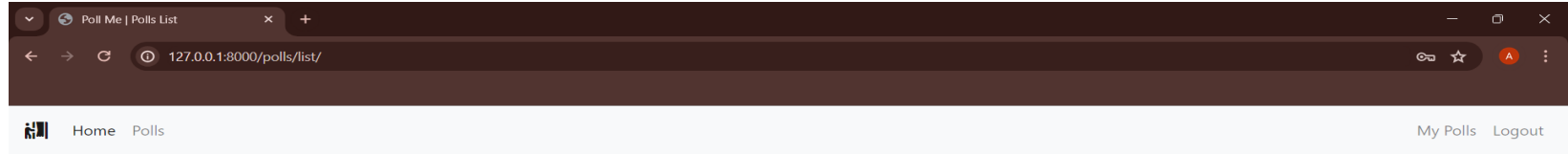
The proposed solution for a voting application using the Django framework is to create a secure and user-friendly online voting platform. The application will allow users to register, vote, and view real-time results. To build the application, the Django framework will be used as the foundation due to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

Poll Me | Polls List

127.0.0.1:8000/polls/add/

☆ A ⋮

 Home

 Polls

My Polls

Logout

Create new poll

Text:

what is your favourite food?

Choice 1:

biryani

Choice 2:

shawarma

Add Poll


Back

Voting Details Page

Poll Me | Polls List

127.0.0.1:8000/polls/list/

☆ A ⋮

 Home Polls

My Polls Logout

Welcome to polls List!

Poll Updated successfully. ✕

👤 Name

🕒 Date

🗳️ Vote









Add +

Search

🔍

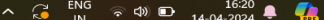
what is your favourite food?	<div><div></div><div></div></div>
what is your favourite subject?	<div><div></div><div></div></div>
what does the outfit cost?	<div><div></div><div></div></div>

38°C
Hot weather

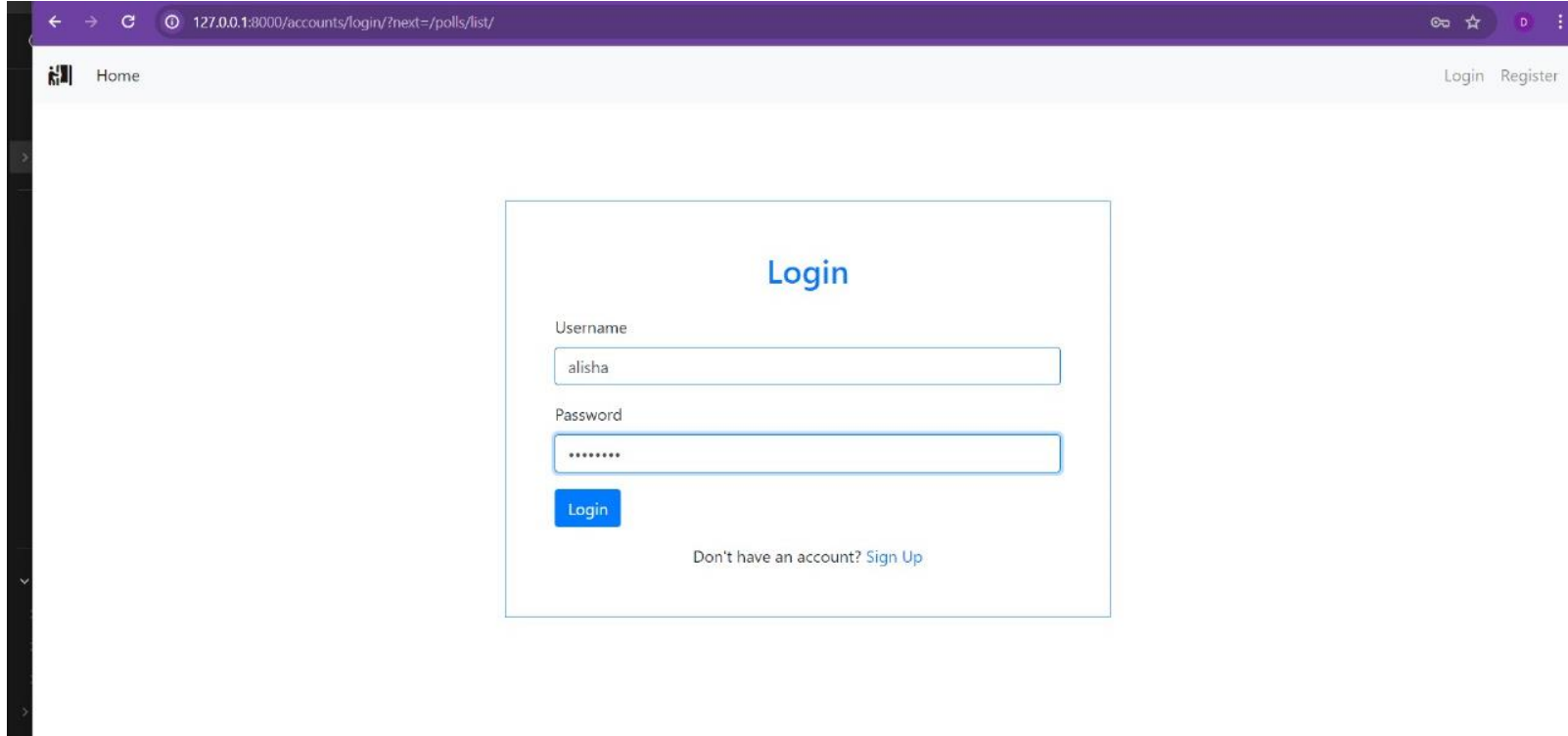
 Search       

ENG
IN

16:20
14-04-2024



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 dark purple header with navigation links "Home", "Login", and "Register". The main content area features a white box with the title "Login" in blue. Below the title are two input fields: "Username" with the value "alisha" and "Password" with masked characters "*****". A blue "Login" button is positioned below the password field. At the bottom of the box, there is a link: "Don't have an account? [Sign Up](#)".

127.0.0.1:8000/accounts/login/?next=/polls/list/

Home Login Register

Login

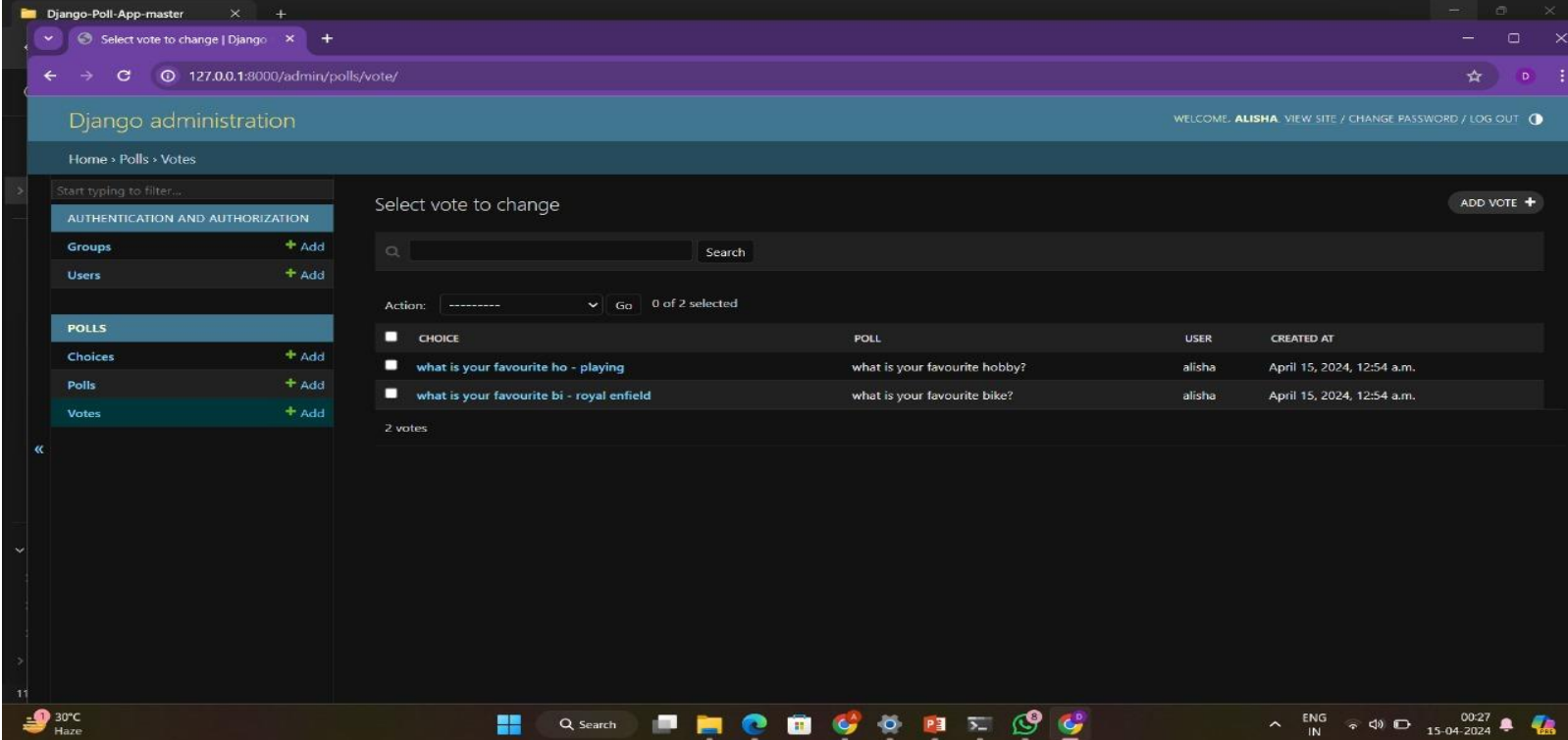
Username

Password

Login

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

Admin Home Page



The screenshot shows the Django administration interface for a Polls application. The browser window is titled "Django-Poll-App-master" and the address bar shows "127.0.0.1:8000/admin/polls/vote/". The page header includes "Django administration" and a welcome message for "ALISHA" with links to "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 vote to change" and features a search bar and an "ADD VOTE +" button. Below the search bar, there is an "Action:" dropdown menu and a "Go" button, indicating that 0 of 2 items are selected.

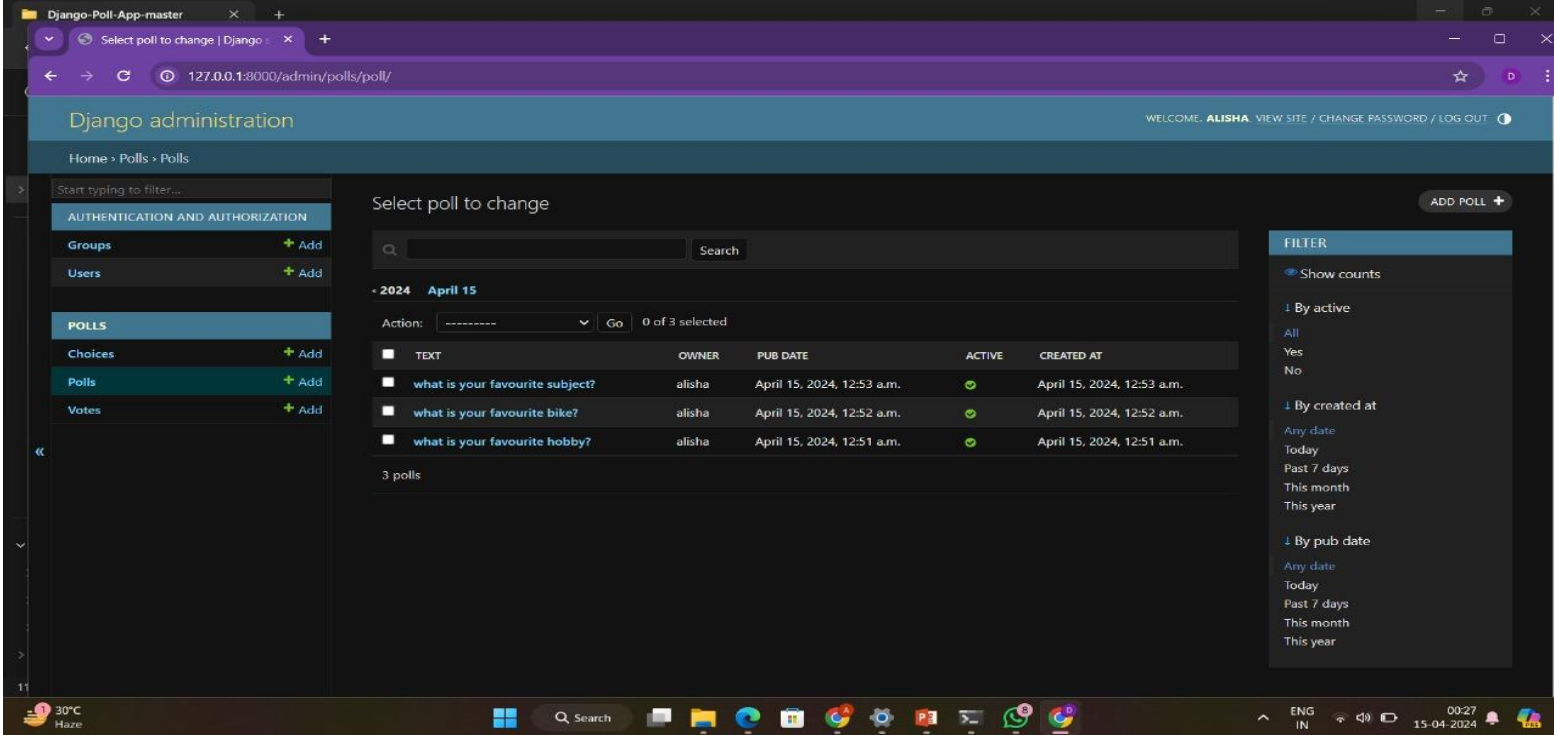
The table displays the following data:

CHOICE	POLL	USER	CREATED AT
<input type="checkbox"/> what is your favourite ho - playing	what is your favourite hobby?	alisha	April 15, 2024, 12:54 a.m.
<input type="checkbox"/> what is your favourite bi - royal enfield	what is your favourite bike?	alisha	April 15, 2024, 12:54 a.m.

Below the table, it indicates "2 votes".

The Windows taskbar at the bottom shows the system clock as 00:27 on 15-04-2024, along with various application icons and a weather widget showing 30°C Haze.

Authentication and Authorization Page



The screenshot shows the Django administration interface for a Polls application. The browser address bar indicates the URL is `127.0.0.1:8000/admin/polls/poll/`. The page title is "Django-Poll-App-master". The Django administration header shows "Django administration" and a welcome message for "ALISHA" with links for "VIEW SITE / CHANGE PASSWORD / LOG OUT".

The left sidebar contains the following navigation items:

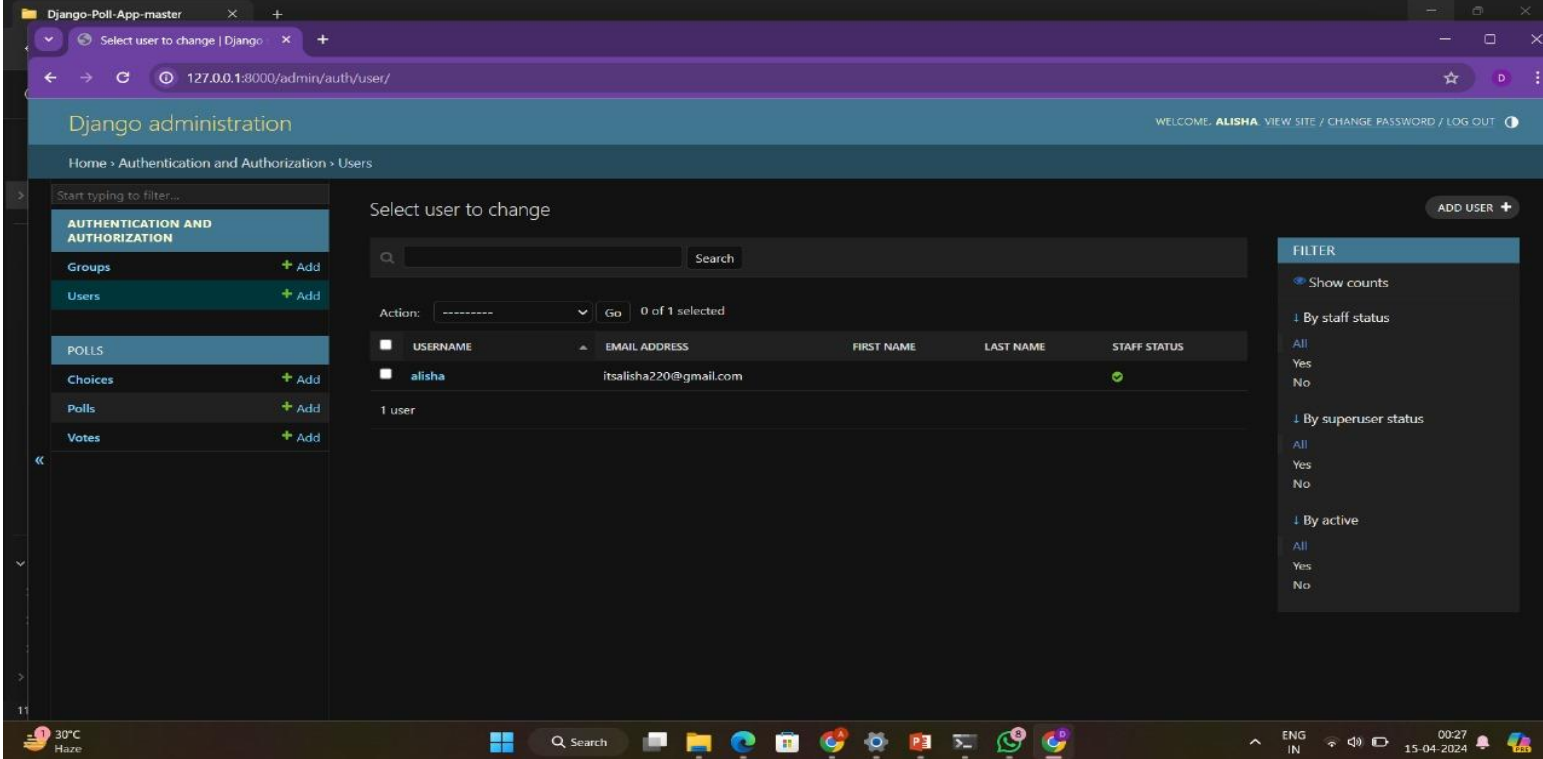
- Home > Polls > Polls
- Start typing to filter...
- AUTHENTICATION AND AUTHORIZATION
 - Groups + Add
 - Users + Add
- POLLS
 - Choices + Add
 - Polls + Add
 - Votes + Add

The main content area is titled "Select poll to change". It includes a search bar and a table of polls. The table has columns: TEXT, OWNER, PUB DATE, ACTIVE, and CREATED AT. There are 3 polls listed, all created on April 15, 2024.

TEXT	OWNER	PUB DATE	ACTIVE	CREATED AT
what is your favourite subject?	alisha	April 15, 2024, 12:53 a.m.	✓	April 15, 2024, 12:53 a.m.
what is your favourite bike?	alisha	April 15, 2024, 12:52 a.m.	✓	April 15, 2024, 12:52 a.m.
what is your favourite hobby?	alisha	April 15, 2024, 12:51 a.m.	✓	April 15, 2024, 12:51 a.m.

Below the table, it says "3 polls". On the right side, there is a "FILTER" sidebar with options to "Show counts", "By active", "By created at", and "By pub date".

Questions Adding Section Page



Django administration

WELCOME, ALISHA VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Authentication and Authorization > Users

Select user to change

ADD USER +

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups + Add
- Users + Add

POLLS

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

1 user

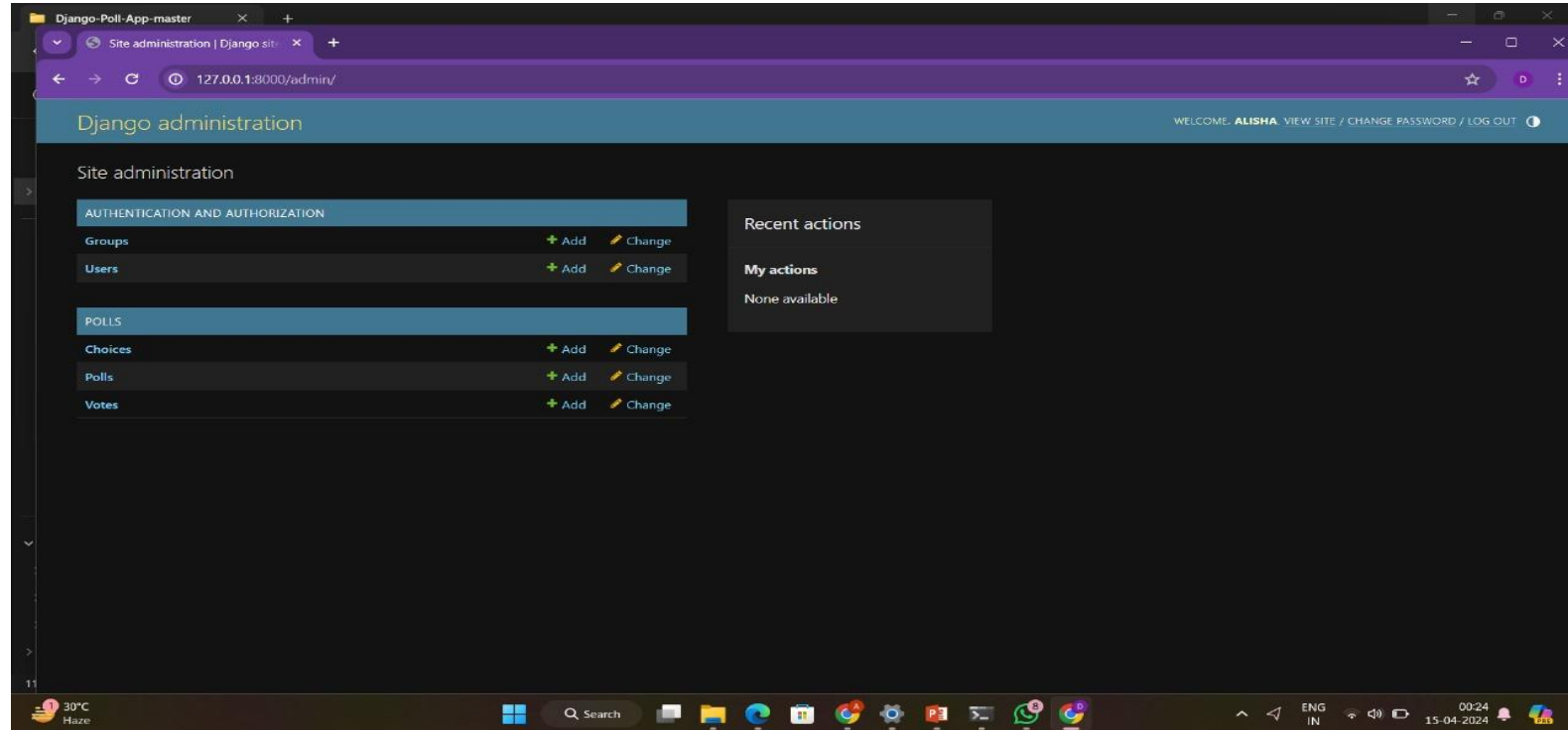
USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
alisha	itsalisha220@gmail.com			Yes

1 user

FILTER

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

Voting Details Page



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!