



# NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Student Name :Mohammed shafeek K  
Student ID :au820621104052

College Name

Arasu Engineering College

# CAPSTONE PROJECT SHOWCASE

## Project Title

Voting Application using Django Framework-Mohammed  
shafeek(820621104052,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:

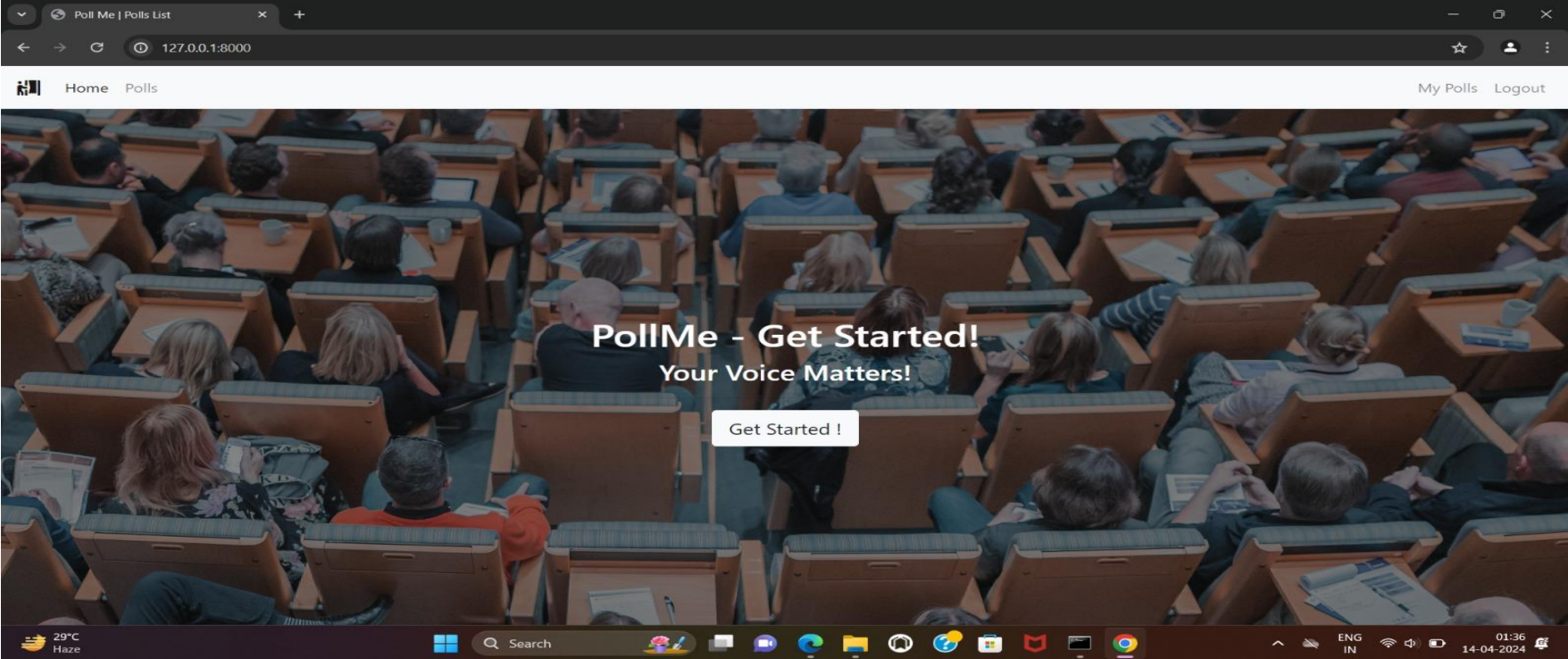
- 1.Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2.Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3.Creating User Authentication:** Implement user authentication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

127.0.0.1:8000

Home Polls

My Polls Logout

**PollMe - Get Started!**  
Your Voice Matters!

Get Started !

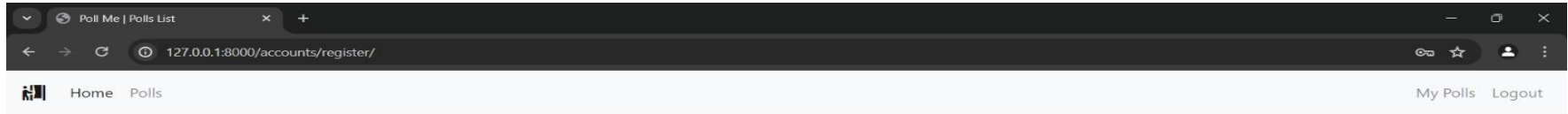
29°C Haze

Search

ENG IN

01:36  
14-04-2024

## Home login



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

Username:

Email:

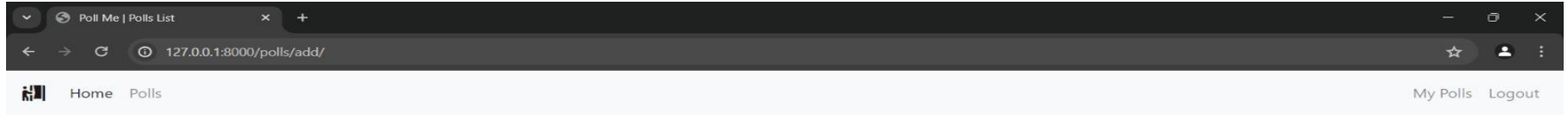
Password:

Confirm Password:

[Sign Up](#)



## create new poll



### Create new poll

Text:

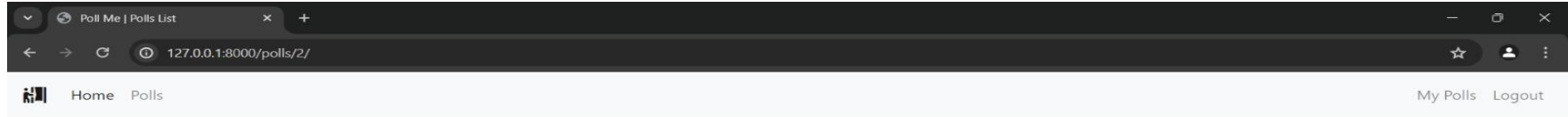
Choice 1:

Choice 2:

Add Poll

Back

## Voting Page



### Polls details page

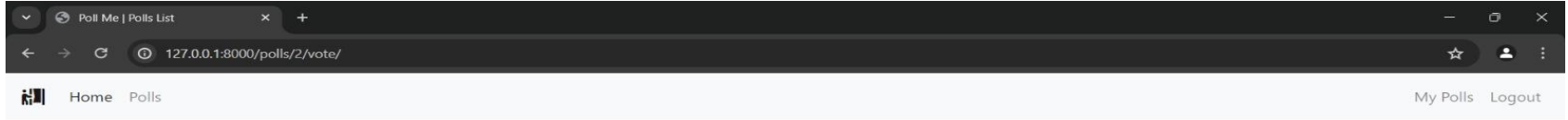
Which of the following HTML element is used for creating an unordered list?

- ☐ <ul>
- ☐ <ui>
- ☐ <li>

Vote

Cancel

## Voting Details Page



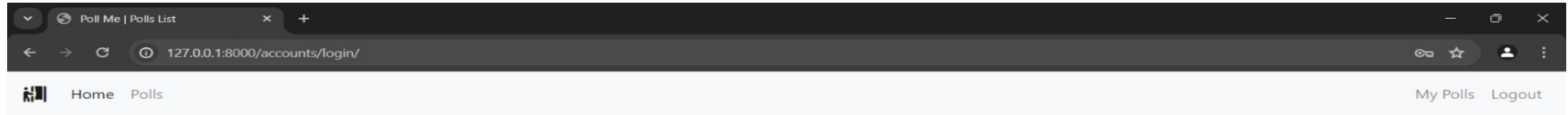
Result for: Which of the following HTML element is used for creating an unordered list?

Total: 1 votes

<ul> - 100%	
<ul>	1
<ui>	0
<li>	0

[Back To Polls](#)

## Admin Login Page



### Login

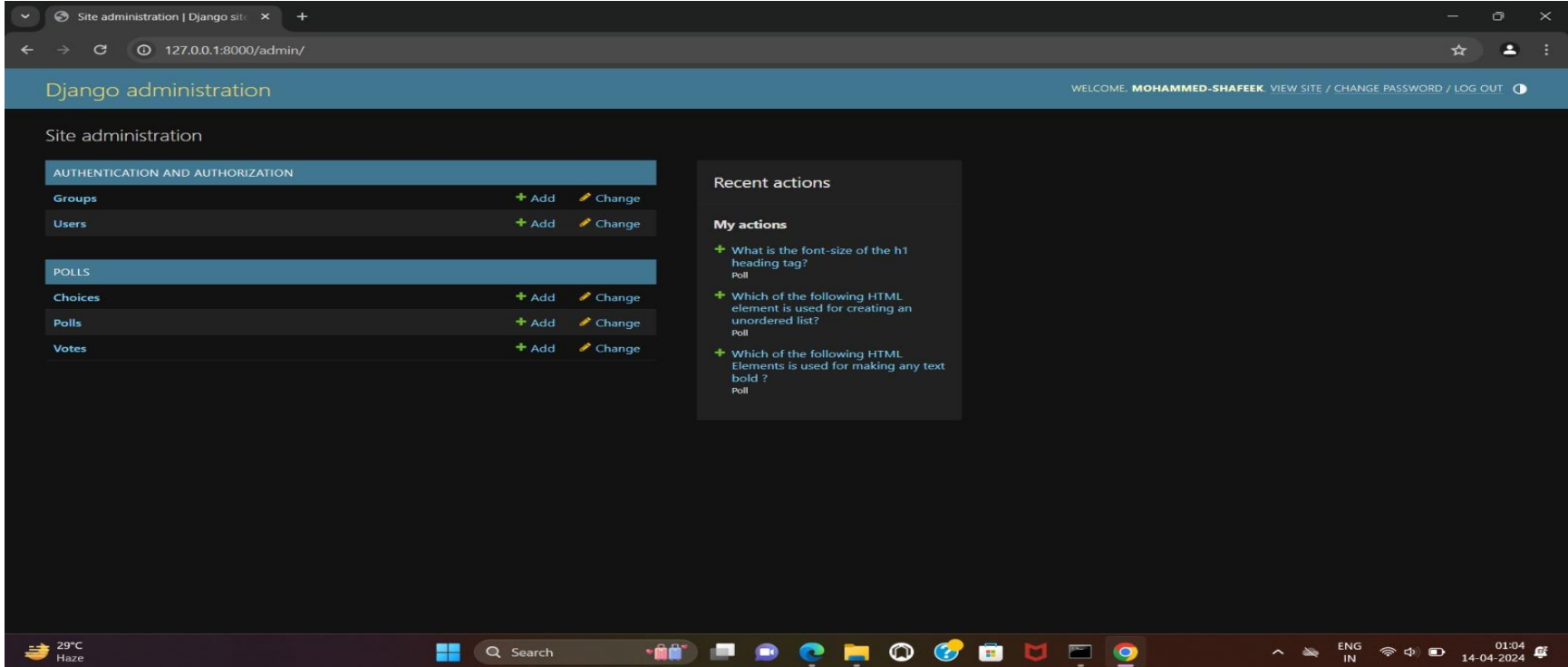
Username

Password

[Login](#)

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

## 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 "MOHAMMED-SHAFEK" 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" and "POLLS".

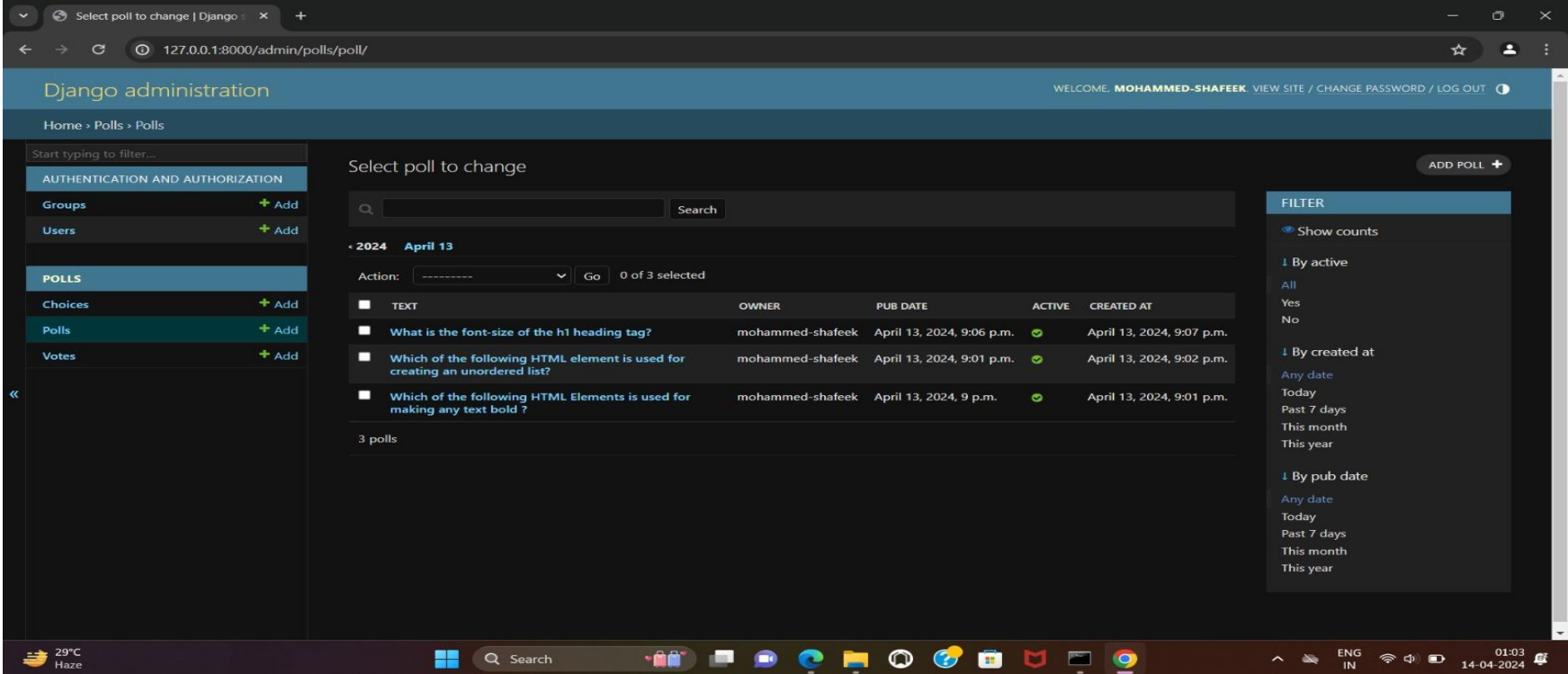
- AUTHENTICATION AND AUTHORIZATION**
  - Groups**: + Add, Change
  - Users**: + Add, Change
- POLLS**
  - Choices**: + Add, Change
  - Polls**: + Add, Change
  - Votes**: + Add, Change

The right column is titled "Recent actions" and contains a section "My actions" with three entries, each marked with a green plus icon and a "Poll" label:

- + What is the font-size of the h1 heading tag? Poll
- + Which of the following HTML element is used for creating an unordered list? Poll
- + Which of the following HTML Elements is used for making any text bold ? Poll

The bottom of the image shows a Windows taskbar with the date "14-04-2024" and time "01:04".

## Authentication and Authorization Page



The screenshot displays the Django administration interface for managing polls. The browser address bar shows the URL `127.0.0.1:8000/admin/polls/poll/`. The page title is "Django administration". The user is logged in as "MOHAMMED-SHAFEEL" 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 table of polls. The table has columns for ACTION, TEXT, OWNER, PUB DATE, ACTIVE, and CREATED AT. There are 3 polls listed.

ACTION	TEXT	OWNER	PUB DATE	ACTIVE	CREATED AT
<input type="checkbox"/>	What is the font-size of the h1 heading tag?	mohammed-shafeel	April 13, 2024, 9:06 p.m.	✓	April 13, 2024, 9:07 p.m.
<input type="checkbox"/>	Which of the following HTML element is used for creating an unordered list?	mohammed-shafeel	April 13, 2024, 9:01 p.m.	✓	April 13, 2024, 9:02 p.m.
<input type="checkbox"/>	Which of the following HTML Elements is used for making any text bold ?	mohammed-shafeel	April 13, 2024, 9 p.m.	✓	April 13, 2024, 9:01 p.m.

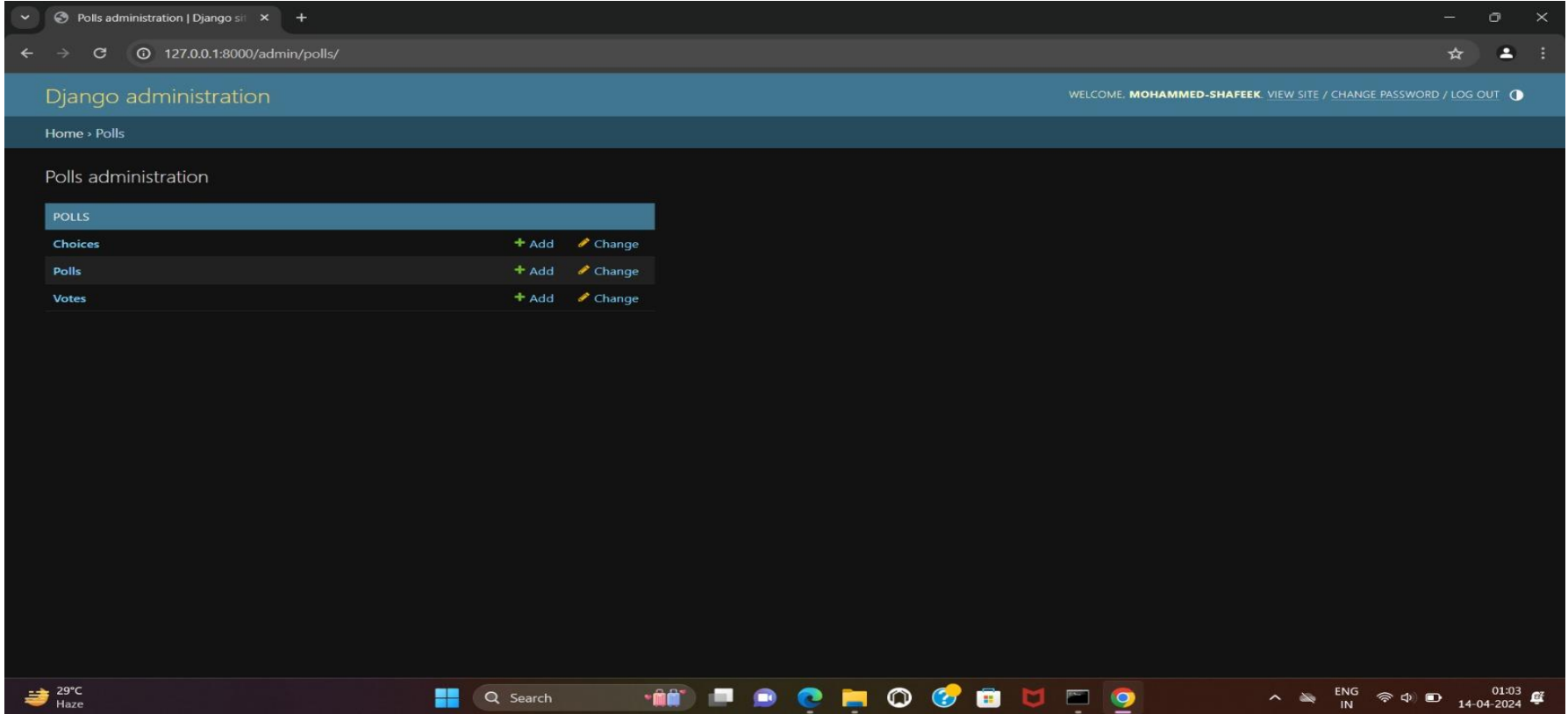
3 polls

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

The bottom of the screen shows the Windows taskbar with the date and time: 01:03 14-04-2024.

## Voting Details Page



The screenshot shows a web browser window displaying the Django administration interface for a Polls application. The browser's address bar shows the URL `127.0.0.1:8000/admin/polls/`. The page title is "Django administration". In the top right corner, a welcome message reads "WELCOME. **MOHAMMED-SHAFEER** [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)". Below the navigation bar, the breadcrumb "Home > Polls" is visible. The main content area is titled "Polls administration" and contains a table with three rows: "POLLS", "Choices", and "Votes". Each row has a "+ Add" button and a "Change" button with a pencil icon.

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>

## 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!**