



# NEXT GEN EMPLOYABILITY PROGRAM

| Creating a future-ready workforce

Student Name :Naveen R  
Student ID :au820621104056

College Name

Arasu Engineering College

# CAPSTONE PROJECT SHOWCASE

## Project Title

Voting Application using Django Framework-Naveen R(820621104056,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:

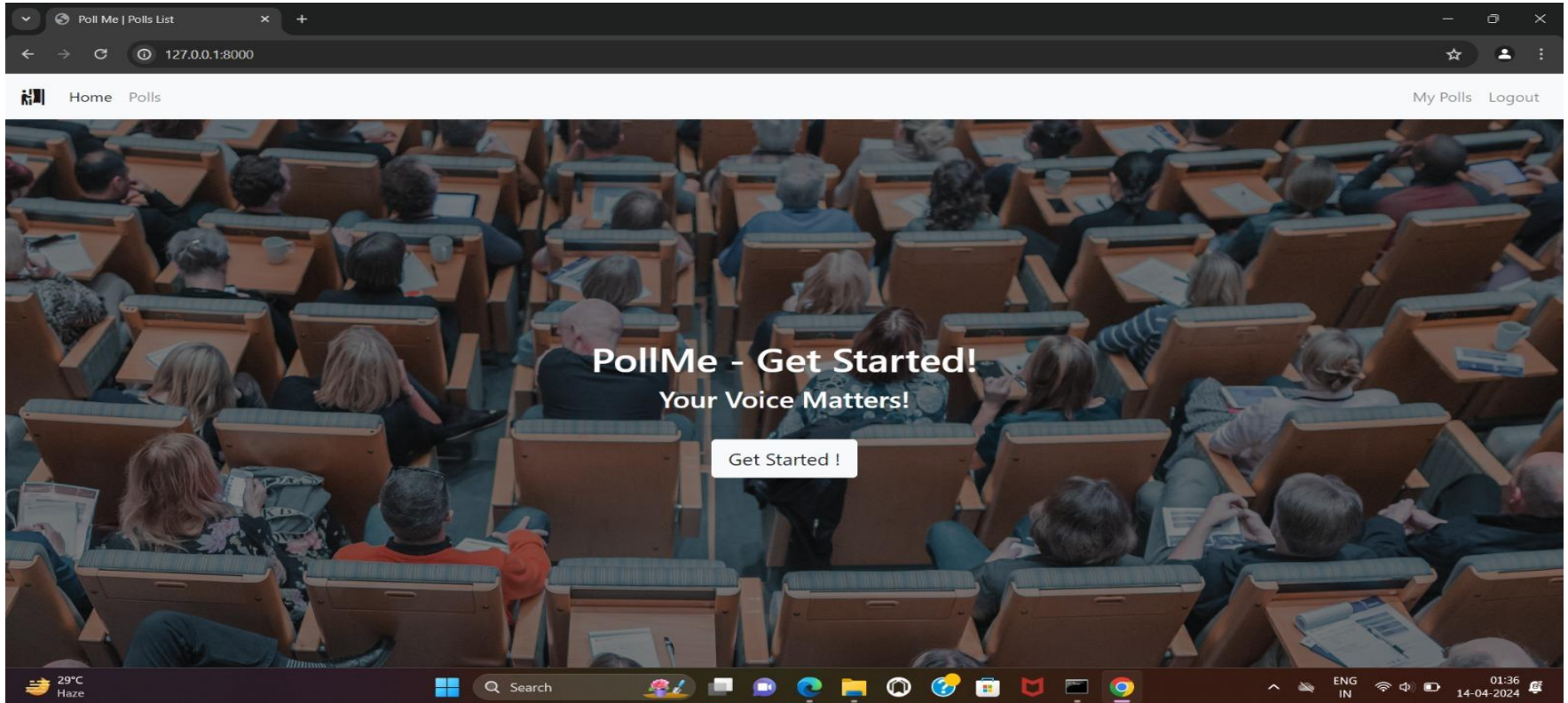
- 1.Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2.Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3.Creating User Authentication:** Implement user authentication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



The screenshot shows a web browser window displaying the PollMe website. The browser's address bar shows the URL 127.0.0.1:8000. The website has a dark blue header with the text "Poll Me | Polls List" and navigation links for "Home" and "Polls". The main content area features a large background image of an audience in a lecture hall. Overlaid on this image is the text "PollMe - Get Started! Your Voice Matters!" and a white button labeled "Get Started !". The bottom of the browser window shows a Windows taskbar with various application icons, a search bar, and system status information including the temperature (29°C), weather (Haze), and time (01:36 on 14-04-2024).

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:

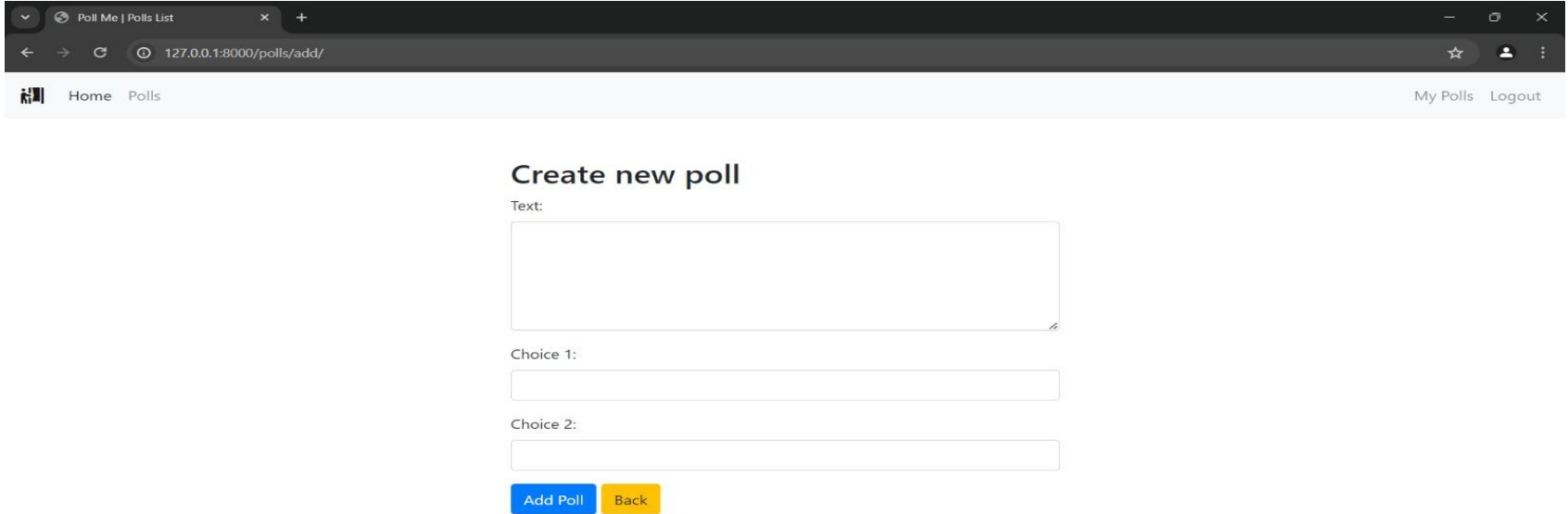
Password:

Confirm Password:

[Sign Up](#)



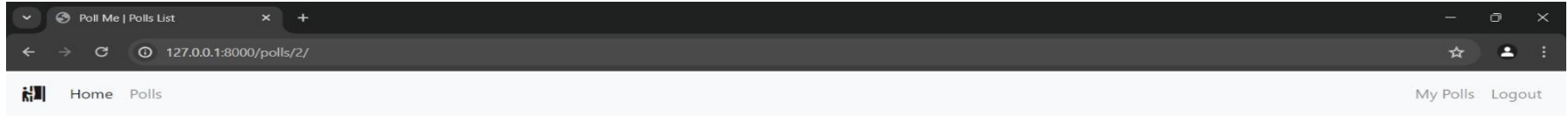
## Create new poll



The screenshot shows a web browser window with the title 'Poll Me | Polls List'. The address bar displays '127.0.0.1:8000/polls/add/'. The browser's navigation bar includes 'Home' and 'Polls' links, and a 'Logout' link. The main content area is titled 'Create new poll' and contains the following form elements:

- Text:** A large text input field.
- Choice 1:** A single-line text input field.
- Choice 2:** A single-line text input field.
- Buttons:** Two buttons at the bottom, 'Add Poll' (blue) and 'Back' (yellow).

## Voting Page



### Polls details page

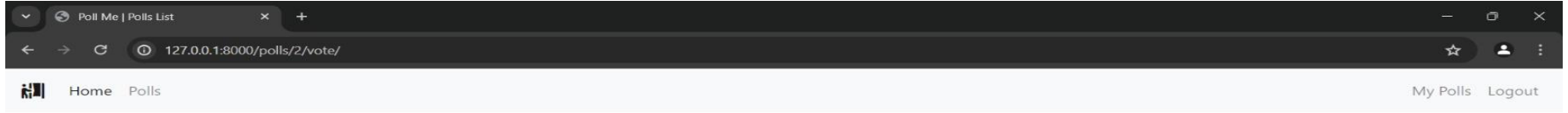
How is document type initialized in HTML5.?

- ☐ </DOCTYPE>
- ☐ <!DOCTYPE HTML>
- ☐ </DOCTYPE html!>

Vote

Cancel

## Voting Details Page



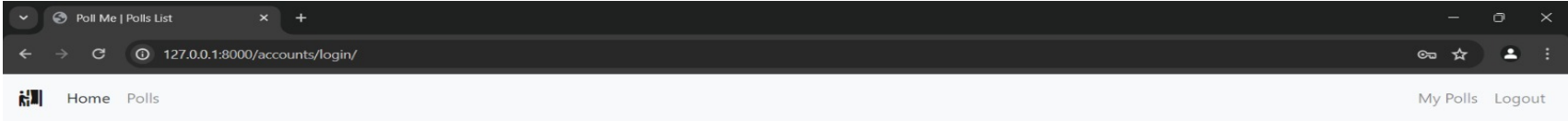
Result for: How is document type initialized in HTML5.?

Total: 1 votes

<!DOCTYPE HTML>--100%	
</DOCTYPE>	0
<!DOCTYPE HTML>	1
</DOCTYPE html>	0

[Back To Polls](#)

## Admin Login Page



Poll Me | Polls List

127.0.0.1:8000/accounts/login/

Home Polls

My Polls Logout

### Login

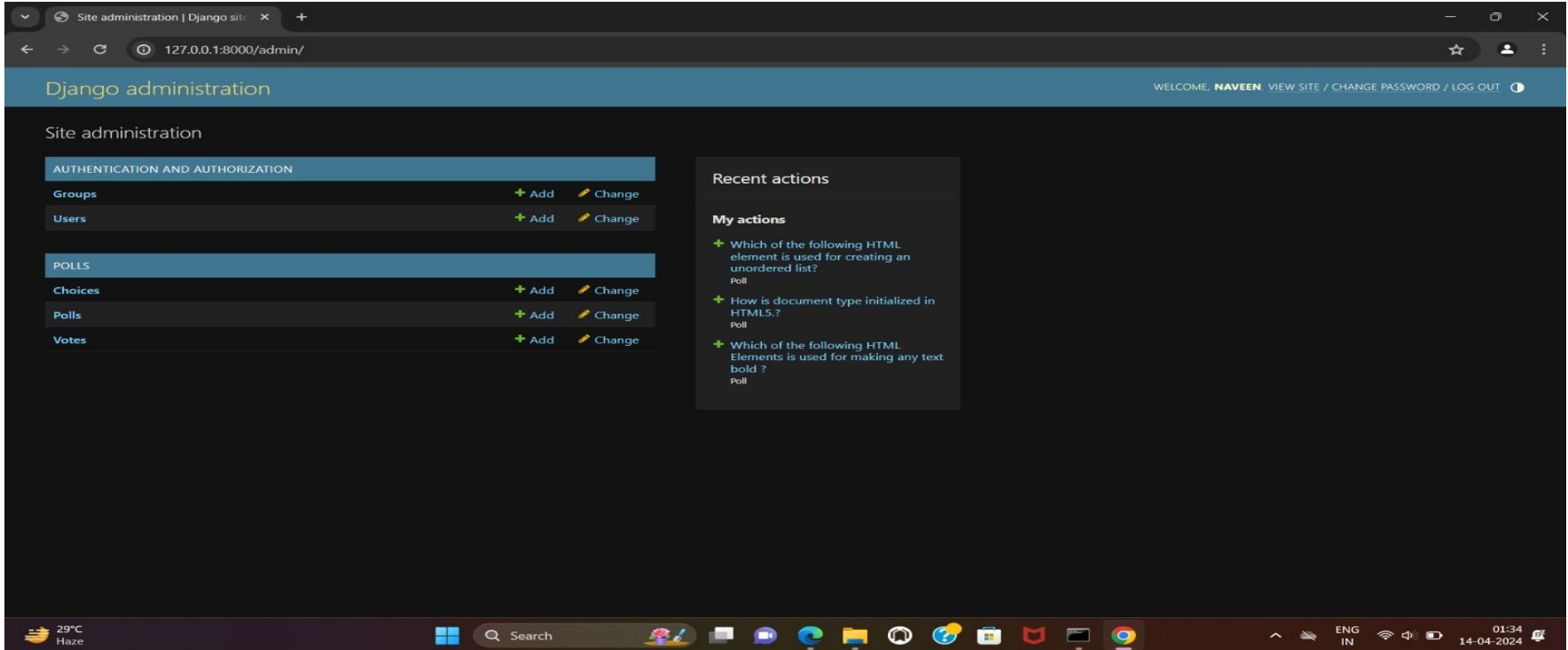
Username

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 "NAVEEN" 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".

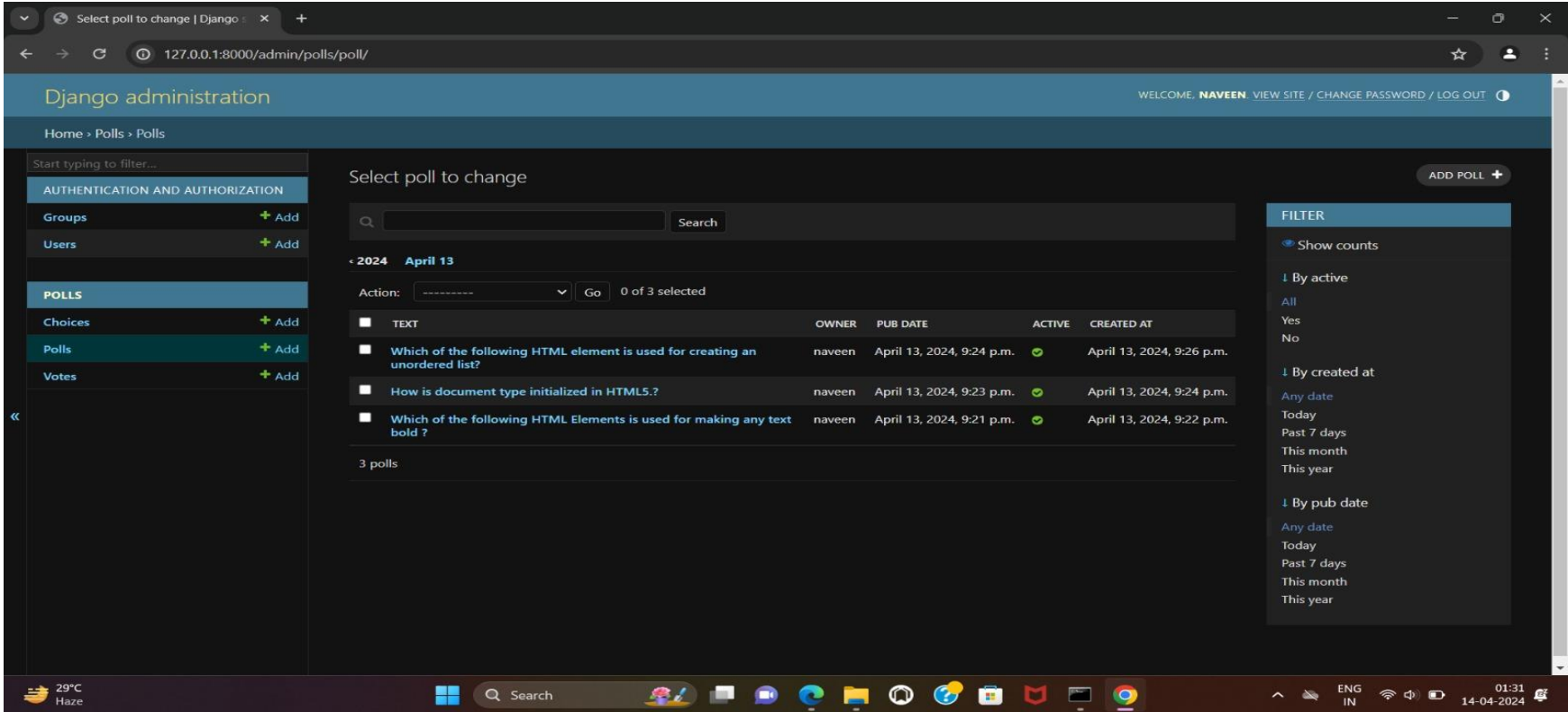
- AUTHENTICATION AND AUTHORIZATION**
  - Groups**: Includes links for "+ Add" and "Change".
  - Users**: Includes links for "+ Add" and "Change".
- POLLS**
  - Choices**: Includes links for "+ Add" and "Change".
  - Polls**: Includes links for "+ Add" and "Change".
  - Votes**: Includes links for "+ Add" and "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 labeled as a "Poll":

- Which of the following HTML element is used for creating an unordered list?
- How is document type initialized in HTML5.?
- Which of the following HTML Elements is used for making any text bold ?

The bottom of the image shows a Windows taskbar with the system clock at 01:34 on 14-04-2024, the language set to ENG IN, and various application icons.

## Authentication and Authorization Page



The screenshot displays the Django administration interface for a poll application. 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 "NAVEEN" and can view the site, change their password, or 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 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". There are 3 polls listed, all created on April 13, 2024.

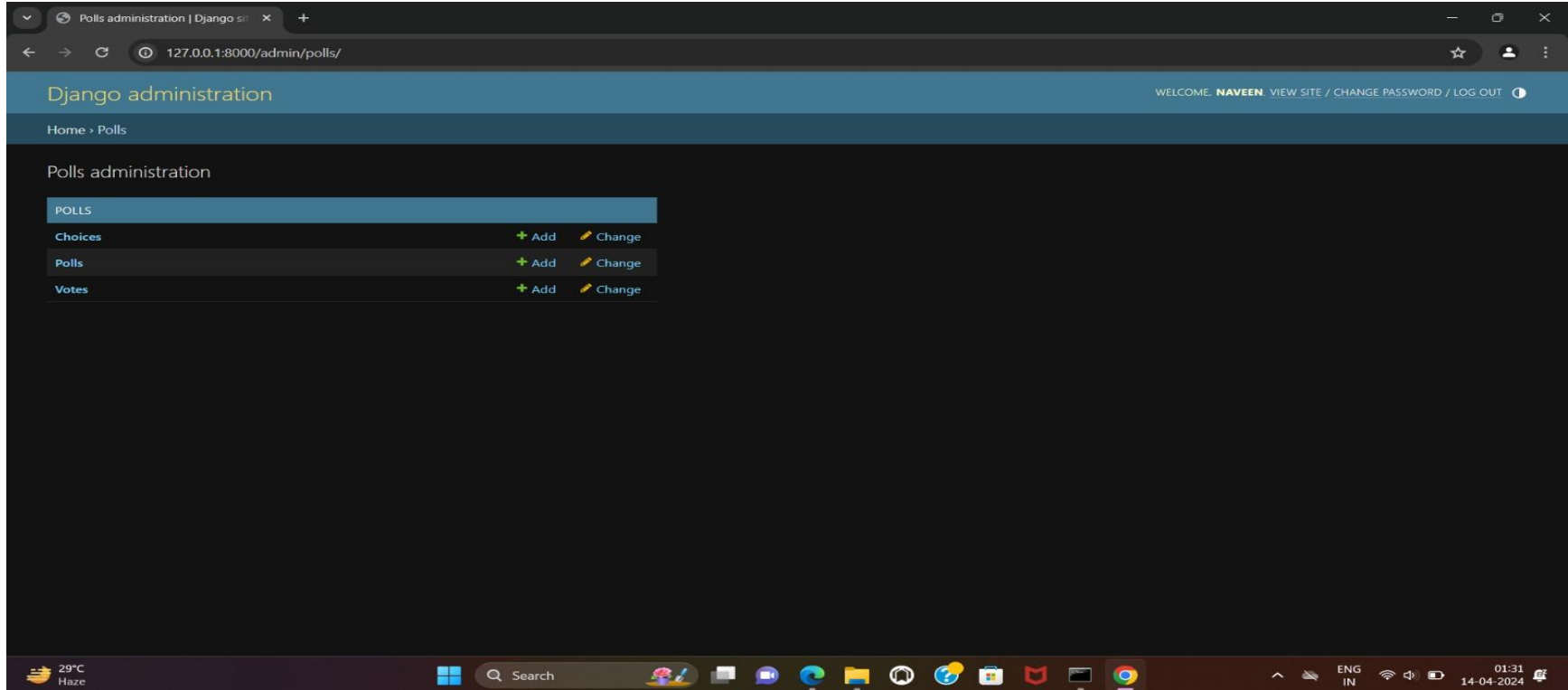
TEXT	OWNER	PUB DATE	ACTIVE	CREATED AT
<input type="checkbox"/> Which of the following HTML element is used for creating an unordered list?	naveen	April 13, 2024, 9:24 p.m.	✓	April 13, 2024, 9:26 p.m.
<input type="checkbox"/> How is document type initialized in HTML5.?	naveen	April 13, 2024, 9:23 p.m.	✓	April 13, 2024, 9:24 p.m.
<input type="checkbox"/> Which of the following HTML Elements is used for making any text bold ?	naveen	April 13, 2024, 9:21 p.m.	✓	April 13, 2024, 9:22 p.m.

Below the table, it indicates "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

## 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 Django administration header is blue, with the title "Django administration" on the left and the user "NAVEEN" on the right, along with links for "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT". Below the header, a breadcrumb trail shows "Home > Polls". The main content area is titled "Polls administration" and contains a table with four rows: "POLL", "Choices", "Polls", and "Votes". Each row has a "+ Add" button and a "Change" button with a pencil icon. The "POLL" row is highlighted in blue. The Windows taskbar at the bottom shows the system clock as 01:31 on 14-04-2024, the temperature as 29°C with Haze, and various application icons.

Browser: Polls administration | Django site

URL: 127.0.0.1:8000/admin/polls/

Django administration

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

Home > Polls

Polls administration

POLL	
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>

29°C Haze

Search

01:31 14-04-2024

## 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!**