



NEXT GEN EMPLOYABILITY PROGRAM

| Creating a future-ready workforce

Student Name :Logeshwaran V K
Student ID :au820621104310

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Voting Application using Django Framework-Logeshwaran(4310,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:

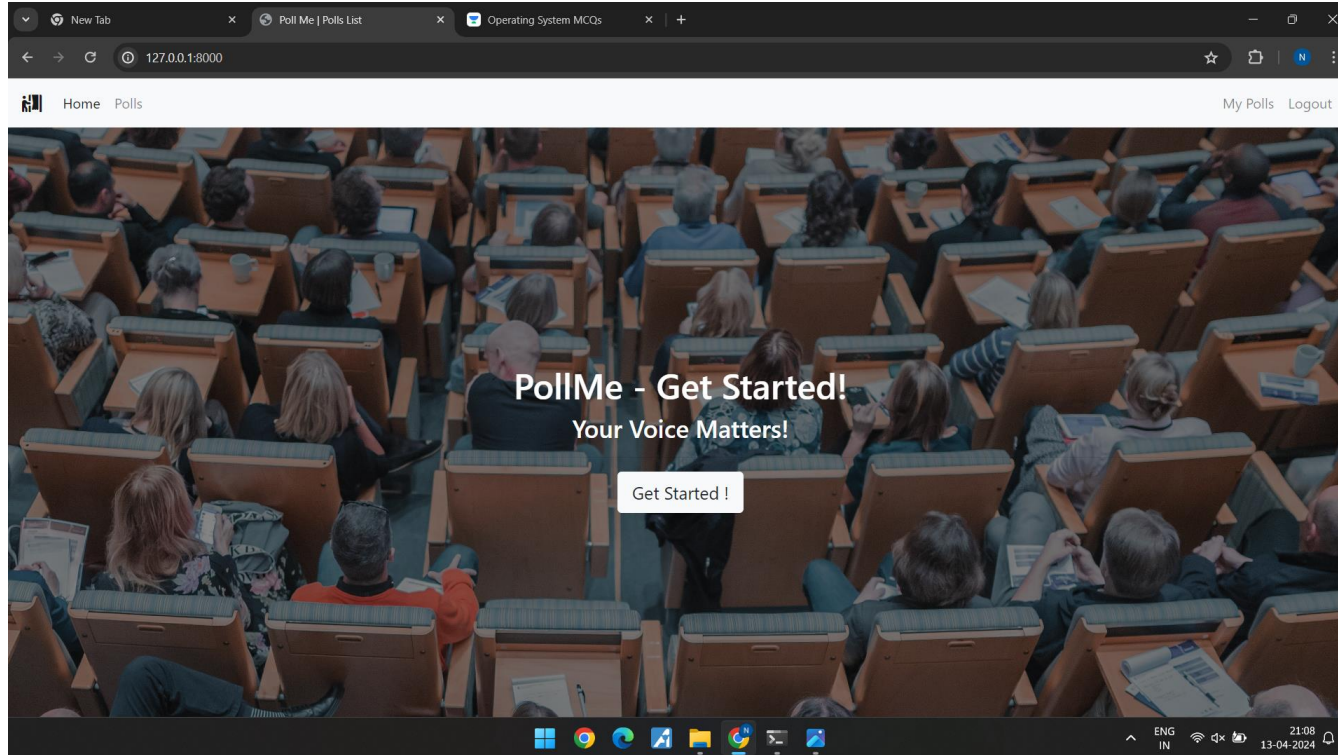
- 1.Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2.Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3.Creating User Authentication:** Implement user authentication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

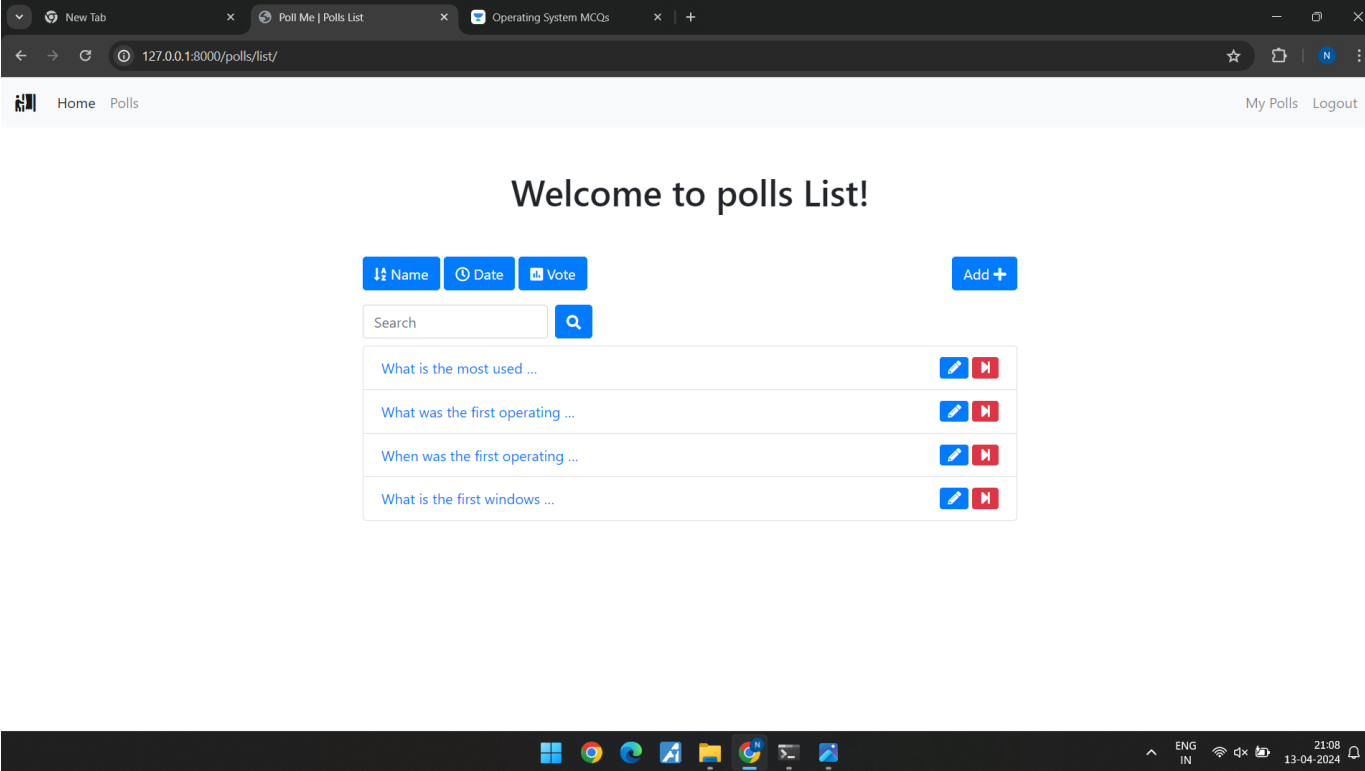
The proposed solution for a voting application using the Django framework is to create a secure and user-friendly online voting platform. The application will allow users to register, vote, and view real-time results. To build the application, the Django framework will be used as the foundation due to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



New Tab x Poll Me | Polls List x Operating System MCQs x +









127.0.0.1:8000/polls/list/

Home Polls My Polls Logout

Welcome to polls List!

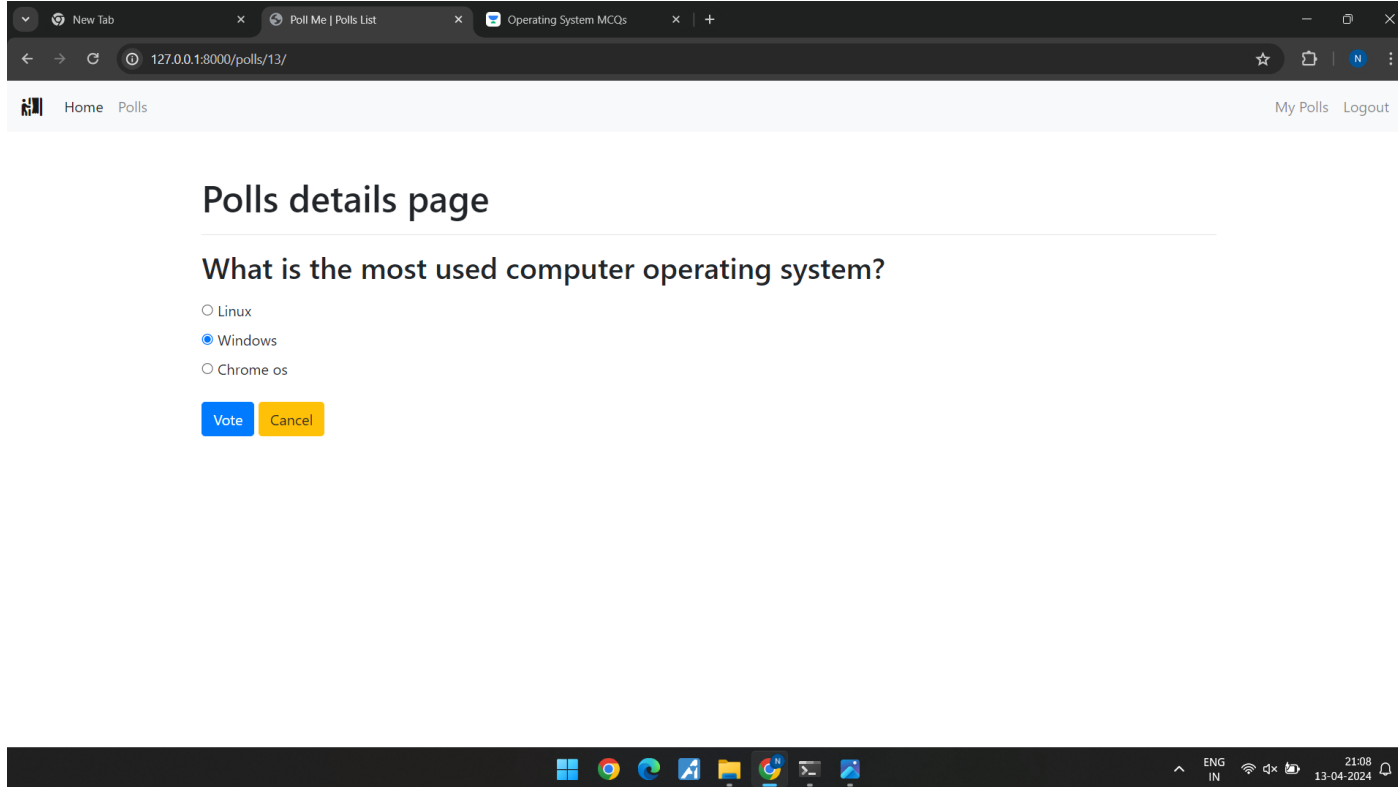
↕ Name ⌚ Date 🗳 Vote Add +

Search 🔍

What is the most used ...	 
What was the first operating ...	 
When was the first operating ...	 
What is the first windows ...	 

Windows Taskbar: 21:08 13-04-2024

Voting Page



New Tab x Poll Me | Polls List x Operating System MCQs x +

127.0.0.1:8000/polls/13/

Home Polls My Polls Logout

Polls details page

What is the most used computer operating system?

☐ Linux

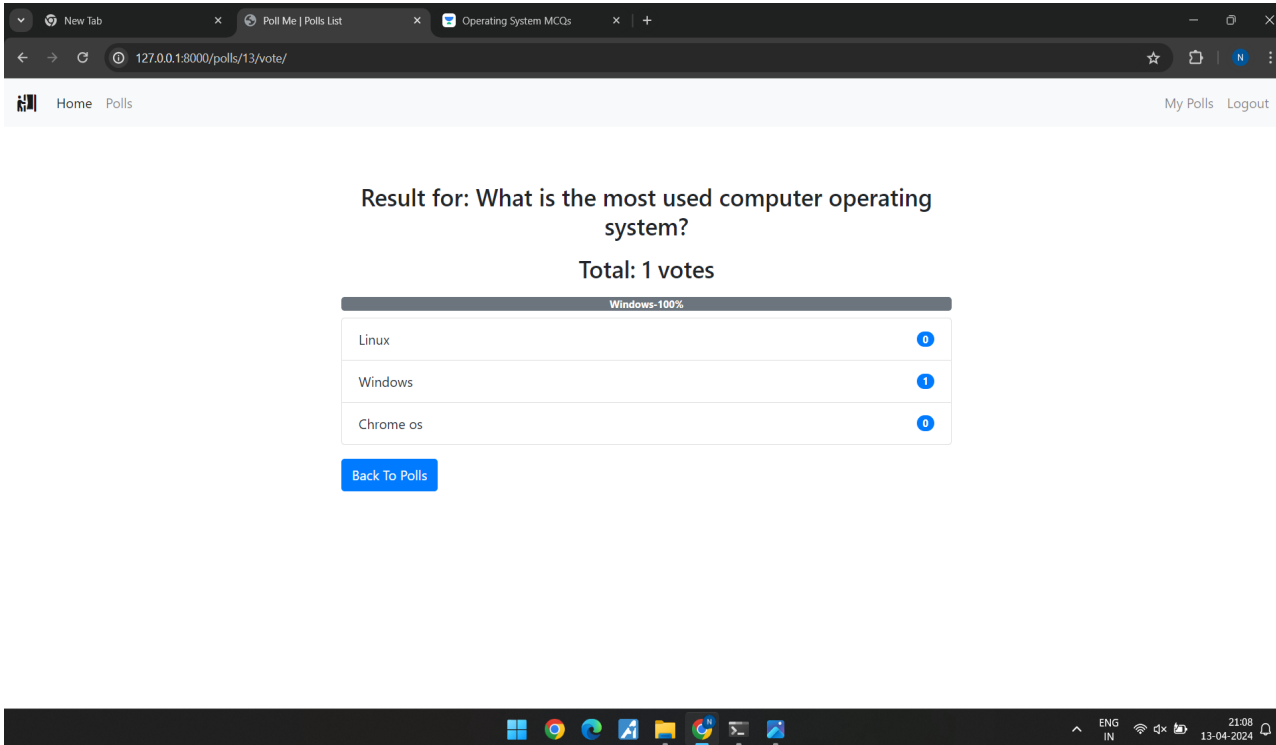
☒ Windows

☐ Chrome os

Vote Cancel

ENG IN 21:08 13-04-2024

Voting Details Page



The screenshot shows a web browser with three tabs: 'New Tab', 'Poll Me | Polls List', and 'Operating System MCQs'. The address bar shows the URL '127.0.0.1:8000/polls/13/vote/'. The page has a navigation bar with 'Home' and 'Polls' links, and 'My Polls' and 'Logout' buttons. The main content area displays the poll result for 'What is the most used computer operating system?'. It shows a total of 1 vote, with a bar chart indicating that Windows received 100% of the votes. The results are listed in a table below the chart.

Result for: What is the most used computer operating system?

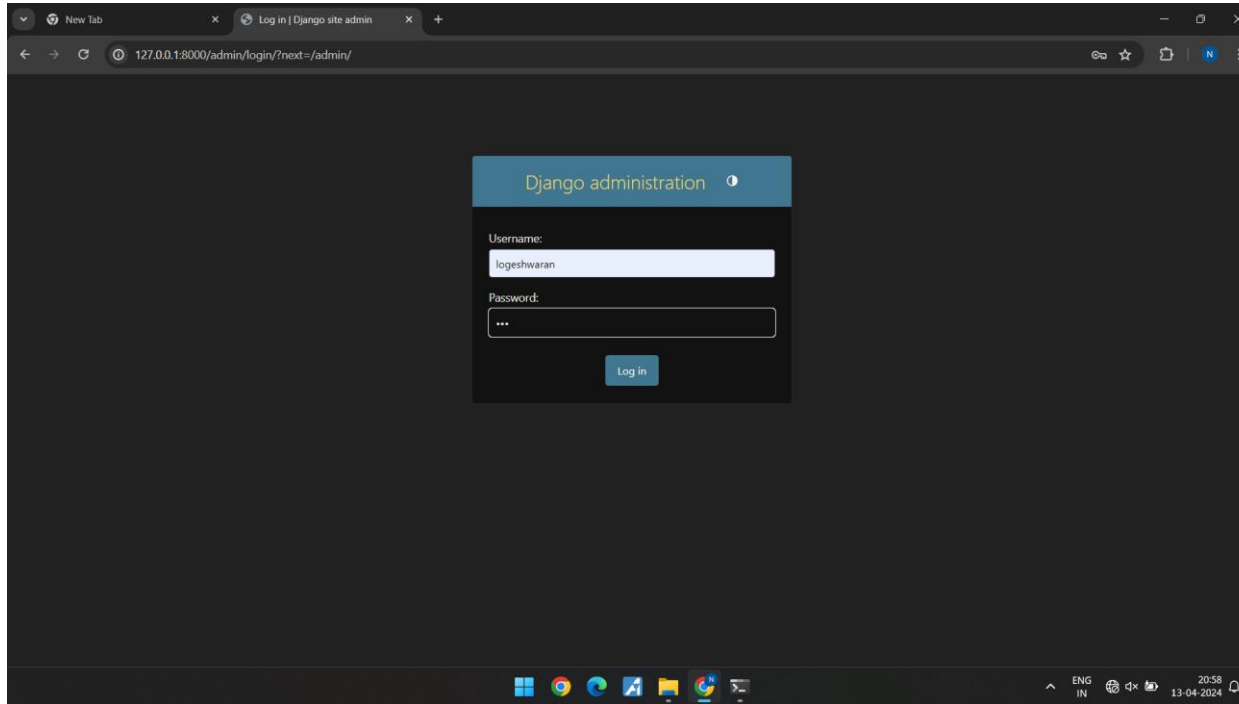
Total: 1 votes

Windows-100%	
Linux	0
Windows	1
Chrome os	0

[Back To Polls](#)

The Windows taskbar at the bottom shows the Start button, task view button, and several pinned applications including Chrome, Edge, and File Explorer. The system tray on the right shows the language set to 'ENG IN', network and volume icons, and the date and time as '21:08 13-04-2024'.

Admin Login Page



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/admin/login/?next=/admin/`. The page content is a dark-themed login form for Django administration. The form has a title bar that says "Django administration" with a help icon. Below the title, there are two input fields: "Username:" with the value "logeshwaran" and "Password:" with masked characters "...". A "Log in" button is positioned below the password field. The browser's taskbar at the bottom shows various application icons and the system clock indicating 20:58 on 13-04-2024.

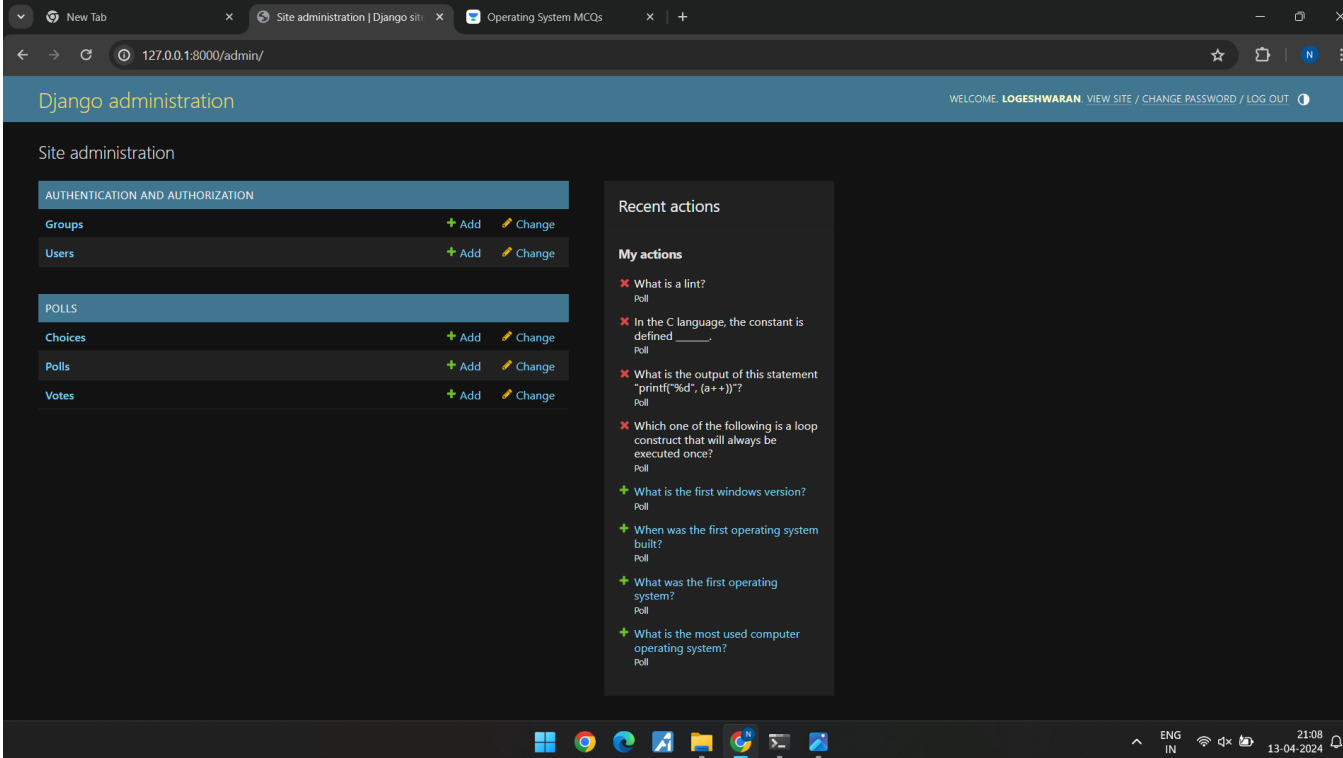
Django administration ⓘ

Username:

Password:

Log in

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 "LOGESHWARAN" with links to "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT".

The main content area is titled "Site administration" and is organized into two primary sections:

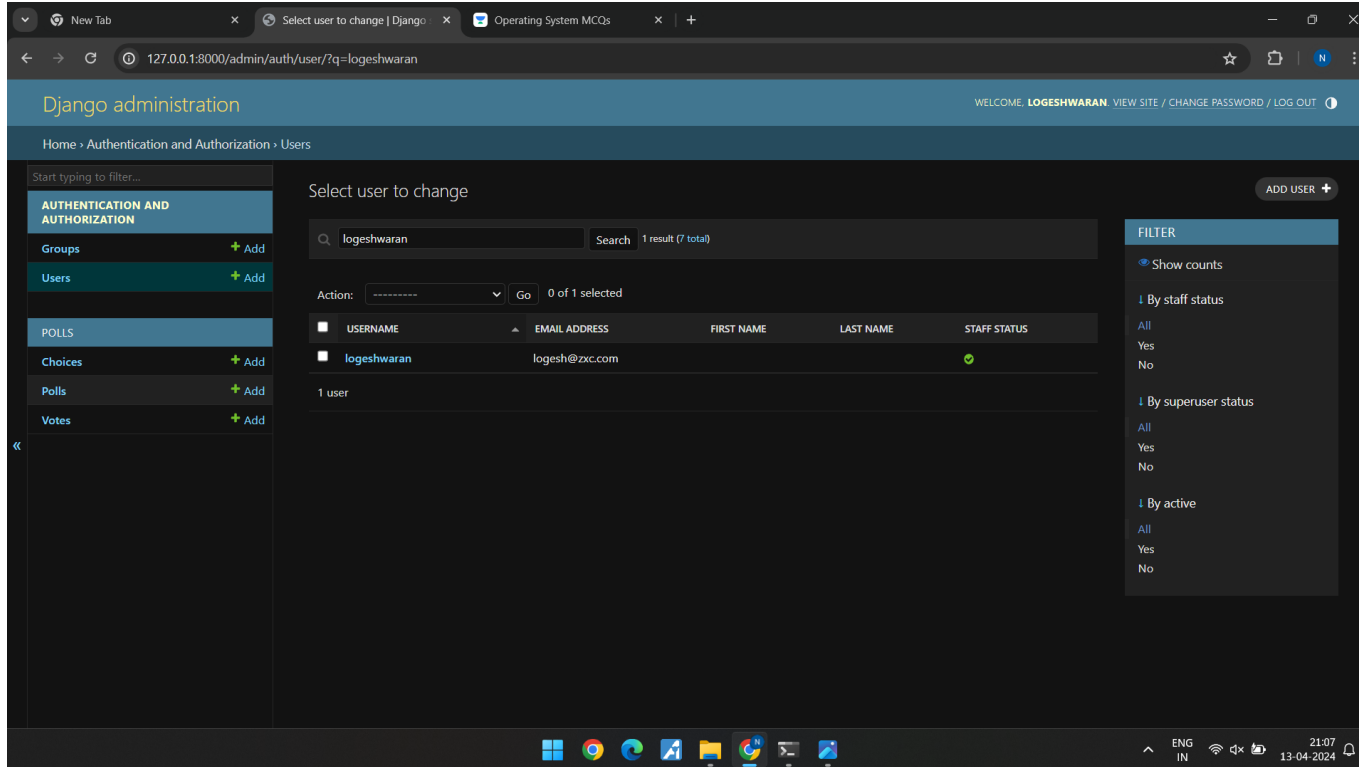
- AUTHENTICATION AND AUTHORIZATION**: This section contains two sub-sections: "Groups" and "Users". Each sub-section has a "+ Add" button and a "Change" link.
- POLLS**: This section contains three sub-sections: "Choices", "Polls", and "Votes". Each sub-section has a "+ Add" button and a "Change" link.

On the right side of the page, there is a "Recent actions" section titled "My actions". It lists several actions, each preceded by a red "X" icon, indicating they are recent or failed actions. The actions are:

- What is a lint? Poll
- In the C language, the constant is defined _____. Poll
- What is the output of this statement "printf("%d", (a++))"? Poll
- Which one of the following is a loop construct that will always be executed once? Poll
- What is the first windows version? Poll
- When was the first operating system built? Poll
- What was the first operating system? Poll
- What is the most used computer operating system? Poll

The bottom of the screenshot shows the Windows taskbar with various application icons and the system clock displaying "21:08" on "13-04-2024".

Authentication and Authorization Page



The screenshot displays the Django administration interface for the 'Authentication and Authorization' section. The browser address bar shows the URL `127.0.0.1:8000/admin/auth/user/?q=logeshwaran`. The page title is 'Django administration'.

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 user to change'. It features a search bar with the text 'logeshwaran' and a 'Search' button. Below the search bar, it indicates '1 result (7 total)'. The 'Action:' dropdown is set to '-----' and the 'Go' button is visible. The table below shows the search results:

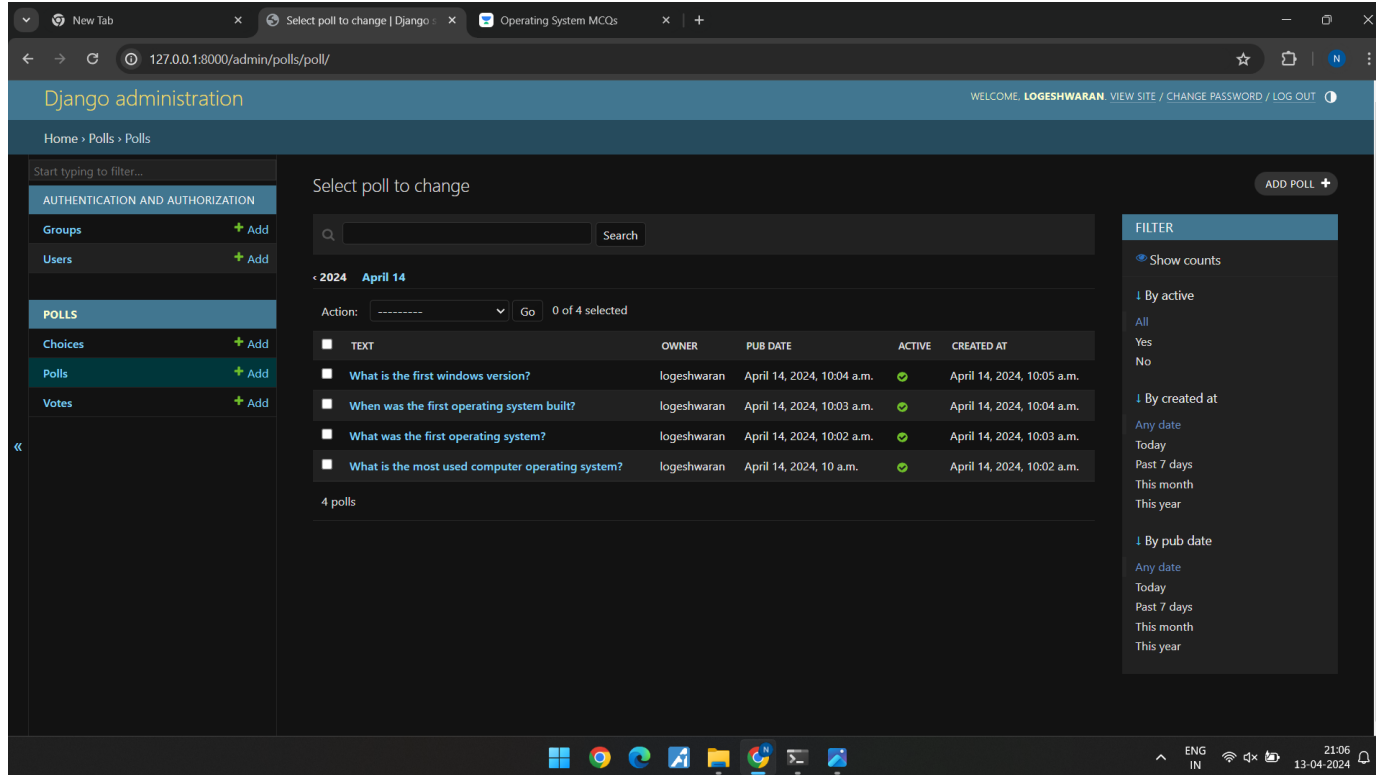
USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
logeshwaran	logesh@zxc.com			Yes

Below the table, it states '1 user'. On the right side of the main content area, there is a 'FILTER' sidebar with the following options:

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

The bottom of the screen shows the Windows taskbar with various application icons and the system clock displaying '21:07' on '13-04-2024'.

Questions Adding Section Page



The screenshot shows the Django administration interface for a poll management system. The browser address bar indicates the URL is `127.0.0.1:8000/admin/polls/poll/`. The page title is "Django administration" and the user is logged in as "LOGESHWARAN".

The sidebar on the left 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

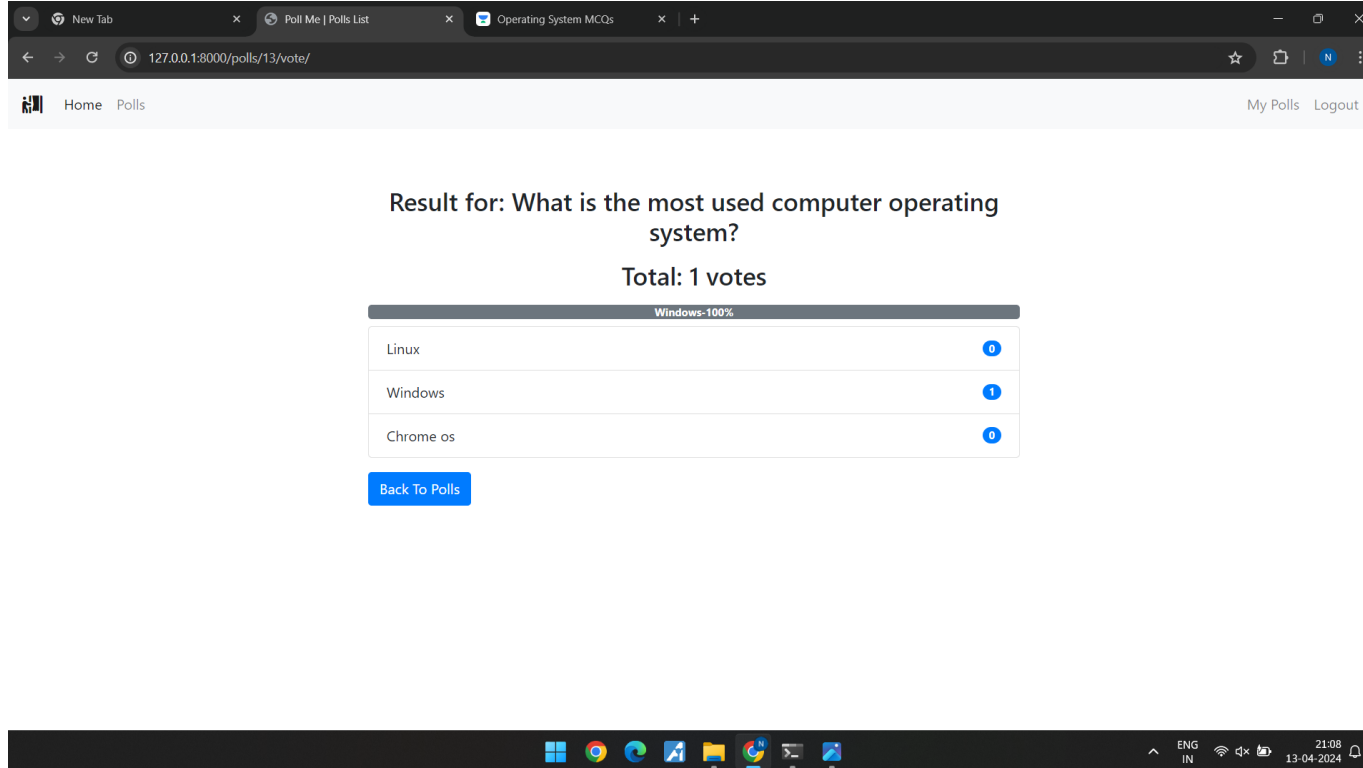
The main content area is titled "Select poll to change". It features a search bar and a table of polls. The table has the following columns: TEXT, OWNER, PUB DATE, ACTIVE, and CREATED AT. There are 4 polls listed, all created on April 14, 2024.

TEXT	OWNER	PUB DATE	ACTIVE	CREATED AT
What is the first windows version?	logeshwaran	April 14, 2024, 10:04 a.m.	Yes	April 14, 2024, 10:05 a.m.
When was the first operating system built?	logeshwaran	April 14, 2024, 10:03 a.m.	Yes	April 14, 2024, 10:04 a.m.
What was the first operating system?	logeshwaran	April 14, 2024, 10:02 a.m.	Yes	April 14, 2024, 10:03 a.m.
What is the most used computer operating system?	logeshwaran	April 14, 2024, 10 a.m.	Yes	April 14, 2024, 10:02 a.m.

The filter sidebar on the right allows filtering by:

- 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 with three tabs: 'New Tab', 'Poll Me | Polls List', and 'Operating System MCQs'. The address bar shows the URL '127.0.0.1:8000/polls/13/vote/'. The page has a navigation bar with 'Home' and 'Polls' links, and 'My Polls' and 'Logout' links on the right. The main content area displays the poll result for 'What is the most used computer operating system?' with a total of 1 vote. A bar chart shows 'Windows-100%'. Below the chart is a table with three rows: 'Linux' with 0 votes, 'Windows' with 1 vote, and 'Chrome os' with 0 votes. A 'Back To Polls' button is located below the table. The Windows taskbar is visible at the bottom of the screen.

Result for: What is the most used computer operating system?

Total: 1 votes

Windows-100%	
Linux	0
Windows	1
Chrome os	0

[Back To Polls](#)

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!