



NEXT GEN EMPLOYABILITY PROGRAM

| Creating a future-ready workforce

Student Name :Dharani S
Student ID :autb21cst001

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Voting Application using Django Framework-DHARANI SIVADASS(4701,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:

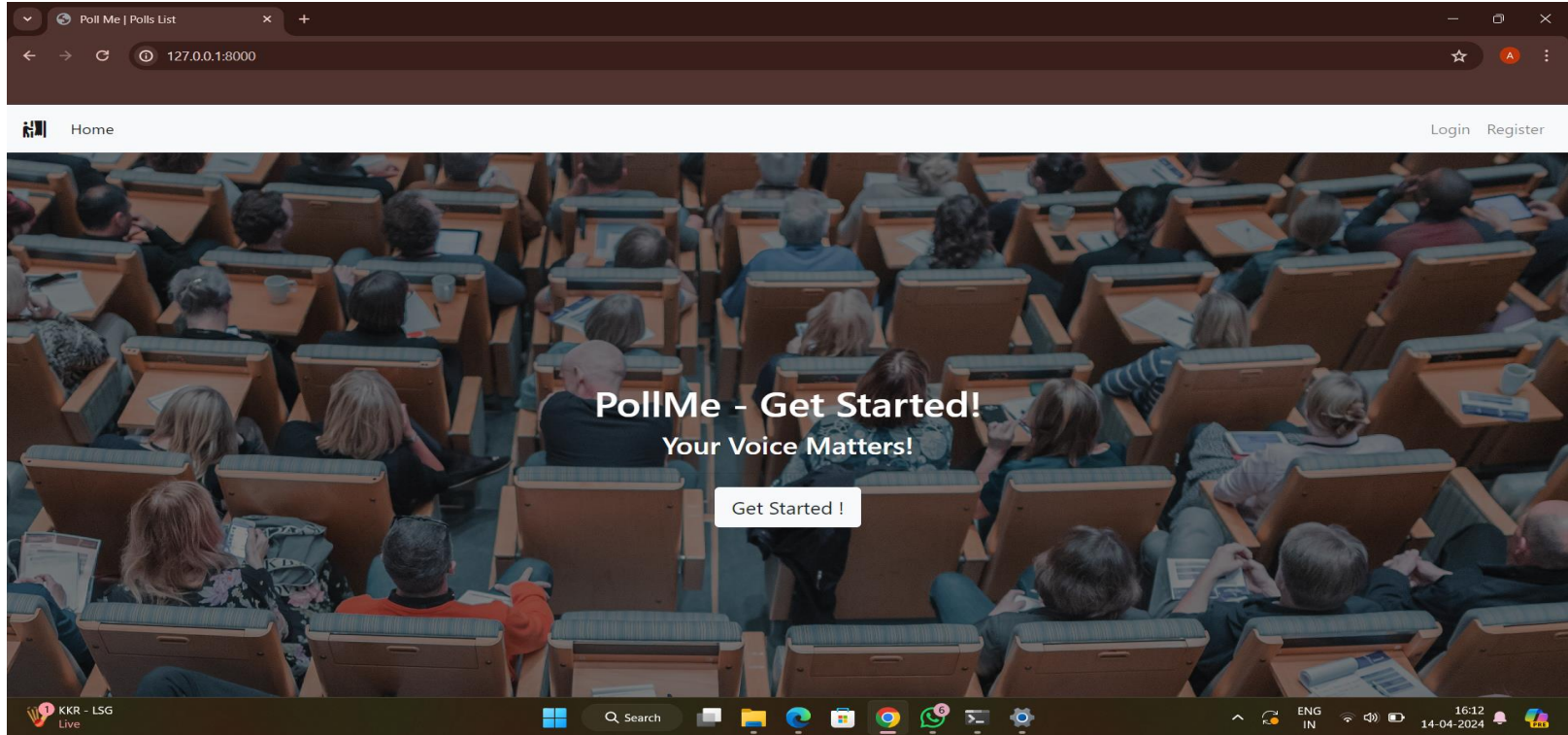
- 1.Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2.Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3.Creating User Authentication:** Implement user authentication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

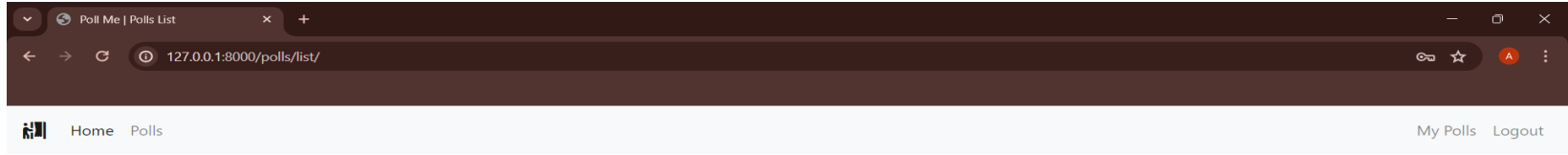
The proposed solution for a voting application using the Django framework is to create a secure and user-friendly online voting platform. The application will allow users to register, vote, and view real-time results. To build the application, the Django framework will be used as the foundation due to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

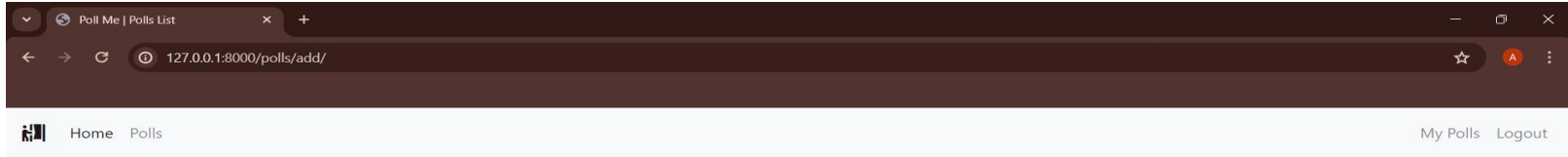
🗳 Vote

Add +

Search



Voting Page



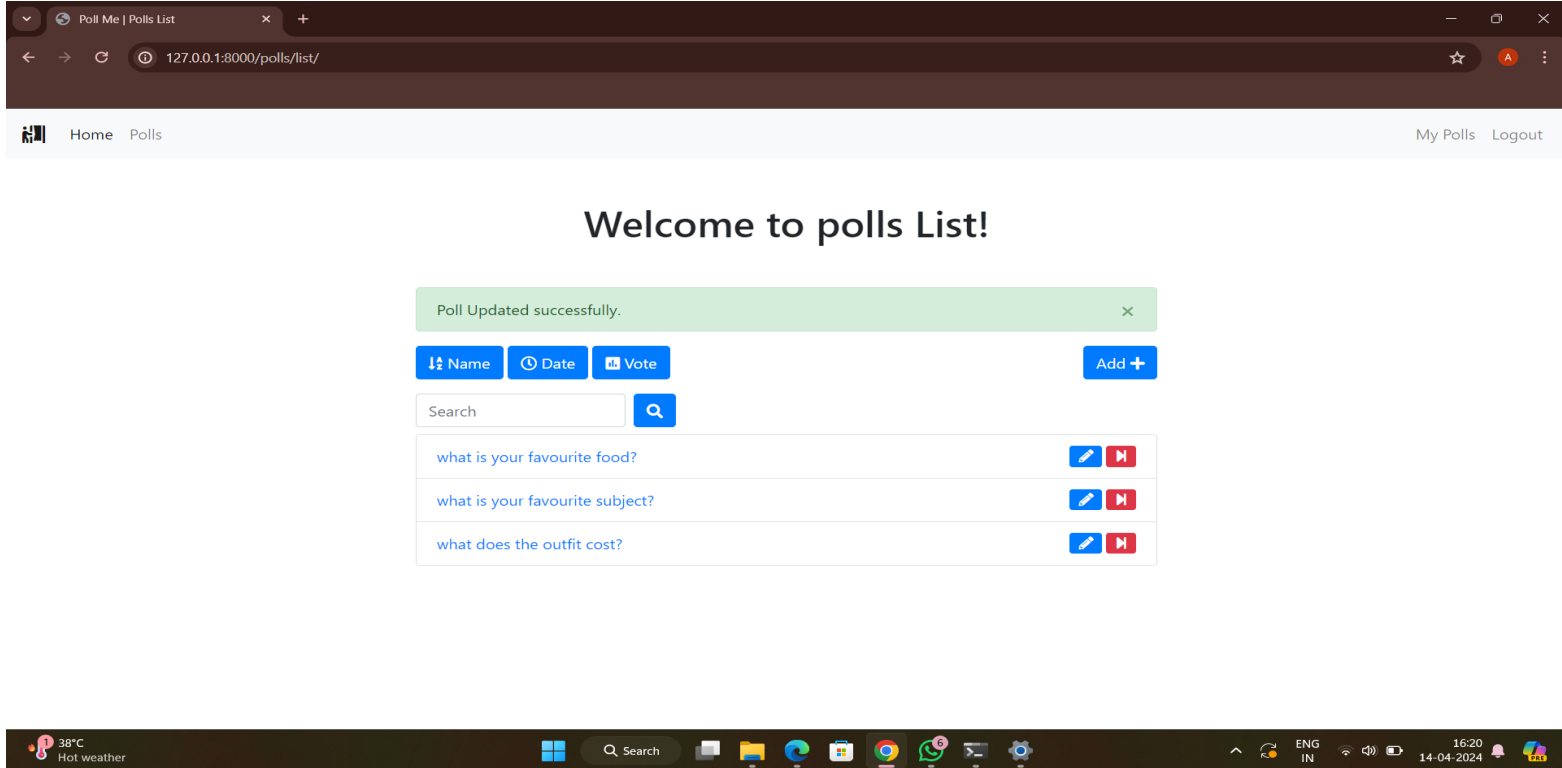
Create new poll

Text:

Choice 1:

Choice 2:

Voting Details Page

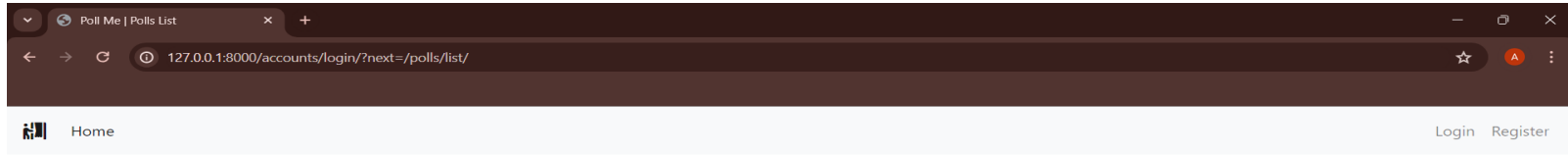


The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/polls/list/". The page has a navigation bar with "Home" and "Polls" links, and "My Polls" and "Logout" links on the right. The main content area is titled "Welcome to polls List!". Below this, a green notification box states "Poll Updated successfully.". There are three buttons: "Name" (with a sort icon), "Date" (with a clock icon), and "Vote" (with a ballot icon), followed by an "Add +" button. A search bar with a magnifying glass icon is present. Below the search bar, there is a list of three poll questions, each with a blue edit icon and a red delete icon:

- what is your favourite food?
- what is your favourite subject?
- what does the outfit cost?

The Windows taskbar at the bottom shows the date as 14-04-2024 and the time as 16:20. The system tray includes icons for weather (38°C Hot weather), language (ENG IN), and other system utilities.

Admin Login Page



Login

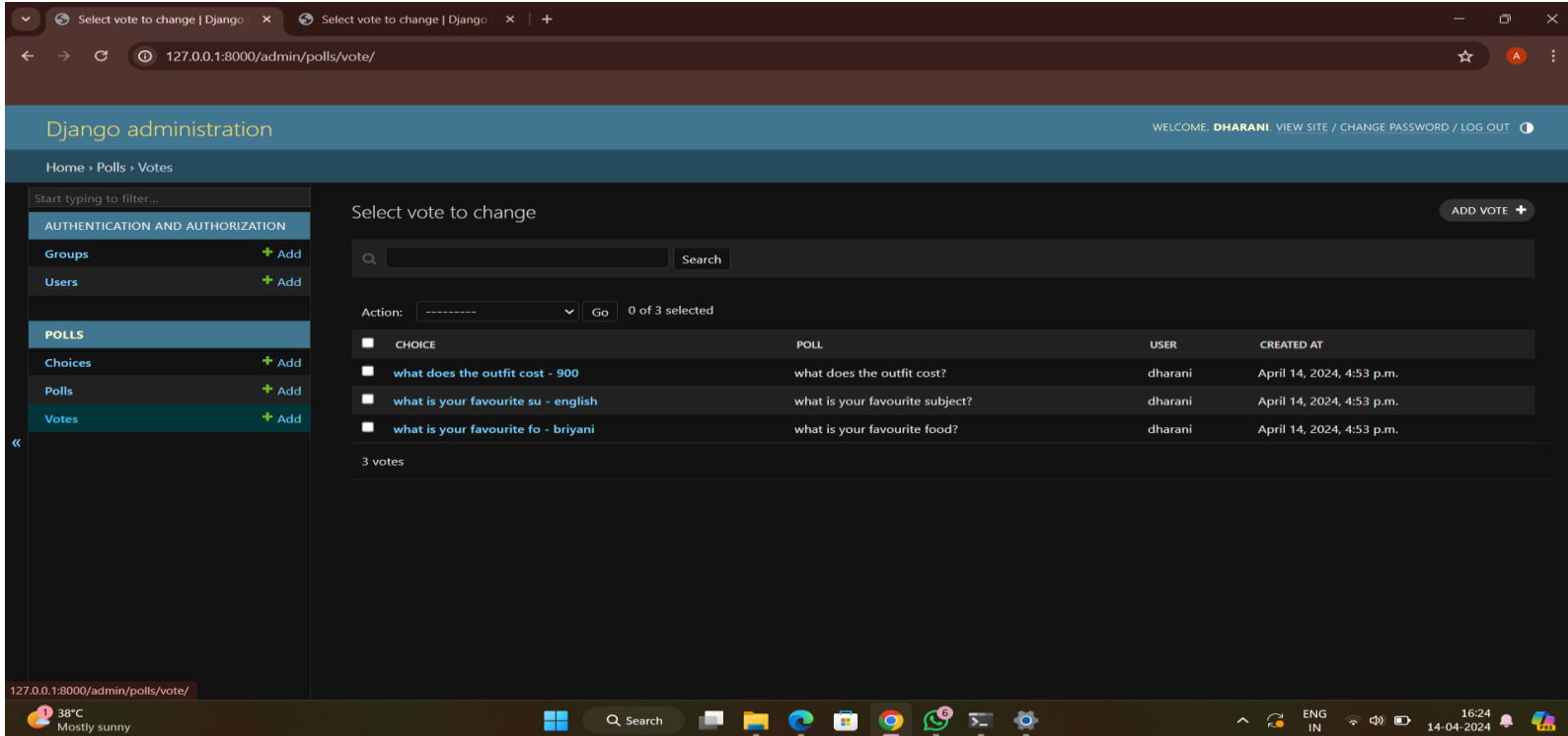
Username

Password

Login

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

Admin Home Page



The screenshot displays the Django administration interface for the 'polls' app. The browser address bar shows the URL `127.0.0.1:8000/admin/polls/vote/`. The page title is 'Django administration', and the user is logged in as 'DHARANI'.

The left sidebar contains the following navigation items:

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

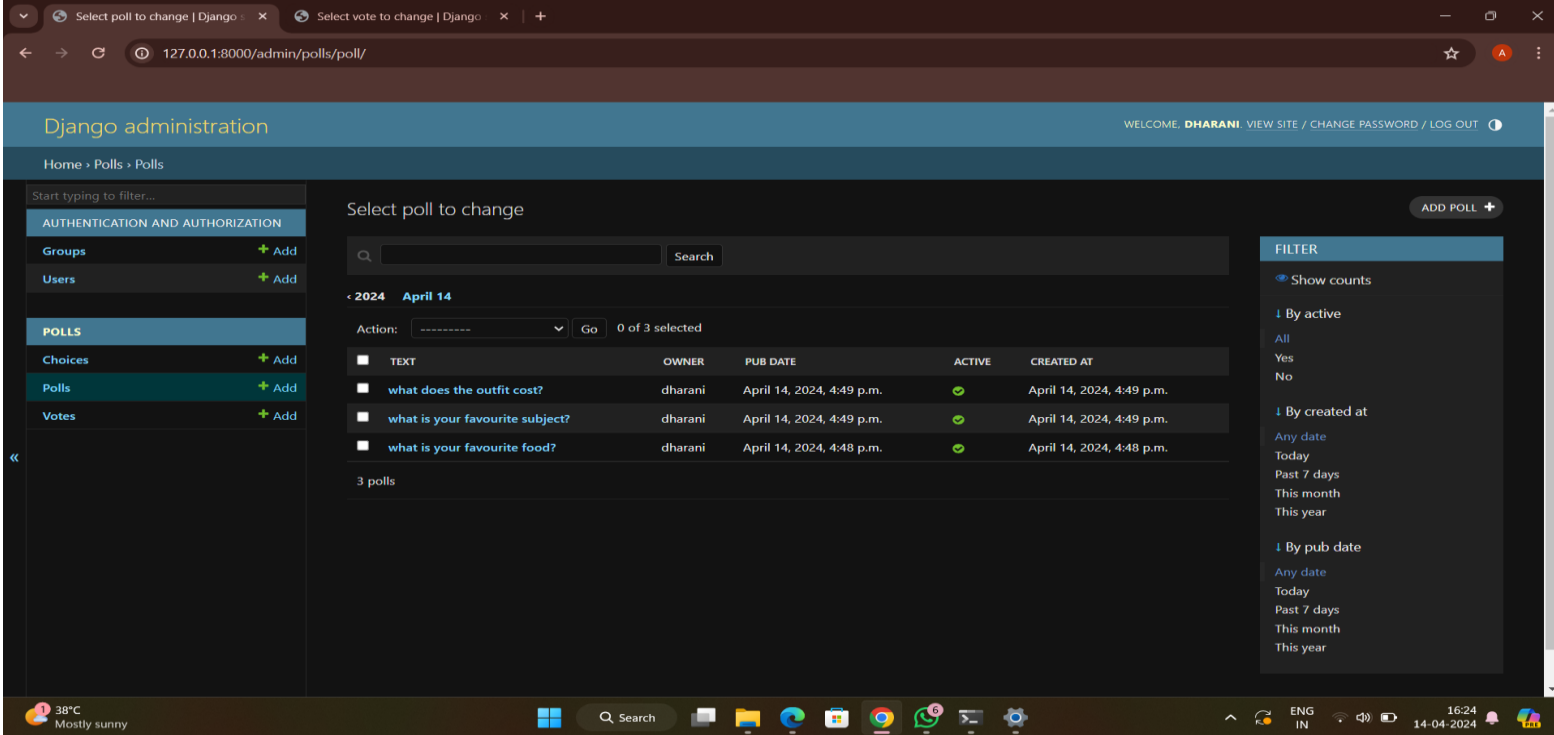
The main content area is titled 'Select vote to change' and includes an 'ADD VOTE +' button. A search bar is present above a table of votes. The table has columns for 'CHOICE', 'POLL', 'USER', and 'CREATED AT'.

CHOICE	POLL	USER	CREATED AT
<input type="checkbox"/> what does the outfit cost - 900	what does the outfit cost?	dharani	April 14, 2024, 4:53 p.m.
<input type="checkbox"/> what is your favourite su - english	what is your favourite subject?	dharani	April 14, 2024, 4:53 p.m.
<input type="checkbox"/> what is your favourite fo - briyani	what is your favourite food?	dharani	April 14, 2024, 4:53 p.m.

Below the table, it indicates '3 votes'.

The bottom of the screen shows a Windows taskbar with the date '14-04-2024' and time '16:24'.

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 "DHARANI".

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 features 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.

TEXT	OWNER	PUB DATE	ACTIVE	CREATED AT
what does the outfit cost?	dharani	April 14, 2024, 4:49 p.m.	✓	April 14, 2024, 4:49 p.m.
what is your favourite subject?	dharani	April 14, 2024, 4:49 p.m.	✓	April 14, 2024, 4:49 p.m.
what is your favourite food?	dharani	April 14, 2024, 4:48 p.m.	✓	April 14, 2024, 4:48 p.m.

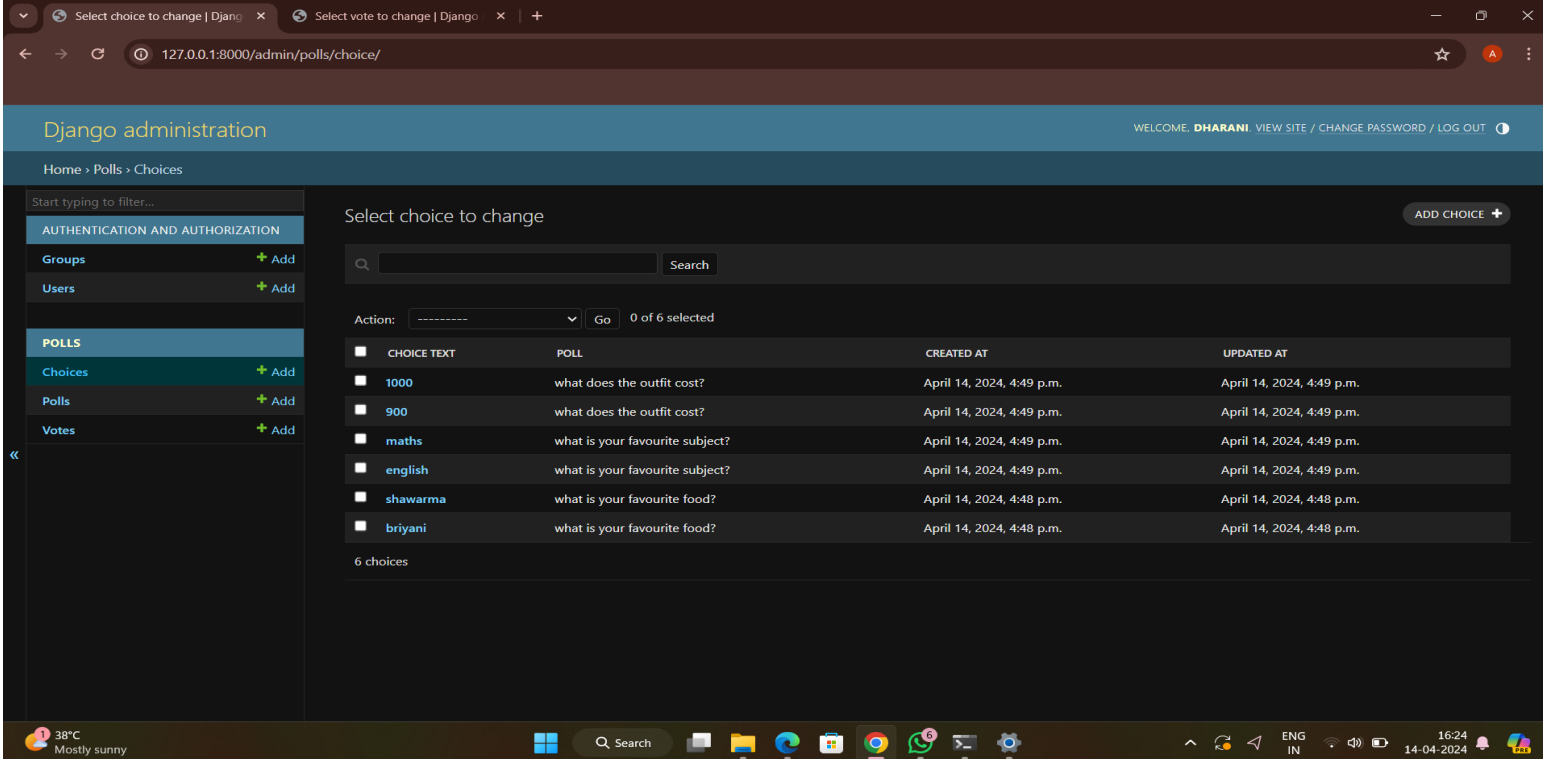
Below the table, it indicates "3 polls".

On the right side, there is a "FILTER" panel 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 Windows taskbar at the bottom shows the date and time as 16:24 on 14-04-2024, along with system icons for weather (38°C, Mostly sunny), search, and various applications.

Questions Adding Section Page



The screenshot shows the Django administration interface for the 'Polls' app, specifically the 'Choices' section. The browser address bar shows the URL '127.0.0.1:8000/admin/polls/choice/'. The page title is 'Django administration' and the user is logged in as 'DHARANI'. The left sidebar shows the navigation menu with 'Polls' selected, and 'Choices' is the current view. The main content area is titled 'Select choice to change' and features a search bar and an 'ADD CHOICE +' button. Below this, there is a table of existing choices. The table has four columns: 'CHOICE TEXT', 'POLL', 'CREATED AT', and 'UPDATED AT'. There are six rows of data, each with a checkbox in the first column. The choices are: '1000' (poll: what does the outfit cost?), '900' (poll: what does the outfit cost?), 'maths' (poll: what is your favourite subject?), 'english' (poll: what is your favourite subject?), 'shawarma' (poll: what is your favourite food?), and 'briyani' (poll: what is your favourite food?). All choices were created on April 14, 2024, at 4:49 p.m. except for 'shawarma' and 'briyani' which were created at 4:48 p.m. The bottom of the screen shows a Windows taskbar with the date and time '16:24 14-04-2024'.

Django administration

WELCOME, DHARANI | VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Polls > Choices

Select choice to change

ADD CHOICE +

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

POLLS

Choices + Add

Polls + Add

Votes + Add

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

CHOICE TEXT	POLL	CREATED AT	UPDATED AT
<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"/> briyani	what is your favourite food?	April 14, 2024, 4:48 p.m.	April 14, 2024, 4:48 p.m.

6 choices

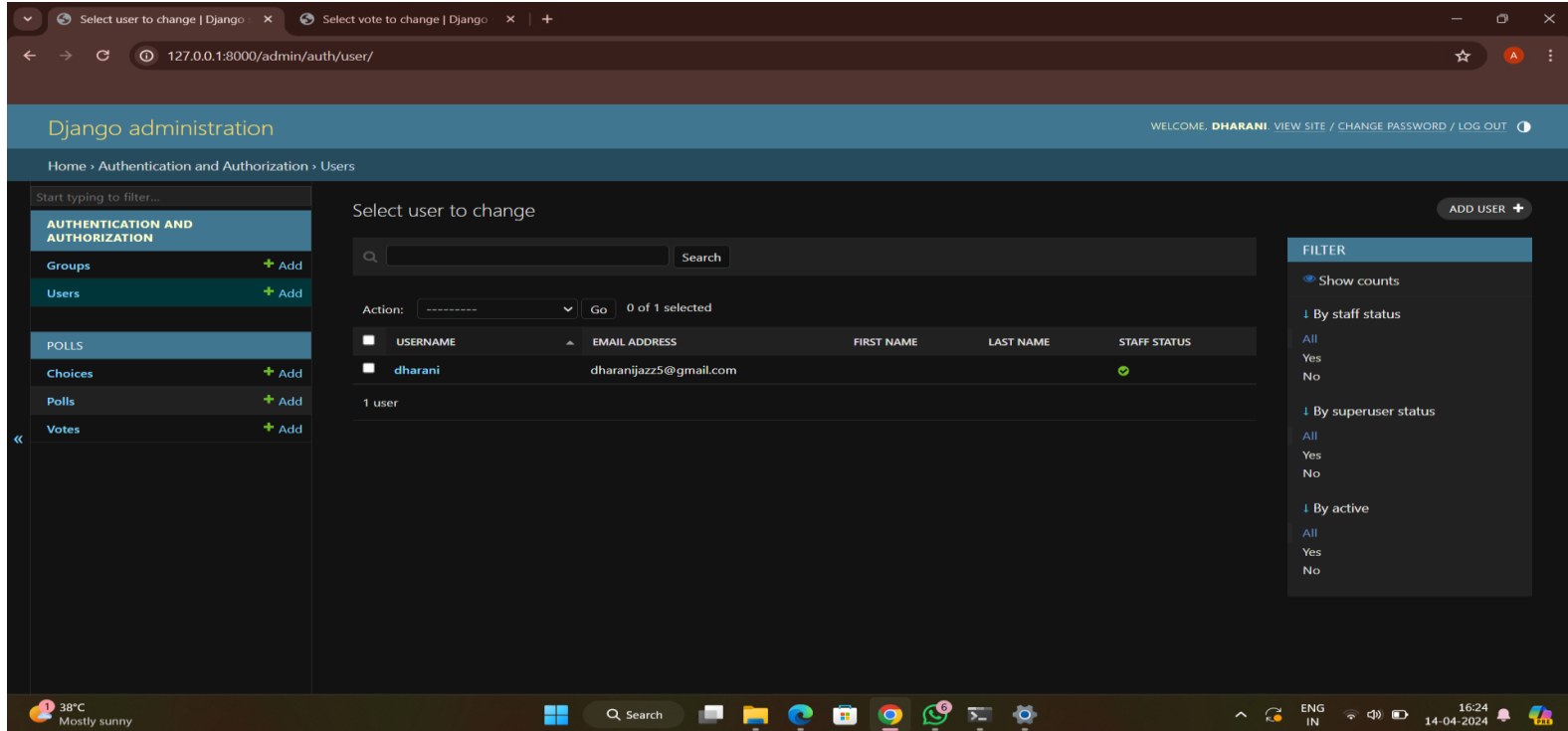
38°C Mostly sunny

Search

ENG IN

16:24 14-04-2024

Voting Details 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/auth/user/`. The page title is "Django administration". The user is logged in as "DHARANI", and the top right corner provides links for "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT".

The left sidebar contains a navigation menu with the following items:

- 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 and a "Search" button. Below the search bar, there is an "Action:" dropdown menu and a "Go" button, indicating that 0 of 1 selected items are shown.

The table below lists the available users:

USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
dharani	dharanijazz5@gmail.com			Yes

Below the table, it states "1 user".

On the right side, there is a "FILTER" panel 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 screenshot shows a Windows taskbar with the date and time "14-04-2024 16:24".

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!