



NEXT GEN EMPLOYABILITY PROGRAM

| Creating a future-ready workforce

Student Name : GOKUL M

Student ID : autb21csl017

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Django Voting-GOKUL(4308,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:

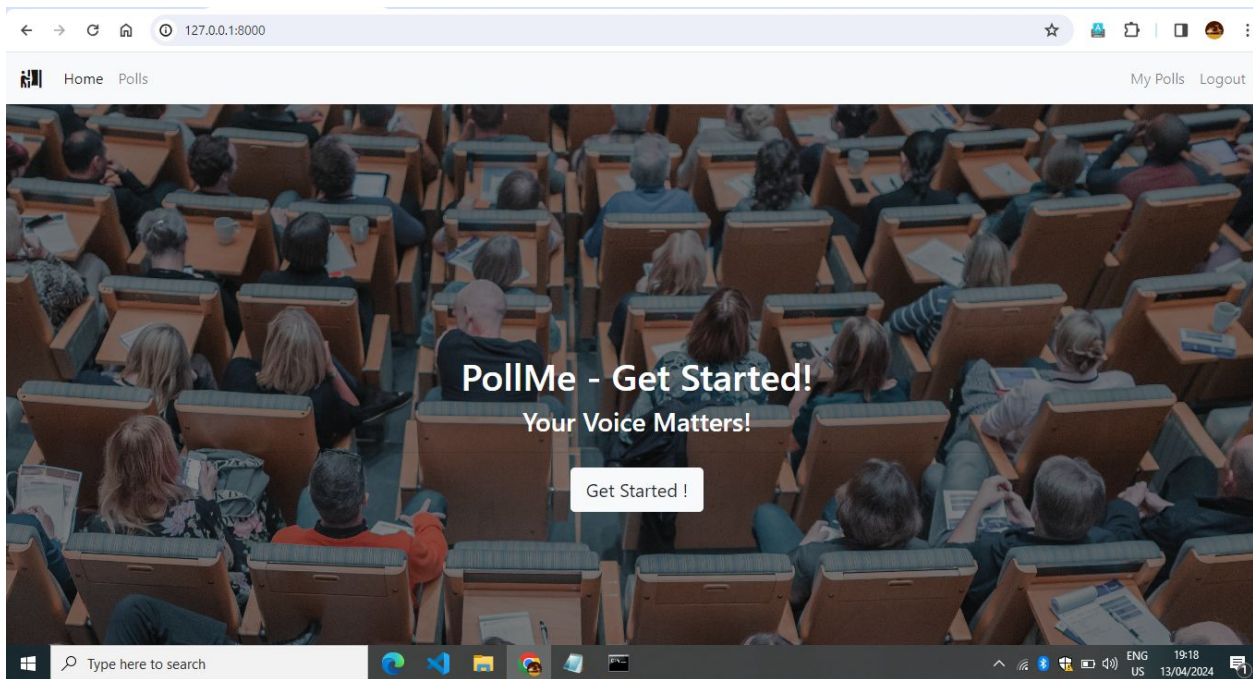
- 1. Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2. Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3. Creating User Authentication:** Implement user authentication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

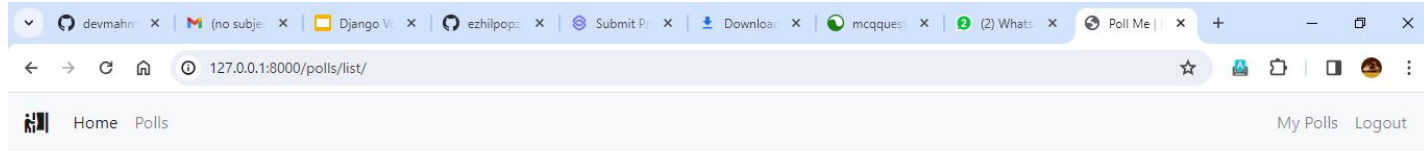
The proposed solution for a voting application using the Django framework is to create a secure and user-friendly online voting platform. The application will allow users to register, vote, and view real-time results. To build the application, the Django framework will be used as the foundation due to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

🔍

what is your favorite food?

✎ 🗑

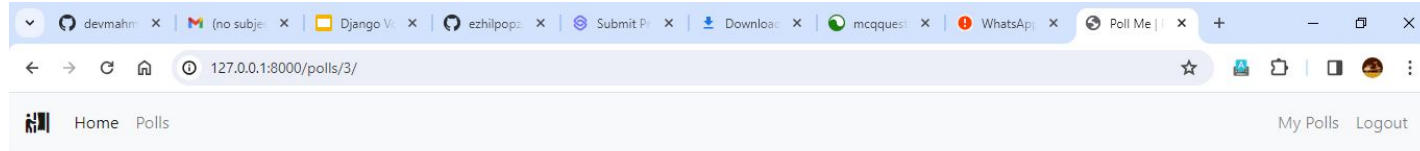
what is your recent Vacation ...

✎ 🗑

Which of the following is ...

✎ 🗑

Voting Page



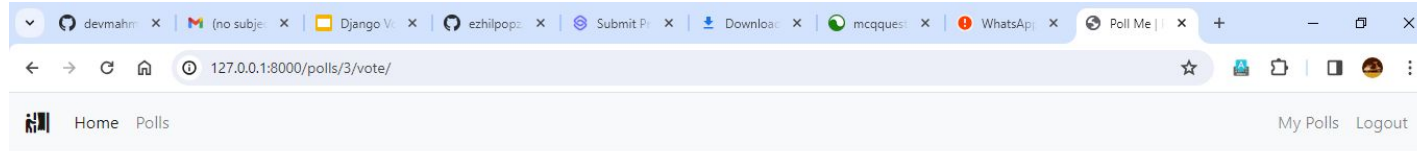
Polls details page

Which of the following is NOT a portable device ?

- ☐ Desktop computers
- ☐ Laptops

[Vote](#) [Cancel](#)

Voting Details Page



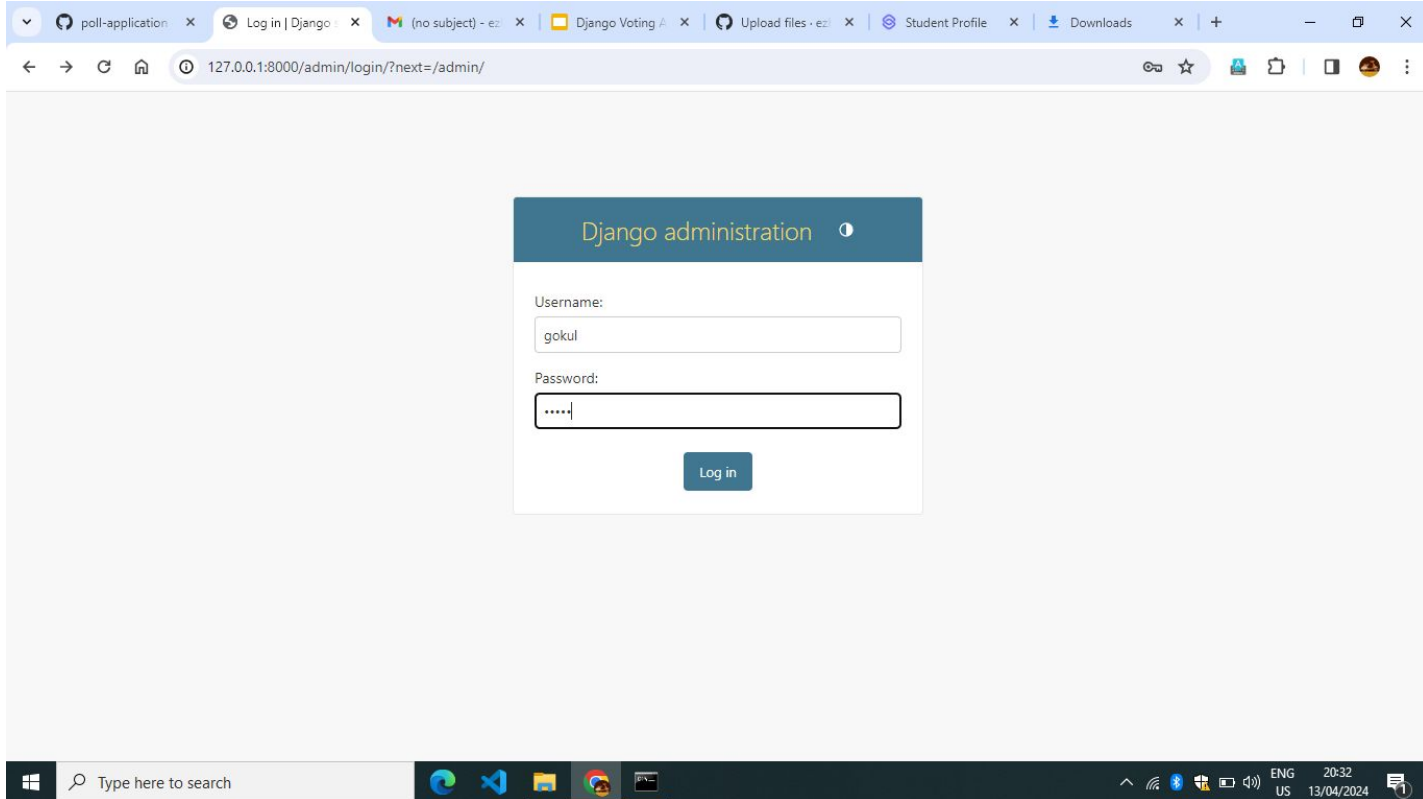
Result for: Which of the following is NOT a portable device ?

Total: 1 votes

Desktop computers-100%	
Desktop computers	1
Laptops	0

[Back To Polls](#)

Admin home



poll-application x Log in | Django x (no subject) - e: x Django Voting A x Upload files - e: x Student Profile x Downloads x +

127.0.0.1:8000/admin/login/?next=/admin/

Django administration

Username:

gokul

Password:

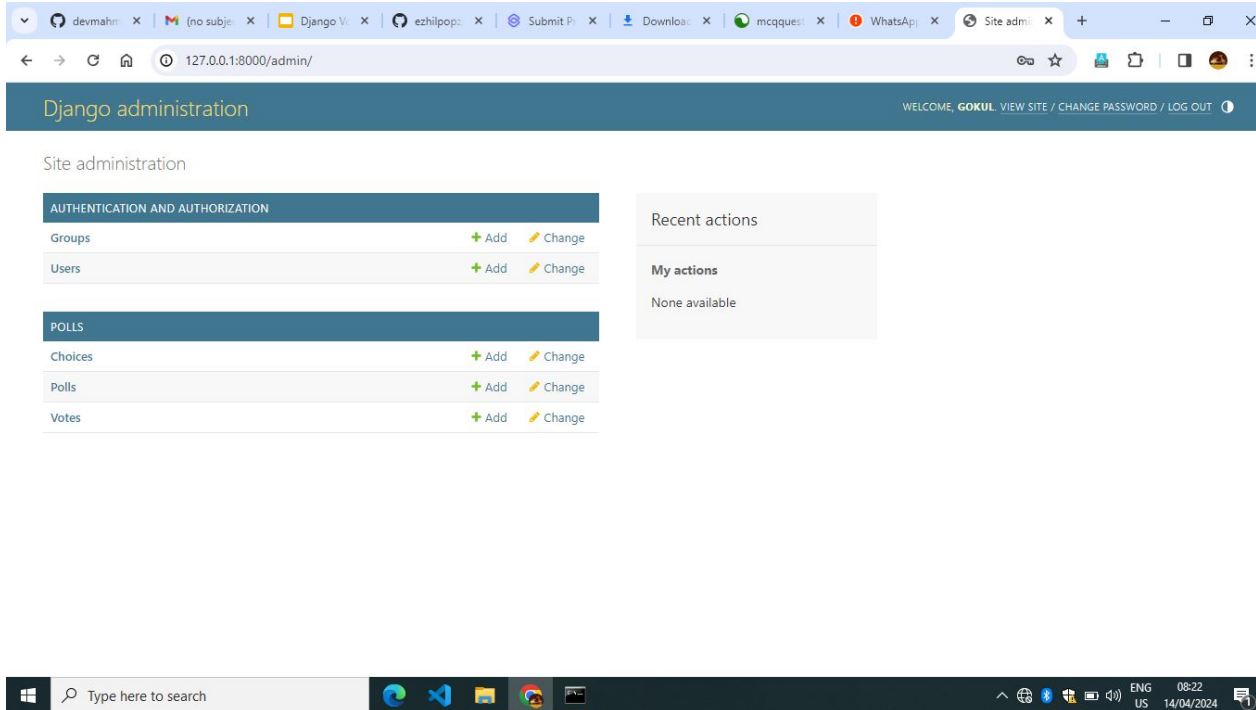
.....

Log in

Type here to search

ENG US 20:32 13/04/2024

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 title "Django administration" and a welcome message for "GOKUL" 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:

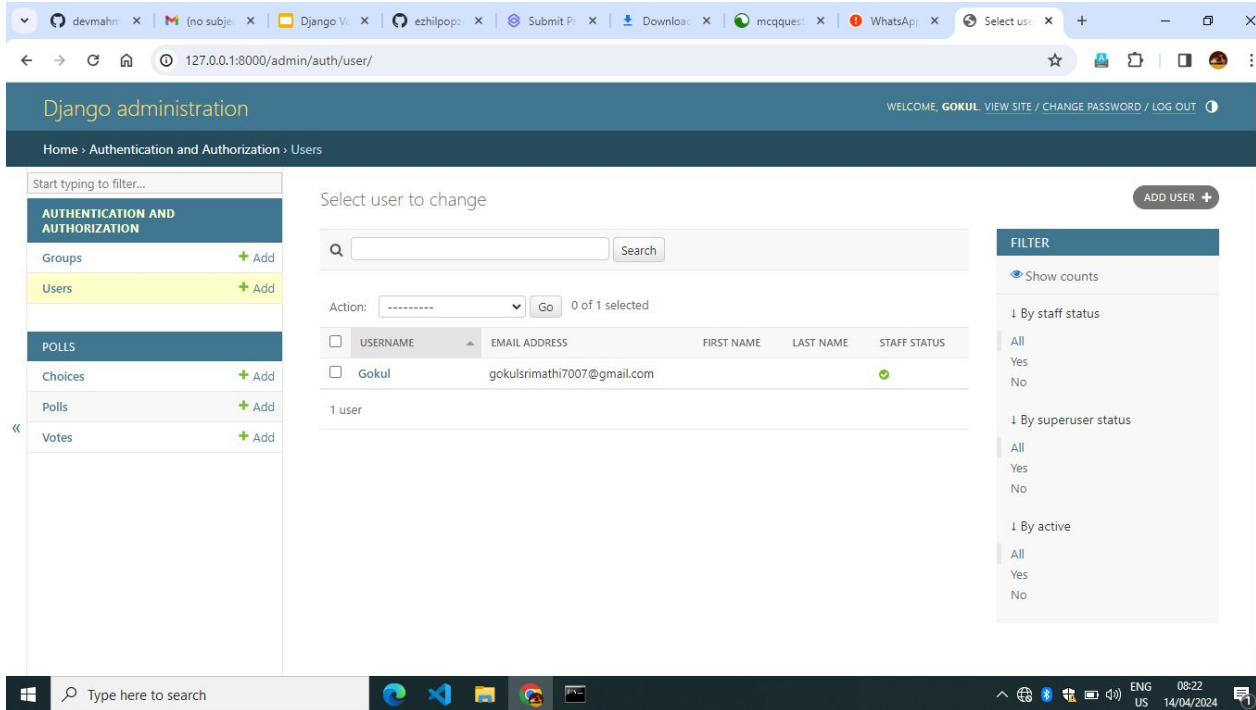
- AUTHENTICATION AND AUTHORIZATION**: This section includes two rows, "Groups" and "Users", each with a "+ Add" link and a "Change" link (represented by a pencil icon).
- POLLS**: This section includes three rows, "Choices", "Polls", and "Votes", each with a "+ Add" link and a "Change" link (represented by a pencil icon).

The right column contains two sections:

- Recent actions**: This section is currently empty.
- My actions**: This section displays the text "None available".

The Windows taskbar at the bottom of the screen shows the search bar with the text "Type here to search", several application icons (including Edge, VS Code, File Explorer, and Chrome), and system information indicating the time is 08:22 and the date is 14/04/2024.

Authentication and Authorization Page



The screenshot displays the Django administration interface for the 'Authentication and Authorization' section, specifically the 'Users' page. The browser address bar shows the URL '127.0.0.1:8000/admin/auth/user/'. The page header includes the Django administration logo and a welcome message for 'GOKUL' with links to '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

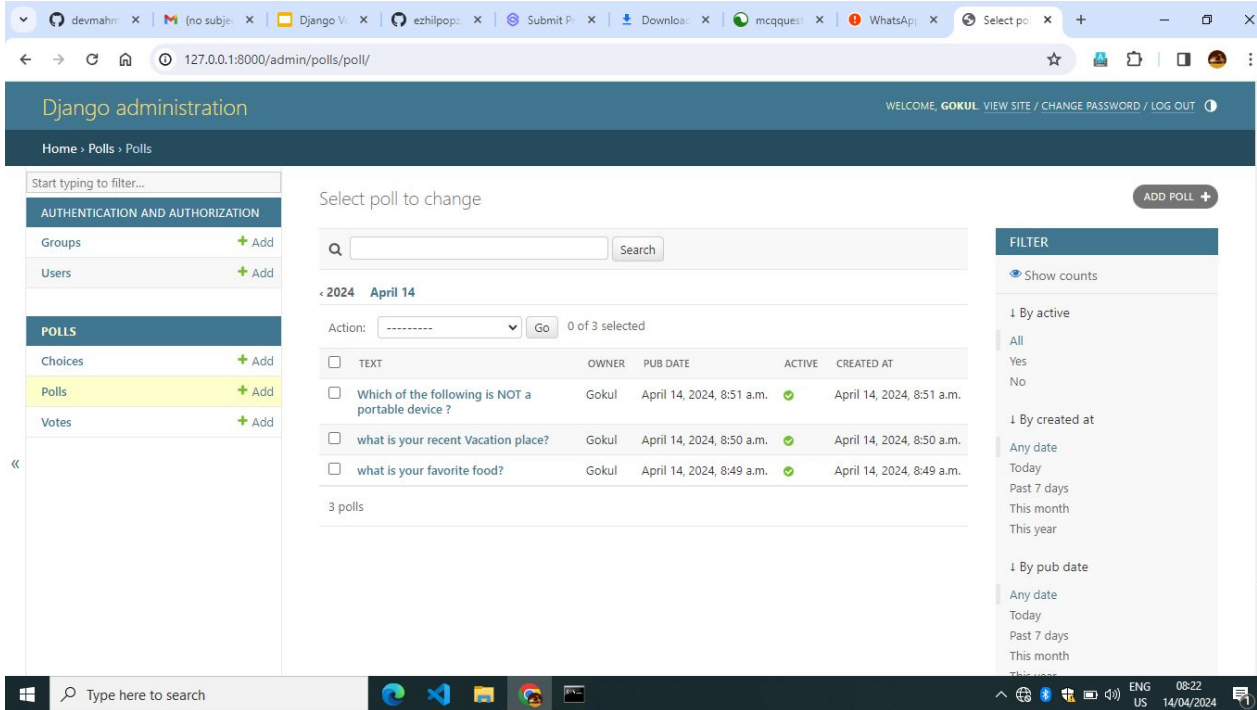
The main content area is titled 'Select user to change'. It features a search bar and a table of users. The table has columns for USERNAME, EMAIL ADDRESS, FIRST NAME, LAST NAME, and STAFF STATUS. One user, 'Gokul', is listed with the email 'gokulsrinathi7007@gmail.com' and a staff status of 'Yes'.

The right sidebar contains a 'FILTER' section 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 the Windows taskbar with the search bar and various application icons.

Questions Adding Section Page



The screenshot shows the Django administration interface for the 'Polls' app. The browser address bar indicates the URL is 127.0.0.1:8000/admin/polls/poll/. The interface includes a sidebar with navigation links for Authentication and Authorization, Polls, and Votes. The main content area displays a list of polls with columns for Action, Text, Owner, Pub Date, Active status, and Created At. A search bar and a filter sidebar are also visible.

Django administration WELCOME, GOKUL. [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

Home > Polls > Polls

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

- Groups [+ Add](#)
- Users [+ Add](#)

POLLS

- Choices [+ Add](#)
- Polls** [+ Add](#)
- Votes [+ Add](#)

Select poll to change

Q

< 2024 April 14

Action: 0 of 3 selected

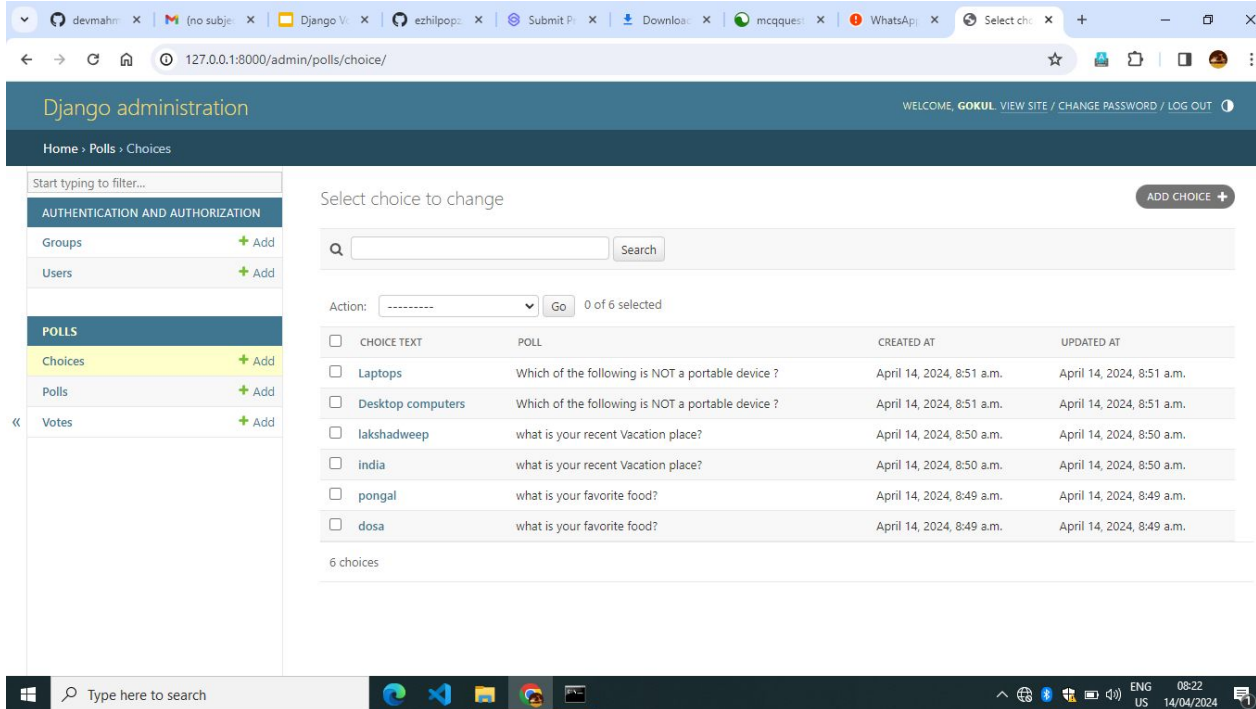
<input type="checkbox"/>	TEXT	OWNER	PUB DATE	ACTIVE	CREATED AT
<input type="checkbox"/>	Which of the following is NOT a portable device ?	Gokul	April 14, 2024, 8:51 a.m.	✓	April 14, 2024, 8:51 a.m.
<input type="checkbox"/>	what is your recent Vacation place?	Gokul	April 14, 2024, 8:50 a.m.	✓	April 14, 2024, 8:50 a.m.
<input type="checkbox"/>	what is your favorite food?	Gokul	April 14, 2024, 8:49 a.m.	✓	April 14, 2024, 8:49 a.m.

3 polls

FILTER

- 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. The address bar shows the URL `127.0.0.1:8000/admin/polls/choice/`. The page title is "Django administration". The user is logged in as "GOKUL" and can view the site, change password, or log out.

The left sidebar shows the navigation menu with the following items:

- Home > Polls > Choices
- 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 choice to change" and includes an "ADD CHOICE +" button. Below this is a search bar and a table of choices.

Search: Search

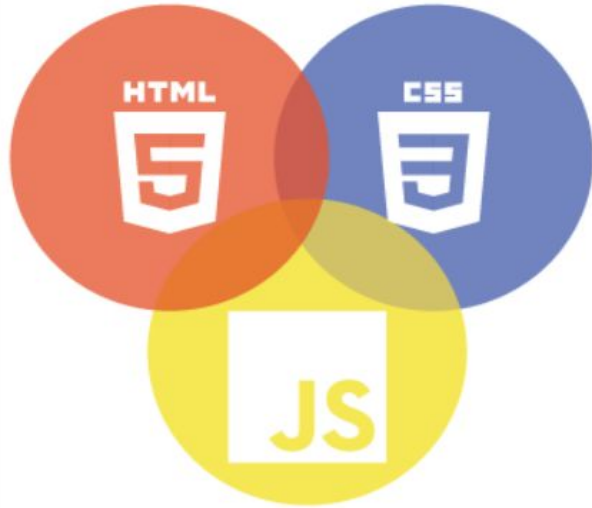
Action: Go 0 of 6 selected

<input type="checkbox"/>	CHOICE TEXT	POLL	CREATED AT	UPDATED AT
<input type="checkbox"/>	Laptops	Which of the following is NOT a portable device ?	April 14, 2024, 8:51 a.m.	April 14, 2024, 8:51 a.m.
<input type="checkbox"/>	Desktop computers	Which of the following is NOT a portable device ?	April 14, 2024, 8:51 a.m.	April 14, 2024, 8:51 a.m.
<input type="checkbox"/>	lakshadweep	what is your recent Vacation place?	April 14, 2024, 8:50 a.m.	April 14, 2024, 8:50 a.m.
<input type="checkbox"/>	india	what is your recent Vacation place?	April 14, 2024, 8:50 a.m.	April 14, 2024, 8:50 a.m.
<input type="checkbox"/>	pongal	what is your favorite food?	April 14, 2024, 8:49 a.m.	April 14, 2024, 8:49 a.m.
<input type="checkbox"/>	dosa	what is your favorite food?	April 14, 2024, 8:49 a.m.	April 14, 2024, 8:49 a.m.

6 choices

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!