



NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Student Name : Sakthi Aravind S
Student ID : au820621104072

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Voting Application using Django Framework-Sakthi Aravind S(4072,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:

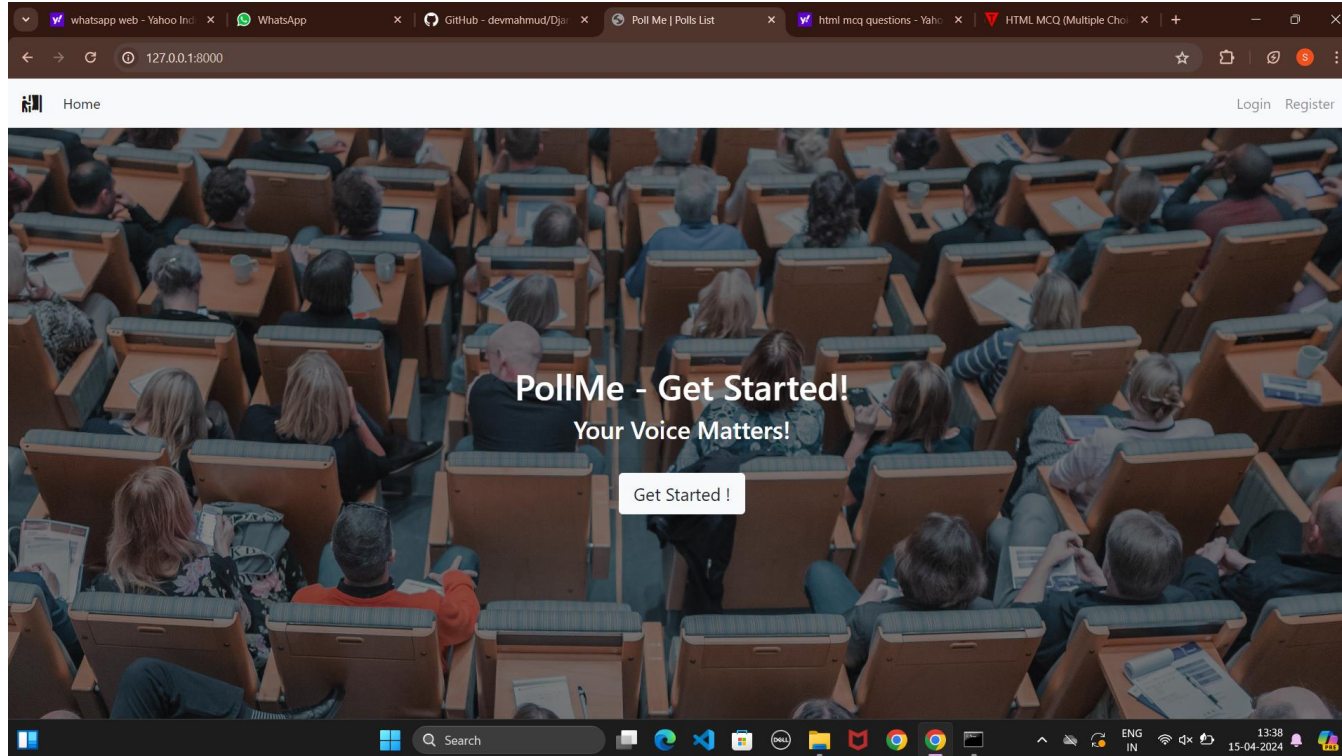
- 1.Setting up a Django Project:** Create a Django project to serve as the foundation for the voting application.
- 2.Designing the Database Schema:** Define the database structure to store user information, votes, and other relevant data.
- 3.Creating User Authentication:** Implement user authentication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

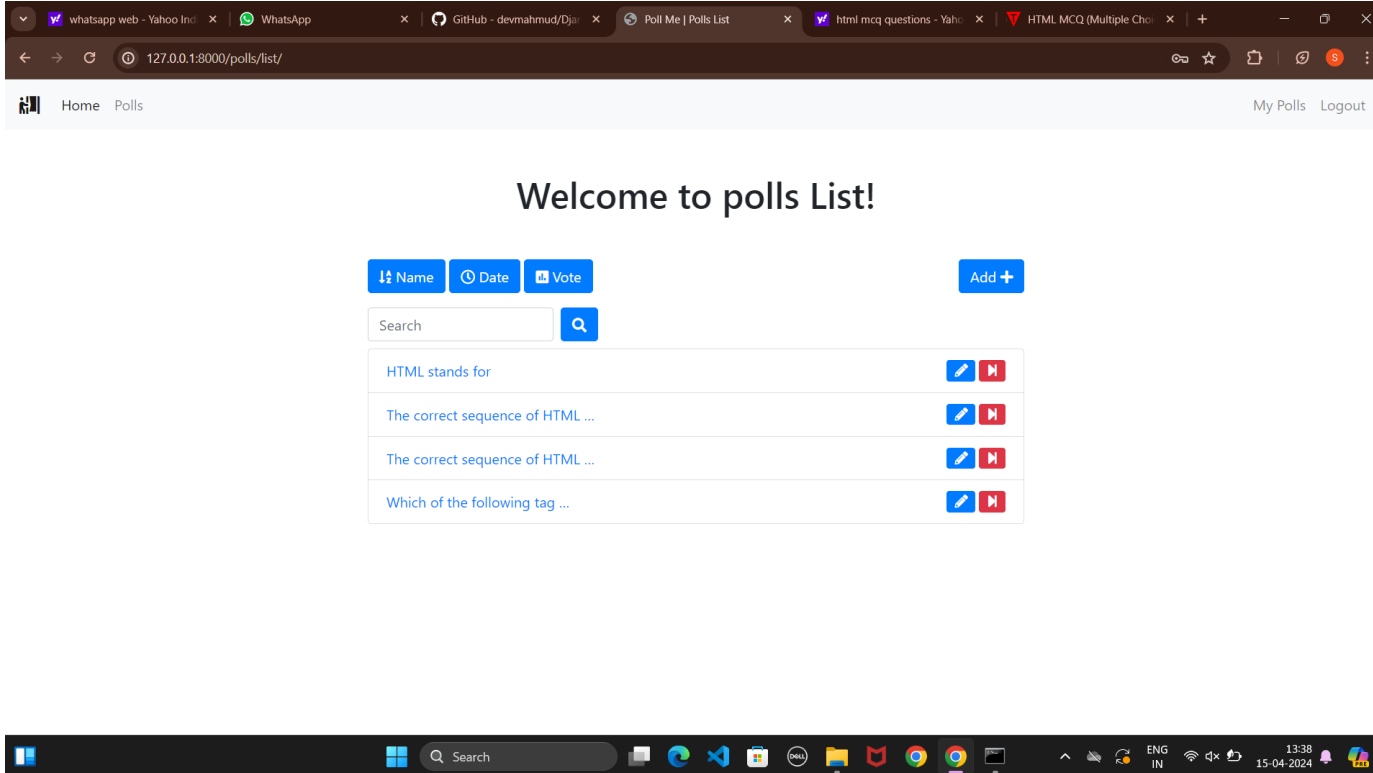
The proposed solution for a voting application using the Django framework is to create a secure and user-friendly online voting platform. The application will allow users to register, vote, and view real-time results. To build the application, the Django framework will be used as the foundation due to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



whatsapp web - Yahoo Inc x WhatsApp x GitHub - devmahmud/Dja x Poll Me | Polls List x html mcq questions - Yaho x HTML MCQ (Multiple Cho x +









127.0.0.1:8000/polls/list/

Home Polls My Polls Logout

Welcome to polls List!

⬆️ Name ⌚ Date 🗳️ Vote Add +

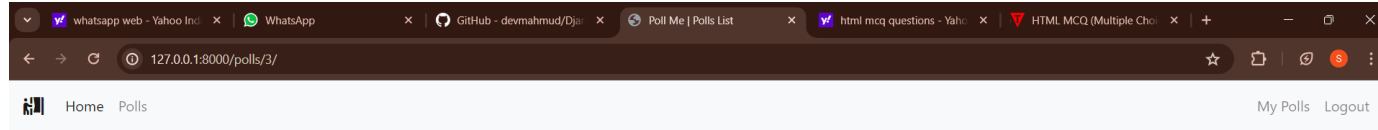
Search 🔍

HTML stands for	 
The correct sequence of HTML ...	 
The correct sequence of HTML ...	 
Which of the following tag ...	 

15

Search ENG IN 13:38 15-04-2024

Voting Page



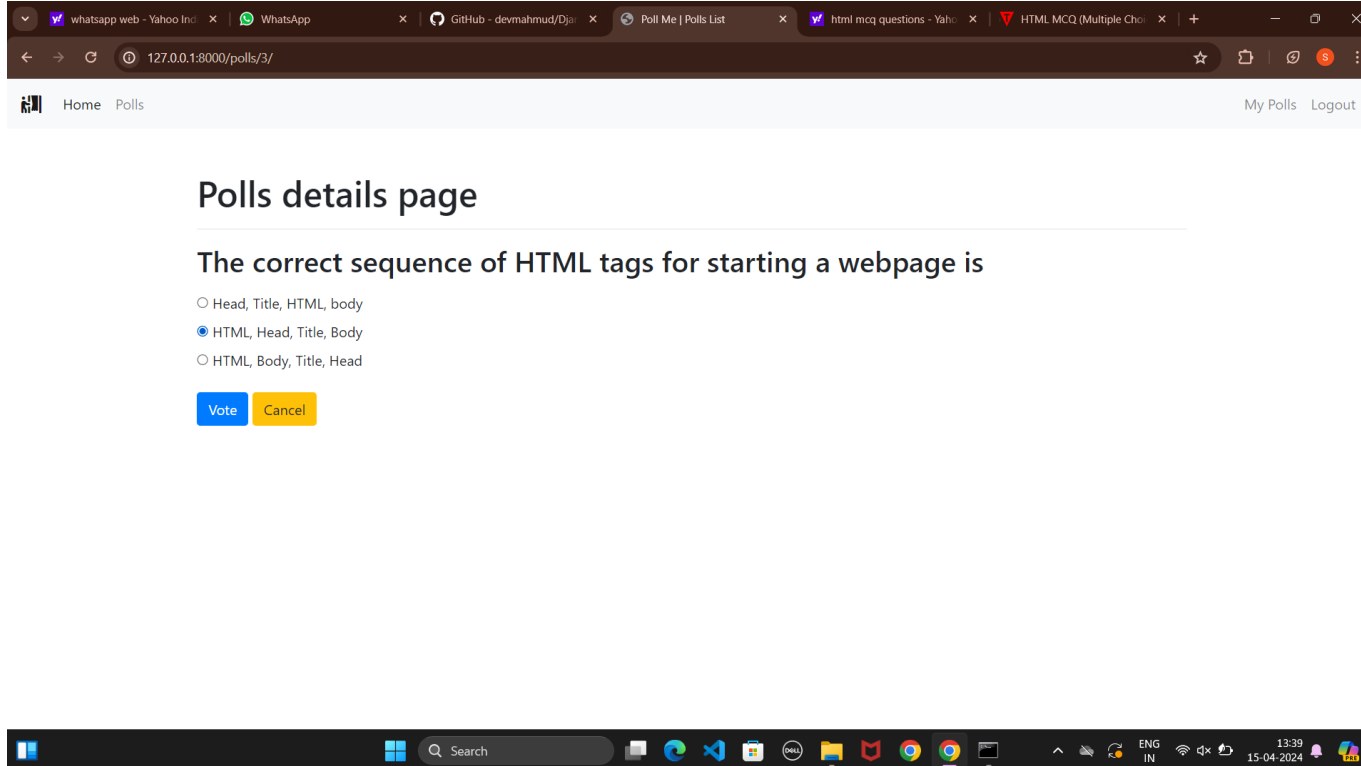
Polls details page

The correct sequence of HTML tags for starting a webpage is

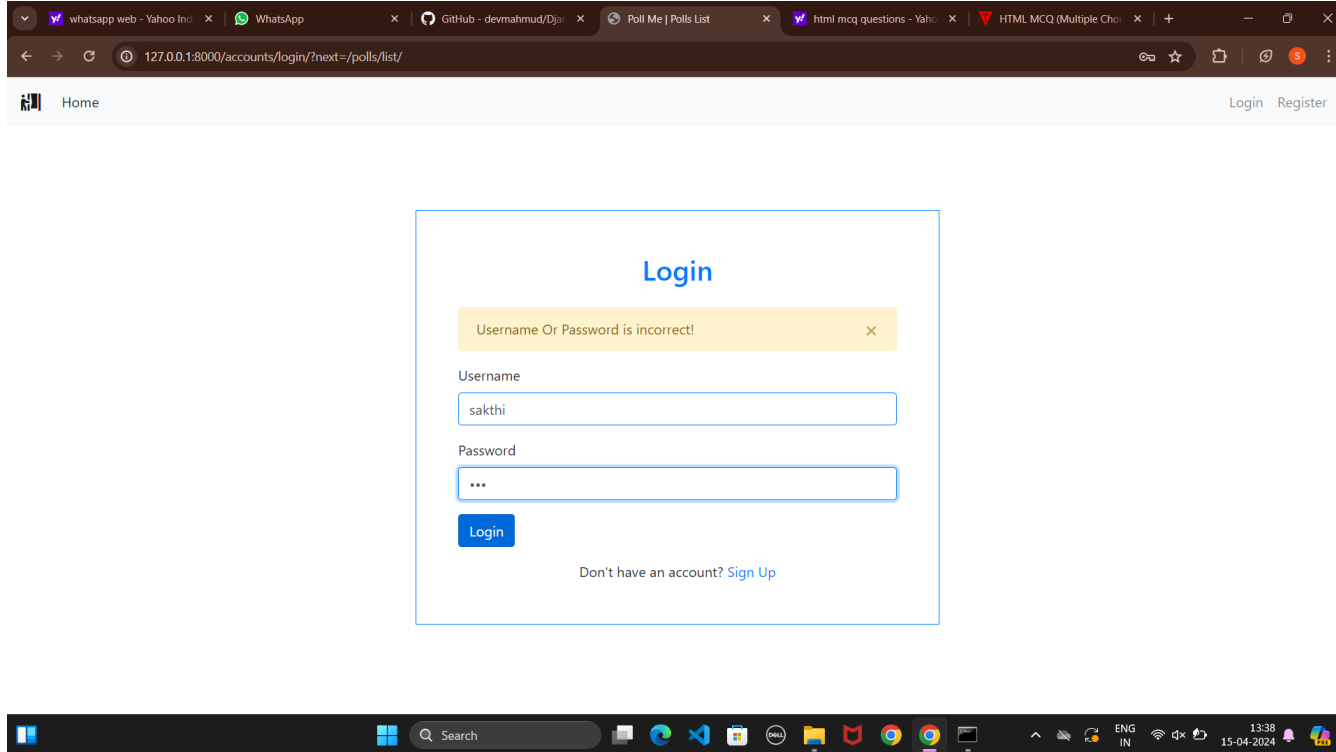
- ☐ Head, Title, HTML, body
- ☐ HTML, Head, Title, Body
- ☐ HTML, Body, Title, Head

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

Voting Details Page



Admin Login Page



whatsapp web - Yahoo Inc. x WhatsApp x GitHub - devmahmud/Djar x Poll Me | Polls List x html mcq questions - Yahoo x HTML MCQ (Multiple Choice) x

127.0.0.1:8000/accounts/login/?next=/polls/list/

Home Login Register

Login

Username Or Password is incorrect! ✕

Username

Password

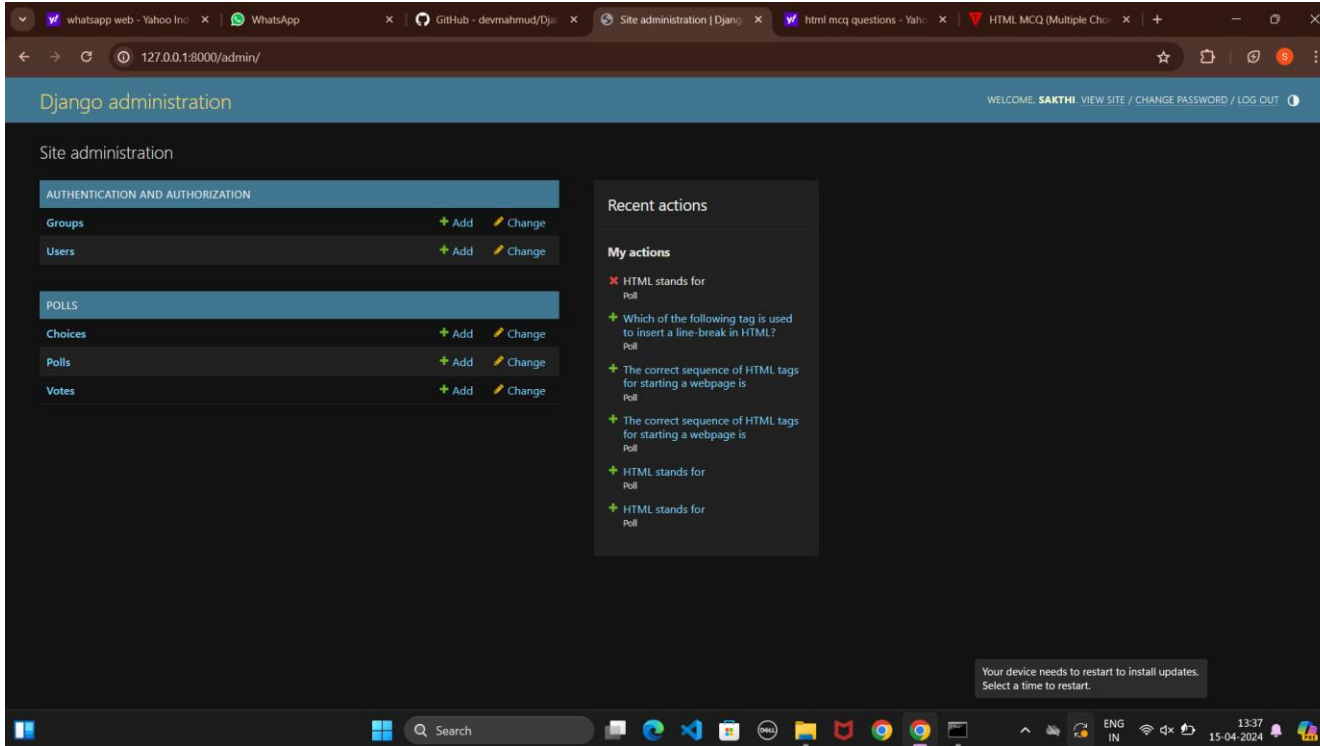
Login

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

Search

ENG IN 13:38 15-04-2024

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 text "Django administration" and a welcome message for "SAKTHI" 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:

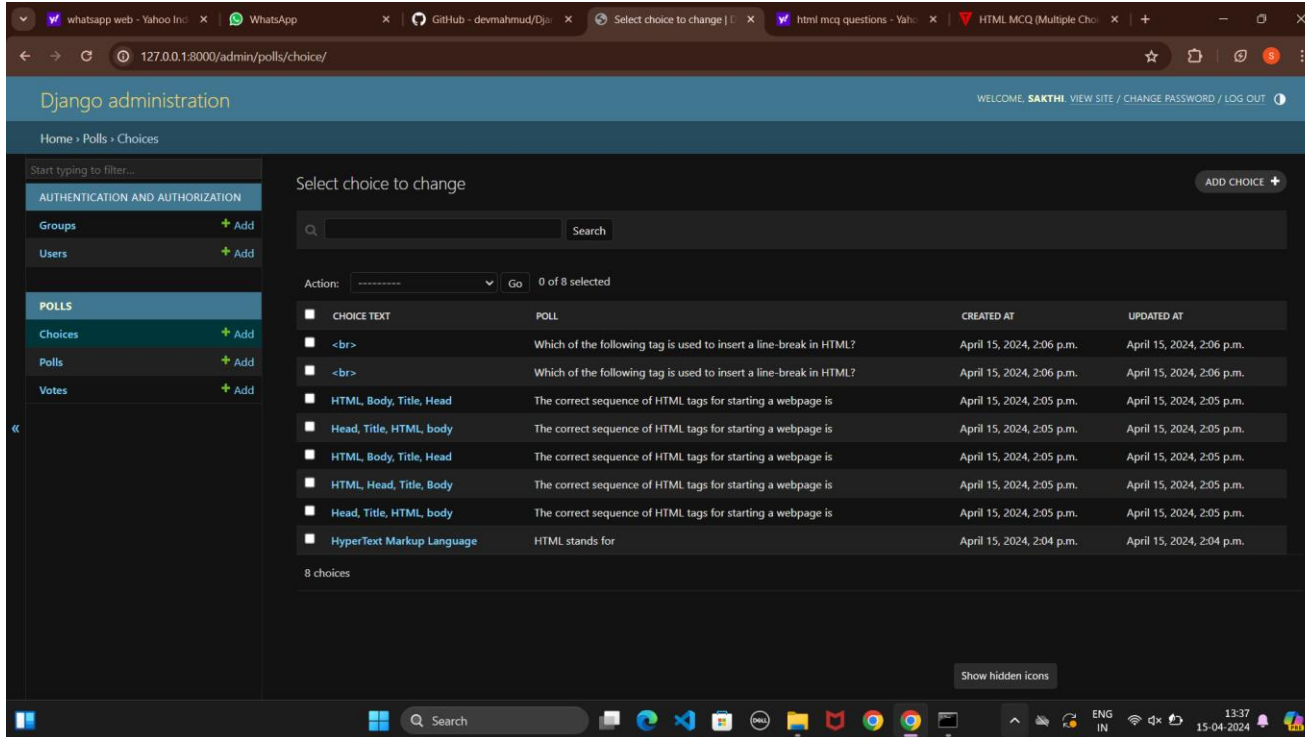
- AUTHENTICATION AND AUTHORIZATION**: Includes links for "Groups" and "Users", each with "Add" and "Change" options.
- POLLS**: Includes links for "Choices", "Polls", and "Votes", each with "Add" and "Change" options.

The right column features a "Recent actions" section with a "My actions" list. The list contains several entries, each with a green plus icon, a description, and a "Poll" label:

- HTML stands for
- Which of the following tag is used to insert a line-break in HTML?
- The correct sequence of HTML tags for starting a webpage is
- The correct sequence of HTML tags for starting a webpage is
- HTML stands for
- HTML stands for

A notification at the bottom right states: "Your device needs to restart to install updates. Select a time to restart." The Windows taskbar at the bottom shows the time as 13:37 on 15-04-2024.

Authentication and Authorization Page



The screenshot shows a web browser window displaying the Django administration interface. The URL is `127.0.0.1:8000/admin/polls/choice/`. The page title is "Django administration". The user is logged in as "SAKTHI" 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](#)

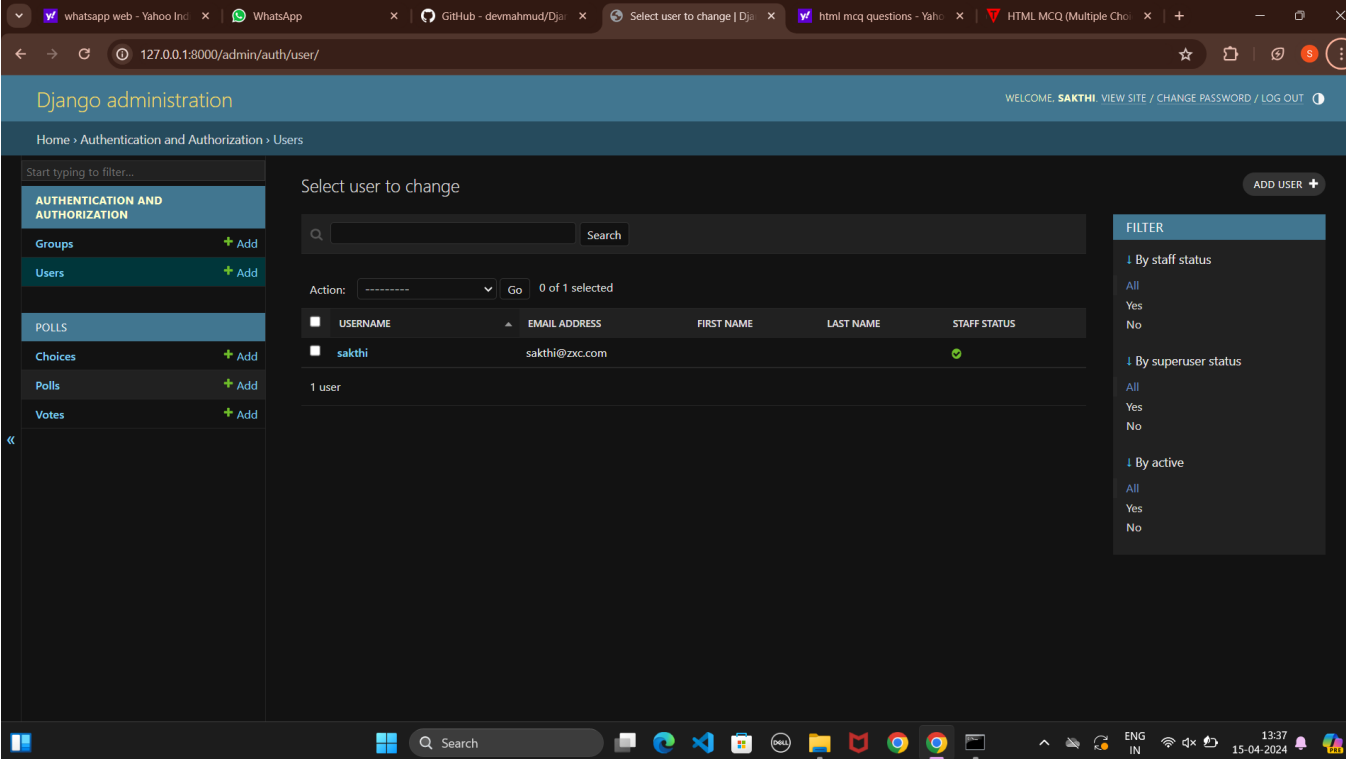
The main content area is titled "Select choice to change". It features a search bar and a table of choices. The table has columns for "CHOICE TEXT", "POLL", "CREATED AT", and "UPDATED AT". There are 8 choices listed.

CHOICE TEXT	POLL	CREATED AT	UPDATED AT
<input type="checkbox"/> <code>
</code>	Which of the following tag is used to insert a line-break in HTML?	April 15, 2024, 2:06 p.m.	April 15, 2024, 2:06 p.m.
<input type="checkbox"/> <code>
</code>	Which of the following tag is used to insert a line-break in HTML?	April 15, 2024, 2:06 p.m.	April 15, 2024, 2:06 p.m.
<input type="checkbox"/> HTML, Body, Title, Head	The correct sequence of HTML tags for starting a webpage is	April 15, 2024, 2:05 p.m.	April 15, 2024, 2:05 p.m.
<input type="checkbox"/> Head, Title, HTML, body	The correct sequence of HTML tags for starting a webpage is	April 15, 2024, 2:05 p.m.	April 15, 2024, 2:05 p.m.
<input type="checkbox"/> HTML, Body, Title, Head	The correct sequence of HTML tags for starting a webpage is	April 15, 2024, 2:05 p.m.	April 15, 2024, 2:05 p.m.
<input type="checkbox"/> HTML, Head, Title, Body	The correct sequence of HTML tags for starting a webpage is	April 15, 2024, 2:05 p.m.	April 15, 2024, 2:05 p.m.
<input type="checkbox"/> Head, Title, HTML, body	The correct sequence of HTML tags for starting a webpage is	April 15, 2024, 2:05 p.m.	April 15, 2024, 2:05 p.m.
<input type="checkbox"/> HyperText Markup Language	HTML stands for	April 15, 2024, 2:04 p.m.	April 15, 2024, 2:04 p.m.

8 choices

At the bottom of the page, there is a "Show hidden icons" button.

Questions Adding Section Page



The screenshot displays the Django administration interface for user management. The browser address bar shows the URL `127.0.0.1:8000/admin/auth/user/`. The page title is "Django administration" and the user is logged in as "SAKTHI". The sidebar on the left contains navigation links for "AUTHENTICATION AND AUTHORIZATION" (Groups, Users) and "POLLS" (Choices, Polls, Votes). The main content area is titled "Select user to change" and includes a search bar, a table of users, and a filter sidebar on the right.

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

Home > Authentication and Authorization > Users

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

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

POLLS

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

Select user to change

ADD USER [+](#)

Search

Action: [-----](#) Go 0 of 1 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input checked="" type="checkbox"/>	sakthi	sakthi@zxc.com			

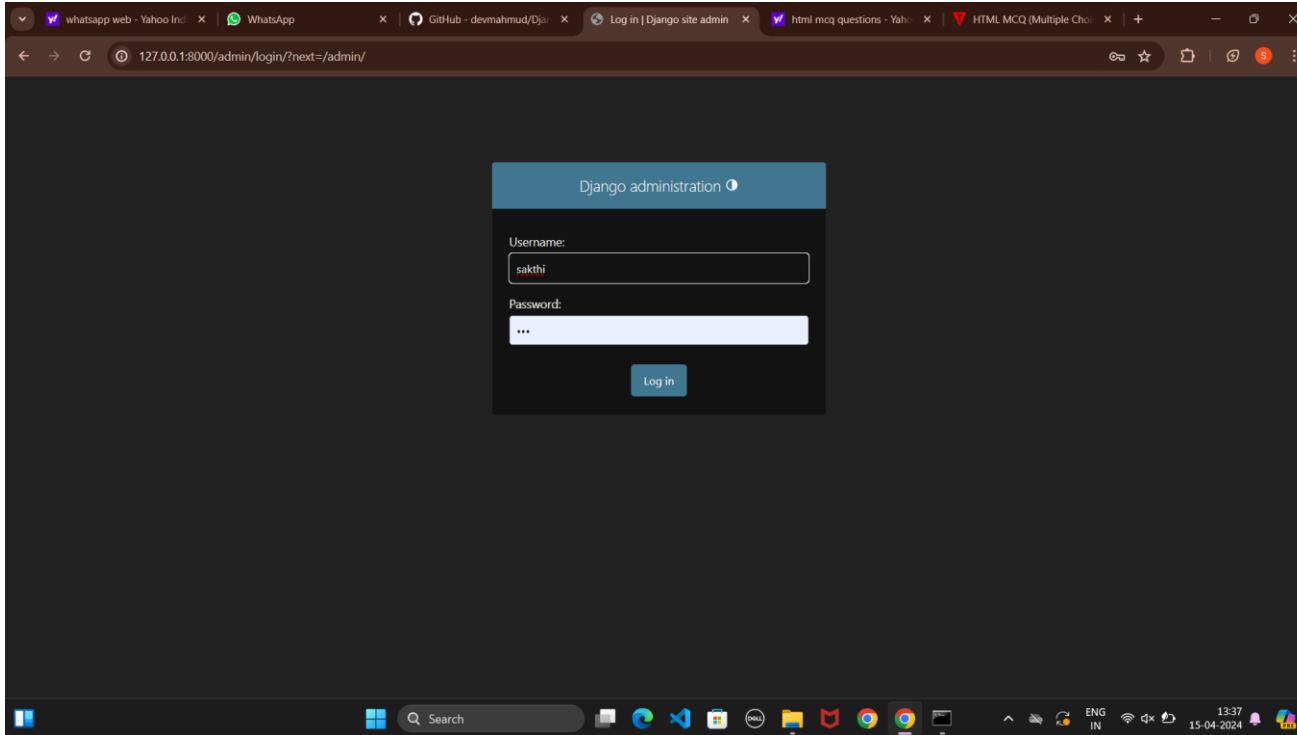
1 user

FILTER

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

Windows taskbar: Search, File Explorer, Edge, VS Code, Outlook, Mail, Chrome, 13:37, 15-04-2024

Admin Login Page



The screenshot shows a web browser window with the URL `127.0.0.1:8000/admin/login/?next=/admin/`. The page displays the Django administration login interface. At the top, there is a header bar with the text "Django administration" and a small information icon. Below this, the "Username:" label is followed by a text input field containing the value "sakthi". The "Password:" label is followed by a password input field with masked characters "...". A "Log in" button is positioned below the password field. The browser's address bar and tabs are visible at the top, and the Windows taskbar is at the bottom.

Django administration ⓘ

Username:

Password:

Log in

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!