



NEXT GEN EMPLOYABILITY PROGRAM

| Creating a future-ready workforce

Student Name :Varsha G
Student ID :au820621104086

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Voting Application using Django Framework-Varsha(4086,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

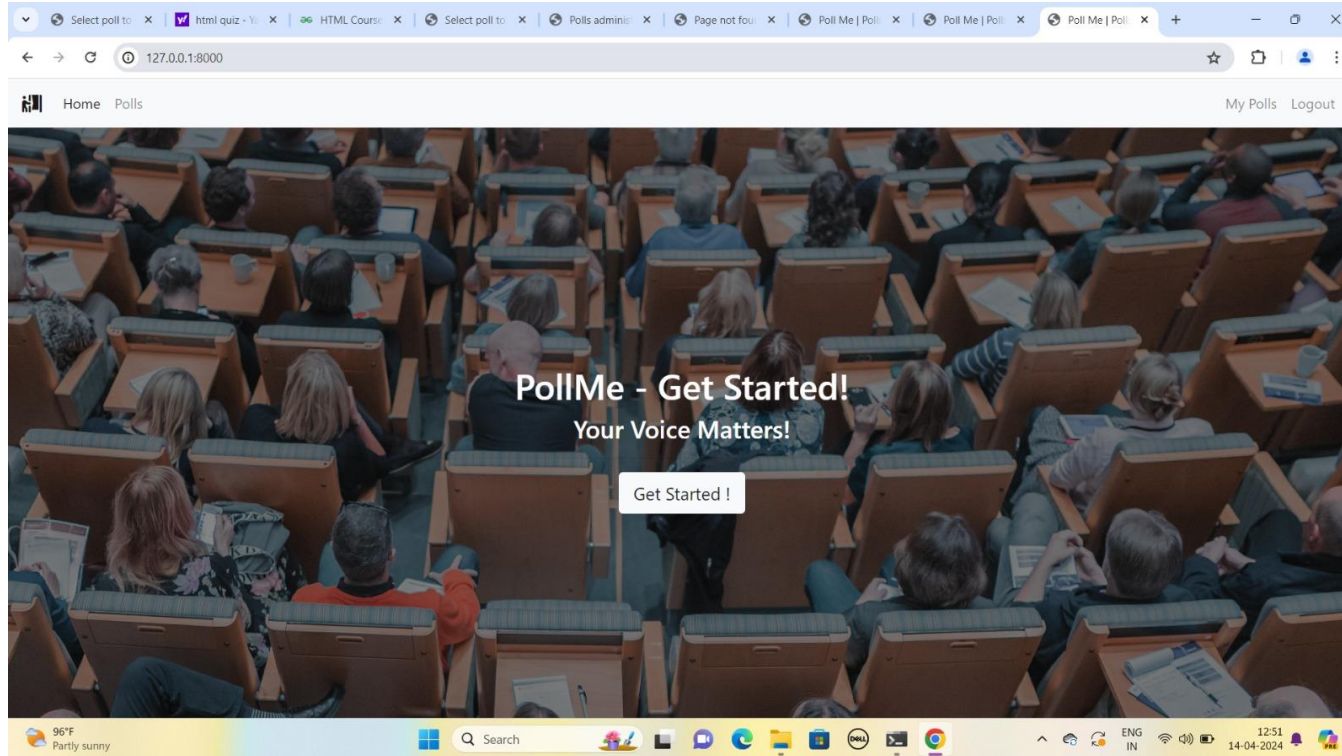
The project overview for a voting application using the Django framework involves creating a secure and user-friendly online voting system. The application allows users to register, vote, and view real-time results. Here is a steps involved in building the voting application:

- 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.

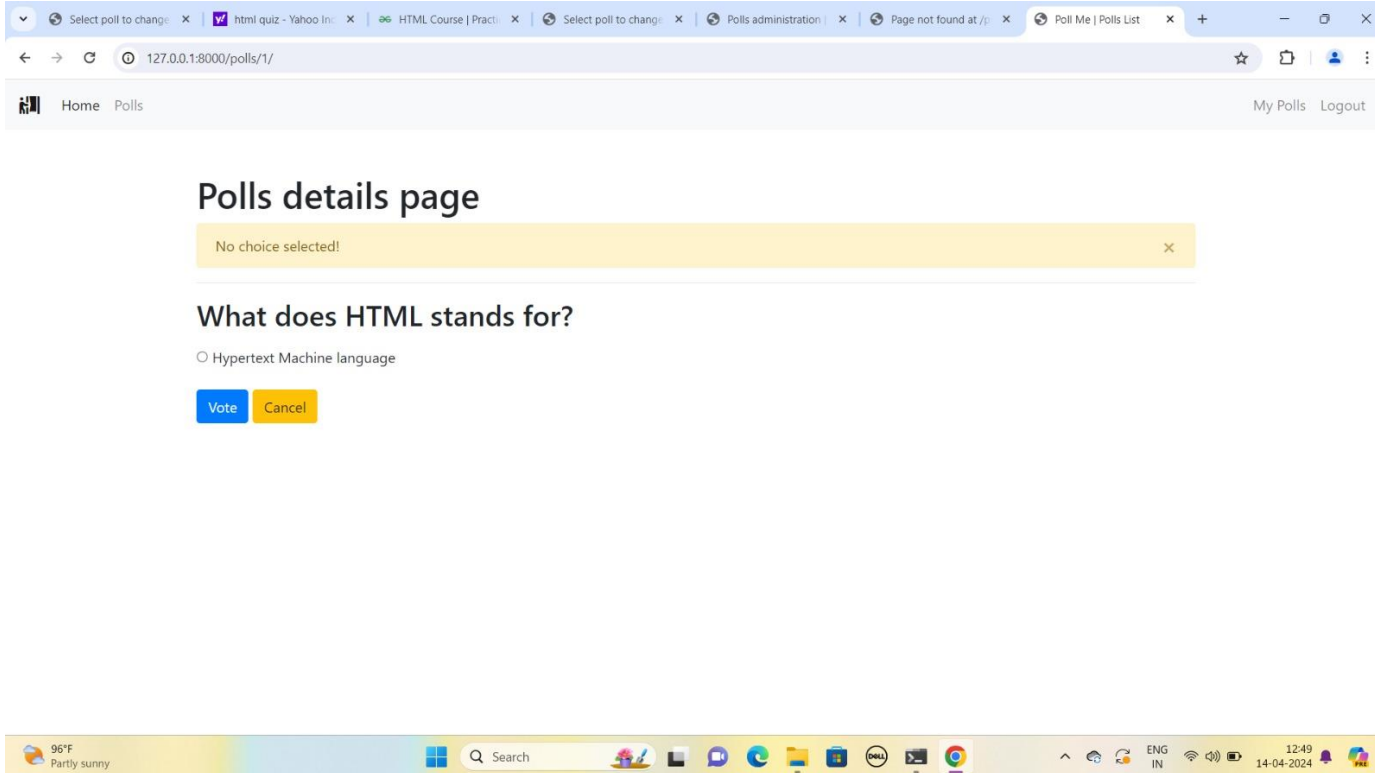
Proposed Solution

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



Select poll to change x html quiz - Yahoo Inc x HTML Course | Practi x Select poll to change x Polls administration x Page not found at /p x Poll Me | Polls List x

127.0.0.1:8000/polls/1/

Home Polls My Polls Logout

Polls details page

No choice selected! x

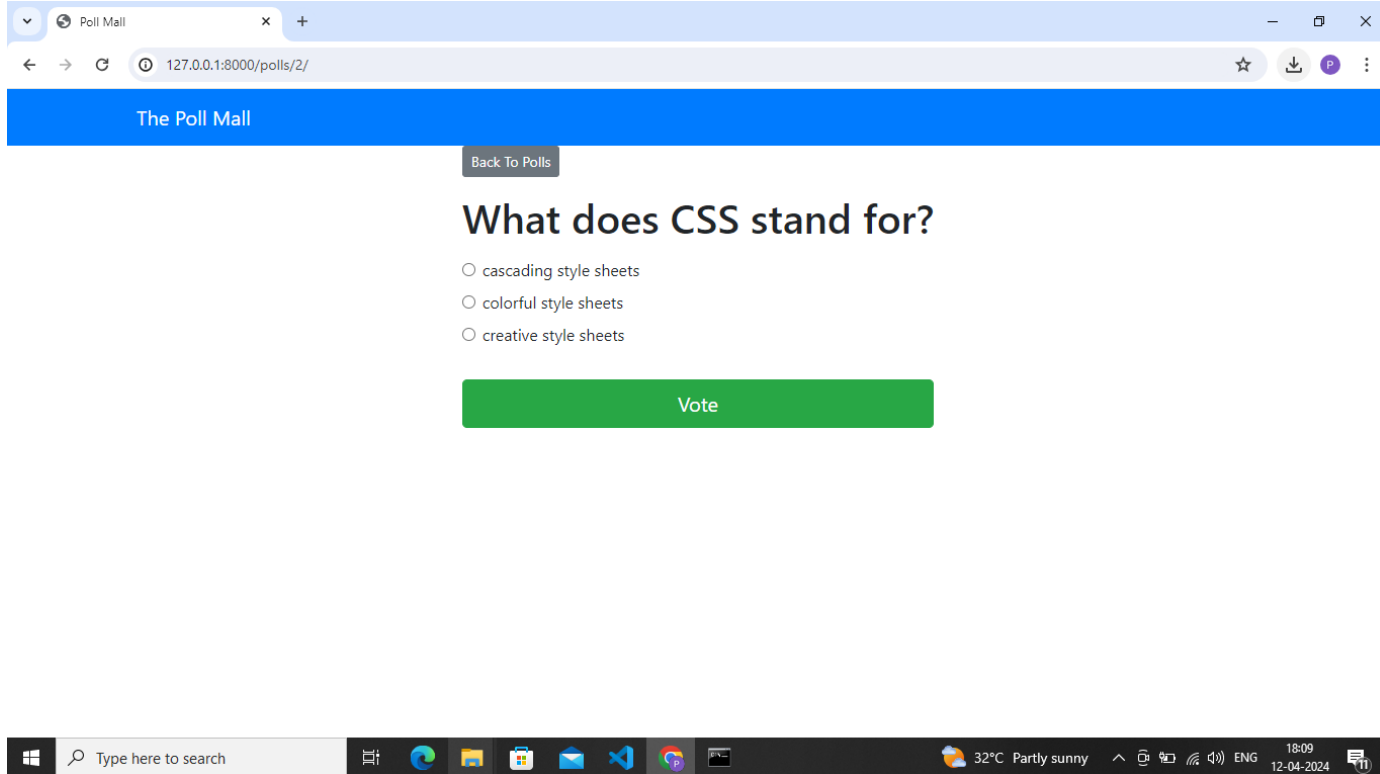
What does HTML stands for?

☐ Hypertext Machine language

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

95°F Partly sunny Search 12:49 14-04-2024

Voting Page



Poll Mall

127.0.0.1:8000/polls/2/

The Poll Mall

[Back To Polls](#)

What does CSS stand for?

☐ cascading style sheets

☐ colorful style sheets

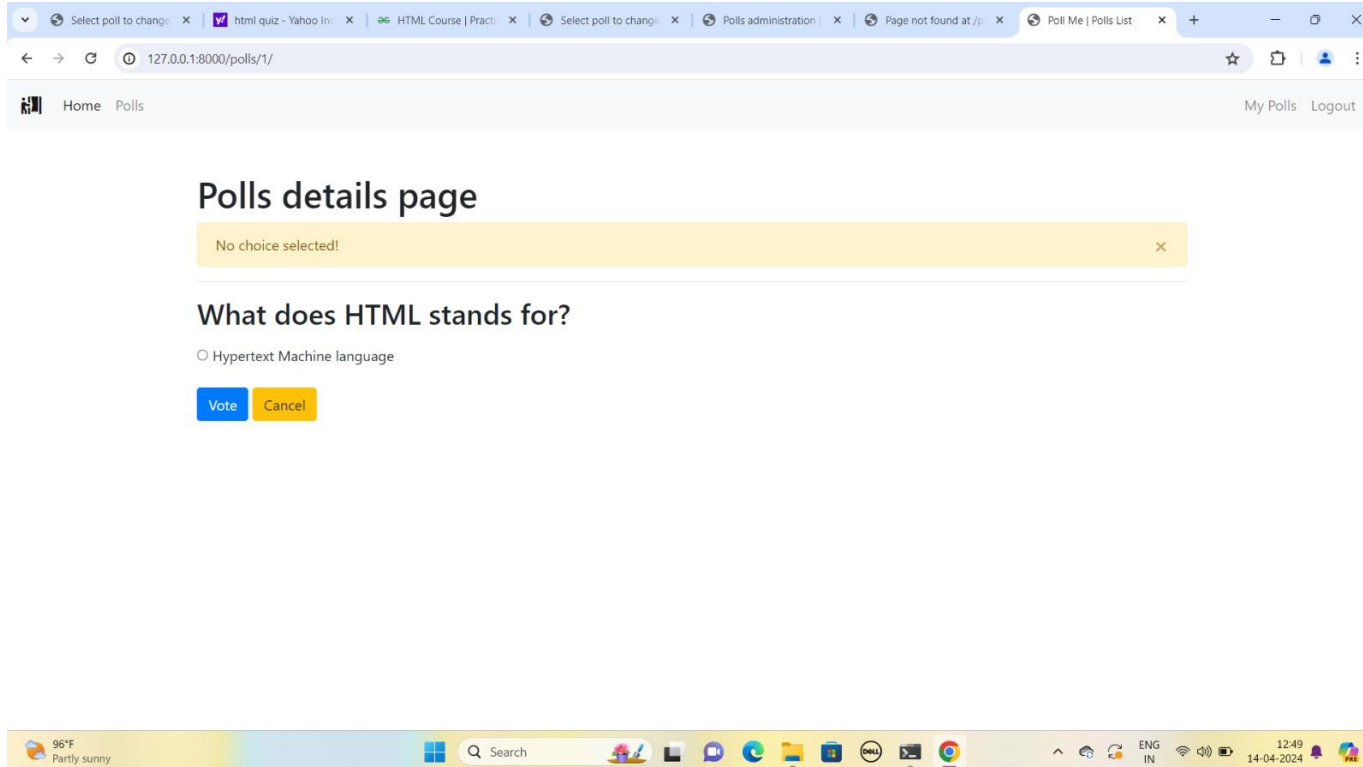
☐ creative style sheets

[Vote](#)

Type here to search

32°C Partly sunny 18:09 12-04-2024

Voting Details Page



Select poll to change x | html quiz - Yahoo In: x | HTML Course | Practi: x | Select poll to change: x | Polls administration | x | Page not found at /: x | Poll Me | Polls List x

127.0.0.1:8000/polls/1/

Home Polls My Polls Logout

Polls details page

No choice selected!

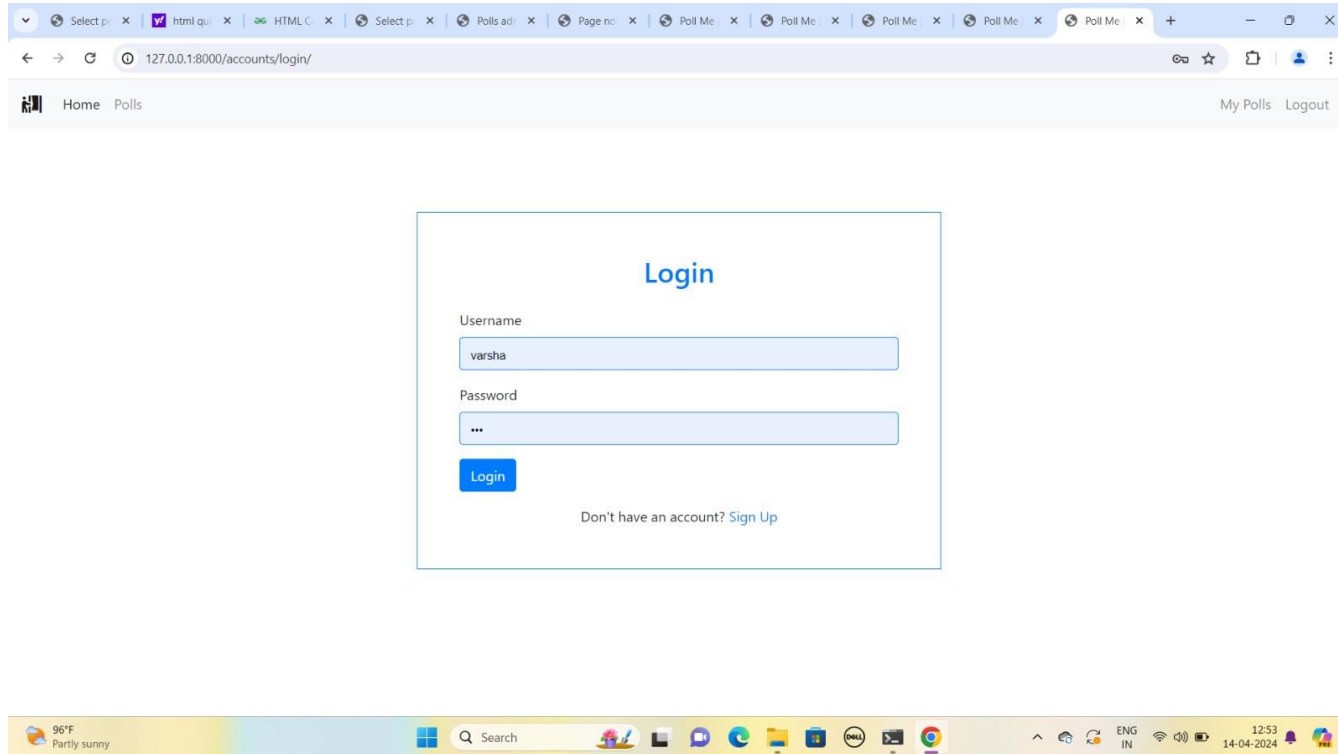
What does HTML stands for?

☐ Hypertext Machine language

Vote Cancel

96°F Partly sunny Search 12:49 14-04-2024

Admin Login Page



The screenshot shows a web browser window with the URL `127.0.0.1:8000/accounts/login/`. The browser's address bar and tabs are visible at the top. Below the browser window, the login form is displayed within a light blue border. The form has a title "Login" in blue text. It contains two input fields: "Username" with the value "varsha" and "Password" with masked characters "***". A blue "Login" button is positioned below the password field. At the bottom of the form, there is a link that says "Don't have an account? [Sign Up](#)".

127.0.0.1:8000/accounts/login/

Home Polls My Polls Logout

Login

Username

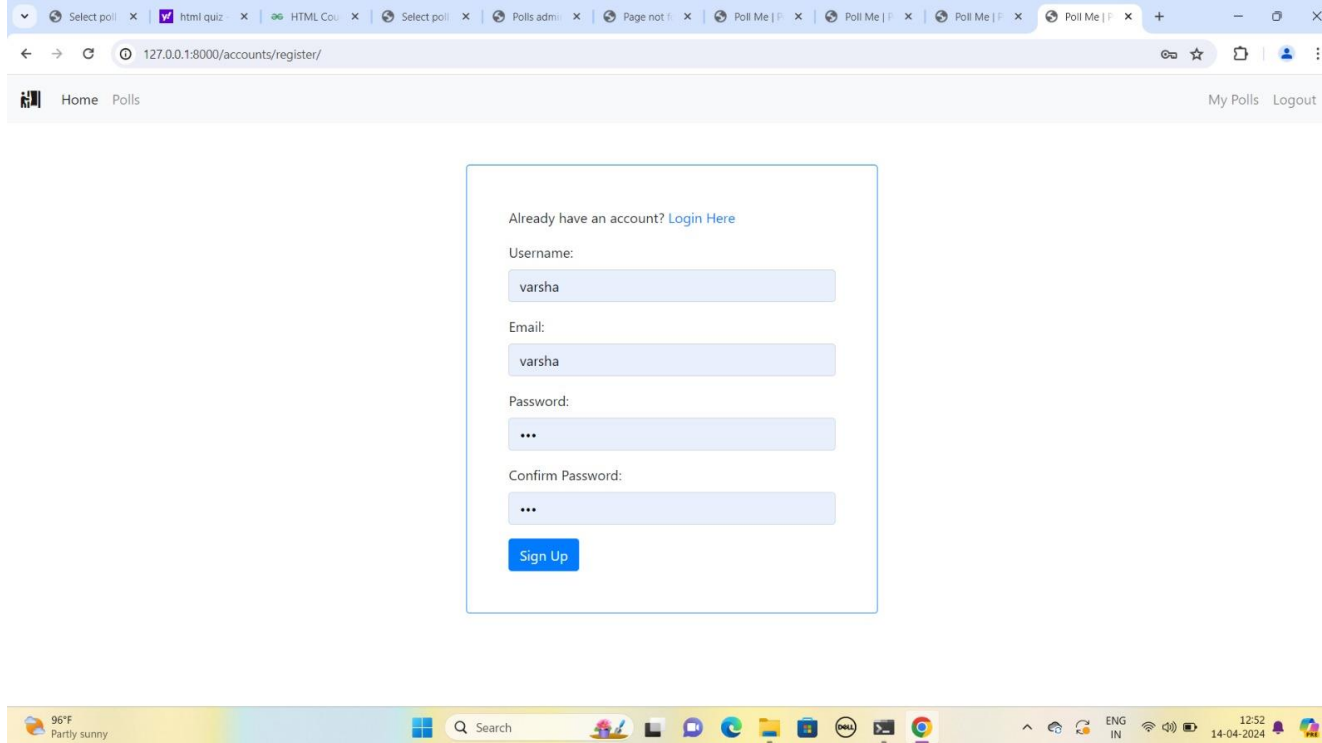
Password

Login

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

96°F Partly sunny Search ENG IN 12:53 14-04-2024

Admin Home Page



The screenshot shows a web browser window with the URL `127.0.0.1:8000/accounts/register/`. The browser's tab bar contains several tabs, including "Select poli", "html quiz", "HTML Cou", "Select poli", "Polls admin", "Page not f", "Poll Me | F", "Poll Me | F", "Poll Me | F", and "Poll Me | F". The browser's address bar shows the URL `127.0.0.1:8000/accounts/register/`. The browser's navigation bar includes a home icon, a "Home" link, a "Polls" link, and a "My Polls" link. The browser's status bar shows the temperature `96°F`, the weather `Partly sunny`, the time `12:52`, and the date `14-04-2024`.

Home Polls My Polls Logout

Already have an account? [Login Here](#)

Username:

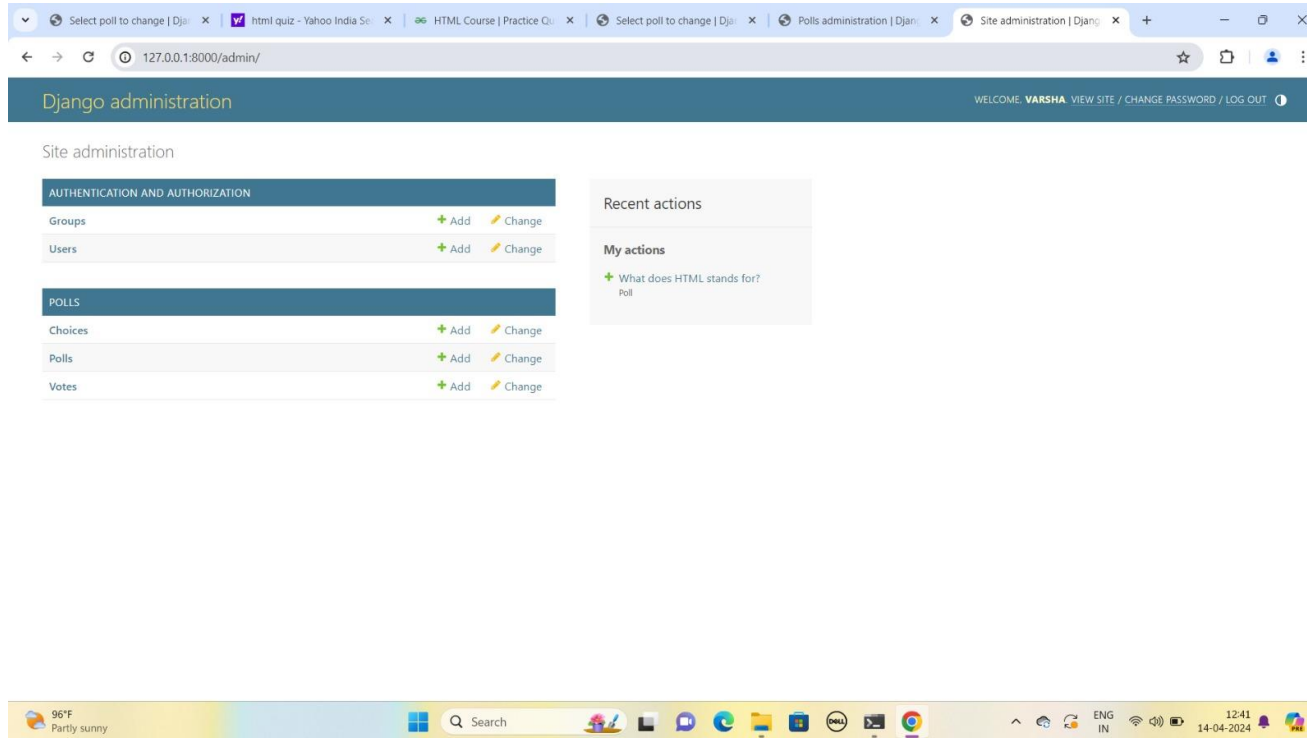
Email:

Password:

Confirm Password:

[Sign Up](#)

Authentication and Authorization 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 title is "Django administration", and a welcome message for "VARSHA" is visible in the top right corner, along with links for "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT".

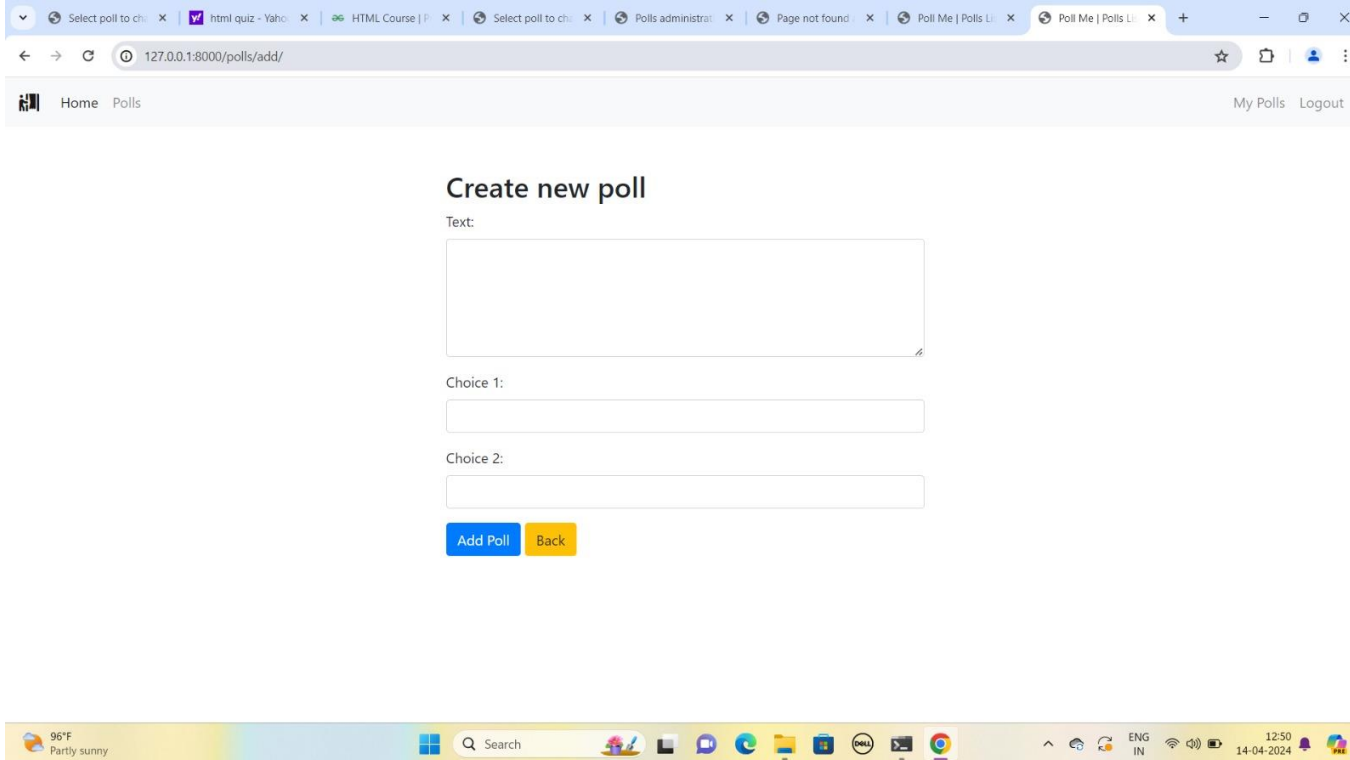
The main content area is titled "Site administration" and is divided into two primary sections:

- AUTHENTICATION AND AUTHORIZATION**: This section contains two sub-sections:
 - Groups**: Includes links for "Add" and "Change".
 - Users**: Includes links for "Add" and "Change".
- POLLS**: This section contains three sub-sections:
 - Choices**: Includes links for "Add" and "Change".
 - Polls**: Includes links for "Add" and "Change".
 - Votes**: Includes links for "Add" and "Change".

On the right side of the page, there is a "Recent actions" section and a "My actions" section. The "My actions" section lists a single action: "What does HTML stands for?" with a sub-entry "poll".

The Windows taskbar at the bottom of the screen shows the system time as 12:41 on 14-04-2024, along with various system icons and the weather forecast (96°F, Partly sunny).

Questions Adding Section Page



Select poll to ch... x | html quiz - Yahoo... x | HTML Course | P... x | Select poll to ch... x | Polls administrat... x | Page not found... x | Poll Me | Polls Li... x | Poll Me | Polls Li... x

127.0.0.1:8000/polls/add/

Home Polls My Polls Logout

Create new poll

Text:

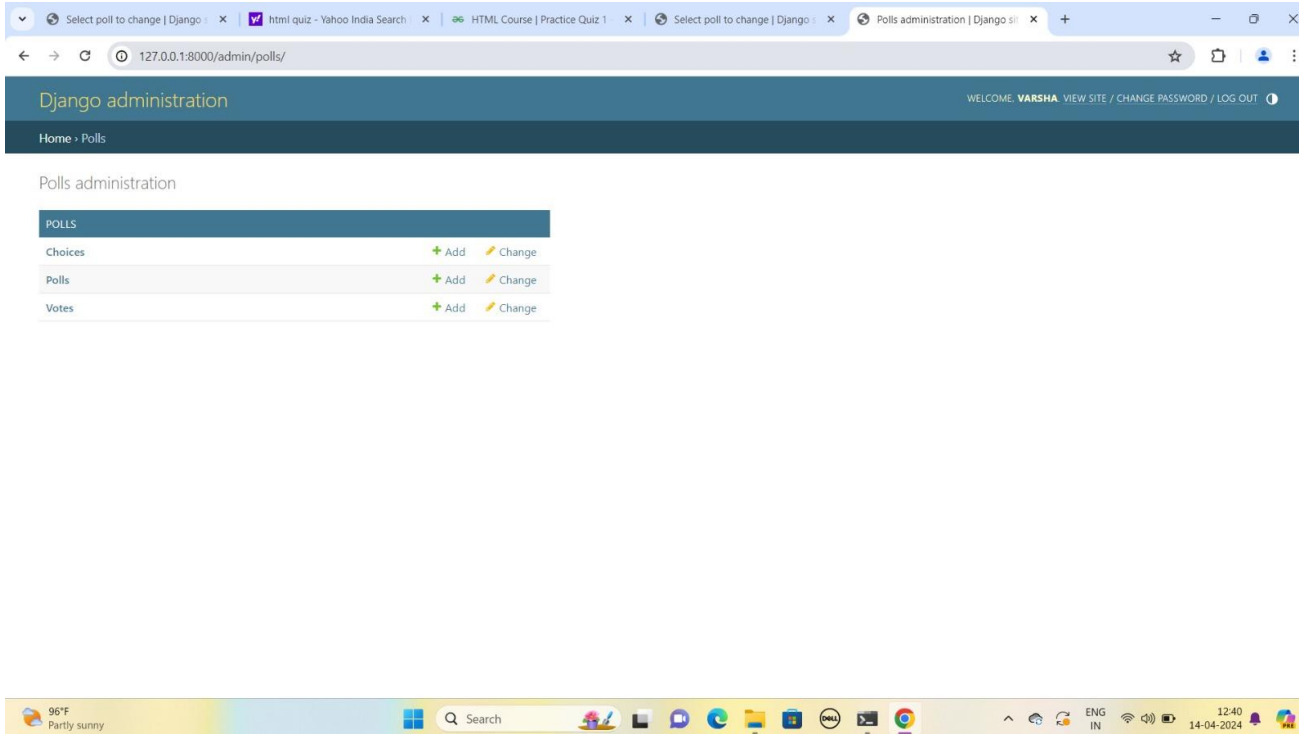
Choice 1:

Choice 2:

Add Poll Back

96°F Partly sunny Search 14-04-2024 12:50

Voting Details Page



The screenshot shows a web browser window with multiple tabs. The active tab is 'Polls administration | Django site'. The address bar shows the URL '127.0.0.1:8000/admin/polls/'. The page header is 'Django administration' with a welcome message for 'VARSHA' and links for 'VIEW SITE', 'CHANGE PASSWORD', and 'LOG OUT'. The breadcrumb trail is 'Home > Polls'. The main content area is titled 'Polls administration' and contains a table with the following data:

POLLS	
Choices	+ Add Change
Polls	+ Add Change
Votes	+ Add Change

The Windows taskbar at the bottom shows the date and time as 12:40 on 14-04-2024, along with various system icons and the search bar.

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!