



NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready
workforce

Student Name :Nirmal kumar S
Student ID :au820621104059

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Voting Application using Django Framework-Nirmal kumar S(820621104059,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

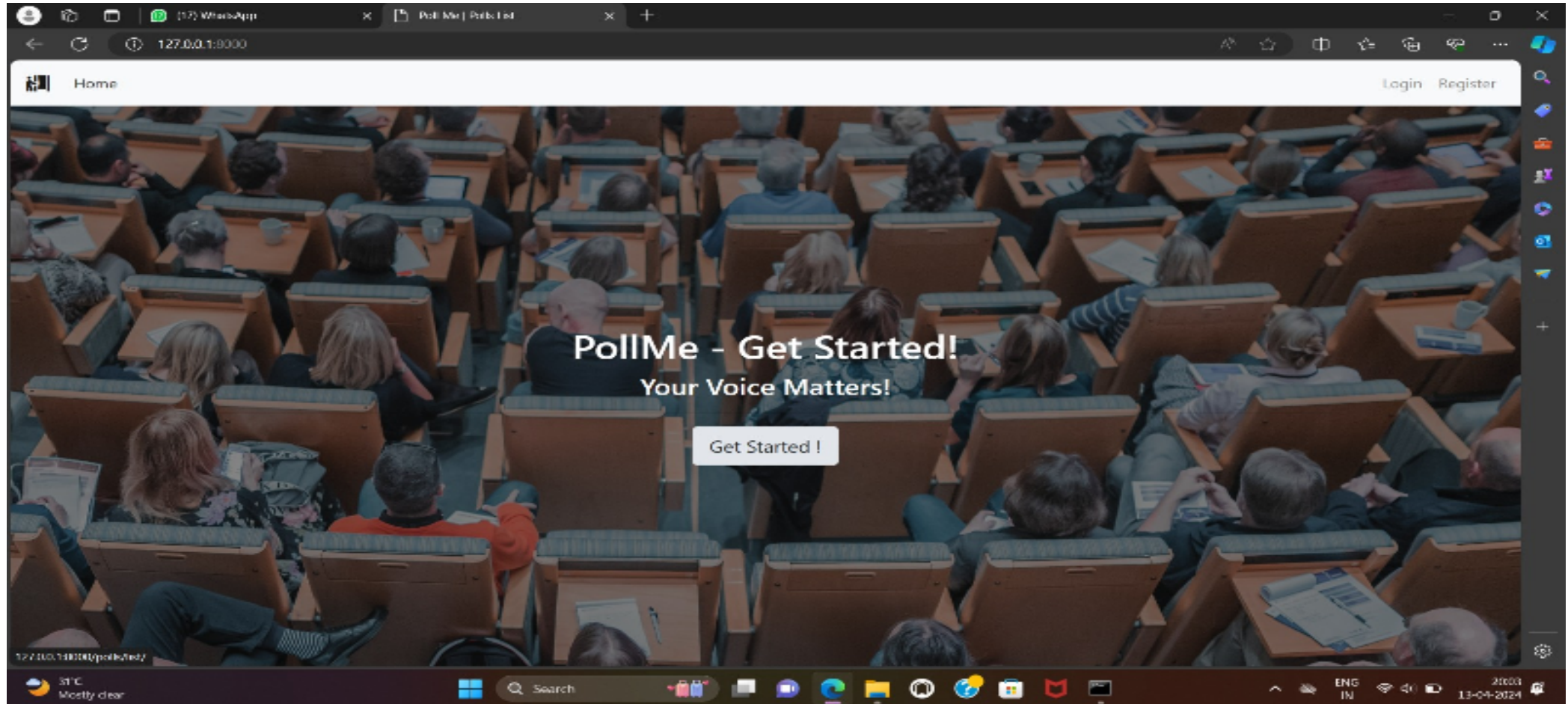
The project overview for a voting application using the Django framework involves creating a secure and user-friendly online voting system. The application allows users to register, vote, and view real-time results. Here is a steps involved in building the voting application:

- 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.

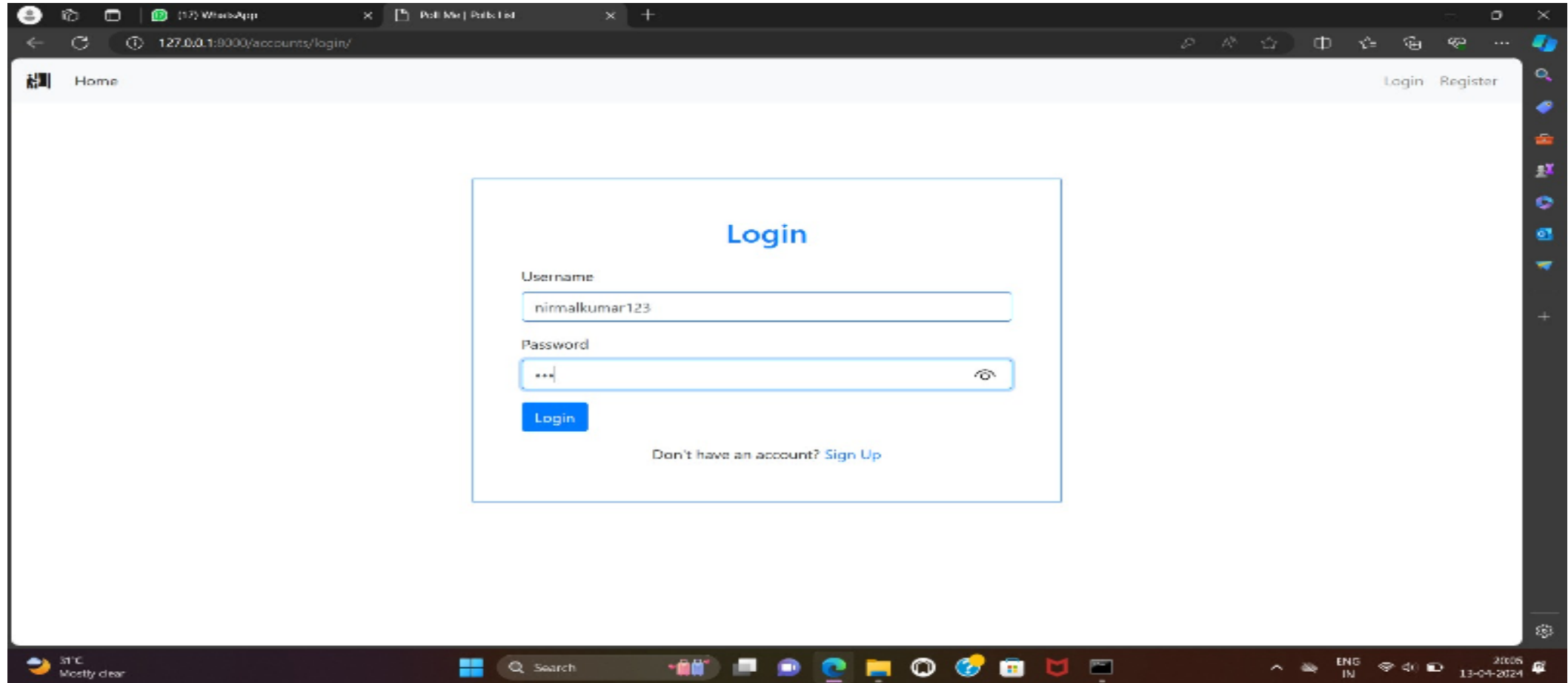
Proposed Solution

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



Admin Login Page



The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/accounts/login/`. The page has a light gray header with a "Home" link on the left and "Login" and "Register" links on the right. The main content area features a central "Login" form with a blue border. Inside the form, the title "Login" is centered in blue. Below it, there are two input fields: "Username" with the value "nirmalkumar123" 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". The browser's taskbar at the bottom shows the system clock as 2:00 PM on 13-04-2024, along with various application icons and a search bar.

Home Login Register

Login

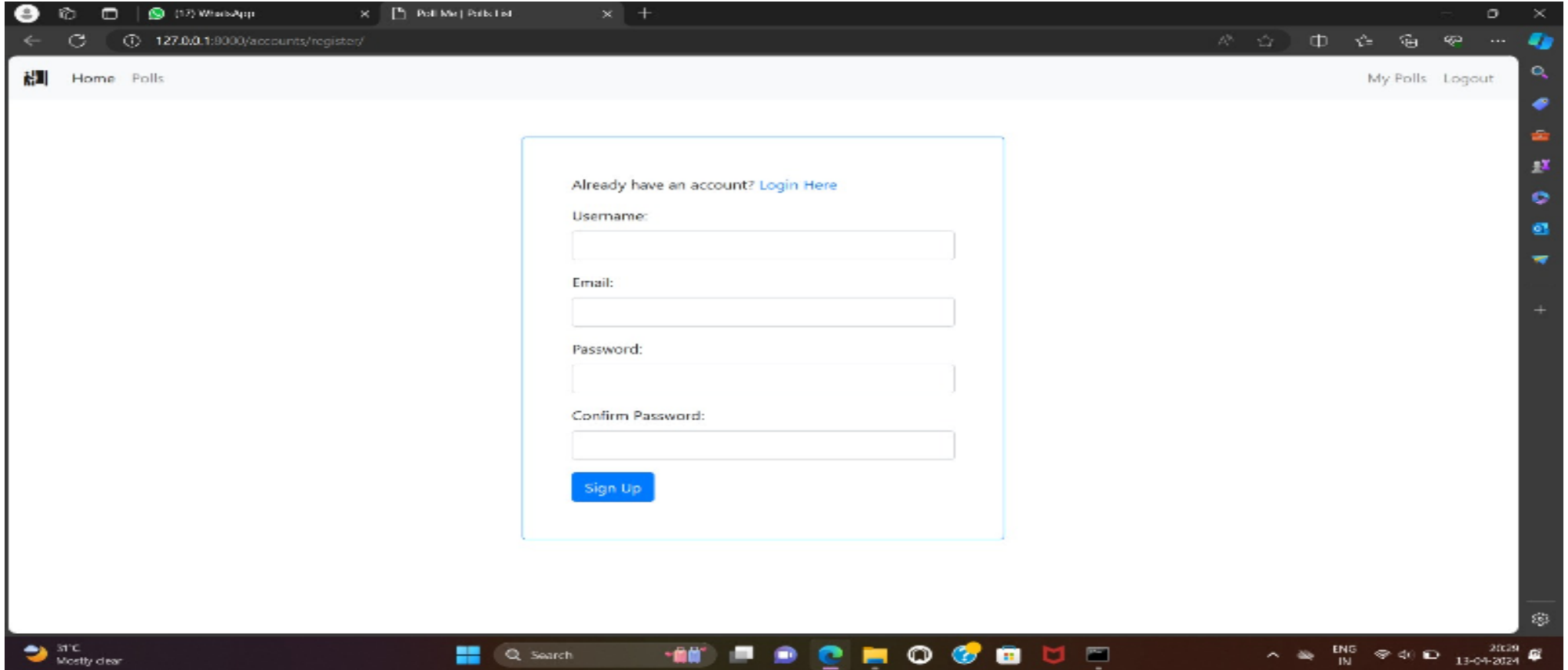
Username
nirmalkumar123

Password

Login

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

home login

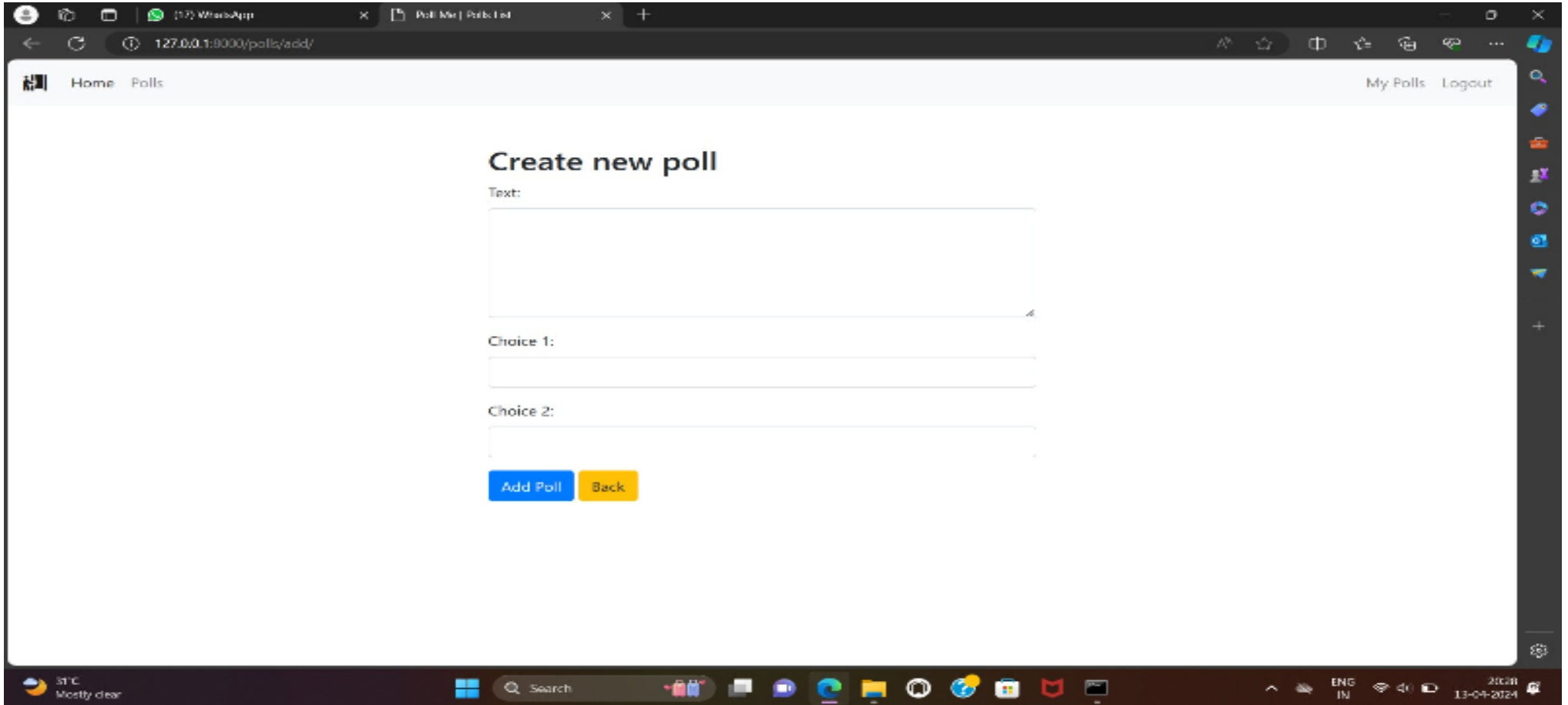


The screenshot shows a web browser window with the address bar displaying `127.0.0.1:9000/accounts/register/`. The page has a navigation bar with 'Home' and 'Polls' on the left, and 'My Polls' and 'Logout' on the right. The main content area features a registration form with the following elements:

- A link: "Already have an account? [Login Here](#)"
- A label: "Username:" followed by a text input field.
- A label: "Email:" followed by a text input field.
- A label: "Password:" followed by a text input field.
- A label: "Confirm Password:" followed by a text input field.
- A blue button labeled "Sign Up" at the bottom of the form.

The Windows taskbar at the bottom shows the system clock as 21:29 on 13-04-2024, with a temperature of 31°C and the text "Mostly clear".

create new poll

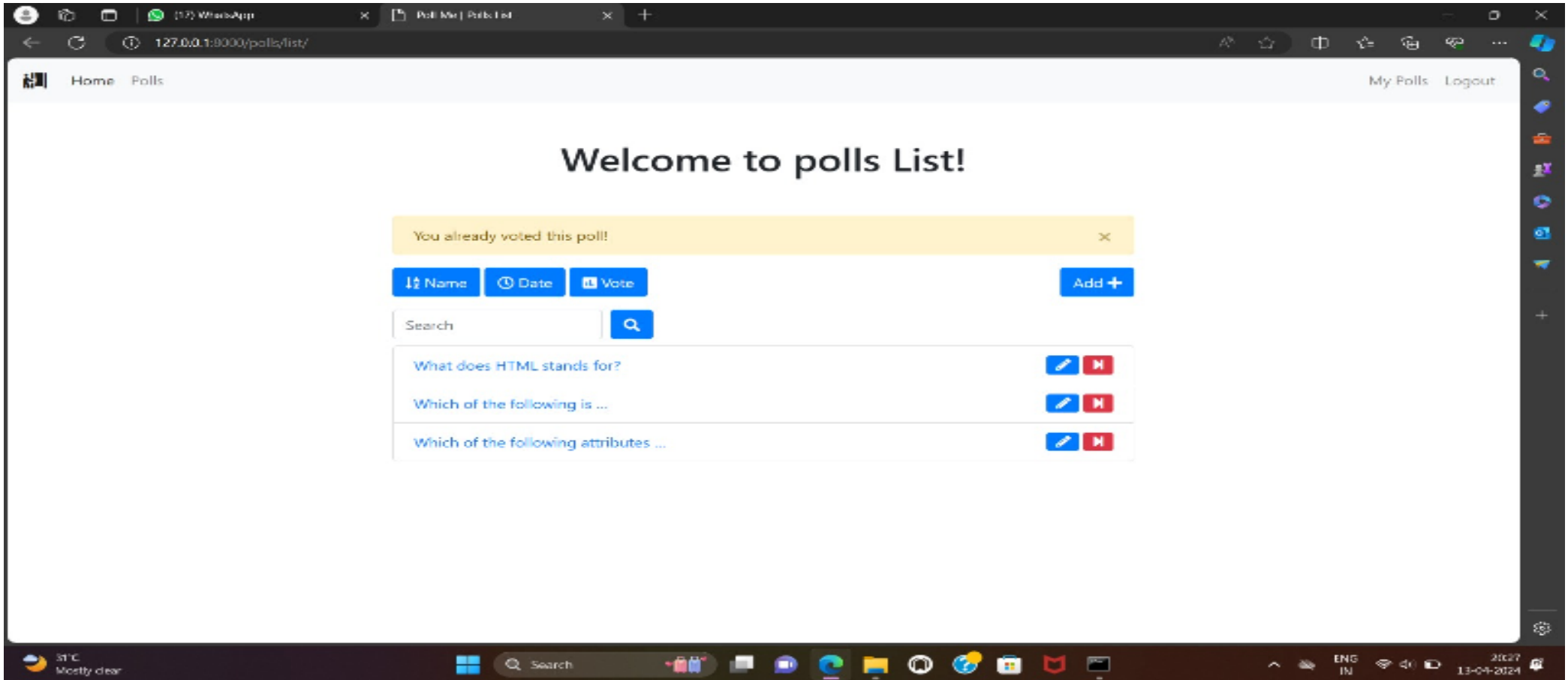


The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/polls/add/`. The browser has two tabs: "(17) WhatsApp" and "Poll Me | Polls List". The website's navigation bar includes "Home" and "Polls" links on the left, and "My Polls" and "Logout" on the right. The main content area is titled "Create new poll" and contains the following form elements:

- A "Text:" label followed by a large text input field.
- A "Choice 1:" label followed by a single-line text input field.
- A "Choice 2:" label followed by a single-line text input field.
- At the bottom, two buttons: a blue "Add Poll" button and a yellow "Back" button.

The Windows taskbar at the bottom shows the system clock as 2:02 PM on 13-04-2024, with the language set to ENG IN.

Poll Page

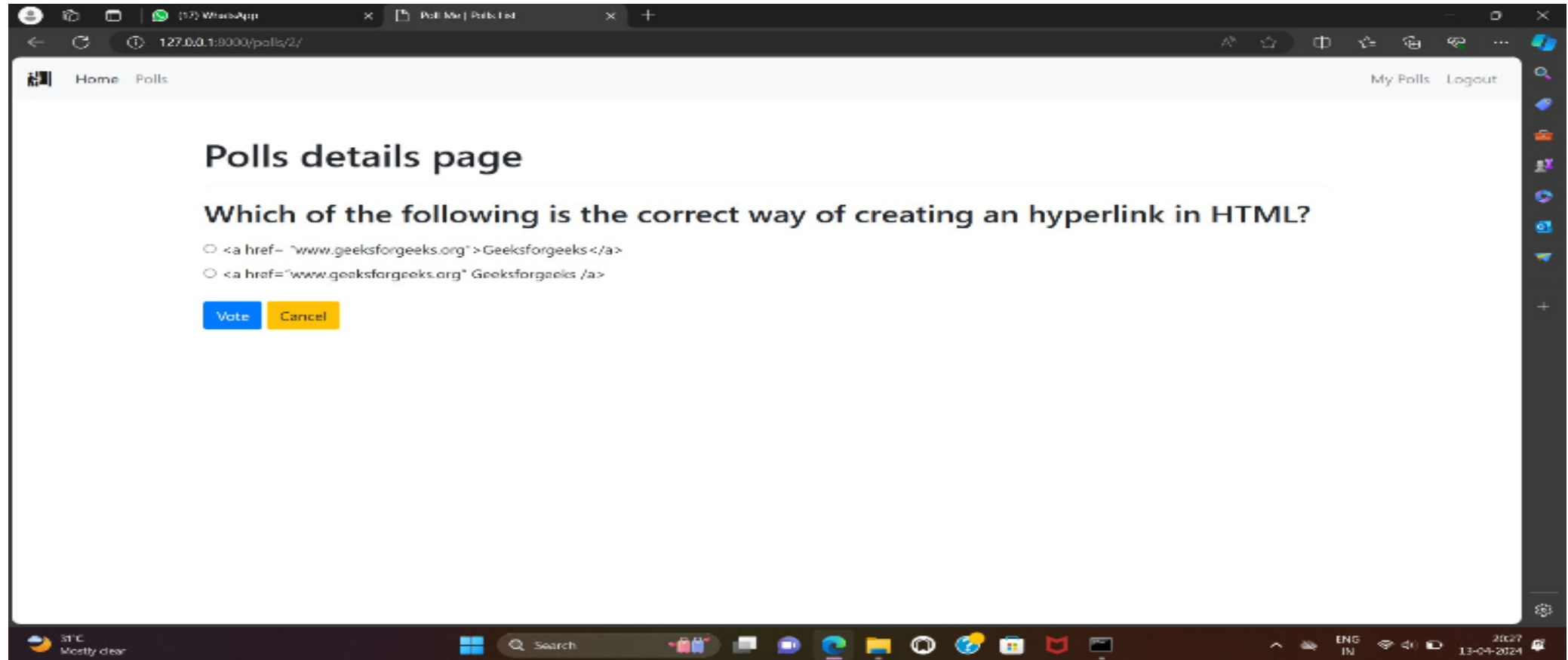


The screenshot shows a web browser window with the address bar displaying `127.0.0.1:8000/polls/list/`. The browser has two tabs: "(17) WhatsApp" and "Poll Me | Polls List". The website's navigation bar includes "Home" and "Polls" links, with "My Polls" and "Logout" on the right. The main heading is "Welcome to polls List!". A yellow notification box states "You already voted this poll!". Below this are three filter buttons: "Name", "Date", and "Vote", followed by an "Add +" button. A search bar with a magnifying glass icon is present. The poll list contains three items, each with an edit icon (pencil) and a delete icon (X):

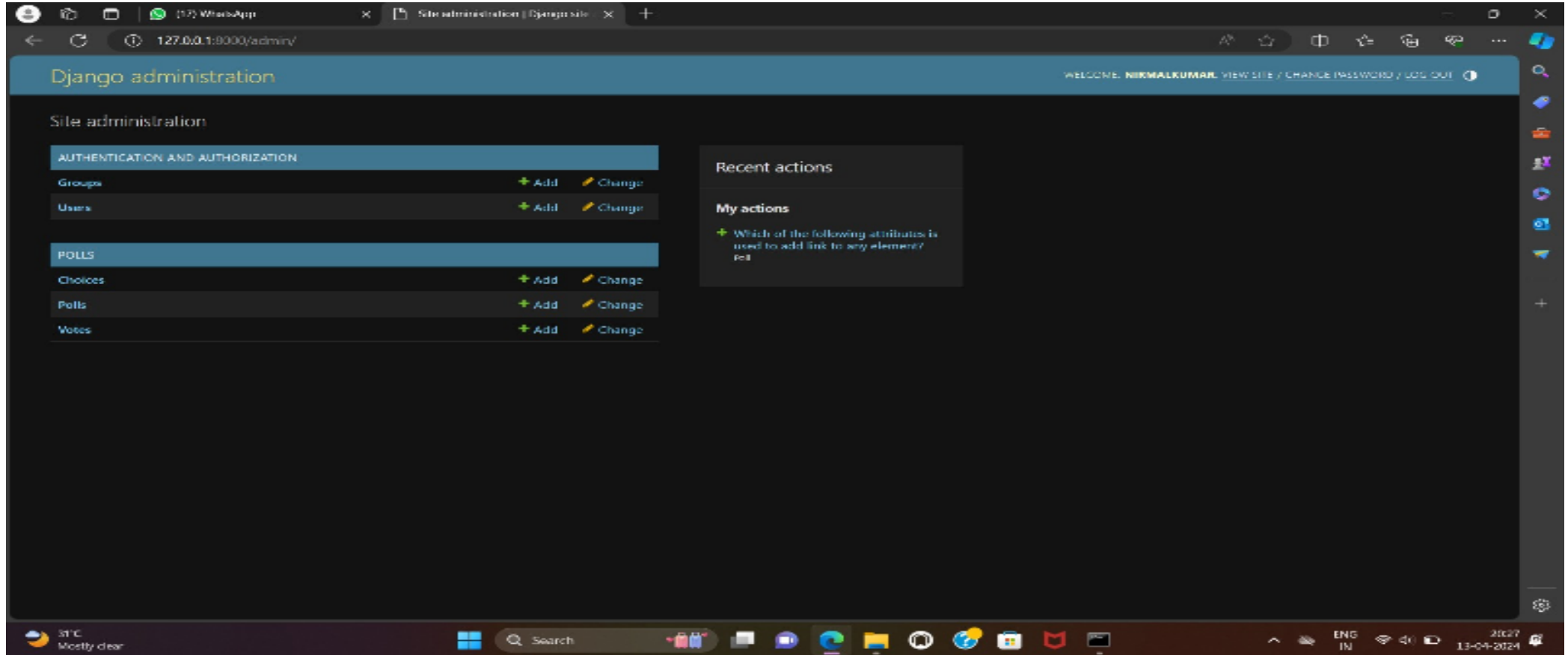
- What does HTML stands for?
- Which of the following is ...
- Which of the following attributes ...

The Windows taskbar at the bottom shows the date and time as 21:27, 13-04-2024, and the language as ENG IN.

Voting Page



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 title is "Django administration". The user is logged in as "NIRMALKUMAR" and can perform actions like "VIEW SITE", "CHANGE PASSWORD", and "LOG OUT".

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

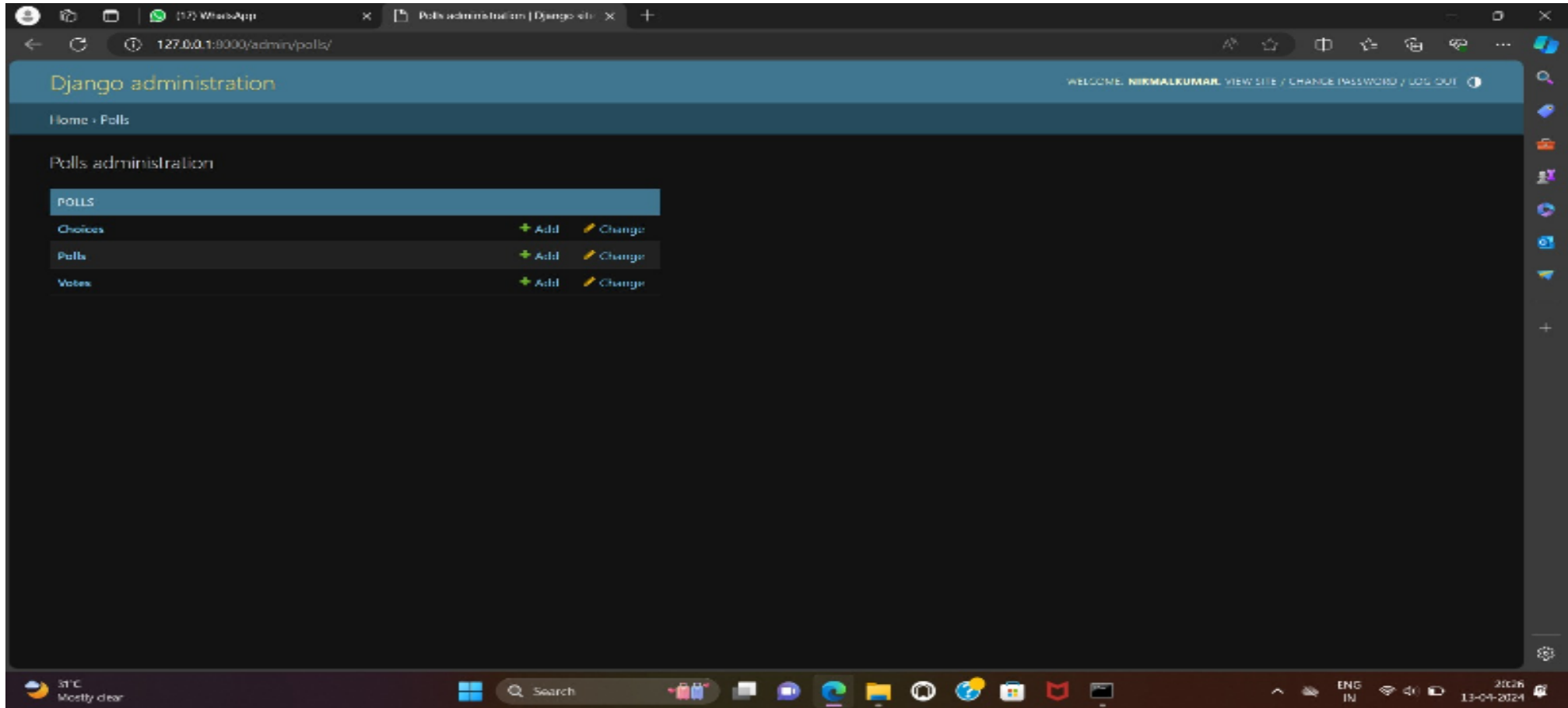
- AUTHENTICATION AND AUTHORIZATION**
 - Groups**: + Add, Change
 - Users**: + Add, Change
- POLLS**
 - Choices**: + Add, Change
 - Polls**: + Add, Change
 - Votes**: + Add, Change

On the right side, there are two panels:

- Recent actions**: Empty.
- My actions**: Contains a single action: "+ Which of the following attributes is used to add link to any element? (all)".

The Windows taskbar at the bottom shows the date and time as 2027 13-04-2024, the language as ENG IN, and the weather as 31°C Mostly clear.

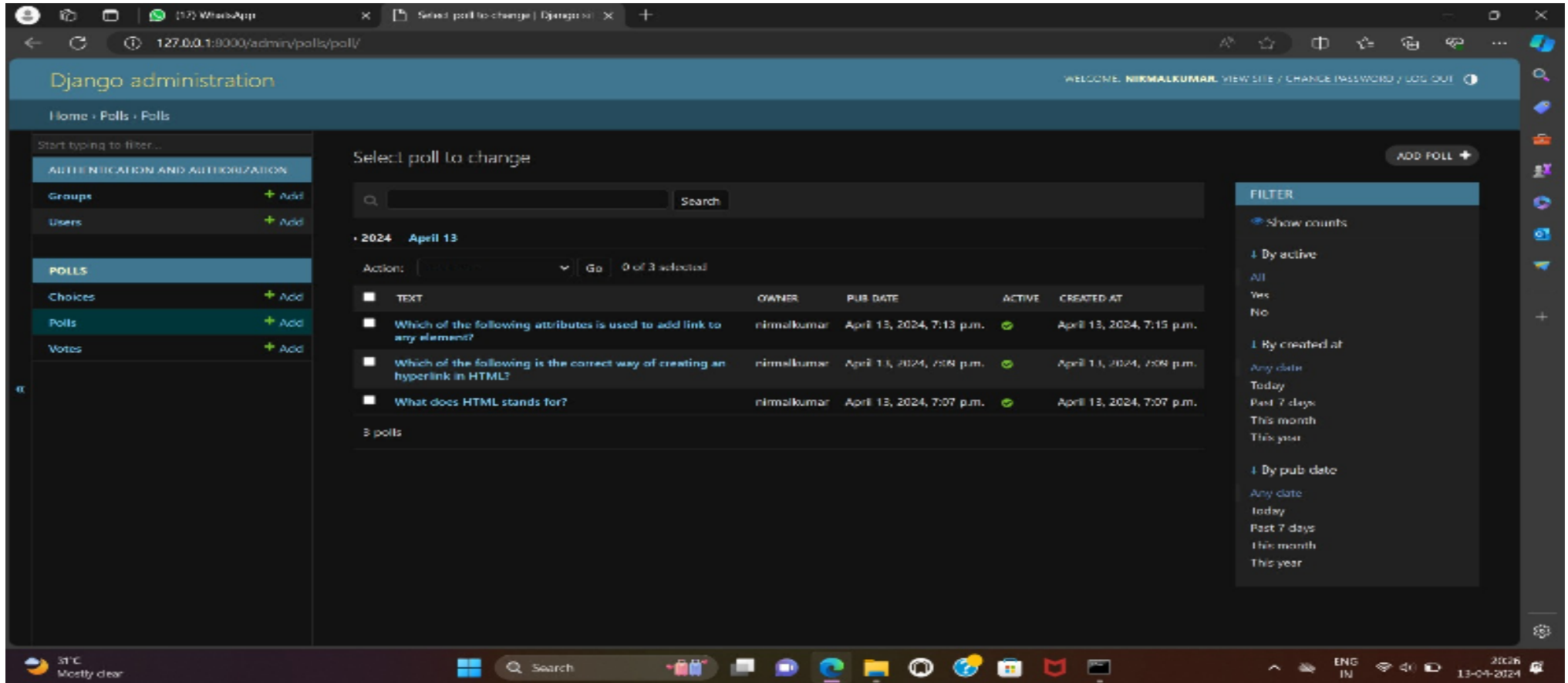
polls administration



The screenshot shows a web browser window displaying the Django administration interface for a 'Polls' application. The browser's address bar shows the URL '127.0.0.1:8000/admin/polls/'. The Django admin header is blue and includes the text 'Django administration' and a welcome message for 'NIRMALKUMAR'. The breadcrumb trail is 'Home > Polls'. The main content area is titled 'Polls administration' and contains a table with three rows: 'POLLS', 'Choices', and 'Polls'. Each row has '+ Add' and 'Change' links. The Windows taskbar at the bottom shows the date as 13-04-2024 and the time as 21:26.

| POLLS | |
|---------|--|
| Choices | + Add Change |
| Polls | + Add Change |
| Votes | + Add Change |

Authentication and Authorization Page



The screenshot displays the Django administration interface for a poll application. The left sidebar shows the navigation menu with 'Authentication and Authorization' and 'Polls' sections. The main content area is titled 'Select poll to change' and features a search bar, a date filter for '2024 April 13', and a table of polls. A 'FILTER' dropdown is open on the right, showing options for filtering by active status, created date, and published date.

Django administration WELCOME, NIRMALKUMAR. 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

Search

2024 April 13

Action: [v] Go 0 of 3 selected

| | TEXT | OWNER | PUB DATE | ACTIVE | CREATED AT |
|--------------------------|---|-------------|---------------------------|--------|---------------------------|
| <input type="checkbox"/> | Which of the following attributes is used to add link to any element? | nirmalkumar | April 13, 2024, 7:13 p.m. | ✓ | April 13, 2024, 7:15 p.m. |
| <input type="checkbox"/> | Which of the following is the correct way of creating an hyperlink in HTML? | nirmalkumar | April 13, 2024, 7:08 p.m. | ✓ | April 13, 2024, 7:08 p.m. |
| <input type="checkbox"/> | What does HTML stands for? | nirmalkumar | April 13, 2024, 7:07 p.m. | ✓ | April 13, 2024, 7:07 p.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

5°C Mostly clear

Search

ENG IN 2024 13-04-2024

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!