



# NEXT GEN EMPLOYABILITY PROGRAM

| Creating a future-ready workforce

**Student name:** MARY NIVEDHITHA.A

**Student ID :**  
au820621104041

**College Name**

Arasu Engineering College

# CAPSTONE PROJECT

## Project Title

Voting Application using Django Framework- MAGESWARAN.R(4034,AEC)

Abstract | Problem Statement | Project Overview | Proposed Solution |  
Technology Used | Modelling & Results | Conclusion



## ABSTRACT

The Django framework, assures a solid and scalable base. The web-based architecture of the proposed voting application enables users to establish and take part in online votes. The Django framework, a well-liked and robust Python-based web framework that offers a solid foundation for creating scalable and secure online applications, is used in the construction of the application. Along with being scalable and adaptable, the program has a modular architecture that makes modification and extension simple. Because of this, it may be applied to a variety of use cases, ranging from intimate internal voting to extensive public elections. All things considered, the suggested voting application provides a safe, convenient, and adaptable online voting platform. Its emphasis on security and user experience, along , making it an excellent and provides a better security to web applications.

## PROBLEM STATEMENT

With more organizations and governments using digital platforms to hold elections and surveys, online voting has grown in popularity in recent years. Online voting does, however, come with a number of drawbacks, mostly with regard to security and integrity. The suggested voting application aims to tackle the issues of security and integrity in online voting, while simultaneously offering an intuitive platform for conducting votes online. Its emphasis on security and user experience, along with its usage of the Django framework, will make it an excellent option for a variety of voting scenarios. The program will emphasize user experience in addition to security, with a clear and simple 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.

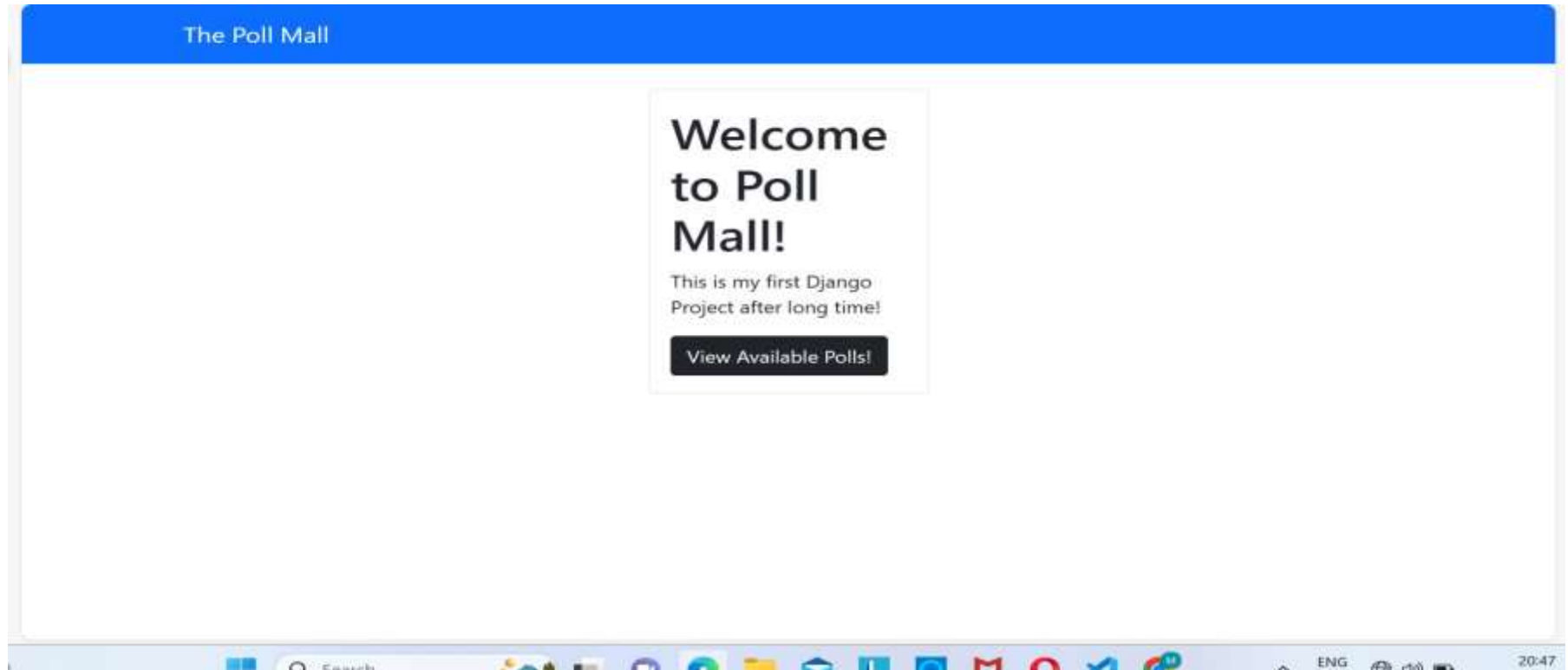
## PROJECT OVERVIEW

- 1.Establishing a Django Project:** To provide the framework for the voting application, create a Django project.
- 2.Creating the Database Schema:** Establish the framework for the database that will hold user data, vote tallies, and other pertinent information.
- 3.Establishing User Authentication:** To enable users to sign up, log in, and take part in voting, implement user authentication.
- 4.Creating the Voting Interface:** Create the user interface that allows users to browse options, make selections, and cast ballots.
- 5.Putting Real-time Results into Practice:** Present the voting results in real-time to give users immediate feedback.
- 6.Creating an Admin Panel:** To efficiently manage the candidates, user accounts, and voting process, create an admin panel.

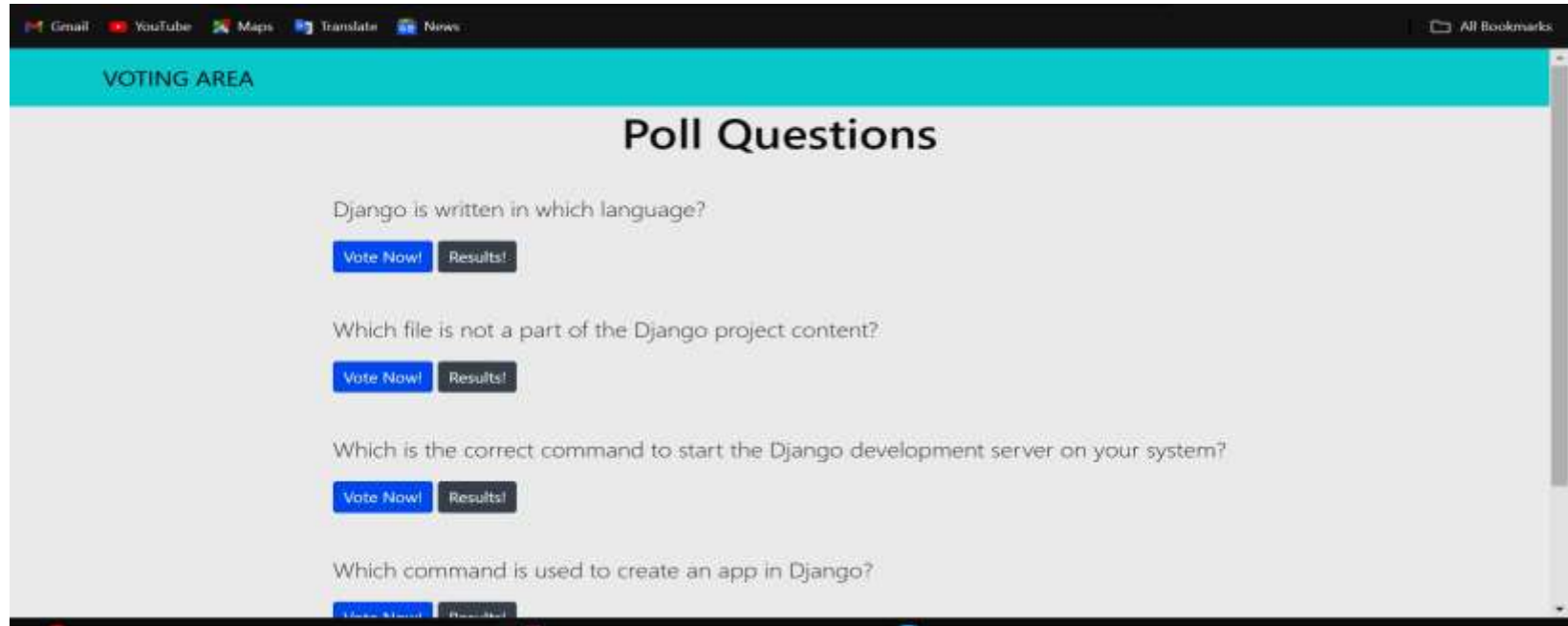
## DETAILED SOLUTION

Developing a safe and user-friendly online voting platform is the suggested solution for a voting application built with the Django framework. Users of the app will be able to register, cast ballots, and see the results in real time. The Django framework will serve as the cornerstone for developing the application because of its scalability and resilience. In order to manage elections, candidates, and user accounts effectively, the program will feature an admin panel, a secure database, real-time results, and an easy-to-use interface. In conclusion, the Django framework-based voting application that has been suggested offers a safe, adaptable, and user-friendly online voting platform. Its emphasis on security and user experience, along with its use of the Django framework, assures a solid and scalable base, making it an excellent ideal choice for a wide range of voting scenarios.

## HOME PAGE



## POLL PAGE



The screenshot shows a web browser window with a black top bar containing links to Gmail, YouTube, Maps, Translate, and News. On the right of the top bar is a link to 'All Bookmarks'. Below the top bar is a teal header with the text 'VOTING AREA'. The main content area has a light gray background and is titled 'Poll Questions'. It contains four poll questions, each with a blue 'Vote Now!' button and a dark gray 'Results!' button.

**VOTING AREA**

### Poll Questions

Django is written in which language?

[Vote Now!](#) [Results!](#)

Which file is not a part of the Django project content?

[Vote Now!](#) [Results!](#)

Which is the correct command to start the Django development server on your system?

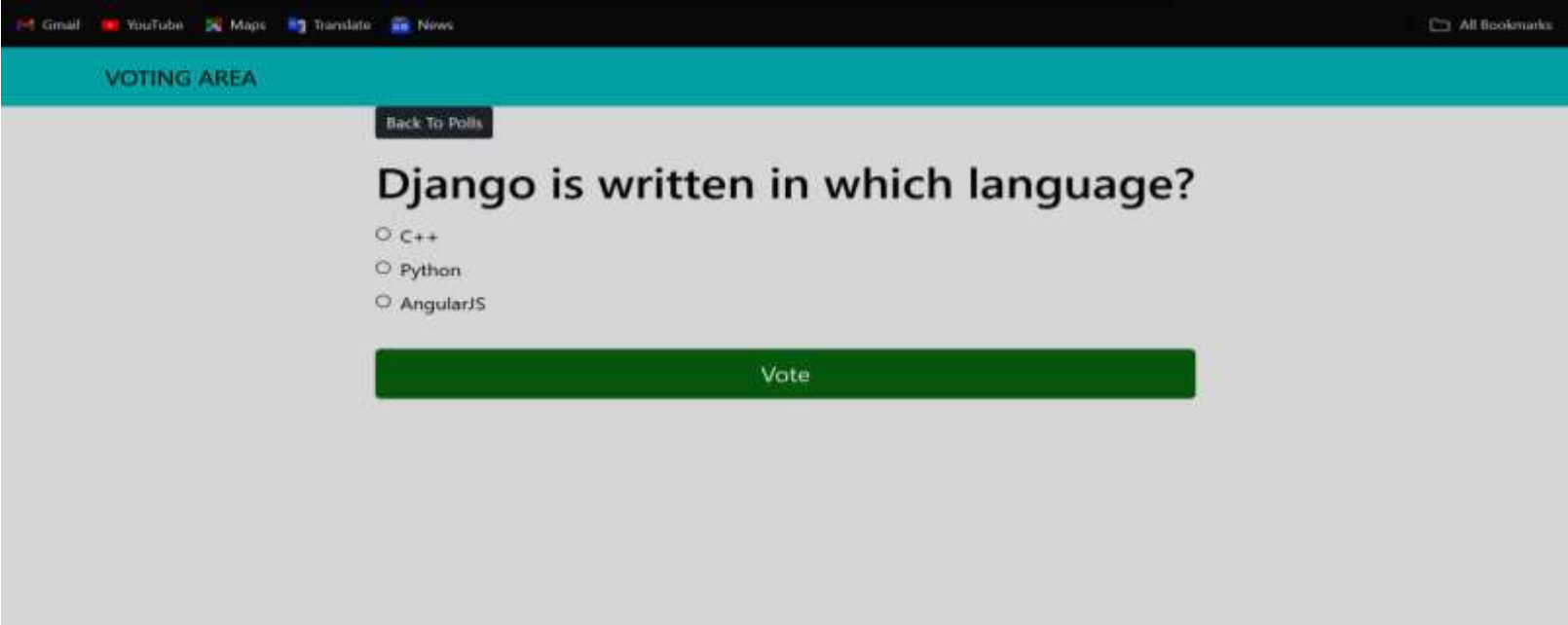
[Vote Now!](#) [Results!](#)

Which command is used to create an app in Django?

[Vote Now!](#) [Results!](#)



## VOTING PAGE



VOTING AREA

[Back To Polls](#)

Django is written in which language?

☐ C++

☐ Python

☐ AngularJS

[Vote](#)

## VOTING DETAILS PAGE

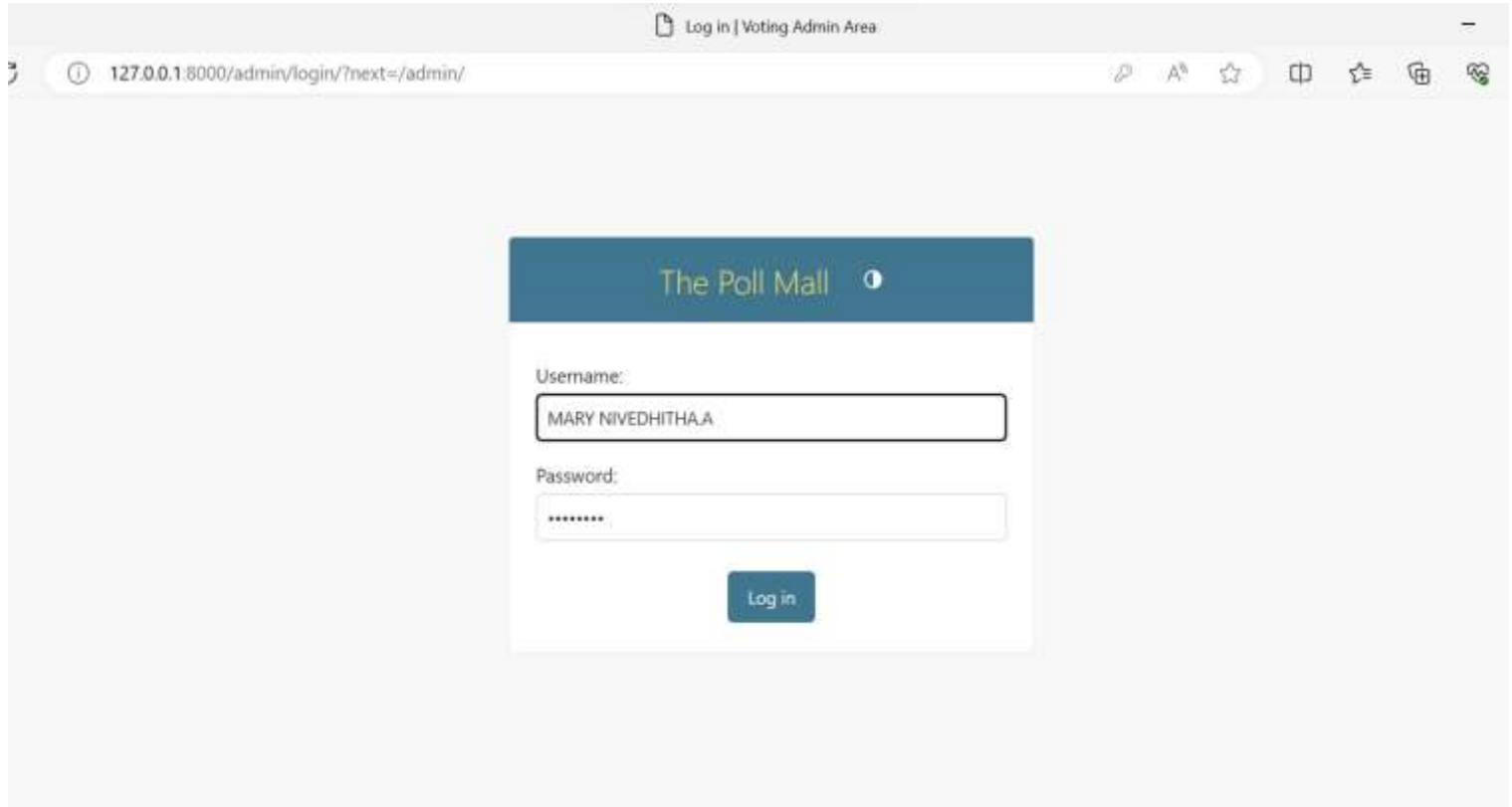
VOTING AREA

Django is written in which language?

C++	4 votes
Python	10 votes
AngularJS	6 votes

[Back To Polls](#) [Vote again?](#)

## ADMIN LOGIN PAGE



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:8000/admin/login/?next=/admin/". The page title is "Log in | Voting Admin Area". The main content area features a login form titled "The Poll Mall" with a help icon. The form includes a "Username:" field with the text "MARY NIVEDHITHA.A", a "Password:" field with masked characters "\*\*\*\*\*", and a "Log in" button.

Log in | Voting Admin Area

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

The Poll Mall ⓘ

Username:

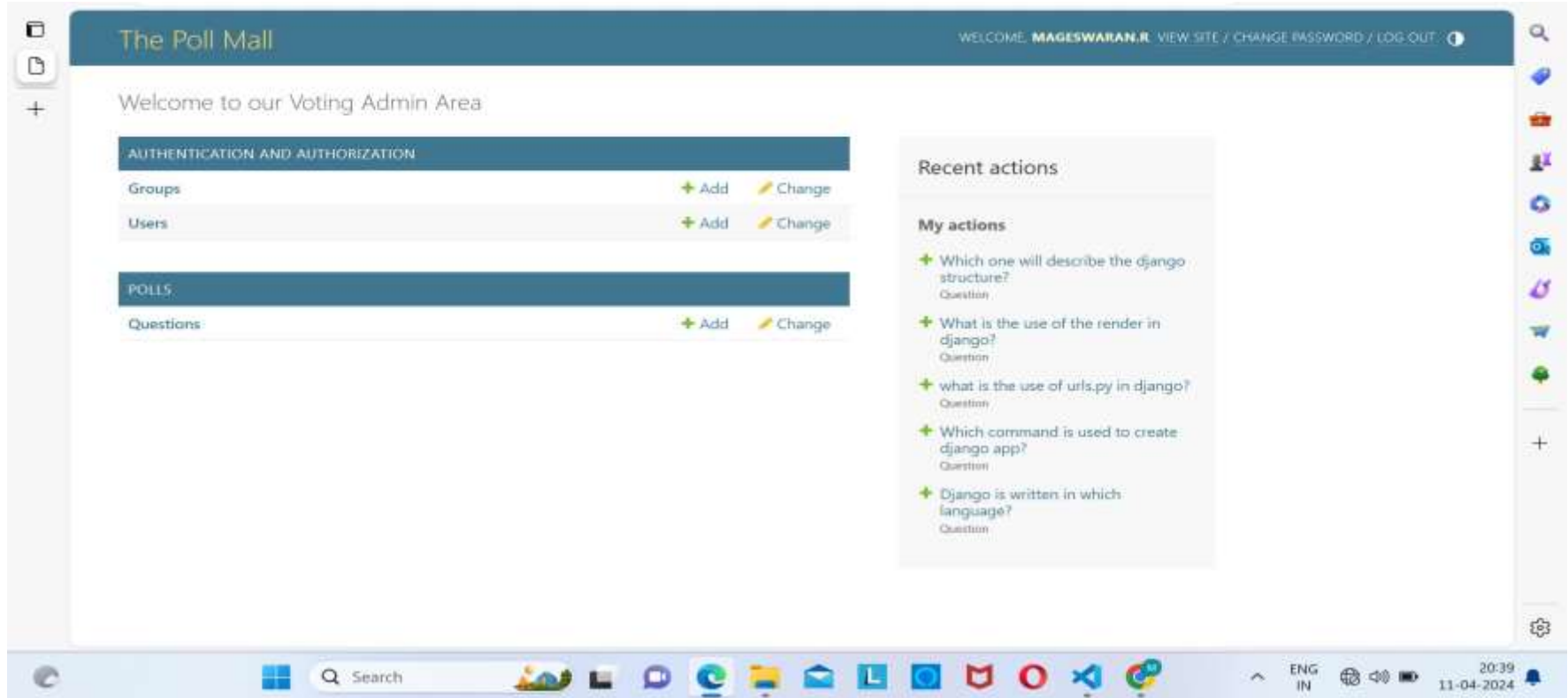
MARY NIVEDHITHA.A

Password:

\*\*\*\*\*

Log in

## ADMIN HOME PAGE



The screenshot displays the 'The Poll Mall' admin interface. The top navigation bar is teal with the title 'The Poll Mall' on the left and user information 'WELCOME, **MAGESWARAN.R** VIEW SITE / CHANGE PASSWORD / LOG-OUT' on the right. A vertical sidebar on the left contains icons for home, documents, and a plus sign. The main content area has a teal header with the title 'The Poll Mall' and a welcome message 'Welcome to our Voting Admin Area'. Below this, there are two main sections: 'AUTHENTICATION AND AUTHORIZATION' and 'POLLS'. The 'AUTHENTICATION AND AUTHORIZATION' section contains two rows: 'Groups' and 'Users', each with '+ Add' and 'Change' links. The 'POLLS' section contains one row: 'Questions', also with '+ Add' and 'Change' links. To the right of the 'POLLS' section, there is a 'Recent actions' box titled 'Recent actions' and 'My actions'. The 'My actions' box lists five actions, each with a green plus icon and a question text: 'Which one will describe the django structure?', 'What is the use of the render in django?', 'what is the use of urls.py in django?', 'Which command is used to create django app?', and 'Django is written in which language?'. The bottom of the screen shows a Windows taskbar with various application icons, a search bar, and system tray icons including the date and time '20:39 11-04-2024'.

**The Poll Mall**

WELCOME, **MAGESWARAN.R** VIEW SITE / CHANGE PASSWORD / LOG-OUT

Welcome to our Voting Admin Area

**AUTHENTICATION AND AUTHORIZATION**

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

**POLLS**

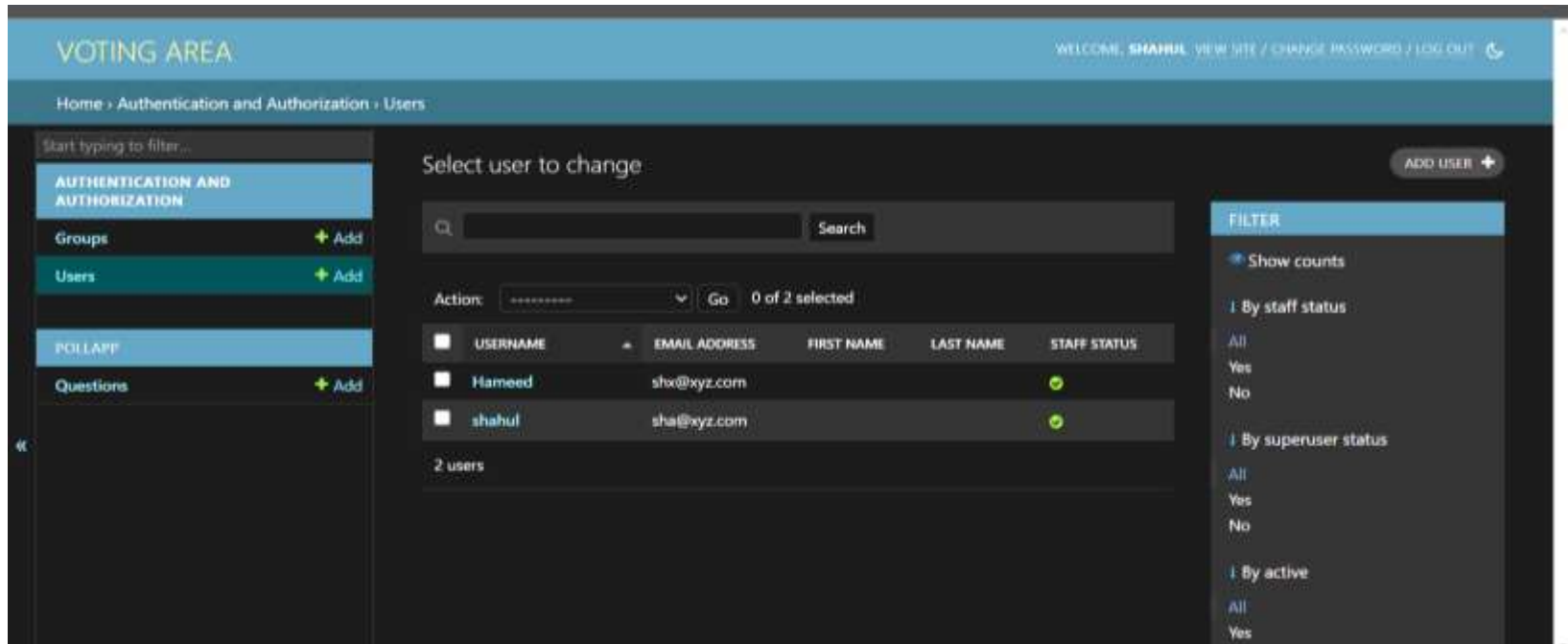
- Questions [+ Add](#) [Change](#)

**Recent actions**

**My actions**

- + Which one will describe the django structure?  
Question
- + What is the use of the render in django?  
Question
- + what is the use of urls.py in django?  
Question
- + Which command is used to create django app?  
Question
- + Django is written in which language?  
Question

## AUTHENTICATION AND AUTHORIZATION PAGE



The screenshot displays the 'VOTING AREA' interface. At the top, a navigation bar includes 'WELCOME, SHAHUL', 'VIEW SITE', 'CHANGE PASSWORD', and 'LOG OUT'. Below this, a breadcrumb trail reads 'Home > Authentication and Authorization > Users'. The left sidebar contains a menu with 'AUTHENTICATION AND AUTHORIZATION' (expanded), 'Groups', 'Users', 'POLL APP', and 'Questions'. The main content area is titled 'Select user to change' and features a search bar, an 'ADD USER' button, and an 'Action:' dropdown. A table lists two users: 'Hameed' and 'shahul', both with email addresses ending in '@xyz.com' and active status (green checkmarks). The bottom of the table indicates '2 users'. A right sidebar titled 'FILTER' offers options to 'Show counts', filter by 'staff status' (All, Yes, No), 'superuser status' (All, Yes, No), and 'active' (All, Yes).

VOTING AREA

WELCOME, SHAHUL VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Authentication and Authorization > Users

Start typing to filter...

**AUTHENTICATION AND AUTHORIZATION**

Groups + Add

Users + Add

**POLL APP**

Questions + Add

Select user to change

ADD USER +

Search

Action: [dropdown] Go 0 of 2 selected

<input type="checkbox"/>	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
<input type="checkbox"/>	Hameed	shx@xyz.com			✓
<input type="checkbox"/>	shahul	sha@xyz.com			✓

2 users

**FILTER**

Show counts

By staff status

All  
Yes  
No

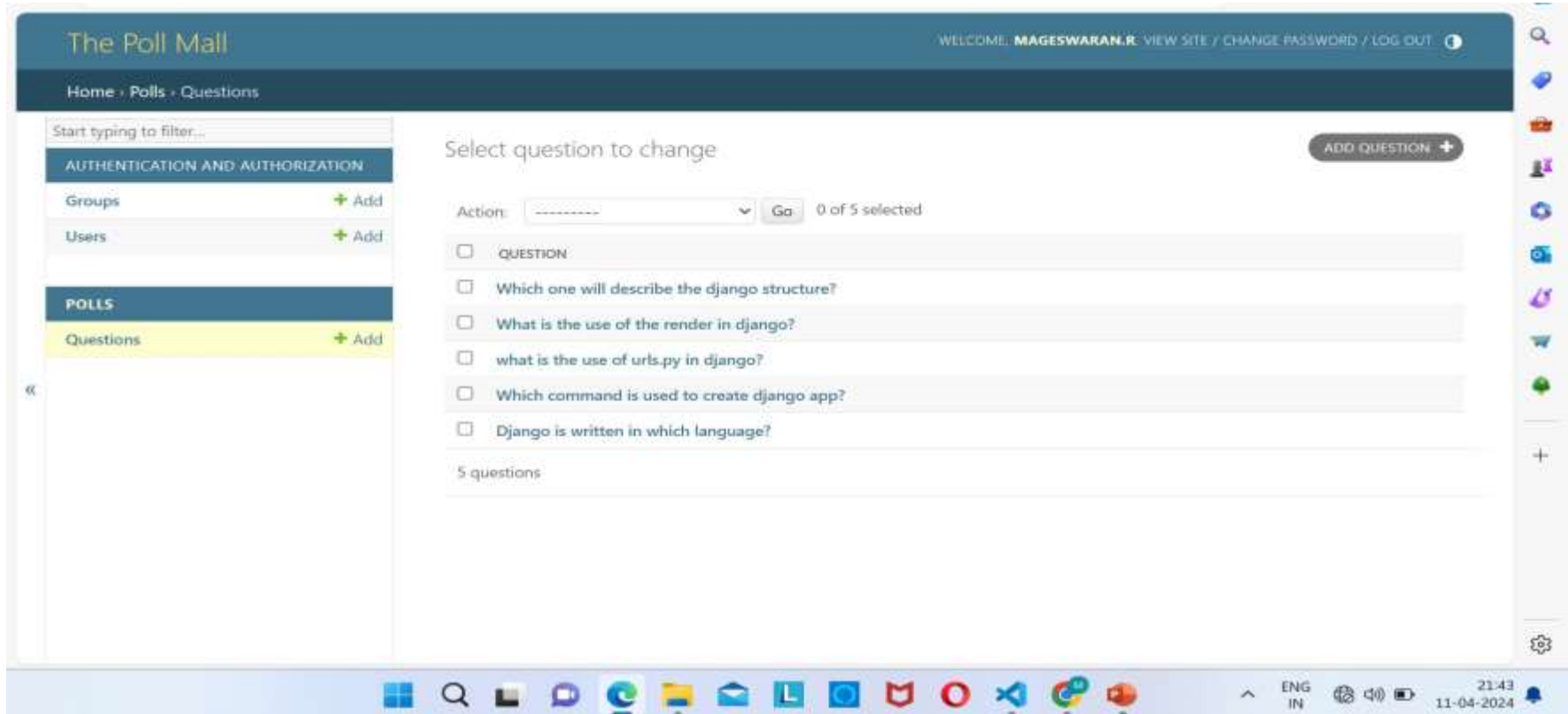
By superuser status

All  
Yes  
No

By active

All  
Yes

## QUESTIONS ADDING SECTION PAGE



The screenshot displays the 'The Poll Mall' web application interface. The top navigation bar includes the site name and a user welcome message: 'WELCOME: **MAGESWARAN.R.** VIEW SITE / CHANGE PASSWORD / LOG OUT'. Below this, a breadcrumb trail shows 'Home > Polls > Questions'. A left sidebar contains a search bar and two main categories: 'AUTHENTICATION AND AUTHORIZATION' (with sub-items 'Groups' and 'Users', each with an 'Add' button) and 'POLLS' (with a 'Questions' sub-item and an 'Add' button). The main content area is titled 'Select question to change' and features an 'ADD QUESTION +' button. It lists five questions, each with a checkbox and an 'Action' dropdown menu. The questions are: 'QUESTION', 'Which one will describe the django structure?', 'What is the use of the render in django?', 'what is the use of urls.py in django?', and 'Which command is used to create django app?'. Below the list, it indicates '5 questions'. The bottom of the image shows a Windows taskbar with various application icons and a system tray with the date '11-04-2024' and time '21:43'.

**The Poll Mall** WELCOME: **MAGESWARAN.R.** VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Polls > Questions

Start typing to filter...

**AUTHENTICATION AND AUTHORIZATION**

- Groups + Add
- Users + Add

**POLLS**

- Questions + Add

Select question to change

ADD QUESTION +

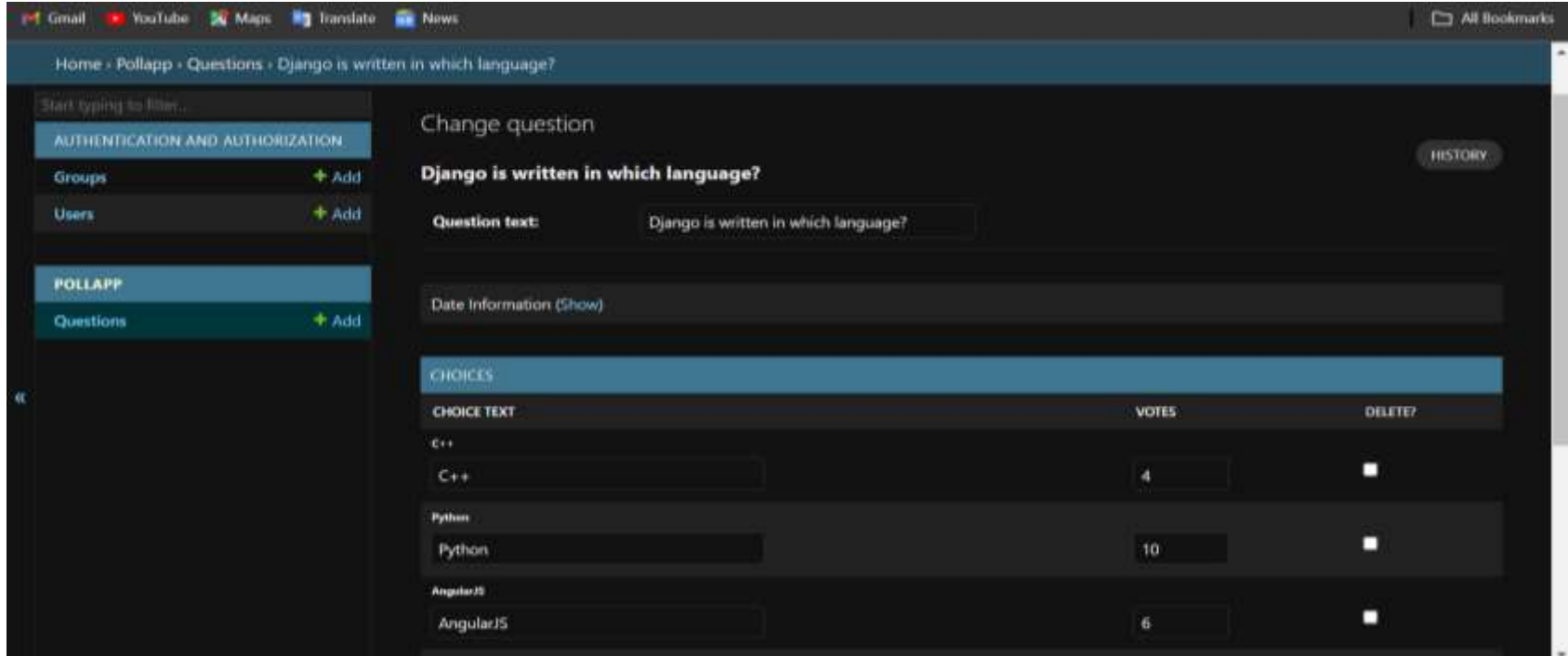
Action: [-----] Go 0 of 5 selected

- ☐ QUESTION
- ☐ Which one will describe the django structure?
- ☐ What is the use of the render in django?
- ☐ what is the use of urls.py in django?
- ☐ Which command is used to create django app?
- ☐ Django is written in which language?

5 questions

Windows taskbar: 21:43, 11-04-2024

## VOTING DETAILS PAGE



The screenshot shows a web application interface for managing a poll. The top navigation bar includes links for Home, Pollapp, Questions, and the current question: "Django is written in which language?". A left sidebar contains a search bar and two main sections: "AUTHENTICATION AND AUTHORIZATION" with links for Groups and Users, and "POLLAPP" with a link for Questions. The main content area is titled "Change question" and features a "HISTORY" button. Below this, the "Question text" is displayed as "Django is written in which language?". A "Date Information (Show)" link is also present. The "CHOICES" section contains a table with three columns: "CHOICE TEXT", "VOTES", and "DELETE?". The table lists three choices: "C++" with 4 votes, "Python" with 10 votes, and "AngularJS" with 6 votes. Each choice has a delete icon next to it.

CHOICE TEXT	VOTES	DELETE?
C++	4	<input type="checkbox"/>
Python	10	<input type="checkbox"/>
AngularJS	6	<input type="checkbox"/>

## Technology Used

### FRONT-END



### BACK-END





## CONCLUSION

A good understanding of HTML, CSS, Bootstrap, Django, and Python programming to design a voting application using Django. The development process entails starting a fresh Django project, building a Django application, specifying models, views, templates, and locations. Features like a user-friendly interface, a secure database design, and real-time results can further improve the program. An admin panel for controlling elections, candidates, and user accounts may also be included. All things considered, a voting application built using the Django framework is a strong and adaptable way to build online voting systems that meet different needs and use cases.