



NEXT GEN EMPLOYABILITY PROGRAM

| Creating a future-ready workforce

Student Name :SARAVANA.G
Student ID :au820621104076

College Name

Arasu Engineering College

CAPSTONE PROJECT SHOWCASE

Project Title

Voting Application using Django Framework-SARAVANA.G(820621104076,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.

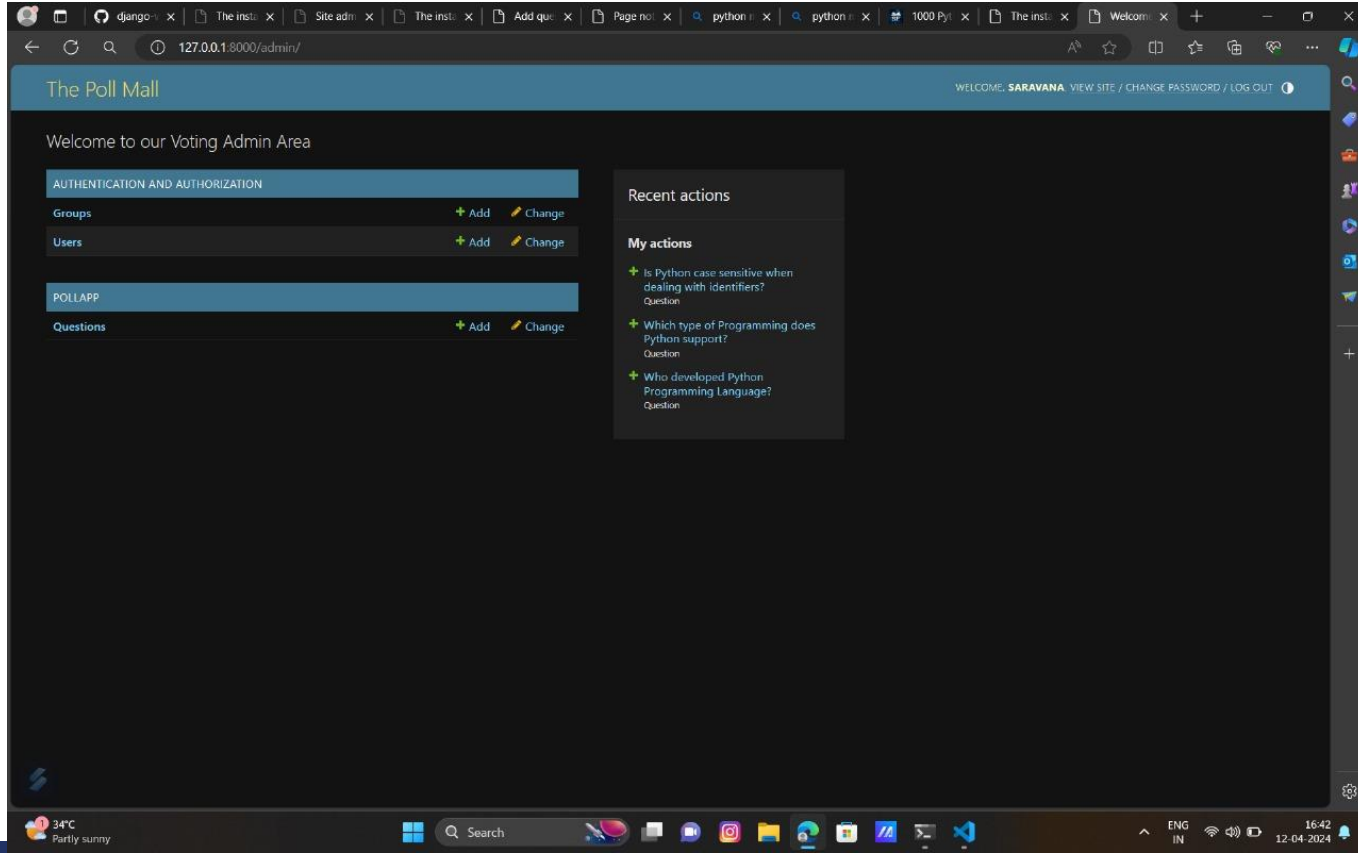
In summary, the proposed solution for a voting application using the Django framework is a secure, user-friendly, and flexible platform for conducting online votes. Its use of the Django framework ensures a robust and scalable foundation, while its focus on security and user experience makes it an ideal choice for a wide range of voting scenarios.

The Poll Mall

Welcome to Poll Mall!

This is my first Django Project after long time!

[View Available Polls!](#)



The screenshot shows a web browser window displaying the Django Admin interface for a project named 'The Poll Mall'. The browser's address bar shows the URL '127.0.0.1:8000/admin/'. The page has a dark theme. At the top, a teal header bar contains the site name 'The Poll Mall' on the left and a user welcome message 'WELCOME, SARAVANA' with links for 'VIEW SITE', 'CHANGE PASSWORD', and 'LOG OUT' on the right. Below the header, the main content area is divided into several sections. On the left, there's a 'Welcome to our Voting Admin Area' message. Below this, there are two main sections: 'AUTHENTICATION AND AUTHORIZATION' and 'POLLAPP'. The 'AUTHENTICATION AND AUTHORIZATION' section contains two items: 'Groups' and 'Users', each with '+ Add' and 'Change' links. The 'POLLAPP' section contains one item: 'Questions', also with '+ Add' and 'Change' links. On the right side of the main content area, there's a 'Recent actions' section titled 'My actions' which lists three recent actions, each with a green plus icon and a question text: 'Is Python case sensitive when dealing with identifiers?', 'Which type of Programming does Python support?', and 'Who developed Python Programming Language?'. The bottom of the browser window shows a Windows taskbar with various application icons, a search bar, and system status information including temperature (34°C), weather (Partly sunny), time (16:42), and date (12-04-2024).

The Poll Mall

WELCOME, SARAVANA VIEW SITE / CHANGE PASSWORD / LOG OUT

Welcome to our Voting Admin Area

AUTHENTICATION AND AUTHORIZATION

- Groups + Add Change
- Users + Add Change

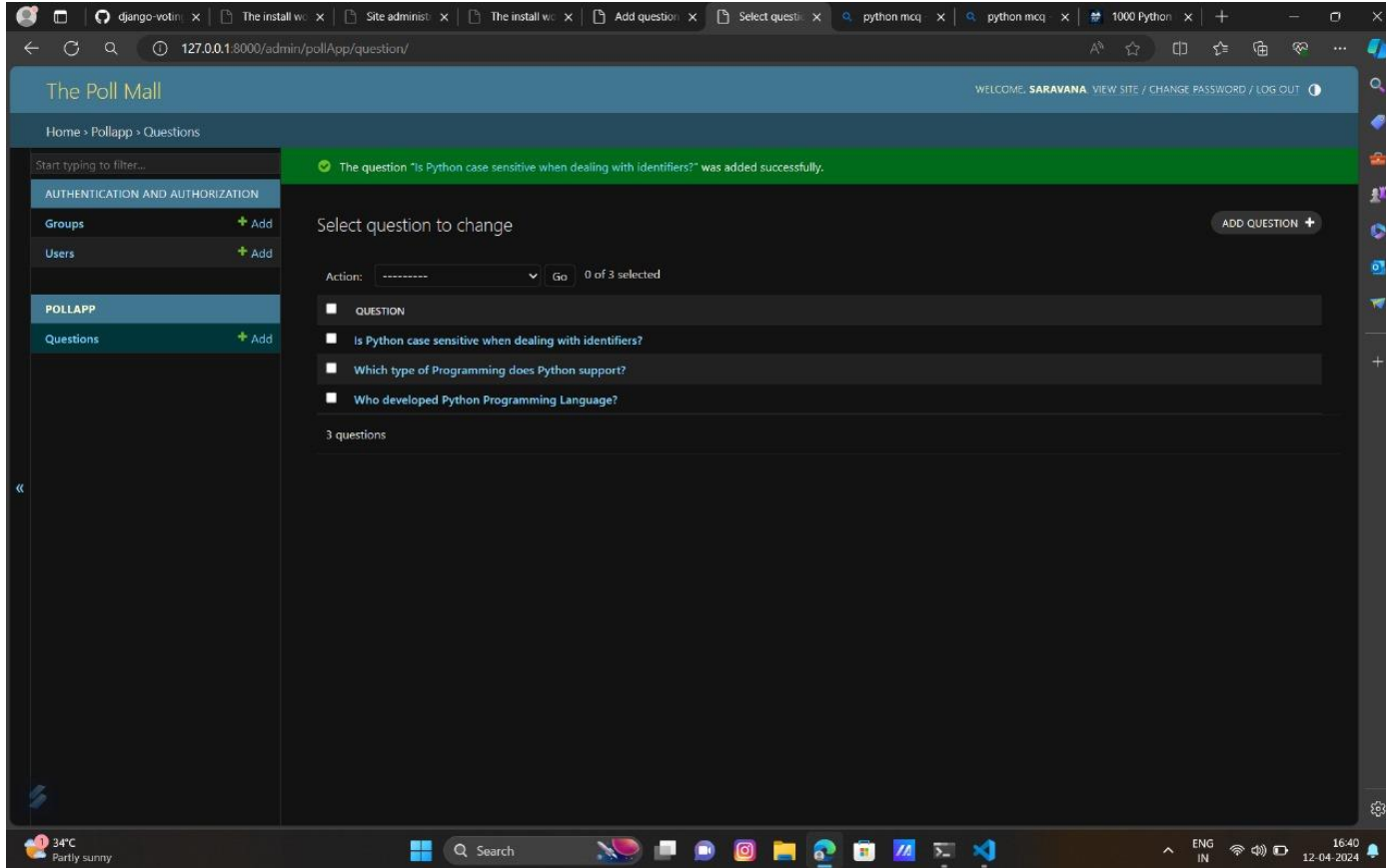
POLLAPP

- Questions + Add Change

Recent actions

My actions

- + Is Python case sensitive when dealing with identifiers? Question
- + Which type of Programming does Python support? Question
- + Who developed Python Programming Language? Question



The screenshot displays the 'The Poll Mall' web application interface. The browser's address bar shows the URL '127.0.0.1:8000/admin/pollApp/question/'. The page header includes the site name 'The Poll Mall' and a user greeting 'WELCOME, SARAVANA' with links for 'VIEW SITE', 'CHANGE PASSWORD', and 'LOG OUT'. A green notification banner at the top states: 'The question "Is Python case sensitive when dealing with identifiers?" was added successfully.'

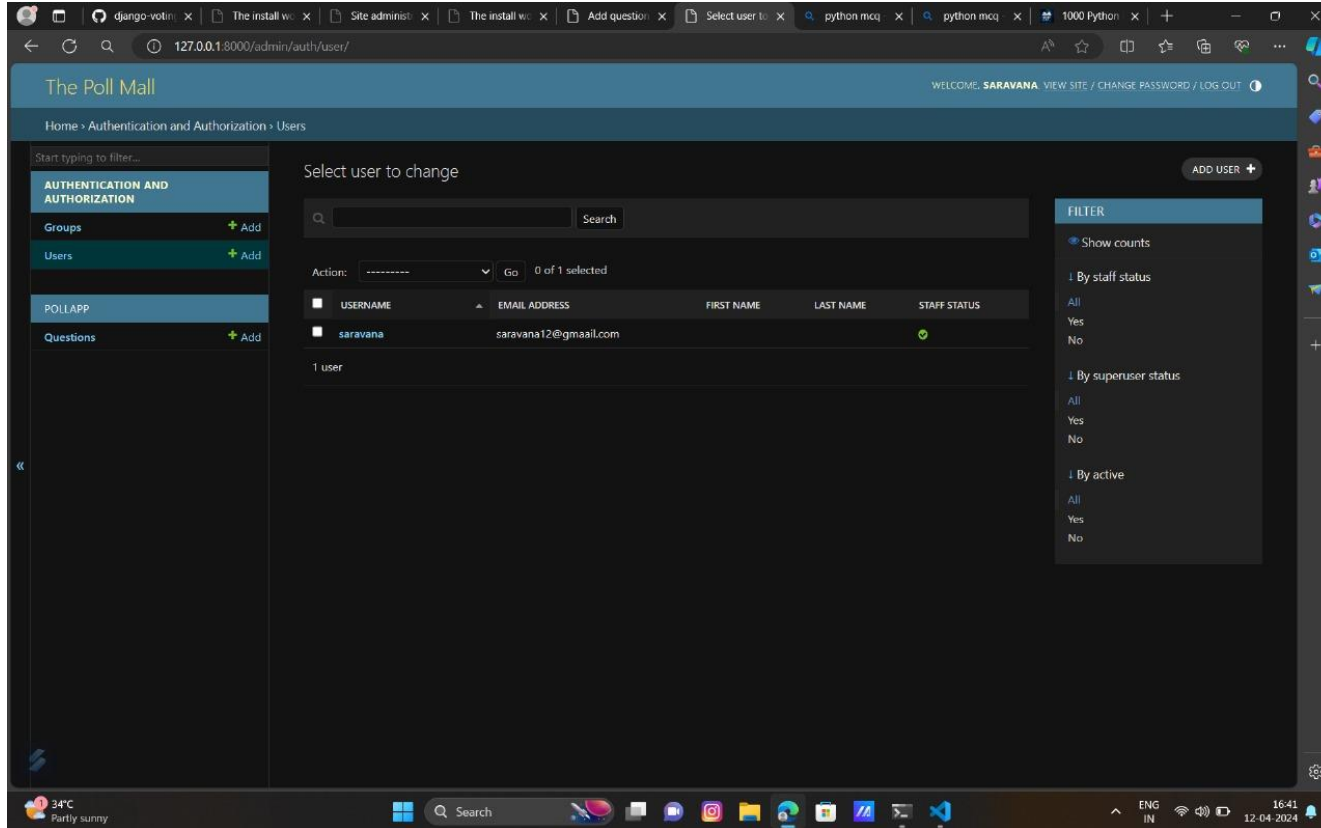
The left sidebar contains a navigation menu with the following sections:

- Start typing to filter...
- AUTHENTICATION AND AUTHORIZATION
 - Groups + Add
 - Users + Add
- POLLAPP
 - Questions + Add

The main content area is titled 'Select question to change' and features an 'ADD QUESTION +' button. Below this, there is an 'Action:' dropdown menu and a 'Go' button, followed by the text '0 of 3 selected'. A list of three questions is displayed, each with a checkbox:

- ☐ QUESTION
- ☐ Is Python case sensitive when dealing with identifiers?
- ☐ Which type of Programming does Python support?
- ☐ Who developed Python Programming Language?

At the bottom of the list, it indicates '3 questions'. The Windows taskbar at the bottom shows the system clock as 16:40 on 12-04-2024, with a temperature of 34°C and weather 'Partly sunny'.



The screenshot displays the Django Admin interface for 'The Poll Mall' application, specifically the 'Users' management page. The browser's address bar shows the URL '127.0.0.1:8000/admin/auth/user/'. The page header includes the site title 'The Poll Mall' and a welcome message for 'SARAVANA' 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
- POLLAPP
 - Questions + Add

The main content area is titled 'Select user to change' and 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, 'saravana', is listed with the email 'saravana12@gmail.com' and a staff status of 'Yes' (indicated by a green circle).

Below the table, it indicates '1 user'. On the right side, there is a 'FILTER' dropdown menu with the following options:

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

The Windows taskbar at the bottom shows the system clock as 16:41 on 12-04-2024, along with weather information (34°C, Partly sunny) and various application icons.

django-votini x The install w... x Site administ... x The install w... x Add question x Add question x python mcq x python mcq x 1000 Python x + -

127.0.0.1:8000/admin/pollApp/question/add/

The Poll Mall

WELCOME, SARAVANA VIEW SITE / CHANGE PASSWORD / LOG OUT

Home > Pollapp > Questions > Add question

Start typing to filter...

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

POLLAPP

Questions + Add

Add question

Question text:

Date Information (Show)

CHOICES		
CHOICE TEXT	VOTES	DELETE?
<input type="text" value="no"/>	<input type="text" value="0"/>	
<input type="text" value="yes"/>	<input type="text" value="0"/>	
<input type="text" value="none of the mentioned"/>	<input type="text" value="0"/>	
+ Add another Choice		

34°C Partly sunny Search ENG IN 16:41 12-04-2024

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!