NEXT GEN EMPLOYABILITY PROGRAM

Creating a future-ready workforce

Team Members

Student Name :Ahamed Haris B

Student ID : 950821104004

College Name:

Government college of Engineering, Tirunelveli

CAPSTONE PROJECT SHOWCASE

Project Title

Voting Web Application using Django Framework

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

Abstract

Our project aims to develop a sophisticated polling application using Django, a powerful Python-based web framework. This web-based polling application will allow users to create custom polls, vote on existing polls, and view real-time results. The app will incorporate user authentication to ensure that only authenticated users can create, manage, or participate in polls. The front-end will be designed using HTML, CSS, and JavaScript to create an intuitive and responsive user interface. Through Django's ORM (Object-Relational Mapping), poll data will be stored and managed efficiently in a relational database, ensuring scalability and performance. The application will implement with sqite for seamless integration with potential future applications. This project showcases the use of Django's capabilities in developing robust and scalable web applications while adhering to best practices in web development and user experience design..

Source :

Problem Statement

The problem at hand is to develop a comprehensive polling application using Django that addresses the need for a user-friendly, secure, and scalable platform for creating and conducting polls. The project aims to overcome the following challenges:

- **User Authentication and Authorization**: Implement secure user authentication and authorization to ensure that only authenticated users can create new polls, manage their own polls, and participate by voting.
- **Poll Creation and Management**: Enable users to create new polls with custom questions and multiple choices. Allow poll creators to edit or delete their polls as needed.
- **Voting Mechanism**: Develop a robust and intuitive interface for users to vote on existing polls. Ensure that each user can only vote once per poll and display real-time results after voting.
- **Data Storage and Management**: Utilize Django's ORM to efficiently store poll data in a relational database, ensuring data integrity and scalability as the number of polls and users grows.
- **User Experience and Interface Design**: Design a responsive and user-friendly interface using HTML, CSS, and JavaScript to provide an engaging experience across different devices and screen sizes.

Source :

Project Overview

User Authentication and Authorization:

- Users will be able to register, log in, and log out.
- Only authenticated users will have permission to create new polls or vote in existing polls.

Poll Creation and Management:

- Authenticated users can create new polls by providing a question and multiple choices.
- User can mention date and time for publishing the result of Polls.

Voting Mechanism:

- Users can view existing polls and vote for their preferred choice.
- Each user can only vote once per.
- Question and Choice of polls are stored in database(sqlite).

Voting result:

- User can see the result of poll only after after date Time which is mention at the time poll creation.
- User can't vote after the result published.
- Result will show include count of each choice and which choice won overall based on max count of choice.

Log out:

- User finally log out.

Source :

Proposed Solution

.

To address the challenges and objectives outlined in the project overview, we propose the development of an E-Voting Web Application using the Django Framework. This solution will leverage modern technologies and best practices to create a robust and user-friendly platform for conducting elections online.

It includes:

- User registration
- Login
- Voting
- Result publishing

Source :

Solution Components:

User Registration and Authentication:

- Implement a secure user registration process where voters can create accounts by providing necessary identification details.
- Utilize strong authentication mechanisms (e.g., username/password, two-factor authentication) to verify voter identities and prevent unauthorized access.

Voting Interface:

- Develop an intuitive and responsive web interface that allows voters to view candidates/parties and cast their votes electronically.
- Ensure accessibility and usability across various devices (desktops, tablets, smartphones) to cater to a diverse user base.

Source :

Result Calculation and Publication:

- Design algorithms to automate the process of counting votes securely and accurately.
- Develop functionalities to generate reports and publish election results in a transparent manner, maintaining the integrity of the electoral process.

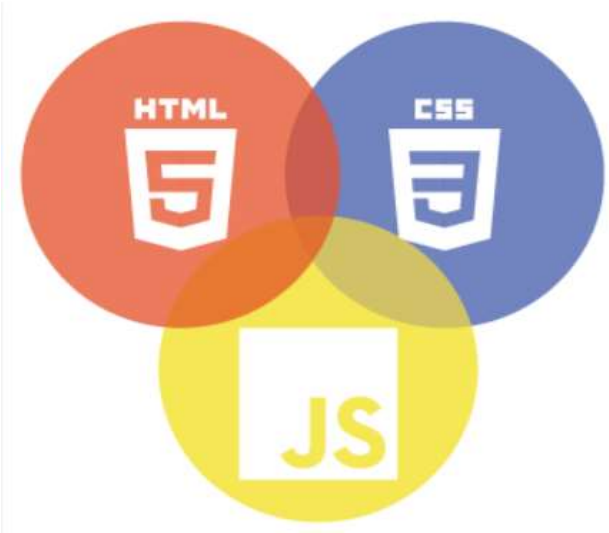
Admin Dashboard:

- Create an administrative dashboard for election officials to manage user registrations, monitor voting activities, and oversee the entire election process.
- Include features for generating audit logs and tracking system activities to ensure accountability and transparency.

Source :

Technology Used

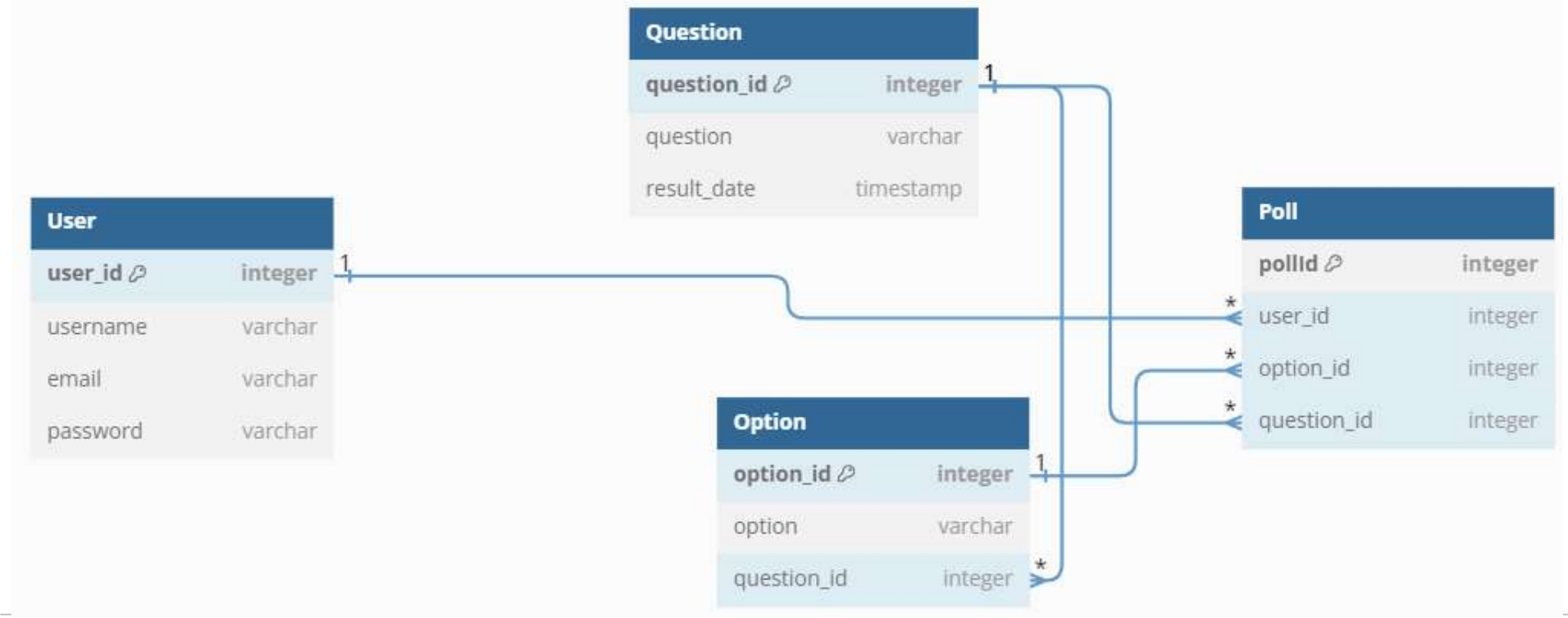
Front-end



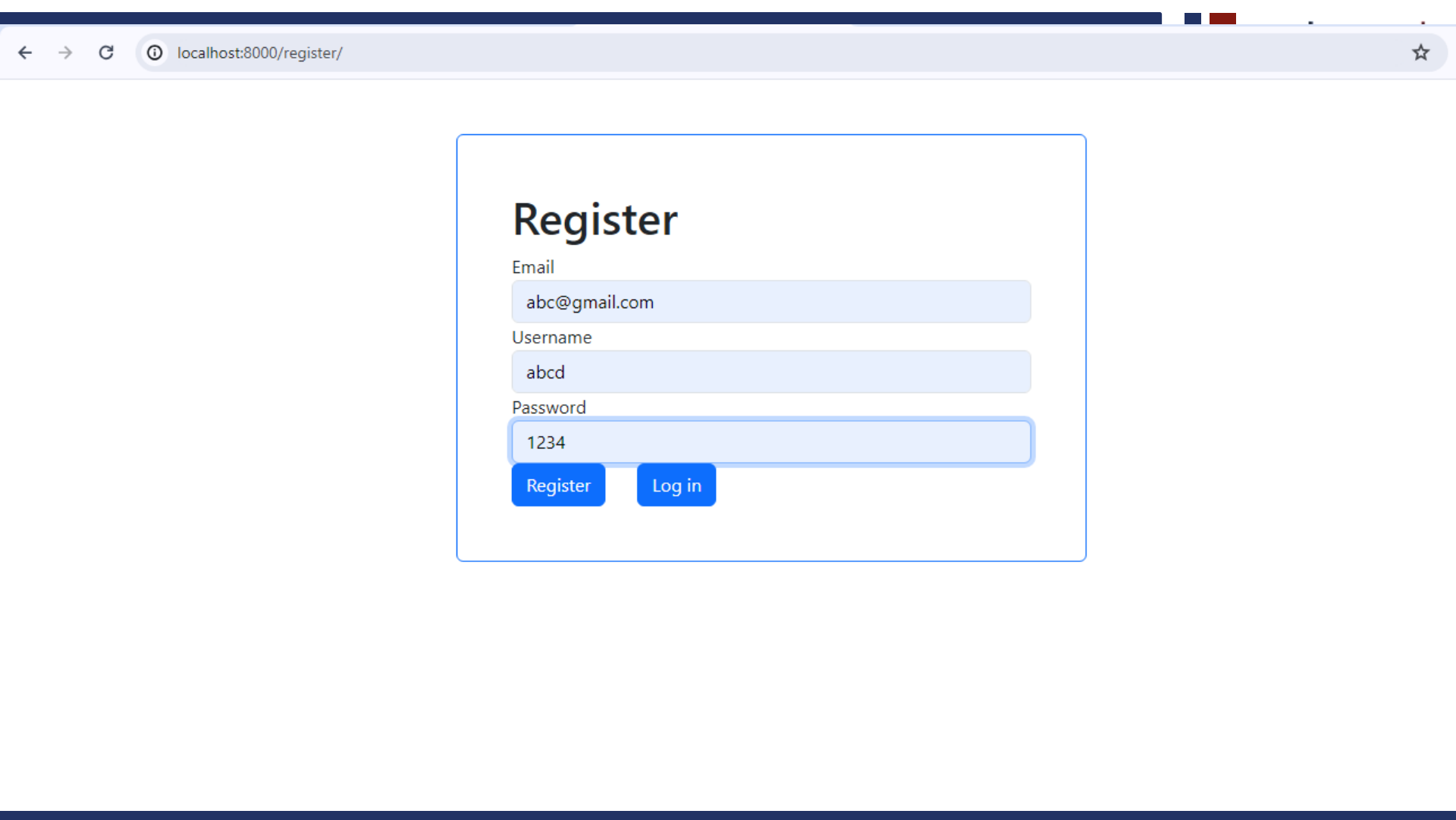
Back-end

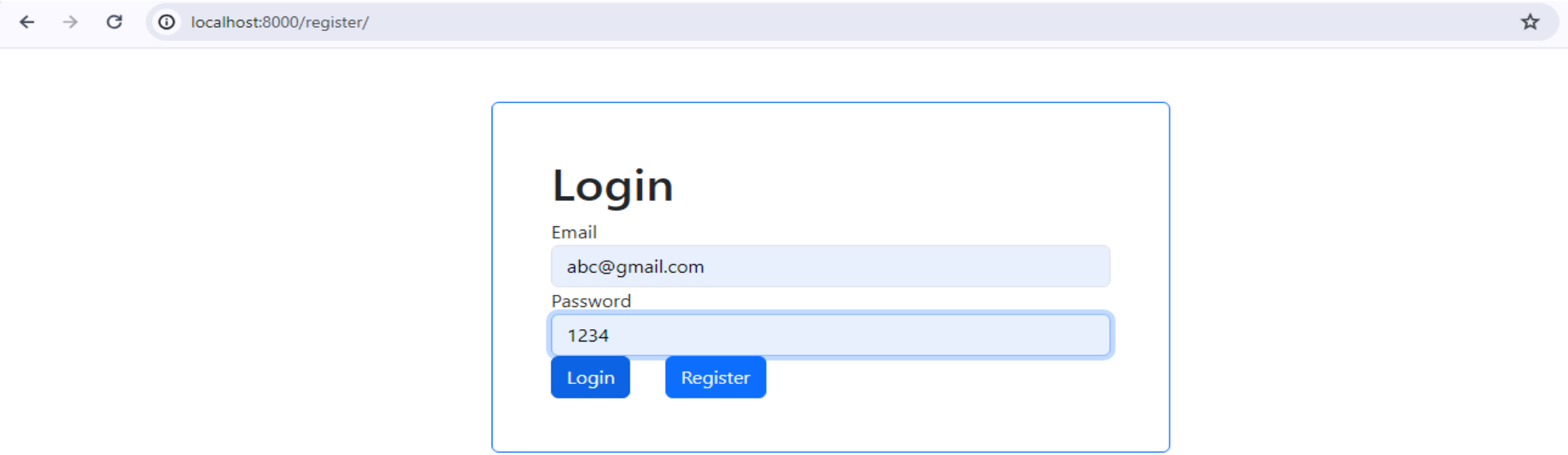
Source :

Modelling & Results

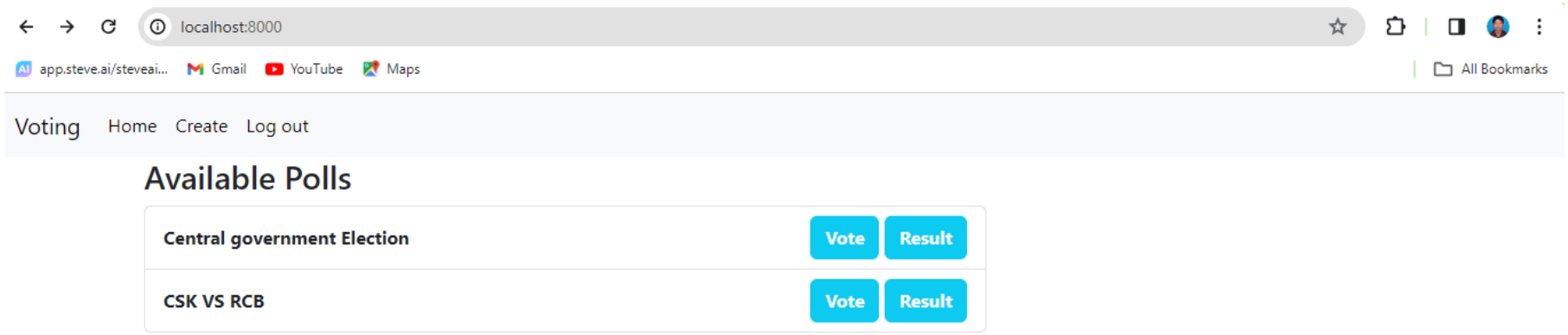


Source :

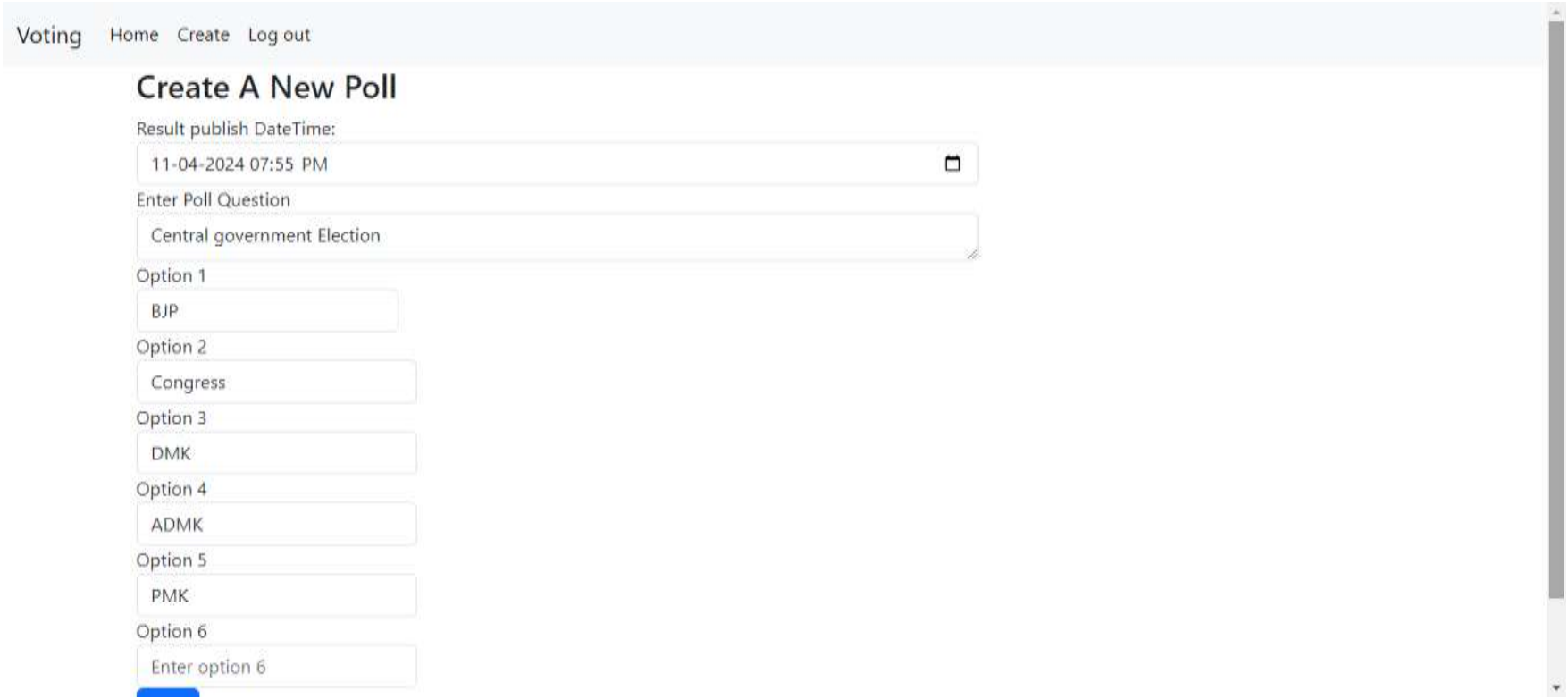

localhost:8000/register/
Register
Email
abc@gmail.com
Username
abcd
Password
1234
Register
Log in


localhost:8000/register/
Login
Email
abc@gmail.com
Password
1234
Login
Register

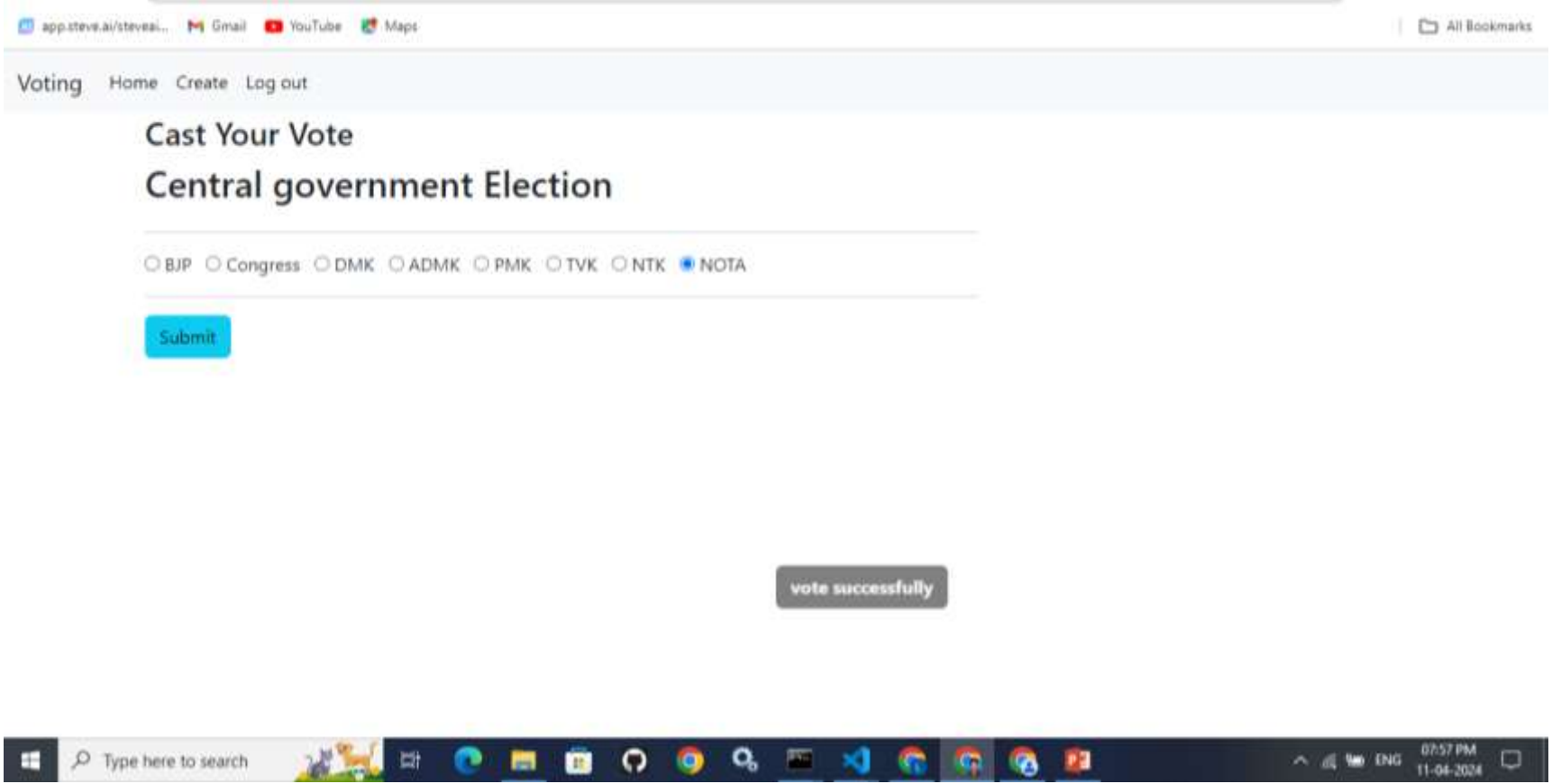
Home Page


localhost:8000
app.steve.ai/steveai...
Gmail
YouTube
Maps
All Bookmarks
Voting
Home
Create
Log out
Available Polls
Central government Election
Vote
Result
CSK VS RCB
Vote
Result

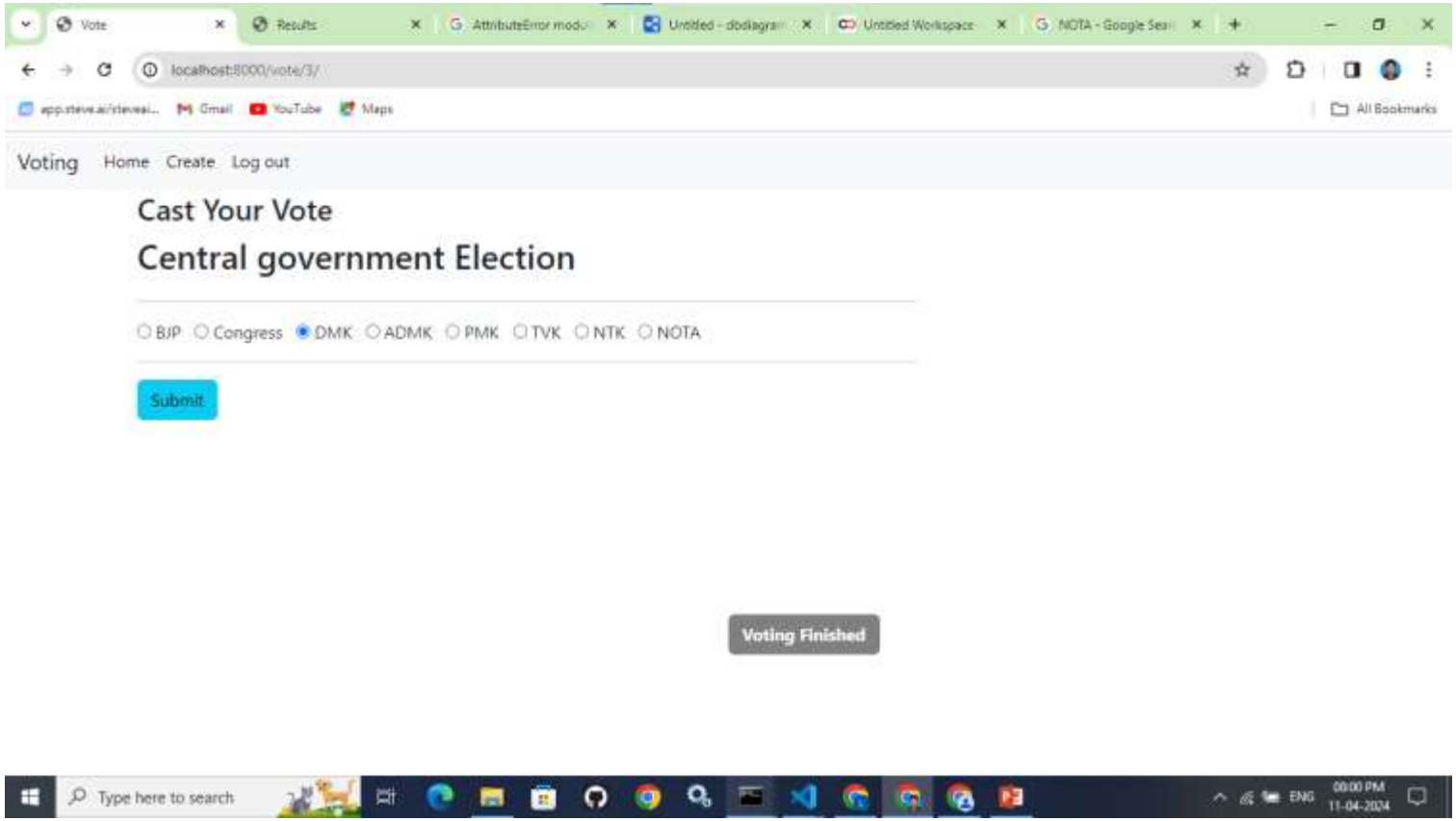
Create Poll


Voting
Home
Create
Log out
Create A New Poll
Result publish DateTime:
11-04-2024 07:55 PM
Enter Poll Question
Central government Election
Option 1
BJP
Option 2
Congress
Option 3
DMK
Option 4
ADMK
Option 5
PMK
Option 6
Enter option 6

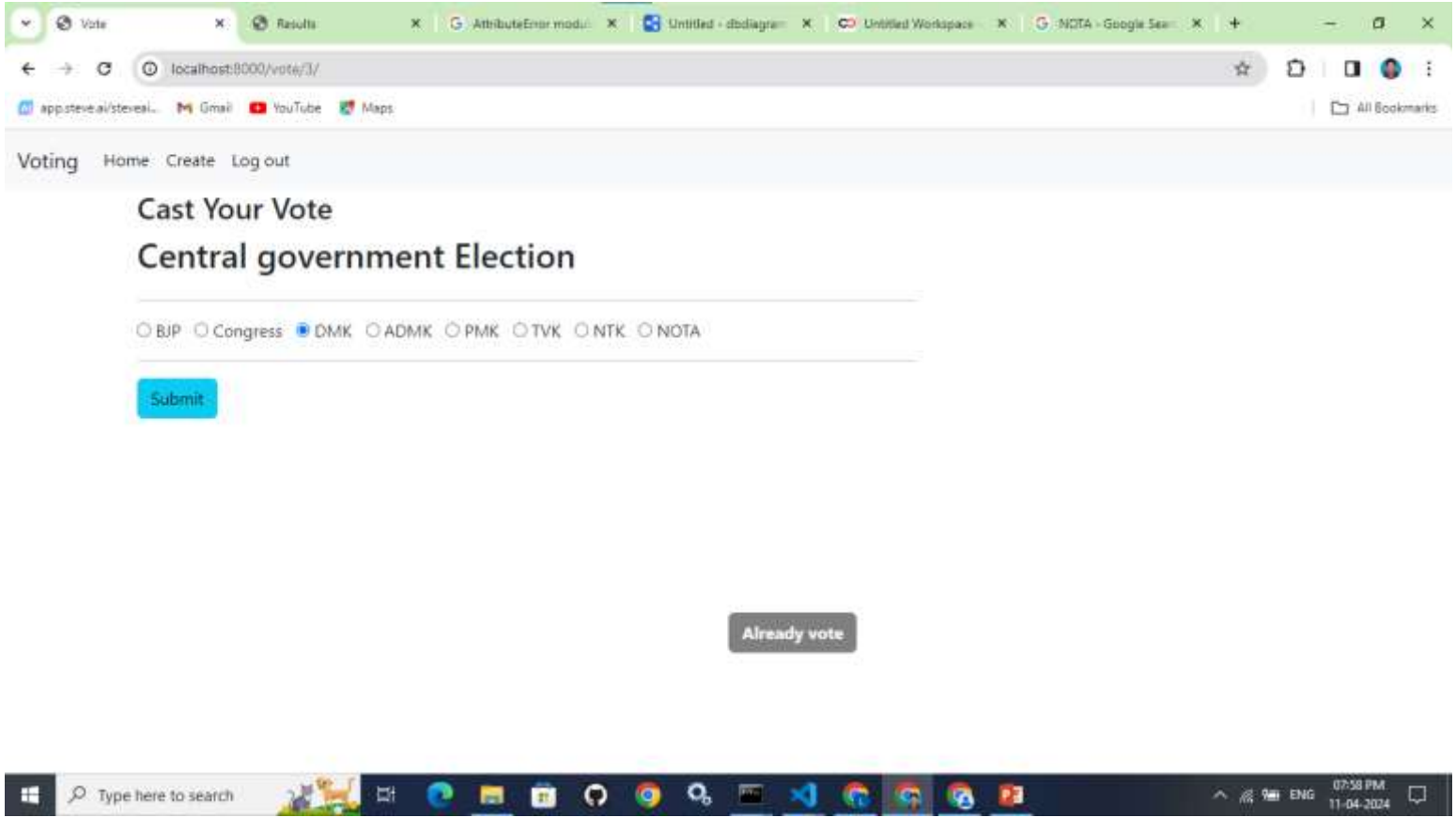
Voting


app.steve.ai/steveai...
Gmail
YouTube
Maps
All Bookmarks
Voting
Home
Create
Log out
Cast Your Vote
Central government Election
BJP
Congress
DMK
ADMK
PMK
TVK
NTK
NOTA
Submit
vote successfully
Type here to search
ENG
07:57 PM
11-04-2024

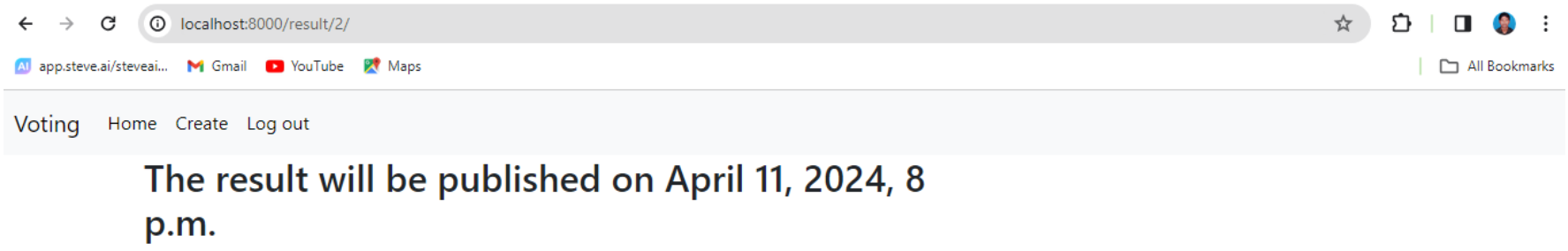
If user Vote after result:



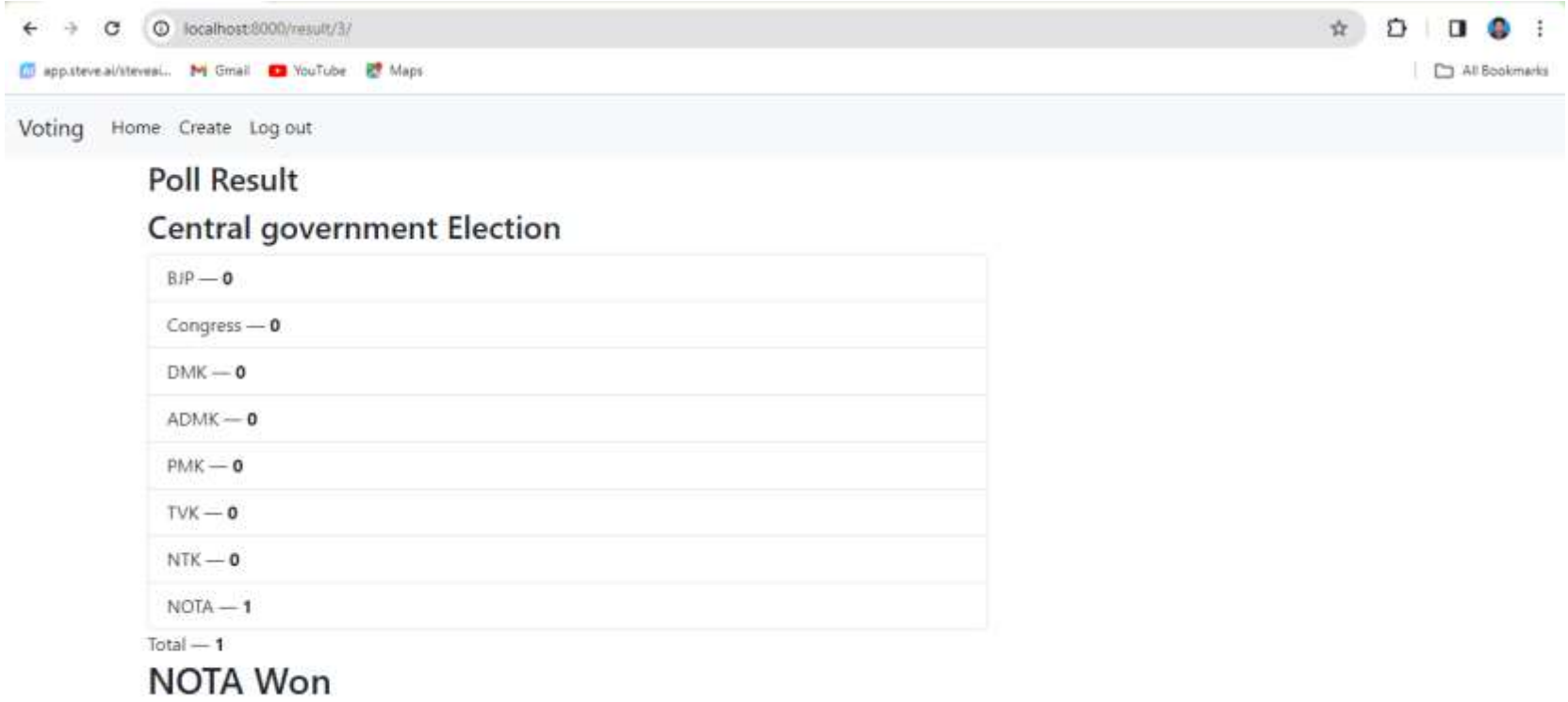
If user Vote second time:



If user click result before publish:



If user click result after publish:



Future Enhancements:

For future enhancements to your Django-based voting project, several key features and optimizations can be considered to improve functionality, scalability, and user experience. Here are some potential enhancements:

1. **User Authentication and Authorization**: Enhance user authentication with features like OAuth integration (e.g., Google, Facebook) for seamless login/signup and granular authorization controls based on user roles (admin, regular user).
2. **Improved User Interface**: Implement a modern and responsive user interface using frontend frameworks like React or Vue.js, enhancing the interactivity and user experience of the polling application.
3. **Real-time Updates**: Integrate WebSocket technology (e.g., Django Channels) to enable real-time updates of poll results, providing users with immediate feedback as votes are cast.
4. **Advanced Result Visualization**: Enhance result presentation with dynamic charts and graphs (using libraries like Chart.js or D3.js) to visually represent poll outcomes, making data interpretation more accessible.
5. **User Engagement Features**: Implement features such as commenting on polls, sharing polls on social media, or adding multimedia content (images, videos) to enrich the poll creation and voting experience.

Future Enhancements:

6.Email Notifications: Set up email notifications for users to receive updates on poll activity (e.g., new polls, poll results) and reminders to participate in ongoing poll
7.Performance Optimization: Optimize database queries with Django's query optimization techniques, use caching (e.g., Redis) for frequently accessed data, and leverage Django's built-in caching mechanisms for improved performance.
8.Internationalization and Localization: Implement support for multiple languages (internationalization) and regional preferences (localization) to make the polling application accessible to a global audience.
9.Security Enhancements: Implement additional security measures such as rate limiting, content validation, and input sanitization to protect against potential vulnerabilities and ensure data integrity.
10.API Development: Develop RESTful APIs to enable integration with other applications (mobile apps, third-party services), allowing external developers to leverage poll data programmatically.

Conclusion

- The voting project developed using Django presents a comprehensive and functional platform for creating, managing, and participating in polls. Throughout the project, key features were implemented to ensure a smooth user experience and efficient data management.
- The project leverages Django's powerful capabilities, including its ORM for database interactions, built-in authentication system for user management, and the Django templating engine for generating dynamic and responsive user interfaces.
- Users can register, log in, and create custom polls with multiple options, while authenticated users can vote on existing polls and view real-time results. The project also incorporates relational database design, with relationships established between users, questions, options, and polls, facilitating seamless data retrieval and manipulation.

Source :

Thank You!