Online Voting Platform

Project Overview

1. Technology Stack

- **Frontend**: HTML, CSS, JavaScript, Bootstrap
- **Backend**: Python (Django Framework)
- **Database**: SQLite (default), can be configured for PostgreSQL/MySQL
- **Version Control**: Git & GitHub
- **Deployment (Optional)**: Heroku / Render / PythonAnywhere

Objective

To design and develop a secure and user-friendly web-based voting platform that enables users to vote online in elections. The system ensures vote integrity, user authentication, and election management for a streamlined voting experience.

2. Key Features

- **User Authentication**
  - Registration & login system for voters and administrators
  - Role-based access control (Admin vs Voter)
- **Admin Functionalities**
  - Create and manage elections
  - Add/edit/delete candidates
  - View results and statistics
  - Manage voter registrations
- **Voter Functionalities**
  - Register and log in
  - View available elections
  - Cast vote (only once per election)
  - View election results after voting ends
- **Security Features**
  - Encrypted passwords (Django's built-in authentication)
  - Session management
  - Restriction of multiple votes
  - Admin-only election setup access

3. System Architecture

Frontend (HTML/CSS/JS) <-> Django Views <-> Django Models <-> SQLite DB

- **Models:** Represent Elections, Candidates, Votes, Users
- **Views:** Handle the business logic and render templates
- **Templates:** Render HTML interfaces for user interactions
- **Forms**: For user input and validation
- **URLs**: Route requests to appropriate views

4. Modules Description

4.1 Authentication Module

- Uses Django's built-in User model
- Login, logout, registration forms
- Role assignment (Admin/Voter)

4.2 Election Module

- Create and manage elections
- Set election start and end times
- Toggle election visibility

4.3 Candidate Module

- Admin adds candidates to specific elections
- Candidate profile: Name, Photo, Description

4.4 Voting Module

- Voters see ongoing elections
- Can vote only once per election
- Vote is recorded in Vote model

4.5 Results Module

- Once voting ends, results are computed
- Admin and users can view election results
- Pie charts/graphs for visual stats (optional via Chart.js)

5. Future Enhancements

- OTP/email verification for voter authentication
- Blockchain integration for tamper-proof voting
- SMS/email notifications
- Voter ID upload and approval
- Responsive UI improvements

6. Conclusion

This project successfully demonstrates the development of a secure and functional Online Voting Platform using Django. It incorporates authentication, vote casting, and result management, providing a foundation for scalable, digital electoral processes.

DEPLOYMENT LINK