

Secure Online College Voting System

Jasim Muhammed Kasim K

Afzal A.s

Abdul Rahman V.P

Vignesh Manikandan .P



Visionary Leadership

Rajitha M

Assistant Professor

Department of Computer Science

Abstract: Secure College Elections

- A **web-based application** developed to digitize and secure college elections.
- Students vote online using their **college ID**, verified by the admin.
- Enforces the principle of **one person – one vote**.
- Provides **real-time result viewing** for transparency.
- Ensures **tamper-proof voting** through secure login protocols and admin-controlled access.
- Built using the **Django framework** with separate modules for **admins** and **voters**.
- Optional features: **OTP/email verification** and **downloadable result logs**.
- Offers a **reliable, accessible, and efficient** alternative to outdated manual voting method

System Modules

Admin Module

- Manages the entire election process including uploading and managing real-time students voter data such as college Id and email.
- Can create, edit or delete election events and approve or reject candidates as well as voter registrations.
- Monitor real time vote counts, publish or download election results in CSV/PDF formats

Result & Audit Modules

- The admin can download detailed vote statistics including total votes, candidate-wise counts and turnout report.
- the activities are logged with timestamps, voter ID and IP addresses for audit purpose.

Notification & Communication

- Sends one time password (OTP) to student's email during login
- After a student votes, a confirmation message is shown on the screen as well as a message also deliver to their respective email

Student (voter) Module

- Secure and user-friendly interface for students upon logging using their College ID
- Voters can view ongoing election, cast a single vote per election, receive on screen confirmation, check published results and get email after voting

Verification & Authentication Module

- Admins upload a pre-verified list containing college ID, name, class and email
- the system checks and sent OTP to the corresponding email and verifies with a one-time 5-minutes expiry limit
- this sub module ensures one person - one vote mechanisms

Existing Manual System

The current college election system is primarily manual and paper-based. Students must be physically present at polling booths to cast votes using printed ballots. Voter identity is manually verified via college ID cards or printed lists, often by faculty or volunteers. The entire process—from registration to counting—is time-consuming and prone to human error.

Manual vote counting increases the risk of miscounts, manipulation, and delays. There is minimal transparency, no real-time result tracking, and no audit trail. Absent students cannot participate, reducing turnout and inclusivity. This traditional approach lacks automation, is vulnerable to malpractice, and offers limited accessibility, highlighting the need for a secure, efficient, and transparent online alternative.



Drawbacks of Existing Systems



Lack of Security

EVM are prone to be hacked or manipulated with hardware/firmware



Time-Consuming

Manual processes are inefficient.



Limited Accessibility

Remote voters cannot participate.



Trust Issues

Political Parties have publicly claimed against EVM which increased dis-believes



No Real-Time Results

Without VVPAT(voter verifiable paper audit trial) voters cannot confirm if their votes are recorded correctly



High Operational Costs

Due to printing and staffing.



Negative Environmental Impact

Paper and physical resources usage.

Key Characteristics

1 Web-Based Platform

Accessible from any device with internet connectivity.

2 College ID-Based Verification

Ensures only eligible students can vote.

3 OTP-Based Secure Login

Adds a second layer of security for login.

4 One-Person-One-Vote Enforcement

Prevents duplication or fraud.

5 Admin-Controlled Management

Admins handle voter data, elections, verification, and results.

6 Real-Time Vote Counting

Votes are tallied automatically for quick insights.

7 Result Transparency

Results are publicly visible to students after polls close.

8 Audit-Ready Logs

Tamper-proof audit trail for verification.

9 Email Communication System

Notifies users of OTPs, confirmations, and reminders.

10 Time-Limited Voting Access

Voter portal active only during official election periods.

11 Responsive and User-Friendly Interface

Easy to use on both mobile and desktop devices.

12 Secure Session Handling

Prevents unauthorized access after logout or timeout.

Software Specification

Frontend

- HTML5 – Structuring the web pages
- CSS3 – Styling and layout design
- Bootstrap or Tailwind CSS – Responsive and mobile-friendly UI components
- JavaScript – Client-side interactivity and dynamic content
- Django Templates (or React.js - optional) – Rendering pages and components

Backend

- Python – Core programming language
- Django Framework – Full-stack web framework for backend logic, routing, and server-side rendering
- Django REST Framework (optional) – For building APIs if required Database
- MySQL – Secure and scalable relational database for production

Hardware Specification

1. Server (where the system is hosted)

- **Processor:** Intel i5 / AMD equivalent (quad-core) or higher
- **RAM:** Minimum **8 GB** (16 GB recommended for smooth performance)
- **Storage:** 250 GB SSD (fast database & log storage)
- **Network:** High-speed internet connection (minimum 100 Mbps)
- **Backup:** External storage / cloud backup for logs & results

2. Client (Students/Admin Devices)

- **Processor:** Dual-core (Intel i3 or higher)
- **RAM:** 4 GB minimum
- **Storage:** 50 GB (basic OS + browser support)
- **Display:** Standard 720p / 1080p resolution
- **Browser:** Latest version of Chrome / Firefox / Edge
- **Network:** Internet connection with at least 1–2 Mbps speed

Conclusion

The SecureOnlineCollege Voting System offers a modern, efficient, and reliable alternative to traditional manual voting methods in college elections. By leveraging a secure web-based platform built with Django, the system ensures authenticity, transparency, and ease of access for both administrators and student voters.

Features such as OTP-based login, college ID verification, real-time result tracking, and audit logs enhance both the security and integrity of the voting process. This system not only reduces administrative burden and operational costs but also promotes higher student participation through remote access and timely notifications.

By addressing the key challenges of the existing system—such as tampering risks, manual errors, and limited accessibility—this solution successfully creates a tamper-proof, one-person-one-vote, and fully transparent digital voting environment. This project demonstrates how technology can be effectively used to modernize democratic processes within academic institutions, paving the way for more inclusive, secure, and trustworthy elections.