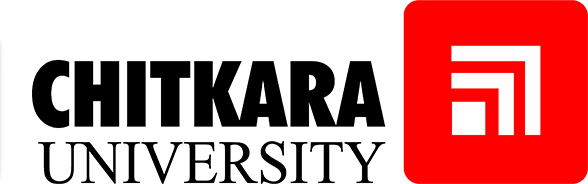
**Database Management System**

Semester-III (Batch-2024)

Online Voting And Election System



**Supervised By: Submitted By:**

Mr. Sachine Rishabh Aggarwal,2410991813 (G-15)

Rishi,2410991815 (G-15)

Rythme,2410991812 (G-15)

**Department of Computer Science and Engineering** **Chitkara University Institute of Engineering & Technology,**

**Chitkara University, Punjab**

### **Online Voting And Election System**

**1. Introduction**

Voting stands as a fundamental pillar of democracy, requiring the collection and management of highly sensitive data that directly shapes governance. Traditionally, elections have relied on paper ballots and manual counting—methods that can be time-consuming, susceptible to errors, and inefficient for large-scale events.

In today’s digital era, the demand for a secure, transparent, and reliable Online Voting System is more pressing than ever. The proposed solution offers a unified platform for voter registration, secure vote casting, identity verification, and the real-time generation of accurate election results.

This system is designed not only to simplify the voting process but also to enhance security, transparency, and accessibility for both voters and administrators. By incorporating strong encryption techniques and database management principles, it ensures data integrity, confidentiality, and scalability, making it suitable for use in institutions, organizations, and even national elections

**2. Problem Overview**

Online voting involves multiple interconnected entities, including **Voters, Candidates, Elections, Political Part, Results, and Administrators**. Without a well-structured voting system, several challenges emerge:

* Delays in processingdue to **manual verification** of voter eligibility and vote counting.
* **Data inconsistency and redundancy** resulting from scattered or poorly maintained records.
* Difficulty in **tracking voter participation** and election results across regions.
* Lack of **centralized management**, limiting transparency and coordination.
* **Security vulnerabilities**, such as unauthorized access, duplicate voting, or vote tampering.
* **Limited monitoring and reporting**, restricting real-time oversight, analytics, and decision-making.

For solving these issues a well-designed Online Voting System can offer a centralized, secure, and relational platform with clearly defined entities, relationships, and constraints. This ensures accuracy, efficiency, security, and transparency throughout the electoral process.

**3. Scope of the Project**

The aim of this project is to design and implement a relational database framework capable of handling all essential elements of a secure and scalable voting environment. Specifically, the system will:

* **Data Storage & Organization**: Maintain detailed information about voters, polling stations, candidates, election officials, parties, and cast ballots in a consistent format.
* **Entity Relationships**: Model the real-world interactions between voters, elections, and results through a well-structured schema that minimizes redundancy and ensures data integrity.
* **Querying & Operations**: Support efficient retrieval of voter lists, candidate details, polling results, and election summaries, while also allowing fast updates (e.g., vote counts, voter validations).
* **Scalability**: Be designed to accommodate expansion, such as an increase in the number of registered voters, future elections, and additional constituencies without performance issues.
* **Security & Access Control**: Incorporate mechanisms to safeguard sensitive electoral data, ensuring privacy, controlled access, and protection against tampering or unauthorized use.

**4. Objectives**

The key objective for make this project is to provide a safe ,secure and reliable platform of online voting and election system for the user and ensure the data validity and safety

* **Centralized Election Data Repository**Develop a unified database that eliminates duplication while keeping complete, accurate records of voters, candidates, and results.
* **Quick & Accurate Data Access**Optimize the database for fast retrieval of voter lists, live results, candidate information, and real-time election summaries.
* **Clear Mapping of Relationships**Model essential voting relationships such as:
  + Voters linked to polling constituencies.
  + Candidates associated with political parties and elections.
  + Votes tied to both voters (validated) and candidates.
  + Officials managing specific polling stations.
* **Data Integrity & Reliability**Use primary keys, foreign keys, and constraints to prevent duplication, fake votes, or inconsistent election records.
* **High-Level Security Measures**Apply encryption, secure login systems, and role-based permissions to protect voter identities and ensure anonymity of ballots.
* **Scalable Infrastructure**Design to accommodate expansion in voters, additional elections (local, state, national), and simultaneous voting events.
* **Reports & Analytics Support**Provide detailed reports including:
  + Voter turnout percentages.
  + Candidate-wise vote counts.
  + Live election dashboards.
  + Post-election statistical analysis.
* **Minimal Redundancy & Optimized Storage**Apply normalization techniques to ensure efficient storage, faster performance, and consistency across records.

**5. Significance of the Project**

## The proposed Online Voting and Election System will significantly transform the election process by:

* **Ensuring Security and Privacy**Protecting voter identities and election data through encryption, authentication, and strict access controls to prevent fraud, tampering, or unauthorized usage.
* **Enhancing Efficiency in Voting**Replacing manual paperwork and ballot counting with a digital platform that enables quick voter registration, instant vote recording, and automated tallying of results.
* **Improving Accuracy and Transparency**Eliminating human counting errors and ensuring that every vote is correctly recorded and reflected in real-time, thereby increasing public trust in the electoral process.
* **Strengthening Accessibility and Inclusiveness**Allowing eligible voters to conveniently cast their votes online, regardless of their location, thereby making the democratic process more inclusive.
* **Providing Real-Time Reports and Analytics**Generating instant results, turnout statistics, and performance insights for election administrators and stakeholders, enabling timely decision-making and clear communication.
* **Supporting Scalability for Future Elections**Offering a robust and flexible architecture that can be adapted for institutional, organizational, or national-level elections with thousands or even millions of voters.

**Main Entities and Attributes:**

## **1. Voter**

Stores details of all registered voters and their authentication credentials.

* Voter\_ID (Primary Key)
* Name
* Date\_of\_Birth
* Gender
* Address
* Aadhaar\_No / Govt\_ID (for verification & authentication)
* Voter\_Status (Active / Inactive / Already Voted / Not Voted)

## **2. Candidate**

Represents candidates contesting in elections.

* Candidate\_ID (Primary Key)
* Name
* Party\_Name (or Party\_ID if linked to Party entity)
* Age
* Gender
* Symbol (image/file of party symbol)
* Constituency\_ID (Foreign Key → Constituency)

## **3. Election**

Contains details of an election event (parliamentary, assembly, local, etc.).

* Election\_ID (Primary Key)
* Election\_Name (e.g., “General Elections 2025”)
* Election\_Type (Parliamentary / Assembly / Local)
* Constituency\_ID (Foreign Key → Constituency)

## **4. Constituency**

Defines constituencies/regions in which elections are held.

* Constituency\_ID (Primary Key)
* Name
* State / Region
* Type (Rural / Urban / Reserved)
* No of Voters

## **5. Admin / Election Commission**

Stores details of system administrators and election officials.

* Admin\_ID (Primary Key)
* Role (System Admin / Returning Officer)
* Username
* Password
* Email

## **6. Political Party**

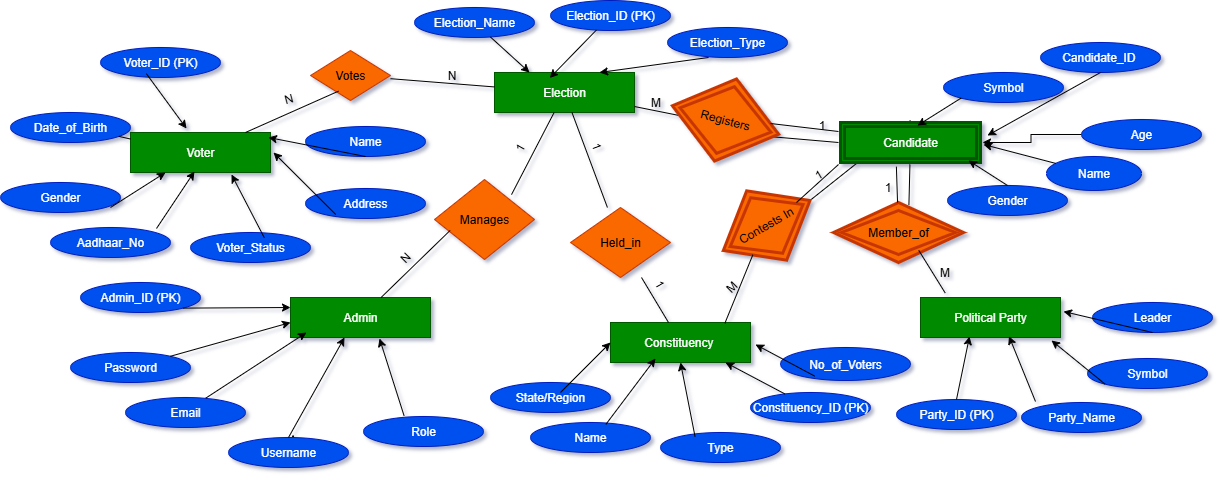
Stores metadata about political parties.

* Party\_ID (Primary Key)
* Party\_Name
* Symbol (Party logo/image if not in Candidate table)
* Leader

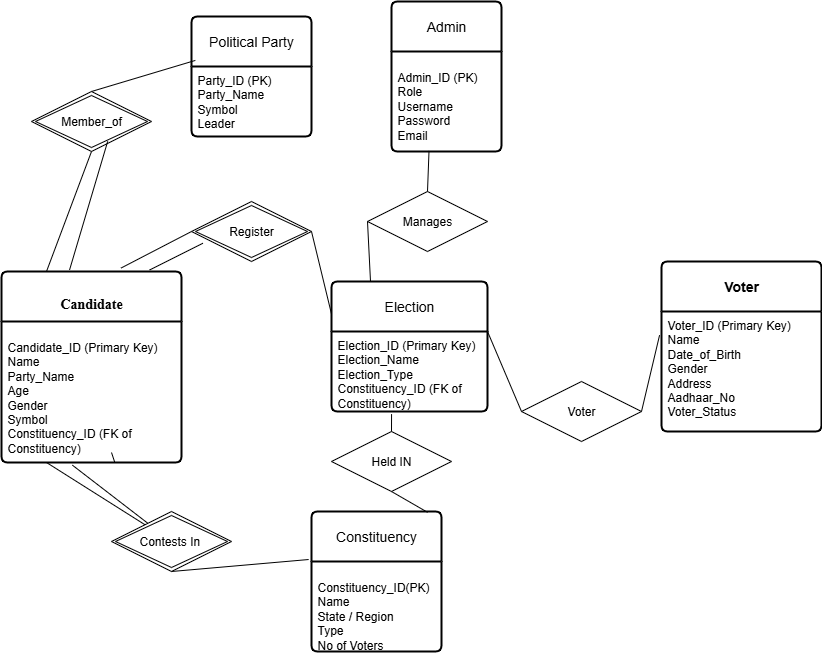
**Relationships:**

* **Voter-Election**: Many-to-Many (Many votes can vote in many elections which happen in a year)
* **Political Party-Election**: One-to-Many (One political party can take part in many election)
* **Candidate-Election**: Many-to-One (One election have many candidates at a time)
* **Candidate-Constituency:** Many-to-One (A constituency can have many candidates at a time)
* **Candidate- Political Party:** Many-to-One (A political party can have many candidates standing for them)

**ER DIAGRAM**

****

**ER MODAL**

****

**ER to Relational Model (Schema)**

**Voter**(Voter\_ID **PK,**Name,Date\_of\_Birth,Gender,Address,Aadhaar\_No Govt\_ID,Voter\_Status)

**Candidate**(Candidate\_ID **PK**,Name,Age,Gender,Symbol,Party\_ID **FK** → Political\_Party(Party\_ID),Constituency\_ID **FK** → Constituency(Constituency\_ID))

**Election(**Election\_ID **PK**,Election\_Name,Election\_Type,Constituency\_ID **FK** Constituency(Constituency\_ID))

**Constituency**(Constituency\_ID **PK**,Name,State/Region,Type,No\_of\_Voters)

**Admin**(Admin\_ID **PK**,Role,Username,Password,Email)

**Political\_Party**(Party\_ID **PK**,Party\_Name,Symbol,Leader)

Vote(Vote\_ID PK, Voter\_ID FK → Voter(Voter\_ID), Election\_ID FK → Election(Election\_ID), Candidate\_ID FK → Candidate(Candidate\_ID))

**Relational Algebra (RA)**

Q1. Get all details of voters who are Active.

**→ σ Voter\_Status= ′Active′ ​(Voter)**

Q2. Get names and Aadhaar numbers of all voters.

**→ π Name, Aadhaar\_No​ (Voter)**

Q3. Find names of voters who have not voted.

**→ π Name​(σ Voter\_Status= ′NotVoted′​ (Voter))**

Q4. Retrieve voter names who are either Inactive or Already Voted.

**→ π Name​(σ Voter\_Status= ′Inactive′​ (Voter)) ∪ π Name​(σ Voter\_Status=′AlreadyVoted′​(Voter))**

Q5. Find voters who are Active and Not Voted yet.

**→ π Name​(σ Voter\_Status= ′Active′​ (Voter)) ∩ π Name​(σ Voter\_Status= ′NotVoted′ ​(Voter))**

Q6. Get voters with their constituency name (assuming Address ~ Constituency.Name).

**→ π Voter.Name,Constituency.Name (Voter ⋈Voter.Address=Constituency.Name Constituency)**

Q7. Get names of all candidates

**→ π Name​ (Candidate)**

Q8. Find details of female candidates.

**→ σ Gender= ′Female′​ (Candidate)**

Q9. List candidates above 40 years old.

**→ σ Age > 40 ​(Candidate)**

Q10. Get candidate names with their party names**.**

**→ π Candidate.Name,Party.Party\_Name​(Candidate ⋈Candidate.Party\_Name=Party.Party\_Name​Party)**

Q11. Candidates contesting in constituency ID = 101.

**→ π Candidate.Name​(σ Constituency\_ID= 101​ (Candidate))**

Q12. Find candidates who are from Party “ABC” or “XYZ”.

**→ σ Party\_Name= ′ABC′​ (Candidate)∪σ Party\_Name= ′XYZ′ ​(Candidate)**

Q13. Candidates belonging to Reserved constituencies.

**→πCandidate.Name​(Candidate⋈Candidate.Constituency\_ID=Constituency.Constituency\_ID​ σ Type= ′Reserved′​ (Constituency))**

Q14. Get all election names.

**→ πElection\_Name​(Election)**

Q15. Find all Parliamentary elections.

**→ σ Election\_Type= ′Parliamentary′​ (Election)**

Q16. List election IDs and their constituency IDs.

**→ π Election\_ID, Constituency\_ID​ (Election)**

Q17. Elections of type Assembly or Local.

**→ σ Election\_Type=′Assembly′​(Election)∪σ Election\_Type=′Local′​(Election)**

Q18. Parliamentary elections in constituency ID = 101.

**→ σ Election\_Type=′Parliamentary′​(Election) ∩ σ Constituency\_ID=101​(Election)**

Q19. List elections with their constituency name.

**→ π Election.Election\_Name,Constituency.Name​(Election ⋈Election.Constituency\_ID=Constituency.Constituency\_ID​Constituency)**

Q20. List candidates contesting in “General Elections 2025”.

**→πCandidate.Name(Candidate⋈Candidate.Constituency\_ID=Election.Constituency\_ID σ Election\_Name=′GeneralElections2025′(Election))**

Q21. Get all constituency name.

**→ π Name​ (Constituency)**

Q22. Find urban constituencies.

**→ σ Type= ′Urban′​ (Constituency)**

Q23. Find constituencies in Delhi**.**

**→ σ State= ′Delhi′ ​(Constituency)**

Q24. Constituencies that are Rural or Reserved.

**→σType=′Rural′​(Constituency)∪σType=′Reserved′​(Constituency)**

Q25. List constituencies with candidates contesting in them.

**→πConstituency.Name,Candidate.Name​(Constituency⋈Constituency.Constituency\_ID=Candidate.Constituency\_ID​Candidate)**