

**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

Online Voting System

**A PROJECT REPORT**

**Submitted to**

**Department of Computer Application**

**Academia International College**

***In partial fulfillment of the requirements for the Bachelors in Computer Application***

Submitted by

Khagendra Malla

[6-2-346-15-2021]

Sujal Bajracharya

[6-2-346-35-2021]

April 2025

Under the Supervision of

**Ananda Adhikari**



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**Academia International College**

**Supervisor’s Recommendation**

I hereby recommend that this project prepared under my supervision by Khagendra Malla &

Sujal Bajracharya entitled “**Online Voting System**” in partial fulfillment of the requirements for the degree of Bachelor of Computer Application is recommended for the final evaluation.

**SIGNATURE**

Ananda Adhikari

**SUPERVISOR**

Lecturer

IT Department

Gwarko chowk, Lalitpur



**Tribhuvan University**

**Faculty of Humanities and Social Sciences**

**Academia International College**

# LETTER OF APPROVAL

This is to certify that this project prepared by Khagendra Malla & Sujal Bajracharya entitled “**Online Voting System**” in partial fulfillment of the requirements for the degree of Bachelor in Computer Application has been evaluated. In our opinion it is satisfactory in the scope and quality as a project for the required degree.

|  |  |
| --- | --- |
| **SIGNATURE of Supervisor**  Name and Academic designation  Department name and full address of the College | **SIGNATURE of HOD/Coordinator**  Name and Academic designation  Department name and full address of the College |
| **SIGNATURE of Internal Examiner Internal Examiner** | **SIGNATURE of Internal Examiner External Examiner** |

# ABSTRACT

This project details the development of an Online Voting System aimed at addressing inefficiencies and security concerns in traditional voting. Key objectives include enhancing accessibility, efficiency, accuracy, and security. This mid-term report covers the initial phases, including requirements analysis, feasibility assessment, and system design progress. The system utilizes PHP, MySQL, HTML, CSS, and JavaScript. Initial progress includes user interface mockups and database schema design. The final system is expected to provide a secure, transparent, and cost-effective platform for various election types.

# ACKNOWLEDGEMENT

We would like to thank our Supervisor, coordinator, Assistant coordinator, teachers and fellow students in their contribution to the development of our project.

**Table of Contents**

[LETTER OF APPROVAL 3](#_Toc196146851)

[ABSTRACT 4](#_Toc196146852)

[ACKNOWLEDGEMENT 5](#_Toc196146853)

[List of Abbreviations 7](#_Toc196146854)

[List of Figures 8](#_Toc196146855)

[List of Tables 9](#_Toc196146856)

[Chapter 1: Introduction 10](#_Toc196146857)

[1.1 Introduction 10](#_Toc196146858)

[1.2 Problem Statement 10](#_Toc196146859)

[1.3 Objectives 10](#_Toc196146860)

[1.4 Scope and Limitations 11](#_Toc196146861)

[1.5 Report Organization 11](#_Toc196146862)

[Chapter 2: Background Study & Literature Review 12](#_Toc196146863)

[2.2 Literature Review 13](#_Toc196146864)

[Chapter 3: System Analysis and Design 14](#_Toc196146865)

[3.1 System Analysis 14](#_Toc196146866)

[3.1.1 Requirement Analysis 14](#_Toc196146867)

[3.1.2 Feasibility Study 14](#_Toc196146868)

[3.2 System Design 14](#_Toc196146869)

[3.3 Algorithm 15](#_Toc196146870)

[Chapter 4: Implementation and Testing (Progress) 16](#_Toc196146871)

[4.1 Implementation Status 16](#_Toc196146872)

[4.2 Testing Status 16](#_Toc196146873)

[Chapter 5: Conclusion and Future Work 17](#_Toc196146874)

# List of Abbreviations

* CSS: Cascading Style Sheets
* EVM: Electronic Voting Machine
* FPTP: First-Past-The-Post
* HTML: HyperText Markup Language
* IIA: Independence of Irrelevant Alternatives
* IRV: Instant Runoff Voting
* JS: JavaScript
* MySQL: My Structured Query Language
* PHP: PHP: Hypertext Preprocessor
* RCV: Ranked Choice Voting
* SRS: Software Requirements Specification
* STV: Single Transferable Vote
* UI: User Interface

# List of Figures

# List of Tables

# Chapter 1: Introduction

## 1.1 Introduction

Traditional voting systems, relying on paper ballots or Electronic Voting Machines (EVMs), often face logistical hurdles, potential for miscounts, security vulnerabilities, and significant administrative costs. These challenges can impact voter turnout and trust in the electoral process. The advancement of digital technology offers an opportunity to modernize elections through online voting systems. Such systems aim to provide a more accessible, efficient, and potentially more secure platform for casting and counting votes. This project undertakes the development of such an online system tailored for academic or organizational elections, focusing on usability and core voting functionalities.

## 1.2 Problem Statement

The primary problems with existing traditional voting methods include:

* **Inefficiency:** Manual counting and logistical arrangements are time-consuming and resource-intensive.
* **Accessibility Issues:** Physical presence requirements can disenfranchise voters who are remote, ill, or have mobility issues.
* **Security Concerns:** Paper ballots can be lost or tampered with, while EVMs can face concerns regarding transparency and unauthorized access.
* **High Costs:** Expenses related to ballot printing, polling station setup, and personnel are substantial.
* **Lack of Transparency:** Verifying votes accurately and transparently can be challenging in traditional systems.

This project addresses these issues by proposing a web-based online voting system designed to be secure, efficient, accessible, and transparent.

## 1.3 Objectives

The main objective is to develop a functional and secure Online Voting System. Specific objectives include:

* To enhance **accessibility** via a user-friendly web interface for remote voting.
* To improve **efficiency** by automating vote casting and counting.
* To ensure **accuracy** in vote tabulation, minimizing errors and manipulation.
* To implement robust **authentication** mechanisms to prevent fraudulent voting.
* To achieve **cost-effectiveness** by utilizing digital infrastructure.
* To design a **scalable** system adaptable to various election sizes.
* To promote **transparency** through secure vote verification mechanisms.

## 1.4 Scope and Limitations

* **Scope:** The system will facilitate user registration, secure login, ballot presentation based on election type, casting of votes for multiple voting methods (e.g., FPTP, Approval, RCV), secure vote storage, automated counting, and result presentation. It will include administrator functionalities for managing elections, users, and results.
* **Limitations:** This is an academic project; therefore, aspects like large-scale deployment infrastructure, advanced cryptographic techniques beyond standard practices (e.g., homomorphic encryption), and legal compliance for national elections are outside the scope. The focus is on demonstrating core functionalities and addressing security at a level appropriate for organizational/academic use.

## 1.5 Report Organization

This mid-term report details the progress made in the first half of the project lifecycle. Chapter 2 discusses the background and literature review. Chapter 3 presents the system analysis and design undertaken so far. Chapter 4 outlines the initial implementation and testing efforts. Chapter 5 concludes with the current status and planned future work. References and appendices follow.

# Chapter 2: Background Study & Literature Review

## 2.1 Background Study

**2.1.1 Existing Voting Systems**

Traditional voting relies on paper ballots or EVMs. While widely used, these face challenges like cost, logistics, and potential inaccuracies. Online voting systems have emerged as alternatives, aiming to leverage technology for efficiency and accessibility. Early trials and implementations exist, but concerns regarding security and digital divides remain prominent areas of research.

**2.1.2 Challenges in Online Voting**

* **Security:** Ensuring voter authentication, preventing coercion, maintaining ballot secrecy, and securing the transmission and storage of votes against tampering or hacking are paramount challenges. Cryptographic techniques are essential but complex to implement correctly.
* **Fairness and Accuracy:** Designing systems that accurately reflect voter intent without bias is complex. Social choice theory highlights inherent difficulties. Arrow's Impossibility Theorem demonstrates that no ranked voting system can simultaneously satisfy a set of desirable fairness criteria (like non-dictatorship, Pareto efficiency, and independence of irrelevant alternatives - IIA) when there are three or more options. Common methods like Plurality, Borda Count, and IRV all violate at least one of these axioms. This implies trade-offs are necessary. The spoiler effect, often linked to IIA violations, where a minor candidate alters the outcome between major ones, is a practical concern.
* **Accessibility and Digital Divide:** While aiming to increase accessibility, online systems might exclude those without internet access or digital literacy.
* **Transparency and Verifiability:** Voters need assurance that their vote was recorded correctly and contributes accurately to the final tally, without compromising anonymity. End-to-end verifiable (E2EV) systems are an active research area but often complex.

**2.1.3 Relevant Technologies**

The proposed system leverages common web technologies:

* **Frontend:** HTML, CSS, JavaScript for user interface and interaction.
* **Backend:** PHP for server-side logic and database interaction.
* **Database:** MySQL for storing user data, election details, and encrypted votes.
* **Security:** Standard practices like HTTPS, password hashing, input validation, and potentially basic encryption for vote data will be employed.

**2.1.4 Supported Voting Methods**

The system aims to support various methods, including:

* First-Past-The-Post (FPTP)
* Approval Voting
* Ranked Choice Voting (RCV) / Instant Runoff Voting (IRV)
* Single Transferable Vote (STV)
* Score Voting
* Condorcet Method The implementation will need to carefully consider the aggregation logic for each method, acknowledging the theoretical limitations discussed (e.g., Arrow's Theorem ).

## 2.2 Literature Review

The design and implementation of secure and reliable online voting systems have been subjects of extensive academic research and practical experimentation. Significant literature explores the vulnerabilities and potential solutions associated with digital elections.

**2.2.1 Security and Authentication**

Security remains the paramount concern. Research highlights the need for robust voter authentication to prevent impersonation and ensure eligibility. Techniques explored range from traditional username/password systems to multi-factor authentication (MFA) and more advanced methods like biometric authentication, although the latter often raises privacy concerns. Protecting vote secrecy and integrity during transmission and storage requires strong cryptographic measures, such as end-to-end encryption (E2EE). Ensuring the backend infrastructure is resilient against denial-of-service attacks, malware, and insider threats is also critical. While mentioned as an area of innovation, complex cryptographic solutions like zero-knowledge proofs or fully homomorphic encryption, which allow computation on encrypted data, are generally considered outside the scope of typical academic projects due to their complexity. Similarly, blockchain technology has been proposed for enhancing transparency and immutability, offering a decentralized ledger for recording votes. However, implementing blockchain securely and scalably for voting presents its own set of challenges, including key management and ensuring voter anonymity. Note: The proposal mentions blockchain, but the prototype report limitations suggest focusing on more standard database security for this academic project.

**2.2.2 Fairness, Paradoxes, and Voting Methods**

Social choice theory, particularly Arrow's Impossibility Theorem and the Gibbard-Satterthwaite Theorem, provides critical insights into the inherent trade-offs in voting system design. Arrow demonstrated that no ranked voting system with 3+ options can simultaneously satisfy Unrestricted Domain, Non-Dictatorship, Pareto Efficiency, IIA, and Social Ordering. This implies that all ranked systems involve compromises. Common paradoxes illustrate these issues:

Condorcet Paradox: Majority preferences can be cyclical (A>B, B>C, C>A), meaning no single winner defeats all others pairwise. This challenges the idea of a clear majority will.

Spoiler Effect: Violations of IIA mean a non-winning candidate can change the outcome between the main contenders. Plurality voting is highly susceptible, but ranked methods like Instant Runoff Voting (IRV) can also exhibit this.

The Gibbard-Satterthwaite theorem adds that any non-dictatorial, deterministic, ordinal voting system with 3+ outcomes is susceptible to strategic voting (manipulation). Voters may have incentives to vote insincerely (e.g., "lesser-evil" voting in Plurality, "burying" in Borda Count) to achieve a preferred outcome.

Given these theoretical limitations, research explores various voting methods supported by this project and their properties:

* First-Past-The-Post (FPTP): Simple but prone to spoiler effects and may not elect a majority-preferred candidate.
* Ranked Choice Voting (RCV) / Instant Runoff Voting (IRV): Aims to achieve majority support through vote transfers but can violate monotonicity and IIA, leading to potential strategic voting.
* Single Transferable Vote (STV): Used for multi-member districts, aims for proportional representation but involves complex counting.
* Approval Voting: Voters approve multiple candidates; considered simple and potentially less prone to spoilers than FPTP. Studies suggest it can reflect consensus well.
* Score Voting (Range Voting): Allows nuanced preference expression but can be vulnerable to strategic score inflation/deflation.
* Condorcet Methods: Aim to elect the Condorcet winner (if one exists), considered resistant to spoilers but computationally complex and can fail if cycles exist.

# Chapter 3: System Analysis and Design

## 3.1 System Analysis

### 3.1.1 Requirement Analysis

1. **Functional Requirements:**
   * User Registration (Voter, Administrator)
   * Secure User Login/Authentication
   * Election Creation & Management (Admin: setup candidates, dates, voting method)
   * Ballot Presentation (Display candidates/options according to election)
   * Vote Casting (Support for selected methods like FPTP, Approval, RCV etc.)
   * Vote Encryption & Secure Storage
   * Automated Vote Counting/Tallying (per selected method)
   * Result Generation & Display
   * Basic Vote Verification Mechanism (e.g., confirmation code)
2. **Non-Functional Requirements:**
   * **Security:** Protect against unauthorized access, data breaches, vote tampering.
   * **Usability:** Intuitive and accessible interface for diverse users.
   * **Reliability:** System should function correctly and consistently.
   * **Scalability:** Ability to handle a moderate number of voters and elections.
   * **Performance:** Responsive interface and timely vote processing.
   * **Maintainability:** Code should be well-structured and documented.

### 3.1.2 Feasibility Study

* **Technical Feasibility:** The project utilizes standard, widely available technologies (PHP, MySQL, HTML/CSS/JS) with ample documentation and community support, making it technically feasible.
* **Operational Feasibility:** The system is designed with a simple UI, requiring minimal training for voters and administrators, ensuring operational feasibility.
* **Economic Feasibility:** Development relies on open-source software (PHP, MySQL) and standard hardware, minimizing direct costs and making it economically viable for academic purposes.

**System Modelling (Structured Approach)**

**Data Modelling: ER Diagram** (Placement for ER Diagram: Insert ER Diagram here). Entities and relationships similar to the previous draft, focusing on Users, Elections, Candidates, Votes, EligibleVoters using MySQL data types.

**Process Modelling: DFD** (Placement for DFDs: Insert Context Diagram, Level 0 DFD, Level 1 DFDs here).

• Context Diagram: Shows interactions between Voter/Administrator and the "Online Voting System".

• Level 0 DFD: Major processes: Manage Users, Manage Elections, Conduct Voting, Tally Results. Data stores: User\_Store, Election\_Store, Candidate\_Store, Vote\_Store (representing MySQL tables).

• Level 1 DFDs: Decompose major processes (e.g., Conduct Voting into Authenticate Voter, Display Ballot, Record Vote).

## 3.2 System Design

**Architectural Design** The system follows a traditional Web Server Architecture suitable for PHP applications:

* Client Tier (Browser): Renders HTML, CSS, executes JavaScript for interactivity and makes requests to the server.
* Web Server Tier (XAMPP - Apache): Hosts the PHP application files. Apache handles HTTP requests, passing PHP requests to the PHP interpreter.
* Application Logic Tier (PHP): PHP scripts handle business logic, process user input, interact with the database, and generate HTML responses.
* Data Tier (XAMPP - MySQL): MySQL database stores all application data. PHP interacts with MySQL using extensions like MySQLi or PDO.

(Placement for Architecture Diagram: Insert a diagram illustrating this architecture, showing Browser, Apache Web Server, PHP Interpreter, and MySQL Database interaction, potentially within the XAMPP context).

**Database Schema Design** (Placement for Database Schema: Insert the detailed MySQL schema here, based on the ERD). Example Snippet (MySQL syntax):

SQL

CREATE TABLE Users (

UserID INT AUTO\_INCREMENT PRIMARY KEY,

Username VARCHAR(50) UNIQUE NOT NULL,

HashedPassword VARCHAR(255) NOT NULL,

Role ENUM('Voter', 'Admin') NOT NULL,

FullName VARCHAR(100),

Email VARCHAR(100) UNIQUE

-- other fields

) ENGINE=InnoDB;

CREATE TABLE Elections (

ElectionID INT AUTO\_INCREMENT PRIMARY KEY,

Title VARCHAR(255) NOT NULL,

Description TEXT,

StartDate DATETIME NOT NULL,

EndDate DATETIME NOT NULL,

VotingMethod VARCHAR(50) NOT NULL, -- e.g., 'FPTP', 'RCV', 'Approval'

Status VARCHAR(20) NOT NULL DEFAULT 'Pending', -- e.g., 'Pending', 'Active', 'Closed'

AdminUserID INT,

FOREIGN KEY (AdminUserID) REFERENCES Users(UserID)

) ENGINE=InnoDB;

-- Other tables: Candidates, Votes, EligibleVoters...

**Interface Design (UI/UX)**

* Focus: Simplicity and clarity using standard HTML, CSS, and JavaScript.
* Voter UI: Login page, dashboard listing elections, ballot page adapting to the voting method, confirmation page.
* Admin UI: Secure login, dashboard for election/user management, results view. Use clear forms and tables.
* Responsiveness: Basic responsiveness using CSS for usability on different screen sizes.
* (Placement for UI Mockups/Wireframes: Insert mockups/wireframes for key screens developed using HTML/CSS/JS).

**Physical DFD** (Placement for Physical DFD: Insert Physical DFD here). Shows mapping to browser, Apache server process, PHP scripts, MySQL database server within the XAMPP environment.

## 3.3 Algorithm

Vote Casting:

1. Authenticate voter.
2. Fetch relevant election details and candidates.
3. Present ballot according to the specified voting method (e.g., checkboxes for Approval, ranked list for RCV).
4. Capture voter input.
5. Encrypt vote data.
6. Store encrypted vote with voter/election identifiers and timestamp.
7. Generate and display a unique confirmation code.

Vote Tallying (Example: FPTP):

1. Decrypt votes for the specific election (requires secure key management).
2. Iterate through votes, incrementing count for the selected candidate.
3. Determine candidate with the highest count.

Security Mechanisms (Planned):

* HTTPS for encrypted communication.
* Password hashing (e.g., bcrypt).
* Input validation to prevent injection attacks.
* Role-based access control (Voter vs. Admin).
* Basic encryption for stored vote data.

# Chapter 4: Implementation and Testing (Progress)

## 4.1 Implementation Status

* Development Environment: Setup complete (XAMPP).
* Database Schema: Initial schema created in MySQL(XAMPP) based on design.
* User Interface: Basic HTML/CSS structure for main pages (Login, Dashboard) developed.
* User Authentication: Backend logic for user registration and login (with password hashing) implemented in PHP.

## 4.2 Testing Status

* Unit Testing: Initial tests performed on user authentication module functions.
* Interface Testing: Basic checks on page rendering and navigation.

# Chapter 5: Conclusion and Future Work