**Online Voting / Polling System – Project Synopsis**

**Shreyas G K – U03ME23S0019**

**Mohammed Bilal – U03ME23S0049**

**Problem Statement**

In recent years, concerns over the transparency and fairness of voting processes have grown. Allegations of vote tampering, rigging, and lack of audit trails have caused public distrust in electoral systems. Even in educational institutions and organizations, managing free and fair elections using traditional paper-based or uncontrolled online methods is challenging.

There is a need for a secure, transparent, and user-friendly online voting system that prevents multiple voting, ensures accurate results, and encourages participation, while being accessible and simple to implement in smaller-scale environments like schools, colleges, and internal organizational polls.

**Objectives**

* To develop a web-based online voting platform that ensures one user, one vote policy.
* To implement secure authentication of users via registration and login.
* To allow poll creation and management by authorized users or admins.
* To enable real-time vote counting and result display.
* To provide a basic admin dashboard to manage users, polls, and view statistics.
* To create a reliable backend system using PHP and MySQL, supporting full CRUD operations.

**Project Description**

This project is a dynamic web application that enables registered users to vote on polls created by admins or other users. The application supports:

* User authentication
* Poll creation and management
* One-time voting enforcement
* Real-time result display
* A clean and responsive user interface

The system is designed for schools, colleges, clubs, or small organizations needing a simple and transparent voting platform. It incorporates fundamental web development concepts including form handling, session management, database operations, and optional AJAX.

**Technology Stack**

| Layer | Tools/Technologies |
| --- | --- |
| Frontend | HTML, CSS, (JavaScript for AJAX or charts) |
| Backend | PHP (for form handling, sessions, and logic) |
| Database | MySQL (data storage for users, polls, votes) |
| Server Environment | Apache with XAMPP/WAMP/LAMP |
| Optional Libraries | Chart.js for visual results, jQuery for AJAX |

**System Architecture**

**➤ Client Side:**

* Displays all UI components (polls, login, forms)
* Sends form submissions (votes, poll creation) via HTTP POST
* Optionally fetches live results using AJAX

**➤ Server Side (PHP):**

* Handles form processing
* Validates user sessions
* Controls poll visibility and voting rights
* Interacts with the database securely using prepared statements

**➤ Database Layer (MySQL):**

* Stores persistent data like users, polls, votes, and inquiries
* Enforces relationships and constraints (e.g., one vote per user per poll)

**Modules / Features**

**1. User Registration & Login**

* Secure login system using hashed passwords.
* Session management using $\_SESSION.
* Role-based access (admin vs regular user).

**2. Poll Creation (Admin or Users)**

* Create polls with 2+ options and optional expiry date.
* Store options in JSON format or a related table.
* Set poll status as active or closed.

**3. Voting Module**

* Registered users can vote only once per poll.
* Votes are stored in a separate table.
* Results update in real-time (via refresh or AJAX).

**4. Poll Result Display**

* Show vote count and percentage per option.
* Optional chart-based visual (pie or bar charts).
* Support expired polls display as "Closed".

**5. Contact Form**

* For general inquiries or feedback.
* Stores messages in inquiries table.
* Admin can view messages via dashboard.

**6. Admin Dashboard**

* View all polls, votes, users, and inquiries.
* Create or delete polls.
* Moderate content or ban users.

**Database Design**

| **Table Name** | **Fields** |
| --- | --- |
| users | id, name, email, password, is\_admin |
| polls | id, title, created\_by, options\_json, status, expiry\_date |
| votes | id, user\_id, poll\_id, selected\_option |
| inquiries | id, name, email, message |

**CRUD Operations Mapping**

| **Module** | **Create** | **Read** | **Update** | **Delete** |
| --- | --- | --- | --- | --- |
| Users | Registration | Login/Profile | Password Update | Admin/User deletion |
| Polls | Create Poll | View Poll | Edit Poll | Delete Poll |
| Votes | Cast Vote | View Results | Change vote | Remove vote |
| Inquiries | Submit Message | Admin View | Mark as replied | Delete inquiry |

**Advanced Features**

* Live poll updates via AJAX (without page refresh)
* Chart.js-based real-time result graphs
* Poll expiration and automatic status change
* Anonymous voting toggle for specific polls
* Admin analytics: most voted polls, active users, etc.

**Expected Outcomes**

* A fully functional, responsive web voting system accessible on all devices.
* Authenticated user system ensuring one vote per person.
* Admin capabilities to manage the platform.
* A scalable backend structure to allow more features in the future.
* Improved transparency and fairness in small-scale elections.

**Conclusion**

This Online Voting System addresses the real-world problem of voting transparency and participation by providing a simple, secure, and intuitive web-based platform. Through the integration of core web technologies (HTML, CSS, PHP, MySQL), it demonstrates the application of CRUD operations, form handling, and session management in a meaningful context.

The system is easily extensible, making it suitable for deployment in schools, clubs, organizations, and small institutions, promoting fair digital participation.