



Web Development Minor Project

Online Voting System

Submitted by:

Name: Pema Dorji

UID: 23BCA10400

Section: 23BCA6"A"

Submitted to:

Mrs. Preeti(E15171)

Online Voting System Case Study Report

1. Introduction

The advent of digital technology has revolutionized numerous aspects of modern society, including the electoral process. The online voting system is designed to offer a secure, efficient, and user-friendly alternative to traditional paper-based voting methods. By leveraging web technologies such as PHP, MySQL, and responsive CSS, this system simplifies voter registration, authentication, and ballot casting, while ensuring that each voter can only cast one vote per election.

This project demonstrates how a robust digital infrastructure can streamline election administration by reducing manual errors, expediting vote counting, and enhancing accessibility for a diverse voter base. In addition, the system provides a dedicated administrative panel for election management, where administrators can easily add, update, and manage candidate information, including uploading candidate photos and monitoring real-time voting results.

Overall, the online voting system represents a significant step toward modernizing the electoral process, ensuring transparency, security, and convenience, and ultimately contributing to a more efficient and inclusive democratic process.

2. Objectives

- Develop a fully functional PHP-based Online Voting System.
- Implement secure user authentication, including voter registration and login.
- Provide a voting interface where registered voters can select and vote for candidates.
- Create an admin panel that allows a single administrator to manage candidates (adding, editing, deleting) and to view the voting results.
- Ensure that vote counts are hidden from the main admin panel and are only displayed on the results page.
- Design the system with a clean and responsive UI using HTML, CSS, and JavaScript.

3. System Requirements

3.1 Hardware Requirements

- **Server:** A web server with Apache.

- **RAM:** Minimum 2 GB.
- **Storage:** Minimum 10 GB free space.

3.2 Software Requirements

- **Backend:** PHP 7.4 or higher, MySQL configured to run on port 3307.
- **Frontend:** HTML, CSS, JavaScript.
- **Development Environment:** XAMPP for local development.

4. System Design

4.1 Database Design

The system uses two main tables:

- **users:** Stores all user information.
Columns: id, username, password, role (ENUM: 'admin', 'user'), and voted (BOOLEAN).
- **candidates:** Stores candidate information.
Columns: id, name, designation, photo_path, and votes.

The database is named **voting**.

4.2 System Architecture

The application follows a standard PHP web application architecture:

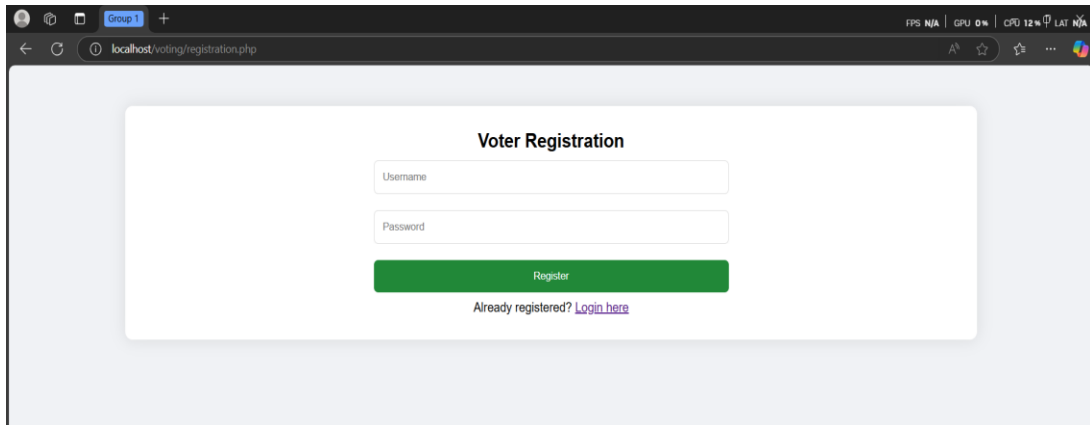
- **Presentation Layer:** HTML/CSS for user interfaces (homepage, voter registration, login, voting, admin panel, and results).
- **Business Logic Layer:** PHP scripts that handle registration, login, vote processing, candidate management, and result display.
- **Data Access Layer:** MySQL database accessed via MySQLi.

5. Implementation

5.1 Modules Description

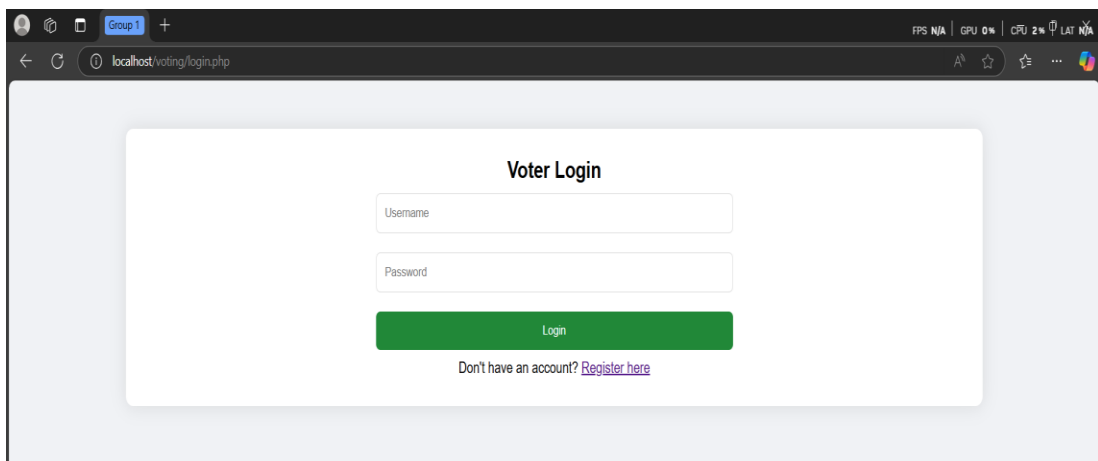
User Authentication

- **Registration Module:** Voters can register via the registration form.



A screenshot of a web browser displaying the 'Voter Registration' form. The browser's address bar shows 'localhost/voting/registration.php'. The form is centered on a light gray background and contains the following elements: a title 'Voter Registration', a 'Username' input field, a 'Password' input field, a green 'Register' button, and a link 'Already registered? [Login here](#)'.

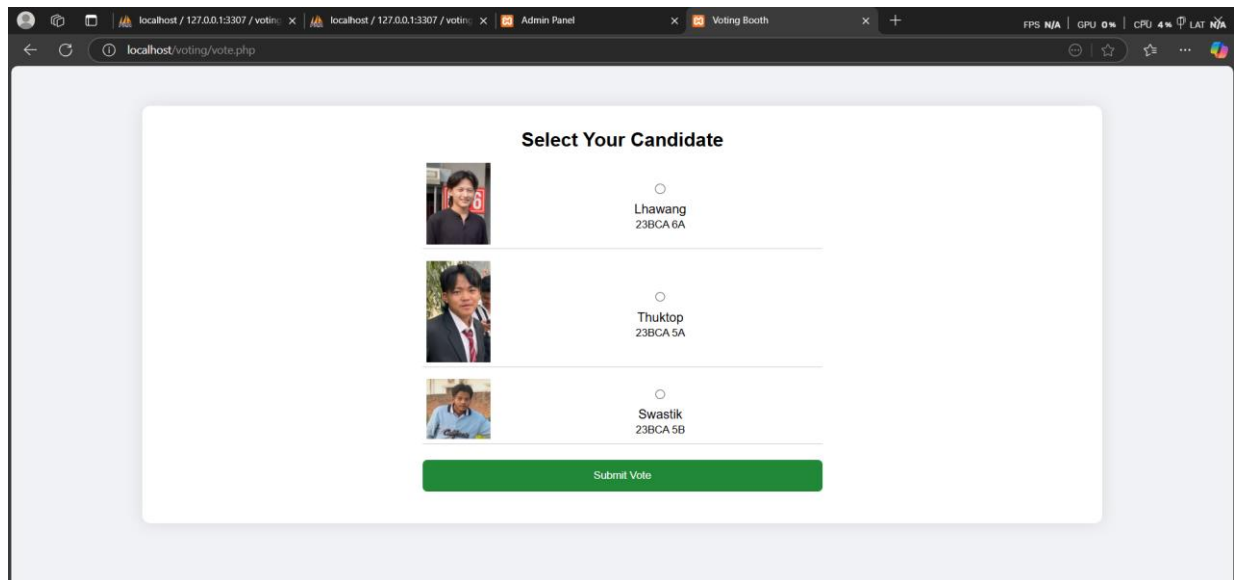
- **Login Module:** Voters and the admin can log in using their credentials. If a voter has already voted, they are prevented from logging in and shown a prompt.



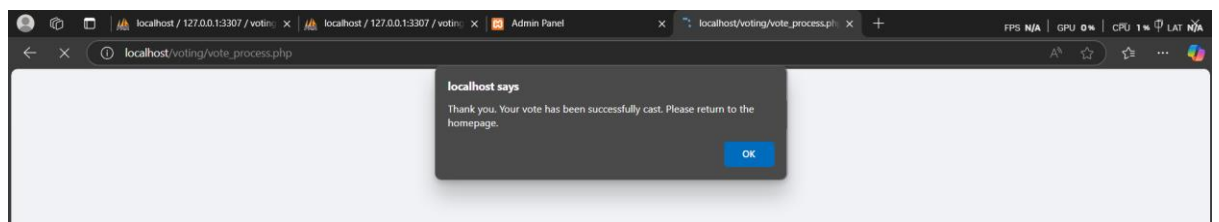
A screenshot of a web browser displaying the 'Voter Login' form. The browser's address bar shows 'localhost/voting/login.php'. The form is centered on a light gray background and contains the following elements: a title 'Voter Login', a 'Username' input field, a 'Password' input field, a green 'Login' button, and a link 'Don't have an account? [Register here](#)'.

Voting Module

- **Voting Interface:** Displays a list of candidates (with small photos, names, and designations) for voters to select.

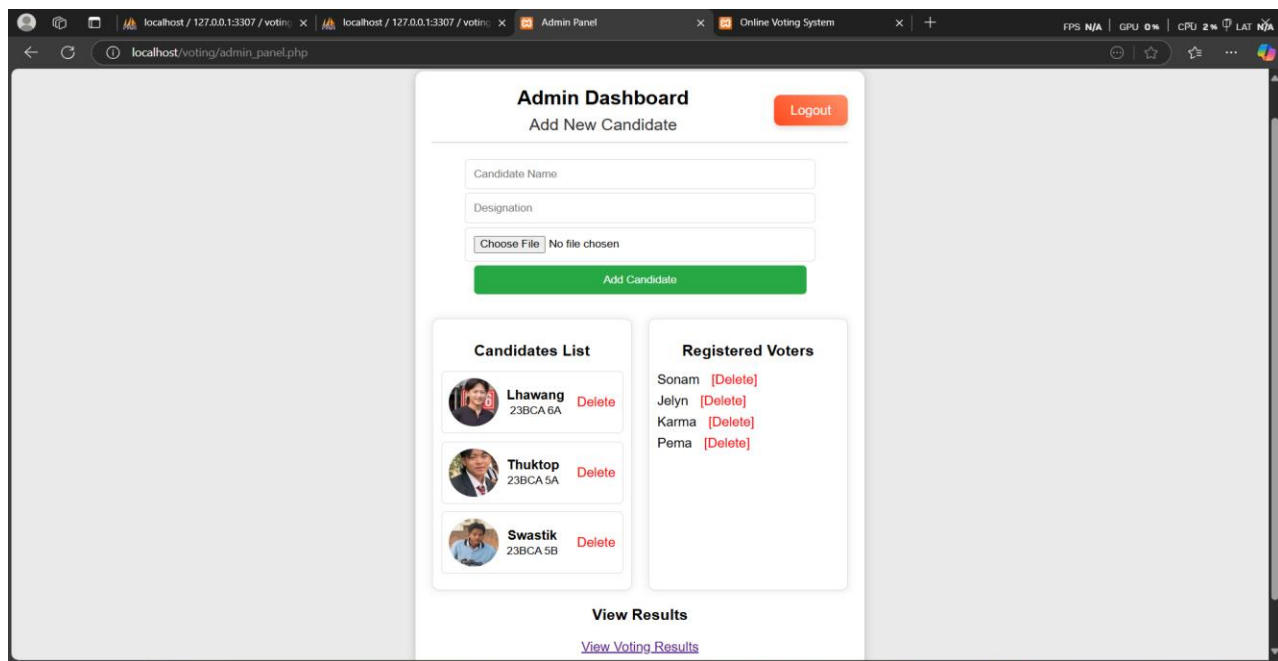


- **Vote Processing:** When a vote is cast, the selected candidate's vote count is incremented and the voter is marked as having voted.

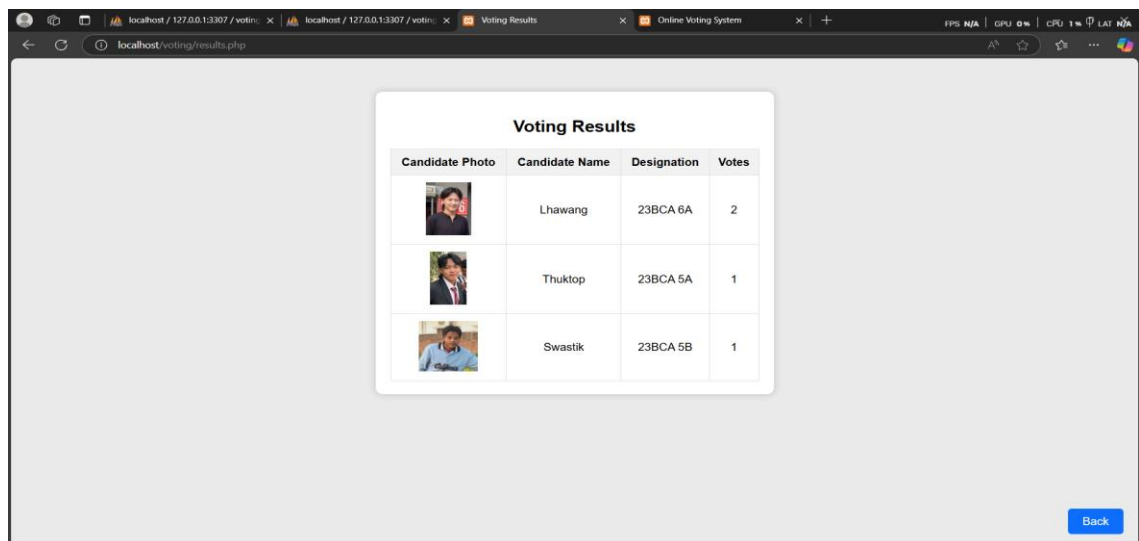


Admin Panel

- **Candidate Management:** The admin can add new candidates (with photo uploads), view the list of candidates (without vote counts), and delete candidates.
- **Voter Management:** The admin panel includes a list of registered voters with an option to delete voters.
- **Navigation:** A fancy logout button and a well-aligned header are included to improve usability and aesthetics.



- **Results Viewing:** A separate results page displays all candidate details—including photos, names, designations, and vote counts—in a table. The vote counts are hidden from the main admin panel.

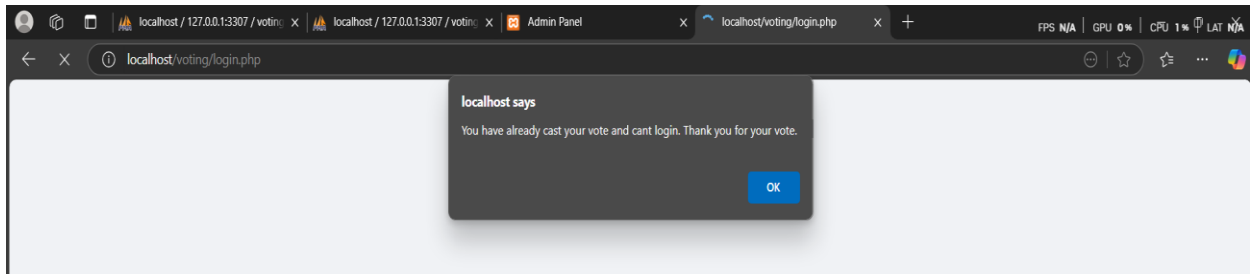


5.2 Key Code Features

- **Session Management:** PHP sessions are used to manage login states for voters and the admin.
- **File Upload:** Candidate photos are uploaded to an uploads folder.
- **Transaction Handling:** Vote submissions are handled using MySQL transactions to ensure consistency.
- **Responsive Design:** CSS is used to create a clean, responsive interface for both the admin and voter pages.

6. Testing

- **User Registration and Login:** Verify that voters can register and log in, and that the admin can log in with the default credentials (username: admin, password: 12345).
- **Voting Process:** Ensure that once a voter casts a vote, they are prevented from logging in again.



- **Admin Operations:** Check that the admin can add, delete, and manage candidates and voters.
- **Results Display:** Confirm that the results page accurately shows candidate details and vote counts, and that the “Back” button functions correctly.

7. Challenges & Solutions

- **Preventing Multiple Votes:**
The system uses a voted flag in the users table to ensure each voter can vote only once.
- **File Upload & Image Handling:**
Candidate images are stored in an uploads folder and displayed with CSS that limits their size.
- **Session and Role Management:**
Separate sessions and role checks ensure that only the admin can access the admin panel and that voters cannot vote more than once.

8. Conclusion & Future Enhancements

The online voting system represents a significant step towards modernizing and streamlining the electoral process. By leveraging web technologies such as PHP, MySQL, and responsive CSS, the system provides a secure, efficient, and user-friendly platform for voters and administrators alike. The project successfully demonstrates how multiple user roles—namely, voters and a single administrator—can interact within a unified digital environment, ensuring that only eligible voters can cast a single vote per election while enabling the administrator to manage candidate data and view results in real time.

Furthermore, the system's design addresses key issues such as data integrity, secure user

authentication, and easy management of candidate information, including features for uploading and displaying candidate photos. While the current implementation meets essential functional requirements, there is potential for future enhancements, such as incorporating advanced security measures (e.g., password hashing and multi-factor authentication), scalability improvements, and more robust audit trails to ensure full transparency and accountability during elections.

Overall, this online voting system offers a practical solution for modern electoral challenges, paving the way for more accessible and reliable voting processes in various organizational and governmental contexts.

Future enhancements may include:

- Implementing password hashing (using **password_hash** and **password_verify**) for better security.
- Adding responsive design improvements for mobile devices.
- Incorporating additional features such as candidate editing and advanced analytics on voting patterns.
- Adding a feature to register using student id or citizen id so that a single person can only register and login once to prevent from vote multicasting by a single user under different identity.
- Adding a graph and chart to give a better visual representation on the number of votes received by the candidates.

9. References

PHP Documentation: <https://www.php.net/>

MySQL Documentation: <https://dev.mysql.com/doc/>

XAMPP Documentation: <https://www.apachefriends.org/>

Idea Source: <https://www.guvi.in/blog/php-project-ideas/>

Online Voting System Idea: <https://www.geeksforgeeks.org/online-voting-system-in-c/>

PHP Website Creation: <https://www.w3schools.com/php/>
<https://www.codeproject.com/Articles/759094/Step-by-Step-PHP-Tutorials-for-Beginners-Creating>

<https://www.zend.com/blog/php-tutorial>
<https://www.youtube.com/watch?v=CIPdU6RqBqk>

Database Creation:<https://www.geeksforgeeks.org/php-mysql-creating-database/>

Database Connection:https://www.w3schools.com/php/php_mysql_connect.asp

Homepage/Index Page:https://www.w3schools.com/howto/howto_website.asp
<https://www.youtube.com/watch?v=oOi2KIQlm6o>
https://www.youtube.com/watch?v=9_Uqyfml2s

Button Tag:https://www.w3schools.com/tags/tag_button.asp
https://www.youtube.com/watch?v=_2wARy-oevQ

Form Handling:https://www.w3schools.com/php/php_forms.asp
<https://www.geeksforgeeks.org/php-form-processing/>
<https://www.studytonight.com/php/php-form-handling>

Hyperlink:https://www.w3schools.com/Php/func_filesystem_link.asp

PHP Header: <https://www.bluehost.com/help/article/php-redirect>
<https://stackoverflow.com/questions/7467330/php-headerlocation-force-url-change-in-address-bar>

Login/Register Page: <https://www.geeksforgeeks.org/creating-a-registration-and-login-system-with-php-and-mysql/>
<https://www.geeksforgeeks.org/how-to-create-admin-login-page-using-php/>

Storage of Data: <https://www.youtube.com/watch?v=Te53-QbaUc8>
<https://www.geeksforgeeks.org/how-to-store-data-in-a-php-session/>
<https://stackoverflow.com/questions/46490966/php-best-practice-for-saving-storing-very-small-amounts-of-data>

Prompt Box:<https://www.geeksforgeeks.org/how-to-pop-an-alert-message-box-using-php/>

Image Upload:https://www.w3schools.com/php/php_file_upload.asp
<https://www.youtube.com/watch?v=adMjzWiG21U>
<https://www.geeksforgeeks.org/how-to-upload-image-into-database-and-display-it-using-php/>
<https://www.youtube.com/watch?v=JaRq73y5MJk>
<https://www.geeksforgeeks.org/how-to-upload-image-in-html/>

Table Creation:https://www.w3schools.com/php/php_mysql_create_table.asp
<https://www.youtube.com/watch?v=kZ83evWEDVE>
<https://www.youtube.com/watch?v=EbektmPPgs8>

PHP GET/POST and HTML Form:https://www.tutorialspoint.com/php/php_get_post.htm
https://www.w3schools.com/php/php_superglobals_post.asp
<https://www.youtube.com/watch?v=6AzAYU8AOHQ>
https://www.w3schools.com/html/html_forms.asp
<https://softuni.org/dev-concepts/handling-an-html-form/>

