1. Security Enhancements

✅ CSRF Protection: Added CSRF tokens in forms and validated them in processing scripts to prevent cross-site request forgery (CSRF) attacks.

✅ Session Authentication: Ensured users are properly authenticated before performing actions like voting, commenting, and managing profiles.

✅ Error Handling & Logging: Errors are now logged with `error\_log()` instead of being displayed to users, preventing sensitive details from being exposed.

✅ Secure Redirects: No output is sent before `header("Location: ...")` to ensure proper redirection without conflicts.

2. Input Validation & Sanitization

✅ Proper Data Sanitization: Used `htmlspecialchars()` and `test\_input()` functions to sanitize user input before storing or displaying it.

✅ Improved Email & Poll ID Validation: Checked for valid email formats and ensured poll IDs are positive integers before database queries.

3. SQL Query Security

✅ Prepared Statements: Replaced direct database queries with prepared statements to prevent SQL injection.

✅ Duplicate Entry Checks: Ensured that login IDs and emails were unique before inserting them into the database.

4. File Upload Security

✅ Image Validation: Verified that uploaded files are actual images before processing.

✅ File Size Limitations: Enforced a 2MB max file size for uploads to prevent oversized files.

✅ Allowed File Types: Restricted uploads to JPG, GIF, and PNG only to avoid malicious scripts disguised as images.

✅ Unique File Names: Used an MD5 hash with a timestamp to generate unique file names for uploaded profile pictures.

5. Improved User Experience

✅ Consistent Navigation (`nav.php`): Ensured session starts correctly before checking authentication.

✅ Dynamic Year in `footer.php`: Replaced hardcoded `"2025"` with `date("Y")`.

✅ Enhanced Error Messages (`error.php`): Added missing error codes to provide better feedback to users.

✅ Prevented Email Enumeration in Password Reset: Users now receive a generic message regardless of whether their email exists, preventing attackers from determining valid accounts.

6. Code Optimization & Readability

✅ Strict Comparisons (`===`): Used instead of `==` for better performance.

✅ Better Indentation & Structure: Improved readability of HTML and PHP code.

✅ Reduced Redundant Code: Removed unnecessary outputs before redirects and replaced direct outputs with structured responses.