Phase 4: Securing the Website (Deadline: 23 March 2025) (Subtotal: 32')

In this phase, you will protect your website against many popular web application security threats.

1. No XSS Injection and Parameter Tampering Vulnerabilities in the whole website
   - [UI Enhancement Only] Proper and vigorous client-side input restrictions for all forms _____/ 1'
   - Proper and vigorous server-side input sanitizations and validations for all forms _____/ 2'
   - Proper and vigorous context-dependent output sanitizations _____/ 2'
   - Proper Content Security Policy Header set to defense XSS / 2'
     Reference: https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP
2. Mitigate SQL Injection Vulnerabilities in the whole website _____/ 2'
   - Apply parameterized SQL statements; Avoid template literals.
3. Mitigate CSRF Vulnerabilities in the whole website _____/ 2'
   - Apply and validate secret nonces for every form
   - Apply an extra measure to prevent CSRF other than the nonce in form data
   - ALL forms must defend against Traditional and Login CSRF
4. Authentication for Admin Panel (Filling in the details in Lecture Notes)
   - Create a user table (or a separate DB with only one user table) _____/ 1'
     - Required columns: *userid (primary key), email, password*
     - Data: *at least 2 users of your choice, 1 admin and 1 normal user (using admin flag)*
     - Security: Passwords must be properly salted and hashed before storage
   - Build a *login page* that requests for *email* and *password* _____/ 3'
     - Upon validated and authenticated, redirect the user to the *admin panel* or main page
     - Indicate user name (or "guest" if not logged in) in your website
     - Otherwise, prompt for errors (i.e. either email or password is incorrect)
     - A separated normal user login page is not compulsory
   - Maintain an authentication token using Cookies (with httpOnly)
     - Proper name(s) and value(s); property: httpOnly _____/ 2'
     - Cookies persist after browser restart (i.e. 0 < expires < 3 days) _____/ 1'
     - No Session Fixation Vulnerabilities (rotate session ID upon successful login) _____/ 1'
     - Configure all authentication cookies to use the Secure and HttpOnly flags _____/ 1'
   - Validate the authentication token before revealing and executing admin features _____/ 3'
     - If successful, let admin users access the admin panel and **execute** admin features
     - Otherwise (e.g. empty or tampered token), redirect back to the *login page* or main page
     - Security: Both the admin panel and admin-process APIs must validate the auth. token
   - Node* & SQL: Provide a logout feature that clears the authentication token _____/ 1'
   - Supporting Change of Password _____/ 2'
     - Must validate the current password first; update the database.
     - Logout user after the password is changed
5. All generated session IDs and nonces are not guessable throughout the whole assign. _____/ 1'
   - e.g., the login token must not reveal the original password in plaintext
   - e.g., the CSRF nonce when applied in a hidden field must be random
6. Apply SSL certificate for the assigned domain.
   - Certificate Application (No other CA services are allowed except you have custom domain) _/ 2'
     - Apply the certificate via Let's Encrypt (Recommended to use certbot)
   - Certificate Installation
     - Install the issued certificate and apply security configurations to web server _____/ 1'
       - Apply strong algorithms and secure cipher suites
     - Host admin panel at https://[your_domain_name]/admin _____ / 2'

- Redirect users to **https** if they use HTTP with Nginx/Apache or other secure means.

Reference: https://wiki.apache.org/httpd/RedirectSSL, https://docs.nginx.com/nginx/admin-guide/web-server/web-server/#rewriting-uris-in-requests
See also: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security