Security Evaluation of LLM generated code

Mandatory questions are marked with an asterisk (*)

Survey Statement

We are conducting this survey as part of a study on evaluating a security mechanism (for inclusivity) generated using Large Language Models (LLMs). The following questions guide the research.

Effectiveness: How effectively can LLMs generate code that implements security and inclusivity requirements, particularly for neurodivergent users?

Influencing Factors: What factors (e.g., prompt design, model choice, and specification level) influence the quality of LLM-generated code with respect to security and inclusivity?

Evaluation: How does LLM-generated code perform when reviewed by humans and LLMs in terms of identifying security vulnerabilities and inclusivity gaps?

In this study, we used prompts with varying levels of detail regarding inclusivity considerations in the development of security mechanisms—from minimal to highly detailed specifications (three cases in total); however, static security requirements were specified across three cases.

You are requested to evaluate the code from each case for security. The survey will take approximately 20 minutes to complete.

We sincerely appreciate your time and participation.

Please note that no personally identifiable information will be collected during this survey.

Participation consent *

□ I agree to participate in the survey

Current role

____

Number of years of experience

____

Case I: No Inclusivity Specification

Security Criteria

1. Broken Access Control
   - The system must prevent cross-site request forgery (CSRF). Sessions must not allow unintended actions to be triggered.
   - The portal must not expose usernames, course folders, or other private identifiers.
   - ll sensitive routes must enforce proper access control with no insecure direct object reference (IDOR) vulnerabilities.
   - CSRF tokens must be unique per session and validated on all sensitive requests.
2. Injection (XSS)
   - Malicious input must not be able to execute scripts, steal credentials, or redirect users to fake pages.
   - All user inputs must be sanitized and outputs escaped.
   - No inline or untrusted scripts are allowed.
   - Framework-level auto-escaping must be enabled (for example React, Angular, or Django templates).
3. Security Misconfiguration
   - HTTPS must be enforced, and HSTS, CSP, and secure headers must be configured.
   - Password reset tokens must be random, single-use, and short-lived.
   - Debug information, directory listings, and stack traces must be disabled in production.
   - Weak HTTPS or unsecured networks must not expose user sessions.
   - Password reset links must not be predictable or interceptable.
   - The portal or backend must not expose patient, user, or student data due to misconfiguration.
4. Identification and Authentication Failures
   - The system must mitigate risks from fake login pages or deceptive emails (phishing).
   - The password reset flow must prevent unauthorized access.
   - Automated guessing attempts (brute force) must be throttled or blocked.
   - Passwords must be hashed using bcrypt or Argon2, never stored in plaintext.
   - Multi-factor authentication (MFA) or single sign-on (SSO) should be implemented.
   - A strong password policy must be enforced.
   - Login attempts must have throttling, CAPTCHA, or account lockout after repeated failures.
   - The password reset flow must validate tokens securely and provide clear feedback to the user.
5. SSRF and Social Engineering
   - Outgoing URLs must be validated or whitelisted to prevent SSRF or open redirects.
   - The system must not allow attackers to impersonate staff or support to gain sensitive information.

- Users must be informed of safe authentication practices, such as never sharing passwords via email.

Please access the code here:
https://gist.github.com/Japskua/d91a92fa8a509b95ea11e1a4a8686344

Security Evaluation criteria

Scale (1-5)
1: Strongly disagree, 2: Disagree, 3: Neutral, 4: Agree, 5: Strongly Agree

A01:2025 *

	1	2	3	4	5
The system checks that reset code belongs to the user. *	○	○	○	○	○
The same reset code works for any other user. *	○	○	○	○	○
The previous sessions still work after a password reset by the user. *	○	○	○	○	○

A02:2025 *

	1	2	3	4	5
The website uses HTTPs with latest TLS version. *	○	○	○	○	○
The reset codes and user data stored are using appropriate encryption algorithm. *	○	○	○	○	○
The email message guard/hide user’s private data. *	○	○	○	○	○

A03:2025 *

	1	2	3	4	5
The code sanitizes user supplied data like Email and passwords. *	○	○	○	○	○
The redirect links are safe and go only to the hospital’s site. *	○	○	○	○	○
The email templates are safe and contain no script or HTML from user text. *	○	○	○	○	○

A05:2025 *

	1	2	3	4	5
The user’s private information or tokens are shown in error pages or logs. *	○	○	○	○	○
The passwords and secret codes are stored safely and not in the code. *	○	○	○	○	○
The system allows default credentials. *	○	○	○	○	○

A07:2025 *

	1	2	3	4	5
The system bars the user from making several wrong attempts. *	○	○	○	○	○
The application show if the username/email does not exist. *	○	○	○	○	○
The password reset code is random, complex and user specific. *	○	○	○	○	○

Case II: Moderate Inclusivity Specification and Same Security Criteria

Please access the code here:
https://gist.github.com/Japskua/a0e1bea52a543e508ad870f51ce9a823

Security Evaluation criteria

Scale (1-5)
1: Strongly disagree, 2: Disagree, 3: Neutral, 4: Agree, 5: Strongly Agree

A01:2025 *

	1	2	3	4	5
The system checks that reset code belongs to the user. *	○	○	○	○	○
The same reset code works for any other user. *	○	○	○	○	○
The previous sessions still work after a password reset by the user. *	○	○	○	○	○

A02:2025 *

	1	2	3	4	5
The website uses HTTPs with latest TLS version. *	○	○	○	○	○
The reset codes and user data stored are using appropriate encryption algorithm. *	○	○	○	○	○
The email message guard/hide user’s private data. *	○	○	○	○	○

A03:2025 *

	1	2	3	4	5
The code sanitizes user supplied data like Email and passwords. *	○	○	○	○	○
The redirect links are safe and go only to the hospital’s site. *	○	○	○	○	○
The email templates are safe and contain no script or HTML from user text. *	○	○	○	○	○

A05:2025 *

	1	2	3	4	5
The user’s private information or tokens are shown in error pages or logs. *	○	○	○	○	○
The passwords and secret codes are stored safely and not in the code. *	○	○	○	○	○
The system allows default credentials. *	○	○	○	○	○

A07:2025 *

	1	2	3	4	5
The system bars the user from making several wrong attempts. *	○	○	○	○	○
The application show if the username/email does not exist. *	○	○	○	○	○
The password reset code is random, complex and user specific. *	○	○	○	○	○

Case III: Detailed Inclusivity Specification and Same Security Criteria

Please access the code here:
https://gist.github.com/Japskua/55b8f48b3fd2af999721596724722103

Security Evaluation criteria

Scale (1-5)
1: Strongly disagree, 2: Disagree, 3: Neutral, 4: Agree, 5: Strongly Agree

A01:2025 *

	1	2	3	4	5
The system checks that reset code belongs to the user. *	○	○	○	○	○

	1	2	3	4	5
The same reset code works for any other user. *	○	○	○	○	○
The previous sessions still work after a password reset by the user. *	○	○	○	○	○

A02:2025 *

	1	2	3	4	5
The website uses HTTPs with latest TLS version. *	○	○	○	○	○
The reset codes and user data stored are using appropriate encryption algorithm. *	○	○	○	○	○
The email message guard/hide user’s private data. *	○	○	○	○	○

A03:2025 *

	1	2	3	4	5
The code sanitizes user supplied data like Email and passwords. *	○	○	○	○	○
The redirect links are safe and go only to the hospital’s site. *	○	○	○	○	○
The email templates are safe and contain no script or HTML from user text. *	○	○	○	○	○

A05:2025 *

	1	2	3	4	5
The user’s private information or tokens are shown in error pages or logs. *	○	○	○	○	○
The passwords and secret codes are stored safely and not in the code. *	○	○	○	○	○

	1	2	3	4	5
The system allows default credentials. *	○	○	○	○	○

A07:2025 *

	1	2	3	4	5
The system bars the user from making several wrong attempts. *	○	○	○	○	○
The application show if the username/email does not exist. *	○	○	○	○	○
The password reset code is random, complex and user specific. *	○	○	○	○	○