Risk ID	Technical Risk	Technical Risk Indicators	CWE IDs	Impact Rating	Impact	Mitigation	Validation Steps
1	SQL Injection	Modified, missing or altered SQL tables, modification of tables in database	CWE ID 89	H	Leakage or unauthorized access to sensitive information	Sanitize input information, drop all special characters	SQL injection arguments (like 1 'or' 1 '='1) could not be executed , special character (like ") cannot be found in the database
2	Cross-site Scripting	Insertion of unexpected operations (like requests, redirects, etc.)	CWE ID 79	M	Loss of control to webpage	Sanitize input (especially ‹› because of tags)	Try inputing HTML or JS elements (like ‹script› tags), and make sure that they are never run
3	Cookie Tampering	Unexpected cookies, or potential unauthorized admin logins	CWE ID 565	L	Admin access by unauthorized users	Make cookie values encrypted, or unable to use after change; avoid cookies for security-related decision	Visiting/logging on webpages should invoke cookie integrity checks, logins after modifying cookie values should not be allowed
4	Unsalted Hashes	Hashes are not salted	CWE ID 759	M	Easy decryption of sensitive information (like passwords)	Salt cashes, or use encryption frameworks	Hash results include salts, and take significantly more time to decrypt
5	Use of Risky Cryptographic Algorithm	Using cryptographic algorithms that can be easily decrypted	CWE ID 327	M	Easy decryption of sensitive information (like passwords)	Use more sophisticated and reliable encryption	Hash results take significantly more time to decrypt
6	Path Traversal	External users can access files and directories outside the webpage presented	CWE ID 23 and 36	M	Files can be viewed by unauthorized users	Ensure that private files are authorized, and arbitrary code (such as ../)cannot be executed	File authorization and use WAFs (Web Application Firewalls); disable directory listing
7	Eval Injection	No sanitization of code before calling eval(), thus suspicious requests	CWE ID 95	H	Unauthorized users may gain access to private information (control server, database, etc)	Sanitize user input, also there is possibility to avoid risky functions such as eval()	Ensure that code (esp. PHP code in this case) are not executed through input; sanitize input

8	Information Exposure through Error Message	Information shown by error message	CWE ID 209	L	Users gain sensitive information through error message	Custom error pages that do not expose sensitive information	Customized error pages do not show sensitive information
9	Weak Password Requirements	Strong passwords are not required	CWE ID 521	M	User passwords can be easily deciphered	Require more complicated passwords, such as those with special characters and numbers	When users sign up, strong passwords are required