

SonarQube Analyse

De code vertoont veel kleine problemen op het gebied van beveiliging, betrouwbaarheid, onderhoudbaarheid en testdekking. Gelukkig zijn de oplossingen voor deze problemen niet al te complex.

High severity issues

Title: Add password protection to this database.	
File: src/database.php	Type: Security
Solution: The vulnerability can be fixed by using a strong password retrieved from an environment variable <code>MYSQL_SECURE_PASSWORD</code> . This environment variable is set during deployment. It should be strong and different for each database.	
Reason: Not having a password is not safe.	

Title: Add curly braces around the nested statement(s). (8 times)	
File: src/index.php	Type: Maintainability
Solution: Add curly braces.	
Reason: While not technically incorrect, the omission of curly braces can be misleading and may lead to the introduction of errors during maintenance.	

Title: Add curly braces around the nested statement(s). (11 times)	
File: src/move.php	Type: Maintainability
Solution: Add curly braces.	
Reason: While not technically incorrect, the omission of curly braces can be misleading and may lead to the introduction of errors during maintenance.	

Title: Add curly braces around the nested statement(s). (4 times)	
File: src/play.php	Type: Maintainability
Solution: Add curly braces.	
Reason: While not technically incorrect, the omission of curly braces can be misleading and may lead to the introduction of errors during maintenance.	

Title: Add curly braces around the nested statement(s). (9 times)	
File: src/util.php	Type: Maintainability
Solution: Add curly braces.	

Reason: While not technically incorrect, the omission of curly braces can be misleading and may lead to the introduction of errors during maintenance.

Medium severity issues

Title: Use a specific version tag for the image.

File: Dockerfile

Type: Maintainability

Solution: Use specific version tags for Dockerfile image.

Reason: While always using the latest version may seem convenient, the build cannot be repeated because it is not clear which was the last version. In addition, it can lead to unpredictability and issues such as version mismatch and potential security vulnerabilities.

Title: This statement will not be executed conditionally; only the first statement of this 2-statement block will be. The rest will execute unconditionally.

File: src/index.php

Type: Maintainability

Solution: Rework the if statement, add curly braces.

Reason: Having inconsistent indentation and omitting curly braces from a control structure, such as an if statement or for loop, is misleading and can induce bugs.

Title: This function has 4 returns, which is more than the 3 allowed.

File: src/util.php

Type: Maintainability

Solution: Turn the three if statements into a single if or statement.

Reason: Having too many return statements in a function increases the function's essential complexity because the flow of execution is broken each time a return statement is encountered. This makes it harder to read and understand the logic of the function.

Title: This function has 4 returns, which is more than the 3 allowed.

File: src/util.php

Type: Maintainability

Solution: Merge the top two if statements using or operator.

Reason: Having too many return statements in a function increases the function's essential complexity because the flow of execution is broken each time a return statement is encountered. This makes it harder to read and understand the logic of the function.