VijayalakshmiKYADGI commented 3 hours ago — Owner

No description provided.

VijayalakshmiKYADGI added 2 commits 5 days ago

`2nd commit` a9fbf14

`new commit` f807bee

VijayalakshmiKYADGI commented 3 hours ago

View reviewed changes

VijayalakshmiKYADGI left a comment — Owner, Author

AI Code Review Summary ⚠️

Open **shdbjhfbv** #4

VijayalakshmiKYADGI wants to merge 2 commits into `main` from `feature/test`

Sent by CrewAI Lead Engineer

This comment was marked as duplicate. Show comment

VijayalakshmiKYADGI commented now

View reviewed changes

VijayalakshmiKYADGI left a comment

Owner Author

AI Code Review Summary ⚠️

This review highlights several critical and high-severity security vulnerabilities that require immediate attention, including SQL injection risks, insecure password handling, hardcoded credentials, and the use of `eval()` with user input. Additionally, running the Flask app with debug mode enabled in production is a significant security concern.

There are also 12 low-severity findings related to code quality and performance, such as unused variables, inefficient imports and object instantiations within route handlers, and less safe dictionary access patterns. While not critical, addressing these would improve the overall robustness and efficiency of the codebase.

Sent by CrewAI Lead Engineer

test-project/api.py

```
16 + def get_user(user_id):
17 +     from database import Database
18 +     db = Database('app.db')
19 +     query = f"SELECT * FROM users WHERE id = {user_id}"
```

VijayalakshmiKYADGI now

Owner Author

🟠 HIGH Severity (SECURITY): SQL Injection vulnerability detected. User-controlled input 'user_id' is directly formatted into an SQL query. To prevent this, please use parameterized queries.

Agent: Lead Software Engineer

Reply...

Resolve conversation

test-project/api.py

```
25 +     search_term = request.args.get('q')
26 +     from database import Database
27 +     db = Database('app.db')
28 +     query = f"SELECT * FROM users WHERE name LIKE '%{search_term}%'"
```

VijayalakshmiKYADGI now

Owner Author

🟠 HIGH Severity (SECURITY): SQL Injection vulnerability detected. User-controlled input 'search_term' is directly formatted into an SQL query. To prevent this, please use parameterized queries.

Agent: Lead Software Engineer

Reply...

Resolve conversation

test-project/api.py

```
37 +     from database import Database
38 +     db = Database('app.db')
39 +     user = db.find_user_by_name(username)
40 +     if user and user[0][2] == password:
```

VijayalakshmiKYADGI now Owner Author

🟡 HIGH Severity (SECURITY): Plain text password comparison is insecure. Passwords should be hashed and securely compared using algorithms like bcrypt.

Agent: Lead Software Engineer

Reply...

Resolve conversation

test-project/api.py

```
38 +     db = Database('app.db')
39 +     user = db.find_user_by_name(username)
40 +     if user and user[0][2] == password:
41 +         return jsonify({'token': 'abc123', 'user_id': user[0][0]})
```

VijayalakshmiKYADGI now Owner Author

🟡 HIGH Severity (SECURITY): Hardcoded token 'abc123'. Authentication tokens should be securely generated and managed, not hardcoded. Consider using environment variables or a secrets management system.

Agent: Lead Software Engineer

Reply...

Resolve conversation

test-project/api.py

```
45 + def execute_code():
46 +     data = request.get_json()
47 +     code = data.get('code')
48 +     result = eval(code)
```

VijayalakshmiKYADGI now Owner Author

🔴 CRITICAL Severity (SECURITY): Using `eval()` with user-provided input ('code') allows for arbitrary code execution on the server, posing a severe security risk. This should be avoided.

Agent: Lead Software Engineer

Reply...

Resolve conversation

test-project/api.py

```
49 +     return jsonify({'result': result})
50 +
51 + if __name__ == '__main__':
52 +     app.run(debug=True, host='0.0.0.0')
```

VijayalakshmiKYADGI now Owner Author

🟡 HIGH Severity (SECURITY): Running Flask application with 'debug=True' in a production environment is a security risk. It exposes detailed error messages and can enable code execution. Please disable debug mode for production deployments.

Agent: Lead Software Engineer

Reply...

Resolve conversation

test-project/auth_handler.py

```
...  ...  @@ -0,0 +1,21 @@
     1  + import os
     2  + import sqlite3
```

VijayalakshmiKYADGI now Owner Author

🟡 HIGH Severity (SECURITY): Hardcoded API key 'sk-abc123hardcoded'. Sensitive credentials should never be stored directly in source code. Please use environment variables or a secrets management system.

Agent: Lead Software Engineer

Reply...

Resolve conversation

test-project/auth_handler.py

```
...  ...  @@ -0,0 +1,21 @@
     1  + import os
     2  + import sqlite3
     3  + import pickle
```

VijayalakshmiKYADGI now Owner Author

🟡 HIGH Severity (SECURITY): Hardcoded password 'admin123'. Sensitive credentials should never be stored directly in source code. Please use environment variables or a secrets management system.

Agent: Lead Software Engineer

Reply...

Resolve conversation

Reviewers

No reviews

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

1 participant