ASSIGNMENT – 20

Name-JANAGANI VARSHITHA

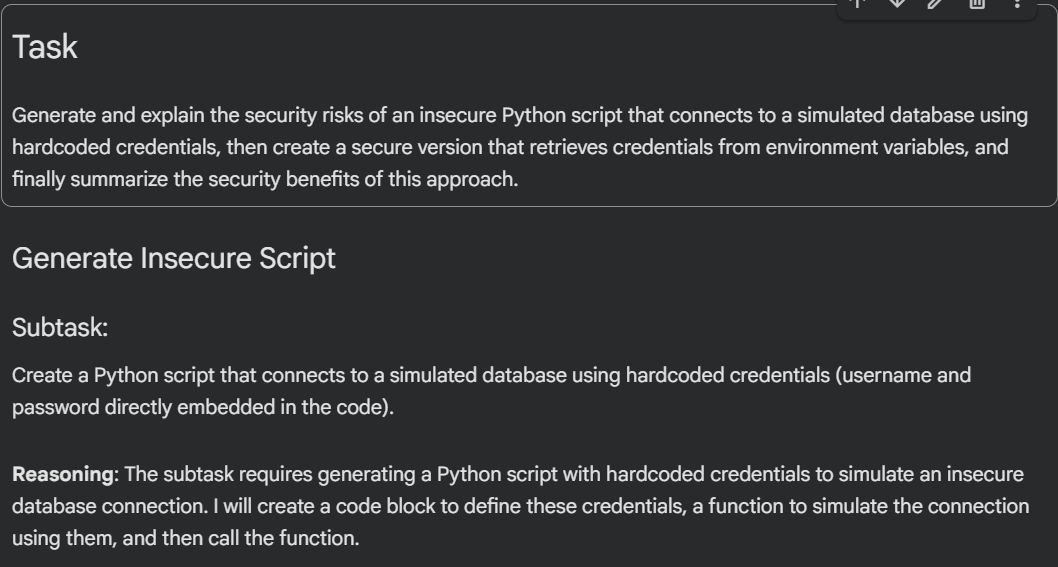
Batch-04

2403a54085

-----------------------------------------------------------------------------------------------------------

**Task 1: Detect Hardcoded Credentials**

PROMPT:  
Generate a Python script that connects to a database and includes hardcoded credentials. Then show a secure version that loads the username and password from environment variables using os.getenv().  
  
CODE and OUTPUT:



A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A black background with white text

AI-generated content may be incorrect.  
  
**Task 2: Identify SQL Injection Vulnerabilities**

Prompt:

Generate a simple Python login using SQLite that contains a SQL-injection vulnerability.  
Show a test using the input ' OR '1'='1 and the vulnerable SQL it produces.  
Then rewrite the code to use parameterized queries (e.g., cursor.execute(..., (user, pwd))) to prevent SQL injection.

CODE AND OUTPUT:

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

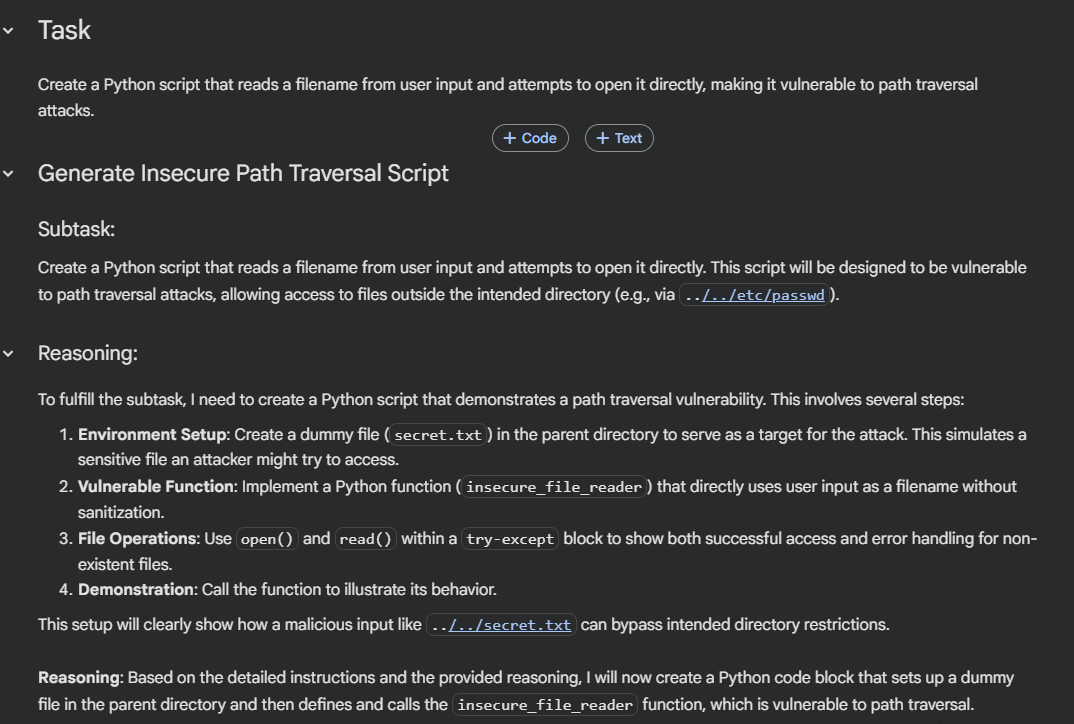
A screenshot of a computer

AI-generated content may be incorrect.

**Task 3: Detect Unsafe File Handling**

PROMPT:

Generate a Python script that reads a filename from user input and opens it (insecure version that allows path traversal like ../../etc/passwd).  
Then provide a secure version that only allows files inside a data/ folder (validate/sanitize the path or normalize and check the final path).

CODE AND OUTPUT:  




A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

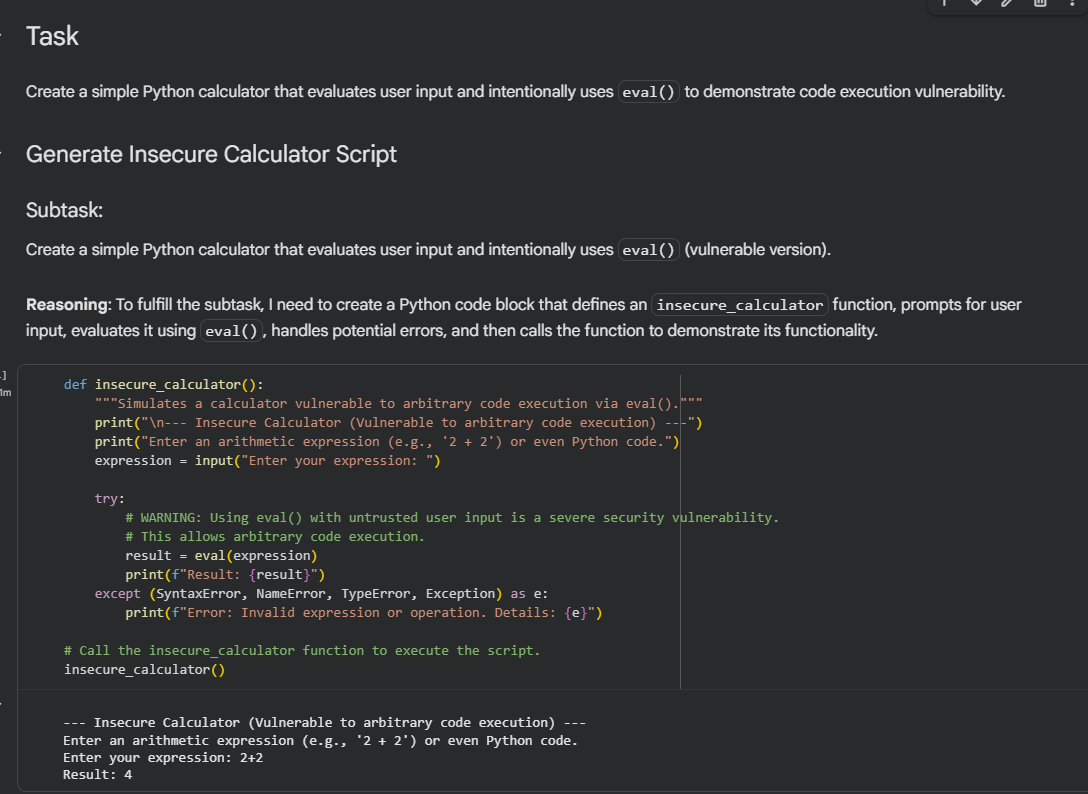
A screenshot of a computer

AI-generated content may be incorrect.

**Task 4: Identify Insecure Use of eval() or exec()**

PROMPT:

Generate a simple Python calculator that evaluates user input and intentionally uses eval() (vulnerable version).  
Show a test with malicious input like \_\_import\_\_('os').system('rm -rf /') to demonstrate the risk.  
Then provide a secure version that replaces eval()/exec() with ast.literal\_eval() (or a safe parser) and explain in one line why it’s safer.  
  
CODE AND OUTPUT:



A screenshot of a computer program

AI-generated content may be incorrect.

A black screen with white text

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer code

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.

A computer screen shot of a program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screen shot of a computer program

AI-generated content may be incorrect.A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

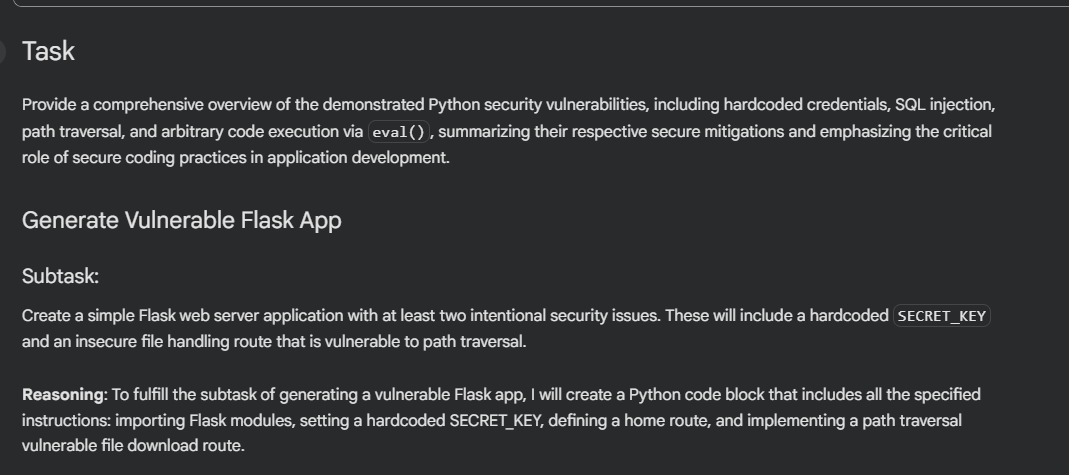
AI-generated content may be incorrect.

**Task 5: Static Security Testing with Linting Tools**

PROMPT:

Generate a short Flask web server with at least two security issues (e.g., hardcoded SECRET\_KEY, insecure file handling, use of eval/subprocess, or weak crypto).  
Output the vulnerable code, run **bandit** and paste the scan results showing the flagged issues, then provide a fixed version of the code and re-run **bandit** showing no issues.

CODE and OUTPUT:



A screenshot of a computer program

AI-generated content may be incorrect.

A black background with pink text

AI-generated content may be incorrect.