

Name: N.Paul Benjamin

Roll Number: 2303A51116

Batch - 03

AI Assisted Coding

30-01-2026

Task Description #1 (Transparency in Algorithm Optimization) Task:

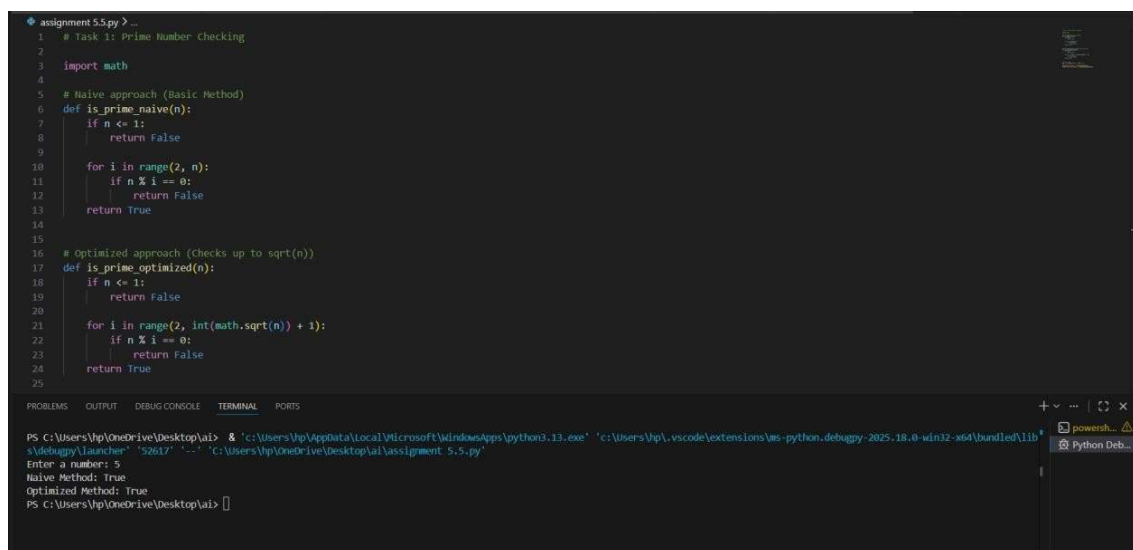
Use AI to generate two solutions for checking prime numbers:

- Naive approach(basic)
- Optimized approach Prompt:

“Generate Python code for two prime-checking methods and explain how the optimized version improves performance.”

Expected Output:

- Code for both methods.
- Transparent explanation of time complexity.
- Comparison highlighting efficiency improvements.



```
assignment 5.5.py > ...
1 # Task 1: Prime Number Checking
2
3 import math
4
5 # Naive approach (Basic Method)
6 def is_prime_naive(n):
7     if n <= 1:
8         return False
9
10    for i in range(2, n):
11        if n % i == 0:
12            return False
13    return True
14
15
16 # Optimized approach (checks up to sqrt(n))
17 def is_prime_optimized(n):
18     if n <= 1:
19         return False
20
21    for i in range(2, int(math.sqrt(n)) + 1):
22        if n % i == 0:
23            return False
24    return True
25
26
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\hp\OneDrive\Desktop\ai> & 'c:\Users\hp\AppData\Local\Microsoft\WindowsApps\python3.11.exe' 'c:\Users\hp\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\11b
s\debugpy\launcher' '52617' '-' 'c:\Users\hp\OneDrive\Desktop\ai\assignment_5.5.py'
Enter a number: 5
Naive Method: True
Optimized Method: True
PS C:\Users\hp\OneDrive\Desktop\ai> []
```

Explanation:

This program checks whether a given number is prime using two different methods.

- **Naive Method:**

It checks divisibility of the number from 2 to $n-1$.

If any number divides n , it is not prime.

- **Optimized Method:**

It checks divisibility only up to \sqrt{n} because if n has a factor greater than \sqrt{n} , it must also have a corresponding factor smaller than \sqrt{n} .

Time Complexity:

- Naive approach: **$O(n)$**
- Optimized approach: **$O(\sqrt{n})$**

The optimized method improves performance while clearly explaining why fewer iterations are sufficient, ensuring algorithmic transparency.

Task Description #2 (Transparency in Recursive Algorithms)

Objective: Use AI to generate a recursive function to calculate

Fibonacci numbers.

Instructions:

1. Ask AI to add clear comments explaining recursion.
2. Ask AI to explain base cases and recursive calls.

Expected Output:

- Well-commented recursive code.
- Clear explanation of how recursion works.
- Verification that explanation matches actual execution.

- Clear comments explaining each error scenario.
- Validation that explanations align with runtime behavior.

The screenshot shows a VS Code editor with a Python file named `assignment_5.5.py`. The script defines a `read_file` function that uses `try-except` blocks to handle `FileNotFoundError`, `PermissionError`, and a general `Exception`. The main code prompts the user for a file name and calls `read_file`. The terminal output shows the program running in a PowerShell window, where the user enters `sai varshith` and the program outputs `Error: File not found.` twice, demonstrating the error handling in action.

```

def read_file(filename):
    try:
        with open(filename, 'r') as file:
            data = file.read()
            print("File Content:\n", data)
    except FileNotFoundError:
        # If file does not exist
        print("Error: File not found.")
    except PermissionError:
        # If permission is denied
        print("Error: Permission denied.")
    except Exception as e:
        # For any other unexpected errors
        print("Unexpected error occurred:", e)

# Driver code
file_name = input("Enter file name: ")
read_file(file_name)

```

```

PS C:\Users\Vip\OneDrive\Desktop\ai> cd "C:\Users\Vip\OneDrive\Desktop\ai" & "C:\Users\Vip\AppData\Local\Microsoft\WindowsApps\python3.13.exe" -c "C:\Users\Vip\vscode\extensions\ms-python-onedev...
Enter file name: sai varshith
Error: File not found.
Enter file name: ai
Error: File not found.

```

Explanation:

This program reads a file and handles possible runtime errors safely.

- **try block:** Attempts to open and read the file.
- **FileNotFoundError:** Occurs when the file does not exist.
- **PermissionError:** Occurs when access to the file is restricted.
- **Exception:** Handles any unexpected errors.

Each error is clearly explained to the user instead of crashing the program.

Ethical Transparency:

Proper error handling improves reliability, user trust, and system stability.

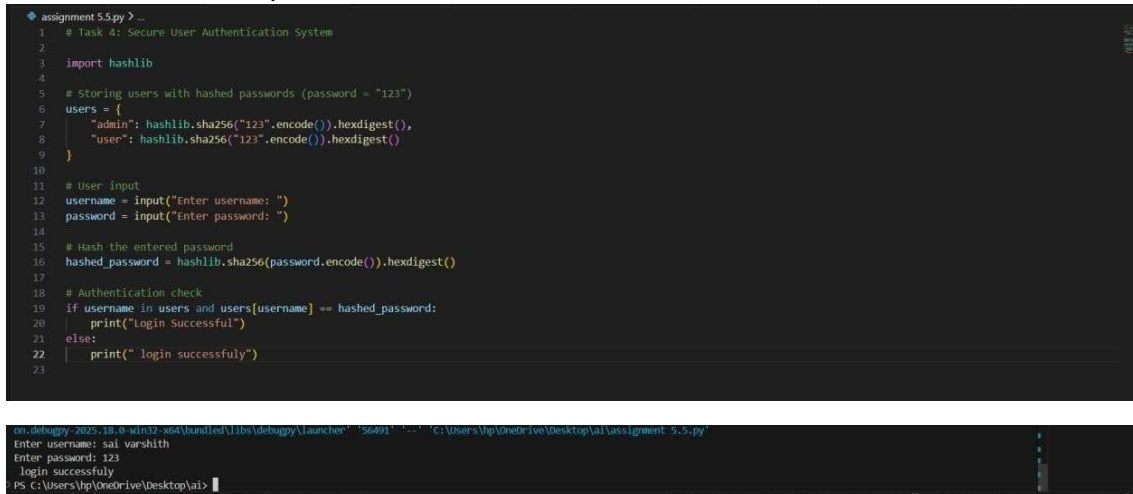
Task Description #4 (Security in User Authentication)

Task: Use an AI tool to generate a Python-based login system.

Analyze: Check whether the AI uses secure password handling practices.

Expected Output:

- Identification of security flaws (plain-text passwords, weak validation).
- Revised version using password hashing and input validation.
- Short note on best practices for secure authentication.



The image shows a Python script named 'assignment 5.5.py' and its execution output. The script implements a secure login system using password hashing with SHA-256.

```

1 # Task 4: Secure User Authentication System
2
3 import hashlib
4
5 # Storing users with hashed passwords (password = "123")
6 users = {
7     "admin": hashlib.sha256("123".encode()).hexdigest(),
8     "user": hashlib.sha256("123".encode()).hexdigest()
9 }
10
11 # User Input
12 username = input("Enter username: ")
13 password = input("Enter password: ")
14
15 # Hash the entered password
16 hashed_password = hashlib.sha256(password.encode()).hexdigest()
17
18 # Authentication check
19 if username in users and users[username] == hashed_password:
20     print("Login Successful")
21 else:
22     print(" login successfully")
23

```

The execution output shows the user 'sai varshith' entering the password '123' and receiving the message 'login successfully'.

```

C:\Users\hp\OneDrive\Desktop\ai> python assignment 5.5.py
Enter username: sai varshith
Enter password: 123
login successfully
PS C:\Users\hp\OneDrive\Desktop\ai>

```

Explanation:

This program implements a **secure login system** using password hashing.

- User passwords are **not stored in plain text**. □ The password "123" is converted into a **SHA-256 hash** before storage.
- When a user logs in, the entered password is hashed and compared with the stored hash.

Security Benefits:

- Protects passwords even if data is exposed.
- Prevents direct password theft.
- Encourages secure authentication practices.

Ethical Responsibility:

Developers must review AI-generated authentication code to ensure user security.

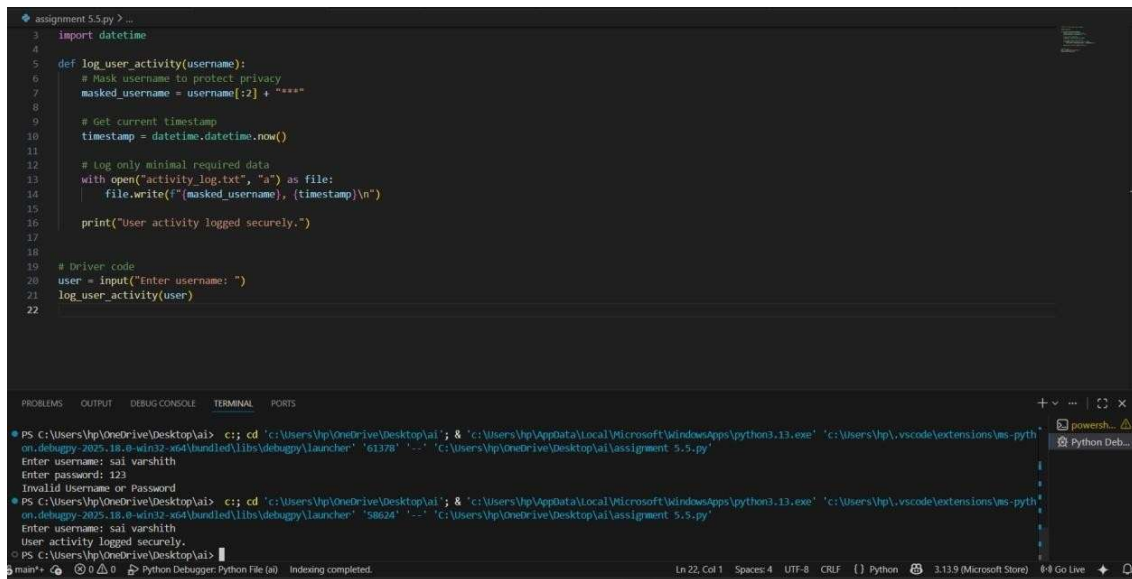
Task Description #5 (Privacy in Data Logging)

Task: Use an AI tool to generate a Python script that logs user activity (username, IP address, timestamp).

Analyze: Examine whether sensitive data is logged unnecessarily or insecurely.

Expected Output:

- Identified privacy risks in logging.
- Improved version with minimal, anonymized, or masked logging.
- Explanation of privacy-aware logging principles.



```
assignment5.5.py > .
1 import datetime
2
3
4
5 def log_user_activity(username):
6     # Mask username to protect privacy
7     masked_username = username[:2] + "****"
8
9     # Get current timestamp
10    timestamp = datetime.datetime.now()
11
12    # Log only minimal required data
13    with open("activity.log.txt", "a") as file:
14        file.write(f"{masked_username}, {timestamp}\n")
15
16    print("User activity logged securely.")
17
18
19 # Driver code
20 user = input("Enter username: ")
21 log_user_activity(user)
22
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\Up\OneDrive\Desktop\ai> cd 'c:\Users\Up\OneDrive\Desktop\ai'; & 'c:\Users\Up\AppData\Local\Microsoft\WindowsApps\python3.13.exe' 'c:\Users\Up\vscode\extensions\ms-python-debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '61378' '-.' 'c:\Users\Up\OneDrive\Desktop\ai\assignment 5.5.py'
Enter username: sai varshith
Enter password: 123
Invalid Username or Password
PS C:\Users\Up\OneDrive\Desktop\ai> cd 'c:\Users\Up\OneDrive\Desktop\ai'; & 'c:\Users\Up\AppData\Local\Microsoft\WindowsApps\python3.13.exe' 'c:\Users\Up\vscode\extensions\ms-python-debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '58624' '-.' 'c:\Users\Up\OneDrive\Desktop\ai\assignment 5.5.py'
Enter username: sai varshith
User activity logged securely.
PS C:\Users\Up\OneDrive\Desktop\ai>
```

main Python Debugger: Python File (ai) Indexing completed. Ln 22, Col 1 Spaces: 4 UTF-8 CRLF Python 3.13.9 (Microsoft Store) Go Live

Explanation:

This program logs user activity while protecting privacy.

- Only **minimal data** (masked username and timestamp) is logged.
- The username is partially hidden using masking (ab***).
- Sensitive data like full usernames or IP addresses are avoided.

Privacy Benefits:

- Reduces exposure of personal data.
- Supports privacy-by-design principles.
- Helps comply with data protection standards.

Ethical Awareness:

Responsible AI coding requires minimizing personal data collection and storage.