AI-ASSISTANT CODING

ASSIGNMENT-5.5

B.HARSHAVARDHAN

2303A52T02

Lab 5: Ethical Foundations – Responsible AI Coding Practices

Lab Objectives:

• To explore the ethical risks associated with AI-generated

• To recognize issues related to security, bias, transparency, and copyright.

• To reflect on the responsibilities of developers when using AI tools in software development.

• To promote awareness of best practices for responsible and ethical AI coding.

Lab Outcomes (LOs):

After completing this lab, students will be able to:

• Identify and avoid insecure coding patterns generated by AI tools.

• Detect and analyze potential bias or discriminatory logic in AI-generated outputs.

• Evaluate originality and licensing concerns in reused AI-generated code.

• Understand the importance of explainability and transparency in AI-assisted programming.

• Reflect on accountability and the human role in ethical AI coding practices.

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:

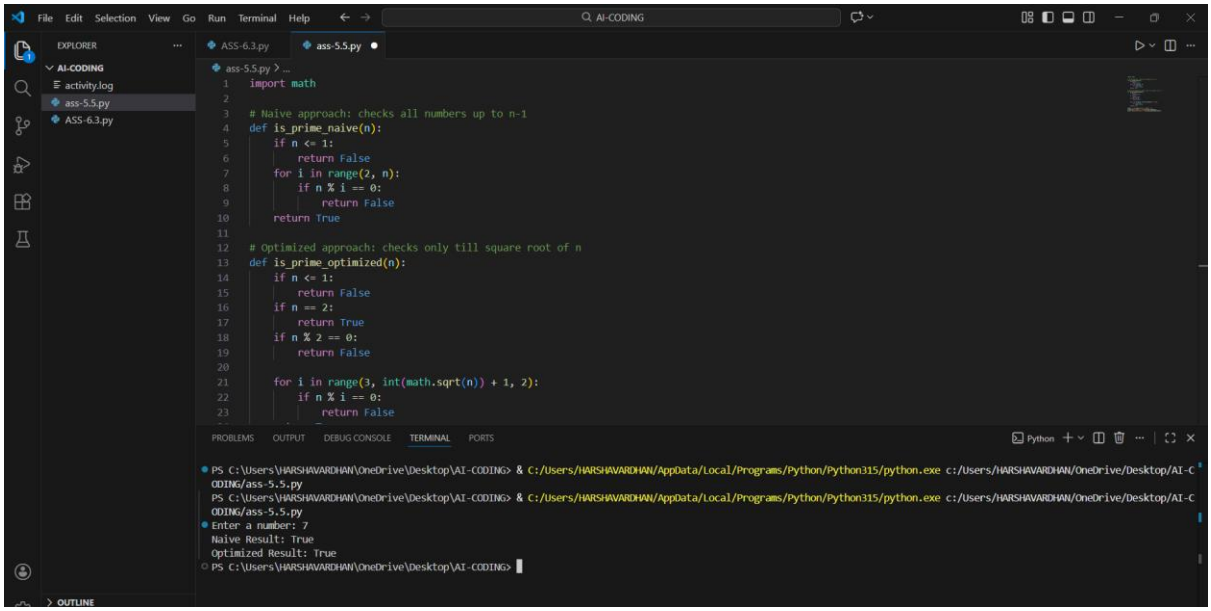
• Code for both methods.

• Transparent explanation of time complexity.

• Comparison highlighting efficiency improvements.

✅ **Prompt Used**

“Generate Python code for two methods to check whether a number is prime: one using a naive approach and another using an optimized approach. Explain how the optimized version improves performance and include time complexity analysis.”



✅ **Justification / Ethical Analysis**

- The naive method checks all numbers up to n, which is inefficient.
- The optimized method checks divisibility only up to √n, reducing unnecessary operations.
- Transparent explanation of time complexity (**O(n)** vs **O(√n)**) helps developers understand performance trade-offs.
- This task highlights **explainability and optimization transparency**, which are important when using AI-generated algorithms

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:

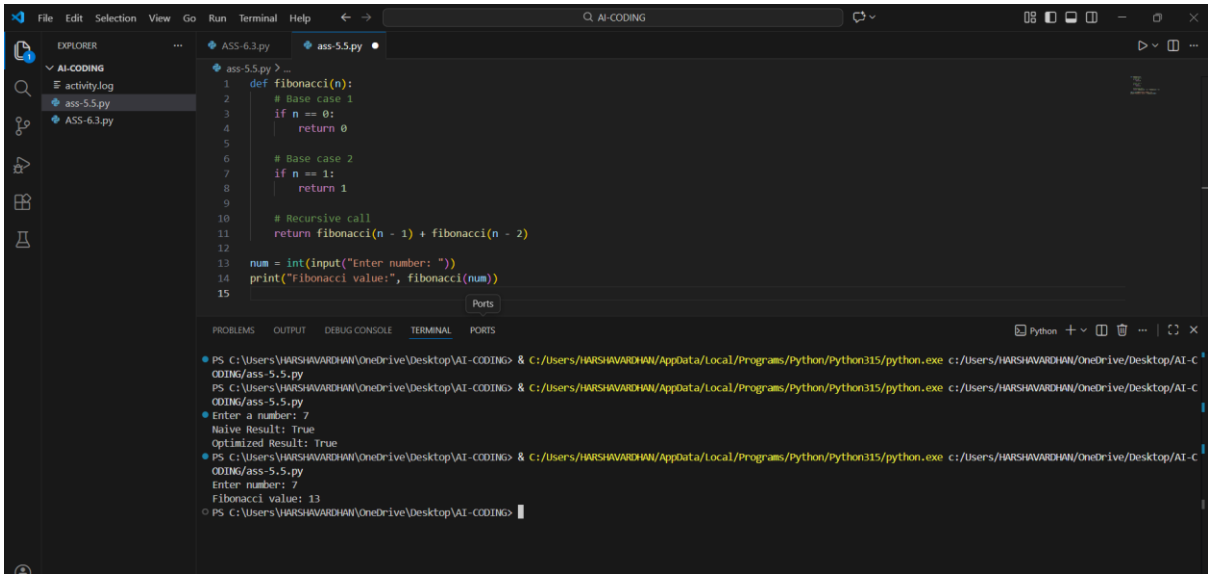
• Well-commented recursive code.

• Clear explanation of how recursion works.

• Verification that explanation matches actual execution.

✅ Prompt Used

“Generate a recursive Python function to calculate Fibonacci numbers. Add clear comments explaining recursion, base cases, and recursive calls. Also explain how recursion works step by step.”



✅ Justification / Ethical Analysis

- Clear comments help understand how recursion breaks problems into smaller sub-problems.
- Base cases prevent infinite recursion and ensure correctness.
- Verifying that the explanation matches actual execution ensures **truthfulness and transparency**.
- This task emphasizes that AI-generated code must be **understandable and explainable**, not just correct

Task Description #3 (Transparency in Error Handling)

Task: Use AI to generate a Python program that reads a file and processes data.

Prompt:

“Generate code with proper error handling and clear explanations for each exception.”

Expected Output:

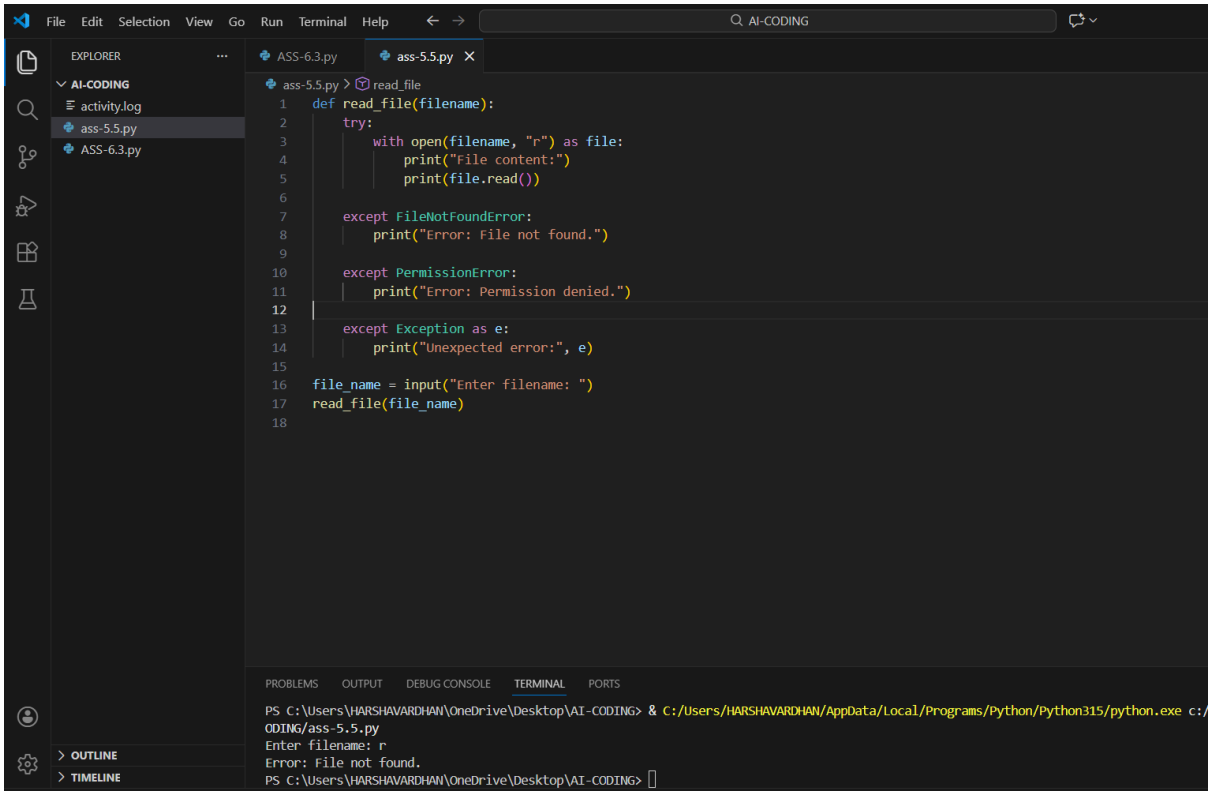
• Code with meaningful exception handling.

• Clear comments explaining each error scenario.

• Validation that explanations align with runtime behavior.

✅ **Prompt Used**

“Generate a Python program that reads and processes a file using proper error handling. Include clear explanations and comments for each exception such as FileNotFoundError and PermissionError.”



✅ Justification / Ethical Analysis

- Meaningful exception handling prevents application crashes and unsafe behavior.
- Clear error messages improve user trust and debugging clarity.
- Validating that explanations match runtime behavior ensures **responsible AI usage**.
- This task shows that AI-generated code should be **robust, safe, and transparent** in failure scenarios.

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:

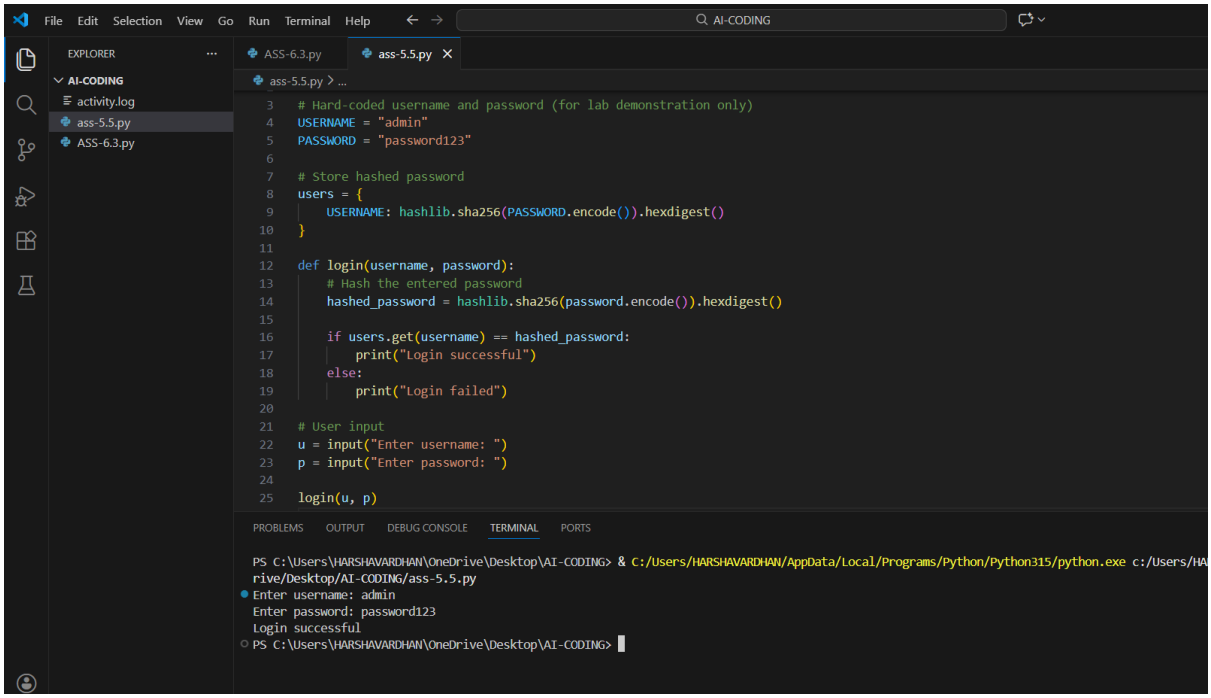
• Identification of security flaws (plain-text passwords, weak validation).

• Revised version using password hashing and input validation.

• Short note on best practices for secure authentication.

✅ Prompt Used

“Generate a Python-based login system. Review whether it follows secure password handling practices. Identify security flaws and revise the code using password hashing and proper input validation.”



✅ **Justification / Ethical Analysis**

- Initial AI-generated code may store passwords in plain text, which is insecure.
- Password hashing (SHA-256) protects credentials even if data is leaked.
- This task demonstrates that developers must **review and improve AI output**.
- Highlights ethical responsibility to protect user data and prevent security breaches

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.

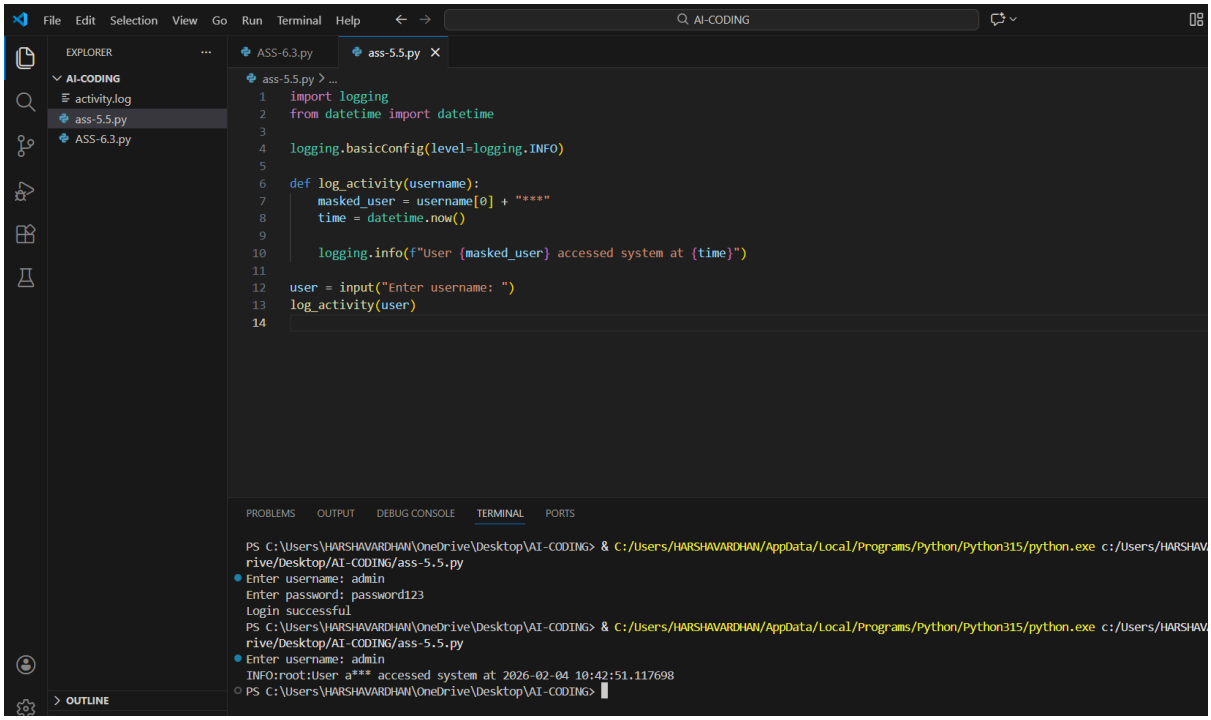
• Improved version with minimal, anonymized, or masked

logging.

• Explanation of privacy-aware logging principles.

✅ **Prompt Used**

"Generate a Python script that logs user activity including username, IP address, and timestamp. Analyze privacy risks and improve the code using anonymized or masked logging."



✅ Justification / Ethical Analysis

- Logging full usernames and IP addresses can violate user privacy.
- Masking or minimizing logged data reduces risk of misuse.
- Privacy-aware logging follows ethical and legal standards such as data minimization.
- This task reinforces that AI-generated logging must respect **user privacy and confidentiality**