

AI-ASSITED CODING

ASSIGNMENT 5.3

H.T NO - 2303A51511

NAME – K.PRADEEP REDDY

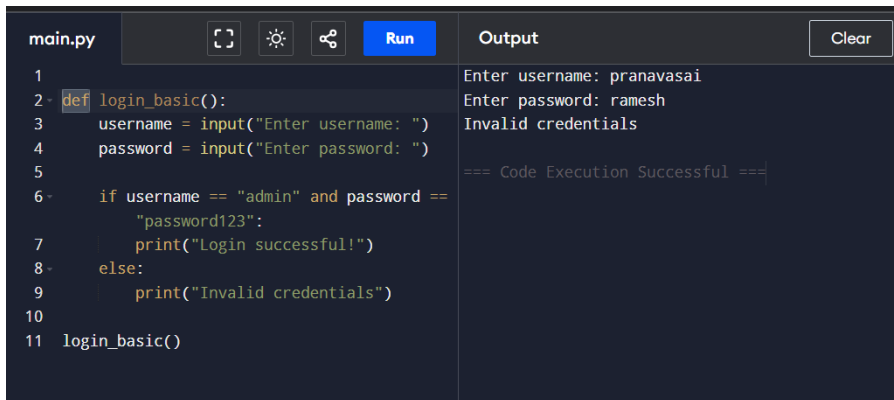
BATCH-27

Ethical and Secure Use of AI-Generated Code

Task 1: Privacy and Data Security in AI-Generated Code

PROMPT: GENERATE AN LOGIN CODE IN PYTHON

CODE AND OUTPUT:



The screenshot shows a code editor with a file named 'main.py'. The code defines a function 'login_basic()' that prompts for a username and password. It checks if the username is 'admin' and the password is 'password123'. If correct, it prints 'Login successful!'; otherwise, it prints 'Invalid credentials'. The function is then called. The output pane shows the execution results: 'Enter username: pranavasai', 'Enter password: ramesh', 'Invalid credentials', and '=== Code Execution Successful ==='.

```
main.py  [Icons] Run Output Clear
1
2 def login_basic():
3     username = input("Enter username: ")
4     password = input("Enter password: ")
5
6     if username == "admin" and password ==
       "password123":
7         print("Login successful!")
8     else:
9         print("Invalid credentials")
10
11 login_basic()
```

Enter username: pranavasai
Enter password: ramesh
Invalid credentials
=== Code Execution Successful ===

Security Issues Identified:

- Credentials are hardcoded
- Password is stored and compared in plain text
- No input validation or hashing

PROMPT: GENERATE A SECURE CODE

CODE AND OUTPUT:

<pre>1 import getpass, hashlib 2 3 stored_hash = hashlib.sha256('securePass' .encode()).hexdigest() 4 user_pass = getpass.getpass('Password: ') 5 6 if hashlib.sha256(user_pass.encode ()).hexdigest() == stored_hash: 7 print('Login successful') 8 else: 9 print('Login failed') 10 11</pre>	<pre>Password: Login failed === Code Execution Successful ===</pre>
--	--

Improvements:

- Removed hardcoded credentials
- Used password hashing
- Used secure password input

Task 2: Bias Detection in AI-Generated Decision Systems

PROMPT:

Generate A Loan Approval Code (Biased Example):

CODE AND OUTPUT:

<pre>main.py 1 applicant_gender = "male" 2 income = 60000 3 4 if applicant_gender == "male" and income > 50000: 5 approve = True 6 else: 7 approve = False 8 9 print("Loan Approved:", approve) 10</pre>	<pre>Loan Approved: True === Code Execution Successful ===</pre>
---	---

Bias Identified:

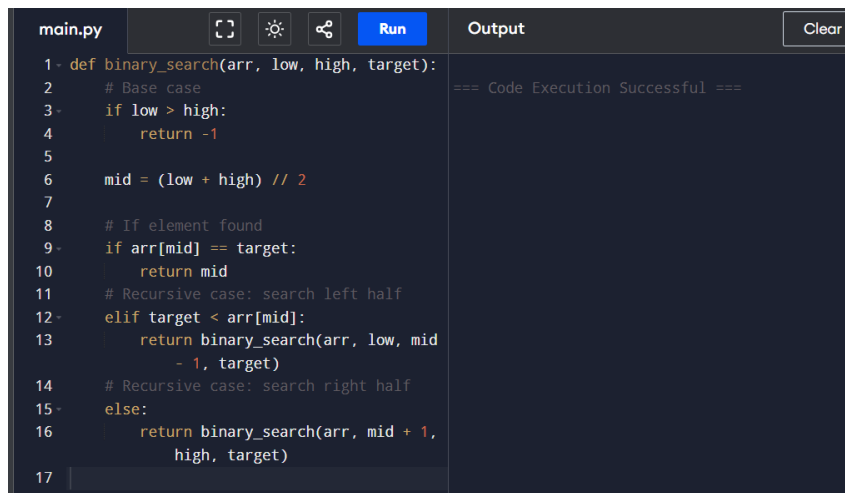
- Approval depends on gender
- Gender is irrelevant to loan eligibility

Fairness Discussion:

Such logic unfairly disadvantages applicants based on gender.

Mitigation Strategies:

- Remove gender from decision logic
- Use only financial criteria
- Regular bias audits



```
main.py  [ ] [ ] [ ] [ ] Run Output Clear
1- def binary_search(arr, low, high, target):
2-     # Base case
3-     if low > high:
4-         return -1
5-
6-     mid = (low + high) // 2
7-
8-     # If element found
9-     if arr[mid] == target:
10-         return mid
11-     # Recursive case: search left half
12-     elif target < arr[mid]:
13-         return binary_search(arr, low, mid
14-                               - 1, target)
15-     # Recursive case: search right half
16-     else:
17-         return binary_search(arr, mid + 1,
18-                               high, target)
```

=== Code Execution Successful ===

Task 3:
Transparency and
Explainability
(Recursive Binary
Search)

PROMPT: Generate a Recursive Binary Search Code:

Code and Output:

Explanation:

- Base case stops recursion when search range is invalid
- Recursive case reduces problem size each call

Code is beginner-friendly, well-commented, and logically clear.

Task 4: Ethical Evaluation of AI-Based Scoring Systems

PROMPT:Generate a Scoring Code:

CODE AND OUTPUT:

Bias Analysis:

- No gender or name used
- Logic is objective and skill-based

Ethical Evaluation:

Scoring is fair as it relies on job-relevant attributes.

main.py	Output
<pre>1 2 skills = float(input("Enter skills score: ")) 3 experience = float(input("Enter experience score: ")) 4 education = float(input("Enter education score: ")) 5 6 score = skills * 0.5 + experience * 0.3 + education * 0.2 7 8 9 print("Final Score:", score) 10</pre>	<pre>Enter skills score: 15 Enter experience score: 34 Enter education score: 45 Final Score: 26.7 === Code Execution Successful ===</pre>

Task 5: Inclusiveness and Ethical Variable Design

PROMPT: generate a code for bonus based on if gender is male

CODE:

```
python

if gender == 'male':
    bonus = 1000
else:
    bonus = 500
```

Issues:

- Gender-based assumptions
- Non-inclusive logic

Revised Inclusive Code:

PROMPT: GENERATE A CODE BASED ON PERFORMANCE

CODE AND OUPUT:

<pre>1 # Input performance rating 2 performance_rating = float(input("Enter performance rating (1 to 5): ")) 3 4 # Bonus calculation based on performance 5 if performance_rating >= 4: 6 bonus = 1000 7 else: 8 bonus = 500 9 10 # Output 11 print("Bonus Amount:", bonus) 12 13</pre>	<pre>Enter performance rating (1 to 5): 4 Bonus Amount: 1000 === Code Execution Successful ===</pre>
--	---

Improvements:

- Removed gender dependency
- Used performance-based logic