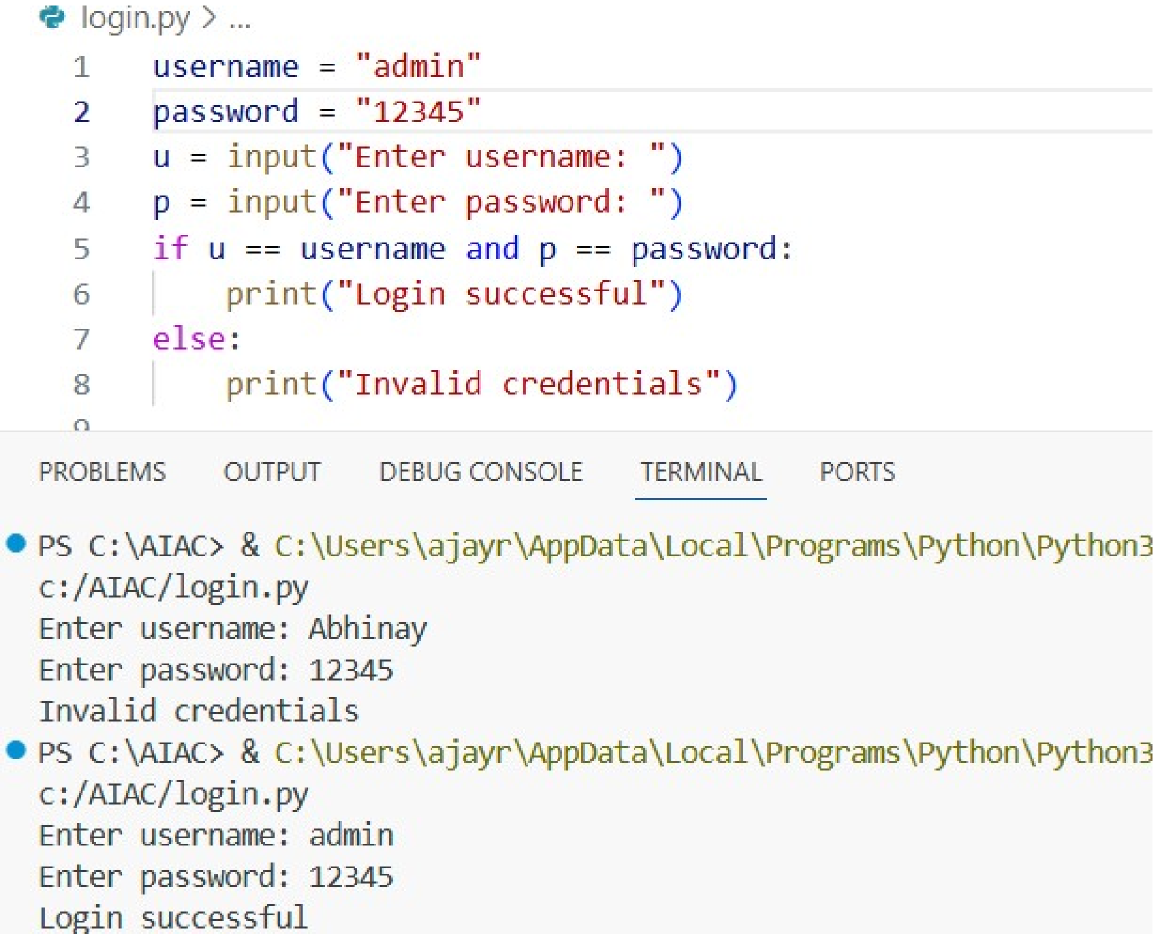
AI - Assisted Coding-Assignment -5

HTNO:2303A52228
Batch:37

Lab 5: Ethical Foundations – Responsible AI Coding Practices

Task Description-1
Privacy and Data Security in AI-Generated Code
Expected Output:
AI-generated login code
Identification of security risks
Revised secure version of the code
Brief explanation of improvements
Prompt:
Generate a simple login system in Python using username and password.
CODE AND OUTPUT:



```python
username = "admin"
password = "12345"
u = input("Enter username: ")
p = input("Enter password: ")
if u == username and p == password:
    print("Login successful")
else:
    print("Invalid credentials")
```

```
PS C:\AIAC> & C:\Users\ajayr\AppData\Local\Programs\Python\Python3
c:/AIAC/login.py
Enter username: Abhinay
Enter password: 12345
Invalid credentials
PS C:\AIAC> & C:\Users\ajayr\AppData\Local\Programs\Python\Python3
c:/AIAC/login.py
Enter username: admin
Enter password: 12345
Login successful
```

Explanation:
The original AI-generated login system was insecure because it used hardcoded credentials and compared passwords in plain text, which makes it easy for attackers to

view or steal sensitive information directly from the source code. To improve security, the revised version replaces plain-text passwords with SHA-256 hashing, so the actual password is never stored or compared directly. Instead, only the hashed value is checked, which protects user data even if the code is exposed. Basic input validation using strip() was also added to avoid accidental spaces during login.

Task Description-2

Bias Detection in AI-Generated Decision Systems

Expected Output:

Python code generated by AI

Identification of biased logic (if any)

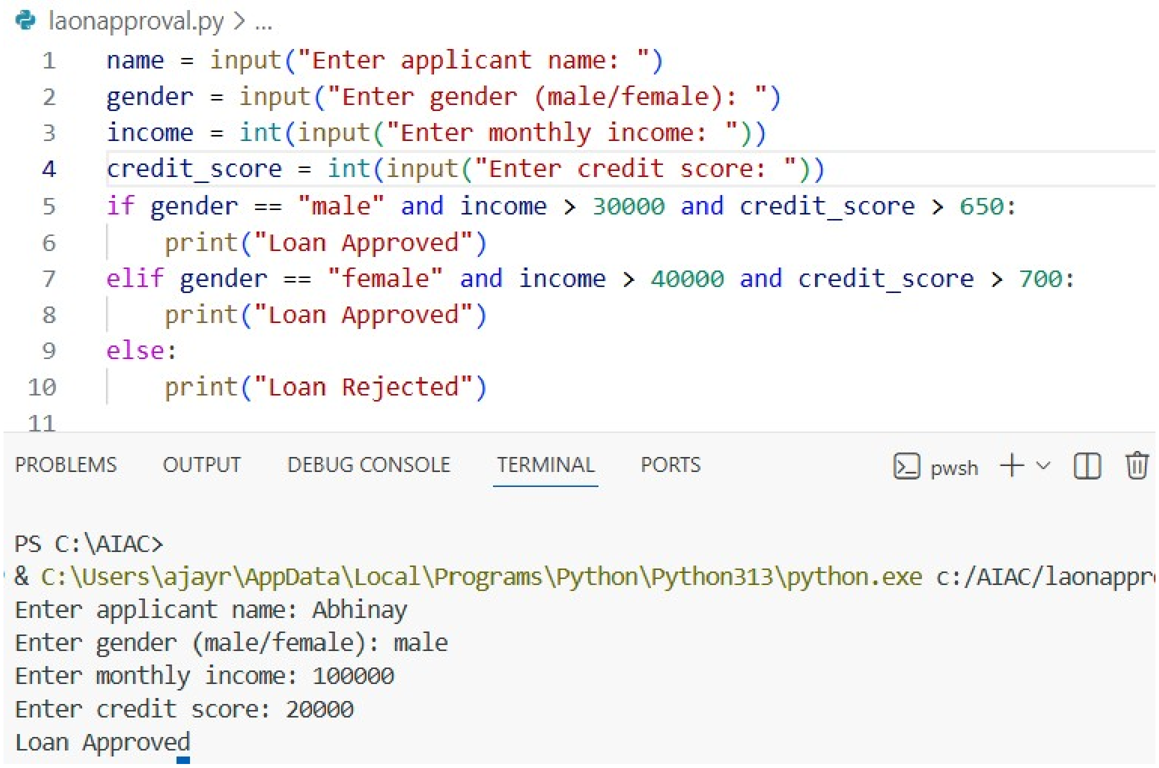
Discussion on fairness issues

Mitigation strategies

Prompt:

Create a loan approval system for applicants named Rahul (male) and Priya (female).

CODE AND OUTPUT:



```python
name = input("Enter applicant name: ")
gender = input("Enter gender (male/female): ")
income = int(input("Enter monthly income: "))
credit_score = int(input("Enter credit score: "))
if gender == "male" and income > 30000 and credit_score > 650:
    print("Loan Approved")
elif gender == "female" and income > 40000 and credit_score > 700:
    print("Loan Approved")
else:
    print("Loan Rejected")
```

```
PS C:\AIAC>
& C:\Users\ajayr\AppData\Local\Programs\Python\Python313\python.exe c:/AIAC/laonappr
Enter applicant name: Abhinay
Enter gender (male/female): male
Enter monthly income: 100000
Enter credit score: 20000
Loan Approved
```

Explanation:

The system unfairly favors male applicants by setting lower eligibility thresholds, while female applicants must meet stricter conditions for the same loan. This creates gender-based discrimination and violates fairness principles. Personal attributes like gender or name should never affect financial decisions. Such biased logic can lead to unequal opportunities and reinforces societal stereotypes, making the AI system unreliable and unethical.

Task Description-3

Transparency and Explainability in AI-Generated Code Recursive Binary Search.

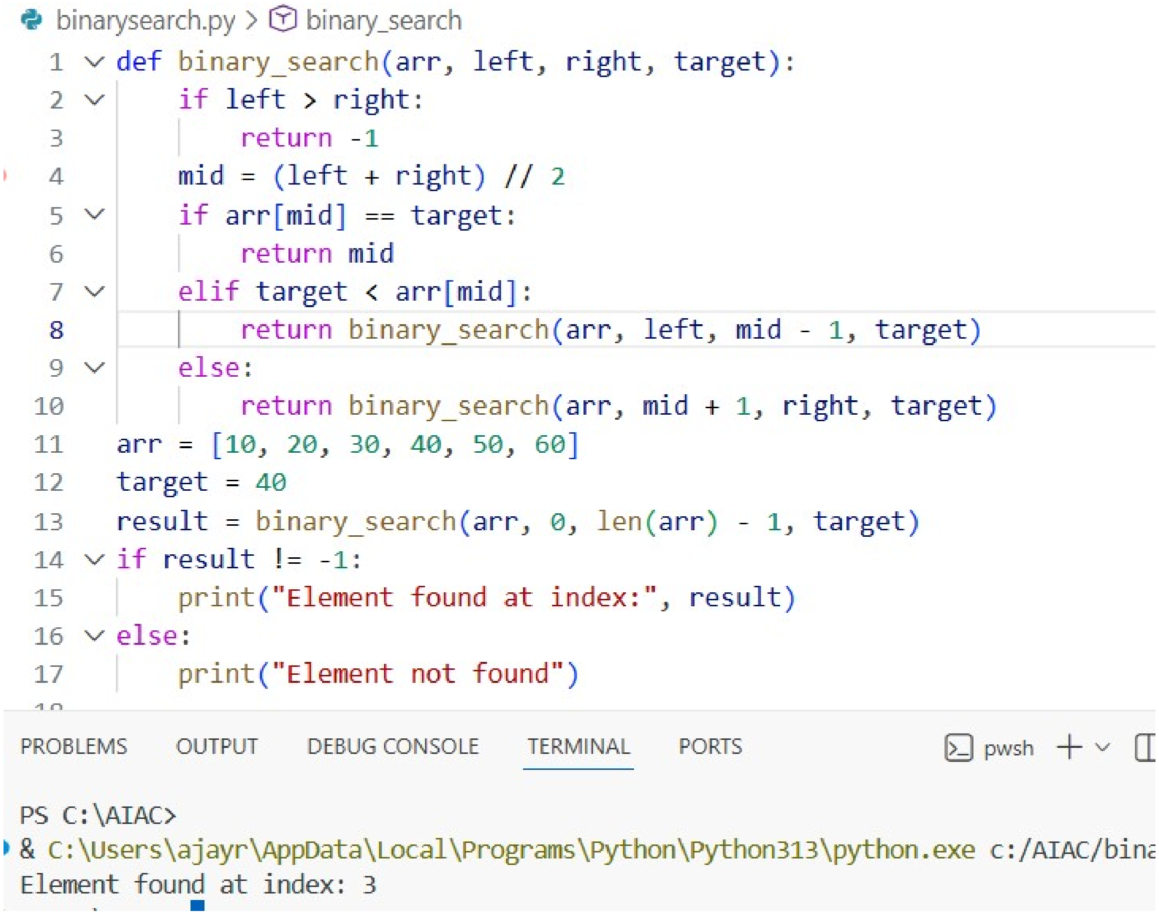
Expected Output:

Python program for recursive binary search
AI-generated comments and explanation
Student’s assessment on clarity, correctness, and transparency

Prompt:

Generate a Python program that implements Binary Search using recursion.
Include clear inline comments and a step-by-step explanation of how recursion works.

CODE AND OUTPUT:



```python
def binary_search(arr, left, right, target):
    if left > right:
        return -1
    mid = (left + right) // 2
    if arr[mid] == target:
        return mid
    elif target < arr[mid]:
        return binary_search(arr, left, mid - 1, target)
    else:
        return binary_search(arr, mid + 1, right, target)
arr = [10, 20, 30, 40, 50, 60]
target = 40
result = binary_search(arr, 0, len(arr) - 1, target)
if result != -1:
    print("Element found at index:", result)
else:
    print("Element not found")
```

```
PS C:\AIAC>
& C:\Users\ajayr\AppData\Local\Programs\Python\Python313\python.exe c:/AIAC/bina
Element found at index: 3
```

Explanation:

The recursive cases are well explained by showing how the function calls itself on either the left or right half depending on the comparison with the middle element.

Task Description-4

Ethical Evaluation of AI-Based Scoring Systems.

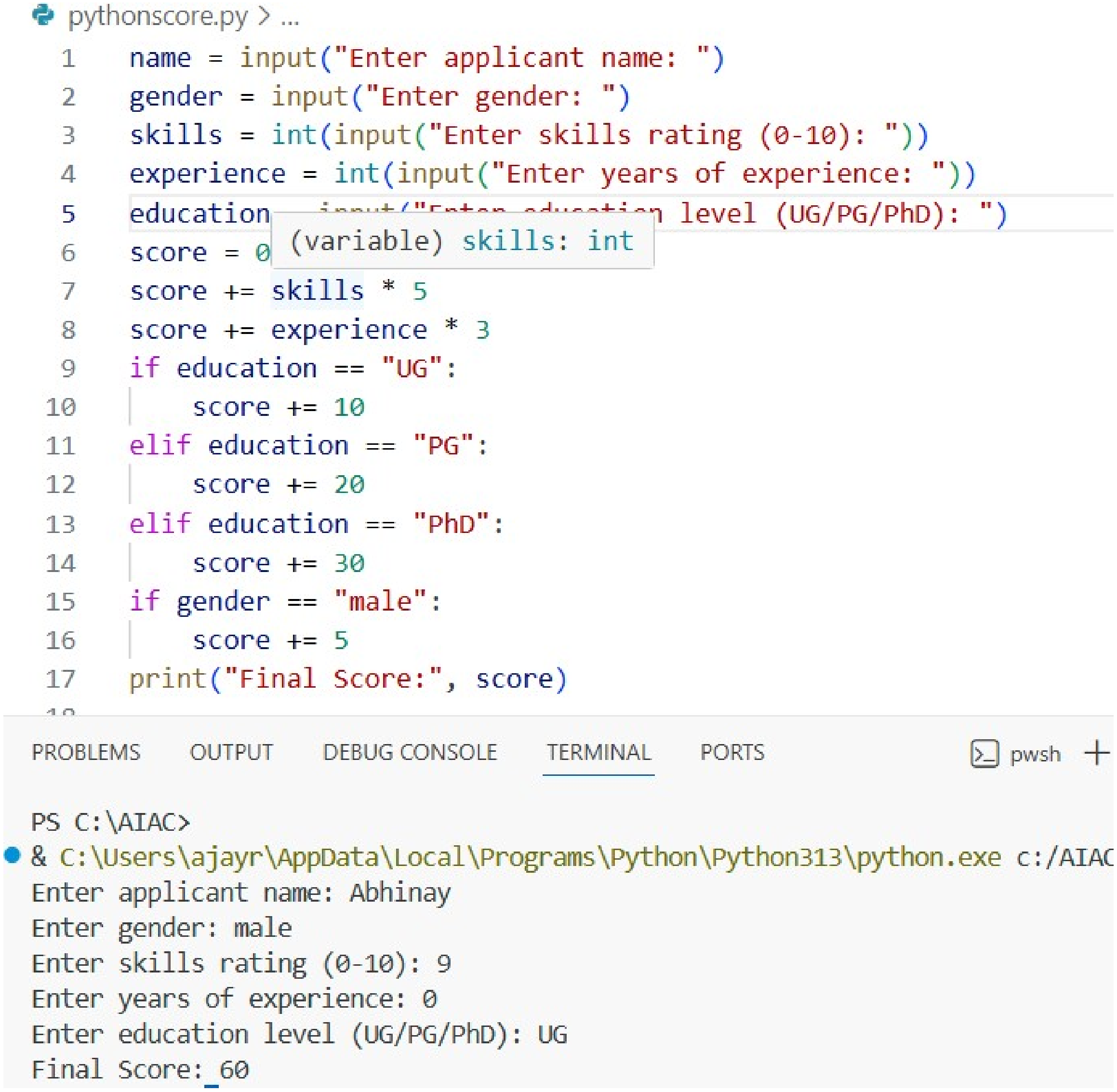
Expected Output:

Python scoring system code
Identification of potential bias (if any)
Ethical analysis of the scoring logic

Prompt:

Generate a Python program to score job applicants based on skills, experience, and education.

CODE AND OUTPUT:



```python
name = input("Enter applicant name: ")
gender = input("Enter gender: ")
skills = int(input("Enter skills rating (0-10): "))
experience = int(input("Enter years of experience: "))
education = input("Enter education level (UG/PG/PhD): ")
score = 0
score += skills * 5
score += experience * 3
if education == "UG":
    score += 10
elif education == "PG":
    score += 20
elif education == "PhD":
    score += 30
if gender == "male":
    score += 5
print("Final Score:", score)
```

```
PS C:\AIAC>
& C:\Users\ajayr\AppData\Local\Programs\Python\Python313\python.exe c:/AIAC
Enter applicant name: Abhinay
Enter gender: male
Enter skills rating (0-10): 9
Enter years of experience: 0
Enter education level (UG/PG/PhD): UG
Final Score: 60
```

Explanation:

The scoring system is ethically problematic because it includes gender as a factor, granting male applicants additional points regardless of their qualifications. This introduces discrimination and violates principles of fairness and equal opportunity. Hiring decisions should be based strictly on relevant criteria such as skills, experience, and education. Collecting personal attributes like name and gender is unnecessary and increases the risk of biased outcomes.

Task Description-5

Inclusiveness and Ethical Variable Design

Expected Output:

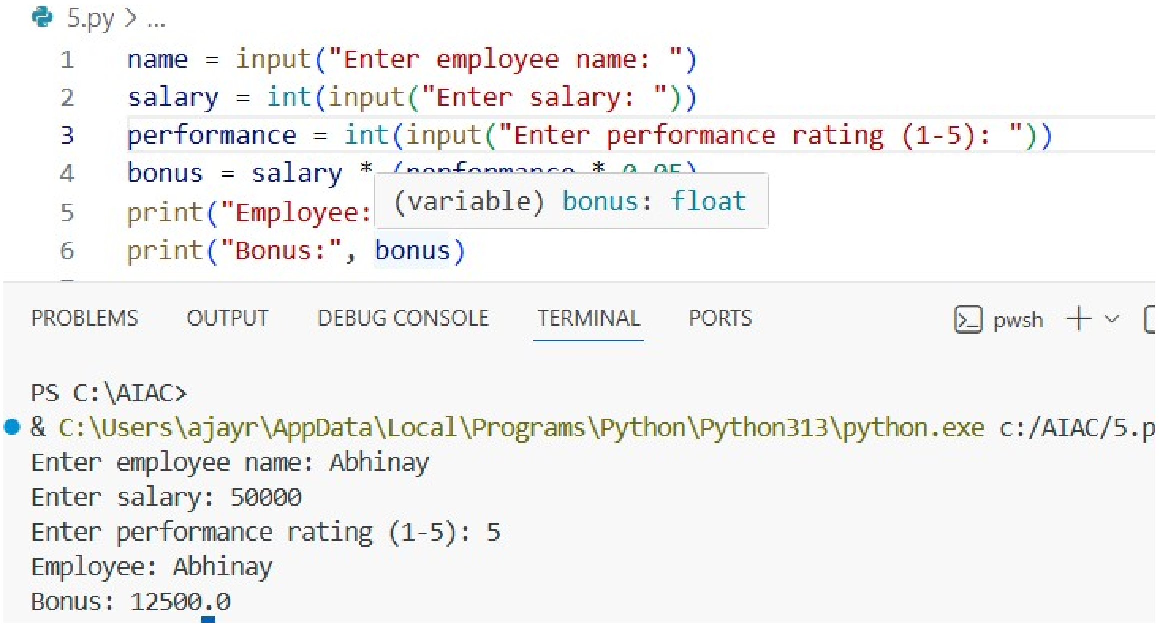
Original AI-generated code snippet

Revised inclusive and gender-neutral code

Prompt:

Generate a Python program that processes employee details including name, gender, and salary.

CODE AND OUTPUT:



```python
name = input("Enter employee name: ")
salary = int(input("Enter salary: "))
performance = int(input("Enter performance rating (1-5): "))
bonus = salary * (performance * 0.05)
print("Employee: (variable) bonus: float
print("Bonus:", bonus)
```

```
PS C:\AIAC>
& C:\Users\ajayr\AppData\Local\Programs\Python\Python313\python.exe c:/AIAC/5.p
Enter employee name: Abhinay
Enter salary: 50000
Enter performance rating (1-5): 5
Employee: Abhinay
Bonus: 12500.0
```

Explanation:

In the original code, gender-specific variables (male and female) were used to decide bonus amounts, which assumes differences in reward based purely on gender. This is non-inclusive and unfair because compensation should depend on professional factors rather than identity. The revised version removes gender entirely and replaces it with a performance-based rating, ensuring that bonuses are calculated using objective criteria.