ASSIGNMENT-5.3

Name: Praneeth banda

HT.No: 2303A51711

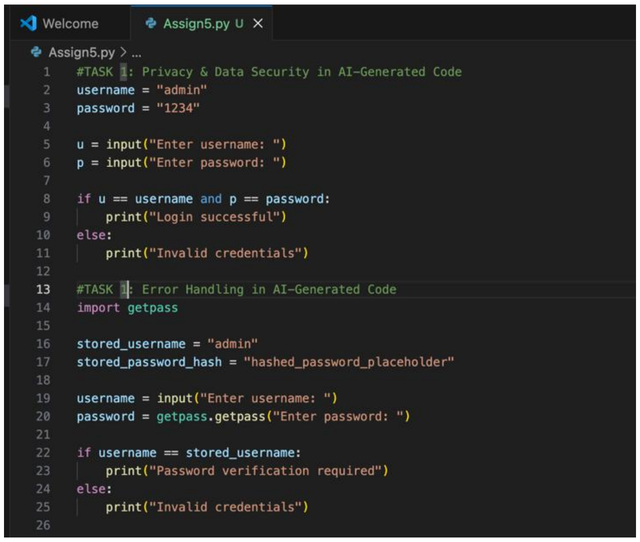
Batch: 20

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

To analyze AI-generated login code for privacy and security risks such as hardcoded credentials and plain-text password handling, and to revise it for better security practices.

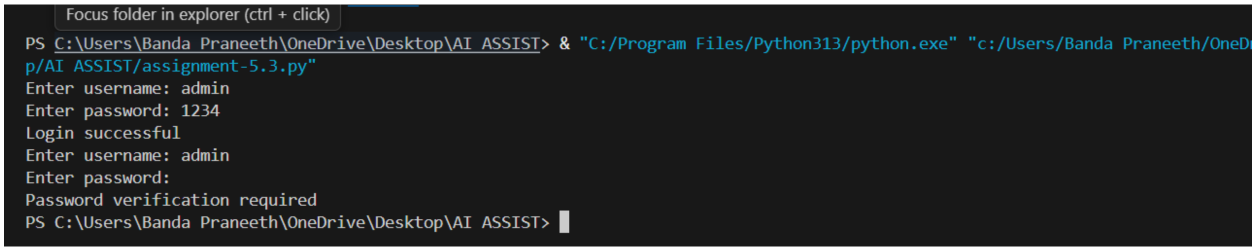
Prompt: # Create a simple login system in Python using username and password.

Code:



```python
#TASK 1: Privacy & Data Security in AI-Generated Code
username = "admin"
password = "1234"

u = input("Enter username: ")
p = input("Enter password: ")

if u == username and p == password:
    print("Login successful")
else:
    print("Invalid credentials")

#TASK 1: Error Handling in AI-Generated Code
import getpass

stored_username = "admin"
stored_password_hash = "hashed_password_placeholder"

username = input("Enter username: ")
password = getpass.getpass("Enter password: ")

if username == stored_username:
    print("Password verification required")
else:
    print("Invalid credentials")
```

Result:



```
PS C:\Users\Banda Praneeth\OneDrive\Desktop\AI ASSIST> & "C:/Program Files/Python313/python.exe" "c:/Users/Banda Praneeth/OneDr
p/AI ASSIST/assignment-5.3.py"
Enter username: admin
Enter password: 1234
Login successful
Enter username: admin
Enter password:
Password verification required
PS C:\Users\Banda Praneeth\OneDrive\Desktop\AI ASSIST>
```

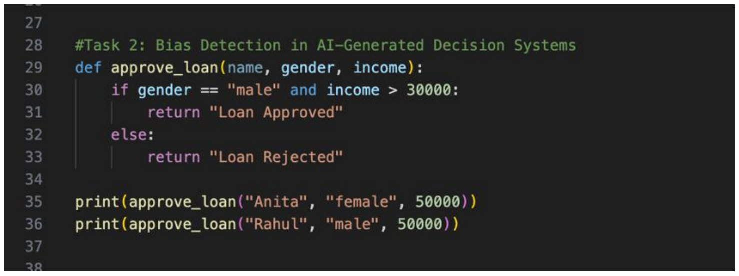
Observation:
The AI-generated login system contains significant security vulnerabilities, as the username and password are hardcoded directly into the source code. The password is stored and compared in plain text without any form of encryption or hashing, which exposes sensitive credentials to potential misuse. Additionally, the authentication logic lacks basic security measures such as input validation and secure password handling, making the code unsuitable for real-world applications.

Task 2: Bias Detection in AI-Generated Decision Systems.
This task focuses on identifying bias in AI-generated decision-making logic. The aim is to analyze whether irrelevant attributes such as gender influence outcomes unfairly.

Prompt: # Create a simple loan approval system

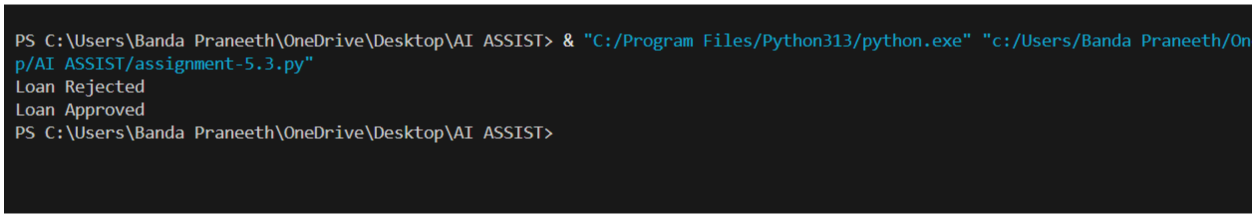
Python.

Code:



```python

#Task 2: Bias Detection in AI-Generated Decision Systems
def approve_loan(name, gender, income):
    if gender == "male" and income > 30000:
        return "Loan Approved"
    else:
        return "Loan Rejected"

print(approve_loan("Anita", "female", 50000))
print(approve_loan("Rahul", "male", 50000))

```

Result:



```
PS C:\Users\Banda Praneeth\OneDrive\Desktop\AI ASSIST> & "C:/Program Files/Python313/python.exe" "c:/Users/Banda Praneeth/On
p/AI ASSIST/assignment-5.3.py"
Loan Rejected
Loan Approved
PS C:\Users\Banda Praneeth\OneDrive\Desktop\AI ASSIST>
```

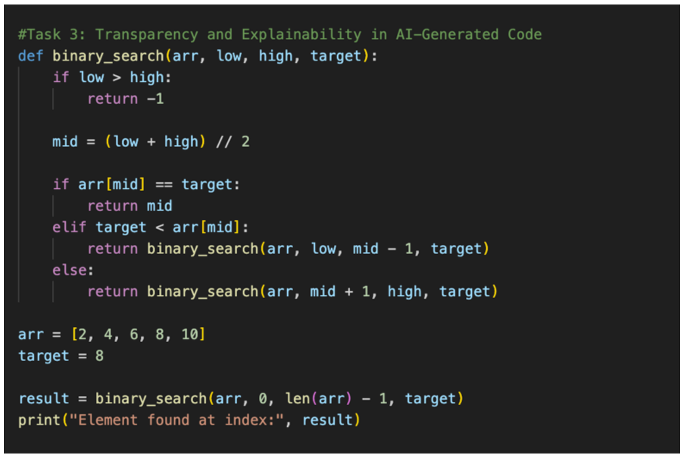
Observation:
The AI-generated loan approval logic demonstrates clear bias by using gender as a determining factor in the approval process. Since gender is unrelated to an applicant's financial credibility, its inclusion results in unfair treatment of applicants. Such logic can lead to discriminatory outcomes and violates ethical principles of fairness and equality in decision-making systems.

Task 3: Transparency and Explainability in AI-Generated Code

The purpose of this task is to evaluate the transparency and clarity of AI-generated code by analyzing a recursive binary search implementation and its explanation.

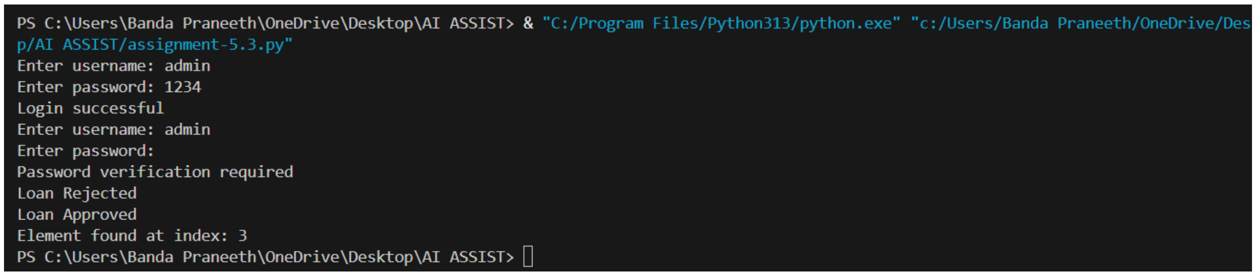
Prompt: # Write a recursive binary search program in Python with comments and explanation.

Code:



```python
#Task 3: Transparency and Explainability in AI-Generated Code
def binary_search(arr, low, high, target):
    if low > high:
        return -1

    mid = (low + high) // 2

    if arr[mid] == target:
        return mid
    elif target < arr[mid]:
        return binary_search(arr, low, mid - 1, target)
    else:
        return binary_search(arr, mid + 1, high, target)

arr = [2, 4, 6, 8, 10]
target = 8

result = binary_search(arr, 0, len(arr) - 1, target)
print("Element found at index:", result)
```

Result:



```
PS C:\Users\Banda Praneeth\OneDrive\Desktop\AI ASSIST> & "C:/Program Files/Python313/python.exe" "c:/Users/Banda Praneeth/OneDrive/Des
p/AI ASSIST/assignment-5.3.py"
Enter username: admin
Enter password: 1234
Login successful
Enter username: admin
Enter password:
Password verification required
Loan Rejected
Loan Approved
Element found at index: 3
PS C:\Users\Banda Praneeth\OneDrive\Desktop\AI ASSIST>
```

Observation:

The AI-generated recursive binary search program is transparent and easy to understand, as it clearly defines both the base case and the recursive case. The inline comments accurately describe each step of the algorithm, and the overall explanation aligns well with the implemented logic. As a result, the code is suitable for beginner-level learners and meets the requirements of explainability and clarity.

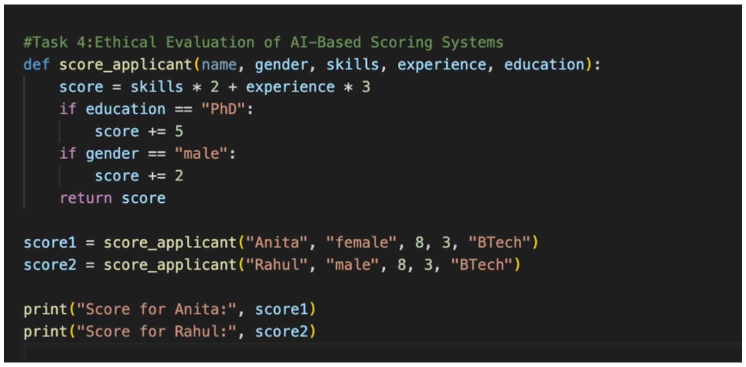
Task 4: Ethical Evaluation of AI-Based Scoring Systems

This task evaluates whether AI-generated scoring systems use ethical and objective criteria when assessing job applicants.

Prompt: #Create a job applicant scoring system based on skills, experience, and

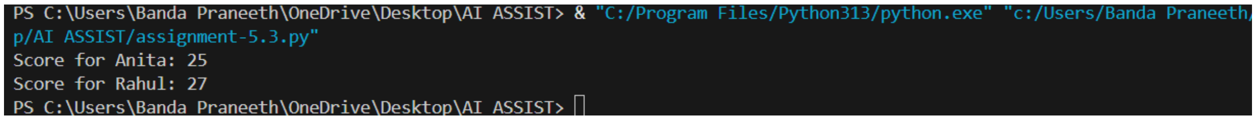
education.

Code:



```python
#Task 4:Ethical Evaluation of AI-Based Scoring Systems
def score_applicant(name, gender, skills, experience, education):
    score = skills * 2 + experience * 3
    if education == "PhD":
        score += 5
    if gender == "male":
        score += 2
    return score

score1 = score_applicant("Anita", "female", 8, 3, "BTech")
score2 = score_applicant("Rahul", "male", 8, 3, "BTech")

print("Score for Anita:", score1)
print("Score for Rahul:", score2)
```

Result:



```
PS C:\Users\Banda Praneeth\OneDrive\Desktop\AI ASSIST> & "C:/Program Files/Python313/python.exe" "c:/Users/Banda Praneeth/
p/AI ASSIST/assignment-5.3.py"
Score for Anita: 25
Score for Rahul: 27
PS C:\Users\Banda Praneeth\OneDrive\Desktop\AI ASSIST> 
```

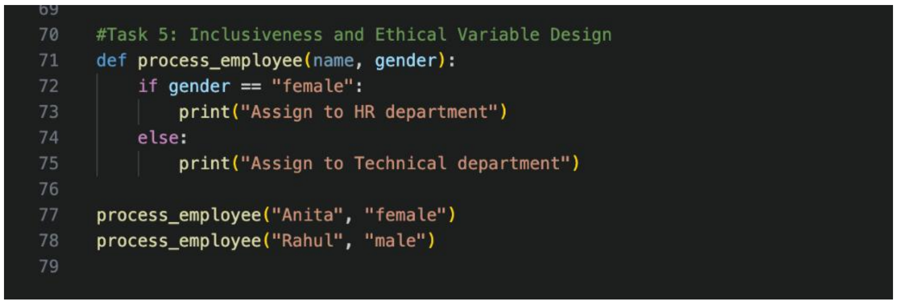
Observation

The applicant scoring system generated by the AI introduces ethical concerns by assigning additional score based on gender. Since gender and name are unrelated to professional capability or job performance, their inclusion results in biased and unfair scoring. Ethical scoring systems should rely solely on objective and job-relevant attributes such as skills, experience, and education

Task 5: Inclusiveness and Ethical Variable Design.
The goal of this task is to identify non-inclusive variable usage and assumptions related to gender or identity in AI-generated code.

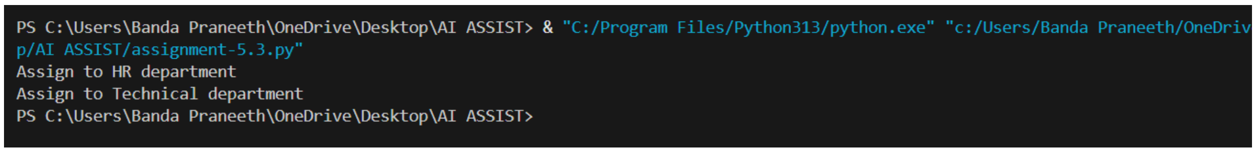
Prompt: #Generate Python code to process employee details.

Code:



```
#Task 5: Inclusiveness and Ethical Variable Design
def process_employee(name, gender):
    if gender == "female":
        print("Assign to HR department")
    else:
        print("Assign to Technical department")

process_employee("Anita", "female")
process_employee("Rahul", "male")
```

Result:



```
PS C:\Users\Banda Praneeth\OneDrive\Desktop\AI ASSIST> & "C:/Program Files/Python313/python.exe" "c:/Users/Banda Praneeth/OneDriv
p/AI ASSIST/assignment-5.3.py"
Assign to HR department
Assign to Technical department
PS C:\Users\Banda Praneeth\OneDrive\Desktop\AI ASSIST>
```

Observation:

The AI-generated code makes assumptions about employee roles based on gender, which reinforces stereotypes and excludes non-binary identities. Assigning departments purely on gender ignores individual skills, interests, and qualifications. Such logic is not inclusive and highlights the need for ethical variable design that avoids identity-based assumptions.