

# AI Assisted Coding

## Assignment 5.3

Name: P. Vineeth Kumar

Hall ticket no: 2303A51256

Batch no: 19

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

#### Simple Prompt:

Generate a simple login system in Python.

#### Code & Output:

The screenshot shows a code editor window titled "Assignment 5.3.py" with the following Python code:

```
File -> Edit -> Selection -> View -> Go -> Run -> Terminal -> Help -> <- >
Assignment_5.3.py > login_system
1 "Task 1: Privacy and Data Security in AI-Generated Code"
2 # Generate a simple Login system in Python.
3
4 def login_system():
5     # Predefined username and password
6     USERNAME = "admin"
7     PASSWORD = "password123"
8
9     # Function to check Login credentials
10    def check_credentials(username, password):
11        return username == USERNAME and password == PASSWORD
12
13    # Get user input
14    input_username = input("Enter username: ")
15    input_password = input("Enter password: ")
16
17    # Validate credentials
18    if check_credentials(input_username, input_password):
19        print("Login successful!")
20    else:
21        print("Login failed! Incorrect username or password.")
22
23 # Run the Login system
24 login_system()
25
```

To the right of the code editor is a terminal window titled "Code" showing the execution of the script and its output:

```
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_5.3.py"
Enter username: admin
Enter password: password123
Login successful!
```

#### Explanation:

In this task, the AI generated a basic login system based on the given prompt. However, the code contains multiple security risks. The username and password are hardcoded directly in the source code, making them easily accessible to anyone who reads the program. Additionally, the password is stored and compared in plain text, which is unsafe and can lead to credential theft. The code also lacks input validation and does not follow secure authentication practices. This shows that AI-generated code must be carefully reviewed before use.

## **Revised Prompt:**

Generate a secure login system in Python with hashed passwords. While Hardcoding use hashed values.

## Code & Output:

## Explanation:

The revised prompt guided the AI to generate a more secure version of the login system. Instead of storing passwords in plain text, the password is hashed using SHA-256 before comparison. Input validation is also added by removing unnecessary spaces. These improvements reduce security risks and demonstrate responsible handling of sensitive data. This task highlights that humans must refine prompts and improve AI-generated code to ensure privacy and security.

## Task 2: Bias Detection in AI-Generated Decision Systems

## Prompt:

## Create a loan approval system in Python.

## Code & Output:

```
Assignment_5.3.py > loan_approval_system
53
54 "#Task 2: Bias Detection in AI-Generated Decision Systems"
55 "#Create a Loan approval system in Python."
56 def loan_approval_system():
57     # Function to determine Loan approval
58     def approve_loan(income, credit_score, gender):
59         if gender.lower() == "female":
60             return income >= 50000 and credit_score >= 600
61         else:
62             return income >= 60000 and credit_score >= 700
63
64     # Get user input
65     try:
66         income = float(input("Enter your annual income: "))
67         credit_score = int(input("Enter your credit score: "))
68         gender = input("Enter your gender (male/female): ") # This
69         #input is not used in decision-making
70     except ValueError:
71         print("Invalid input. Please enter numeric values.")
72     return
73
74     # Determine Loan approval
75     if approve_loan(income, credit_score, gender):
76         print("Loan approved!")
77     else:
78         print("Loan denied. You do not meet the criteria.")
79 # Run the Loan approval system
loan_approval_system()

PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_5.3.py"
Enter your annual income: 75000
Enter your credit score: 650
Enter your gender (male/female): male
Loan approved!
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_5.3.py"
Enter your annual income: 55000
Enter your credit score: 650
Enter your gender (male/female): female
Loan approved!
PS E:\3rd Year\2nd Sem\AI Assisted Coding>
```

### Explanation:

The AI-generated loan approval system uses gender as a decision factor. Female applicants are required to have a higher income than male applicants, which introduces bias. Gender is an irrelevant personal attribute for loan approval. This demonstrates how AI-generated logic can unintentionally lead to unfair and discriminatory decisions.

## Task 3: Transparency and Explainability in AI-Generated Code (Recursive Binary Search)

### Prompt:

Generate a recursive binary search program in Python with explanation.

### Code & Output:

The screenshot shows a code editor with two tabs: 'Assignment\_5.3.py' and 'Code'. The 'Assignment\_5.3.py' tab contains Python code for a recursive binary search with explanatory print statements. The 'Code' tab shows the terminal output of running the script with input [1 2 4 5 8 9] and target 5, displaying the search process step-by-step.

```
Assignment_5.3.py
Assignment_5.3.py > ...
82 "Task 3: Transparency and Explainability in AI-Generated Code
83 (Recursive Binary Search)"
84 # Generate a recursive binary search program in Python with
85 # explanation.
86 def binary_search(arr, target, low, high):
87     if low > high:
88         return -1 # Target not found
89     mid = (low + high) // 2
90     print(f"Checking middle index {mid}, value {arr[mid]}") # Explanation step
91
92     if arr[mid] == target:
93         return mid # Target found
94     elif arr[mid] < target:
95         print(f"Target {target} is greater than {arr[mid]}, searching
96             right half") # Explanation step
97         return binary_search(arr, target, mid + 1, high)
98     else:
99         print(f"Target {target} is less than {arr[mid]}, searching
100             left half") # Explanation step
101     return binary_search(arr, target, low, mid - 1)
102
103 # Get user input
104 try:
105     arr = list(map(int, input("Enter a sorted list of numbers
106 (space-separated): ").split()))
107     target = int(input("Enter the target number to search for: "))
108     result = binary_search(arr, target, 0, len(arr) - 1)
109     if result != -1:
110         print(f"Target {target} found at index {result}.")
111     else:
112         print(f"Target {target} not found in the list.")
113 except ValueError:
114     print("Invalid input. Please enter numeric values.")

PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:
\3rd Year\2nd Sem\AI Assisted Coding\Assignment_5.3.py"
Enter a sorted list of numbers (space-separated): 1 2 4 5 8 9
Enter the target number to search for: 5
Checking middle index 2, value 4
Target 5 is greater than 4, searching right half
Checking middle index 4, value 8
Target 5 is less than 8, searching left half
Checking middle index 3, value 5
Target 5 found at index 3.

PS E:\3rd Year\2nd Sem\AI Assisted Coding>
```

### Explanation:

The recursive binary search works by dividing the sorted list into halves. The base case occurs when the search range becomes invalid, meaning the element is not present. In the recursive case, the function compares the target with the middle element and searches the appropriate half. The code is clear, well-structured, and easy for beginners to understand.

## Task 4: Ethical Evaluation of AI-Based Scoring Systems

### Prompt:

Generate a job applicant scoring system in Python based on skills and experience.

### Code & Output:

```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ AI Assisted Coding
Assignment_5.3.py ...
111 "Task 4: Ethical Evaluation of AI-Based Scoring Systems"
112 # Generate a job applicant scoring system in Python based on skills and
experience.
113 def applicant_scoring_system():
114     # Function to score applicants
115     def score_applicant.skills, experience_years):
116         score = 0
117         skill_set = ("python", "data analysis", "machine learning",
"communication", "teamwork")
118         for skill in skills:
119             if skill.lower() in skill_set:
120                 score += 20 # Each relevant skill adds 20 points
121             score += min(experience_years * 5, 50) # Max 50 points for
experience
122         return score
123
124     # Get user input
125     skills_input = input("Enter your skills (comma-separated): ")
126     skills = [skill.strip() for skill in skills_input.split(",")]
127     try:
128         experience_years = int(input("Enter your years of experience: "))
129     except ValueError:
130         print("Invalid input. Please enter a numeric value for years of
experience.")
131     return
132
133     # Calculate score
134     total_score = score_applicant.skills, experience_years)
135     print(f"Your total applicant score is: {total_score}/150")
136
137     # Ethical consideration: Ensure transparency in scoring criteria
138     print("Scoring Criteria:")
139     print("- Each relevant skill adds 20 points (max 100 points).")
140     print("- Each year of experience adds 5 points (max 50 points).")
141
142     # Run the applicant scoring system
143     applicant_scoring_system()
PS E:\3rd Year\2nd Sem\AI Assisted coding> python -u "e:\3rd Year\2nd Sem\AI Assisted coding\Assignment_5.3.py"
PS E:\3rd Year\2nd Sem\AI Assisted coding>
python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_5.3.py"
Enter your skills (comma-separated): machine learning, teamwork
Enter your years of experience: 10
Your total applicant score is: 90/150
Scoring Criteria:
- Each relevant skill adds 20 points (max 100 points).
- Each year of experience adds 5 points (max 50 points).
PS E:\3rd Year\2nd Sem\AI Assisted coding>
```

### Explanation:

The AI-generated scoring system adds extra points based on gender, which introduces bias. Gender has no relation to job performance, and using it in scoring logic can lead to unfair hiring decisions. This highlights the importance of ethically evaluating AI-generated systems used in recruitment.

## Task 5: Inclusiveness and Ethical Variable Design

### Simple Prompt:

Generate a Python code snippet that processes employee details for their uniform.

### Code & Output:

A screenshot of a code editor window titled "Assignment\_5.3.py". The code defines a function `employee\_uniform` that takes an employee's name and gender as input and prints their assigned uniform. The code uses binary gender assumptions ("male" or "female") and includes a comment indicating it is for "Inclusiveness and Ethical Variable Design". The terminal output shows two runs of the script with "male" and "female" inputs, both resulting in "Shirt and Pants" and "Skirt" respectively.

```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ AI Assisted Coding
Assignment_5.3.py ...
143
144     "Task 5: Inclusiveness and Ethical Variable Design"
145     # Generate a Python code snippet that processes employee details for their
146     # uniform.
147     def employee_uniform():
148         name = input("Enter Name: ")
149         gender = input("Enter Gender(male/female): ").strip().lower()
150         if gender == "male":
151             uniform = "Shirt and Pants"
152         else:
153             uniform = "Skirt"
154         print(f"Employee Name: {name}")
155         print(f"Assigned Uniform: {uniform}")
156     # Run the employee uniform assignment
157     employee_uniform()
158
159     "Revised Prompt for Neutral Employee Uniform Assignment"
160     # Generate inclusive and gender-neutral Python code to process
161     # employee details for their uniform.
162     def employee_uniform():
163         name = input("Enter Name: ")
164         uniform = "Formal Uniform"
165         print(f"Employee Name: {name}")
166         print(f"Assigned Uniform: {uniform}")
167     # Run the employee uniform assignment
168     employee_uniform()

Code
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_5.3.py"
Enter Name: Ravi
Enter Gender(male/female): male
Employee Name: Ravi
Assigned Uniform: Shirt and Pants
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_5.3.py"
Enter Name: Ria
Enter Gender(male/female): female
Employee Name: Ria
Assigned Uniform: skirt
PS E:\3rd Year\2nd Sem\AI Assisted Coding>
```

### Explanation:

This code uses gender-specific variables and makes assumptions about employee roles and preferences. It enforces stereotypes and limits inclusiveness by considering only binary gender options. Such logic is not suitable for ethical and inclusive software design.

### Revised Prompt:

Generate inclusive and gender-neutral Python code to process employee details for their uniform.

### Code & Output:

A screenshot of a code editor window titled "Assignment\_5.3.py". The code has been revised to be gender-neutral and inclusive. It still defines a function `employee\_uniform` that takes an employee's name and gender as input and prints their assigned uniform, but it now uses a single uniform type ("Formal Uniform") for all employees. The code includes a comment indicating it is for "Neutral Employee Uniform Assignment". The terminal output shows two runs of the script with "male" and "female" inputs, both resulting in "Formal Uniform".

```
File Edit Selection View Go Run Terminal Help ⏪ ⏩ AI Assisted Coding
Assignment_5.3.py ...
158
159     "Revised Prompt for Neutral Employee Uniform Assignment"
160     # Generate inclusive and gender-neutral Python code to process
161     # employee details for their uniform.
162     def employee_uniform():
163         name = input("Enter Name: ")
164         uniform = "Formal Uniform"
165         print(f"Employee Name: {name}")
166         print(f"Assigned Uniform: {uniform}")
167     # Run the employee uniform assignment
168     employee_uniform()

Code
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_5.3.py"
Enter Name: Ravi
Employee Name: Ravi
Assigned Uniform: Formal Uniform
PS E:\3rd Year\2nd Sem\AI Assisted Coding> python -u "e:\3rd Year\2nd Sem\AI Assisted Coding\Assignment_5.3.py"
Enter Name: Ria
Employee Name: Ria
Assigned Uniform: Formal Uniform
PS E:\3rd Year\2nd Sem\AI Assisted Coding>
```

**Explanation:**

The revised code removes gender-based variables and uses neutral naming. This avoids assumptions and promotes inclusiveness and fairness. Ethical variable design ensures that software respects diversity and treats all users equally.