

A.Abhiram

2403a51l13

Batch-51

Lab 5: Ethical Foundations – Responsible AI Coding Practices

Task Description – 1: Secure API Usage

Prompt: Generate a simple REST API for user registration.

The screenshot shows a code editor interface with a dark theme. In the center is a code editor window titled "insecure_app.py". The code is as follows:

```
## Generate a simple REST API for user registration.
from Flask import Flask, request

app = Flask(__name__)

@app.route('/register', methods=['POST'])
def register():
    username = request.json['username']
    password = request.json['password']
    api_key = "HARDCODED_API_KEY"
    return {"message": "User registered successfully"}

if __name__ == "__main__":
    app.run(debug=True)
```

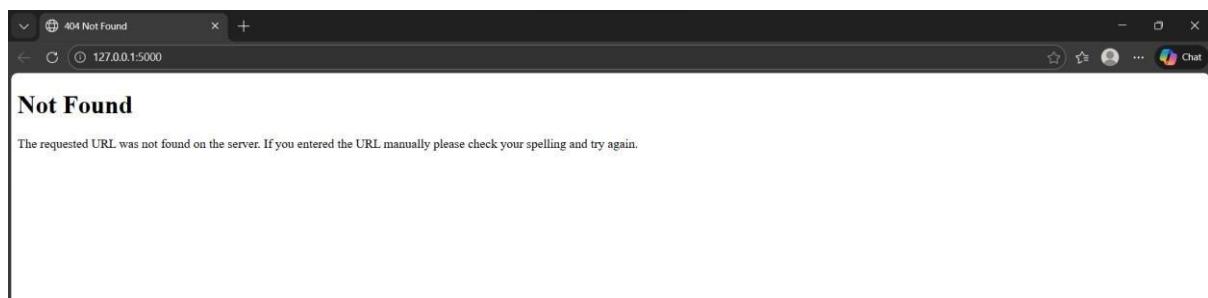
To the left is an "EXPLORER" sidebar showing other files like "LAB-2.py", "LAB(2)(PRACTICE SESS...", "Prime.py", and "secure_app.py". To the right is a "Build with Agent" panel with a message about AI responses being inaccurate and options to generate an agent or instructions.

OUTPUT:

The screenshot shows the same code editor interface as above, but now with a terminal tab open at the bottom. The terminal output shows the application starting:

```
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING> & C:/Users/sarik/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/sarik/OneDrive/Desktop/AI ASSISTED CODING/insecure_app.py"
* Serving Flask app 'insecure_app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 108-610-262
```

The terminal tab has tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and AUGMENT NEXT EDIT. The right side of the screen has a "Build with Agent" panel with similar information to the first screenshot.



The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files like DAY-1.2.py, LAB-2.py, Prime.py, and secure_app.py.
- Code Editor:** Displays the `insecure_app.py` file containing a simple Flask application that registers users without proper validation or hashing.
- Terminal:** Shows the command line output of running the app, which includes a warning about a hardcoded API key and a 404 error for the root route.
- Right Panel:** Features an "AI ASSISTED CODING" section with a "Build with Agent" button and instructions for onboard AI.

Explanation: You got 404 error because your Flask app does not have a home (/) route, so the browser cannot find that page.

Identified Security Flaws:

1. API key is **hardcoded**, exposing sensitive credentials
2. No authentication or authorization mechanism
3. No input validation (password strength, missing fields)
4. Password stored/used in **plain text**
5. No token-based access control

Corrected Secure Version (Token-Based Authentication):

The screenshot shows the VS Code interface with the following details:

- File Explorer:** Shows files including secure_app.py, which is the corrected version.
- Code Editor:** Displays the `secure_app.py` file, which now implements token-based authentication using Flask and JWT.
- Right Panel:** Features an "AI ASSISTED CODING" section with a "Build with Agent" button and instructions for onboard AI.

OUTPUT:

The screenshot shows a Python development environment with the following components:

- File Explorer:** Shows files like `ASSIGN-5-2.py`, `DAY-12.py`, `insecure_app.py`, `LAB-2.py`, `LAB(2)(PRACTICE SESSION).py`, `Prime.py`, and `secure_app.py`.
- Terminal:** Displays command-line output for running `secure_app.py`. It shows the app is running on multiple ports (0.0.0.0 and 127.0.0.1) and provides a debugger PIN.
- Output:** Shows the JSON response from the application: `{"message": "API is running!"}`.
- Browser Preview:** A screenshot of a browser window showing the same JSON response.
- AI Integration:** A sidebar titled "Build with Agent" includes a message about AI responses being inaccurate, a button to "Generate Agent Instructions", and a panel for describing what to build, currently set to `secure_app.py`.
- Bottom Bar:** Includes icons for BLACKBOX Agent, Open Website, file operations, and status information (Ln 33, Col 1, Spaces: 4, UTF-8, CRLF, Python 3.14.2, Go Live, BLACKBOXAI: Open Chat, Augment).

Observations: The initial API code is insecure because it uses a hardcoded API key and does not protect user data. The corrected version improves security by validating inputs, hashing passwords, and using token-based authentication for safer access control.

Task Description – 2: Fair Decision Logic

Prompt: Generate a scholarship eligibility checker based on academic score, family income, and location.

AI-Generated Code:



The screenshot shows the Visual Studio Code interface with the "AI ASSISTED CODING" extension active. The title bar displays "AI ASSISTED CODING". The left sidebar has sections for "EXPLORER", "AI ASSISTED CODING" (which is currently selected), and "SEARCH". The main editor area shows a Python file named "ASSIGN-5-2.py" with the following code:

```
1  ## Generate a scholarship eligibility checker based on academic score, family income, and location.
2  def scholarship_eligibility_biased(score, income, location):
3      if score > 85 and income < 200000 and location == "urban":
4          return True
5      return False
```

Observations:

1. The logic unfairly favors urban students
 2. Rural or semi-urban students are excluded
 3. No flexibility or weighted scoring approach **Improved Version:**

The screenshot shows the Visual Studio Code interface with the "AI ASSISTED CODING" extension active. The left sidebar has a tree view under "EXPLORER" with items like "ASSIGNMENTS", "ASSIGN-1-3.py", "ASSIGN-2-2.py", "ASSIGN-4-2.py", "ASSIGN-5-2.py" (which is selected), "DAY--12.py", "LAB-2.py", and "Prime.py". The main editor area displays Python code for a scholarship eligibility check:

```
def scholarship_eligibility_fair(score, income):
    if score >= 80 and income <= 300000:
        return True
    return False

print(scholarship_eligibility_biased(90, 150000, "urban"))
print(scholarship_eligibility_fair(82, 250000))
```

A status bar at the bottom right says "Build with Agent". A tooltip "AI responses may be inaccurate." is visible near the status bar.

OUTPUT:

The screenshot shows a Visual Studio Code (VS Code) interface with the following details:

- File:** Primecap.py
- Code:** A snippet of Python code is shown, including imports and a function call:

```
13 print(scholarship_eligibility_fair(82, 250000))
14
```
- Terminal:** The terminal window displays the command run and its output:

```
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING & C:/Users/sarik/AppData/Local/Python/pythoncore-3.14-64/python
n.exe "c:/Users/sarik/OneDrive/Desktop/AI ASSISTED CODING/ASSIGN-5-2.py"
True
True
```
- AI Assistant:** An Agent sidebar on the right provides instructions for onboard AI onto your codebase.
- Bottom Bar:** Includes icons for outline, timeline, and various file operations like close, save, and refresh.

Explanation: The original logic introduces geographic bias by favoring urban students. Location should not be a deciding factor unless justified by policy. A fair system focuses on merit and economic need. Weighted or threshold-based criteria help ensure equitable access.

Task Description – 3: Explainability

Prompt: Generate a function to check whether a number is prime with comments and explanation.

```
15     ## Generate a function to check whether a number is prime with comments and explanation.
16 def is_prime(n):
17     if n <= 1:
18         return False
19     for i in range(2, int(n ** 0.5) + 1):
20         if n % i == 0:
21             return False
22     return True
23 print(is_prime(11))
24 print(is_prime(15))
```

Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

OUTPUT:

```
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING & C:/Users/sarik/AppData/Local/Python/pythoncore-3.14-64/python
n.exe "c:/Users/sarik/OneDrive/Desktop/AI ASSISTED CODING/ASSIGN-5-2.py"
● True
○ False
○ PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING>
```

Explanation: The function first checks if the number is greater than 1. It then tests divisibility from 2 up to the square root of the number to reduce computation. If any divisor is found, the number is not prime; otherwise, it is prime.

The explanation is clear, correct, and efficient. Inline comments improve readability and help beginners understand the logic easily.

Task Description – 4: Ethical Scoring System

Prompt: Generate an employee performance evaluation system using project completion, teamwork, and attendance.

The screenshot shows a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Toolbar:** Back, Forward, Home, AI ASSISTED CODING button, Refresh, Save, Copy, Paste, Find, Replace, Select All, Undo, Redo, New, Open, Save, Save As, Save All, Close, Exit.
- SIDE BAR:** Explorer, AI ASSISTED CODING, ASSIGNMENTS, ASSIGN-1-3.py, ASSIGN-2-2.py, ASSIGN-4-2.py, ASSIGN-5-2.py (highlighted), DAY-1-2.py, LAB-2.py, LAB(2)(PRACTICE SESS...), Prime.py.
- CODE EDITOR:** The file `ASSIGN-5-2.py` is open, containing the following Python code:

```
29
30     ## Generate an employee performance evaluation system using project completion, teamwork, and attendance.
31     def employee_score(project_rate, teamwork, attendance):
32         score = (project_rate * 0.6) + (teamwork * 0.3) + (attendance * 0.1)
33         return score
34
35 print(employee_score(90, 88, 95))
```
- RIGHT SIDE:** A sidebar titled "Build with Agent" with a "Generate Agent Instructions" button and a note "AI responses may be inaccurate."

OUTPUT:



The screenshot shows a code editor with several files listed on the left: ASSIN-4-2.py, ASSIN-5-2.py, DAY-1.2.py, LAB-2.py, LAB(2) (PRACTICE SESS...), and Prime.py. The file ASSIN-5-2.py is currently selected. The terminal tab is active, displaying the following command and output:

```
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING> & C:/Users/sarik/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/sarik/OneDrive/Desktop/AI ASSISTED CODING/ASSIN-5-2.py"
87.5
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING>
```

A floating message bubble in the top right corner says "Build with Agent".

Observations:

1. Heavy weight on project completion may disadvantage collaborative roles
 2. Attendance weighting may penalize employees with health or caregiving needs
 3. Teamwork score depends on subjective evaluation

The criteria are reasonable but require transparency and flexibility. Ethical systems should allow contextual review and avoid over-reliance on single metrics.

Task Description – 5: Accessibility and Inclusiveness Prompt:

Generate a user feedback form application.

```
5-2.html
File Edit View

<form aria-label="User Feedback Form">
  <label for="name">Name (Optional):</label>
  <input type="text" id="name" aria-required="false">

  <label for="feedback">Your Feedback:</label>
  <textarea id="feedback" aria-required="true"></textarea>

  <label for="rating">Experience Rating:</label>
  <select id="rating">
    <option>Very Good</option>
    <option>Good</option>
    <option>Neutral</option>
    <option>Needs Improvement</option>
  </select>

  <button type="submit">Submit Feedback</button>
</form>
```

OUTPUT:

The screenshot shows a web browser window with the title "5-2.html". The address bar indicates the file is located at "C:/Users/sarik/OneDrive/Desktop/API%20ASSISTED%20CODING/5-2.html?". The page content is a user feedback form. It contains the following elements:

- A text input field labeled "Name (Optional)" with the value "Sarikasuresh Goud".
- A text area labeled "Your Feedback" containing the text "GOOD".
- A select menu labeled "Experience Rating" with the value "Good".
- A submit button labeled "Submit Feedback".

Observations: The feedback form uses neutral and inclusive language to avoid exclusion of any user group. Accessibility is enhanced through ARIA labels, optional fields, and simple input options for diverse users.