

2403a51L40

B-52

Lab 5: Ethical Foundations – Responsible AI Coding Practices

Task Description – 1: Secure API Usage

Prompt: Generate a simple REST API for user registration.

```

File Edit Selection View Go Run ... ↻ → AI ASSISTED CODING
EXPLORER ... insecure_app.py 1 insecure_app.py •
insecure_app.py > ...
1 ## Generate a simple REST API for user registration.
2 from flask import Flask, request
3
4 app = Flask(__name__)
5
6 @app.route('/register', methods=['POST'])
7 def register():
8     username = request.json['username']
9     password = request.json['password']
10    api_key = "HARDCODED_API_KEY"
11    return {"message": "User registered successfully"}
12
13 if __name__ == "__main__":
14     app.run(debug=True)

```

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

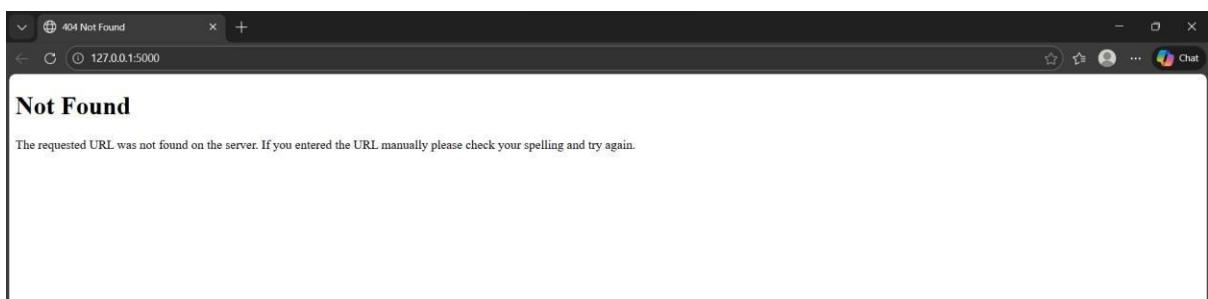
OUTPUT:

```

insecure_app.py
LAB-2.py
LAB(2)PRACTICE SESS...
Prime.py
secure_app.py 1
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS AUGMENT NEXT EDIT Python + v ⚡ ... | ⚡ X
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/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

```

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



```

    password = request.json['password']
    api_key = "HARDCODED_API_KEY"
    return {"message": "User registered successfully"}

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

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AUGMENT NEXT EDIT

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"

* Running on http://127.0.0.1:5000
* Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 108-610-262

127.0.0.1 - - [28/Jan/2026 21:46:17] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [28/Jan/2026 21:46:17] "GET /favicon.ico HTTP/1.1" 404 -

BLACKBOX Agent Open Website Ln 14, Col 24 Spaces: 4 UTF-8 CRLF { } Python 3.14.2 Go Live BLACKBOXAI: Open Chat Augment

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):

```

## Secure API (Corrected - Token-Based Authentication)
from flask import Flask, request, jsonify
app = Flask(__name__)
app.config['SECRET_KEY'] = os.getenv("SECRET_KEY", "mysecretkey")
@app.route('/', methods=['GET'])
def index():
    return jsonify({"message": "API is running!"})
@app.route('/register', methods=['POST'])
def register():
    data = request.get_json()
    if not data or not data.get('username') or not data.get('password'):
        return jsonify({"error": "Invalid input"}), 400
    hashed_password = generate_password_hash(data['password'])
    token = jwt.encode(
        {
            'user': data['username'],
            'exp': datetime.datetime.utcnow() + datetime.timedelta(hours=1)
        },
        app.config['SECRET_KEY'],
        algorithm='HS256'
    )
    return jsonify({"token": token})
if __name__ == "__main__":
    app.run(debug=True, host="0.0.0.0", port=5000)

```

EXPLORER secure_app.py ... AI ASSISTED CODING 5-2.html ASSIGN-1-3.py ASSIGN-2-2.py ASSIGN-4-2.py ASSIGN-5-2.py DAY-1-2.py insecure_app.py LAB-2.py Prime.py secure_app.py

Build with Agent

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

Describe what to build

BLACKBOX Agent Open Website Ln 33, Col 1 Spaces: 4 UTF-8 CRLF { } Python 3.14.2 Go Live BLACKBOXAI: Open Chat Augment

OUTPUT:

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

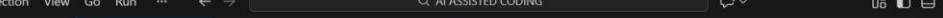
- File Explorer:** Shows files like `ASSIGN-5-2.py`, `DAY--12.py`, `insecure_app.py`, `LAB-2.py`, `LAB(2)PRACTICE SESS...`, `Prime.py`, and `secure_app.py`.
- Terminal:** Displays the command-line interface for running the application. The command `python secure_app.py` is run, and the output shows the application is serving on `http://127.0.0.1:5000`. It also includes a warning about using a development server in production.
- Output:** Shows the JSON response from the application: `{"message": "API is running!"}`.
- Code Editor:** Shows the source code for `secure_app.py`, which contains a Flask application setup with routes for `/register` and `/`.
- Bottom Status Bar:** Includes icons for file operations, a search bar, and status information like "In 33, Col 1", "Spaces: 4", "UTF-8", "CRLF", "Python 3.14.2", "Go Live", and "BLACKBOXAI Open Chat".

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 a code editor interface with a dark theme. The top bar includes standard file operations like File, Edit, Selection, View, Go, Run, and a search bar labeled "AI ASSISTED CODING". On the left, there's a sidebar titled "EXPLORER" with a "AI ASSISTED CODING" section containing links to "ASSIGNMENTS", "ASSIGN-1-3.py", "ASSIGN-2-2.py", "ASSIGN-4-2.py", and "ASSIGN-5-2.py". The main workspace displays a Python script named "ASSIGN-5-2.py" with the following code:

```
# Generate a scholarship eligibility checker based on academic score, family income, and location.
def scholarship_eligibility_biased(score, income, location):
    if score > 85 and income < 200000 and location == "urban":
        return True
    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 a code editor interface with the following details:

- File Menu:** File, Edit, Selection, View, Go, Run, ...
- Toolbar:** Back, Forward, AI ASSISTED CODING, Refresh, Save, Copy, Paste, Find, Replace, Select All, Undo, Redo.
- Sidebar (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 SESSION), Prime.py.
- Code Area:** The file `ASSIGN-5-2.py` contains the following code:

```
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))
```
- Right Panel:** Build with Agent, AI responses may be inaccurate, Generate Agent Instructions to onboard AI onto your codebase.

OUTPUT:

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

- Terminal:** The terminal window displays the command "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/ASSIGN-5-2.py"" followed by the output "True" and "True".
- Code Editor:** The code editor shows a single line of Python code: "print(scholarship_eligibility_fair(82, 250000))".
- Status Bar:** The status bar at the bottom indicates "Ln 13, Col 40" and "CRLF { Python 3.14.2".

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
20     for i in range(2, int(n ** 0.5) + 1):
21         if n % i == 0:
22             return False
23     return True
24
25 print(is_prime(11))
26 print(is_prime(15))
```

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

OUTPUT:

```
23     if n % i == 0:
24         return False
25     return True
26 print(is_prime(11))
27 print(is_prime(15))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS AUGMENT NEXT EDIT Python + ⚡ 🗑️ ... | 🔍

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/ASSIGN-5-2.py"
● True
○ False
○ PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING>

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

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 Explorer:** Shows a folder named "AI ASSISTED CODING" containing several Python files: "ASSIGN-1-3.py", "ASSIGN-2-2.py", "ASSIGN-4-2.py", "ASSIGN-5-2.py", "DAY-1.2.py", "LAB-2.py", "LAB(2)PRACTICE SESS...", and "Prime.py".
- Code Editor:** Displays the content of "ASSIGN-5-2.py".

```
29 ## Generate an employee performance evaluation system using project completion, teamwork, and attendance:
30 def employee_score(project_rate, teamwork, attendance):
31     score = (project_rate * 0.6) + (teamwork * 0.3) + (attendance * 0.1)
32     return score
33
34 print(employee_score(90, 80, 95))
```
- Right Panel:** A sidebar titled "Build with Agent" with the sub-section "AI responses may be inaccurate. Generate Agent Instructions to onboard AI onto your codebase.".

OUTPUT:

The screenshot shows a terminal window with the following details:

- File Explorer:** Shows the same files as the previous screenshot.
- Terminal:** Displays the command "python ASSIGN-5-2.py" being run and its output: "87.5".

```
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/ASSIGN-5-2.py"
87.5
PS C:\Users\sarik\OneDrive\Desktop\AI ASSISTED CODING>
```
- Right Panel:** The same "Build with Agent" sidebar as the code editor.

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/Desktop/AP%20ASSISTED%20CODING/5-2.html?". The page displays a user feedback form. The "Name (Optional)" field contains "Sarikasuresh Goud". The "Your Feedback:" field contains "6000". The "Experience Rating" dropdown menu is open, showing "Good" as the selected option. A "Submit Feedback" button is visible.

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.