

Vyshnavi Parisha

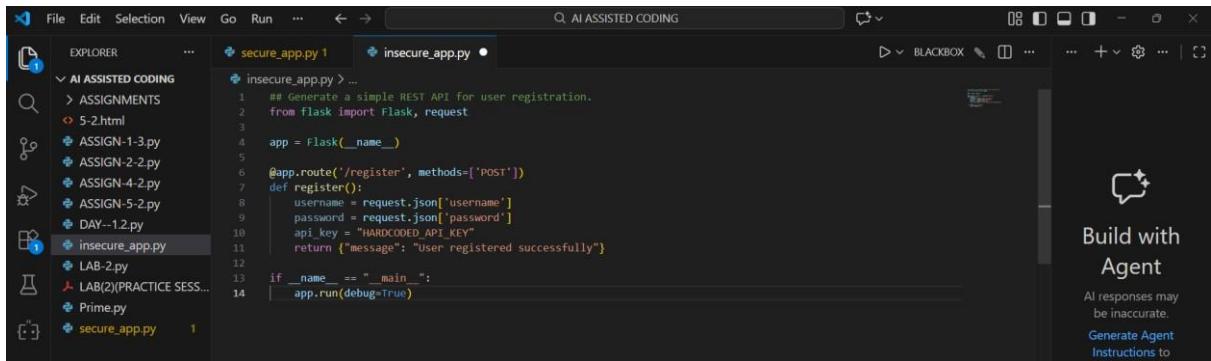
2403a51l34

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.

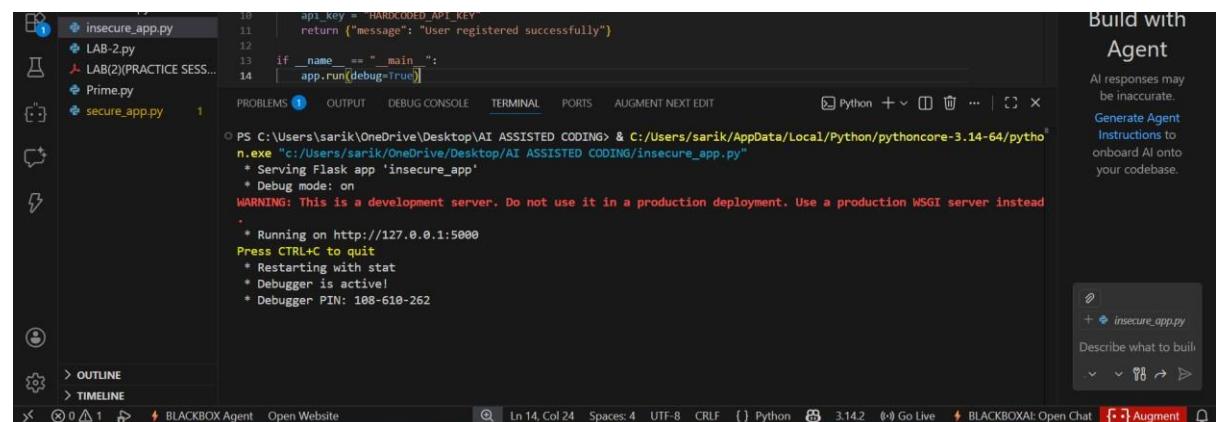


The screenshot shows a code editor interface with a dark theme. In the center, there is a code editor window titled "insecure_app.py". The code is as follows:

```
# 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 right of the code editor, there is a sidebar with the title "Build with Agent". It contains the text "AI responses may be inaccurate." and "Generate Agent Instructions to onboard AI onto your codebase." There is also a button labeled "Build with Agent".

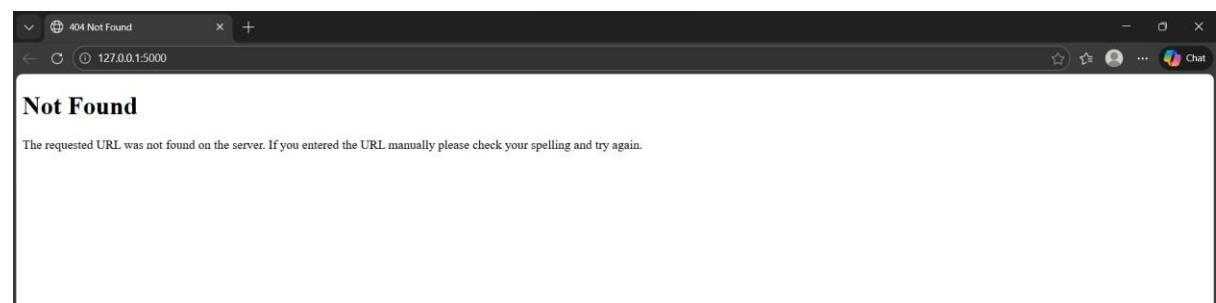
OUTPUT:



The screenshot shows a terminal window with the following output:

```
PS C:\Users\sarik\Desktop\AI ASSISTED CODING & C:/Users/sarik/AppData/Local/Python/pythoncore-3.14-64\python
n.exe "C:/Users/sarik/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
```

To the right of the terminal, there is a sidebar with the title "Build with Agent". It contains the text "AI responses may be inaccurate." and "Generate Agent Instructions to onboard AI onto your codebase." There is also a button labeled "Build with Agent".



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

- File Explorer:** Shows files like `DAY-1.2.py`, `insecure_app.py` (selected), `LAB-2-PY`, `LAB2(PRACTICE SESSION).py`, `Prime.py`, and `secure_app.py`.
- Terminal:** Displays Python code for a user registration API endpoint. The code includes imports for `json` and `http.server`, defines a function to handle POST requests, and runs the application with debug mode enabled.
- Output:** Shows the command `python insecure_app.py` running and outputting logs. The logs indicate the application is running on port 127.0.0.1:15000, and two requests are handled: a 404 for the root path and a 404 for the favicon icon.
- Status Bar:** Shows the file `insecure_app.py` is selected, along with other status indicators.

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 AI ASSISTED CODING feature in Visual Studio Code. The left sidebar displays a tree view of files, with 'secure_app.py' selected. The main code editor window contains the following Python code:

```
## 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)
```

The right side of the interface features a 'Build with Agent' panel with the following text:
Build with Agent
AI responses may be inaccurate.
Generate Agent Instructions to onboard AI onto your codebase.

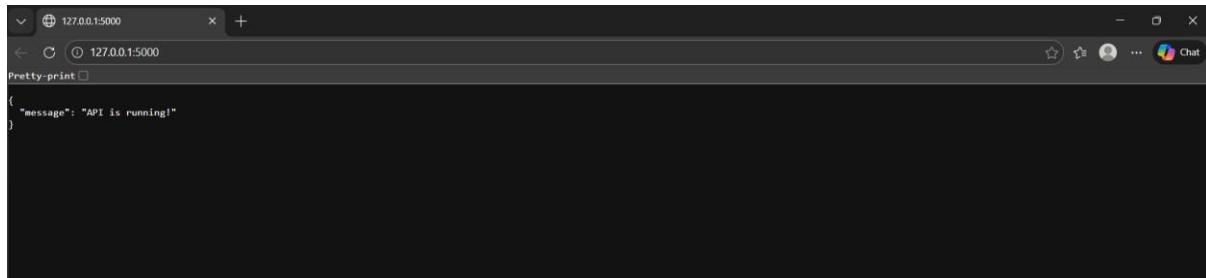
OUTPUT:

```

12 def index():
13     return jsonify({"message": "API is running!"})
14
15 @app.route('/register', methods=['POST'])
16 def register():

```

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/secure_app.py"
* Serving Flask app 'secure_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 all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.3.48.143:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 108-610-262



```

13     return jsonify({"message": "API is running!"})
14
15 @app.route('/register', methods=['POST'])
16 def register():

```

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/secure_app.py"
* Serving Flask app 'secure_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 all addresses (0.0.0.0)
* Running on http://127.0.0.1:5000
* Running on http://10.3.48.143:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 108-610-262
127.0.0.1 - - [28/Jan/2026 21:41:10] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [28/Jan/2026 21:41:10] "GET /favicon.ico HTTP/1.1" 404 -
10.3.48.143 - - [28/Jan/2026 21:41:46] "GET / HTTP/1.1" 200 -
10.3.48.143 - - [28/Jan/2026 21:41:46] "GET /favicon.ico HTTP/1.1" 404 -

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:

```
File Edit Selection View Go Run ... ← → Q AI ASSISTED CODING
EXPLORER ASSIGN-5-2.py
AI ASSISTED CODING
ASSIGNMENTS
ASSIGN-1-3.py
ASSIGN-2-2.py
ASSIGN-4-2.py
ASSIGN-5-2.py
ASSIGN-5-2.py > ...
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
6
7
```

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:

```
File Edit Selection View Go Run ... ← → Q AI ASSISTED CODING
EXPLORER ASSIGN-5-2.py
AI ASSISTED CODING
ASSIGNMENTS
ASSIGN-1-3.py
ASSIGN-2-2.py
ASSIGN-4-2.py
ASSIGN-5-2.py
ASSIGN-5-2.py > ...
1 def scholarship_eligibility_fair(score, income):
2     if score >= 80 and income <= 300000:
3         return True
4     return False
5
6 print(scholarship_eligibility_biased(90, 150000, "urban"))
7 print(scholarship_eligibility_fair(82, 250000))
```

OUTPUT:

```
File Edit Selection View Go Run ... ← → Q AI ASSISTED CODING
EXPLORER Prime.py
AI ASSISTED CODING
ASSIGNMENTS
ASSIGN-1-3.py
ASSIGN-2-2.py
ASSIGN-4-2.py
ASSIGN-5-2.py
ASSIGN-5-2.py > ...
13 print(scholarship_eligibility_fair(82, 250000))
14
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/ASSIGN-5-2.py"
True
True
```

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

+ ASSIGN-5-2.py
Describe what to build

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.

The screenshot shows a code editor interface with a dark theme. In the Explorer sidebar, there are several files listed under 'AI ASSISTED CODING': ASSIGN-1-3.py, ASSIGN-2-2.py, ASSIGN-4-2.py, DAY--1.2.py, LAB-2.py, LAB(2)(PRACTICE SESSION), and Prime.py. The main editor area displays the content of 'ASSIGN-5-2.py'. The code defines a function 'is_prime' that checks if a number is prime by testing divisibility from 2 up to the square root of the number. It includes inline comments explaining the logic. A sidebar on the right is titled 'Build with Agent' and contains a note that AI responses may be inaccurate, along with a button to 'Generate Agent Instructions'.

```
15
16     ## Generate a function to check whether a number is prime with comments and explanation.
17     def is_prime(n):
18         if n <= 1:
19             return False
20
21         for i in range(2, int(n ** 0.5) + 1):
22             if n % i == 0:
23                 return False
24
25         return True
26
27 print(is_prime(11))
28 print(is_prime(15))
```

OUTPUT:

This screenshot shows the same code editor environment as the previous one, but the terminal tab is active in the bottom navigation bar. The terminal window displays the command 'python ASSIGN-5-2.py' being run, followed by the output: 'True' and 'False'. The sidebar on the right still shows the 'Build with Agent' section.

```
23     if n % 1 == 0:
24         return False
25     return True
26 print(is_prime(11))
27 print(is_prime(15))
```

PS C:\Users\srak\OneDrive\Desktop\AI ASSISTED CODING> & C:/Users/srak/AppData/Local/Python/pythoncore-3.14-64/python.exe "c:/Users/srak/OneDrive/Desktop/AI ASSISTED CODING/ASSIGN-5-2.py"
● True
○ False
○ PS C:\Users\srak\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 title "AI ASSISTED CODING". The left sidebar is titled "EXPLORER" and lists several files under "AI ASSISTED CODING" and "ASSIGNMENTS". The main area displays a Python script named "ASSIGN-5-2.py". The code defines a function "employee_score" that calculates a score based on project rate, teamwork, and attendance. The output of the script is shown in the terminal on the right.

```
## Generate an employee performance evaluation system using project completion, teamwork, and attendance.
def employee_score(project_rate, teamwork, attendance):
    score = (project_rate * 0.6) + (teamwork * 0.3) + (attendance * 0.1)
    return score
print(employee_score(90, 80, 95))
```

OUTPUT:

The screenshot shows the same code editor interface as above, but the terminal output is visible. The command "python ASSIGN-5-2.py" is run, and the output shows the calculated score: 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>
```

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 page displays a user feedback form. The "Name (Optional)" field contains "Sarikasuresh Goud". The "Your Feedback" field contains "Good". The "Experience Rating" dropdown menu is open, showing options: "Very Good", "Good", "Neutral", and "Needs Improvement", with "Good" selected. A "Submit Feedback" button is visible at the bottom right of the form area.

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.