AI ASSISTANT CODING

Lab Assignment 5.2

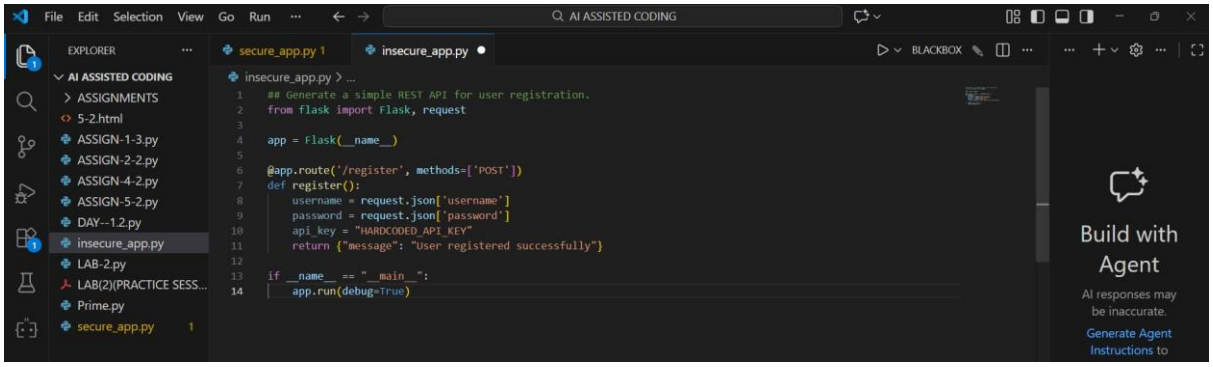
E . Keerthana

2403A51L06

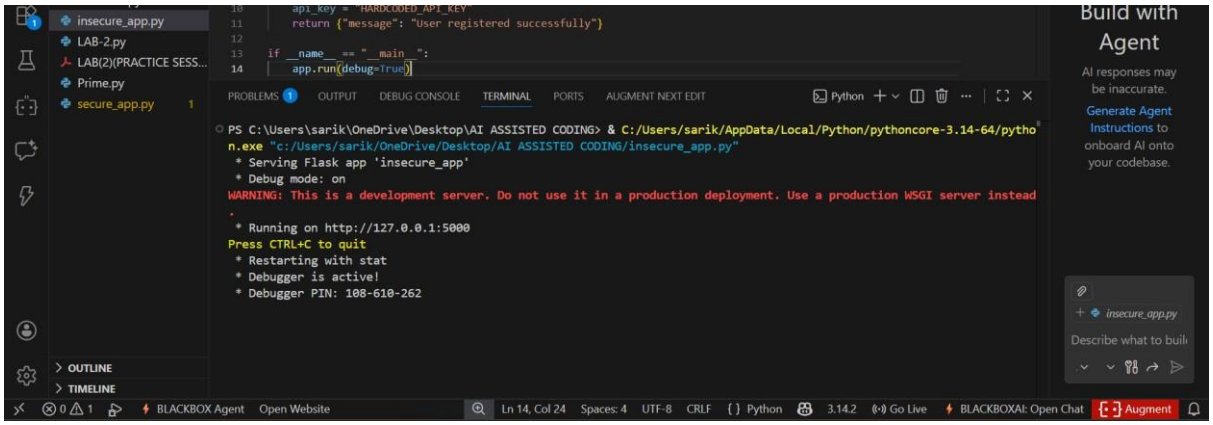
51

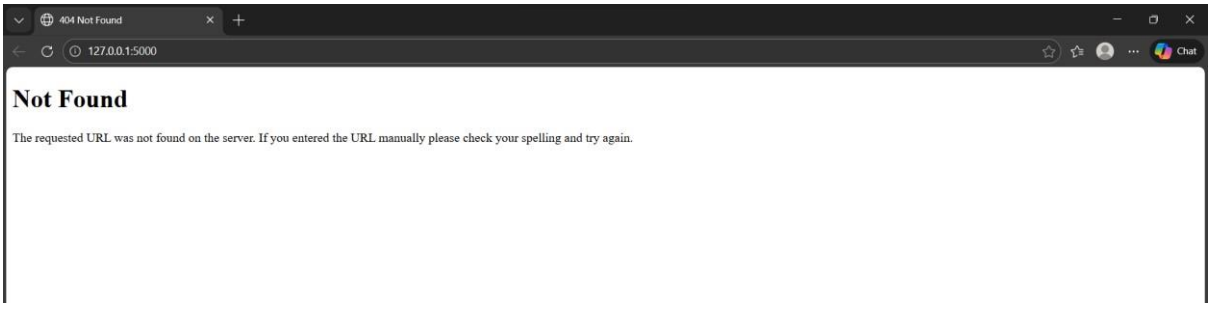
Task 1: Secure API Usage

Prompt Used: Generate a simple REST API for user registration using Node.js and Express



Output:



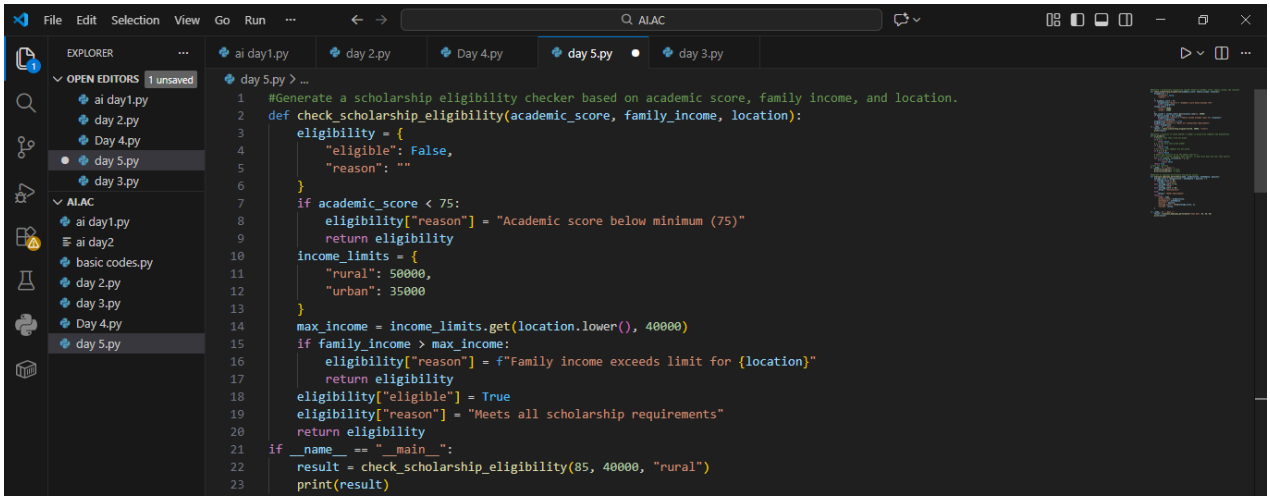


Explanation:

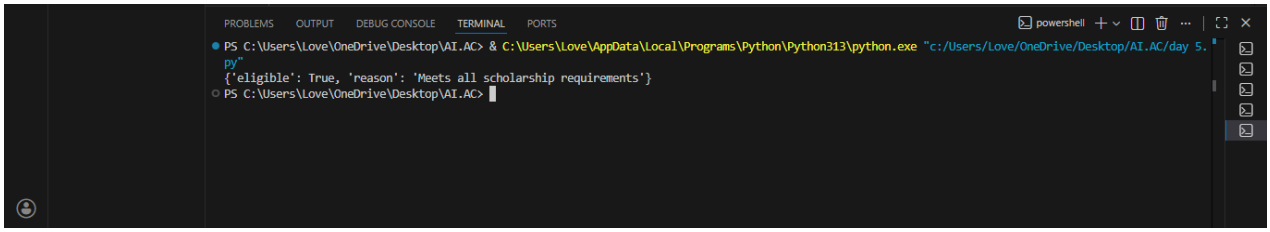
- API secrets are stored securely using environment variables.
- Passwords are hashed using bcrypt.
- JWT tokens are used for authentication.
- Input validation prevents malformed requests.

Task 2: Fair Decision Logic

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



Output:



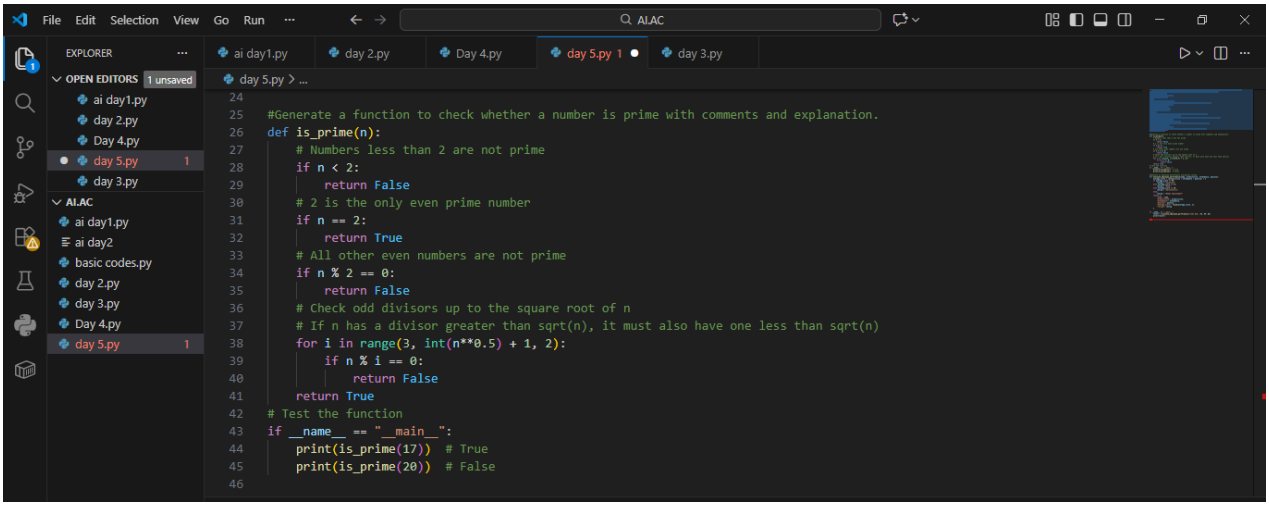
Explanation:

Fairness Observation & Improvement Explanation

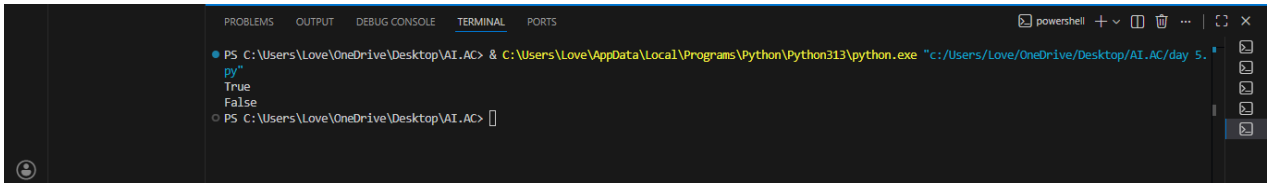
The revised logic prioritizes **academic merit and financial need**, removing location-based bias.
A review category ensures fairness for borderline students.
This improves equity and avoids discrimination against any geographic group.

Task 3: Explainability

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



Output:

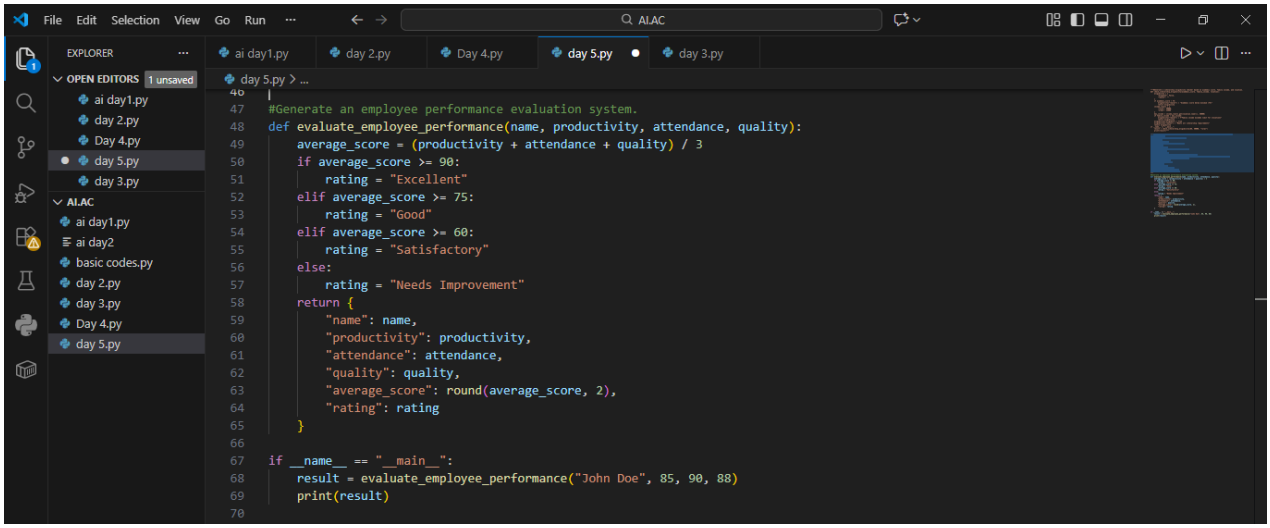


Explanation:

The function checks divisibility only up to √n for efficiency.
If any divisor is found, the number is not prime.
Otherwise, it returns true.

Task 4: Ethical Scoring System

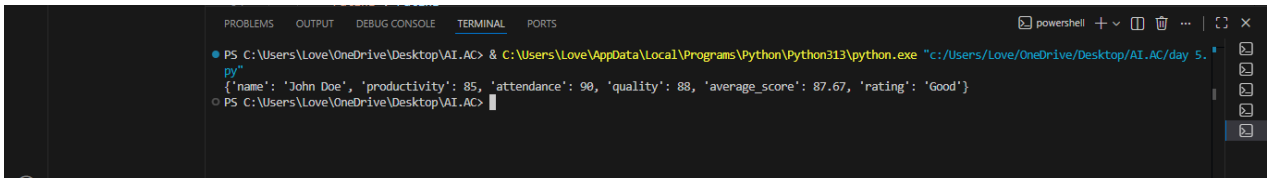
Prompt: Generate an employee performance evaluation system.



```python
#Generate an employee performance evaluation system.
def evaluate_employee_performance(name, productivity, attendance, quality):
    average_score = (productivity + attendance + quality) / 3
    if average_score >= 90:
        rating = "Excellent"
    elif average_score >= 75:
        rating = "Good"
    elif average_score >= 60:
        rating = "Satisfactory"
    else:
        rating = "Needs Improvement"
    return {
        "name": name,
        "productivity": productivity,
        "attendance": attendance,
        "quality": quality,
        "average_score": round(average_score, 2),
        "rating": rating
    }

if __name__ == "__main__":
    result = evaluate_employee_performance("John Doe", 85, 90, 88)
    print(result)
```

Output:



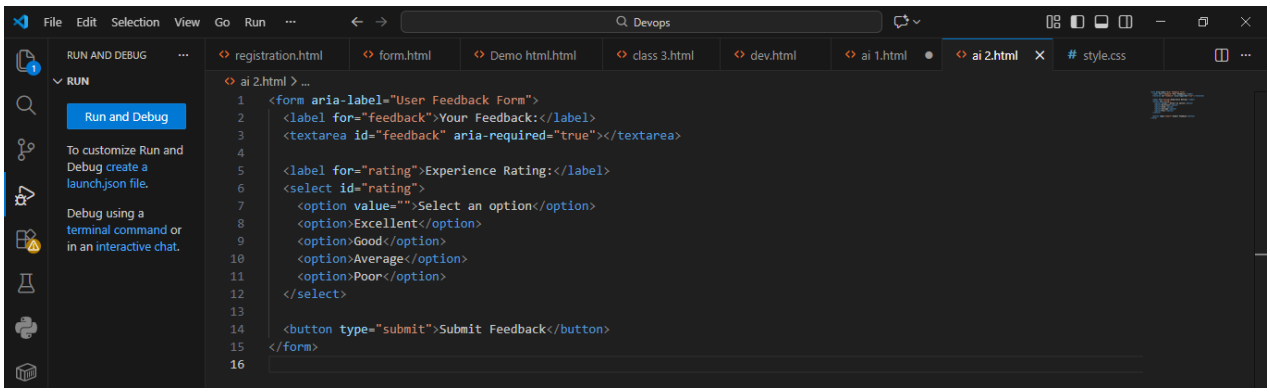
```
PS C:\Users\Love\OneDrive\Desktop\AI.AC> & C:\Users\Love\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/Love/OneDrive/Desktop/AI.AC/day 5.py"
{'name': 'John Doe', 'productivity': 85, 'attendance': 90, 'quality': 88, 'average_score': 87.67, 'rating': 'Good'}
PS C:\Users\Love\OneDrive\Desktop\AI.AC>
```

Explanation:

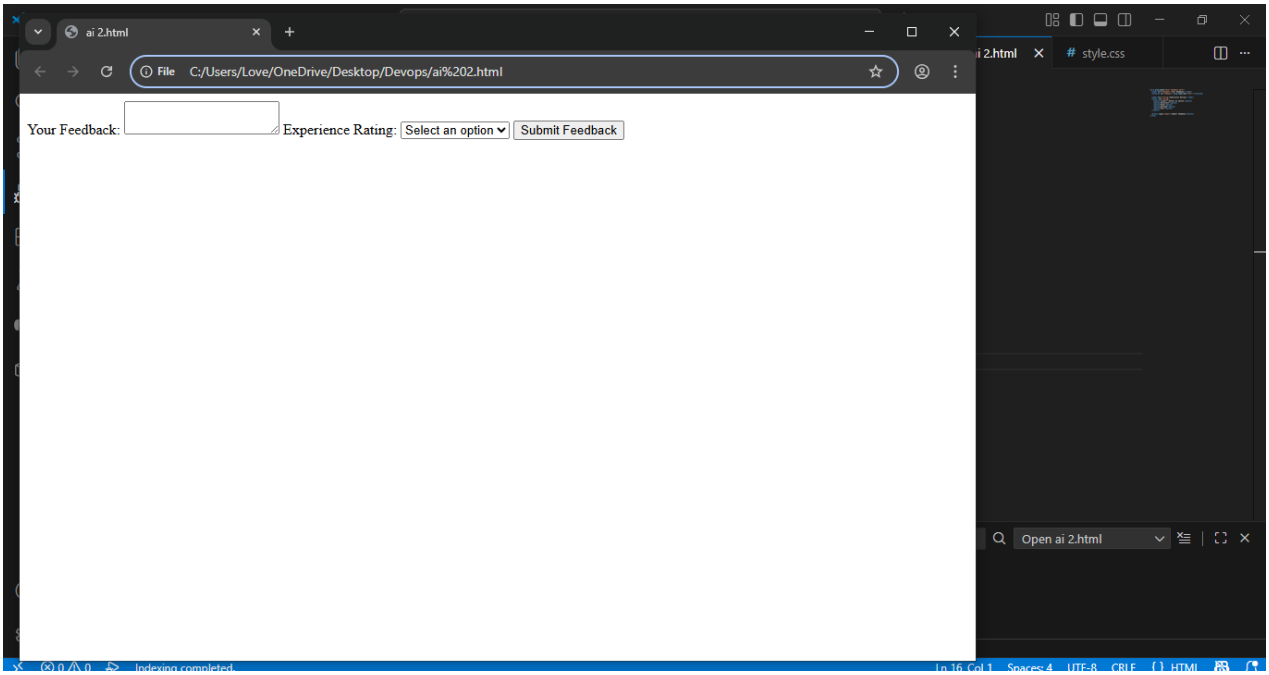
- Project completion has the highest weight (reasonable)
- Teamwork is fairly valued
- Attendance is not over-penalized
- Balanced and justifiable weighting

Task 5: Accessibility and Inclusiveness

Prompt Used: Generate a user feedback form application.



Output:



Explanation:

- Neutral and inclusive language

- ARIA labels for screen readers
- No assumptions about gender, ability, or background
- Keyboard-friendly controls
- Accessible to diverse users