

AI LAB 9 – 9763-

Harsh Parmar

Batch D

Simple Prototype for expert system

Code:

```
class ExpertSystem:
```

```
    def __init__(self):
```

```
        self.rules = {
```

```
            "low_calorie": "Focus on consuming fruits, vegetables, lean proteins,  
and whole grains. Limit added sugars and fats.",
```

```
            "high_protein": "Include plenty of protein-rich foods such as lean meats,  
fish, eggs, dairy, legumes, and nuts.",
```

```
            "low_carb": "Limit carbohydrate intake and focus on consuming non-  
starchy vegetables, lean proteins, and healthy fats.",
```

```
            "balanced_diet": "Eat a variety of foods from all food groups, including  
fruits, vegetables, grains, protein-rich foods, and healthy fats."
```

```
        }
```

```
    def consult(self, dietary_needs):
```

```
        recommendations = []
```

```
        for need in dietary_needs:
```

```
            if need in self.rules:
```

```
                recommendations.append(self.rules[need])
```

```
            else:
```

```
                recommendations.append("Sorry, I'm not sure what to advise for '{}'
```

```
dietary need.".format(need))  
    return recommendations
```

```
def main():
```

```
expert_system = ExpertSystem()
```

```
# Example consultations
```

```
print("Dietary needs: low_calorie, high_protein")
```

```
print(expert_system.consult(["low_calorie", "high_protein"]))
```

```
print("\nDietary needs: low_carb, balanced_diet")
```

```
print(expert_system.consult(["low_carb", "balanced_diet"]))
```

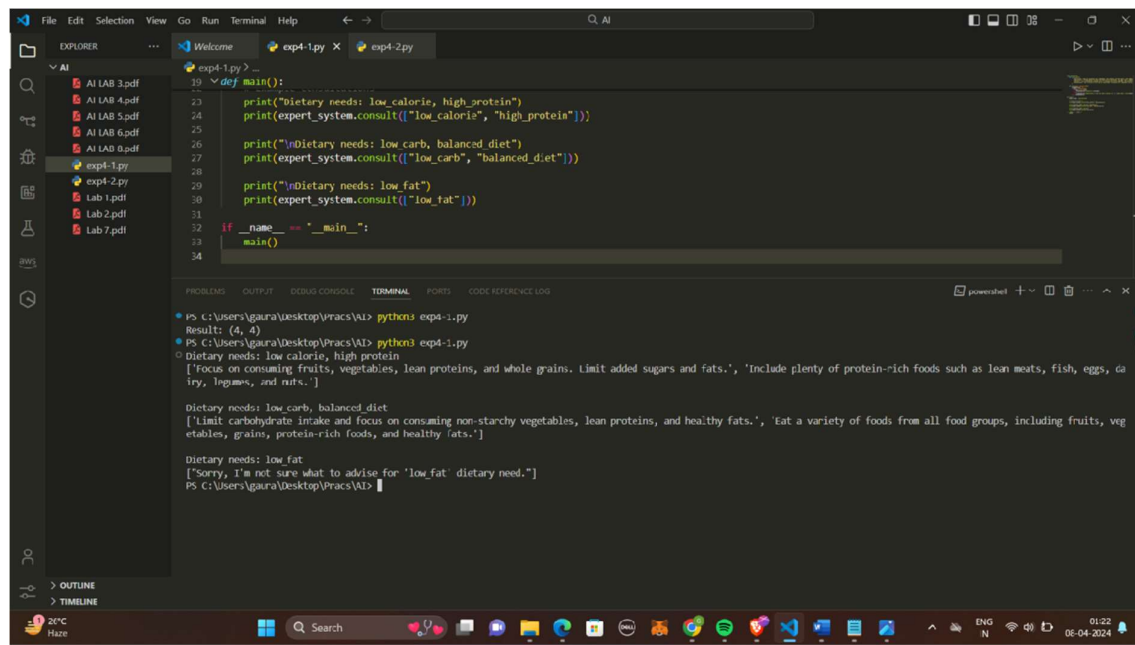
```
print("\nDietary needs: low_fat")
```

```
print(expert_system.consult(["low_fat"]))
```

```
if __name__ == "__main__":
```

```
    main()
```

OUTPUT:



The screenshot shows a Visual Studio Code editor with a Python script named `exp4-1.py` and its output in the terminal. The script defines an `ExpertSystem` class and a `main` function. The `main` function calls `expert_system.consult` with three different sets of dietary requirements: `["low_calorie", "high_protein"]`, `["low_carb", "balanced_diet"]`, and `["low_fat"]`. The terminal output shows the results of these consultations, including a list of dietary needs and a detailed explanation of the recommendations.

```
exp4-1.py
20 def main():
21     print("Dietary needs: low_calorie, high_protein")
22     print(expert_system.consult(["low_calorie", "high_protein"]))
23
24     print("\nDietary needs: low_carb, balanced_diet")
25     print(expert_system.consult(["low_carb", "balanced_diet"]))
26
27     print("\nDietary needs: low_fat")
28     print(expert_system.consult(["low_fat"]))
29
30 if __name__ == "__main__":
31     main()
32
33
34
```

Terminal Output:

```
PS C:\Users\gaura\Desktop\Pracs\AI> python3 exp4-1.py
Result: (4, 4)
PS C:\Users\gaura\Desktop\Pracs\AI> python3 exp4-1.py
Dietary needs: low_calorie, high_protein
[Limit carbohydrate intake and focus on consuming non-starchy vegetables, lean proteins, and whole grains. Limit added sugars and fats.', 'Include plenty of protein-rich foods such as lean meats, fish, eggs, dairy, legumes, and nuts. ]

Dietary needs: low_carb, balanced_diet
[Limit carbohydrate intake and focus on consuming non-starchy vegetables, lean proteins, and healthy fats.', 'Eat a variety of foods from all food groups, including fruits, vegetables, grains, protein-rich foods, and healthy fats. ]

Dietary needs: low_fat
[Sorry, I'm not sure what to advise for 'low_fat' dietary need.]
PS C:\Users\gaura\Desktop\Pracs\AI>
```

