Rudra Vaja A. 240905041036

AI Career Advisor – **Practical Implementation Based on AI Components**

Overview

This project, AI Career Advisor, is a Python-based simulation of a decision-making AI system that recommends an ideal career path to users based on their interests and preferences. It applies fundamental AI principles such as reasoning, learning behavior simulation, and user interaction — as explained in the theoretical section of the manual.

Components of AI Demonstrated

1. Learning (Simulated)

While not trained on real-world datasets, the system mimics **supervised learning** by using a predefined rule-based knowledge structure (user inputs → career output).

Example: If a user enjoys coding and math, the system infers a high score for the "Engineer" role.

2. Reasoning

This project uses **deductive reasoning**: it applies a set of logical rules to infer the best-matching career based on user choices. Each answer affects the score of one or more career paths.

3. Limited Memory

The program retains **short-term input data (answers to five questions) during runtime, and uses this memory to perform scoring and make a decision. This is a good example of a "Limited Memory AI system", which uses recent data to improve results without long-term memory storage.

4. NLP/Interaction

Even though it doesn't use advanced NLP, the interface allows simple natural language interaction using `input()` — making the system interactive and user-driven.

Categorization of Intelligent System Used:

AI Type	Characteristics in this Project
Limited Memory	Remembers temporary answers and calculates scores.
Reasoning System	Applies rule-based logic to infer a career path.
Reactive Machine	Does not store previous sessions; works based on current input.

Real-World Application Connection

This project simulates basic functionality used in:

Career Counseling Platforms (e.g., Mettl, CareerGuide)

Skill Assessment Engines

AI Chatbots for Education & HR

CODE :

```
# AI Career Advisor Project
def ask_questions():
    print("Welcome to the AI Career Advisor!\nAnswer the following questions with Y/N:")

    q1 = input("Do you enjoy solving math problems? (Y/N): ").strip().lower()
    q2 = input("Do you like coding or building software? (Y/N): ").strip().lower()
    q3 = input("Are you interested in human anatomy or biology? (Y/N): ").strip().lower()
    q4 = input("Do you like drawing, designing, or creating visuals? (Y/N): ").strip().lower()
    q5 = input("Are you good at communicating and leading people? (Y/N): ").strip().lower()

    return q1, q2, q3, q4, q5

def suggest_career(answers):
    score = {
        "Engineer": 0,
        "Doctor": 0,
        "Artist": 0,
        "Manager": 0
    }

    q1, q2, q3, q4, q5 = answers

    if q1 == 'y':
        score["Engineer"] += 1
        score["Doctor"] += 1
    if q2 == 'y':
        score["Engineer"] += 2
    if q3 == 'y':
        score["Doctor"] += 2
    if q4 == 'y':
        score["Artist"] += 2
    if q5 == 'y':
        score["Manager"] += 2

    best_fit = max(score, key=score.get)
    return best_fit

def main():
 answers = ask_questions()
    career = suggest_career(answers)
    print(f"\n🔍 Based on your answers, a suitable career path for you is: {career}")


if __name__ == "__main__":
    main()
print("\nRudra Vaja\n240905041036")
```

OUTPUT :

```
Welcome to the AI Career Advisor!

Answer the following questions with Y/N:

Do you enjoy solving math problems? (Y/N): y

Do you like coding or building software? (Y/N): y

Are you interested in human anatomy or biology? (Y/N): y

Do you like drawing, designing, or creating visuals? (Y/N): y

Are you good at communicating and leading people? (Y/N): y


 Based on your answers, a suitable career path for you is: **Engineer**
```

Rudra Vaja

240905041036

Real-world Application Relevance:

This project mimics real-world **AI-based career counseling tools** that:

- Help students choose streams based on interest
- Provide job suggestions from aptitude test data
- Power online career recommendation platforms (like CareerGuide, Mettl, etc.)

Advantages:

- Easy to use
- Provides quick feedback
- Helps in decision-making based on personal interests

Future Improvements:

- Add **machine learning** to predict careers using past data
- Use **GUI** (Tkinter) for better interaction
- Integrate **natural language understanding** for open-ended answers