

# Assignment 02: Building an Expert System Using Rule-Based Systems

---

## 1. Objective

The objective of this assignment is to **design and implement a simple Expert System** that can provide **career path suggestions** based on student interests.

This demonstrates the basic principles of **Artificial Intelligence (AI)**, **decision-making algorithms**, and **rule-based systems**.

---

## 2. Problem Statement

Students often face confusion while selecting the right career path based on their interests and strengths.

An **Expert System** can help by mimicking a human advisor and providing suggestions based on predefined rules.

In this assignment, we create:

1. A **basic version** using Python `if-else` conditions.
  2. An **advanced version** using the `experta` rule-based AI library.
- 

## 3. What is an Expert System?

An **Expert System** is a computer program that **simulates the decision-making ability of a human expert**.

It consists of:

- **Knowledge Base:** Stores facts and rules about a specific domain.

- **Inference Engine:** Applies rules to the knowledge base to draw conclusions or make decisions.

In our case, the domain is **career guidance** based on **student interests**.

---

## 4. Tools and Technologies

- **Programming Language:** Python
  - **Interface:** Command Line Interface (CLI)
  - **Logic Engine:** `experta` (Python library for Expert Systems)
- 

## 5. Knowledge Base (Rules)

Sample rules for career suggestions:

- IF `Maths` AND `Physics` → **Mechanical Engineering**
  - IF `Programming` AND `Maths` → **Computer Engineering**
  - IF `Biology` AND `Chemistry` → **Biotechnology**
  - IF `Circuits` AND `Maths` → **Electronics Engineering**
  - IF `Programming` AND `Statistics` → **AI and Data Science**
  - IF `Programming` AND `AI Concepts` → **AI and Machine Learning**
- 

## 6. Implementation

### 6.1 Method 1 – Using If-Else Conditions (Basic Version)

```
def career_expert_system(interests):
```

```

interests = [i.strip().lower() for i in interests] # clean input

if "maths" in interests and "physics" in interests:
    return "Mechanical Engineering"
elif "programming" in interests and "maths" in interests:
    return "Computer Engineering"
elif "biology" in interests and "chemistry" in interests:
    return "Biotechnology"
elif "circuits" in interests and "maths" in interests:
    return "Electronics Engineering"
elif "programming" in interests and "statistics" in interests:
    return "Artificial Intelligence and Data Science"
elif "programming" in interests and "ai concepts" in interests:
    return "Artificial Intelligence and Machine Learning Engineering"
else:
    return "Sorry, no matching career found. Try different interests."

def main():
    print("Welcome to the Career Path Expert System!")
    user_input = input("Enter your interests separated by commas (e.g., Maths, Physics, Programming): ")
    interests = user_input.split(',')

    suggestion = career_expert_system(interests)
    print("Suggested Career Path:", suggestion)

if __name__ == "__main__":
    main()

```

---

## 6.2 Method 2 – Using **experta** Rule-Based System (Advanced Version)

```

from experta import *

# Define facts
class StudentFacts(Fact):
    pass

# Define rules
class CareerExpertSystem(KnowledgeEngine):

    @Rule(StudentFacts(likes='Maths'), StudentFacts(likes='Physics'))
    def mechanical(self):
        print("Suggested Career Path: Mechanical Engineering")

```

```

@Rule(StudentFacts(likes='Programming'), StudentFacts(likes='Maths'))
def computer(self):
    print("Suggested Career Path: Computer Engineering")

@Rule(StudentFacts(likes='Biology'), StudentFacts(likes='Chemistry'))
def biotech(self):
    print("Suggested Career Path: Biotechnology")

@Rule(StudentFacts(likes='Circuits'), StudentFacts(likes='Maths'))
def electronics(self):
    print("Suggested Career Path: Electronics Engineering")

@Rule(StudentFacts(likes='Programming'), StudentFacts(likes='Statistics'))
def ai_data_science(self):
    print("Suggested Career Path: Artificial Intelligence and Data Science")

@Rule(StudentFacts(likes='Programming'), StudentFacts(likes='AI Concepts'))
def ai_ml(self):
    print("Suggested Career Path: Artificial Intelligence and Machine Learning Engineering")

def main():
    engine = CareerExpertSystem()
    engine.reset() # Prepare engine

    print("Welcome to the Career Path Expert System!")
    interests = input("Enter your interests separated by commas (e.g., Maths, Physics,
Programming): ").split(',')

    # Declare facts for each interest
    for interest in interests:
        engine.declare(StudentFacts(likes=interest.strip()))

    engine.run() # Run rule engine

if __name__ == "__main__":
    main()

```

---

## 7. Sample Output

### Case 1: Input → Maths, Physics

Welcome to the Career Path Expert System!

Enter your interests separated by commas (e.g., Maths, Physics, Programming): Maths, Physics

Suggested Career Path: Mechanical Engineering

### Case 2: Input → Programming, Maths

Welcome to the Career Path Expert System!

Enter your interests separated by commas (e.g., Maths, Physics, Programming): Programming, Maths

Suggested Career Path: Computer Engineering

### Case 3: Input → Biology, Chemistry

Welcome to the Career Path Expert System!

Enter your interests separated by commas (e.g., Maths, Physics, Programming): Biology, Chemistry

Suggested Career Path: Biotechnology

---

## 8. Conclusion

In this assignment, we successfully built an **Expert System for Career Path Suggestion** using:

1. **If-Else Conditions** – Simple decision-making system.
2. **Rule-Based System with Experta** – Real AI-style expert system.

This project demonstrates how **rule-based systems** can mimic human decision-making and can be applied in real-world problems like **career counseling, medical diagnosis, and financial advising**.