Finance Expert System Using **Symbolic AI**

Name **:** Kavy Singh Ranawat
Registration No : GF202342831
Class : CSU2219

Table of Contents

Abstract

This project presents a Finance Expert System developed using Symbolic Artificial Intelligence (AI). Symbolic AI, also known as rule-based AI, represents knowledge in the form of explicit rules and facts, enabling logical reasoning to derive conclusions. The system is designed to assist individuals in making informed financial decisions by analyzing key factors such as income, expenses, age, risk appetite, savings, debt, and emergency funds.

The Finance Expert System applies forward-chaining inference rules to provide personalized recommendations on savings, investments, debt management, and emergency planning. It demonstrates how symbolic AI can be applied in real-life domains to build simple yet effective advisory systems. The system's modular design allows for future enhancements, such as tax planning, retirement planning, insurance recommendations, and goal-based investment strategies.

This work highlights the practical application of symbolic AI concepts in building expert systems and shows how rule-based reasoning can be applied effectively in the field of financial advisory.

Objectives

The main objectives of this Finance Expert System project are:

1. **To understand and apply Symbolic AI concepts** by developing a rule-based expert system.

2. **To provide financial recommendations** tailored to a user's income, expenses, savings, debt, age, and risk appetite.

3. **To demonstrate the use of a knowledge base and inference engine** for logical decision-making.

4. **To encourage better financial habits** by advising on savings, investments, debt management, and emergency funds.

5. **To design a modular and extendable system** that can be enhanced with additional features such as tax planning, retirement planning, insurance, and goal-based investing.

Methodology

The development of the **Finance Expert System** follows a structured symbolic AI approach. This section explains the methodology used to build the system.

1. **Problem Definition**
    - The project aims to assist users in making better financial decisions using symbolic AI.
2. **Knowledge Acquisition**
    - Financial rules were gathered from common guidelines in personal finance (savings, investments, debt, emergency funds).
3. **Knowledge Representation**
    - The knowledge is represented as **IF-THEN rules** in Python code.
    - Example:
        - *IF savings ratio < 20% → Recommend: Increase savings.*
        - *IF risk appetite = high AND age < 40 → Recommend: Invest in stocks and mutual funds.*
4. **Inference Mechanism**
    - A simple **forward chaining approach** is used.
    - User inputs are matched against the knowledge base, and appropriate advice is generated.
5. **Implementation**
    - The system is implemented in Python with a command-line interface.
    - Inputs are collected (income, expenses, savings, debt, age, risk appetite, emergency funds).
    - Outputs are displayed as personalized financial recommendations.

System Design

The Finance Expert System is designed using a **rule-based architecture**, which consists of four major components:

Knowledge Base

- Contains financial rules related to savings, investments, debt, and emergency funds.

Example rule:
`IF savings_ratio < 20% → Recommend: Increase savings.`

`IF risk_appetite = high AND age < 40 → Recommend: Stocks and Mutual Funds.`

Inference Engine

- Implements forward chaining using conditional `if-else` statements in Python.
- It applies rules to the user's input and derives conclusions (financial advice).

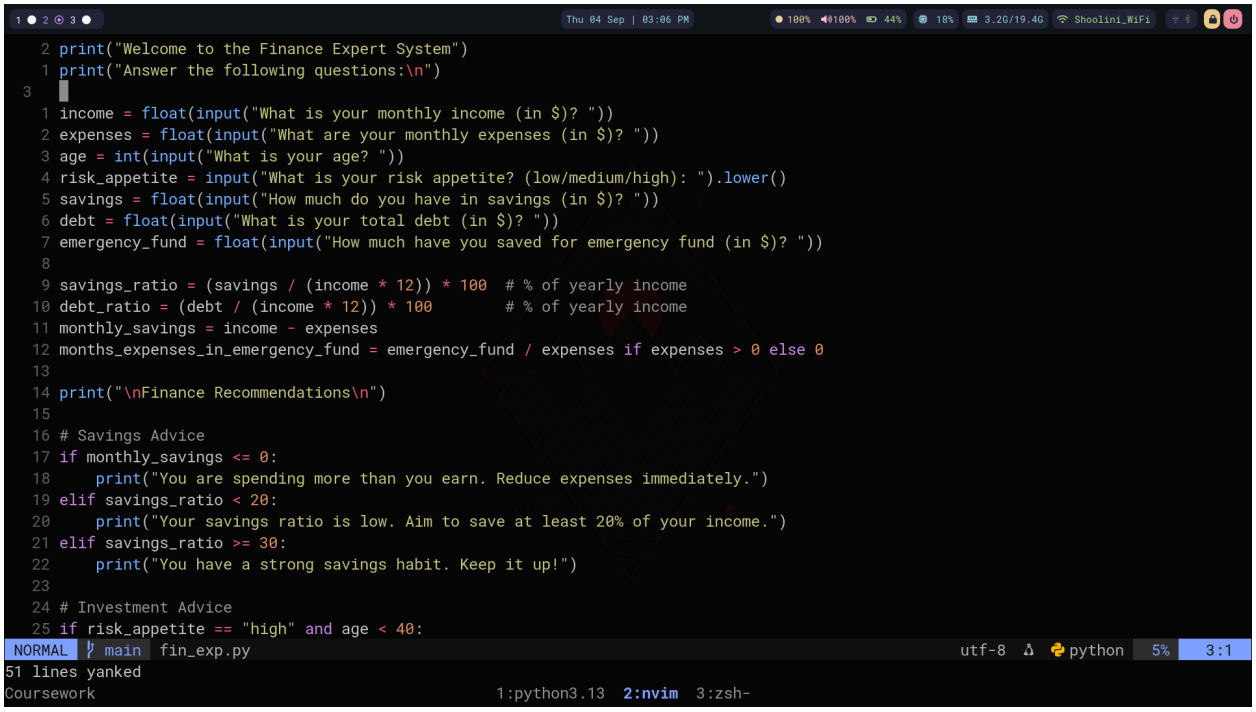
User Interface

- A command-line interface where the user inputs financial details such as income, expenses, age, savings, debt, and emergency funds.
- Displays personalized recommendations based on the applied rules.

Process Flow

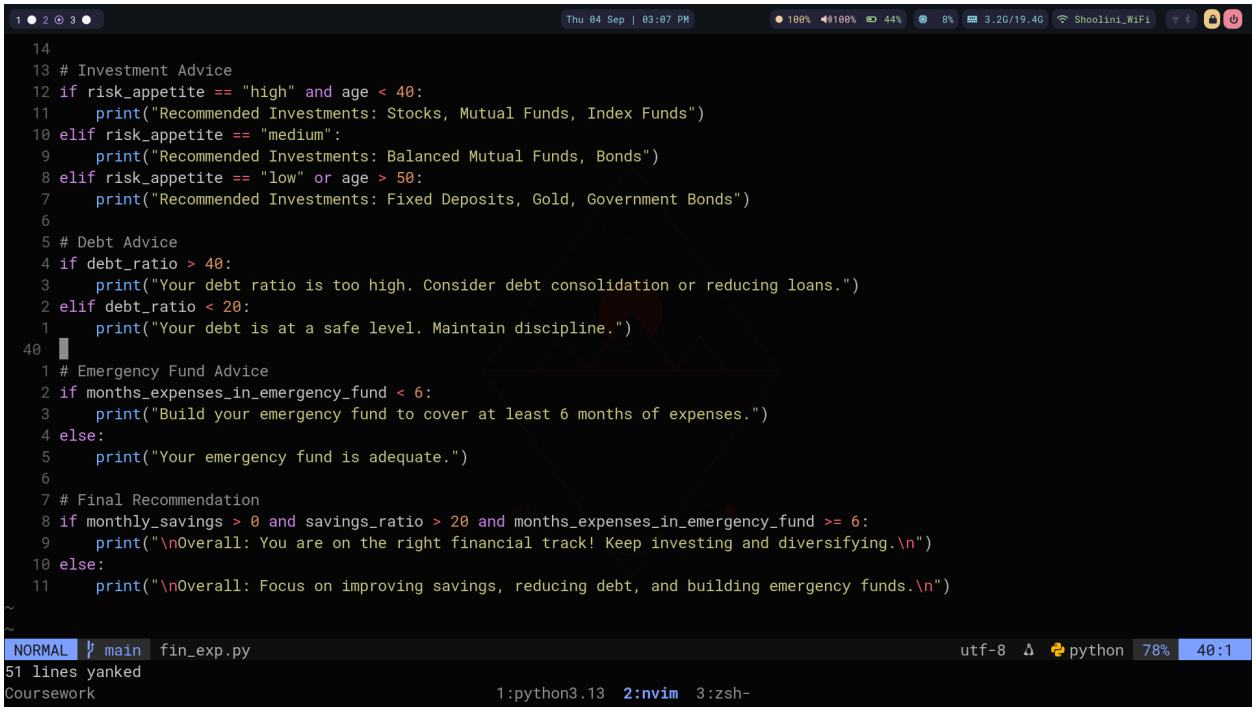
- The user provides input data.
- The inference engine applies relevant rules from the knowledge base.
- The system outputs financial advice, including savings guidance, investment suggestions, debt management tips, and emergency fund planning.

Implementation



```
print("Welcome to the Finance Expert System")
print("Answer the following questions:\n")

income = float(input("What is your monthly income (in $)? "))
expenses = float(input("What are your monthly expenses (in $)? "))
age = int(input("What is your age? "))
risk_appetite = input("What is your risk appetite? (low/medium/high): ").lower()
savings = float(input("How much do you have in savings (in $)? "))
debt = float(input("What is your total debt (in $)? "))
emergency_fund = float(input("How much have you saved for emergency fund (in $)? "))

savings_ratio = (savings / (income * 12)) * 100  # % of yearly income
debt_ratio = (debt / (income * 12)) * 100        # % of yearly income
monthly_savings = income - expenses
months_expenses_in_emergency_fund = emergency_fund / expenses if expenses > 0 else 0

print("\nFinance Recommendations\n")

# Savings Advice
if monthly_savings <= 0:
    print("You are spending more than you earn. Reduce expenses immediately.")
elif savings_ratio < 20:
    print("Your savings ratio is low. Aim to save at least 20% of your income.")
elif savings_ratio >= 30:
    print("You have a strong savings habit. Keep it up!")

# Investment Advice
if risk_appetite == "high" and age < 40:
```

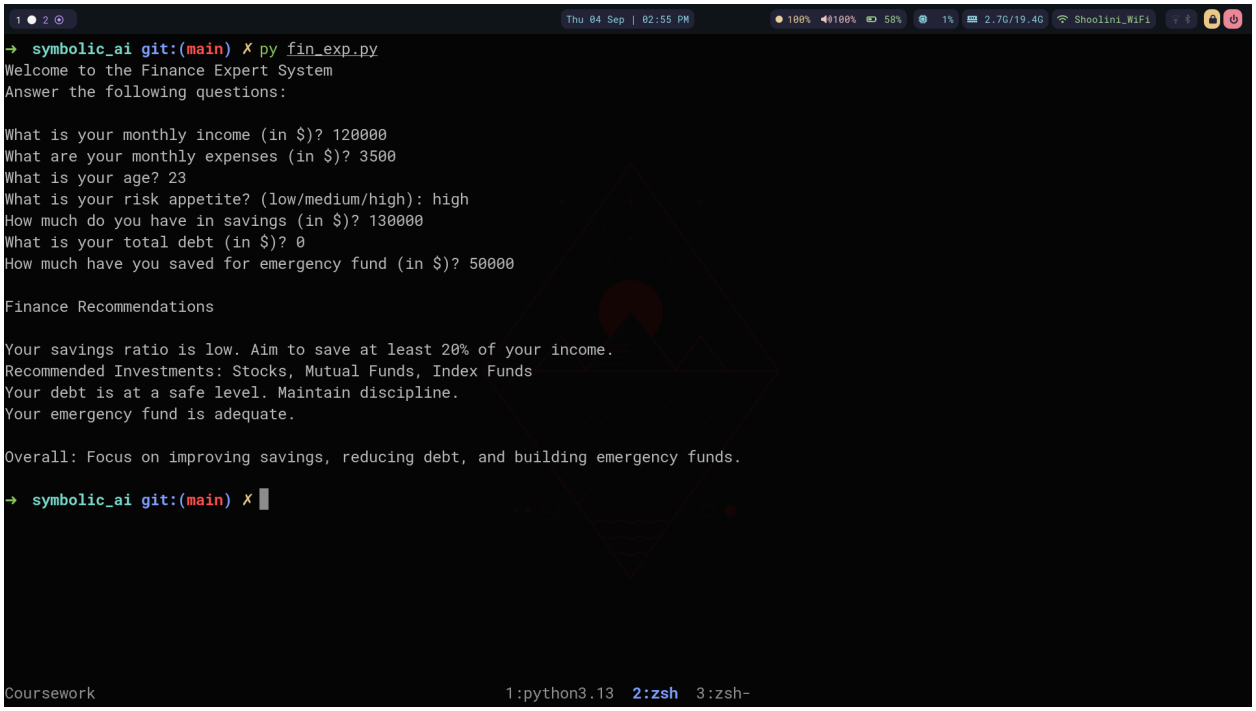
```
NORMAL  main  fin_exp.py                                   utf-8  python  5%  3:1
51 lines yanked
Coursework                          1:python3.13  2:nvim  3:zsh-
```



```
# Investment Advice
if risk_appetite == "high" and age < 40:
    print("Recommended Investments: Stocks, Mutual Funds, Index Funds")
elif risk_appetite == "medium":
    print("Recommended Investments: Balanced Mutual Funds, Bonds")
elif risk_appetite == "low" or age > 50:
    print("Recommended Investments: Fixed Deposits, Gold, Government Bonds")

# Debt Advice
if debt_ratio > 40:
    print("Your debt ratio is too high. Consider debt consolidation or reducing loans.")
elif debt_ratio < 20:
    print("Your debt is at a safe level. Maintain discipline.")

# Emergency Fund Advice
if months_expenses_in_emergency_fund < 6:
    print("Build your emergency fund to cover at least 6 months of expenses.")
else:
    print("Your emergency fund is adequate.")

# Final Recommendation
if monthly_savings > 0 and savings_ratio > 20 and months_expenses_in_emergency_fund >= 6:
    print("\nOverall: You are on the right financial track! Keep investing and diversifying.\n")
else:
    print("\nOverall: Focus on improving savings, reducing debt, and building emergency funds.\n")
```

```
NORMAL  main  fin_exp.py                                   utf-8  python  78%  40:1
51 lines yanked
Coursework                          1:python3.13  2:nvim  3:zsh-
```

Results


Thu 04 Sep | 02:55 PM
Shoolini_WiFi
→ symbolic_ai git:(main) ✗ py fin_exp.py
Welcome to the Finance Expert System
Answer the following questions:
What is your monthly income (in $)? 120000
What are your monthly expenses (in $)? 3500
What is your age? 23
What is your risk appetite? (low/medium/high): high
How much do you have in savings (in $)? 130000
What is your total debt (in $)? 0
How much have you saved for emergency fund (in $)? 50000
Finance Recommendations
Your savings ratio is low. Aim to save at least 20% of your income.
Recommended Investments: Stocks, Mutual Funds, Index Funds
Your debt is at a safe level. Maintain discipline.
Your emergency fund is adequate.
Overall: Focus on improving savings, reducing debt, and building emergency funds.
→ symbolic_ai git:(main) ✗
Coursework
1:python3.13 2:zsh 3:zsh-

Conclusion & Future Work

The Finance Expert System successfully applies **Symbolic AI** to personal financial planning. Using a rule-based approach, it provides tailored advice on savings, investments, debt, and emergency funds. The project shows that simple **knowledge representation and inference** can simulate expert reasoning and guide users toward better financial habits.

Future Work

The system can be improved with:

- **Tax and Retirement Planning** modules.
- **Insurance and Goal-based Recommendations**.
- A **Graphical User Interface (GUI)** for easier interaction.
- **Dynamic rule updates** and integration with real financial data.

With these enhancements, the system could grow into a comprehensive personal finance advisor.