University of Moratuwa
Faculty of Information Communication Technology

CM3320 - Logic Programming & Artificial Cognitive Systems
Assignment 02

FRAMEWIZARD

(EXPERT SYSTEM)

Name: -Thareejan P.

Index Number: -204216D

Table of Contents

1. Introduction

Introducing FrameWizard, an intelligent system built with CLIPS, a potent rule-based programming language. FrameWizard simplifies the complex task of choosing frameworks for software projects. Using a set of rules, it navigates project size, programming language, development speed, and user-friendliness to offer personalized recommendations. While not the most user-friendly, FrameWizard leverages CLIPS' robust capabilities to provide a comprehensive solution for framework decisions, ensuring projects align with optimal development environments. Explore FrameWizard, where precision meets complexity in the world of software framework selection.

2. Identified Problem

The problem identified that led to the creation of FrameWizard is the difficulty developers face when choosing the right backend frameworks for their projects. In the vast world of programming languages and frameworks, developers often find it challenging to navigate the numerous available options. The complexity arises from factors such as project size, language preferences, development speed, and beginner-friendliness. The lack of a straightforward decision-making process results in time-consuming research and trial-and-error methods, leading to frustration and inefficiency. FrameWizard is designed to tackle this problem by offering a structured and rule-based system to guide developers through the maze of choices. Its development is driven by the acknowledgment of the challenges developers encounter in finding the best backend framework, with the goal of simplifying this process and improving the efficiency of software development decisions.

3. Proposed Solution

The proposed solution, FrameWizard, presents a systematic and rule-driven approach to streamline the intricate process of selecting backend frameworks. By leveraging the capabilities of CLIPS, a powerful rule-based programming language, FrameWizard offers a structured decision-making framework. Developers can now navigate the complexities of project size, programming language, development speed, and beginner-friendliness with precision. This expert system aims to eliminate the struggle associated with researching and experimenting with various frameworks, providing a reliable and efficient tool for informed decision-making in software development. With FrameWizard, developers gain a comprehensive solution that simplifies the framework selection process and enhances the overall efficiency of backend development.

3.1. Architecture Diagram

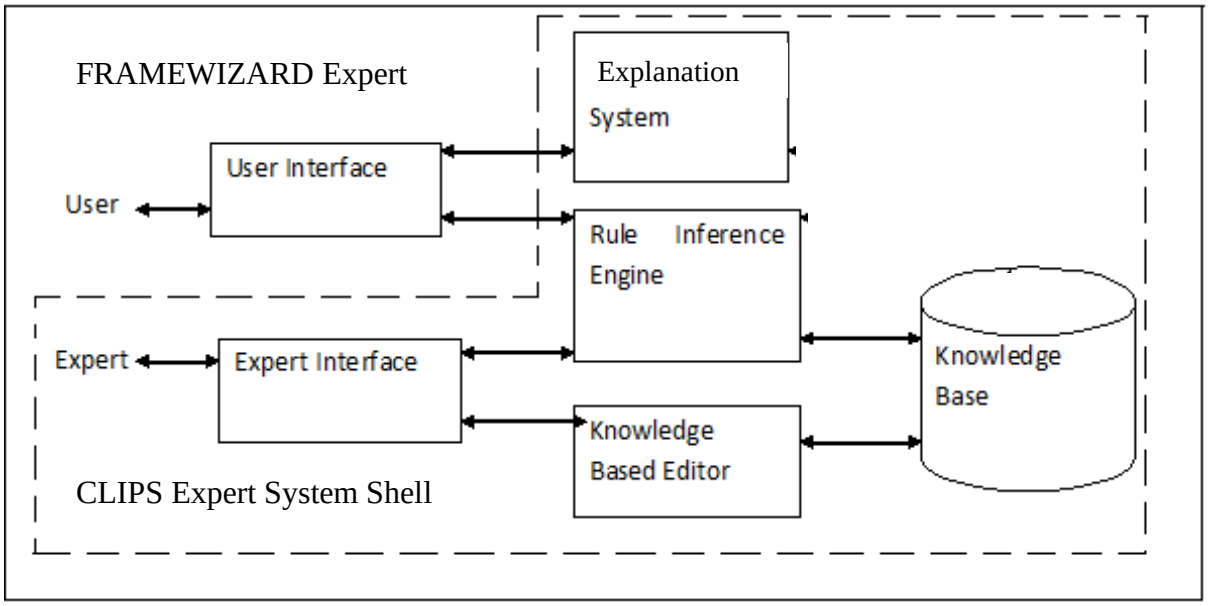


Figure 1 Architecture Diagram

3.1.1. Knowledge Base

The Knowledge Base of the FrameWizard Expert System serves as the repository of rules, logic, and information essential for making informed recommendations in the realm of backend framework selection. Developed using CLIPS, a rule-based programming language, the Knowledge Base encompasses a comprehensive set of rules that encapsulate the nuanced criteria for choosing a suitable framework. It navigates through variables such as project size, preferred programming language, development speed, and the expertise level of developers. Each rule is meticulously crafted to capture the intricate relationships between these factors, ensuring that the system can adeptly analyze user inputs and provide tailored suggestions. The Knowledge Base stands as the backbone of FrameWizard, empowering it to navigate the complexities of framework decision-making with precision and efficiency.

3.1.1.1. Facts

The Knowledge Base of the FrameWizard Expert System is built upon a foundation of extensive research and industry best practices. It incorporates a vast array of facts, encompassing details about various backend frameworks, programming languages, and their respective strengths and weaknesses. Each fact is carefully curated to reflect the current landscape of software development, considering factors such as project scalability, performance, community support, and developer preferences

1. **Spring Boot:**
    - Project Size: Large
    - Language: Java
    - Development Speed: Medium
    - Beginner-Friendly: Hard
2. **Grails:**
    - Project Size: Large
    - Language: Java
    - Development Speed: Very High
    - Beginner-Friendly: Beginner-Friendly
3. **Django:**
    - Project Size: Large
    - Language: Python
    - Development Speed: Very High
    - Beginner-Friendly: Beginner-Friendly
4. **Ruby on Rails:**
    - Project Size: Large
    - Language: Ruby
    - Development Speed: Very High
    - Beginner-Friendly: Beginner-Friendly
5. **ASP.NET Core:**
    - Project Size: Large
    - Language: C#
    - Development Speed: Medium
    - Beginner-Friendly: Hard
6. **Symfony:**
    - Project Size: Large
    - Language: PHP
    - Development Speed: Medium
    - Beginner-Friendly: Hard
7. **NancyFX:**
    - Project Size: Large
    - Language: PHP
    - Development Speed: Medium
    - Beginner-Friendly: Hard

8. **NestJS:**
    - Project Size: Large
    - Language: JavaScript
    - Development Speed: Medium
    - Beginner-Friendly: Hard
9. **Hapi.js:**
    - Project Size: Large
    - Language: JavaScript
    - Development Speed: Very High
    - Beginner-Friendly: Beginner-Friendly
10. **Express.js:**
    - Project Size: Medium/Small
    - Language: JavaScript
    - Development Speed: Very High
    - Beginner-Friendly: Hard
11. **Koa.js:**
    - Project Size: Medium/Small
    - Language: JavaScript
    - Development Speed: Very High
    - Beginner-Friendly: Beginner-Friendly
12. **Flask:**
    - Project Size: Medium/Small
    - Language: Python
    - Development Speed: Medium
    - Beginner-Friendly: Beginner-Friendly
13. **FastAPI:**
    - Project Size: Medium/Small
    - Language: Python
    - Development Speed: Very High
    - Beginner-Friendly: Beginner-Friendly
14. **Sinatra:**
    - Project Size: Medium/Small
    - Language: Ruby
    - Development Speed: Very High
    - Beginner-Friendly: Beginner-Friendly
15. **Laravel:**
    - Project Size: Medium/Small
    - Language: PHP
    - Development Speed: Very High
    - Beginner-Friendly: Hard
16. **SlimPHP:**
    - Project Size: Medium/Small
    - Language: PHP
    - Development Speed: Very High

- Beginner-Friendly: Beginner-Friendly

17. **Dropwizard:**
    - Project Size: Medium/Small
    - Language: Java
    - Development Speed: Medium
    - Beginner-Friendly: Beginner-Friendly

3.1.1.2. Rules

1. **Spring Boot:**
    - If the project size is large and the language is Java, and development speed is medium, and it's hard for beginners, then Spring Boot is suitable.
2. **Grails:**
    - If the project size is large, the language is Java, and development speed is very high, and it's beginner-friendly, then Grails is recommended.
3. **Django:**
    - If the project size is large, the language is Python, development speed is very high, and it's beginner-friendly, then Django is a good choice.
4. **Ruby on Rails:**
    - If the project size is large, the language is Ruby, development speed is very high, and it's beginner-friendly, then Ruby on Rails is recommended.
5. **ASP.NET Core:**
    - If the project size is large, the language is C#, development speed is medium, and it's hard for beginners, then ASP.NET Core is suitable.
6. **Symfony:**
    - If the project size is large, the language is PHP, development speed is medium, and it's hard for beginners, then Symfony is a viable choice.
7. **NancyFX:**
    - If the project size is large, the language is PHP, development speed is medium, and it's hard for beginners, then NancyFX is recommended.
8. **NestJS:**
    - If the project size is large, the language is JavaScript, development speed is medium, and it's hard for beginners, then NestJS is suitable.
9. **Hapi.js:**
    - If the project size is large, the language is JavaScript, development speed is very high, and it's beginner-friendly, then Hapi.js is a good choice.
10. **Express.js:**
    - If the project size is medium/small, the language is JavaScript, development speed is very high, and it's hard for beginners, then Express.js is recommended.
11. **Koa.js:**
    - If the project size is medium/small, the language is JavaScript, development speed is very high, and it's beginner-friendly, then Koa.js is a viable choice.

12. **Flask:**
    - If the project size is medium/small, the language is Python, development speed is medium, and it's beginner-friendly, then Flask is recommended.
13. **FastAPI:**
    - If the project size is medium/small, the language is Python, development speed is very high, and it's beginner-friendly, then FastAPI is suitable.

14. **Sinatra:**
    - If the project size is medium/small, the language is Ruby, development speed is very high, and it's beginner-friendly, then Sinatra is a good choice.
15. **Laravel:**
    - If the project size is medium/small, the language is PHP, development speed is very high, and it's hard for beginners, then Laravel is recommended.
16. **SlimPHP:**
    - If the project size is medium/small, the language is PHP, development speed is very high, and it's beginner-friendly, then SlimPHP is a viable choice.
17. **Dropwizard:**
    - If the project size is medium/small, the language is Java, development speed is medium, and it's beginner-friendly, then Dropwizard is recommended.

3.1.2. Inference Engine

The inference engine used in this expert system is the one provided by CLIPS (C Language Integrated Production System). CLIPS includes a forward-chaining inference engine that evaluates the defined rules and executes actions based on the given facts. In this system, the inference engine processes user inputs through a series of rules, considering conditions and making recommendations for the most suitable backend technology. The forward-chaining mechanism allows the system to derive conclusions from the available information, providing a dynamic and responsive decision-making process.

3.1.3. UI

The UI in this expert system is command-line interface (CLI)-based, providing a text-based interaction between the user and the system. Users are prompted with questions related to their software project requirements, and they respond by entering answers through the command line. The system processes these inputs, evaluates rules, and dynamically progresses through the decision-making process. The CLI interface allows for a straightforward and user-friendly interaction, making it accessible for users to receive recommendations for suitable backend technologies based on their preferences and project specifications.

3.2. Knowledge Representation

Table 1 Knowledge Representation in Tabular Format

Framework	**Project Size**	**Language**	**Development Speed**	**Beginner-Friendly**
Spring Boot	Large	Java	Medium	Hard
Grails	Large	Java	Very High	Beginner-Friendly
Django	Large	Python	Very High	Beginner-Friendly
Ruby on Rails	Large	Ruby	Very High	Beginner-Friendly
ASP.NET Core	Large	C#	Medium	Hard
Symfony	Large	PHP	Medium	Hard
NancyFX	Large	PHP	Medium	Hard
NestJS	Large	JavaScript	Medium	Hard
Hapi.js	Large	JavaScript	Very High	Beginner-Friendly

Express.js	Medium/Small	JavaScript	Very High	Hard
Koa.js	Medium/Small	JavaScript	Very High	Beginner-Friendly
Flask	Medium/Small	Python	Medium	Beginner-Friendly
FastAPI	Medium/Small	Python	Very High	Beginner-Friendly
Sinatra	Medium/Small	Ruby	Very High	Beginner-Friendly
Laravel	Medium/Small	PHP	Very High	Hard
SlimPHP	Medium/Small	PHP	Very High	Beginner-Friendly
Dropwizard	Medium/Small	Java	Medium	Beginner-Friendly

3.3. Implementation

CLIPS Script

```
; Thareejan P.
; Index Number 204216D
; FrameWizard – Backend Technology Recommendation for Software Projects
; Expert System - CLIPS
(deffacts startup
   (start))

(defrule p1
   ?p <- (start)
   =>
   (printout t "What is the size of your project? (Large/Medium)" crlf)
   (assert (projectsize (read)))
   (retract ?p))

(defrule p2
   (projectsize Large)
   =>
   (printout t "What is your desired development speed? (Very-High/Medium)" crlf)
   (assert (speed (read))))

(defrule p3
   (projectsize Large)
   (speed Very-High)
   =>
    (printout t "Is beginner-friendliness important to you? (Yes/No)" crlf)
    (assert (beginnerfriendly (read))))

(defrule p4
   (projectsize Large)
   (speed Very-High)
   (beginnerfriendly Yes)
   =>
      (printout t "Which programming language do you prefer? (Java/Python/Ruby/C#/PHP/JavaScript)" crlf)
      (assert (language (read))))

(defrule rule1
```

```
    (projectsize Large)
    (speed Very-High)
    (beginnerfriendly Yes)
    (language Java)
    =>
     (printout t "Based on your preferences, the recommended framework for your project is Grails" crlf))

(defrule rule2
    (projectsize Large)
    (speed Very-High)
    (beginnerfriendly Yes)
    (language Python)
    =>
     (printout t "Based on your preferences, the recommended framework for your project is Django" crlf))

(defrule rule3
    (projectsize Large)
    (speed Very-High)
    (beginnerfriendly Yes)
    (language Ruby)
    =>
     (printout t "Based on your preferences, the recommended framework for your project is Ruby on Rails" crlf))

(defrule rule4
    (projectsize Large)
    (speed Very-High)
    (beginnerfriendly Yes)
    (language JavaScript)
    =>
     (printout t "Based on your preferences, the recommended framework for your project is Hapi.js" crlf))

(defrule p5
    (projectsize Large)
    (speed Medium)
    =>
     (printout t "Is beginner-friendliness important to you? (Yes/No)" crlf)
     (assert (beginnerfriendly (read))))

(defrule p6
    (projectsize Large)
```

```
    (speed Medium)
    (beginnerfriendly No)
    =>
        (printout t "Which programming language do you prefer? (Java/Python/Ruby/C#/PHP/JavaScript)" crlf)
        (assert (language (read))))

(defrule rule5
    (projectsize Large)
    (language Java)
    (speed Medium)
    (beginnerfriendly No)
    =>
    (printout t "Based on your preferences, the recommended framework for your project is Spring Boot" crlf))

(defrule rule6
    (projectsize Large)
    (language C#)
    (speed Medium)
    (beginnerfriendly No)
    =>
    (printout t "Based on your preferences, the recommended framework for your project is ASP.NET Core" crlf))

(defrule rule7
    (projectsize Large)
    (language PHP)
    (speed Medium)
    (beginnerfriendly No)
    =>
    (printout t "Based on your preferences, the recommended framework for your project is Symfony" crlf))

(defrule rule8
    (projectsize Large)
    (language PHP)
    (speed Medium)
    (beginnerfriendly No)
    =>
    (printout t "Based on your preferences, the recommended framework for your project is NancyFX" crlf))

(defrule rule9
    (projectsize Large)
```

```
    (language JavaScript)
    (speed Medium)
    (beginnerfriendly No)
    =>
    (printout t "Based on your preferences, the recommended framework for your project is NextJS" crlf))

(defrule rule10
        (projectsize Large)
        (language JavaScript)
        (speed Medium)
        (beginnerfriendly No)
        =>
        (printout t "Based on your preferences, the recommended framework for your project is ExpressJS" crlf))

(defrule rule11
        (projectsize Large)
        (language JavaScript)
        (speed Medium)
        (beginnerfriendly No)
        =>
        (printout t "Based on your preferences, the recommended framework for your project is NestJS" crlf))


(defrule rule13
        (projectsize Medium)
        (language JavaScript)
        (speed Very-High)
        (beginnerfriendly Beginner-Friendly)
        =>
        (printout t "Based on your preferences, the recommended framework for your project is Koa.js" crlf))

(defrule rule14
        (projectsize Medium)
        (language Python)
        (speed Medium)
        (beginnerfriendly Beginner-Friendly)
        =>
        (printout t "Based on your preferences, the recommended framework for your project is Flask" crlf))

(defrule rule15
```

```
        (projectsize Medium)
        (language Python)
        (speed Very-High)
        (beginnerfriendly Beginner-Friendly)
        =>
        (printout t "Based on your preferences, the recommended framework for your project is FastAPI" crlf))

(defrule rule16
        (projectsize Medium)
        (language Ruby)
        (speed Very-High)
        (beginnerfriendly Beginner-Friendly)
        =>
        (printout t "Based on your preferences, the recommended framework for your project is Sinatra" crlf))

(defrule rule17
        (projectsize Medium)
        (language PHP)
        (speed Very-High)
        (beginnerfriendly Hard)
        =>
        (printout t "Based on your preferences, the recommended framework for your project is Laravel" crlf))

(defrule rule18
        (projectsize Medium)
        (language PHP)
        (speed Very-High)
        (beginnerfriendly Beginner-Friendly)
        =>
        (printout t "Based on your preferences, the recommended framework for your project is SlimPHP" crlf))

(defrule rule19
        (projectsize Medium)
        (language Java)
        (speed Medium)
        (beginnerfriendly Beginner-Friendly)
        =>
        (printout t "Based on your preferences, the recommended framework for your project is Dropwizard" crlf))
```

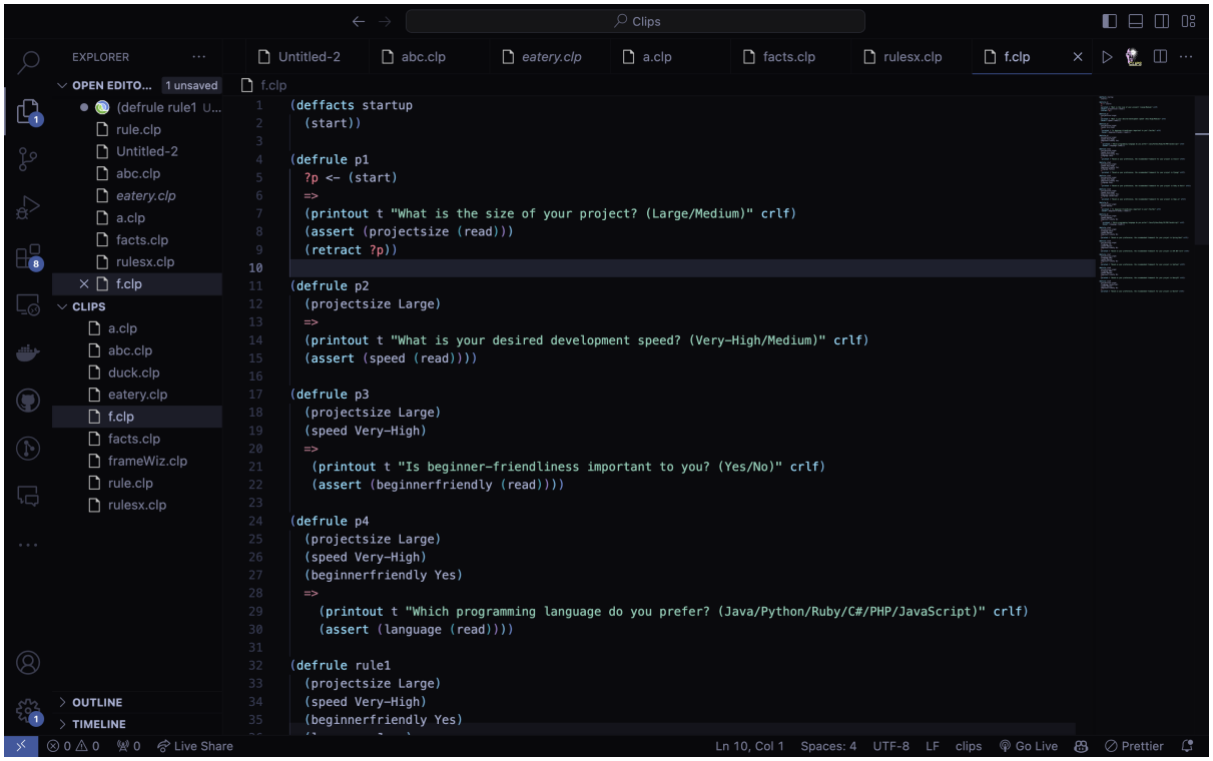


Figure 3 Implementation 1

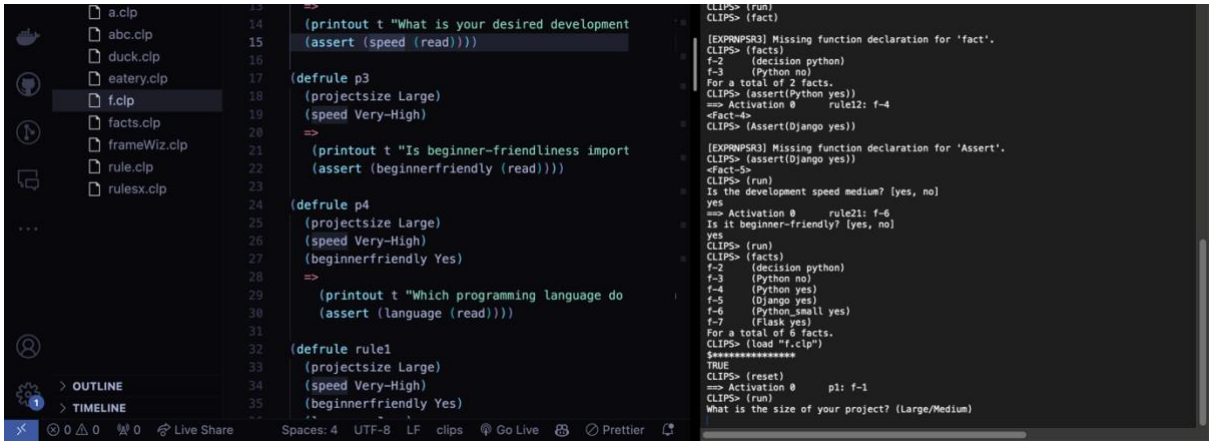


Figure 2 Implementation 2

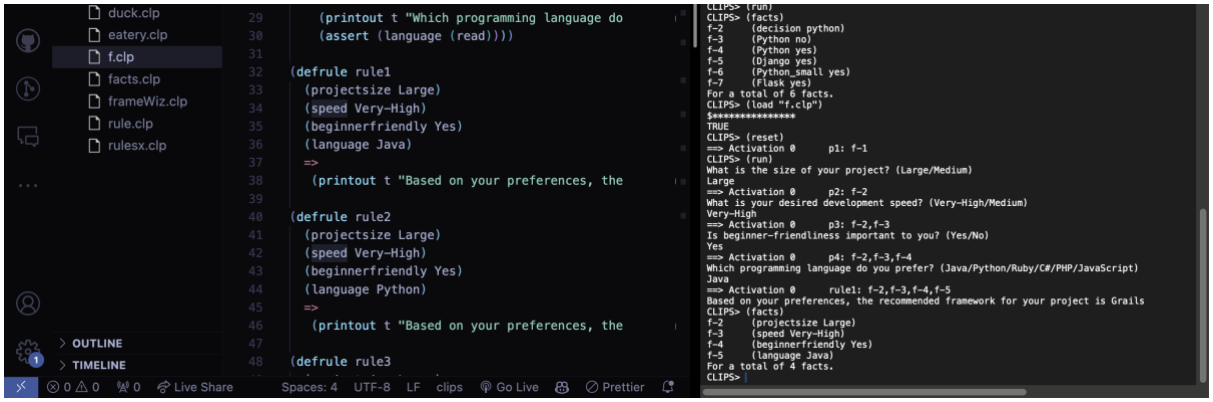


Figure 4Implementation 3

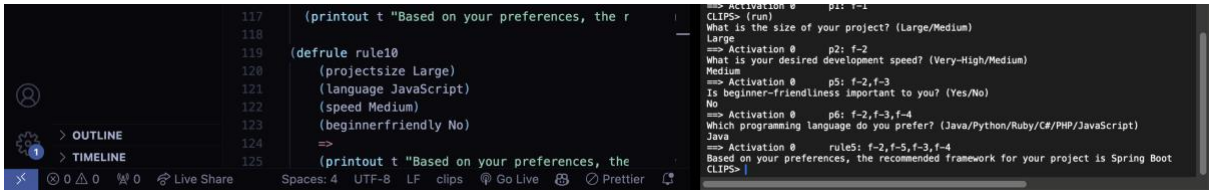


Figure 6 Implementation 4

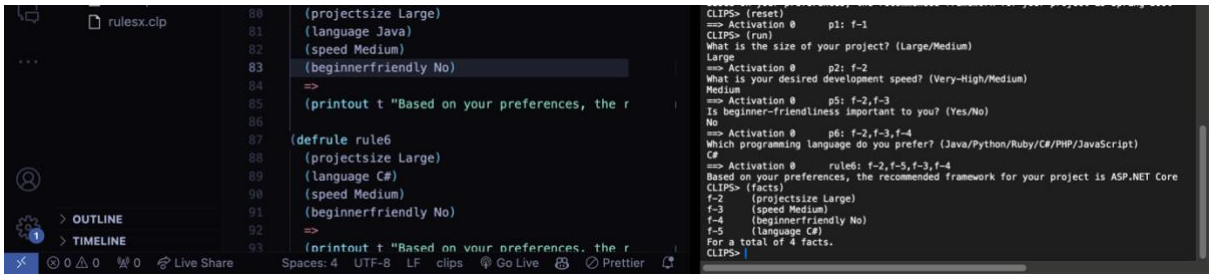


Figure 5 Implementation 5

4. Tools and Technologies

4.1. CLIPS

CLIPS (C Language Integrated Production System) is a rule-based programming language designed for building expert systems. It excels in knowledge representation and reasoning tasks. In this expert system, I utilized CLIPS to implement a backend technology recommendation system. The rules in the code evaluate user input regarding project size, programming language, development speed, and beginner-friendliness to make informed recommendations for suitable backend frameworks. CLIPS provides a robust framework for expressing logical relationships, making it an effective tool for developing expert systems like this one.

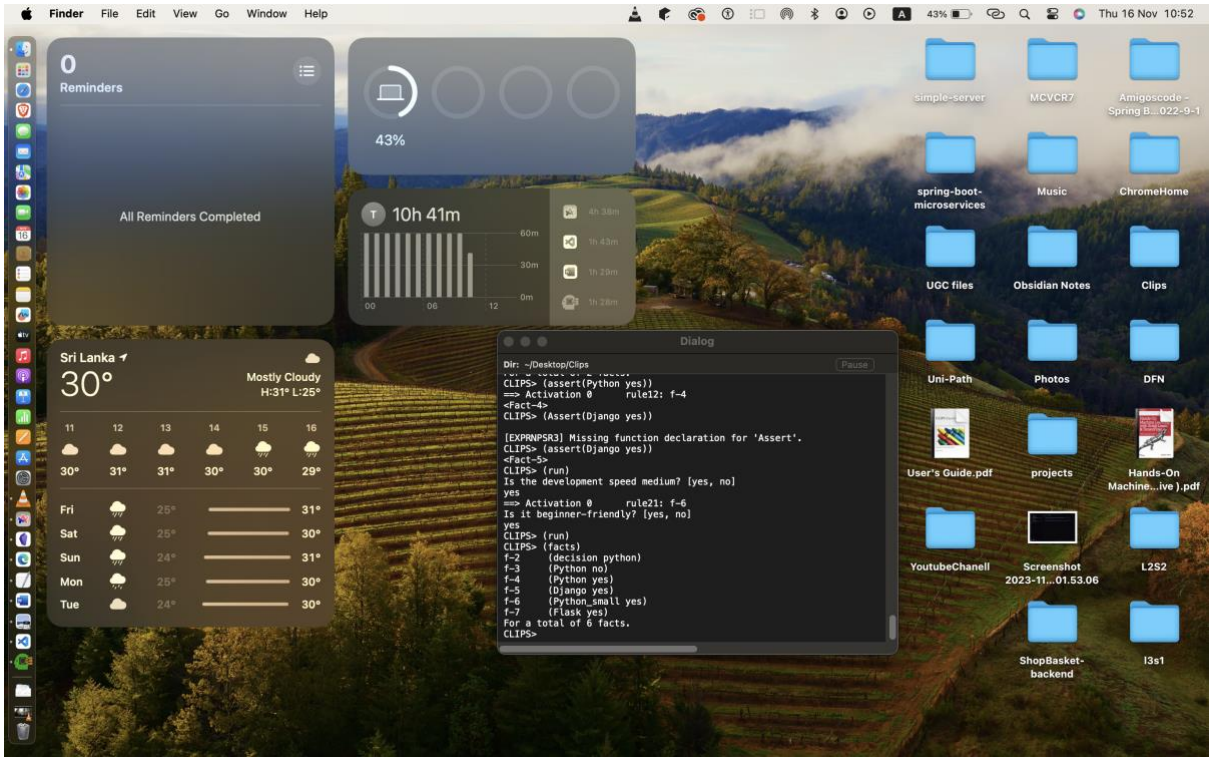


Figure 7 CLIPS Interface

4.2. VS-CODE

In this project, I utilized Visual Studio Code (VSCode) as my integrated development environment (IDE) for programming in CLIPS. VSCode provided a versatile and user-friendly platform for coding, with features such as syntax highlighting and code completion tailored to the CLIPS language. Notably, I leveraged the assistance of GitHub Copilot, an AI-powered code completion tool integrated into VSCode. Copilot contributed suggestions and generated code snippets based on the context of my programming tasks, enhancing the efficiency and productivity of the development process in CLIPS.

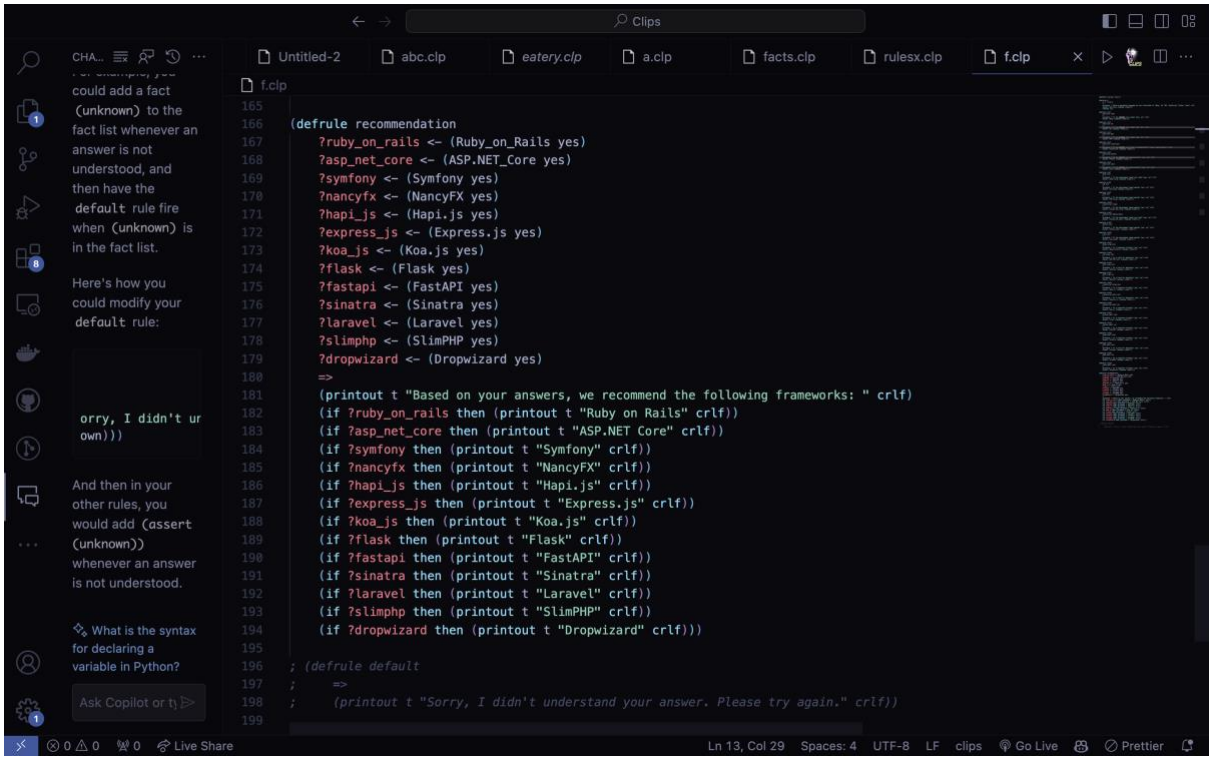


Figure 8 VS code Interface

5. Conclusion

In conclusion, FrameWizard, an intelligent system developed for backend framework selection, showcases the power of CLIPS in simplifying complex decision-making processes. The architecture, driven by a robust Knowledge Base and CLIPS' forward-chaining Inference Engine, ensures precision in recommendations. The CLI-based UI enhances user interaction, while the knowledge representation through rules and facts facilitates systematic evaluation. By addressing the identified problem of framework selection challenges, FrameWizard emerges as a valuable tool, streamlining decision-making for developers and contributing to the efficiency of software development.

6. References

[1] J. C. Giarratano, "CLIPS User's Guide," *Clipsrules.net*. [Online]. Available: https://clipsrules.net/documentation/v631/ug631.pdf. [Accessed: 16-Nov-2023].

[2] "Volume II advanced programming guide," *Sourceforge.io*. [Online]. Available: https://clipsrules.sourceforge.io/documentation/v640/apg.pdf. [Accessed: 16-Nov-2023].

[3] "CLIPS Rule Based Programming Language," *SourceForge*, 05-Feb-2019. [Online]. Available: https://sourceforge.net/projects/clipsrules/. [Accessed: 16-Nov-2023].

[4] A. Abraham, "130: Rule-based expert systems," *Handbook of Measuring System Design, edited by Peter H. Sydenham and Richard Thorn, John Wiley & Sons, Ltd. ISBN: 0-470-02143*, 2005.

[5] B. Lutkevich, "What is an expert system?," *Enterprise AI*, 18-Apr-2023. [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/expert-system. [Accessed: 16-Nov-2023].

[6] S. Osadchuk, "Top Backend Technologies For Your Project: pros & cons [2023]," *DOIT*, 22-Oct-2021. [Online]. Available: https://doit.software/blog/backend-technologies. [Accessed: 16-Nov-2023].

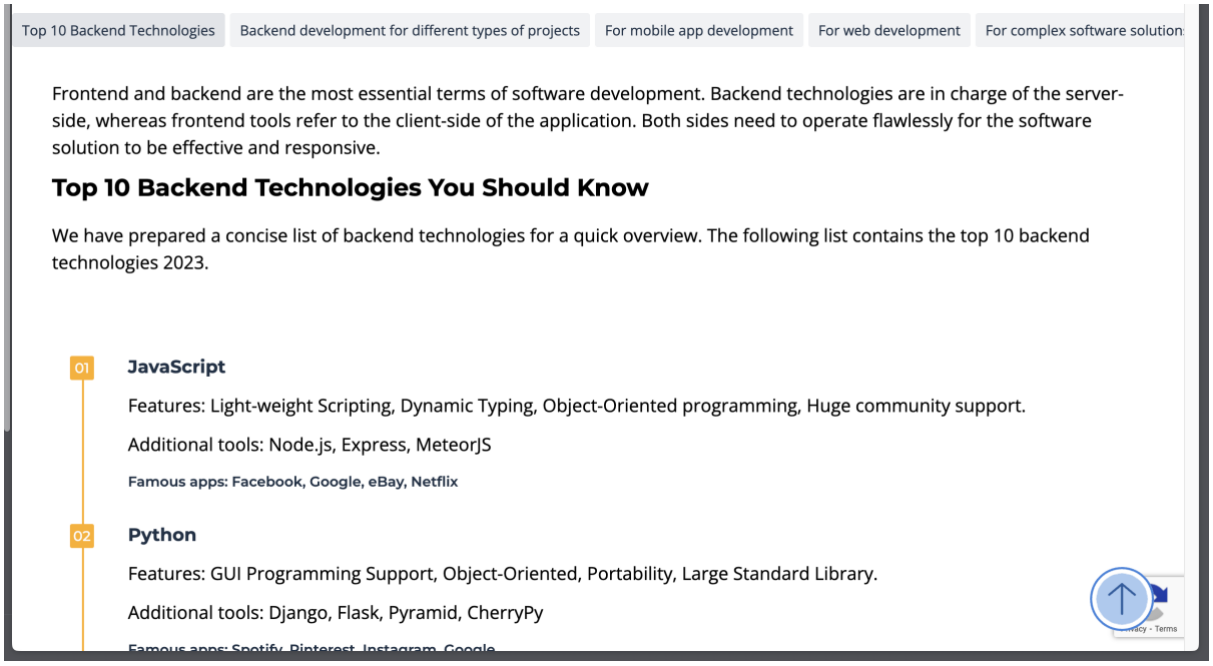
[7] A. A. Follow, "Top 10 backend technologies you must know," *GeeksforGeeks*, 23-Jun-2023. [Online]. Available: https://www.geeksforgeeks.org/backend-technologies/. [Accessed: 16-Nov-2023].

[8] J. Clark, "Top 10 backend technologies," *Back4App Blog*, 31-Aug-2022. [Online]. Available: https://blog.back4app.com/backend-technologies/. [Accessed: 16-Nov-2023].

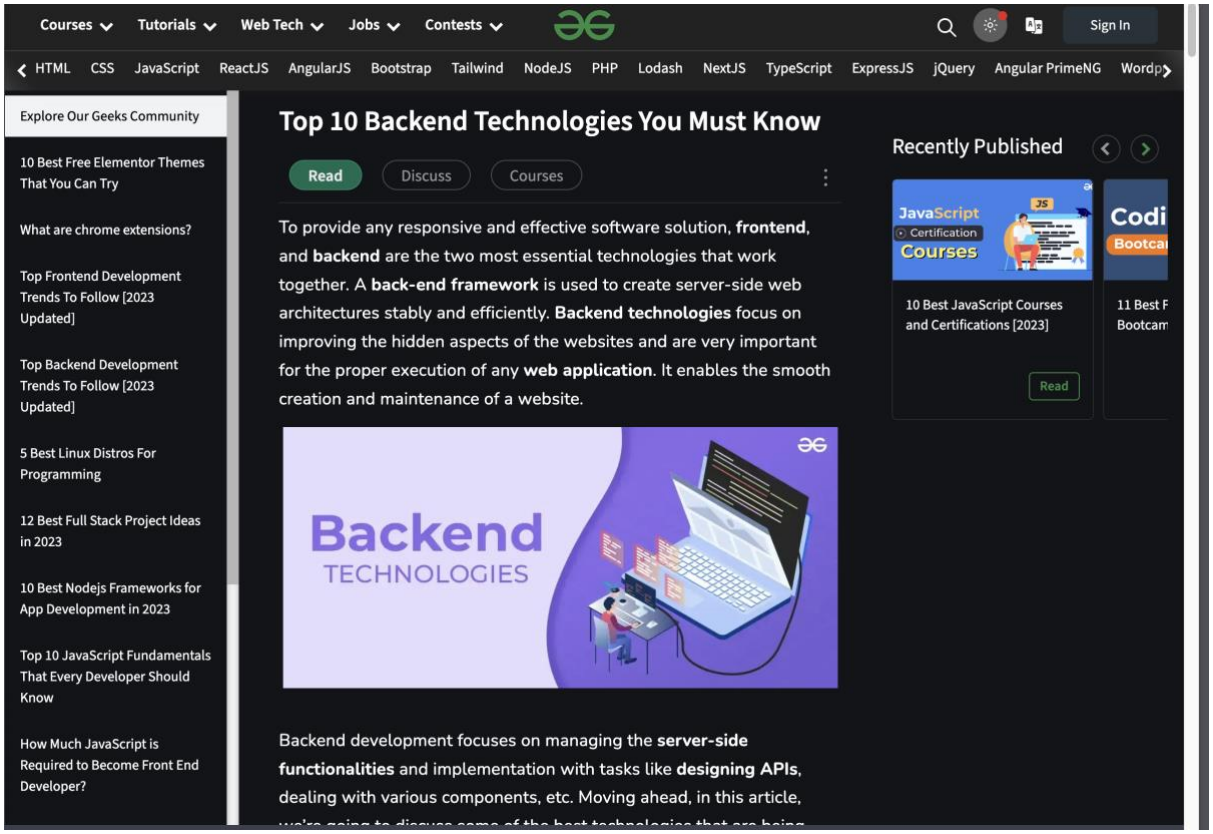
7. Appendices

GitHub Repository Link: - https://github.com/r33j4n/FrameWizard

Data Sources



https://doit.software/blog/backend-technologies.



https://www.geeksforgeeks.org/backend-technologies/