

Project Documentation: Rule Evaluation System

Github Project Link: <https://github.com/rahul420206/Rule-Engine-with-AST>

My Portfolio Link: rahul420206.github.io/Portfolio/


Introduction

This document provides an overview of the Rule Evaluation System, which evaluates user-defined rules based on input data. The system allows users to create, combine, and evaluate rules using a graphical web interface.

Features

- Create rules: Users can create rules using a simple syntax.

Rule Engine with AST

Create/Update Rule

((age > 30 AND department = 'Sales') OR (age < 25 AND department = 'Marketing')) AND (salary > 50000 OR experience > 5)

Save Rule

Existing Rules

Combine Rules

AND ▾

Combine Selected Rules

Evaluate a Rule

Age:

Existing Rules

1: ((age > 30 AND department = 'Sales') OR (age < 25 AND department = 'Marketing')) AND (salary > 50000 OR experience > 5)

Edit

Delete ☐

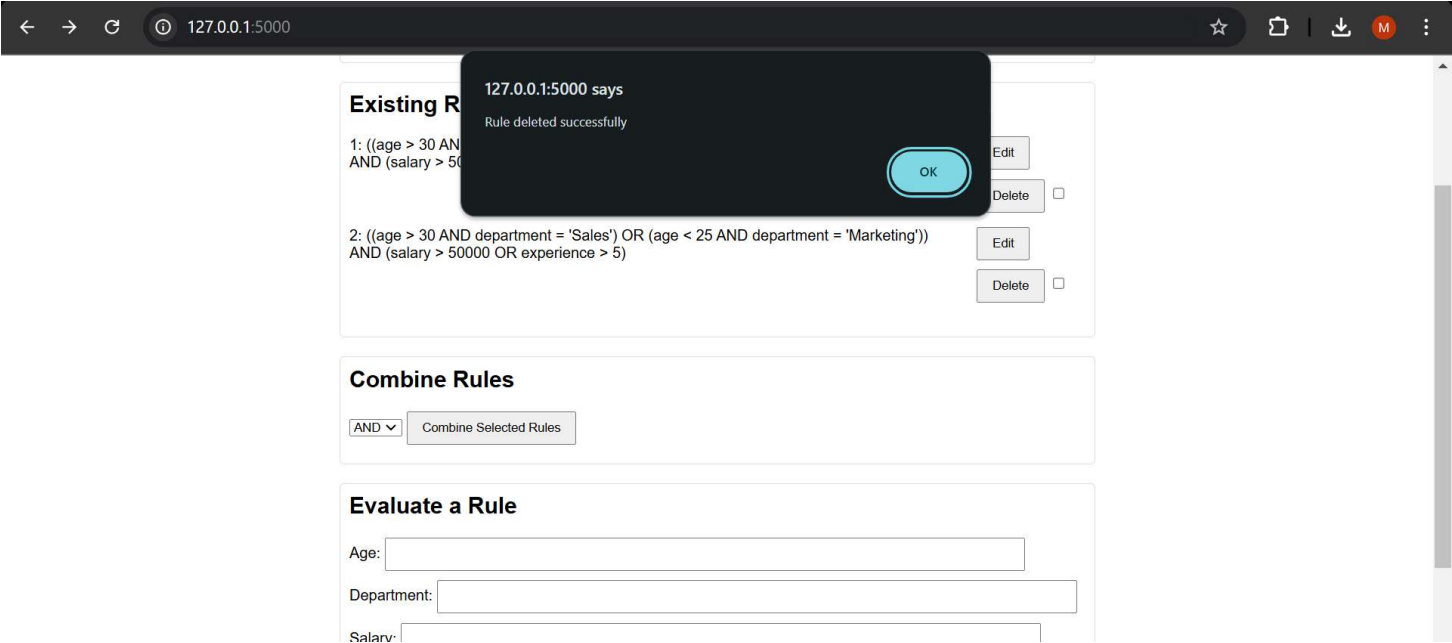
2: ((age > 30 AND department = 'Sales') OR (age < 25 AND department = 'Marketing')) AND (salary > 50000 OR experience > 5)

Edit

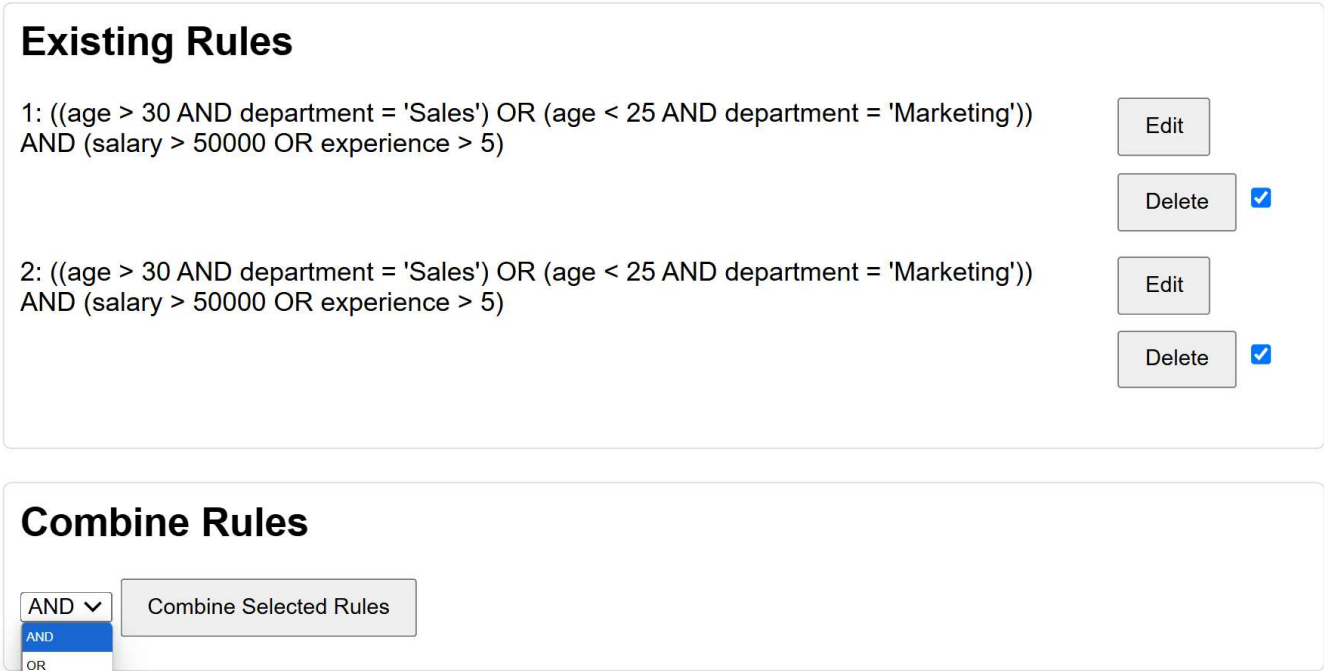
Delete ☐

- Manage stored rule: Edit or Select or delete

te the existing rules



- Combine rules: Multiple rules can be combined using logical operators AND or OR.



- Evaluate rules: The system evaluates user inputs against the defined rules.

Evaluate a Rule

Age:

Department:

Salary:

Experience:

Evaluate Rule

Evaluation Result:

No evaluation done yet.

- In-memory storage: Rules are stored in memory for quick access and manipulation.

Stored Inputs (Satisfied the Rule):

Age	Department	Salary	Experience
-----	------------	--------	------------

- User-friendly interface: The web interface allows easy rule management and evaluation.

Rule Engine with AST

Create/Update Rule

Enter rule (e.g., age > 18)

Save Rule

Existing Rules

2: ((age > 30 AND department = 'Sales') OR (age < 25 AND department = 'Marketing')) AND (salary > 50000 OR experience > 5)

Edit

Delete

Combine Rules

AND

Combine Selected Rules

Evaluate a Rule

Age:

Department:

Salary:

Experience:

Evaluate Rule

Evaluation Result:

No evaluation done yet.

Stored Inputs (Satisfied the Rule):

Age	Department	Salary	Experience
-----	------------	--------	------------

Code Overview

api.py

The main API of the application is defined in api.py. This file contains routes for creating, retrieving, deleting, and combining rules.

- The Node class represents the structure of the Abstract Syntax Tree (AST).
- The evaluate_condition function evaluates conditions based on user input data.
- The evaluate_ast function recursively evaluates the AST.

Node Class

class Node:

```
    def __init__(self, node_type, value=None, left=None, right=None):
```

```
self.node_type = node_type # 'operator' or 'operand'  
  
self.value = value # Operand value, like 'age > 30'  
  
self.left = left # Left child node  
  
self.right = right # Right child node
```

Rule Evaluation Functions

- evaluate_condition: Parses and evaluates conditions based on the input data.
- evaluate_ast: Recursively evaluates the AST against user input.

utils.py

The utils.py file contains utility functions for tokenizing rules and combining ASTs.

- The tokenize_rule function splits a rule into tokens for processing.
- The combine_ast function combines multiple ASTs into a single structure.

Conclusion

The Rule Evaluation System is a flexible and powerful tool for managing and evaluating business rules. It can be extended further by integrating additional functionalities such as persistence in a database or support for more complex rule definitions.