## **Automatic Code Generation in Prolog**

Objective: Create a Prolog program that generates code snippets in a target programming language (e.g., Python) based on high-level specifications or rules. This project involves defining a set of rules for generating code and translating those rules into executable code snippets.

**Project Overview**

1. Define the Target Language Syntax: Specify the syntax of the target programming language (e.g., Python). Represent common constructs such as variable declarations, loops, conditionals, and functions.
2. Create a Specification Language: Define a high-level specification language that describes the code to be generated. Examples of specifications might include variable assignments, loops, and conditionals in a simplified format.
3. Generate Code from Specifications: Implement a Prolog program that translates high-level specifications into code snippets in the target language.

**Example:**

assign(x, 10),

loop(i, 0, 5, [assign(sum, 'sum + i')]),

if\_else('x > 0', [assign(y, 'x - 1')], [assign(y, 'x + 1')])

% Expected output:

% x = 10

% for i in range(0, 5):

% sum = sum + i

% if x > 0:

% y = x - 1

% else:

% y = x + 1

**Advanced Tasks**

1. Support for Multiple Target Languages: Extend the program to support code generation for multiple target languages (e.g., Python, JavaScript).
2. Interactive Code Generation: Create an interactive interface for specifying high-level code and generating target language code.
3. Optimization and Refactoring: Implement optimization techniques to improve the generated code. Example: Combine multiple assignments into a single statement where possible.

**Submission**

* Submit the Prolog source file containing the target language syntax, specification language, and code generation functions.
* Include a README file with documentation and explanations.
* Provide a report with test cases and their outcomes.
* Make and submit a presentation to explain the solution.