Graphical user interface, text, application, email

Description automatically generated

**Report =>**

**Background:**

The logical programming language known as the prolog, also known as the prolog family tree, can process symbolic and non-numeric data. It works particularly effectively for solving issues that call for the use of objects and interactions between those items.

Facts and regulations can be used to express these connections.

There are many different sorts of relationships, some of which can also be laws. Even if a relationship is not stated clearly as a fact, a rule can learn about it.

**Problem Statement:**

Constructing a family tree using prolog or any language.

**Input Data:**

Names of the family members having relation.

**Method:**

* By converting facts and regulations into a family tree, we can then perform certain queries on the data.
* By listing the n-tuples of objects based on the information provided in the family tree, we have defined the parent relation.
* The user can easily ask the Prolog system questions regarding the program's defined relations.
* A Prolog program is made up of clauses that are finished with full stops.
* Among other things, the arguments of relations can be specific objects, constants (like pat and jim), or basic objects like X and Y. In our program, the first kind of objects are referred to as atoms. Variables are the name for the second type of objects.
* One or more goals are contained in questions to the system.

**Result:**

Presented the query, the program will identify the relation based on the query and displays name of the family member. If it doesn't exist, then it returns false.

**Code:**

As present in .pl file attached.

**Note:**

The program has been executed in the SWI Prolog Online Editor using Prolog language. Separate files are attached. For simple view screenshot can be opened to view the output and code.