NAME - GAURAV KUMAR

Reg no. – 25BSA10169

COURSE – FUNDAMENTALS OF AI AND ML

Project Report: Student Grading and Eligibility System (Prolog)

1. Introduction

This report documents the design and implementation of a logic-based system for managing student performance records using the Prolog programming language. The primary goal of this project is to establish a knowledge base that can automatically process student marks and apply predefined academic rules to determine outcomes such as Pass/Fail status, letter grades, and eligibility for academic scholarships.

The use of Prolog is ideal for this application due to its declarative nature, allowing rules to be defined explicitly based on logical conditions rather than procedural steps.

2. Project Objectives

The system was developed to meet the following specific objectives:

1. **Data Storage:** Create a robust and easily queryable database of student names and their corresponding raw marks.
2. **Result Determination:** Implement rules to clearly distinguish between passing and failing students.
3. **Grading:** Establish a multi-tiered grading structure (S, A, B, C, D, F) based on defined mark ranges.
4. **Eligibility Screening:** Implement a rule to quickly identify students who qualify for a scholarship based on merit.
5. **Query Interface:** Allow users (or other modules) to query the system for any student's status, grade, or eligibility.

3. Knowledge Base Structure

The system is composed of two primary sections: Facts and Rules.

3.1. Facts (Data)

Student data is stored using the predicate student(Name, Marks).

Example Facts:

Name	Marks
aditya	92
aniket	85
rohit	67
kiran	43

3.2. Rules (Logic)

The rules define the system's intelligence and are grouped into three main logical predicates:

A. Pass/Fail Criteria (result/2)

Determines if a student has achieved a passing mark.

Rule	Condition	Prolog Implementation
Pass	Marks >= 50	result(Name, pass) :- student(Name, Marks), Marks >= 50.
Fail	Marks < 50	result(Name, fail) :- student(Name, Marks), Marks < 50.

B. Grading Rules (grade/2)

Assigns a letter grade based on a sequential range check. Note that the rules are structured hierarchically to ensure correct grade assignment.

Grade	Marks Range	Prolog Implementation (Simplified)
S	Marks >= 90	grade(Name, 'S') :- student(Name, Marks), Marks >= 90.
A	80 <= Marks < 90	grade(Name, 'A') :- student(Name, Marks), Marks >= 80, Marks < 90.
B	70 <= Marks < 80	grade(Name, 'B') :- student(Name, Marks), Marks >= 70, Marks < 80.
C	60 <= Marks < 70	grade(Name, 'C') :- student(Name, Marks), Marks >= 60, Marks < 70.
D	50 <= Marks < 60	grade(Name, 'D') :- student(Name, Marks), Marks >= 50, Marks < 60.
F	Marks < 50	grade(Name, 'F') :- student(Name, Marks), Marks < 50.

C. Scholarship Eligibility (scholarship/2)

Checks if a student qualifies for the scholarship merit threshold.

Eligibility	Condition	Prolog Implementation
Eligible	Marks >= 85	scholarship(Name, eligible) :- student(Name, Marks), Marks >= 85.
Not Eligible	Marks < 85	scholarship(Name, not_eligible) :-

		student(Name, Marks), Marks < 85.

4. Conclusion and Future Scope

This Prolog implementation successfully demonstrates the power of declarative logic programming for automating academic record analysis. By separating the data (Facts) from the decision-making processes (Rules), the system remains highly flexible; for instance, grading thresholds or scholarship criteria can be updated without altering the core student data.

Future Scope includes:

- Adding predicates for calculating class averages and ranking students.
- Integrating external file input/output (e.g., reading student data from a CSV) for easier data management.
- Implementing conditional rules based on the student's grade (e.g., "A student with an 'S' grade is automatically placed on the Dean's List").