UNIVERSITY OF TECHNOLOGY JAMAICA

Assignment: Path-Finder for Rural Roads Network in Python/Prolog

Group Size: 3-4

Given: Week of September 29, 2025 **Due: Week of November 17, 2025**

Objective

Develop a Python program that models a Jamaican rural road network and implements a path-finding algorithm to determine the **shortest or most efficient route** between two locations (villages, towns, or landmarks). The system should allow interactions with users and administrators. The Python program will interact with a Prolog file. Python will be used as the interface and Prolog will be used to store the network data and execute the Path-Finding Algorithm.

Scenario

You are tasked with helping a rural municipality to digitize its road network to improve connectivity and emergency response. The roads may be **unpaved**, **one-way**, or **seasonally blocked**. Your system should be able to:

- Represent the road network
- Using AI, the system must facilitate dynamic updates (e.g., **blocked roads**, **broken cisterns, potholes that are very deep).**
- Find the shortest or best route based on user-defined criteria

Requirements

1. Model the Road Network

- Represent the network as a **graph**: nodes are villages/towns, edges are roads.
- Each road (edge) should have:
    - Distance (in km)
    - Road type (paved/unpaved/broken cisterns/deep potholes)
    - Travel time (optional)
    - Status (open/closed)

2. Input Format

The system can be initialized with hardcoded data but should allow an administrator to add/update data via user input. The network data should be stored in a Prolog file.

Sample network *Input*:

Source	Destination	Distance_Km	Type	Status
A	B	5	paved	open
B	C	3	unpaved	open
C	D	7	paved	closed

3. Path-Finding Algorithm

Implement the below algorithms. The appropriate algorithm should be executed in Prolog based on the criteria selected.

- **Dijkstra's algorithm** (for shortest distance)
- **A* algorithm** (if you want to include heuristics like terrain difficulty)
- **BFS/DFS** (if constraints are loose)

Allow user to choose the criteria:

- Shortest distance
- Fastest route (if time is available)
- Avoid unpaved roads
- Avoid closed roads
- Avoid roads with broken cisterns
- Avoid roads with deep potholes

4. User Interface

- Allow the user to:
    - Input source and destination
    - Choose path criteria
    - Display the resulting path and total distance/time

5. Output

Show:

- List of places on the path
- Total distance
- Total estimated travel time (if applicable)
- Notification if no path is available

There will be a demonstration of the system which is scheduled for the Week of November 17, 2025.

Required:

Students should work in groups of no more than four (4). The group should present the completed program along with commented code and project documentation.

You are required to submit the complete source code, and documentation. The documentation should include the following: System Design, User Manual and Project Group Report (highlighting the contribution of each member and lessons learnt) by November 22, 2025.

Marking Scheme:

Marks will be awarded as follows:

- Facts (10%)

- Rules (15%)

- Persistence (10%)

- Functionality (30%)

    o Proper use facts and rules [5%]

    o Proper use of prolog search mechanisms [5%]

    o Storing of User Response [5%]

    o Use of User Response in arriving at conclusion / solution [5%]

    o Robustness [5%]

    o Correctness [5%]

- Documentation (15%)
    - o Internal : comments, indentation and naming conventions
    - o External: hardcopy – formatting, neatness, sample run, group report and declaration of authorship (one per member)
- User Interface & Ease of Use (10%)
- Originality & Ingenuity (10%)

There will be an additional 10% for groups who provide a graphical user interface (GUI) for this system. Marks will be subtracted for late assignments at a rate of 7.5% per day. Assignments more than one week late will not be accepted.