**Mini Expert System Report**

**Course:** CCST-101-3A  
**Activity:** Expert System Rules Implementation  
**Student:** Harley Yuan B.Quinay  
**Date:** September 05,2025

**1. Introduction**

An expert system is a computer program that uses logical rules to mimic human decision-making. For this activity, I implemented a **Mini Expert System in Python** that evaluates student data according to a set of predefined rules. The program was tested with three different students, and the results were automatically logged into a CSV file for record-keeping.

The goal of this task is to:

1. Understand how expert systems process conditions using rules.
2. Test different scenarios for multiple students.
3. Save all outputs in a structured CSV file.
4. Extend the system by adding one new custom rule.

**2. Rules Tested**

The program applied five rules. Each rule evaluates a condition and produces an output:

1. **Attendance Rule**
   * Checks if a student’s attendance is **≥ 75%**.
   * If true → *Attendance OK*.
   * If false → *Attendance Low*.
2. **Grading Rule**
   * Checks if a student’s grade is **≥ 75**.
   * If true → *Pass*.
   * If false → *Fail*.
3. **Login System Rule**
   * Verifies a student’s username and password against stored credentials.
   * If matched → *Login Successful*.
   * If incorrect → *Login Failed*.
4. **Bonus Points Rule**
   * Checks if the student has completed an extra activity.
   * If true → *Bonus Points Awarded*.
   * If false → *No Bonus*.
5. **New Rule: Library Borrowing**
   * Checks if a student has a valid ID.
   * If true → *Allowed to Borrow*.
   * If false → *Not Allowed*.

**3. Testing Procedure**

The program was tested with **three students (Alice, Bob, and Charlie)** who had different conditions:

* **Alice:** High attendance, passing grade, correct login, did the activity, valid ID.
* **Bob:** Low attendance, failing grade, wrong password, no activity, invalid ID.
* **Charlie:** Sufficient attendance, borderline grade, wrong password, did the activity, valid ID.

Each student went through all five rules, and the results were logged in **logic\_results.csv**.

**4. Sample Program Execution**

When the program was run, it printed the results for each student in the console, for example:

--- Results for Alice ---

Attendance OK

Pass

Login Successful

Bonus Points Awarded

Allowed to Borrow

All results were also recorded in the CSV file in the format:

Student,Rule,Condition,Result

Alice,Attendance Rule,18/20,Attendance OK

Alice,Grading Rule,Score = 85,Pass

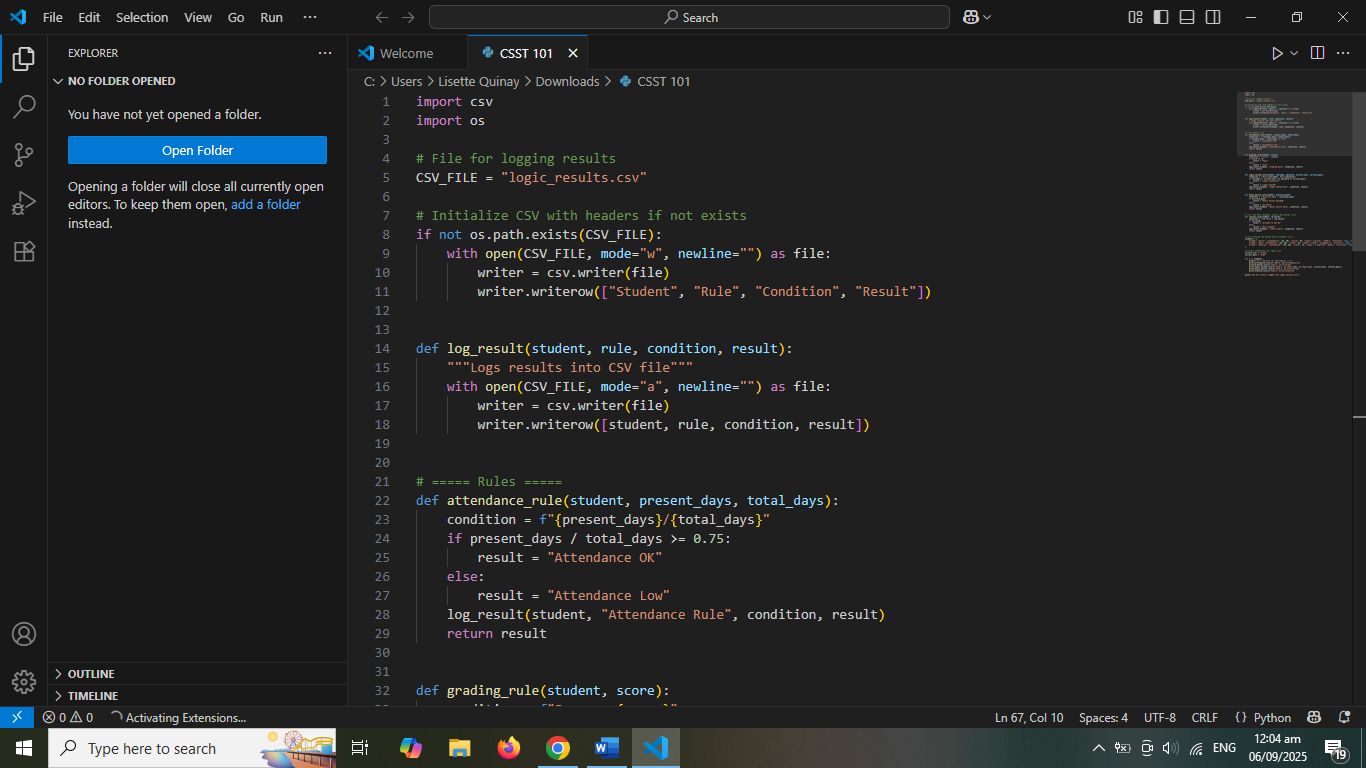
Alice,Login System Rule,User=alice, Pass=1234,Login Successful

Alice,Bonus Points Rule,Activity Done = True,Bonus Points Awarded

Alice,Library Rule,ID Valid = True,Allowed to Borrow

...

**5. Screenshots**



**6. Results and Discussion**

* The **attendance rule** correctly identified students who met the 75% requirement.
* The **grading rule** showed clear pass/fail outcomes based on scores.
* The **login system rule** confirmed that only students with the correct username and password could access the system.
* The **bonus points rule** provided incentives for students who completed additional activities.
* The **library borrowing rule** demonstrated how easily the system can be extended to handle new conditions.

This proves that the Mini Expert System is flexible and reliable in evaluating logical conditions and recording outcomes.

**7. Conclusion**

The Mini Expert System successfully implemented multiple rules and recorded all outcomes in a structured CSV file. By extending the system with a new rule (**Library Borrowing**), it was shown that the framework can be easily adapted to other scenarios, such as enrollment clearance or laboratory access.

This activity helped me understand how **rule-based expert systems** work and how they can be applied to real-world problems like student management, authentication, and resource access.