Pusat
Sains Matematik

CENTRE OF MATHEMATICAL SCIENCES
UNIVERSITI MALAYSIA PAHANG AL-SULTAN ABDULLAH

BSD2513 ARTIFICIAL INTELLIGENCE
SEM I 2025/2026

LAB REPORT 3

TITLE	Chapter 3: Knowledge Representation System
LECTURER'S NAME	Dr. Ku Muhammad Na'im Ku Khalif
NAME	Sara Khadija binti Saidin
ID NUMBER	SD23061
SUBMISSION DATE	30/11/2025

Question 1

General Knowledge

Discuss three applications of rule-based systems in real-world phenomena. Give references.

(5 Marks)
(CO1 PO1)

1. **Clinical Decision Support and Medical Diagnosis**

   Rule-based systems continue to play a significant role in assisting medical professionals with diagnosis and decision-making by encoding medical expertise as explicit "if–then" logic. Modern expert systems help process patient symptoms, suggest diagnoses, and provide explanations that support clinician reasoning. For example, enhanced rule-based and case-based reasoning systems have been developed and evaluated for disease diagnosis, showing competitive performance compared to other methods and helping to maintain interpretability in high-stakes clinical contexts.

2. **Decision Support in Administrative and Security Contexts**

   Beyond healthcare, rule-based systems are used as decision support systems (DSS) in administrative, organizational, and security fields. They structure complex knowledge into condition/action rules that guide administrative decisions and help detect vulnerabilities or threats. A recent literature review highlights how modern rule-based DSS are applied to manage decisions, evaluate risks, and address cybersecurity considerations across domains.

3. **Industrial and Autonomous Monitoring**

   In industry and automation, rule-based logic provides reliable monitoring and control by encoding safety thresholds, operational constraints, and expert knowledge for real-time decision-making. Even as machine

learning shifts some workloads toward data-driven techniques, rule-based systems remain valuable for predictable, interpretable monitoring in regulated or safety-critical environments. Recent comparative studies emphasize their continued utility, especially when combined in hybrid frameworks with data-driven methods.

References

Mustafa, E. M., Saad, M. M., & Rizkallah, L. W. (2023). Building an enhanced case-based reasoning and rule-based systems for medical diagnosis. *Journal of Engineering and Applied Science*, *70*(1). https://doi.org/10.1186/s44147-023-00315-4

Savvadelli, E., Kiouvrekis, Y., & Kokkinaki, A. (2025). A Literature Review on Rule-Based Systems as Decision Support Systems. *IFIP Advances in Information and Communication Technology*, 376–388. https://doi.org/10.1007/978-3-032-02504-3_26

Giovanni, D. G., & Facchini, S. D. (2025). *A Comparative Study of Rule-Based and Data-Driven Approaches in Industrial Monitoring*. ArXiv.org. https://arxiv.org/abs/2509.15848

Question 2

Python: Knowledge Representation System (Rule-Based System)

Generate and implement a rule-based system for scholarship advisory to assist the university in deciding the eligibility and type of scholarship to be awarded to applicants. You are required to consider using EXACTLY these into their JSON editor in the app.

(5 Marks, CO2PO2)
(15 Marks, CO3PO3)

Coding:

```
import streamlit as st

# ----------------------------
# RULES (unchanged)
# ----------------------------
RULES_JSON = [
    {
        "name": "Top merit candidate",
        "priority": 100,
        "conditions": [
            ["cgpa", ">=", 3.7],
            ["co_curricular_score", ">=", 80],
            ["family_income", "<=", 8000],
            ["disciplinary_actions", "==", 0]
        ],
        "action": {
            "decision": "AWARD_FULL",
            "reason": "Excellent academic & co-curricular performance,
with acceptable need"
        }
    },
    {
        "name": "Good candidate - partial scholarship",
        "priority": 80,
        "conditions": [
            ["cgpa", ">=", 3.3],
            ["co_curricular_score", ">=", 60],
```

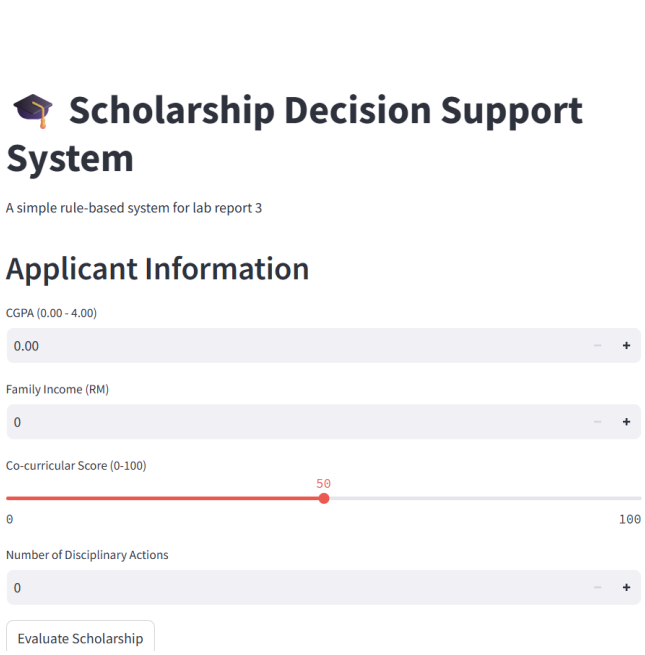
```
            ["family_income", "<=", 12000],
            ["disciplinary_actions", "<=", 1]
        ],
        "action": {
            "decision": "AWARD_PARTIAL",
            "reason": "Good academic & involvement record with moderate
need"
        }
    },
    {
        "name": "Need-based review",
        "priority": 70,
        "conditions": [
            ["cgpa", ">=", 2.5],
            ["family_income", "<=", 4000]
        ],
        "action": {
            "decision": "REVIEW",
            "reason": "High need but borderline academic score"
        }
    },
    {
        "name": "Low CGPA - not eligible",
        "priority": 95,
        "conditions": [
            ["cgpa", "<", 2.5]
        ],
        "action": {
            "decision": "REJECT",
            "reason": "CGPA below minimum scholarship requirement"
        }
    },
    {
        "name": "Serious disciplinary record",
        "priority": 90,
        "conditions": [
            ["disciplinary_actions", ">=", 2]
        ],
        "action": {
            "decision": "REJECT",
```

```
            "reason": "Too many disciplinary records"
        }
    }
]


# ----------------------------
# EVALUATOR
# ----------------------------
def evaluate_rule(rule, facts):
    for cond in rule["conditions"]:
        field, operator, value = cond
        fact_value = facts.get(field)

        # if a fact is missing, treat as non-matching
        if fact_value is None:
            return False

        if operator == ">=" and not (fact_value >= value):
            return False
        if operator == "<=" and not (fact_value <= value):
            return False
        if operator == ">" and not (fact_value > value):
            return False
        if operator == "<" and not (fact_value < value):
            return False
        if operator == "==" and not (fact_value == value):
            return False
        if operator == "!=" and not (fact_value != value):
            return False

    return True


def evaluate_scholarship(facts):
    matched = []
    for rule in RULES_JSON:
        if evaluate_rule(rule, facts):
            matched.append(rule)
```

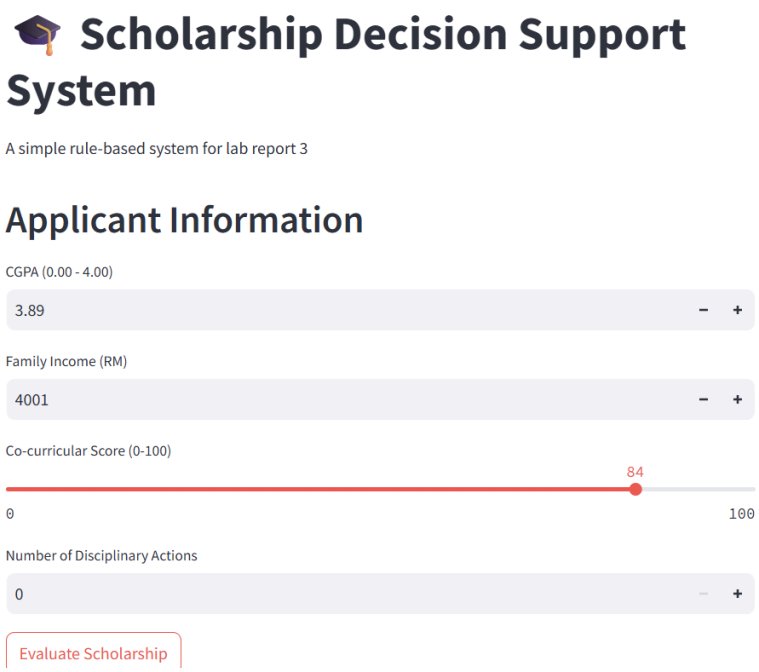
```
    if not matched:
        return {"decision": "NO_MATCH", "reason": "No rules matched."}

    best_rule = max(matched, key=lambda r: r["priority"])
    return {
        "rule_name": best_rule["name"],
        "decision": best_rule["action"]["decision"],
        "reason": best_rule["action"]["reason"],
        "priority": best_rule["priority"]
    }


# ------------------------------
# STREAMLIT UI
# ------------------------------
st.set_page_config(page_title="Scholarship Decision Support",
layout="centered")
st.title("🎓 Scholarship Decision Support System")
st.write("A simple rule-based system for lab report 3")

st.header("Applicant Information")

# CGPA: float
cgpa = st.number_input("CGPA (0.00 - 4.00)", min_value=0.0, max_value=4.0,
step=0.01, format="%.2f")

# Family income: integer (RM)
family_income = st.number_input("Family Income (RM)", min_value=0, step=1,
format="%d")

# Co-curricular score: integer slider 0-100
co_curricular = st.slider("Co-curricular Score (0-100)", 0, 100, 50)

# Disciplinary actions: integer
disciplinary = st.number_input("Number of Disciplinary Actions",
min_value=0, max_value=10, step=1, format="%d")

if st.button("Evaluate Scholarship"):
    # Ensure types: cgpa float, others ints
    student_facts = {
```

```
        "cgpa": float(cgpa),
        "family_income": int(family_income),
        "co_curricular_score": int(co_curricular),
        "disciplinary_actions": int(disciplinary)
    }

    result = evaluate_scholarship(student_facts)

    st.subheader("Decision Result")
    st.write(f"**Rule Triggered:** {result.get('rule_name', 'None')}")
    st.write(f"**Decision:** {result['decision']}")
    st.write(f"**Reason:** {result['reason']}")
    st.write(f"**Priority:** {result.get('priority', '-')}")
```

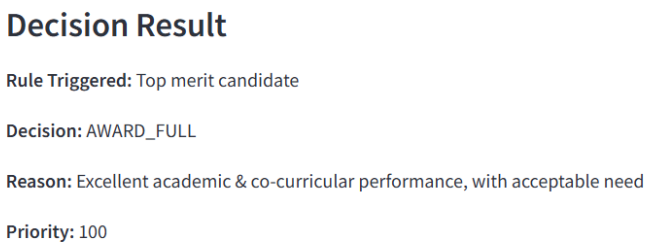
Output:



This is the output before the user input any information. Users need to insert the CGPA, family income in RM, co-curriculum score, and the number of disciplinary actions taken by the user.

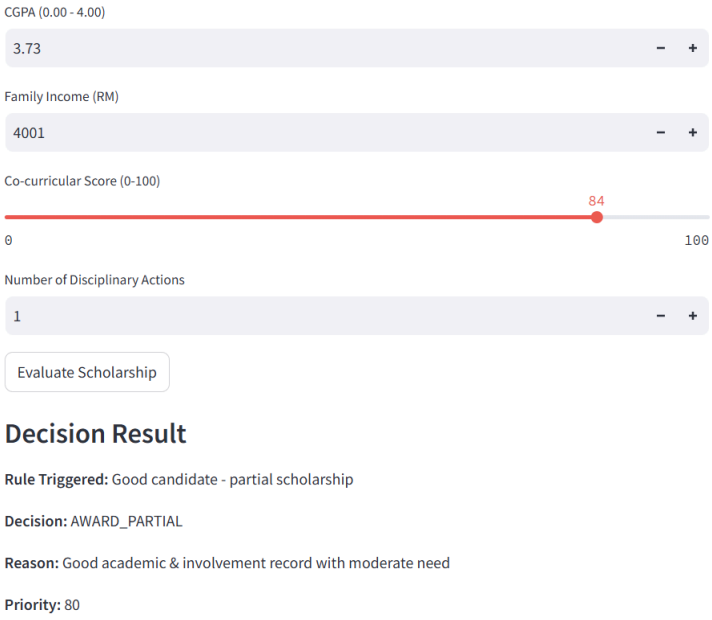


As example, when user input CGPA of 3.89, family income as much as RM4001, co-curriculum score of 84, and no disciplinary actions, the output is like this:

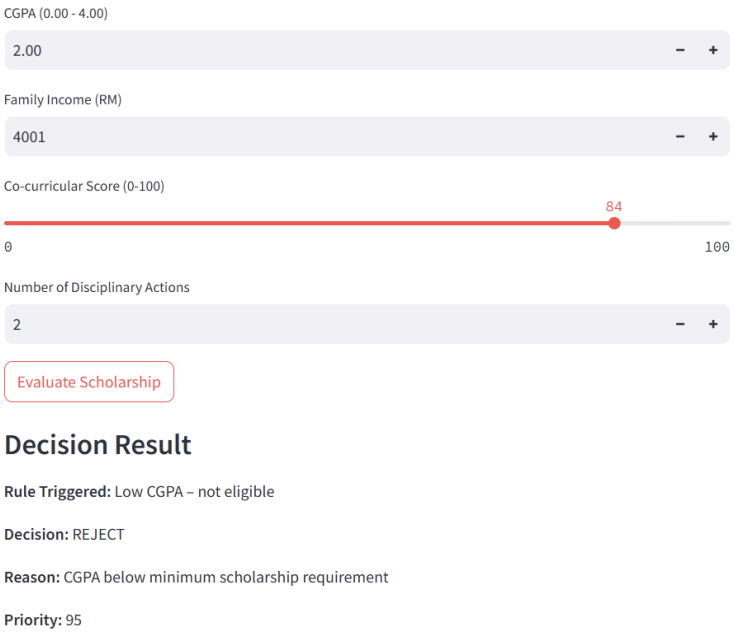


The user is a top merit student, so he/ she can be rewarded with a full award with the above reason.

Basically, all the outcome depends on the student's input user. Other examples:



This is when the user is partially awarded with priority of 80%.



The worsen situation is when the user gets rejected due to the CGPA being below the scholarship requirement.

GitHub: https://github.com/SaraKhadija/sd23061_lab3
Streamlit: https://labreportailab3.streamlit.app/