	SUBJECT: BSD2513 ARTIFICIAL INTELLIGENCE	**MARKS**: 25(5%)
	TOPIC: Chapter 3: Knowledge Representation System	
	LAB REPORT 3	

CLO	Description	PLO Mapping	Percentage	Marks
CLO1	Acquire the artificial intelligence concepts and methodologies in data science.	PLO1: Knowledge and Understanding C3: Application	1%	5
CLO2	Demonstrate critical thinking ideas of artificial intelligence knowledge in problem-solving situation.	PLO2: Cognitive Skills and Functional work skills with focus on Numeracy skills CLO3: Application	1%	5
CLO3	Develop an artificial intelligence system prototype using appropriate software.	PLO3: Functional work skills with focus on Practical, and Digital skills P4: Mechanism	3%	15

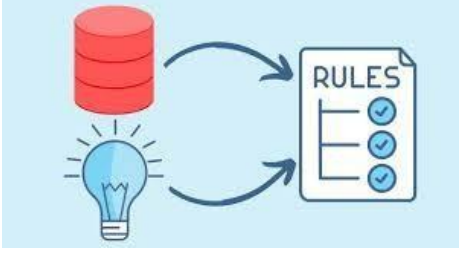
Laboratory Report Objectives

By the end of this lab, students should be able to:

1. articulate AI capability types and their relevance to real deployments;
2. explain how rule-based systems represent knowledge using rules, conditions, and actions; and
3. implement and deploy a minimal, reliable web app (Streamlit) that uses a rule engine to evaluate cases and supports different rule configurations.

CASE STUDY:

A university wants to support its scholarship committee with a transparent decision support tool. Instead of purely manual screening, they decide to use a rule-based system that can be reviewed and adjusted by the academic and financial aid committees.



Each scholarship applicant is described by several facts, such as:

a. Cumulative GPA (CGPA)
b. Monthly family income
c. Co-curricular involvement score (0–100)
d. Community service hours
e. Current semester of study
f. Number of disciplinary actions on record

Question 1
General Knowledge

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

(5 Marks)
(CO1 PO1)

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 use EXACTLY these into their JSON editor in the app.

```
[
  {
    "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"
    }
  }
]
```

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

Save your work in both .py and PDF formats. Name your files using the following format: StudentID_LabX. Submit both files through the Kalam platform by 26th November 2025, 11:59 PM. In addition, deploy your Streamlit application and include the public URL to your deployed app and GitHub repository link inside your report. Late submissions will only be considered with prior approval.

Pusat Sains Matematik	**SUBJECT: BSD3513** **INTRODUCTOION TO ARTIFICIAL INTELLIGENCE**	**MARKS**: 25(5%)
	TOPIC: Chapter 3: Knowledge Representation System	
	LAB REPORT 3	

NAME: ZAHIN ZIKRI BIN ZAWAWI **STUDENT ID: SD23019** **SECTION:02G**

Mark for CO1: /5

Rubric for CO2.

Instruction: For CO2, assess each item using the given scales.

Demonstrate critical thinking ideas of artificial intelligence knowledge in problem-solving situation.								
Item Assessed (Cognitive)	**Very Poor 0**	**Poor 1**	**Fair 2**	**Good 3**	**Very Good 4**	**Excellent 5**	**Weigh tage**	**Score**
Apply and analyse relevant artificial intelligence knowledge.	The work has not done.	Poorly applied and analysed relevant artificial intelligence knowledge and results.	Applied and analysed relevant artificial intelligence knowledge but failed to achieve successful results.	Applied and analysed relevant artificial intelligence knowledge but arrive at satisfactory results.	Applied and analysed relevant artificial intelligence knowledge to arrive at successful results.	Applied and analysed relevant artificial intelligence knowledge to arrive at excellent results.	0.5	
Using logical, rational or problem-solving appropriate to the artificial intelligence problems.	The work has not done.	The work needs to demonstrate logical, rational or problem-solving understanding appropriate to the artificial intelligence problems.	The work has demonstrated some logical, rational or problem-solving understanding appropriate to the artificial intelligence problems.	The work has demonstrated logical, rational or problem-solving understanding appropriate to artificial intelligence problems.	The work has demonstrated a thorough, logical, rational or problem-solving understanding appropriate to artificial intelligence problems.	The work has demonstrated a thorough and classy logical, rational or problem-solving understanding appropriate to artificial intelligence problems.	0.5	
						Total Score	**1**	**/5**

Rubric for CO3.

Instruction: For CO3, assess each item using the given scales.

CO3: Develop an artificial intelligence system prototype using appropriate software.							
Item Assessed (Cognitive)	**Very Poor 0**	**Poor 1**	**Fair 2**	**Good 3**	**Very Good 4**	**Excellent 5**	**Score**
Utilizing the appropriate tools / software effectively	No relevant tool used.	Tools used but did not enhance solution or information clarity.	Tools used but with limited enhancement; minimal functionality demonstrated.	Tools used appropriately to produce a functional solution with clear output.	Tools used effectively to enhance clarity, performance, and solution quality.	Tools used optimally with advanced features, clear design, and effective interaction to display the solution.	
Code functionality, clarity & structure	No code constructed.	Code incomplete or mostly non-functional; unclear and poorly structured.	Partially functional code; errors present; structure somewhat difficult to follow.	Mostly functional code with minor errors; clear structure and readable.	Fully functional and well-structured code; clearly commented and readable.	Fully functional, optimized, modular, and well-documented code; demonstrates best practices.	
Deployment & Version Control (GitHub + Streamlit or etc)	No deployment and no GitHub repository.	GitHub repo exists but incomplete OR app deploy attempt failed.	GitHub repo available with basic files; deployment page exists but app not functioning correctly.	Working deployment provided; GitHub repo contains main code files.	Working deployment with complete repository (README, code, requirements); clearly accessible.	Fully deployed app with professional GitHub repo (README, screenshots, instructions, modules, tags); live Streamlit app runs smoothly and reliably.	
						Total Score	**/15**

Question 1: General Knowledge

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

1. Medical Diagnosis and Clinical Decision Support

Rule-based systems are critical in healthcare for assisting doctors with complex decision-making. These systems function by applying strict expert logic to patient data to ensure safety and accuracy. For example, Friedman and Frank (1983) demonstrated that rule-based structures are ideal for oncology practice because they offer modularity of structure and explainability, which are essential for patient care. In modern applications, these systems have evolved to be highly precise; Aloufi et al. (2024) reported that in dental specialties, hybrid rule-based systems combined with neural networks achieved up to 96% accuracy in diagnosis and treatment planning.

2. Automated Claims Processing (Business Operations)

In the financial and administrative sectors, rule-based engines are used to automate high-volume transactions that require strict adherence to policy. Sawar et al. (2010) discuss the application of these systems in medical billing, where a rule engine is integrated with software to identify billing errors in medical claims at real-time. This application is significant because it allows organizations to handle a large number of knowledge-oriented checks instantly, ensuring that every claim complies with financial regulations before it is processed.

3. Cybersecurity and Intrusion Detection

Rule-based systems are extensively deployed in cybersecurity to protect digital infrastructure against attacks. According to Vassilev (2024), these systems are used for "unauthorized intrusion detection, malicious interference protection, and digital forensics." Unlike probabilistic AI, rule-based approaches in security allow for precise control, executing multiple operations in a single transaction based on heuristic rules. This ensures that specific threats trigger immediate, defined defensive actions within heterogeneous AI environments.

References :

Aloufi, S. H., Alrige, M., & Bukhary, D. M. (2022). *Clinical applications of rule-based systems in different dental specialties: Scoping review*. JMIR Preprints. https://doi.org/10.2196/preprints.36465

Friedman, R. B., & Frank, A. D. (1983). Use of conditional rule structure to automate clinical decision support: A comparison of artificial intelligence and deterministic programming techniques. *Computers and Biomedical Research, 16*(4), 378–394. https://www.sciencedirect.com/science/article/pii/0010480983900617?via%3Dihub

Sawar, M. J., Abdullah, U., & Ahmed, A. (2010). *Enhanced design of a rule based engine implemented using structured query language*. Semantic Scholar. https://www.semanticscholar.org/paper/Enhanced-Design-of-a-Rule-Based-Engine-Implemented-Sawar-Abdullah/969e03094efba078dbf81c7a802a08d9f743590c

Vassilev, V. T. (2024). Is the rule-based order in AI dead, or is still kicking? *Journal of Robotics and Automation Research, 5*(3), 41-47. https://www.opastpublishers.com/open-access-articles/is-the-rulebased-order-in-ai-dead-or-is-still-kicking.pdf

Question 2

System Implementation Overview

To address the requirements of the scholarship advisory case study, a rule-based expert system was developed using the Python programming language and the **Streamlit** framework for the web interface. The system simulates human decision-making by evaluating applicant data against a strict set of conditions (rules) to determine scholarship eligibility.

The core logic is built upon a **Knowledge Base** represented as a list of JSON-like dictionaries. Each dictionary contains:

- **Conditions:** The specific criteria (e.g., CGPA thresholds, income limits) that must be met.
- **Actions:** The resulting decision (e.g., "AWARD FULL") if the conditions are true.
- **Priority:** A numerical value ensuring that specific rules are checked in the correct order.

User Interface Design

The application features a user-friendly sidebar interface that allows the scholarship committee or applicants to input key data points. As shown in the figures below, the system accepts inputs for **Academic Performance** (CGPA), **Financial Status** (Family Income), **Extracurriculars** (Co-curricular Score), and **Behavioral Record** (Disciplinary Actions).

Testing and Validation

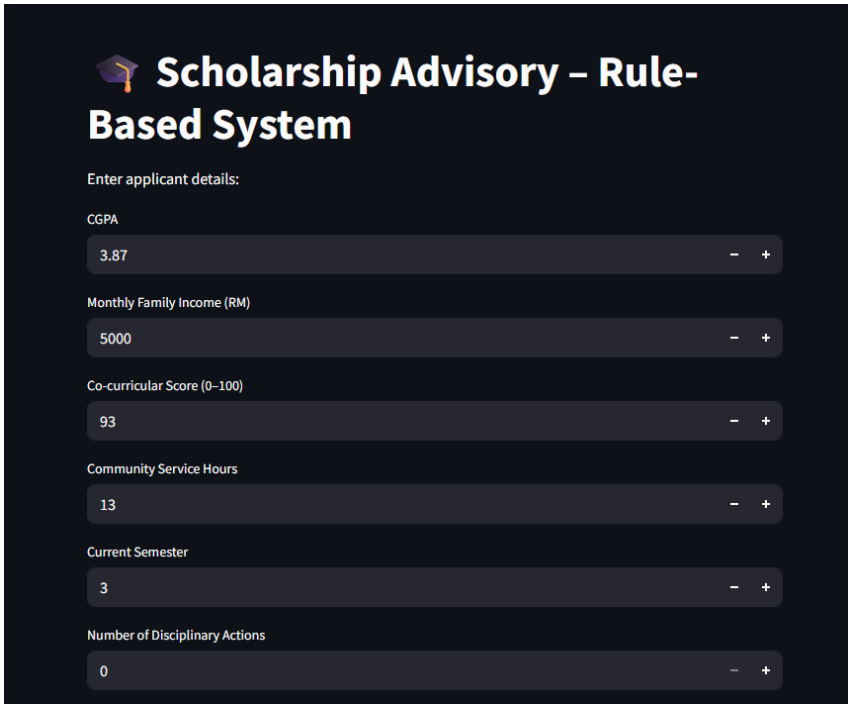
The system was validated using distinct test cases to ensure that the inference engine correctly applies the logic rules based on their assigned priorities.

Test Case A: Successful Full Scholarship

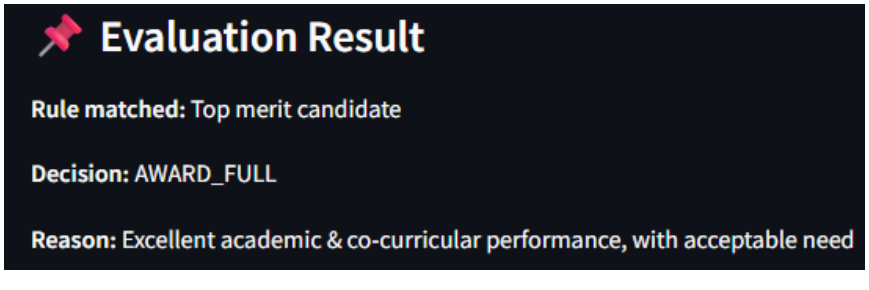
In this scenario, a high-performing candidate was simulated with the following attributes:

- **CGPA:** 3.87 (Exceeds the 3.7 threshold)
- **Monthly Family Income:** RM 5,000 (Below the RM 8,000 limit)
- **Co-curricular Score:** 93 (Exceeds the 80-point threshold)
- **Disciplinary Actions:** 0

Result: The system correctly identified that all conditions for the highest priority rule were met. It triggered the **"Top merit candidate"** rule (Priority 100) and output the decision **"AWARD_FULL"** with the justification: *"Excellent academic & co-curricular performance, with acceptable need."*



Input details for a high-merit candidate.



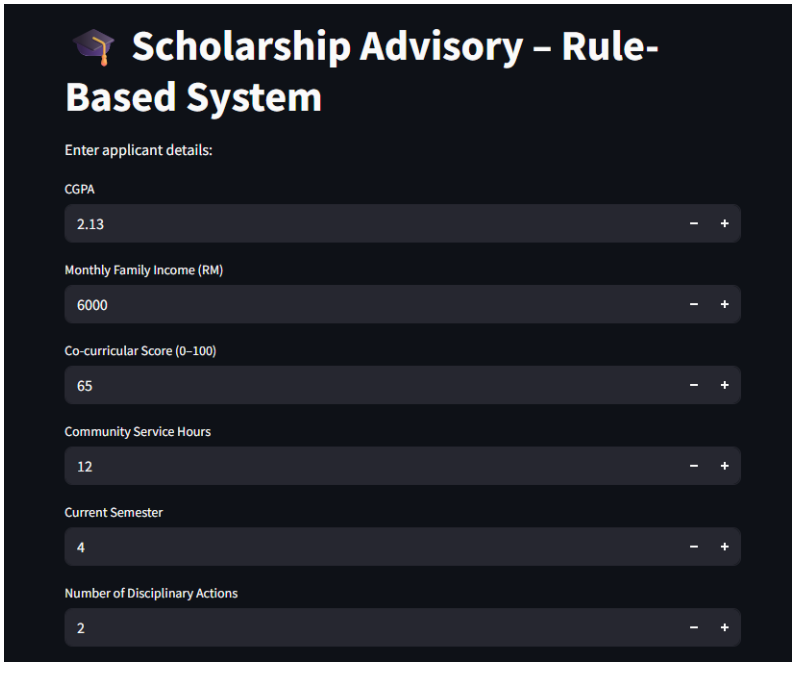
Evaluation result showing a successful full scholarship award.

Test Case B: Application Rejection (Priority Logic Check)

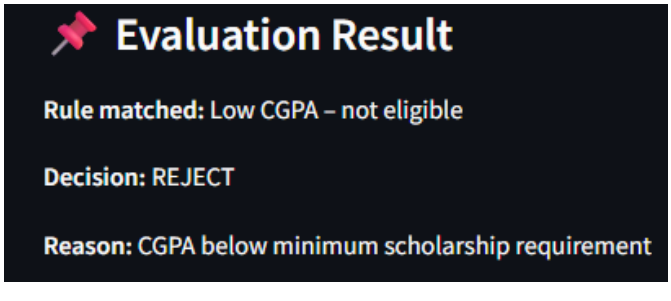
In the second scenario, an applicant was simulated with multiple disqualifying factors to test the system's priority handling:

- **CGPA:** 2.13 (Below the 2.5 minimum)
- **Monthly Family Income:** RM 6,000
- **Disciplinary Actions:** 2 (High disciplinary record)

Result: Although the applicant had 2 disciplinary actions (which matches the "Serious disciplinary record" rule, Priority 90), the system correctly triggered the **"Low CGPA - not eligible"** rule first because it has a higher priority (**Priority 95**). This confirms that the inference engine is correctly sorting and evaluating rules by importance.



Input details for an ineligible candidate with low CGPA and disciplinary issues.



Evaluation result showing a "REJECT" decision triggered by the Low CGPA rule, taking precedence over the disciplinary record.

Deployment

The application is deployed on the Streamlit Community Cloud and is accessible publicly. The source code is maintained in a GitHub repository.

- **App URL:** https://labreport3.streamlit.app/
- **GitHub Repo:** https://github.com/zahinzikrizawawi/LabReport3

Appendix: Source Code

```
import json
import streamlit as st

# ============================================================
#  LOAD RULES (exactly as given in the question)
# ============================================================

RULES = [
    {
        "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"
        }
    },s
    {
        "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"
        }
    }
]


# ========================================================
#   RULE ENGINE
# ========================================================

def evaluate_conditions(applicant, conditions):
    for field, operator, value in conditions:
        applicant_value = applicant[field]

        if operator == ">=" and not applicant_value >= value:
            return False
        if operator == "<=" and not applicant_value <= value:
            return False
        if operator == ">" and not applicant_value > value:
            return False
        if operator == "<" and not applicant_value < value:
            return False
        if operator == "==" and not applicant_value == value:
            return False

    return True
```

```python
def run_rule_engine(applicant):
    # Sort by priority (highest first)
    sorted_rules = sorted(RULES, key=lambda r: r["priority"], reverse=True)

    for rule in sorted_rules:
        if evaluate_conditions(applicant, rule["conditions"]):
            return rule["name"], rule["action"]["decision"], rule["action"]["reason"]

    return None, "NO_DECISION", "No rule matched"


# ========================================================
#   STREAMLIT APPLICATION UI
# ========================================================

st.title("Scholarship Advisory – Rule-Based System")
st.write("Enter applicant details:")

cgpa = st.number_input("CGPA", 0.0, 4.0, step=0.01)
income = st.number_input("Monthly Family Income (RM)", 0, 50000)
cocu = st.number_input("Co-curricular Score (0–100)", 0, 100)
community_hours = st.number_input("Community Service Hours", 0, 500)
semester = st.number_input("Current Semester", 1, 10)
disciplinary = st.number_input("Number of Disciplinary Actions", 0, 10)

if st.button("Evaluate"):
    applicant_data = {
        "cgpa": cgpa,
        "family_income": income,
        "co_curricular_score": cocu,
        "community_service": community_hours,
        "semester": semester,
        "disciplinary_actions": disciplinary
    }

    rule_name, decision, reason = run_rule_engine(applicant_data)

    st.subheader("Evaluation Result")
    st.write(f"**Rule matched:** {rule_name}")
    st.write(f"**Decision:** {decision}")
    st.write(f"**Reason:** {reason}")

    st.success("Evaluation completed successfully!")
```