Portfolio Summary

AI-Assisted Jira to Qase Test Case Automation

Author: Reza Paramarta
Role: QA Engineer
Tools: Jira, Qase.io, n8n, OpenAI (LLM)

---

1. Project Overview

This project demonstrates an AI-assisted QA workflow that converts Jira user stories into structured test cases in Qase.io using n8n.

The objective is not to fully automate testing, but to **assist QA engineers** by reducing repetitive work while maintaining test quality, clarity, and ownership.

---

2. Problem Statement

In many teams, QA engineers manually create test cases based on Jira user stories.
This process is often:

- Time-consuming
- Inconsistent across engineers
- Difficult to scale when requirements grow

---

3. Solution Approach

I built an automation workflow that:

1. Retrieves a Jira issue via REST API
2. Extracts description and acceptance criteria
3. Uses an LLM to generate **one structured test case**

4. Sends the test case to Qase.io via API
5. Allows QA engineers to review and refine the result

AI is used as an assistant, not as a decision-maker.

---

4. Workflow Architecture

```
Jira Issue
    ↓
n8n HTTP Request
    ↓
Field Mapping
    ↓
LLM (Test Case Generation)
    ↓
Structured JSON Output
    ↓
Qase.io API
```

---

5. Test Case Design Principles

- One test case per workflow execution
- Based strictly on Jira acceptance criteria
- Human-readable steps with explicit test data
- Compatible with Qase.io schema
- Designed for manual execution (automation-ready later)

---

6. Key Technical Decisions

- **Structured Output Enforcement**
  Ensures generated test cases always match the Qase schema.

- **Explicit Test Data in Steps**
  Qase does not support a separate “Data” field, so test data is embedded in step actions.

- **Single Responsibility per Run**
  One Jira issue → one test case (simpler traceability).

---

7. Limitations & Future Improvements

Current limitations:

- Only positive scenarios generated
- One test case per execution

Planned improvements:

- Negative test case generation
- Multiple test cases per Jira issue
- Suite assignment via Qase API
- Jira label → Qase metadata mapping

---

8. Why This Project Matters

This project shows:

- Strong understanding of QA fundamentals
- Practical automation skills
- Responsible use of AI in QA

- Ability to bridge requirements and test artifacts

---

9. Key Takeaway

Automation should support QA engineers, not replace them.
AI helps accelerate test creation, but **quality remains a human responsibility**.