QA Metrics Dashboard – Project Summary

Project Overview

The QA Metrics Dashboard is a business intelligence solution developed using **Power BI** and **Microsoft Fabric** to provide a real-time overview of software quality metrics across multiple product releases.

This dashboard was created as a hands-on data visualization project focused on:

- Understanding software quality patterns
- Tracking KPIs over time
- Enabling better QA and release decisions

Key Questions Answered

- What is the current test pass rate across all test types?
- How many bugs were raised per release and per module?
- How quickly are bugs being resolved?
- Which modules or releases require QA attention?

Tools & Technologies Used

Tool	Purpose
Power BI	Interactive dashboard visualizations
Microsoft Fabric	Data modeling and ingestion using Lakehouse
DAX	Custom KPI calculations and logic
CSV Files	Simulated QA data for testing

Dataset Description

The dashboard is powered by four CSV files containing simulated QA data:

- `test_cases.csv`: Contains test case results (status, type, date)
- `bug_reports.csv`: Logs bug IDs, severities, resolution timelines
- `releases.csv`: Captures release cycles and feature counts
- `code_modules.csv`: Module-level data including LOC and ownership

KPIs Tracked

KPI	Description
Test Pass Rate	% of test cases that passed
Bug Density (per KLOC)	Bugs per 1000 lines of code
Avg. Bug Resolution Time	Days taken to resolve a bug
QA Health Score	Weighted composite quality score

Visual Elements

Visual Type	Purpose
KPI Cards	Highlight pass rate, bug density, resolution time, and quality score
Matrix Table	Breakdown of bug count and test pass rate per module
Bar Chart	Distribution of bugs by severity
Line Chart	Bug resolution trend over time
Pie Chart	Test case outcome (pass vs fail)
Test Type Chart	Coverage across unit, integration, system, and regression tests

Filters & Interactivity

The dashboard supports filtering by:

- Release
- Module
- Severity
- Test Type

Each filter dynamically updates all visuals and KPIs.

Summary Insights (Based on Mock Data)

- The current release shows an **80% pass rate** and **average bug resolution time of 8.3 days**.
- Modules like **Authentication** and **Payments** have higher bug densities and need regression focus.
- Resolution trends show decrease in performance over time, with bugs being resolved slower across later releases.

Next Steps

- Set up a production-ready refresh flow via OneLake or OneDrive sync
- Extend to live bug tracking tools (e.g., Jira, DevOps)
- Add Smart Narrative or AI visuals to automate insights