

Peng Yueh Wu

Software Development Engineer In Test (SDET)



-  Taipei, TPE, 111
-  +886960653533
-  kolyfish2@gmail.com

 [Bold Profile](#)

Results-oriented SDET with 3+ years of experience in software engineering, transitioning from a strong Java Backend background to Automated Testing. Leverages 'Black-box' testing perspectives to design high-coverage test scripts and optimize code quality. Proficient in Playwright, with hands-on experience in CI/CD pipelines (GCP). Adept at utilizing AI-driven tools (Cursor IDE / playwright MCP) to accelerate development workflows. Committed to enhancing software delivery quality and efficiency through robust automation strategies.

Portfolios

- <https://github.com/kolyfish/SmartWarehouse>
- https://github.com/kolyfish/Smart_Stock_Light

【Technical Skills】

- Automation Frameworks: Playwright (Python/JS), Selenium, JUnit, Pytest 
- Languages: Python (Proficient), Java (Spring Boot/MVC), SQL 
- DevOps & Tools: Git, GitHub Actions, Jenkins, Docker, GCP (Google Cloud Platform) 
- Database & Testing: MSSQL, Cloud Firestore (NoSQL), Postman (API Testing), JMeter 
- Methodologies: TDD (Test-Driven Development), BDD (Behavior-Driven Development), ISTQB Principles 
- Test script creation 
- Regression testing expertise 
- Manual testing 
- Expert test automation 
- Version control systems 
- Continuous integration expertise 
- Functional testing 



Work history

2025-03 - 2025-12

Automation QA Engineer

Up Young Industry, Taipei, Taipei City

- **Automation Strategy:** Spearheaded the Web and API testing strategy, migrating manual test cases to automated scripts using Playwright (Python), significantly reducing regression testing cycle time.
- **CI/CD Integration:** Managed CI/CD pipelines on Google Cloud Platform (GCP), ensuring automated test execution upon deployment and monitoring logs for release stability.
- **AI-Enhanced Workflow:** Integrated Cursor IDE and AI Agents (MCP) to accelerate DOM element targeting and script generation, boosting overall testing efficiency.
- **Data Integrity:** Validated data consistency across NoSQL (Cloud Firestore) and utilized Elastic Stack to analyze API responses, ensuring synchronization between frontend and backend.

2022-01 - 2024-06

Backend Engineer

Jiefeng International Logistics, Taipei, Taipei City

- **Core System Development:** Developed and maintained the Warehouse Management System (WMS) using Java Spring MVC and MSSQL, managing core modules including Receiving, Picking, and Distribution.
- **Complex Logic Implementation:** Engineered the 'Inventory Hold' mechanism using SQL locking strategies to prevent the shipment of expired or abnormal goods; this experience laid the foundation for deep understanding of boundary value testing.
- **API Integration:** Led the integration with Shopee API and tax systems; optimized performance for bulk Excel (Workbook) data processing.
- **Legacy Maintenance:** Maintained legacy JSP frontends and Java Swing tools to ensure seamless data transfer and operational continuity.



Education

2012-01

M.S.: Graduate Institute of Environmental Education

National Taiwan Normal University - Taipei

2010-01

B.S.: Dept. of Urban and Environmental Planning

Chinese Culture University - Taipei



【Side Projects】

Smart_Stock_Light

- **Stock-to-Light** is a Python-based IoT application that visualizes real-time stock prices by dynamically changing a smart bulb's color to signal buy/sell targets.
- The system features a multi-threaded architecture integrating a Tkinter desktop GUI, a Flask local web server, and automated hardware control via the Tapo API.
- It offers a seamless cross-device experience, allowing users to configure stock parameters remotely by scanning a dynamically generated QR

code on the dashboard.

SmartWarehouse

- **Developed a robust web crawler using Python and Playwright to successfully bypass dynamic anti-scraping mechanisms on PTT forums.**
- **Implemented Test-Driven Development (TDD) principles with Pytest to ensure high code reliability and maintainability before implementation.**
- **Established a fully automated CI/CD pipeline via GitHub Actions to streamline unit testing and code quality checks upon every commit.**



Certifications



Python TQC+ v.2 professional

2025-03