

STRENGTHS

-  **Automation Judgment**
Selected stable, repetitive checks for automation after reviewing manual coverage, preserving human review for exploratory cases and unusual behavior.
-  **Evidence-Based Debugging**
Collected logs, screenshots, SQL results, and reproduction steps during defect triage; clearer evidence helped peers reach decisions with less back-and-forth.
-  **Cross-Functional Communication**
Presented failed tests and coverage gaps during sprint reviews; developers and product peers received practical context for deciding next fixes.
-  **Test Design**
Used boundary values, mocked responses, and user stories in coursework and lab work, catching cases that happy-path checks could miss.
-  **Reliable Collaboration**
Peer teams sought help with Python lab issues and test setup; patient explanations kept shared work moving when failures were hard to reproduce.

SKILLS

<u>Python</u>	<u>pytest</u>	<u>unittest</u>	<u>Selenium</u>
<u>requests</u>	<u>SQL</u>	<u>REST API testing</u>	
	<u>Functional testing</u>		
	<u>Regression testing</u>		
	<u>Integration testing</u>		
	<u>Exploratory testing</u>		
<u>Test-case design</u>	<u>Defect tracking</u>		
<u>AQA planning</u>	<u>Agile sprints</u>		

SUMMARY

Early-career **Python development** and QA automation candidate with supervised university lab, internship, and capstone experience building scripts that validate web applications, APIs, databases, networks, and software workflows. Practical work includes Python, pytest, unittest, Selenium, requests, SQL, PostgreSQL, SQLite, Git, GitHub Actions, **REST API testing**, browser testing, defect tracking, and **test-case design**. Converts repeatable manual checks into maintainable automated coverage, documents errors with useful evidence, and communicates findings clearly with developers, **manual QA engineers**, product managers, and peer reviewers. Brings curiosity about product design, patience during debugging, clear C1 English communication, availability Monday through Friday from 9 AM to 6 PM Eastern Standard Time, and flexibility for occasional work outside traditional local hours.

EDUCATION

Bachelor's Degree in Computer Science
University of New Haven  GPA: 3.8  2026  New Haven, CT
***Coursework:** Python programming, software testing, database systems, web application development*

TECHNICAL SKILLS

- **Programming Languages:** Python, SQL, Bash
- **Python Testing:** pytest, unittest, fixtures
- **Browser Automation:** Selenium, Chrome, Firefox
- **API Testing:** requests, REST, mocked responses
- **Database Testing:** PostgreSQL, SQLite, SQL
- **Defect Tracking:** Jira, logs, screenshots
- **Source Control:** Git, GitHub, branching
- **CI Automation:** GitHub Actions, workflows, artifacts
- **QA Methods:** Functional testing, regression testing, integration testing
- **Test Design:** Boundary values, parameterization, test cases
- **Planning Practices:** AQA planning, Agile sprints, scope review
- **Documentation:** Test strategy, technical reports, reproduction steps

EXPERIENCE

QA Automation Intern
University Software Quality Lab  September 2025 - May 2026  New Haven, CT

Supported quality engineering for a university language-learning platform through Python automation, browser validation, API checks, database verification, defect analysis, and sprint planning. Worked under QA supervision with student developers, manual QA reviewers, and product stakeholders, helping convert stable manual coverage into repeatable checks that gave release reviews clearer evidence.

- Built and maintained 34 **Python** and **pytest** checks for authentication, lesson navigation, account settings, and subscription workflows, giving reviewers repeatable product validation.
- Added **Selenium** fixtures, reusable page objects, and environment variables for consistent Chrome and Firefox regression runs, reducing setup variation across test sessions.
- Extended API and database validation with requests and SQL queries against **PostgreSQL**, confirming user actions created accurate records and exposing cross-layer defects.
- Documented reproducible Jira defects with logs, screenshots, expected results, and severity recommendations, helping student developers reach triage decisions faster.
- Reviewed manual QA checklists with a supervising QA engineer, selected stable repetitive cases for automation, and kept exploratory scenarios manual where human judgment mattered.
- Presented sprint results during **AQA planning**, sharing failed-test evidence, coverage gaps, and proposed fixes with **developers** and the product owner.

LANGUAGES



MY CAREER



- QA Automation Intern at University Software Quality Lab (8 Months)
- Python Test Automation Developer at Capstone Project (4 Months)
- Software Testing Project Analyst at Student QA Automation Lab (3 Months)

Python Test Automation Developer

Capstone Project 📅 January 2025 - May 2025 📍 New Haven, CT

Created a supervised Python test automation solution for a mock language-company website, covering checkout and course enrollment workflows that slowed release reviews. Combined framework structure, functional analysis, REST validation, continuous integration, defect seeding, and evidence-based reporting so faculty and peers could assess maintainability and coverage.

- Designed a **Python** automation framework for mock website checkout and course enrollment, replacing repetitive manual review with a clearer, repeatable validation path.
- Implemented pytest fixtures, parameterized data, and Selenium page objects across 52 functional scenarios, covering valid, invalid, and boundary-value inputs.
- Added REST checks for enrollment and payment-status endpoints, using mocked responses to examine timeout, authorization, and malformed-payload behavior safely.
- Connected test execution with **GitHub Actions** and published readable pass/fail artifacts, letting instructors review results without recreating local setup.
- Compared automated outcomes with an independently prepared manual test set, using defect-seeding exercises to identify missed cases and false positives.
- Delivered a written test strategy and live demonstration that received an exemplary faculty assessment for traceability, maintainability, and evidence-based evaluation.

Software Testing Project Analyst

Student QA Automation Lab 📅 September 2024 - December 2024 📍 New Haven, CT

Investigated a simulated client portal through supervised **software testing** projects involving user-story analysis, Python scripting, database checks, defect reporting, exploratory review, and team presentations. Partnered with a teaching assistant and six-person student team to connect observed user friction with actionable fixes before a final demonstration.

- Mapped user stories to functional, regression, integration, and exploratory scenarios, clarifying coverage for inconsistent client-portal behavior before test execution.
- Wrote introductory **Python scripts** with **unittest** and requests for login, profile, search, and notification workflows against a local application.
- Used **SQLite** queries and controlled test data to compare responses with stored records, isolating duplicate and missing-entry defects for focused investigation.
- Maintained a shared defect log with reproduction steps, environment details, screenshots, and retest status, giving weekly reviews consistent evidence.
- Joined brainstorming sessions that proposed testable improvements for error messages, navigation, and form validation based on observed user friction.
- Prepared a concise testing dashboard and presented findings to a six-person project team, supporting prioritization of three fixes before final demonstration.

PROJECTS

Language Platform API Reliability Study 📅 2025

Built a separate REST testing study for enrollment and payment-status services. Used Python, requests, mocked responses, and SQL evidence to examine authorization, timeout, malformed-payload, and record-integrity behavior.

Cross-Browser Regression Evidence Hub 📅 2025

Created a distinct browser-testing concept using Selenium page objects, pytest fixtures, environment variables, and GitHub Actions artifacts. Focused on readable evidence that helps reviewers compare Chrome and Firefox behavior.

LEADERSHIP & AWARDS

- Dean's List, University of New Haven, Fall 2024 and Spring 2025; sustained academic focus across core computer science coursework.
- Outstanding Capstone Testing Strategy, Computer Science Department, 2025; recognized for traceability, maintainability, and practical test evidence.
- Faculty Recognition for Technical Communication, 2025; honored for clear testing reports, demonstrations, and collaboration with peers.

CERTIFICATIONS

- Python for Everybody Specialization, Coursera 📅 2025
- Software Testing and Automation Foundations, IBM SkillsBuild 📅 2025
- Git and GitHub Essentials, Microsoft Learn 📅 2024

PROFESSIONAL AFFILIATIONS

- QA Study Group Coordinator, University of New Haven Computer Science Club, 2025-2026; organized peer sessions around Python and software testing.
- Peer Tutor, Introductory Python Laboratory, University of New Haven, 2024-2026; helped classmates debug scripts and build confidence with core concepts.
- Participant, Northeast Collegiate Software Testing Challenge, 2025; practiced collaborative test analysis in a competitive academic setting.

ADDITIONAL INFORMATION

Work Status : Authorized to work in United States. No sponsorship required.

REFERENCES

AVAILABLE ON REQUEST