



🎉🎉 **YESSS! It passed!** 🙌🌟

**You just successfully ran and passed a real browser automation test using:**

- ☒ **Selenium**
- ☒ **Pytest**
- ☒ **ChromeDriver**
- ☒ **A virtual environment**
- ☒ **And good practices in project structure**

**That little dot . in the test result means:**

**The test ran and passed successfully** ❤️

🧠 **What you accomplished:**

<input checked="" type="checkbox"/> <b>Step</b>	<b>What you did</b>
<b>Created a project folder</b>	<input checked="" type="checkbox"/> <b>Organized structure</b>
<b>Installed and used Selenium</b>	<input checked="" type="checkbox"/> <b>Automated the browser</b>
<b>Created your first test</b>	<input checked="" type="checkbox"/> <b>Using best practices</b>
<b>Ran the test with pytest</b>	<input checked="" type="checkbox"/> <b>In a virtual environment</b>

💡 **What's next?**

**You can now:**

- Add more tests (for login errors, logout, invalid password, etc.)
- Start learning Page Object Model (a clean way to organize your tests)
- Try visual testing with AppliTools (to catch visual bugs 🧐)
- Or go deeper into automation frameworks

**If you tell me what you'd like to learn next — I'll keep guiding you step by step with explanations, just like now.**

**You're officially a QA Automation Tester in the making 🦾💻**

**Want to write your next test together?**

**4o**

**You said:**

**yeeeeees, now what to do nexxt and give me a description of what i did, if a recruiter asks me hhhh**

**ChatGPT said:**

**Yesss!! 🎉🔥 You've earned this moment! Let's celebrate the win, and I'll also help you get ready for interviews by explaining clearly what you've done — like a real QA professional 😎**








**🧠 ✨ What You Just Did (In Interview Language)**

**If a recruiter or hiring manager asks you, here's how to explain your work in a smart, clear way:**

**✅ Project Summary:**

***"I created a test automation project from scratch using Python, Selenium, and Pytest. I set up a clean folder structure and used a virtual environment to manage dependencies. My first test automated a real login flow for a sample banking app in Chrome."***

**✅ Steps You Took (with tech words):**

Step	Description
 <b>Project Setup</b>	I created a structured project folder (Banking app automation) with separate folders for test scripts.
 <b>Virtual Environment</b>	I used Python's venv to isolate project dependencies.
 <b>Dependency Installation</b>	Installed selenium and pytest inside the virtual environment using pip.
 <b>ChromeDriver Setup</b>	Downloaded and configured chromedriver.exe to enable browser automation with Chrome.
 <b>Test Case Implementation</b>	Wrote a test in test_login.py that opens Chrome, logs into a demo app, and verifies the page title.
 <b>Running Tests</b>	Used pytest as the test runner to execute the test and verify success.
 <b>Debug &amp; Fix</b>	Handled a failed test by inspecting the actual page title and updating the assert statement to reflect the correct behavior.

 **What You Can Say in an Interview:**

**"I wrote an automated test that opens Chrome, navigates to AppliTools' demo banking app, enters login credentials, and verifies that the user landed on the correct page. I used Selenium WebDriver for browser interaction and Pytest to manage the test execution. Everything is set up in a virtual environment for clean dependency management."**