Start Here: The Functionize Studio Exercise Series

Exercises · https://docs.functionize.com/hc/en-us/articles/43766996528791-Start-Here-The-Functionize-Studio-Exercise-Series

This is a self-paced series that teaches Functionize Studio by building something. You work against our own demo store, so you can follow every step exactly without touching your own application, and nothing you do here can affect a real system.

Eight exercises, about an hour and three quarters end to end. Each stands on its own and takes ten to fifteen minutes, so you can stop after any of them and come back later.

Tip: click any screenshot to open it full size.

The demo application

Every exercise uses the **Anaqa demo store**, a practice e-commerce site we host:

```
https://ecom.functionizeapp.com/testsites/anaqa/index.php?route=common/home
```

You will never need to sign in. Every exercise uses the store as a guest — browsing, searching, opening products and using the basket all work without an account. That means no credentials to manage, and nothing in these exercises can lock you out.

Open the store in a browser tab and click around for a minute before you start. Knowing what the application does makes the rest far easier, and it is the same advice we would give you about your own application.

What is in the store

Electronics > Laptops	2 products — Vostro 3458 at $175.00, iBall CompBook at $560.00
Electronics > Air Conditioner	5 products — including Boltas 2 Ton 3 Star Split AC at $966.00
Electronics > Mobiles	30 products
Electronics > Computer Accessories	3 products

Small, predictable categories make good assertions — "exactly 2 products" is something a test can actually prove.

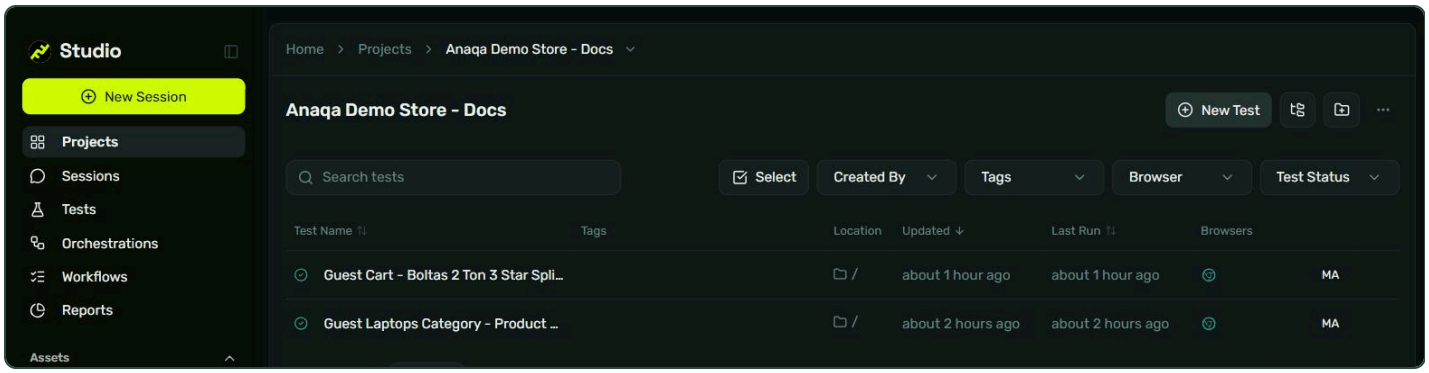
Set up once

Make a project to keep the exercises in, so they stay separate from real work.

1. Open **Projects** in the left sidebar.

2. Choose **New Project**.
3. Name it something obvious, such as **Studio Exercises**.
4. Open it.

A project with a couple of tests in it looks like this:



That is the whole setup. Nothing to install — tests run on our infrastructure in a real browser.

The exercises

1. Create your first test from a prompt Describe an outcome in a sentence and watch Studio build and run it.	10 min
2. Write assertions that prove something The difference between a test that clicks things and a test that checks things.	10 min
3. Read a run Steps, screenshots, and the Data tab that answers most questions.	10 min
4. Change a test by asking Extend and edit an existing test in conversation.	10 min
5. Break it on purpose, then diagnose it Make a test fail for a real reason and work out why.	15 min
6. Group tests into an orchestration Turn your tests into a suite that runs as one job.	10 min
7. Drive Studio from your own AI tool Connect over MCP and work from your editor.	15 min
8. Tag a suite and give it a schedule Tag from the Tests screen, build a suite from the tag, and set the run order, re-runs and alerts deliberately.	15 min

How to get the most out of these

Type the prompts rather than pasting them. Every exercise gives you an exact prompt so you are never stuck, but the skill you are building is phrasing a request well. Typing it makes you read it.

Change one thing and see what happens. Ask for a different product, a different category, a different assertion. Nothing here is precious.

Read what the agent says back. It tells you what it understood before it builds anything, and what it was unsure about when it finished. Skimming past that is the most common way to end up with a test that does not do what you meant.

Expect to be asked questions. If a request cannot work, the agent says so rather than building something broken.

If you get stuck

Ask the agent — it has your test and your run in front of it, and "why did this fail?" is a perfectly good prompt. Otherwise see Diagnosing a Failed Test in Functionize Studio, or raise a ticket with your test ID and the time of the run.

Ready? Start with Exercise 1.

After the exercises: getting a whole team up to speed

The eight exercises make one person productive. Three more pages turn that into something a team shares:

- **Study Kit: Getting a Team Up to Speed in a Week** — a five-day plan of about ninety minutes a day, with two sessions that need everyone present. It uses the exercises below plus the strategy templates.
- **Flash Cards: The Vocabulary Your Team Needs** — five decks covering the product, classic quality engineering, the AI-driven development lifecycle, evaluating AI and measurement. Readable here, or printable as cut-out cards.
- **Check Yourself: Twenty Questions on Testing With AI** — twenty questions with the answers behind them. Best run with a team: read each one aloud and let people answer before anyone reveals the card, because the disagreements are the useful part.