QA Automation — Expert Module

Client Guide: Scripts Execution and Explanation

Playwright Automation — One-button run setup

Prepared for: Client

Prepared by: Yahya M. Mirza

Contents

1 Overview

This document explains how to run and understand the automated test suite for the **Expert Module** of your web application. It mirrors the same client-friendly format used for the Admin Module and focuses on what matters to you: how to run tests, what input data they expect, how to read results, and what to check before running tests.

The automation uses Playwright and is configured so you can run all tests with a single double-click and get a clear HTML report at the end.

2 Why This Matters

Automated tests give you fast, repeatable checks of the most important user journeys in the Expert Module (expert signup, consultations, events and profile updates). Instead of manually re-checking these flows every release, you run the suite and quickly see whether any critical functionality regressed.

Benefits for you:

- Confidence: key flows verify themselves every run.
- Speed: a one-button run completes in minutes and produces a human-friendly report.
- Clarity: the report highlights exactly which step failed and why.

3 Test Data — The Input of the Tests

All client-facing tests rely on a single source of truth for expected data. This is the file `testdata.ts`, located inside the project fixtures. Think of this file as the list of records and names the tests expect to find in your app before running.

3.1 Naming Conventions

Fields in `testdata.ts` use clear, descriptive names so you (or your content admin) know what to check in the application. Example:

```
EXPERT.PROFILE.NAME = "John Doe - Test Expert"
```

This tells the test to look for an expert profile with that exact name. If the name in your live site changes, update `testdata.ts` to match.

3.2 Common Fields for the Expert Module

Typical entries in `testdata.ts` for the Expert Module include (but are not limited to):

- Expert profile names and emails (used for creating/locating profiles)
- Service titles and categories (used for listing services)
- Booking slots / availability templates (used in booking flows)
- Pricing / payout settings (used in payment-related tests)
- Review text samples (used for testing review display)

3.3 File Path

`project-root/tests/fixtures/testdata.ts`

3.4 Your Responsibility (Data Ownership)

Because your app changes over time, please ensure the records defined in `testdata.ts` exist in the specified environment (staging or testing) before running the tests. If a record is missing or renamed, the related test will either fail or, depending on the test, be skipped.

4 Prerequisites (one-time setup)

The following are required once when setting up the automation on a machine:

1. Node.js LTS (v18 or newer) — verify by running: `node -v`
2. Install project dependencies: `npm ci` (or `npm install`)
3. Install Playwright browsers (first-time only): `npx playwright install`

5 How to Run Everything (One Button)

5.1 Windows (most common)

1. Open the project folder.
2. Double-click `run-tests.bat`.
3. The console will run all Expert Module tests (typical run time: 6–8 minutes).
4. At the end, the HTML report will automatically open in your browser.

5.2 macOS / Linux

If you prefer a terminal approach, open a terminal in the project root and run:

```
npm run test:all
```

which behaves the same as the one-button script.

5.3 Report

The report shows **PASSED** / **FAILED** / **SKIPPED** for each test and provides details for failures so you know what to fix (data vs UI).

6 Skipping a Test (Optional)

If you want to temporarily disable a specific test from the run, the script author can mark it as skipped in the test file. For example (this is an instruction for your automation contact; you do not need to edit code yourself):

```
test.skip('This test will be skipped', async () => { ...  })
```

Skipped tests will appear as **SKIPPED** in the report instead of failing.

7 Fixtures & Test Files

All images, CSVs, and other file inputs used by the Expert Module tests are stored in:

```
tests/fixtures/
```

7.1 Purpose of This Folder

Files in `tests/fixtures/` act as inputs for tests that need file uploads (e.g., profile pictures, sample CSVs for bulk service import).

7.2 How to Change Fixtures

- **Images:** Replace the image file inside the folder with your own, or update the path in `testdata.ts` to point to a different image.
- **CSV Files:** Edit an existing CSV or create a new one and update the path in `testdata.ts` for bulk-import tests.

8 Test Data Structure Explanation

We use a consistent new / edit pattern across the Expert Module tests so clients can easily see what will be created and what will be changed during automated runs.

8.1 Example — Expert Service

In `testdata.ts` you may see an object similar to:

```
service: {
  new: {
    title: "Test Consultation",
    category: "Business",
    duration: "60",
    price: "50",
    description: "Short test service",
  },
  edit: {
    oldTitle: "Test Consultation",
    newTitle: "Test Consultation - Updated",
    price: "45",
    successMsg: "Service updated successfully",
  }
}
```

8.2 Breaking This Down

- **new:** Fields used when the test needs to create a new service/profile. Example: `title, category, price`. The test will create a record with these attributes.
- **edit:** Fields used when the test edits an existing record. Example: `oldTitle` is the exact record name the test will search for; `newTitle` is the name after editing.

8.3 Why This Matters

This pattern (new/edit) ensures tests can verify both creation and modification flows. If the `oldTitle` (or equivalent) does not exist in your environment, the edit test will fail or be skipped.

9 Troubleshooting & Reading Reports

9.1 Common Cases and Fixes

1. **Can't find element / selector changed** The app's UI changed and the automation cannot locate a button, field or link. Action: notify the automation owner to update selectors, or provide screenshots for the change.
2. **Data-dependent failure** A required record (e.g., Expert profile: "John Doe - Test Expert") is missing or renamed. Fix: recreate/rename the record in the app or update `testdata.ts`.
3. **Slow or flaky network responses** Network slowness can cause timeouts. We usually increase timeouts temporarily while investigating, or run tests again when the environment is stable.

9.2 How to Read a Failed Test in the Report

Open the HTML report after a run. Failed tests are marked in red; click a failed test to see:

- Error message (what was expected vs. what was found)
- Full step-by-step trace and screenshots (if configured)

Use this information to determine whether the failure is caused by:

- *Missing data* in the environment (fix in app or testdata.ts), or
- *UI changes* that require selector updates in the automation.

9.3 If the Report Doesn't Open Automatically

Open a command line in the automation folder and run:

```
npx playwright show-report
```

This will open the latest generated HTML report in your default browser.

10 Responsibilities (Important)

Before running tests, please ensure:

- All records listed in `testdata.ts` exist in the environment you plan to run tests against (staging/test).
- If a record is missing, either recreate it in the app or update `testdata.ts` to match the current record name.
- If multiple team members will run tests, make sure they use the same environment (staging/test) to avoid false failures.

11 Support & Change Requests

- **Small tweaks** (selector updates, small test additions) — quick turnaround.
- **New modules or major flow changes** — estimated separately based on scope.
- **Re-activating a skipped test** — simply remove the `.skip` from the test file; we can help if you prefer we do it.

12 Final Notes

- One double-click runs the entire Expert Module test suite and opens a clear HTML report.
- Keep `testdata.ts` updated to match your live/staging data to avoid false failures.
- Typical run time: 6–8 minutes (may vary depending on environment).
- The report provides direct guidance on what failed and why — this speeds up triage.

If you want this LaTeX compiled to PDF or want any wording changed to match specific expert flows (e.g., "appointments", "payouts", or "consultation types"), tell us which pieces to adjust and we will update the document.