

HW04 – Automation Testing - En

HW04 – Automation Testing

1. General Information

Exercise ID	HW04-AI
Duration	10 hours
Deadline	Please refer to the submission link on Moodle
Form	Individual Assignment
Submission	Moodle (report)
Lecturers & TAs	Dr. Lam Quang Vu / Dr. Tran Duy Hoang / MSc. Tran Thi Bich Hanh / MSc. Truong Phuoc Loc / MSc. Ho Tuan Thanh
Contact	lqv@fit.hcmus.edu.vn / tdhoang@fit.hcmus.edu.vn / ttbhanh@fit.hcmus.edu.vn / tploc@fit.hcmus.edu.vn / hthanh@fit.hcmus.edu.vn
AI Policy	Open — a declaration and an attached AI Audit Report are mandatory
Required Bloom-AI Level	G9.1 → G9.6, depending on the homework (see the <i>CLO Mapping</i>)

2. Guiding Principles

These principles define how you are expected to work throughout the series of assignments in this course. Read them carefully before you begin, as your submission will be evaluated against them.

- **AI-First strategy.** You are required to apply AI to the testing techniques covered in class. However, this does not mean issuing a single, generic prompt such as *"write all the automation scripts for this feature."* Instead, you must guide the AI through every step of the technique as it was taught, using the AI as a disciplined assistant rather than a black box.
- **Human review.** Every result produced by the AI must be carefully reviewed by you, the student. You are fully responsible for the correctness of these results. You are expected to make any necessary corrections and

refinements — submitting the raw AI output without review is not acceptable.

- **AI Audit Report.** The entire process of using AI must be recorded in a complete log. You are encouraged to build Agent Skills that can automatically perform these activities on similar exercises. If you do **not** use AI, you must still declare this explicitly.
- **Documentation.** The whole working process must be documented in a text-based format such as Markdown.
- **Quality over completion.** Your work will be graded not merely on whether it is complete, but on the quantity and quality of the deliverables: the automation scripts, data files, HTML reports, the bug report, the demo video, and referenced links.

3. Learning Outcomes

By completing this assignment, you will be able to:

- Use AI to generate automation test scripts for the SUT's web frontend with Playwright or Selenium, then review and refine them.
- Apply data-driven testing and assertion patterns, and execute the suite fully across multiple browsers.
- Critically review the AI-generated automation: fix it, analyse what the AI got wrong or missed, and produce complete outputs (HTML reports and bug reports).
- Demonstrate Bloom-AI competencies at levels **G9.2 (Apply)**, **G9.3 (Analyse)**, and **G9.4 (Collaborate with AI)**.

4. System Under Test (SUT)

SUT: EShop — a Vietnamese e-commerce demo application designed for testing practice.

Repository: <https://github.com/ttbhanh/eshop-sut>

The application's features are organised into the following pools:

- **Pool A — Authentication, Categories, and Products**
 - FR-01: Account registration
 - FR-02: Login and account logout

- FR-03: Forgot password and password reset (two steps)
- FR-04: Personal profile management
- FR-05: Product listing and search
- FR-06: Product detail view
- **Pool B — Shopping Cart and Checkout**
 - FR-07: Shopping cart
 - FR-08: Checkout
 - FR-09: Discount coupons
 - FR-10: Order state machine
 - FR-11: Order history view (user)
- **Pool C — Web Admin**
 - FR-12: Access control
 - FR-13: Dashboard
 - FR-14: Category management (CRUD)
 - FR-15: Product management (CRUD)
 - FR-16: Product import from CSV
 - FR-17: Coupon management (CRUD)
 - FR-18: Order management (admin)
 - FR-19: User management (admin)
- **Pool D — Mobile App**

5. Feature Selection

- Automate the **same three (3) web features** you selected in HW02 — one each from **Pools A, B, and C**. The Pool D (mobile) feature is **not** used in this homework, because here you automate the web frontend.
- If you have not completed HW02, self-declare three web features from Pools A–C directly in this report and clearly state the reason HW02 is unavailable.

- Within each group, ensure that your selection is **not duplicated** among the members of the group, as in HW02.

6. Requirements

For each of the following tasks, document your process in the main report and attach the required evidence. Review the relevant course lectures on test automation before you begin.

Task 1 — AI-generated automation scripts

Following the AI-first strategy, use an AI tool to generate the automation scripts, then review, fix, and take full responsibility for them.

- **Generate with AI.** For **each** of your three features, drive an AI tool — step by step, not with a single generic prompt — to convert **at least 12 test cases** into automation scripts. The 12 may be any combination of positive, negative, and edge cases — all types count toward the minimum.
- **Make the scripts data-driven.** The test data must be stored in a **separate .csv or .json file** (hardcoded inline arrays or objects in the script are not accepted), and the scripts must use **at least three distinct assertion patterns**.
- **Run on at least 3 browsers** (Chromium / Firefox / WebKit, or Chrome / Edge / Firefox). Each feature must run on all three browsers — at least **9 browser runs** in total across the suite. Each run must produce an **HTML report** (Allure or the Playwright HTML reporter) that visibly displays **"Run by: {StudentID}"** (in the title, header, footer, or report metadata).
- **Review and fix (human review).** Critically review the AI-generated scripts and correct them. Report what the AI got wrong or missed — for example, fragile selectors, weak or missing assertions, missing edge cases, or flaky waits — and explain *why* it missed them (prompt quality, model limitations, or characteristics of the feature). You are fully responsible for the final scripts.
- **Make the automation as complete as possible.** Aim for the suite to execute end to end and produce as full a set of outputs as you can: the multi-browser HTML reports and, wherever a failing assertion reveals a genuine defect, a **bug report**. Log such bugs both in the Markdown report and on your GitHub Issues page, attaching a **screenshot** to each issue. Document any test cases you could not automate and explain why.

Task 2 — Demo video

- Record an **unlisted YouTube video of at least 5 minutes**, narrated in Vietnamese, demonstrating **one** of your automation scripts running end to end (including the multi-browser run and the generated HTML report).
- Narrate at least one fix you made to the AI-generated script during your review.
- The video must evidence authorship by showing **either your face-cam or a terminal running `whoami` and `hostname`**.

7. Agent Skill

- You are encouraged to build an Agent Skill that applies this automation workflow (data-driven, multi-browser script generation and maintenance), so that it can be reused on additional features in future testing tasks.
- Submit the skill together with a demonstration video (YouTube link) that shows, end to end, how you used the skill on a complete feature.

8. Allowed Tools and Bloom-AI Level

You may use the following tools, and you must declare them in your AI Audit Report:

- Any AI tool of your choice (e.g., ChatGPT, Claude, Gemini, Copilot, Cursor) — for generating the automation scripts.
- Playwright (recommended) or Selenium 4+.
- Allure or the Playwright HTML reporter.

The required Bloom-AI level for this homework is **G9.2 (Apply)**, **G9.3 (Analyse)**, and **G9.4 (Collaborate)**.

9. AI Audit Report (Mandatory Appendix)

Attach the AI Audit Report as an appendix. Use the content of the given AI Templates if needed.

- If you did not use AI, declare: *"I do not use any AI help in this exercise."*
- If you did use AI, declare: *"I use AI tools for the following tasks,"* and include the following information for each interaction:

- Name of the AI tool
- Date and time
- Your prompt
- The AI output

To simplify this process, you are encouraged to create a skill or rule that extracts the information above automatically after an AI session.

10. AI Critique (200–300 words, Mandatory)

Write a paragraph of 200–300 words critiquing the AI. Address the following questions: Where did the AI get something wrong, biased, or incomplete? Why did it fail to catch the issue? What principle have you learned about collaborating with AI during this assignment?

Use the content of the given AI Templates if needed.

11. Anti-AI-Cheat Constraints

This homework relies on real, attributable execution evidence. The following must not be AI-generated or fabricated, and the TAs verify them during grading:

- The HTML reports, which must contain **"Run by: {StudentID}"** together with an ISO timestamp.
- The demo video, which must contain your own voice narration and show either your face-cam or a terminal running `whoami` and `hostname`.

12. Git Commit Log

- Maintain a **public GitHub repository** with a meaningful history: **at least 8 commits**. Only commits that change test-script files (`.spec.js`, `.spec.ts`, or equivalent) count toward the 8-commit minimum; commits touching only the README, PDF, or other non-test documents do not count.
- Provide the Git commit log in a text-based file format.

13. Oral Defense

A randomly selected **30% of students** may be invited to a 5–7-minute oral defense during the week following the deadline, to explain how they completed

this homework.

14. Submission Regulations

- **Filename format:** `<StudentID>_HW04_AI_Automation_<SelfAssessedGrade>.zip`
 - *SelfAssessedGrade*: a 3-digit number in the range [000, 100].
 - *Example*: `25127001_HW04_AI_Automation_090.zip`
- **Required contents of the .zip :**
 - Main report (Markdown + PDF), including the automation report and your review / gap analysis of the AI-generated scripts.
 - The public GitHub repository link (scripts, data files, and HTML reports).
 - The multi-browser HTML reports (Allure / Playwright).
 - The unlisted YouTube demo video link.
 - AI Critique and AI Audit Report (Markdown + PDF).
 - Git commit log (text file).
 - Bug report, with screenshots of the bugs on the GitHub Issues page (if any).
 - A `README.md` containing the self-assessment table (below) and a test summary report: number of features; number of test cases automated, executed, passed, and failed; number of browser runs; number of bugs; and the demo video link.
 - Any other supporting materials.
- Submit to Moodle. For the deadline, refer to the submission link.

15. Assessment Template

No.	Criteria	Grade	Self-Assessed Grade
1	Task 1 - Feature A	25	
1	Task 1 - Feature B	25	
1	Task 1 - Feature C	25	
2	Task 2 — Demo video	15	
3	Agent Skills	10	

No.	Criteria	Grade	Self-Assessed Grade
	Total	100	

16. References

- ISTQB Foundation Level Syllabus (latest edition).
- Hardman, P. (2025). *A Post-AI Learning Taxonomy*.
- Fuster Rabella, M. (2025). *OECD Education Working Paper No. 338*.
- Anthropic (2025). *Building Reliable AI Test Agents* — engineering blog.
- DeepEval & Promptfoo documentation — LLM testing frameworks.

17. Other Regulations

- Late submission is **not permitted**.
- Missing any required document results in **0 points**.
- Copying between students — **including prompts** — results in a **grade of 0 for both parties**.