AI Usage Log — Full Session (timestamps from JSONL)

AI Tool: Claude Code (Claude Opus 4.8)
Session file: `02afd25e-2c05-49c2-bb70-f3bd37adee00.jsonl`
Genuine user interactions: 48
Timezone: UTC as recorded (`Z`). Timestamps span 2026-07-03 to 2026-07-06.

Interaction 1

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T15:47:19.489Z
- **AI response timestamp:** 2026-07-03T15:47:28.960Z
- **Tools used:** Read, Glob, Glob, Read, Read, Read, Grep, Grep, Write

User Prompt

```
Selected features:

FR-05: Product listing and search
FR-09: Discount coupons
FR-14: Category management (CRUD)
FR-13: Dashboard (Mobile)

For each of your selected features, complete the following tasks:

Domain Testing. With the support of AI tools, apply the domain testing technique to design a comprehensive set of test cases. You are er
Boundary Value Analysis. Likewise, apply the boundary value analysis technique to design a comprehensive set of test cases, adding more
AI gap analysis. If your AI tools miss any test cases or bugs, you must report them. Explain why the AI missed them — for example, due t
Bug reporting. Report all discovered bugs both in the Markdown report and on your group's GitHub Issues page. Remember to attach bug scr

Agent Skill: build Agent Skills that apply the domain testing and boundary value analysis techniques, so that they can be reused on addi

Please write a workflow in docs folder with AI collaboration to complete above tasks precisely
```

AI Output

```
I'll start by understanding the project structure and the specs for your four selected features, then write a precise AI-collaboration w

Let me read the API spec and the backend implementation for the four selected features to ground the workflow precisely.

Let me check the seed data and the mobile dashboard implementation to ground the workflow accurately.

I now have a precise picture of all four features and their planted-bug hotspots. Let me create the workflow document in the `docs` folc

Key grounding facts I'll build the workflow around:
- **FR-05** search: `WHERE name LIKE '%${searchQuery}%'` (string concat → SQLi/XSS), `/products/:id` returns `{}`+200 on miss, even-id p
- **FR-09** coupon: `total_amount > min_order_amount` (should be `>=` — boundary bug), broken percent formula `total*(1-discount_value)`
- **FR-14** category: no empty-name validation, no `role=='admin'` check.
- **FR-13** dashboard: no mobile dashboard endpoint/screen at all; revenue rule (`status='delivered'` only) + the `canceled→delivered` s

I've created the workflow at `docs/ai-testing-workflow.md`. It's grounded in the actual EShop code and specs, not generic boilerplate —

## What's in it

**§0 Deliverables map** — every artifact the assignment wants (D1–D7) with its exact target path, so nothing gets missed.

**§1 AI-collaboration model** — a who-does-what table with three golden rules, the most important being: **the `README.md` spec is the c

**§2 Setup** — the 4-terminal run sequence + the exact seed data (accounts, coupons, prices) you'll reference in boundary cases.

**§3 Lecture-review checklist** — one-line self-checks for Domain Testing and BVA to confirm before designing.

**§4 The reusable 6-step method** + a **paste-ready AI prompt scaffold** (tasks a–d).

**§5 Per-feature playbooks** — the payload. For each of your four features I listed the variables, equivalence classes, boundaries, and
- **FR-05**: SQL/HTML payloads in search, product-id boundaries, single-`<h1>`/alt-text rules
- **FR-09**: the `total == min_order_amount` boundary (the `>` vs `>=` off-by-one — the single most important case), broken percent forr
- **FR-14**: empty-name acceptance + the non-admin-token authorization row (SEC-03)
- **FR-13**: dashboard may be *absent* on mobile (a finding in itself), plus the `canceled→delivered` revenue-inflation cross-link to FR

**§6–8** — execution/recording table format, **AI gap analysis** with the 3 required buckets (prompt quality / tool limitation / inherer

**§9 Agent Skills** — exact folder layout and required `SKILL.md` contents for `domain-testing` and `boundary-value-analysis`, plus a va

**§10 Definition of Done** + **Appendices** with ready-to-paste prompts and templates.

## Suggested next steps (pick any, I can do them now)
1. **Scaffold the two Agent Skills** (`.claude/skills/domain-testing/SKILL.md` + `boundary-value-analysis/SKILL.md`) so they're immediat
2. **Generate the first test-design file** (e.g., `FR-09-bva.md`) as a worked example following the workflow.
3. **Create the `docs/test-design/` + `docs/evidence/` folders** with the templates pre-filled.

Want me to proceed with any of these?
```

Interaction 2

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T15:53:35.435Z
- **AI response timestamp:** 2026-07-03T15:54:48.564Z
- **Tools used:** Write, Write

User Prompt

```
Scaffold the two Agent Skills (.claude/skills/domain-testing/SKILL.md + boundary-value-analysis/SKILL.md) so they're immediately reusabl
```

AI Output

```
Both Agent Skills are scaffolded and ready to use:
```

.claude/skills/ ├── domain-testing/SKILL.md └── boundary-value-analysis/SKILL.md

```
## What each contains

**`domain-testing`** — YAML frontmatter (`name` + trigger-rich `description`), then:
- **Guardrails**: spec-is-oracle, never rule PASS/FAIL, mark `spec-undefined`, demand concrete inputs.
- **5-step method**: extract rules → identify variables/state → partition into valid/invalid classes (incl. format, empty, injection/mar
- A **paste-ready LLM prompt scaffold**, a **worked EShop FR-14 example** (empty-name + non-admin-token matrix), and an output checklist

**`boundary-value-analysis`** — parallel structure:
- **4-step method**: find boundaries → decide inclusive vs exclusive from the spec operator → generate `{b−1, b, b+1}` (6-value set for
- **Worked EShop FR-09 example** built around the `total >= min_order_amount` inclusive edge — the case that catches a `>` vs `>=` off-b

## Design choices worth noting
- **Project-agnostic core, EShop-grounded examples** — the method transfers to any spec-driven feature, but each skill shows a concrete
- **The two skills cross-reference each other** (domain-testing hands ordered/length/date variables to BVA; BVA notes it avoids duplicat
- Both bake in the **spec-is-the-oracle guardrail** — essential for this deliberately-buggy SUT, where reasoning from code would validat

The skills become invocable in a new session. Per the workflow's §9, the last validation step is to **trial them on a 5th unselected fea

Want me to run that validation trial now, or generate the first real test-design file (e.g., `docs/test-design/FR-09-bva.md`) using thes
```

Interaction 3

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T15:56:54.715Z
- **AI response timestamp:** 2026-07-03T15:57:32.610Z
- **Tools used:** Write

User Prompt

```
run validation trial now
```

AI Output

```
Per §9 of the workflow, the validation trial applies both skills to a **fifth, unselected feature** and confirms they produce a usable c

One note before I execute: I'm running the trial as a **design-only** exercise (producing the case tables the skills generate). I'm deli

Validation trial complete — recorded at `docs/skill-validation-trial.md`.

## What the trial did
Applied **both skills to FR-01 (Registration)** — a feature outside your selected set — using only the spec + seed data as input, follow

**`domain-testing` produced 19 cases (DT-01…DT-19):** valid/invalid classes for every variable, plus the non-obvious classes the skill's

**`boundary-value-analysis` produced 10 cases (BVA-01…BVA-10):** the inclusive `≥ 8` password-length edge isolated as `{7→reject, 8→acce

## Verdict: PASSED

The trial confirms all eight criteria — usable table from spec alone, valid+invalid coverage, adversarial/interaction cases, correct inc

## Two things worth flagging
1. **No skill edits were needed** — the two improvement ideas that surfaced (enum-set edge cases; "required field ⇒ length-0 boundary")
2. The trial is **design-only by choice** — I noted in the doc that live execution was skipped to avoid the `require('./database')` rese

The skills are now proven reusable. Next logical step per the workflow would be generating the first *real* deliverable for a selected f
```

Interaction 4

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T16:01:19.868Z
- **AI response timestamp:** 2026-07-03T16:01:27.621Z
- **Tools used:** Skill, Skill, Write, Write

User Prompt

```
generate the test-design file for FR-05
```

AI Output

```
This matches the two testing skills directly — let me invoke them to follow their methods precisely.

Now the BVA skill:

Both skills loaded. Now I'll produce the two FR-05 test-design deliverables following the skill methods. Anchoring every *Expected* to `

The FR-05 test design is complete — two deliverables under `docs/test-design/`, produced by invoking the two skills:

## `FR-05-domain.md` — Domain Testing (26 cases)
- **Step 1** distills FR-05 into 8 atomic rules (R1–R8).
- **A. Matching classes** (DT-01…11): exact/partial/multi-match, case, Vietnamese diacritics, empty, whitespace — with `spec-undefined`
- **B. Adversarial classes** (DT-12…19): HTML markup, `<script>`, `img onerror`, `' OR '1'='1`, SQL comment, `DROP TABLE`, single-quote
- **C. State classes** (DT-20…22): all products, empty catalog, slow-network loading state.
- **D. Display invariants** (DT-23…26): `alt` text, `đ` thousands format, single `<h1>`, grid layout.
- Coverage check confirms every rule R1–R8 maps to ≥1 case.

## `FR-05-bva.md` — Boundary Value Analysis (15 cases)
Three genuine ordered/length dimensions (FR-05 has no money-range field, so I noted that explicitly):
- **B1 keyword length**: `0→show-all, 1, 255, 256, 10 000`.
- **B2 result-set size**: the `0→1→all` empty-state trigger.
- **B3 `LIKE` literal-vs-wildcard edge**: `%` and `_` as literals — plus match-position edges.

The doc calls out **BVA-10 (`search = %`)** as the single highest-value boundary: a name search must treat `%` as literal (0 results), b

## Design choices
- **Skill discipline held**: spec-as-oracle throughout, `spec-undefined` used honestly, **no PASS/FAIL** assigned, and clean division of
- **Execution-ready**: each file ends with execution notes (DB-reset points, capture raw JSON for injection/wildcard cases to catch leak

Both files are design-only, as the workflow prescribes — the PASS/FAIL verdicts get filled in when you execute against the running SUT.
```

Interaction 5

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T16:11:25.573Z
- **AI response timestamp:** 2026-07-03T16:11:31.672Z
- **Tools used:** Read, Bash, Bash, Bash, Write, Write

User Prompt

```
Scaffold the two Agent Skills (.claude/skills/domain-testing/SKILL.md + boundary-value-analysis/SKILL.md) so they're immediately reusabl
```

AI Output

```
Let me read the lecture PDF so I can align the skills with the course's specific domain-testing methodology and terminology.

The page-render path needs poppler. Let me try text extraction another way.

Good — this is the classic Myers/Kaner equivalence-partitioning material. Let me get the boundary-analysis portion and the coverage rule

I have the full lecture methodology now. The course (CSC13003 S04) uses the classic **4-step General Approach** with two details my curr

Both skills are re-scaffolded to align with the course lecture **CSC13003 — S04 Domain Testing**. I extracted the PDF via `pdftotext` (t

## What changed vs. the first draft

**`domain-testing/SKILL.md`**
- Reframed around the lecture's **"General Approach — 4 steps"**: (1) identify Input **& Output** variables → (2) equivalence classes →
- Added the missing **Output-variable partitioning** (success result *and* each distinct error message are their own classes — the lectu
- Added the lecture's **Step-2 guidelines table**: range → 1 valid + 2 invalid; set → 1 valid per member + 1 invalid; "must be" → 1 vali
- Added the crucial **Step-3 coverage rules** my first version lacked: **combine valid classes** into few dense tests, but give **each i
- Split output into two tables — **(a) complete partition set (EC1…)** then **(b) selected cases** — matching the lecture worksheet, plu

**`boundary-value-analysis/SKILL.md`**
- Reframed as **"Step 4 of Domain Testing"** with the lecture's rationale — the two edge-defect types it names: **inequality mis-specifi
- Adopted the lecture's **"up to ~9 test cases per partition"** model: `LB−1/LB/LB+1`, `UB−1/UB/UB+1`, **smallest/largest allowed via UI
- Tables now name the **operator** per boundary and use `LB/UB` probe labels.

## Preserved
The project-specific **spec-is-the-oracle guardrail**, `spec-undefined` handling, no-PASS/FAIL rule, and the EShop worked examples (FR-1

One consistency note: the `FR-05-domain.md` file I generated earlier uses the older single-table format. If you want, I can regenerate i
```

Interaction 6

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T16:14:25.683Z
- **AI response timestamp:** 2026-07-03T16:14:31.399Z
- **Tools used:** Skill, Skill, Write, Write

User Prompt

```
generate test design for FR-05
```

AI Output

```
Regenerating the FR-05 test design in the new lecture-aligned format (complete partition table + selected cases, output variables, valic

Both skills loaded. Regenerating the two FR-05 files in the lecture-aligned format.

Both FR-05 files regenerated in the lecture-aligned (CSC13003 S04) format.

## `FR-05-domain.md` — now follows the 4-step General Approach
- **Step 1** lists **Input, State, and Output** variables — the output side (grid, empty-state msg, loading indicator, safe-render, alt
- **Step 2** is the **complete partition table** (EC1–EC19 + EC-O1–EC-O7) with a best representative each, built via the lecture guidel
- **Step 3** applies the two **coverage rules**: 8 **valid-combined** tests (DT-01 packs exact-match + alt + price + `<h1>` outputs into

## `FR-05-bva.md` — now uses the ~9-value model
- Boundaries labeled with **operators and LB/UB** probe tags.
- Three dimensions: keyword length (`LB=0` show-all through the 10 000-char extreme), result-count (the `0→1` empty-state trigger, `UB=5
- Boundary rationale cites the lecture's two edge-defect types; **BVA-10 (`search = %`)** is flagged as the FR-05 analogue of FR-09's `

Both remain **design-only** with spec-as-oracle, `spec-undefined` used honestly, and no PASS/FAIL — verdicts fill in at execution (work

The two files are now consistent with the updated skills. The other three features (FR-09, FR-14, FR-13) still need their design files -
```

Interaction 7

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T18:38:47.412Z
- **AI response timestamp:** 2026-07-03T18:38:51.690Z
- **Tools used:** Skill, Skill, Write, Write

User Prompt

```
generate the test design for FR-09
```

AI Output

```
I'll generate the FR-09 (coupon) test design — the richest feature for both techniques, centered on the `>=` threshold boundary. Invokir

Both skills loaded. Writing the two FR-09 deliverables.

FR-09 test design complete — two lecture-aligned files under `docs/test-design/`.

## `FR-09-domain.md` — Domain Testing (17 cases)
- **Step 1** partitions the 5 conditions into inputs (`code`, `total_amount`, login), state (`type`, usage, expiry), and — critically —
- **Step 2**: complete partition table EC1–EC19 + output classes, with the coupon reference table.
- **Step 3** applies the coverage rules: 3 **valid-combined** positive paths (percent via SAVE10, fixed via BIGBUY, usage-below-max via
- **Interaction cases** DT-15/16/17 target the highest-risk defects: expiry-enforced-below-min (C2×C3), the **anonymous-user usage-cap b

## `FR-09-bva.md` — BVA (22 cases)
Four boundary families, each with operator + inclusivity stated:
- **B1** `total >= min` (C3, inclusive) — the `LB−1/LB/LB+1` straddle for **all three** coupon minimums (300k/500k/300k).
- **B2** `uses < max` (C5, **exclusive** at max) — the `<`-vs-`<=` edge for both caps (1 and 2).
- **B3** expiry `now < expired_at` (C2) — day before/of/after, using today = 2026-07-04.
- **B4** amount sign/type — 0, negative, extreme.

The doc flags **BVA-02/05/08 (`total == min`)** as the assignment's signature boundary: with C3 specified as `>=`, the coupon *must* app

Both stay design-only (spec-as-oracle, `spec-undefined` where the spec is silent on negatives/same-day-time/case, no PASS/FAIL) with exe

Two features designed (FR-05, FR-09); **FR-14** and **FR-13** remain. Want me to continue with FR-14 next?
```

Interaction 8

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T18:42:24.587Z
- **AI response timestamp:** 2026-07-03T18:42:27.299Z
- **Tools used:** Skill, Skill, Write, Write

User Prompt

```
generate the test design for FR-14
```

AI Output

```
I'll generate the FR-14 (Category CRUD) test design. This is where the authorization matrix (SEC-03) and empty-name validation are the k

Both skills loaded. Writing the FR-14 deliverables.

FR-14 test design complete — two lecture-aligned files under `docs/test-design/`.

## `FR-14-domain.md` — Domain Testing (14 cases)
- **Step 1** treats **actor auth** as a "set handled differently" (lecture guideline) → three classes: no-token / non-admin user token /
- **Step 2**: complete table EC1–EC15 with the authorization matrix front and center.
- **Step 3** coverage rules applied: 2 **valid-combined** tests (DT-01 packs valid name + diacritics + admin + list-view) and 8 **invali
  - **DT-03/04** — empty and whitespace-only name (FR-14 "name required").
  - **DT-05/07** — non-admin *user* token on POST/DELETE must return **403** (FR-12/SEC-03).
- **DT-14 interaction** (empty name × non-admin token) cleverly isolates *ordering*: whether the auth gate fires before validation — a 4
- Three honest **`spec-undefined` probes** (uniqueness, delete-with-products referential integrity, numeric-only name) — run them, but t

## `FR-14-bva.md` — BVA (14 cases)
- **B1** the empty→non-empty **`name` length edge** (`0→reject, 1→accept`, + whitespace-trims-to-0) — flagged as FR-14's highest-value b
- **B2** upper length (255/256/10 000) marked `spec-undefined` (no max stated).
- **B3** category `id` on DELETE: `LB−1=0`, `LB=1`, `UB=3`, `UB+1=4`, far-extreme, and non-numeric.
- All name-length cases run under an **admin** token so a length result can't be masked by an auth rejection (keeping the two concerns c

Both design-only, spec-as-oracle, no PASS/FAIL, with execution notes on the destructive-delete DB-reset discipline.

Three features done (FR-05, FR-09, FR-14); **FR-13 (Dashboard, Mobile)** is the last. Want me to generate it? Note it has a wrinkle wort
```

Interaction 9

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T18:46:54.087Z
- **AI response timestamp:** 2026-07-03T18:47:26.876Z
- **Tools used:** Skill, Skill, Write, Write

User Prompt

```
I want to generate test design for FR-10
```

AI Output

```
FR-10 (Order State Machine) is a strong fit for domain testing via a full state-transition matrix, with the actor dimension (user self-c

Both skills loaded. Writing the FR-10 deliverables — the domain file carries the full state-transition matrix; the BVA file frames the t

FR-10 test design complete — two files under `docs/test-design/`.

## `FR-10-domain.md` — Domain Testing as state-transition testing (15 cases)
- Frames the "input" as the triple **(current state, requested target, actor)** and builds the **full 5×5 transition matrix** as the par
- **Step 3** applies the coverage rules faithfully: **DT-01 is one happy-path walk** (pending→confirmed→shipping→delivered) that packs a
- The two planted-defect probes are called out explicitly:
  - **DT-07 — `canceled → delivered`** (final-state exit must be rejected).
  - **DT-12 — user self-cancels a `shipping` order** (spec: admin-only when shipping).
- Handles the genuine **spec ambiguity** honestly: the diagram shows `canceled` only from pending/confirmed, but the text implies admin

## `FR-10-bva.md` — BVA on the ordinal edges (15 cases)
Since FR-10 has no numeric range, I mapped the lecture's two edge-defect types to their **state-machine analogues** and tested three edg
- **B1 terminal-state edge** — last-non-final (accept) vs on-final (reject), covering both final states.
- **B2 user-cancel permission edge** — `pending`(inside) / **`confirmed`(LB, allowed)** / **`shipping`(LB+1, forbidden)** / `delivered`
- **B3 forward-adjacency edge** — step sizes −1/0/+1/+2/+3.

Highest-value: **BVA-04 (`canceled→delivered`)** and **BVA-08 (user cancels shipping)** — each an off-by-one that fails *only* at that b

Both design-only, spec-as-oracle, no PASS/FAIL, with fixture-building notes.

You now have designs for **FR-05, FR-09, FR-14, FR-10**. Note FR-10 wasn't in your original selected set (FR-05/09/14/13) — happy to hav
```

Interaction 10

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T18:49:53.182Z
- **AI response timestamp:** 2026-07-03T18:49:58.784Z

User Prompt

```
no I don't want that anymore
```

AI Output

```
Understood — no FR-13.

You currently have complete test designs for **FR-05, FR-09, FR-14, and FR-10** (domain + BVA each), plus the workflow doc, the two reus

Just so the deliverables map stays accurate: your original assignment brief listed **FR-05, FR-09, FR-14, FR-13**, and `docs/ai-testing

Want me to do that swap in the workflow doc, or leave it? And is there anything else you'd like next (e.g., the execution/results phase
```

Interaction 11

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T18:50:26.156Z
- **AI response timestamp:** 2026-07-03T18:50:29.507Z
- **Tools used:** Grep, Read, Read, Edit, Edit, Read, Read, Edit, Edit, Read, Edit, Read, Edit, Grep, Read, Edit

User Prompt

```
update please
```

AI Output

```
I'll update the workflow doc to swap FR-13 → FR-10. Let me find all the FR-13 references first.

Let me read the sections needing edits.

Now I'll make the edits.

Now the §5.4 playbook — replace the FR-13 Dashboard section with FR-10:

Now the two references in the gap-analysis / miss-patterns section (lines ~319, ~324):

Let me verify no stray FR-13/Dashboard/mobile references remain in the workflow doc.

The workflow doc is fully updated — FR-13 → FR-10 swap is complete with no stray references left. Changes made across `docs/ai-testing-w

| Location | Change |
|----------|--------|
| **Selected features** (§ header) | FR-13 Dashboard (Mobile) → **FR-10 Order state machine** |
| **Deliverables map** (D1/D2) | File lists now name all four explicitly, ending in `FR-10-domain.md` / `FR-10-bva.md` |
| **Setup** (§2 terminals) | Dropped the mobile/Expo terminal; Terminal 3 now covers FR-14 + FR-10; added a note that FR-10/FR-09 are dr
| **Execution tools** (§2) | Removed "Expo Go/emulator for mobile"; mapped Postman/cURL and browser tools to the actual four features |
| **§5.4 playbook** | Fully rewritten: FR-10 state-transition matrix (domain), the three ordinal edges (BVA: terminal-state, user-cancel
| **§7 gap analysis** | "FR-13↔FR-10 revenue inflation" → "FR-10 state-machine edges"; miss-patterns list now cites the final-state leak

Everything now consistently reflects **FR-05, FR-09, FR-14, FR-10** — matching your four generated test-design file pairs under `docs/te

One thing I left untouched: `docs/skill-validation-trial.md` still uses FR-01 as its trial feature — that's intentional and correct (the
```

Interaction 12

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T18:53:46.592Z
- **AI response timestamp:** 2026-07-03T18:54:06.610Z
- **Tools used:** Read, Read, Write

User Prompt

```
now I want to accumulate all test design into 1 main report
```

AI Output

```
I'll build a single consolidated report. Let me read all eight test-design files to accumulate their tables accurately.

I have all eight designs (FR-05 read fresh; FR-09/FR-14/FR-10 verbatim from when I authored them this session). Now assembling the consc

Done — the consolidated master report is at `docs/test-design-report.md`. It accumulates all four features' domain + BVA designs into or

## Structure
1. **Header + oracle discipline** — spec-as-oracle, `spec-undefined`/`spec-ambiguous`, no PASS/FAIL in design.
2. **§1 Shared methodology** — the CSC13003 S04 material stated *once* (4-step domain approach, valid-combined/invalid-isolated coverage
3. **§2 Environment & seed data** — accounts, categories, products, the coupon table, reset discipline.
4. **§3 Coverage summary** — one table with per-feature case counts + the signature case, plus the consolidated hotspot list.
5. **Per-feature sections (FR-05, FR-09, FR-14, FR-10)** — each with its variables, full equivalence-class table, selected cases, and BV
6. **§4 Traceability matrix** — every spec rule (R-numbers, C1–C5, SEC-03/04/05, final states) → the cases covering it.
7. **§5 Results template** — the copy-per-row Actual/Status/Evidence format that turns this design doc into the execution record, with p

## Design choices
- **Single source of truth for the tables**; the eight `docs/test-design/FR-xx-*.md` files remain as fuller per-file references (each fe
- **Per-feature case IDs kept as-is** (DT-01, BVA-01…) since they're unambiguous under each feature heading; the traceability matrix and
- Dropped the repeated method preambles to keep it readable at ~130 cases.

One caveat worth noting: this report is a **hand-assembled accumulation**, so if you later edit an individual `FR-xx-*.md` file, you'll
```

Interaction 13

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T19:38:13.936Z
- **AI response timestamp:** 2026-07-03T19:38:30.448Z

User Prompt

```
Generate an AI usage report for the current conversation.



For the most recent interaction, output:



# AI Usage Log



- AI Tool Name: Claude Code

- Date and Time: {{current_datetime}}



## User Prompt

```text

{{user_prompt}}
```
```

AI Output

```
{{assistant_response}}
```

Do not summarize. Include the complete prompt and complete response.

```
### AI Output
```text
# AI Usage Log

- AI Tool Name: Claude Code
- Date and Time: 2026-07-04

## User Prompt
```text
now I want to accumulate all test design into 1 main report
```
```

# AI Output

```
I'll build a single consolidated report. Let me read all eight test-design files to accumulate their tables accurately.

[Tool calls: Read FR-05-domain.md and FR-05-bva.md in parallel; the FR-09, FR-14, and FR-10 file contents were already present verbatim

I have all eight designs (FR-05 read fresh; FR-09/FR-14/FR-10 verbatim from when I authored them this session). Now assembling the consc

[Tool call: Write docs/test-design-report.md — a consolidated master report accumulating all four features' domain + BVA designs, 130 de
- Header + oracle discipline (spec-as-oracle, spec-undefined/spec-ambiguous, no PASS/FAIL in design)
- §1 Shared methodology (CSC13003 S04: 4-step domain approach, valid-combined/invalid-isolated coverage rules, BVA edge model + two edge
- §2 Environment & seed data (accounts, categories, products, coupon table, reset discipline)
- §3 Coverage summary & highest-value cases (per-feature counts + signature case + hotspot list)
- Per-feature sections FR-05 / FR-09 / FR-14 / FR-10, each with variables, full equivalence-class table, selected cases, BVA table, and
- §4 Traceability matrix (spec rule → cases)
- §5 Results-recording template with execution prerequisites]

Done — the consolidated master report is at `docs/test-design-report.md`. It accumulates all four features' domain + BVA designs into or

Structure
1. **Header + oracle discipline** — spec-as-oracle, `spec-undefined`/`spec-ambiguous`, no PASS/FAIL in design.
2. **§1 Shared methodology** — the CSC13003 S04 material stated *once* (4-step domain approach, valid-combined/invalid-isolated coverage
3. **§2 Environment & seed data** — accounts, categories, products, the coupon table, reset discipline.
4. **§3 Coverage summary** — one table with per-feature case counts + the signature case, plus the consolidated hotspot list.
5. **Per-feature sections (FR-05, FR-09, FR-14, FR-10)** — each with its variables, full equivalence-class table, selected cases, and BV
6. **§4 Traceability matrix** — every spec rule (R-numbers, C1–C5, SEC-03/04/05, final states) → the cases covering it.
7. **§5 Results template** — the copy-per-row Actual/Status/Evidence format that turns this design doc into the execution record, with p

Design choices
- **Single source of truth for the tables**; the eight `docs/test-design/FR-xx-*.md` files remain as fuller per-file references (each fe
- **Per-feature case IDs kept as-is** (DT-01, BVA-01…) since they're unambiguous under each feature heading; the traceability matrix and
- Dropped the repeated method preambles to keep it readable at ~130 cases.

One caveat worth noting: this report is a **hand-assembled accumulation**, so if you later edit an individual `FR-xx-*.md` file, you'll
```
```

```
---

## Interaction 14
- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T19:39:59.489Z
- **AI response timestamp:** 2026-07-03T19:40:15.929Z

### User Prompt
```text
Generate an AI usage report for the current conversation.



For the most recent interaction, output:



# AI Usage Log



- AI Tool Name: Claude Code

- Date and Time: {{current_datetime}}



## User Prompt

```text

{{user_prompt}}
```

AI Output

```
{{assistant_response}}
```

Do not summarize. Include the complete prompt and complete response.

```
### AI Output
```text
# AI Usage Log

- AI Tool Name: Claude Code
- Date and Time: 2026-07-04

## User Prompt
```text
/ai-log
Generate an AI usage report for the current conversation.

For the most recent interaction, output:

# AI Usage Log

- AI Tool Name: Claude Code
- Date and Time: {{current_datetime}}

## User Prompt
```text
{{user_prompt}}
```
```

AI Output

```
{{assistant_response}}
```

Do not summarize. Include the complete prompt and complete response.

```
## AI Output
```text
# AI Usage Log

- AI Tool Name: Claude Code
- Date and Time: 2026-07-04

## User Prompt
```text
now I want to accumulate all test design into 1 main report
```

AI Output

```
I'll build a single consolidated report. Let me read all eight test-design files to accumulate their tables accurately.

[Tool calls: Read FR-05-domain.md and FR-05-bva.md in parallel; the FR-09, FR-14, and FR-10 file contents were already present verbatim

I have all eight designs (FR-05 read fresh; FR-09/FR-14/FR-10 verbatim from when I authored them this session). Now assembling the consc

[Tool call: Write docs/test-design-report.md — a consolidated master report accumulating all four features' domain + BVA designs, 130 de
- Header + oracle discipline (spec-as-oracle, spec-undefined/spec-ambiguous, no PASS/FAIL in design)
- §1 Shared methodology (CSC13003 S04: 4-step domain approach, valid-combined/invalid-isolated coverage rules, BVA edge model + two edge
- §2 Environment & seed data (accounts, categories, products, coupon table, reset discipline)
- §3 Coverage summary & highest-value cases (per-feature counts + signature case + hotspot list)
- Per-feature sections FR-05 / FR-09 / FR-14 / FR-10, each with variables, full equivalence-class table, selected cases, BVA table, and
- §4 Traceability matrix (spec rule → cases)
- §5 Results-recording template with execution prerequisites]

Done — the consolidated master report is at `docs/test-design-report.md`. It accumulates all four features' domain + BVA designs into or

## Structure
1. **Header + oracle discipline** — spec-as-oracle, `spec-undefined`/`spec-ambiguous`, no PASS/FAIL in design.
2. **§1 Shared methodology** — the CSC13003 S04 material stated *once* (4-step domain approach, valid-combined/invalid-isolated coverage
3. **§2 Environment & seed data** — accounts, categories, products, the coupon table, reset discipline.
4. **§3 Coverage summary** — one table with per-feature case counts + the signature case, plus the consolidated hotspot list.
5. **Per-feature sections (FR-05, FR-09, FR-14, FR-10)** — each with its variables, full equivalence-class table, selected cases, and BV
6. **§4 Traceability matrix** — every spec rule (R-numbers, C1–C5, SEC-03/04/05, final states) → the cases covering it.
7. **§5 Results template** — the copy-per-row Actual/Status/Evidence format that turns this design doc into the execution record, with p

## Design choices
- **Single source of truth for the tables**; the eight `docs/test-design/FR-xx-*.md` files remain as fuller per-file references (each fe
- **Per-feature case IDs kept as-is** (DT-01, BVA-01…) since they're unambiguous under each feature heading; the traceability matrix and
- Dropped the repeated method preambles to keep it readable at ~130 cases.

One caveat worth noting: this report is a **hand-assembled accumulation**, so if you later edit an individual `FR-xx-*.md` file, you'll
```

Interaction 15

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T19:43:43.024Z
- **AI response timestamp:** 2026-07-03T19:44:50.056Z

User Prompt

```
Read the current Claude Code session log.



For each interaction extract:



- Name of AI Tool

- Timestamp

- User Prompt

- AI Output



Output as Markdown.
```

AI Output

```
# AI Usage Log — Full Session

**AI Tool:** Claude Code (model: Claude Opus 4.8)
**Project:** `eshop-sut` (software-testing SUT)
**Note on timestamps:** the session context exposes calendar dates but not wall-clock times. Interactions are listed in order; dates are

---

## Interaction 1
- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-03

### User Prompt
```text
Selected features:

FR-05: Product listing and search
FR-09: Discount coupons
FR-14: Category management (CRUD)
FR-13: Dashboard (Mobile)

For each of your selected features, complete the following tasks:

Domain Testing. With the support of AI tools, apply the domain testing technique to design a comprehensive set of test cases. You are er
Boundary Value Analysis. Likewise, apply the boundary value analysis technique to design a comprehensive set of test cases, adding more
AI gap analysis. If your AI tools miss any test cases or bugs, you must report them. Explain why the AI missed them — for example, due t
Bug reporting. Report all discovered bugs both in the Markdown report and on your group's GitHub Issues page. Remember to attach bug scr

Agent Skill: build Agent Skills that apply the domain testing and boundary value analysis techniques, so that they can be reused on addi

Please write a workflow in docs folder with AI collaboration to complete above tasks precisely
```
```

AI Output

Read `README.md`, `api_specification.md`, `backend/server.js`, `backend/database.js`, and grepped the mobile app for dashboard logic to ground the workflow in the real feature specs and planted-bug hotspots. Created **`docs/ai-testing-workflow.md`** — a full AI-collaboration workflow with: §0 deliverables map (D1–D7), §1 human/AI collaboration model + golden rules (spec-as-oracle), §2 environment setup + seed data, §3 lecture-review checklist, §4 reusable 6-step method + AI prompt scaffold, §5 per-feature playbooks (FR-05/09/14/13) with variables, partitions, boundaries and hotspots, §6 execution/recording, §7 AI gap analysis (3 buckets), §8 bug reporting (Markdown + `gh issue create` + screenshots), §9 Agent Skills blueprint, §10 Definition of Done, appendices of paste-ready prompts/templates. Offered three next steps (scaffold skills / generate first test-design file / create folders).

Interaction 2

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-03

User Prompt

```
Scaffold the two Agent Skills (.claude/skills/domain-testing/SKILL.md + boundary-value-analysis/SKILL.md) so they're immediately reusabl
```

AI Output

Created **`.claude/skills/domain-testing/SKILL.md`** and **`.claude/skills/boundary-value-analysis/SKILL.md`**, each with YAML frontmatter (name + trigger-rich description), guardrails (spec-is-oracle, never rule PASS/FAIL, mark `spec-undefined`), a step method, a paste-ready LLM prompt scaffold, an EShop-grounded worked example, and an output checklist. Noted the skills cross-reference each other and bake in the spec-as-oracle guardrail; suggested validating them on a 5th feature.

Interaction 3

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-03

User Prompt

```
run validation trial now
```

AI Output

Applied both skills to **FR-01 (Registration)** — a feature outside the selected set — as a design-only trial (avoiding a live backend run that would reseed the uncommitted `database.sqlite`). Created **`docs/skill-validation-trial.md`**: 19 domain cases (DT-01…19) and 10 BVA cases (BVA-01…10) including the inclusive `≥ 8` password-length edge and required-character count boundaries, an 8-criteria verdict table (**PASSED**), and notes that no skill edits were needed. Reported the outcome and offered to proceed to a real feature design.

Interaction 4

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-03

User Prompt

```
generate the test-design file for FR-05
```

AI Output

Invoked the `domain-testing` and `boundary-value-analysis` skills, then created **`docs/test-design/FR-05-domain.md`** (26 cases: matching/adversarial/state/display-invariant classes) and **`docs/test-design/FR-05-bva.md`** (15 cases across keyword length, result-set size, `LIKE` literal-vs-wildcard, match-position edges), flagging **BVA-10 (`search = %`)** as the highest-value boundary. Both design-only, spec-as-oracle, no PASS/FAIL.

Interaction 5

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-03

User Prompt

```
Scaffold the two Agent Skills (.claude/skills/domain-testing/SKILL.md + boundary-value-analysis/SKILL.md) so they're immediately reusabl
```

AI Output

Extracted the lecture PDF text via `pdftotext -layout` (page-render path lacked poppler). Rewrote both `SKILL.md` files to align with **CSC13003 — S04 Domain Testing**: added the lecture's 4-step "General Approach," **Output-variable partitioning**, the range/set/must-be/split guidelines table, the **Step-3 coverage rules** (combine valid classes, isolate each invalid class), the two-table output format, and the

lecture's Strengths/blind spots. BVA reframed as "Step 4 of Domain Testing" with the two edge-defect types (inequality mis-specified / boundary value mistyped) and the ~9-values-per-partition (LB/UB) model. Preserved the spec-as-oracle guardrails and EShop worked examples. Flagged that the earlier FR-05 file used the old format.

Interaction 6

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-03

User Prompt

```
generate test design for FR-05
```

AI Output

Re-invoked both skills and regenerated `FR-05-domain.md` (now with Input/State/Output variables, complete partition table EC1–EC19 + EC-O1…O7, valid-combined DT-01…08 / invalid-isolated DT-09…16 / interaction DT-17/18) and `FR-05-bva.md` (operators + LB/UB probe labels, three ordered dimensions, BVA-10 signature callout) in the lecture-aligned format.

Interaction 7

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-03

User Prompt

```
generate the test design for FR-09
```

AI Output

Invoked both skills; created `docs/test-design/FR-09-domain.md` (17 cases: 5-condition partitioning C1–C5 with 6 distinct error outputs, 3 valid-combined + 11 invalid-isolated + 3 interaction cases including anonymous-user usage-cap bypass and percent-formula check) and `docs/test-design/FR-09-bva.md` (22 cases across the `total >= min` threshold per coupon, `uses < max` exclusive edge, expiry date, and amount sign/type), flagging **BVA-02/05/08 (`total == min`)** as the signature `>=` boundary.

Interaction 8

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-03

User Prompt

```
generate the test design for FR-14
```

AI Output

Invoked both skills; created `docs/test-design/FR-14-domain.md` (14 cases: authorization matrix no-token/user-token/admin-token as the SEC-03 defect surface, empty/whitespace name isolation, ordering interaction DT-14, `spec-undefined` probes) and `docs/test-design/FR-14-bva.md` (14 cases: empty→non-empty name length edge, upper-length `spec-undefined`, category-id delete boundaries LB−1/LB/UB/UB+1 + non-numeric). Highlighted DT-05/07 (403) and BVA-01/02/04 (empty name).

Interaction 9

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-04

User Prompt

```
I want to generate test design for FR-10
```

AI Output

Invoked both skills; created `docs/test-design/FR-10-domain.md` (state-transition testing: full 5×5 transition matrix, 15 cases — one happy-path walk DT-01, isolated illegal transitions, `canceled→delivered` DT-07 and user-cancel-while-shipping DT-12 as key defects, `spec-ambiguous` DT-14, ownership interaction DT-15) and `docs/test-design/FR-10-bva.md` (15 cases mapping the two lecture edge-defects to state-machine analogues: terminal-state edge, user-cancel permission edge, forward-adjacency edge). Noted FR-10 wasn't in the original selected set and asked whether FR-13 was still wanted.

Interaction 10

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-04

User Prompt

```
no I don't want that anymore
```

AI Output

Acknowledged dropping FR-13. Summarized completed deliverables (FR-05/09/14/10 designs + workflow + skills + validation trial), noted `docs/ai-testing-workflow.md` still referenced FR-13, and offered to update the workflow doc to swap FR-13 → FR-10.

Interaction 11

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-04

User Prompt

```
update please
```

AI Output

Grepped for FR-13/Dashboard/mobile references and edited **`docs/ai-testing-workflow.md`**: swapped the selected-features list, deliverables map (D1/D2 file lists), setup terminals (removed mobile/Expo, mapped API-layer testing), the §5.4 playbook (rewrote as FR-10 state machine with matrix, ordinal BVA edges, fixture recipe), and the §7 gap-analysis/miss-pattern references. Verified no stray FR-13/Dashboard/mobile references remained; left `skill-validation-trial.md` (FR-01) intentionally unchanged.

Interaction 12

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-04

User Prompt

```
now I want to accumulate all test design into 1 main report
```

AI Output

Read the FR-05 files fresh (FR-09/14/10 content already in session) and created **`docs/test-design-report.md`** — a consolidated master report with **130 designed cases** (64 domain + 66 BVA): shared

methodology stated once, environment/seed data, coverage summary + hotspots, per-feature sections (variables, equivalence-class tables, selected cases, BVA tables, signature callouts; FR-10 keeps its matrix), a spec-rule→case traceability matrix, and a results-recording template. Noted it's a hand-assembled accumulation and offered a "last synced" note.

Interaction 13

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-04

User Prompt

```
/ai-log  — Generate an AI usage report for the current conversation. For the most recent interaction, output the User Prompt and AI Outp
```

AI Output

Produced an **AI Usage Log** for the most recent interaction (the "accumulate all test design into 1 main report" turn), including the complete user prompt and the complete assistant response.

Interaction 14

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-04

User Prompt

```
/ai-log  — Generate an AI usage report for the current conversation. For the most recent interaction, output the User Prompt and AI Outp
```

AI Output

Produced an **AI Usage Log** whose "most recent interaction" was the previous `/ai-log` command (nested prompt + response reproduced in full).

Interaction 15 (current)

- **AI Tool:** Claude Code
- **Timestamp:** 2026-07-04

User Prompt

```
/ai-log — Read the current Claude Code session log. For each interaction extract: Name of AI Tool, Timestamp, User Prompt, AI Output. Ou
```

AI Output

This full-session AI Usage Log (all interactions above), rendered as Markdown.

> **Accuracy note:** This log is reconstructed from the in-context conversation transcript, not from a raw on-disk session-log file with per-message epoch timestamps (that file isn't exposed to me here). Prompts are reproduced verbatim; AI outputs are faithful descriptions of each response and the files created/edited. If you need exact wall-clock timestamps, they'd come from the Claude Code session `.jsonl` under your project's history directory.

```
---

## Interaction 16
- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T19:49:14.327Z
- **AI response timestamp:** 2026-07-03T19:49:17.033Z
- **Tools used:** Bash, Bash

### User Prompt
```text
Find the current Claude Code session JSONL file.



Extract for every message:

- timestamp

- role

- content



Then generate an AI usage report containing:



- Name of AI Tool

- Timestamp

- User Prompt

- AI Output



Use timestamps from the JSONL file only.

Do not invent timestamps.
```
```

AI Output

```
I'll locate the actual session JSONL file and extract real timestamps from it — no invented times.
```

Interaction 17

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T19:50:47.553Z
- **AI response timestamp:** 2026-07-03T19:50:52.449Z
- **Tools used:** Glob, Bash, Bash, Bash, Bash, Bash

User Prompt

```
Find the current Claude Code session JSONL file.



Extract for every message:

- timestamp

- role

- content



Then generate an AI usage report containing:



- Name of AI Tool

- Timestamp

- User Prompt

- AI Output



Use timestamps from the JSONL file only.

Do not invent timestamps.
```

AI Output

```
Let me locate the session file using the Glob tool.

The current session is `02afd25e-...` (matches the scratchpad path). Let me parse it for real timestamps, roles, and content.

Structure confirmed — each message has a real `timestamp` and `message.{role,content}`. Let me extract genuine prompts and assistant out

The JSONL reveals the real timestamps are **all `2026-07-03` (UTC)** — the "date is now 2026-07-04" note was a harness injection, not a

I parsed the actual session JSONL and generated the report from real timestamps only. Full report written to **`docs/ai-usage-log.md`**

## Key correction from the JSONL
**Every recorded timestamp is `2026-07-03` (UTC).** My two earlier `/ai-log` runs labeled the FR-10-onward turns as `2026-07-04` — that

## Method
- Located the session file via Glob: `02afd25e-…jsonl` (matches this session's ID). Skipped the older `4e8e8c57-…jsonl`.
- Parsed 412 lines; kept `user`/`assistant` messages with their `timestamp` + `message.content`.
- **Filtered out non-human turns** so counts are accurate: Skill-injected `"Base directory for this skill:…"` messages, `<command-messag

## Extracted interactions (AI Tool = Claude Code for all)

| # | User prompt ts (UTC) | AI response ts | Prompt |
|---|---|---|---|
| 1 | 15:47:19.489Z | 15:47:28.960Z | Selected features / write workflow |
| 2 | 15:53:35.435Z | 15:54:48.564Z | Scaffold two Agent Skills |
| 3 | 15:56:54.715Z | 15:57:32.610Z | run validation trial now |
| 4 | 16:01:19.868Z | 16:01:27.621Z | generate the test-design file for FR-05 |
| 5 | 16:11:25.573Z | 16:11:31.672Z | Scaffold skills using S04 PDF |
| 6 | 16:14:25.683Z | 16:14:31.399Z | generate test design for FR-05 |
| 7 | 18:38:47.412Z | 18:38:51.690Z | generate the test design for FR-09 |
| 8 | 18:42:24.587Z | 18:42:27.299Z | generate the test design for FR-14 |
| 9 | 18:46:54.087Z | 18:47:26.876Z | I want to generate test design for FR-10 |
| 10 | 18:49:53.182Z | 18:49:58.784Z | no I don't want that anymore |
| 11 | 18:50:26.156Z | 18:50:29.507Z | update please |
| 12 | 18:53:46.592Z | 18:54:06.610Z | accumulate all test design into 1 main report |
| 13 | 19:38:13.936Z | 19:38:30.448Z | /ai-log (report, most recent) |
| 14 | 19:39:59.489Z | 19:40:15.929Z | /ai-log (report, most recent) |
| 15 | 19:43:43.024Z | 19:44:50.056Z | /ai-log (read session log) |
| 16 | 19:49:14.327Z | 19:49:17.033Z | /ai-log (find JSONL) — interrupted |
| 17 | 19:50:47.553Z | 19:50:52.449Z | /ai-log (find JSONL) — current |

The full report with complete User Prompt and AI Output blocks per interaction is in `docs/ai-usage-log.md`. Note the AI response timest
```

Interaction 18

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T20:04:44.988Z
- **AI response timestamp:** 2026-07-03T20:05:37.197Z
- **Tools used:** Write, Write

User Prompt

```
now build agent skill for generate test script from @docs/test-design-report.md and skill for running all test cases plus reporting ther
```

AI Output

```
The report is the perfect machine-readable source — each case already has ID, endpoint, request, precondition, and a spec-anchored Expe

Now the second skill — running the harness and reporting.

Both skills are built and immediately reusable.
```

.claude/skills/ ├── generate-test-script/SKILL.md └── run-tests-and-report/SKILL.md

```
## `generate-test-script` — report → executable harness
Turns `docs/test-design-report.md` into a **dependency-free Node.js harness** (uses built-in `fetch` + `node:test`-free runner, so nothi
- **Step 2 classifies every case** — `auto` (HTTP-observable: most of FR-09/10/14 + API side of FR-05), `manual` (FR-05 DOM/visual: `<h1
- **Step 3** gives a translation table from the report's Expected phrasing → concrete assertions (e.g. "Apply; discount 30,000; final 27
- **Step 4** maps each Precondition to a fixture (`resetDb`, `login`, `seedCouponUsage`, `makeOrder(state)`, `makeTempCoupon` for expiry
- Emits four files (`tests/harness.mjs`, `cases.mjs`, `runner.mjs`, `README.md`) with **full working templates** — including real genera

## `run-tests-and-report` — execute + report
Runs the harness and produces `docs/test-results.md`:
- Preflight (backend up, Node ≥18, records commit + DB-reset flag), run, aggregate, write report, route failures to bug reporting.
- Full report template: summary matrix, **candidate-bug table** (case → spec rule → Expected vs Actual → suspected defect), full results
- Feeds verified FAILs into `docs/bug-report.md` + `gh issue` drafts (workflow §8), cross-linking case ↔ bug ↔ issue.

## The critical design decision (baked into both skills' guardrails)
**Tests encode the spec, so on this buggy SUT a FAIL = a candidate bug, not a broken test.** Both skills forbid weakening assertions or

One caveat worth flagging: the DB-reset helper (`node database.js`) is **destructive** and is guarded behind `ESHOP_ALLOW_DB_RESET=1` —

Want me to now **invoke `generate-test-script`** to actually emit the `tests/` harness for all 130 cases, or leave the skills as reusabl
```

Interaction 19

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T20:17:25.206Z
- **AI response timestamp:** 2026-07-03T20:17:39.424Z
- **Tools used:** Read, Read, Write, Write

User Prompt

```
I want to fix FR-10 test design for mobile version
```

AI Output

```
FR-10 on mobile is really the **user-cancel** side of the state machine (per FR-20: mobile cancel only when `pending`/`confirmed`; no ac

The mobile code tells a clear story. Key findings from `frontend-mobile/App.js`:
- **`cancelOrder` (line 312)** calls `PUT /api/orders/:id/cancel` with the user's Bearer token — the *only* FR-10 action available on mo
- **The "Hủy đơn" button is rendered only when `o.status === "pending" || o.status === "confirmed"`** (line 961) — the mobile UI *correc
- `statusLabel` (line 331) maps all five states to Vietnamese; button uses the red danger style.

The crucial testing insight: the mobile UI's client-side gate **masks** the backend's planted bug (the API still lets you cancel a `ship

FR-10 mobile test design is in place — two new files grounded in the actual `frontend-mobile/App.js`:
- `docs/test-design/FR-10-mobile-domain.md` (13 cases)
- `docs/test-design/FR-10-mobile-bva.md` (10 cases)

## What makes the mobile version different (and why it needed fixing)
The original `FR-10-*.md` files were written for the **admin/API** view (full 5×5 transition matrix). Mobile is a fundamentally narrower

1. **Only the user self-cancel slice exists on mobile** — there's no admin panel on the device (`cancelOrder` → `PUT /api/orders/:id/can

2. **The two-layer principle** — the headline insight. The mobile UI renders "Hủy đơn" **only for `pending`/`confirmed`** (App.js:961),
   - **L1 (UI):** button hidden (MDT-04/05/06, MBVA-05).
   - **L2 (API):** direct `PUT /cancel` must still be rejected (MDT-07/08/09, MBVA-06).

   **MDT-11 / MBVA-06** make this explicit: *button hidden + API accepts = the UI is masking an FR-10 server defect.* A UI-only test wou

3. **Mobile-specific classes** added: Vietnamese `statusLabel` correctness (FR-21), red danger button (FR-21), empty-history state (FR-2

4. **BVA signature** is the `confirmed → shipping` permission flip, tested at both layers — **MBVA-05** (UI hides button, likely correct

Execution notes cover the mobile realities: set `API_URL` to the LAN IP, build states via the admin API, screenshot the order card for

Two things I did **not** touch, so flag them for your call:
- The **API/web `FR-10-domain.md` / `FR-10-bva.md`** are unchanged — mobile is additive, not a replacement. If you'd rather rename those
- The **master `test-design-report.md`** doesn't yet include the mobile FR-10 section or its 23 cases in the coverage totals. Want me to
```

Interaction 20

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T20:23:06.717Z
- **AI response timestamp:** 2026-07-03T20:23:36.298Z
- **Tools used:** Skill, Skill, Write, Write, Bash

User Prompt

```
regenerate @docs/test-design/FR-10-bva.md and @docs/test-design/FR-10-domain.md dedicate for mobile version using proper skills
```

AI Output

```
The intent is clear: make the **canonical** `FR-10-domain.md` and `FR-10-bva.md` the mobile-dedicated versions (superseding the interim

Both skills loaded. Now I'll overwrite the canonical FR-10 files with mobile-dedicated designs.

The canonical FR-10 files are now mobile-dedicated. That makes last turn's interim `FR-10-mobile-*.md` files redundant duplicates — I'll

Done. Both canonical FR-10 files are regenerated as **mobile-dedicated** designs (via the `domain-testing` and `boundary-value-analysis`

## What changed
The files were **rewritten from the admin/API view to the mobile view**:

- **Scope narrowed to the user self-cancel slice** — mobile has no admin panel, so admin forward transitions become *fixtures via the ad
- **New spec anchors**: FR-10 + **FR-20** (mobile cancel only when `pending`/`confirmed`) + FR-11 (own orders) + FR-21 (VN labels / red
- **The two-layer principle is the backbone**: since the app hides "Hủy đơn" for `shipping`/`delivered`/`canceled` (App.js:961), every t

## Domain file (`FR-10-domain.md`) — 14 cases
- Full partition table EC1–EC15 + outputs; valid-combined (DT-01/02/03) / invalid-isolated split across **L1** (DT-04/05/06 button hidde
- **DT-11** explicitly compares L1 vs L2 at `shipping` — the masked-bug gap. **DT-14** handles the diagram/text `spec-ambiguous` admin-d

## BVA file (`FR-10-bva.md`) — 10 cases
- The ordinal **cancel-permission edge** straddled at both layers: `pending`(LB–1) / `confirmed`(LB) / `shipping`(LB+1) / `delivered`(LB
- **Signature: BVA-05 + BVA-06** (`confirmed→shipping` flip) — BVA-05 (UI hides button, likely passes) is a *false positive* for FR-10;

Two follow-ups worth noting:
- The master **`test-design-report.md`** still contains the *old admin/API* FR-10 section and its counts. Want me to regenerate that sec
- The report's coverage table and traceability reference the old FR-10 case IDs (e.g. "DT-07 canceled→delivered", "BVA-04"); those mappi
```

Interaction 21

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T20:26:05.257Z
- **AI response timestamp:** 2026-07-03T20:27:06.221Z
- **Tools used:** Edit, Edit, Edit, Edit, Edit, Edit, Grep

User Prompt

```
regenerate and resync
```

AI Output

```
Resyncing the master report to the new mobile FR-10 design. New per-feature counts: FR-05 18/15, FR-09 17/22, FR-14 14/14, **FR-10 (mobi

Now the main FR-10 feature section — replacing the entire admin/API state-machine block with the mobile design.

Now the §4 traceability FR-10 rows.

Let me verify no stale references remain (old total, old FR-10 admin-transition framing).

Report fully resynced. The remaining `BVA-11..15` hits are all legitimate (they belong to FR-05, FR-09, and FR-14 tables, which still ha

## Summary of the resync to `docs/test-design-report.md`

| Location | Change |
|----------|--------|
| **Selected features** | FR-10 → "Order cancellation state machine — **Mobile**" |
| **§2 Environment** | FR-10 moved from Admin `:5174` to the **Mobile app** (Expo, LAN IP); notes UI-layer (L1) + API-layer (L2) testing
| **§3 Coverage table** | FR-10 row: endpoint `PUT /api/orders/:id/cancel` (+ admin fixtures), **14 domain / 10 BVA**, signature = DT-07
| **Totals** | **63 domain + 61 BVA = 124** (was 64/66/130) |
| **Hotspots** | user-cancel-while-shipping now notes the mobile UI gate can **mask** the server defect |
| **FR-10 feature section** | Fully rewritten: mobile scope, two-layer principle, EC1–EC15 + outputs, 14 selected cases (L1 button-hidde
| **§4 Traceability** | FR-10 rows remapped to the new mobile case IDs (FR-20/FR-10/FR-11/FR-21/FR-24 anchors, L1+L2) |

The master report is now internally consistent with the regenerated `FR-10-domain.md` / `FR-10-bva.md`, and the redundant `FR-10-mobile

One downstream note: if you later run the `generate-test-script` skill, the FR-10 cases now split into **L1 (manual/UI on device)** and
```

Interaction 22

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T20:36:14.541Z
- **AI response timestamp:** 2026-07-03T20:36:17.135Z
- **Tools used:** Skill, Write, Write, Write, Write, Bash, Edit, Bash

User Prompt

```
generate test scripts using proper skill
```

AI Output

```
I'll invoke the `generate-test-script` skill to build the harness from the master report.

Skill loaded. Now emitting the four harness files covering all 124 cases (84 auto / 22 manual / 18 probe), with deterministic fixtures

Now the case registry — all 124 cases:

Now the runner and the README:

Let me syntax-check the three `.mjs` files (no backend needed) to confirm the harness is valid.

One missing brace in the `cancel` arrow factory. Fixing:

The harness is generated and verified. All three files pass `node --check`, `cases.mjs` imports cleanly, and the registry has **exactly

## What was produced (`tests/`)
| File | Contents |
|------|----------|
| `harness.mjs` | `api`/`login`/`resetDb` + fixtures: `makeOrder(state)`, `makeTempCoupon`, `seedCouponUsage`, `makeForeignOrder`, `date
| `cases.mjs` | **124 cases** — 84 auto / 22 manual / 18 probe |
| `runner.mjs` | executes → `tests/results.json` + per-feature tally |
| `README.md` | run instructions, env vars, determinism notes, hotspot-FAIL list, manual checklist |

**Verified counts** (matches report §3): FR-05 33 · FR-09 39 · FR-14 28 · FR-10 24 = **124**, all IDs unique.

## Key skill-guided decisions
- **Classification per the skill:** `auto` = HTTP-observable; `manual` = device/DOM (FR-05 display invariants, **all FR-10 L1 button-vis
- **Tests encode the spec, so FAIL = candidate bug** — assertions were *not* weakened to match the buggy implementation. The README list
- **Determinism without cross-case coupling:** coupon usage/expiry boundaries use **throwaway coupons** (`makeTempCoupon` + runtime-rela
- **Honest adaptations flagged:** e.g. `FR05-BVA-07` uses a one-char-off term instead of destructively deleting the Keychron product — s
- **FR-10 mobile split respected:** L2 API-enforcement cases (`DT-07/BVA-06` — the masked-bug edge) are `auto`; L1 button-visibility cas

## To run (needs the backend up on :3000)
```bash
cd backend && node server.js        # reseeds DB on boot
node tests/runner.mjs               # writes tests/results.json
```
```

I did **not** execute it — running requires starting the backend, which reseeds `database.sqlite` (your working tree has an uncommitted copy). When you're ready, say the word and I'll invoke the **run-tests-and-report** skill to start the backend, run the suite, and produce `docs/test-results.md` with the candidate-bug table.

```
---

## Interaction 23
- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T23:47:56.177Z
- **AI response timestamp:** 2026-07-03T23:48:04.170Z
- **Tools used:** Skill, Bash, Bash, Bash, Bash, Bash, Write, Bash, Read, Bash, KillShell

### User Prompt
```text
run test cases and report using proper skill
```
```

AI Output

```
I'll invoke the `run-tests-and-report` skill. This requires starting the backend (which reseeds the DB on boot).

Starting Step 1 — preflight. Checking Node, backend deps, git commit, and whether the backend is already up.

Backend is down (Node v22 ✓, deps ✓, commit `6cdd618`). Starting `server.js` in the background — note this reseeds the DB on boot (expe

Backend is up. Step 2 — running the 124-case suite.

Clean run — 0 errors, **23 FAILs** (candidate bugs). Let me extract the FAILs and probe actuals from `results.json` to build the report

Results confirm the design's predictions precisely — the 23 FAILs collapse into **7 distinct planted defects**. Let me generate the repo

Report written (321 lines). Let me verify the top of it and stop the background backend.

The report looks correct. Now stopping the background backend.

Test run complete and reported. Backend stopped.

## Execution summary — `docs/test-results.md`

Ran all **124 cases** against a freshly-reseeded backend (commit `6cdd618`, Node v22, **0 ERRORs**):

| | PASS | FAIL | MANUAL | PROBE |
|---|---:|---:|---:|---:|
| **Total** | 61 | **23** | 22 | 18 |
| FR-05 | 18 | 5 | 7 | 3 |
| FR-09 | 26 | 9 | 1 | 3 |
| FR-14 | 9 | 7 | 1 | 11 |
| FR-10 (Mobile) | 8 | 2 | 13 | 1 |

The 23 FAILs collapse into **7 distinct planted defects** — every one predicted by the design's hotspot list, none passing unexpectedly

| Bug | Sev | Defect | Evidence |
|-----|-----|--------|----------|
| **BUG-01** | High | FR-09 `>` instead of `>=` — coupon rejected at exactly the minimum | `400 "chưa đủ giá trị tối thiểu"` at total==r
| **BUG-02** | High | FR-09 percent formula `total×(1−value)` → **negative discounts** | discount `-9,000,000` / `-89,999,999,991` (3 ca
| **BUG-03** | High | FR-09 coupon applies **anonymously** (C4 unenforced) | `200 success` with no token/user_id |
| **BUG-04** | Critical | FR-05 **SQL injection + LIKE-wildcard leak** | `' OR '1'='1` → 5 rows; `%` → 5 rows; `Mac_ook`/`Mac%Pro` → Mac
| **BUG-05** | Medium | FR-14 empty/whitespace category name accepted | `200` created (4 cases) |
| **BUG-06** | High | FR-14 **non-admin token accepted** (missing SEC-03 role check) | user token create/delete → `200` instead of `403`
| **BUG-07** | High | FR-10 **server lets a user cancel a `shipping` order** | `PUT /cancel` → `200` — the mobile UI hides the button (

**BUG-07 is the mobile two-layer payoff:** the L2 auto cases (`FR10-DT-07`, `FR10-BVA-06`) caught exactly what a UI-only test would have

Also surfaced in **probes** (spec-undefined, escalated for review): deleting a category with products **orphans** them (`FR14-DT-12`), a

The report contains the full 124-row results table, the grouped candidate-bug tables (Expected vs Actual), the 22-item manual checklist

## Next step
Per the skill's Step 5, the verified FAILs are ready to route to **`docs/bug-report.md`** + **GitHub Issues** (`gh issue create`) with c
```

Interaction 24

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T23:54:55.188Z
- **AI response timestamp:** 2026-07-03T23:55:32.675Z
- **Tools used:** Write, Write, Write, Write, Write, Write, Write, Write, Write, Bash

User Prompt

```
draft 7 bug reports plus GitHub issue commands
```

AI Output

```
I'll create the Markdown bug report, seven issue-body files, and a `gh` script. Writing the master `docs/bug-report.md` first.

Now the seven issue-body files and the `gh` script:

Now the `gh` script that creates labels and the 7 issues:

Everything is drafted. Let me check whether `gh` is available/authenticated and whether the repo has a GitHub remote, so you know what's

All 7 bug reports and the GitHub issue tooling are drafted.

## Files created

**`docs/bug-report.md`** — the master Markdown report: severity-ordered summary table + full write-up of all 7 bugs (spec ref, cURL step

**`docs/issues/bug-01…07.md`** — seven self-contained issue bodies for `gh --body-file`.

**`docs/issues/create-issues.sh`** — creates the labels (idempotent) and all 7 issues.

## The 7 bugs

| # | Sev | Title | Cases |
|---|-----|-------|-------|
| BUG-04 | **Critical** | FR-05 SQL injection & LIKE-wildcard leak | 5 |
| BUG-01 | High | FR-09 coupon rejected at exactly min (`>` vs `>=`) | 5 |
| BUG-02 | High | FR-09 percent formula → negative discount | 3 |
| BUG-03 | High | FR-09 coupon applies without login (C4) | 1 |
| BUG-06 | High | FR-14 non-admin token accepted (SEC-03) | 3 |
| BUG-07 | High | FR-10 user cancels `shipping` order via API | 2 |
| BUG-05 | Medium | FR-14 empty/whitespace name accepted | 4 |

## Good news on tooling
`gh` is **installed and authenticated** (as `linhkhoi1309`), and the remote is **`github.com/linhkhoi1309/eshop-sut`** — so the script i

```bash
bash docs/issues/create-issues.sh
```
```

Two things before you run it

1. **Creating issues is an outward-facing, public action** — I've drafted everything but haven't run it. Say the word and I'll execute the script (it'll create 7 real issues on your repo), or you can run it yourself.
2. **Screenshots are required by the assignment**, and `gh` can't attach images. Each issue body links `docs/evidence/bug-0X.png`. You'll need to either commit those PNGs under `docs/evidence/` or drag them into each issue on github.com after creation. The API bugs (01–07 except the manual ones) can be screenshotted from the cURL/JSON output; **BUG-07's** shipping-cancel and any FR-05/FR-10 UI aspects are best captured from Postman / the device.

Want me to run `create-issues.sh` now, or leave it for you to execute?

```
---

## Interaction 25
- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-03T23:59:15.467Z
- **AI response timestamp:** 2026-07-03T23:59:20.279Z
- **Tools used:** Bash, Bash, Bash, Edit, Bash, Edit

### User Prompt
```text
run create-issues.sh now
```
```

AI Output

```
Running it now — this creates 7 real issues on `github.com/linhkhoi1309/eshop-sut`.

Label creation was silently swallowed by `|| true`, so the issues referenced labels that don't exist, and `set -e` stopped before any is

`gh` is defaulting to the upstream `ttbhanh/eshop-sut` (this repo is a fork), which you can't write to. Your origin is `linhkhoi1309/esh

You have ADMIN on the target repo but **Issues are disabled** there, and `gh` defaults to the upstream fork parent. I'll enable Issues a

All 7 issues created on `github.com/linhkhoi1309/eshop-sut`. Let me record the URLs in the bug report's summary table.

All 7 GitHub issues are live and cross-linked back into the bug report.

## What happened
The first run failed because this repo is a **fork of `ttbhanh/eshop-sut`**, so `gh` defaulted issue/label operations to the upstream (w

## Issues created — `github.com/linhkhoi1309/eshop-sut`
| # | Issue | Bug |
|---|-------|-----|
| [#1](https://github.com/linhkhoi1309/eshop-sut/issues/1) | [FR-05] SQL injection & LIKE-wildcard leak | BUG-04 (Critical) |
| [#2](https://github.com/linhkhoi1309/eshop-sut/issues/2) | [FR-09] Coupon rejected at exactly min (`>` vs `>=`) | BUG-01 |
| [#3](https://github.com/linhkhoi1309/eshop-sut/issues/3) | [FR-09] Percent formula → negative discount | BUG-02 |
| [#4](https://github.com/linhkhoi1309/eshop-sut/issues/4) | [FR-09] Coupon applies without login | BUG-03 |
| [#5](https://github.com/linhkhoi1309/eshop-sut/issues/5) | [FR-14] Non-admin token accepted (SEC-03) | BUG-06 |
| [#6](https://github.com/linhkhoi1309/eshop-sut/issues/6) | [FR-10] User cancels shipping order | BUG-07 |
| [#7](https://github.com/linhkhoi1309/eshop-sut/issues/7) | [FR-14] Empty/whitespace category name | BUG-05 |

Each has labels (`bug` + `FR-xx` + `severity:*`), the spec ref, reproducible cURL steps, Expected vs actual response, and case traceabil

## One remaining action — screenshots (required by the assignment)
The issue bodies reference `docs/evidence/bug-0X.png`, but no images are attached yet (`gh` can't upload images). To complete the requir
- **drag a screenshot** into the issue on github.com (easiest — uploads to GitHub's CDN), or
- commit the PNGs under `docs/evidence/` and push (the relative links will then resolve).

Capture suggestions: BUG-01/02/03/04/05/06 from the cURL/Postman JSON responses; **BUG-07** from the Postman sequence (or the mobile orc
```

Interaction 26

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T00:13:22.671Z
- **AI response timestamp:** 2026-07-04T00:14:32.863Z

User Prompt

```
are there more bugs
```

AI Output

```
Yes — quite a few. The 7 filed issues are only the **API-observable bugs the automated suite caught within my 4 selected features**. Her

## 1. Within my 4 features — candidates my auto-suite flagged but didn't file
These came back as **PROBE/MANUAL** (not auto-FAIL), so I left them out pending judgement:

- **FR-14 orphans products on category delete** — `DELETE /api/categories/1` (a category with products) returns `200` and leaves those p
- **FR-14 silent no-op delete** — deleting a non-existent/non-numeric id returns `200 "Category deleted"` (`FR14-DT-08`, `FR14-BVA-14`)
- **22 manual UI cases never executed** — FR-05 safe-render/XSS (`<script>` reflected), single `<h1>`, `alt` text, `đ` format, loading/e

## 2. ⚠️ A gap I introduced — the FR-10 **admin-side** bug is now untested
When we refocused FR-10 to the **mobile user-cancel** slice, the tests stopped covering admin transitions. But `server.js` still has a p

## 3. Beyond my 4 features — the code is full of other planted defects
Visible in `server.js`/`database.js` (likely your teammates' scope, but real):

| Spec | Defect in code |
|------|----------------|
| FR-02 | failed-login counter increments by **+2** (`login_attempts + 2`), not 1; lockout is **180s**, spec says 30s |
| FR-03 / SEC-07 | reset OTP is **4 digits** (`1000–9999`), spec requires 6; no expiry |
| FR-04 / **SEC-06** | `PUT /users/me` lets the client change its own **`role`** → privilege escalation to admin |
| FR-01 / SEC-01 | register does **no** validation (password strength, email format, confirm-match); `users.email` has **no UNIQUE** cor
| FR-06 | `GET /products/:id` on a missing id returns `{}` **200** (not 404); even-numbered ids return `price` as a **string** |
| FR-08 | `checkout` trusts the client-supplied `total_amount` (backend never recomputes) |
| **SEC-03** | the missing admin-role check isn't just categories — **every** `/api/admin/*` route + coupon CRUD + import all rely on `a
| FR-19 | `DELETE /admin/users/:id` has no "can't delete yourself" guard, and no role check |
| FR-23 | mobile logout button is labeled **"Thoát"** (App.js:941), spec requires **"Đăng xuất"** |

## Suggested next actions
1. **File the FR-10 admin `canceled→delivered` bug** (tier 2) — it's in-scope for FR-10 and I have the case ready; I'd add an admin-tran
2. **File the FR-14 orphan-delete bug** (tier 1) — strong, API-reproducible.
3. **Run the 22 manual cases** against the web/mobile UI to confirm the FR-05 XSS/`<h1>`/alt and FR-10 button-visibility bugs, then file

Tier 3 is real but likely other group members' assigned features — want me to note them in `docs/bug-report.md` as "out-of-scope, observ

Want me to add auto-cases for #1 and #2 to the harness, run them, and file the issues?
```

Interaction 27

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T00:16:10.271Z
- **AI response timestamp:** 2026-07-04T00:17:08.770Z
- **Tools used:** Edit, Bash, Bash, Bash, Bash, Write, Write, Write, Bash, Edit, Edit, KillShell

User Prompt

```
auto cases for #1 and #2 to the harness run them and file the issues
```

AI Output

```
On it. First I'll add the auto-cases to the harness: the FR-10 admin-side final-state bug (the gap the mobile rewrite left) and the FR-1

Syntax check, then start the backend and run just the two supplemental features.

Both new bugs confirmed (controls pass, so the harness logic is sound):
- **FR10-ADM-01** → `200` — a **canceled order resurrected to delivered** (the admin final-state bug the mobile rewrite left uncovered)
- **FR14-INTEG-01** → product **orphaned** (references category_id 4 which no longer exists).

Writing the two issue bodies:

Now the script for the two new issues, and I'll run it:

Both filed. Now updating `docs/bug-report.md` — adding the two rows, the full sections, and removing the orphan item from the probe-find

Stopping the backend now that the run is done.

Done — both bugs added to the harness, executed (confirmed FAIL with passing controls), and filed.

## Two new auto-cases added to `tests/cases.mjs` (now 128 cases)
| Case | Result | Meaning |
|------|--------|---------|
| `FR10-ADM-01` | **FAIL** (`200`) | canceled → delivered accepted — the bug |
| `FR10-ADM-02` | PASS (`400`) | control: delivered → shipping correctly rejected |
| `FR10-ADM-03` | PASS (`200`) | control: pending → confirmed correctly accepted |
| `FR14-INTEG-01` | **FAIL** | product orphaned (references a deleted category) |

The passing controls matter: they prove the state machine *is* otherwise enforced, so `FR10-ADM-01` is a genuine planted carve-out, not

## Two new GitHub issues filed
- **[#8](https://github.com/linhkhoi1309/eshop-sut/issues/8)** — [FR-10] Admin can move a canceled order to delivered (final-state viola
- **[#9](https://github.com/linhkhoi1309/eshop-sut/issues/9)** — [FR-14] Deleting a category orphans its products — **Medium**

## Docs updated
`docs/bug-report.md` now has **9 bugs** — added BUG-08 and BUG-09 with full write-ups and issue links, and I promoted the orphan finding

The backend has been stopped. Two honesty notes on BUG-09: the spec basis is a bit indirect — FR-14 itself doesn't mandate cascade behav

Still outstanding if you want to keep going: the **22 manual UI cases** (FR-05 XSS-render/`<h1>`/alt, FR-10 mobile button-visibility) ar
```

Interaction 28

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T00:25:29.512Z
- **AI response timestamp:** 2026-07-04T00:25:54.246Z
- **Tools used:** Glob, Read, Grep, Write, AskUserQuestion, Write, Write, Write, Bash, Edit, Edit

User Prompt

```
manual UI verification next
```

AI Output

```
Real talk on constraints first: this environment has no browser-automation or screenshot capability, so I can't literally click through

`Home.jsx` is revealing — several FR-05 UI violations are now confirmed statically. Let me check the admin category rendering (FR-14 DT

Verified. Admin category name renders as plain escaped text (`{c.name || ""}`, App.jsx:321) → that one's a **PASS**. The mobile pieces l

Manual UI verification complete — documented in **`docs/manual-ui-verification.md`**. I verified all 22 by static source inspection (det

## Verdicts

**FR-05 Web (`Home.jsx`) — 8 FAILs**, a genuinely bad cluster:
- **Reflected XSS** — `Home.jsx:64` renders the search term via `dangerouslySetInnerHTML`. `<b>hi</b>` shows bold (not literal); **`<img
- **Display compliance** — empty `alt=""` (:82), **"VND" instead of `₫`** (:87), **two `<h1>`** on the page (:43 and :110), **no loading

**FR-14 Admin — PASS.** Category name renders as escaped text (`{c.name || ""}`, App.jsx:321) — no XSS. Honest negative result.

**FR-10 Mobile — all UI cases PASS.** Button-gating (only `pending`/`confirmed`), VN labels, empty state, and offline handling are all c

**Also spotted (bonus, FR-23):** the mobile logout button is labeled **"Thoát"** (App.js:941); spec requires **"Đăng xuất"**.

## Two new candidate bugs (not yet filed)
| # | Sev | Title |
|---|-----|-------|
| BUG-10 | **Critical** | Reflected XSS in web search (`dangerouslySetInnerHTML`) — R5/SEC-04 |
| BUG-11 | Low | FR-05 listing GUI non-compliance (2×`<h1>`, empty alt, "VND"≠`₫`, no loading/empty state) |

These are new bugs, and filing publishes them — so before I create issues #10/#11, how do you want to proceed?

Filing both. Creating the issue bodies:

The `severity:low` label doesn't exist yet — creating it, then running the script.

Both filed. Updating `docs/bug-report.md` with the two new rows and full sections.

Manual UI verification done and both bugs filed.

## Verification outcome (`docs/manual-ui-verification.md`)
All 22 manual cases verified by static source inspection with `file:line` evidence:
- **FR-05 Web — 8 FAILs** (the reflected XSS + five display-compliance breaches)
- **FR-14 Admin — PASS** (category name escaped)
- **FR-10 Mobile — all PASS** (UI layer correct; the FR-10 defect is purely server-side = BUG-07)
- Plus the bonus **FR-23** "Thoát" logout-label observation

## Two new issues filed
- **[#10](https://github.com/linhkhoi1309/eshop-sut/issues/10)** — [FR-05] Reflected XSS in web search — **Critical**
- **[#11](https://github.com/linhkhoi1309/eshop-sut/issues/11)** — [FR-05] Listing GUI non-compliance — **Low**

`docs/bug-report.md` now tracks **11 bugs** across 11 GitHub issues.

## Full bug ledger

| # | Sev | Feature | Issue |
|---|-----|---------|-------|
| BUG-04 | Critical | FR-05 SQL injection / wildcard leak | [#1](https://github.com/linhkhoi1309/eshop-sut/issues/1) |
| BUG-10 | Critical | FR-05 reflected XSS (search) | [#10](https://github.com/linhkhoi1309/eshop-sut/issues/10) |
| BUG-01 | High | FR-09 `>` vs `>=` at min | [#2](https://github.com/linhkhoi1309/eshop-sut/issues/2) |
| BUG-02 | High | FR-09 percent formula | [#3](https://github.com/linhkhoi1309/eshop-sut/issues/3) |
| BUG-03 | High | FR-09 coupon without login | [#4](https://github.com/linhkhoi1309/eshop-sut/issues/4) |
| BUG-06 | High | FR-14 non-admin token accepted | [#5](https://github.com/linhkhoi1309/eshop-sut/issues/5) |
| BUG-07 | High | FR-10 user cancels shipping (API) | [#6](https://github.com/linhkhoi1309/eshop-sut/issues/6) |
| BUG-08 | High | FR-10 canceled→delivered (admin) | [#8](https://github.com/linhkhoi1309/eshop-sut/issues/8) |
| BUG-05 | Medium | FR-14 empty category name | [#7](https://github.com/linhkhoi1309/eshop-sut/issues/7) |
| BUG-09 | Medium | FR-14 orphan-on-delete | [#9](https://github.com/linhkhoi1309/eshop-sut/issues/9) |
| BUG-11 | Low | FR-05 GUI non-compliance | [#11](https://github.com/linhkhoi1309/eshop-sut/issues/11) |

## Honesty notes
- The UI verdicts are from **static code analysis**, not a live browser — deterministic for these render/label/visibility behaviors, but
- **FR-10 mobile is the clean two-layer demonstration**: every UI case passed, yet FR-10 is still violated at the server (BUG-07) — a UI

Remaining if you want to continue: the **tier-3 out-of-scope defects** in other features (FR-02 counter/lockout, SEC-06 role escalation
```

Interaction 29

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T00:33:19.114Z
- **AI response timestamp:** 2026-07-04T00:34:34.425Z
- **Tools used:** Bash

User Prompt

```
are FR-05, FR-09, FR-10, FR-14 no more unfound bugs
```

AI Output

```
Good question — and the honest answer is **no, not exhaustively**. "All tests passed" only shows absence of bugs *in what was tested*. S

Confirmed the gap. Here's the honest per-feature status — and no, they're **not** fully exhausted: a code-level review turned up at leas

## Answer by feature

**FR-09 — likely complete.** The `apply-coupon` logic has exactly the 3 planted defects filed (BUG-01 `>=`, BUG-02 formula, BUG-03 no-au

**FR-14 — likely complete.** BUG-05/06/09 cover it. The extra `PUT /api/categories/:id` update endpoint (line 257) is untested but only

**FR-05 — search/listing path complete** (BUG-04 SQLi, BUG-10 XSS, BUG-11 GUI). Caveat: adjacent product bugs exist but belong to *other

**FR-10 — NOT complete. Found a new one:**

## Residual findings

| Finding | Scope | Confidence | Evidence |
|---------|-------|-----------|----------|
| **`PUT /api/admin/orders/:id/status` has no admin-role check** → any logged-in **non-admin can drive any order's state machine** (conf
| `GET /api/orders/:id` has **no auth at all** → anyone can read any order (total, shipping address) — IDOR | FR-11 (order viewing; adja
| Admin **cannot** cancel a `shipping` order (`shipping→canceled` not allowed for anyone) | FR-10 | **spec-ambiguous** — text says "admi
| `GET /api/admin/orders` no role check → any user lists all orders | FR-18 (out of scope) | High | `server.js:510` |
| `POST/PUT/DELETE /api/products` have **no auth at all** | FR-15 (out of scope) | High | `server.js:167/179/191` |
| `/api/products/:id`: missing id → `{}`+200 (not 404); even-id price → string | FR-06 (out of scope) | High | `server.js:159-164` |

## Bottom line
For your four features, the **one clearly in-scope unfiled bug is BUG-12: the missing admin-role check on the order-status endpoint** —

I can't *prove* zero remaining (tests show presence, not absence; subtle logic bugs could hide), but I've now reviewed every route + the

Want me to **add an auto-case for BUG-12, run it, and file it** (and optionally the `GET /orders/:id` IDOR if you count FR-11 as in-scop
```

Interaction 30

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T00:43:06.236Z
- **AI response timestamp:** 2026-07-04T00:44:07.838Z
- **Tools used:** AskUserQuestion, Read, Edit

User Prompt

```
## Probes (spec-undefined / ambiguous — record & decide)

| Case | Actual | Note |
|------|--------|------|
| FR05-DT-04 | 1 rows | spec-undefined — case-insensitivity of search |
| FR05-DT-18 | 0 rows | spec-undefined — accent-folding × case (`ban phim`) |
| FR05-BVA-05 | status 200, 0 rows | spec-undefined — 256-char (any cap?) |
| FR09-DT-14 | status 404 {"error":"Mã giảm giá không tồn tại hoặc đã bị vô hiệu hóa"} | spec-undefined — coupon code case-sensitivity
| FR09-BVA-17 | status 400 {"error":"Mã giảm giá đã hết hạn"} | spec-undefined — coupon expiring today (same-day time-of-day) |
| FR09-BVA-21 | status 400 {"error":"Đơn hàng chưa đủ giá trị tối thiểu 300,000 ₫ để áp dụng mã … | spec-undefined — negative total (-1
| FR14-DT-08 | status 200 {"message":"Category deleted"} | spec: reject/no-op on non-existent id 9999 (no-op vs error undefined) |
| FR14-DT-11 | status 200 | spec-undefined — duplicate name 'Laptop' |
| FR14-DT-12 | status 200 | spec-undefined — delete category id=1 that has products (integrity) |
| FR14-DT-13 | status 200 | spec-undefined — numeric-only name '12345' |
| FR14-BVA-05 | status 200 | spec-undefined — 255-char name (no max stated) |
| FR14-BVA-06 | status 200 | spec-undefined — 256-char name |
| FR14-BVA-09 | status 200 | spec: id 0 (LB-1) reject/no-op |
| FR14-BVA-10 | status 200 | spec: id -1 (negative) reject/no-op |
| FR14-BVA-12 | status 200 | spec: id just past max (UB+1) not found/no-op |
| FR14-BVA-13 | status 200 | spec: id far above range no-op |
| FR14-BVA-14 | status 200 | spec-undefined — non-numeric id 'abc' |
| FR10-DT-14 | status 200 | spec-ambiguous — any device path to cancel a shipping order? | ask me open questions consecutively to decide
```

AI Output

```
You want to adjudicate the probes. I'll walk you through them in clusters (a few rounds). Note: **FR14-DT-12** (delete category with pro

The user stepped away — I'll proceed with best judgment, record provisional decisions (clearly marked so they can override), and **not**

I'll rewrite the probe section with a **Decision** column recording my provisional adjudication.

I've adjudicated all 18 probes with best-judgment decisions (recorded in `docs/test-results.md`, clearly marked provisional so you can c

## How the 18 probes resolved

| Decision | Count | Cases |
|----------|------:|-------|
| **NOT-A-BUG** (spec-compliant/defensible) | 11 | search case-insensitive, 256-char search, code case-sensitivity, negative total, nume
| **SPEC-GAP** (not code bug; spec should define) | 3 | category name uniqueness (DT-11), max length (BVA-05/06), search accent-folding
| **CLARIFY** (genuinely ambiguous → ask spec owner) | 2 | same-day coupon expiry (BVA-17), admin-cancel-from-shipping (DT-14) |
| **FILED** | 1 | delete-with-products → already **BUG-09** |
| **Low-priority note** (file only if you want) | (5, within NOT-A-BUG) | invalid/non-existent delete returns `200 "deleted"` |

## The two items that genuinely need *your* input (I couldn't resolve from the spec)
1. **`FR09-BVA-17` — same-day expiry.** C2 says "current date must be *before* `expired_at`." The code parses `expired_at` as midnight,
2. **`FR10-DT-14` — admin cancel of a `shipping` order.** FR-10's **text** ("when shipping, only admin may act") implies admin *can* can

## My verdict on your original question
Net-net for the four features: the probes surfaced **no new confirmed code bug** beyond BUG-09 (already filed). The only outstanding *co

When you're back, tell me:
- **BUG-12** — verify + file? (I'd add an auto-case, run, and open the issue.)
- The **low-priority invalid-delete** note — file as a Low bug, or leave as an observation?
- The **2 CLARIFY items** — want them written up as spec-clarification notes (or filed as bugs if you consider the text authoritative)?

I re-ran none of the app for this (pure spec/code adjudication), so nothing was left running.
```

Interaction 31

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T00:52:41.096Z
- **AI response timestamp:** 2026-07-04T00:53:04.392Z

User Prompt

```
| FR09-DT-14 | 404 | code case-sensitivity (`save10`)? | **NOT-A-BUG** — case-sensitive codes are fine (web/mobile upper-case before ser
```

AI Output

```
Because to the **database**, `save10` and `SAVE10` are *not* the same string. The lookup is an exact-match SQL comparison:

```js
db.get("SELECT * FROM coupons WHERE code = ? AND is_active = 1", [code], ...)
```
```

In SQLite, the `=` operator on text uses **BINARY collation by default — it's case-sensitive**. The seeded row stores `code = 'SAVE10'` (uppercase, from `database.js`). So `WHERE code = 'save10'` matches **zero rows** → the handler returns `404 "Mã giảm giá không tồn tại hoặc đã bị vô hiệu hóa"`.

They look "the same" to a human (same coupon), but they're different byte strings to `=`. (Note this is the opposite of the `LIKE` used in product search — `LIKE` *is* case-insensitive for ASCII in SQLite, which is why `iphone` matched `iPhone` in FR05-DT-04. Different operators, different behavior.)

Why I still marked it NOT-A-BUG

Two reasons — but it's a judgment call:

1. **The spec (FR-09) never says coupon codes must be case-insensitive.** No rule is violated.
2. **Real users can't hit it through the UI** — both frontends normalize before sending. Mobile: `App.js:366` → `couponCode.trim().toUpperCase()`; the web checkout does the same. So a user typing `save10` actually sends `SAVE10`, which works.

The honest caveat

It *is* an inconsistency at the **raw API layer**: `POST /api/apply-coupon` with `save10` fails, even though the "same" coupon exists. If you think coupon codes *should* be case-insensitive (a reasonable robustness expectation, and it's inconsistent with the case-insensitive product search), this is a **Low-severity robustness gap** — the fix would be `WHERE code = ? COLLATE NOCASE`.

So: **not a spec violation, but a defensible-either-way call.** Want me to reclassify it from NOT-A-BUG to a **Low-priority note** (or file it as a Low bug)? I'd lean "note it, don't file" — but you have the context on how your grader treats API-vs-UI consistency.

```
---

## Interaction 32
- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T00:53:30.117Z
- **AI response timestamp:** 2026-07-04T00:53:42.451Z
- **Tools used:** Write, Bash, Bash, Edit, Edit, Edit, Edit

### User Prompt
```text
file it as low bug
```
```

AI Output

```
Filing it as **BUG-13** (Low). Creating the issue body:

Filed as issue **#12**. I'll keep numbering sequential (this is **BUG-12**), rename the file to match, and update the docs.

Now update the probe adjudication in `test-results.md` — flip FR09-DT-14 to FILED.

Filed as **BUG-12** → [issue #12](https://github.com/linhkhoi1309/eshop-sut/issues/12).

## What I did
- **Created** `docs/issues/bug-12.md` and opened issue #12 (labels `bug` / `FR-09` / `severity:low`), with the root cause (SQLite `WHERE
- **`docs/bug-report.md`** — added the BUG-12 row and full section (now **12 bugs**).
- **`docs/test-results.md`** — flipped the `FR09-DT-14` probe from NOT-A-BUG to **FILED → BUG-12**, and updated the Outcome summary.
- Kept numbering sequential (BUG-12 ↔ issue #12) and renamed the file to `bug-12.md` to match.

## Current bug ledger — 12 filed
| Sev | Count | Bugs |
|-----|------:|------|
| Critical | 2 | BUG-04 (SQLi), BUG-10 (XSS) |
| High | 6 | BUG-01, BUG-02, BUG-03, BUG-06, BUG-07, BUG-08 |
| Medium | 2 | BUG-05, BUG-09 |
| Low | 2 | BUG-11 (GUI), BUG-12 (coupon case) |

One consistency note so the numbering stays clean: the **order-status missing-role-check** I earlier called "BUG-12" in conversation is
```

Interaction 33

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T00:56:07.982Z
- **AI response timestamp:** 2026-07-04T00:56:20.031Z
- **Tools used:** Edit, Bash, Bash, Bash, Write, Bash, KillShell, Edit, Edit

User Prompt

```
file that one too
```

AI Output

```
Verifying first (I'll add the auto-case and run it before filing). Adding `FR10-SEC-01` to the harness:

Starting the backend to verify the case empirically.

Confirmed — a **non-admin token drove the order status to `confirmed` (200)**. Filing **BUG-13**.

Filed as issue **#13**. Stopping the backend and updating the docs.

Done — verified and filed as **BUG-13** → [issue #13](https://github.com/linhkhoi1309/eshop-sut/issues/13). Backend stopped.

## What happened
- Added auto-case `FR10-SEC-01` to the harness, ran it against a live backend → **FAIL: `200`** (a non-admin user token drove an order t
- Opened issue #13 (`bug`/`FR-10`/`severity:high`) with root cause (`server.js:525` uses `authenticateToken` only, no role check), the r
- Updated `docs/bug-report.md` (row + full section).

## Full bug ledger — 13 filed, all in-scope for FR-05/09/10/14

| Sev | Bugs (issues) |
|-----|---------------|
| **Critical** ×2 | BUG-04 SQLi (#1), BUG-10 XSS (#10) |
| **High** ×7 | BUG-01 `>=` (#2), BUG-02 formula (#3), BUG-03 anon coupon (#4), BUG-06 category role (#5), BUG-07 user-cancel-shipping
| **Medium** ×2 | BUG-05 empty name (#7), BUG-09 orphan delete (#9) |
| **Low** ×2 | BUG-11 GUI (#11), BUG-12 coupon case (#12) |

That clears the last **confirmed in-scope defect** I'd identified. Remaining open items are only:
- **2 CLARIFY questions** for the spec owner (same-day coupon expiry `BVA-17`; admin-cancel-from-shipping `DT-14`) — not bugs, need a sp
- **Screenshots** still to attach to all 13 issues (the assignment requires them; `gh` can't upload images).
- Tier-3 **out-of-scope** defects in other features (FR-01/02/03/04/06/08/15/18/23) — real, but likely teammates' scope.

Want me to draft the 2 spec-clarification write-ups, or help with the screenshot capture plan next?
```

Interaction 34

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T01:07:56.822Z
- **AI response timestamp:** 2026-07-04T01:08:51.248Z
- **Tools used:** Edit, Bash, Bash, Bash, Write, Bash, KillShell, Edit, Edit, Edit, Edit

User Prompt

```
| FR10-DT-14 | 200 | admin cancel a shipping order? | **CLARIFY** — FR-10 text ("admin may act" when shipping) conflicts with the diagra
```

AI Output

```
Good catch to flag — and let me be precise, because there's a subtlety in what `FR10-DT-14` actually tested. The probe was *labeled* "ac

The admin question is separate and needs its own check: does `PUT /api/admin/orders/:id/status {status:"canceled"}` work on a shipping c

Confirmed: admin cancel of a shipping order → **`400 "Invalid state transition from shipping to canceled"`**. Per your decision (text au

Filed as issue **#14**. Updating the docs.

Now updating the probe adjudication in `test-results.md` so the record reflects the decision.

Verified and filed as **BUG-14** → [issue #14](https://github.com/linhkhoi1309/eshop-sut/issues/14). Backend stopped.

## Important correction I made along the way
The `FR10-DT-14` probe was mislabeled — it *said* "admin cancel a shipping order" but the code actually ran a **user** cancel, so its `2
- **BUG-07** — *user* can cancel shipping (should be rejected, but returns 200) ✗
- **BUG-14** — *admin* cannot cancel shipping (should be allowed, but returns 400) ✗

## Docs updated
- `docs/bug-report.md` — BUG-14 row + section (**14 bugs** now).
- `docs/test-results.md` — `FR10-DT-14` probe flipped to **RESOLVED → BUG-14** (with the user-path/admin-path clarification); Outcome no

## Full ledger — 14 filed
| Sev | Bugs |
|-----|------|
| Critical ×2 | BUG-04, BUG-10 |
| High ×7 | BUG-01, 02, 03, 06, 07, 08, 13 |
| Medium ×3 | BUG-05, 09, **14** |
| Low ×2 | BUG-11, 12 |

## What's left
- **1 CLARIFY item** for the spec owner: `FR09-BVA-17` — same-day coupon expiry (date-vs-datetime). Not a code bug; needs a spec decisio
- **Screenshots** for all 14 issues (assignment requirement; `gh` can't attach them).
- Tier-3 **out-of-scope** defects in other features remain unaddressed (likely teammates').

FR-05/09/10/14 are now, to the best of a route-level + logic + UI review, fully mined for in-scope bugs.
```

Interaction 35

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T01:15:48.557Z
- **AI response timestamp:** 2026-07-04T01:16:21.818Z
- **Tools used:** Bash, Edit, Bash, Edit, Edit, Edit, Edit

User Prompt

```
revert bug 14 since this is a valid
```

AI Output

```
Reverting BUG-14 — you've decided the admin-cannot-cancel-shipping behavior is valid (diagram authoritative). I'll close the issue with

Now reverting the artifacts. Removing the `FR10-ADM-04` case, deleting the issue body, and reverting the docs.

Now reverting the `test-results.md` probe adjudication for FR10-DT-14 back to not-a-bug.

BUG-14 fully reverted.

## What I did
- **Closed issue [#14](https://github.com/linhkhoi1309/eshop-sut/issues/14)** as "not planned" with a comment explaining the reversal (c
- **Removed** the `FR10-ADM-04` case from `tests/cases.mjs` (syntax re-checked ✓) and deleted `docs/issues/bug-14.md`.
- **`docs/bug-report.md`** — removed the BUG-14 row and section (back to **13 bugs**).
- **`docs/test-results.md`** — flipped `FR10-DT-14` to **NOT-A-BUG** (diagram authoritative), and updated the Outcome summary.

## Net effect
`shipping → canceled` being disallowed for everyone is now recorded as **correct** per the diagram. The two FR-10 cancellation facts sta
- **BUG-07** (still valid) — a *user* can cancel a `shipping` order (should be rejected) ✗
- Admin can't cancel `shipping` — **intended** (diagram: `canceled` only from `pending`/`confirmed`) ✓

## Ledger — back to 13 filed
| Sev | Bugs |
|-----|------|
| Critical ×2 | BUG-04, BUG-10 |
| High ×7 | BUG-01, 02, 03, 06, 07, 08, 13 |
| Medium ×2 | BUG-05, 09 |
| Low ×2 | BUG-11, 12 |

Remaining open item is the single CLARIFY (`FR09-BVA-17` same-day expiry) plus screenshots for the 13 live issues. Anything else to adju
```

Interaction 36

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T01:22:56.943Z
- **AI response timestamp:** 2026-07-04T01:23:29.240Z
- **Tools used:** Edit, Edit, Bash

User Prompt

```
resync the @docs/bug-report.md
```

AI Output

```
Resyncing after the BUG-14 revert and the supplemental additions. The main drift: the summary table isn't cleanly severity-ordered (BUG

Let me verify the resync is clean — no BUG-14 remnants and 13 detail sections.

Resync complete and verified — **0 BUG-14 remnants**, **13 summary rows = 13 detail sections**, all issue links intact.

## What I fixed
1. **Summary table reordered** to proper severity order — BUG-10 (Critical) moved up next to BUG-04; the stranded Critical is gone. Orde
2. **Intro resynced** — dropped the stale wording, added the supplemental gap cases (`FR10-ADM-01`, `FR10-SEC-01`, `FR14-INTEG-01`) to t
3. **Removed the doubled `---`** left where the BUG-14 section was deleted.

## One remaining cosmetic note
The **detail sections** below the table are still in *discovery/filing* order (e.g., BUG-05 Medium sits between BUG-07 and BUG-08; BUG-1

The report is otherwise fully consistent with the current 13-issue state on GitHub (issue #14 closed as not-planned, not referenced).
```

Interaction 37

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T01:28:16.479Z
- **AI response timestamp:** 2026-07-04T01:28:19.161Z
- **Tools used:** Bash, Bash, Bash, Bash, Bash

User Prompt

```
run the web app for me to manual cross check
```

AI Output

```
I'll start the backend and the web app so you can manually cross-check the FR-05 UI findings. Both need to run; starting them in the bac

Deps present. Starting the backend (:3000) and the web dev server (:5173).

Both are running. Let me grab the Vite URL from the dev server output.

Both servers are live. Open **http://localhost:5173/** in your browser.

- **Backend API:** http://localhost:3000 (running, DB freshly seeded)
- **Web app:** http://localhost:5173 (Vite)
- Seeded logins if needed: user `test@eshop.com / Test1234!`, admin `admin@eshop.com / Admin123!`

## Manual cross-check guide — FR-05 bugs (open DevTools: F12)

**BUG-10 — Reflected XSS (Critical)** — the headline one:
1. In the search box type **`<b>hi</b>`** → submit. **Bug:** the "Kết quả tìm kiếm cho:" line shows **bold** "hi" (HTML rendered), not l
2. Now search **`<img src=x onerror=alert(1)>`** → submit. **Bug:** an **`alert(1)` popup fires** = live XSS. (A bare `<script>alert(1)<
3. Screenshot the alert box for the issue evidence.

**BUG-11 — GUI non-compliance (Low)** — on the home page:
- **(a) two `<h1>`** → DevTools Console, run `document.querySelectorAll('h1').length` → returns **2** ("Danh sách sản phẩm" + "Hiển thị
- **(b) empty alt** → inspect any product `<img>` → `alt=""` (should be descriptive).
- **(c) wrong currency** → product price reads "**… VND**", not "**₫**".
- **(d) no loading state** → DevTools → Network → throttle to "Slow 3G", search → no spinner appears.
- **(e) no empty-state** → search **`zzzzz`** (no matches) → you get a **blank area**, no friendly "no results" message.

**BUG-04 — SQL injection / wildcard leak (Critical)** — visible in the UI too:
- Search **`%`** → **all 5 products** returned (a literal `%` should match none).
- Search **`' OR '1'='1`** → **all 5** returned.
- Search **`Mac_ook`** → returns **MacBook Pro M3** (the `_` acted as a wildcard).

**Not-a-bug sanity checks** (should behave correctly):
- Search **`iphone`** (lowercase) → still finds iPhone (case-insensitive, FR05-DT-04 — fine).
- Search **`Mac`** → MacBook only.

## When you're done
Tell me and I'll shut down both background servers (they keep running until stopped). If any observation differs from what I predicted s
```

Interaction 38

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T01:32:26.401Z
- **AI response timestamp:** 2026-07-04T01:33:01.649Z
- **Tools used:** Read, Edit, Edit, Edit, Read, Edit

User Prompt

```
FR-05 bug  Khi không có kết quả tìm kiếm phải hiển thị thông báo empty state phù hợp.
```

AI Output

```
You've confirmed it in the browser — that's the **R7 / FR-24 empty-state** violation ("when a search returns no results, a suitable empt

Now the same update in `docs/bug-report.md`:

Confirmed and recorded. Let me also mark it browser-verified in the verification doc.

Confirmed and recorded — you've now **browser-verified** the FR-05 empty-state bug (R7 / FR-24).

**What you saw:** search `zzzzz` → 0 results but **no "no results" message**, just a blank area. That's the spec violation: *"Khi không

**Updated in 3 places (marked browser-verified):**
- `docs/bug-report.md` → BUG-11 finding (e) now names the no-search-results case + repro
- `docs/issues/bug-11.md` → same, with a browser-verified callout and the Vietnamese spec quote
- `docs/manual-ui-verification.md` → the empty-state row flipped to "FAIL — browser-verified"
- Traceability now cites `FR05-DT-07` (no-match search) alongside `FR05-DT-16` (empty catalog)

This stays under **BUG-11** (Low, GUI bundle) → [issue #11](https://github.com/linhkhoi1309/eshop-sut/issues/11). Screenshot the blank `

## Servers still running (for more cross-checking)
- Web http://localhost:5173 · Backend http://localhost:3000

Keep going — good next checks while you're in there:
- **BUG-10 XSS:** search `<img src=x onerror=alert(1)>` → alert pops (the big one to screenshot)
- **BUG-04 leak:** search `%` → all 5 products
- **BUG-11 others:** console `document.querySelectorAll('h1').length` → 2; price shows "VND"

Want me to **post the browser-verified note as a comment on issue #11**, **start the admin app (:5174)** for the FR-14 checks, or **stop
```

Interaction 39

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-04T01:36:15.231Z
- **AI response timestamp:** 2026-07-04T01:36:58.844Z
- **Tools used:** AskUserQuestion, Write, Write, Write, Write, Write, Bash, Edit, Edit, Edit, Bash

User Prompt

```
### Findings (all on `frontend-web/src/pages/Home.jsx`)
| # | Rule | Expected | Actual | Code |
|---|------|----------|--------|------|
| a | R8 | exactly **one** `<h1>` per page | **two** `<h1>` ("Danh sách sản phẩm" + "Hiển thị N sản phẩm") | `:43`, `:110` |
| b | R2 / FR-24 | product image has a **descriptive `alt`** | `alt=""` (empty) on every product image | `:82` |
| c | R3 / FR-21 | price uses **`₫`** + thousands separator | shows "`… VND`" (wrong currency symbol) | `:87` |
| d | R6 | **loading** state while fetching | no loading indicator; `fetchProducts` sets no loading flag | `:12-33` |
| e | R7 / FR-24 | **empty-state** message when a search returns **no results** (and when the catalog is empty) — "Khi không có kết quả
```

AI Output

```
Good question — it's a bug-granularity call. My take:

**Yes, I'd split them.** In testing practice you file **one issue per distinct defect** when each has an independent root cause and fix

Nuance worth noting: (b) empty `alt`, (c) "VND" vs `₫`, and (a) two-`<h1>` are trivial one-line fixes; (d) missing loading state and (e

Since this changes what I file (outward-facing), let me confirm the approach:

Splitting into 5. I'll repurpose issue #11 for finding (a) and open 4 new issues for (b)–(e). Writing the five issue bodies first.

Now update issue #11 (repurpose to finding a) and create the 4 new issues.

Issue #11 repurposed; 4 new issues created (**BUG-14 → #15, BUG-15 → #16, BUG-16 → #17, BUG-17 → #18**). Now resyncing `bug-report.md` -

Let me verify the split resynced correctly.

Split done and resynced — **17 summary rows = 17 detail sections**, all links correct.

## What changed
The BUG-11 GUI bundle is now **5 separate Low issues**, one per FR rule:

| Bug | Finding | Rule | Issue |
|-----|---------|------|-------|
| BUG-11 | Two `<h1>` on the page | R8 | [#11](https://github.com/linhkhoi1309/eshop-sut/issues/11) *(repurposed)* |
| BUG-14 | Empty image `alt` | R2/FR-24 | [#15](https://github.com/linhkhoi1309/eshop-sut/issues/15) |
| BUG-15 | "VND" instead of `₫` | R3/FR-21 | [#16](https://github.com/linhkhoi1309/eshop-sut/issues/16) |
| BUG-16 | No loading state | R6 | [#17](https://github.com/linhkhoi1309/eshop-sut/issues/17) |
| BUG-17 | No empty-state message | R7/FR-24 | [#18](https://github.com/linhkhoi1309/eshop-sut/issues/18) |

Each has its own issue body (`docs/issues/bug-11.md`, `bug-14…17.md`) with a focused repro, spec rule, `file:line`, and case traceabilit

## Ledger — now 17 bugs
**2 Critical · 7 High · 2 Medium · 6 Low.** (Issue #14 remains the closed/reverted one — not counted.)

Note the BUG-ID ↔ issue-number offset: because #14 was used-then-reverted, BUG-14/15/16/17 map to issues #15/#16/#17/#18. That's recorde

## Servers still running
Web http://localhost:5173 · Backend http://localhost:3000 — ready for you to keep cross-checking (BUG-10 XSS, BUG-04 `%` leak, etc.) and

Want me to start the **admin app (:5174)** for the FR-14 checks, **stop the servers**, or keep going?

Both background servers were stopped (backend :3000 and web :5173 are now down). Nothing was lost — all the bug reports, issue files, an

Current state:
- **No servers running.**
- **17 bugs filed** (2 Critical · 7 High · 2 Medium · 6 Low) across issues #1–#13 and #15–#18.
- Bug report, issue bodies, test results, and manual-verification docs all in sync.

Just say the word if you want me to:
- **Restart** the backend + web (and/or start the admin app on :5174) to keep cross-checking,
- **Post browser-verified notes / screenshots** to any issues, or
- wrap up here.
```

Interaction 40

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-06T06:22:25.492Z
- **AI response timestamp:** 2026-07-06T06:22:38.581Z
- **Tools used:** Bash, Bash, Bash, Bash, Bash, Bash

User Prompt

```
start the application
```

AI Output

```
Starting the full stack — backend, web, and admin. Let me check the admin app's deps and launch all three.

All deps present. Launching backend (:3000), web (:5173), and admin (:5174).

Backend and web are up, but the admin server reported a failure — let me check its output (my probe got a 200 on :5174, so something's i

All three are accessible. The admin "failure" is harmless: **port 5174 was already in use** — an admin dev server from earlier is still

## Application is running
| App | URL | Status |
|-----|-----|--------|
| **Backend API** | http://localhost:3000 | ✅ up (this session) |
| **Web (shopper)** | http://localhost:5173 | ✅ up (this session) |
| **Admin panel** | http://localhost:5174 | ✅ up (pre-existing process) |

**Logins:** admin `admin@eshop.com / Admin123!` · user `test@eshop.com / Test1234!`

## Quick cross-check pointers
- **Web (:5173)** — FR-05 bugs: search `<img src=x onerror=alert(1)>` (XSS, BUG-10), `%` → all products (BUG-04), `zzzzz` → no empty-sta
- **Admin (:5174)** — FR-14 bugs: log in as admin, add a category with an **empty name** → it's accepted (BUG-05); or add one whose name

## One heads-up
The admin server on :5174 is a **pre-existing process not managed by this session**, so when you're done I can stop the backend + web I

Want me to leave everything running, or is there something specific you'd like to walk through?
```

Interaction 41

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-06T06:26:23.801Z
- **AI response timestamp:** 2026-07-06T06:26:54.869Z
- **Tools used:** Write

User Prompt

```
where to put evidence
```

AI Output

```
**Put them in `docs/evidence/`** — that's the folder every issue body and the bug report already point to. Let me create it with a namin

Created **`docs/evidence/`** with a naming guide. Short version:

## Where
Put screenshots in **`docs/evidence/`**, named **`bug-01.png` … `bug-17.png`** (matching each bug's `Evidence:` line and each issue body

## Important GitHub caveat
`gh` can't attach images, and **repo-relative image links don't render inside GitHub issue bodies** — only in repo Markdown files like `

1. **Drag-drop the PNG into each issue** on github.com → GitHub uploads it to its CDN and embeds it. ✅ This is what satisfies the requ
2. **Also commit** the PNGs to `docs/evidence/` and push → they then render in your local `docs/bug-report.md` deliverable and give you

Do both for full marks.

## Mind the BUG↔issue number offset
Because issue #14 was reverted, the file/bug numbers don't equal issue numbers for the last four:

| File | Bug | **Issue** |
|------|-----|----------|
| bug-14.png | BUG-14 | **#15** |
| bug-15.png | BUG-15 | **#16** |
| bug-16.png | BUG-16 | **#17** |
| bug-17.png | BUG-17 | **#18** |

The full "what to capture" table for all 17 is in `docs/evidence/README.md`.

The app is still running (backend :3000, web :5173, admin :5174) so you can capture everything now. Want me to walk through capturing an
```

Interaction 42

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-06T06:29:42.957Z
- **AI response timestamp:** 2026-07-06T06:29:53.959Z
- **Tools used:** Bash

User Prompt

```
why the issue in github repo didnot see
```

AI Output

```
Let me verify the issues actually exist and on which repo, so we can pinpoint why you're not seeing them.

They **do** exist — all 17 open (plus #14 closed) on **`linhkhoi1309/eshop-sut`**. You're almost certainly looking at the **wrong repo*

## The cause: fork vs upstream
Your local repo has two remotes:
- `origin` → **`github.com/linhkhoi1309/eshop-sut`** ← *your fork — the issues are HERE*
- `upstream` → **`github.com/ttbhanh/eshop-sut`** ← *the original you forked from (no issues, and you don't have write access — that's w

If you opened the repo from the original/group link (`ttbhanh/eshop-sut`), the Issues tab there is empty/different. The issues were file

## Direct link — open this
👉 **https://github.com/linhkhoi1309/eshop-sut/issues**

Make sure you're **signed in as `linhkhoi1309`** and on the **Issues** tab (I had to enable it — it was off by default; if you had the p

You should see, newest first: #18/#17/#16/#15 (FR-05 GUI), #13 (order-status role), #12 (coupon case), #11 (two-h1), #10 (XSS), … down t

## If your assignment wants them on the *group* repo
The issues are on your personal fork, not on `ttbhanh/eshop-sut`. If the group is supposed to collect issues on the upstream `ttbhanh` r

Want me to print the clickable list of all 17 issue URLs, or help point the submission at the right repo?
```

Interaction 43

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-06T06:31:03.937Z
- **AI response timestamp:** 2026-07-06T06:31:23.470Z
- **Tools used:** Bash

User Prompt

```
how to reference evidence in github issue with the current repo
```

AI Output

```
Good question — the mechanism depends on whether the repo is public, so let me check that first.

Repo is **public**, default branch **`main`** — so you can embed in-repo images directly. The trick: **use an absolute `raw.githubuserco

## The pattern
1. Save the screenshot to `docs/evidence/bug-04.png`
2. Commit + push it:
   ```bash
   git add docs/evidence && git commit -m "Add bug evidence screenshots" && git push origin main
```

3. In the issue body, reference it with the **raw** URL:

   ```
   ![bug-04](https://raw.githubusercontent.com/linhkhoi1309/eshop-sut/main/docs/evidence/bug-04.png)
   ```

   That renders the image inline in the issue. (Equivalent alt form: `https://github.com/linhkhoi1309/eshop-sut/blob/main/docs/evidence/bug-04.png?raw=true`.)
   ```

General shape for any bug:

```
https://raw.githubusercontent.com/linhkhoi1309/eshop-sut/main/docs/evidence/bug-<NN>.png
```

Because the repo is public, these URLs work for anyone with no auth. (If it were private, raw URLs wouldn't render for others and you'd have to drag-drop instead.)

Two caveats

- The image only appears **after** you commit + push the PNG — until then the URL 404s.
- The issue bodies I filed currently use the **relative** `../evidence/bug-XX.png` (fine for `docs/bug-report.md`, but won't render in the issues).

Offer

I can **swap all 17 issue bodies** from the relative path to the absolute raw URL and **re-push them** with `gh issue edit` (correct bug→issue mapping handled). Then you just: capture the PNGs → `docs/evidence/` → commit + push, and every image pops into its issue automatically.

Want me to do that swap-and-re-push across all 17 now?

```
---

## Interaction 44
- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-06T06:32:05.210Z
- **AI response timestamp:** 2026-07-06T06:32:09.140Z
- **Tools used:** Bash, Bash, Edit, Bash

### User Prompt
```text
yes swap and re-push all 17
```
```

AI Output

```
I'll swap the relative evidence paths to absolute raw URLs in all 17 issue body files, then re-push each to its issue. First the URL swa

The swap ran (the `../evidence` grep matches were false positives — `.` is a regex wildcard matching `cs/evidence` inside the new raw UR

`bug-12.md` had a stale `bug-13.png` reference (leftover from the earlier rename). Fixing it to the correct raw URL:

All 17 bodies now use raw URLs. Re-pushing each to its issue with the correct bug→issue mapping.

All 17 issue bodies re-pushed with absolute raw-URL image embeds. Two issues fixed along the way:
- `bug-12.md` had a stale `bug-13.png` reference (leftover from the earlier rename) → corrected to `bug-12.png`.
- The `../evidence` grep "matches" were false positives (`.` is a regex wildcard hitting `cs/evidence` inside the new URLs) — no real re

Every issue now references:
```

https://raw.githubusercontent.com/linhkhoi1309/eshop-sut/main/docs/evidence/bug-.png

```
## What's left for the images to appear
The URLs 404 until the PNGs are committed. So:
1. Save screenshots to **`docs/evidence/bug-01.png … bug-17.png`** (names in `docs/evidence/README.md`; remember BUG-14→#15, 15→#16, 16-
2. Commit + push:
   ```bash
   git add docs/evidence && git commit -m "Add bug evidence screenshots" && git push origin main
```

3. Refresh each issue — the image renders inline automatically.

You can verify one issue now (e.g. open #1) — it'll show a broken-image placeholder until step 2, then resolve.

The app is still running (backend :3000, web :5173, admin :5174) so you can capture all the screenshots now. Want me to keep the servers up, or stop them when you're done?

```
---

## Interaction 45
- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-06T06:48:55.722Z
- **AI response timestamp:** 2026-07-06T06:49:43.665Z
- **Tools used:** Write, Bash

### User Prompt
```text
please write .http file to test bug relating to API in @docs/bug-report.md
```

AI Output

```
I'll create a `.http` file (VS Code REST Client format) covering the **11 API-testable bugs** — the FR-05 UI bugs (BUG-10/11/14/15/16/17

Created **`docs/api-bug-tests.http`**. Let me confirm the backend is still up so it's ready to run.

Backend is up (`200`), so the file is ready to run.

## `docs/api-bug-tests.http` — what's in it
Covers the **11 API-testable bugs**, VS Code REST Client format (click "Send Request" above each block):

| Bug | Requests | The tell |
|-----|----------|----------|
| BUG-04 | 3 GETs | `%`, `' OR '1'='1`, `Mac_ook` → return products instead of `[]` |
| BUG-01 | 1 POST | SAVE10 @ exactly 300,000 → `400` (should apply) |
| BUG-02 | 1 POST | SAVE10 @ 1,000,000 → `discount_amount: -9000000` |
| BUG-03 | 1 POST | no token/user_id → `200 success` |
| BUG-12 | 1 POST | `save10` → `404` |
| BUG-05 | 2 POSTs | empty / whitespace name → `200 created` |
| BUG-06 | 2 (user token) | create + delete category → `200` (should be `403`) |
| BUG-13 | 2 chained | user drives admin status → `200` |
| BUG-07 | 4 chained | user cancels a shipping order → `200` |
| BUG-08 | 3 chained | admin `canceled → delivered` → `200` |
| BUG-09 | 5 chained | delete category → product left orphaned |

## How to run
1. Backend already running (`:3000`).
2. **Run "Login as USER" and "Login as ADMIN" first** — they populate `{{userToken}}`, `{{adminToken}}`, `{{userId}}` via REST Client's
3. Run each bug section top-to-bottom. The multi-step ones (BUG-07/08/09/13) chain values like `{{checkout07.response.body.$.orderId}}`

Each request has an `EXPECT` vs `ACTUAL` comment, so a bug is confirmed when the response differs from EXPECT.

## Notes
- **Excluded (not HTTP-observable):** BUG-10 (XSS) and BUG-11/14/15/16/17 (FR-05 listing GUI) — those are browser-render defects; verify
- Requires the **REST Client** extension (`humao.rest-client`) in VS Code, or a compatible `.http` runner. JetBrains' HTTP client uses a
- Some bugs write data (orders/categories); re-run `node backend/database.js` (or restart the server) for a clean slate between full pas
```

Interaction 46

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-06T07:17:00.820Z
- **AI response timestamp:** 2026-07-06T07:17:26.658Z
- **Tools used:** Bash, Edit, Edit, Edit, Edit, Bash

User Prompt

```
i cannot login as admin in @docs/api-bug-tests.http
```

AI Output

```
Let me diagnose directly — testing the admin login against the running backend to see what actually comes back.

The credentials are fine — admin login returns `200` with a valid token via curl, and the account isn't locked. So the failure is in the

The robust fix is the documented REST Client idiom: **reference the login response inline** in each request and drop the intermediate `@

Now replacing the `{{...Token}}`/`{{userId}}` references with inline response lookups:

Now the user token and user id references:

Finally the `{{userId}}` references:

The remaining "userId is not found" entries are stale (my edit reported all occurrences replaced). Let me confirm the file is clean now

Fixed. All the flaky `{{adminToken}}` / `{{userToken}}` / `{{userId}}` indirection is gone — every request now references the login resp

## Why "can't log in as admin" happened
The admin login request itself was fine (curl returns `200` + token). The problem was the file-level `@adminToken = {{adminLogin.respons

## What the remaining diagnostics mean (they're not errors to fix)
The ⚠ "Only JSON… request has not been sent" and ● "Request 'X' has not been sent" are **static-analysis notices**, not bugs. They just

## How to run it now
1. **Send "Login as USER"** ▶ then **"Login as ADMIN"** ▶ (once per session). After this, all the ⚠ token warnings resolve.
2. Run each bug section. For the chained ones, send in order:
   - BUG-13: 13a → 13b
   - BUG-07: 07a → 07b → 07c → 07d
   - BUG-08: 08a → 08b → 08c
   - BUG-09: 09a → 09b → 09c → 09d → 09e

The backend is up (`:3000`), so you can send the two logins right now and the admin token will populate. Want me to add a JetBrains-HTTP
```

Interaction 47

- **AI Tool:** Claude Code
- **User prompt timestamp:** 2026-07-06T07:32:56.914Z
- **AI response timestamp:** 2026-07-06T07:33:21.025Z
- **Tools used:** Bash

User Prompt

```
@docs/api-bug-tests.http how about bug-13
```

AI Output

```
Let me actually run the BUG-13 flow end-to-end via curl against the live backend, so you can see exactly what it should produce.

BUG-13 reproduces exactly as documented — the `.http` file is correct. Here's the proof I just ran against the live backend:
```

BUG-13a checkout as user → order id 5 created BUG-13b user PUT /api/admin/orders/5/status {status:“confirmed”} → HTTP 200 {“message”:“Order status updated”}

```
**Expected:** `403 Forbidden` (a normal user shouldn't touch `/api/admin/*`).
**Actual:** `200` — the non-admin drove the state machine. ✅ bug confirmed.

## The one thing to watch in the `.http` file
BUG-13b depends on BUG-13a's response (`{{checkout13.response.body.$.orderId}}`). So you must **send 13a first, then 13b** in the same s

Correct sequence for BUG-13:
1. ▶ **Login as USER** (once)
2. ▶ **BUG-13a** — creates the order (captures `orderId`)
3. ▶ **BUG-13b** — user drives admin status → you'll see `200` (should be `403`)

Note it doesn't even need the admin login — the whole point of BUG-13 is that a *user* token reaches an admin route.

So nothing's wrong with BUG-13 in the file. If you were seeing it fail, it was almost certainly the "run 13a before 13b" ordering, or th
```