**How to change directory in the terminal and run full backend and frontend tests**

* Change directory, then run:

cd ..\frontend

cd ..\backend

npm test

npm start

* Using absolute path (if you prefer):
  + npm --prefix C:\Dev\Merkato\frontend test
  + npm --prefix C:\Dev\Merkato\frontend run e2e:smoke

You’re in backend, so --prefix frontend points to backend/frontend (which doesn’t exist). Use ..\frontend (or an absolute path), or cd into the frontend first. Also, don’t type narrator text like “Run tests:” — PowerShell treats it as a command.

Quick options

**Change directory, then run:**

cd ..\frontend

npm test

npm start

**Stay in backend and target frontend with prefix:**

Tests: npm --prefix ..\frontend test

Dev server: npm --prefix ..\frontend run start

E2E (Docker smoke): npm --prefix ..\frontend run e2e:smoke

E2E (Docker core): npm --prefix ..\frontend run cy:run:docker

E2E (local runner): npm --prefix ..\frontend run e2e:core

Using absolute path (if you prefer):

npm --prefix C:\Dev\Merkato\frontend test

npm --prefix C:\Dev\Merkato\frontend run e2e:smoke

Tip: If backend is running in that terminal, open a new one or stop it with Ctrl+C before starting the frontend dev server.

Here are the E2E commands you can run from the terminal. Use either repo root with --prefix or cd into frontend first.

From repo root

Smoke (Docker, fastest/most reliable): npm --prefix ..\frontend run e2e:smoke

Core set (Docker): npm --prefix ..\frontend run cy:run:docker

Local runner (no Docker): npm --prefix ..\frontend run e2e:core

Single spec (local runner):

PowerShell: $env:E2E\_SPEC="cypress/e2e/smoke.cy.js"; npm --prefix ..\frontend run e2e:run

From frontend directory

Smoke (Docker): npm run e2e:smoke

Core (Docker): npm run cy:run:docker

Local runner (all core specs): npm run e2e:core

Single spec (local runner):

PowerShell: $env:E2E\_SPEC="cypress/e2e/card\_intent.cy.js"; npm run e2e:run

* Backend tests: npm --prefix backend test
* Frontend tests: npm --prefix frontend test
* E2E (Docker): npm --prefix frontend run cy:run:docker
* E2E (CI Nightly): push/merge or trigger “E2E Nightly”

cd "C:\Dev\Merkato\backend>

cd "C:\Dev\Merkato\frontend>

cd "C:\Dev\Merkato>

PS C:\Dev\Merkato\backend>

PS C:\Dev\Merkato\frontend>

PS C:\Dev\Merkato>

npm --prefix frontend run cy:run:docker

npm --prefix frontend run cy:run:docker

npm --prefix backend test –silent

npm --prefix frontend run e2e:core

* Force local Mongo for E2E scripts
  + Updated both runners to set MONGO\_URI to mongodb://127.0.0.1:27017/merkato\_e2e, avoiding slow Atlas calls.
  + Files: frontend/scripts/run-e2e.js, frontend/scripts/run-e2e\_docker.js
  + Make sure your local MongoDB is running.
* Cypress defaults to reduce flake
  + Increased timeouts and enabled retries.
  + File: frontend/cypress.config.js (defaultCommandTimeout, pageLoadTimeout, retries)

How to run all specs

* Local runner (all):
  + PowerShell: Remove-Item Env:E2E\_SPEC
  + Then: npm run e2e:run (from frontend)
* Docker runner (all, more consistent):
  + npm run cy:run:docker (from frontend)
* Smoke on Docker (fast subset): npm run e2e:smoke

Targeted triage plan (to raise pass rate fast)

* Fix login-based flows first (most failures are auth/role redirects)
  + Ensure tests use seeded emails:
    - Admin: admin@test.com
    - Vendor: vendor@test.com
    - Customer: customer@test.com
  + Wait explicitly on login/XHRs to avoid race conditions:
    - cy.intercept('POST', '/api/auth/login').as('login'); cy.wait('@login')
  + Confirm redirects with url check and a simple dom sentinel:
    - cy.url().should('include', '/admin')
    - cy.get('[data-cy=dashboard-content]').should('contain', 'Welcome back, Admin')
* Stabilize selectors and waits
  + Use data-cy= hooks where available (we added many), and avoid implicit .contains(...) for dynamic text.
  + Add route waits around admin/vendor pages that fetch lists.
* Cache sessions to speed and reduce flake
  + Use cy.session() around logins in beforeEach to reuse authenticated state.
* Keep network local
  + Confirm your E2E run logs show local Mongo (127.0.0.1) — the scripts now enforce this, which improves speed dramatically.

Suggested run sequence

* Verify the improvement with a core subset:
  + npm run e2e:smoke
* Then run the curated core set:
  + npm run e2e:core
* Finally run the full suite (local or Docker):
  + Local: Remove-Item Env:E2E\_SPEC; npm run e2e:run
  + Docker: npm run cy:run:docker

If you want, I can go file-by-file on the failing specs (auth\_roles, smoke, basic\_navigation, checkout\_payment, etc.), add the missing waits/selectors, and push a second pass. The usual pattern is to add cy.intercept + waits around login and page data fetches, and adjust to the seeded credentials.

3 files changed+22-3

UndoView changes

cypress.config.js+5-0

run-e2e.js+9-1

run-e2e\_docker.js+8-2

Action for tomorrow

Sounds good — great progress today.

Quick recap

* Backend: 198/198 tests green; payments validation + verify/webhooks in place; unified customer model solid.
* Frontend: 146/146 tests green; checkout uses dynamic methods and artifacts; frictionless paths aligned.
* E2E: Runners set up with local Mongo, retries/timeouts tweaked; smoke + payment stubs ready; CI jobs added (unit, smoke, nightly).

When you’re back, I can:

* Triage failing E2E specs (auth, basic nav, checkout) with targeted intercepts/waits and selector fixes.
* Extend smoke to cover login, stripe/paypal flows, and a simple vendor/admin path.
* Optionally add cy.session to speed stateful specs.

Rest well — I’ll be ready to pick up right where we left off.