**Test Cases**

**1) Authentication — Login returns 401 for incorrect credentials**

**Precondition:**  
A POST /api/auth/login (or similar) endpoint exists, which must return **401 Unauthorized** when credentials are invalid.

**Steps (UI):**

1. Open http://localhost:5173/login.
2. Enter a valid email (that does not exist) **or** an incorrect password for an existing email.
3. Click **“Login”**.

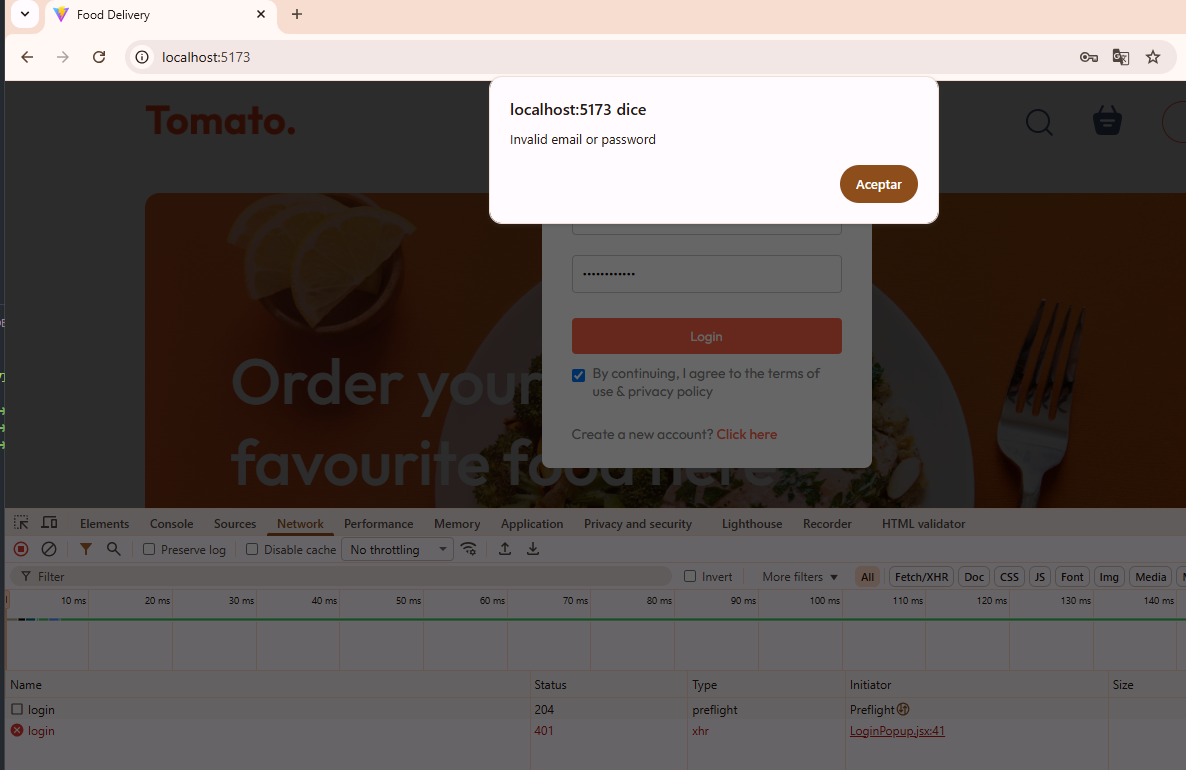
**Expected Result (UI):**

* A clear error message is displayed (e.g., “Invalid credentials”).
* The user is **not redirected** to the dashboard/home.
* **No token** is stored in localStorage.

**Expected Result (Network):**

* In **DevTools → Network**, the login request responds with **HTTP 401**.
* The response body may include { message: "Unauthorized" } or similar.
* **No cookies or tokens** are set.

**Acceptance Notes:**  
Any case with invalid credentials **must return 401**, not 200 with an error message in the body.



**2) Authentication — Registration returns 401 if the user already exists**

**Precondition:**  
You already have a registered user (you can register it once beforehand).

**Steps (UI):**

1. Open http://localhost:5173/register.
2. Fill out the form using the **same email** that is already registered.
3. Click **Submit**.

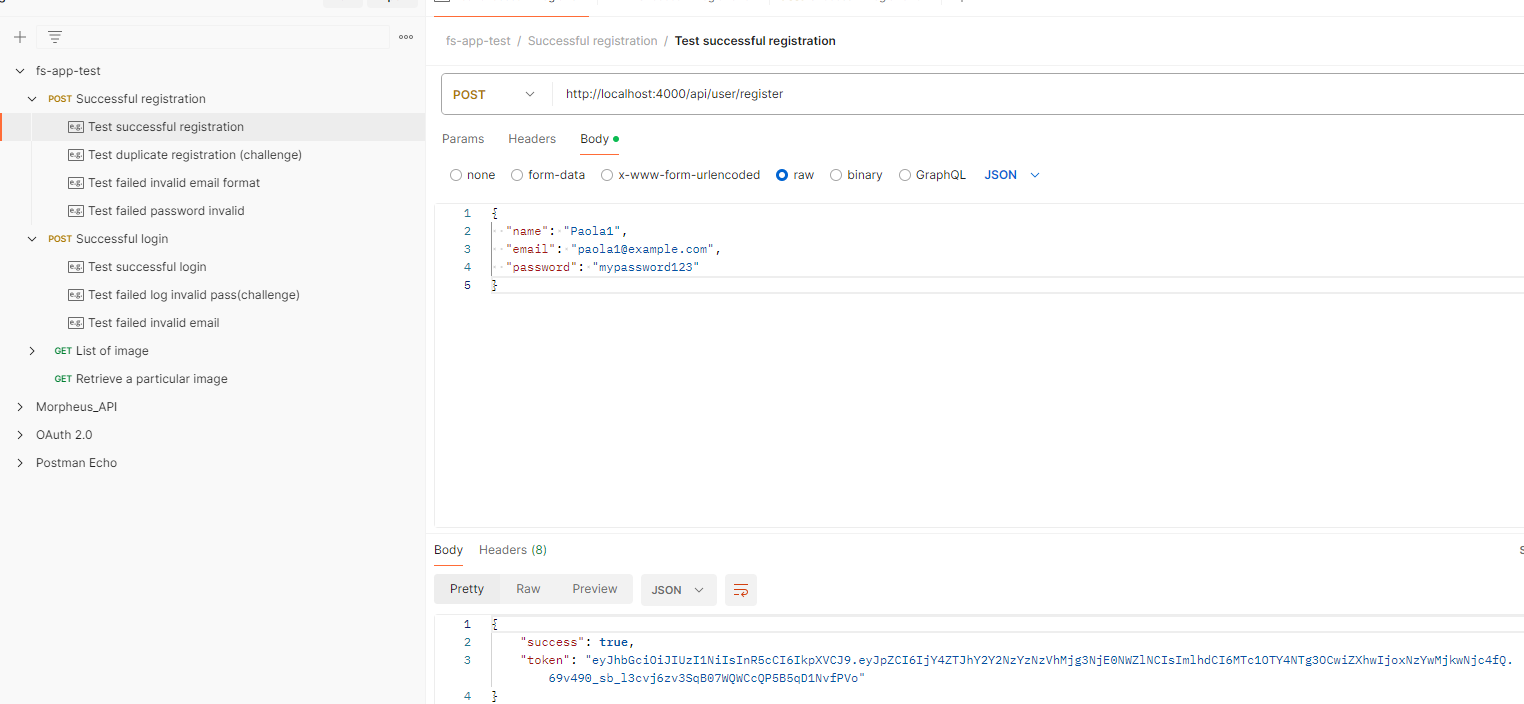
**Expected Result (UI):**

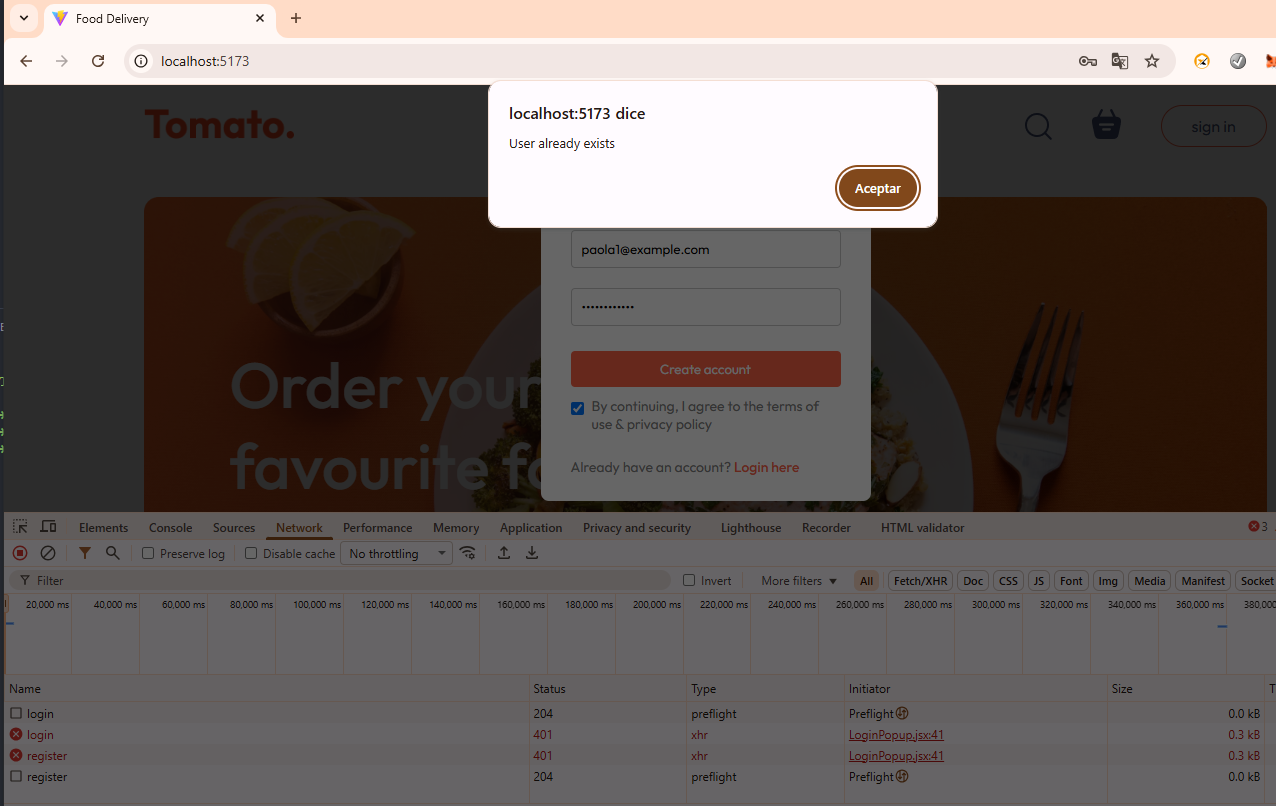
* A message is displayed such as **“User already exists”** or **“Email already registered.”**
* You remain on the registration screen.

**Expected Result (Network):**

* The response is **HTTP 401** (or **409 Conflict**, if preferred, but the challenge specifies 401).
* No token is returned, and no new user is created.

**Acceptance Notes:**  
The original bug was returning **200 OK** — it must now return **401 Unauthorized.**

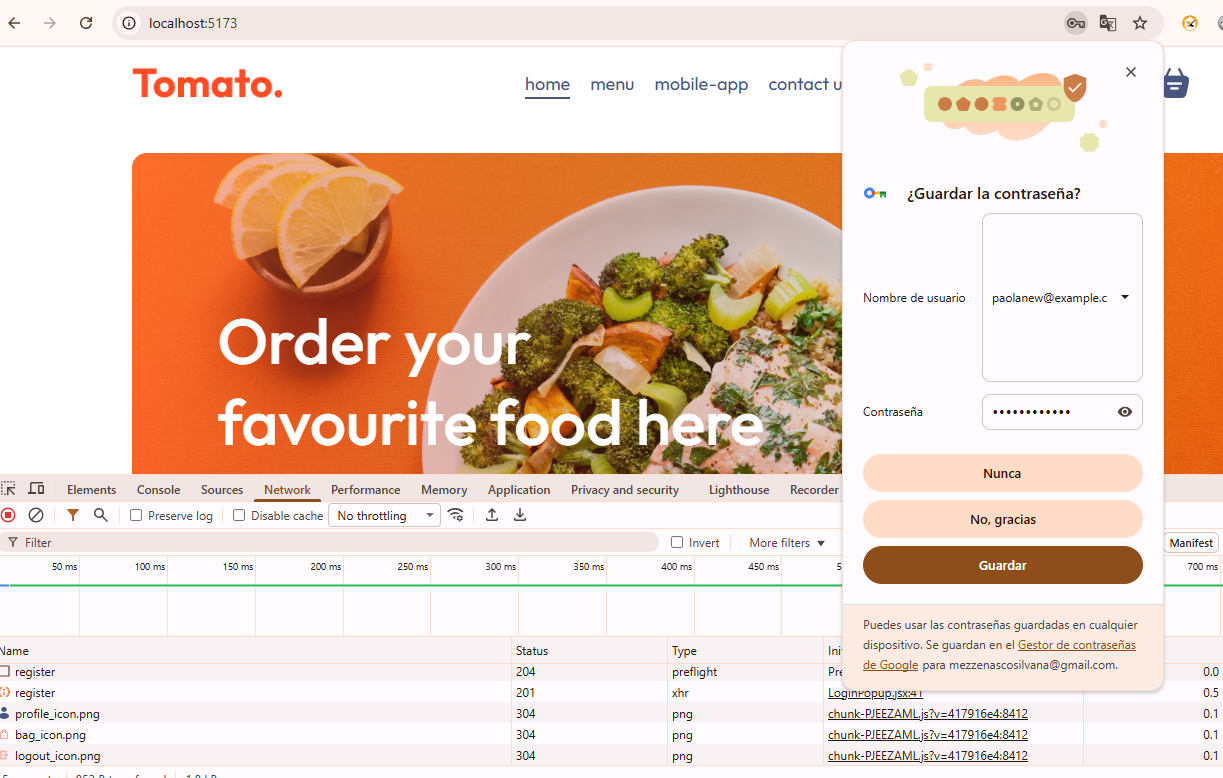




**3) Authentication — Happy Flows (to ensure existing functionality still works)**

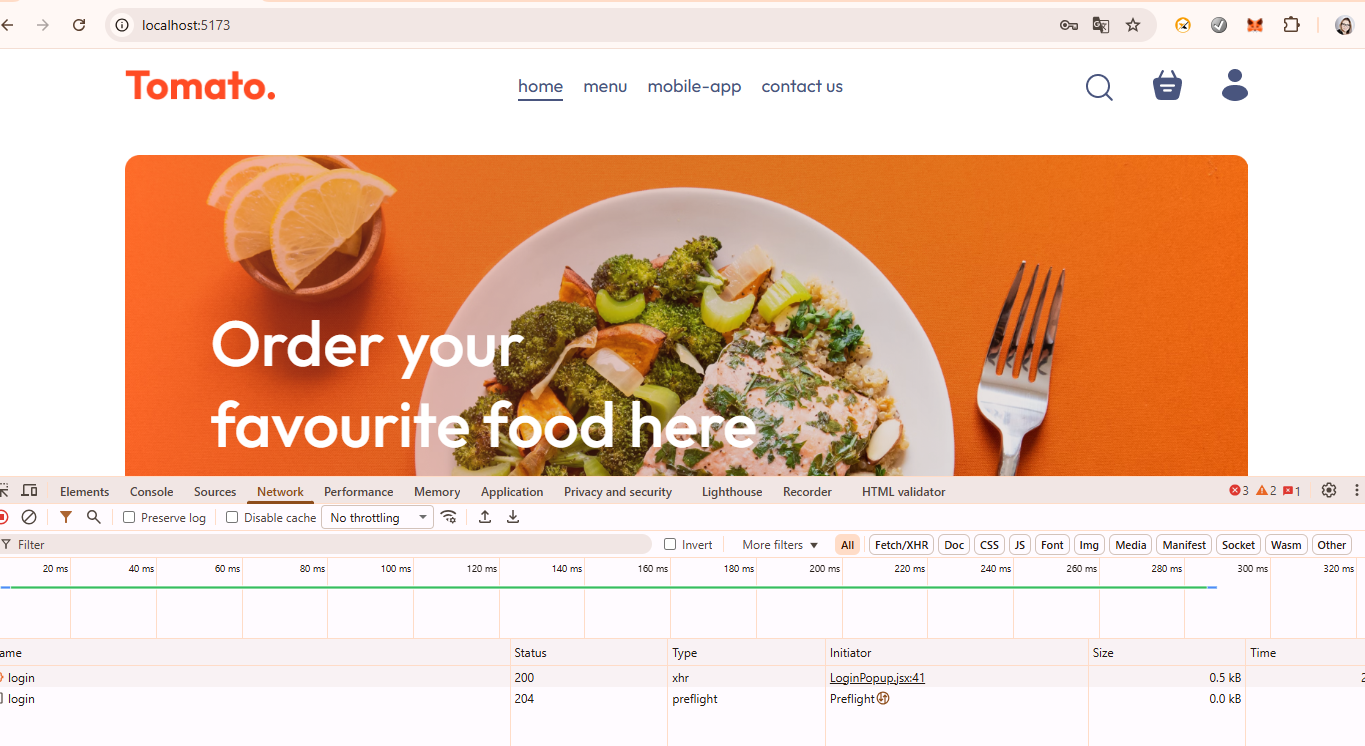
**3.1 Successful Registration**  
**Steps:** Register with a **new valid email**.  
**Expected (UI):** Success message **or** redirection to **login/home**.  
**Expected (Network):** **HTTP 201/200** with a token or confirmation message — **not 401**.

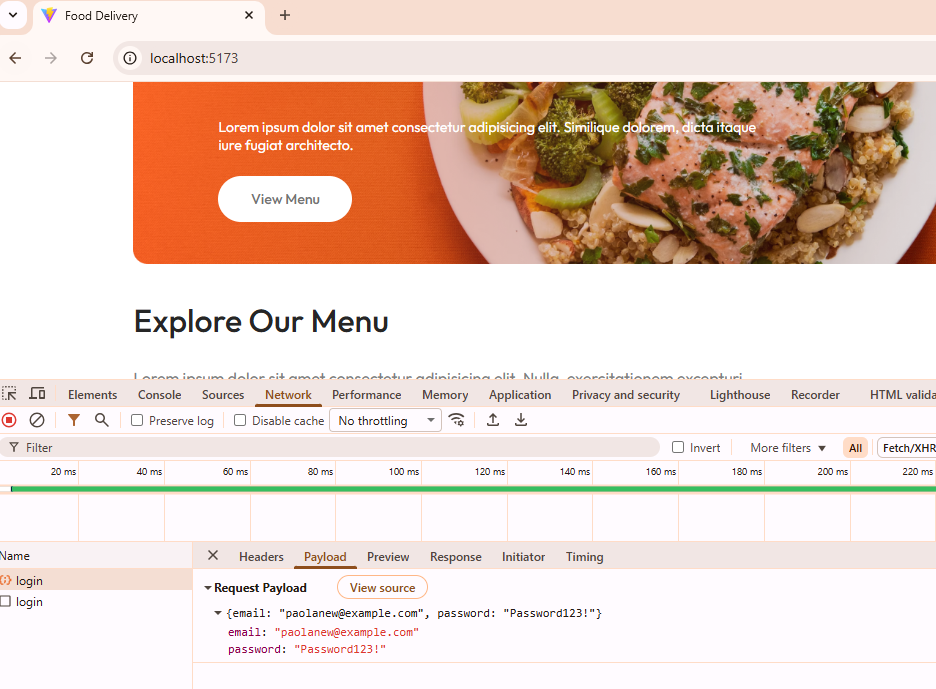
From Postman: Collection attached : Test successful registration

From UI : 

**3.2 Successful Login**  
**Steps:** Log in with **valid email and password**.  
**Expected (UI):** Redirects to **home**, shows **logged-in navbar/state**, and **cart is visible**.  
**Expected (Storage):** JWT token is persisted (**localStorage** or **cookie**).  
**Expected (Network):** **HTTP 200** with valid token or session.

From Postman: Collection attached > Test successful login

From UI: 



**4) UI Enhancement — Zoom-in on hover for menu images**

**Precondition:**  
Menu page is available (list of dishes/products with images).

**Steps (UI):**

1. Open http://localhost:5173
2. Hover the mouse over each product image.

**Expected Result (Visual):**

* The image smoothly zooms in (e.g., transform: scale(1.05–1.1) with transition: transform 200–300ms).
* No **layout shift** occurs (the card itself should not move or resize — only the image crops internally).
* **Keyboard focus (TAB)** should also trigger the same effect for accessibility — optional but recommended.

**How to Verify: Video attached**

**5) Order UI Fix — The “–” icon remains fixed when changing quantities**

**Precondition:**  
Product or cart view includes **+** and **–** controls.

**Steps (UI):**

1. Add an item to the cart from the menu.
2. Open the cart.
3. Press **+** several times (0 → 1 → 2 → 3…).
4. Press **–** several times (3 → 2 → 1 → 0).
5. Switch to another product and repeat.

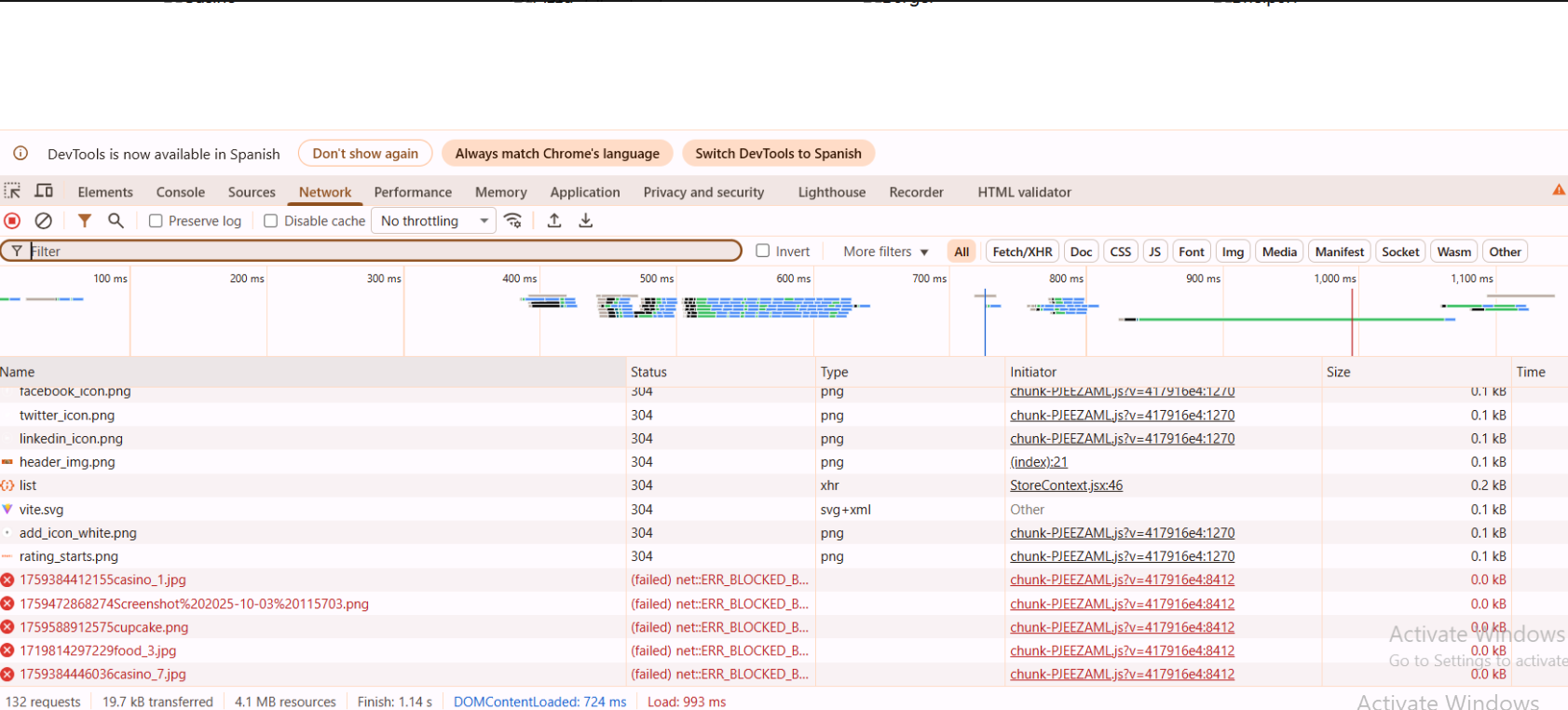
**Expected Result (Visual/UX):**

* The **“–” icon** keeps its exact position and alignment regardless of the quantity value.
* The quantity container has a **fixed width** (min-width or width) or uses a **monospaced font** for the number.
* There are **no jumps or layout shifts** of the button or surrounding elements.

**How to Verify:**

* In **DevTools**, inspect the “–” button and the quantity container — their boxes should remain unchanged as the number changes.
* Test again by adjusting browser zoom (e.g., 90%, 110%) — alignment must remain consistent.
* Video attached

***Note*** Please note that the images are being blocked by my browser.  
I can confirm that the backend is working correctly because I’m able to retrieve the full list of images from Postman, and I can also access each individual image directly through its URL.  
The issue only occurs in the browser, which blocks the image requests due to local security or ad-blocking settings.

******