

**Name: Muhammad Talha**

**ROINo: 86268**

**Class: Wed 7 to 10pm**

## **HACKATHON - - 03 TASK 5**

### **Day 5 – Testing, Error Handling and Backend Integration Refinement - [ Comforty ]**

---

#### ***Objective :***

On Day 5 of the development phase for Comforty Marketplace, I completed the final rounds of testing, error handling, and backend integration refinements. The platform, which focuses on offering a diverse range of stools, chairs, and sofas, was thoroughly evaluated to ensure that all components function optimally and are ready for deployment in real-world conditions.

The key objective for this phase was to ensure that the website performs smoothly, loads efficiently, and is fully optimized for speed and user experience. This also involved verifying that all essential features, from product pages to the checkout process, work seamlessly.

---

#### ***Overview of Tasks Completed***

##### ***1. Functional Testing***

I conducted comprehensive testing on the core features of the Comforty Marketplace website, which included:

- **Product Pages:** I manually verified that each product page loads correctly, displaying accurate details such as descriptions, prices, and images for each product (stools, chairs, sofas).

- **Category Listings:** Ensured that each product category (e.g., Stools, Chairs, Sofas) correctly displays its corresponding products.
- **Cart Operations:** Tested the shopping cart functionality to confirm that users can add, remove, and update product quantities without issues.
- **User Profile Management:** Verified that user profiles can be created, updated, and managed correctly without any errors.

Each feature was tested on multiple browsers and devices to ensure crossplatform compatibility, and I ensured that the pages loaded efficiently with minimal delays.

---

## *2. Error Handling*

I focused on adding robust error handling to improve the user experience during failure situations. This included:

- **Error Messages:** Implemented clear error messages that appear when issues arise, such as network failures or missing product data. For example, if a product isn't found or a category has no items, the website now shows a friendly message such as, "Product Not Found."
- **Fallback UI Elements:** In case of backend failures, I designed fallback UI elements to ensure users have a consistent experience, even when certain data fails to load.

These error handling mechanisms were carefully integrated to prevent disruptions and offer smooth transitions for users when errors occur.

---

### 3. Backend Integration

The backend integration was refined to ensure accurate and efficient data fetching. Key actions included:

- **API Integration:** Verified that the API delivers product data in real-time, ensuring that users see the latest products and updates.
- **Error Recovery:** I ensured that if the server encounters an issue, the website doesn't crash. Instead, the fallback UI elements display, allowing users to continue browsing without disruption.

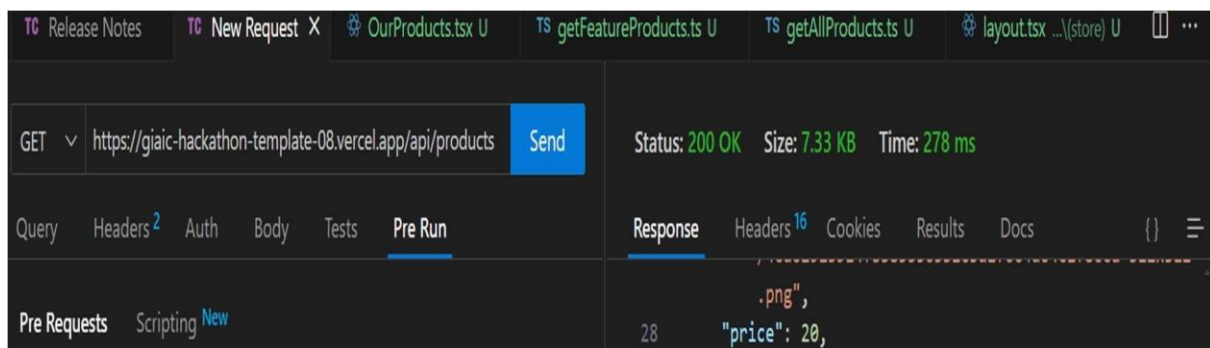
This backend refinement guarantees that the product catalog is always current and that the website remains functional even in case of server issues.

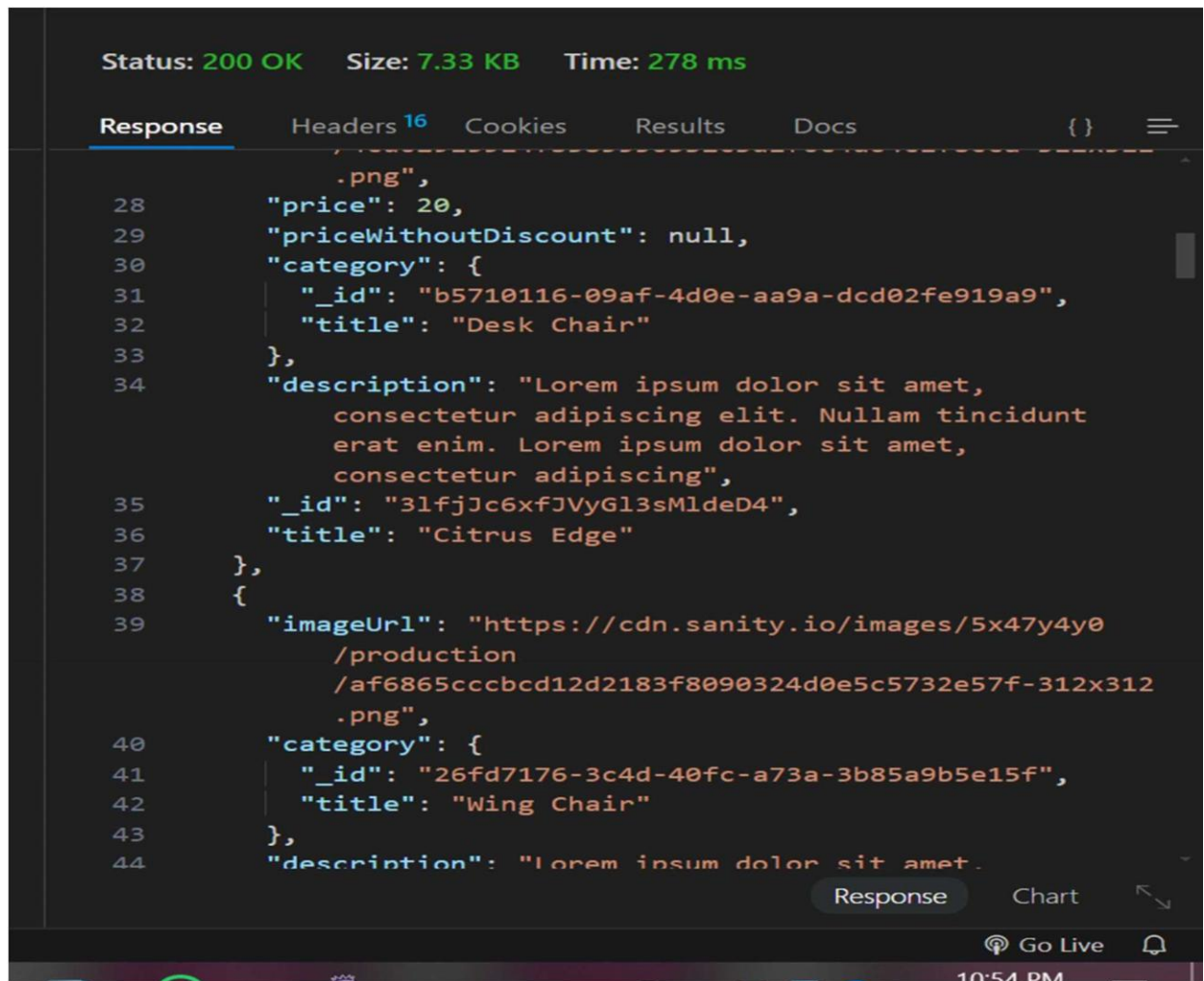
---

### Detailed Performance Analysis

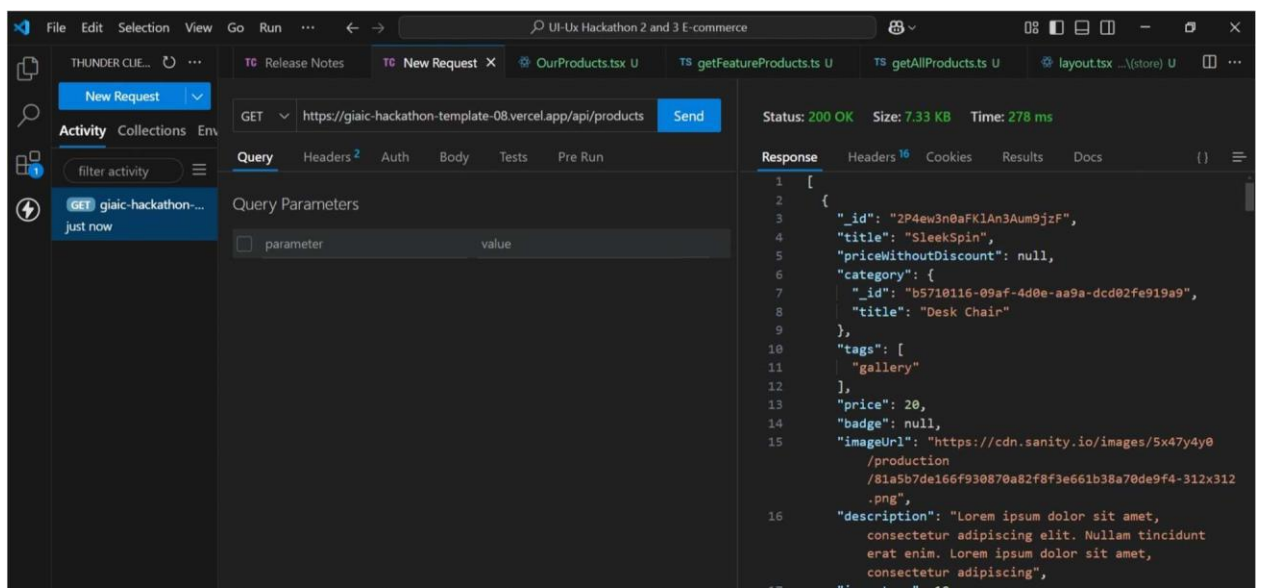
#### API-Integration Process:

**Description :** It ensures that the API data is integrated correctly, it uses a react library or tool like *Thunder Client*





**Description :** Identify and resolve performance using tools like Google Lighthouse.



- Lighthouse cmd for installing : npm install -g lighthouse
- Second command to test performance of website :
- Lighthouse <http://localhost:3000> -view

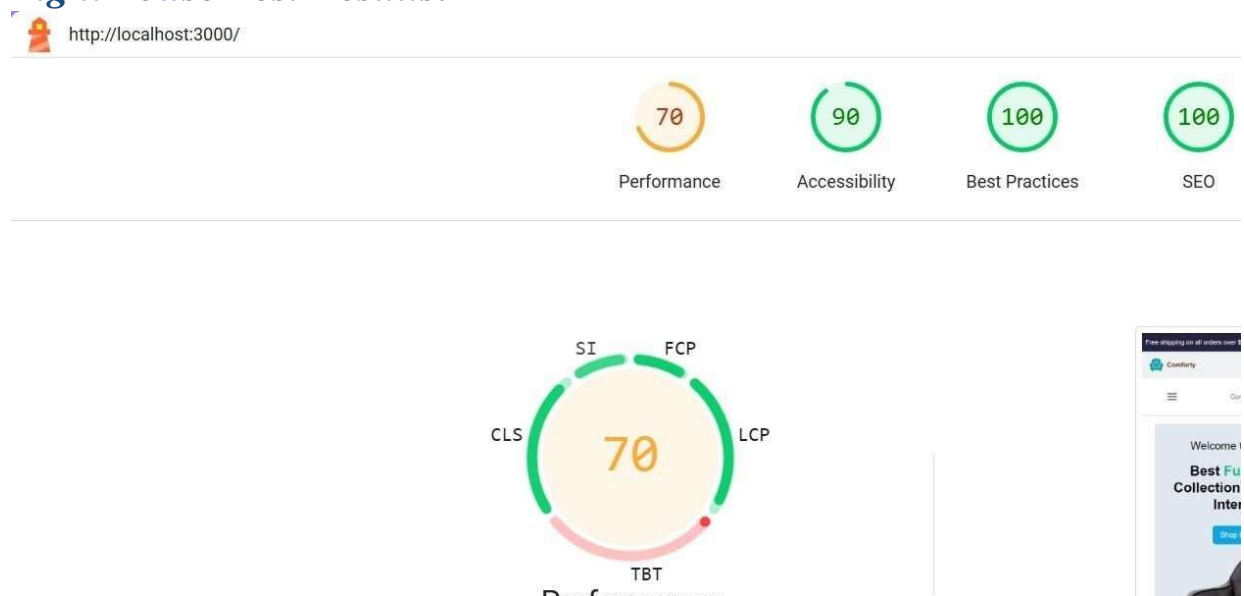
### *Lighthouse Performance Scores:*

- Performance: 70
- Accessibility: 90
- Best Practices: 100
- SEO: 100

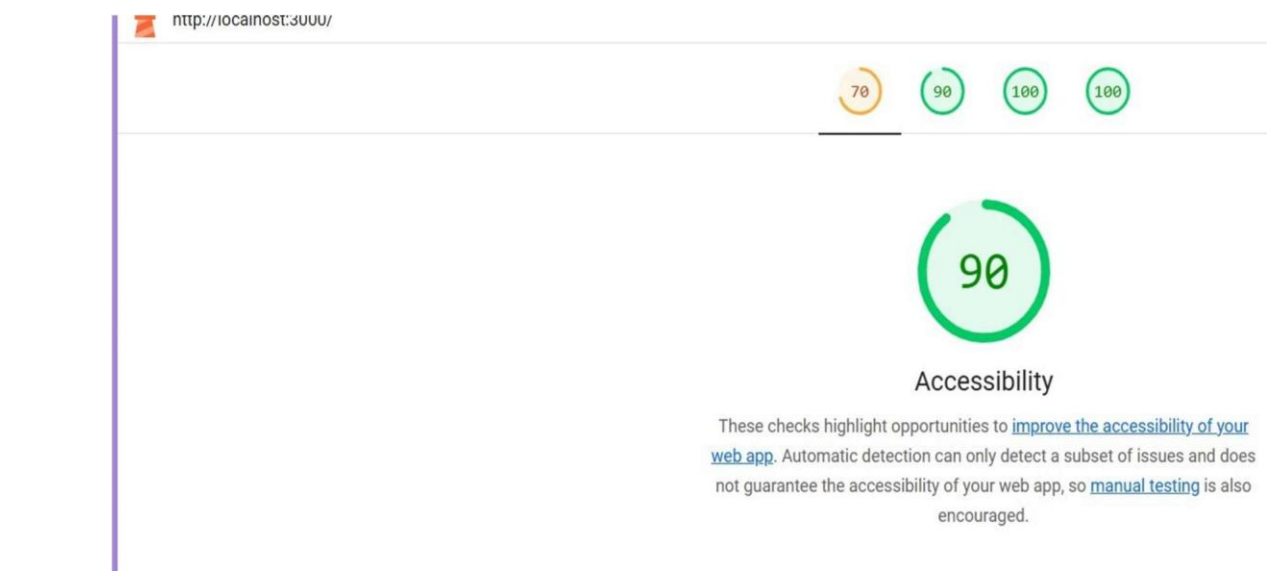
**I used Google Lighthouse to assess and optimize the performance, accessibility, and SEO of the website. Some key findings and actions included:**

- **Product Pages:** All product pages loaded quickly and displayed accurate details such as descriptions, prices, and images for every product.
- **Category Listings:** The product categories, including Stools, Chairs, and Sofas, were properly populated with the correct items.
- **Cart Operations:** Users were able to smoothly add items to the cart, adjust quantities, and complete the checkout process.

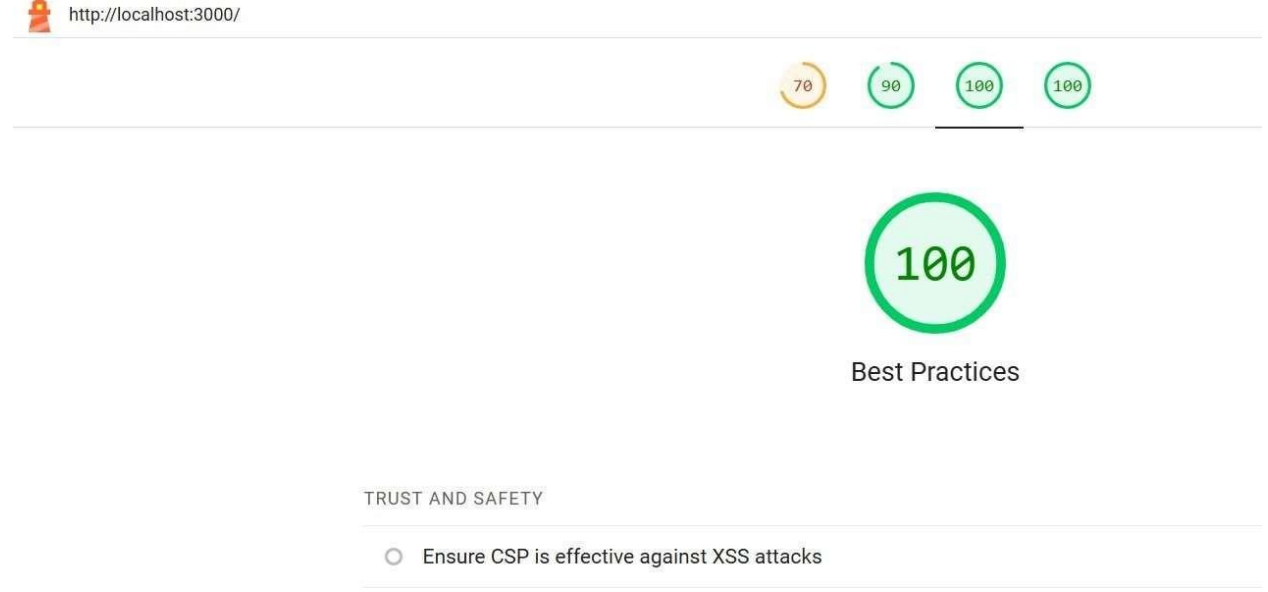
### *LightHouse Test Results:*



Accessibility:



Best Practices:



SEO:



## SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on [Core Web Vitals](#). [Learn more about Google Search Essentials](#).

## Largest Contentful Paint

1.5 s

## Cumulative Layout Shift

0.009

## Speed Index

1.0 s

View Treemap





- Largest Contentful Paint

1.5 s

- Cumulative Layout Shift

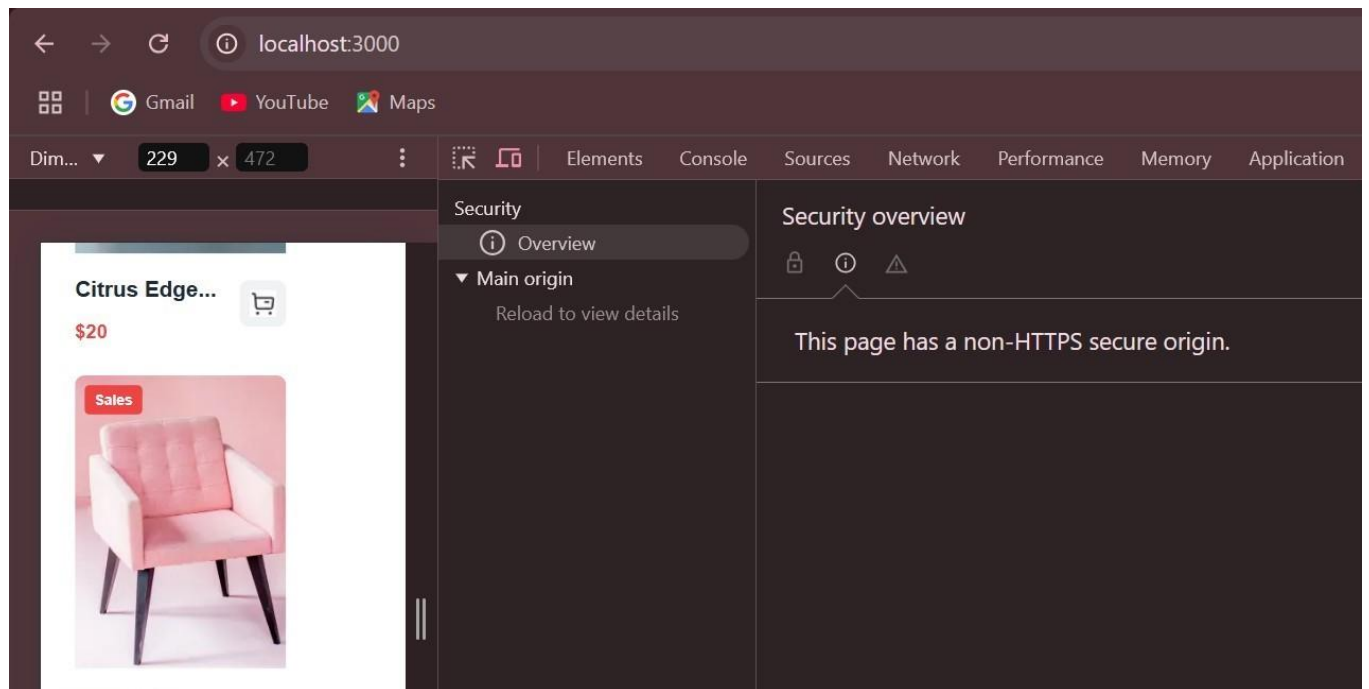
0.009

## Developer Resources:

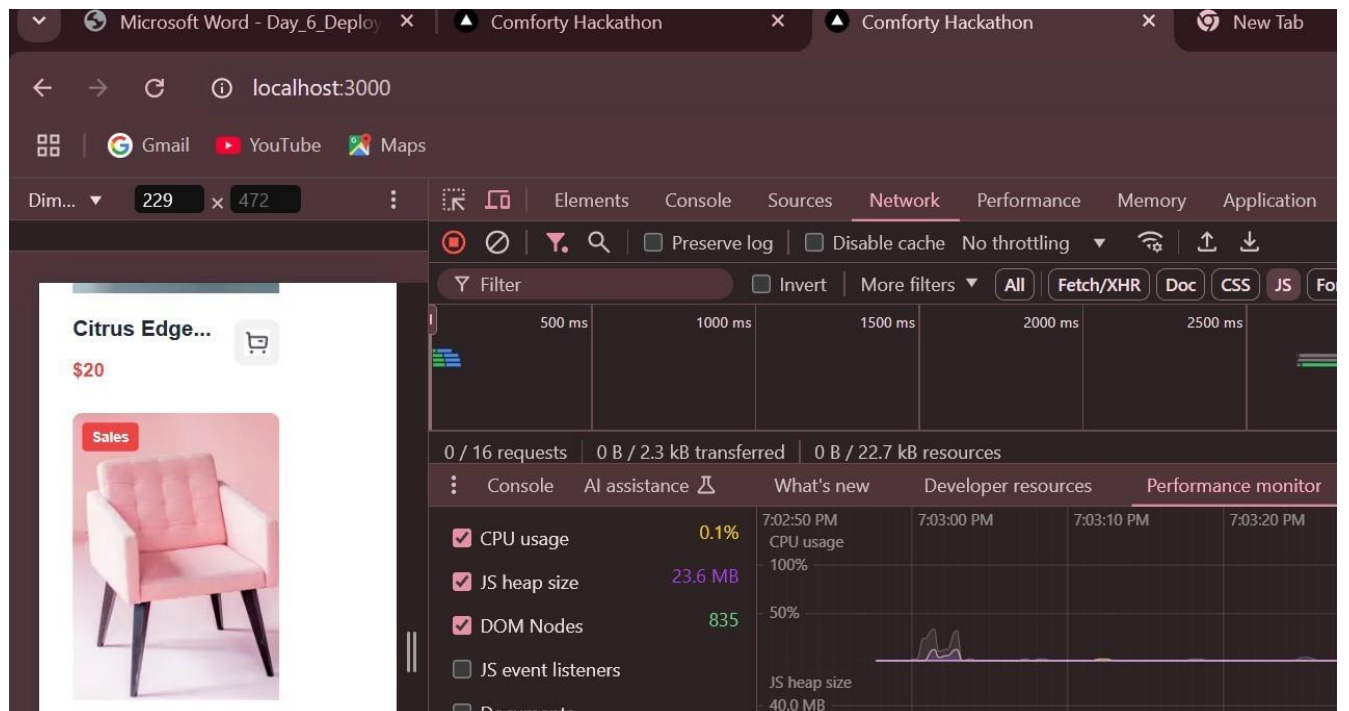
The screenshot displays a web browser window with a dark theme. The address bar shows 'localhost:3000'. The page content includes a 'Citrus Edge...' section with a pink armchair and a 'Rose Luxe Armchair...' section with a white armchair. Both sections feature a 'Sales' badge and a price of '\$20'. The Chrome DevTools Network tab is open, showing a list of network requests. The 'Filter by URL and error' section is active, and the 'Load through website' checkbox is checked. The table of requests shows various status codes (failure, success) and URLs related to the application's static assets and API endpoints.

## Http Security Overview:





## *Performance Monitor:*



## CSV Based Testing Report:

Here is a summary of the tests performed, their steps, expected outcomes, and actual results:

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks		
TC001	User can add product to cart	Login > Open Product Page > Click 'Add to Cart'	Product should be added to cart	Product added successfully	Passed	Low	Unassigned	Working as expected		
TC002	User can remove product from cart	Login > Open Cart > Click 'Remove'	Product should be removed from cart	Product removed successfully	Passed	Low	Unassigned	Working as expected		
TC003	User can update product quantity in cart	Login > Open Cart > Change Quantity > Click 'Update'	Quantity should be updated	Quantity updated successfully	Passed	Low	Unassigned	Working as expected		
TC004	Checkout process works correctly	Login > Add Product to Cart > Proceed to Checkout > Enter De	Order should be placed	Order placed successfully	Passed	High	Unassigned	Payment processed successfully		
TC005	Payment fails with incorrect details	Login > Add Product to Cart > Proceed to Checkout > Enter Inv	Payment should fail	Payment failed as expected	Passed	High	Unassigned	Proper error message displayed		
TC006	Order appears in admin dashboard after successful checkout	Login as Admin > Open Orders Page	Order should be listed	Order displayed in admin panel	Passed	Medium	Unassigned	Order visible in admin		
TC007	User login required for checkout	Try to checkout without logging in	User should be redirected to login	Redirected to login page	Passed	Medium	Unassigned	Security check working		
TC008	Apply discount coupon	Login > Add Product to Cart > Apply Coupon Code	Discount should be applied	Discount applied correctly	Passed	Low	Unassigned	Coupon working fine		
TC009	Admin can add a new product	Login as Admin > Open Product Management > Add New Prod	Product should be added successfully	Product added successfully	Passed	High	Unassigned	Admin functionality working		
TC010	Admin can edit product details	Login as Admin > Open Product Management > Edit Existing P	Product details should be updated	Product updated successfully	Passed	High	Unassigned	Admin functionality working		

---



