# Hackthon Day 5 Testing, Error Handling, And Backend Integration Refinement

Website Frontend Overview

My website is built with Next.js, a powerful React framework. The frontend is modern and user-friendly, ensuring fast rendering and seamless navigation.

My website is built with **Next.js**, a powerful React framework. The frontend is **modern, user-friendly, and fully responsive**, ensuring an optimal experience across all devices, including desktops, tablets, and smartphones. The **UI (User Interface)** is clean, intuitive, and designed with a focus on seamless interaction, making navigation and shopping enjoyable for users.

# Functionality

A key feature is the "Add to Cart" functionality, which operates through an API. When a user adds a product to the cart, a request is sent to the API, which then stores and manages the data on the backend. This ensures a smooth and efficient shopping experience.

The website also includes a localhost button, which is used for testing and debugging during local development. This button helps in testing specific functionalities or triggering a local API endpoint.

# Responsive

 **Fluid Layout:** The website design utilizes CSS Flexbox/Grid or percentage-based widths, allowing the content to adjust seamlessly.

 **Media Queries:** Custom styles are applied for different screen sizes, ensuring the UI renders correctly on all devices.

 **Mobile-Friendly Navigation:** The navbar and buttons are touch-friendly, providing a smooth interaction for mobile users.

 **Optimized Images & Text:** Images and fonts are responsive, maintaining both readability and performance across devices

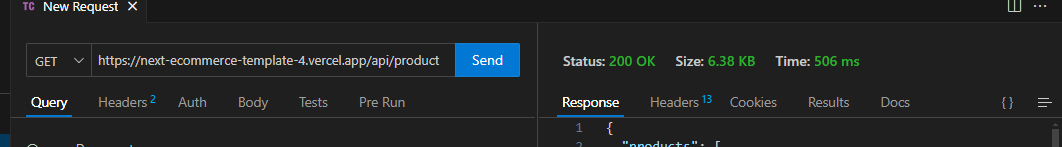
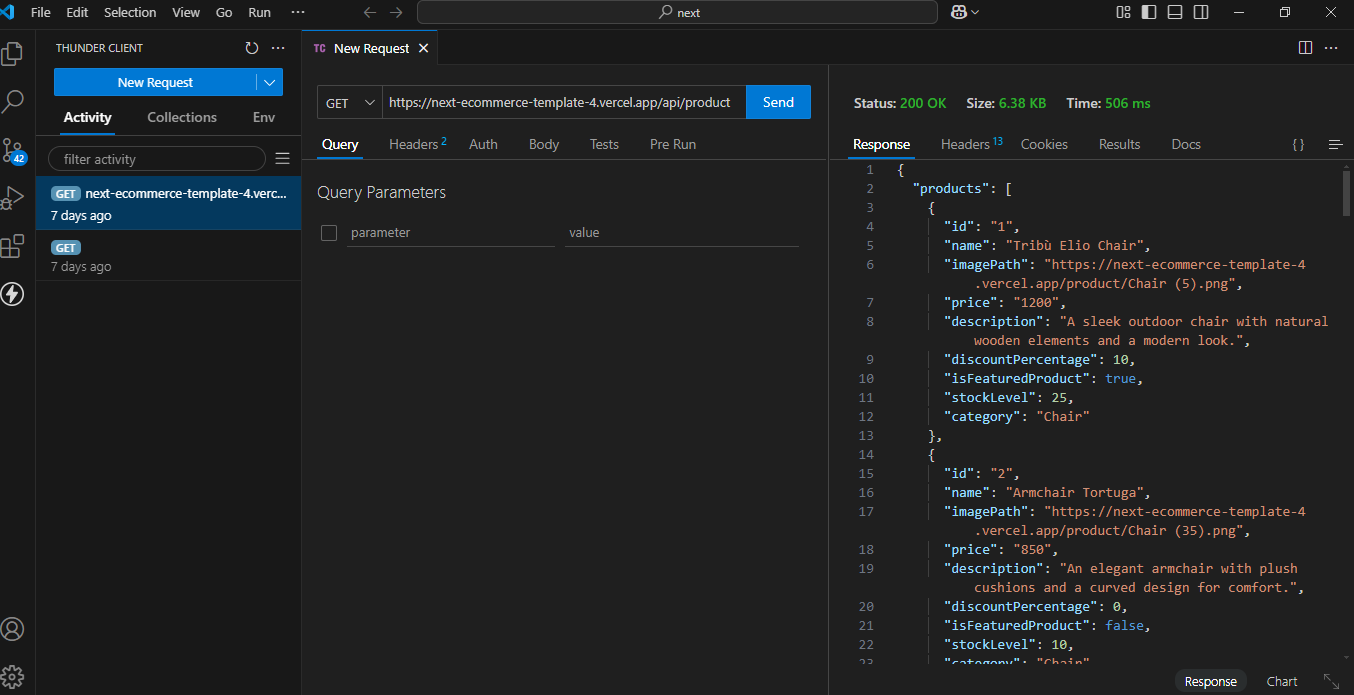
# Api Working

A key feature is the **"Add to Cart"** functionality, which operates through an **API**. When a user adds a product to the cart, a request is sent to the API, which then stores and manages the data on the backend. Additionally, **product images are dynamically fetched from the API**, ensuring that the latest visuals are always displayed. This enhances the shopping experience with real-time updates.

The website also includes a **localhost button**, which is used for testing and debugging during local development. This button helps in testing specific functionalities or triggering a local API endpoint.

If needed, more details can be added, such as the styling approach (**Tailwind CSS, Styled Components, or CSS Modules**) and any additional features.

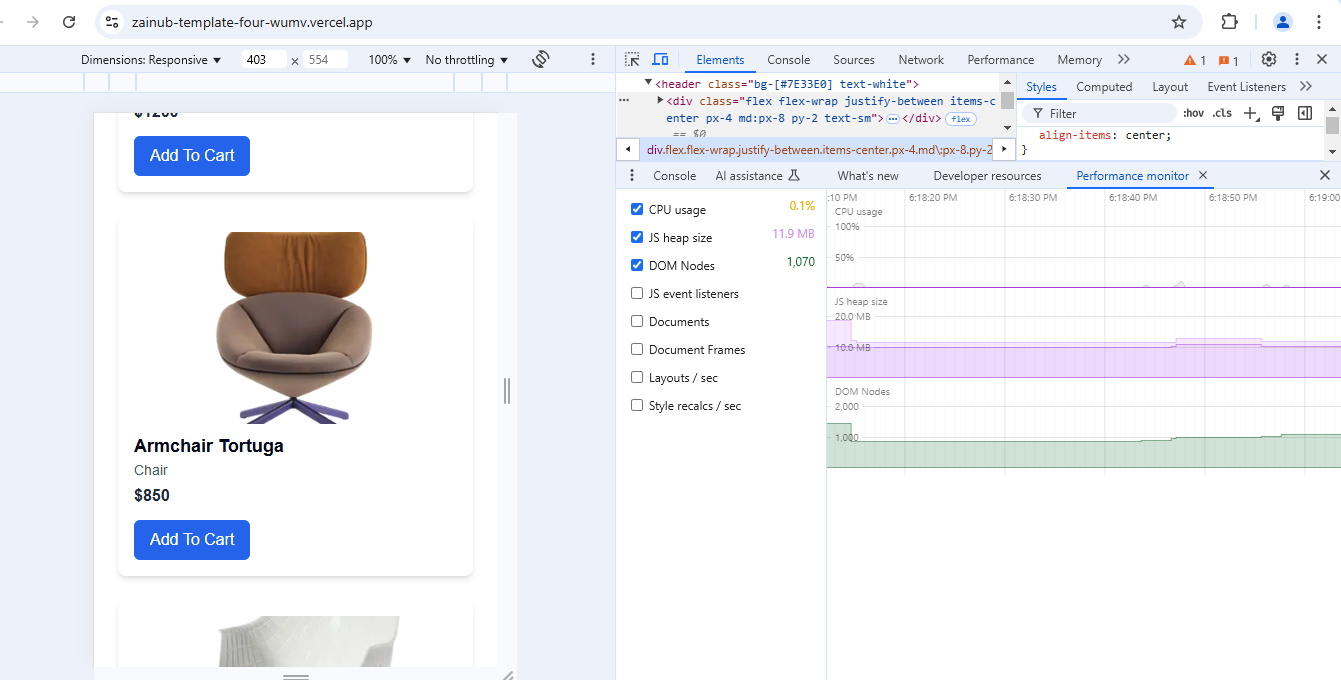
<https://next-ecommerce-template-4.vercel.app/api/product>



# Testing Report Sample.csv

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Test Steps** | **Expected Result** | **Actual Result** | **Status** | **Severity Level** | **Assigned To** | **Remarks** |
| 2 | TC001 | Validate product listing page | Open product page > Verify products | Products displayed correctly | Products displayed correctly | Passed | Low | - | No issues found |
| 3 | TC002 | Test API error handling | Disconnect API > Refresh page | Show fallback UI with error message | Error message shown | Passed | Medium | - | Handled gracefully |
| 4 | TC003 | Check cart functionality | Add product to cart > Verify cart contents | Cart updates with added product | Cart updates as expected | Passed | High | - | Works as expected |
| 5 | TC004 | Ensure responsiveness on mobile | Resize browser window > Check layout | Layout adjusts properly to screen size | Responsive layout working as intended | Passed | Medium | - | Test successful |

# Performance Monitor



# Security

