MraketPlace Furniro E-Commerce Website Documentation

Table of Contents

1. Introduction

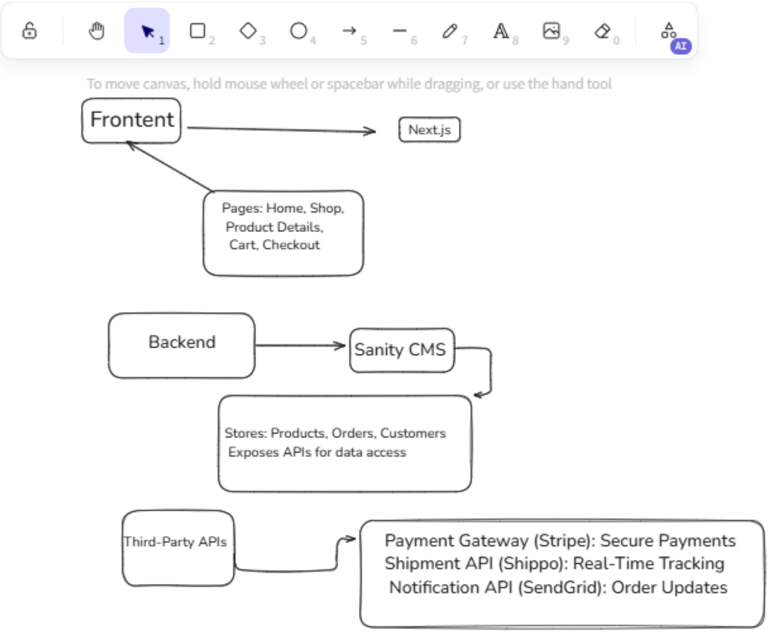
This document provides a detailed overview of the Furniro E-Commerce Website project, developed for the Marketplace Builder Hackathon 2025. It outlines the project's technical framework, development phases, core functionalities, testing approaches, and performance insights. The objective of this documentation is to facilitate understanding, ensure maintainability, and support the project's scalability.

2. Project Overview

Furniro is a cutting-edge e-commerce platform designed to offer users a seamless shopping journey. It features product listings, a shopping cart, secure checkout, and dynamic content management powered by Sanity CMS. The project utilizes Next.js for the frontend and integrates custom APIs for efficient data management.

3. Technical Architecture

Furniro follows a headless architecture to decouple the frontend from the backend, ensuring scalability and flexibility.


To move canvas, hold mouse wheel or spacebar while dragging, or use the hand tool
AI

Frontent

Next.js

Pages: Home, Shop, Product Details, Cart, Checkout

Backend

Sanity CMS

Stores: Products, Orders, Customers
Exposes APIs for data access

Third-Party APIs

Payment Gateway (Stripe): Secure Payments
Shipment API (Shippo): Real-Time Tracking
Notification API (SendGrid): Order Updates

Tools and Libraries:

Testing: **React Testing Library, Postman, OWASP ZAP**

Performance Optimization: **Lighthouse, TinyPNG**

Deployment: **Vercel**

4. Development Process

Day 1: Project Setup and Planning

Objective: Establish the project foundation and define its scope.Tasks:

Initialized the Next.js project.

Configured Sanity CMS for content management.

Defined schema for products, categories, and user data.

Planned API integration.Tools Used: Next.js, Sanity CMS, Postman.

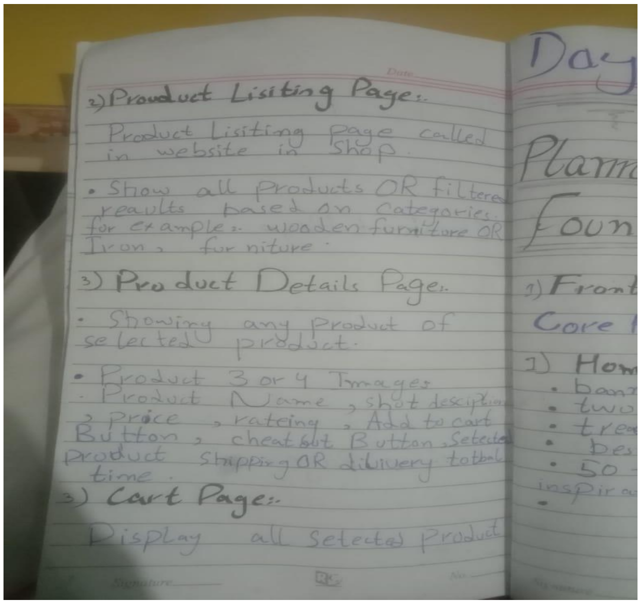
Day 2: Schema Design and Data Modeling

Objective: Develop the database schema and prepare for data migration.Tasks:

Designed schemas for products, categories, and user data in Sanity CMS.

Validated schema compatibility with API data structure.

Prepared data migration scripts.Tools Used: Sanity CMS, JSON, CSV.



2) Prouduct Lisiting Page:-

Product Lisiting page called in website in Shop.

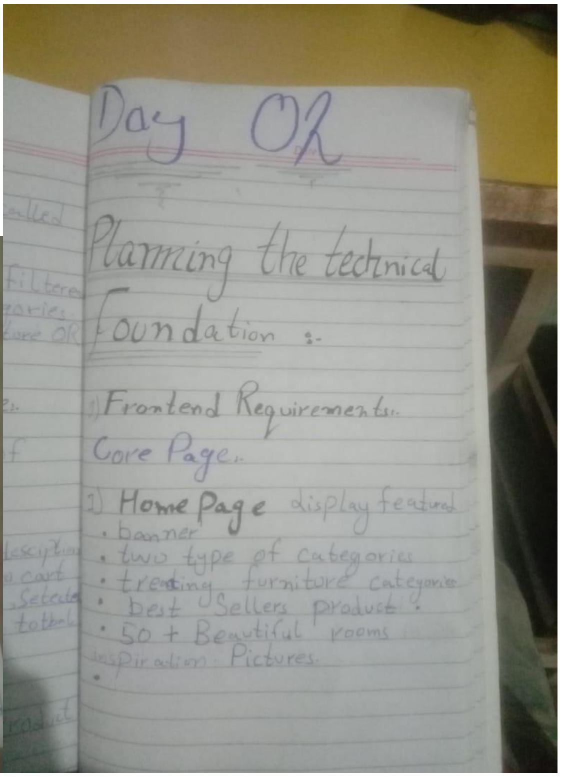
- Show all Products OR filtered reaults based on Categories. for example:- wooden furniture OR Iron, furniture.

3) Product Details Page:-

- Showing any Product of selected product.
- Product 3 or 4 Images
- Product Name, shot desciption, price, rateing, Add to cart Button, cheat out Button, Selected product shipping OR dilivery totbal time.

3) Cart Page:-

Display all Selected Product



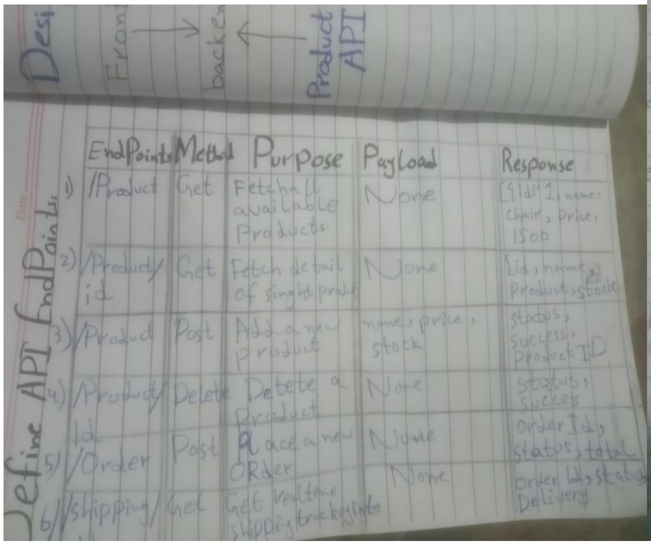
Day 02

Planning the technical Foundation :-

1) Frontend Requirements:-

Core Page:-

1) Home Page display featured
- banner
- two type of categories
- treding furniture categories
- best Sellers product.
- 50 + Beautiful rooms inspiration Pictures.
-

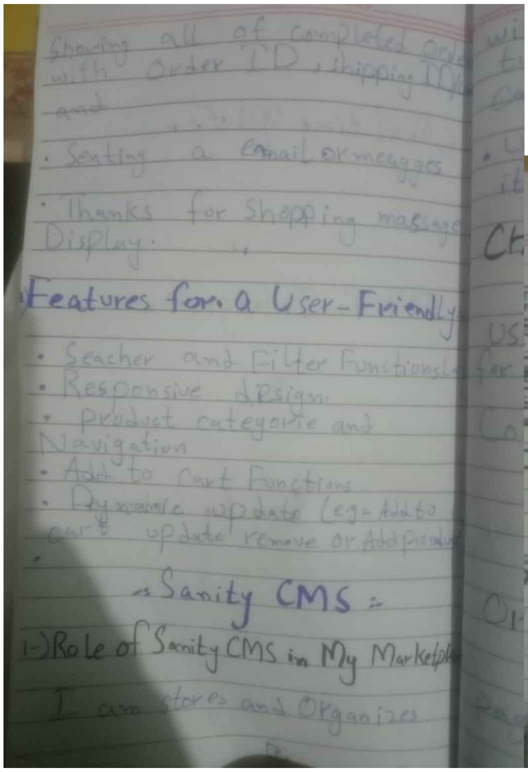


Desi

Front → backen ← Product API

Define API EndPoints:

	EndPoints	Method	Purpose	Payload	Response
1)	/Product	Get	Fetch all available Products.	None	[{id: 1, name: chair, price: 1500
2)	/Product/id	Get	Fetch detail of single product	None	{id, name, Product, stock
3)	/Product	Post	Add a new Product	name, price, stock	status, success, Product ID
4)	/Product/Id	Delete	Delete a Product	None	status, success
5)	/Order	Post	Place a new ORder	None	order Id, status, total
6)	/shipping/	Get	Get realtime shipping tracking info	None	order id, status, Delivery



Showing all of completed Order with Order ID, shipping ID and
~~and~~

- Senting a email or meagges
- Thanks for Shopping massage Display.

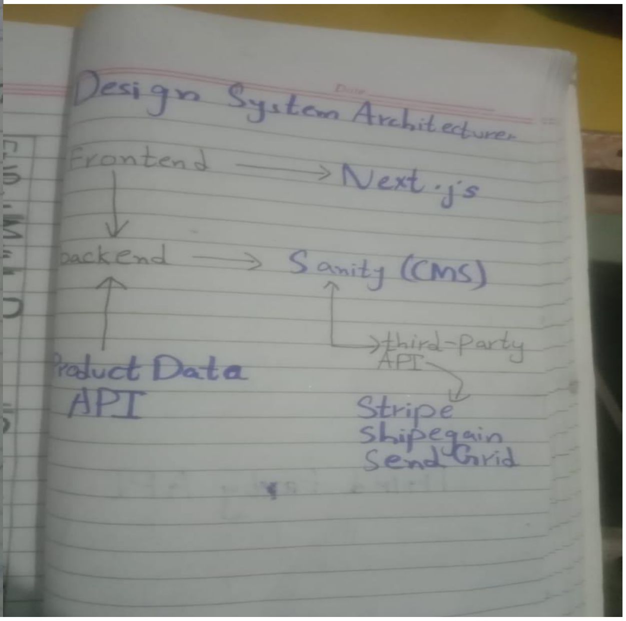
)Features for. a User-Friendly:

- Seacher and Filter Functions
- Responsive design.
- product categorie and Navigation
- Add to Cart Functions
- Dynamic update (e.g. Add to cart update remove or Add produ
-

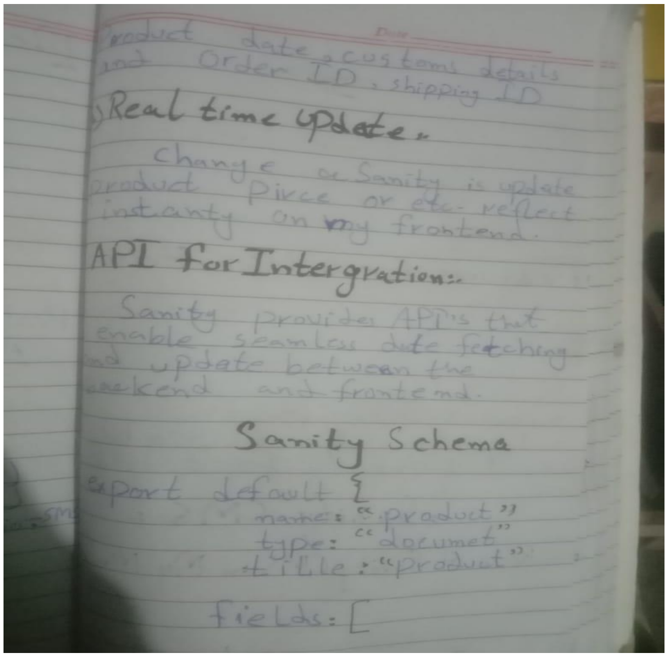
:Sanity CMS :-

1-)Role of Sanity CMS in My Marketpl

I am stores and Organizes



Design System Architecturer



product date, customs details and Order ID, shipping ID

)Real time update.

Change a Sanity is update product Pirce or etc. reflect instanty on my frontend.

API for Intergration:-

Sanity provides API's that enable seamless date fetching and update between the backend and frontend.

Sanity Schema

```
export default {
        name: "product"
        type: "documet"
       tille: "product"

     fields: [
```

Day 3: API Integration and Data Migration

Objective: Integrate APIs and migrate data to Sanity CMS.Tasks:

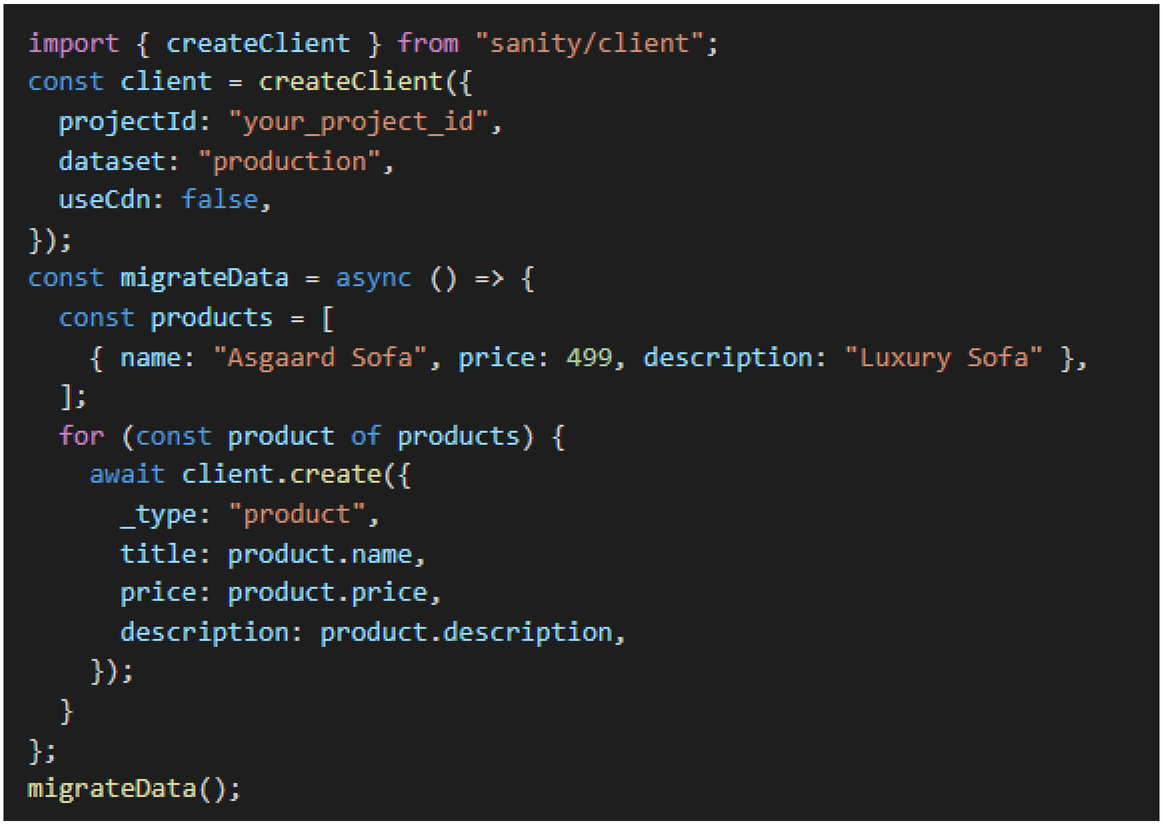
Reviewed API documentation and identified endpoints.

Aligned the Sanity schema with API data.

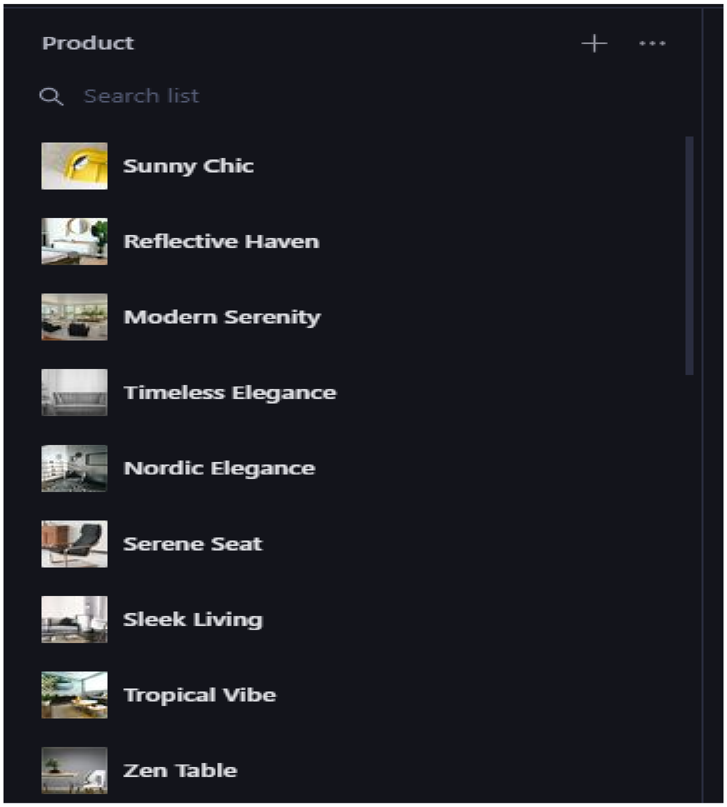
Migrated data using scripts and manual import.

Integrated APIs into the Next.js frontend.Tools Used: Sanity CMS, Next.js, Postman, React Testing Library.

Code Snippet:



```
import { createClient } from "sanity/client";
const client = createClient({
  projectId: "your_project_id",
  dataset: "production",
  useCdn: false,
});
const migrateData = async () => {
  const products = [
    { name: "Asgaard Sofa", price: 499, description: "Luxury Sofa" },
  ];
  for (const product of products) {
    await client.create({
      _type: "product",
      title: product.name,
      price: product.price,
      description: product.description,
    });
  }
};
migrateData();
```



Sunny Chic

Reflective Haven

Modern Serenity

Timeless Elegance

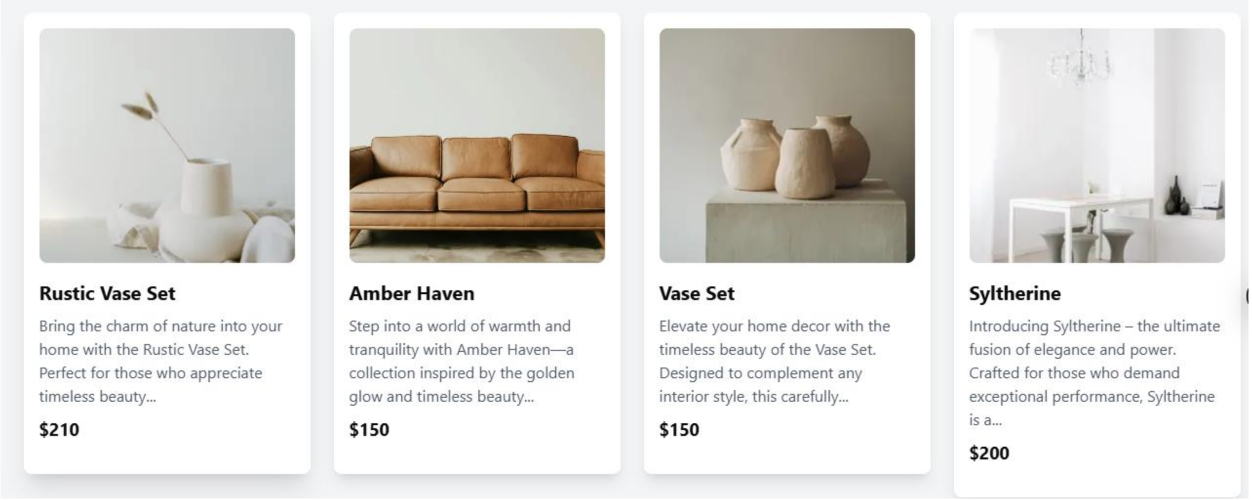
Nordic Elegance

Serene Seat

Sleek Living

Tropical Vibe

Zen Table



Rustic Vase Set

Bring the charm of nature into your home with the Rustic Vase Set. Perfect for those who appreciate timeless beauty...

$210

Amber Haven

Step into a world of warmth and tranquility with Amber Haven—a collection inspired by the golden glow and timeless beauty...

$150

Vase Set

Elevate your home decor with the timeless beauty of the Vase Set. Designed to complement any interior style, this carefully...

$150

Syltherine

Introducing Syltherine – the ultimate fusion of elegance and power. Crafted for those who demand exceptional performance, Syltherine is a...

$200

Day 4: Testing and Debugging

Objective: Ensure website functionality and reliability.Tasks:

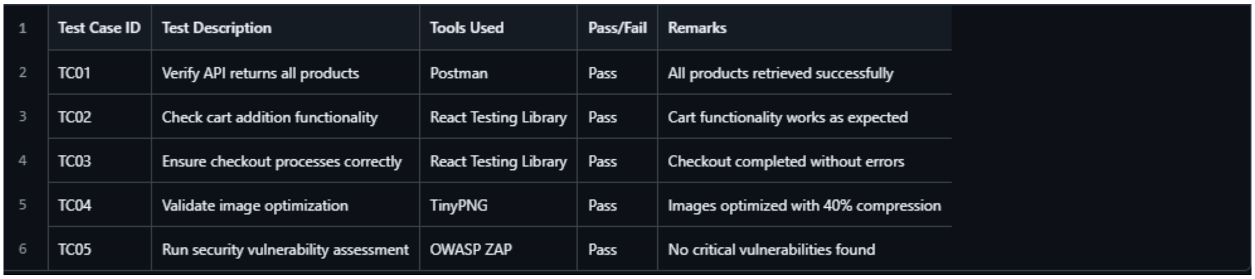
Verified API responses using Postman.

Tested cart addition and checkout functionalities.

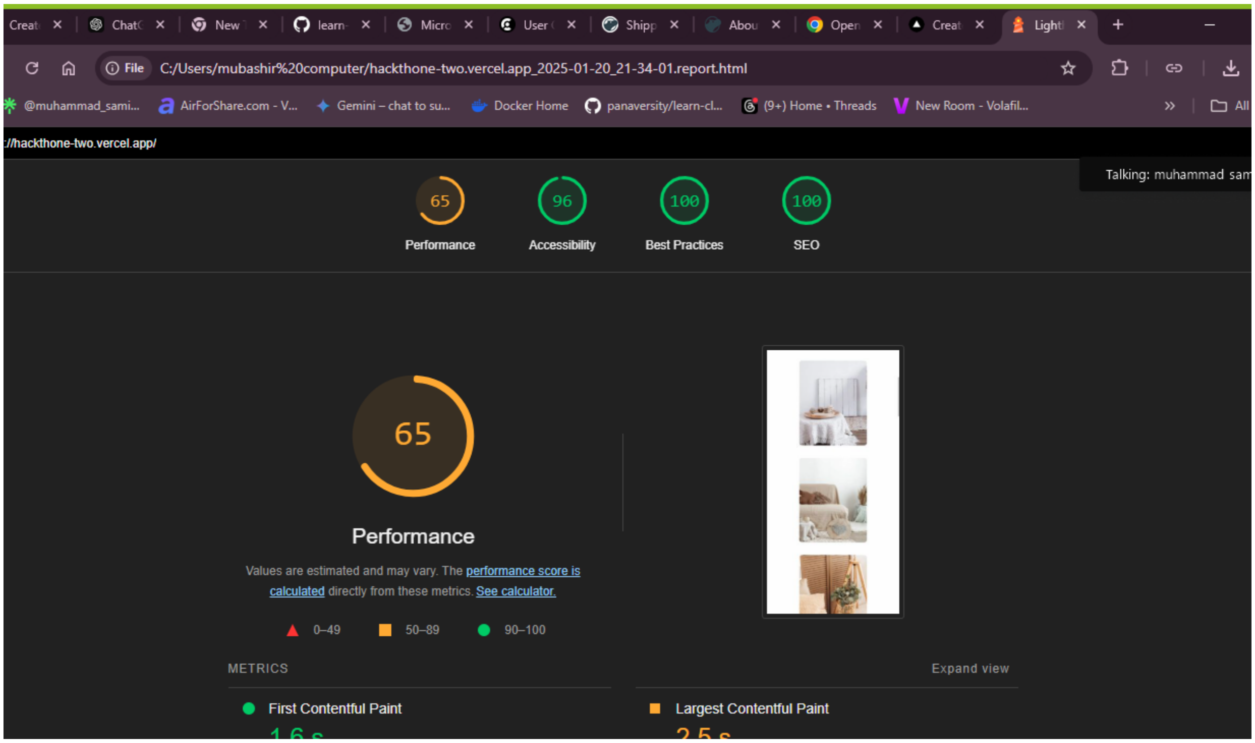
Optimized images using TinyPNG.

Conducted security assessments using OWASP ZAP.

Testing Report:



Test Case ID	Test Description	Tools Used	Pass/Fail	Remarks
TC01	Verify API returns all products	Postman	Pass	All products retrieved successfully
TC02	Check cart addition functionality	React Testing Library	Pass	Cart functionality works as expected
TC03	Ensure checkout processes correctly	React Testing Library	Pass	Checkout completed without errors
TC04	Validate image optimization	TinyPNG	Pass	Images optimized with 40% compression
TC05	Run security vulnerability assessment	OWASP ZAP	Pass	No critical vulnerabilities found



File C:/Users/mubashir%20computer/hackthone-two.vercel.app_2025-01-20_21-34-01.report.html

://hackthone-two.vercel.app/

65 Performance

96 Accessibility

100 Best Practices

100 SEO

65

Performance

Values are estimated and may vary. The performance score is calculated directly from these metrics. See calculator.

▲ 0–49 ■ 50–89 ● 90–100

METRICS

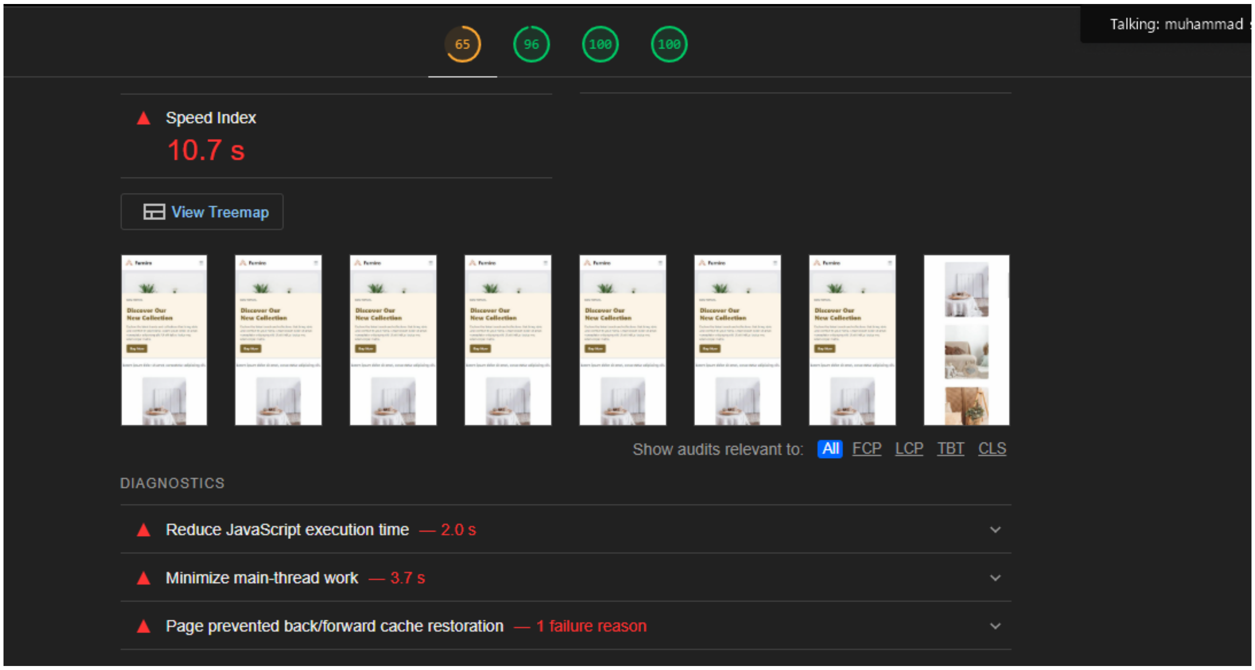
Expand view

● First Contentful Paint

1.6 s

■ Largest Contentful Paint

2.5 s



65 96 100 100

Talking: muhammad

▲ Speed Index
10.7 s

View Treemap

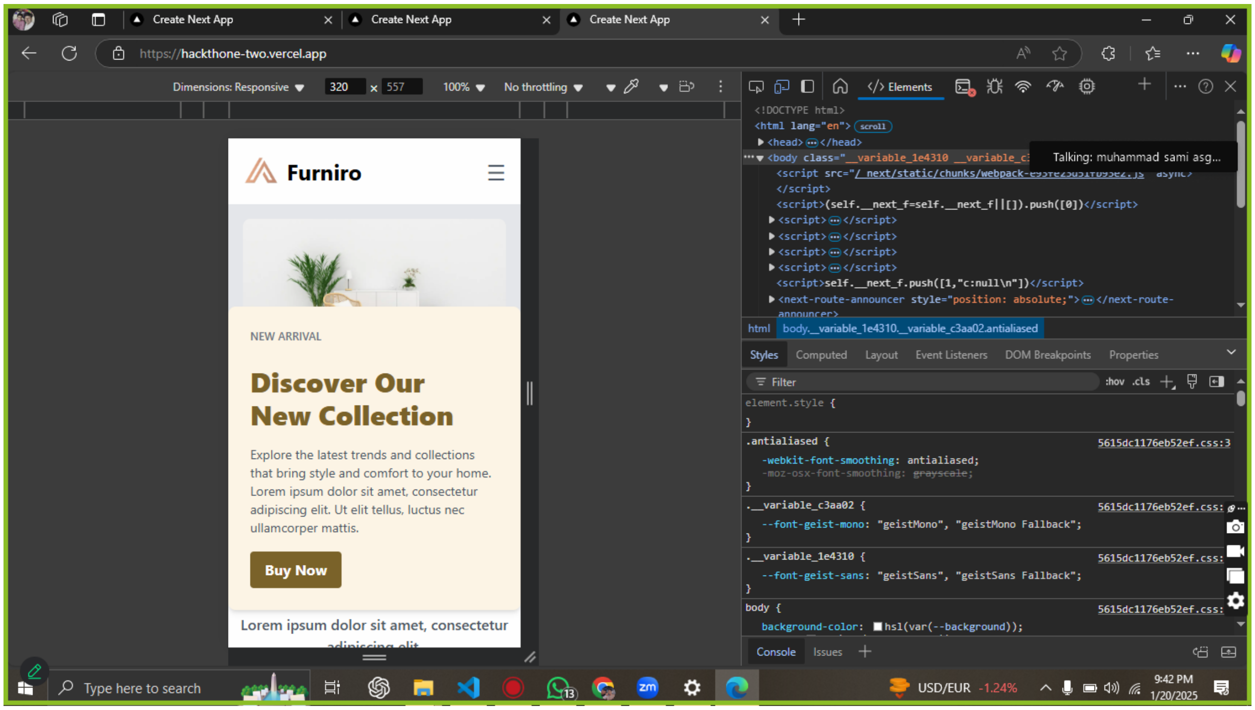
Show audits relevant to: All FCP LCP TBT CLS

DIAGNOSTICS

▲ Reduce JavaScript execution time — 2.0 s

▲ Minimize main-thread work — 3.7 s

▲ Page prevented back/forward cache restoration — 1 failure reason



Furniro

NEW ARRIVAL

Discover Our New Collection

Explore the latest trends and collections that bring style and comfort to your home. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut elit tellus, luctus nec ullamcorper mattis.

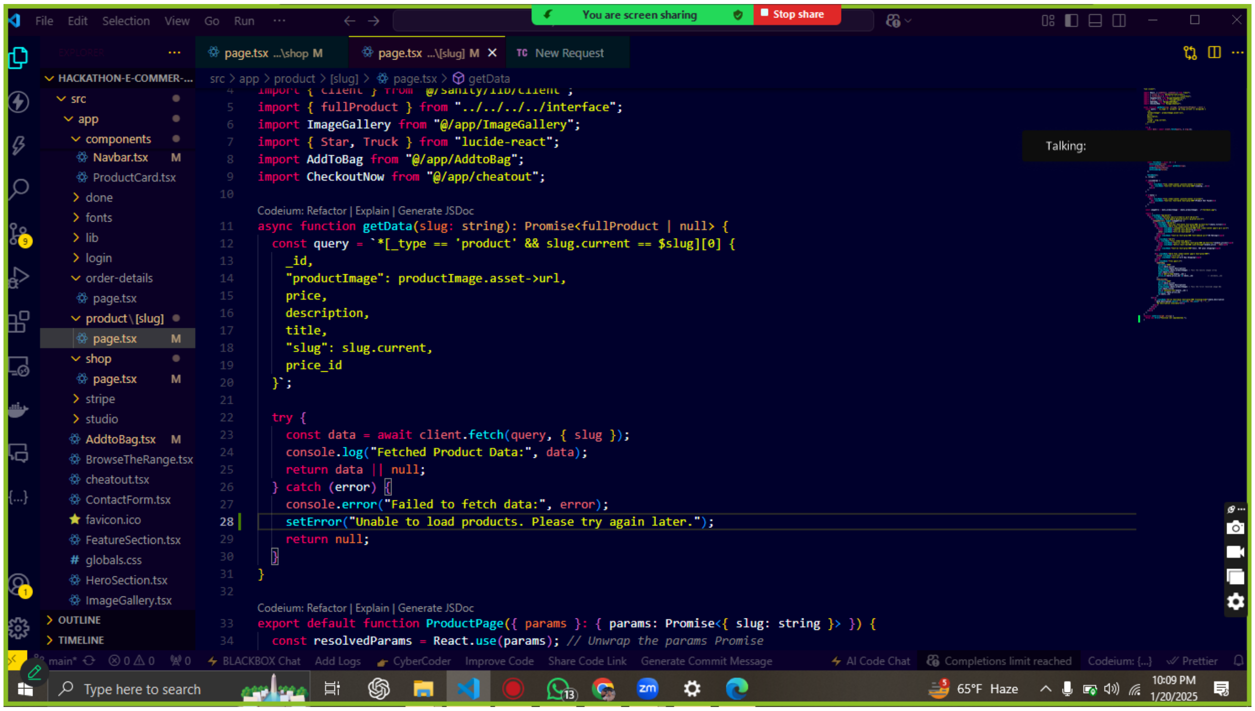
Buy Now

Lorem ipsum dolor sit amet, consectetur adipiscing elit

```
<!DOCTYPE html>
<html lang="en"> scroll
 ▶<head> ··· </head>
 ▼<body class="__variable_1e4310 __variable_c
    <script src="/_next/static/chunks/webpack-e95fe25d51fb95e2.js" async>
    </script>
    <script>(self.__next_f=self.__next_f||[]).push([0])</script>
   ▶<script> ··· </script>
   ▶<script> ··· </script>
   ▶<script> ··· </script>
   ▶<script> ··· </script>
    <script>self.__next_f.push([1,"c:null\n"])</script>
   ▶<next-route-announcer style="position: absolute;"> ··· </next-route-
    announcer>
```

html   body.__variable_1e4310.__variable_c3aa02.antialiased

```
element.style {
}
.antialiased {                                        5615dc1176eb52ef.css:3
   -webkit-font-smoothing: antialiased;
   -moz-osx-font-smoothing: ~~grayscale~~;
}
.__variable_c3aa02 {                                  5615dc1176eb52ef.css:
   --font-geist-mono: "geistMono", "geistMono Fallback";
}
.__variable_1e4310 {                                  5615dc1176eb52ef.css:
   --font-geist-sans: "geistSans", "geistSans Fallback";
}
body {                                                5615dc1176eb52ef.css:
   background-color: hsl(var(--background));
```



HACKATHON-E-COMMER-...
- src
  - app
    - components
      - Navbar.tsx M
      - ProductCard.tsx
    - done
    - fonts
    - lib
    - login
    - order-details
      - page.tsx
    - product \ [slug]
      - page.tsx M
    - shop
      - page.tsx M
    - stripe
    - studio
    - AddtoBag.tsx M
    - BrowseTheRange.tsx
    - cheatout.tsx
    - ContactForm.tsx
    - favicon.ico
    - FeatureSection.tsx
    - globals.css
    - HeroSection.tsx
    - ImageGallery.tsx
- OUTLINE
- TIMELINE

src > app > product > [slug] > page.tsx > getData

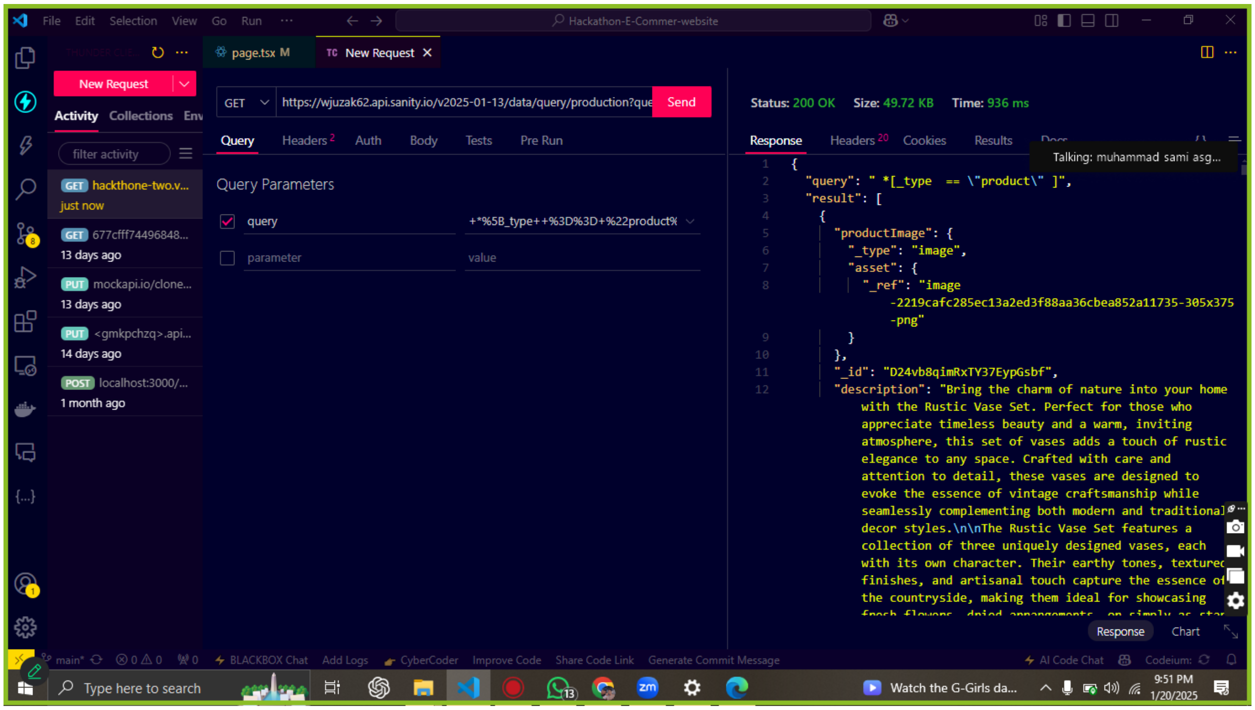
```
import { client } from "@/sanity/lib/client";
import { fullProduct } from "../../../../interface";
import ImageGallery from "@/app/ImageGallery";
import { Star, Truck } from "lucide-react";
import AddToBag from "@/app/AddtoBag";
import CheckoutNow from "@/app/cheatout";

Codeium: Refactor | Explain | Generate JSDoc
async function getData(slug: string): Promise<fullProduct | null> {
  const query = `*[_type == 'product' && slug.current == $slug][0] {
    _id,
    "productImage": productImage.asset->url,
    price,
    description,
    title,
    "slug": slug.current,
    price_id
  }`;

  try {
    const data = await client.fetch(query, { slug });
    console.log("Fetched Product Data:", data);
    return data || null;
  } catch (error) {
    console.error("Failed to fetch data:", error);
    setError("Unable to load products. Please try again later.");
    return null;
  }
}

Codeium: Refactor | Explain | Generate JSDoc
export default function ProductPage({ params }: { params: Promise<{ slug: string }> }) {
  const resolvedParams = React.use(params); // Unwrap the params Promise
```

main* ⊗ 0 ⚠ 0 BLACKBOX Chat Add Logs CyberCoder Improve Code Share Code Link Generate Commit Message AI Code Chat Completions limit reached Codeium: {...} Prettier



File Edit Selection View Go Run ···
Hackathon-E-Commer-website
page.tsx M
New Request
Activity Collections Env
filter activity
Query Headers 2 Auth Body Tests Pre Run
Response Headers 20 Cookies Results Docs
Response Chart
Talking: muhammad sami asg...
Type here to search
Watch the G-Girls da...
9:51 PM
1/20/2025

GET https://wjuzak62.api.sanity.io/v2025-01-13/data/query/production?que Send

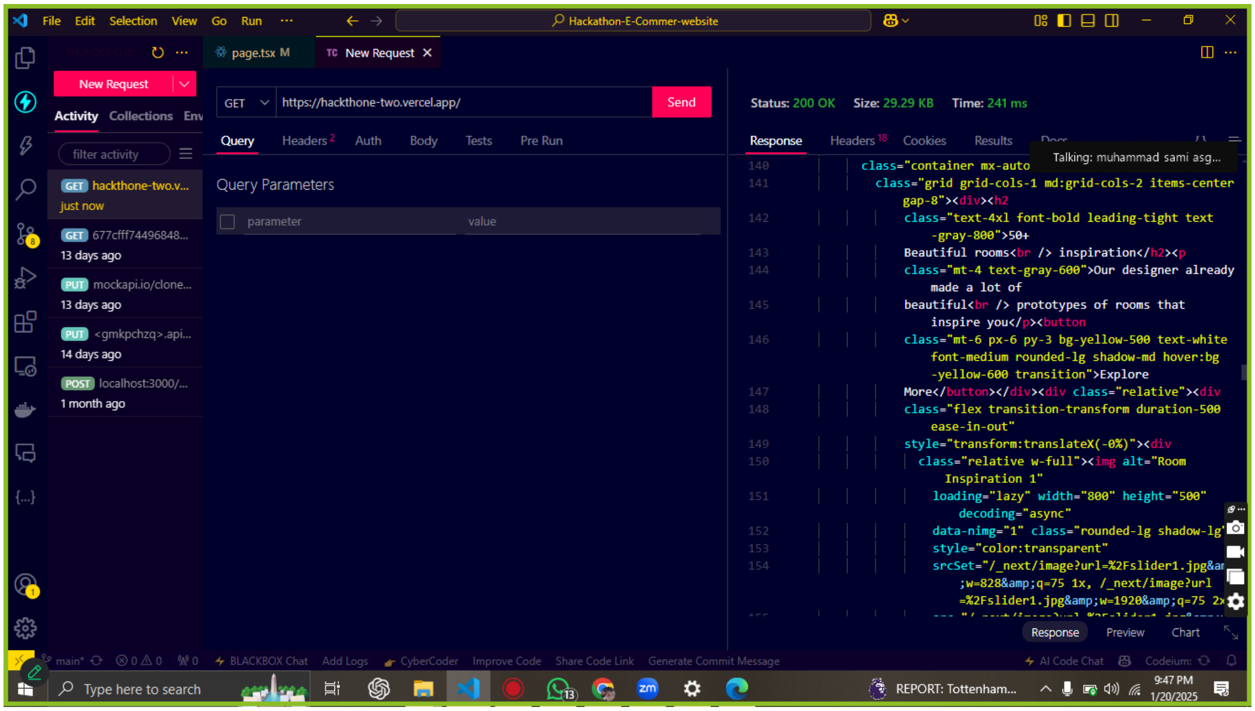
Query Parameters

☑ query +*%5B_type++%3D%3D+%22product%

□ parameter value

Status: 200 OK Size: 49.72 KB Time: 936 ms

```
{
  "query": " *[_type  == \"product\" ]",
  "result": [
    {
      "productImage": {
        "_type": "image",
        "asset": {
          "_ref": "image
            -2219cafc285ec13a2ed3f88aa36cbea852a11735-305x375
            -png"
        }
      },
      "_id": "D24vb8qimRxTY37EypGsbf",
      "description": "Bring the charm of nature into your home
        with the Rustic Vase Set. Perfect for those who
        appreciate timeless beauty and a warm, inviting
        atmosphere, this set of vases adds a touch of rustic
        elegance to any space. Crafted with care and
        attention to detail, these vases are designed to
        evoke the essence of vintage craftsmanship while
        seamlessly complementing both modern and traditional
        decor styles.\n\nThe Rustic Vase Set features a
        collection of three uniquely designed vases, each
        with its own character. Their earthy tones, texturec
        finishes, and artisanal touch capture the essence of
        the countryside, making them ideal for showcasing
```



File Edit Selection View Go Run ···
Hackathon-E-Commer-website
page.tsx M
New Request
Activity Collections Env
filter activity
Query Headers 2 Auth Body Tests Pre Run
Response Headers 18 Cookies Results Docs
Response Preview Chart
Talking: muhammad sami asg...
Type here to search
REPORT: Tottenham...
9:47 PM
1/20/2025

GET https://hackthone-two.vercel.app/ Send

Query Parameters

□ parameter value

Status: 200 OK Size: 29.29 KB Time: 241 ms

```
class="container mx-auto
  class="grid grid-cols-1 md:grid-cols-2 items-center
    gap-8"><div><h2
    class="text-4xl font-bold leading-tight text
      -gray-800">50+
    Beautiful rooms<br /> inspiration</h2><p
    class="mt-4 text-gray-600">Our designer already
      made a lot of
    beautiful<br /> prototypes of rooms that
      inspire you</p><button
    class="mt-6 px-6 py-3 bg-yellow-500 text-white
      font-medium rounded-lg shadow-md hover:bg
      -yellow-600 transition">Explore
    More</button></div><div class="relative"><div
    class="flex transition-transform duration-500
      ease-in-out"
    style="transform:translateX(-0%)"><div
      class="relative w-full"><img alt="Room
        Inspiration 1"
        loading="lazy" width="800" height="500"
          decoding="async"
        data-nimg="1" class="rounded-lg shadow-lg"
        style="color:transparent"
        srcSet="/_next/image?url=%2Fslider1.jpg&am
          ;w=828&amp;q=75 1x, /_next/image?url
          =%2Fslider1.jpg&amp;w=1920&amp;q=75 2x
```

Day 5: Backend Refinement and Performance Optimization

Objective: Enhance performance and scalability.Tasks:

Minimized JavaScript execution time.

Implemented caching strategies.

Assessed performance via Lighthouse.Performance Metrics:

First Contentful Paint: 1.6s

Largest Contentful Paint: 2.5s

Speed Index: 10.7s

Performance Score: 65/100

Day 6: Frontend Enhancements and User Experience

Objective: Enhance UI and responsiveness.Tasks:

Added interactive elements and animations.

Improved responsiveness across devices.Tools Used: React, Tailwind CSS, Chrome DevTools.

Day 7: Final Testing and Deployment

Objective: Deploy and finalize project.Tasks:

Conducted end-to-end testing.

Deployed to Vercel.Tools Used: Vercel, Postman, React Testing Library.

5. Key Features

Dynamic product listings.

Shopping cart and checkout.

Responsive design.

Sanity CMS integration.

6. Testing and Quality Assurance

Unit Testing: React Testing Library

API Testing: Postman

Security Testing: OWASP ZAP

Performance Testing: Lighthouse