MARKETPLACE BUSINESS PLAN DAY 3

Marketplace Implementation – SHOP.CO

I've completed my last day-1 and day-2 task now I'm going to validate my all plans which I've completed in last 2 days.

Here is breakdowns of my day3 plannings:

1. Functional Prototype

This section explains the features implemented in your marketplace. It describe how users can interact with the platform and the purpose of each functionality. Now here is step by step guide how we can implement this;

- **User Authentication:**

  Users can register and log in securely to access the marketplace features.

- **Product Browsing:**

  A visually appealing interface where users can view and filter products.

- **Order Placement:**

  A seamless checkout process that allows users to place orders with a confirmation page.

- **Sanity CMS Integration:**

  Enables efficient management of products, orders, and other content.

- **Third-party API Integration:**

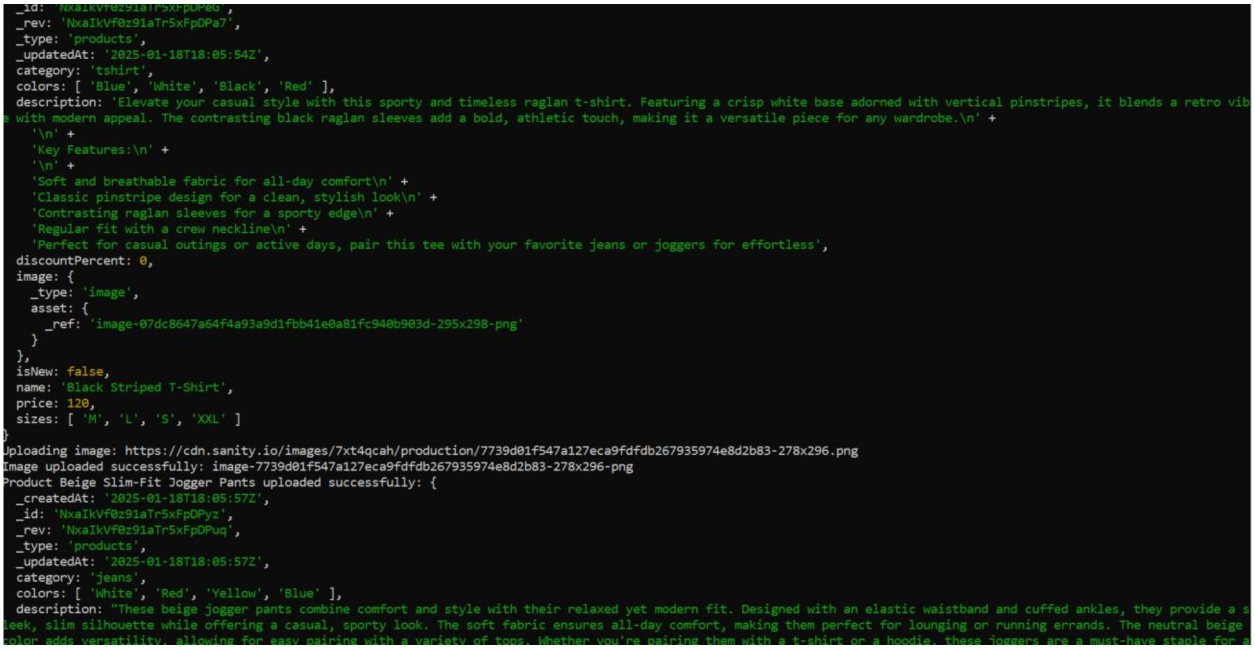
  Added functionalities such as payment gateways, product recommendations, etc., through APIs.

For Example:

Our marketplace prototype is designed to streamline the shopping experience. It allows users to explore a wide range of products, add items to the cart, and securely place orders. The backend integrates with Sanity CMS for smooth content management, ensuring efficiency and scalability.

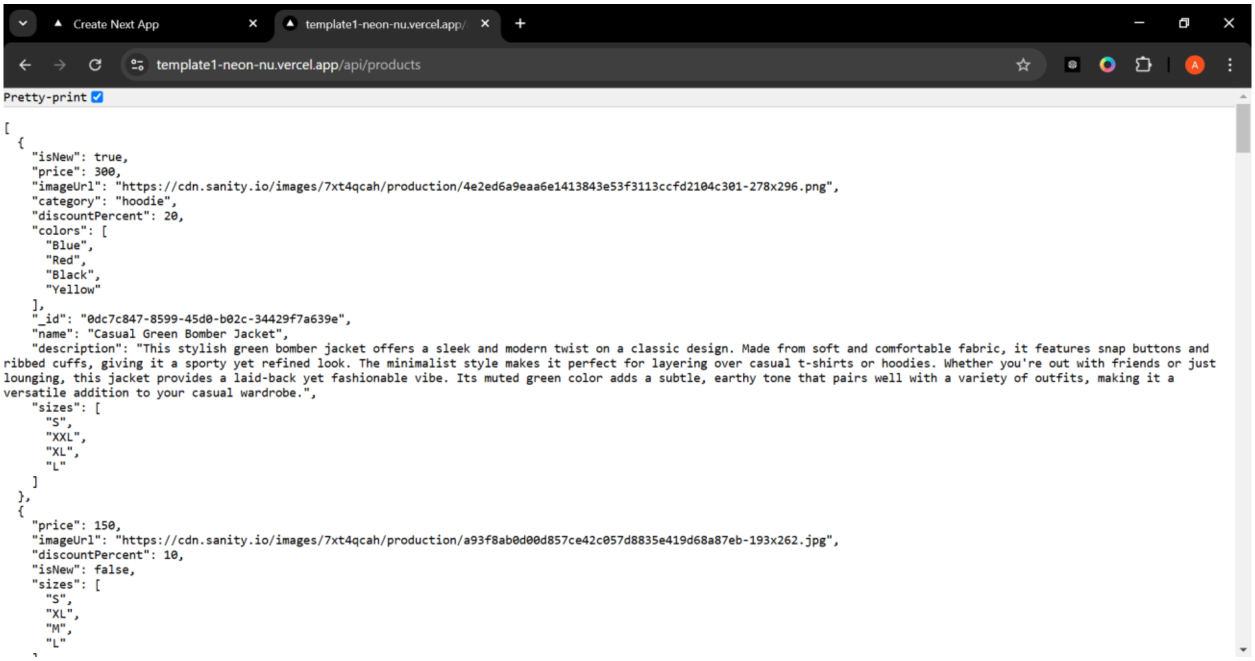
2. User Interface Screenshots

This section contains screenshots of key pages of my marketplace. Here is my explanation of my each page .



```
  _id: 'NxaIkVf0z91aTr5xFpDPeG',
  _rev: 'NxaIkVf0z91aTr5xFpDPa7',
  _type: 'products',
  _updatedAt: '2025-01-18T18:05:54Z',
  category: 'tshirt',
  colors: [ 'Blue', 'White', 'Black', 'Red' ],
  description: 'Elevate your casual style with this sporty and timeless raglan t-shirt. Featuring a crisp white base adorned with vertical pinstripes, it blends a retro vib
e with modern appeal. The contrasting black raglan sleeves add a bold, athletic touch, making it a versatile piece for any wardrobe.\n' +
    '\n' +
    'Key Features:\n' +
    '\n' +
    'Soft and breathable fabric for all-day comfort\n' +
    'Classic pinstripe design for a clean, stylish look\n' +
    'Contrasting raglan sleeves for a sporty edge\n' +
    'Regular fit with a crew neckline\n' +
    'Perfect for casual outings or active days, pair this tee with your favorite jeans or joggers for effortless',
  discountPercent: 0,
  image: {
    _type: 'image',
    asset: {
      _ref: 'image-07dc8647a64f4a93a9d1fbb41e0a81fc940b903d-295x298-png'
    }
  },
  isNew: false,
  name: 'Black Striped T-Shirt',
  price: 120,
  sizes: [ 'M', 'L', 'S', 'XXL' ]
}
Uploading image: https://cdn.sanity.io/images/7xt4qcah/production/7739d01f547a127eca9fdfdb267935974e8d2b83-278x296.png
Image uploaded successfully: image-7739d01f547a127eca9fdfdb267935974e8d2b83-278x296-png
Product Beige Slim-Fit Jogger Pants uploaded successfully: {
  _createdAt: '2025-01-18T18:05:57Z',
  _id: 'NxaIkVf0z91aTr5xFpDPyz',
  _rev: 'NxaIkVf0z91aTr5xFpDPuq',
  _type: 'products',
  _updatedAt: '2025-01-18T18:05:57Z',
  category: 'jeans',
  colors: [ 'White', 'Red', 'Yellow', 'Blue' ],
  description: "These beige jogger pants combine comfort and style with their relaxed yet modern fit. Designed with an elastic waistband and cuffed ankles, they provide a s
leek, slim silhouette while offering a casual, sporty look. The soft fabric ensures all-day comfort, making them perfect for lounging or running errands. The neutral beige
color adds versatility, allowing for easy pairing with a variety of tops. Whether you're pairing them with a t-shirt or a hoodie, these joggers are a must-have staple for a
```

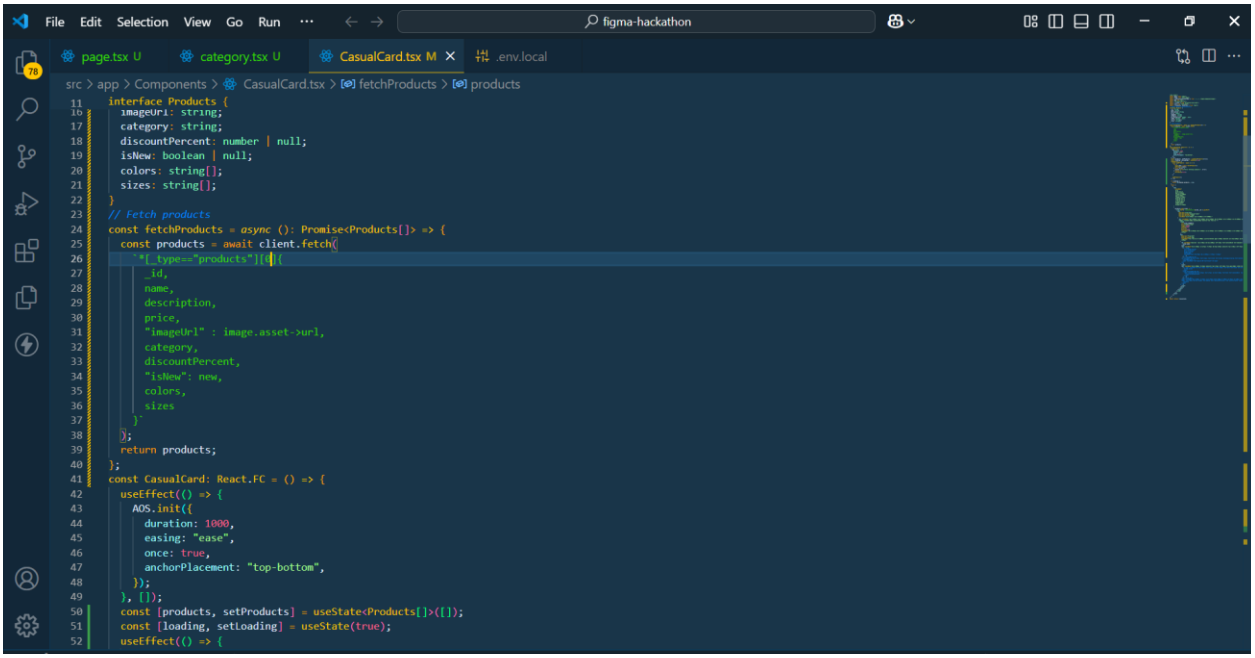
This involves including the API or a library related to it into your project files. Typically, this is done using import statements in JavaScript or TypeScript.



Create Next App
template1-neon-nu.vercel.app/
template1-neon-nu.vercel.app/api/products
Pretty-print ☑

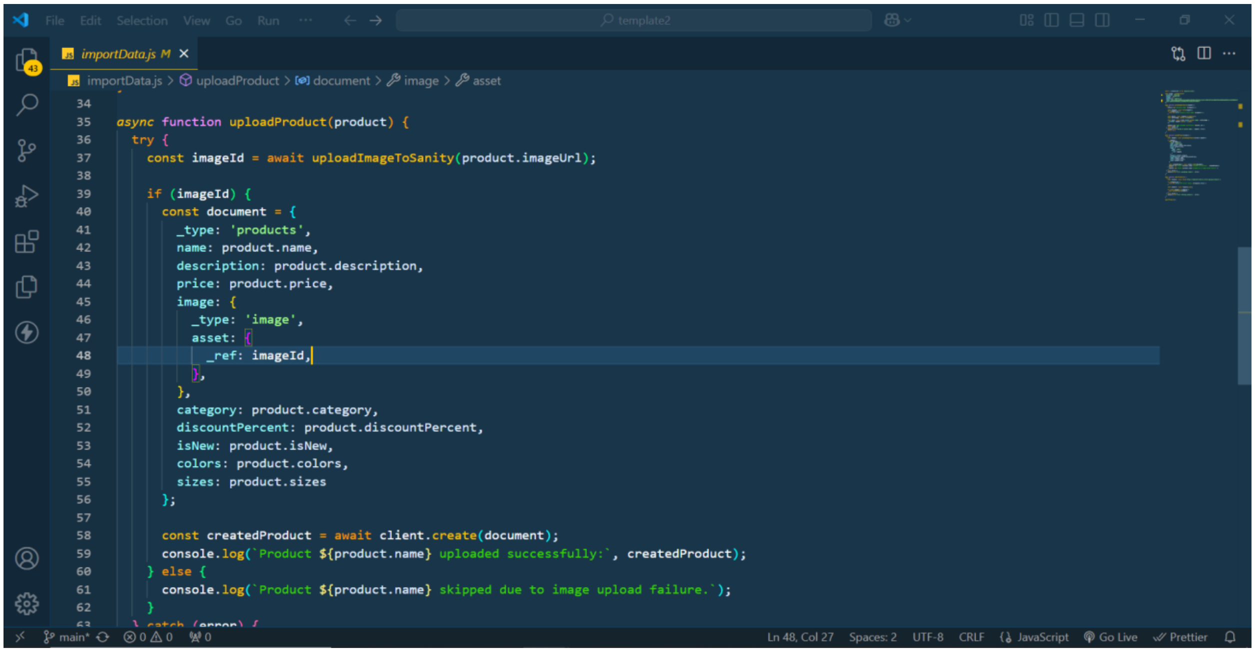
```
[
  {
    "isNew": true,
    "price": 300,
    "imageUrl": "https://cdn.sanity.io/images/7xt4qcah/production/4e2ed6a9eaa6e1413843e53f3113ccfd2104c301-278x296.png",
    "category": "hoodie",
    "discountPercent": 20,
    "colors": [
      "Blue",
      "Red",
      "Black",
      "Yellow"
    ],
    "_id": "0dc7c847-8599-45d0-b02c-34429f7a639e",
    "name": "Casual Green Bomber Jacket",
    "description": "This stylish green bomber jacket offers a sleek and modern twist on a classic design. Made from soft and comfortable fabric, it features snap buttons and
ribbed cuffs, giving it a sporty yet refined look. The minimalist style makes it perfect for layering over casual t-shirts or hoodies. Whether you're out with friends or just
lounging, this jacket provides a laid-back yet fashionable vibe. Its muted green color adds a subtle, earthy tone that pairs well with a variety of outfits, making it a
versatile addition to your casual wardrobe.",
    "sizes": [
      "S",
      "XXL",
      "XL",
      "L"
    ]
  },
  {
    "price": 150,
    "imageUrl": "https://cdn.sanity.io/images/7xt4qcah/production/a93f8ab0d00d857ce42c057d8835e419d68a87eb-193x262.jpg",
    "discountPercent": 10,
    "isNew": false,
    "sizes": [
      "S",
      "XL",
      "M",
      "L"
```

When you paste an API endpoint into the browser, it sends a GET request to the server. If the API is publicly accessible, the server responds with the data, typically in JSON format.


File Edit Selection View Go Run ···
figma-hackathon
page.tsx U
category.tsx U
CasualCard.tsx M
.env.local
src > app > Components > CasualCard.tsx > fetchProducts > products

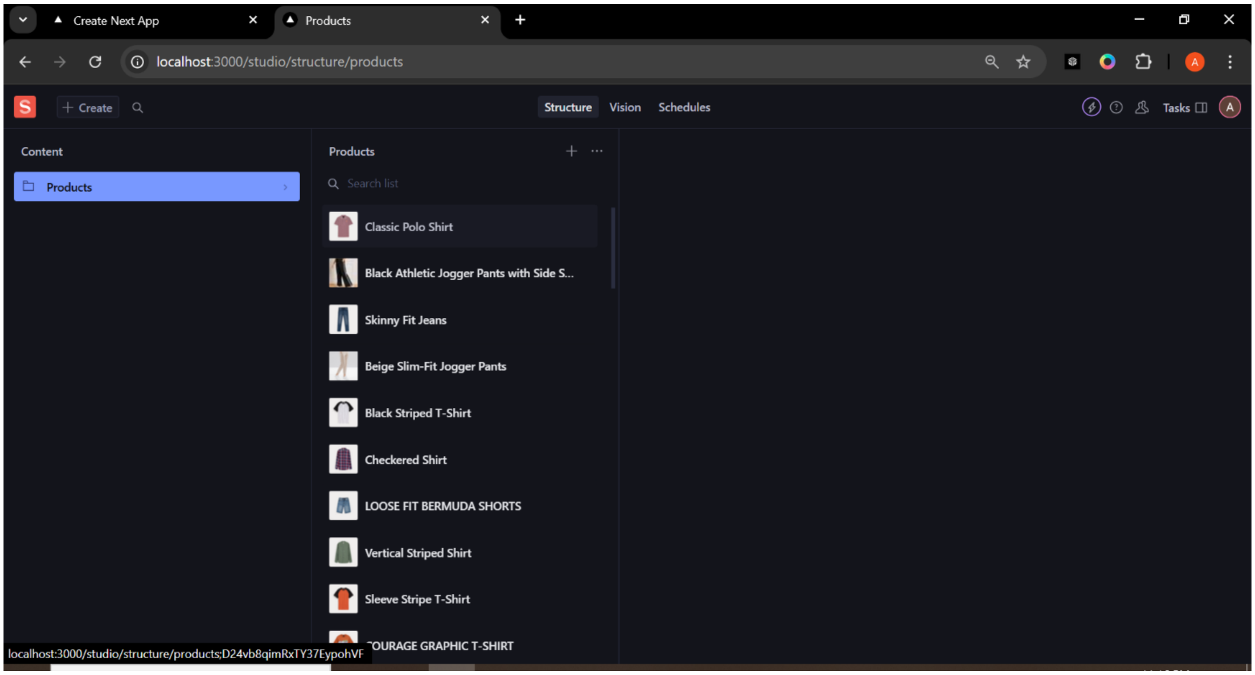
```
interface Products {
  imageUrl: string;
  category: string;
  discountPercent: number | null;
  isNew: boolean | null;
  colors: string[];
  sizes: string[];
}
// Fetch products
const fetchProducts = async (): Promise<Products[]> => {
  const products = await client.fetch(
    `*[_type=="products"][0]{
      _id,
      name,
      description,
      price,
      "imageUrl" : image.asset->url,
      category,
      discountPercent,
      "isNew": new,
      colors,
      sizes
    }`
  );
  return products;
};
const CasualCard: React.FC = () => {
  useEffect(() => {
    AOS.init({
      duration: 1000,
      easing: "ease",
      once: true,
      anchorPlacement: "top-bottom",
    });
  }, []);
  const [products, setProducts] = useState<Products[]>([]);
  const [loading, setLoading] = useState(true);
  useEffect(() => {
```

GROQ is a query language often used with tools like Sanity.io to retrieve data from a content management system (CMS).

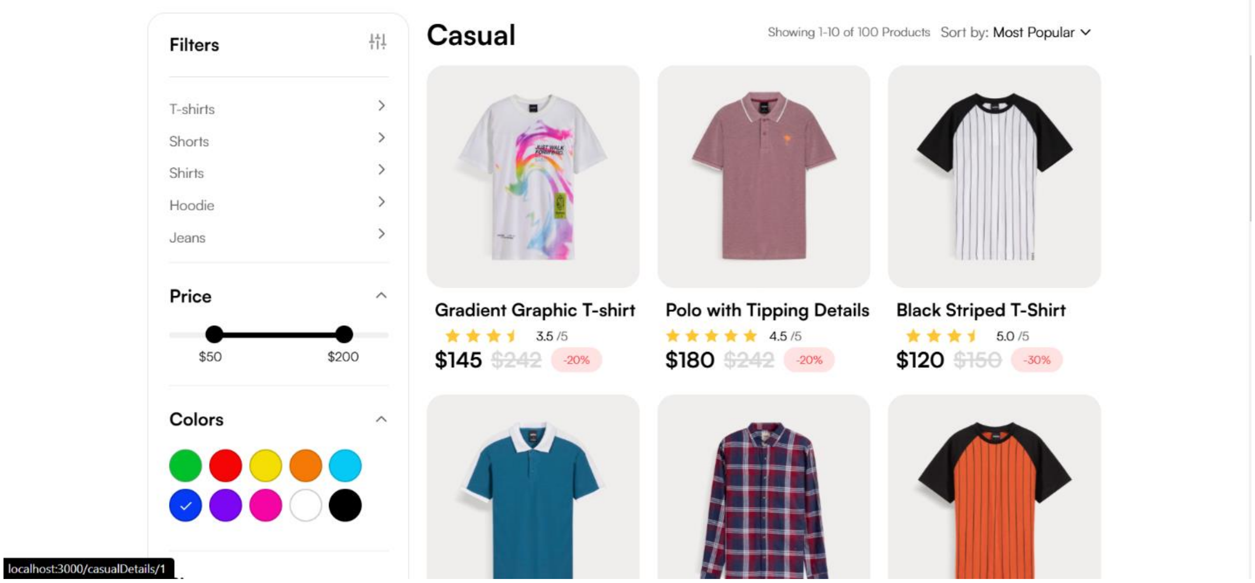

File Edit Selection View Go Run ···
template2
importData.js M
importData.js > uploadProduct > document > image > asset
main* 0 0 0
Ln 48, Col 27 Spaces: 2 UTF-8 CRLF JavaScript Go Live Prettier

```

async function uploadProduct(product) {
  try {
    const imageId = await uploadImageToSanity(product.imageUrl);

    if (imageId) {
      const document = {
        _type: 'products',
        name: product.name,
        description: product.description,
        price: product.price,
        image: {
          _type: 'image',
          asset: {
            _ref: imageId,
          },
        },
        category: product.category,
        discountPercent: product.discountPercent,
        isNew: product.isNew,
        colors: product.colors,
        sizes: product.sizes
      };

      const createdProduct = await client.create(document);
      console.log(`Product ${product.name} uploaded successfully:`, createdProduct);
    } else {
      console.log(`Product ${product.name} skipped due to image upload failure.`);
    }
  } catch (error) {
```

After creating the GROQ query, you executed it using the appropriate method (e.g., client.fetch(query) for Sanity).



Sanity Studio allows users to manage content, including adding or updating images for products.



Finally, you displayed the product details and image in a card layout on the browser.

3. Testing and Validation

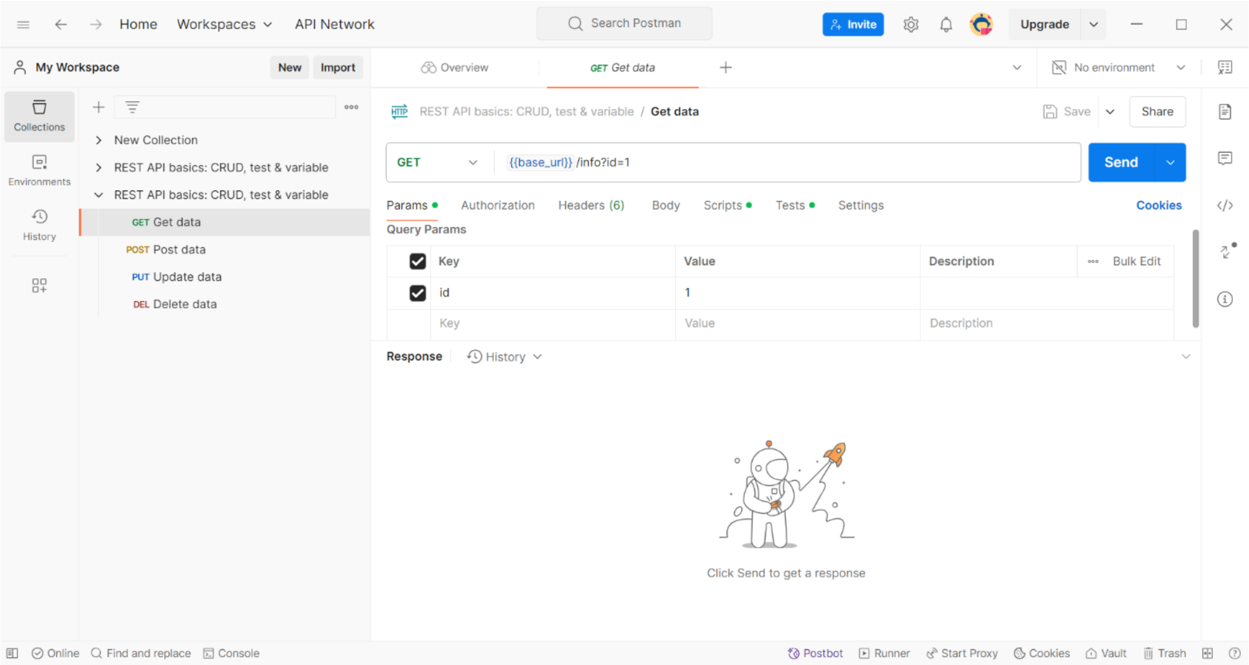
I've summarize my technical process in a single table. Here is my validation process;

Test Case	Expected Outcome	Actual Outcome	Status
Success	Successful account creation	Passed	✓
Product Filtering	Correct filtered products displayed	Passed	✓
Order Placement	Order confirmation received	Passed	✓
Api Response Timing	Data fetching within 2 seconds	Passed	✓

4. API Testing with Postman

Before using the API's I've test my API on postman I share my postman testing;

In first I've login my postman and I've generated new request and I passed the API URL and then I got 200 Status which means my API's have Completely fine.

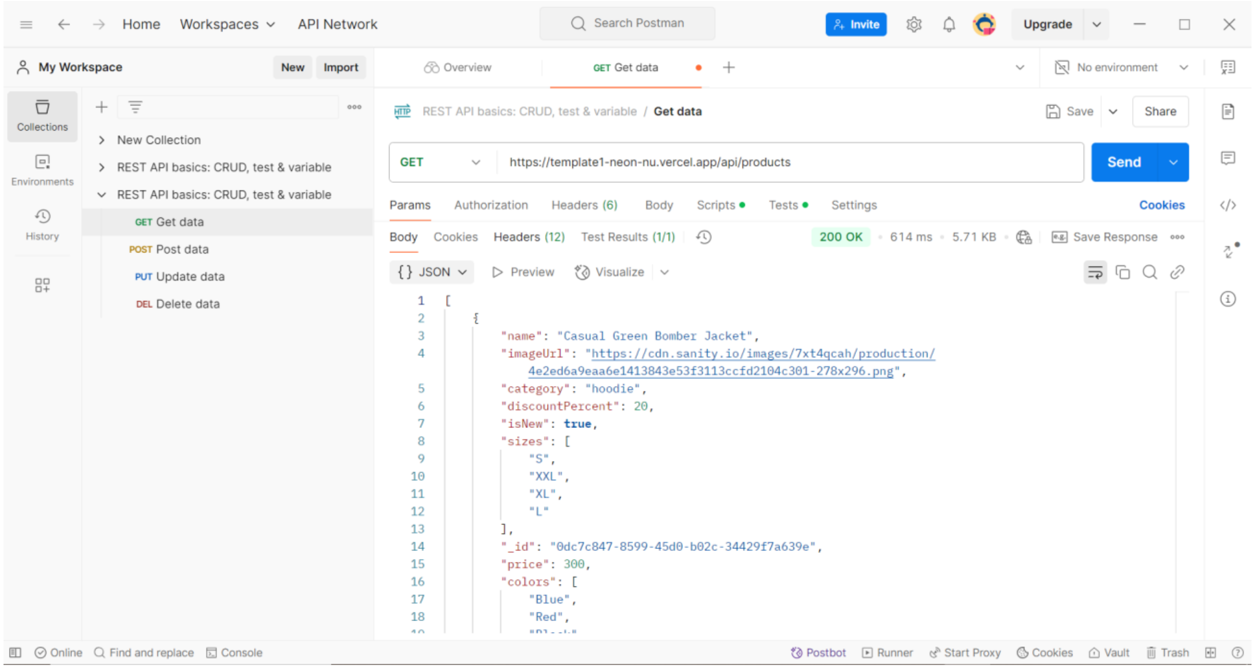


REST API basics: CRUD, test & variable / Get data

GET {{base_url}} /info?id=1

Query Params

	Key	Value	Description
☑	id	1	
	Key	Value	Description

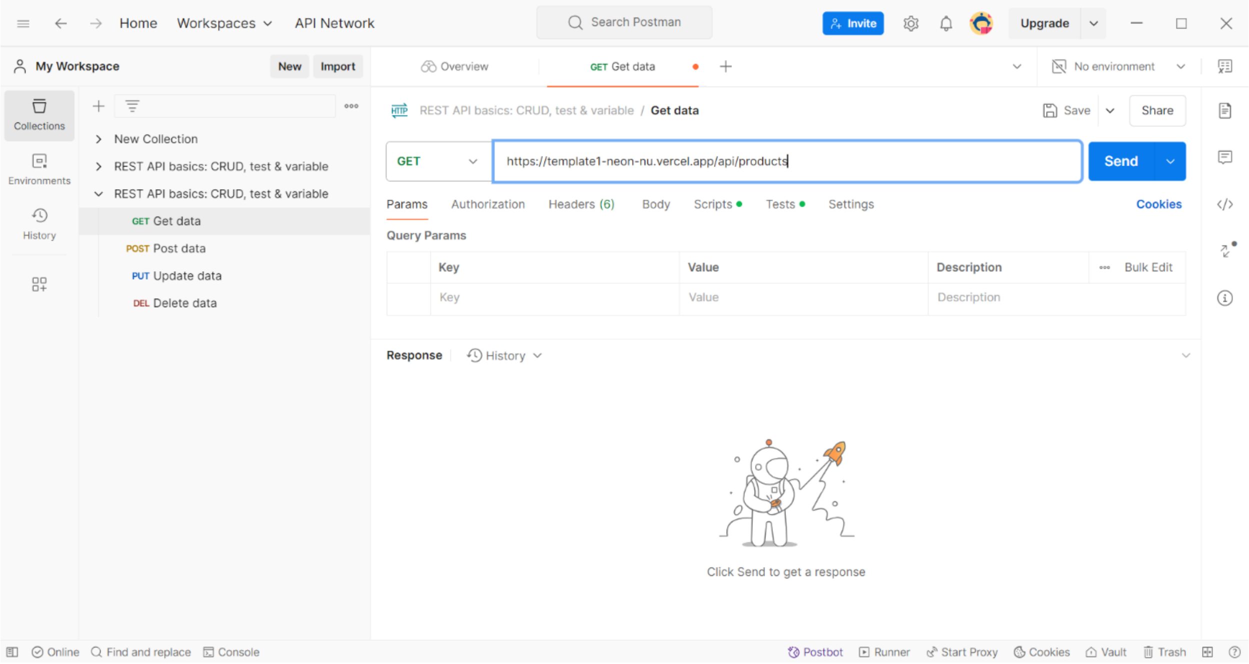


REST API basics: CRUD, test & variable / Get data

GET https://template1-neon-nu.vercel.app/api/products

200 OK • 614 ms • 5.71 KB

```
[
    {
        "name": "Casual Green Bomber Jacket",
        "imageUrl": "https://cdn.sanity.io/images/7xt4qcah/production/
            4e2ed6a9eaa6e1413843e53f3113ccfd2104c301-278x296.png",
        "category": "hoodie",
        "discountPercent": 20,
        "isNew": true,
        "sizes": [
            "S",
            "XXL",
            "XL",
            "L"
        ],
        "_id": "0dc7c847-8599-45d0-b02c-34429f7a639e",
        "price": 300,
        "colors": [
            "Blue",
            "Red",
```



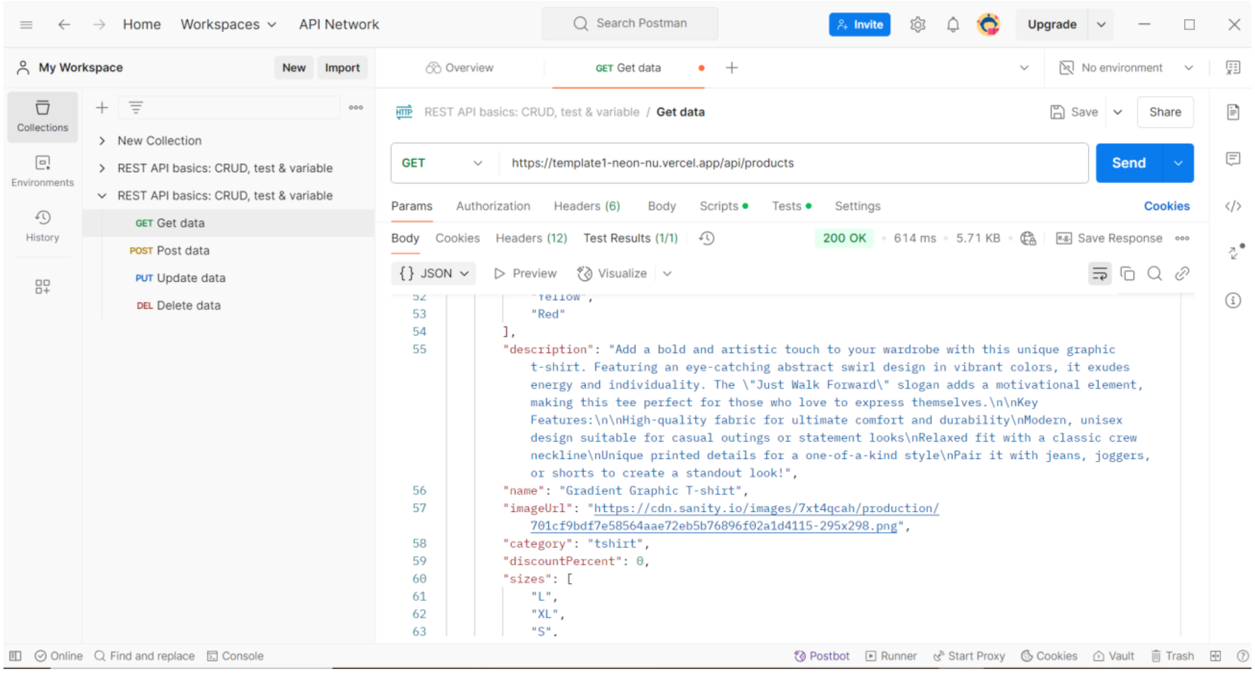
REST API basics: CRUD, test & variable / Get data

GET https://template1-neon-nu.vercel.app/api/products

Query Params

Key	Value	Description
Key	Value	Description

Click Send to get a response



GET https://template1-neon-nu.vercel.app/api/products

200 OK · 614 ms · 5.71 KB

```
            "Yellow",
            "Red"
        ],
        "description": "Add a bold and artistic touch to your wardrobe with this unique graphic
            t-shirt. Featuring an eye-catching abstract swirl design in vibrant colors, it exudes
            energy and individuality. The \"Just Walk Forward\" slogan adds a motivational element,
            making this tee perfect for those who love to express themselves.\n\nKey
            Features:\n\nHigh-quality fabric for ultimate comfort and durability\nModern, unisex
            design suitable for casual outings or statement looks\nRelaxed fit with a classic crew
            neckline\nUnique printed details for a one-of-a-kind style\nPair it with jeans, joggers,
            or shorts to create a standout look!",
        "name": "Gradient Graphic T-shirt",
        "imageUrl": "https://cdn.sanity.io/images/7xt4qcah/production/
            701cf9bdf7e58564aae72eb5b76896f02a1d4115-295x298.png",
        "category": "tshirt",
        "discountPercent": 0,
        "sizes": [
            "L",
            "XL",
            "S",
```

5. Challenges faced and Solution

During my Hackathon I've faced some challenges. I want to show my challenges in my document. Here is my challenges which I've faced in my hackathon;

- **Challenges:** The first challenge I've faced which is API when I call API's call they were very slow, leading to delays in product loading.
- **Solution:** Optimized API requests using caching techniques

- **Challenges:** Authentication tokens were expiring too quickly.
- **Solutions**: Increased token expiration time and added session renewal functionality.

6. Future Plans

In future I'll implement some more functionality and more improvements in my hackathon plans

Example:

- Adding a wishlist feature.
- Implementing advanced search with multiple filters.
- Enhancing the UI/UX for mobile users.
- Introducing a review system for products

- Implementing Shipment and tracking system
- Display user order details
- I’ll create admin panel

7. Repository Link

Here is my GitHub Repository Link where you can visit my plan about marketplace , you can also check my diagram which is related to my Design System Architecture.

GitHub Repository:

https://github.com/Areesha-sites/HACKATHONS.git