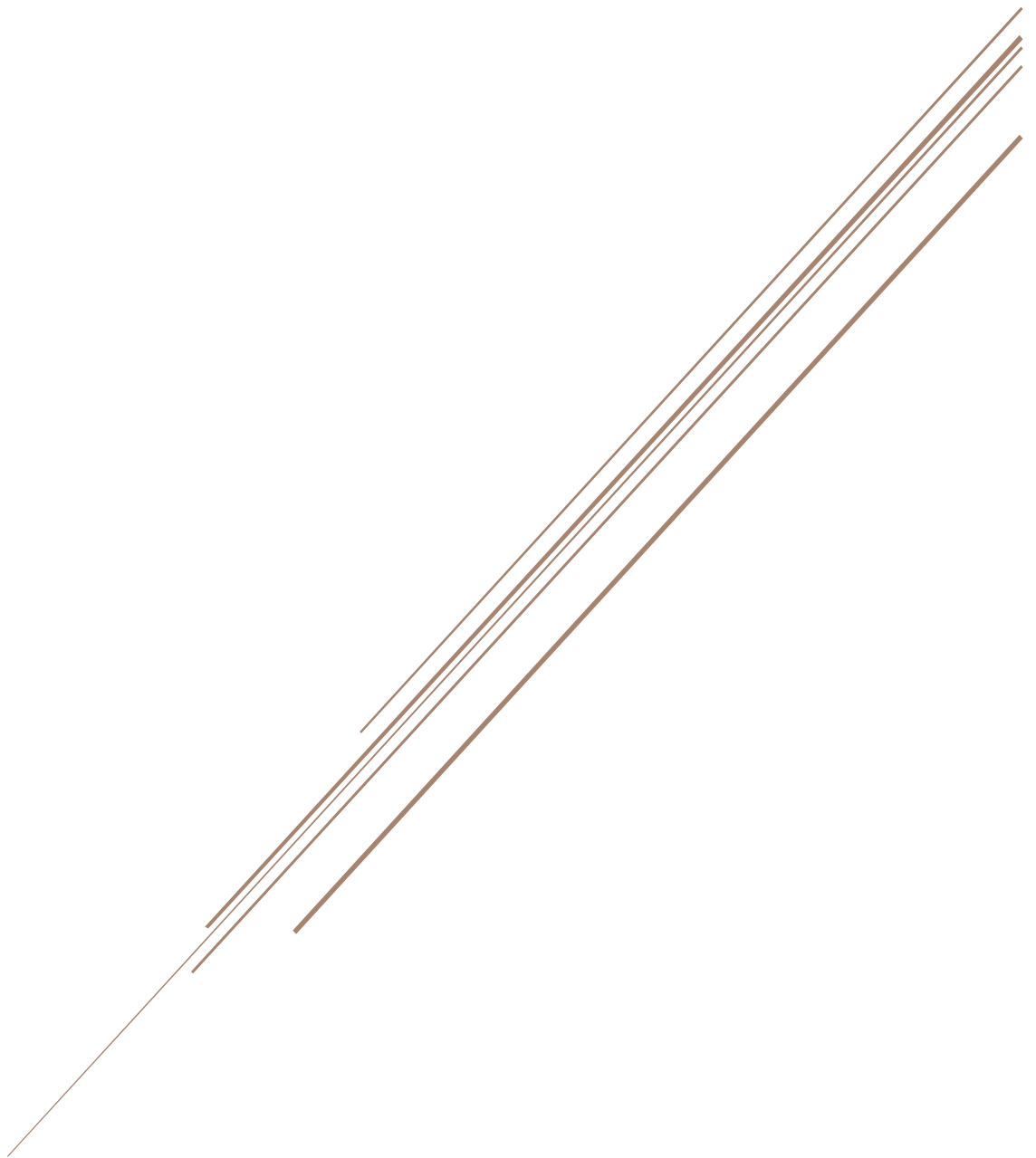
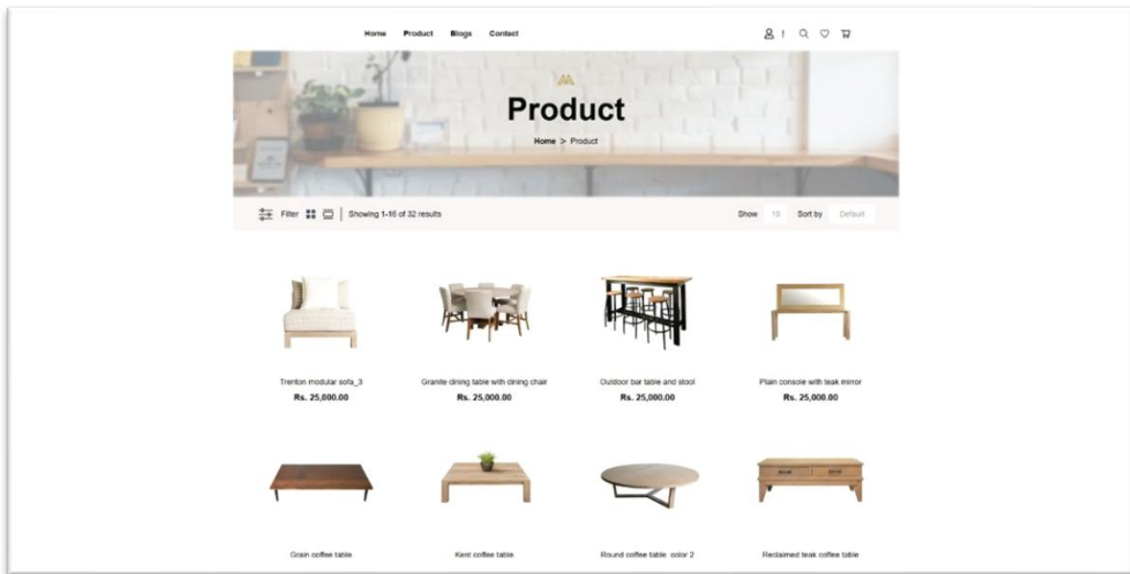


HACKATHON DAY 5





1. INTRODUCTION

A furniture marketplace website enables buyers to explore and purchase sofas and chairs, while sellers list their inventory for sale. The platform should emphasize **usability**, **scalability**, and **performance** to cater to a wide range of customers and sellers.

2. KEY FEATURES OF THE MARKETPLACE

The marketplace should include the following key features:

- **Product Listings:** Display sofas and chairs with high-quality images, descriptions, and prices.
- **Advanced Search & Filters:** Enable filtering by material, color, size, style, price, etc.
- **User Profiles:** Buyers can view order history and manage shipping information. Sellers can manage product listings, view sales data, and access dashboards.
- **Shopping Cart & Checkout:** Support adding items to a cart and secure payment gateways.
- **Admin Panel:** Manage users, products, and sales analytics.
- **Mobile Responsiveness:** Ensure the website works seamlessly across all devices and browsers.
- **Error Handling:** Provide meaningful error messages, such as 'No products available' or 'API error.'

3. STEP-BY-STEP DEVELOPMENT PROCESS

STEP 1: REQUIREMENT ANALYSIS

Define buyer and seller personas and key features. Research competitor websites for ideas. Plan a testing strategy for user workflows and backend systems.

STEP 2: DESIGN PHASE

The design phase includes:

- **Wireframing:** Create layout drafts for product listing pages, seller dashboards, and the shopping cart. Tools: Figma, Adobe XD, or Sketch.
- **UI/UX Design:** Prioritize easy navigation and accessibility. Ensure consistency in typography, colors, and element spacing.

STEP 3: FRONTEND DEVELOPMENT

Frontend development involves:

- **Framework:** Use React.js, Next.js, or Angular for modern and responsive UIs.
- **Components:** Create reusable components for product cards, navigation bars, filters, etc. Ensure modular design for scalability.
- **Cross-Browser Testing:** Verify layout and interactivity across Chrome, Firefox, Safari, and Edge.

STEP 4: BACKEND DEVELOPMENT

Backend development includes:

- **API Integration:** Create RESTful APIs for product fetching, cart management, and order placement.
- **Error Handling:** Use try-catch blocks for API calls and provide user-friendly fallback messages

```

"scripts": {
  "dev": "next dev --turbo",
  "build": "next build",
  "start": "next start",
  "lint": "next lint"
},
  "import-data": "node scripts/import-data.mjs"
},
"dependencies": {
  "@sanity/client": "^6.25.0",
  "@sanity/image-url": "^1.1.0",
  "@sanity/vision": "^3.70.0",
  "axios": "^1.7.9",

```

```

async function importData() {
  try {
    console.log('Fetching products from API...');
    const response = await axios.get('https://next-ecommerce-templat
    const products = response.data;
    console.log(Fetched ${products.length} products);

    for (const product of products) {
      console.log(Processing product: ${product.title});

      let imageRef = null;
      if (product.image) {
        imageRef = await uploadImageToSanity(product.image);
      }

```

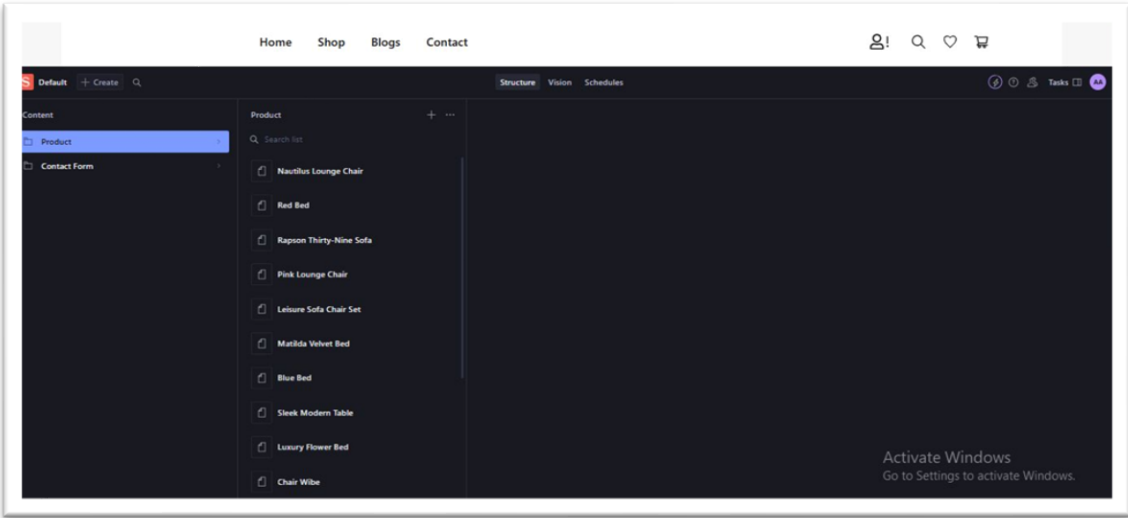
```

    console.log(Uploading product to Sanity: ${sanityProduct.name});
    const result = await client.create(sanityProduct);
    console.log(Product uploaded successfully: ${result._id});
  }

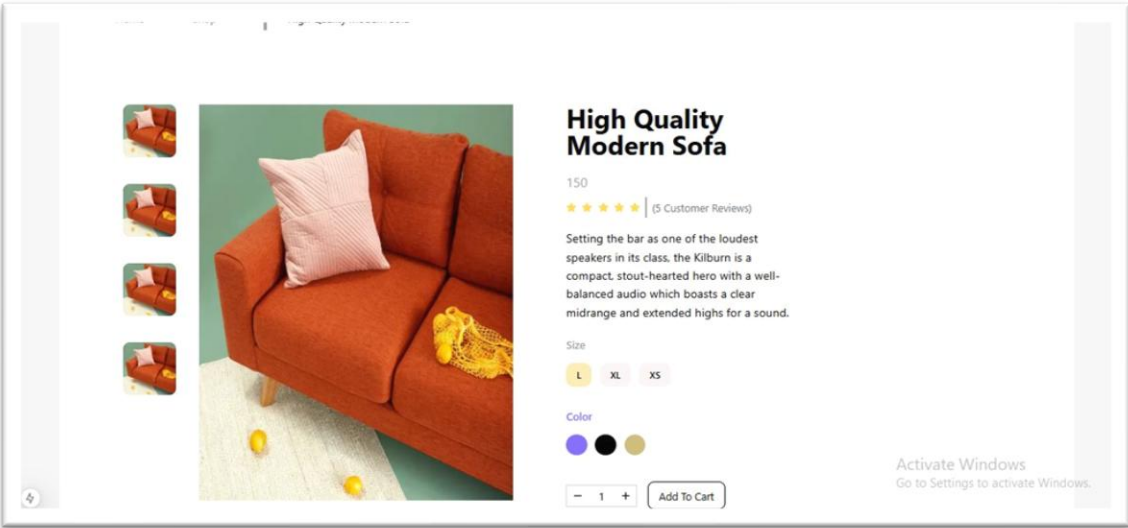
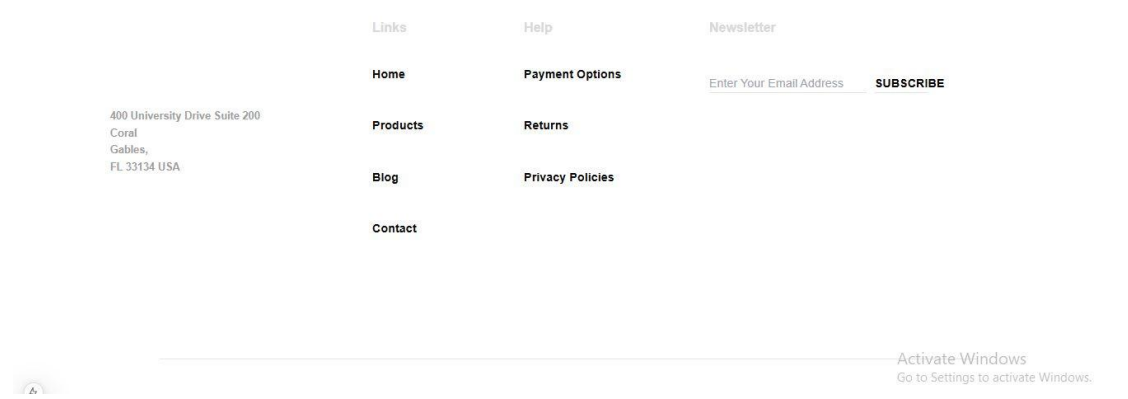
  console.log('Data import completed successfully!');
} catch (error) {
  console.error('Error importing data:', error);
}

importData();

```



Error: Failed to fetch



STEP 5: TESTING

Testing should cover:

- **Functional Testing:** Validate product listings, cart operations, and search features using tools like Cypress, React Testing Library, or Jest.
- **Performance Testing:** Optimize images, minify JavaScript, and implement caching.
- **Cross-Browser and Device Testing:** Use BrowserStack or LambdaTest to test on different screen sizes (mobile, tablet, desktop).
- **Security Testing:** Prevent SQL injection, XSS, and other vulnerabilities. Use HTTPS and store API keys securely.

STEP 6: PERFORMANCE OPTIMIZATION

Optimize assets like images using TinyPNG, enable lazy loading, and minify CSS, JavaScript, and HTML.

STEP 7: DEPLOYMENT

Deploy the website using Vercel, Netlify, or AWS. Automate builds and deployments using CI/CD pipelines with tools like GitHub Actions or Jenkins.

STEP 8: USER ACCEPTANCE TESTING (UAT)

Simulate buyer and seller activities (e.g., placing orders, managing inventory). Collect feedback from users to improve workflows.

STEP 9: DOCUMENTATION

Maintain a Testing Report in CSV format including Test Case ID, Description, Steps, Expected vs. Actual Results, and Status. Create a README file summarizing the project setup and deployment steps.

Test Case ID	Description	Steps	Expected Result	Actual Result	Status
TC001	Verify homepage loads correctly	1. Open homepage	Homepage should load with all elements visible	Homepage loaded with all elements visible	Pass
TC002	Verify navigation to 'Chairs' category	1. Click 'Chairs' category	Chairs category page should display chair products	Chairs page displayed correctly	Pass

TC003	Verify navigation to 'Sofas' category	1. Click 'Sofas' category	Sofas category page should display sofa products	Sofas page displayed correctly	Pass
TC004	Verify search functionality	1. Enter 'recliner' in search bar	Search results should display recliner chairs	Recliner chairs displayed in search results	Pass
TC005	Verify add-to-cart functionality	1. Select a product and click 'Add to Cart'	Selected product should be added to the cart	Product successfully added to the cart	Pass
TC006	Verify cart functionality	1. View cart	Cart should display the added product with correct details	Cart displays the product correctly	Pass
TC007	Verify checkout process	1. Proceed to checkout	Checkout page should display shipping and payment options	Checkout page displays options correctly	Pass
TC008	Verify order confirmation	1. Complete checkout	Order confirmation page should display order details	Order confirmation page displays details correctly	Pending
TC009	Verify product filtering	1. Apply filters (e.g., price, material)	Filtered results should display products matching the criteria	Filtered results display correctly	Pass
TC010	Verify user login/registration	1. Create or log in to an account	User should be able to create or log in to an account	User account created/logged in successfully	Pending

Test Case ID	Description	Steps	Expected Result	Actual Result	Status
TC001	Verify homepage loads correctly	1. Open homepage	Homepage should load with all elements visible	Homepage loaded with all elements visible	Pass
TC002	Verify navigation to 'Chairs' category	1. Click 'Chairs' category	Chairs category page should display chair products	Chairs page displayed correctly	Pass
TC003	Verify navigation to 'Sofas' category	1. Click 'Sofas' category	Sofas category page should display sofa products	Sofas page displayed correctly	Pass
TC004	Verify search functionality	1. Enter 'recliner' in search bar	Search results should display recliner chairs	Recliner chairs displayed in search results	Pass
TC005	Verify add-to-cart functionality	1. Select a product and click 'Add to Cart'	Selected product should be added to the cart	Product successfully added to the cart	Pass
TC006	Verify cart functionality	1. View cart	Cart should display the added product with correct details	Cart displays the product correctly	Pass
TC007	Verify checkout process	1. Proceed to checkout	Checkout page should display shipping and payment options	Checkout page displays options correctly	Pass
TC008	Verify order confirmation	1. Complete checkout	Order confirmation page should display order details	Order confirmation page displays details correctly	Pass

4. EXPECTED OUTCOME

A fully functional furniture marketplace with responsive design. Error handling and fallback mechanisms for a robust user experience. Comprehensive testing reports and performance optimizations.

5. SUBMISSION CHECKLIST

Ensure you have the following **functional deliverables**:

1. **Functional Deliverables:** Working frontend with responsive design, API integrations, and backend logic.
2. **Testing Artifacts:** Testing reports (functional, performance, cross-browser) and before-and-after performance snapshots.
3. **Documentation:** Testing and optimization steps, along with a README file in the repository.