Day 6 - Deployment Preparation and Staging Environment

[Dynamic Furniture E-commerce Website]

Project Overview:

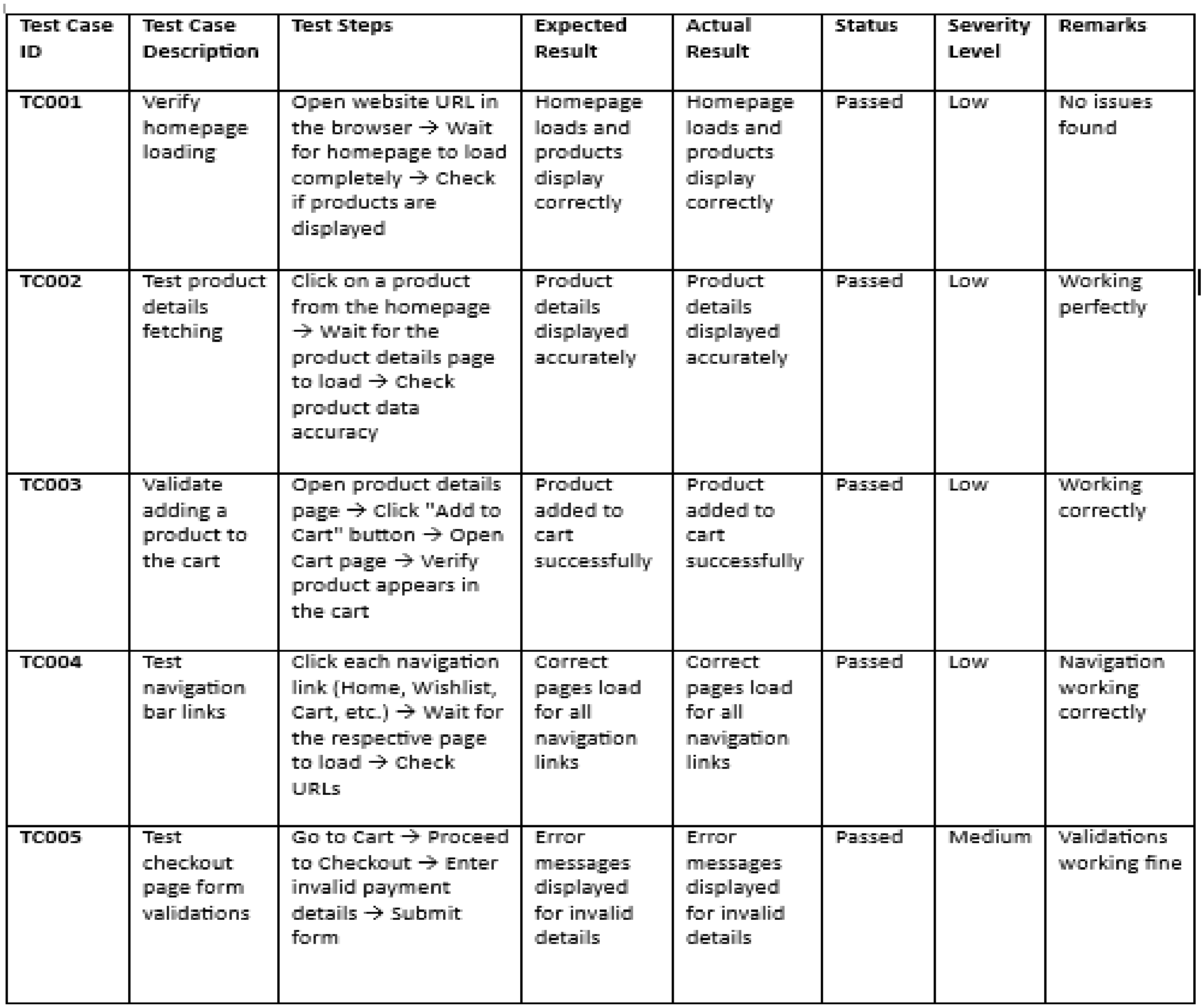
- **Project Name:** Dynamic Furniture Marketplace
- **Core Features:**
  - Product Listing and Detail Pages
  - Add to Cart and Wishlist Functionality
  - Search and Filter Options
  - Checkout flow
  - User profile component
  - Reviews and ratings component
  - Notification component
  - Fully Dynamic and Responsive Website
- **Goal:** To create a seamless user experience with optimized performance and scalability.

Deployment Preparation:

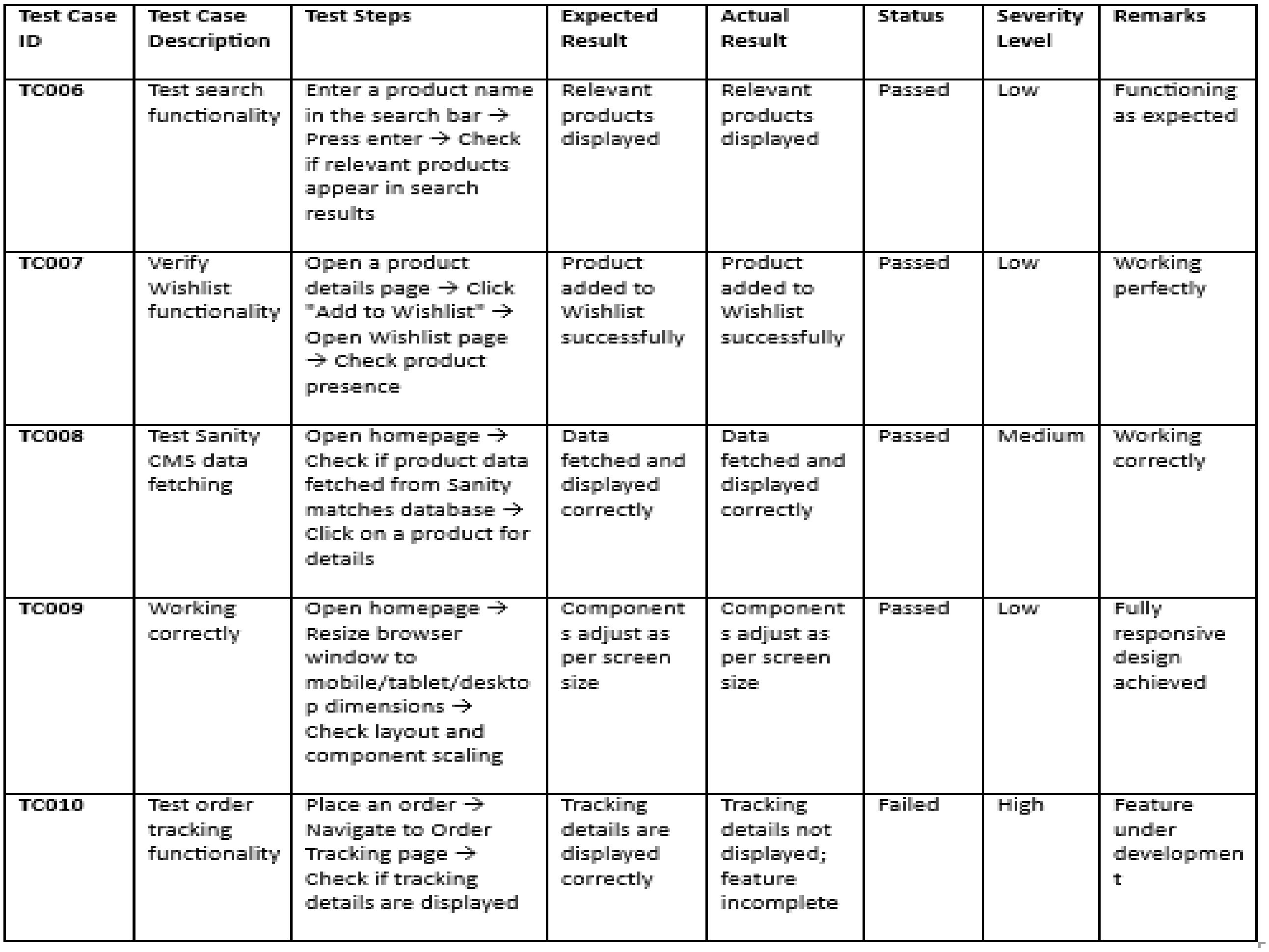
Key Steps Taken:

1. **Code Review:**
   - Conducted a detailed review of the codebase to ensure clean and efficient code.
   - Removed redundant files and unused dependencies.
2. **Environment Variables:**
   - Secured sensitive data (e.g., API keys) using .env files.
   - Ensured correct variables for both staging and production.
3. **Build Optimization:**
   - Ran npm run build to check for any compilation issues.
   - Minimized large libraries to reduce build size.

4. **Testing:**
   - Conducted Unit, Integration, and End-to-End testing for stability.
   - Verified responsive design across multiple screen sizes.
5. **CSV Report:**
   - Generated a CSV report to track errors, performance metrics, and fixes applied.



Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
TC001	Verify homepage loading	Open website URL in the browser → Wait for homepage to load completely → Check if products are displayed	Homepage loads and products display correctly	Homepage loads and products display correctly	Passed	Low	No issues found
TC002	Test product details fetching	Click on a product from the homepage → Wait for the product details page to load → Check product data accuracy	Product details displayed accurately	Product details displayed accurately	Passed	Low	Working perfectly
TC003	Validate adding a product to the cart	Open product details page → Click "Add to Cart" button → Open Cart page → Verify product appears in the cart	Product added to cart successfully	Product added to cart successfully	Passed	Low	Working correctly
TC004	Test navigation bar links	Click each navigation link (Home, Wishlist, Cart, etc.) → Wait for the respective page to load → Check URLs	Correct pages load for all navigation links	Correct pages load for all navigation links	Passed	Low	Navigation working correctly
TC005	Test checkout page form validations	Go to Cart → Proceed to Checkout → Enter invalid payment details → Submit form	Error messages displayed for invalid details	Error messages displayed for invalid details	Passed	Medium	Validations working fine



Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Remarks
TC006	Test search functionality	Enter a product name in the search bar → Press enter → Check if relevant products appear in search results	Relevant products displayed	Relevant products displayed	Passed	Low	Functioning as expected
TC007	Verify Wishlist functionality	Open a product details page → Click "Add to Wishlist" → Open Wishlist page → Check product presence	Product added to Wishlist successfully	Product added to Wishlist successfully	Passed	Low	Working perfectly
TC008	Test Sanity CMS data fetching	Open homepage → Check if product data fetched from Sanity matches database → Click on a product for details	Data fetched and displayed correctly	Data fetched and displayed correctly	Passed	Medium	Working correctly
TC009	Working correctly	Open homepage → Resize browser window to mobile/tablet/desktop dimensions → Check layout and component scaling	Components adjust as per screen size	Components adjust as per screen size	Passed	Low	Fully responsive design achieved
TC010	Test order tracking functionality	Place an order → Navigate to Order Tracking page → Check if tracking details are displayed	Tracking details are displayed correctly	Tracking details not displayed; feature incomplete	Failed	High	Feature under development

Staging Environment Configuration:

Hosting on Vercel:

- **Deployment Platform:** Vercel was chosen for its simplicity, performance, and automatic build capabilities.

Configuration Details:

- Set up a **staging branch** with its own environment variables.
- Enabled Continuous Deployment (CD) pipelines to automate deployments.
- Used **Vercel analytics** for real-time monitoring.

Error Handling and Challenges During Deployment:

1. Environment Variable Issues:

- **Problem:** Variables were not loading correctly in staging.
- **Cause:** Misconfigurations in the .env file and missing keys.
- **Solution:**
  - Double-checked the. env setup.
  - Used Vercel's dashboard to securely store and manage environment variables.

2. Runtime Errors:

- **Problem:** Uncaught errors appeared in the console during staging.
- **Cause:** Edge cases in cart and Wishlist functionality.
- **Solution:**
  - Integrated **Sentry** to track and debug runtime errors.
  - Improved error handling using try-catch blocks and default values.

3. Responsiveness Issues:

- **Problem:** Certain layouts were breaking on mobile devices.
- **Solution:**
  - Tested using Chrome DevTools and physical devices.
  - Refined Tailwind CSS classes and adjusted breakpoints.

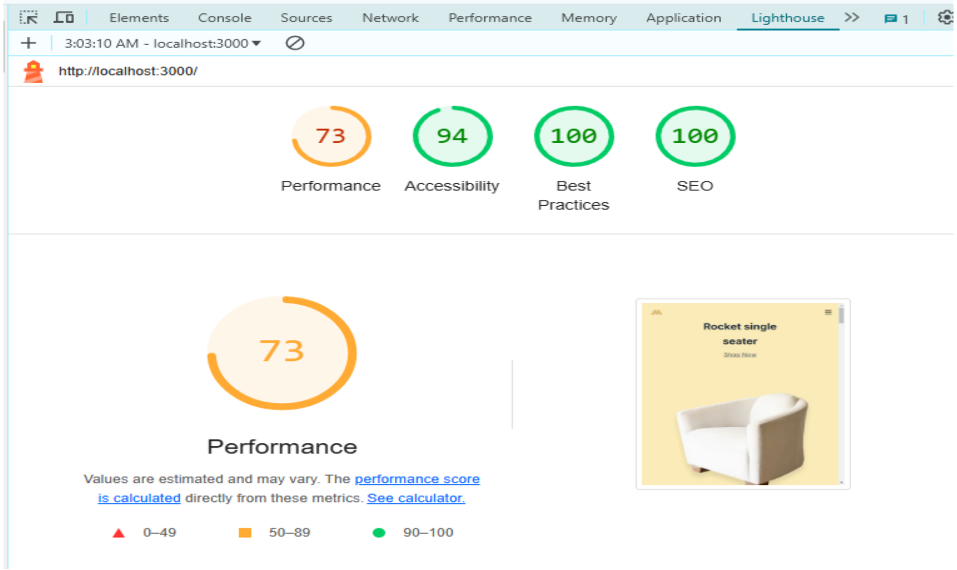
Testing Process:

1. Functional Testing:

- Ensured that all core features functioned as expected:
  - Adding and removing items from the cart.
  - Saving and removing items from the Wishlist.
  - Searching and filtering products.
- Verified navigation between pages worked without errors.

2. Performance Testing:

- Used **Lighthouse** to test performance, accessibility, SEO, and best practices:
  - **Performance Score:** Optimized to reduce initial load time.
  - **Accessibility Score:** Fixed contrast and semantic issues.
  - **SEO Score:** Ensured meta tags and alt attributes were in place.



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

▲ 0–49 ■ 50–89 ● 90–100

- Used **Vercel's built-in analytics** to monitor speed and performance metrics.

3. Responsive Design Testing:

- Tested the website on multiple screen sizes:
  - Mobile, tablet, and desktop resolutions.
  - Verified no layout shifts or overflow issues.

Deployment Workflow:

Pre-Deployment Steps:

1. Ensured the database was backed up.
2. Validated all environment variables.
3. Ran a final build (npm run build) to ensure no errors.

Deployment on Vercel:

1. Pushed the staging branch to GitHub.

2. Configured **Vercel** for automatic deployment from the production branch.
3. Monitored deployment logs to confirm success.

Post-Deployment Testing:

- Verified all functionalities, including cart, Wishlist, and filters, on the live site.
- Conducted manual and automated tests for critical workflows.

Challenges and How They Were Resolved:

1. Environment Variable Management:

- **Issue:** Missing. env configurations led to build failures.
- **Solution:** Used Vercel's Environment Variable Manager to handle sensitive data.

2. Layout Issues on Small Screens:

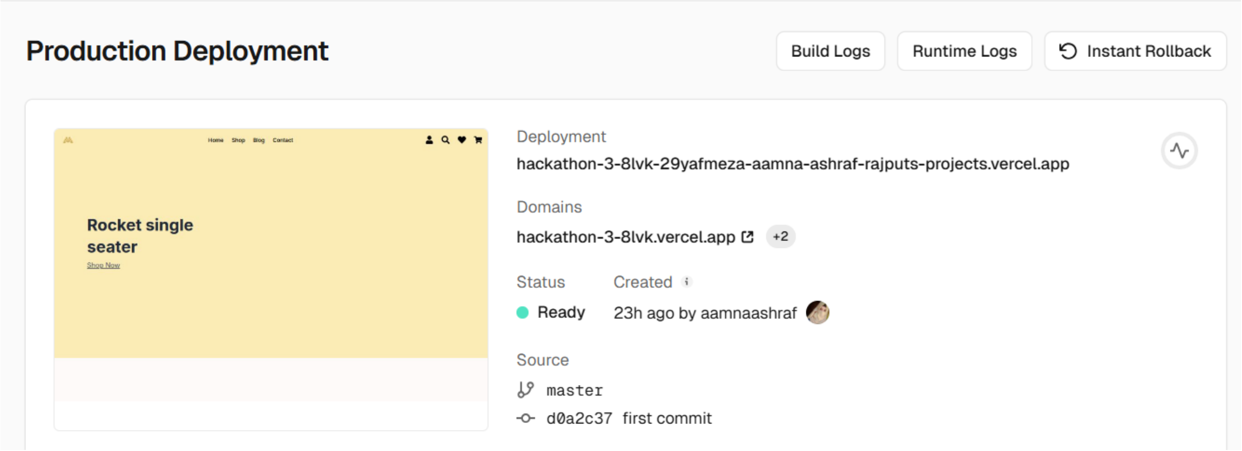
- **Issue:** Elements were misaligned or overlapping on mobile.
- **Solution:** Tested multiple devices and adjusted Tailwind CSS classes for consistent layouts.

3. Deployment Errors:

- **Issue:** Initial deployment on Vercel failed due to missing build dependencies.
- **Solution:** Added required dependencies in package. Json and reran the build.

Final Outcome:

- Successfully deployed a **scalable, dynamic furniture e-commerce website** on Vercel.
- Achieved high performance and accessibility scores through Lighthouse testing.
- Resolved all critical errors and ensured smooth functionality across devices.



Production Deployment

Deployment
hackathon-3-8lvk-29yafmeza-aamna-ashraf-rajputs-projects.vercel.app

Domains
hackathon-3-8lvk.vercel.app +2

Status: Ready

Created: 23h ago by aamnaashraf

Source
master
d0a2c37 first commit

Next Steps:

1. Monitor the live environment using **Vercel Analytics** and **Sentry** for continuous improvements.
2. Plan future enhancements like user reviews and advanced sorting.
3. Regularly update and test to maintain security and scalability.

Conclusion:

The development of this dynamic furniture e-commerce platform demonstrates a user-focused and scalable solution. Featuring product listings, detailed pages, search, filters, and Wishlist options, it delivers a seamless shopping experience. Deployment on Vercel involved resolving challenges like runtime errors and image optimization, ensuring a stable application. Rigorous testing with tools like Lighthouse and responsive checks on multiple devices enhanced the website's performance and usability. This project highlights a balance of technical expertise and commitment to creating a reliable, high-quality e-commerce platform.