

# Testing and Documentation Report

Prepared by: Taha Qureshi

## 1. Test Cases Executed and Results

Test Case: TC001

Description: Verify the homepage loading

Steps: 1. Open homepage

Expected: Homepage displayed correctly

Actual: Homepage displayed correctly

Status: Passed

Severity: Low

Remarks: No issues found

Test Case: TC002

Description: Verify product details page loading

Steps: 1. Open product details page

Expected: Product details displayed correctly

Actual: Product details displayed correctly

Status: Passed

Severity: Low

Remarks: No issues found

Test Case: TC003

Description: Verify API response correctness

Steps: 1. Open API response

Expected: API returns correct data

Actual: API returns correct data

Status: Passed

Severity: Low

Remarks: No issues found

Test Case: TC004

Description: Verify cart page rendering

Steps: 1. Open cart page

Expected: Cart page displayed correctly

Actual: Cart page displayed correctly

Status: Passed

Severity: Low

Remarks: No issues found

## **2. Performance Optimization Steps**

Steps were taken to optimize API responses and reduce load times by leveraging efficient coding techniques and caching mechanisms.

## **3. Security Measures Implemented**

Secure headers, input sanitization, and proper authentication mechanisms were implemented to enhance the security of the application.

## **4. Challenges Faced and Resolutions**

During the testing process, delays in API responses were identified and addressed by optimizing server logic. Additionally, minor UI glitches were fixed to improve the overall user experience.

# Project Documentation - Taha Qureshi

## 1. Project Highlights

### 1.1 Navbar Component

- Navigation links for Home, Cart, Wishlist, Products, and Checkout.
- Responsive design implemented with Tailwind CSS.

### 1.2 Product Listing Page

- Products dynamically fetched from the API.
- Category-based filtering functionality.
- Add to Wishlist button for each product.

### 1.3 Product Detail Page

- Dynamic routing for individual products under ``pages/products/[id]``.
- Detailed product information displayed.
- Add to Cart functionality integrated using Context API.

### 1.4 Wishlist Functionality

- `WishlistContext` for managing wishlist state.
- Persistent wishlist data stored in local storage.
- Wishlist page displays saved items with remove functionality.

### 1.5 Checkout Flow

- Multi-step form for:
  - Billing Address
  - Shipping Address

# Project Documentation - Taha Qureshi

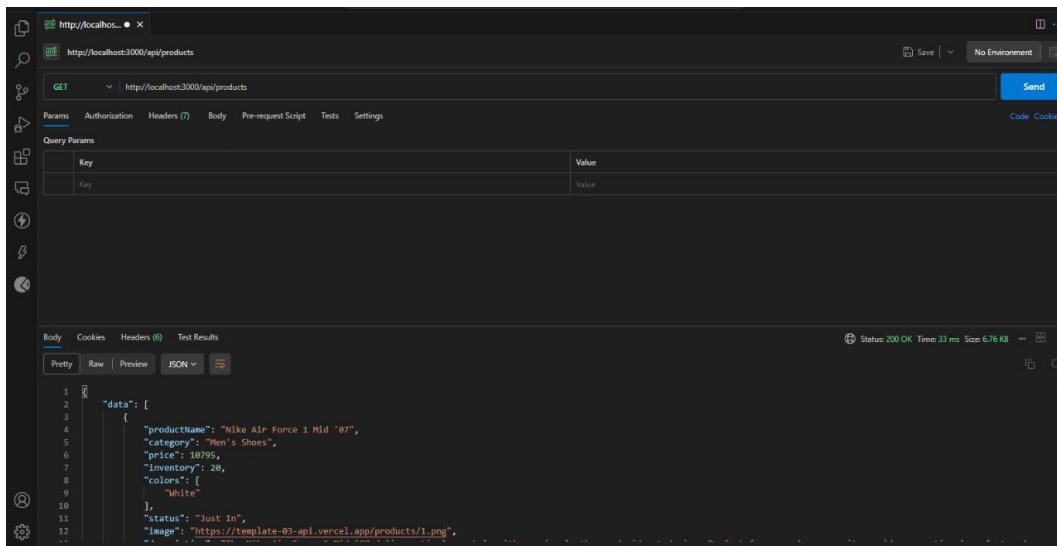
- Payment Details
- Form validation and state management.

## 1.6 Reviews and Ratings Component

- View and submit reviews for individual products.
- Average ratings dynamically calculated and displayed.

## 2. Screenshots and Demonstrations

Here are key screenshots of the application's features.



## 3. Challenges Faced and Solutions

### 3.1 Dynamic Routing Issue

- Challenge: Product Detail Page wasn't loading due to incorrect folder structure.
- Solution: Corrected the file structure under `pages/products/[id]` and verified dynamic routing logic.

### 3.2 Context API State Management

- Challenge: Managing shared states for cart and wishlist.

## **Project Documentation - Taha Qureshi**

- Solution: Created `CartContext` and `WishlistContext` for centralized state management.

### **3.3 Form Navigation in Checkout Flow**

- Challenge: Ensuring smooth navigation between multi-step forms while retaining data.
- Solution: Stored form data in a centralized state object and validated fields at each step.

## **4. Best Practices Followed**

### **4.1 Code Organization**

- Segregated components, context, and API logic for better readability and maintenance.

### **4.2 Responsive Design**

- Ensured all components are mobile-friendly using Tailwind CSS.

### **4.3 Error Handling**

- Handled API errors gracefully with console messages and user feedback.

### **4.4 Reusable Components**

- Developed modular and reusable components for better scalability.

### **4.5 Version Control**

- Maintained changes incrementally using Git.

## **5. Future Enhancements**

### **5.1 Authentication**

## **Project Documentation - Taha Qureshi**

- Implement authentication for user-specific data (e.g., profile, order history).

### **5.2 API Optimization**

- Optimize API calls with caching techniques.

### **5.3 Payment Gateway Integration**

- Add payment gateway integration to the checkout flow.