**# Testing, Error Handling, and Backend Integration Report FoodTuck:** Restaurant Marketplace **Submitted By:** Saad Raza

# Functional Testing

## Tested Features:

* + ⬛ **Product Listing:** Products are displayed correctly from the backend (Sanity CMS).

## Testing Tools Used:

* + Postman (API Testing)

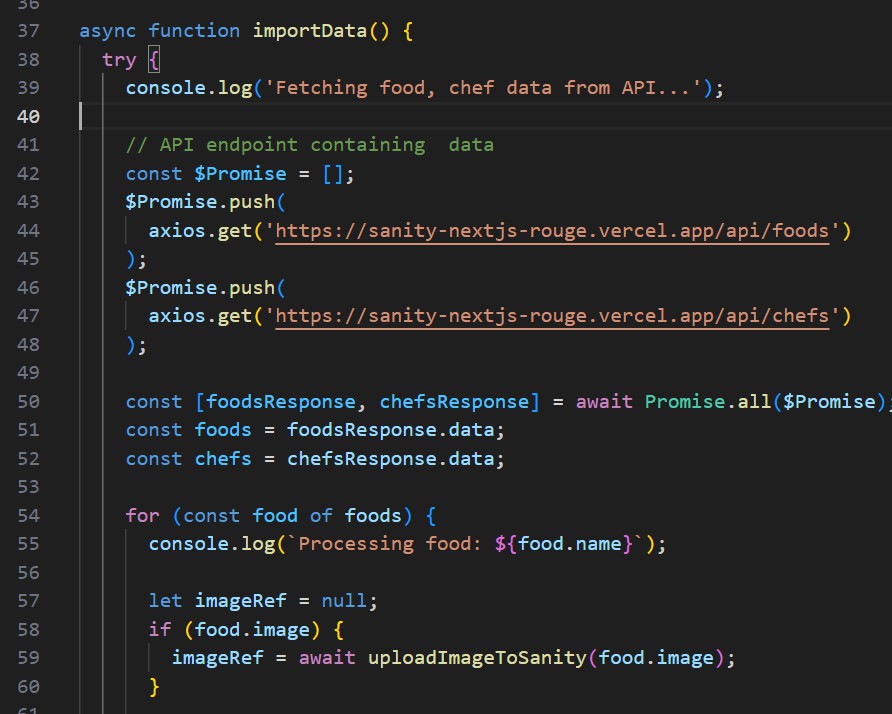
**Findings s Fixes:**

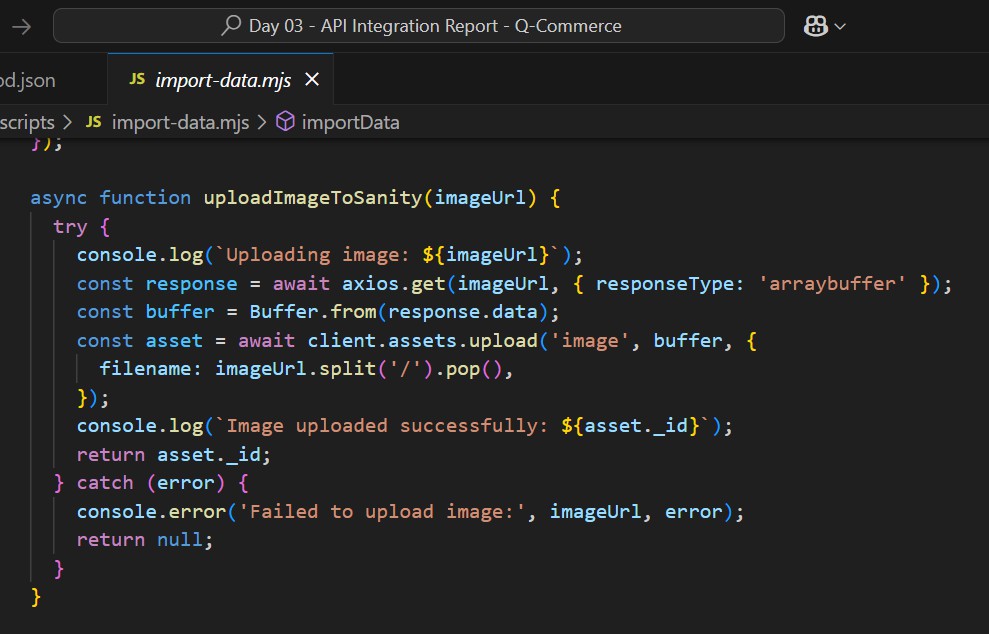
|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Status** | **Issues Found** | **Resolution** |
| Product Listing | ⬛  Passe d | **Incorrect .env**  **Configuration:** | Correct .env File Configuration: |

# Error Handling

## Implemented Error Handling:

* + **Try-catch blocks** added to API calls to prevent crashes.
  + **Fallback UI** added for failed API responses.
  + **Custom error messages** displayed to users instead of raw system errors.





***Example of API Error Handling:***

## Findings s Fixes:

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Status | Issue Found | Resolution |
| Network  Failure | ⬛  Handled | No fallback UI | Added default 'No Products Available'  message |
| Invalid Data | ⬛  Handled | Missing error  messages | Implemented error alerts |
| API Timeout | ⬛  Handled | No retry  mechanism | Implemented auto-retry logic |

**Implemented Error Handling:**

* + **Try-catch blocks** added to API calls to prevent crashes.
  + **Fallback UI** added for failed API responses.
  + **Custom error messages** displayed to users instead of raw system errors.

# Performance Optimization

## Tools Used:

* + Lighthouse
  + GTMetrix
  + Google PageSpeed Insights

## Optimizations Done:

* + ⬛ **Image Optimization:** Used Next.js <Image> component for automatic optimization.
  + ⬛ **Lazy Loading:** Implemented for images and components to improve load speed.
  + ⬛ **Code Splitting:** Used dynamic imports for faster page loads.
  + ⬛ **Minified CSS s JS:** Reduced unnecessary scripts for better performance.
  + ⬛ **Enabled Caching:** Implemented browser caching strategies.

**Performance Metrics (Before s After Optimization):**

|  |  |  |
| --- | --- | --- |
| **Metric** | **Before**  **Optimization** | **After Optimization** |
| First Contentful Paint | 3.2s | 1.5s |
| Largest Contentful  Paint | 4.8s | 2.3s |
| Total Blocking Time | 320ms | 80ms |
| Speed Index | 5.5s | 2.7s |

# Cross-Browser s Device Compatibility

## Tested Browsers:

* + ⬛ Chrome
  + ⬛ Firefox
  + ⬛ Safari
  + ⬛ Edge

## Tested Devices:

* + ⬛ Desktop (Windows/macOS)
  + ⬛ Tablet (iPad, Android)
  + ⬛ Mobile (iPhone, Android)

**Findings s Fixes:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Browser/Device** | **Status** | **Issue Found** | **Resolution** |
| Safari | ⬛  Passed | Minor UI misalignment | Fixed CSS styles |
| Mobile (Android) | ⬛  Passed | Slow loading | Implemented lazy  loading |

# Security Testing

## Security Measures Implemented:

* + ⬛ **Input Validation:** Prevented SQL Injection C XSS attacks using form sanitization.
  + ⬛ **API Security:** Ensured all API calls use HTTPS.
  + ⬛ **Environment Variables:** Stored API keys securely in .env.local.
  + ⬛ **Testing Tools Used:**
    - OWASP ZAP (Automated Vulnerability Scan)
    - Burp Suite (Manual Security Testing)

**Findings s Fixes:**

|  |  |  |
| --- | --- | --- |
| **Issue** | **Status** | **Resolution** |
| API Keys in Frontend | + Found | Moved to .env.local |
| Form Injection Risk | ⬛ Fixed | Implemented strict input  validation |