

# MARKETPLACE BUSSINESS PLAN DAY 5

## Testing, Error Handling and Backend integration

### Foodtuck

In the last day 4 I've focused on dynamic component and functional component in my hackathon after created all component I'm going to fulfill my day 5 requirement where I've to do testing, error handling and backend integration refinement. Where these key areas are include;

#### Key Learning Outcomes:

1. Perform testing , core functionalities and security testing etc...
2. Implement robust error handling mechanisms
3. Optimized marketplace for speed , responsiveness and perform metrics.
4. Cross-browser compatibility and device responsive.
5. Professional testing documentation that meets industry level.

#### Key Areas of Focus:

1. **Functional Testing:** In this I've to focused on validate marketplace and ensure that my core functionalities like product listing, product details, add to cart and user profile management.
2. **Error Handling:** Implement error messages like network failures, invalid and missing data, unexpected server errors etc...
3. **Performance Testing:** Identify bottlenecks we can use any tool like lighthouse, GTmetrix and etc..
4. **Cross-browser and Device Testing:** Test my marketplace on popular browser chrome safari etc... and also check my responsiveness on many devices like (Mobile, Tablets and desktop).
5. **Security Testing:** In this I've to validate input fields , use HTTPS for secure communication, avoid exposing my sensitive api keys etc...
6. **User Acceptance Testing (UAT):** Simulate real-world scenarios by interacting with my marketplace as a user.

#### Step for Implementation:

##### *Step 1 : Functional Testing*

Validate core functionalities of key component ensure they work as expected.

- **Product Listing:** Ensure product display properly as expected.
- **Filter and Search:** Ensure filters and search bar work correctly.

- **Cart Operation:** Ensure cart work properly like user can remove the product add the product and save in local storage .
- **Dynamic Routing :** Ensure dynamic routing work as expected when user click on the particular product they will navigate on the actual product.
- **Navigation Links:** Ensure all navigation links work properly and navigate easily.

#### Tool Used:

- **Postman:** For API response testing .
- **React Testing Library:** For component behavior testing.
- **Cypress:** For end-to-end testing

#### **Step 2 : Error Handling**

Implement mechanism for all error provide user-friendly feedback.

- Use try-catch blocks to handle API errors. 

```
try {  
  // Simulate searching for the product  const product =  
  products.find((p) => p.id === productId); if (!product) { throw  
  new Error(`Product with ID ${productId} not found.`);  
}  
  
console.log("Product found:", product);  
}  
  
catch (error) { console.error("Error:",  
error.message);  
}
```

For example:

- **Fallback UI:** Display alternative content when data is unavailable.
- **For Example:** “No item found” for an empty product list.

#### **Step 3 : Performance Optimization**

Identify and resolve performance bottlenecks using tool like google light house.

- **Optimized Assets:** use compress images tools like TinyPNG. Implement lazy loading for large assets.
- **Analyze Performance:** Use light house for speed and performance. Implemented fixes like unused CSS code and bundles of JAVASCRIPT.

- **Test Load Time** : Measuring initial loading and interaction time. For example initial page load within 2 seconds.

#### ***Step 4: Cross-browser and Device Testing***

- **Browser Testing**: Check website on different browser like chrome, safari, Firefox etc...
- **Device Tested**: Used responsive design like different screens. Manually check responsiveness on mobile devices.

#### ***Step 5 : Security Testing***

- **Input Validation**: Sanitize inputs to prevent SQL injection or XSS attacks.
- **Secure API Communication**: Ensure API calls are made over HTTPS. Store all API keys in `.env.local` environment variables.
- **Testing Tools**: OWASP for automated vulnerability scanning.
- **Burp Suit**: For advance penetration testing.

#### ***Step 6 : User Acceptance Testing (UAT)***

Simulate real-world user interactions to identify usability issues

- **Simulate Real-world Usage**: Browsing products, adding item add to cart, remove from cart, quantity increasing and decreasing and checkout process
- **Feedback Collection**:
  - Minor UI issue and now resolved
  - Improve workflows for better experience.
  - Optimized visual browsing products because of user-friendly.

#### ***Step 7 : Documentation Updates:***

Finding and resolutions into a professional report.

- Test case descriptions and results.
- Performance optimized steps.
- Security measure implemented.
- Screenshots of issue and fixes
- Additional insights into areas for future improvement.

## **CSV Content**

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	Verify user login functionality	1. Open login page 2. Enter valid credentials 3. Click login button	User successfully logs in	User successfully logs in	Passed	Low	Bameen Alam	No issues found.
TC002	Validate login with incorrect credentials	1. Open login page 2. Enter invalid credentials 3. Click login button	Error	Error				

message displayed,Error message displayed,Passed,Low,Bameen Alam,Functioning as expected. TC003,Check login button disabled for empty fields,1. Open login page 2. Leave fields empty 3. Verify button state,Login button is disabled,Login button is disabled,Passed,Low,Bameen Alam,No issues found. TC004,Verify user session timeout,1. Login to the system 2. Stay idle for 15 mins 3. Try accessing a secure page,User is redirected to login page,User is redirected to login page,Passed,Medium,Bameen Alam,Session handling working fine. TC005,Test product search functionality,1. Open search bar 2. Enter product name 3. Click search button,Relevant products displayed,No products displayed,Failed,High,Bameen Alam,Search API integration issue. TC006,Validate secure data transmission,1. Perform login 2. Monitor network traffic,Data is encrypted,Data is encrypted,Passed,High,Bameen Alam,SSL encryption working fine. TC007,Test backend integration for cart,1. Add item to cart 2. Check cart count in API response,Correct count displayed,Incorrect count displayed,Failed,High,Bameen Alam,Cart backend synchronization issue. TC008,Verify error handling for network failures,1. Simulate network failure 2. Perform an action requiring network,Error message displayed,Error message displayed,Passed,Medium,Bameen Alam,Error messages are user-friendly. TC009,Validate responsive design on mobile,1. Open site on mobile browser 2. Check UI layout,All elements align correctly,Alignment issues in footer,Failed,Medium,Bameen Alam,Footer needs responsive fixes. TC010,Check loading time optimization,1. Measure page load time using tool 2. Compare with target time,Page loads within 2 seconds,Page loads within 1.8 seconds,Passed,Low,Bameen Alam,Performance optimized.

## CSV-Based Testing Report

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	status	Severity Level	Assigned To	Remarks

TC001	Verify user login functionality	1.Open login page 2.Enter valid credentials 3.Click login button	User successfully logs in	User successfully logs in	Passed	Low	Ameen Alam	No Issues found
TC002	Validate login with incorrect credentials	1.Open login page 2.Enter invalid credentials 3.Click login button	Error message displayed	Error message displayed	Passed	Low	Ameen Alam	Functioning as expected
TC003	Check login button disabled for empty fields	1.Open login page 2.Leave fields Empty 3.Verify button state	Login button is disabled	Login button is disabled	Passed	Low	Ameen Alam	No issues found
TC004	Verify user session timeout	1.Login to the system 2.Stay idle for 15 mins 3.Try accessing a secure page	User is redirected to login page	User is redirected to login page	Passed	Medium	Ameen Alam	Session handling working fine
TC005	Test product search functionality	1.Open search bar 2.Enter product name 3.Click search button	Relevant products displayed	No products displayed	Failed	High	Ameen Alam	Search API integration issue
TC006	Validate secure data transmission	1.Perform login 2. Monitor network traffic	Data is encrypted	Data is encrypted	Passed	High	Ameen Alam	SSL encryption working fine

TC007	Test backend integration for cart	1.Add item to cart 2.Check cart count in API response	Correct count displayed	Incorrect count displayed	Failed	High	Ameen Alam	Cart backend synchronization
TC008	Verify error handling for network failure	1.Simulate network failure 2.Perform an action requiring network	Error message displayed	Error message displayed	Passed	Medium	Ameen Alam	Error messages are userfriendly
TC009	Validate responsive design on mobile	1.Open site on mobile browser 2.Check UI layout	All elements align correctly	Alignment issues in footer	Failed	Medium	Ameen Alam	Footer needs responsive fixes
TC010	Check loading time optimization	1.Measure page load time using tool. 2. Compare with target time.	Page loads within 2 seconds	Page loads within 1.8 seconds	Passed	Low	Ameen Alam	Performance Optimization

## Conclusion:

Finally I've Fulfilled all requirement of Day 5 where I've implemented testing, error handling and backend integration refinement where I've used tools and many other things. Specially focused on core functionalities like product listing , categories, search bar, error handling, SEO enhancement, improved performance, after these all implementation I'll include further concept and refine the development and deploy process.

After explaining all the key outcome I create CSV report content those content which I've implemented in my website and also create a table of CVS report where clearly see the roadmap of CSV report.