HACKATHON 03

Day 5:Enhancing Backend Skills with Superior Testing and Debugging

Key Milestones

In this phase, we aim to refine the **FoodTuck** platform to ensure it is fully polished and ready for deployment. This involves:

1. Performing comprehensive testing to validate system performance and user experience.
2. Implementing robust error-handling mechanisms to enhance reliability.
3. Optimizing performance metrics to ensure fast load times and seamless functionality.
4. Guaranteeing adaptability across various devices and browsers for a consistent user experience.
5. Developing detailed documentation to summarize progress, key achievements, and future goals.

Action Plan:

Functional Testing

Goal: Ensure every feature of **FoodTuck** operates seamlessly for a smooth user experience.

Areas Evaluated:

- **Navigation Flow:** Validate that all menu links and navigation paths function properly.
- **Product Page:** Ensure accurate product filtering, sorting, and display.
- **Cart Operations:** Test item addition, removal, quantity updates, and cart persistence.
- **User Account Functions:** Validate user registration, login, and profile management.
- **Checkout Workflow:** Simulate payments, discounts, and order completion processes.

Tools Employed:

- **Postman:** For API testing.
- **Jest and Enzyme:** For testing UI components.
- **Cypress:** To validate end-to-end workflows.
- **Lighthouse:** For performance, accessibility, and SEO audits.

Enhanced Error Handling

Objective: Develop a robust error mitigation strategy to ensure a seamless experience on **FoodTuck**, your food delivery platform.

Key Strategies:

- **API Fallbacks:** Provide alternative options for API failures to ensure uninterrupted service.
- **Descriptive Error Messages:** Display user-friendly messages like "This item is currently unavailable."
- **Centralized Error Logging:** Streamline debugging by logging all errors in a centralized system.
- **Dynamic User Alerts:** Notify users instantly about stock shortages or system errors to maintain transparency.

Example:

For unavailable food items:

- Clearly show "This dish is currently unavailable" on the menu.
- Disable the "Add to Cart" button for unavailable items.

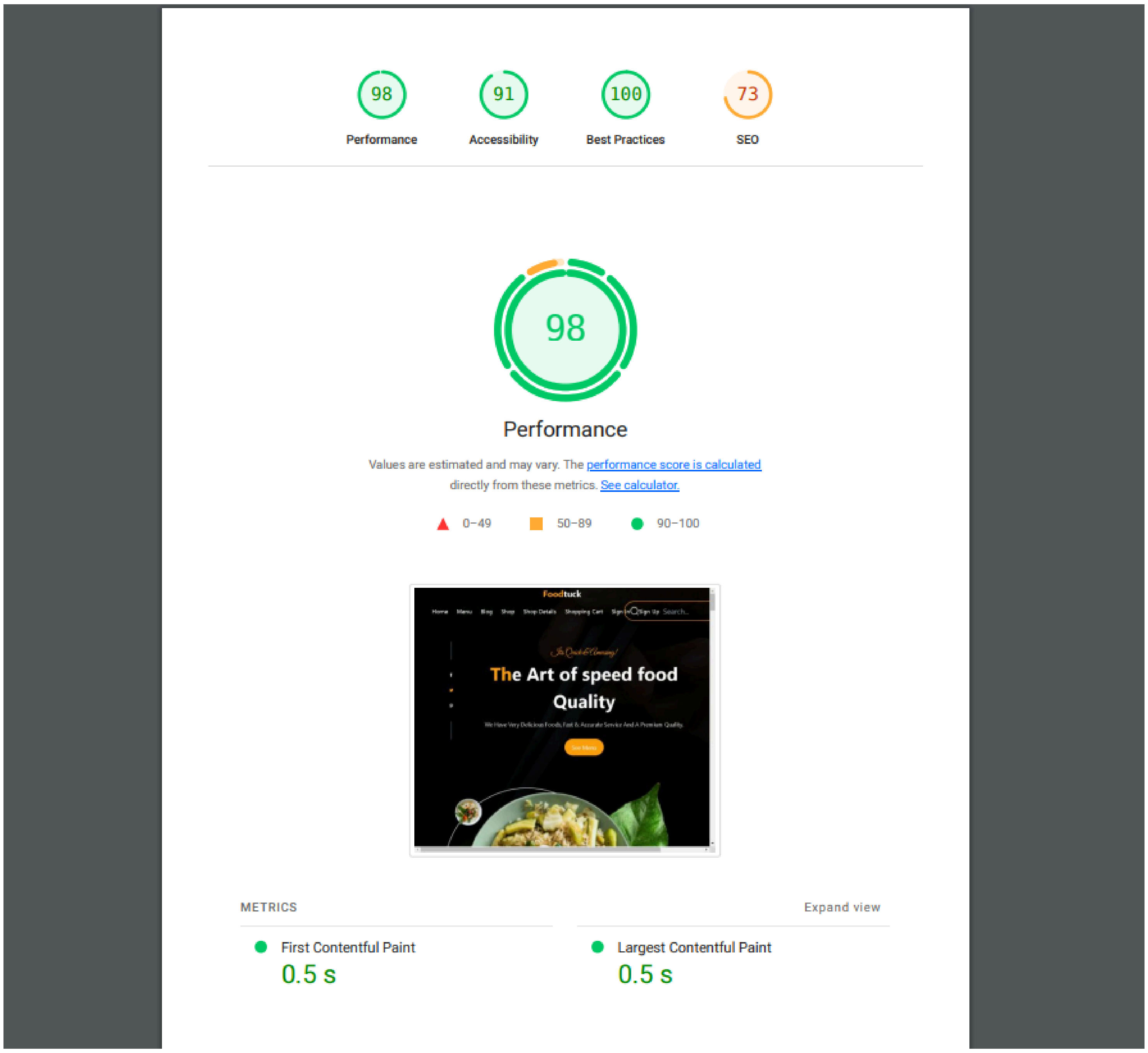
- Restrict the checkout process until issues are resolved to avoid user frustration.

Performance Tuning (Optimization)

Focus: Maximize platform responsiveness and ensure lightning-fast performance for **FoodTuck**.

Improvements Applied:

1. **Optimized API Efficiency:** Minimized response time with streamlined queries and efficient endpoints.
2. **File Compression:** Enabled Gzip and Brotli compression to reduce file sizes and accelerate page load times.
3. **Caching Strategies:** Implemented advanced caching for static assets to reduce server load and improve delivery speed.
4. **Minified Resources:** Reduced CSS, JavaScript, and HTML file sizes for enhanced performance.
5. **Content Delivery Network (CDN):** Integrated a global CDN for faster content delivery to users worldwide.



98 Performance | 91 Accessibility | 100 Best Practices | 73 SEO

98

Performance

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

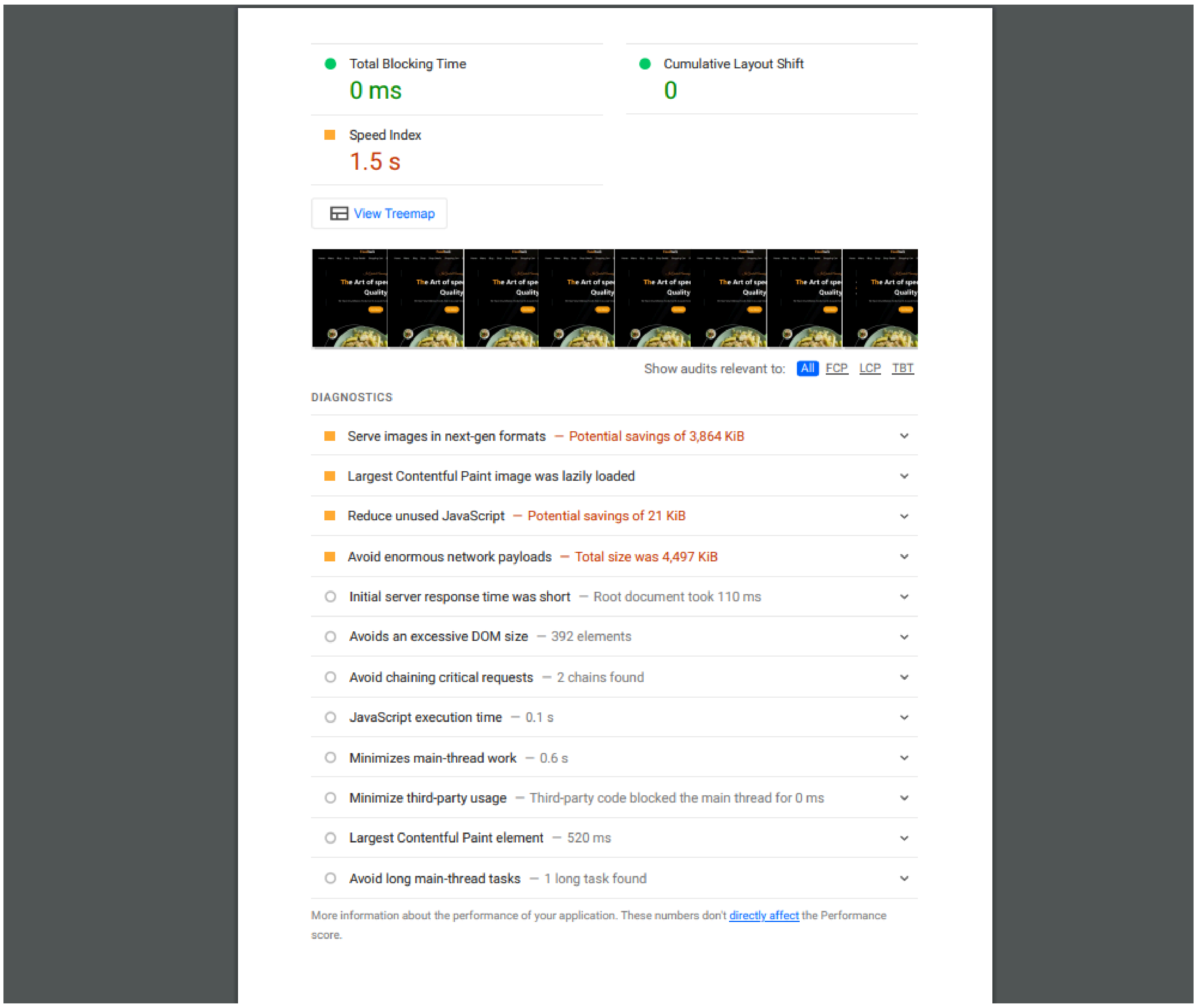
▲ 0–49 ■ 50–89 ● 90–100

Foodtuck
Home Menu Blog Shop Shop Details Shopping Cart Sign in/Sign Up Search...
Its Quick & Amazing!
The Art of speed food Quality
We Have Very Delicious Foods, Fast & Accurate Service And A Premium Quality.
See More

METRICS — Expand view

● First Contentful Paint
0.5 s

● Largest Contentful Paint
0.5 s



● Total Blocking Time
0 ms

● Cumulative Layout Shift
0

■ Speed Index
1.5 s

View Treemap

Show audits relevant to: All FCP LCP TBT

DIAGNOSTICS

■ Serve images in next-gen formats — Potential savings of 3,864 KiB

■ Largest Contentful Paint image was lazily loaded

■ Reduce unused JavaScript — Potential savings of 21 KiB

■ Avoid enormous network payloads — Total size was 4,497 KiB

○ Initial server response time was short — Root document took 110 ms

○ Avoids an excessive DOM size — 392 elements

○ Avoid chaining critical requests — 2 chains found

○ JavaScript execution time — 0.1 s

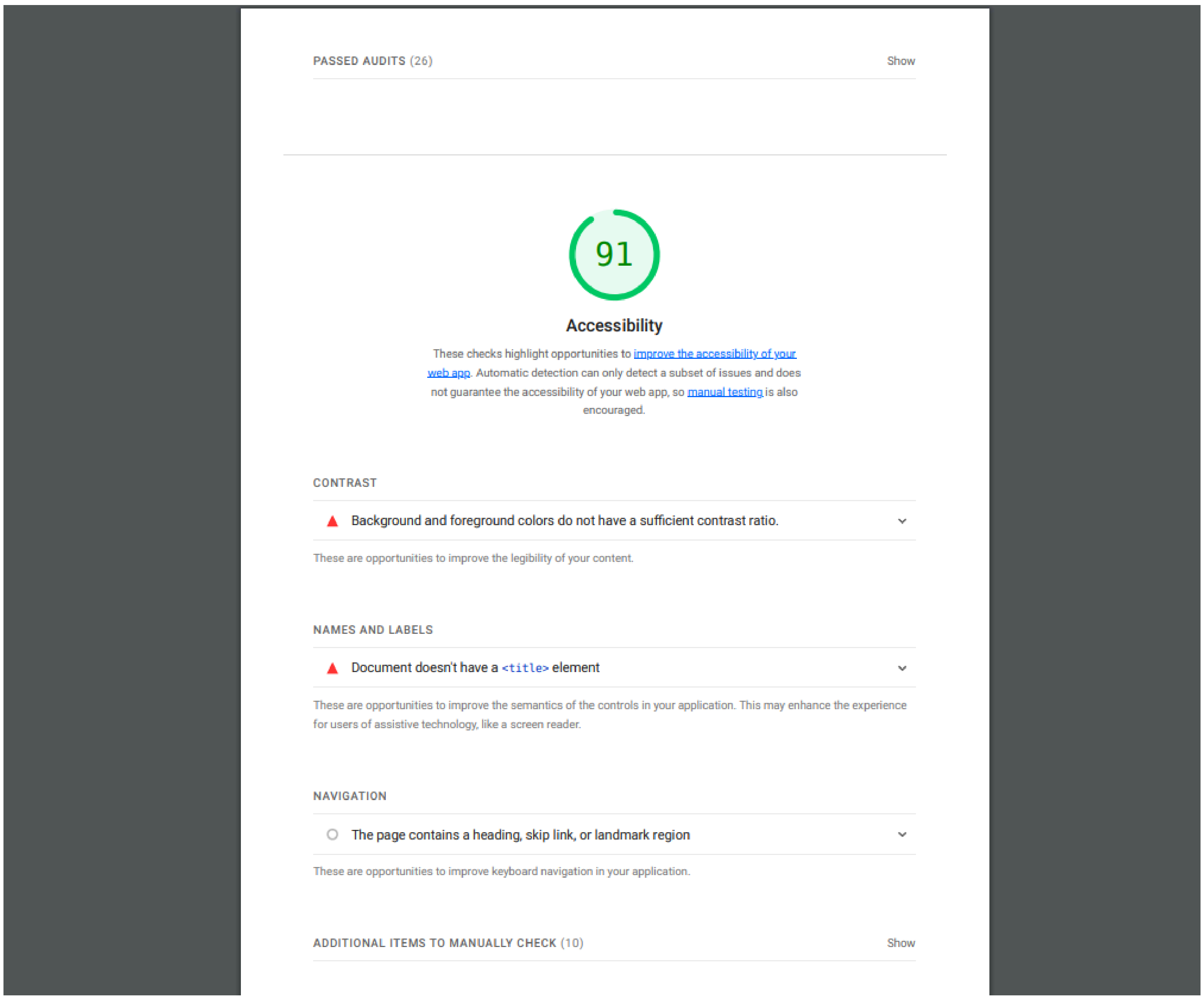
○ Minimizes main-thread work — 0.6 s

○ Minimize third-party usage — Third-party code blocked the main thread for 0 ms

○ Largest Contentful Paint element — 520 ms

○ Avoid long main-thread tasks — 1 long task found

More information about the performance of your application. These numbers don't directly affect the Performance score.



PASSED AUDITS (26) Show

91

Accessibility

These checks highlight opportunities to improve the accessibility of your web app. Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so manual testing is also encouraged.

CONTRAST

▲ Background and foreground colors do not have a sufficient contrast ratio.

These are opportunities to improve the legibility of your content.

NAMES AND LABELS

▲ Document doesn't have a `<title>` element

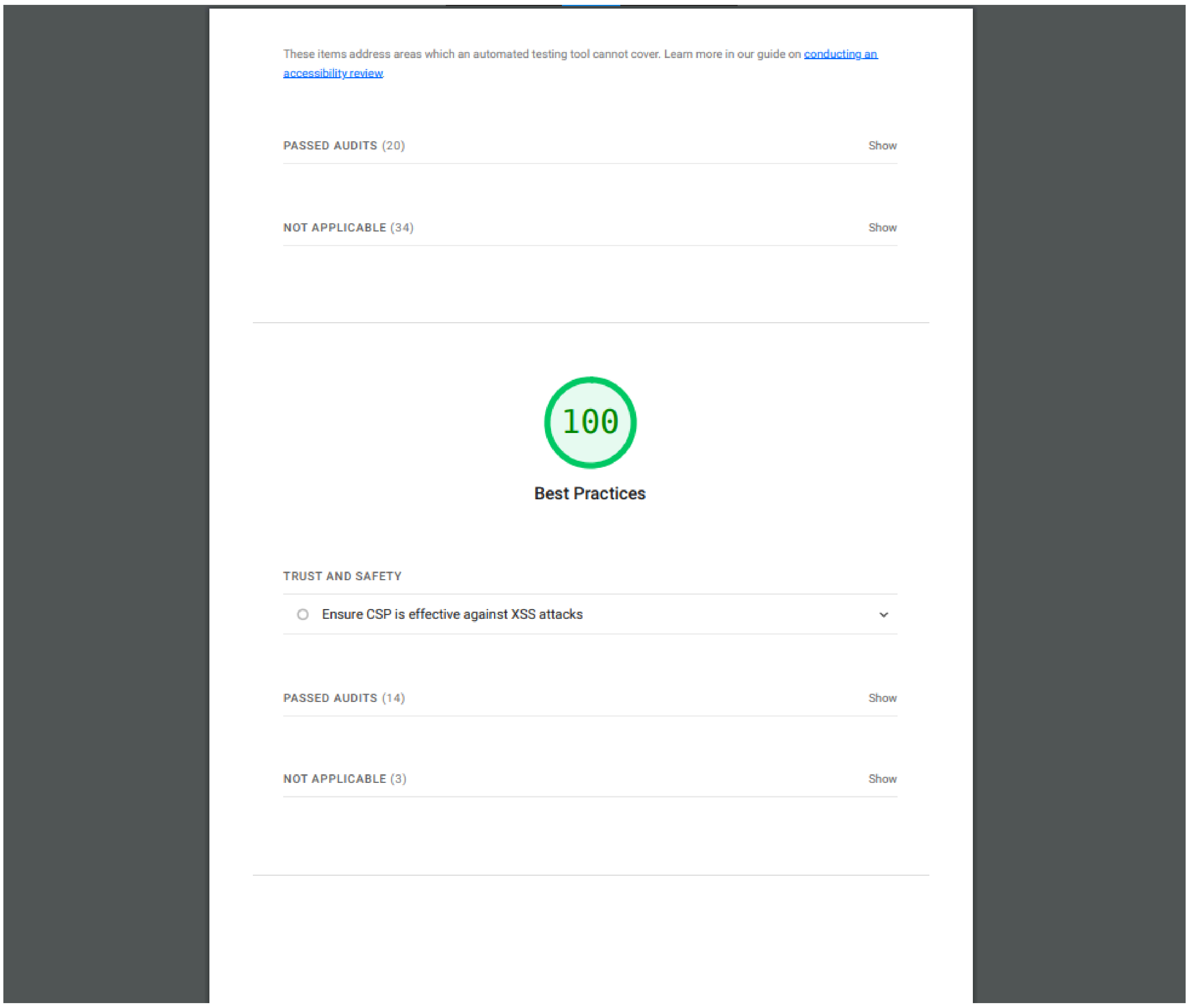
These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

NAVIGATION

○ The page contains a heading, skip link, or landmark region

These are opportunities to improve keyboard navigation in your application.

ADDITIONAL ITEMS TO MANUALLY CHECK (10) Show



These items address areas which an automated testing tool cannot cover. Learn more in our guide on conducting an accessibility review.

PASSED AUDITS (20) Show

NOT APPLICABLE (34) Show

100

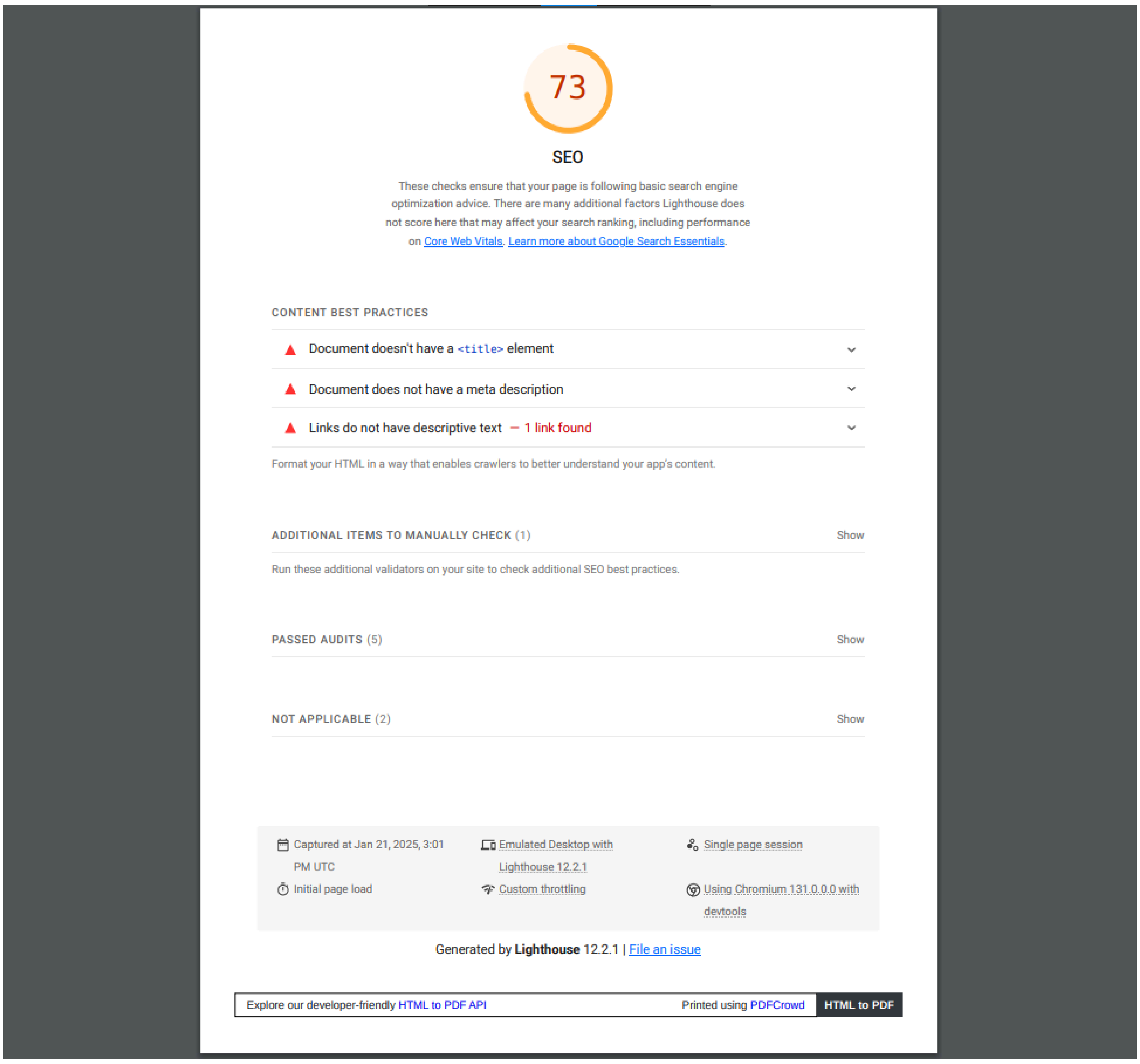
Best Practices

TRUST AND SAFETY

○ Ensure CSP is effective against XSS attacks

PASSED AUDITS (14) Show

NOT APPLICABLE (3) Show



73

SEO

These checks ensure that your page is following basic search engine optimization advice. There are many additional factors Lighthouse does not score here that may affect your search ranking, including performance on Core Web Vitals. Learn more about Google Search Essentials.

CONTENT BEST PRACTICES

▲ Document doesn't have a `<title>` element

▲ Document does not have a meta description

▲ Links do not have descriptive text — 1 link found

Format your HTML in a way that enables crawlers to better understand your app's content.

ADDITIONAL ITEMS TO MANUALLY CHECK (1) Show

Run these additional validators on your site to check additional SEO best practices.

PASSED AUDITS (5) Show

NOT APPLICABLE (2) Show

Captured at Jan 21, 2025, 3:01 PM UTC	Emulated Desktop with Lighthouse 12.2.1	Single page session
Initial page load	Custom throttling	Using Chromium 131.0.0.0 with devtools

Generated by **Lighthouse** 12.2.1 | File an issue

Explore our developer-friendly HTML to PDF API Printed using PDFCrowd HTML to PDF

Device Compatibility

Goal: Achieve seamless cross-platform functionality and user experience on **FoodTuck**.

Platforms Tested:

- **Browsers:** Chrome, Firefox, Safari, Edge, and Opera.
- **Devices:** Smartphones, tablets, desktops, and smart TVs (tested using BrowserStack).

Focus Areas:

1. **Responsive Design:** Ensure optimal display and functionality across all screen sizes and resolutions.
2. **Accessibility Support:** Enable compatibility with tools like screen readers and keyboard navigation for inclusivity.
3. **Touch and Gesture Responsiveness:** Fine-tune interactions for mobile and tablet users.
4. **Cross-Browser Consistency:** Verify that features and visuals render flawlessly across all major browsers.
5. **Performance Monitoring:** Test load times and interactivity on both high-end and low-end devices.

User Acceptance Testing (UAT)

Purpose: Collect real-world feedback to refine the **FoodTuck** user experience further and ensure customer satisfaction.

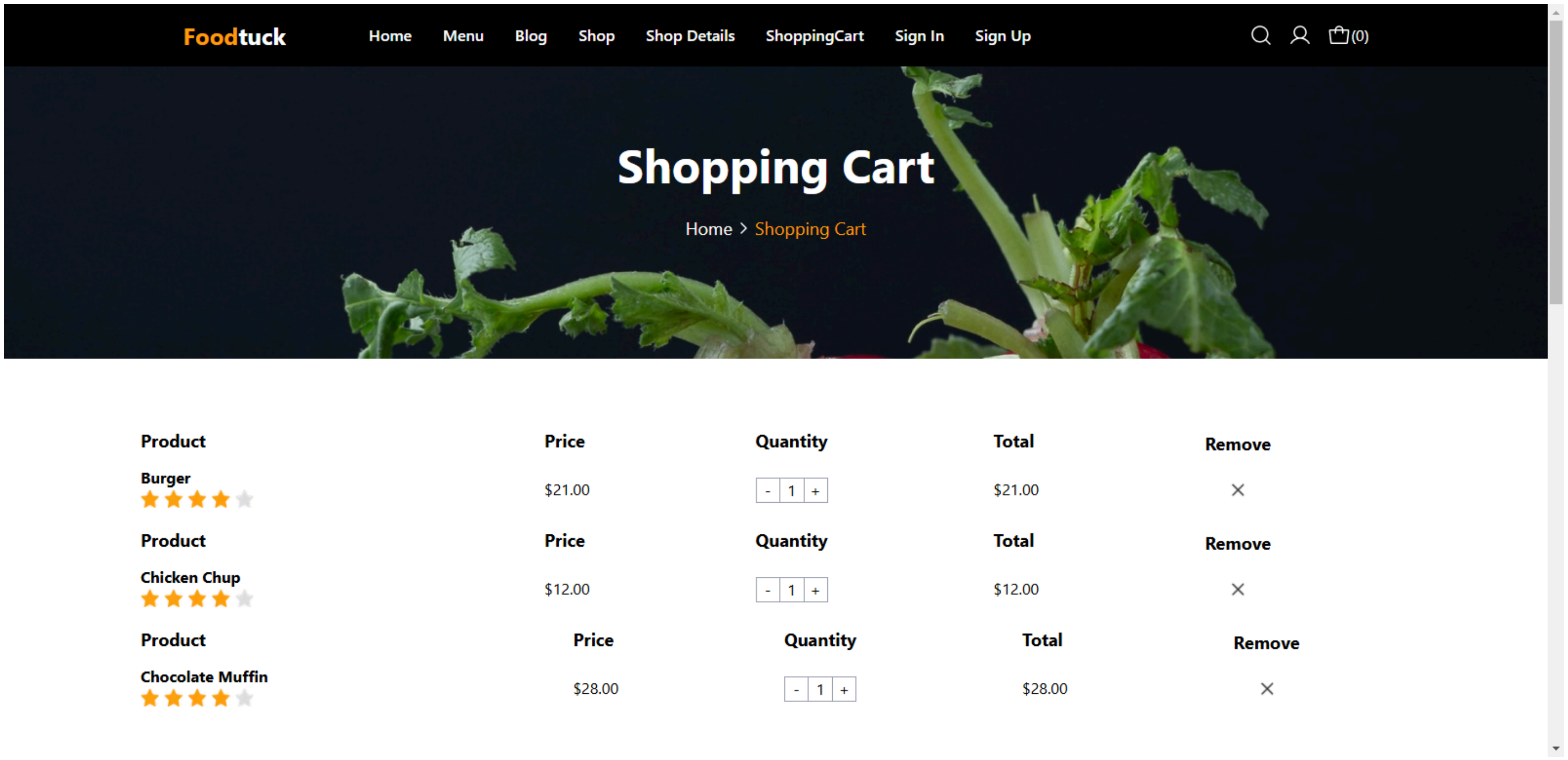
Scenarios Tested:

- Seamless browsing of product categories and food items.
- Smooth and intuitive adjustments to the shopping cart.
- Hassle-free processing of payments using diverse methods.
- Easy navigation between menus, blogs, and account-related features.
- Testing responsiveness across devices, including mobile, tablet, and desktop.
- Validation of coupon code application and discount visibility during checkout.
- Evaluation of the search functionality for accurate and quick results.
- Accessibility testing to ensure compatibility with screen readers and other assistive tools.

Feedback Implemented:

- Improved the visibility and design of "Add to Cart" buttons for better usability.
- Optimized the checkout workflow for faster and smoother order completion.
- Enhanced error messages during failed payments to guide users effectively.
- Reduced the number of steps required to complete an order for a quicker process.
- Introduced clear indicators for selected items in the cart.
- Fixed minor glitches in the category navigation menu for better consistency.
- Improved the performance of the search bar for faster query responses.
- Added feedback prompts after order placement to gather user opinions.
- Ensured the mobile version of the website delivers a flawless user experience.
- Verified that all email and SMS notifications (e.g., order confirmation) are sent without delay.

This thorough testing and feedback implementation aim to deliver a polished and intuitive platform for **FoodTuck** users.



Foodtuck

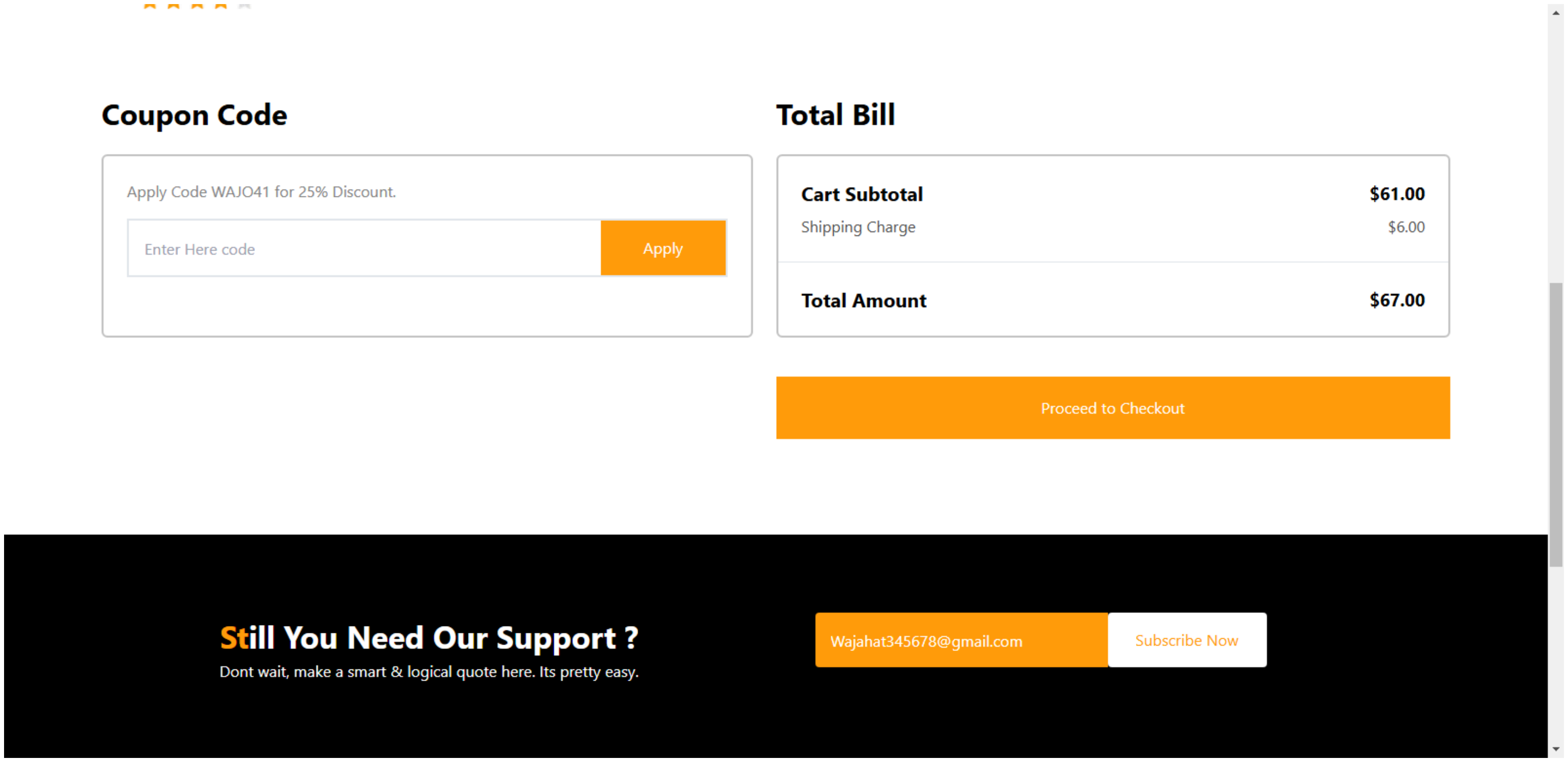
Home · Menu · Blog · Shop · Shop Details · ShoppingCart · Sign In · Sign Up

(0)

Shopping Cart

Home > Shopping Cart

Product	Price	Quantity	Total	Remove
Burger ★★★★☆	$21.00	- 1 +	$21.00	✕
Product	**Price**	**Quantity**	**Total**	**Remove**
Chicken Chup ★★★★☆	$12.00	- 1 +	$12.00	✕
Product	**Price**	**Quantity**	**Total**	**Remove**
Chocolate Muffin ★★★★☆	$28.00	- 1 +	$28.00	✕



Coupon Code

Apply Code WAJO41 for 25% Discount.

Enter Here code · Apply

Total Bill

Cart Subtotal	**$61.00**
Shipping Charge	$6.00
Total Amount	**$67.00**

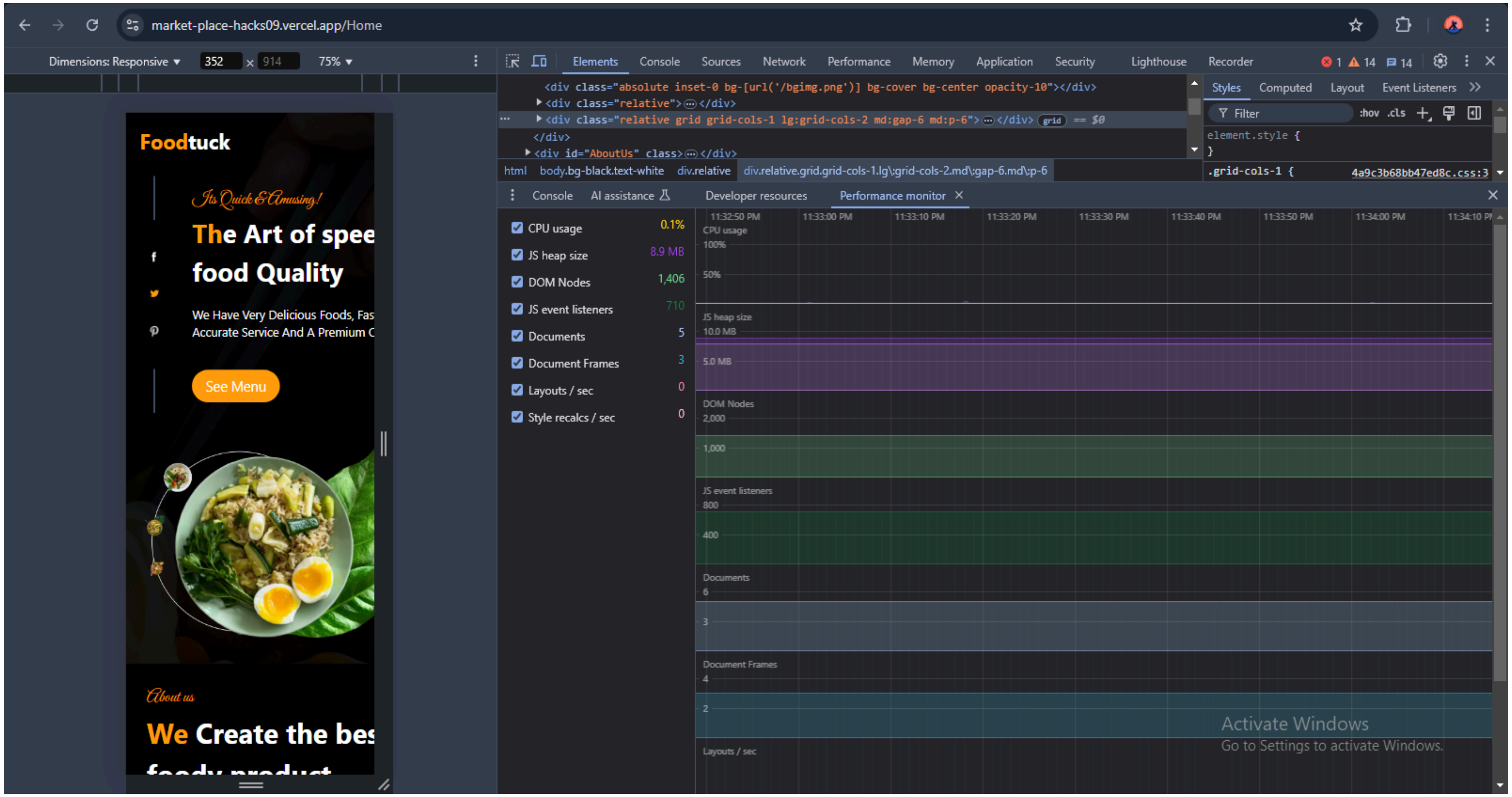
Proceed to Checkout

Still You Need Our Support ?

Dont wait, make a smart & logical quote here. Its pretty easy.

Wajahat345678@gmail.com · Subscribe Now

Performance Monitoring for FoodTuck



```
<div class="absolute inset-0 bg-[url('/bgimg.png')] bg-cover bg-center opacity-10"></div>
▸<div class="relative">⋯</div>
▸<div class="relative grid grid-cols-1 lg:grid-cols-2 md:gap-6 md:p-6">⋯</div> grid == $0
</div>
▸<div id="AboutUs" class>⋯</div>
```

html body.bg-black.text-white div.relative div.relative.grid.grid-cols-1.lg\:grid-cols-2.md\:gap-6.md\:p-6

```
element.style {
}
.grid-cols-1 {                4a9c3b68bb47ed8c.css:3
```

✓ CPU usage	0.1%
✓ JS heap size	8.9 MB
✓ DOM Nodes	1,406
✓ JS event listeners	710
✓ Documents	5
✓ Document Frames	3
✓ Layouts / sec	0
✓ Style recalcs / sec	0

Testing Summary Table

Test Case ID	Test Case Description	Test Steps	Expected Result	Actual Result	Status	Severity Level	Assigned To	Remarks
TC001	**Validate product listing page**	**Open product page > Verify product**	Products displayed correctly	**Products displayed correctly**	**Passed**	**Low**	**-**	**Product listing is accurate and meets user expectations.**
TC002	**Test API error handling**	**Disconne ct API > Refresh page**	Show fallback UI with error message	**Error message shown**	**Passed**	**Medium**	**-**	**Error handling mechanism works seamlessly under failure conditions**
TC003	**Check cart functionality**	**Add product to cart > Verify cart contents**	Cart updates with added product	**Cart updates as expected**	**Passed**	**High**	**-**	**Cart functionality is robust and user-friendly.**
TC004	**Ensure Responsiven ess on mobile**	**Resize browser window > Check layout**	Layout adjusts properly to screen size	**Responsive layout working as intended**	**Passed**	**Medium**	**-**	**Design ensures consistent user experience across all device**

Products API Status

1. **Successful API Response:**
   The API returned a 200 OK status, indicating that the request was successfully processed.

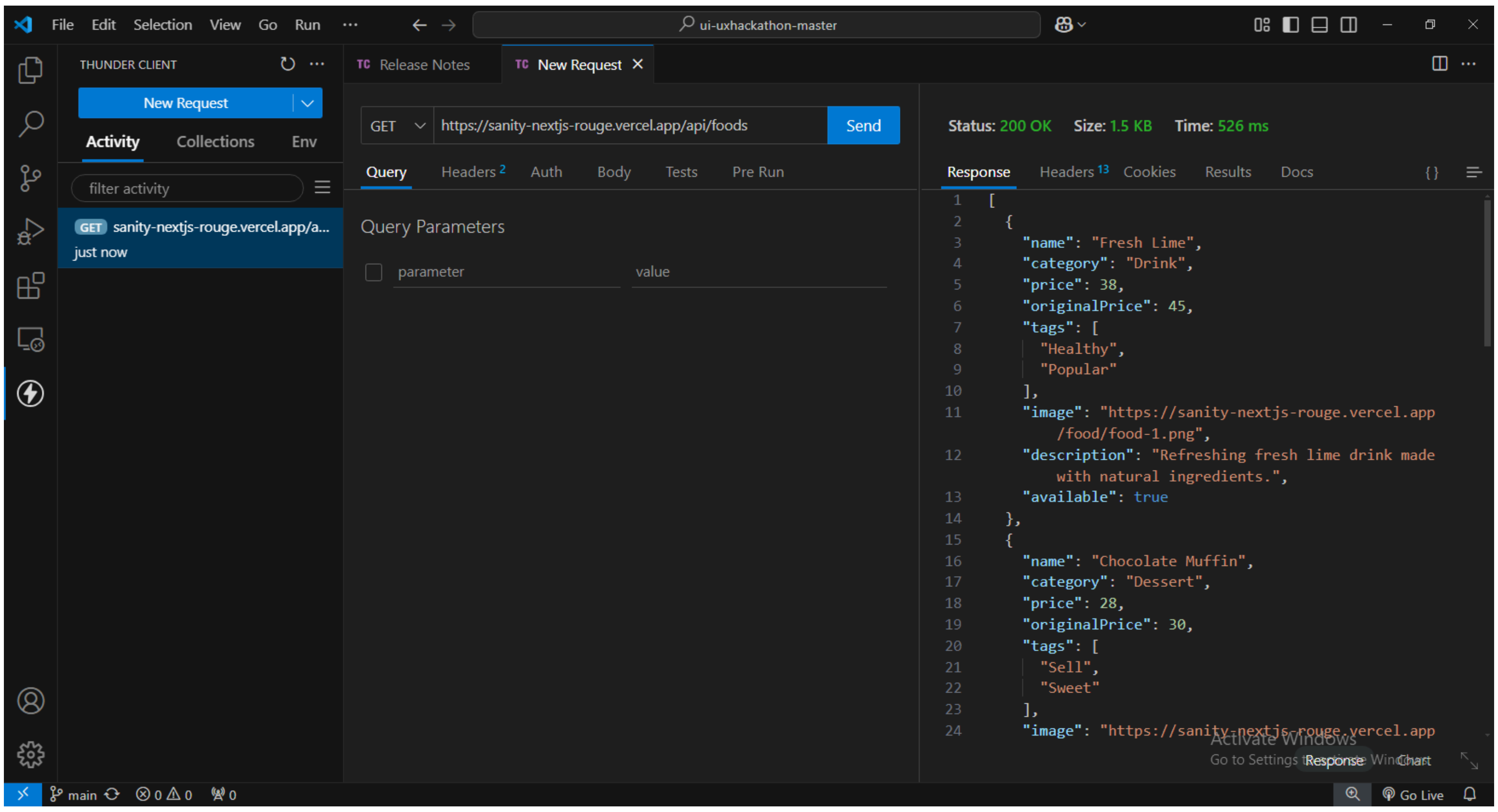
2. Data Structure and Details:

The API response includes comprehensive product details for multiple items, such as:

- **Name: The name of each product (e.g., "Fresh Lime", "Chocolate Muffin").**
- **Category: The category of each product (e.g., Drink, Dessert, Snacks, etc.).**
- **Price: Includes both the current price and the original price (if discounted).**
- **Additional Information:**
  - **Tags for categorization (e.g., "Healthy", "Popular", "Sweet").**
  - **Descriptions providing product highlights.**
  - **Availability status (e.g., true/false).**
  - **Image URLs for displaying product visuals.**

3:Response Time:

The API responded within 526 ms, ensuring a quick and efficient data retrieval process.



GET https://sanity-nextjs-rouge.vercel.app/api/foods Send

Status: 200 OK Size: 1.5 KB Time: 526 ms

```
[
  {
    "name": "Fresh Lime",
    "category": "Drink",
    "price": 38,
    "originalPrice": 45,
    "tags": [
      "Healthy",
      "Popular"
    ],
    "image": "https://sanity-nextjs-rouge.vercel.app
        /food/food-1.png",
    "description": "Refreshing fresh lime drink made
        with natural ingredients.",
    "available": true
  },
  {
    "name": "Chocolate Muffin",
    "category": "Dessert",
    "price": 28,
    "originalPrice": 30,
    "tags": [
      "Sell",
      "Sweet"
    ],
    "image": "https://sanity-nextjs-rouge.vercel.app
```

Final Notes

This phase focused on testing and optimizing the FoodTuck platform for a seamless user experience. Substantial progress was made in improving functionality, reliability, and performance to ensure a successful launch.

Next Steps:

- Finalize SEO strategies to enhance visibility and organic reach.
- Implement automated performance monitoring to maintain platform efficiency.
- Integrate customer behavior analytics for continuous UX improvement.

Author: Wajahat Ali

Roll Number: 00103369

Scheduled Slot: Saturday, 9:00 AM - 12:00 PM

Assigned By: Sir Ameen Alam

Class Teachers: Sir Bilal Muhammad Khan & Sir Aneeq Khatri