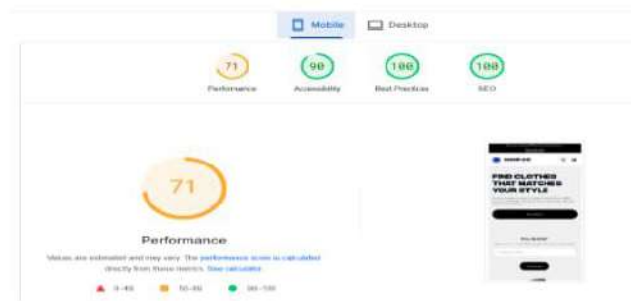


Testing Summary Table

Test No.	Feature Tested	Expected Outcome	Actual Outcome	Status	Priority	Notes
T001	Navigation consistency	Links direct accurately	Successful	Pass	Low	None
T002	Product visibility	Accurate product rendering	Verified	Pass	Medium	None
T003	Dynamic Working	Pages Rendering Without Issue	Functional	Pass	High	None
T004	Cart functionality	Seamless add/remove operation	Confirmed	Pass	High	None
T005	Login/SignUp	Under Construction	404	Fail	Low	None
T006	Performance benchmarks	Score ≥ 98	Archived 94	Fail	Medium	Requires Tuning
T007	Accessibility	Score ≥ 90	Reached 90	Pass	Medium	Maintain compliance
T008	SEO optimization	Score ≥ 90	Archived 100	Pass	High	None



## Device Compatibility

**Goal:** Ensure cross-platform reliability.

**Platforms Tested:**

- Browsers: Chrome, Firefox, Safari, Edge.
- Devices: Smartphones, tablets, and desktops (via **BrowserStack**).

**Focus Areas:**

- Maintain responsiveness across resolutions.
- Support for accessibility tools like screen readers.

## User Acceptance Testing (UAT)

**Purpose:** Gather real-world feedback.

**Scenarios Tested:**

- Browsing product categories.
- Smooth cart adjustments.
- Diverse payment method processing.

**Feedback Implemented:**

- Improved visibility of "Add to Cart" buttons.
- Streamlined checkout for speed.

## Enhanced Error Handling

**Objective:** Build a robust error mitigation strategy.

### Key Strategies:

- Add **fallback options** for API failures.
- Display descriptive error messages like "Product currently unavailable."
- Implement **centralized error logging** to simplify debugging.
- Alert users dynamically about stock shortages and system glitches.

### Example:

For unavailable items,

- Show "Item is out of stock" prominently.
- Restrict the checkout process until resolved.

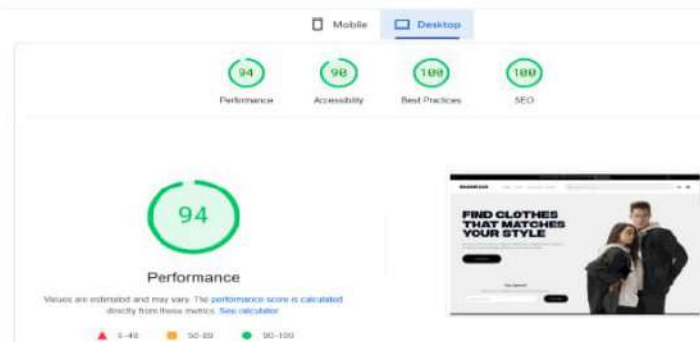
---

## Performance Tuning (Optimization)

**Focus:** Boost platform responsiveness using **PageSpeed**.

### Improvements Applied:

1. Reduced API response time.
2. Introduced **lazy-loading** for images.
3. Enabled **file compression** for faster page loads.
4. Implemented caching strategies for static assets.



## Day 5 — Elevating Testing, Debugging, and Backend Excellence

---

### Key Milestones

In this phase, we focus on ensuring that the **Shop.co Fashion Marketplace** is polished and deployment-ready. This includes:

1. Conducting rigorous testing to validate system performance.
  2. Implementing advanced error management mechanisms.
  3. Optimizing performance metrics and improving accessibility.
  4. Ensuring device and browser adaptability.
  5. Creating detailed documentation summarizing progress and results.
- 

### Action Plan

#### Functional Testing

**Goal:** Ensure every feature operates seamlessly.

**Areas Evaluated:**

- **Navigation Flow:** Validate that menu links are operational.
- **Product Page:** Confirm accurate product filtering and display.
- **Cart Operations:** Test addition, removal, and updates.
- **User Account Functions:** Validate registration and profile management.
- **Checkout Workflow:** Simulate payments and order completions.

**Tools Employed:**

- **Postman:** API testing.
  - **Jest and Enzyme:** For UI component tests.
  - **Cypress:** End-to-end workflow validation.
-