

## **MERN Stack Training**

**Company:** Sensation Software Solutions

**Student Name:** Tajinder Kaur

**Training Duration:** 6 Months

**Days:** 80

---

## **Objective of the Day**

The primary objective of Day 80 was to finalize the advanced frontend modules of the GadgetShop project. The modules targeted for final testing, polishing, and integration included:

1. Cart Component and Cart Page interaction flow.
2. Checkout Page interface, including summary structure.
3. Product Search functionality integrated with product list.
4. FilterSidebar for category-based and frontend-based product filtration.
5. Ensuring authentication validation for all the above modules.
6. Testing full workflow from product discovery to checkout readiness.
7. Achieving uniform styling and consistent user interface behaviour across components.

This day was focused on completing the end-to-end functional flow of the user journey on the platform, ensuring that the modules worked cohesively rather than in isolation.

---

## **Work Done on Day 80**

The finalization process for each module was broken down into iterative improvement stages. The tasks completed were as follows:

### **1. Component Review and Cleanup**

- Checked file structure to ensure every component was placed in the intended directories.
- Reviewed component names, folder names, and imports for consistency.
- Removed unused states, variables, and event handlers from the components.
- Rewrote comments inside components for easier understanding.
- Standardized naming conventions such as:
  - Cart.jsx
  - Checkout.jsx
  - SearchBar.jsx
  - FilterSidebar.jsx
- Verified that all components used React hooks correctly, mainly:
  - useState, useEffect, useContext, and routing hooks like useNavigate.

## **2. Authentication Dependency Integration**

- Ensured that users must be authenticated to perform the following tasks:
  - Adding items to the cart.
  - Viewing the cart.
  - Proceeding to checkout.
  - Searching within the shop.
  - Applying filters to product lists.
- Added redirection triggers:
  - If unauthenticated user attempts to open /cart, redirect to /login.
  - If unauthenticated user selects checkout, redirect to /login.
- Added conditional rendering logic to hide restricted buttons when user is logged out.

## **3. Cart Component Finalization**

- Validated that the component fetches the correct cart state from context.
- Checked that cart items display:
  - Dynamic product name
  - Unit price
  - Product quantity
  - Subtotal calculation
- Confirmed interaction elements:
  - Increment quantity button
  - Decrement quantity button
  - Remove item button
- Re-tested the cart summary area to verify:
  - Grand total calculation
  - Link to checkout button
  - Empty cart message display
- Ensured responsiveness:
  - Two-column layout on large screens
  - Single-column layout on mobile
- Verified layout alignments using Tailwind's:
  - grid, flex, justify-between, gap, and space-y

## **4. Checkout Page Completion**

- Verified that the checkout page does not open with an empty cart.
- Validated structural layout:
  - Billing Form Section
  - Cart Summary Section
- Billing Form Final Checks:
  - Shipping address fields and validation
  - Contact details fields with mandatory indicators
  - Payment method radio selection (frontend only)
- Summary Section Final Checks:
  - Subtotal, tax placeholder, and final total output
  - Order review panel displaying cart items again
- Verified navigation flow:
  - Back to cart button

- Confirm order placeholder button
- Designed consistency checks:
  - Same button styles across the platform
  - Uniform input focus behavior

## 5. SearchBar Final Enhancement

- Positioning confirmed at the top of the Shop page in a fixed structure.
- Input field configured for:
  - Typing delay handling
  - Real-time filtering without page reload
- Display logic confirmed:
  - Clear input resets the product listing
  - Search query matches both uppercase and lowercase inputs
- Verified that SearchBar interacts with:
  - Product list array stored in state
  - Filtering function without backend request

## 6. FilterSidebar Polishing

- Finalized filter categories including:
    - Gadget categories (e.g., mobiles, laptops, accessories)
    - Price range selection through slider or range inputs
    - Rating filtering placeholder for frontend
  - Confirmed interaction behaviour:
    - Filters applied instantly on selection
    - Multiple filters stack together instead of replacing each other
  - Added reset filter button to return to full product list
  - Implemented mobile-responsive behavior:
    - Sidebar collapsible
    - Expand/collapse triggered by button
- 

## Conclusion

Day 80 serves as a major milestone achievement where the user journey pipeline Shop → Discover → Select → Cart → Checkout is fully operational on the frontend and prepared for the next development phase.