Epic

EPIC-001: Enhance User Shopping Experience on Tira Beauty Website  
Description: Improve the overall shopping flow including search, product details, checkout, and payment integration to increase conversion rate and customer satisfaction.

User Stories

(Each story links to the Epic)

Story-001: Implement and Optimize Product Search

* As a shopper
* I want to search products by name, brand, and category
* So that I can quickly find the items I need
* Acceptance Criteria:
  + Search bar is visible on all pages
  + Search results are relevant and sorted by popularity/relevance
  + Filters and sorting options available

Story-002: Improve Product Detail Page

* As a customer
* I want to see clear product images, descriptions, reviews, and price details
* So that I can make an informed purchase decision
* Acceptance Criteria:
  + High-resolution images with zoom-in feature
  + Ingredients, usage instructions, and brand info displayed
  + Customer reviews section functional

Story-003: Enhance Add-to-Cart and Checkout Flow

* As a user
* I want a smooth add-to-cart and secure checkout experience
* So that I can complete my purchase without frustration
* Acceptance Criteria:
  + Add-to-cart works from product list and detail pages
  + Cart updates instantly without page reload
  + Payment gateway integration with success and failure handling

Story-004: Mobile Responsiveness

* As a mobile shopper
* I want the site to be responsive and load fast on mobile devices
* So that I can shop easily from anywhere
* Acceptance Criteria:
  + Pages adjust to different screen sizes
  + No overlapping elements or broken layouts on mobile

Tasks

(Example breakdown for Story-001: Implement and Optimize Product Search)

1. TASK-001: Design UI for search bar and results page
2. TASK-002: Integrate backend API for product search
3. TASK-003: Implement filters (category, brand, price range)
4. TASK-004: Write automated test cases for search functionality
5. TASK-005: Conduct usability testing on search

(Example breakdown for Story-003: Enhance Add-to-Cart and Checkout Flow)

1. TASK-006: Implement AJAX add-to-cart feature
2. TASK-007: Integrate payment gateway (Razorpay/UPI/Cards)
3. TASK-008: Add cart persistence for logged-in users
4. TASK-009: Test payment success/failure scenarios
5. TASK-010: Write regression test cases

Bugs

(Example defects found during exploratory testing on Tira Beauty)

1. BUG-001: Search results not updating when changing filters (Severity: High, Priority: P1)
2. BUG-002: Product images not zooming on mobile (Severity: Medium, Priority: P2)
3. BUG-003: Checkout page hangs after payment success (Severity: Critical, Priority: P1)
4. BUG-004: Some menu items are not clickable in mobile view (Severity: High, Priority: P1)
5. BUG-005: Page load time exceeds 5 seconds on home page (Severity: Medium, Priority: P3)

**Epic**

**EPIC-001: Tira Red Module Development & Quality Assurance**  
**Description:**  
Implement and test the new "Tira Red" loyalty and rewards module in the Tira Beauty application, enabling customers to earn, redeem, and track loyalty points across purchases.

**Goals:**

* Seamless integration of Tira Red loyalty program
* Smooth UI/UX for earning and redeeming points
* Accurate tracking and reporting of loyalty data

**User Stories**

**Story 1: Customer Enrollment in Tira Red**

**ID:** STORY-001  
**Description:**  
As a customer, I want to enroll in the Tira Red loyalty program so that I can earn points on my purchases.

**Acceptance Criteria:**

* Enrollment option is available in the app and web platform
* Successful enrollment confirmation displayed
* User account updated with Tira Red membership status

**Story 2: Earn Points on Purchase**

**ID:** STORY-002  
**Description:**  
As a customer, I want to earn loyalty points automatically on eligible purchases so that I can redeem them later.

**Acceptance Criteria:**

* Points calculation based on purchase amount and tier rules
* Points visible in user’s account after purchase confirmation
* Points reflected in backend loyalty database

**Story 3: Redeem Points at Checkout**

**ID:** STORY-003  
**Description:**  
As a customer, I want to redeem my loyalty points during checkout so that I can get discounts on my purchases.

**Acceptance Criteria:**

* Redemption option available at checkout
* Discount applied based on redeemed points value
* Points balance updated after redemption

**Tasks (Sample for Story 2: Earn Points on Purchase)**

**TASK-001:** Integrate backend API for points calculation  
**TASK-002:** Update frontend UI to display earned points after purchase  
**TASK-003:** Implement database triggers for points update  
**TASK-004:** Test API response for different purchase amounts and tiers  
**TASK-005:** Perform UI testing for points display on purchase confirmation page

**Bugs (Sample for Story 3: Redeem Points)**

**BUG-001:** Redeemed points not deducted from user account after checkout  
**BUG-002:** Incorrect discount value applied for redeemed points  
**BUG-003:** Points redemption option missing in mobile app checkout flow  
**BUG-004:** Error message displayed when redeeming points for guest users  
**BUG-005:** UI alignment issue in checkout page when redemption field is visible

**EPIC**

**EPIC-NEW-001: “What’s New” Module Enhancement**  
**Description:**  
Launch a dynamic “What’s New” content module for Tira Beauty that highlights fresh trends, products, and editorial content, ensuring users discover the latest offerings seamlessly.

**User Stories**

**Story 1: Editorial “What’s New” Content Feed**

**ID: STORY-NEW-001**  
**As a** beauty enthusiast, **I want** to view curated editorial content (e.g., “7 lipstick and lip liner combos to buy right now”) **so that** I can stay updated on trending beauty advice and guides.  
**Acceptance Criteria:**

* Editorial posts visible in a feed on the “What’s New” page
* Each item shows title, summary, date (e.g., August 8, 2025), and an image [Tira](https://www.tirabeauty.com/article?srsltid=AfmBOooxZu23ddWRBk_UZvr1W_5n-vJ2_hGS03UDVwCoOnAeB65MeSP_&utm_source=chatgpt.com)
* Clicking an item opens full content

**Story 2: “Newly Arrived” Product Carousel**

**ID: STORY-NEW-002**  
**As a** frequent shopper, **I want** a carousel showcasing newly added products and brands (e.g., Chantecaille, Dr. Jart⁺) **so that** I can easily explore the latest offerings.  
**Acceptance Criteria:**

* A visually engaging carousel “New On Tira”
* Auto-scrolling, user-controlled navigation
* Product tiles include image, brand, name, and badge (“New”)

**Story 3: Personalized Trend Highlights**

**ID: STORY-NEW-003**  
**As a** user, **I want** trending themes like “Guava Girl Makeup” or “Peptide Everything” tailored to my browsing history **so that** the “What’s New” feels relevant.  
**Acceptance Criteria:**

* Trend modules dynamically generated based on user behavior
* At least 3 personalized trend tiles displayed
* Refreshed on each login or weekly

**Tasks (aligned under each user story)**

**For Story 1:**

* TASK-NEW-101: Design editorial feed UI mockups
* TASK-NEW-102: Build backend API to fetch editorial items
* TASK-NEW-103: Integrate editorial feed in frontend
* TASK-NEW-104: QA visual and functional consistency across viewports

**For Story 2:**

* TASK-NEW-201: Design new-product carousel UI
* TASK-NEW-202: Develop API to retrieve newly launched products
* TASK-NEW-203: Implement carousel with lazy loading
* TASK-NEW-204: Test responsive behavior and accessibility

**For Story 3:**

* TASK-NEW-301: Analyze user data to define trend personalization logic
* TASK-NEW-302: Create backend service for personalized trend suggestions
* TASK-NEW-303: Frontend implementation of personalized trend tiles
* TASK-NEW-304: A/B testing setup to measure engagement uplift

**Bug Examples (to capture during testing)**

* **BUG-NEW-001:** Editorial posts show incorrect publish dates
* **BUG-NEW-002:** Carousel items fail to load or images overlap
* **BUG-NEW-003:** Trend personalization doesn’t update per user – stale content shown
* **BUG-NEW-004:** Clicking an editorial tile throws a 404 error
* **BUG-NEW-005:** Carousel doesn’t display correctly on mobile devices (UI overlap or cutoff)

**Epic**

**EPIC-HC-001: Help Center Implementation & Optimization**  
**Description:**  
Develop and integrate a centralized Help Center in the Tira Beauty application and website, allowing customers to find FAQs, submit support tickets, and track resolutions easily.

**Goals:**

* Improve self-service customer support
* Reduce support ticket load
* Enhance customer experience with clear, accessible information

**User Stories**

**Story 1: Browse FAQs**

**ID:** STORY-HC-001  
**As a** customer, **I want** to browse categorized FAQs **so that** I can quickly find answers to common questions without contacting support.

**Acceptance Criteria:**

* FAQs grouped by category (Orders, Payments, Returns, Loyalty, etc.)
* Search functionality within FAQs
* Responsive design for desktop and mobile

**Story 2: Submit a Support Ticket**

**ID:** STORY-HC-002  
**As a** customer, **I want** to submit a support ticket from the Help Center **so that** I can get assistance when FAQs don’t resolve my issue.

**Acceptance Criteria:**

* Ticket form with required fields (Name, Email, Issue Category, Description, Attachments)
* Ticket ID generated and displayed after submission
* Email confirmation sent to customer

**Story 3: Track Ticket Status**

**ID:** STORY-HC-003  
**As a** customer, **I want** to track the status of my submitted ticket **so that** I know when my issue will be resolved.

**Acceptance Criteria:**

* Ticket status page with filters (Open, In Progress, Resolved)
* Status auto-updated from backend
* Option to reply or add notes to an active ticket

**Tasks**

**For Story 1: Browse FAQs**

* TASK-HC-101: Design Help Center FAQ page UI
* TASK-HC-102: Create FAQ category and content management backend
* TASK-HC-103: Implement search functionality with keyword highlighting
* TASK-HC-104: Test FAQ display and search on desktop/mobile

**For Story 2: Submit a Support Ticket**

* TASK-HC-201: Build support ticket submission form UI
* TASK-HC-202: Connect ticket form to backend API
* TASK-HC-203: Implement email notification service for ticket confirmation
* TASK-HC-204: Validate ticket form fields and error handling

**For Story 3: Track Ticket Status**

* TASK-HC-301: Create ticket status tracking UI
* TASK-HC-302: Integrate with backend to fetch ticket status updates
* TASK-HC-303: Add filter and search for tickets
* TASK-HC-304: Enable adding notes or follow-ups to tickets

**Bugs**

* **BUG-HC-001:** FAQ search returns irrelevant results for certain keywords
* **BUG-HC-002:** Ticket form allows submission without mandatory fields
* **BUG-HC-003:** Ticket confirmation email not sent after submission
* **BUG-HC-004:** Ticket status not updating in real-time
* **BUG-HC-005:** Mobile Help Center layout breaks on smaller screens

**Epic**

**EPIC-HC-001: Help Center Implementation & Optimization**  
**Description:**  
Develop and integrate a centralized Help Center in the Tira Beauty application and website, allowing customers to find FAQs, submit support tickets, and track resolutions easily.

**Goals:**

* Improve self-service customer support
* Reduce support ticket load
* Enhance customer experience with clear, accessible information

**User Stories**

**Story 1: Browse FAQs**

**ID:** STORY-HC-001  
**As a** customer, **I want** to browse categorized FAQs **so that** I can quickly find answers to common questions without contacting support.

**Acceptance Criteria:**

* FAQs grouped by category (Orders, Payments, Returns, Loyalty, etc.)
* Search functionality within FAQs
* Responsive design for desktop and mobile

**Story 2: Submit a Support Ticket**

**ID:** STORY-HC-002  
**As a** customer, **I want** to submit a support ticket from the Help Center **so that** I can get assistance when FAQs don’t resolve my issue.

**Acceptance Criteria:**

* Ticket form with required fields (Name, Email, Issue Category, Description, Attachments)
* Ticket ID generated and displayed after submission
* Email confirmation sent to customer

**Story 3: Track Ticket Status**

**ID:** STORY-HC-003  
**As a** customer, **I want** to track the status of my submitted ticket **so that** I know when my issue will be resolved.

**Acceptance Criteria:**

* Ticket status page with filters (Open, In Progress, Resolved)
* Status auto-updated from backend
* Option to reply or add notes to an active ticket

**Tasks**

**For Story 1: Browse FAQs**

* TASK-HC-101: Design Help Center FAQ page UI
* TASK-HC-102: Create FAQ category and content management backend
* TASK-HC-103: Implement search functionality with keyword highlighting
* TASK-HC-104: Test FAQ display and search on desktop/mobile

**For Story 2: Submit a Support Ticket**

* TASK-HC-201: Build support ticket submission form UI
* TASK-HC-202: Connect ticket form to backend API
* TASK-HC-203: Implement email notification service for ticket confirmation
* TASK-HC-204: Validate ticket form fields and error handling

**For Story 3: Track Ticket Status**

* TASK-HC-301: Create ticket status tracking UI
* TASK-HC-302: Integrate with backend to fetch ticket status updates
* TASK-HC-303: Add filter and search for tickets
* TASK-HC-304: Enable adding notes or follow-ups to tickets

**Bugs**

* **BUG-HC-001:** FAQ search returns irrelevant results for certain keywords
* **BUG-HC-002:** Ticket form allows submission without mandatory fields
* **BUG-HC-003:** Ticket confirmation email not sent after submission
* **BUG-HC-004:** Ticket status not updating in real-time
* **BUG-HC-005:** Mobile Help Center layout breaks on smaller screens

If you want, I can **put this into a Jira importable Excel/CSV** with proper Epic → Story → Task → Bug hierarchy so you can load it directly into your Jira project without retyping anything.

Do you want me to prepare that file?

**You said:**

create epic , story, task and bugs for tira beauty cart

**ChatGPT said:**

Here’s a **Jira-ready backlog structure** (Epic → Story → Task → Bug) for the **Tira Beauty – Cart Module**.

**Epic**

**EPIC-CART-001: Tira Beauty Shopping Cart Enhancement**  
**Description:**  
Develop, optimize, and test the Tira Beauty shopping cart feature to allow users to seamlessly add, update, and remove products, view cart totals, and proceed to checkout without issues.

**Goals:**

* Smooth cart management experience on web and mobile
* Accurate price, quantity, and discount calculations
* Reliable sync between user sessions and accounts

**User Stories**

**Story 1: Add Items to Cart**

**ID:** STORY-CART-001  
**As a** customer, **I want** to add products to my shopping cart **so that** I can review and purchase them later.

**Acceptance Criteria:**

* Add-to-cart button available on product listing and product details pages
* Success message displayed after adding item
* Cart icon updates with correct item count

**Story 2: Update Cart Items**

**ID:** STORY-CART-002  
**As a** customer, **I want** to change the quantity or remove items from my cart **so that** I can adjust my purchase before checkout.

**Acceptance Criteria:**

* Quantity update instantly reflects new subtotal
* Remove button deletes product from cart
* Cart total recalculates automatically

**Story 3: Apply Discounts & Coupons**

**ID:** STORY-CART-003  
**As a** customer, **I want** to apply promo codes or coupons in the cart **so that** I can get discounts before checkout.

**Acceptance Criteria:**

* Coupon field accepts valid codes
* Error message displayed for invalid/expired codes
* Discount applied to cart total instantly

**Story 4: Persist Cart Across Devices**

**ID:** STORY-CART-004  
**As a** logged-in customer, **I want** my cart to be synced across devices **so that** I can shop from multiple devices without losing my selections.

**Acceptance Criteria:**

* Cart items saved in backend linked to user ID
* Cart loads correctly on web and mobile for the same account

**Tasks**

**For Story 1: Add Items to Cart**

* TASK-CART-101: Design and place “Add to Cart” button on product pages
* TASK-CART-102: Implement frontend logic for cart icon update
* TASK-CART-103: Create backend API for adding items to cart
* TASK-CART-104: Test adding items under different network conditions

**For Story 2: Update Cart Items**

* TASK-CART-201: Implement quantity selector with +/– buttons
* TASK-CART-202: Create API endpoint for updating and removing items
* TASK-CART-203: Test subtotal and total recalculation logic

**For Story 3: Apply Discounts & Coupons**

* TASK-CART-301: Develop promo code input and validation logic
* TASK-CART-302: Integrate backend coupon validation service
* TASK-CART-303: Verify discount percentage and fixed amount logic

**For Story 4: Persist Cart Across Devices**

* TASK-CART-401: Store cart items in backend DB for logged-in users
* TASK-CART-402: Sync cart API for cross-device consistency
* TASK-CART-403: Test cart persistence after logout/login

**Bugs**

* **BUG-CART-001:** Cart icon not updating after adding item
* **BUG-CART-002:** Quantity update not reflected until page refresh
* **BUG-CART-003:** Invalid coupon code accepted without error message
* **BUG-CART-004:** Cart items disappear after session timeout
* **BUG-CART-005:** Cart total miscalculates when applying multiple discounts
* **BUG-CART-006:** Remove button fails on mobile web version

**Epic**

**EPIC-FRAG-001: Tira Beauty Fragrance Shopping Experience**  
**Description:**  
Build and enhance the fragrance shopping section in Tira Beauty, allowing customers to browse, search, filter, and purchase perfumes, deodorants, and luxury scents seamlessly.

**Goals:**

* Curated fragrance catalog with rich product details
* Advanced filters and sorting for scent type, brand, and price
* Smooth add-to-cart and checkout integration

**User Stories**

**Story 1: Browse Fragrance Catalog**

**ID:** STORY-FRAG-001  
**As a** customer, **I want** to browse a fragrance catalog **so that** I can discover perfumes and scents that match my preferences.

**Acceptance Criteria:**

* Fragrance listing with brand, image, price, and “New/Best Seller” tags
* Pagination or infinite scroll for smooth browsing
* Filters for gender, fragrance family, brand, and price range

**Story 2: Product Details Page for Fragrances**

**ID:** STORY-FRAG-002  
**As a** customer, **I want** detailed fragrance descriptions and scent notes **so that** I can make informed purchase decisions.

**Acceptance Criteria:**

* Display top/middle/base notes in a scent pyramid format
* Show ratings, reviews, and related products
* Add-to-cart button available directly on details page

**Story 3: Search & Filter for Fragrances**

**ID:** STORY-FRAG-003  
**As a** customer, **I want** to search and apply filters to find specific fragrances **so that** I can quickly locate products I’m interested in.

**Acceptance Criteria:**

* Keyword search with autocomplete suggestions
* Multiple filter selection with instant results update
* “Clear all filters” option

**Story 4: Personalized Recommendations**

**ID:** STORY-FRAG-004  
**As a** returning customer, **I want** fragrance recommendations based on my past purchases and browsing history **so that** I can find scents I might like.

**Acceptance Criteria:**

* Recommendation engine integrated on fragrance homepage and product page
* Minimum of 4 recommended products displayed
* Recommendations refresh weekly

**Tasks**

**For Story 1: Browse Fragrance Catalog**

* TASK-FRAG-101: Design fragrance listing page UI
* TASK-FRAG-102: Implement backend API for fetching fragrance products
* TASK-FRAG-103: Integrate filters and sorting in frontend
* TASK-FRAG-104: Test pagination and performance

**For Story 2: Product Details Page**

* TASK-FRAG-201: Create fragrance detail template with scent pyramid
* TASK-FRAG-202: Connect detail page to backend product data
* TASK-FRAG-203: Implement “Add to Cart” functionality
* TASK-FRAG-204: Test layout on desktop and mobile

**For Story 3: Search & Filter**

* TASK-FRAG-301: Build search API with keyword suggestions
* TASK-FRAG-302: Implement multi-filter selection UI
* TASK-FRAG-303: Optimize search speed for large catalog

**For Story 4: Personalized Recommendations**

* TASK-FRAG-401: Create recommendation logic based on user data
* TASK-FRAG-402: Integrate recommendation API into frontend
* TASK-FRAG-403: Test recommendation accuracy with sample accounts

**Bugs**

* **BUG-FRAG-001:** Filters not applying correctly when multiple options selected
* **BUG-FRAG-002:** Fragrance detail page missing scent notes for some products
* **BUG-FRAG-003:** Search autocomplete returns irrelevant products
* **BUG-FRAG-004:** Recommendations not updating after new purchases
* **BUG-FRAG-005:** Pagination skips products on mobile version
* **BUG-FRAG-006:** Add-to-cart button unresponsive on product detail page

**Epic**

**EPIC-MKP-001: Tira Beauty Makeup Shopping & Discovery Experience**  
**Description:**  
Develop and optimize the Makeup section in Tira Beauty, enabling customers to explore, filter, and purchase a wide range of makeup products with a smooth, engaging, and personalized shopping experience.

**Goals:**

* Rich product catalog with detailed makeup information
* Filters by product type, brand, price, and shades
* Integration with virtual try-on and personalized recommendations

**User Stories**

**Story 1: Browse Makeup Catalog**

**ID:** STORY-MKP-001  
**As a** customer, **I want** to view all makeup products in a well-organized catalog **so that** I can easily find what I need.

**Acceptance Criteria:**

* Grid/list view of products with image, price, brand, and rating
* Filters for category (lipstick, foundation, etc.), brand, price, and shade
* Sort options by price, popularity, and newest arrivals

**Story 2: Makeup Product Details Page**

**ID:** STORY-MKP-002  
**As a** customer, **I want** to see detailed product descriptions, shade availability, and usage tips **so that** I can make informed purchase decisions.

**Acceptance Criteria:**

* Shade swatches with real-time image change
* Ingredients and product benefits listed
* “Add to Cart” and “Add to Wishlist” options available

**Story 3: Virtual Try-On Feature**

**ID:** STORY-MKP-003  
**As a** customer, **I want** to virtually try on makeup shades using my camera or uploaded photo **so that** I can see how the product looks on me before buying.

**Acceptance Criteria:**

* Virtual try-on for lipstick, eyeshadow, blush, etc.
* Option to switch between shades instantly
* Works on both mobile app and desktop

**Story 4: Personalized Makeup Recommendations**

**ID:** STORY-MKP-004  
**As a** returning customer, **I want** recommendations based on my skin tone, past purchases, and browsing history **so that** I can discover relevant products.

**Acceptance Criteria:**

* Recommendation section on Makeup homepage and product page
* Recommendations refresh weekly
* Minimum of 4 products shown per recommendation section

**Tasks**

**For Story 1: Browse Makeup Catalog**

* TASK-MKP-101: Design makeup listing page UI
* TASK-MKP-102: Implement backend API for fetching makeup products
* TASK-MKP-103: Integrate filters and sorting options in frontend
* TASK-MKP-104: Test performance and responsiveness

**For Story 2: Makeup Product Details Page**

* TASK-MKP-201: Create product detail template with shade swatches
* TASK-MKP-202: Implement dynamic image change based on selected shade
* TASK-MKP-203: Connect “Add to Cart” and “Wishlist” APIs
* TASK-MKP-204: Test mobile and desktop layouts

**For Story 3: Virtual Try-On Feature**

* TASK-MKP-301: Integrate AR SDK for makeup try-on
* TASK-MKP-302: Implement real-time shade switching
* TASK-MKP-303: Test try-on accuracy with different lighting conditions

**For Story 4: Personalized Recommendations**

* TASK-MKP-401: Build recommendation logic using user data
* TASK-MKP-402: Integrate recommendation API into frontend
* TASK-MKP-403: Test recommendation accuracy and relevance

**Bugs**

* **BUG-MKP-001:** Filters not working for multiple shade selections
* **BUG-MKP-002:** Product images not updating when shade selected
* **BUG-MKP-003:** Virtual try-on feature crashes on certain mobile devices
* **BUG-MKP-004:** Recommendations showing irrelevant products
* **BUG-MKP-005:** Add-to-Cart button unresponsive in some browsers
* **BUG-MKP-006:** Makeup product page loading time too long

**Epic**

**EPIC-MB-001: Tira Beauty Mom & Baby Shopping & Content Module**  
**Description:**  
Develop and optimize the Mom & Baby section in Tira Beauty, providing customers with a curated selection of products and resources for mothers and infants, along with personalized recommendations and helpful guides.

**Goals:**

* Wide range of mom & baby products with detailed info
* Easy-to-use filters and sorting options
* Educational content for parenting and newborn care
* Seamless cart and checkout integration

**User Stories**

**Story 1: Browse Mom & Baby Catalog**

**ID:** STORY-MB-001  
**As a** customer, **I want** to browse all mom & baby products in an organized catalog **so that** I can easily find what I need.

**Acceptance Criteria:**

* Grid/list view of products with images, brand, price, and ratings
* Filters for product type (diapers, skincare, feeding, toys), brand, age group, and price
* Sort options for price, popularity, and newest arrivals

**Story 2: Product Details Page for Mom & Baby Items**

**ID:** STORY-MB-002  
**As a** customer, **I want** to see detailed descriptions, ingredients, usage instructions, and safety certifications **so that** I can make informed purchases.

**Acceptance Criteria:**

* Clear product images from multiple angles
* Safety certifications and age suitability
* “Add to Cart” and “Wishlist” buttons available

**Story 3: Parenting & Baby Care Articles**

**ID:** STORY-MB-003  
**As a** parent, **I want** to access articles and tips for baby care and parenting **so that** I can get trusted guidance.

**Acceptance Criteria:**

* Articles displayed in a separate “Parenting Tips” section
* Search and category filter for articles
* Share option for articles via social media

**Story 4: Personalized Recommendations for Mom & Baby**

**ID:** STORY-MB-004  
**As a** returning customer, **I want** personalized product recommendations **so that** I can discover relevant products for my child’s needs.

**Acceptance Criteria:**

* Recommendations based on child’s age, past purchases, and browsing history
* Recommendations update weekly
* At least 4 relevant products shown per section

**Tasks**

**For Story 1: Browse Mom & Baby Catalog**

* TASK-MB-101: Design Mom & Baby category listing page
* TASK-MB-102: Implement backend API for fetching products
* TASK-MB-103: Integrate filters and sorting options in frontend
* TASK-MB-104: Test catalog performance on web & mobile

**For Story 2: Product Details Page**

* TASK-MB-201: Create product detail page with multiple image views
* TASK-MB-202: Display product safety and age suitability
* TASK-MB-203: Implement “Add to Cart” and “Wishlist” functionality
* TASK-MB-204: Test UI for accessibility and responsiveness

**For Story 3: Parenting & Baby Care Articles**

* TASK-MB-301: Design parenting tips section UI
* TASK-MB-302: Build CMS for article management
* TASK-MB-303: Implement search and category filter for articles
* TASK-MB-304: Enable social media sharing for articles

**For Story 4: Personalized Recommendations**

* TASK-MB-401: Build recommendation logic using child profile data
* TASK-MB-402: Integrate recommendation API with frontend
* TASK-MB-403: Test relevance of recommended products

**Bugs**

* **BUG-MB-001:** Filters not applying correctly for age group selection
* **BUG-MB-002:** Safety certification badge missing for some products
* **BUG-MB-003:** Articles not loading under “Parenting Tips” section
* **BUG-MB-004:** Recommendations showing irrelevant products for age group
* **BUG-MB-005:** Add-to-Cart button unresponsive for certain baby products
* **BUG-MB-006:** Mobile layout misalignment in Mom & Baby section