Fifty Firsts Wellness — SHOP Specification

1. Core Shop Features Overview

Feature	Description / Business Logic	Frontend Tasks	Backend Tasks
Product Listing Cards	Each product appears as a card showing image, name, price, and CTA buttons (Pre-Order / Add to Cart).	Display product grid with filters (category, availability). Add Pre-order and Notification icons.	API endpoint to fetch all products with attributes: `id, name, price, stock, preorder_count, discount, reviews, image_url, category`.
Pre-Order Functionality	Users can "Pre-Order" an item not yet available. The system tracks total pre-orders per product.	"Pre-order" button visible when `product.status = 'coming_soon'`. Update state in UI after click.	Endpoint: `/api/products/preorder` → Records user_id, product_id, timestamp. Update `preorder_count`. Return new preorder total.
Out-of-Stock Notification (Bell Icon)	When a product's stock = 0, bell icon appears. Users can opt to be notified when item restocks.	Display bell icon conditionally. On click → trigger "notify me when available" modal.	Endpoint: `/api/products/notify-me` → stores user_id + product_id. On stock update > 0, auto-trigger Maileroo API to email users.
Dashboard Tracking (User Activity)	Users can view all marketplace interactions (preorders, notifications, purchases) in dashboard.	Add new "Marketplace" tab in user dashboard. Display lists of user preorders, waitlist items, and orders.	API endpoints: `/api/users/{id}/marketplace-activity` returns preorders, notifications, purchases. Data persisted in `user_activity` table.
Royal Mail Address Verification + Delivery Tracking	During checkout, users must enter a verified UK address using Royal Mail API. Delivery tracking also linked.	Checkout form uses address autocomplete from Royal Mail API. Show tracking ID in user dashboard after purchase.	Integrate Royal Mail API: validate address before order submission, create shipping entry with `tracking_id`. Endpoint: `/api/orders/shipping`.
Shopping Cart Page	Dedicated cart page accessible from top-right cart icon. (Refer to Abiola's Figma design).	Cart icon redirects to `/cart` page. Display item summary, quantity, subtotal, discount if active.	Maintain user cart in DB or session. `/api/cart` for CRUD operations. Support persistent carts (tied to user_id).
Discounts & Promotions	Admin can set global or product-specific discounts.	Display discounted price dynamically if discount active.	Admin portal field `discount_percentage` per product. On save, backend recalculates `discounted_price`.
Product Reviews	Users can submit and view reviews per product. Admin can manage and filter reviews.	Review form (rating, comment). Display reviews list.	`/api/products/{id}/reviews` GET/POST/DELETE. Admin dashboard shows moderation view with filters by date, rating, keyword.
User Deletion & Data Retention	Admin can permanently delete test or inactive users.	Admin UI: "Delete User" action (confirmation modal).	Implement soft delete (`is_active=false`) and hard delete option. Retain purchase history in soft-delete mode for reactivation.
User Reactivation & Data Persistence	If user is reactivated, all history remains intact.	On reactivation, frontend re-fetches all previous data from DB.	Keep user-related records in linked tables (`user_id` foreign key). Ensure soft delete only affects authentication layer, not data tables.

2. Database Schema (Simplified Overview)

Table Name	Key Fields	Notes
`products`	`id`, `name`, `price`, `discount`, `stock`, `status`, `image_url`, `preorder_count`, `category_id`	Stores all product metadata.
`preorders`	`id`, `user_id`, `product_id`, `timestamp`	Tracks all preorders per user and product.
`notifications`	`id`, `user_id`, `product_id`, `type ('restock', 'new_product')`, `is_sent`	Tracks which users should be emailed upon restock.
`user_activity`	`id`, `user_id`, `action`, `product_id`, `timestamp`	Logs all marketplace actions for user dashboards.
`reviews`	`id`, `user_id`, `product_id`, `rating`, `comment`, `created_at`, `is_visible`	Product review system.
`orders`	`id`, `user_id`, `product_id`, `quantity`, `total_price`, `status`, `tracking_id`, `address_id`	Handles order and tracking.
`addresses`	`id`, `user_id`, `address_line1`, `city`, `postcode`, `royal_mail_verified`	Linked to Royal Mail API validation.
`discounts`	`id`, `product_id`, `percentage`, `is_active`, `start_date`, `end_date`	For promotions.

3. Implementation Plan (Step-by-Step)

Phase	Task Group	Responsible	Details / Tools
1. Setup & Planning	Confirm DB schema and API routes	Backend	Finalize endpoints, define relationships, and integrate ORM (Prisma / Sequelize).
2. Product Listing & Preorders	Build core product grid & preorder logic	Frontend + Backend	Backend: `/api/products`, `/api/preorders`. Frontend: React/Vue grid, add Preorder button states.
3. Notifications System	Integrate Maileroo for restock alerts	Backend	On product stock update → trigger email batch to users in `notifications`.
4. Dashboard Integration	Link user marketplace actions to dashboard	Frontend + Backend	Create dashboard API endpoint to retrieve marketplace activity.
5. Checkout & Royal Mail Integration	Integrate address validation and shipping tracking	Backend	Use Royal Mail API for UK addresses. Store verified addresses.
6. Cart Page Implementation	Follow Abiola's design	Frontend	Cart page with editable quantities, discount line, total, checkout button.
7. Discounts & Admin Controls	Add discount management to Admin Dashboard	Frontend + Backend	Backend: `/api/admin/discounts`. Frontend: Admin form + dynamic price updates.
8. Reviews & Moderation	Build review system	Frontend + Backend	Backend: `/api/products/{id}/reviews`. Frontend: star ratings + comment UI.

9. Data Deletion & User Reactivation	Clean up admin data management	Backend	Implement soft delete and reactivation logic.
10. Testing & QA	End-to-end flow tests	Full Team	Test preorder, notification, checkout, discount, and reactivation flows.

4. Additional Notes

- Maileroo API will handle **email notifications** for restocks and preorders.
- Royal Mail API is only triggered during checkout; validated addresses are cached for returning users.
- Admin dashboard will require a **new section: "Product Management"** to toggle stock, set discounts, and review preorders.
- User dashboard will gain a **"Marketplace"** sub-section that consolidates all marketplace-related user activities.