VR Integrated Furniture E commerce — Product Specification

1. Overview
- Purpose: Cross platform furniture marketplace (Expo/React Native + Web) with 3D/VR previews, cart, COD checkout, user listings, profile, environmental impact, and admin operations.
- Platforms: iOS, Android, Web (react-native-web).
- Backend: Supabase (Auth, Postgres, Storage, RLS).
- Personas: Visitor, Authenticated User, Admin (email-allowlisted, e.g., thammikt@gmail.com).

2. Roles & Access
- Visitor
  - Browse categories/products, view Product Detail and VR Preview.
- Authenticated User
  - Add to Cart, update/remove items, Checkout (Cash on Delivery), view Order Success.
  - Profile with environmental impact and "Your Orders".
  - Submit furniture Listing (with photos, phone, location).
- Admin
  - All user capabilities.
  - Admin Dashboard (stats and quick actions).
  - Order Management: view all orders, update status (pending → confirmed → shipped → delivered), cancel.
  - Listings Management: approve/reject user listings.

Admin detection: email allowlist in app; isAdmin true for thammikt@gmail.com.

3. Navigation (High Level)
- Bottom Tabs: HomeStack, CartStack, SellStack, ProfileStack.
- Stacks include: Home, ProductList, ProductDetail, VRView, Cart, Checkout, OrderSuccess, Profile, SellFurniture.
- Admin routes (from Profile Admin Panel): AdminDashboard, AdminOrderManagement, AdminListingsManagement.

4. Key Flows & Acceptance Criteria

4.1 Browse & Product Detail
- Home shows featured products and categories.
- Tapping a product opens ProductDetail with name, price, description, impact metrics (trees_saved, carbon_reduced), and actions.
- If product has sketchfab_id, "View in 3D/VR" opens VRView.
- Closing VRView replaces back to ProductDetail (no extra stack). Back from ProductDetail returns to Home.

4.2 Cart
- Add to Cart from Home/ProductDetail stores item in Supabase cart_items with quantity.
- Cart lists image, name, optional description, unit price, quantity, delete button.
- Delete prompts confirmation and removes item server-side; UI updates.
- Cart total = sum(price × quantity).

4.3 Checkout (COD) & Order Success
- Proceed to Checkout only when cart is non-empty.
- Required shipping fields: full name, phone, address, city, state, zip. Button disabled until valid.
- Placing order creates orders row (status=pending, shipping_address, total_amount) and order_items rows; clears cart.
- Navigate to OrderSuccess with order id and total.
- "Continue Shopping" goes to Home tab; "View Profile" opens Profile tab.

4.4 Profile & Orders
- Profile shows environmental impact and "Your Orders" (id short, date, total, items_count, status).

4.5 Sell Furniture (Listings)
- Users can create listing with title, photos (uploaded to Supabase Storage bucket listings), phone, location text/coords.
- Listing status defaults to pending for admin review.

4.6 Admin
- Dashboard: stats (total/pending orders, total/pending listings, total revenue), recent activity, quick actions.
- Order Management: list all orders with customer info; update status flow and cancel.
- Listings Management: list all submissions with images and user info; approve/reject.

5. Non Functional Requirements
- Performance: Home initial load < 3s on broadband; lists virtualized.
- Reliability: All write operations confirm success or show error toast.
- Security: RLS enforced (users only access own cart/orders), products readable by all; admin via application allowlist.
- Compatibility: Web via Expo web; native via Expo runtime.
- Accessibility: Buttons have labels/icons; status badges with sufficient contrast.

6. Data Model (Simplified)
- products(id, name, description, price numeric, image, category, sketchfab_id, trees_saved numeric, carbon_reduced numeric, created_at)
- user_profiles(id uuid PK → auth.users.id, full_name, email, avatar_url, total_trees_saved, total_carbon_reduced, total_waste_diverted)
- cart_items(id, user_id → user_profiles.id, product_id → products.id, quantity, created_at)
- orders(id, user_id, status text, shipping_address text, total_amount numeric, created_at)
- order_items(id, order_id → orders.id, product_id → products.id, quantity, price numeric)
- listings(id, user_id, title, description, condition, price, location_text, latitude, longitude, phone, image_urls[], status text, created_at)

7. External Integrations
- Sketchfab embed in VRView via iframe/WebView using sketchfab_id.
- Supabase Storage for listing images (bucket: listings).

8. Environment & Run
- Dev: Expo.
- Web command: expo start --web.
- Testsprite Local Port: 8081 (ensure http://localhost:8081 is reachable).

9. Test Account (for Testsprite)
- Username (email): thammikt@gmail.com
- Password: provide in Testsprite UI (not stored in repo).
- Admin: true (email allowlist).

10. Feature Checklist (For Automated Tests)
- Home renders categories and featured products.
- ProductDetail shows fields; VR button appears when sketchfab_id exists; VR close → ProductDetail; ProductDetail back → Home.
- Add to Cart works from Home/ProductDetail; success toast shown.
- Cart renders image, title, optional description, price, delete; total updates on add/remove.
- Checkout validates; creates orders + order_items; success page; Continue Shopping → Home.

- Profile shows impact and user orders for the logged-in user.
- Listing creation uploads images and creates pending listing.
- Admin Dashboard shows stats; Order Management filters and updates statuses; Listings Management approves/rejects.

11. Risks & Assumptions
- Supabase schema and RLS configured; USE_MOCK_DATA=false for E2E tests.
- Email allowlist for admin acceptable for current phase.