

## PRODUCT REQUIREMENTS DOCUMENT (PRD)

Product Name: ShopWizz.ai

Version: MVP v1.0

### Overview:

ShopWizz.ai is a responsive web application (desktop, laptop, tablet, mobile) with best-in-class mobile optimization, offering AI-powered personal shopping assistance primarily on Amazon. Users navigate to the site, upload images, enter text, or use voice prompts to describe desired products. Leveraging GPT-4 and ElevenLabs voice APIs, ShopWizz.ai returns ranked Amazon product links with embedded affiliate tags, AI-generated star ratings, and concise “Why buy” summaries. Customers filter results by price, brand, quality, reviews, shipping time, and pickup availability. Revenue is earned solely via Amazon Associates affiliate partnerships.

### Target User:

- Primary user: Online shoppers on desktop or mobile seeking personalized Amazon product discovery without switching tabs.
- Secondary users (future): Merchants, Support Agents, Super Admins.

### Goal of MVP:

- Success Criteria:
  1. Fully responsive web UI optimized for mobile, tablet, and desktop: chat widget and filters adapt to screen size.
  2. End-to-end conversational search (image/text/voice) → ranked Amazon results → affiliate links.
  3. Core filters functional and persisting per anonymous session (localStorage) and permanently for logged-in users (Supabase).
  4. Optional signup prompt to save search/check-out history via Supabase.
  5. Affiliate tagging via Amazon Associates on all returned links.
- Out of Scope:
  - Support for non-Amazon retailers.
  - In-app cart assembly or direct payment handling.

- Mobile apps, merchant dashboards, advanced admin features.
- Social sharing or loyalty programs.

## CORE FUNCTIONAL REQUIREMENTS (DO NOT SKIP)

### 1. Authentication & Session

- **Anonymous Default:** Users immediately access chat via text, voice, or image without signing up.
- **Optional Signup Prompt:** Under chat input: “Save your search history? [Sign up]” triggers Supabase Auth modal.
- Must include: email/password registration, email verification, forgot password. OAuth optional later.
- **Guardrail:** Signup optional; core features never blocked.

### 2. Responsive UI / Main Interface

- **Layout:** Single-page app built in Next.js with Tailwind CSS.
- **Chat Widget Placement:**
  - Desktop/tablet: fixed right-side panel (width: 400px).
  - Mobile: bottom sheet full-width.
- **Widget Elements:**
  - Text input “Ask anything.”
  - Image upload button.
  - Voice record button (ElevenLabs STT).
  - Recommended prompts below input.
  - Filters toggle collapsible section (price, brand, quality, reviews, shipping, pickup).
  - Signup prompt link.

- Voice Chat Toggle to open floating voice panel.
- **Adaptive Behavior:**
  - Collapse filters by default on screens <768px.
  - Ensure readable font sizes and tap targets ≥44px.
- **Load Data:** from localStorage for anonymous, Supabase for registered.

### 3. Feature Modules

#### 3.1 Chat & Input Module

- Inputs: textPrompt, voiceBlob, imageFile.
- Validation: ≥1 input; file <10 MB; text ≤500 chars.
- Output: searchQuery object.

#### 3.2 AI Processing Module

- Use OpenAI GPT-4 API to parse inputs into parsedQuery JSON.
- ElevenLabs STT for voice transcription and TTS for response audio.

#### 3.3 Amazon Product API Module

- Query Amazon Product Advertising API with parsedQuery + filters.
- Output: productItems[] with asin, title, imageUrl, price, brand, reviewCount, avgRating, affiliateUrl.
- Validation: exclude items missing price or image.

#### 3.4 Results Display Module

- Render vertical list of cards:
  - Image (200×200), title (≤60 chars), AI star rating + “Why buy” (≤100 chars), price, “Buy on Amazon” button → affiliateUrl.
- Pagination/infinite scroll: 10 items per load.
- For mobile, use swipe gestures for card navigation.

### 3.5 Voice Chat Module

- **Voice Input:** record via widget mic; send to ElevenLabs STT.
- **Voice Output:** render response audio via ElevenLabs TTS in floating panel.
- UI Controls: play/pause, mic mute, close panel.
- **Guardrail:** respect device mute settings.

### 3.6 Revenue & Monetization Module

- **Amazon Associates:**

- Append tag=<associateTag> parameter to affiliateUrl.
- Comply with Associates Program TOS (link format, disclosure banner).

## 4. Settings / Configurations

- Filters: price range, brandList, minRating, minReviewCount, maxShippingDays, pickupAvailability.
- Voice Chat settings: STT/TTS toggle, playback volume.
- Save preferences: anonymous → localStorage; registered → Supabase on Save.
- **Guardrail:** explicit Save click required.

## 5. Admin Panel (future stub)

- Roles: user, admin, super\_admin.
- Planned features: user management, associateTag management, analytics dashboard.

## NON-FUNCTIONAL REQUIREMENTS

- Security:
  - OWASP Top 10 compliance; HTTPS for API calls.
  - Input sanitization to prevent XSS.

- JWT via Supabase for registered users; localStorage for anonymous.
  - Performance:
- Widget and results load <2s on 4G.
- Debounce API calls (500ms).
  - Responsiveness:
- Tailwind CSS breakpoints: mobile (<640px), tablet (641–1024px), desktop (>1025px).
- Ensure WCAG 2.1 AA contrast and touch target sizes.
  - Data Storage:
- localStorage for anonymous filters and searchHistory.
- Supabase PostgreSQL: users table (UUID PK, email, password\_hash, role, preferences JSON, searchHistory JSON, timestamps).

#### GUARDRAILS FOR AI/DEVS (STRICT)

- Do not change variable/field names.
- Do not assume missing logic; clarify before coding.
- No placeholders unless marked.
- Do not alter UI/UX flows without approval.
- Follow component naming and order exactly.
- Keep implementation modular and DRY.

#### TOOLS + STACK

- Frontend: Next.js (React) + Tailwind CSS
- Backend: Supabase (Auth, Postgres)
- AI: OpenAI GPT-4 API; ElevenLabs STT/TTS
- Amazon API: Amazon Product Advertising API

- Hosting: Vercel for frontend; Supabase for backend services

*End of PRD*