Product Requirements Document (Technical / Agent-Ready)

Project Name: PC Part Sniper
Version: 1.2 (Revised and Expanded)
Scope: MVP for Incubator Project — A next-generation PC building and part comparison platform inspired by PCPartPicker, with a more fluid design, real-time affiliate pricing, and improved component visualization.

0. Product Summary

PC Part Sniper is a web platform that helps PC builders search, compare, and assemble computer parts. It combines an AI recommendation system with a compatibility checker to simplify the building process for beginners and enthusiasts. The MVP focuses on part search, build creation, and AI-based compatibility evaluation.

---

1. Website Architecture

The site follows a structure similar to PCPartPicker — a modular, component-based system where users can browse PC parts, create custom builds, check compatibility, and purchase parts through affiliate links.
The difference: **PC Part Sniper** uses smoother transitions, a cleaner, modern CSS design, and a more immersive build creator experience.

1.1 Home Page — `index.php`

Purpose:
The Home page introduces users to the platform, highlights trending builds, and provides a fast entry point to the parts database. It acts as the “dashboard” equivalent of PCPartPicker’s landing page — but with a more visual layout and fluid animations.

Buttons / Functions:

- **Search** → takes input → redirects to `parts.php?query=...`.

- **View Featured Build** → queries `builds` table for sample builds → opens `build.php?build_id=...`.

- **Affiliate Ad Click** → logs event in `click_tracking` → redirects user to external store.

Data Flow:

  - `User enters query → search_controller.php executes SQL SELECT on parts table`
  - `Featured build section → SELECT * FROM builds JOIN build_parts`
  - `Affiliate click → INSERT INTO click_tracking (user_id, part_id, merchant_id)`

MVP Data Ops:

- READ: Featured builds, trending parts.

- WRITE: Log affiliate link clicks.

User Flow:
User lands on home page → scrolls through featured builds (similar to PCPartPicker's "Trending Builds" section) → searches for a part or component type → redirected to the parts listing page.

1.2 Part Listings Page — `parts.php`

Purpose:
Displays PC components filtered by category, brand, price, and other specs. This mirrors PCPartPicker's parts list system, but with real-time filtering and smoother UI transitions.

Buttons / Functions:

- **Category Selector** → loads relevant part types (CPU, GPU, etc.).

- **Filter Dropdowns** → applies SQL `WHERE` conditions (brand, price, form factor).

- **Used/New Toggle** → toggles `is_used` field.

- **Select Part** → redirects to `details.php?part_id=...`.

Data Flow:

  - `Filter selection → AJAX request to /api/get_parts`
  - `Backend → SELECT * FROM parts JOIN part_prices JOIN merchants`
  - `Front-end → renders cards with price and merchant info`
  - `Price Graph → requests /api/get_price_history (Chart.js)`

MVP Data Ops:

- READ: Filtered part listings + merchant pricing.
- JOIN: `parts`, `part_prices`, `merchants`.

User Flow:
User browses or filters parts by category (e.g. “NVIDIA GPUs”) → sees prices from multiple merchants like Amazon or Newegg (affiliate linked) → clicks one part to view its details.
The browsing experience is **nearly identical to PCPartPicker’s “Parts List” page**, but with live updates instead of reloads.

1.3 Part Detail Page — `details.php`

Purpose:
Displays in-depth technical specifications, historical pricing, reviews, and affiliate purchase links. It’s equivalent to PCPartPicker’s “Product Page,” but with integrated compatibility data previews.

Buttons / Functions:

- **Add to Build** → INSERT INTO `build_parts`.
- **Write Review** → INSERT INTO `reviews`.
- **Buy from Merchant** → redirects to `part_prices.url` (affiliate tracking).

Data Flow:

- `On load → SELECT * FROM parts WHERE part_id = ?`
- `Fetch merchant pricing → SELECT * FROM part_prices WHERE part_id = ?`
- `User adds review → INSERT INTO reviews`
- `User clicks buy → INSERT INTO click_tracking + redirect to affiliate URL`

MVP Data Ops:

- READ: Specs, reviews, and merchant links.
- WRITE: Reviews, build additions, click tracking.

User Flow:
User clicks a part → reads specs and comparisons → sees compatible parts preview → adds part to build or buys via affiliate link.
Compared to PCPartPicker, this page adds smoother animations and compatibility highlights (like "✓ Compatible with your current motherboard").

1.4 Build Creator — `build.php`

Purpose:

This is the centerpiece — the "Build Your PC" interface, equivalent to PCPartPicker's builder. Users assemble parts, check compatibility, and save builds.
PC Part Sniper improves on PCPartPicker by offering a **more dynamic compatibility engine**, **instant price summaries**, and **fluid UI transitions** between categories.

Buttons / Functions:

- **Add Part** → inserts into `build_parts`.
- **Compatibility Check** → runs PHP validation rules comparing sockets, form factor, and wattage.
- **Save Build** → inserts/updates `builds` table.
- **Share Build** → generates shareable URL.

Data Flow:

- `Add Part → INSERT build_parts`
- `Check Compatibility → SELECT * FROM parts WHERE build_id = ?`
- `Compare (CPU.socket = MB.socket, PSU.wattage ≥ sum(TDPs), etc.)`
- `Save Build → INSERT builds (user_id, build_name)`

User Flow:
User opens build creator → adds CPU, motherboard, GPU, etc. → system checks compatibility in real time (like PCPartPicker's green checkmarks) → total price updates dynamically → user saves or shares the build.

1.5 User Profile — `profile.php`

Purpose:
Acts as the user's personal hub — stores all saved builds, reviews, and settings. Similar in spirit to PCPartPicker's "My Builds" dashboard, but with a cleaner layout and responsive cards.

Buttons / Functions:

- **Edit Profile** → UPDATE users.
- **View Build** → redirects to `build.php?build_id=...`.
- **Delete Build** → DELETE FROM builds.
- **Review History** → SELECT FROM reviews.

Data Flow:

- `SELECT builds WHERE user_id = ?`
- `JOIN build_parts → show all parts per build`
- `UPDATE users on profile edit`
- `DELETE builds on user command`

User Flow:
User visits their profile → manages builds → reviews or edits previous setups → clicks to reopen a build for modification or sharing.

1.6 Checkout — `checkout.php`

Purpose:
Aggregates all selected build parts and redirects users to affiliate merchant pages for each item. Functions like PCPartPicker's "Buy This Build" page, with seamless redirection.

Buttons / Functions:

- **Buy Part** → redirect to affiliate link for individual part.
- **Buy Full Build** → open all merchant links in new tabs (or list them).

Data Flow:

- `SELECT build_parts JOIN part_prices JOIN merchants`
- `Render list with affiliate URLs`
- `On click → INSERT click_tracking + redirect`

User Flow:
User finalizes build → reviews merchant list and total cost → clicks "Buy Now" → redirected through affiliate links.

1.7 Contact — `contact.php`

Purpose:
Customer support and general inquiry form.

Buttons / Functions:

- **Submit Ticket** → INSERT INTO support_tickets.
- **View FAQ** → expand static answers.

Data Flow:

- `POST contact form → INSERT support_tickets`

User Flow:
User submits support request → receives confirmation → admin later reviews tickets.

1.8 Chatbot — `chatbot.js`

Purpose:
AI helper for navigation and basic Q&A about compatibility (no monetization).
Mirrors PCPartPicker's compatibility guide but automated via chat UI.

Buttons / Functions:

- **Ask Question** → queries backend for part data.
- **Suggest Build** → returns compatible parts via backend logic.

Data Flow:

- `Frontend JS → /api/chatbot → queries parts DB`
- `Returns compatible suggestions`

---

2. Backend Data Schema

Includes all tables from prior version, plus:

- `click_tracking` for affiliate analytics
- `price_history` for trend visualization
- Removed token/ad system.

(Schema identical to prior version with the additions above.)

---

3. Expanded User Flow (End-to-End)

Overview:

PC Part Sniper mirrors the **user journey of PCPartPicker**, maintaining familiarity while modernizing the interaction and visual design.

1. **Discover (Home → Search → Listings)**
    - User lands on homepage → enters keyword (e.g., “RTX 5090”).
    - System performs SQL query → returns matching components.
    - User filters and sorts results.
2. **Research (Listings → Details)**
    - User clicks a part → reads detailed specs and compatibility info.
    - Sees pricing from multiple merchants with affiliate links.
    - May read or write reviews.
3. **Build (Details → Build Creator)**
    - User adds part to a new or existing build.
    - System validates compatibility dynamically.
    - Price totals update automatically.
    - User saves or shares the build publicly.
4. **Purchase (Build → Checkout)**
    - Checkout page lists all parts with current prices and affiliate merchants.
    - User clicks merchant links → redirected through affiliate URLs (tracked in `click_tracking`).

    - Monetization occurs through affiliate partnerships (Amazon, Newegg, Micro Center).

5. **Manage (Profile)**
    - User reviews saved builds.
    - Edits or deletes configurations.
    - Can reaccess merchant links anytime.
6. **Support / Guidance (Chatbot, Contact)**
    - Chatbot assists with part compatibility or navigation.
    - Contact form provides manual support option.

Data Lifecycle Summary:

- `User Action → Controller Logic → SQL Query → Render View → (Optional) Affiliate Redirect`

Affiliate Tracking Flow:

- `User clicks merchant link → INSERT INTO click_tracking (user_id, part_id, merchant_id, timestamp)`
- `Merchant processes referral → revenue logged externally`

---

4. Design Specification

Layout:

- Modular, grid-based, similar to PCPartPicker for familiarity.
- Smooth transitions between filters and build updates.

Color Scheme:

- **Primary:** Deep charcoal `#1E1E1E`
- **Accent:** Electric blue `#0088FF`
- **Secondary:** Cool gray `#2A2A2A`
- **Highlight:** Neon green `#3BFF7F` (for “Compatible”)

Visual Differentiation:

- Glassmorphism-inspired transparency on build cards.
- Animated hover states and dynamic search bar.
- Simplified icons and high-contrast readability for dark mode.