**Note for Future Updates: This document is the primary source of truth for the project's state, designed to be consumed by AI assistants. It must be as technical and detailed as possible. NEVER edit or remove previous entries. All new updates must be appended to the bottom with a new timestamp to maintain a complete, chronological history log.**

**Project 'Market Pulse' - Changelog**

***Initial document generated: Saturday, July 5, 2025***

**1. Core Project Specification**

* **Objective: A web application to provide hotel performance metrics, including a live dashboard and a "vs. The Market" comparison tool.**
* **Frontend: public/index.html and public/script.js. Vanilla JavaScript.**
* **Backend: server.js (Node.js/Express). Serves the frontend and provides a JSON API.**
* **Database: Neon PostgreSQL.**
* **Deployment: Vercel, connected to the main branch of the GitHub repository.**
* **Sensitive Data: All secrets (API keys, database URLs) are managed via a local .env file and as Environment Variables in the Vercel project settings.**

**Entry: Saturday, July 5, 2025 - Morning Session**

* **Objective: Migrate the application from a local-only setup to a fully operational, cloud-native solution on Vercel.**
* **Key Files Modified: server.js, vercel.json, daily-refresh.js.**
* **Backend Changes:**
  + **server.js: Logic updated to use process.env.DATABASE\_URL for the production database connection.**
  + **daily-refresh.js: Converted to a Vercel Serverless Function format using import / export and a default handler function.**
* **Deployment & Configuration:**
  + **vercel.json: Created to define build outputs and configure a cron job.**
  + **Cron Job: path: "/api/daily-refresh", schedule: "0 1 \* \* \*" to run daily-refresh.js daily at 1 AM UTC.**

**Entry: Saturday, July 5, 2025 - Afternoon Session (Data Seeding)**

* **Objective: Solve the "cold start" problem for the "Us vs. The Market" feature by creating and importing a rich set of mock competitor data.**
* **Key Files Modified: seed-market-data.js (new file).**
* **Database Schema Changes:**
  + **hotels table created to store static hotel information (name, city, coordinates).**
  + **Foreign key constraint added between daily\_metrics\_snapshots.hotel\_id and hotels.hotel\_id.**
* **Backend Changes:**
  + **seed-market-data.js: New Node.js script created to parse daily\_metrics\_snapshots\_5hotels.csv and perform a batch INSERT ... ON CONFLICT operation into the daily\_metrics\_snapshots table.**
* **Data Artifacts:**
  + **daily\_metrics\_snapshots\_5hotels.csv: CSV file created with 3,650 rows of mock data for 5 competitor hotels.**

**Entry: Saturday, July 5, 2025 - 2:40 PM CEST (UI & Backend Aggregation)**

* **Objective: Transform the raw competitor data into a true "like-for-like" comparison view and improve the UI for direct comparison.**
* **Key Files Modified: index.html, script.js, server.js.**
* **Backend Changes (server.js):**
  + **/api/competitor-metrics: The SQL query in this endpoint was fundamentally changed. It now uses AVG(), SUM(), and GROUP BY stay\_date to perform data aggregation directly in the database. This ensures the API returns a single, averaged "market" row per day.**
* **Frontend Changes:**
  + **index.html: Restructured to use a two-column div layout (<div class="comparison-container">) to place "Your Hotel" and "Competitor Market" tables side-by-side.**
  + **script.js: The renderCompetitorMetricsTable function was simplified. It no longer needs to handle multiple rows per day or a hotel\_id column. Headers were updated to "Market ADR", etc.**

**Entry: Saturday, July 5, 2025 - 3:15 PM CEST (Granularity Feature)**

* **Objective: Implement a feature allowing users to view the comparison data with daily, weekly, or monthly granularity.**
* **Key Files Modified: index.html, script.js, server.js.**
* **Backend Changes (server.js):**
  + **/api/metrics-from-db & /api/competitor-metrics: Both endpoints were enhanced to accept a granularity query parameter.**
  + **A logic block was added to sanitize the input (e.g., 'weekly' -> 'week') and dynamically build the SQL query.**
  + **For non-daily views, the SQL queries now use DATE\_TRUNC('<granularity>', stay\_date) and GROUP BY DATE\_TRUNC(...) to perform the aggregation at the requested time interval.**
* **Frontend Changes:**
  + **index.html:**
    - **Added a div with class toggle-group containing three buttons: [ Daily ] [ Weekly ] [ Monthly ].**
    - **Changed the Number of Days input to an End Date picker (<input type="date" id="master-end-date">).**
  + **script.js:**
    - **Added a global variable currentGranularity to hold the state.**
    - **Added setGranularity(granularity) function to handle button clicks and update the state.**
    - **loadAllDbData() function updated to read from the new master-end-date input and pass the currentGranularity to the API calls.**
    - **Table rendering functions updated to change the date column header based on the selected granularity.**

**Entry: Saturday, July 5, 2025 - 3:35 PM CEST (UX & Cosmetic Refinements)**

* **Objective: Polish the user interface and improve the overall user experience.**
* **Key Files Modified: index.html, script.js.**
* **Frontend Changes:**
  + **index.html:**
    - **The CSS <style> block was completely overhauled with a new, more modern design system (CSS variables, refined color palette, improved typography and spacing) for a "dashboard-like" feel.**
    - **The "Live API Forecast" section was moved above the "Master Controls" box.**
    - **The container for property information was changed to an empty div with a button: <div id="hotel-info-container"><button onclick="fetchHotelDetails()">...</button>...</div>.**
  + **script.js:**
    - **The DOMContentLoaded event listener that automatically called fetchHotelDetails() was removed. The function is now only triggered by the new button's onclick event.**
    - **A new block was added to the DOMContentLoaded listener to programmatically set the master-start-date to today and master-end-date to one month in the future.**
    - **The DATASET\_7\_MAP constant was updated to rename the occupancy metric's name property from "Occupancy (Direct)" to "Occupancy".**

**Entry: Saturday, July 5, 2025 - 8:27 PM CEST (Vercel Deployment Fix)**

* **Objective: Diagnose and resolve a critical deployment failure on Vercel where the application was non-functional despite working locally.**
* **Problem Summary:**
  + **The deployed application on Vercel was serving a 404: NOT\_FOUND error for the root page and all static assets (e.g., script.js, dashboard.html).**
  + **This resulted in a non-interactive page where JavaScript-driven features, such as the date pickers and data loading buttons, did not work.**
  + **The root cause was a vercel.json configuration that did not correctly instruct the Vercel build system on how to handle the static frontend files.**
* **Debugging & Resolution:**
  + **Initial analysis of the vercel.json file suggested the routing rules were incorrectly sending all traffic to the server.js backend, which does not have access to the static files in a serverless environment.**
  + **After an initial attempt to fix the routes failed, a detailed review of the Vercel Build Logs revealed the true issue: Vercel was incorrectly compiling frontend files (like script.js) as if they were backend functions, instead of deploying them as static content. The key log line was Compiling "script.js" from ESM to CommonJS....**
  + **The final, successful fix involved adding an explicit build rule to vercel.json: { "src": "public/\*\*", "use": "@vercel/static" }. This rule forces the Vercel build system to recognize the /public directory's contents as static assets and deploy them correctly, resolving the 404 errors.**
* **Key Files Modified: vercel.json**

**Entry: Saturday, July 5, 2025 - 8:56 PM CEST (Feature Enhancements)**

* **Objective: Improve performance and add more contextual information to the dashboard.**
* **Key Files Modified: server.js, index.html, script.js.**
* **Feature 1: Performance Optimization (DB-first Hotel Details)**
  + **Backend: A new endpoint, /api/hotel-details-from-db, was created in server.js to query the hotels table directly.**
  + **Frontend: The script.js was updated to call this new endpoint on page load (DOMContentLoaded), displaying property info instantly. The previous live API call was kept behind a manual "Refresh" button. The fetchHotelDetailsFromDB function was made dynamic to render all non-null fields returned from the API, preventing missing data.**
* **Feature 2: Competitor Set Context**
  + **Backend: The /api/competitor-metrics endpoint in server.js was enhanced to perform two new calculations:**
    1. **A COUNT query to get the number of competitor hotels.**
    2. **A SUM query on the most recent capacity\_count for each competitor to get the total room inventory.**
  + **The API response was changed to an object: { metrics: [], competitorCount: X, totalCapacity: Y }.**
  + **Frontend: A new placeholder div was added to index.html. The loadCompetitorMetrics function in script.js was updated to parse the new response object and display the text, e.g., "Based on a competitive set of 5 hotels, with a total of 500 rooms."**