Project Specification: Campaign Flux - Marketing CRM (v7.0)

1. Project Overview

The objective is to build "Campaign Flux," a high-performance Marketing Customer Relationship Management (CRM) tool. This platform enables marketing teams to manage massive lead databases, track real-time campaign performance, and orchestrate complex multi-channel marketing flows. The system must remain fully functional in offline environments (capturing leads at events), process heavy analytics in the background, and maintain a seamless 60fps UI despite high-frequency data updates from global marketing channels.

2. Core Concepts to Demonstrate

- **The Browser Event Loop:** Managing high-frequency UI updates without blocking the main thread.
- **Storage Tiers:** LocalStorage (Layouts), SessionStorage (Active Campaign Drafts), and IndexedDB (Offline Lead Repository).
- **Advanced Concurrency:** Web Workers for calculating "Lead Scoring" and "Sentiment Analysis" on bulk data.
- **Data Structures:** `Map` for Lead Segmenting and `Set` for unique categorical Interest Tagging.
- **Communication Matrix:** WebSockets (Live Interaction Feed), SSE (Campaign ROI Metrics), Short Polling (Service Health), and Long Polling (Third-Party Export Handshakes).
- **Idempotency:** Ensuring bulk lead uploads or status changes (e.g., "Unsubscribe") are not duplicated during network retries.
- **DOM Mechanics:** Event delegation for high-density lead grids and propagation control for nested action menus.
- **Memory & Logic:** `WeakMap`/`WeakSet` for DOM metadata and sync tracking; `try/catch/finally` for safe binary uploads.
- **DevTools & Memory Profiling:** Identifying memory leaks, analyzing Heap Snapshots, and observing Garbage Collection (GC) behavior.

3. Technical Deliverables

D1: The Persistence & Offline Reliability Layer

- **Requirement 1 (IndexedDB - The Lead Vault):** Store the full database of leads and campaign history.
    - **The Offline Queue & Idempotency:** If a user captures a new lead or updates a status while offline, queue the request.
    - **Constraint:** Each request must carry a unique **Idempotency Key**. Upon reconnection, the sync logic must ensure that a "New Lead" or "Unsubscribe" action isn't processed twice if the network times out during transmission.

- **Requirement 2 (LocalStorage - Workspace Persistence):** Store the user's dashboard configuration (e.g., widget placement and dark mode).
- **Requirement 3 (SessionStorage - Campaign Context):** Store the state of an "Active Campaign Draft" so the user can navigate to other parts of the CRM without losing their work.

D2: Real-Time Data Management (Maps & Sets)

- **Requirement 4 (The Segmenting Map):** Use a `Map` to track lead distribution across different segments (`segmentId => { count: 1500, avgScore: 82 }`). Update this Map dynamically to power real-time dashboard visualizations.
- **Requirement 5 (The Interest Set):** Use a `Set` to track unique "Interest Tags" encountered across a batch of leads (e.g., "SaaS", "Enterprise", "B2B"). This set will power a "Smart Filter" system.

D3: The High-Availability Communication Matrix

- **Requirement 6 (WebSockets - Live Interaction Feed):** A bi-directional stream showing every live customer interaction (e.g., "User X opened Email Y"). Include heartbeat logic to detect dead connections.
- **Requirement 7 (SSE - Global ROI Feed):** A read-only stream showing live conversion metrics and ROI across all active global campaigns.
- **Requirement 8 (Long Polling - Data Export Handshake):** Initiate a long-hanging request when a user attempts to export a large dataset to a third-party tool (e.g., Salesforce). The request waits for an "Export Complete" token.
- **Requirement 9 (Short Polling - System Health):** A 5-second check using `fetch` and `AbortController` to monitor the API health. Handle `429` rate-limits with exponential backoff.

D4: Binary Assets & Background Computing

- **Requirement 10 (Multipart Asset Upload):** Implement a feature to upload marketing assets (banners/logos) or CSV lead lists using `multipart/form-data`. Generate a local preview for images via `URL.createObjectURL`.
- **Requirement 11 (Web Worker - Lead Scoring Engine):** Implement a "Score Leads" button. This processes a dataset of 10,000+ leads to calculate priority based on historical behavior. This **must** run in a Web Worker to keep the UI fluid.

D5: DevTools & Memory Profiling Lab

- **Requirement 12 (Leak Detection):** Include a "Stress Test" mode that generates 1,000 temporary lead rows. Trainees must ensure these are properly garbage collected when the view is cleared.
- **Requirement 13 (Heap Analysis):** Use the **Memory Tab** to take Heap Snapshots before and after the "Stress Test." Identify any detached DOM nodes or uncleaned closures.
- **Requirement 14 (Allocation Instrumentation):** Use the "Allocation instrumentation on timeline" to observe memory spikes during high-frequency WebSocket interaction updates.

- **Requirement 15 (GC Observation):** Explicitly demonstrate how the `WeakMap` allows the GC to reclaim memory once lead rows are removed from the DOM.

D6: DOM Architecture & Memory Management

- **Requirement 16 (Event Delegation):** A single event listener on the main Lead Table for all actions (Edit, Delete, Score, Tag).
- **Requirement 17 (Propagation Control):** Inside a Lead Row, there is a "Quick Action" dropdown. Clicking inside the dropdown should not trigger the "Select Lead" event of the row (use `stopPropagation`).
- **Requirement 18 (WeakMap/WeakSet Usage):**
    - Use a `WeakMap` to store temporary "Interaction Tooltips" associated with specific Lead DOM elements.
    - Use a `WeakSet` to track "Selected" leads for bulk actions.

4. UI/UX Specification

Layout Structure

A high-density "Marketing Command Center" dark theme (`bg-slate-950`).

1. Header (The Control Hub)

- **Connectivity LEDs:** Status indicators for WS, SSE, LP, and SP protocols.
- **Sync Badge:** Counter for offline leads pending in the IndexedDB queue.
- **Diagnostic Button:** Triggers the console log sequence of Macrotasks vs Microtasks.

2. The Marketing Dashboard

- **Lead Grid:** High-density list using event delegation. Each row shows lead score and status.
- **Segment Visualizer:** Real-time data visualization powered by the `Map` data structure.
- **Memory Visualizer:** A small sparkline showing real-time JS Heap usage trends.

3. Side Panels

- **Live Feed:** A scrolling terminal showing WebSocket-driven interaction events.
- **Worker Status:** A progress indicator for the Lead Scoring Web Worker.

5. Submission Requirements

1. **The Code:** A single-file implementation.
2. **The Narrative:** * Explain the **Idempotency Strategy** used for lead capture.
    - Describe the **Memory Audit**: Detail how DevTools confirmed no memory leaks exist during row deletion.
3. **The Audit (Screenshots Required):**
    - **Memory Tab:** Before/After Heap Snapshots during the "Stress Test."
    - **Network Tab:** Evidence of binary `multipart` requests and the long-poll "Pending" state.

- **Console:** The output of the Event Loop diagnostic explaining execution priority.