**Progress We Had Made (Before the "Blank Map" Regressed):**

1. **Core Map Functionality Solid:**
   * **Map Initialization:** We had the map initializing correctly (Effect #1 using mapContainerReady state).
   * **Markers:**
     + Loading from Firestore was working.
     + Displaying on the map was working.
     + Adding new markers via UI was working (saving to state, saving to Firestore, displaying).
     + Show/Hide markers was working.
   * **Paths:**
     + Loading from Firestore was working (with the corrected coordinate transformation).
     + Displaying saved paths was working.
     + Drawing new paths (and finishing with double-click) was working.
     + Saving new paths to React state and Firestore was working.
     + Show/Hide paths was working.
   * **Building Conditions (Admin):**
     + Loading from Firestore was working (assuming correct admin auth for rules).
     + Displaying conditions on buildings (color-coding) was working.
     + Selecting a building and changing its condition was working.
   * **UI Overlays (Logos, Basic Panels):** These were rendering.
2. **Firestore Security Rules:**
   * We drafted and (presumably) deployed a set of security rules to differentiate between public and admin access for markers, paths, and buildingConditions. This is a crucial security step.
3. **Cloud Function for Server-Side Validation (Paths):**
   * You successfully set up Firebase Functions in your project.
   * You upgraded your project to the Blaze plan.
   * You enabled all necessary APIs (Cloud Functions, Cloud Build, Artifact Registry, Eventarc, etc.).
   * You successfully deployed the validatePathWritten (V2 Firestore trigger) Cloud Function. This function provides robust server-side validation for newly created paths.

**Where We Stumbled (Leading to the Current "Blank Map"):**

The goal was to implement "User Click Tracking" by:

1. Creating a saveMapClick useCallback hook to write click data to a new mapClicks Firestore collection.
2. Modifying the existing map click useEffect (Effect #6) to call saveMapClick under appropriate conditions.

The "blank map" issue and the subsequent ReferenceErrors (saveMapClick is not defined, exportToCSV is not defined, etc.) arose primarily from **incorrect placement and ordering of these new/modified hooks within your large StakeholderMap.js component.** Hooks have strict rules about where they can be called (top-level only, not inside other hooks, not conditionally). When these rules are broken, React's internal mechanism for managing hooks fails, leading to unpredictable behavior, including refs not being populated or functions appearing as "not defined."

**The State Before We Started "User Click Tracking":**  
If we could revert your StakeholderMap.js to the state it was in right *before* we started adding the saveMapClick function and modifying the map click useEffect, your map and all its features (markers, paths, UI overlays) were likely working correctly.

**Why the "Minimal Test" Approach Was Suggested:**  
When a component gets into a state where basic refs like mapContainerRef.current are not being populated as expected, it often indicates a fundamental issue with React's rendering or hook execution order for that component. The "minimal test" was an attempt to strip away all potentially interfering logic to confirm that the absolute core (rendering a div, attaching a ref, and initializing Mapbox in a basic useEffect) works. If even that fails, the problem is very deep. However, given your previous success, the issue is almost certainly the recent structural changes we made for click tracking.

**Do We Need to Start a New Chat?**  
Not necessarily, unless you feel this one has become too cluttered. I have the context. The main thing is to get your StakeholderMap.js back to a working state for map display, and then very carefully re-introduce the click tracking.

**Recommended Path Forward (Reset and Re-Integrate Carefully):**

1. **Goal 1: Get the Map Displaying Again (Rollback Recent Changes to Click Logic).**
   * **Option A (If you use Git):** If you have your code in Git, can you revert StakeholderMap.js to the commit *before* we started adding saveMapClick and modifying Effect #6 (the map click handler)? This would be the cleanest way to get back to a known working state.
   * **Option B (Manual "Undo"):**
     + **Remove** the const [mapContainerReady, setMapContainerReady] = useState(false); line.
     + **Remove** the useEffect block that you added specifically to set mapContainerReady.
     + **Revert "Effect 1: Initialize Map"** to its simpler form that *doesn't* depend on or check mapContainerReady. It should primarily check if (!mapContainerRef.current || mapRef.current) return;. Its dependency array should just be [config].