Listing Wizard Redesign Requirements Document

Current Flow

1. User visits `/properties/list`
2. User selects Property Type and Listing Type
3. User clicks "Create Your Listing" button
4. System navigates directly to the wizard

Proposed Changes

Core Requirements

1. **Database Storage for Flow Tracking**
   - Create a temporary table (`temp_property_listing`)
   - Required fields:
     - `id` (UUID, primary key)
     - `owner_id` (UUID, foreign key to profiles)
     - `created_at` (timestamp with time zone)
     - `updated_at` (timestamp with time zone)
     - `property_details` (JSONB)
     - `status` (text, default 'draft')
2. **Flow Initialization**
   - When user selects Property Type and Listing Type and clicks "Create Your Listing":
     - Create an entry in the temporary table
     - Record the flow type (e.g., residential_rent)
     - Store the flow template
     - Generate a unique flow ID
     - Set initial status to 'draft'
3. **URL-based Flow Identification**
   - Include the unique flow ID in the wizard URL
   - Example: `/properties/wizard/{unique_flow_id}`
4. **Database-driven Flow Detection**
   - Use only the database value to determine flow type
   - Eliminate all client-side flow detection logic

    - Query the database using the flow ID from the URL

5. **Progressive Data Storage**
    - Save data to the temporary table at each step
    - Update the database when user clicks "Next"
    - Persist partial progress in case of interruptions
6. **Final Submission Process**
    - At the review step, maintain existing functionality
    - On final submission:
        - Copy data from temporary table to `public.properties_v2`
        - Change status in temporary table from 'draft' to 'complete'

Implementation Principles

1. **Create New Files**
    - Implement changes in new files rather than modifying existing ones
    - Allow for parallel operation during transition
2. **Follow Single Responsibility Principle**
    - Break down functionality into smaller, focused components
    - Each file should have a single clear purpose
3. **Database Code Organization**
    - Isolate database-related code in separate files
    - Use a consistent naming convention for database operations

Technical Details

1. **Temporary Table Status Values**
    - Initial status: 'draft'
    - Final status: 'complete'
2. **Migration Script Requirement**
    - Script needed for temporary table creation
3. **Security Considerations**
    - Implement appropriate row-level security policies
    - Ensure users can only access their own listings

This document outlines the core requirements for the Listing Wizard redesign without proposing specific implementation details or file structures.