Listing Wizard Redesign Implementation Plan (Revised)

Phase 0: Analysis and Architecture Planning

Tasks:

1. **Analyze Existing Structure**
   - Document current wizard flow and component relationships
   - Identify components that can remain unchanged
   - Map data flow from UI to storage and back
2. **Design Database Changes**
   - Plan the temporary table structure
   - Design database functions needed
   - Document how flow ID will integrate with existing components
3. **Architecture Planning**
   - Create service/hook architecture that works with existing UI
   - Plan folder structure for new services within existing wizard folder
   - Document integration points between new services and existing components

Phase 1: Database Infrastructure Setup

Tasks:

1. **Create Database Migration Script**
   - Create the `temp_property_listing` table with required fields
   - Set up appropriate indexes for performance
   - Implement row-level security policies for data protection
   - Create foreign key relationship to profiles table
2. **Develop Database Functions**
   - Function to create new temporary listing
   - Function to update existing temporary listing
   - Function to query listing by ID
   - Function to finalize and copy to permanent table
3. **Testing & Validation**
   - Test database functions through SQL client

- ○ Verify RLS policies function correctly
- ○ Ensure proper cleanup of abandoned drafts

Phase 2: Service Layer Development

Tasks:

1. **Create New Services Within Existing Structure**
    - Develop flow initialization service for creating entries in temp table
    - Create flow retrieval service to get data by flow ID
    - Build update service for saving step progress
    - Implement finalization service for publishing
2. **Flow State Management**
    - Create new hooks that work with existing UI components
    - Build adapters to connect existing UI to new storage approach
    - Implement URL parameter handling for flow ID
3. **Testing & Validation**
    - Test services with existing UI components
    - Verify data flow between UI and database
    - Validate flow state persistence

Phase 3: Adapting Entry Point

Tasks:

1. **Modify Listing Selection**
    - Update "Create Your Listing" button handler
    - Add flow creation in temporary table
    - Implement redirect with flow ID parameter
    - Ensure backward compatibility
2. **URL Parameter Integration**
    - Add flow ID parameter to routes
    - Create utility to extract and validate flow ID
    - Handle cases of invalid or missing flow IDs
3. **Testing & Validation**
    - Test flow initialization and redirection
    - Verify correct flow type is recorded
    - Ensure URL parameters work as expected

Phase 4: Adapting Existing Components

Tasks:

1. **Create Connection Layer**
    - Develop adapter hooks for existing UI components
    - Build bridge between UI state and database
    - Implement auto-save functionality
2. **Modify Navigation Logic**
    - Update step navigation to save to database
    - Adapt progress tracking to use database state
    - Ensure browser refresh maintains state
3. **Testing & Validation**
    - Test step progression with database updates
    - Verify state maintenance across page reloads
    - Validate that UI components display correct data

Phase 5: Review and Submission Update

Tasks:

1. **Adapt Review Component**
    - Update data source to use flow ID from URL
    - Maintain existing UI presentation
    - Connect to new submission service
2. **Update Submission Process**
    - Modify submission to use finalization service
    - Implement status update from 'draft' to 'complete'
    - Ensure proper redirection after submission
3. **Testing & Validation**
    - Test complete flow from start to finish
    - Verify data integrity in final submission
    - Ensure proper handling of edge cases

Phase 6: Migration Support

Tasks:

1. **Progressive Rollout Strategy**

   - Implement feature flags for controlled access
   - Support both storage approaches during transition
   - Create fallback mechanisms for any issues
2. **Monitoring Implementation**

   - Add logging for critical operations
   - Implement error tracking specific to the new flow
   - Create dashboard for monitoring completion rates
3. **Documentation & Training**

   - Update technical documentation
   - Document troubleshooting procedures
   - Create internal knowledge base for support

Phase 7: Legacy Cleanup

Tasks:

1. **Identify Redundant Code**

   - Map unused services and utilities
   - Document dependencies before removal
   - Create cleanup plan with minimal risk
2. **Incremental Removal**

   - Remove redundant flow detection logic
   - Clean up unnecessary utility functions
   - Maintain UI components and public interfaces
3. **Final Validation**

   - Verify system integrity after cleanup
   - Ensure no regressions in functionality
   - Confirm all flows work as expected

This revised implementation plan focuses on creating new services and hooks within the existing wizard structure, maintaining the current UI components while changing the underlying data flow and storage mechanisms. The approach minimizes disruption while achieving the goal of database-driven flow management.