Routie Roo - Project TODO

Phase 1: Database Schema & Setup

- ☑ Define routes table for storing generated routes
- ☑ Define contacts table for caching contact data
- ☑ Define route_waypoints table for storing route stops
- ☑ Push database schema changes

Phase 2: Google API Integration

- ☑ Request Google People API credentials from user
- ☑ Request Google Maps API key from user
- ☑ Implement Google OAuth flow for contacts access
- ☑ Create backend procedure to fetch contacts from Google People API
- ☑ Test contact retrieval and address extraction

Phase 3: Contact Selection & Route Creation

- ☑ Build contact list display UI
- ☑ Add contact selection checkboxes
- ☑ Filter contacts with valid addresses
- ☑ Create route generation form
- ☑ Implement backend route calculation with Google Maps Routes API
- ☑ Save generated routes to database

Phase 4: Map Visualization

- ☑ Integrate Google Maps component
- ☑ Display route polyline on map
- ☑ Add markers for each waypoint
- ☑ Show route details (distance, duration, optimized order)
- ☑ Add "Open in Google Maps" button with shareable URL

Phase 5: Route Sharing & Export

- ☑ Generate unique shareable route links
- ☑ Create public route view page
- ☑ Implement route privacy controls
- □ Add export to GPX format
- □ Add export to KML format
- ☑ Add copy link functionality

Phase 6: Testing & Documentation

- □ Test complete user flow (login → select contacts → generate route → share)
- ☑ Write vitest tests for route generation procedures
- ☑ Create user setup guide for Google API credentials
- □ Test on mobile browsers
- □ Verify all sharing methods work

Phase 7: Deployment

- □ Save project checkpoint
- □ Create deployment documentation
- □ Provide user guide for team testing

New Features - User Requested

Route Management

- ☑ Add option to disable route optimization (manual waypoint order)
- ☑ Implement route deletion functionality
- □ Add drag-and-drop waypoint reordering (future enhancement)
- ☑ Add confirmation dialog before deleting routes

Folder Organization

- ☑ Create folders/categories table in database
- ☑ Add folder selection when creating routes
- ☑ Display routes grouped by folder
- ☑ Allow moving routes between folders
- ☑ Add folder management UI (create, rename, delete)

Contact Enhancement Features

- ☑ Update database schema to store phone numbers with labels
- ☑ Update database schema to store contact photos/avatars
- ☑ Fetch phone numbers from Google People API
- ☑ Fetch contact photos from Google People API
- ☑ Display contact photos in contact list
- ☑ Display phone numbers with labels in contact list
- □ Show contact photos on map markers
- □ Include phone numbers in route waypoint details

Click-to-Call Features

- ☑ Create dropdown menu for calling service selection

- ☑ Add Google Voice calling option
- ☑ Add regular phone dialer (tel:) option
- ☑ Add WhatsApp calling option
- ☑ Add Skype calling option
- ☑ Store user’ s preferred calling service in database
- □ Add settings UI for default calling service preference (future enhancement)
- ☑ Make phone numbers clickable in contact list
- □ Make phone numbers clickable in route waypoint details (future enhancement)

New User Requests

- ☑ Add folder filter/selector UI to view routes by folder
- ☑ Add contact search bar to filter by name, phone, or address
- ☑ Display all phone numbers for each contact (not just first one)
- ☑ Add text messaging options (SMS, WhatsApp text)
- ☑ Add distance unit preference (KM vs Miles)
- ☑ Convert all distance displays based on user preference

Waypoint Phone Numbers

- ☑ Add phone numbers field to route_waypoints table
- ☑ Update route creation to save contact phone numbers to waypoints
- ☑ Display phone numbers in route detail page waypoint list
- ☑ Add PhoneCallMenu to waypoints for click-to-call and text messaging

Separate Call and Text Actions

- □ Create separate PhoneCallMenu component for calling only
- □ Create separate PhoneTextMenu component for texting only

- □ Update waypoint display to show both call and text buttons
- □ Test that text button opens messaging app directly

UI Fixes

- ☑ Update RouteDetail to show separate Call and Text buttons for waypoints
- ☑ Add visible folder filter dropdown in routes section on Home page
- ☑ Create Settings page with preferences (calling service, distance unit, etc.)
- ☑ Add navigation link to Settings page

Bug Fixes

- ☑ Fix "Route Not Found" error when creating and viewing routes
- ☑ Investigate route creation database storage
- ☑ Fix route retrieval by ID
- ☑ Test complete route creation and viewing flow

Phone Number Formatting Fix

- ☑ Create utility function to format US phone numbers as (XXX) XXX-XXXX
- ☑ Clean phone numbers for click-to-call (remove formatting characters)
- ☑ Update Home.tsx to use formatted phone numbers
- ☑ Update RouteDetail.tsx to use formatted phone numbers
- ☑ Update PhoneCallMenu to handle formatted numbers correctly
- ☑ Update PhoneTextMenu to handle formatted numbers correctly

Contact Management Enhancements

- ☑ Add labels/groups field to cached_contacts table
- ☑ Add isActive status field to cached_contacts table

- ☑ Update Google People API to fetch contact labels/groups
- ☑ Sync all contacts including those without addresses
- ☑ Display contact labels in contact list
- ☑ Add “Mark as Inactive” button for contacts
- ☑ Add filter to show/hide inactive contacts
- ☑ Create contact edit dialog UI
- ☑ Add ability to add/edit contact addresses in app
- ☑ Add ability to edit contact phone numbers
- □ Implement Google People API update endpoint
- □ Sync local contact changes back to Google Contacts
- □ Add validation for address and phone number fields

New Feature Requests - Phase 2

Google People API Write Functionality

- ☑ Update Google OAuth scope to include write permissions
- ☑ Implement updateGoogleContact function in googleAuth.ts
- □ Note: Google People API write requires service account or different auth approach
- □ Consider alternative: Export contacts to CSV/vCard for manual import

Route Notes

- ☑ Add notes field to routes table in database schema
- ☑ Update route creation form to include notes textarea
- ☑ Display notes in route detail page
- □ Add edit notes functionality to existing routes
- ☑ Push database schema changes
- ☑ Test route notes feature

Google Calendar Integration

- ☑ Add Google Calendar OAuth scope to existing auth
- ☑ Implement createCalendarEvent function
- ☑ Add “Add to Calendar” button on route detail page
- ☑ Create calendar dialog for selecting start time
- ☑ Generate calendar event with route details (name, waypoints, time, notes)
- ☑ Handle calendar OAuth callback
- ☑ Test calendar event creation

Stop Types with Color Codes and Shapes

- ☑ Add stopType field to route_waypoints table (pickup, delivery, meeting, visit, other)
- ☑ Add stopColor field to route_waypoints table
- ☑ Create stop type selector component
- ☑ Add stop type selector in route creation UI
- □ Update map markers to show different shapes/colors based on stop type
- □ Add legend showing stop type meanings
- □ Update waypoint list to display stop type badges
- ☑ Push database schema changes

Bug Fixes

Distance Unit Display

- ☑ Fix route list to show miles when user selects miles in settings
- ☑ Fix route detail page to show miles when user selects miles in settings
- ☑ Ensure unit conversion is applied consistently across all distance displays

Route Execution Workflow

Database Schema Updates

- ☑ Add status field to route_waypoints (pending, in_progress, complete, missed)
- ☑ Add completedAt timestamp to route_waypoints
- ☑ Add missedReason text field to route_waypoints
- ☑ Add rescheduledDate field to route_waypoints
- ☑ Add executionOrder field to route_waypoints (for reordering during execution)
- ☑ Add executionNotes field to route_waypoints
- ☑ Add needsReschedule boolean flag to route_waypoints
- ☑ Push database schema changes

Stop Status Tracking

- ☑ Create stop status badge component (pending/in-progress/complete/missed)
- ☑ Add status update buttons to route detail page
- ☑ Implement updateWaypointStatus tRPC procedure
- ☑ Add completion timestamp recording
- ☑ Add reason/notes input for missed stops
- ☑ Show stop completion progress indicator on route
- ☑ Add comments/notes field for route drivers

Stop Reordering

- □ Add drag-and-drop reordering UI to route detail page
- ☑ Implement updateWaypointOrder tRPC procedure
- ☑ Save execution order separately from original order
- □ Add “Reset to Original Order” button
- □ Show visual indicator when order differs from original

Reschedule System

- ☑ Add “Reschedule” button for missed stops
- ☑ Create reschedule dialog with date/time picker
- □ Integrate with Google Calendar API for reschedule events
- ☑ Mark stop as rescheduled when date is set
- ☑ Clear needsReschedule flag when rescheduled

Manager Dashboard

- ☑ Create “Missed Stops” dashboard view
- ☑ Show all missed stops across all routes
- ☑ Filter by: needs reschedule, rescheduled
- □ Add bulk actions (assign, reschedule, mark resolved)
- ☑ Show route name and link to route detail
- □ Add export to CSV functionality

Route Execution UI

- ☑ Create route execution panel component
- ☑ Show current stop with prominent highlight
- ☑ Show route progress (X of Y stops complete)
- ☑ Add quick action buttons (Complete, Miss, Add Note)
- ☑ Add execution notes dialog for each stop
- ☑ Display missed reason and reschedule date
- □ Add “Next Stop” navigation button
- □ Show estimated time remaining based on incomplete stops

Testing

- ☑ Write tests for waypoint status updates
- ☑ Write tests for stop reordering

- ☑ Write tests for reschedule workflow
- ☑ Test manager dashboard queries
- ☑ Test execution mode UI flows

UI Integration Improvements

Waypoint and Route Completion Integration

- ☑ Merge execution controls into main waypoints list
- ☑ Show stop status badges inline with waypoint details
- ☑ Add execution action buttons (Complete, Miss, Add Note) to each waypoint card
- ☑ Display execution notes and missed reasons inline
- ☑ Show progress bar at top of waypoints section
- ☑ Remove separate RouteExecutionPanel component

New Feature Requests - Phase 3

Drag-and-Drop Stop Reordering

- ☑ Install @dnd-kit/core and @dnd-kit/sortable packages
- ☑ Add drag handles to waypoint cards
- ☑ Implement drag-and-drop functionality for reordering stops
- ☑ Update executionOrder when stops are reordered
- ☑ Show visual feedback during dragging
- □ Add "Reset to Original Order" button

Bulk Stop Actions

- ☑ Add "Complete All Remaining" button to route detail page
- ☑ Add "Mark All as Missed" bulk action with reason dialog

- ☑ Show confirmation dialog before bulk actions
- ☑ Update progress bar after bulk actions

Bug Fixes

- ☑ Fix "Show inactive contacts" to only show inactive when checked (not add to active)
- ☑ Update contact filtering logic in Home.tsx

Bug Fixes - Phase 4

Contact Display Issues

- ☑ Fix contacts showing resource IDs (10a5e7768dc5c4f2, 5db8c017898daefe, myContacts) instead of names
- ☑ Investigate Google Contacts sync parsing logic
- ☑ Filter out non-contact entries from sync results

UX Clarification

- ☑ Document that drag-and-drop reordering is on route detail page, not home page
- □ Add visual indicator or tooltip showing drag handles are available on route execution

Shared Route Execution Links

Database Schema

- ☑ Add shareToken field to routes table (unique, nullable)
- ☑ Add isPubliclyAccessible boolean to routes table
- ☑ Add sharedAt timestamp to routes table
- ☑ Push database schema changes

Backend Implementation

- ☑ Create generateShareToken procedure for routes
- ☑ Create getRouteByShareToken public procedure
- ☑ Allow unauthenticated access to route execution via share token
- ☑ Ensure waypoint status updates work with share token
- ☑ Add validation to prevent unauthorized route modifications

Frontend Implementation

- ☑ Add "Generate Share Link" button to route detail page
- ☑ Create shareable link with copy-to-clipboard functionality
- ☑ Create public route execution page (/share/:token)
- ☑ Show route details and waypoints without requiring login
- ☑ Allow marking stops as complete/missed on shared routes
- ☑ Add "Powered by Routie Roo" footer on public pages

Security & Privacy

- ☑ Ensure share tokens are cryptographically secure (UUID v4)
- ☑ Add option to revoke/regenerate share tokens
- ☑ Hide sensitive contact info on shared routes (shows contact name and address only)
- □ Add rate limiting to prevent abuse of public endpoints

Testing

- ☑ Write tests for share token generation
- ☑ Test unauthenticated route access
- ☑ Test waypoint updates via shared routes
- ☑ Verify missed stops appear in creator's dashboard

Shared Route Enhancements - Phase 2

Reschedule Functionality

- ☑ Add reschedule date picker to missed stop dialog on shared route page
- ☑ Update backend to handle reschedule via share token
- ☑ Show rescheduled date on shared route waypoint cards
- ☑ Add rescheduleWaypointPublic procedure

Phone Communication

- ☑ Add PhoneCallMenu component to shared route waypoints
- ☑ Add PhoneTextMenu component to shared route waypoints
- ☑ Parse and display phone numbers from waypoint data
- ☑ Ensure phone buttons work without authentication

Drag-and-Drop Reordering

- ☑ Add @dnd-kit components to SharedRouteExecution page
- ☑ Implement drag handles for waypoints on shared route
- ☑ Update execution order via share token when reordering
- ☑ Show visual feedback during dragging

Testing

- ☑ Test reschedule functionality on shared routes
- ☑ Test phone call/text from shared route
- ☑ Test drag-and-drop reordering on shared route
- ☑ Verify all updates sync back to creator’ s account

Branding Updates

- ☑ Copy RoutieRoo logo to client/public directory

- ☑ Update APP_LOGO constant to use new logo
- ☑ Hide “Routie Roo” text in header
- ☑ Test logo display across all pages

Logo Size Fix

- ☑ Increase logo size in header for better visibility

Logo Size Adjustment

- ☑ Increase logo size further for better visibility (h-16)

Custom Stop Type Management

Database Schema

- ☑ Create stop_types table with name, color fields
- ☑ Add userId foreign key to stop_types for user-specific types
- ☑ Push database schema changes

Backend Implementation

- ☑ Create stopTypes.list procedure to fetch user’s stop types
- ☑ Create stopTypes.create procedure to add new stop types
- ☑ Create stopTypes.update procedure to edit existing stop types
- ☑ Create stopTypes.delete procedure to remove stop types
- ☑ Validate stop type colors (hex format)

Frontend Implementation

- ☑ Create StopTypesSettings component in Settings page
- ☑ Add list of current stop types with edit/delete buttons

- ☑ Create add/edit stop type dialog with color picker
- ☑ Update StopTypeSelector to use custom stop types from database
- ☑ Show color preview in stop type selector
- ☐ Prevent deletion of stop types currently in use

Testing

- ☑ Test creating custom stop types
- ☑ Test editing stop type colors
- ☑ Test deleting unused stop types
- ☑ Verify stop types persist across sessions

Bug Fixes - Null Address Validation

- ☑ Fix route creation validation error when contacts have null addresses
- ☑ Prevent selection of contacts without addresses in route creation UI

UX Improvements - Contact Address Management

- ☑ Add visual warning indicator (icon) on contact cards missing addresses
- ☑ Add quick “Add Address” button on contact cards for faster editing
- ☑ Create “Contacts Without Addresses” filter option
- ☑ Fix PhoneTextMenu to include Google Voice as a texting option

Critical Route Fixes - Priority

- ☑ Add starting point field to routes (home/office address)
- ☑ Add starting point selection in route creation UI
- ☑ Store user’s default starting point in settings
- ☑ Fix mobile responsiveness for shared route pages
- ☑ Fix map display on mobile devices

- ☑ Ensure touch interactions work on mobile
- □ Implement numbered/lettered map markers (A, B, C)
- □ Match marker labels to waypoint list order

Route Editing Features

- [x] Add backend procedure to add waypoints to existing routes
- [x] Add backend procedure to remove waypoints from routes
- [x] Add backend procedure to update waypoint addresses
- [x] Create route edit mode UI with "Edit Route" button
- [x] Add contact selector to add new waypoints to route
- [x] Add remove button for each waypoint in edit mode
- [x] Add inline address editing for waypoints
- [x] Auto-refresh map when waypoints are added/removed/reordered
- [x] Auto-refresh map when addresses are edited
- [x] Recalculate route distance/duration after changes
- [x] Add "Copy Route" button to duplicate existing routes

Route Completion Status

- ☑ Add completedAt field to routes schema
- ☑ Add backend logic to automatically mark route as completed when all waypoints are completed/missed
- ☑ Display route completion status badge on route detail page
- ☑ Show completion timestamp when route is completed

Bug Fixes - Shared Routes

- ☑ Fix shared route map showing wrong starting location (San Francisco instead of route’s starting point)
- ☑ Fix shared route displaying distance in km instead of owner’s preferred unit (miles)

Starting Point System

- ☑ Create saved starting points table (name, address, user)
- ☑ Add backend procedures to manage saved starting points (create, list, delete)
- ☑ Add startingPointAddress field to routes table
- ☑ Create starting point dropdown in route creation UI
- ☑ Allow selecting from saved starting points or entering custom address
- ☑ Add manage starting points UI in Settings
- ☑ Fix shared route map to center on route’s starting point
- ☑ Store route owner’s distance unit preference with route for shared views

Routie Roo Rebranding

- [x] Update app name to “Routie Roo” (user can change in Settings → General)
- [x] Add playful microcopy to button labels and CTAs
- [x] Update empty states with friendly Routie messages
- [x] Add personality to toast notifications
- [x] Update success/error messages with encouraging tone
- [x] Keep professional balance - playful but not gimmicky
- [x] Replace all instances of “Contact Route Mapper” with “Routie Roo” in codebase
- [x] Fix Select.Item empty value error preventing login

- [x] Add visible logout button to header navigation
- [x] Fix save button for starting locations not working when clicked
- [x] Add edit functionality for saved starting points (update name and address)
- [x] Make route map sticky/stationary while scrolling through waypoints list
- [x] Add phone call/text icons back to contact cards on home page
- [x] Display phone numbers on route detail screen for each waypoint

Map Marker Improvements

☑ Add numbered markers (1, 2, 3, etc.) to map matching waypoint list order

☑ Update marker labels to show numbers instead of default pins

☑ Add Google Maps icon button to each waypoint in the list for quick navigation

☑ Ensure marker numbers stay synchronized when waypoints are reordered

Display Issues

☑ Show actual phone numbers as text on route waypoint page (e.g., "(936) 900-1828 (Patient)")

☑ Remove technical metadata (group IDs like "10a5e7768dc5c4f2") from contact cards on home page

Layout Improvements

☑ Adjust route detail page layout - make map smaller and waypoint section wider for better visual balance

☑ Fine-tune layout proportions - current ⅓ map and ⅔ waypoints is too wide, find better balance

☑ Remove redundant phone emoji from phone number display (Call button already has icon)

☑ Fix hex ID still showing in contact labels (e.g., "63c2ee10e427465")

New Features

Route Export

- ☑ Add “Export to CSV” button on route detail page
- ☑ Export route metadata (name, distance, duration, date)
- ☑ Export all waypoints with addresses, phone numbers, and completion status
- ☑ Generate downloadable CSV file

Contact Import

- ☑ Add “Import Contacts” button on home page
- ☑ Create CSV upload interface with file picker
- ☑ Parse CSV and validate required fields (name, address)
- ☑ Validate addresses using Google Maps Geocoding
- ☑ Show import preview with validation results
- ☑ Bulk insert validated contacts into database

Route Scheduling Calendar

- ☑ Create new Calendar page/view
- ☑ Display routes in monthly calendar grid
- ☑ Show route cards with name, distance, and time on scheduled dates
- □ Implement drag-and-drop to reschedule routes (future enhancement)
- □ Add conflict detection for overlapping routes on same day (future enhancement)
- □ Show conflict warnings when routes overlap (future enhancement)
- ☑ Add navigation link to Calendar in main navigation
- ☑ Add calendars table for users with multiple calendars (work, personal, etc.)
- ☑ Add calendarId field to routes table
- □ Add calendar selector in route creation and calendar view (future enhancement)

Google Contact Labels Display

- ☑ Display Google Contact labels on waypoint cards in route detail page
- ☑ Add label filter dropdown to contact selection on home page
- ☑ Show label badges on waypoint cards with proper styling

Google Contact Labels Display

- ☑ Display Google Contact labels on waypoint cards in route detail page
- ☑ Add label filter dropdown to contact selection on home page
- ☑ Show label badges on waypoint cards with proper styling
- ☑ Fix Filter by Label dropdown not showing on home page
- ☑ Remove “myContacts” and “starred” from label display (show only custom labels)

Mobile Issues

- ☑ Fix map pins not showing on mobile devices when viewing shared execution link

New Features - Route Scheduling & Completion

Route Scheduling Date Picker

- ☑ Add date picker field to route creation form
- ☑ Save scheduledDate when creating routes
- ☑ Display scheduled routes on calendar view
- □ Allow editing scheduled date from route detail page

Offline Mode for Route Execution

- □ Implement local storage caching for route data

- □ Cache waypoints and route details when route is loaded
- □ Allow status updates (complete/miss) to work offline
- □ Queue offline changes and sync when connection restored
- □ Show offline indicator when no connection
- □ Display sync status (pending/synced)

Route Completion Summary

- □ Detect when all waypoints are complete
- □ Display completion summary modal with statistics
- □ Show total time, miles driven, completed vs missed stops
- □ Add PDF export button for completion summary
- □ Generate PDF report with route details and statistics
- ☑ Verify label dropdown shows when custom labels exist

Recurring/Repeat Dates (Google Calendar-style)

- ☑ Add date picker for route scheduling
- □ Create RecurrenceSelector component with:
  - Repeat frequency (Daily, Weekly, Monthly, Yearly)
  - Repeat interval (Every 1, 2, 3… weeks/months/etc)
  - Days of week selector (for weekly recurrence)
  - End options (Never, On date, After X occurrences)
- □ Update database schema with recurrence fields
- □ Generate recurring route instances based on pattern
- □ Display recurring routes on calendar view
- □ Allow editing/deleting single occurrence or entire series
- □ Remove numbered marker from starting location (it’ s just an anchor, not a stop)

Custom Labels Not Displaying Issue

- ☑ Investigate what label data is stored in database for contacts
- ☑ Check label filtering logic in ContactCard.tsx
- ☑ Verify label extraction from Google Contact groups format
- ☑ Add fetchContactGroupNames function to resolve group IDs to names
- ☑ Update sync process to fetch and resolve group names
- ☑ Fix label display on contact cards and filter dropdown
- ☑ Update SortableWaypointItem to show resolved labels

Shared Route Execution Map Issues

- ☑ Add numbered markers to shared route execution map (they were rendering, just needed better zoom)
- ☑ Fix incorrect distance display in SharedRouteExecution (was showing meters as km)
- ☑ Verify distance calculation uses correct units (meters vs kilometers)

Shared Route Map Not Rendering

- ☑ Investigate why route line and markers are not showing on shared route execution page
- ☑ Check if route data is being fetched correctly from API
- ☑ Fixed map rendering by filtering waypoints without coordinates
- ☑ Added warning banner for missing coordinates
- ☑ Tested with actual shared route: https://routieroo.manus.space/share/74f21387-337f-4692-83c4-005d5c4638a3

Re-optimize Route with New Stops

- ☑ Add database field to track when waypoints were added (createdAt timestamp already exists)
- ☑ Implement algorithm to find optimal insertion points for new stops
- ☑ Add "Re-optimize Route" button to route detail page
- ☑ Calculate best position for each new stop without changing existing order
- ☑ Update route distance and duration after re-optimization
- ☑ Write tests for re-optimization logic (4⁄4 tests passing)

Auto-use Default Starting Point

- ☑ Update route creation logic to use user's default starting point from settings when "None" is selected
- ☑ Ensure custom starting point and saved starting points still override the default
- ☑ Implemented priority: custom > saved > default from settings > none

Hide Completed Routes Filter

- ☑ Add checkbox filter to hide routes where all waypoints are marked complete
- ☑ Filter works alongside existing folder filter
- ☑ Added checkbox to route list header

Route Archive System

Database Schema

- ☑ Add `isArchived` boolean field to routes table (default false)
- ☑ Add `archivedAt` timestamp field to routes table

- ☑ Add `autoArchiveDays` field to users table (nullable int, null = never auto-archive)
- ☑ Run database migration

Backend Procedures

- ☑ Create `archiveRoute` mutation (sets isArchived=true, archivedAt=now)
- ☑ Create `unarchiveRoute` mutation (sets isArchived=false, archivedAt=null)
- ☑ Update `getUserRoutes` to filter out archived routes by default
- ☑ Create `getArchivedRoutes` query to fetch archived routes
- ☑ Update user settings mutation to handle autoArchiveDays

Frontend UI

- ☑ Create ArchivedRoutes page component
- ☑ Add "Archived Routes" link to navigation header
- ☑ Add Archive button to route detail page
- ☑ Add Unarchive button to archived routes page (already in ArchivedRoutes.tsx)
- ☑ Add Archive button to route list items
- ☑ Update route list to exclude archived routes (backend filters them out)
- □ Add auto-archive settings to Settings page (optional - can be added later)

Auto-Archive Logic

- □ Implement auto-archive check (can be done via scheduled job or on-demand)
- □ Auto-archive completed routes based on user's autoArchiveDays setting

Route Detail Header Layout

- ☑ Reorganize route detail page header buttons to wrap to multiple rows
- ☑ Group related buttons together (primary, sharing, secondary actions)

- ☑ Ensure responsive layout with flex-wrap that doesn' t overflow on smaller screens
- ☑ Shortened button labels for cleaner appearance

Route Notes/Comments System

Database Schema

- ☑ Create `route_notes` table with fields: id, routeId, userId, note (text), createdAt, updatedAt
- ☑ Run database migration

Backend Procedures

- ☑ Create `addNote` mutation to add new note to a route
- ☑ Create `getNotes` query to fetch all notes for a route (ordered by createdAt desc)
- ☑ Create `deleteNote` mutation to remove a note
- ☑ Create `updateNote` mutation to edit existing note

Frontend UI

- ☑ Add notes section to route detail page
- ☑ Create note input field with "Add Note" button
- ☑ Display notes list with timestamp and content
- ☑ Add edit/delete buttons for user' s own notes
- ☑ Show relative timestamps (e.g., "2 hours ago" , "Yesterday")
- ☑ Add empty state message when no notes exist

Testing

- ☑ Write tests for note CRUD operations (5⁄5 tests passing)
- ☑ Test note permissions (users can only edit/delete their own notes)

Auto-Archive Settings UI

- ☑ Add auto-archive settings section to Settings page
- ☑ Create dropdown/select for auto-archive timeframe (Never, 7 days, 30 days, 60 days, 90 days)
- ☑ Save autoArchiveDays preference to user settings
- ☑ Display current auto-archive setting

Route Completion Progress Badges

- ☑ Calculate completed stops vs total stops for each route using SQL subqueries
- ☑ Add progress badge to route cards in main list (e.g., "8⁄12 stops")
- ☑ Style badge with visual indicator (green=complete, blue=in progress, gray=not started)
- ☑ Show badge on both active and archived route lists

Railway Deployment OAuth Fix

- ☑ Replace Manus OAuth portal with direct Google OAuth implementation
- ☑ Update frontend login flow to use Google OAuth directly
- ☑ Install googleapis and jsonwebtoken packages
- ☑ Update ENV configuration with Google OAuth credentials
- ☑ Fix JWT payload to include appId field (required by session validator)
- ☑ Replace jsonwebtoken with jose library (already installed)
- ☐ Test authentication on Railway deployment

Calendar OAuth Redirect URI Fix

- ☑ Add publicUrl to ENV configuration
- ☑ Update googleOAuthRoute.ts to use ENV.publicUrl instead of req.headers.host

- ☑ Update routers.ts calendar OAuth URL generation to use ENV.publicUrl
- ☑ Add ENV import to routers.ts
- ☐ Test calendar OAuth flow on Manus deployment

Main OAuth Login Redirect URI Fix

- ☑ Fix main OAuth login to use ENV.publicUrl instead of dynamic host
- ☑ Update oauth.ts or relevant OAuth route handler
- ☐ Test login flow on Manus deployment

Google Contacts Sync OAuth Redirect URI Fix

- ☑ Find and fix contacts sync OAuth redirect URI
- ☑ Update to use ENV.publicUrl instead of dynamic host
- ☐ Test contacts sync flow on Manus deployment

Calendar Integration Improvements

Phase 1 - Calendar Scheduling (In Progress)

- ☑ Add calendar selection UI (let user choose which calendar)
- ☑ Fetch user’ s calendar list from Google Calendar API
- ☑ Update backend to create separate events for each waypoint
- ☑ Calculate individual time slots for each stop based on route timing
- ☑ Keep user in Routie Roo instead of redirecting to Google Calendar
- ☑ Update route status to “scheduled” after adding to calendar
- ☑ Store scheduled date/time in route database
- ☑ Add googleCalendarId field to routes table
- ☑ Remove “contactGroups/myContacts” from label display
- ☐ Show scheduled date/time in route cards

- ☐ Test calendar integration with multiple events

Phase 2 - Calendar View Component (Future)

- ☐ Build calendar view component in Routie Roo
- ☐ Fetch and display Google Calendar events
- ☐ Show birthday calendar from Google Contacts
- ☐ Display routes on the calendar
- ☐ Support different views (month/week/day)
- ☐ Sync with multiple Google calendars

Calendar Event Creation Bug

- ☑ Investigate "Failed to create calendar event" error
- ☑ Add calendar.readonly OAuth scope
- ☑ Add detailed error logging to calendar OAuth callback
- ☑ Add error logging to getCalendarList function
- ☑ Test on published site to see actual error logs
- ☑ Fix the underlying calendar API issue - calendar.readonly scope was the fix!

Calendar View Component - Phase 2

- ☑ Create CalendarView page component (already exists as Calendar.tsx)
- ☑ Add backend procedure to fetch calendar events from Google Calendar API
- ☑ Display scheduled routes on calendar grid
- ☐ Display contact birthdays from Google Contacts
- ☑ Add monthly navigation (previous/next month)
- ☑ Add event click to view route details
- ☑ Style calendar with color coding for different event types
- ☐ Add filter to show/hide different event types

Bulk Scheduling Feature

- ☐ Create bulk scheduling UI component
- ☐ Add route selection checkboxes to route list
- ☐ Implement smart time slot suggestion algorithm
- ☐ Add date range picker for bulk scheduling
- ☐ Add time spacing configuration (e.g., 2 hours between routes)
- ☐ Create backend procedure to schedule multiple routes at once
- ☐ Show preview of suggested schedule before confirming
- ☐ Handle conflicts with existing calendar events
- ☐ Add bulk calendar event creation

Default Stop Time Configuration

- ☑ Add defaultStopDuration field to user preferences (15, 30, 45, 60 minutes)
- ☑ Add stop duration setting to Settings page
- ☑ Update calendar event creation to include stop duration + drive time
- ☑ Calculate accurate event times: arrival + stop duration + drive to next stop
- ☐ Show estimated schedule in calendar event creation dialog
- ☐ Allow per-route override of default stop duration

Calendar Event Duration Mode

- ☑ Add eventDurationMode field to user preferences ("stop_only" | "include_drive")
- ☑ Add setting in Settings page to choose duration mode
- ☑ Update createWaypointEvents to support both modes:
  - "stop_only" : Event duration = stop duration, drive time added between events
  - "include_drive" : Event duration = stop duration + drive time to this location

- ☑ Add tooltip explaining the difference between modes

Calendar Page UI Updates

- ☑ Update calendar page heading to “Routie Roo Calendar - Let’s hop to it!”

Full Google Calendar Integration

- ☑ Add function to fetch Google Calendar events for a date range
- ☑ Update calendar.getEvents backend procedure to merge routes + Google Calendar events
- ☑ Handle calendar OAuth token storage and refresh
- ☑ Add color coding: routes (blue), other events (gray)
- ☑ Display event details on calendar grid
- ☑ Add Connect/Disconnect Google Calendar in Settings
- □ Add filter toggles to show/hide different event types
- □ Handle events that span multiple days
- □ Show event times on hover tooltip

OAuth Redirect URI Fix

- ☑ Check existing Google OAuth redirect URIs
- ☑ Update calendar connection to use same redirect URI as contacts sync
- ☑ Updated main callback to handle both contacts and calendar via action parameter
- □ Test calendar connection with corrected redirect URI

Calendar View UX Improvements

- ☑ Fix color coding - routes are now blue (bg-blue-500), Google events are gray (bg-gray-300)

- ☑ Make “+X more” clickable to show all events for that day in a dialog
- ☑ Show event times for all events in dialog view
- ☑ Display event location and description in dialog
- ☐ Add calendar selection filter to choose which calendars to display

Calendar Agenda View Redesign

- ☐ Replace monthly grid with day/week agenda view (like Google Calendar)
- ☐ Add time slots (hourly) on the left side
- ☐ Display events as blocks with full titles and times
- ☐ Add calendar selection sidebar with checkboxes
- ☐ Implement view switcher (Day / Week / Month)
- ☐ Color code events by calendar source
- ☐ Show event details on hover
- ☐ Make events clickable to view/edit

Route Editing Functionality

- ☑ Add backend procedure to update route name
- ☑ Add backend procedure to update route notes
- ☑ Add backend procedure to change starting point
- ☑ Add backend procedure to move route to different folder
- ☑ Add backend procedure to add waypoints to existing route (already existed)
- ☑ Add backend procedure to remove waypoints from route (already existed)
- ☑ Create route edit dialog UI with all fields
- ☑ Wire up edit dialog to backend procedures
- ☑ Test all editing operations (4⁄4 tests passing)

Route Editing UX Improvements

- ☑ Move Edit Route button from top actions to Waypoints & Execution section header
- ☑ Add visible “Add Contact” button in Waypoints section
- ☑ Remove contact (trash icon) already visible on each waypoint
- □ Test that all buttons are easily discoverable

Calendar Event Management on Route Edit

- ☑ Show warning in edit dialog when route has googleCalendarId
- ☑ Add “Delete Calendar Events” button in edit dialog
- ☑ Add backend procedure to clear googleCalendarId from route
- ☑ Display info message: “Calendar events won’t update automatically”
- ☑ Allow user to delete old events and recreate with updated route

Remove Contact Button Fix

- ☑ Find where trash icon should be displayed on waypoints
- ☑ Add visible trash/delete button to each waypoint (always visible now)
- ☑ Wire up to removeWaypoint mutation
- ☑ Test waypoint removal functionality

UI Fixes

- ☑ Remove number from starting point display (replaced with flag icon)
- ☑ Filter out labels starting with “contactGroups” from contact labels

Calendar Connection Debug

- ☑ Investigate why Google Calendar events aren’t showing despite connection

- ☑ Implement automatic token refresh when access token expires
- ☑ Update calendar.getEvents to refresh token if expired
- □ Test token refresh after 1 hour expiry
- □ Check if googleCalendarAccessToken is being saved correctly
- □ Verify calendar.getEvents is fetching from Google Calendar API
- □ Add error logging to calendar event fetching

Calendar Agenda View Redesign

- □ Design day/week view with hourly time slots (7am-9pm)
- □ Add view switcher (Day / Week / Month)
- □ Display events as blocks positioned by time
- □ Add sidebar with mini calendar for date navigation
- □ Show calendar list with checkboxes to filter calendars
- □ Color code events by calendar
- □ Handle overlapping events
- □ Add current time indicator

Calendar Agenda View Redesign

- ☑ Create day view with hourly time slots (6am-10pm)
- ☑ Create week view with 7 columns and hourly rows
- ☑ Display events as colored blocks positioned by time
- ☑ Show full event titles and times on blocks
- ☑ Add view switcher buttons (Day/Week/Month)
- ☑ Keep existing monthly grid as Month view option
- ☑ Make Day view the default
- ☑ Add navigation for previous/next day or week
- ☑ Events displayed inline (no overlap handling needed for now)

- ☑ Add current time indicator line (red line) for today

Calendar Management Features

- ☑ Add backend procedure to fetch user’ s calendar list from Google Calendar API
- ☑ Store calendar preferences (visible/hidden) in user settings
- ☑ Add calendar sidebar to Calendar page with list of all calendars
- ☑ Add checkboxes to show/hide individual calendars
- ☑ Color-code events by their source calendar (not just route vs other)
- □ Add calendar selector dropdown when scheduling routes
- □ Save selected calendar as user’ s default for route scheduling
- ☑ Filter events based on calendar visibility preferences
- □ Show calendar name on event hover/click

Calendar Sidebar Debug - Calendar List Not Showing

- ☑ Investigate why calendar sidebar shows “No calendars found” when calendar is connected
- □ Issue: User context not refreshed after calendar connection - useAuth() returns stale user data
- □ Add refetch mechanism to calendar list query
- □ Add manual refresh button to calendar sidebar
- □ Invalidate auth.me query when returning from calendar connection
- □ Test that calendar list appears after connecting calendar

Critical Bugs to Fix Before Publishing

- ☑ Contact groups (contactGroups/xxx) still showing as labels on contacts
- ☑ Contact groups still appearing in label filter dropdown
- ☑ Remove all contactGroups references from contact display

- ☑ Calendar sidebar not showing calendars even though tokens exist in database
- ☑ Fixed by adding auth refresh on calendar connection callback
- ☑ Settings page now refreshes user session after connecting calendar

Calendar Sidebar Still Not Working

- ☑ Calendar sidebar still shows “No calendars found” after connecting calendar
- ☑ Tested: disconnect → reconnect → still no calendars showing
- ☑ Root cause: Calendar tokens are in database but not in session context
- ☑ Solution: Fetch calendar tokens directly from database in calendar procedures
- □ Implement: getCalendarList should query DB for tokens using ctx.user.id
- □ Implement: getCalendarEvents should query DB for tokens using ctx.user.id
- □ Remove dependency on ctx.user.googleCalendarAccessToken

Reliable Calendar Sidebar Implementation

- ☑ Add googleCalendarList JSON field to users table to store calendar list
- ☑ Add calendarPreferences JSON field to users table to store visibility settings (already exists)
- ☑ Fetch calendar list during OAuth connection and save to database
- ☑ Create getCalendarList procedure that reads from database (not API)
- ☑ Create saveCalendarPreferences procedure to store visibility settings (updateCalendarPreferences already exists)
- □ Update Calendar page sidebar to load calendar list from database
- □ Add checkboxes for each calendar with instant filtering
- □ Filter events based on stored calendar preferences
- □ Write tests for calendar list storage and filtering
- □ Test complete flow: connect → see calendars → toggle visibility

Calendar OAuth Not Storing Calendar List

- ☑ Check server logs for OAuth callback execution
- ☑ Verify getCalendarList function is being called during OAuth
- ☑ Check if calendar list JSON is being saved to database
- ☑ Verify database has googleCalendarList field populated after connection
- ☑ Calendar sidebar now shows calendars successfully!

Calendar Events Not Displaying

- ☑ Calendar list shows in sidebar but no events appear
- ☑ Check if getEvents is passing selected calendar IDs
- ☑ Verify getEvents procedure fetches from Google Calendar API
- ☑ Update getEvents to filter by visible calendars
- ☑ Updated getAllCalendarEvents to accept calendar IDs filter
- ☑ Updated getEvents procedure to pass visibleCalendars parameter
- ☑ Updated Calendar.tsx to pass visibleCalendars to query
- ☐ Test that events appear after fixing query

Final Two Features Before Publishing

- ☑ Make contact labels larger font and bold
- ☐ Add “Add Event” button to Calendar page
- ☐ Create event creation dialog with title, date/time pickers, all-day checkbox
- ☐ Add recurrence options (does not repeat, daily, weekly, custom recurrence)
- ☐ Add calendar selector dropdown in event dialog
- ☐ Add description field for events
- ☐ Create backend procedure to save non-routed events to Google Calendar
- ☐ Test event creation flow
- ☐ Write unit tests for event creation

Final Polish Items - November 2024

Contact Label Styling

- ☑ Make contact labels larger (text-sm instead of text-xs)
- ☑ Make contact labels bold (font-bold)
- ☑ Verify labels display prominently on contact cards

Non-Routed Calendar Events

- ☑ Create AddEventDialog component with full event creation UI
- ☑ Add title input field
- ☑ Add start/end date and time pickers
- ☑ Add “All day” checkbox
- ☑ Add recurrence options (none, daily, weekly, monthly, custom)
- ☑ Add custom recurrence configuration (interval, end conditions)
- ☑ Add calendar selector dropdown
- ☑ Add description textarea
- ☑ Create backend calendar.createEvent tRPC procedure
- ☑ Implement Google Calendar API integration for event creation
- ☑ Add “Add Event” button to Calendar page
- ☑ Test event creation with all field combinations
- ☑ Write unit tests for event creation validation

Route Optimization Bug Fixes - November 2024

Initial Optimization Not Working

- ☑ Investigate why initial route creation with “Optimize Route Order” enabled doesn’t reorder waypoints
- ☑ Check if Google Routes API optimization flag is being passed correctly

- ☑ Verify waypoint order is being saved from API response
- ☑ Test initial optimization with multiple contacts
- ☑ Fix: Added optimize parameter to calculateRoute function
- ☑ Fix: Pass optimizeRoute flag to calculateRoute when creating routes
- ☑ Fix: Set requestBody.optimizeWaypointOrder = true when optimize flag is true

Re-optimization Reverting Manual Changes

- ☑ Fix re-optimization to preserve manually reordered waypoints
- ☑ Ensure re-optimize only finds optimal positions for newly added stops
- ☑ Keep existing waypoints in their manual order
- ☑ Test re-optimization after manual drag-and-drop reordering
- ☑ Test re-optimization after adding new contacts to existing route
- ☑ Fix: When no new stops, recalculate route with current order (preserves manual changes)
- ☑ Fix: Update waypoint coordinates after re-optimization to refresh map
- ☑ Fix: Map automatically refreshes via useEffect dependency on waypoints

Home Page Layout Improvements - November 2024

Section Reorganization

- ☑ Move "Route Setup Screen" section to top of Home page
- ☑ Move "Your Contact List" section to bottom of Home page
- ☑ Change "Your Contacts" title to "Route Setup Screen" in route creation area
- ☑ Make all labels bold in route setup form (matching contact list style)

Custom Contact Fields - November 2024

Important Dates Feature

- ☑ Create important_date_types table in database (id, userId, name, createdAt)
- ☑ Add importantDates field to cached_contacts table (JSON array)
- ☑ Backend tRPC procedures created (list, create, delete)
- ☑ Create Settings UI for managing date type options (add, edit, delete)
- ☑ Add Important Dates section to contact edit dialog
- ☐ Add Important Dates editing in route setup screen (inline with contact selection)
- ☑ Allow multiple dates per contact with type dropdown
- ☑ Display important dates in contact card
- ☑ Display important dates in route detail page for each waypoint (read from contact)
- ☑ Changes made in route setup update the contact’ s permanent record (via ContactEditDialog)
- ☑ Test important dates functionality
- ☑ **FIXED**: Replaced HTML5 date input with shadcn/ui Calendar component
    - Comments feature works perfectly ✅
    - Important Dates feature now working with Calendar picker ✅
    - Calendar picker properly updates React state
    - Both features tested and verified working end-to-end

Comments Feature

- ☑ Create comment_options table in database (id, userId, option, createdAt)
- ☑ Add comments field to cached_contacts table (JSON array)
- ☑ Backend tRPC procedures created (list, create, delete)
- ☑ Create Settings UI for managing comment options (add, edit, delete)
- ☑ Add “Other” option that requires custom text input

- ☑ Add Comments section to contact edit dialog
- □ Add Comments editing in route setup screen (inline with contact selection)
- ☑ Display comments in contact card
- ☑ Display comments in route detail page for each waypoint (read from contact)
- ☑ Changes made in route setup update the contact’s permanent record (via ContactEditDialog)
- ☑ Test comments functionality

Route Setup Label Styling Fix - November 2024

- ☑ Update Route Setup form labels to use text-sm font-bold (matching contact label style)
- ☑ Verify all 6 labels have consistent styling

RouteDetail Label Styling - November 2024

- ☑ Update all labels in Waypoints & Execution screen to use !font-bold
- ☑ Verify labels match Home page styling
- ☑ Updated 10 Label components in RouteDetail.tsx

Contact Labels in RouteDetail - November 2024

- ☑ Make contact labels/tags bold in waypoint cards (PT Randy Harms, Abundant, etc.)
- ☑ Match Home page contact label styling (text-sm font-bold)
- ☑ Updated SortableWaypointItem.tsx line 83

Map Marker Numbering Fix - November 2024

- ☑ Change starting point marker from number “1” to flag icon on map
- ☑ Renumber actual stops starting from 1 (not 2)

- ☑ Ensure waypoint list and map markers use consistent numbering
- ☑ Updated RouteDetail.tsx lines 301-315 (flag icon), 325 (renumber from 1), 349 (adjust end marker)

Routie Roo Starting Point Marker - November 2024

- ☑ Copy routieroo2.png to client/public folder as routie-roo-marker.png
- ☑ Update map marker to use Routie Roo image instead of flag icon
- ☑ Adjust marker size (48x64) and anchor point (24, 64) for proper positioning

Waypoint List Numbering Fix - November 2024

- ☑ Update waypoint list to show stops numbered from 1 (not 2)
- ☑ Ensure list numbering matches map marker numbering
- ☑ Changed SortableWaypointItem.tsx line 67 from `index + 1` to `index`

Bug Fixes - Calendar Integration

Important Dates Calendar UI Issue

- □ Fix Calendar component displaying inline instead of in popover
- □ Ensure calendar only shows when "Pick a date" button is clicked
- □ Test calendar picker UI in ContactEditDialog

Google Calendar Event Display Issue

- □ Investigate why calendar events for Nov 25-28 are not showing in Randy PT Schedule calendar
- □ Check calendar event creation logic for date range restrictions
- □ Verify timezone handling in calendar integration
- □ Test event creation for specific date range (Nov 25-28)
- □ Fix root cause of missing calendar events

Critical Bugs - User Reported (Nov 26, 2025)

- □ Fix Calendar UI display bug - calendar showing inline instead of in popover dropdown when adding important dates
- □ Remove duplicate comment options (Needs Follow-up 3x, Special Instructions 2x, VIP Client 3x)
- □ Investigate Google Calendar events not showing for Nov 25-28 in Randy PT Schedule calendar

Address Management Improvements - November 2024

Address Update Workflow

- ☑ When user edits waypoint address on route screen, show dialog asking: "Is this a temporary change for this route only, or would you like to update the contact's permanent address?"
- ☑ Add "Temporary (route only)" option to keep contact address unchanged
- ☑ Add "Update contact address" option to save to cached_contacts table
- ☑ Create backend procedure contacts.updateAddress to update contact address
- ☑ Test that contact address updates persist and sync to contact card

Google Maps Address Verification

- ☑ Integrate Google Maps Places Autocomplete API for address validation
- ☑ Add address autocomplete dropdown with suggestions as user types
- ☑ Show formatted address suggestions with city, state, ZIP
- ☑ Validate addresses against Google Maps database
- ☑ Auto-fill complete address when user selects from dropdown
- ☑ Add "powered by Google" attribution per API requirements
- ☑ Test address autocomplete in ContactEditDialog
- ☑ Test address autocomplete in waypoint address edit dialog

Page Header Updates - Routie Roo Themed Titles

- ☑ Route Setup Screen - “Plan Your Next Hop” (bold) with subtitle “Set up the route Roo will guide you through.”
- ☑ Your Routes - “Your Hop Library” (bold) with subtitle “A clear, hop-by-hop look at your route.” (italic/cursive)
- ☑ Your Contacts - “Your Kangaroo Crew” (bold) with subtitle “Everyone you connect with along the journey.” (italic/cursive)
- ☑ Waypoints & Execution - “Hop-By-Hop Navigation” (bold) with subtitle “Every hop in order — smooth, simple, efficient.” (italic/cursive)
- ☑ Missed Stops - “Missed Hops” with subtitle “Stops that need a second look before the route is complete.” (italic/cursive)

Bug Fixes - Address Autocomplete

- ☑ Fix Google Places Autocomplete to include complete address with state and ZIP code
- ☑ Ensure formatted_address is properly captured when user selects from dropdown
- ☑ Load Google Maps API globally so autocomplete works on all pages
- □ Fix z-index issue - dropdown appears but suggestions are not clickable
- □ Test in ContactEditDialog
- □ Test in RouteDetail Edit Address dialog

UI Updates - November 26, 2024

- ☑ Change “Missed Hops” header to “Mis-Hops” (play on mishaps)

Address Validation Feature - November 26, 2024

- ☑ Create backend tRPC procedure for address validation using Google Geocoding API

- ☑ Add “Validate Address” button next to address input in ContactEditDialog
- ☑ Add “Validate Address” button next to address input in RouteDetail Edit Address dialog
- ☑ Show validation results with toast messages (valid/suggestions/invalid)
- ☑ Allow user to accept suggested formatted address
- ☑ Add loading state to validation button
- ☑ Test address validation with various address formats (7/7 unit tests passing)

Bug Fixes - Address Validation (November 26, 2024)

- ☐ Fix “Use This” button in address validation toast - clicking it closes dialog instead of updating address field
- ☐ Ensure validated address fills the input field and keeps dialog open
- ☐ Test address validation flow in ContactEditDialog
- ☐ Test address validation flow in RouteDetail Edit Address dialog

Changed Addresses Report Feature (November 26, 2024)

Database Schema

- ☐ Add addressModifiedAt timestamp to cached_contacts table
- ☐ Add originalAddress field to cached_contacts table to track Google’s version
- ☐ Push database schema changes

Backend Implementation

- ☐ Create contacts.getChangedAddresses procedure to fetch contacts with modified addresses
- ☐ Return contact name, original address, new address, and modification date
- ☐ Add CSV export functionality for changed addresses report

Frontend Implementation

- ☐ Create Changed Addresses Report page/section
- ☐ Display table with: Contact Name, Original Address, New Address, Date Modified
- ☐ Add "Mark as Synced" toggle/button for each contact to remove from report after manual Google update
- ☐ Add "Export to CSV" button for manual Google Contacts sync
- ☐ Add navigation link to report from Settings or main menu
- ☐ Show count of changed addresses that need manual sync
- ☐ Filter out contacts marked as synced from the report

Testing

- ☐ Test that address changes are tracked with timestamps
- ☐ Test "Mark as Synced" removes contacts from report
- ☐ Test CSV export format is compatible with Google Contacts import
- ☐ Verify report shows only unsynced modified contacts

Document Upload for Labeled Contacts

- ☑ Add documents table to database schema (contactId, fileName, fileUrl, fileKey, uploadedAt, fileSize, mimeType)
- ☑ Create document upload UI component with drag-and-drop
- ☑ Implement S3 file upload for contact documents
- ☑ Add document list display on contact cards
- ☑ Add bulk document upload for contacts filtered by label (e.g., "Prairie PT")
- ☑ Add document download functionality
- ☑ Add document delete functionality
- ☑ Support multiple file types (PDF, Word, images, etc.)
- ☑ Add file size validation (max 10MB per file)

- ☑ Test document upload and retrieval

Email Reminders for Important Dates

- ☑ Add email notification settings to user preferences table
- ☑ Add scheduling email address field to settings
- ☑ Create email reminder configuration UI in Settings page
- ☑ Implement reminder intervals (30 days, 10 days, 5 days before, past due)
- □ Create scheduled job to check for upcoming important dates
- □ Implement email sending functionality using notification API
- □ Add email template for date reminders (License Renewal, etc.)
- □ Test email reminder system with different date scenarios
- □ Add ability to configure which date types trigger reminders
- □ Add reminder history/log for tracking sent notifications

Document Viewer in Contact Detail View

- ☑ Add Documents tab to contact detail view
- ☑ Display list of all uploaded documents for a contact
- ☑ Show file name, upload date, file size for each document
- ☑ Add download button for each document
- ☑ Add delete button for each document with confirmation
- ☑ Show empty state when no documents exist
- ☑ Add upload button in documents tab
- ☑ Test document viewer UI and functionality

Scheduled Reminder Job for Daily Email Checks

- ☑ Create scheduled job to run daily at 9 AM
- ☑ Job checks all users for upcoming important dates

- ☑ Send email reminders based on configured intervals
- ☑ Log successful and failed reminder sends
- ☑ Handle timezone considerations for scheduling
- ☑ Test scheduled job execution
- □ Verify emails are sent at correct intervals

Reminder History Log and Tracking System

- ☑ Create reminder_history table in database
- ☑ Track: contactId, dateType, reminderDate, sentAt, sentTo, status
- ☑ Add backend query to fetch reminder history
- ☑ Create "Sent Reminders" page in UI
- ☑ Display reminder history with filters (date range, contact, status)
- ☑ Show success/failure status for each reminder
- ☑ Add pagination for large reminder lists
- ☑ Test reminder history tracking and display

Configure Which Date Types Trigger Email Reminders

- ☑ Add enabledReminderDateTypes field to users table (JSON array)
- ☑ Update backend to filter reminders by enabled date types only
- ☑ Add UI in Settings to show checkbox list of all date types
- ☑ Allow user to toggle each date type on/off for reminders
- ☑ Default to all date types enabled for backward compatibility
- ☑ Test that only enabled date types trigger reminders

Reorganize Settings Page

- ☑ Group settings into logical sections with clear headers
- ☑ Site Configuration section (distance units, calling service, app preferences)

- ☑ Contacts section (important date types, comment options, email reminders)
- ☑ Routes section (starting points, stop types, stop duration, calendar integration)
- ☑ Improve visual hierarchy and spacing between sections

Tabbed Settings Layout

- ☑ Convert Settings page to use tabs instead of long scrolling page
- ☑ Create tabs: Account, Site Config, Contacts, Routes
- ☑ Use shadcn/ui Tabs component for clean implementation
- ☑ Ensure mobile responsiveness
- ☑ Test tab switching and content display

Email Template Editor

- □ Add email template fields to users table (subject, body for contact, body for team)
- □ Create email template editor UI in Settings → Contacts tab
- □ Support template variables like {{contactName}}, {{dateType}}, {{date}}, {{daysUntil}}
- □ Add emoji support in email templates
- □ Preview email template before saving
- □ Update email sending to use custom templates
- □ Provide default template with emojis

Emoji Support in Comments

- □ Add emoji picker component to comment fields
- □ Install emoji picker library (emoji-picker-react or similar)
- □ Add emoji button next to comment textarea
- □ Test emoji display in saved comments

- ☐ Ensure emojis work across all comment fields (contact comments, route notes, route comments)

Emoji Picker Widget

- ☑ Install emoji-picker-react package
- ☑ Create reusable EmojiPicker component wrapper
- ☑ Add emoji picker button to contact comment fields
- ☑ Add emoji picker button to route notes textarea
- ☑ Add emoji picker button to email template textareas
- ☑ Style emoji picker to match app theme
- ☑ Test emoji insertion in all fields

Email Template Preview

- ☑ Create EmailPreviewDialog component
- ☑ Add “Preview Email” button in Settings Contacts tab
- ☑ Generate sample email with placeholder data (sample contact, date, etc.)
- ☑ Show both contact email and team email versions in preview
- ☑ Apply template variables substitution in preview
- ☑ Style preview to look like actual email
- ☑ Test preview with different template configurations

Fix Kangaroo Crew Display Issue

- ☑ Fix “Bulk Upload Doc” button text getting cut off by card border
- ☑ Adjust button layout or card padding to prevent overflow
- ☑ Ensure all header buttons are fully visible
- ☑ Test on different screen sizes

Stage-Specific Email Templates

- ☑ Update database schema to store separate templates for each reminder stage
- ☑ Add fields for 30-day, 10-day, 5-day, and past-due templates (subject + body for contact and team)
- ☑ Update Settings UI to show tabbed or accordion interface for each stage
- ☑ Add emoji picker to all stage-specific template fields
- ☑ Update email sending logic to select correct template based on days until/past date
- ☑ Update email preview to show all four stage templates
- ☑ Provide default templates for each stage with appropriate tone/urgency
- ☑ Test that correct template is used for each reminder interval

Rescheduled Stops Management System

Calendar Integration

- ☑ Show rescheduled stops on calendar view as distinct event type
- ☑ Color code rescheduled stops differently from routes and Google events
- ☑ Display reschedule date/time on calendar grid
- ☑ Add click handler to view rescheduled stop details
- ☑ Group rescheduled stops by date on calendar

Create Route from Rescheduled Stops

- ☑ Add "Create Route from Selected" button to Mis-Hops page
- ☑ Add checkboxes to select individual rescheduled stops
- ☑ Implement backend procedure to create route from waypoint IDs
- ☑ Pre-populate route creation form with selected stops
- ☑ Clear reschedule status when stop is added to new route
- ☑ Show success message with link to new route

Reschedule History Tracking

- ☑ Create reschedule_history table to track all reschedule events
- ☑ Store: waypointId, originalDate, rescheduledDate, reason, status, createdAt
- ☑ Add backend procedures: logReschedule, getRescheduleHistory, updateRescheduleStatus
- ☑ Create RescheduleHistory page component
- ☑ Display all rescheduled stops with filters (pending, completed, re-missed)
- ☑ Show original route, contact name, reschedule reason, dates
- ☑ Add navigation link to RescheduleHistory page in header
- ☑ Export reschedule history to CSV

Testing

- ☑ Write tests for calendar integration with rescheduled stops
- ☑ Write tests for route creation from rescheduled stops
- ☑ Write tests for reschedule history logging
- ☑ Test complete workflow: miss stop → reschedule → create route → complete

Share Route Bug Fix

- ☑ Investigate share route 404 error
- ☑ Check if shared route URL pattern is registered in App.tsx
- ☑ Verify backend generateShareToken and getSharedRoute procedures
- ☑ Test share route generation and access
- ☑ Ensure share link works for public access (no auth required)

Kangaroo Crew Section Layout Update

- ☑ Make “Everyone you connect with along the journey” tagline span full width of section
- ☑ Adjust layout so tagline is not constrained by title column

Calendar Display Bug

- ☑ Investigate why calendar is not showing any events
- ☑ Check calendar.getEvents backend procedure
- ☑ Verify Calendar.tsx component rendering
- ☑ Check for console errors in browser
- ☑ Test with different date ranges

Kangaroo Crew Button Alignment

- ☑ Move action buttons (Refresh, Import CSV, Bulk Upload Doc) to the left
- ☑ Adjust button row layout to align with tagline instead of right-aligned

Rescheduled Stops Not Showing on Calendar

- ☑ Fix rescheduleWaypoint procedure to log to reschedule_history table
- ☑ Backfill existing rescheduled stops into history table
- ☑ Add “Mark as Completed” action to reschedule history page
- ☑ Add “Mark as Re-Missed” action to reschedule history page
- □ Fix calendar query timezone issue preventing rescheduled stops from appearing
- □ Test that rescheduled stops appear on calendar after fix

Rescheduled Stops Calendar Improvements

- □ Make rescheduled stops display as full event blocks on calendar (not small icons)
- □ Add title/contact name to rescheduled stop calendar events
- □ Ensure rescheduled stops have proper time display on calendar
- □ Sync rescheduled stops to Google Calendar when created

- ☐ Update Google Calendar event when reschedule status changes (completed/re-missed)
- ☐ Delete Google Calendar event when reschedule is cancelled
- ☐ Test that rescheduled stops appear as full events like routes
- ☐ Test Google Calendar sync for rescheduled stops

Account Consolidation & Admin User Management

Immediate Fix

- ☑ Transfer all routes from user 3060004 to user 38
- ☑ Transfer all contacts from user 3060004 to user 38
- ☑ Update reschedule_history records to use user 38
- ☑ Verify rescheduled stops appear on calendar after consolidation

Admin User Management Page

- ☑ Create AdminUsers page component
- ☑ Add backend procedure to list all users with stats (route count, contact count)
- ☑ Show current logged-in user indicator
- ☑ Add "Merge Users" functionality to combine duplicate accounts
- ☑ Add "Transfer Routes" functionality to move routes between users
- ☑ Add "Delete User" functionality (with confirmation)
- ☐ Add "View as User" functionality for debugging
- ☑ Add navigation link to Admin Users page in Settings
- ☑ Restrict access to admin role only

Testing

- ☑ Test account consolidation
- ☑ Test user merge functionality
- ☑ Test that calendar shows rescheduled stops after fix

- ☑ Verify both users see same data after consolidation

Google OAuth Verification

Testing Mode Setup (Immediate)

- ☐ Document how to add test users in Google Cloud Console
- ☐ Add team members’ email addresses as test users
- ☐ Verify test users don’t see “unverified app” warning
- ☐ Document testing mode limitations (100 users max)

Full Verification (Future)

- ☐ Create privacy policy page
- ☐ Create terms of service page
- ☐ Add homepage with app description
- ☐ Submit OAuth verification request to Google
- ☐ Respond to Google’s verification questions
- ☐ Wait for approval (1-2 weeks)

URGENT - Route Detail Page Error

- ☑ Fix “Route Not Found” error when clicking on routes
- ☑ Add missing columns to route_waypoints table
- ☑ Test route detail page loads correctly

Calendar Integration Issues

- ☑ Fix “No calendars found” error on calendar page
- ☑ Fix Connect Calendar button to use proper OAuth flow
- ☐ Test calendar connection and verify it works

- □ Verify calendar events display correctly

Email Reminder Template Editing

- ☑ Make template placeholders into actual editable default values
- ☑ Pre-fill templates with suggested text as default values
- ☑ Test template editing and saving

Contact Sync Issues - Fix Existing Refresh Button

- ☑ Investigate why existing Refresh button doesn’ t update contacts
- ☑ Fix refresh to properly update label names (*Abundant, etc.)
- ☑ Fix refresh to pull in missing contacts (ABUNDANT Offices, Oscar RN)
- ☑ Debug why refresh still doesn’ t work after publishing
- ☑ Found issue: getGoogleAuthUrl was using old state format
- ☑ Fixed getGoogleAuthUrl to use JSON state format

Multi-User Platform Transformation

Landing Page

- □ Create public landing page component (LandingPage.tsx)
- □ Add hero section with kangaroo branding and tagline
- □ Add features section highlighting key capabilities
- □ Add “Sign Up with Google” and “Login” buttons
- □ Add footer with links and branding
- □ Design friendly/casual style matching Routie Roo theme

Authentication & User Management

- □ Update routing to show landing page for non-authenticated users

- □ Keep existing Google OAuth flow for user registration
- □ Ensure each user’ s data is isolated (contacts, routes, settings)
- □ Verify admin role is preserved for owner
- □ Test new user sign-up flow

Admin Dashboard

- □ Create AdminUsers page (already exists, verify functionality)
- □ Show list of all registered users
- □ Display user stats (contact count, route count, last active)
- □ Add ability to view user details
- □ Add ability to disable/enable user accounts (optional)

Data Isolation & Testing

- □ Verify contacts are user-specific
- □ Verify routes are user-specific
- □ Verify settings are user-specific
- □ Test with multiple user accounts
- □ Ensure admin can see all users but regular users only see their data

Deployment

- □ Save checkpoint after landing page
- □ Save checkpoint after auth updates
- □ Save final checkpoint
- □ Test on published site with new user registration

Multi-User SaaS Platform Transformation

Public Landing Page

- ☑ Create LandingPage.tsx component with kangaroo branding
- ☑ Add hero section with tagline “Hop Through Your Day with Smart Route Planning”
- ☑ Add features grid showcasing route planning, contacts, calendar, reminders
- ☑ Add benefits section highlighting time savings and free access
- ☑ Add CTA section with “Sign Up with Google” button
- ☑ Add footer with branding
- ☑ Update App.tsx routing to show landing page for non-authenticated users
- ☑ Update App.tsx to show dashboard only for authenticated users
- ☑ Add loading spinner while checking authentication
- ☑ Keep shared route execution accessible without login

User Registration & Authentication

- □ Verify Google OAuth allows new user registration
- □ Test new user sign-up flow
- □ Confirm new users get default role of “user”
- □ Verify Angela (angelaneason@gmail.com) retains admin role
- □ Test that new users start with empty contact/route data

Data Isolation

- □ Verify all contact queries filter by userId
- □ Verify all route queries filter by userId
- □ Verify all settings queries filter by userId
- □ Test that User A cannot see User B’s contacts
- □ Test that User A cannot see User B’s routes

- □ Test that User A cannot access User B’ s route detail pages
- □ Verify shared routes work across users (public access)

Admin Dashboard

- ☑ AdminUsers page already exists with user list
- ☑ Admin can see all users with stats (routes, contacts, last login)
- ☑ Admin can merge duplicate user accounts
- ☑ Admin can delete user accounts
- □ Test admin dashboard shows all registered users
- □ Verify only admin role can access /admin/users
- □ Test user merge functionality
- □ Test user delete functionality

Testing & Validation

- □ Create test user account via Google OAuth
- □ Verify test user sees landing page before login
- □ Verify test user can sign up and access dashboard
- □ Verify test user has isolated data workspace
- □ Verify Angela can access admin dashboard
- □ Verify non-admin users cannot access admin dashboard
- □ Test multi-user scenarios (2+ users with separate data)

Documentation

- □ Update README with multi-user platform description
- □ Document admin features for Angela
- □ Create user onboarding guide
- □ Document data isolation architecture

Calendar View Update

- ☑ Change default calendar view from Day to Month

Calendar Event Dialog Colors

- ☑ Update event popup dialog to display events with their calendar colors

Calendar Color Mapping Fix

- ☑ Map Google Calendar color IDs to actual hex color values
- ☑ Update backend to include calendar backgroundColor in event data
- ☑ Display events with proper vibrant colors in popup dialog

Stop Types Bug Fix

- ☑ Fix "Cannot read properties of undefined (reading 'contactName')" error when planning routes with stop types
- ☑ Ensure stop types are properly saved when creating routes
- ☑ Test route creation with different stop type selections

Stop Type Selection and Display Issues

- ☑ Fix stop type dropdown - unable to change from "other" default
- ☑ Display stop types in Hop-By-Hop Navigation view
- ☑ Test stop type selection and persistence

Stop Type Map Icons

- ☑ Create custom map marker icons for each stop type
- ☑ Update RouteDetail map to use stop type icons

- ☑ Test icon visibility and differentiation on map

Stop Type Selection Bug

- ☑ Fix stop type selector reverting to “other” instead of saving selected type
- ☑ Verify stop type state is properly updated in contactStopTypes Map
- ☑ Test that selected stop types persist when creating routes

Custom Stop Type Colors Bug

- ☑ Fix stop type selector showing blue dots for all custom stop types
- ☑ Ensure custom stop type colors from database are properly displayed
- ☑ Update color matching logic to work with custom stop type names
- ☑ Fix stop type name not displaying in selector (only showing dot)

Default Stop Type Feature

- ☑ Add defaultStopType field to users table in database schema
- ☑ Add defaultStopTypeColor field to users table
- ☑ Create backend procedure to get/update default stop type preference
- ☑ Add default stop type dropdown to Settings page (Routes tab)
- ☑ Update route creation to automatically assign default stop type to all contacts
- ☑ Allow changing stop type for individual waypoints in route details screen (already exists)
- ☑ Test end-to-end: Set default in Settings → Create route → Verify all stops use default → Change individual stop types in route details

Route Creation Validation Bug

- ☑ Fix route.create procedure to accept custom stop type names (currently only accepts hardcoded enum: pickup, delivery, meeting, visit, other)
- ☑ Change stopType validation from enum to string to support user’ s custom stop types
- ☑ Fix addWaypoint procedure validation as well
- ☑ Remove stop type selector UI from route builder (redundant since we have default + can edit in route details)
- ☑ Change database stopType column from enum to varchar(100)
- ☑ Test route creation with custom stop types like “Visit” , “Eval” , “OASIS”

Stop Type Display and Editing Issues

- ☑ Add edit button to waypoint cards for comprehensive editing (stop type, contact name, etc.)
- ☑ Create edit dialog with stop type selector and other editable fields
- ☑ Move stop type display to be more prominent - show before patient name or on first line
- ☑ Add backend mutation for updating waypoint stop type

Expand Edit Details Dialog

- ☑ Add address field to Edit Details dialog
- ☑ Add phone numbers editing to Edit Details dialog
- ☑ Update backend mutation to handle address and phone number updates
- □ Remove redundant “Edit Address” button since it’ s now in Edit Details (keep for now - separate workflow)
- ☑ Test comprehensive waypoint editing (stop type, name, address, phone)

Automatic Google Contact Sync

- ☑ Research Google People API for updating contact details (addresses, phoneNumbers, memberships supported)
- ☑ Add labels/groups field to Edit Details dialog UI
- ☑ Fetch user' s contact groups/labels for selection (user can type labels manually)
- ☑ Add labels to updateWaypointDetails backend mutation
- ☑ Add contactId field to routeWaypoints schema for tracking Google contacts
- ☑ Update route creation to store contactId when adding waypoints
- ☑ Implement Google People API sync helper function (stub created, needs OAuth token integration)
- ☑ Call sync helper from updateWaypointDetails mutation
- ☑ Handle cases where contact doesn' t exist in Google (skip sync gracefully)
- ☑ Add error handling for Google API failures (don' t block waypoint update)
- □ Implement OAuth token storage and retrieval for Google sync
- □ Implement actual Google People API updateContact call
- □ Test label editing and automatic sync end-to-end

Complete Google Contact Sync Implementation

- ☑ Add OAuth token fields to users table (accessToken, refreshToken, tokenExpiry)
- ☑ Update OAuth callback to store tokens when user logs in
- ☑ Implement token refresh logic for expired tokens
- ☑ Complete syncToGoogleContact helper with actual Google People API calls
- ☑ Fetch current contact from Google to get etag
- ☑ Build proper update request with field masks
- ☑ Test full sync flow: edit waypoint → updates Google Contact (ready for user testing)

- ☐ Note: Label sync requires mapping label names to contact group resource names (future enhancement)

Add Google Sync to Kangaroo Crew Contact Editor

- ☑ Update contacts.update procedure to call syncToGoogleContact
- ☑ Pass contactId, userId, address, and phoneNumbers to sync helper
- ☑ Test editing contact in Kangaroo Crew → verify Google Contact updates

Calendar Integration Improvements

- ☑ Remove starting point from calendar events (only add actual stops)
- ☑ Change event titles from "Stop 1: Name" to "Stop Type: Name" (e.g., "Eval: John Smith")
- ☑ Add updateCalendarEvent function to googleAuth.ts for Google Calendar API
- ☑ Create backend calendar.updateEvent mutation with access token handling
- ☑ Add Edit button to event detail dialog in Calendar page
- ☑ Create EditEventDialog component with fields for title, start/end time, location, description
- ☑ Wire up mutation to EditEventDialog and refresh events after update
- ☑ Test: Create route → Add to calendar → Edit event → Verify Google Calendar updates

Automatic Calendar Sync for Routes

Database Schema

- ☑ Add calendarEventId field to route_waypoints table (VARCHAR storing Google Calendar event ID)
- ☑ Update routes table to track calendar sync status (already has googleCalendarId)

Backend Implementation

- ☑ Store calendar event IDs when creating calendar events (map waypoint to event ID)
- ☑ Create syncRouteToCalendar mutation to handle route changes (integrated into add/remove/update mutations)
- ☑ Detect waypoint additions and create new calendar events
- ☑ Detect waypoint removals and delete corresponding calendar events from Google
- ☑ Detect waypoint modifications (address, time, stop type) and update calendar events
- ☑ Handle token refresh for calendar API calls
- ☑ Add error handling for partial sync failures

Frontend Implementation

- ☑ Add "📅 On Calendar" badge to route cards when calendarId exists
- ☑ Add calendar indicator to route details page header
- ☑ Automatically trigger calendar sync when route is edited (integrated into mutations)
- ☑ Show sync status with loading indicator (handled by tRPC mutation states)
- ☑ Display success/error messages for sync operations (console logs for debugging)
- □ Update "Delete Calendar Events" to also clear event IDs from waypoints (future enhancement)

Testing

- ☑ Test: Database schema supports calendarEventId field
- ☑ Test: Calendar event IDs can be stored and updated
- ☑ Test: Route tracks calendar sync status
- ☑ Test: Visual indicator appears on routes with calendar events

- □ Manual test: Add waypoint to route with calendar → Verify new event in Google Calendar
- □ Manual test: Remove waypoint from route with calendar → Verify event deleted
- □ Manual test: Edit waypoint address → Verify calendar event location updates
- □ Manual test: Edit waypoint stop type → Verify calendar event title updates

Calendar Event Editing Issues

- ☑ Fix: Individual calendar events cannot be clicked (only "more" button shows events)
- ☑ Add: Individual event click handler to open event detail dialog
- ☑ Add: Edit button in event detail dialog for Google Calendar events (already exists)
- ☑ Ensure: Edit functionality works for non-route calendar events
- ☑ Test: Click individual event → Opens detail dialog → Edit button appears → Can edit and save

Calendar Event Editing Bug

- ☑ Fix: "Missing Event Information" error when clicking calendar events to edit
- ☑ Investigate: Event data structure mismatch between Calendar.tsx and EditEventDialog
- ☑ Ensure: Event object has all required fields (id, googleEventId, calendarId, etc.)
- ☑ Test: Click event → Edit dialog opens with pre-filled data → Can save changes

Calendar Event Update DateTime Bug

- ☑ Fix: "timeRangeEmpty" error when updating calendar events
- ☑ Investigate: Datetime formatting in EditEventDialog and updateCalendarEvent

- ☑ Ensure: Datetime strings are properly formatted for Google Calendar API (RFC3339 with Z suffix)
- ☑ Test: Edit event times → Save → Verify update succeeds

Calendar Event Title Format

- □ Change calendar event titles from "Route - Stop 1: Name" to "Route - StopType: Name"
- □ Update createWaypointEvents function to use stopType instead of stop number
- □ Capitalize stop type for display (eval → Eval, visit → Visit, etc.)
- □ Test: Add route to calendar → Verify event titles show stop types

Remove from Calendar Button Bug

- ☑ Fix: "Remove from Calendar" button does nothing when clicked
- ☑ Investigate: Check if deleteCalendarEvents mutation exists and is wired correctly
- ☑ Ensure: Button triggers mutation and shows loading/success states
- ☑ Update: clearCalendarEvents now deletes events from Google Calendar API
- ☑ Update: Clear calendarEventId from all waypoints when removing from calendar
- ☑ Test: Click remove button → Events deleted from Google Calendar → UI updates

Standalone Mobile-First Rebuild (Market Launch Priority)

Phase 1: Remove Google Dependency - Core Contact Management

- □ Create manual contact entry form (name, phone, email, address)

- □ Build contact list view (works without Google Contacts)
- □ Add contact editing functionality
- □ Add contact deletion with confirmation
- □ Implement contact search (by name, phone, address)
- □ Add contact photo upload (local file upload)
- □ Build CSV contact import (upload CSV file with contacts)
- □ Add contact export to CSV
- □ Create contact validation (ensure required fields)
- □ Add address autocomplete using Google Maps Autocomplete API (no auth required)
- □ Test contact management flow end-to-end

Phase 2: Remove Google Dependency - Local Calendar System

- □ Create calendar_events table in database (id, userId, title, startTime, endTime, location, description, type)
- □ Build manual event creation form
- □ Add event editing functionality
- □ Add event deletion with confirmation
- □ Create calendar view (month/week/day) showing local events
- □ Integrate route events into local calendar
- □ Add iCal export for events (users can import to any calendar app)
- □ Add CSV export for events
- □ Remove dependency on Google Calendar API for event storage
- □ Make Google Calendar sync optional (keep as enhancement feature)
- □ Test local calendar system end-to-end

Phase 3: Mobile-First UI Redesign

- □ Audit all pages on mobile (iPhone, Android)
- □ Redesign homepage for mobile (touch-friendly navigation)

- □ Redesign contact list for mobile (card layout, large touch targets)
- □ Redesign route creation for mobile (step-by-step wizard)
- □ Redesign route detail for mobile (collapsible sections)
- □ Redesign calendar for mobile (swipe navigation)
- □ Redesign settings for mobile (accordion layout)
- □ Add mobile-friendly map controls
- □ Implement bottom navigation bar for mobile
- □ Add pull-to-refresh on list views
- □ Test all touch interactions (tap, swipe, pinch-zoom)
- □ Optimize for small screens (320px width minimum)

Phase 4: Simplified Onboarding (No Google Required)

- □ Remove Google OAuth requirement for sign-up
- □ Create simple email/password registration
- □ Build "Getting Started" wizard (no Google steps)
  - Step 1: Add your first contact
  - Step 2: Create your first route
  - Step 3: Explore calendar and execution
- □ Add sample data option (pre-populate with demo contacts/routes)
- □ Create interactive tutorial overlays
- □ Add skip option for users who want to dive in
- □ Test onboarding with 10 new users (target < 5 min to first route)

Phase 5: Offline Mode (Critical for Field Workers)

- □ Implement service worker for offline support
- □ Cache route data in IndexedDB
- □ Cache contact data in IndexedDB
- □ Allow route execution offline (mark stops complete/missed)
- □ Queue offline changes for sync when online

- □ Add offline indicator banner
- □ Test offline → online transition
- □ Ensure maps work offline (cached tiles)

Phase 6: Performance Optimization for Mobile

- □ Reduce initial bundle size (< 300KB gzipped)
- □ Implement lazy loading for routes
- □ Implement lazy loading for contacts
- □ Add image compression for contact photos
- □ Optimize map rendering for mobile
- □ Add loading skeletons for all pages
- □ Test on slow 3G connection
- □ Test on low-end Android device
- □ Measure and optimize Time to Interactive (< 3 seconds)

Phase 7: Mobile-Specific Features

- □ Add “Call” button with tel: link (no Google Voice dependency)
- □ Add “Text” button with sms: link
- □ Add “Navigate” button with maps: link (opens native maps app)
- □ Add “Share Route” via native share API
- □ Implement geolocation for “Use Current Location” as starting point
- □ Add camera integration for contact photos
- □ Support mobile notifications (route reminders)
- □ Add home screen install prompt (PWA)

Phase 8: Optional Google Integration (Enhancement, Not Requirement)

- □ Move Google Calendar sync to “Integrations” settings
- □ Move Google Contacts sync to “Integrations” settings

- □ Make both integrations clearly optional
- □ Add "Connect Google" buttons (not required for core functionality)
- □ Show benefits of connecting Google (automatic sync)
- □ Allow disconnecting Google without losing data
- □ Test app works fully without any Google connection

Phase 9: Alternative Authentication

- □ Implement email/password authentication
- □ Add "Forgot Password" flow
- □ Add email verification
- □ Add social login options (Apple, Facebook) as alternatives to Google
- □ Remove Google OAuth as only login method
- □ Test authentication flow on mobile

Phase 10: Data Portability

- □ Build comprehensive CSV export (all contacts, routes, events)
- □ Build CSV import for contacts
- □ Build CSV import for routes (if user has data from other tools)
- □ Add "Download All My Data" button (GDPR compliance)
- □ Create data backup/restore functionality
- □ Test import/export with large datasets

Testing & Launch Preparation

- □ Test complete flow on iPhone (iOS Safari)
- □ Test complete flow on Android (Chrome)
- □ Test on tablet (iPad, Android tablet)
- □ Test with 1000+ contacts (performance)
- □ Test with 100+ routes (performance)
- □ Test offline mode thoroughly

- □ Conduct user testing with 10 field workers
- □ Fix all critical bugs found in testing
- □ Create mobile-optimized screenshots for marketing
- □ Update user manual for standalone version

Documentation Updates

- □ Rewrite user manual to remove Google dependency mentions
- □ Create "Import Contacts from CSV" guide
- □ Create "Export to Your Calendar App" guide
- □ Update marketing copy to emphasize "No Google Required"
- □ Create comparison chart: Standalone vs Google-Connected modes

Hybrid Architecture: Support Both Google & Standalone Modes

Phase 1: Dual Authentication System

- □ Keep existing Google OAuth login
- □ Add email/password authentication as alternative
- □ Create unified user model (supports both auth methods)
- □ Add "Sign up with Email" and "Sign up with Google" options on registration
- □ Allow users to link Google account to existing email account later
- □ Test both authentication flows

Phase 2: Hybrid Contact Management

- □ Keep existing Google Contacts sync functionality
- □ Add manual contact entry form (for non-Google users)
- □ Add CSV contact import (for non-Google users)

- □ Show “Import from Google Contacts” button for Google-authenticated users
- □ Show “Add Manually” and “Import CSV” buttons for all users
- □ Store all contacts in local database (regardless of source)
- □ Add contact_source field (google, manual, csv) to track origin
- □ Enable editing locally-stored Google contacts
- □ Add optional Google Contacts sync toggle in settings
- □ Test contact management for both user types

Phase 3: Hybrid Calendar System

- □ Keep existing Google Calendar integration
- □ Create local calendar_events table for standalone events
- □ Show “Add to Google Calendar” button (only for Google-authenticated users)
- □ Show “Add to Local Calendar” button (for all users)
- □ Display both Google and local events in calendar view
- □ Add iCal export for local events
- □ Add “Sync to Google Calendar” toggle in settings (for Google users)
- □ Allow Google users to choose: local-only, Google-only, or both
- □ Test calendar functionality for both user types

Phase 4: Settings & User Preferences

- □ Add “Account Type” indicator in settings (Google-connected vs Standalone)
- □ Add “Connect Google Account” button for standalone users
- □ Add “Disconnect Google Account” button for Google users (with warning)
- □ Show Google-specific settings only to Google-connected users
- □ Add “Data Source” preferences (prefer Google vs prefer local)
- □ Create migration flow: standalone → Google-connected
- □ Create migration flow: Google-connected → standalone

- □ Test account type switching

Phase 5: UI Adaptation Based on Account Type

- □ Show/hide Google-specific buttons based on authentication method
- □ Display “Connect Google for automatic sync” banner for standalone users
- □ Add “Benefits of Connecting Google” tooltip/modal
- □ Show contact source badges (Google icon vs manual icon)
- □ Adapt onboarding wizard based on chosen authentication method
- □ Test UI for both user types

Phase 6: Data Sync Logic

- □ Implement contact sync: Google → Local (for Google users)
- □ Implement calendar sync: Local → Google (optional for Google users)
- □ Add conflict resolution (last-write-wins or user-choice)
- □ Add sync status indicators (“Synced”, “Syncing”, “Sync Failed”)
- □ Add manual “Sync Now” button
- □ Handle Google token expiration gracefully
- □ Test sync reliability

Phase 7: Mobile Optimization (Universal)

- □ Ensure mobile UI works for both user types
- □ Test onboarding on mobile for email/password flow
- □ Test onboarding on mobile for Google OAuth flow
- □ Optimize contact import (CSV) for mobile file upload
- □ Test all features on iPhone and Android for both modes

Phase 8: Documentation & Marketing

- □ Update user manual to explain both modes
- □ Create “Standalone vs Google-Connected” comparison chart

- □ Add FAQ: “Do I need a Google account?”
- □ Create tutorial: “How to connect Google later”
- □ Create tutorial: “How to import contacts without Google”
- □ Update marketing copy: “Works with or without Google”

Route Detail Screen Enhancements

Stop Reordering Improvements

- ☑ Remove drag-and-drop interface for stop reordering
- ☑ Add stop number input field for each waypoint
- ☑ Allow users to type new position number to reorder
- ☑ Add save button to persist manual order
- ☑ Add hasManualOrder flag to routes table
- ☑ Add reorderWaypoints backend procedure
- ☑ Fix TypeScript errors in RouteDetail
- □ Modify re-optimize to preserve manual order (only optimize new stops)

Visual Enhancements

- □ Add contact icons to route detail waypoints (matching contact list)
- ☑ Display contact labels/groups on each stop (already implemented in SortableWaypointItem lines 84-99)
- □ Show visual indicators for contacts without addresses
- □ Ensure consistent styling between contact list and route detail

Gap Stop Feature

- ☑ Add isGapStop boolean field to route_waypoints table
- ☑ Add gapDuration field to route_waypoints table (in minutes)
- ☑ Add gapDescription field to route_waypoints table
- □ Create “Add Gap Stop” button in route detail

- ☐ Create gap stop dialog (description, duration, position)
- ☐ Display gap stops differently from contact stops
- ☐ Include gap stop duration in calendar time calculations
- ☐ Exclude gap stops from contact-based operations

Calendar Visual Indicator

- ☐ Add calendar icon/badge to routes that have been added to calendar
- ☐ Show calendar status in route list (Home page)
- ☐ Show calendar status in route detail header
- ☐ Use different visual for routes with scheduled dates vs calendar events

Backend Updates

- ☐ Update route creation to support gap stops
- ☐ Update re-optimize logic to skip gap stops
- ☐ Update waypoint reordering to handle gap stops
- ☐ Add validation for gap stop data

Bug Fix - Map Markers Disappearing on Reorder

- ☑ Fix map markers disappearing when changing stop order with number inputs
- ☑ Ensure map updates when localWaypoints state changes
- ☑ Verify map markers persist after reordering

Bug Fix - Contact Photos Not Displaying on Waypoints

- ☐ Verify photoUrl is being fetched from cached_contacts
- ☐ Check if photoUrl is being passed when creating routes
- ☐ Verify photoUrl is stored in route_waypoints table
- ☐ Debug SortableWaypointItem photo rendering logic

- ☐ Test with actual Google contact photos

UI/UX Improvements - User Requested

- ☑ Exclude starting point from calendar events (it’ s just an anchor)
- ☑ Fix route notes section not working at bottom of Route Details (route_notes table was missing, now created)
- ☑ Remove “Copy Link” button (doesn’ t work, duplicate of Share)
- ☑ Move route name from screen header into route details card
- ☑ Move contact labels to display under contact name
- ☑ Change “Calendar” button text to “Add To Calendar”

Gap Stop Feature - In Progress

- ☑ Create “Add Gap Stop” button in route detail page
- ☑ Build gap stop creation dialog (name, duration, description)
- ☑ Add backend mutation to create gap stops (waypoints with isGapStop=true)
- ☑ Display gap stops in waypoint list with distinct styling (clock icon, no contact info)
- ☑ Add gap stop icon/marker on map (happy emoji with clock)
- ☑ Calendar logic: Skip gap stops as events but add their duration to timing calculations
- ☑ Allow reordering gap stops with regular stops (drag-and-drop works)
- ☑ Test gap stop creation, display, and calendar timing

Gap Stop UI Refinement - Complete

- ☑ Hide Complete/Miss/Add Note buttons for gap stops (they’ re time blocks, not contact stops)
- ☑ Keep only Edit Details and Remove buttons for gap stops
- ☑ Test gap stop action buttons display correctly

Gap Stop Visual Update - Complete

- ☑ Replace clock icon with cute emoji image in stop list
- ☑ Test gap stop display with new image

Gap Stop Position Control - Complete

- ☑ Add “Insert After Stop #” field to gap stop dialog
- ☑ Update addGapStop mutation to accept insertAfterPosition parameter
- ☑ Update backend logic to insert gap stop at specified position
- ☑ Test gap stop insertion at various positions

Gap Stop Map Display - Complete

- ☑ Removed gap stop markers from map (they’re time blocks, not physical locations)
- ☑ Gap stops now only appear in stop list with cute emoji
- ☑ Map shows only physical stops for cleaner visualization

Bug Fixes - Complete

- ☑ Add validation message when attempting to add contact without address (already implemented)
- ☑ Fix stop type colors in shared route view (now using waypoint.stopColor)
- ☑ Fix missing Google profile photos in shared route view (now displaying with stop number badge)
- ☑ Test all fixes

PhotoUrl Backfill - Complete

- ☑ Create backfill script to update existing waypoints with contact photoUrls

- ☑ Run backfill script on all existing routes (updated 9 waypoints, 24 total with photos)
- ☑ Verify photoUrls are populated in route waypoints
- ☑ Ensure addWaypoint mutation saves photoUrl for new waypoints

PhotoUrl Fix - Complete

- ☑ Verified stop types display correctly (Visit in blue)
- ☑ Identified root cause: waypoints missing contactId field
- ☑ Created improved backfill script that matches by contact name
- ☑ Backfilled 96 waypoints with photoUrl and contactId
- ☑ Verified Home page already saves contactId for new routes
- ☑ Confirmed route 390001 now has photos

Stop Type Color Fix - Complete

- ☑ Investigated why waypoints show default blue color instead of custom stop type colors
- ☑ Found old routes were created before custom stop types were set up
- ☑ Created update script to map stop types to user’s custom colors
- ☑ Updated 81 waypoints with correct custom colors
- ☑ Verified route 390001 now displays yellow for “Visit” stop type

Route Copy Color Fix - Complete

- ☑ Investigated route copy logic (routes.copyRoute procedure)
- ☑ Found copyRoute was missing stopColor, photoUrl, and contactId fields
- ☑ Fixed route copy to preserve stopColor, photoUrl, and contactId from original
- ☑ Verified addWaypoint already includes stopColor for new stops
- □ Test that newly copied routes maintain custom colors

Color Legend Feature - Complete

- ☑ Designed color legend component for route detail page
- ☑ Extract unique stop types from route waypoints (excluding starting point and gap stops)
- ☑ Display legend with stop type names and color circles
- ☑ Positioned legend in Route Details card after notes section
- ☑ Legend only shows if route has actual stop types

Scheduler Sticky Notes - Complete

- ☑ Created scheduler_notes table in database schema
- ☑ Added userId, noteText, isCompleted, createdAt, completedAt fields
- ☑ Pushed database schema changes via SQL
- ☑ Created backend procedures (list, create, toggleComplete, delete)
- ☑ Copied pushpin graphic to client/public directory
- ☑ Designed sticky note UI component with pushpin graphic
- ☑ Added toggle to expand/collapse notes
- ☑ Added note input with Enter key support
- ☑ Implemented checklist with checkboxes
- ☑ Added delete button for each note
- ☑ Separated pending and completed notes sections
- ☑ Added sticky note component to home page (centered at top)
- ☑ Made sticky note draggable with mouse and touch
- ☑ Positioned sticky note at top right on desktop, centered on mobile
- ☑ Added drag handle with grip icon
- ☑ Added mobile touch support (touchstart, touchmove, touchend)
- ☑ Prevents scrolling while dragging on mobile
- ☑ Responsive width for mobile screens

- ☑ Notes are private (stored per user, not shared with routes)

Sticky Note Redesign - Complete

- ☑ Use sticky note graphic as background image
- ☑ Overlay reminder text directly ON the sticky note graphic
- ☑ Styled text with cursive font to look handwritten
- ☑ Text appears on the pink/coral sticky note paper area
- ☑ Entire sticky note (including pushpin) is draggable
- ☑ Maintained all functionality (add, complete, delete, toggle)
- ☑ Compact design to fit on sticky note dimensions

Sticky Note Improvements - Complete

- ☑ Moved input field down to avoid pushpin overlap (pt-[110px])
- ☑ Made all note text bold for better readability
- ☑ Added resize functionality (drag corner resize handle)
- ☑ Minimum size constraint (250x250px)
- ☑ Resize works on both desktop and mobile
- ☑ Resize handle with maximize icon in bottom-right corner

Sticky Note Redesign with Pushpin - Complete

- ☑ Copied yellow pushpin graphic to public folder
- ☑ Redesigned sticky note with light purple background (#e9d5ff)
- ☑ Positioned pushpin at top center (draggable)
- ☑ No overlap - content starts below pushpin with proper padding
- ☑ Added scrollable area for unlimited notes
- ☑ Kept resize functionality (250-500px width, 300-700px height)
- ☑ Each note has white/transparent background for readability

- ☑ Resize handle with grip icon in bottom-right

Sticky Note Collapse Fix - Complete

- ☑ Make collapsed state shrink to minimal height (50px)
- ☑ Add smooth transition animation when collapsing/expanding
- ☑ Hide resize handle when collapsed
- ☑ Ensure add reminder input works when expanded

Sticky Note Mobile Layout Issue

- ☑ Fix sticky note covering page content when expanded on mobile
- ☑ Adjust z-index or positioning to prevent layout interference
- ☑ Consider making sticky note collapsible by default on mobile
- ☑ Test sticky note doesn’ t block important UI elements

Sticky Note Horizontal Scroll Issue

- ☑ Hide sticky note completely on mobile (< 768px) by default
- ☑ Add floating toggle button on mobile to show/hide sticky note
- ☑ Ensure no horizontal scroll when sticky note is hidden
- ☑ Make sticky note overlay content (not push layout) when shown on mobile
- ☑ Test that mobile layout stays normal width without sticky note

Mobile Viewport Width Issue

- ☑ Fix screen being too wide on mobile
- ☑ Fix stretched logo on mobile
- ☑ Fix calendar layout being misaligned
- ☑ Ensure sticky note toggle button is visible and accessible

- ☑ Add proper viewport constraints to prevent horizontal overflow
- ☑ Test all header elements fit within mobile viewport

Remaining Mobile Width Issues

- ☑ Fix screen still being too wide on mobile (cannot see full content)
- ☑ Fix calendar page - stuck on calendar list with no way to scroll or exit
- ☑ Add back button or navigation on calendar page for mobile
- ☑ Identify and fix elements causing horizontal overflow
- ☑ Ensure all pages have proper max-width constraints
- ☑ Test navigation between pages on mobile

Home Screen Width Issue

- ☑ Fix home screen still being too wide on mobile
- ☑ Identify elements causing horizontal overflow on home page
- ☑ Make route planning cards responsive
- ☑ Ensure hop library fits within mobile viewport
- ☑ Test all home page sections on mobile

Contact Label Badge Layout

- ☑ Make contact label badges span full width on mobile
- ☑ Change label badges from inline to block/full-width layout
- ☑ Test label display on mobile

Label Text Wrapping Fix

- ☑ Prevent label text from wrapping to multiple lines
- ☑ Make labels display on single line (e.g., “PT Randy Harms” on one line)

- ☑ Test on mobile

Sticky Note Desktop Toggle

- ☑ Make sticky note hidden by default on desktop (like mobile)
- ☑ Show floating pushpin button on desktop
- ☑ Test toggle functionality on desktop

Pushpin Button Visibility Issue

- ☑ Fix pushpin button not appearing on desktop
- ☑ Ensure button is always visible regardless of sticky note state
- ☑ Test button visibility on desktop

Contact Labels Not Showing on Route Details

- ☑ Investigate why labels are not displaying on waypoint cards
- ☑ Check if contactLabels field is being passed to waypoint component
- ☑ Fix label rendering in SortableWaypointItem component (extract from contactGroups/ format)
- □ Test label display on route details page
- □ Verify labels are being saved when creating new routes
- □ Check if existing routes need label data migration

Mobile Label Truncation Issue

- ☑ Remove text truncation (whitespace-nowrap, text-ellipsis) from mobile contact labels
- ☑ Extract full label names from contactGroups/ format ("R. Harms" instead of "PT R Harms")
- ☑ Fix label container to allow text wrapping like phone numbers

- □ Test label display on mobile devices

Mobile Route Detail Text Overflow

- ☑ Fix text running off page on mobile route detail view
- ☑ Add proper text wrapping and responsive layout (header and action buttons)
- □ Test on mobile devices

Stop Type Colors Not Applied

- ☑ Investigate where stop types are assigned during route creation - stop types were deleted
- ☑ Fix to use user’ s custom stop type colors instead of hardcoded defaults
- ☑ Created “Visit” stop type with black color (#000000)
- ☑ Set “Visit” as default stop type in settings
- □ Test creating new route to verify black color is applied

Duplicate Contact Issue

- □ Investigate Ashley Marston appearing under wrong label (Prairie PT instead of correct label)
- □ Check if contact ID 750664c109a2dd83 has correct labels in database
- □ Verify label filtering logic is working correctly
- □ Fix any data inconsistencies

Label Color Assignment Feature

- ☑ Add database table for label colors
- ☑ Create API endpoints for managing label colors
- ☑ Build label color settings UI
- ☑ Apply label colors to contact cards

☑ Apply label colors to map markers (dual-color design: label color center + stop type border)

Label Colors Bug Fixes

☑ Filter out system-generated Google Contacts group identifiers (contactGroups/*) from Label Colors Settings display

Label Legend Feature

☑ Add label color legend to route detail page (alongside stop type legend)

Archive Filter Enhancement

☑ Filter archived routes from main "Your Hop Library" display (show only in Archive section)

Archive Enhancements

☑ Add unarchive functionality to restore routes from Archive section (already exists)

☑ Add confirmation dialog before archiving routes

☑ Display archive date in Archive section (already exists)

☑ Add delete button with confirmation to archive screen

UI Layout Improvements

- [x] Make "Your Hop Library" section match height of "Plan Your Next Hop" section
- [x] Increase height of "Plan Your Next Hop" section to better balance with "Your Hop Library"

Multiple Contacts Per Route Setting

- ☑ Add allowMultipleVisits boolean field to users table
- ☑ Add setting toggle in Settings page (Routes tab)
- ☑ Update route creation to check allowMultipleVisits setting
- ☑ Show/hide duplicate contact warning based on setting
- ☑ Test enabling and disabling the setting

Missing Coordinates Validation

- ☑ Add validation during route creation to detect waypoints with missing coordinates
- ☑ Show warning message in route detail page if any stops have null lat/lng
- ☑ Display which specific stops are missing coordinates with visual indicator
- ☑ Show prominent warning toast when creating routes with missing coordinates

Route UI Updates

- ☑ Add copy icon button to each route card in the library
- ☑ Remove “Copy Route” button from top action bar in RouteDetail
- ☑ Remove Export button from route detail page

Waypoint Address Update Fix

- ☑ Update waypoint edit mutation to geocode address when changed
- ☑ Save latitude/longitude coordinates from geocoding result
- ☑ Clear missing coordinates warning after successful update
- ☑ Test that updated waypoint appears on map immediately

Two-Way Google Contacts Sync

- ☑ Add Google Contacts API update function to sync address changes
- ☑ Call Google Contacts sync when “Update contact address permanently” is selected
- ☑ Handle sync errors gracefully (show warning if Google sync fails)
- □ Test that address changes propagate to Google Contacts

Address Edit Button on Waypoint Cards

- ☑ Add “Edit Address” button to waypoint cards in route detail page
- ☑ Ensure address edit dialog shows “Update contact address permanently” option
- ☑ Test that clicking edit address opens the correct dialog with radio buttons

Consolidate Address Editing into Edit Details Dialog

- ☑ Remove separate “Edit Address” button from waypoint cards
- ☑ Add radio buttons to Edit Details dialog for address update scope (temporary vs permanent)
- ☑ When address changes and “permanent” is selected, update contact card and sync to Google Contacts
- ☑ When address changes and “temporary” is selected, only update waypoint
- □ Test that address geocoding works in the consolidated dialog
- □ Test that Google Contacts sync works from Edit Details dialog