Admin Support & Communication Hub Guide

For the Replit Development Agent

Version: 1.0
Date: 2025-08-21
Author: Replit Coach Too
Status: Draft

1. Objective

This document provides a detailed technical guide for the Replit Agent to build the **Admin Support & Communication Hub**. This is a major enhancement to the existing Super Admin Dashboard. Its purpose is to provide the Avallen Solutions team with the tools to view detailed user profiles, proactively identify missing information, communicate directly with users, and manage the document verification process.

2. Part 1: Backend & Database Enhancements

2.1. New Database Tables for Messaging

To support the internal messaging system, two new tables are required.

- New Table: conversations

Column Name	Data Type	Constraints	Description
id	UUID	PRIMARY KEY	Unique identifier for the conversation thread.
company_id	UUID	FOREIGN KEY (companies.id)	Links to the client company in the conversation.
admin_user_id	UUID	FOREIGN KEY (users.id)	Links to the admin who is part of the conversation.
created_at	TIMESTAMPTZ	DEFAULT now()	Timestamp of when the conversation started.
last_message_at	TIMESTAMPTZ		Timestamp of the most recent message, for sorting.

- New Table: messages

Column Name	Data Type	Constraints	Description
id	UUID	PRIMARY KEY	Unique identifier for the message.
conversation_id	UUID	FOREIGN KEY (conversations.id)	Links to the conversation this message belongs to.
sender_id	UUID	FOREIGN KEY (users.id)	The user ID of the sender (can be an admin or a client).

content	TEXT	NOT NULL	The text content of the message.
sent_at	TIMESTAMPTZ	DEFAULT now()	Timestamp of when the message was sent.
is_read	BOOLEAN	DEFAULT false	Read status flag.

2.2. New Table for Document Comments

- New Table: document_comments

Column Name	Data Type	Constraints	Description
id	UUID	PRIMARY KEY	Unique identifier for the comment.
report_id	UUID	FOREIGN KEY (reports.id)	Links to the report or document being commented on.
admin_user_id	UUID	FOREIGN KEY (users.id)	The admin who wrote the comment.
comment_text	TEXT	NOT NULL	The content of the comment.
created_at	TIMESTAMPTZ	DEFAULT now()	Timestamp of the comment.

2.3. New Backend Services & API Endpoints

- **UserProfileService:**
  - Create a new service with a function get_user_profile_completeness(company_id). This function will check all the user's data (company_data, lca_questionnaires, etc.) and return a JSON object flagging incomplete sections.
- **New Admin API Endpoints (/api/admin/users):**
  - GET /api/admin/users: Fetches a list of all client users.
  - GET /api/admin/users/<company_id>: Fetches a detailed profile for a single company, including the completeness data from the UserProfileService.
- **New Messaging API Endpoints (/api/messages):**
  - GET /api/messages/<conversation_id>: Fetches all messages for a conversation.
  - POST /api/messages/<conversation_id>: Sends a new message in a conversation.
- **New Commenting API Endpoints (/api/admin/reports):**
  - POST /api/admin/reports/<report_id>/comment: Allows an admin to add a comment to a report.

3. Part 2: Frontend Implementation (Admin Dashboard)

This section details the new UI components to be built within the /admin section.

3.1. User Management View (/admin/users)

- **UI:** A new page that displays a searchable table of all client users (role = 'user').
- **Features:**
  - Columns for Company Name, Contact Name, Email.
  - A "View Profile" button for each user, which navigates to their detailed profile page.

3.2. Detailed User Profile View (/admin/users/<company_id>)

This is the central hub for admin-user interaction. It will be a tabbed interface.

- **Tab 1: Profile Overview**
  - **UI:** Displays the user's core company information.
  - **Profile Completion Module:** A prominent section on this page will display the data from the UserProfileService. It will be a checklist or a series of progress bars visually flagging the areas that have not been completed (e.g., "Scope 1 Data: Incomplete", "Packaging Data: Complete").
- **Tab 2: Internal Messaging**
  - **UI:** A simple, chat-style interface.
  - **Logic:** This component will fetch and display the message history for the conversation with this user. It will include a text input and a "Send" button to allow the admin to communicate directly with the user without leaving the platform. New messages will trigger notifications for both the admin and the user.
- **Tab 3: Document & Report Review**
  - **UI:** A table listing all reports and other key documents submitted by the user.
  - **Features:** For each document, the admin can:
    - **View:** Open the document to review its contents.
    - **Comment:** A button to open a modal where the admin can write and submit comments. These comments will be saved to the document_comments table and will be visible to the user on their own report page.
    - **Verify/Approve:** The existing "Approve Report" button will be located here, allowing the admin to finalize the verification process after reviewing and commenting.