HopeLine Assistant – Admin & Analytics Upgrade Spec

Detailed Implementation Guide for Replit

READ THIS FIRST
This document tells Replit EXACTLY what to do to finish and upgrade the HopeLine Assistant admin side so it feels like a real, premium SaaS product.

FOCUS AREAS
1) Add a real Analytics layer (backend + admin UI)
2) Make pre-intake data show up clearly in appointments
3) Wire AI summaries into appointments and emails
4) Finish notifications (email now, SMS when provider is ready)
5) Upgrade the client admin page to a simple Futuristic/Cyber-Clean layout
6) Prepare for super-admin vs client-admin separation (phase 2)

Replit should NOT break existing working behavior. Only add/extend what's specified here.

SECTION 1 – ANALYTICS (BACKEND + ADMIN UI)

GOAL
Provide a simple but real Analytics section that shows:
- Total conversations
- Total appointments
- Conversion rate
- Topic/category breakdown
- (Optional later) Activity over time

1.1 Conversation logging assumptions
There should already be a conversation log table similar to:

- conversationLogs:
  - id
  - sessionId
  - role ("user" | "assistant")
  - content
  - category (string | null)
  - createdAt

If user messages are NOT logged yet:
- Update the /api/chat route to log user messages as well, with an inferred category.
- Use a simple keyword-based category system:
  - pricing, availability, requirements, application_process, crisis_redirect, contact_info, pre_intake, other.

1.2 New backend endpoint: /api/analytics/summary

Create a new route handler (protected by admin auth):

GET /api/analytics/summary

Returns JSON like:

```
{
  "totalSessions": number,
  "totalAppointments": number,
  "conversionRate": number,         // between 0 and 1
  "messagesByCategory": [
```

```
    { "category": "pricing", "count": 23 },
    { "category": "requirements", "count": 18 },
    ...
  ]
}
```

Implementation outline (in storage layer):
- totalSessions:
  - Count DISTINCT sessionId from conversationLogs.
- totalAppointments:
  - Count rows in appointments table.
- conversionRate:
  - totalAppointments / totalSessions (0 if totalSessions == 0).
- messagesByCategory:
  - Group conversationLogs by category and count.

Replit must adapt this to the existing DB library
(Drizzle/Prisma/raw SQL/etc).

1.3 Admin UI: Analytics section

Add an Analytics view accessible from the admin navigation:
- Either as a separate page: /admin/analytics
- Or as a section at the top of /admin/dashboard

UI elements:

A) Summary cards row (3 cards):
- Card 1: Total Conversations
- Card 2: Total Appointments
- Card 3: Conversion Rate (%)

Each card:
- Label (small text)
- Big number (e.g., “124” or “23.4%”)
- Sub-label like “Last 30 days” (for now just static text is OK).

B) Category breakdown panel:
- Title: “Top Question Types”
- List categories and counts from messagesByCategory.
Example:
- Pricing – 23
- Requirements – 18
- Availability – 15
- Application – 9
- Crisis redirects – 2

Replit should use the existing frontend stack (React, React Query
or similar) to:
- Fetch /api/analytics/summary.
- Show loading and error states.
- Render cards + list cleanly.

SECTION 2 – PRE-INTAKE → APPOINTMENT LINK

GOAL
When someone completes a pre-intake flow ("See if I qualify"),
that information should be visible to staff in the appointment
detail view.

2.1 Appointment model

Ensure the appointments table/model has room for pre-intake data.

Options:

Option A — Dedicated fields:
- forWho (string) – e.g.: "self", "loved_one"
- sobrietyStatus (string) – e.g.: "currently_sober", "need_detox", "not_sure"
- financialSupport (string)
- timeline (string)
- preIntakeNotes (text)

Option B — JSON:
- preIntakeData (JSON/text) storing an object with those keys.

Either option is acceptable. For simplicity, Option A (flat fields) is preferred.

2.2 Pre-intake frontend

If not already present, implement a PreIntakeFlow component with fields:
- Who is this for? (Self / Loved one)
- Current sobriety status?
- Ability to help with fees?
- Timeline? (ASAP / Within 30 days / Just exploring)
- Notes (optional)

On submit, associate this pre-intake data with the current sessionId and either:
- Immediately store it to the DB as a pre-intake record keyed by sessionId, or
- Store it in memory/state and send it along when an appointment is created.

2.3 Attach pre-intake to appointments

When the user books an appointment (via AppointmentFlow):
- Look up pre-intake data for that sessionId.
- Attach it to the new appointment record:
  - Either by copying into fields (forWho, sobrietyStatus, etc.)
  - Or assigning JSON into preIntakeData.

2.4 Show pre-intake in admin

In the admin appointment detail panel or drawer:

Add a “Pre-Intake” section that displays, if present:

- For who: Myself / Loved one
- Sobriety: Currently sober / Need detox / Not sure
- Financial support: Yes / Not sure
- Timeline: ASAP / Within 30 days / Just exploring
- Notes: [preIntakeNotes]

If no pre-intake data exists, show: “No pre-intake information recorded.”

\---
SECTION 3 – AI CONVERSATION SUMMARIES
\---

GOAL
Each appointment should have a short AI-generated summary of the conversation so staff don’t need to read entire logs.

3.1 Summary generation on appointment creation

Backend steps after an appointment is created:

1) Get sessionId associated with this appointment (from request or storage).
2) Fetch the last ~20 messages (user + assistant) for that sessionId from conversation logs.
3) Call the OpenAI API (gpt-4.1-mini or similar) with a prompt like:

“Summarize this chat for a staff member at a sober living home.
Include:
- Who is reaching out (self or loved one)
- Basic situation
- How urgent it seems
- What they are hoping for
- Any key context for the first call.
Keep it to 3–6 sentences, neutral and professional.”

4) Save the resulting text into appointment.summary.

3.2 Show summary in admin

In the appointment detail view, add a section:

- Label: “Conversation Summary”
- Content: appointment.summary

If summary is empty or failed to generate, show: “No summary available.”

3.3 Include summary in staff email notifications

Whenever an appointment is created and an email is sent to staff, include a “Summary” section in the email body with appointment.summary.

3.4 Failure handling

If OpenAI summary generation fails for any reason:
- Log the error on the server.
- Do NOT block appointment creation.
- appointment.summary can remain null/empty.

SECTION 4 – NOTIFICATIONS (EMAIL + SMS)

GOAL
Ensure email notifications are real and reliable now. SMS will be wired once a provider is chosen.

4.1 Email notifications (staff)

If not yet implemented:

- Use either Resend, Nodemailer, or an existing mail utility in this project.
- Trigger email after appointment creation.

Email should include:
- Name
- Phone
- Email
- Appointment type

- Preferred time
- Notes
- Pre-intake details (if present)
- Summary (if present)

Subject example:
“New inquiry for The Faith House from {name}”

If email sending fails:
- Log error server-side.
- Do not block appointment creation.

4.2 SMS notifications (later)

Maintain or extend existing placeholders for SMS, but do not hard-crash if SMS is not configured.

Environment variables to expect once ready:
- TWILIO_ACCOUNT_SID
- TWILIO_AUTH_TOKEN
- TWILIO_FROM_NUMBER

Logic outline (when configured):
- On new appointment:
  - SMS to staff: “New inquiry from {name}. Preferred {time}. Check admin panel.”
  - Optional SMS to client if they chose SMS as contactPreference.

For now, email is the more important path to be fully functional.

SECTION 5 – CLIENT ADMIN PAGE (FUTURISTIC / SIMPLE)

GOAL
Upgrade the client admin page (Faith House’s admin) to a simple, Futuristic / Cyber-Clean layout that feels like premium SaaS but is easy to use.

5.1 Overall layout

Use a clean dark-mode layout similar to the simple Futuristic/Cyber-Clean mock provided. Structure:

- Left sidebar (narrow):
  - Logo or icon
  - Nav items:
    - Dashboard
    - Appointments
    - Settings
    - Logout

- Main content top row:
  - Page title (e.g., “Dashboard”)
  - Optional date range selector

- Main content body includes:
  - A row of 2–3 stat cards (e.g., Total Appointments, New This Week)
  - A list or table of recent appointments
  - Quick access buttons (e.g., “View All Appointments”, “Go to Settings”)

Replit should reuse the existing admin data sources; this is a layout/UX refactor, not a rewrite of logic.

5.2 Dashboard content

On the client admin Dashboard view:

- Stat cards:
  - Total appointments (all-time or last 30 days)
  - New appointments in last 7 days
  - (Optional) Conversion rate for this client (if per-client tracking exists)

- Recent appointments list (5–10 items):
  - Name
  - Type
  - Status
  - Created date
  - Click to open full appointment details.

5.3 Appointments view

Keep or refine the existing appointments table for the client, showing:

- Name
- Appointment type
- Contact info
- Status
- Preferred time
- Created date

Add search and basic status filters if not present.

5.4 Settings view (client-facing)

Client admin should be able to edit, at minimum:

- Staff email (for notifications)
- Staff phone (for SMS, future use)
- Appointment types enabled (checkboxes)
- Pre-intake toggle (enable/disable)
- After-hours message text

Super admin (you) can still have deeper controls in a separate area later; this Settings view is client-safe.

SECTION 6 – SUPER-ADMIN VS CLIENT-ADMIN (PHASE 2 PREP)

GOAL
Prepare the system so it can later support multiple businesses with separate admin areas.

6.1 Data model: add clientId where missing

Ensure these entities have a clientId or equivalent:
- appointments
- settings
- conversation logs (optional but recommended)
- analytics summaries (derived)

For now, it is enough to:
- Add clientId fields.
- Use a fixed default clientId for Faith House.

6.2 Logical separation

In the future:
- /super-admin routes will manage all clients.
- /admin routes will manage only a single client (Faith House).

For this phase, just keep the code easy to extend:
- Avoid hardcoding “Faith House” everywhere; rely on settings where possible.

SECTION 7 – DONE CHECKLIST

The upgrade is considered DONE when all of the following are true:

- [ ] /api/analytics/summary exists and returns correct numbers.
- [ ] Admin UI shows:
  - Total conversations,
  - Total appointments,
  - Conversion rate,
  - Category breakdown.
- [ ] Pre-intake data is stored with appointments and visible in the appointment detail view.
- [ ] AI summaries are generated, saved to appointment.summary, and visible in admin + staff email.
- [ ] Email notifications send successfully with key appointment details.
- [ ] SMS placeholders exist and do not break the app if not configured.
- [ ] Client admin page uses a simple Futuristic/Cyber-Clean layout with:
  - Minimal sidebar,
  - Dashboard with stat cards + recent appointments,
  - Appointments list/table view,
  - Basic Settings for notifications and intake toggles.
- [ ] Data structures are ready for clientId usage in a future multi-tenant/super-admin build.

Replit should follow this spec step-by-step and reuse existing logic where possible, only adding and extending what is described above.