Treasure Coast AI

MASTER DOCUMENTATION – SaaS Platform, Admin Manual, Client Guides, Developer Handbook, Brand Kit, and Automation Engine Deep-Dive

Hybrid Neon-Glass Edition

Technical Developer Handbook – Full System Architecture

This handbook provides complete engineering coverage of the entire Treasure Coast AI SaaS platform, including architecture, tooling, deployment strategy, developer workflows, coding conventions, and extension guidelines.

The system is built using Next.js (App Router), React, TailwindCSS, Prisma, PostgreSQL, Stripe, OpenAI API, and JWT authentication.

Each feature is modularized using a domain-based folder structure that separates services cleanly for maintainability.

Key engineering goals include: multi-tenant isolation, clean component architecture, reusable API logic, predictable state flows, automated billing enforcement, robust logging, and AI-driven automation.

Folder Structure Breakdown

/app – API routes, server components, route handlers.

/components – Shared UI components with shadcn/ui styling.

/modules – Business domains such as bots, leads, automations, analytics.

/lib – Utility functions, JWT verification, OpenAI client, Stripe client.

/prisma – Database schema files, migrations, seed scripts.

/styles – TailwindCSS configuration and global styles.

/scripts – Deployment helpers, scheduled tasks, usage resets.

API Route Reference

/api/auth – Login, register, JWT issue, refresh.

/api/workspaces – CRUD operations, plan assignment, super-admin actions.

/api/bots – Create bots, edit settings, clone bots.

/api/knowledge – CRUD KB items, import FAQ.

/api/chat – Main conversation engine with OpenAI routing.

/api/automations – Saves/loads V1 & V2 flows.

/api/stripe – Webhooks, checkout, billing portal.

/api/inbox – Conversations, agent side replies.

/api/analytics – Usage stats, lead charts, time-series messages.

Developer Workflow

Run migrations before development using `npx prisma migrate dev`.

Use a seeded super admin for quick testing.

Use environment variables loaded via `.env.local`.

Follow pull request conventions: linted, typed, documented.

Testing: unit tests for utils, integration tests for chat, E2E tests for widget.

Super Admin Training Manual

This manual trains system administrators to operate and manage the Treasure Coast AI platform from the global super-admin portal.

Super admins can manage every workspace, plan, user, and bot in the system.

Access is restricted to role=SUPER_ADMIN and enforced at every route.

Super Admin Abilities

View global dashboard with revenue, users, bots, leads, and system usage.

Open any workspace, impersonate clients, apply plan changes.

Suspend/restore subscription access manually.

View and filter System Logs.

Review error logs, message failures, and AI outages.

Set feature flags for early-access modules (V2 Automations, CRM, etc.).

Client User Manual – Using the Platform

This guide explains how business owners (clients) log in, create bots, configure automations, view analytics, and manage billing.

Everything is designed for non-technical users with step-by-step UI instructions, tooltips, and guided onboarding.

Bot Creation Walkthrough

Step 1: Click 'Create Bot' on dashboard.

Step 2: Upload avatar, define tone, set welcome message.

Step 3: Add FAQs to Knowledge Base.

Step 4: Configure Automations (V1 or V2).

Step 5: Install widget on your website with one line of script.

Investor Deck – Strategic Overview

Treasure Coast AI is positioned as a next-generation automation SaaS for small businesses, offering AI chat, lead capture, analytics, and advanced drag-and-drop automations.

Market Opportunity: 45M SMBs globally, <2% use AI automation, large whitespace.

Business Model: Subscription SaaS with usage-based expansion.

Moats: Multi-tenant engine, automation builder, white-label ability, local niche expansion.

Brand Kit – Treasure Coast AI

Primary Colors: Neon Blue #00e5ff, Deep Navy #0d47a1, Dark Glass Black rgba(0,0,0,0.85).

UI Style: Glassmorphism panels, neon accents, glows, gradients.

Typography: Inter, bold headings, clean readable UI text.

Imagery: Futuristic Florida coastline aesthetics with neon reflections.

Automations Engine Deep-Dive

The Automations Engine consists of three layers: Flow Builder UI, Flow Storage (JSON), and Flow Executor.

Node Types: Trigger, Condition, AI Condition, Delay, Action, Lead Capture, Variable Setter, End Node.

Execution State Machine: Each conversation stores its active node in metadata.

AI-Assisted Flow Generation: Natural language to structured flow JSON using OpenAI.

Error Handling: Node timeouts, retries, fallback logic.

Event Trigger System

Triggers include: First Message, Keyword Match, Office Hours, Inactivity Timers, Message Count Conditions.

Each trigger maps to a Listener that pushes the conversation into the appropriate automation sequence.

Deployment Guide – Step by Step

Step 1: Configure environment variables (DB, Stripe, OpenAI).

Step 2: Run `prisma migrate deploy`.

Step 3: Deploy Next.js app to Vercel or Replit.

Step 4: Configure Stripe webhook endpoint.

Step 5: Run seed script to initialize super admin and example workspace.

Step 6: Test widget on external site.

Maintenance & Scaling

Use row-level security through workspaceId filtering for isolation.

Add Redis or Upstash for rate limiting, caching, and scheduling automation delays.

Enable log retention and anomaly detection for system health.