Treasure Coast AI — FULL SaaS SYSTEM DOCUMENTATION

Purpose of This Document

This is the COMPLETE, FULL-SYSTEM DOCUMENTATION for the Treasure Coast AI SaaS platform.
It includes:

- Full architecture
- Backend code structure
- Frontend structure
- Control Center design
- Template system
- Bot editing system
- Branding
- Analytics
- Billing (Stripe)
- File structure
- Onboarding forms
- Feature gating
- Everything needed for Replit to build the platform perfectly

1. Final File Structure

```
/client
/src
/components
/ui
Button.tsx
Card.tsx
Input.tsx
Select.tsx
Tabs.tsx
Table.tsx
Badge.tsx
/layout
Sidebar.tsx
Topbar.tsx
ControlCenterLayout.tsx
/charts
TrendLine.tsx
/pages
/super-admin
ControlCenter.tsx
ClientOverview.tsx
BotSettings.tsx
BillingStatus.tsx
AnalyticsPage.tsx
LogsPage.tsx
TemplatesPage.tsx
CreateClientFromTemplate.tsx
/client
Dashboard.tsx
Analytics.tsx
Leads.tsx
/theme
theme.css

/server
app.ts
/routes
chat.ts
bots.ts
clients.ts
templates.ts
analytics.ts
billing.ts
logs.ts
/services
botService.ts
clientService.ts
analyticsService.ts
billingService.ts
```

```
templateService.ts
/utils
logger.ts
crisisCheck.ts
validateStatus.ts
/bots
/clients
/logs
.env
```

2. Theme System (theme.css)

Defines the premium SaaS look using gradients, shadows, and consistent spacing.

Variables used globally:
--color-primary
--color-accent
--radius-card
--radius-button
--shadow-soft
--shadow-strong

3. UI Component Library

All UI follows a premium component system:

- Button.tsx (primary, secondary)
- Card.tsx (rounded, shadowed)
- Input.tsx
- Select.tsx
- Tabs.tsx
- Table.tsx
- Badge.tsx

These ensure the entire platform looks consistent and professional.

4. Control Center (Super Admin)

This is the MOST IMPORTANT page.

From ONE SCREEN, you can:

- Edit any client
- Create new clients
- Use templates
- Modify bots
- View analytics
- View logs
- Update billing
- Pause/activate accounts

Tabs:

- Overview
- Bot Settings
- Billing
- Analytics
- Logs

5. Template System

Templates allow instant client creation.

Templates include:

- Restaurant
- Barber
- Auto Shop
- Home Services
- Gym
- Faith House
- Real Estate
- Med Spa
- Tattoo

Each template is a JSON bot config with:

- Default services
- Default FAQs
- Default hours
- Tone
- Safety rules

6. Bot Editing System

Editable fields:

- Business name
- Phone
- Email
- Website
- Hours
- Services
- FAQs
- Tone
- Integrations

Backend:
GET /api/super-admin/bots/:botId
PUT /api/super-admin/bots/:botId

7. Analytics System

Backend:
GET /api/super-admin/analytics/:clientId

Metrics displayed:

- Messages last 7 days
- Leads
- Bookings
- Trend line chart

8. Billing System (Stripe)

Backend:
POST /api/billing/create-checkout-session
POST /api/billing/webhook

Webhook Events:

- checkout.session.completed → activate
- invoice.payment_failed → pause

9. Faith House Feature Gating

Only appear when client.type === 'sober_living'.

All other clients get simplified UIs.

10. Onboarding Form

Fields:

- Client name
- Business type
- Phone
- Email
- Website
- Hours
- Services
- FAQs
- Template to use

Flow:

1. Fill form
2. Template cloned
3. Bot created
4. Client created
5. Stripe link generated

11. Deployment Instructions

1. Set up Neon DB
2. Add .env variables
3. Build frontend
4. Deploy server
5. Connect domain
6. Configure Stripe webhook

12. Developer Summary

This system is fully modular:
- Clean UI
- Clean backend
- Template-driven
- Subscription-driven
- Multi-tenant
- Premium-grade SaaS

This PDF is the source of truth for Replit.