GTS MEDOH · TECHNICAL DOCUMENTATION

Technical Administration & Maintenance Manual

Architecture, configuration, deployment procedures and maintenance reference for the GTS MedOH platform. Intended for platform owners, developers and support contractors only.

THIS DOCUMENT IS FOR

Platform owner · Developer · Support contractor · Technical administrator

GTS MedOH · May 2026 · Geotech Systems Ltd · Confidential

Platform Architecture Overview

The GTS MedOH platform consists of three interconnected systems sharing a single Supabase database:

SYSTEM	URL	DEPLOYMENT
Marketing Website	gts-medoh-website.pages.dev	Cloudflare Pages — project: gts-medoh-website
Booking & Compliance Platform	gts-medoh-booking.pages.dev	Cloudflare Pages — project: gts-medoh-booking
Database & Auth	peldaxqelwcoujngjnng.supabase.co	Supabase (hosted Postgres + auth + storage + edge functions)

File Reference

All platform HTML files live in the project root at `/Users/alinfarah/Documents/Nurse Training Booking/GTS_MedOH_Booking_responsive_pass/`

FILE	PURPOSE	AUTH REQUIRED
`admin.html`	Full admin panel — all operational tabs	Admin roles only
`admin-intelligence.html`	Management Intelligence dashboard	Admin roles only
`company-portal.html`	Organisation Hub — company booker portal	company_admin role
`trainer-portal.html`	Trainer portal	trainer role
`my-bookings.html`	Delegate portal	Authenticated users
`help.html`	Help Centre — all guides with search	None (public)
`nurse-training-booking.html`	Public booking form	None (public)
`courses.html`	Public course listing	None (public)
`certificate.html`	Certificate verification (token-based)	None (public)
`review.html`	Course feedback submission (token-based)	None (public)

Website files

Marketing website files live in `website/` and are deployed separately via `website-deploy.sh`.

Supabase Configuration

Project details

SETTING	VALUE
Project ref	`peldaxqelwcoujngjnng`
Project URL	`https://peldaxqelwcoujngjnng.supabase.co`
Region	eu-west-2 (London)
Anon key	See `config.js` in project root
Service role key	Supabase Dashboard → Settings → API

Key database tables

TABLE	PURPOSE
`profiles`	User profiles and roles — extends auth.users
`companies`	Organisation registry
`bookings`	All training bookings with delegate details and payment info
`course_sessions`	Scheduled session instances — dates, trainers, locations, capacity
`certificates`	Issued certificates with token-based verification
`employees`	Workforce records linked to companies
`compliance_types`	Configurable compliance category definitions
`compliance_records`	Individual employee certification records
`competency_requirements`	Job role → compliance type mappings per organisation
`assets`	Equipment register with inspection/calibration tracking
`risk_items`	Risk register — assessments, near misses, corrective actions
`ai_report_catalog`	Approved AI report definitions (whitelisted SQL functions)
`report_execution_log`	Audit log of all AI report executions
`report_cache`	15-minute cache for AI report results
`saved_reports`	User-saved AI report configurations
`renewal_revenue_pipeline`	View: expiring compliance records with renewal value estimates
`customer_health_cache`	Nightly computed health scores per organisation
`customer_relationship_cache`	Nightly computed relationship scores per organisation
`account_opportunity_categories`	Extensible opportunity category registry
`account_opportunity_values`	Computed opportunity values per org per category
`mgmt_alert_rules`	Configurable alert thresholds
`mgmt_alerts`	Triggered management alerts
`audit_log`	All create/update/delete operations with user attribution

Row Level Security (RLS)

All tables have RLS enabled and enforced. The security model is role-based:

- `is_ops_or_above()` — superadmin, admin, operations_admin
- `is_company_admin()` — company_admin role, scoped to org_name match
- `is_trainer()` — trainer role, scoped to assigned sessions
- `is_delegate()` — customer role, scoped to own records

Tenancy isolation for compliance data uses `tenant_id` (a reference to `companies.id`). Company admins can only see records where `tenant_id` matches their organisation.

Edge Functions

All edge functions are deployed to Supabase and called via HTTPS from the frontend. They are located in `supabase/functions/`.

FUNCTION	PURPOSE
`compliance-ai`	AI assistant — two-phase classify + synthesise using Anthropic Claude API
`cancel-booking`	Cancels a booking, frees session capacity, sends confirmation email
`notify-waitlist`	Notifies waitlisted delegates when a session place becomes available
`send-session-reminders`	Daily: sends joining instruction reminders and low-attendance warnings
`dispatch-events`	Processes the system_events webhook queue
`apply-prospect-import`	Applies a staged prospect import batch to the companies/contacts tables
`submit-review`	Processes course feedback and triggers certificate issuance
`send-review-requests`	Sends feedback request emails after session completion
`send-cert-expiry-reminders`	Daily: sends certificate expiry reminders at configured intervals
`create-payment-intent`	Creates a Stripe PaymentIntent for card payments
`stripe-webhook`	Handles Stripe payment confirmation webhooks

Required secrets (Supabase Edge Function secrets)

```
ANTHROPIC_API_KEY     # Claude AI — set via: npx supabase secrets set ANTHROPIC_API_KEY=sk-ant-...
RESEND_API_KEY        # Email delivery
STRIPE_SECRET_KEY     # Payment processing
STRIPE_WEBHOOK_SECRET # Stripe webhook signature verification
```

AI Architecture

The AI assistant uses a two-phase orchestration pattern. No raw SQL is ever generated by the AI.

Phase 1 — Classify: The user's message is sent to Claude with the current AI report catalog. Claude selects the best matching report key and extracts any parameters (date ranges, limits, org IDs). It returns JSON only.

Phase 2 — Synthesise: The selected report function is called via Supabase RPC. The structured result is passed to Claude with a synthesise prompt. Claude returns a JSON response: type, summary, chart data (optional), table data (optional) and next actions.

Approved reports are defined in `ai_report_catalog`. The `fnMap` in `compliance-ai/index.ts` maps each `report_key` to a specific `rpt_*` SQL function. Adding a new AI report requires:

1. Create the SQL function (`rpt_*` naming convention, `security definer`)
2. Insert a row into `ai_report_catalog`
3. Add the mapping to `fnMap` in `compliance-ai/index.ts`
4. Deploy the edge function

pg_cron Jobs

JOB NAME	SCHEDULE	FUNCTION
refresh-compliance-status	Daily 01:00	`refresh_compliance_status()`
check-asset-alerts	Daily 06:30	`check_asset_alerts()`
send-weekly-digest	Monday 07:00	`send_weekly_compliance_digest()`
purge-report-cache	Daily 03:00	`purge_expired_report_cache()`
compute-customer-health	Daily 02:00	`compute_customer_health()`
compute-management-alerts	Daily 06:00	`compute_management_alerts()`
compute-customer-relationship	Daily 02:30	`compute_customer_relationship()`
compute-account-opportunities	Daily 03:00	`compute_account_opportunities()`
dispatch-events-worker	Every minute	dispatch-events edge function

Seeding caches after deployment

Cron jobs run nightly. After a new deployment or extended downtime, seed the caches manually:

```
select public.compute_customer_health();
select public.compute_customer_relationship();
select public.compute_account_opportunities();
select public.compute_management_alerts();
```

Deployment Procedures

Deploying the booking platform

```
cd /path/to/project
bash deploy.sh
```

This generates `config.js`, deploys all HTML/JS/CSS files to Cloudflare Pages (project: gts-medoh-booking) and redeploys all 11 edge functions.

Deploying the marketing website

```
bash website-deploy.sh
```

Deploys the `website/` folder to Cloudflare Pages (project: gts-medoh-website).

Deploying the AI edge function only

```
npx supabase functions deploy compliance-ai --project-ref peldaxqelwcoujngjnng
```

Updating the Anthropic API key

```
npx supabase secrets set ANTHROPIC_API_KEY=sk-ant-YOUR_NEW_KEY --project-ref peldaxqelwcoujngjnng
```

Regenerating PDFs

```
node generate-pdfs.js
```

Regenerates all 11 PDFs from the `docs/` source HTML files. Run after updating any guide.

Cloudflare Pages Configuration

SETTING	VALUE
Account ID	`44285dc433d55a34bce1e6805e07db3f`
Booking project name	`gts-medoh-booking`
Website project name	`gts-medoh-website`
API token	Stored in `deploy.sh` (rotate before sharing)

DNS (pending)

The domain `gtsmedoh.com` is currently managed by Wix, which does not support the DNS records required by Resend for custom sender addresses. DNS must be migrated to Cloudflare before custom email sending can be activated.

Until migration: emails send from `onboarding@resend.dev` (Resend sandbox). After migration: change sender to `bookings@gtsmedoh.com` in all four edge functions that send email.

Storage Buckets

BUCKET	ACCESS	CONTENTS
`course-assets`	RLS — trainers see own course assets	Course resources and pre-reading PDFs
`po-documents`	Private — admin and company_admin	Purchase order document uploads
`compliance-evidence`	RLS — org-scoped	Compliance certificate evidence uploads

Key SQL Functions Reference

Compliance

- `refresh_compliance_status()` — updates all compliance_records statuses based on dates
- `generate_compliance_from_certificate()` — trigger: creates compliance record on certificate issue
- `get_compliance_gaps(p_employee_id)` — returns gap analysis for one employee
- `migrate_booking_company_ids()` — backfills company_id on historical bookings

Renewal Intelligence

- `report_renewal_pipeline(p_org_id)` — pipeline by time window
- `report_renewal_by_org(p_days_ahead, p_limit)` — probability scores per org
- `report_at_risk_customers(p_days_ahead, p_min_value_gbp, p_inactive_days)`
- `report_expiring_by_value(p_days_ahead)` — compliance types by value

Management Intelligence

- `rpt_mgmt_kpis()` — all 7 executive KPI cards
- `rpt_mgmt_renewal_pipeline()` — confidence-tiered pipeline
- `rpt_mgmt_customer_health(p_limit)` — health scores inline compute
- `rpt_mgmt_revenue_leakage()` — leakage value, %, trend, traffic light
- `rpt_mgmt_relationship_scores(p_limit)` — relationship scores from cache
- `rpt_mgmt_account_opportunity(p_org_id)` — opportunity value by category
- `compute_customer_health()` — nightly cron
- `compute_customer_relationship()` — nightly cron
- `compute_account_opportunities()` — daily cron

- `compute_management_alerts()` — daily cron

Administrative

- `gdpr_erase_user(p_user_id)` — anonymises PII, retains financial records
- `void_certificate(p_cert_id, p_reason)` — voids and logs
- `refund_booking(p_booking_id)` — triggers Stripe refund
- `transfer_booking(p_booking_id, p_new_session_id)` — moves booking to new session
- `soft_delete(p_table, p_id)` / `soft_restore(p_table, p_id)`

Maintenance Procedures

Rotating the Anthropic API key

1. Generate a new key at console.anthropic.com
2. Run: `npx supabase secrets set ANTHROPIC_API_KEY=sk-ant-NEW_KEY --project-ref peldaxqelwcoujngjnng`
3. Verify the AI assistant still responds correctly
4. Invalidate the old key in the Anthropic console

Adding a new SQL migration

1. Write idempotent SQL (use `create table if not exists`, `create or replace function`, `on conflict do nothing`)
2. Test in the Supabase SQL Editor
3. Append to `supabase-schema.sql` with a clear migration comment header
4. Update the migrations table in `USER_GUIDE.md`

Disabling a cron job

In the Supabase dashboard: Database → Extensions → pg_cron → find the job → unschedule. Or via SQL:

```
select cron.unschedule('job-name');
```

Recovery from a failed deployment

Cloudflare Pages maintains deployment history. Go to the Cloudflare dashboard → Pages → project → Deployments → select a previous deployment → Rollback. This is instant and requires no code changes.

Checking edge function logs

Supabase Dashboard → Edge Functions → select function → Logs. Filter by time range and look for error-level entries.

Account Opportunity Framework — Future Expansion

The `account_opportunity_categories` table is seeded with seven categories. Three are currently active (training_renewal, certification_renewal, compliance_gap). Four are inactive pending the Prospect Sub module (equipment_calibration, face_fit_programme, health_surveillance, consumables).

To activate a new category when the source module is ready:

1. Update the category row: `update account_opportunity_categories set is_active = true where category_key = 'equipment_calibration';`
2. Extend `compute_account_opportunities()` to populate that category's values
3. The Account Opportunity Value section in the Management Intelligence dashboard picks up the new column automatically

---