SALONOS · PLATFORM DOCUMENTATION

Technical Features Overview

A complete tour of what the platform does — booking engine, admin & staff tools, payments, walk-in queue, notifications, the touchscreen kiosk and waiting-room display, and the security model underneath all of it.

Product: SalonOS — self-hosted, single-salon booking & management platform

Stack: PHP 8.2–8.3 · MySQL/MariaDB · Bootstrap 5.3 · vanilla JS (no build step)

Deployment target: standard cPanel shared/reseller hosting

Document date: September 2026 (build 158 — Glass backdrop styles)

Contents

1. Architecture & design principles

SalonOS is a distributable, self-install product: an owner receives the application files and a database schema, installs both onto their own domain and hosting account, and runs the whole business from there. It is not multi-tenant SaaS — each install is one salon's own independent copy, with a single owner account per install and no franchise/multi-branch concept.

PHP 8.2–8.3 **MySQL / MariaDB** **PDO + prepared statements** **Bootstrap 5.3**
Vanilla JS, no framework **No Composer / build step** **Self-hosted Font Awesome** **cPanel-first hosting**

Everything lives inside a single web-root folder, editable directly from cPanel's File Manager with no build pipeline. There is no external JS framework and no bundler — every page is server-rendered PHP with small, targeted vanilla-JS enhancements. Every third-party integration (Stripe, PayPal, Worldpay, GoCardless Pay by Bank, SMTP, SMS, Google Analytics, Maps, reCAPTCHA) is configured from admin forms, never by hand-editing code, and stored in a config file separate from the database-credential config written once at install time.

Salon-type awareness

Every install declares one primary business type at setup, from thirteen: Hair Salon, Barbershop, Nail Salon, Beauty Salon, Spa, Medi-Spa / Aesthetics, Brows & Lashes, Tanning, Afro / Textured Hair, Waxing & Laser, Mobile Hair & Beauty, Bridal & Occasion, and Other. A single declarative profile per type — not per-type code paths — drives vocabulary (what a staff member, a customer and a booking are called, each owner-overridable), the opening hours and booking rules pre-filled at install (a spa’s seven-day week and 48-hour cancellation window; a mobile business’s 30-minute travel buffer; a bridal studio’s year-ahead booking horizon), and a realistic one-click starter service menu offered on the install wizard with UK-typical durations, prices, buffers and deposits. The type sets defaults only; the owner switches feature modules on or off afterwards, so no salon is ever locked out of a capability by its label. Upgrading installs see no data change — their stored type simply begins feeding the same helpers.

Feature modules

Type-specific capability ships as optional modules behind a single switchboard (Business Profile → Features), stored as plain settings so an upgrading install starts with everything off and a fresh install starts with its type's suggestions on. Shipped so far: **Rebooking cycles** — a per-service recommended return interval, surfaced on the client's confirmation, a dashboard count, and a "Due for rebooking" list in both the admin and front-desk realms with one-click email/SMS reminders that are never sent automatically; and **Courses & session packs** — pre-paid packs of a service sold in salon (recorded as a normal payment for reporting), with sessions drawn down automatically at booking and restored on cancellation. The drawdown is derived from the booking lines themselves rather than a separate ledger, so every cancellation path — owner, staff, front desk, the client's own account, and edits that swap services — stays correct by construction.

Consultation & consent forms — an owner-built form library (four question types, tick-to-confirm declarations, an HTML5 signature pad with a typed-name fallback, per-form validity periods), attached per service and completed either online through a per-appointment HMAC-signed link (no login needed for guest bookings) or at the desk on any device from the appointment page. Every completion stores a snapshot of the questions as they stood, the answers, the signature image and who took it. Appointments carry a "Form outstanding" badge across every calendar and list and cannot be marked Completed until signed, with an audited owner/front-desk override. **Patch tests** — typed test records with validity, required per service, blocking online and kiosk self-booking with a prompt to book the patch-test service first, warning-only for staff, and recorded automatically when a patch-test appointment is completed. Both ship with editable per-salon-type templates wired to the starter menu at install.

Rooms & equipment — resource types with interchangeable units, required per service, folded into the slot engine as a second availability axis: a time is offered only when the staff member and a unit of every required type are free for the whole visit, checked with one cached busy-map query per type and date rather than one per slot. Bookings allocate a unit inside the same transaction (and roll back with a plain-English message if it vanished in a race), calendars label the unit, a "By room" day view shows occupancy per unit, and owner/front desk can move a booking between units with a conflict check. Feature-gated, so the engine is byte-for-byte unchanged for installs that leave it off.

Walk-in queue — a per-day queue (walk_ins + snapshotted service lines) for businesses that run on walk-ins. Entries carry "first available" or a preferred staff member; a first-fit simulation over every on-duty staff member's remaining appointments hands each waiting entry, in arrival order, to the eligible person who could start soonest, producing the "about 15 min" estimates shown at the desk, on the kiosk and on the TV. Calling someone creates a real appointment at that instant (source walk_in, checked-in, courses redeemed and rooms allocated inside the same transaction), so every downstream feature — checkout, payments, reports, rebooking — is untouched; waiting entries never reserve calendar time. Three realms share one page body (owner, Control Desk, staff portal — where a staff member can only call people to themselves), the desk page refreshes itself from a tiny version-token poll, the touchscreen kiosk gains a PIN-free "Join the queue" flow, the TV display gains a queue slide folded into its existing content-version refresh, and stale entries expire lazily at the day boundary with no cron job.

Self-service services — a per-service flag (needs a resource type) that removes the staff axis entirely: the slot engine gains a room-only path (business hours × a free unit of every required type, via the same cached busy-map as staffed bookings), every slot carries staff 0, and the four create/update transactions write `appointment_services.staff_id = NULL`. The public, kiosk, admin and front-desk flows skip the staff picker for an all-self-service basket and refuse to mix one with a staffed service; confirmation

email and pages say "Where: Sunbed 2" instead of "With"; the client's reschedule request and the staff portal's ownership rule both understand a booking that belongs to a room rather than a person; and the three staff-keyed day views synthesise one extra column per unit (negative ids, so they can never collide with a staff id) so nothing disappears into the staff-0 bucket. Switching the module off simply makes the flag inert — existing bookings stay exactly where they are.

Online course purchase — courses flagged "Sell online" are listed on a public Courses & packs page (price, per-session price, saving versus single sessions) and in the client account, and offered as an upsell at online checkout for a single-service basket. A purchase rides the same payment-intent pipeline as a deposit: the intent's payload carries kind=course plus a full snapshot of the course, so the sale is honoured even if the owner edits or retires it mid-checkout; the return page and every provider webhook finalise it idempotently into exactly what an in-salon sale writes (a completed payments row with the provider as method, plus the client_courses row with sold_by "online"). The checkout upsell books the first session in the same finalisation, drawing session 1 through the normal redemption path; if that slot was taken while the client was at the bank, the course is kept and nothing is refunded — they simply pick another time. Purchases require a client account (no gifting), and the client receives a course confirmation email with a link to their courses.

Gift vouchers — money vouchers (owner presets, optional custom amount within a range) and service vouchers (priced at the service, named on the voucher), sold online through the payment-intent pipeline (payload kind=voucher, finalised idempotently by the return page or any provider webhook) or at the desk against cash/card, with unambiguous GV-XXXX-XXXX codes, a branded recipient email sent now or on a chosen date (cron plus a lazy fallback), a buyer receipt, and a printable voucher page behind an HMAC link. Redemption is a first-class payment: on the online pay step a voucher covers all or part of a deposit or course purchase (the remainder goes through the gateway; a fully covered purchase uses a gateway-free 'voucher' intent so every finaliser is reused unchanged), and at every checkout screen the "Gift voucher" method takes a code. Every redemption writes a payments row (method gift_voucher, reference = code) and an audit row, balances decrement atomically, partial use carries the balance, expiry is applied lazily, and the owner can look up, resend, print or cancel any voucher.

Group bookings — a party of up to twelve booked as one: a `booking_groups` row (name, organiser, date, start time, source) plus one *ordinary* appointment per member carrying `group_id` (the organiser's with their client_id, guests as guest_name lines, the organiser's contact details on exactly one of them so reminders arrive once). The scheduler offers only "together" starts: for each member it takes the eligible staff (honouring a named preference), computes each staff member's free-start set once through the existing slot engine, then runs a small backtracking assignment at every candidate time so distinct staff are found whenever an arrangement exists (a greedy pick would miss "Bea wants Pia, so the organiser takes Quin"), and finally checks that enough free units of every required room or machine type exist for the members needing one at that time. Creation recomputes the assignment inside one transaction and rolls the whole party back on any failure. The desk page (admin and front desk, shared body) books directly, ignoring online minimum notice; the online page needs a logged-in organiser and, when any service carries a deposit, sends them through the same payment-intent pipeline as deposits, courses and vouchers (payload kind=group, one combined deposit recorded against the organiser's line, gift vouchers accepted, the pay step rechecking the whole party before charging). Calendars wear a group badge, the three appointment pages show the party with "Cancel the whole group" (owner/front desk), clients see their guests under My bookings and can't reschedule a member online, and self-service services are excluded because they carry no staff to assign.

Travel time — a per-service `is_mobile` flag, owner-defined postcode zones (`travel_zones`: comma/space-separated prefixes, one-way minutes, an optional fee) with a default for anything unmatched, and four visit columns on the appointment (`travel_minutes`, `visit_address`, `visit_postcode`, `travel_zone`). Prefix matching is deliberately postcode-aware: an all-letters prefix is an area ("LS" matches LS1 and LS28), a district prefix must equal the outward code ("LS1" never swallows LS10), a longer prefix matches the full code ("LS28 7" = one sector); the longest matching prefix wins. The visit being booked is a request-scoped context (the rule builders take services + profile only), resolved once per request from the session postcode, a desk form field, the payment-intent payload, or the appointment row itself for a reschedule/edit; `service_effective_rules()`/`basket_effective_rules()` expose it as `travel`, `staff_busy_intervals()` returns each existing interval's travel as a third element, and both slot loops require a gap of max(buffer, this travel, neighbour's travel) — one drive between two homes, never a there-and-back double count, nothing extra at the start or end of the day. All four booking transactions call `travel_apply_to_appointment()`, which stamps the visit and writes the zone's fee as an ordinary `appointment_services` line (`service_id NULL`, 0 minutes, sorted last) so checkout, payments and reports need no new arithmetic; an edit re-stamps from the appointment's own values. The public flow takes the postcode at the basket step (so the slot list is honest), the full address at the details step, and rechecks the time if the typed postcode lands in a farther zone; the desk booking screens take the postcode before showing times; the appointment panel lets the owner or front desk correct an address, recomputing zone, minutes and fee line. Kiosk, walk-in queue and group bookings exclude mobile services.

Database & migrations

The schema ships as a single SQL file for a fresh install, and every schema change made after go-live runs through a guarded, self-healing migration mechanism: each numbered upgrade checks whether it has already applied (via a flag row and live column-existence checks), applies only the missing piece, and never throws — a live site simply catches up to the latest schema automatically the next time any page loads, with no manual SQL required from the owner. Every appointment-service line item snapshots its own price and duration at the moment of booking, so a later change to a service's price never rewrites booking history.

2. Public website & SEO

- **Homepage** with two selectable hero styles (in-flow or full-viewport background image), owner-uploaded logo, brand tagline/description, service menu, team section, and a live opening-hours table.
- **Team profile pages** per staff member, with tenure, bio, service list, and availability summary — independently toggleable per person via a "Show on website" flag, separate from whether that person is actually bookable (so a front-desk receptionist can have a public profile without ever appearing in the booking flow, and a bookable stylist can opt out of the public site entirely).
- **Blog** with a full admin authoring flow (draft/publish, featured image, SEO title & description, JSON-LD article markup) and a public paginated listing.
- **Owner-editable static pages**: About, Terms & Conditions, Cookie Policy, Refunds Policy — all four editable from one admin screen, each shipping with sensible starting content.

- **Contact page** with a real working enquiry form that emails the owner through the site's own SMTP integration.
- Per-page SEO meta title/description, Open Graph & Twitter Card tags with a branded social-preview image, sitemap.xml, robots.txt, and a generated favicon/PWA icon set.
- Cookie-consent banner and a Google Analytics 4 integration that only ever injects tracking once a real Measurement ID is saved and enabled.
- Dark/light theme toggle sitewide, with the owner's own brand colours (primary/secondary/accent) and an optional custom brand font overriding the defaults everywhere — public site, admin, and staff portal alike.

3. Online booking engine

One shared availability engine powers every booking surface in the platform — the public website, the client account area, admin/front-desk manual bookings, the staff portal's own booking flow, and the touchscreen kiosk — so "is this slot really free right now" is answered identically everywhere rather than by several drifting implementations.

- **Multi-service basket**: a client can add several services to one visit; duration, buffer, minimum notice, and maximum advance window are all combined sensibly across the whole basket, and staff eligibility is the intersection of everyone able to perform every service in it.
- **Guest checkout or a full client account**, with an account able to register itself onto an existing guest record (e.g. one first created by the front desk) rather than creating a duplicate.
- **Real-time slot computation** against business hours, each staff member's own working hours, approved time off, existing bookings (respecting a configurable buffer), and per-service minimum-notice/maximum-advance rules — with an explicit "Any available" option that assigns the first free eligible person.
- **A transaction-safe double-booking guard**: every booking re-checks the exact requested slot is still free inside the same database transaction immediately before writing it, closing the race condition between viewing a slot and confirming it.
- **Waitlist** for a fully-booked day/service, with an admin screen to notify, mark booked, or expire each entry.
- **Self-service booking history**, cancellation-notice policy, and a "book again with this stylist" deep link from their public profile.
- **Deposit collection** at checkout for any service configured to require one, via card (Stripe or Worldpay), PayPal or Pay by Bank (GoCardless Open Banking) — whichever the owner has enabled, side by side.

4. Client account area

- Register/login/forgot-password (secure hashed reset tokens, one-hour expiry, single-use), self-service email change with confirmation sent to the new address, and profile-photo upload.
- Dashboard with upcoming/past bookings, address, marketing-consent toggle, and a payment breakdown per past visit.

- GDPR self-service: request a full data export (JSON) or account erasure directly from the account area, both actioned by the owner from an admin review screen; erasure anonymises personal fields while deliberately preserving financial/booking records for legitimate retention needs.
- A "Kiosk PIN" panel (shown only when the salon's touchscreen kiosk is switched on) letting a client set the 4–8-digit PIN they'll use in-store alongside their phone number.

5. Owner / admin area

- **Dashboard** with today's bookings, expected revenue, real money collected today, and a live staff roster widget (working/off/on-time-off, with an "Off duty" override for anyone who's stepped out).
- **Calendar**: a staff-columned day view (full-width flexible layout that scales cleanly from 2 to 20+ staff, columns for staff with bookings shown first, empty columns hidden entirely) and a month view; every booking card is collapsible with a page-wide "Compact view" switch, remembered per browser.
- **Client CRM**: search, add/edit, full booking & payment history per client, soft-delete.
- **Staff management**: profile, role, working hours, calendar colour, photo, per-staff service eligibility, archive (with a real-booking-history guard against permanent deletion), and an owner-editable **"Elevated" flag per role** that determines manager-level access (Team calendar, staff picker on bookings, bulk SMS, etc.) independently of whatever the role happens to be named.
- **Services & categories**: price, duration, deposit requirement/amount, and per-service overrides of the salon's default buffer/notice/advance-booking rules.
- **Business profile**: address, contact details, seven-day opening hours, and salon-wide booking rules.
- **Site settings**: logo, brand colours (including per-segment brand-name colouring), brand font, homepage hero style & image, meta keywords/description, favicon.
- **Reports** (owner only, one date range, four tabs): *Revenue* — expected vs. collected vs. outstanding, tips and no-show fees shown separately; *Takings by method* — every payment recorded in the range by cash / card terminal / Stripe / PayPal / Pay by Bank / gift voucher and by type, refunds netted off; *Modules* — a card per module that’s switched on: walk-in queue (joined, seen, left, average wait), self-service rooms and machines (bookings, minutes, by unit), courses (packs sold, sessions used, sessions still owed), gift vouchers (sold, redeemed, outstanding balance, expired), group bookings (people, live, value) and home visits (travel time, travel charges, by zone); *No-shows* — count, rate against completed visits, value not delivered, fees charged, by staff member and booking source. Every table has a **Download CSV** button.
- **Employee of the Month**: an owner-set monthly winner with a homepage badge and a one-day staff-portal banner.
- **Tips**: full tip history with a per-payment paid/unpaid payout status the owner controls, paginated.
- **Closed dates**, **audit log** (every meaningful action across every realm, filterable, with a clear-and-record action), and a **GDPR request queue**.
- **Text Clients**: a bulk marketing-SMS composer (see Section 9).
- **Kiosk Control** and **Control Desk** (PIN-gated admin/front-desk switch) — see Sections 7 & 10.

6. Staff portal

- **"My Day"** (a plain stylist's own schedule) or **"Team calendar"** (every staff member's day, for an elevated login), plus a personal month-view **"My Calendar"**.
- **Self-service "New booking"**: any staff member can book an existing client; an elevated login additionally gets the full salon-wide service menu and a staff picker (including "Any available"), while a plain stylist is scoped to their own assigned services and always books onto their own chair.
- **Checkout**: every role can declare a payment taken (cash/card-terminal/gift-voucher) and see a live "collected today" total; managers/front-desk additionally see the salon-wide figure.
- **Manual "Check in"** for a client who didn't use the touchscreen kiosk — ties into the exact same "Checked In" badge shown everywhere else in the system.
- **Profile self-service** (phone, bio, colour, photo, password, working hours) with role/active/bookable/email deliberately locked to owner control only.
- **My Tips** (own history) or **Team Tips** (everyone, filterable) depending on role.
- **Text Clients** bulk-SMS composer, elevated roles only.

7. Front Desk (Control Desk)

A dedicated reception login with full day-to-day parity to the owner's own admin area for front-of-house tasks: the same calendar (day/month, compact collapsible cards), manual bookings with a full staff picker, client lookup, checkout/payments, waitlist management, Kiosk Control, and the bulk SMS composer. An **Admin ↔ Front Desk PIN switch** lets an owner step into the full admin area from the Control Desk and back again, with a short idle timeout that automatically returns an elevated session to the Control Desk if left unattended.

8. Payments, checkout & reconciliation

- **Stripe**, **PayPal**, **Worldpay** and **Pay by Bank (GoCardless)**, any combination enabled independently, for online deposit collection at checkout — hosted checkout/tokenisation only, so raw card or bank data never touches the application's own database. All four are webhook-verified (HMAC signature checks) with the return page treated as a UX courtesy only, the webhook as the authoritative source of truth.
- **Worldpay** uses Access Worldpay Hosted Payment Pages (the platform behind Worldpay online and Lloyds Cardnet accounts): a payment page is created server-side with a unique transaction reference and six result URLs, the client pays on Worldpay's page, and on return — or via the `Event-Signature`-verified webhook when they never return — the outcome is confirmed through the Payment Queries API before anything is booked. Refunds for lost slots follow the payment's own HAL action links (partial refund, full refund, or cancel before settlement), and post-completion settlement failures, chargebacks and refunds mark the payment failed and email the owner. Every gateway plugs into one small driver registry (`includes/payments/registry.php`: ready / start / confirm / refund), so deposits, course sales, gift vouchers and group deposits gain each new method with no page changes.

- **Pay by Bank** is account-to-account Open Banking via GoCardless Instant Bank Pay (Billing Requests API): the client picks their bank and approves in their banking app, funds settle by Faster Payments (GBP) or SEPA Instant (EUR), and the method is automatically withheld for any other salon currency. A fulfilled webhook completes the booking even when the client never returns from their banking app, and a later payment failure marks the deposit failed and emails the owner with a link to the booking.
- **Payment-gateway driver registry**: every online method is a driver with one contract (ready / start / confirm), so the pay step, the return page and the "is a deposit booking possible?" check are provider-agnostic. Adding a bank's own card acquirer (Barclaycard Smartpay, Tyl, Cardnet, Global Payments…) is one driver file plus one Integrations card, with nothing else in the booking flow changing.
- **Manual payment recording** (cash, card terminal, gift voucher) from any of the admin/front-desk/staff checkout screens, with a running balance-due calculation that correctly excludes tips and no-show fees from the "amount owed" figure.
- **Refunds**, tracked against the original appointment and correctly reflected in balance-due math.
- **Tip payout tracking**: an owner-controlled paid/unpaid status per tip.
- **Glass look, site-wide**: one setting (`ui_theme`, glass by default, Classic one radio button away) becomes a `<body>` class emitted by every layout header, and a single scoped CSS layer (`body.sos-theme-glass …`) restyles the shared classes — no page markup changes, so Classic is the exact pre-155 rendering. Build 157's recipe: brand-derived backdrop colours (`ui_orb_colour()` lifts each brand colour to a saturation floor and pinned lightness, emitted as `--sos-orb-N` / `-dark` custom properties alongside the brand colours) drawn on two fixed `body::before/::after` layers with a 3% SVG grain, drifting on a 60/75 s transform loop while `body.sos-glass-motion` is present (owner switch `ui_glass_motion`; `prefers-reduced-motion` stills it regardless); panels at ~52% white with `blur(18px) saturate(160%)`, a translucent rim, inset top highlight, a diagonal sheen as an extra background layer (no pseudo-elements, so nothing needs `overflow:hidden`) and a brand-tinted shadow; three tiers (chrome most transparent and most blurred, cards, then nested elements with no fill); blur only ever on containers — a calendar's hundreds of appointment cards are translucent tints with no filter; gradient stat numbers and a dark-mode ink lift for brand-coloured text; `prefers-reduced-transparency` and `@supports not (backdrop-filter)` fallbacks. The kiosk and TV carry the same recipe in their own scoped blocks with solid fallbacks above. **Build 158** adds three alternative shapes for that same backdrop — *Orbs* (three big glows, the original), *Mesh* (calmer: several smaller, further-spread, sometimes blended blobs), *Aurora* (broad flowing diagonal bands via angled `linear-gradients`, not rotated elements — a static rotation would fight the drift keyframes' own transform), and *Geometric* (a faint two-direction diagonal grid of hairlines plus a soft colour wash, deliberately less organic) — picked from Site settings → Look & feel as swatch thumbnails rendered in the salon's own live brand colours. A new setting (`ui_glass_backdrop`, clamped server- and template-side to a known value, default `orbs`) controls which one shows; all four are still driven off the identical `--sos-orb-*` custom properties, so the setting only ever rearranges them — it never changes their colour source — and all four reuse the same drift keyframes and motion toggle, so no new animation logic was needed. Kiosk and TV read the identical setting, so a chosen backdrop style is consistent everywhere without a separate kiosk-only choice.
- **Reports** as tabs on one owner-only page sharing a validated date range: the original revenue reconciliation; takings by method × type from the payments ledger with refunds netted per method; a Modules tab whose sections exist only while their feature module is on (average walk-in wait

computed in PHP from joined/called stamps, self-service from staff-less lines joined to their unit, a live course liability of unused unexpired sessions valued per session, voucher sold / redeemed / outstanding / expired, group live-member value, home-visit travel minutes and travel-line charges by zone); and no-shows with a rate against completed visits. Every section streams a UTF-8 CSV (BOM for Excel) through one helper; all queries run identically on MySQL and the SQLite test stand-in (DATE() ranges, no DATE_ADD/TIMESTAMPDIFF/GROUP_CONCAT).

9. Notifications & marketing SMS

- **Transactional email** via a real SMTP client (implicit-SSL or STARTTLS, AUTH LOGIN) with a dark, salon-branded HTML template used for every notification the app sends — booking confirmation, reminders, cancellation notices, status-change updates, password/email-change confirmations, receipts, GDPR acknowledgements, and internal owner alerts whenever a staff member (not the owner) takes an action worth knowing about.
- **Appointment reminders**, sent by a daily cron job at an owner-configurable lead time.
- **Transactional SMS** via The SMS Works (UK) for reminders, status updates and owner-composed bulk texts.
- **Text Clients**: a manually-composed bulk SMS broadcast tool (admin, front desk, and elevated staff) restricted to clients who have explicitly given marketing consent, with an optional "only clients who've actually visited within the last N months" recency filter, a single `{first_name}` merge token, a live segment-count estimate, and a test-send option before broadcasting for real.

10. Touchscreen kiosk & waiting-room TV display

Two optional, off-by-default client-facing screens, each reached via a long unguessable token URL that never expires until the owner regenerates it, managed from one Kiosk Control page (full parity between admin and front desk).

Touchscreen booking kiosk

- A client identifies themselves with just their phone number and a PIN they set on their own account — a narrow, separate session realm from a real client login, so a shared walk-up device can never leak one client's account to the next.
- Two entry points, **Check In** (for today's own appointment, with a green confirmation once done) and **Make New Booking** (the full services → staff → date/time → confirm flow), both landing on the same identity step first.
- PIN brute-force lockout scoped per client account, a 30-second idle timeout that resets the session, and every step re-checking the kiosk is still switched on.
- Built without any external CDN dependency, since a kiosk device may sit on a restricted network or an old browser.

Waiting-room TV display

- A fullscreen, non-interactive, auto-rotating slideshow: welcome branding, opening hours, upcoming closed dates, who's working today, automatic staff birthday & work-anniversary slides, an optional live

services & price list, and owner-authored manual announcement slides (with an emoji picker, drag-to-reorder, and a one-click pause).
- Cross-fades between slides, staff avatars, and slide headings/branding rendered in the salon's own brand colour.
- Automatically detects when its content has changed (a new slide, a setting change, a date-driven event like a birthday) and reloads itself at the end of the current rotation cycle — no one needs to walk up and refresh the browser by hand.
- Built to survive old, low-powered embedded TV browsers: flexbox-only layout, overscan-safe padding, no WebSocket/polling connection, a periodic clean reload as a safety net, and an optional fullscreen toggle.

11. People Onsite live panel

An always-visible panel at the bottom of the admin, front-desk, and staff-portal navigation showing, in real time: which staff are rota'd in today (with a one-click "Off duty"/"Back on duty" override for anyone who's stepped out), and which clients are currently checked in and still on the premises. A client is inferred to be onsite from the moment they check in (via the kiosk or a manual check-in) until a genuine checkout payment is recorded or their booking is marked Completed — then they stay visible for a further two minutes before quietly dropping off the list, so nobody watching the panel sees someone vanish mid-glance. The panel polls every ten seconds and is collapsible per section, remembered per browser.

A daily cleanup (run both by the site's own scheduled cron job and as a live safety-net the first time the panel is viewed each day) clears the "Checked In" flag from any appointment left over from a previous day where the visit was never actually completed or paid — so a forgotten check-in never shows as "Checked In" forever on that client's later booking history.

12. Security, roles & compliance

- **Five independent session realms** (owner/admin, front desk, staff, client, and the kiosk's own narrow identity-only session), each with its own login, lockout, and password-reset flow.
- **Role-based access** driven by an owner-editable **"Elevated" flag** per staff role (Site Settings), rather than hardcoded matching on a role's name — a custom role called "Supervisor" gets full manager-level access the instant the owner ticks the box, with zero code changes needed.
- **CSRF protection** on every state-changing form, checked with a timing-safe comparison independent of any client-side gating.
- **Brute-force lockout** on every login form, plus a separate, PIN-specific lockout for the Control Desk switch and the kiosk's own client PIN, each with its own counter so one can never affect the other.
- **Full audit trail**: every meaningful action across every realm (status changes, payments, staff/client edits, integration changes, duty toggles, broadcasts, and more) is recorded with actor identity and a JSON detail blob.
- **GDPR tooling**: consent capture, self-service export/erasure requests, an admin review queue, and a soft-delete convention (rather than hard deletes) across clients, appointments, staff, and services so historical/financial records are never silently lost.

- **Secrets** for every third-party integration are stored outside the main config file, never re-displayed once saved (a blank field always means "keep the existing value"), and payment card data never touches the application's own database at all — every gateway is hosted-checkout/tokenised only.

13. Reporting & integrations

A single Admin → Integrations screen holds every third-party connection, each independently enabled and each with a genuine live "test" action before it can be relied on: Stripe, PayPal, Worldpay, GoCardless Pay by Bank, SMTP email, The SMS Works SMS, Google Analytics 4, Google Maps, and reCAPTCHA. Appointment reminders and the People Onsite daily cleanup are both driven by one scheduled cPanel Cron Job, with the exact command line shown directly on this page.

14. Engineering & quality discipline

Every feature in this document was built and verified against an extensive, continuously-maintained automated regression suite covering validation logic, database behaviour, and real end-to-end HTTP flows across every login realm, before being delivered. Schema changes always run through the guarded, self-healing migration mechanism described in Section 1, so an existing live installation always catches up automatically — there is no manual SQL step for an owner to perform when upgrading to a newer version of the application.