



Modernize – application overview and features

English translation of `alkalmazas-bemutato-es-funkciok.md`.

Current application version: 1.1.1. This document describes the implementation on the repository's current `main` branch; `ImplementationSpec.md` is the primary source for the development contract and the product boundaries.

Modernize is a Hungarian and English booking system that can be branded for a local service business. One deployment handles one business, one location, and one shared provider calendar; customer data stays in a separate Vercel and Supabase environment.

A fresh developer/demo deployment starts with Modernize software-development branding and a service catalog. On a production client, the client-specific seed and the admin settings replace that.

The flows below apply to the features enabled on the deployment. The behavior of disabled features is described in **Deployment feature flags**.

Customer side

A visitor can complete a booking without an account:

1. They look at the landing page and the service catalog. In the catalog they can search, filter by category, payment mode, deposit, duration, price, and availability today, and sort.
2. They choose a free day and time based on real capacity. A blue dot on the calendar means that day has a bookable time.
3. They enter their contact details, fill in the service's custom questions, and upload a private file if needed.
4. They can use a promo code, then book with on-premise payment or Stripe online payment. Online payment receives a 31-minute hold; the full price or a deposit can be charged. Before submit they must read and acknowledge the privacy notice; checkout links directly to that page. The booking records the consent timestamp and the checkout policy identifier `gdpr-1.1.2:<locale>:<sha256>`, which also identifies the content of the notice that was shown.
5. The successful confirmation gives a secure management link and, for a confirmed booking, a way to add it to Google Calendar. When the booking steps change, the page scrolls automatically to the current step.

If a day is full, the customer can join the waitlist after acknowledging that they have read the privacy notice. When a place opens, the system sends an exclusive, time-limited booking offer in order.

With the management link — or, when signed in, on their own portal — the customer can view, cancel, or reschedule the booking to a free time. Self-service cancellation is allowed only outside the service's cancellation deadline; inside the deadline the interface asks them to get in touch. Rescheduling is possible until the start, but a customer reschedule inside the cancellation deadline permanently disables later self-service cancellation of that booking. For an eligible prepaid booking they can request a refund with a reason. The portal shows upcoming and past bookings separately.

Sign-in is possible with email/password or a Google account; Facebook appears only with the matching optional deployment setting. For email/password registration the system first sends a one-



hour, single-use verification link, and creates the account only when that link is opened. The verification email can be resent. Forgot password sends a new random temporary password to a customer who has credentials, invalidates the old sessions, and for a Google-only account asks for Google sign-in. The response does not reveal whether an unknown email address has an account. The sign-in and registration screen shows a link to the privacy notice. When a social email is authenticated, earlier guest bookings with the same address can be linked to the account.

On **Profile**, a signed-in customer can change their name and phone number, change their email address through a verified flow, and change their password according to the account type. They can request a GDPR data export, for which they receive a private ZIP download link by email that stays available for seven days, and they can delete their own account after confirmation.

The booking site, guest checkout, and customer accounts are not aimed at users under 16 (GDPR 1.1.0 Finding 22). The booker or account holder is at least 16; an adult can book a service in the catalog that mentions children. There is no age gate and no self-service flow for people under 16. This is an operational audience exclusion, not lawyer-approved legal text.

A guest without an account can request access, correction, export, or erasure at the privacy contact email address shown in the privacy notice. For signed-in customers the notice points directly at the profile `#export` and `#delete` sections. A request must be answered without undue delay, and within one calendar month at the latest. At deployment, the contact address must be set in the StorefrontConfig business email field to an address that is actually monitored.

Public website

The administrator can set the business name, logo, hero image, one optional gallery image, seven brand colors, language, currency, timezone, and contact and social details. A location map can be shown from the iframe `src` URL copied from Google Maps “Embed a map”, but the Google Maps iframe loads only after the visitor consents (Show map). The gallery section is visible only when a gallery image has been uploaded; without one, the about and contact blocks use a two-column layout. Address, phone, and email appear on the location/contact card; the footer contains the navigation, the social links, the language switcher, and the four legal links — privacy notice, terms, cookie inventory, and Impressum.

The system supports localized metadata, an Open Graph image, LocalBusiness structured data, `robots.txt`, and a sitemap. The privacy notice and the terms are built from separate Hungarian/English Markdown fields; an empty field shows a “not configured” notice. The admin Legal documents page provides per-language editing, a safe live preview, and unlimited revision history; earlier versions can be restored after a preview.

Administration

The admin interface also checks the administrator role on the server for every mutating action.

Admin date and time fields — availability rules, blackouts, the promo period, and the analytics custom range — use the interface language's format: in Hungarian `yyyy. mm. dd.` and a 24-hour time, in English `mm/dd/yyyy` and a 12-hour AM/PM form. The fields have their own calendar and time picker; an uninterpretable entry shows an error under the field, and on submit the system passes a



canonical value (`YYYY-MM-DD` , `HH:mm` , `YYYY-MM-DDTHH:mm`), which the server interprets in the business timezone.

Catalog and bookability

- Categories: create, edit, order, active state, and safe deletion; a category that is in use cannot be deleted.
- Services: description, image, price, deposit, payment mode, duration, before/after time, capacity, cancellation deadline, reminders, order, and active state. The slug of the public `/book/<slug>` address is editable: an empty field is generated from the English title, and after save the previous URL is no longer available.
- Custom fields: text, number, dropdown, or a private file of at most 10 MB; required flag, options, and order. The booking keeps a snapshot of the question and the answer.
- Availability: weekly general, category-level, or service-level opening hours, plus one-off blackouts. A more specific rule fully replaces the more general one.

Slot math runs in the business IANA timezone, taking daylight-saving changes into account. Buffer time and capacity are part of the conflict check. At the same start of the same service, several bookings are allowed up to the given capacity, but another overlapping service blocks the shared resource.

Day-to-day operation

- Overview and calendar with real booking data. In the calendar a **Today** button jumps back to today, and entries get a capacity indicator (`bookings/capacity` , with separate text when capacity is full) and an overlap indicator with a tooltip when they overlap another buffered booking.
- An operations view of failed or stalled notifications, bookings that are paid but not yet confirmed, refund errors, and the last successful run of all seven cron jobs; navigation to the booking/refund and a manual retry of eligible notifications are supported.
- A booking list with search and filters for status, category, service, and date, and a selectable page size.
- A booking detail page with price, discount, answer, attachment, status, notification, and Google Calendar sync history.
- Approve or deny for optionally moderated on-premise bookings; admin cancellation with a reason and an optional refund, reschedule, completed and no-show states, and a manual retry of a failed notification.
- Manual booking with customer data, an optional confirmation, and a separate confirmation for a time outside opening hours. Overlap protection still applies.
- A guest list with a summary of booking history.
- Percentage or fixed promotions with a period, a total limit, a per-customer limit, and optional service targeting.
- Waitlist with statuses; promotion is automatic.
- Customer-requested, admin-started, and automatic refunds; approval, a required denial note, Stripe processing, and a safe retry of uncertain automatic outcomes. A permanently failed refund can also be restarted manually by an admin; the row shows the attempt count, a Stripe-classified failure reason, and a Stripe Dashboard link for the payment.



- Date-range analytics: confirmation, cancellation, no-show, revenue, deposit, rebooking rate, trend, and performance per service. For an incomplete or reversed custom range the filter reports an error and does not reload the view.

Settings and privacy

- Branding, media, contact, Google Maps embed URL, language, currency, timezone, and the waitlist booking window.
- Email notification, optional SMS, and the admin-approval rule for on-premise bookings.
- A localized VAT rate. The service price stays gross; new bookings store and display a net, VAT, and gross amount snapshot.
- Connect Google Calendar, select a writable calendar, and disconnect. Sync is one-way: create, reschedule, and cancel are copied to Google Calendar; events there do not change bookability.
- GDPR erasure by email: anonymize bookings that must be kept, delete the waitlist and private files, and delete the customer account. An admin account cannot be deleted this way.
- GDPR Articles 18 and 21: the admin records and handles restriction and objection requests with a logged status history. An applied request pauses waitlist notifications to that email address.
- Admin PII views and erasure write `AdminPiiAccessLog` metadata (GDPR Finding 24).
- The Hungarian and English privacy notice and terms can be published, previewed, and restored from an earlier version on the admin Legal documents page; every publish creates a new, retained revision.

On an erasure request for a future booking, the contact data needed for fulfilment may remain until the booking ends; private uploaded files are deleted. The guest can ask, at the contact address, for the future booking to be cancelled and for erasure afterwards. Stripe, Google Calendar, and optional Sentry copies are deleted according to the processor steps in the operations runbook (`app-documents/devops-telepitesi-es-uzemeltetesi-utmutato.md`).

The admin **Billing** page is currently only a visual sample of an invoicing and subscription layout. The working VAT-rate setting is on **Settings**; there is no invoicing or subscription-management backend.

Automatic processes

- booking, change, cancellation, refund, and waitlist notifications;
- timed email reminders per service, and SMS reminders when configured;
- logged retries of failed notifications and Google Calendar operations;
- building, sending, and expiry handling of customer-requested GDPR data exports;
- safe retry of automatic Stripe refunds;
- expiry of unpaid 31-minute holds;
- expiry of waitlist offers and promotion of the next applicant;
- deletion of stale rate-limit counters;
- scheduled GDPR retention: 7-day data-export packages, 90-day outbox and StripeEvent rows, and anonymization of personal data on ended bookings after 24 months (the booking row is kept for analytics).



Deployment feature flags

The version knows nine `FEATURE_*` flags (`lib/features.ts` , sample: `.env.example`). These are Vercel environment variables, not switches on the admin interface. A feature is live only when all three layers allow it:

1. the `FEATURE_*` flag (or the default below, if the variable is unset);
2. the vendor credentials, where the feature has an external account;
3. the admin database setting, where one exists (for example the SMS reminder or on-premise approval). The saved setting remains when the flag is turned off, and applies again when it is turned back on.

There is no `NEXT_PUBLIC_FEATURE_*` variable. The server decides; the browser cannot turn a feature on.

Flag	Default when unset	When off
<code>FEATURE_ONLINE_PAYMENT</code>	on, if both the Stripe secret and the webhook secret exist	no prepaid checkout and no Stripe refund
<code>FEATURE_SMS</code>	on, if the Twilio configuration is complete	no SMS, and the admin SMS switch is hidden
<code>FEATURE_EMAIL_NOTIFICATIONS</code>	on	booking and operations emails are not sent; the account-verification, temporary-password, and data-export emails are
<code>FEATURE_GOOGLE_CALENDAR</code>	on	no calendar sync, and the Settings calendar block is hidden
<code>FEATURE_FACEBOOK_LOGIN</code>	on, if both Facebook keys exist	Facebook sign-in does not appear
<code>FEATURE_GOOGLE_MAPS_EMBED</code>	on	no map field and no public map
<code>FEATURE_WAITLIST</code>	on	the waitlist menu item is hidden, a full day does not offer joining, a claim link does not book, and promotion does not run
<code>FEATURE_PROMO_CODES</code>	on	the promo-codes admin page is unavailable, and checkout does not ask for a code
<code>FEATURE_ON_PREMISE_APPROVAL</code>	on	the on-premise approval switch is hidden, and a new booking is not sent for approval even if the saved admin setting is on

Google-account sign-in has no separate flag: Google OAuth is part of the deployment. Facebook, the map, the calendar, and payment can be switched independently of that.

Rollout order

1. Set branding, contact, the Google Maps embed URL, language, currency, and timezone.
2. Upload Hungarian and English legal content, and the structured Impressum and controller identification data, during deployment. A production `seed:new-client` rejects an incomplete



Impressum configuration.

3. Create categories, services, custom questions, and reminders.
4. Record opening hours and blackouts.
5. Connect Stripe, email, optional SMS, and Google Calendar.
6. Try one full on-premise booking and — if used — one online-payment booking, including the email, the management link, the admin view, and calendar sync.

Product boundaries and external dependencies

Area	Current state
Organization model	One business, one location, one shared provider per deployment
Not part of the current version	Staff selection, multiple locations, recurrence, gift cards, reviews, shift planning, invoicing, and subscription management
Feature flags	Nine <code>FEATURE_*</code> environment SKUs; defaults and off behavior are in Deployment feature flags
Payment and refund	<code>FEATURE_ONLINE_PAYMENT</code> , a Stripe account, and an authenticated webhook are required
Email / SMS	Booking email: <code>FEATURE_EMAIL_NOTIFICATIONS</code> , Resend, and a verified sender. SMS: <code>FEATURE_SMS</code> and a complete Twilio configuration, plus the admin switch
Sign-in / calendar	Email/password works; Google sign-in needs OAuth. Calendar is one-way when <code>FEATURE_GOOGLE_CALENDAR</code> is on. Facebook appears when <code>FEATURE_FACEBOOK_LOGIN</code> and the keys are present
Media and attachments	The public <code>storefront</code> bucket and the private <code>booking-intake</code> Supabase bucket are required
Background jobs	Vercel cron and <code>CRON_SECRET</code> are required

The public cookie inventory is on `/locale/cookies` : the language cookie, the session cookie, and a short-lived OAuth state cookie on social sign-in. If Sentry is enabled on the deployment, the browser sends error reports and performance tracing at a 100% sample rate. There are no advertising pixels; a new cookie audit and consent handling are required before Session Replay or marketing tracking. The public Google Maps embed sends data to a third party (Google), so the iframe loads only after the visitor consents; the browser stores that consent in `localStorage` , not in a first-party cookie.