

Kybele

User guide — how the system works, where every setting lives, why each decision was made.

Generated 2026-05-11

Kybele — User guide

This document explains **how the app works**, **where each setting lives**, and **why the system makes the choices it does**. If something on screen confuses you, it's documented here.

Kybele is a single-user self-hosted tool that holds two parallel workflows in one app:

1. **Personal work** — projects, tasks, notes, calendar
2. **Light freelance ops** — customers, deals, proposals, invoices, expenses, reports

A project can be **personal** (no customer attached) or **customer-linked**. A proposal can stand alone or be tied to a customer + project. Customers do not log in — they are records you keep about people you work with.

Table of contents

1. Concepts
2. The shell — sidebar, topbar, search, language, theme
3. Today dashboard
4. Tasks
5. Projects, Cycles, Modules
6. Notes (block editor + backlinks)
7. Calendar
8. Customers, contacts, pipeline (deals), actions, emails
9. Proposals
10. Invoices, payments, expenses, cashflow
11. Time tracking
12. Reports
13. Import (CSV / Trello / Todoist / Notion)
14. Trash & restore
15. Settings
16. Security model
17. Backups
18. Mobile & offline
19. Keyboard shortcuts
20. Troubleshooting

1. Concepts

Concept	Why it exists	Where it lives
Project	The container for tasks. Either personal or attached to one customer.	/projects, /projects/{slug}
Task	A unit of work. Status (todo / doing / blocked / done / archived), priority (0-3), due date, tags, optional parent task (one level of nesting).	/tasks, /projects/{slug}
Cycle	A time-boxed sprint inside a project (start + end date, status workflow draft → active → completed). Tasks attach optionally. Velocity is auto-computed.	/projects/{slug}/cycles
Module	A thematic workstream inside a project (no fixed dates). Lets you slice "all auth-flow tasks" or "all design-system tasks" across cycles.	/projects/{slug}/modules
Note	A block of structured content. Each block is typed (paragraph / heading / list / todo / code / quote / divider / image / callout). Notes can attach to a task, a project, a customer, or stand alone.	/notes, /notes/{id} (JSON)
Customer	A record about a person/company. Has tax info, status, multiple contact persons, and acts as a parent for projects, proposals, invoices, deals, notes.	/customers, /customers/{id}
Deal	A sales opportunity tracked in a kanban pipeline. Has stage, value, currency, win probability, expected close date.	/deals
Proposal	A quote you send to a customer. Has line items, VAT, discount, validity, locale (EN/TR), status workflow. Generates a branded PDF.	/proposals
Invoice	A payment-tracking record. Kybele does not issue legal e-fatura PDFs — you generate the e-fatura via your GIB integration of choice, then upload that PDF here so Kybele can track partial payments against it.	/invoices
Expense	An outgoing cost — used in the cashflow report.	/expenses
Time entry	A timestamped work block against a task. Start/stop with a live topbar timer, or add manually.	/time
Calendar event	A time-blocked entry in the built-in calendar. Can recur. Can be reminded by email.	/calendar

Out of scope, intentionally: team accounts, customer-facing portals, automated email blasts, e-archive/e-fatura generation, native iOS/Android apps. Kybele is web-only and single-user — those will never be added.

2. The shell

Sidebar

The left navigation is grouped:


- **Personal** — Today / Calendar / Tasks / Projects / Notes
- **Operations** — Customers / Pipeline / Proposals / Expenses / Invoices / Reports
- **System** — Import / Trash / Settings

Each link's icon is colour-coded by module via the `--c-mod-*` token; the body of the page picks up the same accent so you always know what module you're in.

The sidebar collapses on screens < 1024 px wide. Tap the burger to open it; tap outside to close.

Topbar

From left to right:

- **Hamburger** (mobile only) — opens the sidebar
- **Page-specific actions** (varies per page)
- **Live timer** — visible only when a time entry is running. Click  to stop.
- **Global search** — a single input that searches every entity type (tasks, notes, customers, projects, proposals, invoices, deals, expenses, calendar events). Press **Ctrl/Cmd+K** anywhere to jump-focus it. The typeahead shows 4 hits per type; press Enter (or click "See all results") to land on `/search?q=...` for the full results page.
- **Language icon** — a translate icon with a dropdown of every locale listed in `APP_AVAILABLE_LOCALES`. Native names are shown ("Türkçe", "Deutsch"...). Changing the language POSTs to `/locale/{code}`, updates `users.locale`, and reloads.
- **Theme switcher** — sun (light) / moon (dark) / monitor (follow system). The choice is per-user and persists in `users.theme`.

Why no global search modal?

Kybele used to open a modal here, but the modal hid context from the page below it. The inline input + dropdown reads better and lets you keep scanning the page while typing.

3. Today dashboard

/ (default after login).

Layout:

- Top row of counters — "Tasks done this week", "Overdue", "Active customers", "Open quotes", etc.
- "Actions due today" widget — pulls from `customer_actions` where `due_at <= today` AND `completed_at IS NULL`.
- Today's calendar agenda — events that intersect today.
- Today's tasks — grouped into Overdue / Today / Tomorrow / This week / Later / No date.

Why these groups? Daily planning needs the next-action horizon, not a flat list. Grouping by relative time is the same model Things 3 uses.

Click a task title to open its detail; click ► on the row to start the timer; check the box to mark it done.

4. Tasks

/tasks — three views

- **List** — grouped by due-date bucket (Today / Tomorrow / etc.).
- **Kanban** — columns per status. Drag to change status; drop targets auto-update via `POST /tasks/reorder`.
- **Table** — dense, sortable; suitable for bulk operations.
- **Gantt** (per-project only) — read-only 12-week window at `/projects/{slug}/ganttt`.

Creating a task

Two paths:

1. **Inline form on /tasks** — title, due date (defaults to today), priority, recurrence, optional cycle, optional module.
2. **Quick-capture modal** — press `c` (or `Ctrl/Cmd+K` then start typing) to open a single-input modal. The parser understands:
 - `#tag` → tags
 - `! / !! / !!!` or `!important / !urgent` → priority
 - `+project-slug` → attaches to that project
 - Relative dates: `today`, `tomorrow`, `next monday`, `28/05`, `28 May`
 - Times: `14:00`, `2pm`

Example: `Doctor appointment tomorrow 14:00 +personal #health !important`

The same parser runs on the server (so the server is authoritative) and in the browser (so the modal shows the preview).

Recurring tasks

Pick a preset from the "Recurrence" dropdown: daily / weekdays / weekly / monthly / yearly. When you mark the task done, the next occurrence spawns automatically. DST and leap-year edge cases are handled by `Service\RecurringRule`.

Dependencies

A task can depend on another task (it's "blocked by" it). The list view shows a indicator on tasks whose blockers aren't done. Cycle detection prevents $A \rightarrow B \rightarrow A$ loops.

Bulk operations

In the Table view, tick rows and use the bulk bar at the top to:

- Mark done
- Change priority
- Add/remove a tag
- Move to another project

Templates

`/settings/templates` — pre-defined checklists you can apply to any project. The "Apply template" form on the project page spawns one sub-task per template item.

Saved views

Any combination of filters on `/tasks` can be saved (Filter bar → "Save view"). Saved views appear in a quick-pick row at the top of `/tasks`.

5. Projects, Cycles, Modules

Project detail (`/projects/{slug}`)

Header shows: name (editable inline), status dropdown, **Cycles / Modules / Gantt** buttons, **Archive** action.

Body has three view toggles (List / Kanban / Table) for the project's tasks, then the **Files** block (drag-and-drop attachments with thumbnails for images), and finally the project's **Notes**.

The task-add form picks up the project's cycles and modules so you can place a new task into them at creation time. If the project has none, those dropdowns simply don't appear.

Cycles

A cycle is a time-boxed sprint. Open `/projects/{slug}/cycles` to see all of them with a progress bar each. Click into a cycle to see:

- **Stats grid** — total / done / progress % / date range
- **Velocity chart** — a bar per day showing tasks completed
- **Task list** — tasks attached to this cycle, sorted by priority and due

Status workflow: **draft** → **active** → **completed** → **cancelled**. Only the position and dates are editable; the velocity is computed from `tasks.completed_at`.

Modules


A module is a thematic group within a project — for "Auth flow", "Marketing site", "Design system", etc. Modules have a color, a status (planned / active / paused / completed / cancelled), and a progress %. Use them when "this work spans multiple cycles" or "I want to filter all tasks for this feature area".

6. Notes

`/notes` lists all notes. Two view modes:

- **List** — flat grid of cards, pinned on top.
- **Kanban** — grouped by project (one column per project + a "Standalone" column).



The block editor

Click  on any note card to enter the block editor. Each note is an ordered list of typed blocks:

Block type	Markdown shortcut to convert from a paragraph
Paragraph	<i>(default)</i>
Heading 1	<code>#</code>
Heading 2	<code>##</code>
Heading 3	<code>###</code>
Bulleted list	<code>-</code> or <code>*</code>
Numbered list	<code>1.</code>
Todo list	<code>[]</code> (then click each box to check)
Quote	<code>></code>

Block type	Markdown shortcut to convert from a paragraph
Code	<code>```</code>
Divider	<code>---</code>
Callout	<i>(from block menu)</i>
Image	<i>(from block menu)</i>

Block-level controls:

-  to the left of a block opens its menu — change type, delete.
-  below the handle inserts a new paragraph after this block.
- **Enter** in a list/todo creates a new item; on a list item that's empty, Enter exits the list to a new paragraph.
- **Backspace at start of an empty block** deletes the block and focuses the previous one.

Inline formatting

Inside any text block you can use:

- **`**bold**`** → **bold**
- *`*italic*`* → *italic*
- `` inline code `` → inline code`
- Bare URLs (`https://...`) → autolinked

Backlinks

Type `[[` inside any text block to open the backlink picker. As you type after the brackets, Kybele searches across **customers, projects, tasks, notes** in parallel and shows up to 5 hits per type.

- **Bare query:** `[[acme` → searches all four entity types for "acme"
- **Type-filtered query:** `[[customer:elif, [[project:acme, [[task:offline, [[note:pricing` → narrows to that bucket only; the part after the colon is the search term (can be empty — `[[customer:` just browses all customers)

Navigate with **↓/↑**, confirm with **Enter** (or Tab, or click). Kybele replaces the typed `[[query` with the canonical token:

Token shape	Renders as
<code>[[customer:42]]</code>	<code>Acme Holding</code>
<code>[[project:acme-website]]</code>	<code>Acme Website Redesign</code>
<code>[[task:101]]</code>	<code>Build component library</code>
<code>[[note:7]]</code>	<code>Pricing strategy ideas</code>

Customer and note refs are numeric ids; project refs are slugs. The server batches title-resolution across all backlinks in a note (one query per type), so a note with dozens of links doesn't N+1 the database.

Tags

The quick-add form has a Tags input (comma-separated). Tag chips render in the note card footer.

Convert a note to a task

Click → on a note card. The note stays; a task is created with the note's content as the description.

7. Calendar

`/calendar` defaults to the **week view**. Use the toolbar to switch between month / week / day.

- **Click** an empty time slot to create an event there.
- **Drag** an event to reschedule it; release on the new time/day. The PATCH happens server-side.
- **Side-by-side overlap** — when two events conflict, the column is automatically split into lanes by the conflict-detection module.
- **Reminders** — set "X minutes before" on the event modal; the cron at `bin/reminder-send.php` emails you that many minutes before the start.
- **Recurring events** — pick a preset on the create form. Recurring instances are virtual: only one row in `calendar_events`, instances are expanded at query time. Clicking a recurring instance shows the source.

Time-blocking from a task

In any task row, click the small calendar icon. The calendar opens with a pre-filled "create event" modal pointing at that task. The created event keeps a foreign key to the task so you can navigate back.

Google Calendar (one-way sync)

If you connect Google Calendar in `/settings/calendar`, Kybele writes a Google event for every task that has a due date. Updates and completions propagate. **Kybele does not read from Google** — events you create there don't appear in Kybele. The deliberate choice keeps Kybele as the source of truth for your work.

8. Customers & Pipeline

`/customers`

A list of customers with status filter (lead / active / inactive / lost), archived toggle, and full-text search. Click "New customer" to add one.

Customer detail (`/customers/{id}`)

Tabs:

- **General** — contact info, tax info, notes, files, preferred locale.
- **Projects** — customer-linked projects.
- **Proposals** — quotes for this customer.
- **Deals** — pipeline cards for this customer.
- **Actions** — next-step reminders (call/email/meeting). Each has a due date and can be marked completed.
- **Emails** — logged outgoing/incoming emails (you log them manually; Kybele doesn't read mail).
- **Files** — uploaded contracts, business cards, etc.

The "Next action" of the most-overdue or upcoming kind is also surfaced on the Today dashboard.

Pipeline (`/deals`)

A kanban of deals across the stages defined in `pipeline_stages` (seeded with Lead / Discovery / Proposal / Negotiation / Won / Lost — editable in `/settings/pipeline-stages`). Each card shows the customer, value, win probability, and expected close date.

Drag a card horizontally to change its stage; the server-side endpoint `POST /deals/reorder` persists the new positions in one request per drop.

Top of the page shows a **weighted forecast**: $\sum (\text{value} \times \text{win_probability} / 100)$ for non-closed deals, grouped by currency.

Email templates

`/settings/email-templates` lets you write reusable email bodies with `{{customer.name}}`, `{{customer.company}}`, `{{deal.value}}`, etc. placeholders. From a customer's Email tab, pick a template and the body fills in. Kybele doesn't send the email — it puts the rendered body on the clipboard so you can paste it into your real mail client.

9. Proposals

`/proposals` lists every proposal. Status pill on each card: draft / sent / accepted / rejected / expired / converted.

Creating one

`/proposals/new` form fields:

- Customer (autocomplete, optional)
- Project (optional, filtered by customer if one is set)
- **Locale** (en/tr) — this controls the PDF template *and* which proposal numbering pool is used. Defaults to the customer's `preferred_locale`.
- Title, description (markdown), issue date (today), valid until (today + 30)
- **Items table** — description, qty, unit, unit price, tax %, computed line totals. Add/remove rows dynamically.
- Discount (absolute amount)
- Terms (markdown)

Calculation

Server-side only, using `bcmath` at scale 2:

```
line_subtotal = quantity * unit_price
line_tax      = line_subtotal * (tax_rate / 100)
line_total    = line_subtotal + line_tax
subtotal     = Σ line_subtotal
tax_total     = Σ line_tax
total        = subtotal + tax_total - discount
```

Numbering

PRP-YYYY-MNNN — auto-incremented per year, transactional (`SELECT ... FOR UPDATE` in `Service\ProposalNumberer`). The format is language-agnostic, so the same PRP-2026-0042 works for both EN and TR proposals.

PDF

Click **PDF** to download. The renderer is mPDF with DejaVu Sans (proper Turkish glyph coverage). The template per-proposal is chosen by `proposals.locale`. Output is stored at `storage/uploads/pdf/{id}-{number}-{timestamp}.pdf` and `proposals.pdf_path` is updated.

Status workflow

- **Draft** → editable, no PDF.
- **Mark as sent** → status sent, `sent_at` set, PDF freshly generated.
- **Mark accepted / Mark rejected** → terminal-ish states.

- **Convert to project** → spawns a new project linked to the customer with the proposal's title/description. Status moves to `converted`.

The daily cron `bin/expire-proposals.php` moves sent proposals past their `valid_until` to `expired`.

10. Invoices, Expenses, Cashflow

Why Kybele tracks payments but doesn't issue e-fatura

In Turkey, the only legal invoice is one issued via GİB (Gelir İdaresi Başkanlığı) through a registered integrator — Mikrokom, Foriba, etc. Doing that integration is its own project (security clearances, XML schema, signing certificates, support agreements). Kybele intentionally stops at:

1. You issue the e-fatura via your integrator's own portal.
2. You download that PDF.
3. You upload it into Kybele on `/invoices/new` — Kybele stores the PDF as an attachment.
4. Kybele now tracks **the payment** against that invoice: due date, status, partial payments, overdue reminders, etc.

`/invoices`

Listed with status filter and totals. Status workflow: `draft` / `sent` / `paid_partial` / `paid` / `overdue` / `void`.

The status is **auto-derived** from payments:

- $\sum \text{payments} \geq \text{total} \rightarrow \text{paid}$
- $0 < \sum \text{payments} < \text{total} \rightarrow \text{paid_partial}$
- $0 = \sum \text{payments}$ AND `today > due_date` AND `sent` → `overdue`
- Otherwise the manual status is used.

Recurring invoices

An invoice can opt into a recurrence rule (same engine as recurring tasks). Cron at `bin/invoice-cron.php` spawns the next instance when due.

Overdue email reminders

Same cron, at 1 / 7 / 14 / 30 days past `due_date`, sends an email reminder to the customer's primary contact.

Expenses (/expenses)

Light category-tagged outgoings. Used in the cashflow report. No tax handling — they're just net amounts.

Cashflow report (/reports/cashflow)

Tiles:

- **Received this month** — sum of payments paid in this calendar month, per currency.
- **Outstanding** — sum of un-paid sent invoices.
- **Overdue** — sum of overdue invoices.

Chart: 12 trailing months of received payments per currency.

11. Time tracking

Per-project hourly rates

`/settings/company` lets you set a default rate. Individual projects override via `projects.hourly_rate`. Customers can have a default that wins over the project's. Order of precedence (most specific wins): customer → project → company default.

Tracking

Two ways to log time:

1. **Live timer** — click ► on any task row. A timer chip appears in the topbar with the elapsed time updating every second. Click □ to stop and persist the entry.
2. **Manual entry** — `/time` → "Add entry" → pick task, start time, duration.

Reports (/reports/time)

Totals grouped by project (and drilldown per task). Useful when invoicing a customer for hours worked.

12. Reports

Report	URL	What it shows
Dashboard	<code>/reports</code>	Header KPIs
Customers	<code>/reports/customers</code>	Active projects, total proposals, acceptance rate, total accepted, last activity

Report	URL	What it shows
Projects	<code>/reports/projects</code>	Per-project completion %, linked proposal value, personal/customer pie
Proposals	<code>/reports/proposals</code>	Status pie, 12-month sent-vs-accepted bars, acceptance rate, median acceptance time, top customers
Tasks	<code>/reports/tasks</code>	Completed this week/month, by project, by tag, avg completion days
Time	<code>/reports/time</code>	Hours per project
Cashflow	<code>/reports/cashflow</code>	Received/outstanding/overdue + monthly receipts chart

Every report has a **CSV export** at `/reports/{slug}/export.csv`. The output is UTF-8 with BOM and semicolon delimiter so Excel TR opens it without prompting.

Charts use Chart.js, loaded only on report pages to keep the main bundle slim.

13. Import

`/import` accepts four formats:

- **CSV** — generic, with column-name aliasing. Recognised columns: `title / name / task / subject, description, status, priority, due / due_at / due_date, tags / labels, project / project_name / list / board, parent / parent_title`. Tab and semicolon delimiters auto-detected.
- **Trello** — board JSON export from `https://trello.com/b/{id}.json`. Lists become statuses; cards become tasks; checklist items become sub-tasks; labels become tags; closed cards are skipped.
- **Todoist** — JSON backup from Todoist Settings → Backups. Priority 1-4 maps to 0-3. Checked items become done. `parent_id` becomes `parent_title`.
- **Notion** — CSV export of a database view. Emoji prefixes on status (🔄 In Progress) are stripped; the Notion (a1b2c3d4) hash suffix in parent titles is stripped.

Two-step flow

1. Upload → Kybele parses + previews the rows. Diagnostics call out **new projects to create** and **new tags**.
2. Confirm → rows commit in a single transaction. The run is recorded in `import_runs` for audit.

If you provide a target project, every row imports into it regardless of its `project_name`. Otherwise rows need a `project_name` column and Kybele will create projects on the fly.

14. Trash & restore

Soft-deleted entities (tasks, proposals, invoices, expenses, deals, calendar events) live in `/trash` for 30 days. Click **Restore** to undelete. The cron `bin/trash-purge.php` permanently removes anything past the threshold.

15. Settings

`/settings` is the index. Sub-pages:

- `/settings/totp` — view enable state, regenerate recovery codes, disable TOTP.
 - `/settings/calendar` — connect/disconnect Google Calendar (OAuth2). Stores tokens AES-256-GCM encrypted.
 - `/settings/company` — name, address, tax info, IBAN, logo, default tax rate, default currency. All used on proposal/invoice PDFs.
 - `/settings/email-templates` — reusable email bodies with `{{customer.x}}` placeholders.
 - `/settings/pipeline-stages` — reorder / rename the deal pipeline.
 - `/settings/templates` — task templates.
 - `/settings/saved-views` — manage saved task filters.
 - `/settings/logs` — tail `storage/logs/app-YYYY-MM-DD.log`.
 - `/settings/backup` — list of backup files in `storage/backups`.
-

16. Security model

Layer	What's enforced
Auth	Argon2id password (≥ 12 chars, NIST-style length + breach blocklist). TOTP RFC 6238 mandatory. 10 recovery codes, single-use, Argon2-hashed.
Login rate limit	10 fails/IP/15 min and 5 fails/email/15 min \rightarrow 15-min lockout.
Session	Secure + HttpOnly + SameSite=Strict cookies. Session ID rotates on login + TOTP.
CSRF	Required token on every state-changing endpoint. Reject with HTTP 419.
Headers	HSTS, X-Content-Type-Options, X-Frame-Options DENY, CSP <code>script-src 'self'</code> .

Layer	What's enforced
File uploads	MIME whitelist + magic-byte verification + SVG sanitization + image re-encode (EXIF stripped) + decompression-bomb checks. Storage is outside the webroot; served only via authenticated controller.
Encryption at rest	TOTP secret, Google OAuth tokens — AES-256-GCM with random IV.
Backups	Encrypted (AES-256-CBC PBKDF2) and gzipped.
Liveness	/healthz is the only public route besides /login — no side effects.

For Turkish hosting providers that require KVKK alignment, all customer data lives in the database you run; Kybele doesn't ship telemetry, doesn't call home, doesn't analytics.

17. Backups

`docker compose exec app php bin/backup.php:`

1. `mariadb-dump` → `gzip` → AES-256-CBC PBKDF2 encrypted with `APP_KEY`.
2. Includes `storage/uploads/` (proposal PDFs, attachments) as a tar.
3. Lands in `storage/backups/`.

Recommended cron: nightly at 03:00, with a 14-day local rotation + weekly offsite copy (Hetzner Storage Box, S3, etc.).

To **restore** on a fresh server:

```
openssl enc -aes-256-cbc -pbkdf2 -d -k "$APP_KEY" -in backup-2026-05-11.sql.gz.enc  
| gunzip | mariadb -ukybele -p kybele  
tar -xzf uploads-2026-05-11.tar.gz -C /var/www/html
```

18. Mobile & offline

Kybele is web-only. The PWA installs on the home screen on iOS and Android via "Add to Home Screen" / Chrome's install prompt.

- **Bottom navigation** on screens < 768 px. Five primary tabs: Today / Tasks / [+] / Customers / More.
- **Quick-capture FAB** — the central [+] button in the bottom nav.
- **Offline shell** — service worker caches all static assets and the last ~30 HTML pages you visited. A page you've never opened falls back to `/offline.html`. The API is never cached, so the moment you reconnect you see fresh data.

19. Keyboard shortcuts

Shortcut	Action
Ctrl/Cmd + K	Open quick-capture modal (works anywhere; mobile uses the green FAB)
/	Focus topbar global search (works anywhere outside an input)
j / k	Move selection down / up in a task list
e	Edit selected task
x	Toggle complete on selected task
g d	Go to Today
g p	Go to Projects
Esc	Close modal / dropdown

Inline forms also parse NL tokens. The new-task form at the top of `/tasks` runs the natural-language parser whenever the title field contains `#tag`, `+project-slug`, `!` priority markers, or relative time words. The form's project / due-date / priority dropdowns then act as fallbacks for fields the parser didn't fill. So `Doctor #health !! tomorrow 14:00` works the same whether you type it in the modal or the inline form.

20. Troubleshooting

Search doesn't find a row I just created

InnoDB FULLTEXT indexes update lazily. If you just bulk-inserted rows (via `bin/seed-demo.php` or an import) and the topbar search misses them for a few seconds, that's expected. The seed script runs `OPTIMIZE TABLE` at the end to flush; if you wrote the rows manually, search will catch up within a few requests, or you can:

```
docker compose exec db mariadb -ukybele -p kybele -e \  
"SET GLOBAL innodb_optimize_fulltext_only=1; OPTIMIZE TABLE customers; OPTIMIZE \  
TABLE tasks; OPTIMIZE TABLE notes;"
```

Login redirects to TOTP but I lost my authenticator

Use a single-use recovery code from `/settings/totp` (you printed them out, right?). If you didn't keep them, you'll need to disable TOTP at the database level:

```
UPDATE users SET totp_enabled = 0, totp_secret = NULL WHERE email = \  
'you@example.com';
```

Then immediately re-enable TOTP in settings with a fresh secret.

Tasks created via import show wrong dates

Different sources use different date formats. The importer uses PHP's `DateTimeImmutable` parser which is permissive but not magic — `2026-06-08` parses fine, `06/08/2026` is ambiguous (US vs EU). If your dates land wrong, the CSV importer accepts a `due_at` column that you can pre-format to `YYYY-MM-DD HH:MM:SS` for unambiguous parsing.

Proposal PDF shows weird boxes instead of Turkish characters

mPDF ships with DejaVu Sans, which covers Turkish. If you see `□` glyphs, it usually means the runtime is missing the font file. Re-run `docker compose exec app composer install --no-dev` (mPDF reinstalls fonts on install).

Service worker caches an old version of a page

The service worker is `kybele-v5-static / kybele-v5-pages`. On a deploy that ships new HTML, the page-cache evicts and the static cache rebuilds on the next install (within minutes). To force-refresh during development: DevTools → Application → Service Workers → "Update on reload".

`/healthz` returns 503

It only fails if the app can't reach the database. Check `docker compose ps` and `docker compose logs db`. If the DB container is healthy but the app can't connect, your `.env` `DB_HOST / DB_PORT / DB_PASSWORD` are out of sync with the DB container.

For deeper architecture detail (migrations, models, services), read the inline docblocks — every class header explains *why* it exists, not just what it does.