Enterprise Console — Modules & Sub-functionalities

This document maps the application modules to their core functionalities, sub-functionalities, and key files/routes. Use this during development as a quick reference.

1) Authentication & Account Management

* Sign in / Register
* Multi-credential login (email | phone | username), optional base64-encoded username
* View: resources/views/auth/login.blade.php
* Request normalization/auth: app/Http/Requests/Auth/LoginRequest.php
* Routes: standard auth routes (routes/auth.php) and POST /login
* Registration
* View: resources/views/auth/register.blade.php
* Controller: app/Http/Controllers/Auth/RegisteredUserController.php
* Routes: registration endpoints in routes/auth.php
* Password management
* Password reset request & reset
* Views: resources/views/auth/forgot-password.blade.php, reset-password.blade.php
* Controllers: app/Http/Controllers/Auth/PasswordResetLinkController.php, NewPasswordController.php
* Email verification
* Controllers: app/Http/Controllers/Auth/EmailVerificationPromptController.php, EmailVerificationNotificationController.php, VerifyEmailController.php
* Developer override (local-only)
* Ctrl+L client handler + dev-override endpoint
* View/JS: resources/views/auth/login.blade.php (JS handler)
* Controller: app/Http/Controllers/Auth/DevOverrideController.php
* Route: POST /dev-override (name: dev.override)
* Env: DEV\_OVERRIDE\_TOKEN (set locally, do not commit a real secret)
* Forgot-username (email code recovery)
* Send verification code to email; verify code to reveal username
* Controller: app/Http/Controllers/Auth/ForgotUsernameController.php
* Mailable: app/Mail/UsernameRecoveryMail.php
* Email view: resources/views/emails/username\_recovery.blade.php
* Routes: POST /auth/recover-username, POST /auth/verify-username-code
* OAuth (Google)
* Redirect and callback handling, creating/finding users by email on callback
* Controller: app/Http/Controllers/Auth/OAuthController.php
* View/button: resources/views/auth/login.blade.php (Sign in with Google)
* Routes: /oauth/redirect/{provider}, /oauth/callback/{provider}
* Config: config/services.php (google settings) and composer dependency laravel/socialite in composer.json

2) User Profile & Account Settings

* Profile edit/update/destroy
* Controller: app/Http/Controllers/ProfileController.php
* Views: resources/views/profile/\*
* Routes: /profile (protected by auth middleware)

3) Dashboard & UI Shell

* Dashboard home
* View: resources/views/dashboard.blade.php
* Route: /dashboard (middleware: auth, verified)

4) Database / Admin Management (Web UI)

* Company users & user restrictions
* Controllers (Web): app/Http/Controllers/Web/Database/CompanyUserController.php, UserRestrictionController.php
* Routes: resource routes under /database/company-users and /database/user-restrictions

5) Company & Organization Management

* Models: app/Models/Company.php, CompanyUser.php, CompanyUserRestriction.php
* Functionality: manage companies, link users to companies, enforce restrictions

6) Reports & Sheets

* Models: app/Models/Report.php, ReportRun.php, Sheet.php
* Functionality: define reports, queue/execute runs, view run history and results

7) Documents & Mailbox

* Models: app/Models/Document.php, Mailbox.php
* Functionality: upload/manage documents, mailbox processing

8) External Connections & Integrations

* Models: app/Models/DatabaseConnection.php, FtpConnection.php, AiProvider.php, AiSession.php
* Functionality: manage external connections, test connections, AI provider sessions

9) Activity Logging & Auditing

* Model: app/Models/ActivityLog.php
* Functionality: record user and system events; admin views for logs

1. Settings / Grid / UI Preferences

* Model: app/Models/GridSetting.php
* Functionality: save per-user grid/table preferences (columns, filters)

Cross-cutting concerns

* Authorization: spatie/laravel-permission (roles/permissions) — see User model and policies in app/Policies/
* API vs Web controllers: Web UI under app/Http/Controllers/Web/, APIs under app/Http/Controllers/Api/
* Mail templates: resources/views/emails/
* Tests: tests/Feature/ and tests/Unit/

How to use this file

* Keep this file updated as you add controllers, views, models or routes.
* When adding a new feature, add a short line under the relevant module linking the new file(s)/route(s).

Next improvement suggestions

* Produce an endpoints inventory (route, method, controller action) in a machine-readable format (openapi or JSON) for automated docs.
* Add a diagram (PlantUML or Mermaid) showing high-level module relationships.

Generated for developer guidance — update freely as the project evolves.