**Enterprise Console — Modules**

Generated on: 2025-09-13T10:01:04.670520

Table of Contents (update fields in Word: Select All -> F9)

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. This is revised

# 1 . Authentication & Account Management

## 1.1 Sign in / Register

### 1.1.1 Multi-credential login

#### 1.1.1.1 Supports: email | phone | username

#### 1.1.1.2 Optional: base64-encoded username input

#### 1.1.1.3 View: `resources/views/auth/login.blade.php`

#### 1.1.1.4 Request normalization/auth: `app/Http/Requests/Auth/LoginRequest.php`

#### 1.1.1.5 Routes: standard auth routes (`routes/auth.php`) and POST `/login`

### 1.1.2 Registration

#### 1.1.2.1 View: `resources/views/auth/register.blade.php`

#### 1.1.2.2 Controller: `app/Http/Controllers/Auth/RegisteredUserController.php`

#### 1.1.2.3 Routes: registration endpoints in `routes/auth.php`

## 1.2 Password management

### 1.2.1 Password reset request & reset

#### 1.2.1.1 Views: `resources/views/auth/forgot-password.blade.php`, `reset-password.blade.php`

#### 1.2.1.2 Controllers: `app/Http/Controllers/Auth/PasswordResetLinkController.php`, `NewPasswordController.php`

1.3 Email verification

1.3.1 Email verification flows

#### 1.2.1.3 Controllers: `app/Http/Controllers/Auth/EmailVerificationPromptController.php`, `EmailVerificationNotificationController.php`, `VerifyEmailController.php`

1.4 Developer override (local-only)

1.4.1 Local dev override

#### 1.2.1.4 Trigger: Ctrl+L client handler in login view (JS)

#### 1.2.1.5 Endpoint: POST `/dev-override` (route name: `dev.override`)

#### 1.2.1.6 Controller: `app/Http/Controllers/Auth/DevOverrideController.php`

#### 1.2.1.7 View/JS: `resources/views/auth/login.blade.php` (contains the JS handler)

#### 1.2.1.8 Required env: `DEV\_OVERRIDE\_TOKEN` (set locally; do not commit real secret)

1.5 Forgot-username (email code recovery)

1.5.1 Username recovery flow

#### 1.2.1.9 Send verification code to email; verify code to reveal username

#### 1.2.1.10 Controller: `app/Http/Controllers/Auth/ForgotUsernameController.php`

#### 1.2.1.11 Mailable: `app/Mail/UsernameRecoveryMail.php`

#### 1.2.1.12 Email view: `resources/views/emails/username\_recovery.blade.php`

#### 1.2.1.13 Routes: POST `/auth/recover-username`, POST `/auth/verify-username-code`

1.6 OAuth (Google)

1.6.1 Google OAuth sign-in / register

#### 1.2.1.14 Redirect & callback handling; create/find users by email on callback

#### 1.2.1.15 Controller: `app/Http/Controllers/Auth/OAuthController.php`

#### 1.2.1.16 View/button: `resources/views/auth/login.blade.php` (Sign in with Google)

#### 1.2.1.17 Routes: `/oauth/redirect/{provider}`, `/oauth/callback/{provider}`

#### 1.2.1.18 Config: `config/services.php` (google settings) and composer dependency `laravel/socialite` in `composer.json`

2. User Profile & Account Settings

2.1 Profile management

2.1.1 Edit / Update / Destroy

#### 1.2.1.19 Controller: `app/Http/Controllers/ProfileController.php`

#### 1.2.1.20 Views: `resources/views/profile/\*`

#### 1.2.1.21 Routes: `/profile` (protected by auth middleware)

3) Dashboard & UI Shell

#### 1.2.1.22 Dashboard home

#### 1.2.1.23 View: resources/views/dashboard.blade.php

#### 1.2.1.24 Route: /dashboard (middleware: auth, verified)

4) Database / Admin Management (Web UI)

#### 1.2.1.25 Company users & user restrictions

#### 1.2.1.26 Controllers (Web): app/Http/Controllers/Web/Database/CompanyUserController.php, UserRestrictionController.php

#### 1.2.1.27 Routes: resource routes under /database/company-users and /database/user-restrictions

5) Company & Organization Management

#### 1.2.1.28 Models: app/Models/Company.php, CompanyUser.php, CompanyUserRestriction.php

#### 1.2.1.29 Functionality: manage companies, link users to companies, enforce restrictions

6) Reports & Sheets

#### 1.2.1.30 Models: app/Models/Report.php, ReportRun.php, Sheet.php

#### 1.2.1.31 Functionality: define reports, queue/execute runs, view run history and results

7) Documents & Mailbox

#### 1.2.1.32 Models: app/Models/Document.php, Mailbox.php

#### 1.2.1.33 Functionality: upload/manage documents, mailbox processing

8) External Connections & Integrations

#### 1.2.1.34 Models: app/Models/DatabaseConnection.php, FtpConnection.php, AiProvider.php, AiSession.php

#### 1.2.1.35 Functionality: manage external connections, test connections, AI provider sessions

9) Activity Logging & Auditing

#### 1.2.1.36 Model: app/Models/ActivityLog.php

#### 1.2.1.37 Functionality: record user and system events; admin views for logs

10) Settings / Grid / UI Preferences

#### 1.2.1.38 Model: app/Models/GridSetting.php

#### 1.2.1.39 Functionality: save per-user grid/table preferences (columns, filters)

Cross-cutting concerns

#### 1.2.1.40 Authorization: spatie/laravel-permission (roles/permissions) — see User model and policies in app/Policies/

#### 1.2.1.41 API vs Web controllers: Web UI under app/Http/Controllers/Web/, APIs under app/Http/Controllers/Api/

#### 1.2.1.42 Mail templates: resources/views/emails/

#### 1.2.1.43 Tests: tests/Feature/ and tests/Unit/

How to use this file

#### 1.2.1.44 Keep this file updated as you add controllers, views, models or routes.

#### 1.2.1.45 When adding a new feature, add a short line under the relevant module linking the new file(s)/route(s).

Next improvement suggestions

#### 1.2.1.46 Produce an endpoints inventory (route, method, controller action) in a machine-readable format (openapi or JSON) for automated docs.

#### 1.2.1.47 Add a diagram (PlantUML or Mermaid) showing high-level module relationships.

Tools (editing Word docs)

#### 1.2.1.48 A VB.NET sample that edits a .docx using Xceed.Words.NET is available at `tools/vb-word-sample`.

#### 1.2.1.49 A Python fallback is available at `tools/python-docx-sample/edit\_docx.py` which edits `docs/modules.docx` and writes `docs/modules-edited.docx`.

#### 1.2.1.50 A PowerShell wrapper `tools/vb-word-sample/run-sample.ps1` attempts to run the VB sample if `dotnet` is present, otherwise runs the Python fallback. Use this from repository root to edit `docs/modules.docx`.

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