

# KasamaHR

## SaaS Tier Subscription Design

**Current state:** All tenants have unrestricted access to all 21 modules with no billing system.

**Target state:** 3 subscription tiers (Starter, Professional, Enterprise) + Custom plans with module-based gating, per-employee pricing, add-on slots, and PayMongo billing.

**Key design decision:** PayMongo has no Laravel Cashier equivalent and no native per-seat quantity parameter on subscriptions. We use per-tenant dynamic plans where each tenant gets a unique PayMongo plan with  $\text{amount} = \text{unit\_price} \times \text{employee\_count}$ , updated via API when employee count changes.

# Module Inventory

KasamaHR consists of 21 distinct functional modules:

#	Module	Description
1	HR Management	Employee records, HR analytics dashboard, company documents, announcements, certifications, document requests
2	Organization Management	Departments, positions, salary grades, work locations, org chart, work schedules, holidays, competencies
3	Time & Attendance	Attendance logs, daily time records (DTR), device management
4	Leave Management	Leave applications, approvals, calendar, leave types, balances, adjustments
5	Payroll	Payroll entries, periods, adjustments, loans, government contributions (SSS, PhilHealth, Pag-IBIG, Tax), payslips
6	HR Compliance & Reporting	SSS, PhilHealth, Pag-IBIG, BIR reports, BIR 2316 certificates
7	Employee Self-Service	Personal dashboard, DTR, payslips, documents, loans, leaves, goals, evaluations, training, compliance, onboarding
8	User & Access Management	User CRUD, roles, invitations, profile settings, password management, 2FA
9	Recruitment	Job requisitions, postings, candidates, applications, interviews, offer templates, offers, analytics
10	Onboarding & Pre-boarding	Pre-boarding/onboarding checklists, templates, tasks, self-service
11	Training & Development	Course catalog, training sessions, calendar, enrollments, history, categories, materials
12	Performance Management	Performance cycles, goals, KPIs, 360-degree evaluations, competency evaluations, development plans, analytics
13	Probationary Management	Criteria templates, evaluations, approvals, status tracking
14	Manager/Supervisor Module	Team goals management, probationary evaluations, team compliance tracking, leave approvals
15	Help Center	Help articles, categories, search, admin CMS
16	Compliance Training	Compliance dashboard, courses, assignments/rules, reports, assessments, certificates
17	Biometric Integration	Biometric device management, employee sync, sync logs, MQTT commands
18	Background Check & Reference	Background checks, documents, reference checks
19	Audit & Security	Audit logs (model change tracking), password confirmation
20	Careers/Public Portal	Public careers page, job listings, public applications, offer response portal
21	HR Analytics	HR analytics dashboard, workforce metrics, turnover analysis

# Subscription Tiers

## Starter — Core HR Operations (9 modules)

For small companies that need essential HR, payroll, and compliance tools. **■50/employee/mo** · Min 5 employees

Module	Included Features
HR Management	Employee CRUD, company documents, announcements, document requests
Organization Management	Departments, positions, salary grades, work locations, org chart, work schedules, holidays
Time & Attendance	Attendance logs, daily time records
Leave Management	Leave applications, approvals, calendar, types, balances, adjustments
Payroll	Entries, periods, adjustments, loans, government contributions, calculator, payslips
HR Compliance & Reporting	SSS, PhilHealth, Pag-IBIG, BIR reports, BIR 2316
Employee Self-Service (Basic)	My DTR, payslips, leave applications, loan applications, document requests, announcements, BIR 2316
User & Access Management	User CRUD, roles, invitations, profile settings, 2FA
Biometric Integration	Device management, employee sync, MQTT attendance logging (up to 2 devices)
Web Kiosk Clock-In (planned)	Browser-based clock-in/out terminal for offices without biometric hardware

## Professional — Strategic HR (Starter + 8 modules)

For growing companies that need recruitment, training, and performance tools. **■100/employee/mo** · Min 10 employees. Everything in Starter (with up to 10 biometric devices), plus:

Module	Included Features
Recruitment	Job requisitions, postings, candidates, applications, interviews, offer templates, offers, analytics
Onboarding & Pre-boarding	Checklists, templates, tasks, employee self-service
Training & Development	Course catalog, sessions, calendar, enrollments, history, categories, materials
Performance Management	Cycles, goals, KPIs, 360 evaluations, competency evaluations, development plans, analytics
Probationary Management	Criteria templates, evaluations, approvals, status view
Manager/Supervisor Module	Team goals, probationary evaluations, leave approvals
Help Center	Articles, categories, search, admin CMS
HR Analytics	Workforce metrics, turnover analysis, HR analytics dashboard

## Enterprise — Full Platform (Professional + 4 modules)

For large organizations needing full compliance, integrations, and employer branding. **■150/employee/mo** · Min 25 employees. Everything in Professional (unlimited biometric devices), plus:

Module	Included Features
Compliance Training	Dashboard, courses, assignments/rules, reports, assessments, certificates
Background Check & Reference	Background checks, documents, reference checks
Audit & Security	Audit logs, password confirmation for sensitive actions

Module	Included Features
Careers/Public Portal	Public careers page, job listings, public applications, offer responses

## Feature Comparison Matrix

Capability	Starter	Professional	Enterprise	Custom
<b>Core HR</b>				
Employee Records	✓	✓	✓	Pick & choose
Organization Management	✓	✓	✓	Pick & choose
Time & Attendance	✓	✓	✓	Pick & choose
Biometric Integration	✓	✓	✓	Pick & choose
Web Kiosk Clock-In (planned)	✓	✓	✓	Pick & choose
Leave Management	✓	✓	✓	Pick & choose
Payroll Processing	✓	✓	✓	Pick & choose
Government Compliance	✓	✓	✓	Pick & choose
Employee Self-Service (Basic)	✓	✓	✓	Pick & choose
<b>Strategic HR</b>				
Recruitment & Hiring	—	✓	✓	Pick & choose
Onboarding & Pre-boarding	—	✓	✓	Pick & choose
Training & Development	—	✓	✓	Pick & choose
Performance Management	—	✓	✓	Pick & choose
Probationary Management	—	✓	✓	Pick & choose
Manager/Supervisor Tools	—	✓	✓	Pick & choose
Help Center	—	✓	✓	Pick & choose
HR Analytics Dashboard	—	✓	✓	Pick & choose
<b>Enterprise</b>				
Compliance Training	—	—	✓	Pick & choose
Background & Reference Checks	—	—	✓	Pick & choose
Audit Logs	—	—	✓	Pick & choose
Public Careers Portal	—	—	✓	Pick & choose
<b>Limits</b>				
Max Employees	50	250	Unlimited	Negotiated
Max Admin/HR Users	3	10	Unlimited	Negotiated
Max Departments	5	Unlimited	Unlimited	Negotiated
Max Biometric Devices	2	10	Unlimited	Negotiated
Storage	1 GB	10 GB	100 GB	Negotiated
<b>Extras</b>				

Capability	Starter	Professional	Enterprise	Custom
API Access	—	Read-only	Full	Negotiated
Custom Branding	Logo only	Logo + Colors	White-Label	Negotiated
SSO / SAML	—	—	✓	Negotiated
Support	Email	Priority Email	Dedicated AM	Dedicated AM

## Pricing Strategy

**Model:** Per-employee, per-month pricing with annual discount.

Tier	Monthly (per employee)	Annual (per employee)	Savings
Starter	■50/mo	■500/yr	~17%
Professional	■100/mo	■1,000/yr	~17%
Enterprise	■150/mo	■1,500/yr	~17%
Custom	Negotiated	Negotiated	Per-deal

**Minimums:** Starter: 5 employees (■250/mo) · Professional: 10 employees (■1,000/mo) · Enterprise: 25 employees (■3,750/mo)

**PayMongo implementation:** Each tenant gets a unique PayMongo plan with amount = price\_per\_unit × max(employee\_count, tier\_minimum). When employee count changes, the plan amount is updated via PUT /v1/subscriptions/plans/{plan\_id}.

## Add-On Slots

Tenants can purchase additional capacity beyond their plan's included limits without upgrading to a higher tier.

Add-On	Unit	Price (per unit/mo)	Available On
Extra Employee Slots	Pack of 10 employees	■25/mo	Starter, Professional
Extra Biometric Devices	1 device	■50/mo	Starter, Professional

**Effective limit calculation:**

```
effective_max_employees = plan.max_employees + (extra_employee_packs × 10)
effective_max_devices = plan.max_biometric_devices + extra_device_count
```

**Downgrade validation:** When removing add-ons or downgrading plans, the system validates that current usage fits within the new effective limits.

## Custom Plan (Sales-Assisted)

For organizations that need specific modules at scale but don't fit neatly into standard tiers — e.g., a production plant with 2,000 employees that only needs core HR, payroll, and attendance.

Setting	Description
Plan Name	Custom label (e.g., "Starter XL — Acme Manufacturing")
Modules	Cherry-picked from the full 21-module list
Per-Employee Rate	Negotiated price (e.g., ■35/mo for high-volume)
Employee Limit	Set per-deal (e.g., 5,000) or unlimited
Biometric Device Limit	Set per-deal or unlimited
Billing Method	PayMongo subscription or manual invoice
Contract Term	Monthly, annual, or custom (e.g., 2-year contract)

**Example — Acme Manufacturing:** 2,000 factory workers needing only 9 Starter modules. Custom deal: ■35/employee/mo × 2,000 = ■70,000/mo (vs ■100,000/mo on standard Starter).

Custom plans use the same plans table with is\_custom = true. Modules assigned via plan\_modules pivot. Not shown on public pricing — only assignable by super admins.

## Trial Period

Setting	Value
Duration	14 days
Tier	Professional (gives users the strategic HR experience)
Credit card required	No
Expiration	Auto-locks access, requiring plan selection to continue
Enterprise trial	Available by request (sales-assisted, 30 days)

## PayMongo Limitations

Limitation	Impact
No proration	Plan changes take effect on the next billing cycle, not mid-cycle
24-hour payment window	First subscription payment must complete within 24 hours or PayMongo auto-cancels
Cards + Maya only	Subscriptions only support card and Maya payment methods
No customer portal	Must build billing management UI in-app (no Stripe Portal equivalent)
No native per-seat billing	Must use per-tenant dynamic plans with manual amount updates

## Architecture Decisions

Decision	Choice	Rationale
Payment provider	PayMongo	Philippine-focused, PHP currency native, local payment methods
Billable entity	Tenant model	Billing is per-organization, not per-user
Billing DB	Platform database	Billing data is cross-tenant; must survive tenant DB operations
Module gating	Middleware + Inertia shared props	Matches existing EnsureRole pattern
Plan-module mapping	Database table (plan_modules)	Allows runtime customization without deploys
Employee billing	Per-tenant dynamic plans	PayMongo has no per-seat quantity
Trial tracking	trial_ends_at on tenant	Allows trial without payment method upfront
Data on downgrade	Preserved, access restricted	Never delete customer data
Webhook endpoint	Main domain (/paymongo/webhook)	Webhooks are application-wide
Frontend gating	useSubscription composable	Follows existing useTenant pattern
Connection pattern	getConnectionName() method	Matches existing Tenant model pattern

## Module Enum

app/Enums/Module.php — 21 module identifiers grouped by tier:

```
enum Module: string {  
    // Starter  
    case HrManagement = 'hr_management';  
    case OrganizationManagement = 'organization_management';  
    case TimeAttendance = 'time_attendance';  
    case BiometricIntegration = 'biometric_integration';  
    case LeaveManagement = 'leave_management';  
    case Payroll = 'payroll';  
    case HrCompliance = 'hr_compliance';  
    case EmployeeSelfService = 'employee_self_service';  
    case UserAccessManagement = 'user_access_management';  
    // Professional  
    case Recruitment = 'recruitment';  
    case OnboardingPreboarding = 'onboarding_preboarding';  
    case TrainingDevelopment = 'training_development';  
    case PerformanceManagement = 'performance_management';  
    case ProbationaryManagement = 'probationary_management';  
    case ManagerSupervisor = 'manager_supervisor';  
    case HelpCenter = 'help_center';  
    case HrAnalytics = 'hr_analytics';  
    // Enterprise  
    case ComplianceTraining = 'compliance_training';  
    case BackgroundCheckReference = 'background_check_reference';  
    case AuditSecurity = 'audit_security';  
    case CareersPortal = 'careers_portal';  
}
```



## Subscription Status Enum

```
enum SubscriptionStatus: string {
  case Active = 'active';
  case PastDue = 'past_due';
  case Unpaid = 'unpaid';
  case Cancelled = 'cancelled';
  case Incomplete = 'incomplete';
  case IncompleteCancelled = 'incomplete_cancelled';
}
```

## Add-On Type Enum

```
enum AddonType: string {
  case EmployeeSlots = 'employee_slots'; // 10 employees per qty
  case BiometricDevices = 'biometric_devices'; // 1 device per qty
}
```

## Database Schema

All billing tables live in the **platform database** alongside tenants and users.

### plans table

```
Schema::create('plans', function (Blueprint $table) {
  $table->id();
  $table->string('name');
  $table->string('slug')->unique();
  $table->text('description')->nullable();
  $table->boolean('is_active')->default(true);
  $table->boolean('is_custom')->default(false);
  $table->foreignId('tenant_id')->nullable()->constrained()->nullOnDelete();
  $table->integer('sort_order')->default(0);
  $table->json('limits');
  $table->timestamps();
});
```

### plan\_prices table

```
Schema::create('plan_prices', function (Blueprint $table) {
  $table->id();
  $table->foreignId('plan_id')->constrained()->cascadeOnDelete();
  $table->string('billing_interval'); // 'monthly' or 'yearly'
  $table->integer('price_per_unit'); // centavos
  $table->string('currency')->default('PHP');
  $table->boolean('is_active')->default(true);
  $table->timestamps();
});
```

### plan\_modules table

```
Schema::create('plan_modules', function (Blueprint $table) {
  $table->id();
  $table->foreignId('plan_id')->constrained()->cascadeOnDelete();
  $table->string('module');
  $table->timestamps();
  $table->unique(['plan_id', 'module']);
});
```

### subscriptions table

```
Schema::create('subscriptions', function (Blueprint $table) {
  $table->id();
  $table->foreignId('tenant_id')->constrained()->cascadeOnDelete();
  $table->string('name')->default('default');
  $table->string('paymongo_id')->unique()->nullable();
  $table->string('paymongo_plan_id')->nullable();
  $table->string('paymongo_status')->nullable();
});
```

```
$table->foreignId('plan_price_id')->nullable()->constrained()->nullOnDelete();
$table->integer('quantity')->default(1);
$table->timestamp('current_period_end')->nullable();
$table->timestamp('ends_at')->nullable();
$table->timestamps();
$table->index(['tenant_id', 'paymongo_status']);
});
```

## tenant\_addons table

```
Schema::create('tenant_addons', function (Blueprint $table) {
    $table->id();
    $table->foreignId('tenant_id')->constrained()->cascadeOnDelete();
    $table->string('type'); // AddonType enum
    $table->integer('quantity')->default(1);
    $table->integer('price_per_unit'); // centavos
    $table->string('currency')->default('PHP');
    $table->boolean('is_active')->default(true);
    $table->timestamp('expires_at')->nullable();
    $table->timestamps();
    $table->index(['tenant_id', 'type', 'is_active']);
});
```

## Additions to tenants table

```
Schema::table('tenants', function (Blueprint $table) {
    $table->foreignId('plan_id')->nullable()->constrained()->nullOnDelete();
    $table->string('paymongo_customer_id')->nullable()->unique();
    $table->timestamp('trial_ends_at')->nullable();
    $table->integer('employee_count_cache')->default(0);
});
```

## Feature Gate Service

app/Services/FeatureGateService.php — Centralized service for all module and limit access checks.

```
class FeatureGateService {
    public function hasModule(Module $module): bool
    public function hasAnyModule(Module ...$modules): bool
    public function availableModules(): array
    public function isWithinEmployeeLimit(): bool // uses effectiveLimit
    public function isWithinUserLimit(): bool
    public function isWithinDeviceLimit(): bool // uses effectiveLimit
    public function getEffectiveLimit(string $key): int|null
    public function addonCostBreakdown(): array
}
```

## Module Gating Middleware

Two middlewares registered in bootstrap/app.php:

- **EnsureModuleAccess** — Route-level module gate. Returns 403 or redirects to upgrade page.
- **EnsureActiveSubscription** — Checks tenant→hasActiveAccess(). Allows billing routes through.

Modules requiring middleware:

Module Enum Value	Route Prefix/File
recruitment	routes/tenant/web-recruitment.php, api-recruitment.php
onboarding_preboarding	Onboarding route groups
training_development	Training route groups, api-training.php
performance_management	Performance route groups, api-performance.php
probationary_management	Probationary route groups
manager_supervisor	Manager-specific route groups
help_center	Help center route groups
hr_analytics	HR analytics route groups
compliance_training	Compliance route groups
background_check_reference	Background check route groups
audit_security	Audit log route groups
careers_portal	Careers/public portal route groups

## Frontend Gating

Inertia shared props include plan, subscription, available\_modules, effective\_limits, and active\_addons. Vue composables **useSubscription** provides hasModule(), isSubscribed, isOnTrial, planName, planSlug, trialEndsAt.

## Billing Routes & Controller

routes/tenant/web-billing.php — Protected by ensure-role:admin middleware.

Action	Description
index	Billing dashboard — current plan, usage stats, subscription status, invoices
plans	Plan comparison page with pricing toggle (monthly/yearly)
upgrade	Upgrade prompt shown when accessing locked modules

Action	Description
subscribe	Ensures PayMongo customer, creates checkout URL, redirects to PayMongo checkout
changePlan	Validates limits for downgrades, calls changePlan() on service
cancel	Calls cancel, redirects back with confirmation
invoices	List all invoices
success	Post-checkout confirmation callback
addons	Add-ons management page: current add-ons, available add-ons, usage vs limits
purchaseAddon	Validates tier eligibility, creates TenantAddon, updates PayMongo plan amount
updateAddon	Changes add-on quantity, validates usage before decrease
cancelAddon	Validates usage fits within reduced limits, deactivates on next billing cycle

## Notifications

Notification	Trigger	Recipients
TrialExpiredNotification	Trial period ends with no subscription	Tenant admin users
PaymentFailedNotification	PayMongo invoice payment fails	Tenant admin users
SubscriptionCancelledNotification	Subscription cancelled	Tenant admin users

# Implementation Phases

## Phase 1 (Week 1–2)

### Foundation

- Install PayMongo PHP SDK
- Create Module and SubscriptionStatus enums
- Create config/billing.php and config/paymongo.php
- Create platform migrations for plans, plan\_prices, plan\_modules, subscriptions
- Create migration to add billing columns to tenants table
- Create Plan, PlanPrice, PlanModule, Subscription, TenantAddon models
- Add plan/subscription relationships to Tenant model
- Create PlanSeeder
- Create model factories

## Phase 2 (Week 2–3)

### PayMongo Service Layer

- Create PayMongoServiceProvider
- Create PayMongoCustomerService
- Create PayMongoSubscriptionService
- Create PayMongoWebhookService
- Create PayMongoService facade

## Phase 3 (Week 3–4)

### Feature Gating

- Create FeatureGateService
- Create EnsureModuleAccess and EnsureActiveSubscription middleware
- Register in bootstrap/app.php
- Apply module middleware to route groups
- Extend HandleInertiaRequests
- Create useSubscription composible
- Update TenantSidebar.vue

## Phase 4 (Week 4–5)

### Billing UI & Routes

- Create BillingController with Inertia pages
- Create billing pages (Index, Plans, Upgrade, Addons, Success)
- Create PayMongoWebhookController
- Add billing and webhook routes
- Modify TenantRegistrationController for trial provisioning

## **Phase 5 (Week 5–6)**

### **Employee Sync, Limits, Notifications**

- Create UpdateBillingQuantity job
- Create EmployeeObserver
- Create SyncBillingQuantities and CheckExpiredTrials commands
- Add employee/user/device limit checks
- Integrate add-on costs into billing
- Create notification classes

## **Phase 6 (Week 6–7)**

### **Platform Admin Dashboard**

- Create PlatformAdminController
- Create admin layout and Vue pages
- Add /admin/\* routes
- Implement tenant management actions
- Create custom plan CRUD
- Implement impersonation

## **Phase 7 (Week 7–8)**

### **Testing**

- Feature tests for FeatureGateService, middleware, BillingController, TenantAddon, webhooks, employee sync, trials, PlanSeeder, PlatformAdminController

## New Files

File	Purpose
app/Enums/AddonType.php	Add-on type identifiers
app/Enums/Module.php	21 module identifiers
app/Enums/SubscriptionStatus.php	PayMongo subscription statuses
app/Models/Plan.php	Subscription plan model
app/Models/PlanPrice.php	Plan price model
app/Models/PlanModule.php	Plan-to-module pivot
app/Models/Subscription.php	Local subscription record
app/Models/TenantAddon.php	Tenant add-on slots model
app/Services/FeatureGateService.php	Centralized module/limit checks
app/Services/Billing/PayMongoService.php	Main billing facade
app/Services/Billing/PayMongoCustomerService.php	Customer CRUD
app/Services/Billing/PayMongoSubscriptionService.php	Subscription lifecycle
app/Services/Billing/PayMongoWebhookService.php	Webhook validation/routing
app/Providers/PayMongoServiceProvider.php	PayMongo client singleton
app/Http/Middleware/EnsureModuleAccess.php	Route-level module gate
app/Http/Middleware/EnsureActiveSubscription.php	Active subscription gate
app/Http/Controllers/BillingController.php	Billing management UI
app/Http/Controllers/PayMongoWebhookController.php	Webhook handler
app/Http/Controllers/PlatformAdminController.php	Platform admin dashboard
app/Jobs/UpdateBillingQuantity.php	Sync employee count to PayMongo
app/Console/Commands/SyncBillingQuantities.php	Nightly billing sync
app/Console/Commands/CheckExpiredTrials.php	Trial expiration handler
app/Observers/EmployeeObserver.php	Trigger billing sync
app/Notifications/TrialExpiredNotification.php	Trial expiry notice
app/Notifications/PaymentFailedNotification.php	Failed payment notice
app/Notifications/SubscriptionCancelledNotification.php	Cancellation notice
config/billing.php	Billing configuration
config/paymongo.php	PayMongo credentials
database/migrations/xxxx_create_billing_tables.php	All billing schema
database/seeders/PlanSeeder.php	Seed default plans
database/factories/PlanFactory.php	Plan test factory
database/factories/SubscriptionFactory.php	Subscription test factory
database/factories/TenantAddonFactory.php	TenantAddon test factory

File	Purpose
resources/js/composables/useSubscription.ts	Frontend subscription composable
resources/js/pages/Billing/Index.vue	Billing dashboard
resources/js/pages/Billing/Plans.vue	Plan comparison
resources/js/pages/Billing/Upgrade.vue	Upgrade prompt
resources/js/pages/Billing/Addons.vue	Add-ons management
resources/js/pages/Billing/Success.vue	Post-checkout confirmation
routes/tenant/web-billing.php	Billing routes
resources/js/pages/Admin/Dashboard.vue	Platform overview stats
resources/js/pages/Admin/Tenants/Index.vue	Searchable tenant list
resources/js/pages/Admin/Tenants/Show.vue	Tenant detail + actions
resources/js/pages/Admin/Plans/Create.vue	Custom plan creation
resources/js/pages/Admin/Plans/Edit.vue	Custom plan edit
resources/js/layouts/admin/Layout.vue	Admin sidebar layout



## Modified Files

File	Change
app/Models/Tenant.php	Add plan/subscription relationships, trial methods, PayMongo fields
app/Http/Middleware/HandleInertiaRequests.php	Share plan/subscription/modules in tenant context
resources/js/components/TenantSidebar.vue	Add hasModule() visibility checks
bootstrap/app.php	Register module and subscribed middleware aliases
app/Http/Controllers/TenantRegistrationController.php	Assign trial plan + create PayMongo customer
app/Http/Controllers/EmployeeController.php	Add employee limit check in store()
app/Providers/AppServiceProvider.php	Register EmployeeObserver
routes/tenant.php	Include billing routes file
routes/platform.php	Add webhook route + /admin/* route group
routes/console.php	Schedule billing commands

## Platform Admin Dashboard

A centralized dashboard on the main platform domain for super admins to manage all tenants, subscriptions, and billing. Access: super admins only, via `Gate::authorize('super-admin')`.

### Admin Actions

Action	Description
Dashboard	Overview stats: Total Tenants, Active Subscriptions, Active Trials, MRR, Subscriptions by Plan, Trial Conversion Rate
Tenant List	Searchable/filterable list with plan, status, employee count
Tenant Detail	Full info, subscription history, usage vs limits, admin users
Extend Trial	Extend trial period by specified days
Assign Plan	Manually assign/change plan (sales-assisted or support overrides)
Cancel Subscription	Administratively cancel subscription
Create Custom Plan	Module picker, per-employee rate, limits, optional tenant assignment
Edit Custom Plan	Edit modules, limits, pricing for custom plans
Impersonate	Access tenant dashboard via secure redirect token

# Web Kiosk Clock-In

A browser-based clock-in/clock-out terminal providing a software-only alternative to biometric hardware. Available on all tiers (Starter, Professional, Enterprise).

## Kiosk Terminal Mode

**URL:** `https://{tenant}.kasamahr.com/kiosk` (or unique kiosk token URL)

**Flow:** Employee enters 4–6 digit Kiosk PIN → System shows name, photo & employee code → Employee taps Clock In / Clock Out → AttendanceLog created with source = 'kiosk' → Returns to PIN screen.

## Kiosk PIN

A dedicated numeric PIN for kiosk identification, separate from login credentials.

Property	Details
Auto-generated	Random 4–6 digit PIN when kiosk feature is enabled or new employee created
Stored securely	Hashed in database using Hash::make()
Shown once	Via Self-Service dashboard under "My DTR → Kiosk PIN", or by HR during onboarding
Resettable	By the employee (Self-Service) or by an HR admin
Unique per tenant	Validated with unique constraint scoped to tenant database

**Why not use employee code?** Employee codes are often sequential (EMP-0001, EMP-0002) and easy to guess. A random PIN prevents colleagues from clocking in for each other.

## Location Verification (Self-Service)

Self-service clock-in can be restricted by location. Admins configure per work location:

Method	How It Works	Best For	Spoofing Risk
IP Whitelist	Check IP against approved office IPs	Office workers	Low
Geofencing (GPS)	Browser Geolocation API + Haversine distance	Field/mobile workers	Medium
Both required	Must pass IP AND GPS	High-security	Low
Any (default)	Pass either IP or GPS	Flexible policies	Medium
None	No restriction	Fully remote teams	N/A

## Anti-Fraud Measures

Measure	Description
Kiosk PIN	Random numeric PIN per employee, hashed — not guessable like sequential codes
IP Whitelisting	Restrict kiosk and self-service clock-in to specific office IPs
Geofencing (GPS)	Require GPS location within radius; reject if accuracy > 200m
GPS Accuracy Check	Reject readings with poor accuracy (spoofed locations report 0 or extreme)
Photo Capture	Take webcam photo on clock-in for verification

Measure	Description
Cooldown Period	Prevent duplicate clock-ins within configurable window (e.g., 5 min)
Admin Audit	All kiosk/self-service entries flagged with source for filtering
Device Binding	Bind kiosk sessions to specific device fingerprints
Anomaly Detection	Flag impossible location jumps within short time windows

## Kiosk Limits per Tier

Tier	Max Kiosks
Starter	1
Professional	5
Enterprise	Unlimited

**Implementation priority:** This feature should be implemented after the core SaaS billing system (Phases 1–7) is complete. Suggested timeline: Phase 8, Week 9–10.