Karyalay ERP: Roles & Permissions Module – Functional Specification Document

---

🔐Module Name: Roles & Permissions

The **Roles & Permissions Module** provides a secure, flexible, and page-level access control system for **Karyalay ERP**. It enables the creation of multiple roles, assignment of fine-grained permissions per page, and enforcement of restricted access through automatic redirection to fallback pages when unauthorized actions are attempted.

📆1. Functional Overview

This module ensures that every user’s access is governed by clearly defined permissions tied to their assigned role. It allows the **Admin** to: - Create and customize roles with specific permissions per page. - Assign roles to employees. - Modify access rights without altering the codebase. - Enforce page-level and CRUD-level (Create, Read, Update, Delete) access.

Unauthorized page access attempts result in automatic redirection to a configured fallback page (e.g., *Access Denied* or *Not Authorized* page).

🧱2. Feature List

Admin/Owner Side:

- Create, edit, or delete roles.
- Define access levels (View, Create, Edit, Delete, Export, Approve).
- Assign and revoke permissions page-by-page.
- Assign users to one or more roles.
- Manage default redirect fallback pages for each restricted area.

Employee/Manager Side:

- Access only permitted pages and actions.
- View their assigned role and permissions summary.
- Attempted unauthorized actions trigger fallback redirect automatically.

🧮3. Database Schema

Table: `roles`

Field	Type	Description
id	INT, PK, AI	Unique role ID
name	VARCHAR(100)	Role name (e.g., Admin, Manager, Employee)
description	TEXT	Optional role description
created_at	TIMESTAMP	Creation time
updated_at	TIMESTAMP NULL	Last updated time

Table: `permissions`

Field	Type	Description
id	INT, PK, AI	Unique permission ID
page_name	VARCHAR(150)	Page identifier (e.g., crm/leads, crm/tasks)
can_view	BOOLEAN DEFAULT 0	Permission to view the page
can_create	BOOLEAN DEFAULT 0	Permission to create entries
can_edit	BOOLEAN DEFAULT 0	Permission to edit entries
can_delete	BOOLEAN DEFAULT 0	Permission to delete entries
can_export	BOOLEAN DEFAULT 0	Permission to export data
can_approve	BOOLEAN DEFAULT 0	Permission to approve actions
fallback_page	VARCHAR(150)	Redirect route if unauthorized
created_at	TIMESTAMP	Created timestamp
updated_at	TIMESTAMP NULL	Updated timestamp

Table: `role_permissions`

Field	Type	Description
id	INT, PK, AI	Unique ID
role_id	INT, FK	Linked role from `roles` table
permission_id	INT, FK	Linked permission record from `permissions` table

Table: `user_roles`

Field	Type	Description
id	INT, PK, AI	Unique mapping ID
user_id	INT, FK	Employee or system user ID
role_id	INT, FK	Assigned role ID
assigned_at	TIMESTAMP	Assignment timestamp

⚙️4. Permission Levels (CRUD + Extra)

Each page permission record defines one or more of the following access types:

Access Type	Description
View	Allows reading and listing page data
Create	Allows adding new records or entries
Edit	Allows modifying existing records
Delete	Allows removing or archiving records
Export	Allows exporting records to CSV, PDF, or Excel
Approve	Allows verifying, approving, or marking records as final

Each permission check is performed both on the **frontend (UI)** and **backend (API)** layers to prevent bypassing via direct URL entry.

🧱5. Frontend Pages

Page	URL Route	Description	Access Role
Roles Manager	`/settings/roles`	List and manage all roles	Admin
Add/Edit Role	`/settings/roles/edit/:id`	Create or modify roles	Admin
Permissions Manager	`/settings/permissions`	View and assign permissions	Admin
Assign Roles	`/settings/assign-roles`	Map users to specific roles	Admin

Page	URL Route	Description	Access Role
Access Denied Page	`/unauthorized`	Fallback page for unauthorized access	All

🗜️6. Backend Routes (PHP Endpoints)

Method	Route	Purpose	Auth
GET	`/api/roles`	Fetch all roles	Admin
POST	`/api/roles/add`	Create new role	Admin
POST	`/api/roles/update/:id`	Update role info	Admin
DELETE	`/api/roles/delete/:id`	Delete a role	Admin
GET	`/api/permissions`	List all permissions	Admin
POST	`/api/permissions/update`	Update permission matrix	Admin
GET	`/api/user-roles/:id`	Fetch user’s assigned roles	Authenticated user

🪠7. Access Control Flow

1. When a user attempts to access a page, their assigned role is fetched.
2. The system checks corresponding permission flags from `role_permissions`.
3. If permission exists → grant access.
4. If not → redirect to `fallback_page` defined in `permissions`.

Example: - Unauthorized user tries `/crm/leads/edit/5` → redirected to `/unauthorized`. - The fallback page shows message: *“You do not have permission to access this page.”*

📊8. Admin Utilities & Logs

- View all user-role assignments with timestamps.
- Audit log of permission changes (who modified what and when).
- Export current permission matrix.
- Reset to default permission templates (Admin, Manager, Employee).

🔗9. Integration with Other Modules

Module	Integration Type	Purpose
Employees	Direct	Role assignment via employee profile
CRM / Expense / Visitor / Document Vault	Access Control	Page-wise permission enforcement
Branding Panel	Partial	Restrict theme or branding settings to Admin only

🔍10. Validations & Rules

- Each user must have at least one assigned role.
- Deleting a role automatically revokes access for its assigned users.
- Fallback pages must exist and be accessible to all.
- No role should have unrestricted access by default (must be explicitly configured).

📧11. Notifications (Optional Future Enhancement)

Trigger	Type	Recipient	Message
Role Assigned	Email	Employee	You've been assigned the role: [Role Name]
Permission Updated	Email	Admin	Role [Role Name] permissions updated successfully

⏳12. Future Enhancements (Phase 2)

- Module-based grouping for quick role setup
- Role cloning and bulk permission update
- Permission inheritance (e.g., Manager inherits Employee access)
- Time-based access restrictions (temporary access)
- Role activity analytics (most accessed modules per role)

🔗Summary

The **Roles & Permissions Module** enforces strict, customizable access control throughout Karyalay ERP. It provides a transparent and easily maintainable system where every page, action, and data operation is secured under a defined role, ensuring **data safety, workflow consistency, and organizational accountability**.

This system supports page-level granularity, per-user assignments, and automatic redirection on unauthorized access — perfectly balancing **flexibility** with **security** for SME environments.