**🎯 COMPREHENSIVE VISION IMPLEMENTATION**

**PHASE 1: CORE FOUNDATION**

✅ **Data-Driven RBAC**

* 5 core database tables (Users, Roles, Modules, Role\_Permissions, Tenants)
* No hardcoded permissions
* Plug-and-play module architecture

✅ **Two-Space Architecture**

* System Space (Platform administration)
* Tenant Space (Organization operations)
* Complete isolation and security

✅ **Database Design**

* Single users table (system + tenant)
* Unified role system
* Dynamic module registry
* Permission mapping layer

**PHASE 2: SECURITY EXCELLENCE**

✅ **Authentication & Authorization**

* JWT tokens with refresh mechanism
* Bcrypt password hashing
* Session management
* MFA support (optional)

✅ **Zero-Trust Model**

* Default deny principle
* Whitelist-only permissions
* Every request verified
* Role inheritance support

✅ **Tenant Isolation**

* Complete data separation
* SQL-level filtering
* API-level enforcement
* Audit trail per tenant

✅ **Compliance & Governance**

* Complete audit logs (who, what, when, why)
* Separation of duties (SoD)
* Permission delegation
* Change tracking
* GDPR compliance features

**PHASE 3: DEVELOPER EXPERIENCE**

✅ **Modular Design**

* Add modules without touching core code
* Self-registering menu system
* Automatic UI updates
* Framework handles all RBAC

✅ **Easy Extensibility**

* New module = register menu key + actions
* Frontend automatically shows new options
* Backend automatically enforces permissions
* Zero configuration needed

✅ **Type-Safe Development**

* TypeScript throughout
* Strict typing
* Better IDE support
* Fewer runtime errors

✅ **Documentation**

* Complete API docs
* Component guides
* Architecture overview
* Setup instructions
* Code examples

**PHASE 4: PERFORMANCE & SCALABILITY**

✅ **Optimization**

* Permission caching (Redis ready)
* Database query optimization
* Connection pooling
* Pagination & lazy loading
* Minimal bundle size (Angular 20)

✅ **Horizontal Scaling**

* Stateless API design
* Load balancer ready
* Docker container orchestration
* Database replication ready
* CDN compatible

✅ **Real-Time Features**

* WebSocket support
* Live notifications
* Instant permission updates
* Activity feeds

**PHASE 5: MODERN UI/UX**

✅ **Design System**

* Compact, minimal layout
* Professional typography
* Consistent spacing
* Smooth animations

✅ **Dual Themes**

* Light mode (default)
* Dark mode
* System preference detection
* User preference persistence

✅ **Responsive Design**

* Mobile-first approach
* Tablet optimization
* Desktop excellence
* Touch-friendly
* Keyboard navigation

✅ **UI Components**

* Reusable component library
* Forms, tables, modals, badges
* Loading states, empty states
* Error handling
* Toast notifications

**PHASE 6: MULTI-TENANCY**

✅ **Complete Isolation**

* Tenant-specific databases (or schemas)
* Data firewall at every level
* Tenant-specific branding
* Tenant-specific settings

✅ **Tenant Management**

* Self-service onboarding
* Usage tracking
* Billing integration ready
* Automatic provisioning
* Data residency options

✅ **Cross-Tenant Features**

* System admin access (with audit)
* Support staff access (read-only)
* Delegated admin (temporary)
* Audit across tenants

**PHASE 7: OBSERVABILITY**

✅ **Monitoring & Logging**

* Structured logging (JSON)
* Log levels (debug, info, warn, error)
* Request tracking (correlation IDs)
* Performance metrics
* Error tracking

✅ **Audit Trails**

* Every action logged
* User identification
* Timestamp precision
* Before/after values
* Compliance reports

✅ **Dashboards**

* System health
* User activity
* Permission usage
* Performance metrics
* Audit summaries

**PHASE 8: DEVOPS & DEPLOYMENT**

✅ **Containerization**

* Docker for all services
* Docker Compose for local dev
* Production-ready configs
* Health checks
* Graceful shutdown

✅ **CI/CD Pipelines**

* GitHub Actions workflows
* Automated testing
* Code quality checks
* Build automation
* Deployment automation

✅ **Infrastructure**

* Database migrations
* Seed data automation
* Backup strategies
* Disaster recovery
* Zero-downtime deployments

**PHASE 9: TESTING**

✅ **Test Coverage**

* Unit tests (services, utils)
* Integration tests (API endpoints)
* E2E tests (user flows)
* Jest configuration
* Test data factories

✅ **Quality Assurance**

* Code coverage reports
* Performance testing
* Security testing
* Load testing ready
* Accessibility testing

**PHASE 10: DOCUMENTATION**

✅ **Complete Documentation**

* Architecture guide
* RBAC system guide
* API documentation
* Database schema docs
* Deployment guide
* Security guide
* Troubleshooting guide

✅ **Examples & Templates**

* Example modules (Loans, Customers)
* Code snippets
* API requests (Postman collection)
* SQL scripts
* Configuration templates

**🚀 READY TO BUILD?**

This comprehensive vision includes:

✅ **Express API** - 7 controllers, 6 services, complete RBAC middleware  
✅ **Angular Web** - 15+ components, dark/light theme, admin dashboards  
✅ **Ionic Mobile** - MyPortal client with touch optimization  
✅ **PostgreSQL** - 8+ tables, migrations, seed data  
✅ **DevOps** - Docker Compose, GitHub Actions, monitoring  
✅ **Tests** - Jest setup, unit & integration tests  
✅ **Docs** - Complete guides, API docs, examples  
✅ **Security** - RBAC, tenant isolation, audit trails  
✅ **Performance** - Caching, optimization, scaling  
✅ **UX** - Modern, compact, responsive, accessible

ExITS-SaaS-Template/

│

├── api/ # Express Backend

│ ├── src/

│ │ ├── config/

│ │ │ ├── database.js

│ │ │ ├── env.js

│ │ │ └── constants.js

│ │ ├── middleware/

│ │ │ ├── auth.middleware.js

│ │ │ ├── rbac.middleware.js

│ │ │ ├── tenant-isolation.middleware.js

│ │ │ ├── error.middleware.js

│ │ │ └── logger.middleware.js

│ │ ├── controllers/

│ │ │ ├── auth.controller.js

│ │ │ ├── user.controller.js

│ │ │ ├── role.controller.js

│ │ │ ├── module.controller.js

│ │ │ ├── tenant.controller.js

│ │ │ ├── audit.controller.js

│ │ │ ├── permission.controller.js

│ │ │ └── delegation.controller.js

│ │ ├── routes/

│ │ │ ├── auth.routes.js

│ │ │ ├── user.routes.js

│ │ │ ├── role.routes.js

│ │ │ ├── module.routes.js

│ │ │ ├── tenant.routes.js

│ │ │ ├── audit.routes.js

│ │ │ └── index.js

│ │ ├── services/

│ │ │ ├── auth.service.js

│ │ │ ├── user.service.js

│ │ │ ├── role.service.js

│ │ │ ├── permission.service.js

│ │ │ ├── tenant.service.js

│ │ │ ├── cache.service.js

│ │ │ └── audit.service.js

│ │ ├── scripts/

│ │ │ ├── schema.sql

│ │ │ ├── migrate.js

│ │ │ ├── seed.js

│ │ │ └── migrate-fresh.js

│ │ ├── utils/

│ │ │ ├── logger.js

│ │ │ ├── jwt.js

│ │ │ └── validators.js

│ │ ├── server.js

│ │ └── index.js

│ ├── tests/

│ │ ├── auth.test.js

│ │ ├── rbac.test.js

│ │ ├── user.test.js

│ │ └── setup.js

│ ├── .env.example

│ ├── .env.test

│ ├── package.json

│ ├── jest.config.js

│ ├── Dockerfile

│ └── README.md

│

├── web/ # Angular Admin Dashboard

│ ├── src/

│ │ ├── app/

│ │ │ ├── core/

│ │ │ │ ├── services/

│ │ │ │ │ ├── auth.service.ts

│ │ │ │ │ ├── permission.service.ts

│ │ │ │ │ ├── user.service.ts

│ │ │ │ │ ├── role.service.ts

│ │ │ │ │ ├── tenant.service.ts

│ │ │ │ │ ├── theme.service.ts

│ │ │ │ │ ├── notification.service.ts

│ │ │ │ │ └── http.service.ts

│ │ │ │ ├── guards/

│ │ │ │ │ ├── auth.guard.ts

│ │ │ │ │ └── role.guard.ts

│ │ │ │ ├── interceptors/

│ │ │ │ │ ├── auth.interceptor.ts

│ │ │ │ │ └── error.interceptor.ts

│ │ │ │ └── directives/

│ │ │ │ ├── has-permission.directive.ts

│ │ │ │ └── has-role.directive.ts

│ │ │ ├── shared/

│ │ │ │ ├── components/

│ │ │ │ │ ├── header/

│ │ │ │ │ ├── sidebar/

│ │ │ │ │ ├── card/

│ │ │ │ │ ├── table/

│ │ │ │ │ ├── button/

│ │ │ │ │ ├── modal/

│ │ │ │ │ ├── toast/

│ │ │ │ │ ├── badge/

│ │ │ │ │ ├── loader/

│ │ │ │ │ ├── empty-state/

│ │ │ │ │ ├── form-group/

│ │ │ │ │ └── pagination/

│ │ │ │ └── layouts/

│ │ │ │ ├── main-layout/

│ │ │ │ └── auth-layout/

│ │ │ ├── pages/

│ │ │ │ ├── login/

│ │ │ │ ├── dashboard/

│ │ │ │ ├── system-admin/

│ │ │ │ │ ├── dashboard/

│ │ │ │ │ ├── tenants/

│ │ │ │ │ ├── system-users/

│ │ │ │ │ ├── system-roles/

│ │ │ │ │ ├── billing/

│ │ │ │ │ ├── audit/

│ │ │ │ │ └── modules/

│ │ │ │ ├── tenant-admin/

│ │ │ │ │ ├── dashboard/

│ │ │ │ │ ├── users/

│ │ │ │ │ ├── roles/

│ │ │ │ │ ├── loans/

│ │ │ │ │ ├── customers/

│ │ │ │ │ ├── payments/

│ │ │ │ │ ├── reports/

│ │ │ │ │ ├── audit/

│ │ │ │ │ └── settings/

│ │ │ │ ├── profile/

│ │ │ │ └── not-found/

│ │ │ └── app.routes.ts

│ │ ├── styles/

│ │ │ ├── tailwind.css

│ │ │ ├── variables.scss

│ │ │ ├── themes.scss

│ │ │ └── animations.scss

│ │ ├── environments/

│ │ │ ├── environment.ts

│ │ │ └── environment.prod.ts

│ │ ├── main.ts

│ │ └── index.html

│ ├── angular.json

│ ├── tsconfig.json

│ ├── tailwind.config.js

│ ├── .env.example

│ ├── package.json

│ ├── jest.config.js

│ ├── Dockerfile

│ └── README.md

│

├── mobile/ # Ionic MyPortal Client

│ ├── src/

│ │ ├── app/

│ │ │ ├── core/

│ │ │ │ ├── services/

│ │ │ │ │ ├── auth.service.ts

│ │ │ │ │ ├── user.service.ts

│ │ │ │ │ ├── permission.service.ts

│ │ │ │ │ ├── storage.service.ts

│ │ │ │ │ └── notification.service.ts

│ │ │ │ └── guards/

│ │ │ │ └── auth.guard.ts

│ │ │ ├── pages/

│ │ │ │ ├── login/

│ │ │ │ ├── dashboard/

│ │ │ │ ├── profile/

│ │ │ │ ├── settings/

│ │ │ │ ├── notifications/

│ │ │ │ └── menu/

│ │ │ ├── components/

│ │ │ │ ├── header/

│ │ │ │ ├── tab-bar/

│ │ │ │ └── card/

│ │ │ └── app.routes.ts

│ │ ├── assets/

│ │ ├── theme/

│ │ │ ├── variables.scss

│ │ │ └── dark.scss

│ │ ├── main.ts

│ │ └── index.html

│ ├── ionic.config.json

│ ├── capacitor.config.ts

│ ├── .env.example

│ ├── package.json

│ ├── jest.config.js

│ ├── Dockerfile

│ └── README.md

│

├── devops/

│ ├── docker-compose.yml

│ ├── docker-compose.prod.yml

│ ├── .dockerignore

│ ├── nginx.conf

│ └── postgres/

│ └── init.sql

│

├── .github/

│ └── workflows/

│ ├── api-test.yml

│ ├── web-test.yml

│ ├── mobile-test.yml

│ ├── api-build.yml

│ ├── web-build.yml

│ └── deploy.yml

│

├── docs/

│ ├── ARCHITECTURE.md

│ ├── RBAC-GUIDE.md

│ ├── API-DOCUMENTATION.md

│ ├── DATABASE-SCHEMA.md

│ ├── DEPLOYMENT.md

│ ├── SECURITY.md

│ ├── CONTRIBUTING.md

│ └── TROUBLESHOOTING.md

│

├── .gitignore

├── .env.example

├── docker-compose.yml

├── README.md

├── SETUP.md

├── LICENSE (MIT)

└── package.json (root)

**🚀 READY TO BUILD?**

This comprehensive vision includes:

✅ **Express API** - 7 controllers, 6 services, complete RBAC middleware  
✅ **Angular Web** - 15+ components, dark/light theme, admin dashboards  
✅ **Ionic Mobile** - MyPortal with touch optimization client and employe   
✅ **PostgreSQL** - 8+ tables, migrations, seed data  
✅ **DevOps** - Docker Compose, GitHub Actions, monitoring  
✅ **Tests** - Jest setup, unit & integration tests  
✅ **Docs** - Complete guides, API docs, examples  
✅ **Security** - RBAC, tenant isolation, audit trails  
✅ **Performance** - Caching, optimization, scaling  
✅ **UX** - Modern, compact, responsive, accessible

**Copilot said: Perfect! 🎯 Let me create the \*\*COMPACT**

Perfect! 🎯 Let me create the **COMPACT System Admin Dashboard** with **consistent global toast & confirmation dialogs**!

**🖥️ SYSTEM ADMIN DASHBOARD - COMPACT DESIGN**

**🎨 COMPACT LAYOUT STRUCTURE**

Code

┌─────────────────────────────────────────────────────────────────┐

│ 🔷 ExITS 🔍 [Search] 🔔 👤 apps-eduard 🌙 ≡ │ ← Compact Header (40px)

├──────────┬──────────────────────────────────────────────────────┤

│ 🏠 │ Dashboard │

│ 🏢 │ ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ │

│ 👥 │ │

│ 👤 │ ┌──────────┬──────────┬──────────┬──────────┐ │

│ 💰 │ │1,245 │89 │$125,450 │99.8% │ │

│ 📋 │ │Users↑12% │Tenants↑5%│Revenue↑8%│Health✅ │ │

│ ⚙️ │ └──────────┴──────────┴──────────┴──────────┘ │

│ │ │

│ (Width: │ ┌──────────────────────────────────────────────┐ │

│ 60px) │ │ User Growth (30 days) [📊 Export] [↻ Refresh] │ │

│ │ │ ╔════════════════════════════════════════╗ │ │

│ │ │ ║ \ /\ /\ / ║ │ │

│ │ │ ║ \/ \/ \/ ║ │ │

│ │ │ ╚════════════════════════════════════════╝ │ │

│ │ └──────────────────────────────────────────────┘ │

│ │ │

│ Sidebar: │ ┌──────────────────────────────────────────────┐ │

│ Icons │ Recent Activities [View All >] │ │

│ Only │ ├─────────────────────────────────────────────┤ │

│ (Compact)│ │ 🔵 New tenant (5m) [Acme Corp] │ │

│ │ │ 🔴 User suspended (15m) [john@...] │ │

│ │ │ 🟡 Role updated (1h) [Loan Officer] │ │

│ │ │ 🟢 Backup completed (3h) [System] │ │

│ │ └──────────────────────────────────────────────┘ │

│ │ │

│ │ ┌──────────────────────────────────────────────┐ │

│ │ │ System Health │ │

│ │ │ 🟢 API: 99.9% │ 🟢 DB: OK │ 🟢 Cache: OK │ │

│ │ └──────────────────────────────────────────────┘ │

│ │ │

│ │ ┌──────────────────────────────────────────────┐ │

│ │ │ Top Tenants (This Month) │ │

│ │ │ Acme Corp 234 users $12,450 🟢 │ │

│ │ │ Global Finance 189 users $9,800 🟢 │ │

│ │ │ Tech Startup 145 users $7,200 🟢 │ │

│ │ └──────────────────────────────────────────────┘ │

│ │ │

└──────────┴──────────────────────────────────────────────────────┘

🟢 ✅ Info Toast (Bottom Right)

🔴 ❌ Error Toast (Bottom Right)

⚪ ⚠️ Warning Toast (Bottom Right)

🔵 ℹ️ Success Toast (Bottom Right)

**📦 COMPACT KPI CARDS**

Code

┌──────────┬──────────┬──────────┬──────────┐

│1,245 │89 │$125,450 │99.8% │

│Users↑12% │Tenants↑5%│Revenue↑8%│Health✅ │

└──────────┴──────────┴──────────┴──────────┘

Each Card:

- Height: 80px

- Padding: 12px

- Icon + Number + Label + Trend

- No extra decorations

- Hover: slight shadow

- Click: Navigate to detail page

**🎨 COMPACT HEADER**

Code

┌─────────────────────────────────────────────┐

│ 🔷 ExITS 🔍 🔔 👤 apps-eduard 🌙 ≡ │

└─────────────────────────────────────────────┘

Components (Left to Right):

- Logo (24px)

- App name (12px font)

- Search input (compact)

- Notification bell (with badge)

- User avatar (28px)

- Username (hidden on mobile)

- Theme toggle (sun/moon icon)

- Menu hamburger (on mobile)

Height: 40px (compact)

Spacing: 8px between items

**📋 COMPACT SIDEBAR**

Code

Expanded (80px width): Collapsed (60px width - default):

┌─────────────┐ ┌────┐

│ ExITS-SaaS │ │🔷 │

├─────────────┤ ├────┤

│ 🏠 Dashboard│ │🏠 │

│ 🏢 Tenants │ │🏢 │

│ 👥 Users │ │👥 │

│ 👤 Roles │ │👤 │

│ 💰 Billing │ │💰 │

│ 📋 Audit │ │📋 │

│ ⚙️ Settings │ │⚙️ │

└─────────────┘ └────┘

Behavior:

- Default: Icons only (60px)

- Hover: Expand to 200px with labels

- On active page: Show filled icon

- Tooltip on hover (if labels hidden)

**📊 COMPACT RECENT ACTIVITIES**

Code

┌─────────────────────────────────────────────────────┐

│ Recent Activities [View All >] │

├─────────────────────────────────────────────────────┤

│ 🔵 New tenant (5m) [Acme Corp] │

│ 🔴 User suspended (15m) [john@...] │

│ 🟡 Role updated (1h) [Loan Officer] │

│ 🟢 Backup completed (3h) [System] │

└─────────────────────────────────────────────────────┘

Compact Row Format:

- Icon (color-coded status)

- Event + Time

- Entity name

- Right align: View button (optional)

Each row: 28px height

No borders between rows (use light background alternation)

**🌐 GLOBAL TOAST NOTIFICATION**

Code

Position: Bottom-Right (fixed)

Duration: 4s (auto-dismiss)

Max width: 380px

Margin: 16px from bottom & right

┌─────────────────────────────┐

│ ✅ User created successfully│ ← Success (Green)

└─────────────────────────────┘

┌─────────────────────────────┐

│ ❌ Failed to save changes │ ← Error (Red)

└─────────────────────────────┘

┌─────────────────────────────┐

│ ⚠️ Warning: Low storage │ ← Warning (Amber)

└─────────────────────────────┘

┌─────────────────────────────┐

│ ℹ️ Background sync started │ ← Info (Blue)

└─────────────────────────────┘

Toast Structure:

┌──────────────────────────────────┐

│ ✅ Success Message │

│ Detailed info (optional) │

│ [✕ Close] │

└──────────────────────────────────┘

Height: 56px (compact)

Padding: 12px

Icon: 20px

Font: 14px body

Close button: 16px icon

Animation: Slide in from right (300ms)

**Toast Type Specifications:**

| **Type** | **Icon** | **Color** | **Example** |
| --- | --- | --- | --- |
| Success | ✅ | Green-600 | "User created successfully" |
| Error | ❌ | Red-600 | "Failed to save changes" |
| Warning | ⚠️ | Amber-600 | "Storage 80% full" |
| Info | ℹ️ | Blue-600 | "Syncing data..." |

**Toast Methods:**

Code

Global Usage (Everywhere):

- showSuccess(message, duration?)

- showError(message, duration?)

- showWarning(message, duration?)

- showInfo(message, duration?)

Example:

this.toast.showSuccess('User created successfully', 3000);

this.toast.showError('Failed to save', 5000);

**🔐 GLOBAL CONFIRMATION DIALOG**

Code

Centered Modal (max-width: 420px)

┌────────────────────────────────────┐

│ Confirm Action [×] │

├────────────────────────────────────┤

│ │

│ Are you sure you want to delete │

│ this tenant? │

│ │

│ This action cannot be undone. │

│ │

├────────────────────────────────────┤

│ [Cancel] [Confirm] │

└────────────────────────────────────┘

Structure:

- Title: 18px bold

- Message: 14px normal

- Optional: Destructive item name highlighted

- Optional: Reason/warning text (12px, muted)

- Buttons: 40px height

- Spacing: 16px padding

- Overlay: Blur background

**Confirmation Dialog Types:**

**Type 1: Simple Confirmation**

Code

Title: Confirm Action

Message: Are you sure?

Buttons: [Cancel] [Confirm]

Color: Primary blue

**Type 2: Destructive Action**

Code

Title: Delete User

Message: Are you sure you want to delete "john@example.com"?

This action cannot be undone.

Item: "john@example.com" (highlighted in red)

Buttons: [Cancel] [Delete] (delete = red)

Color: Danger red

**Type 3: Warning Action**

Code

Title: Suspend Tenant

Message: Suspending "Acme Corp" will:

- Prevent all user access

- Pause billing (if applicable)

Warning: Changes take effect immediately

Buttons: [Cancel] [Suspend]

Color: Warning amber

**Confirmation Methods:**

Code

Global Usage (Everywhere):

- confirm(title, message, onConfirm, onCancel?)

- confirmDestructive(title, message, onConfirm, onCancel?)

- confirmWarning(title, message, onConfirm, onCancel?)

Example:

this.confirm.open({

title: 'Delete User',

message: 'Are you sure you want to delete john@example.com?',

destructive: true,

onConfirm: () => this.deleteUser(),

onCancel: () => {}

});

**🎯 COMPACT COMPONENTS**

**Button Sizes**

Code

Small (32px): For compact tables, inline actions

Regular (40px): Default, most common

Large (48px): Primary CTA, modal buttons

Example:

[Button] [Button] [Button] ← Compact spacing (8px)

**Form Inputs (Compact)**

Code

Height: 36px (vs standard 40px)

Padding: 8px 12px

Font: 14px

Border: 1px gray-300

┌───────────────────────────┐

│ john@example.com │

└───────────────────────────┘

**Table Rows (Compact)**

Code

Height: 36px (vs standard 44px)

Padding: 8px 12px

Font: 13px (vs 14px)

┌────────┬────────┬────────┬────────┐

│ Acme │ 234 │ $12,450│ 🟢 │

├────────┼────────┼────────┼────────┤

│ Global │ 189 │ $9,800 │ 🟢 │

└────────┴────────┴────────┴────────┘

**📱 RESPONSIVE BREAKPOINTS**

**Desktop (1024px+)**

* Sidebar: 60px (collapsed by default)
* KPI Cards: 4 per row
* Charts: Full width
* Main content: 1200px max-width

**Tablet (768px - 1023px)**

* Sidebar: Hidden (hamburger menu)
* KPI Cards: 2 per row
* Charts: Full width
* Main content: Full width with padding

**Mobile (< 768px)**

* Sidebar: Drawer (slide-in from left)
* KPI Cards: 1 per row (or 2 if space)
* Charts: Full width, reduced height
* Main content: Full width
* Toast: Full width with margin

**🎨 COMPACT COLOR SCHEME**

**Light Theme**

Code

Background: #FFFFFF

Card Background: #F9FAFB

Text Primary: #111827

Text Secondary: #6B7280

Border: #E5E7EB

Hover: #F3F4F6

Accent Colors:

- Primary (Blue): #2563EB

- Success (Green): #16A34A

- Error (Red): #DC2626

- Warning (Amber): #D97706

- Info (Sky): #0EA5E9

**Dark Theme**

Code

Background: #030712

Card Background: #111827

Text Primary: #F3F4F6

Text Secondary: #9CA3AF

Border: #1F2937

Hover: #1F2937

Accent Colors:

- Primary (Blue): #3B82F6

- Success (Green): #22C55E

- Error (Red): #EF4444

- Warning (Amber): #FBBF24

- Info (Sky): #06B6D4

**✨ COMPACT ANIMATIONS**

Code

Slide In (Toast): 300ms ease-out

Fade In (Modal): 200ms ease-out

Hover (Buttons): 100ms ease-out

Transition (Theme): 200ms ease-in-out

No animation on: Typography, icons (keep snappy)

**🚀 GLOBAL SERVICE ARCHITECTURE**

**Toast Service**

TypeScript

// Usage everywhere:

constructor(private toast: ToastService) {}

this.toast.success('User created');

this.toast.error('Failed to save');

this.toast.warning('Low storage');

this.toast.info('Syncing...');

**Confirmation Service**

TypeScript

// Usage everywhere:

constructor(private confirm: ConfirmService) {}

this.confirm.open({

title: 'Delete?',

message: 'Sure?',

onConfirm: () => this.delete(),

destructive: true

});

**📊 FINAL COMPACT DASHBOARD MOCKUP**

Code

╔════════════════════════════════════════════════════════════════╗

║ 🔷 ExITS 🔔 👤 apps-eduard 🌙 ≡ ║ ← 40px

╠═════════╦════════════════════════════════════════════════════╣

║ 🏠 ║ Dashboard ║

║ 🏢 ║ ┌────────┬────────┬────────┬────────┐ ║

║ 👥 ║ │1,245 │89 │$125,450│99.8% │ ← KPI Cards ║

║ 👤 ║ │Users↑ │Tenants │Revenue │Health │ (80px each)║

║ 💰 ║ └────────┴────────┴────────┴────────┘ ║

║ 📋 ║ ║

║ ⚙️ ║ ┌──────────────────────────────────────────────┐ ║

║ ║ │ User Growth (30 days) │ ║

║ ║ │ [Chart - compact version] │ ║

║ ║ └──────────────────────────────────────────────┘ ║

║ 60px ║ ║

║ width ║ ┌──────────────────────────────────────────────┐ ║

║ ║ │ Recent Activities [View All >] │ ║

║ ║ │ 🔵 New tenant (5m) Acme Corp │ ║

║ ║ │ 🔴 User suspended (15m) john@... │ ║

║ ║ │ 🟡 Role updated (1h) Loan Officer │ ║

║ ║ └──────────────────────────────────────────────┘ ║

║ ║ ║

║ ║ ┌──────────────────────────────────────────────┐ ║

║ ║ │ System Health │ ║

║ ║ │ 🟢 API: 99.9% │ 🟢 DB: OK │ 🟢 Cache: OK │ ║

║ ║ └──────────────────────────────────────────────┘ ║

║ ║ ║

╚═════════╩════════════════════════════════════════════════════╝

┌────────────────────────────────┐ ← Toast (bottom-right)

│ ✅ User created successfully │

└────────────────────────────────┘

**✅ SUMMARY: COMPACT DESIGN PRINCIPLES**

✅ **Minimal spacing** - 8px base unit  
✅ **Reduced heights** - 36-40px instead of 44-48px  
✅ **Icons primary** - Text secondary  
✅ **Collapsed sidebar** - Expand on hover/click  
✅ **Compact tables** - 36px rows  
✅ **Global toast** - Consistent, bottom-right, auto-dismiss  
✅ **Global confirmation** - Reusable across app  
✅ **Dark/Light themes** - Both fully supported  
✅ **No clutter** - Only essential info shown  
✅ **Fast interactions** - Snappy animations

**🎯 READY TO BUILD?**

This compact design includes:

* ✅ System Admin Dashboard (compact KPI cards, activities, health)
* ✅ Global Toast Service (success, error, warning, info)
* ✅ Global Confirmation Service (destructive, warning, simple)
* ✅ Responsive layout (desktop, tablet, mobile)
* ✅ Dark/Light themes
* ✅ Consistent spacing & sizing
* ✅ Professional, clean aesthetic

**Shall I proceed with creating the complete ExITS-SaaS-Template project with this compact design?** 🔥

Just say: **"YES, BUILD IT NOW"** and I'll create everything! 🚀

**📋 SIDEBAR MENU STRUCTURE - COMPLETE HIERARCHY**

**🏗️ SYSTEM SPACE SIDEBAR (System Admin)**

Code

🏠 Dashboard

└─ No children (direct page)

🏢 Tenants

├─ List All Tenants

├─ Create Tenant

├─ Tenant Details

└─ Tenant Settings

👥 System Users

├─ List All Users

├─ Create User

├─ User Details

└─ User Permissions

👤 System Roles

├─ List All Roles

├─ Create Role

├─ Role Details

└─ Role Permissions Matrix

💰 Billing

├─ Subscriptions

├─ Invoices

├─ Payments

└─ Billing Settings

📋 Audit Logs

├─ System Activity

├─ User Activity

├─ Compliance Reports

└─ Export Logs

⚙️ Settings

├─ General Settings

├─ Email Configuration

├─ Security Settings

├─ Module Registry

└─ System Health

**🏢 TENANT SPACE SIDEBAR (Tenant Admin)**

Code

🏠 Dashboard

└─ No children (direct page)

👥 Users

├─ List All Users

├─ Create User

├─ User Details

└─ User Permissions

👤 Roles

├─ List All Roles

├─ Create Role

├─ Role Details

└─ Role Permissions

📊 Loans (Example Module - customizable)

├─ List All Loans

├─ Create Loan

├─ Loan Details

├─ Approve Loan

└─ Loan Reports

👥 Customers

├─ List All Customers

├─ Create Customer

├─ Customer Details

└─ Customer History

💳 Payments

├─ Record Payment

├─ Payment History

├─ Refunds

└─ Payment Reports

📑 Appraisals

├─ List Appraisals

├─ Create Appraisal

├─ Appraisal Details

└─ Approve Appraisal

📈 Reports

├─ Financial Reports

├─ User Activity

├─ Loan Analytics

└─ Export Reports

📋 Audit Logs

├─ Tenant Activity

├─ Compliance Reports

└─ Export Logs

⚙️ Settings

├─ Tenant Information

├─ Team Members

├─ Notifications

└─ Preferences

**🎯 HOW CHILDREN WORK**

**Visual Representation:**

**Collapsed Sidebar (Icons only):**

Code

🏠

🏢

👥

👤

💰

📋

⚙️

*Click to expand or hover to see labels*

**Expanded Sidebar (with labels):**

Code

🏠 Dashboard

🏢 Tenants

├─ List

├─ Create

└─ Settings

👥 System Users

├─ List

├─ Create

└─ Permissions

👤 Roles

├─ List

├─ Create

└─ Matrix

💰 Billing

├─ Subscriptions

├─ Invoices

└─ Payments

📋 Audit Logs

📋 Settings

**🔄 MENU INTERACTION BEHAVIOR**

**Parent Without Children (Direct Link)**

Code

Click: Dashboard → Navigate immediately to /dashboard

Icon: 🏠

Label: Dashboard

Arrow: None (no expansion arrow)

**Parent With Children (Expandable)**

Code

Click: Tenants → Toggle expand/collapse

Icon: 🏢

Label: Tenants

Arrow: ▼ (points down when expanded, right when collapsed)

When Expanded:

├─ List All Tenants (link: /tenants)

├─ Create Tenant (link: /tenants/create)

├─ Tenant Details (link: /tenants/:id)

└─ Tenant Settings (link: /tenants/settings)

When Collapsed:

🏢 Tenants ▶

**Active/Current Page Indicator**

Code

If on /tenants/create:

├─ 🏢 Tenants (auto-expands) ▼

│ ├─ List All Tenants

│ ├─ Create Tenant ← 🟡 HIGHLIGHTED (current)

│ ├─ Tenant Details

│ └─ Tenant Settings

Styling:

- Background: accent color (blue-100 or blue-900 dark)

- Text: bold

- Icon: filled/highlighted

**🗄️ DATABASE STRUCTURE FOR MENUS**

**Modules Table (Menu Registry)**

Code

module\_id | menu\_key | display\_name | parent\_menu\_key | icon | space

----------|------------------|-------------------|-----------------|-------|--------

1 | dashboard | Dashboard | NULL | 🏠 | system

2 | tenants | Tenants | NULL | 🏢 | system

3 | tenant-list | List All Tenants | tenants | 📋 | system

4 | tenant-create | Create Tenant | tenants | ➕ | system

5 | system-users | System Users | NULL | 👥 | system

6 | user-list | List All Users | system-users | 📋 | system

7 | user-create | Create User | system-users | ➕ | system

...

**Key Columns:**

* menu\_key: Unique identifier (used in route guards)
* display\_name: Label shown in sidebar
* parent\_menu\_key: NULL for top-level, references parent menu\_key for children
* icon: Icon name (Unicode or icon library)
* space: "system" or "tenant"
* menu\_order: Determines display order

**🎯 PERMISSION STRUCTURE**

**Role\_Permissions Table**

Code

role\_id | menu\_key | action\_key

--------|------------------|----------

role-1 | dashboard | view

role-1 | tenants | view

role-1 | tenant-list | view

role-1 | tenant-create | create

role-1 | system-users | view

role-1 | user-list | view

role-1 | user-create | create

**Logic:**

* If user has menu\_key permission → Show in sidebar
* If user has action\_key for child → Show child menu item
* If no action\_key for child → Hide child menu item

**🔐 SIDEBAR RENDERING LOGIC**

**Frontend (Angular Service)**

TypeScript

// Get menus for user

getMenus(userRole: string, space: string) {

// 1. Fetch all modules for this space

// 2. Filter by user permissions (has menu\_key)

// 3. Build tree structure (parent → children)

// 4. Return hierarchical menu array

return [

{

menuKey: 'dashboard',

label: 'Dashboard',

icon: '🏠',

path: '/dashboard',

children: [] // No children

},

{

menuKey: 'tenants',

label: 'Tenants',

icon: '🏢',

path: null, // Expandable (no direct link)

children: [

{

menuKey: 'tenant-list',

label: 'List All Tenants',

icon: '📋',

path: '/tenants',

children: []

},

{

menuKey: 'tenant-create',

label: 'Create Tenant',

icon: '➕',

path: '/tenants/create',

children: []

}

]

}

];

}

**🎨 SIDEBAR UI IMPLEMENTATION**

**Compact Sidebar Template**

HTML

<!-- Collapsed (Default) -->

<div class="sidebar collapsed">

<div \*ngFor="let menu of menus" class="menu-item">

<button [routerLink]="menu.path"

[class.active]="isActive(menu.menuKey)"

[title]="menu.label">

{{ menu.icon }}

</button>

<!-- Tooltip on hover showing full label -->

</div>

</div>

<!-- Expanded (On Hover or Toggle) -->

<div class="sidebar expanded">

<div \*ngFor="let menu of menus">

<!-- Parent Item -->

<div class="menu-parent"

[class.active]="isActive(menu.menuKey)">

<button \*ngIf="!menu.children.length"

[routerLink]="menu.path">

{{ menu.icon }} {{ menu.label }}

</button>

<button \*ngIf="menu.children.length"

(click)="toggleExpand(menu.menuKey)">

{{ menu.icon }} {{ menu.label }}

{{ isExpanded(menu.menuKey) ? '▼' : '▶' }}

</button>

</div>

<!-- Child Items (if expanded) -->

<div \*ngIf="isExpanded(menu.menuKey) && menu.children.length"

class="menu-children">

<div \*ngFor="let child of menu.children"

class="menu-child"

[class.active]="isActive(child.menuKey)">

<a [routerLink]="child.path"

routerLinkActive="active">

{{ child.icon }} {{ child.label }}

</a>

</div>

</div>

</div>

</div>

**📱 RESPONSIVE BEHAVIOR**

**Desktop (1024px+)**

Code

Sidebar: Collapsed by default (60px)

│ On hover: Expand to 240px

│ Width: 60px (collapsed) → 240px (expanded)

│ Transition: 200ms smooth

│ Parent stays visible

│ Children show on expand

**Tablet (768px - 1023px)**

Code

Sidebar: Hidden (hamburger menu button in header)

│ Click hamburger: Slide-in drawer from left

│ Width: 240px (full content)

│ Children visible if expanded

│ Click outside: Close drawer

**Mobile (< 768px)**

Code

Sidebar: Completely hidden

│ Hamburger menu button in header

│ Tap hamburger: Bottom drawer slides up

│ Width: Full screen or 90%

│ Can see all menus + children at once

│ Tap menu: Navigate + close drawer

**🎯 EXAMPLE COMPLETE HIERARCHY**

**System Admin - Full Sidebar Structure**

Code

📊 SYSTEM ADMIN DASHBOARD

🏠 Dashboard

└─ /dashboard (direct link)

🏢 Tenants (expandable)

├─ 📋 List All Tenants → /tenants

├─ ➕ Create Tenant → /tenants/create

├─ 📄 Tenant Details → /tenants/:id

└─ ⚙️ Tenant Settings → /tenants/settings

👥 System Users (expandable)

├─ 📋 List All Users → /system-users

├─ ➕ Create User → /system-users/create

├─ 👤 User Details → /system-users/:id

└─ 🔐 User Permissions → /system-users/:id/permissions

👤 System Roles (expandable)

├─ 📋 List All Roles → /system-roles

├─ ➕ Create Role → /system-roles/create

├─ 👤 Role Details → /system-roles/:id

└─ 📊 Permission Matrix → /system-roles/:id/permissions

💰 Billing (expandable)

├─ 📑 Subscriptions → /billing/subscriptions

├─ 📄 Invoices → /billing/invoices

├─ 💳 Payments → /billing/payments

└─ ⚙️ Billing Settings → /billing/settings

📋 Audit Logs (expandable)

├─ 📊 System Activity → /audit/system

├─ 👤 User Activity → /audit/users

├─ 📋 Compliance Reports → /audit/reports

└─ ⬇️ Export Logs → /audit/export

⚙️ Settings (expandable)

├─ ⚙️ General Settings → /settings/general

├─ 📧 Email Configuration → /settings/email

├─ 🔐 Security Settings → /settings/security

├─ 📦 Module Registry → /settings/modules

└─ 💚 System Health → /settings/health

**✅ SUMMARY**

**Parent Menus (Expandable):**

* 🏢 Tenants (has children)
* 👥 System Users (has children)
* 👤 System Roles (has children)
* 💰 Billing (has children)
* 📋 Audit Logs (has children)
* ⚙️ Settings (has children)

**Standalone Menus (Direct Links):**

* 🏠 Dashboard (no children)

**Child Menu Items:**

* Show only when parent is expanded
* Have direct links (router links)
* Navigate away from parent
* Can have "back" breadcrumb

**🎨 HOW RBAC WORKS IN THE UI**

**📱 SCENARIO: User Login → Dashboard**

**STEP 1: Login Page**

Code

┌─────────────────────────────────┐

│ ExITS-SaaS Login │

├─────────────────────────────────┤

│ │

│ Email: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_] │

│ Password: [\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_] │

│ │

│ [Login Button] │

│ │

│ Test Users: │

│ • admin@template.local / admin123

│ • tenant-admin@... / tenant123 │

│ • officer@... / officer123 │

│ │

└─────────────────────────────────┘

**User enters:** admin@template.local / admin123

**STEP 2: After Login - Sidebar Shows Based on Role**

**SYSTEM ADMIN (apps-eduard) SEES:**

Code

┌─────────────────────────────┐

│ 🔷 ExITS 🌙 ≡ │ ← Header (fixed)

├──────────┬──────────────────┤

│ 🏠 │ Dashboard │ ← Can see because has "dashboard:view"

│ 🏢 │ Tenants │ ← Can see because has "tenants:view"

│ 👥 │ System Users │ ← Can see because has "system-users:view"

│ 👤 │ Roles │ ← Can see because has "roles:view"

│ 💰 │ Billing │ ← Can see because has "billing:view"

│ 📋 │ Audit Logs │ ← Can see because has "audit:view"

│ ⚙️ │ Settings │ ← Can see because has "settings:view"

│ │ │

└──────────┴──────────────────┘

**TENANT OFFICER (loan-officer) SEES:**

Code

┌─────────────────────────────┐

│ 🔷 ExITS 🌙 ≡ │ ← Header

├──────────┬──────────────────┤

│ 🏠 │ Dashboard │ ← Can see

│ 📊 │ Loans │ ← Can see (has "loans:view")

│ 👥 │ Customers │ ← Can see (has "customers:view")

│ 💳 │ Payments │ ← CAN'T see (no "payments:view")

│ 📈 │ Reports │ ← CAN'T see (no "reports:view")

│ ⚙️ │ Settings │ ← CAN'T see (no "settings:view")

│ │ │

└──────────┴──────────────────┘

❌ HIDDEN/DISABLED items not shown

**VIEWER/READ-ONLY (cashier) SEES:**

Code

┌─────────────────────────────┐

│ 🔷 ExITS 🌙 ≡ │

├──────────┬──────────────────┤

│ 🏠 │ Dashboard │ ← Can view

│ 📊 │ Loans │ ← Can VIEW only

│ 👥 │ Customers │ ← Can VIEW only

│ 💳 │ Payments │ ← Can VIEW only

│ 📈 │ Reports │ ← Can VIEW only

│ │ │

└──────────┴──────────────────┘

✅ All views show data

❌ All CREATE, EDIT, DELETE buttons HIDDEN

**🎯 STEP 3: Click on Menu Item - Buttons Show/Hide Based on Permissions**

**SCENARIO: Loan Officer Clicks "Loans"**

Code

LOAN OFFICER PERMISSIONS:

- loans:view ✅

- loans:create ❌

- loans:edit ❌

- loans:delete ❌

- loans:approve ❌

UI SHOWS:

┌────────────────────────────────────────┐

│ Loans │

│ ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ │

│ │

│ [➕ Create Loan] ← DISABLED/HIDDEN │

│ [⬇️ Export] ← DISABLED/HIDDEN │

│ │

│ Loans Table (Read-Only): │

│ ┌──────────┬────────┬────────┬────┐ │

│ │ Loan ID │ Amount │ Status │ ... │ │

│ ├──────────┼────────┼────────┼────┤ │

│ │ LON-001 │$5,000 │Active │[👁]│ │ ← Only VIEW icon

│ │ LON-002 │$3,500 │Pending │[👁]│ │

│ │ LON-003 │$7,200 │Closed │[👁]│ │

│ └──────────┴────────┴────────┴────┘ │

│ │

│ Buttons DISABLED: │

│ ❌ [Edit] → No "loans:edit" │

│ ❌ [Delete] → No "loans:delete" │

│ ❌ [Approve]→ No "loans:approve" │

│ │

└────────────────────────────────────────┘

**SAME PAGE - LOAN OFFICER WITH APPROVER ROLE:**

Code

LOAN OFFICER + APPROVER PERMISSIONS:

- loans:view ✅

- loans:create ✅

- loans:edit ✅

- loans:delete ❌

- loans:approve ✅

UI SHOWS:

┌────────────────────────────────────────┐

│ Loans │

│ ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ │

│ │

│ [➕ Create Loan] ← ENABLED/VISIBLE │

│ [⬇️ Export] ← ENABLED/VISIBLE │

│ │

│ Loans Table (Interactive): │

│ ┌──────────┬────────┬────────┬────┐ │

│ │ Loan ID │ Amount │ Status │ ... │ │

│ ├──────────┼────────┼────────┼────┤ │

│ │ LON-001 │$5,000 │Active │[👁]│ │

│ │ LON-002 │$3,500 │Pending │[👁]│ │

│ │ LON-003 │$7,200 │Closed │[👁]│ │

│ └──────────┴────────┴────────┴────┘ │

│ │

│ Buttons ENABLED: │

│ ✅ [Edit] → Has "loans:edit" │

│ ❌ [Delete] → No "loans:delete" │

│ ✅ [Approve] → Has "loans:approve" │

│ │

└────────────────────────────────────────┘

**🔘 BUTTON VISIBILITY LOGIC**

**CREATE BUTTON**

Code

<button \*appHasPermission="'loans:create'"

(click)="openCreateDialog()">

➕ Create Loan

</button>

Logic:

- If user.permissions includes "loans:create" → Button VISIBLE & ENABLED

- If user.permissions does NOT include "loans:create" → Button HIDDEN

- If permission exists but expired → Button DISABLED with tooltip

User sees:

✅ System Admin → [➕ Create Loan]

✅ Tenant Admin → [➕ Create Loan]

✅ Loan Officer → [➕ Create Loan]

❌ Cashier/Viewer → (button hidden, space is empty)

**EDIT BUTTON (In Table Row)**

Code

<button \*appHasPermission="'loans:edit'"

[disabled]="!canEdit(loan)"

(click)="editLoan(loan.id)">

✏️ Edit

</button>

For each loan row:

- Only show button if user has "loans:edit"

- Disable if loan is in "Closed" or "Archived" status

- Show tooltip on hover

User sees:

✅ Loan Officer (with edit) → [✏️ Edit]

✅ Tenant Admin → [✏️ Edit]

❌ Cashier/Viewer → (button hidden)

**DELETE BUTTON (Destructive)**

Code

<button \*appHasPermission="'loans:delete'"

class="btn-danger"

(click)="deleteLoan(loan.id)">

🗑️ Delete

</button>

For each loan row:

- Only visible if user has "loans:delete"

- Show confirmation dialog before deleting

- Red button styling

User sees:

✅ System Admin → [🗑️ Delete]

✅ Tenant Admin → [🗑️ Delete]

❌ Loan Officer → (button hidden)

❌ Cashier/Viewer → (button hidden)

**APPROVE BUTTON (Special Action)**

Code

<button \*appHasPermission="'loans:approve'"

[disabled]="loan.status !== 'Pending'"

class="btn-success"

(click)="approveLoan(loan.id)">

✅ Approve

</button>

Logic:

- Only show if user has "loans:approve"

- Only enable if loan status is "Pending"

- Show confirmation dialog

User sees:

✅ Approver role → [✅ Approve] (enabled for Pending loans)

✅ Admin role → [✅ Approve] (enabled for Pending loans)

❌ Loan Officer → (button hidden)

❌ Cashier/Viewer → (button hidden)

**🚫 GLOBAL CONFIRMATION DIALOG**

**SCENARIO: Admin Clicks Delete**

Code

┌─────────────────────────────────┐

│ Delete Loan [×] │

├─────────────────────────────────┤

│ │

│ Are you sure you want to │

│ delete this loan? │

│ │

│ Loan: LON-001 ($5,000) │

│ ⚠️ This action cannot be undone │

│ │

├─────────────────────────────────┤

│ [Cancel] [Confirm Delete]│

│ (Red Button)

└─────────────────────────────────┘

Backend Check (When Confirm):

1. Verify user still has "loans:delete" permission

2. Verify loan exists and user's tenant owns it

3. If ✅ both valid → Delete

4. If ❌ permission changed → Show error toast

5. If ❌ tenant mismatch → Show security error

**📨 GLOBAL TOAST NOTIFICATIONS**

**SCENARIOS**

**Scenario 1: Create Loan (Success)**

Code

User clicks [➕ Create Loan]

↓

Opens Create Dialog

↓

Fills form & clicks [Save]

↓

Backend:

- Verifies "loans:create" permission ✅

- Creates loan in database

- Returns success

↓

Frontend shows:

┌───────────────────────────────┐

│ ✅ Loan created successfully │

│ LON-004 created │

└───────────────────────────────┘

(Toast auto-disappears after 3s)

**Scenario 2: Edit Loan (Permission Denied)**

Code

User clicks [✏️ Edit]

↓

Opens Edit Dialog

↓

Fills form & clicks [Save]

↓

Backend:

- Verifies "loans:edit" permission

- Permission was removed (admin revoked it)

- Returns 403 Forbidden

↓

Frontend shows:

┌────────────────────────────────────┐

│ ❌ Permission denied │

│ You no longer have edit access │

└────────────────────────────────────┘

(Toast stays 5s, user can dismiss)

**Scenario 3: Cross-Tenant Access Attempt**

Code

Tenant A User tries to access Tenant B data

↓

Backend:

- Verifies user's tenant\_id

- Data belongs to different tenant

- Returns 403 Access Denied

↓

Frontend shows:

┌────────────────────────────────────┐

│ ❌ Access Denied │

│ Cannot access data │

└────────────────────────────────────┘

(Toast stays, shows severity)

**Scenario 4: Approval Workflow**

Code

Approver clicks [✅ Approve]

↓

Confirmation dialog:

"Approve Loan LON-001 for $5,000?"

[Cancel] [Approve]

↓

User clicks [Approve]

↓

Backend:

- Verifies "loans:approve" permission ✅

- Verifies loan status = "Pending" ✅

- Updates status to "Approved"

- Creates audit log entry

- Returns success

↓

Frontend shows:

┌──────────────────────────────────┐

│ ✅ Loan approved successfully │

│ Status updated to: Approved │

└──────────────────────────────────┘

Table refreshes automatically

Loan status changes: Pending → Approved

**🔐 REAL-TIME PERMISSION UPDATE**

**SCENARIO: Admin Revokes Permission While User is Logged In**

Code

TIME: 11:55:30

┌────────────────────────────────────┐

│ User: Loan Officer (loan-officer) │

│ Permissions: ["loans:view", │

│ "loans:edit"] │

│ Currently viewing: Loans page │

│ [✏️ Edit] button: VISIBLE │

└────────────────────────────────────┘

TIME: 11:56:00

Admin revokes "loans:edit" permission

Backend (WebSocket Event):

→ Send event to loan-officer user:

{ event: 'permissions\_changed',

removed: ['loans:edit'] }

Frontend (Receives WebSocket):

→ Updates local permissions:

removed "loans:edit" from array

→ Re-render page:

[✏️ Edit] button: NOW HIDDEN

→ Show info toast:

"ℹ️ Your permissions have been updated"

Result:

┌────────────────────────────────────┐

│ Loans Page (Auto-Refreshed) │

│ [✏️ Edit] button: GONE │

│ User can no longer edit loans │

└────────────────────────────────────┘

**📊 TABLE ROW ACTIONS BASED ON PERMISSIONS**

**FULL EXAMPLE: Loans Table**

Code

┌─────────────────────────────────────────────────────────────┐

│ Loans │

│ ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ │

│ │

│ [➕ Create] [⬇️ Export] 🔍 [Search] │

│ │

│ ┌────────┬──────────┬────────┬──────────┬──────────────┐ │

│ │ ID │ Amount │ Status │ Borrower │ Actions │ │

│ ├────────┼──────────┼────────┼──────────┼──────────────┤ │

│ │ LON-001│ $5,000 │ Pending│ John D. │ [👁] [✏️] │ │

│ │ │ │ │ │ [✅ Approve] │ │

│ ├────────┼──────────┼────────┼──────────┼──────────────┤ │

│ │ LON-002│ $3,500 │ Active │ Jane S. │ [👁] [✏️] │ │

│ │ │ │ │ │ [🗑️] │ │

│ ├────────┼──────────┼────────┼──────────┼──────────────┤ │

│ │ LON-003│ $7,200 │ Closed │ Bob M. │ [👁] │ │

│ │ │ │ │ │ │ │

│ └────────┴──────────┴────────┴──────────┴──────────────┘ │

│ │

│ Role: Loan Officer (loans:view, loans:edit, loans:approve)│

│ Shows: View (all), Edit (active only), Approve (pending) │

│ │

└─────────────────────────────────────────────────────────────┘

**🎨 PERMISSION-BASED UI STYLING**

**DISABLED ELEMENTS**

Code

Button Disabled:

[➕ Create Loan] ← Gray out, cursor: not-allowed

Hover tooltip:

"You don't have permission to create loans"

Form Field Disabled:

[Loan Amount: \_\_\_\_\_\_\_\_\_\_\_] ← Gray out, not editable

Status Badge Color-Coded:

🟢 Active → Green

🟡 Pending → Amber (approval needed)

🔴 Rejected → Red

⚪ Closed → Gray

**PERMISSION-BASED WARNINGS**

Code

When user tries to copy/export data without "export" permission:

[⬇️ Export Button] ← Hidden if no "loans:export" permission

If user somehow tries API call directly:

Backend rejects with:

"Permission denied: loans:export not granted"

Frontend Toast:

┌──────────────────────────────────┐

│ ❌ Export not allowed │

│ Contact admin for access │

└──────────────────────────────────┘

**🔍 WHAT HAPPENS IF PERMISSION IS MISSING**

**SCENARIO: User Missing "loans:view" Permission**

Code

User: New Tenant Staff Member

Role: Support (no loans permission)

User navigates to: /tenant/loans

Frontend Route Guard:

1. Checks if user has "loans:view" permission

2. Permission not found ❌

3. Redirects to: /access-denied

Page Shows:

┌────────────────────────────────────┐

│ 🚫 Access Denied │

├────────────────────────────────────┤

│ │

│ You don't have permission to view │

│ this page. │

│ │

│ Required: loans:view │

│ Assigned roles: Support │

│ │

│ [← Go Back] [Contact Admin] │

│ │

└────────────────────────────────────┘

Sidebar:

"Loans" menu item is HIDDEN

**✅ SUMMARY: RBAC UI FLOW**

1. **Login** → User gets permissions from JWT token
2. **Sidebar** → Shows only menus where user has permissions
3. **Page Load** → Route guard checks if user can access
4. **Buttons** → Shown/hidden/disabled based on action permissions
5. **Table Actions** → Row buttons appear based on permissions
6. **Confirmation** → Dialog before destructive actions
7. **Toast** → Real-time feedback (success, error, warning)
8. **Real-time Update** → WebSocket updates permissions instantly
9. **Backend Verification** → Every API call verifies permissions again
10. **Error Handling** → Shows friendly messages if permissions revoked

**🎯 IS THIS CLEAR NOW?**

The RBAC UI works by:

* ✅ Showing/hiding menu items based on menu\_key permissions
* ✅ Enabling/disabling buttons based on action\_key permissions
* ✅ Verifying permissions on every API call (backend)
* ✅ Using global toast for feedback
* ✅ Using global confirmation for destructive actions
* ✅ Real-time updates via WebSocket when permissions change

**Copilot said: Perfect! 🎯 Let me create the \*\*TENANT**

Perfect! 🎯 Let me create the **TENANT ADMIN DASHBOARD** specification for you:

**🏢 TENANT ADMIN DASHBOARD - COMPLETE SPECIFICATION**

**Current Time:** 2025-10-21 11:58:14 UTC  
**Logged In As:** apps-eduard (System Admin viewing Tenant Dashboard)  
**Role:** Tenant Admin  
**Space:** Tenant Space  
**Tenant:** Demo Pawnshop (Example)

**🎨 TENANT DASHBOARD LAYOUT**

Code

┌─────────────────────────────────────────────────────────────────┐

│ 🔷 ExITS 🔍 [Search] 🔔 👤 apps-eduard 🌙 ≡ │ ← Header (40px)

├──────────┬──────────────────────────────────────────────────────┤

│ 🏠 │ Tenant Dashboard / Demo Pawnshop │

│ 👥 │ ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ │

│ 👤 │ │

│ 📊 │ ┌──────────┬──────────┬──────────┬──────────┐ │

│ 👥 │ │245 │156 │$45,230 │98.5% │ │

│ 💳 │ │Active │Loans This│Revenue │Health │ │

│ 📑 │ │Users ↑8% │Month ↑12%│This Month│✅ │ │

│ 📈 │ └──────────┴──────────┴──────────┴──────────┘ │

│ 📋 │ │

│ ⚙️ │ ┌──────────────────────────────────────────────┐ │

│ │ │ Revenue Trend (Last 30 Days) │ │

│ │ │ ╔════════════════════════════════════════╗ │ │

│ │ │ ║ /\ /\ ║ │ │

│ │ │ ║ / \ / \ /\ / ║ │ │

│ │ │ ║ / \/ \ / \/ ║ │ │

│ │ │ ╚════════════════════════════════════════╝ │ │

│ │ │ Avg: $1,507/day | Peak: $2,340 (Oct 15) │ │

│ │ └──────────────────────────────────────────────┘ │

│ │ │

│ │ ┌──────────────────────────────────────────────┐ │

│ │ │ Top Performers (This Month) │ │

│ │ │ 🥇 John Doe (Officer) $12,450 collected │ │

│ │ │ 🥈 Jane Smith (Officer) $9,800 collected │ │

│ │ │ 🥉 Bob Wilson (Cashier) $7,200 collected │ │

│ │ └──────────────────────────────────────────────┘ │

│ │ │

│ │ ┌──────────────────────────────────────────────┐ │

│ │ │ Loan Portfolio │ │

│ │ │ 🟢 Active: 156 ($234,500) │ │

│ │ │ 🟡 Pending Approval: 23 ($78,900) │ │

│ │ │ 🔴 Overdue: 12 ($34,200) │ │

│ │ │ ⚪ Closed: 2,340 ($1,234,500) │ │

│ │ └──────────────────────────────────────────────┘ │

│ │ │

│ │ ┌──────────────────────────────────────────────┐ │

│ │ │ Recent Activity │ │

│ │ │ 🔵 New loan created (5m) LON-2945 $2,000 │ │

│ │ │ 🟢 Payment received (12m) $450 (LON-2940) │ │

│ │ │ 🟡 Loan pending approval (45m) LON-2944 │ │

│ │ │ 🔴 Payment overdue (1h) LON-2920 $300 │ │

│ │ │ 🟢 Customer registered (3h) CUST-5824 │ │

│ │ │ [View All >] │ │

│ │ └──────────────────────────────────────────────┘ │

│ │ │

│ │ ┌──────────────────────────────────────────────┐ │

│ │ │ Quick Stats │ │

│ │ │ Avg Loan Amount: $1,505 | Avg Tenure: 45d │ │

│ │ │ Collection Rate: 94.2% | Default Rate: 3.2%│ │

│ │ │ Customer Satisfaction: 4.8/5 ⭐ │ │

│ │ └──────────────────────────────────────────────┘ │

│ │ │

│ │ ┌──────────────────────────────────────────────┐ │

│ │ │ Alert & Notifications │ │

│ │ │ ⚠️ 12 loans approaching due date │ │

│ │ │ ⚠️ Monthly target: 82% achieved │ │

│ │ │ ℹ️ Backup completed successfully (1h ago) │ │

│ │ │ ✅ All systems operational │ │

│ │ └──────────────────────────────────────────────┘ │

│ │ │

└──────────┴──────────────────────────────────────────────────────┘

**📦 TENANT DASHBOARD SECTIONS (Compact)**

**1. KPI CARDS (Top Row - 4 Cards)**

**Card 1: Active Users**

Code

┌──────────────┐

│ 245 │

│ Active Users │

│ ↑ 8% vs week │

│ [👥 icon] │

└──────────────┘

Click → Navigate to Users page

**Card 2: Loans This Month**

Code

┌──────────────┐

│ 156 │

│ Loans This │

│ Month │

│ ↑ 12% vs avg │

│ [📊 icon] │

└──────────────┘

Click → Navigate to Loans page

**Card 3: Revenue This Month**

Code

┌──────────────┐

│ $45,230 │

│ Revenue │

│ This Month │

│ ↑ 15% target │

│ [💰 icon] │

└──────────────┘

Click → Navigate to Reports page

**Card 4: System Health**

Code

┌──────────────┐

│ 98.5% │

│ Health/ │

│ Uptime │

│ ✅ All OK │

│ [🟢 icon] │

└──────────────┘

Click → View system status details

**2. REVENUE TREND CHART**

Code

Chart Type: Line Chart (Last 30 Days)

┌────────────────────────────────────────┐

│ Revenue Trend (Last 30 Days) │

├────────────────────────────────────────┤

│ │

│ /\ /\ │

│ / \ / \ /\ / │

│ / \/ \ / \/ │

│ │

│ Avg: $1,507/day | Peak: $2,340 │

│ [📊 Download] [↻ Refresh] │

│ │

└────────────────────────────────────────┘

Features:

- Daily revenue bars/line

- Hover for exact amount

- Color code: Green (above avg), Yellow (avg), Red (below avg)

- Peak day highlighted

- Clickable days to drill down

**3. TOP PERFORMERS**

Code

Compact Card:

┌────────────────────────────────────────┐

│ Top Performers (This Month) │

├────────────────────────────────────────┤

│ 🥇 John Doe (Officer) │

│ $12,450 collected | 45 transactions │

│ │

│ 🥈 Jane Smith (Officer) │

│ $9,800 collected | 38 transactions │

│ │

│ 🥉 Bob Wilson (Cashier) │

│ $7,200 collected | 32 transactions │

│ │

│ [View All Performance >] │

└────────────────────────────────────────┘

Click row → View user profile & details

**4. LOAN PORTFOLIO STATUS**

Code

Donut/Pie Chart:

┌────────────────────────────────────────┐

│ Loan Portfolio │

├────────────────────────────────────────┤

│ ╭─────────╮ │

│ ╱ ╲ │

│ ╱ 🟢 156 Active ╲ │

│ │ ($234,500) │ │

│ │ 🟡 23 Pending │ Status: │

│ │ ($78,900) │ ✅ Healthy │

│ ╲ 🔴 12 Overdue ╱ │

│ ╲ ($34,200) ╱ │

│ ╰─────────╯ │

│ ⚪ 2,340 Closed ($1,234,500) │

│ │

│ Total Portfolio: $1,582,100 │

│ [View Detailed Report >] │

└────────────────────────────────────────┘

Click segments:

- Click "Active" → Show active loans list

- Click "Overdue" → Show overdue loans (RED ALERT)

- Click "Pending" → Show loans awaiting approval

**5. RECENT ACTIVITY LOG**

Code

Compact Timeline:

┌────────────────────────────────────────┐

│ Recent Activity [View All]│

├────────────────────────────────────────┤

│ 🔵 New loan created (5m) │

│ LON-2945 | $2,000 | John Doe │

│ │

│ 🟢 Payment received (12m) │

│ $450 collected | LON-2940 │

│ │

│ 🟡 Loan pending approval (45m) │

│ LON-2944 | $1,200 | Awaiting... │

│ │

│ 🔴 Payment overdue (1h) │

│ LON-2920 | $300 | URGENT │

│ │

│ 🟢 Customer registered (3h) │

│ CUST-5824 | John Smith │

│ │

└────────────────────────────────────────┘

Hover effects:

- Expand to show details

- Click → Navigate to entity

Colors:

- 🔵 Blue = Informational

- 🟢 Green = Success/Completed

- 🟡 Yellow = Warning/Pending

- 🔴 Red = Error/Urgent

**6. QUICK STATS**

Code

Single Row Metrics:

┌────────────────────────────────────────┐

│ Quick Stats │

├────────────────────────────────────────┤

│ Avg Loan: $1,505 | Avg Tenure: 45d │

│ Collection Rate: 94.2% ✅ │

│ Default Rate: 3.2% (Below 5% target) │

│ Customer Satisfaction: 4.8/5 ⭐⭐⭐⭐⭐ │

│ │

└────────────────────────────────────────┘

**7. ALERTS & NOTIFICATIONS**

Code

Alert Stack (Top or Bottom):

┌────────────────────────────────────────┐

│ ⚠️ 12 loans approaching due date │

│ [Review] [Dismiss] │

├────────────────────────────────────────┤

│ 📊 Monthly target: 82% achieved │

│ Need 18% more to hit 100% │

│ [Set Reminder] [Dismiss] │

├────────────────────────────────────────┤

│ ℹ️ Backup completed (1h ago) │

│ All data secured │

│ [Dismiss] │

├────────────────────────────────────────┤

│ ✅ All systems operational │

│ No issues detected │

│ [Dismiss] │

└────────────────────────────────────────┘

Dismissible alerts

Auto-dismiss after 5s (info/success)

Stay visible (warning/error)

**🎯 QUICK ACTION BUTTONS**

Code

Top bar near title:

[➕ New Loan] [➕ New Customer] [💳 Record Payment] [📈 View Reports]

Each button navigates to:

- New Loan → /loans/create

- New Customer → /customers/create

- Record Payment → /payments/create

- View Reports → /reports

**🔐 PERMISSION-BASED DASHBOARD**

**TENANT ADMIN SEES (Full Access):**

Code

✅ All KPI cards (all metrics)

✅ Revenue chart

✅ Top performers

✅ Loan portfolio

✅ All recent activities

✅ Quick action buttons ([New Loan], [New Customer], [Payment])

✅ All alerts visible

✅ All quick stats

**BRANCH MANAGER SEES (Limited):**

Code

✅ KPI cards (branch-only data)

✅ Revenue chart (branch data only)

✅ Top performers (branch staff only)

✅ Loan portfolio (branch loans only)

✅ Recent activity (branch activity only)

⚠️ Quick actions limited:

- [New Loan] → allowed

- [New Customer] → allowed

- [Record Payment] → allowed

❌ [View Reports] → hidden (view-only)

**LOAN OFFICER SEES (View Only):**

Code

✅ KPI cards (read-only)

✅ Revenue chart (read-only)

✅ Top performers (read-only)

✅ Loan portfolio (read-only)

✅ Recent activity (own activity only)

❌ Quick action buttons → HIDDEN

(Officers navigate to specific pages instead)

**CASHIER SEES (Minimal):**

Code

✅ KPI cards (payments only)

✅ Recent activity (payment activity only)

❌ Revenue chart → hidden

❌ Top performers → hidden

❌ Loan portfolio → hidden

❌ Quick stats → hidden

**📱 RESPONSIVE BEHAVIOR**

**Desktop (1024px+)**

Code

- Sidebar: 60px (collapsed)

- KPI Cards: 4 per row

- Charts: Full width (50% if side-by-side)

- Activity: Full width

- All sections visible

**Tablet (768px - 1023px)**

Code

- Sidebar: Hidden (hamburger)

- KPI Cards: 2 per row

- Charts: Full width

- Activity: Full width

- Some sections stacked

**Mobile (< 768px)**

Code

- Sidebar: Drawer menu

- KPI Cards: 1 per row or 2 if compact

- Charts: Full width, reduced height

- Activity: Compact view

- Only critical info shown

- Scroll to see all sections

**🌙 DARK/LIGHT THEME**

**Light Theme**

Code

Background: #FFFFFF

Cards: #F9FAFB

Text: #111827

Chart colors: Blues, Greens

Alert colors: Amber, Red, Green

**Dark Theme**

Code

Background: #030712

Cards: #111827

Text: #F3F4F6

Chart colors: Bright Blues, Greens

Alert colors: Bright Amber, Red, Green

**⚡ REAL-TIME UPDATES**

**WebSocket Features:**

* ✅ KPI cards update in real-time
* ✅ Recent activity feed updates instantly
* ✅ Alerts appear immediately
* ✅ Charts refresh every minute
* ✅ Portfolio status updates on new transactions
* ✅ Top performers rank updates

**🎬 USER INTERACTION FLOW**

**1. Tenant Admin Logs In**

Code

Login Page → Credentials → Backend verifies → JWT token

↓

Redirects to: /tenant/dashboard

↓

Dashboard loads with:

- User's tenant data

- User's role permissions

- All tenant statistics

- All tenant activities

**2. Admin Views Dashboard**

Code

Sees all 7 sections (KPI, chart, performers, portfolio, activity, stats, alerts)

Click sections to drill down

Click buttons to take actions

**3. Admin Clicks [New Loan]**

Code

Modal or Page: Create Loan Form

- Form appears with fields

- Pre-filled tenant info

- Validation on submit

- Show confirmation before saving

**4. Form Submitted**

Code

Backend:

1. Verify user has "loans:create" permission

2. Verify tenant\_id matches user's tenant

3. Create loan in database

4. Return success + new loan ID

Frontend:

1. Show toast: "✅ Loan created successfully"

2. Refresh recent activity

3. Update KPI cards

4. Close modal/form

5. Optionally navigate to loan details

**📊 DATA SHOWN (EXAMPLES)**

**Demo Pawnshop Tenant Data**

Code

Active Users: 245

- 12 Admins

- 45 Loan Officers

- 68 Cashiers

- 120 Appraisers

Loans This Month: 156

- Active: 156

- Pending Approval: 23

- Overdue: 12

- Closed (this month): 89

Revenue This Month: $45,230

- Loan Interest: $32,100

- Appraisal Fees: $8,500

- Late Fees: $4,630

Team Performance:

- John Doe: $12,450 (45 transactions)

- Jane Smith: $9,800 (38 transactions)

- Bob Wilson: $7,200 (32 transactions)

Health Metrics:

- System Uptime: 98.5%

- API Response: 85ms avg

- Database: Healthy

- Storage: 45% used

**✅ SUMMARY: TENANT ADMIN DASHBOARD**

**Tenant Admin Dashboard provides:**

* 🎯 **Complete Tenant Overview** → All metrics at a glance
* 📊 **Financial Analytics** → Revenue trends, performance
* 👥 **Team Insights** → Top performers, activity tracking
* 📋 **Portfolio Management** → Loan status overview
* 🔔 **Real-Time Alerts** → Urgent notifications
* 🎮 **Quick Actions** → Common tasks one-click away
* 🔐 **Role-Based** → Shows only accessible data
* 📱 **Responsive** → Works on any device
* 🌙 **Themes** → Dark and light modes
* ⚡ **Real-Time** → WebSocket updates

**🎯 EXPECTED BEHAVIOR**

1. **Load Time:** < 2 seconds
2. **Real-Time Updates:** < 100ms delay
3. **Charts Render:** < 500ms
4. **Responsive:** Smooth on mobile/tablet/desktop
5. **Accessibility:** WCAG compliant
6. **Performance:** Optimized queries, cached data

**Copilot said: Perfect! 🎯 Let me create the \*\*COMPLETE**

Perfect! 🎯 Let me create the **COMPLETE SIDEBAR CHILD MENUS** for both System and Tenant spaces:

**📋 SIDEBAR CHILD MENUS - COMPLETE HIERARCHY**

**Current Time:** 2025-10-21 11:59:51 UTC  
**Logged In As:** apps-eduard (System Admin)

**🏗️ SYSTEM SPACE SIDEBAR (System Admin)**

**PARENT: 🏠 Dashboard**

Code

No children

Direct link: /system/dashboard

Icon: 🏠

Action: Click → Navigate to System Admin Dashboard

**PARENT: 🏢 Tenants**

Code

🏢 Tenants (Expandable)

├─ 📋 List All Tenants

│ Link: /system/tenants

│ Permissions: tenants:view

│ Shows: Table of all tenants (name, users, status, revenue)

│ Actions: View, Edit, Suspend, Delete

│

├─ ➕ Create Tenant

│ Link: /system/tenants/create

│ Permissions: tenants:create

│ Shows: Form to create new tenant

│ Fields: Company name, subdomain, plan, region, etc.

│

├─ 📄 Tenant Details

│ Link: /system/tenants/:id

│ Permissions: tenants:view

│ Shows: Full tenant information

│ Sections: Overview, Users, Settings, Billing, Activity

│

└─ ⚙️ Tenant Settings

Link: /system/tenants/settings

Permissions: tenants:edit

Shows: Global tenant configuration

Settings: Branding, notifications, data residency, etc.

**PARENT: 👥 System Users**

Code

👥 System Users (Expandable)

├─ 📋 List All Users

│ Link: /system/users

│ Permissions: system-users:view

│ Shows: Table of all system users (admins, staff, support)

│ Columns: Name, Email, Role, Status, Last Login

│ Actions: View, Edit, Suspend, Delete, View Permissions

│

├─ ➕ Create User

│ Link: /system/users/create

│ Permissions: system-users:create

│ Shows: Form to create new system user

│ Fields: Email, Name, Role, Status

│

├─ 👤 User Details

│ Link: /system/users/:id

│ Permissions: system-users:view

│ Shows: Complete user profile

│ Sections: Basic Info, Role & Permissions, Activity Log, Audit Trail

│

└─ 🔐 User Permissions

Link: /system/users/:id/permissions

Permissions: system-users:edit

Shows: Detailed permission matrix for user

Actions: Add role, remove role, view inherited permissions

**PARENT: 👤 System Roles**

Code

👤 System Roles (Expandable)

├─ 📋 List All Roles

│ Link: /system/roles

│ Permissions: system-roles:view

│ Shows: Table of all system roles

│ Columns: Role Name, Space, Users Count, Status

│ Example Roles:

│ - Super Admin (all permissions)

│ - System Admin (manage tenants & users)

│ - Support Staff (view only)

│ - IT Support (system management)

│ - Billing Manager (billing only)

│ - Auditor (audit logs only)

│

├─ ➕ Create Role

│ Link: /system/roles/create

│ Permissions: system-roles:create

│ Shows: Form to create new system role

│ Fields: Role name, description, space (system/tenant)

│

├─ 👤 Role Details

│ Link: /system/roles/:id

│ Permissions: system-roles:view

│ Shows: Complete role information

│ Sections: Overview, Permissions, Users, Audit Trail

│

└─ 📊 Permission Matrix

Link: /system/roles/:id/permissions

Permissions: system-roles:edit

Shows: Matrix view (Modules vs Actions)

Example:

╔═══════════════╦═══════╦═══════╦═══════╦═══════╗

║ Module/Action ║ View ║Create ║ Edit ║Delete ║

╠═══════════════╬═══════╬═══════╬═══════╬═══════╣

║ Dashboard ║ ✅ ║ - ║ - ║ - ║

║ Tenants ║ ✅ ║ ✅ ║ ✅ ║ ✅ ║

║ System Users ║ ✅ ║ ✅ ║ ✅ ║ - ║

║ Billing ║ ✅ ║ - ║ - ║ - ║

║ Audit Logs ║ ✅ ║ - ║ - ║ - ║

╚═══════════════╩═══════╩═══════╩═══════╩═══════╝

Actions: Toggle checkboxes, add/remove permissions

**PARENT: 💰 Billing**

Code

💰 Billing (Expandable)

├─ 📑 Subscriptions

│ Link: /system/billing/subscriptions

│ Permissions: billing:view

│ Shows: List of all tenant subscriptions

│ Columns: Tenant Name, Plan, Users, Price/Month, Status, Renewal Date

│ Actions: View, Upgrade/Downgrade, Suspend, Cancel

│

├─ 📄 Invoices

│ Link: /system/billing/invoices

│ Permissions: billing:view

│ Shows: All issued invoices

│ Columns: Invoice ID, Tenant, Amount, Date, Status, Download

│ Filters: Date range, tenant, status

│

├─ 💳 Payments

│ Link: /system/billing/payments

│ Permissions: billing:view

│ Shows: Payment transaction history

│ Columns: Payment ID, Tenant, Amount, Method, Date, Status

│ Filters: Date range, payment method, status

│

└─ ⚙️ Billing Settings

Link: /system/billing/settings

Permissions: billing:edit

Shows: Global billing configuration

Settings: Payment methods, tax rates, invoice templates, etc.

**PARENT: 📋 Audit Logs**

Code

📋 Audit Logs (Expandable)

├─ 📊 System Activity

│ Link: /system/audit/system

│ Permissions: audit:view

│ Shows: All platform-level activities

│ Columns: Timestamp, Event Type, Performed By, Entity, Result

│ Examples:

│ - Tenant created/deleted

│ - System user added/removed

│ - Role permissions changed

│ - System settings modified

│ - Database backup completed

│

├─ 👤 User Activity

│ Link: /system/audit/users

│ Permissions: audit:view

│ Shows: User-level activities across all tenants

│ Columns: Timestamp, User, Action, Tenant, Entity, Result

│ Filters: User, date range, action type

│

├─ 📋 Compliance Reports

│ Link: /system/audit/reports

│ Permissions: audit:view

│ Shows: Pre-built compliance reports

│ Reports: Monthly activity, user access, role changes, failed attempts

│

└─ ⬇️ Export Logs

Link: /system/audit/export

Permissions: audit:export

Shows: Export audit data in various formats

Formats: CSV, PDF, Excel, JSON

Options: Date range, event type, entity filters

**PARENT: ⚙️ Settings**

Code

⚙️ Settings (Expandable)

├─ ⚙️ General Settings

│ Link: /system/settings/general

│ Permissions: settings:view

│ Shows: Platform-wide settings

│ Settings:

│ - Platform name

│ - Logo & branding

│ - Time zone

│ - Default language

│ - Currency

│

├─ 📧 Email Configuration

│ Link: /system/settings/email

│ Permissions: settings:edit

│ Shows: Email service configuration

│ Settings:

│ - SMTP server

│ - From email address

│ - Email templates

│ - Notification settings

│

├─ 🔐 Security Settings

│ Link: /system/settings/security

│ Permissions: settings:edit

│ Shows: Security policies

│ Settings:

│ - Password policies

│ - MFA requirements

│ - Session timeout

│ - IP whitelist

│ - Rate limiting

│

├─ 📦 Module Registry

│ Link: /system/settings/modules

│ Permissions: settings:view

│ Shows: All available modules/features

│ Display: Module key, name, icon, status, actions

│ Actions:

│ - Enable/Disable module

│ - View module details

│ - Configure module settings

│ Example Modules:

│ - Dashboard

│ - Loan Management

│ - Customer Management

│ - Payment Processing

│ - Appraisal System

│ - Reports & Analytics

│ - Audit System

│ - User Management

│

└─ 💚 System Health

Link: /system/settings/health

Permissions: settings:view

Shows: Real-time system status

Metrics:

- 🟢 API Server Status

- 🟢 Database Status

- 🟢 Redis Cache Status

- 🟢 Email Service Status

- 🟢 Storage Status

- 🟢 Notification Queue Status

Details: Uptime, response times, error logs

**🏢 TENANT SPACE SIDEBAR (Tenant Admin)**

**PARENT: 🏠 Dashboard**

Code

No children

Direct link: /tenant/dashboard

Icon: 🏠

Action: Click → Navigate to Tenant Dashboard

**PARENT: 👥 Users**

Code

👥 Users (Expandable)

├─ 📋 List All Users

│ Link: /tenant/users

│ Permissions: users:view

│ Shows: Table of all tenant users

│ Columns: Name, Email, Role, Department, Status, Last Login

│ Actions: View, Edit, Suspend, Delete, View Permissions

│

├─ ➕ Create User

│ Link: /tenant/users/create

│ Permissions: users:create

│ Shows: Form to create new tenant user

│ Fields: Email, First Name, Last Name, Role, Department, Status

│

├─ 👤 User Details

│ Link: /tenant/users/:id

│ Permissions: users:view

│ Shows: Complete user profile

│ Sections: Basic Info, Role, Permissions, Activity Log

│

└─ 🔐 User Permissions

Link: /tenant/users/:id/permissions

Permissions: users:edit

Shows: User's permission matrix

Actions: Assign/remove roles, view inherited permissions

**PARENT: 👤 Roles**

Code

👤 Roles (Expandable)

├─ 📋 List All Roles

│ Link: /tenant/roles

│ Permissions: roles:view

│ Shows: Table of all tenant roles

│ Columns: Role Name, Users Count, Permissions Count, Status

│ Example Roles:

│ - Tenant Admin (all permissions)

│ - Branch Manager (branch-level access)

│ - Loan Officer (loans & customers)

│ - Cashier/Teller (payments only)

│ - Appraiser (appraisals only)

│ - Accountant/Auditor (reports & audit)

│ - Viewer/Read-Only (view all, no actions)

│

├─ ➕ Create Role

│ Link: /tenant/roles/create

│ Permissions: roles:create

│ Shows: Form to create new role

│ Fields: Role name, description, status

│

├─ 👤 Role Details

│ Link: /tenant/roles/:id

│ Permissions: roles:view

│ Shows: Complete role information

│ Sections: Overview, Permissions Matrix, Users, Audit Trail

│

└─ 📊 Permission Matrix

Link: /tenant/roles/:id/permissions

Permissions: roles:edit

Shows: Matrix view (Modules vs Actions)

Example:

╔════════════════╦═══════╦═══════╦═══════╦═══════╗

║ Module/Action ║ View ║Create ║ Edit ║Delete ║

╠════════════════╬═══════╬═══════╬═══════╬═══════╣

║ Loans ║ ✅ ║ ✅ ║ ✅ ║ - ║

║ Customers ║ ✅ ║ ✅ ║ ✅ ║ - ║

║ Payments ║ ✅ ║ ✅ ║ - ║ - ║

║ Appraisals ║ ✅ ║ - ║ - ║ - ║

║ Reports ║ ✅ ║ - ║ - ║ - ║

║ User Mgmt ║ - ║ - ║ - ║ - ║

╚════════════════╩═══════╩═══════╩═══════╩═══════╝

Actions: Toggle permissions, set constraints (if needed)

**PARENT: 📊 Loans**

Code

📊 Loans (Expandable)

├─ 📋 List All Loans

│ Link: /tenant/loans

│ Permissions: loans:view

│ Shows: Table of all tenant loans

│ Columns: Loan ID, Borrower, Amount, Status, Officer, Due Date

│ Filters: Status, date range, amount range, officer

│ Status colors: 🟢 Active, 🟡 Pending, 🔴 Overdue, ⚪ Closed

│ Actions: View, Edit, Approve, Delete, Generate Statement

│

├─ ➕ Create Loan

│ Link: /tenant/loans/create

│ Permissions: loans:create

│ Shows: Form to create new loan

│ Steps:

│ 1. Select customer (or create new)

│ 2. Enter loan details (amount, rate, term)

│ 3. Select collateral (optional)

│ 4. Set terms & conditions

│ 5. Review & submit

│

├─ 📄 Loan Details

│ Link: /tenant/loans/:id

│ Permissions: loans:view

│ Shows: Complete loan information

│ Sections:

│ - Basic Info (amount, rate, term, status)

│ - Borrower Info (customer details)

│ - Payment Schedule (table of payments)

│ - Collateral (appraisal details)

│ - Activity Timeline (all actions on this loan)

│ - Payments (received & pending)

│

├─ ✅ Approve Loan

│ Link: /tenant/loans/:id/approve

│ Permissions: loans:approve

│ Shows: Approval form

│ Actions: Approve, Reject (with reason), Request Changes

│

├─ 📊 Loan Reports

│ Link: /tenant/loans/reports

│ Permissions: loans:view

│ Shows: Loan analytics & reports

│ Reports:

│ - Portfolio summary

│ - Performance metrics

│ - Overdue analysis

│ - Default predictions

│ - Revenue breakdown

│

└─ 📈 Loan Analytics

Link: /tenant/loans/analytics

Permissions: loans:view

Shows: Advanced loan analytics

Charts: Growth, distribution, performance trends

**PARENT: 👥 Customers**

Code

👥 Customers (Expandable)

├─ 📋 List All Customers

│ Link: /tenant/customers

│ Permissions: customers:view

│ Shows: Table of all customers

│ Columns: Customer ID, Name, Email, Phone, Status, Loans, Total Borrowed

│ Filters: Status, date range, loan status

│ Actions: View, Edit, Delete, View Loans, View History

│

├─ ➕ Create Customer

│ Link: /tenant/customers/create

│ Permissions: customers:create

│ Shows: Form to create new customer

│ Fields: Personal info, contact, identification, address

│

├─ 👤 Customer Details

│ Link: /tenant/customers/:id

│ Permissions: customers:view

│ Shows: Complete customer profile

│ Sections:

│ - Personal Info

│ - Contact Information

│ - Identification Documents

│ - Loan History (all loans)

│ - Payment History

│ - Credit Score

│ - Activity Log

│

└─ 📜 Customer History

Link: /tenant/customers/:id/history

Permissions: customers:view

Shows: Complete transaction history

Timeline: All loans, payments, interactions, communications

**PARENT: 💳 Payments**

Code

💳 Payments (Expandable)

├─ 📋 Payment History

│ Link: /tenant/payments

│ Permissions: payments:view

│ Shows: Table of all payments received

│ Columns: Payment ID, Loan, Amount, Date, Method, Status

│ Filters: Date range, payment method, status

│

├─ ➕ Record Payment

│ Link: /tenant/payments/create

│ Permissions: payments:create

│ Shows: Form to record new payment

│ Steps:

│ 1. Select loan or customer

│ 2. Enter payment amount

│ 3. Select payment method (cash, check, transfer)

│ 4. Add notes

│ 5. Submit

│

├─ 💰 Refunds

│ Link: /tenant/payments/refunds

│ Permissions: payments:view

│ Shows: Refund history & processing

│ Actions: Process refund, view refund status

│

└─ 📊 Payment Reports

Link: /tenant/payments/reports

Permissions: payments:view

Shows: Payment analytics

Reports:

- Daily/monthly collection summary

- Payment method breakdown

- Late payment tracking

- Refund summary

**PARENT: 📑 Appraisals**

Code

📑 Appraisals (Expandable)

├─ 📋 List Appraisals

│ Link: /tenant/appraisals

│ Permissions: appraisals:view

│ Shows: Table of all appraisals

│ Columns: Appraisal ID, Loan, Collateral, Amount, Status, Appraiser, Date

│ Filters: Status, date range, appraiser

│

├─ ➕ Create Appraisal

│ Link: /tenant/appraisals/create

│ Permissions: appraisals:create

│ Shows: Form to create appraisal

│ Fields: Select collateral, details, images, valuation

│

├─ 📄 Appraisal Details

│ Link: /tenant/appraisals/:id

│ Permissions: appraisals:view

│ Shows: Complete appraisal report

│ Sections: Item details, valuation, images, condition, status

│

└─ ✅ Approve Appraisal

Link: /tenant/appraisals/:id/approve

Permissions: appraisals:approve

Shows: Approval form

Actions: Approve, Reject, Request Changes

**PARENT: 📈 Reports**

Code

📈 Reports (Expandable)

├─ 💰 Financial Reports

│ Link: /tenant/reports/financial

│ Permissions: reports:view

│ Shows: Financial performance reports

│ Reports:

│ - Revenue summary

│ - Expense breakdown

│ - Profit & loss

│ - Cash flow analysis

│ - Budget vs actual

│

├─ 👤 User Activity Report

│ Link: /tenant/reports/activity

│ Permissions: reports:view

│ Shows: User activity analytics

│ Metrics: Loans created, payments processed, approvals, etc.

│

├─ 📊 Loan Analytics

│ Link: /tenant/reports/loans

│ Permissions: reports:view

│ Shows: Loan performance reports

│ Reports:

│ - Portfolio overview

│ - Performance metrics

│ - Default analysis

│ - Overdue tracking

│

└─ 📋 Custom Reports

Link: /tenant/reports/custom

Permissions: reports:view

Shows: Build custom reports

Features: Select metrics, date range, filters, export options

**PARENT: 📋 Audit Logs**

Code

📋 Audit Logs (Expandable)

├─ 📊 Tenant Activity

│ Link: /tenant/audit

│ Permissions: audit:view

│ Shows: All tenant activities

│ Columns: Timestamp, User, Action, Entity, Result

│ Filters: User, date range, action type

│ Examples: Loans created, payments received, users added, etc.

│

├─ 📋 Compliance Reports

│ Link: /tenant/audit/compliance

│ Permissions: audit:view

│ Shows: Pre-built compliance reports

│ Reports: Monthly activity, user access logs, role changes

│

└─ ⬇️ Export Logs

Link: /tenant/audit/export

Permissions: audit:export

Shows: Export audit data

Formats: CSV, PDF, Excel, JSON

Options: Date range, user filters, action type filters

**PARENT: ⚙️ Settings**

Code

⚙️ Settings (Expandable)

├─ 🏢 Tenant Information

│ Link: /tenant/settings/general

│ Permissions: settings:view

│ Shows: Tenant company information

│ Fields:

│ - Company name

│ - Subdomain

│ - Logo & branding

│ - Contact information

│ - Address

│ - Region & timezone

│

├─ 👥 Team Members

│ Link: /tenant/settings/team

│ Permissions: settings:view

│ Shows: Team member settings

│ Actions:

│ - Invite new members

│ - Manage team members

│ - Set team roles

│

├─ 🔔 Notifications

│ Link: /tenant/settings/notifications

│ Permissions: settings:edit

│ Shows: Notification preferences

│ Settings:

│ - Email notifications

│ - SMS alerts

│ - Push notifications

│ - Alert triggers

│ - Recipient list

│

└─ 👤 Preferences

Link: /tenant/settings/preferences

Permissions: settings:edit

Shows: User preferences

Settings:

- Language

- Timezone

- Theme (dark/light)

- Date & number formats

- Default views

**📊 SIDEBAR MENU STRUCTURE (Visual)**

**SYSTEM SPACE - Fully Expanded**

Code

🔷 ExITS-SaaS

━━━━━━━━━━━━━━━━

🏠 Dashboard

🏢 Tenants ▼

├─ 📋 List All Tenants

├─ ➕ Create Tenant

├─ 📄 Tenant Details

└─ ⚙️ Tenant Settings

👥 System Users ▼

├─ 📋 List All Users

├─ ➕ Create User

├─ 👤 User Details

└─ 🔐 User Permissions

👤 System Roles ▼

├─ 📋 List All Roles

├─ ➕ Create Role

├─ 👤 Role Details

└─ 📊 Permission Matrix

💰 Billing ▼

├─ 📑 Subscriptions

├─ 📄 Invoices

├─ 💳 Payments

└─ ⚙️ Billing Settings

📋 Audit Logs ▼

├─ 📊 System Activity

├─ 👤 User Activity

├─ 📋 Compliance Reports

└─ ⬇️ Export Logs

⚙️ Settings ▼

├─ ⚙️ General Settings

├─ 📧 Email Configuration

├─ 🔐 Security Settings

├─ 📦 Module Registry

└─ 💚 System Health

**TENANT SPACE - Fully Expanded**

Code

🔷 ExITS-SaaS

━━━━━━━━━━━━━━━━

🏠 Dashboard

👥 Users ▼

├─ 📋 List All Users

├─ ➕ Create User

├─ 👤 User Details

└─ 🔐 User Permissions

👤 Roles ▼

├─ 📋 List All Roles

├─ ➕ Create Role

├─ 👤 Role Details

└─ 📊 Permission Matrix

📊 Loans ▼

├─ 📋 List All Loans

├─ ➕ Create Loan

├─ 📄 Loan Details

├─ ✅ Approve Loan

├─ 📊 Loan Reports

└─ 📈 Loan Analytics

👥 Customers ▼

├─ 📋 List All Customers

├─ ➕ Create Customer

├─ 👤 Customer Details

└─ 📜 Customer History

💳 Payments ▼

├─ 📋 Payment History

├─ ➕ Record Payment

├─ 💰 Refunds

└─ 📊 Payment Reports

📑 Appraisals ▼

├─ 📋 List Appraisals

├─ ➕ Create Appraisal

├─ 📄 Appraisal Details

└─ ✅ Approve Appraisal

📈 Reports ▼

├─ 💰 Financial Reports

├─ 👤 User Activity Report

├─ 📊 Loan Analytics

└─ 📋 Custom Reports

📋 Audit Logs ▼

├─ 📊 Tenant Activity

├─ 📋 Compliance Reports

└─ ⬇️ Export Logs

⚙️ Settings ▼

├─ 🏢 Tenant Information

├─ 👥 Team Members

├─ 🔔 Notifications

└─ 👤 Preferences

**🔐 CHILD MENU VISIBILITY (By Role)**

**SYSTEM SPACE - System Super Admin Sees ALL**

Code

✅ Dashboard

✅ Tenants (all children)

✅ System Users (all children)

✅ System Roles (all children)

✅ Billing (all children)

✅ Audit Logs (all children)

✅ Settings (all children)

**SYSTEM SPACE - System Support Sees LIMITED**

Code

✅ Dashboard

✅ Tenants → List All Tenants (view-only)

✅ System Users → List All Users (view-only)

❌ Tenants → Create Tenant (hidden)

❌ Tenants → Tenant Settings (hidden)

❌ System Users → Create User (hidden)

❌ System Users → User Permissions (hidden)

✅ Audit Logs → System Activity (view-only)

❌ Billing (hidden)

❌ Settings (hidden)

**TENANT SPACE - Tenant Admin Sees ALL**

Code

✅ Dashboard

✅ Users (all children)

✅ Roles (all children)

✅ Loans (all children)

✅ Customers (all children)

✅ Payments (all children)

✅ Appraisals (all children)

✅ Reports (all children)

✅ Audit Logs (all children)

✅ Settings (all children)

**TENANT SPACE - Loan Officer Sees LIMITED**

Code

✅ Dashboard

✅ Loans → List All Loans

✅ Loans → Create Loan

✅ Loans → Loan Details

❌ Loans → Approve Loan (if not approver)

❌ Loans → Loan Reports (view only)

✅ Customers → List All Customers (view-only)

❌ Payments (hidden)

❌ Appraisals (hidden)

❌ Audit Logs (hidden)

❌ Settings (hidden)

❌ Roles (hidden)

❌ Users (hidden)

**✅ COMPLETE SUMMARY**

**System Space Sidebar:**

* 1 standalone menu (Dashboard)
* 6 expandable parent menus
* 24 total child menus

**Tenant Space Sidebar:**

* 1 standalone menu (Dashboard)
* 9 expandable parent menus
* 31 total child menus

**Total:** 55 unique menu items across both spaces

**Copilot said: Perfect! 🎯 Let me create a \*\*COMPREHENSIVE**

Perfect! 🎯 Let me create a **COMPREHENSIVE TODO CHECKLIST** for building the complete **ExITS-SaaS-Template** project:

**✅ COMPLETE PROJECT CHECKLIST**

**Project:** ExITS-SaaS-Template  
**Owner:** apps-eduard  
**Start Date:** 2025-10-21 12:03:10 UTC  
**Repository:** apps-eduard/ExITS-SaaS-Template  
**Status:** 🔴 NOT STARTED

**📋 PHASE 1: REPOSITORY SETUP & FOUNDATION**

**A. Repository Creation**

* Create GitHub repository: apps-eduard/ExITS-SaaS-Template
* Set visibility to: Public
* Initialize with README.md
* Add .gitignore (Node.js template)
* Add MIT LICENSE
* Create branch protection rules (main branch)

**B. Root Directory Structure**

* Create /api directory
* Create /web directory
* Create /mobile directory
* Create /devops directory
* Create /.github/workflows directory
* Create /docs directory
* Create root README.md (project overview)
* Create SETUP.md (quick start guide)
* Create .env.example (root level)
* Create .gitignore (root level)
* Create root package.json

**C. Documentation Foundation**

* Create /docs/ARCHITECTURE.md
* Create /docs/RBAC-GUIDE.md
* Create /docs/DATABASE-SCHEMA.md
* Create /docs/API-DOCUMENTATION.md
* Create /docs/DEPLOYMENT.md
* Create /docs/SECURITY.md
* Create /docs/CONTRIBUTING.md
* Create /docs/TROUBLESHOOTING.md

**🗄️ PHASE 2: DATABASE & BACKEND FOUNDATION**

**A. Database Schema & Migration Scripts**

* Create /api/src/scripts/schema.sql
  + Users table (system + tenant)
  + Roles table
  + Modules table (menu registry)
  + Role\_Permissions table
  + Tenants table
  + Create indexes for performance
* Create /api/src/scripts/migrate.js
* Create /api/src/scripts/seed.js with test data:
  + System Super Admin user
  + System Admin user
  + System Support user
  + Tenant Admin user
  + Loan Officer user
  + Cashier user
  + Viewer user
  + System roles with permissions
  + Tenant roles with permissions
  + Sample tenants
  + Sample modules/menus
* Create /api/src/scripts/migrate-fresh.js (reset database)

**B. Backend Project Setup**

* Create /api/package.json with all dependencies:
  + express
  + postgresql (or sequelize/typeorm)
  + jsonwebtoken
  + bcryptjs
  + cors
  + helmet
  + dotenv
  + joi (validation)
  + winston (logging)
  + jest (testing)
  + supertest (API testing)
* Create /api/.env.example
* Create /api/jest.config.js
* Create /api/Dockerfile

**C. Backend Configuration**

* Create /api/src/config/database.js (PostgreSQL connection)
* Create /api/src/config/env.js (environment variables)
* Create /api/src/config/constants.js (app constants)
* Create /api/src/utils/logger.js (structured logging)
* Create /api/src/utils/jwt.js (JWT utilities)
* Create /api/src/utils/validators.js (input validation)

**D. Backend Middleware**

* Create /api/src/middleware/auth.middleware.js
  + JWT token verification
  + User extraction
  + Token refresh logic
* Create /api/src/middleware/rbac.middleware.js
  + Menu key verification
  + Action key verification
  + Permission checking
* Create /api/src/middleware/tenant-isolation.middleware.js
  + Tenant ID verification
  + Data access validation
  + Cross-tenant prevention
* Create /api/src/middleware/error.middleware.js
  + Error handling
  + Status code mapping
  + Error response formatting
* Create /api/src/middleware/logger.middleware.js
  + Request logging
  + Response logging
  + Performance metrics

**E. Backend Services**

* Create /api/src/services/auth.service.js
  + User login logic
  + Password hashing/verification
  + JWT generation
  + Permission fetching
* Create /api/src/services/user.service.js
  + User CRUD operations
  + User filtering
  + Status management
* Create /api/src/services/role.service.js
  + Role CRUD operations
  + Role filtering
  + Permission inheritance
* Create /api/src/services/permission.service.js
  + Permission checking
  + Permission caching
  + Permission updates
* Create /api/src/services/tenant.service.js
  + Tenant CRUD operations
  + Tenant provisioning
  + Tenant data isolation
* Create /api/src/services/cache.service.js
  + Permission caching
  + Cache invalidation
  + Redis connection (optional)
* Create /api/src/services/audit.service.js
  + Audit log creation
  + Activity tracking
  + Compliance logging

**F. Backend Controllers**

* Create /api/src/controllers/auth.controller.js
  + POST /api/auth/login
  + POST /api/auth/logout
  + POST /api/auth/refresh
  + GET /api/auth/me (current user profile)
* Create /api/src/controllers/user.controller.js
  + GET /api/users (list)
  + POST /api/users (create)
  + GET /api/users/:id (details)
  + PUT /api/users/:id (update)
  + DELETE /api/users/:id (delete)
  + GET /api/users/:id/permissions (user permissions)
* Create /api/src/controllers/role.controller.js
  + GET /api/roles (list)
  + POST /api/roles (create)
  + GET /api/roles/:id (details)
  + PUT /api/roles/:id (update)
  + DELETE /api/roles/:id (delete)
  + GET /api/roles/:id/permissions (role permissions matrix)
  + PUT /api/roles/:id/permissions (update permissions)
* Create /api/src/controllers/module.controller.js
  + GET /api/modules (list all modules)
  + POST /api/modules (create module)
  + PUT /api/modules/:id (update module)
  + GET /api/modules/:id (module details)
* Create /api/src/controllers/tenant.controller.js
  + GET /api/tenants (list)
  + POST /api/tenants (create)
  + GET /api/tenants/:id (details)
  + PUT /api/tenants/:id (update)
  + DELETE /api/tenants/:id (delete)
  + PUT /api/tenants/:id/status (suspend/activate)
* Create /api/src/controllers/audit.controller.js
  + GET /api/audit-logs (list activities)
  + GET /api/audit-logs/:id (details)
  + POST /api/audit-logs/export (export logs)
  + GET /api/audit-logs/compliance (compliance reports)
* Create /api/src/controllers/delegation.controller.js
  + POST /api/permissions/delegate (grant temporary access)
  + DELETE /api/permissions/delegate/:id (revoke delegation)
  + GET /api/permissions/delegations (list delegations)

**G. Backend Routes**

* Create /api/src/routes/auth.routes.js
* Create /api/src/routes/user.routes.js
* Create /api/src/routes/role.routes.js
* Create /api/src/routes/module.routes.js
* Create /api/src/routes/tenant.routes.js
* Create /api/src/routes/audit.routes.js
* Create /api/src/routes/index.js (main router)

**H. Backend Entry Point**

* Create /api/src/server.js (Express server setup)
* Create /api/src/index.js (app entry point)
* Setup middleware order (CORS, Helmet, Logger, Auth, etc.)
* Setup error handling

**I. Backend Tests**

* Create /api/tests/setup.js (test environment)
* Create /api/tests/auth.test.js
  + Login endpoint tests
  + JWT generation tests
  + Token refresh tests
* Create /api/tests/rbac.test.js
  + Permission checking tests
  + Role permission tests
  + Tenant isolation tests
* Create /api/tests/user.test.js
  + CRUD operations
  + Permission assignment
  + Validation tests
* Setup test coverage reporting

**J. Backend Documentation**

* Create /api/README.md
  + Setup instructions
  + Environment variables
  + API endpoints
  + Database schema
  + Running tests

**🎨 PHASE 3: FRONTEND (ANGULAR) - SYSTEM ADMIN**

**A. Angular Project Setup**

* Create /web/package.json with dependencies:
  + @angular/core
  + @angular/router
  + @angular/common
  + @angular/forms
  + @angular/http
  + tailwindcss
  + jest
  + typescript
* Create /web/angular.json
* Create /web/tsconfig.json
* Create /web/tailwind.config.js
* Create /web/jest.config.js
* Create /web/.env.example
* Create /web/Dockerfile

**B. Angular Core Services**

* Create /web/src/app/core/services/auth.service.ts
  + Login method
  + Logout method
  + Current user state
  + Token management
  + User permissions getter
* Create /web/src/app/core/services/permission.service.ts
  + Check permission method
  + Get user menus
  + Get available actions
  + Cache permissions
* Create /web/src/app/core/services/user.service.ts
  + Get all users
  + Create user
  + Update user
  + Delete user
  + Get user permissions
* Create /web/src/app/core/services/role.service.ts
  + Get all roles
  + Create role
  + Update role
  + Delete role
  + Get role permissions
  + Update role permissions
* Create /web/src/app/core/services/tenant.service.ts
  + Get all tenants
  + Create tenant
  + Update tenant
  + Get tenant details
  + Update tenant status
* Create /web/src/app/core/services/theme.service.ts
  + Toggle dark/light mode
  + Save theme preference
  + Load theme preference
* Create /web/src/app/core/services/notification.service.ts
  + Show success toast
  + Show error toast
  + Show warning toast
  + Show info toast
  + Show confirmation dialog
* Create /web/src/app/core/services/http.service.ts
  + HTTP interceptor setup
  + Request/response handling
  + Error handling

**C. Angular Guards & Interceptors**

* Create /web/src/app/core/guards/auth.guard.ts
  + Check if user is authenticated
  + Redirect to login if not
* Create /web/src/app/core/guards/role.guard.ts
  + Check menu key permission
  + Redirect to access-denied if not allowed
* Create /web/src/app/core/interceptors/auth.interceptor.ts
  + Add JWT token to requests
  + Handle token expiration
  + Refresh token logic
* Create /web/src/app/core/interceptors/error.interceptor.ts
  + Handle HTTP errors
  + Show error toasts
  + Log errors

**D. Angular Directives**

* Create /web/src/app/core/directives/has-permission.directive.ts
  + \*appHasPermission="'module:action'"
  + Show/hide elements based on permission
* Create /web/src/app/core/directives/has-role.directive.ts
  + \*appHasRole="'roleName'"
  + Show/hide elements based on role

**E. Shared UI Components**

* Create /web/src/app/shared/components/header/header.component.ts
  + Logo & app name
  + Search bar
  + Notification bell
  + User profile dropdown
  + Theme toggle
  + Hamburger menu (mobile)
* Create /web/src/app/shared/components/sidebar/sidebar.component.ts
  + Menu rendering logic
  + Role-based menu filtering
  + Collapse/expand functionality
  + Active menu highlighting
  + Child menu rendering
* Create /web/src/app/shared/components/card/card.component.ts
  + Generic card wrapper
  + Title, content, footer sections
  + Optional click action
* Create /web/src/app/shared/components/table/table.component.ts
  + Generic table component
  + Sortable columns
  + Filterable columns
  + Pagination
  + Row actions
  + Selectable rows
* Create /web/src/app/shared/components/button/button.component.ts
  + Primary, secondary, danger variants
  + Disabled state
  + Loading state
  + Size variants (small, regular, large)
* Create /web/src/app/shared/components/modal/modal.component.ts
  + Generic modal wrapper
  + Title, content, footer
  + Close button
  + Animation
* Create /web/src/app/shared/components/toast/toast.component.ts
  + Success, error, warning, info toasts
  + Auto-dismiss timer
  + Position (bottom-right)
  + Animation
* Create /web/src/app/shared/components/badge/badge.component.ts
  + Status badges
  + Color variants (green, red, amber, blue)
  + Small size
* Create /web/src/app/shared/components/loader/loader.component.ts
  + Loading skeleton
  + Spinner
  + Text loading animation
* Create /web/src/app/shared/components/empty-state/empty-state.component.ts
  + No data message
  + Optional action button
  + Icon
* Create /web/src/app/shared/components/form-group/form-group.component.ts
  + Label, input, error message
  + Validation feedback
* Create /web/src/app/shared/components/pagination/pagination.component.ts
  + Page numbers
  + Previous/Next buttons
  + Items per page selector

**F. Shared Layouts**

* Create /web/src/app/shared/layouts/main-layout/main-layout.component.ts
  + Header
  + Sidebar
  + Main content area
  + Footer
* Create /web/src/app/shared/layouts/auth-layout/auth-layout.component.ts
  + Simple layout for login page

**G. System Admin Pages**

* Create /web/src/app/pages/login/login.component.ts
  + Email input
  + Password input
  + Login button
  + Test credentials display
  + Error handling
  + Remember me (optional)
* Create /web/src/app/pages/dashboard/dashboard.component.ts
  + System Admin Dashboard (as spec)
  + 4 KPI cards
  + User growth chart
  + Recent activities log
  + System health metrics
  + Top tenants
  + Alerts section
* Create /web/src/app/pages/system-admin/tenants/tenants-list.component.ts
  + Table of tenants
  + Search, filter, sort
  + Create, edit, delete buttons
  + Status toggle
  + Pagination
* Create /web/src/app/pages/system-admin/tenants/tenant-create.component.ts
  + Form to create tenant
  + Fields: name, subdomain, plan, region, etc.
  + Validation
  + Submit handler
* Create /web/src/app/pages/system-admin/tenants/tenant-details.component.ts
  + Full tenant information
  + Overview, users, settings, billing, activity sections
* Create /web/src/app/pages/system-admin/system-users/users-list.component.ts
  + Table of system users
  + Search, filter, sort, pagination
  + Create, edit, delete buttons
* Create /web/src/app/pages/system-admin/system-users/user-create.component.ts
  + Form to create system user
  + Fields: email, name, role, status
* Create /web/src/app/pages/system-admin/system-users/user-details.component.ts
  + User profile, role, permissions
* Create /web/src/app/pages/system-admin/system-roles/roles-list.component.ts
  + Table of system roles
  + Create, edit, delete
* Create /web/src/app/pages/system-admin/system-roles/role-create.component.ts
  + Form to create role
* Create /web/src/app/pages/system-admin/system-roles/role-permissions.component.ts
  + Permission matrix (modules vs actions)
  + Checkbox toggling
* Create /web/src/app/pages/system-admin/billing/billing.component.ts
  + Subscriptions list
  + Invoices list
  + Payments list
* Create /web/src/app/pages/system-admin/audit/audit-logs.component.ts
  + System activity table
  + Filters, search, pagination
  + Export button
* Create /web/src/app/pages/system-admin/modules/modules-list.component.ts
  + Module registry table
  + Create, edit, enable/disable modules
* Create /web/src/app/pages/system-admin/settings/general-settings.component.ts
  + Platform settings form
* Create /web/src/app/pages/system-admin/settings/security-settings.component.ts
  + Security policies form
* Create /web/src/app/pages/profile/profile.component.ts
  + User profile page
  + Edit profile
  + Change password
  + Preferences
* Create /web/src/app/pages/not-found/not-found.component.ts
  + 404 page

**H. Styling & Themes**

* Create /web/src/styles/tailwind.css
* Create /web/src/styles/variables.scss
  + Color variables
  + Spacing variables
  + Typography variables
* Create /web/src/styles/themes.scss
  + Light theme variables
  + Dark theme variables
* Create /web/src/styles/animations.scss
  + Slide, fade, scale animations

**I. Angular Routing**

* Create /web/src/app/app.routes.ts
  + Public routes (login)
  + System admin routes
  + Tenant admin routes
  + Route guards
  + Lazy loading (optional)

**J. Angular Main Files**

* Create /web/src/main.ts
* Create /web/src/index.html
* Create /web/src/styles.css

**K. Angular Tests**

* Create /web/tests/auth.service.spec.ts
* Create /web/tests/permission.service.spec.ts
* Create /web/tests/components/header.component.spec.ts
* Setup test coverage

**L. Web Documentation**

* Create /web/README.md

**📱 PHASE 4: MOBILE (IONIC)**

**A. Ionic Project Setup**

* Create /mobile/package.json
* Create /mobile/ionic.config.json
* Create /mobile/capacitor.config.ts
* Create /mobile/jest.config.js
* Create /mobile/.env.example
* Create /mobile/Dockerfile

**B. Mobile Core Services**

* Create /mobile/src/app/core/services/auth.service.ts
* Create /mobile/src/app/core/services/user.service.ts
* Create /mobile/src/app/core/services/permission.service.ts
* Create /mobile/src/app/core/services/storage.service.ts
* Create /mobile/src/app/core/services/notification.service.ts

**C. Mobile Guards**

* Create /mobile/src/app/core/guards/auth.guard.ts

**D. Mobile Pages**

* Create /mobile/src/app/pages/login/login.page.ts
  + Email input
  + Password input
  + Biometric login toggle
  + Login button
* Create /mobile/src/app/pages/dashboard/dashboard.page.ts
  + User info card
  + Quick stats
  + Quick actions
  + Activity feed
* Create /mobile/src/app/pages/profile/profile.page.ts
  + User information
  + Edit profile
  + Settings
  + Logout button
* Create /mobile/src/app/pages/settings/settings.page.ts
  + Theme toggle
  + Language
  + Notifications
* Create /mobile/src/app/pages/notifications/notifications.page.ts
  + Notification list
* Create /mobile/src/app/pages/menu/menu.page.ts
  + Tab-based navigation

**E. Mobile Components**

* Create /mobile/src/app/components/header/header.component.ts
* Create /mobile/src/app/components/tab-bar/tab-bar.component.ts
* Create /mobile/src/app/components/card/card.component.ts

**F. Mobile Styling**

* Create /mobile/src/theme/variables.scss
* Create /mobile/src/theme/dark.scss

**G. Mobile Routing**

* Create /mobile/src/app/app.routes.ts

**H. Mobile Main Files**

* Create /mobile/src/main.ts
* Create /mobile/src/index.html
* Create /mobile/src/global.scss

**I. Mobile Documentation**

* Create /mobile/README.md

**🐳 PHASE 5: DEVOPS & INFRASTRUCTURE**

**A. Docker Setup**

* Create /api/Dockerfile
  + Node.js base image
  + Install dependencies
  + Copy source code
  + Expose port 3000
  + Start command
* Create /web/Dockerfile
  + Node.js build stage
  + Nginx runtime stage
  + Copy built app to Nginx
  + Expose port 4200
* Create /mobile/Dockerfile (optional)
* Create /devops/.dockerignore

**B. Docker Compose**

* Create /docker-compose.yml
  + PostgreSQL service
  + API service (Express)
  + Web service (Angular)
  + Nginx (optional)
  + Redis (optional)
  + Environment variables
  + Volume mappings
  + Health checks
  + Network configuration
* Create /docker-compose.prod.yml
  + Production-specific configs
  + Replicas/scaling
  + Resource limits
* Create /devops/postgres/init.sql
  + PostgreSQL initialization script

**C. Nginx Configuration**

* Create /devops/nginx.conf
  + Reverse proxy setup
  + SSL/TLS (optional)
  + Gzip compression
  + Caching rules

**🔄 PHASE 6: CI/CD PIPELINES (GitHub Actions)**

**A. API Testing Workflow**

* Create /.github/workflows/api-test.yml
  + Trigger: Push to main/develop
  + Node.js setup
  + Install dependencies
  + Run linter
  + Run tests
  + Generate coverage report
  + Upload coverage to Code Coverage service

**B. Web Testing Workflow**

* Create /.github/workflows/web-test.yml
  + Trigger: Push to main/develop
  + Node.js setup
  + Install dependencies
  + Run linter
  + Run tests
  + Build Angular app
  + Upload artifacts

**C. Mobile Testing Workflow**

* Create /.github/workflows/mobile-test.yml
  + Trigger: Push to main/develop
  + Node.js setup
  + Install dependencies
  + Run linter
  + Run tests

**D. API Build Workflow**

* Create /.github/workflows/api-build.yml
  + Trigger: Push to main
  + Build Docker image
  + Push to Docker Hub/Container Registry

**E. Web Build Workflow**

* Create /.github/workflows/web-build.yml
  + Trigger: Push to main
  + Build Docker image
  + Push to Docker Hub/Container Registry

**F. Deployment Workflow**

* Create /.github/workflows/deploy.yml
  + Trigger: Release created
  + Pull latest images
  + Run database migrations
  + Start services
  + Run health checks
  + Notify on success/failure

**🧪 PHASE 7: TESTING & QUALITY**

**A. API Tests**

* Create /api/tests/auth.test.js
* Create /api/tests/rbac.test.js
* Create /api/tests/user.test.js
* Setup test database
* Setup test fixtures/factories

**B. Web Tests**

* Create component tests
* Create service tests
* Create guard tests
* Setup testing module

**C. Code Quality**

* Setup ESLint for all projects
* Setup Prettier for code formatting
* Setup pre-commit hooks
* Setup GitHub branch protection

**📚 PHASE 8: DOCUMENTATION & GUIDES**

**A. API Documentation**

* Document all endpoints
* Document request/response formats
* Create Postman collection (optional)
* Document authentication flow
* Document error codes

**B. Frontend Documentation**

* Document component structure
* Document service methods
* Document routes & guards
* Create component usage examples

**C. Database Documentation**

* Document all tables
* Document relationships
* Document indexes
* Create ER diagram

**D. Deployment Documentation**

* Local development setup
* Docker Compose setup
* Production deployment
* Environment configuration
* Backup & recovery

**E. Security Documentation**

* Authentication & authorization
* Data encryption
* Compliance requirements
* Security best practices
* Incident response

**🚀 PHASE 9: SEED DATA & TEST USERS**

**A. Test Credentials**

* System Super Admin: admin@template.local / admin123
* System Admin: sys-admin@template.local / sysadmin123
* Support Staff: support@template.local / support123
* Tenant Admin: tenant-admin@template.local / tenant123
* Loan Officer: officer@template.local / officer123
* Cashier: cashier@template.local / cashier123
* Viewer: viewer@template.local / viewer123

**B. Sample Data**

* Sample tenants (Demo Pawnshop, Test Company)
* Sample users (various roles)
* Sample system roles
* Sample tenant roles
* Sample modules/menus
* Sample permissions
* Sample loans (if for pawnshop context)
* Sample customers (if for pawnshop context)

**✅ FINAL CHECKLIST**

**A. Pre-Launch Verification**

* All tests passing (API, Web, Mobile)
* Code coverage > 80%
* No security vulnerabilities (npm audit)
* Documentation complete
* Docker builds successfully
* Docker Compose runs without errors
* All environment variables documented
* Seed data populates successfully

**B. Repository & CI/CD**

* GitHub Actions workflows configured
* Branch protection rules enabled
* CODEOWNERS file created (optional)
* Contributing guidelines added
* README badges added (build status, coverage, etc.)

**C. Final Testing**

* Manual login test (all user types)
* Sidebar menu renders correctly (all roles)
* Permissions enforced (UI and API)
* Toasts display correctly
* Confirmations work
* Dark/Light theme toggle works
* Responsive design (mobile, tablet, desktop)
* Performance benchmarks met

**D. Repository Ready**

* README.md complete
* SETUP.md complete & tested
* All documentation done
* License included
* .gitignore configured
* .env.example created
* Changelog started (if needed)

**📊 SUMMARY**

| **Phase** | **Tasks** | **Status** |
| --- | --- | --- |
| 1. Foundation | 10 | 🔴 Not Started |
| 2. Backend | 40 | 🔴 Not Started |
| 3. Frontend (Web) | 80 | 🔴 Not Started |
| 4. Mobile | 35 | 🔴 Not Started |
| 5. DevOps | 15 | 🔴 Not Started |
| 6. CI/CD | 6 | 🔴 Not Started |
| 7. Testing | 8 | 🔴 Not Started |
| 8. Documentation | 15 | 🔴 Not Started |
| 9. Seed Data | 12 | 🔴 Not Started |
| 10. Final | 13 | 🔴 Not Started |
| **TOTAL** | **234 Tasks** | 🔴 **0% Complete** |

**🎯 NEXT STEPS**

**Priority Order:**

1. ✅ **Phase 1: Foundation** (Repository setup)
2. ✅ **Phase 2: Backend** (Core API)
3. ✅ **Phase 3: Frontend** (Web admin)
4. ✅ **Phase 4: Mobile** (Ionic app)
5. ✅ **Phase 5: DevOps** (Docker setup)
6. ✅ **Phase 6: CI/CD** (GitHub Actions)
7. ✅ **Phase 7-10:** Testing, Docs, Seed Data, Final Verification

**🚀 READY TO START?**

**Total Estimated Time:**

* Backend: 6-8 hours
* Frontend: 10-12 hours
* Mobile: 4-6 hours
* DevOps: 2-3 hours
* Testing & Docs: 4-5 hours
* **Total: 26-34 hours** (for one developer)