

# Multi-Tenant Payroll System – User Stories

## Epic 1: Tenant Management & Setup

### US-101: Tenant Registration and Onboarding

**As a** company administrator

**I want to** register my organization as a new tenant

**So that** I can set up and manage payroll for my employees

#### Acceptance Criteria:

- Admin can create a new tenant with company name, email, phone, and industry
- System assigns unique tenant\_id for data isolation
- Initial admin user is auto-created with credentials
- Tenant status defaults to "active"
- Confirmation email sent to tenant contact
- Tenant data is stored with created\_at timestamp

#### Definition of Done:

- Tenants table properly stores multi-tenant data
- Email confirmation implemented
- Audit trail recorded

---

### US-102: Tenant Settings Management

**As a** tenant admin

**I want to** update tenant settings (name, contact info, industry)

**So that** my organization information remains current

#### Acceptance Criteria:

- Admin can edit tenant name, email, phone, industry
- Changes are logged with updated\_at timestamp
- Tenant status can be toggled between active/inactive
- Only tenant admins can modify settings
- Inactive tenants cannot process payroll

---

## Epic 2: User Management & Authentication

### US-201: User Registration and Role Assignment

**As a** tenant admin

**I want to** invite users and assign roles (admin, HR, payroll\_officer, employee)

**So that** I can delegate payroll responsibilities

#### Acceptance Criteria:

- Admin can create users with email, password, first\_name, last\_name
  - Roles: admin, hr\_officer, payroll\_officer, employee
  - Unique email per tenant (users can have same email across tenants)
  - Initial password sent securely
  - User account starts as inactive until first login
  - tenant\_id linked to each user
- 

## US-202: User Authentication (Login/Logout)

**As a** user

**I want to** log in with email and password

**So that** I can access the payroll system securely

**Acceptance Criteria:**

- Users authenticate with email and password
  - Password stored as hash (bcrypt or similar)
  - last\_login timestamp updated on successful login
  - Session token (JWT recommended) issued for 24-48 hours
  - Users can log out and invalidate session
  - Failed login attempts tracked (optional rate limiting)
- 

## US-203: Role-Based Access Control

**As a** system

**I want to** enforce role-based permissions

**So that** users only access features appropriate to their role

**Acceptance Criteria:**

- Admin: Full system access, user management, payroll approval
  - HR Officer: Employee data management, attendance tracking
  - Payroll Officer: Salary structure, payroll runs, payment processing
  - Employee: View own payroll details, download pay stubs
  - Unauthorized role access returns 403 Forbidden
- 

# Epic 3: Organization Structure

## US-301: Department Management

**As a** HR officer

**I want to** create and manage departments

**So that** I can organize employees by business unit

**Acceptance Criteria:**

- HR can create departments with name, description
- Department manager can be assigned (user\_id or employee\_id)
- Departments linked to tenant\_id
- Department can be marked active/inactive
- Delete prevented if employees assigned

- List departments with filter by status
- 

## US-302: Position Management

**As a** HR officer

**I want to** create job positions with salary bands

**So that** I can standardize roles across the organization

**Acceptance Criteria:**

- Create positions with title, description, level (junior/mid/senior)
  - Positions linked to tenant\_id
  - Position can be assigned to multiple employees
  - Position can be marked active/inactive
  - List positions sorted by level
- 

## US-303: Employee Master Data Management

**As a** HR officer

**I want to** add and maintain employee records

**So that** I have accurate personnel data for payroll processing

**Acceptance Criteria:**

- Create employee with personal info (name, email, phone, DOB, gender)
  - Auto-generate employee\_code (unique per tenant)
  - Assign to department and position
  - Link to user account (optional, for employee portal access)
  - Set joining\_date and employment\_type (full-time, part-time, contract)
  - Track employee status (active, on\_leave, terminated)
  - Prevent deletion if payroll records exist
  - Edit timestamps (created\_at, updated\_at) recorded
- 

# Epic 4: Salary Configuration

## US-401: Salary Structure Setup

**As a** payroll officer

**I want to** create salary structures with base salary

**So that** I can standardize compensation

**Acceptance Criteria:**

- Create salary structure with name, base\_salary, effective\_date
  - Mark as active/inactive
  - Multiple structures per tenant allowed
  - Only active structures can be assigned
  - Edit history tracked via timestamps
-

## US-402: Allowance Configuration

**As a payroll officer**

**I want to** define allowances (HRA, dearness, shift allowance, etc.)

**So that** I can include them in payroll calculations

### Acceptance Criteria:

- Create allowances with name, type (fixed/variable), amount, is\_taxable flag
  - Allowances tenant-specific
  - Link multiple allowances to salary structures (join table: payroll\_detail\_allowances)
  - Amount can be decimal (supports cents/paisa)
  - Mark as active/inactive
- 

## US-403: Deduction Configuration

**As a payroll officer**

**I want to** define deductions (tax, insurance, loan recovery)

**So that** I can calculate net salary accurately

### Acceptance Criteria:

- Create deductions with name, type (tax/insurance/loan/other), amount, is\_taxable flag
  - Deductions tenant-specific
  - Link multiple deductions to salary structures
  - Amount can be decimal
  - Mark as active/inactive
  - Tax deductions properly flagged for tax calculation
- 

## US-404: Employee Salary Assignment

**As a payroll officer**

**I want to** assign salary structures to individual employees

**So that** I can customize compensation per employee

### Acceptance Criteria:

- Assign salary structure to employee with effective\_date
  - Support salary changes with end\_date on old assignment and new start\_date
  - Track assigned\_by (user who made assignment)
  - Only active structures can be assigned
  - Query current salary for employee on given date
  - Salary history maintained for audit trail
- 

## Epic 5: Payroll Processing

## US-501: Create Payroll Run

**As a** payroll officer

**I want to** create a payroll run for a period (monthly/bi-weekly)

**So that** I can process bulk payroll for all employees

### Acceptance Criteria:

- Create payroll run with name, period\_start\_date, period\_end\_date, payment\_date
  - Status defaults to "draft"
  - Tenant-specific payroll run
  - Include all active employees in run by default (selectable)
  - Calculate totals: total\_salary, total\_allowances, total\_deductions
  - Timestamp recorded for audit
- 

## US-502: Calculate Payroll Details

**As a** system

**I want to** automatically calculate payroll for each employee

**So that** I reduce manual errors and save time

### Acceptance Criteria:

- For each employee in payroll run:
    - Fetch current salary structure
    - Fetch assigned allowances and deductions
    - Calculate: gross\_salary = base\_salary + allowances
    - Calculate: net\_salary = gross\_salary - deductions - tax
    - Create payroll\_detail record
    - Create payroll\_detail\_allowances and payroll\_detail\_deductions join records
  - Tax calculation invokes tax\_calculations entity
  - Handle exceptions (no salary structure assigned)
  - Store base\_salary, gross\_salary, net\_salary as decimal
- 

## US-503: Overtime Management

**As a** HR officer

**I want to** record overtime for employees

**So that** I can pay overtime compensation in payroll

### Acceptance Criteria:

- Add overtime record with employee, overtime\_date, hours\_worked, rate\_multiplier
  - rate\_multiplier: 1.5x for regular overtime, 2x for holidays (configurable)
  - Calculate amount = hours\_worked × hourly\_rate × rate\_multiplier
  - Overtime marked pending until approved
  - Payroll officer approves overtime (approved\_by field)
  - Approved overtime included in payroll calculation
  - Status: pending → approved → included\_in\_payroll
-

## US-504: Attendance Tracking

**As a** HR officer

**I want to** record daily attendance (check-in/check-out times)

**So that** I can track employee availability for payroll

### Acceptance Criteria:

- Record attendance with check\_in\_time, check\_out\_time
- Auto-calculate work\_hours from timestamps
- Status: present, absent, half\_day, leave
- Remarks field for notes (WFH, medical leave, etc.)
- Query attendance by employee/date range
- Monthly attendance summary available for payroll review

---

## US-505: Tax Calculation & Management

**As a** system

**I want to** calculate taxes based on income and deductions

**So that** tax withholding is accurate

### Acceptance Criteria:

- For each payroll detail, calculate tax based on:
  - taxable\_income (gross\_salary minus tax-exempt allowances)
  - tax\_rate (configurable per tenant, could default to national rates)
- Tax types: income\_tax, professional\_tax, etc.
- Store tax\_calculations with tax\_type, taxable\_income, tax\_rate, tax\_amount
- Link to payroll\_detail via foreign key
- Support tax adjustments/rebates with remarks

---

## US-506: Review & Approve Payroll

**As a** tenant admin

**I want to** review calculated payroll before processing

**So that** I can ensure accuracy and prevent errors

### Acceptance Criteria:

- View payroll run summary (total salary, allowances, deductions, tax)
- View per-employee payroll details with breakdown
- Compare with previous period (if available)
- Flag anomalies (zero salary, excessive deductions, etc.)
- Approve entire payroll run (status: draft → approved)
- Reject with comments (status: draft → rejected)
- Only admins/payroll officers can approve

# Epic 6: Payment Processing

## US-601: Payment Method Management

**As a** HR officer

**I want to** manage employee payment methods (bank account, cash, cheque)

**So that** I can transfer salaries correctly

**Acceptance Criteria:**

- Create payment methods: bank\_transfer, cash, cheque
- Payment methods tenant-specific, reusable across employees
- Mark as active/inactive

---

## US-602: Employee Payment Method Assignment

**As a** employee

**I want to** register my bank account for salary deposit

**So that** I receive payments directly to my account

**Acceptance Criteria:**

- Employee/HR adds payment method with:
  - account\_holder\_name, account\_number, bank\_name, routing\_number
- Mark one payment method as is\_default (default to true for first)
- Multiple payment methods per employee (for salary split)
- Edit/delete payment methods
- Validation: account number format, routing number (if applicable)

---

## US-603: Process Payments

**As a** payroll officer

**I want to** initiate salary payments to employees

**So that** employees receive their compensation on time

**Acceptance Criteria:**

- Select approved payroll run
- Generate payment instructions (bank transfers via API/file export)
- Record payment with:
  - payroll\_detail\_id, employee\_payment\_method\_id, amount
  - payment\_date, transaction\_reference
  - Status: pending → processing → completed → failed
- Link payment to payroll\_detail
- Support bulk payment export (CSV/XML for bank integration)
- Audit trail: who initiated, when, status changes

## US-604: Payment Reconciliation

**As a** finance officer

**I want to** reconcile bank payments with payroll records

**So that** I can identify failed or pending transfers

### Acceptance Criteria:

- Query payments by status (pending, completed, failed)
  - Mark payment as completed when confirmed by bank
  - Update status if payment fails (insufficient funds, invalid account)
  - Generate reconciliation report (expected vs. actual)
  - Flagged mismatches for investigation
- 

## Epic 7: Reporting & Analytics

### US-701: Payroll Summary Report

**As a** tenant admin

**I want to** view payroll summary by period

**So that** I can track total compensation costs

### Acceptance Criteria:

- Report shows: total employees, total salary, total allowances, total deductions, total tax
  - Group by department (optional)
  - Filter by payroll run, date range, employee status
  - Export to PDF/Excel
  - Timestamp of report generation
- 

### US-702: Employee Pay Stub

**As a** employee

**I want to** download my pay stub

**So that** I can verify my compensation

### Acceptance Criteria:

- Employee views/downloads pay stub for each payroll period
  - Pay stub includes:
    - Employee details (name, ID, period)
    - Base salary, allowances (itemized), deductions (itemized), tax, net salary
    - YTD totals (year-to-date)
    - Payment method used, payment date
  - Download as PDF
  - Secure access (only employee's own pay stubs)
-

## US-703: Attendance Report

**As a** HR officer

**I want to** generate monthly attendance reports

**So that** I can track employee attendance patterns

**Acceptance Criteria:**

- Report by employee: total present, absent, half\_day, leave days
  - Report by department: aggregate attendance metrics
  - Filter by month, year, employee, status
  - Export to CSV/Excel
  - Highlight absenteeism issues
- 

## Epic 8: System & Audit

### US-801: Multi-Tenant Data Isolation

**As a** system

**I want to** ensure tenant data is completely isolated

**So that** one tenant cannot access another's data

**Acceptance Criteria:**

- Every query filters by tenant\_id
  - No cross-tenant data visible
  - Foreign keys enforced (referential integrity)
  - Soft delete supported (mark as inactive instead of delete where applicable)
  - Audit logs include tenant\_id
- 

### US-802: Audit Trail & Logging

**As a** compliance officer

**I want to** track all changes to critical data

**So that** I can audit payroll for compliance

**Acceptance Criteria:**

- Log user actions: create, update, delete on sensitive tables
    - Employees, salary structures, payroll runs, payments, users
  - Record: user\_id, action, table, old\_value, new\_value, timestamp, tenant\_id
  - Log is immutable (append-only)
  - Query audit logs by user, table, date range, tenant
  - Retention: minimum 7 years (configurable)
- 

### US-803: Error Handling & Validation

**As a** system

**I want to** validate all inputs and handle errors gracefully

**So that** data integrity is maintained

**Acceptance Criteria:**

- Decimal fields (salary, amount) validated for precision (2 decimal places)
  - Date fields validated (no future dates where inappropriate)
  - Email validation for unique constraint
  - Foreign key validation (no orphaned records)
  - HTTP status codes: 400 (bad request), 403 (forbidden), 404 (not found), 500 (server error)
  - User-friendly error messages
- 

## US-804: Performance Optimization

**As a** system

**I want to** respond quickly to large payroll runs

**So that** payroll processing completes in reasonable time

**Acceptance Criteria:**

- Index on tenant\_id + id (primary lookup)
  - Index on foreign keys (payroll\_run\_id, employee\_id, etc.)
  - Batch processing for large payroll runs (1000+ employees)
  - Query response time < 2 seconds for 10k employee database
  - Pagination for list endpoints (limit 100 per page)
- 

## Epic 9: Integration & Advanced Features (Future)

### US-901: Biometric Attendance Integration

**As a** HR officer

**I want to** sync attendance from biometric devices

**So that** I have automated, accurate attendance records

**Acceptance Criteria:**

- API endpoint to receive attendance data from biometric systems
  - Parse check-in/check-out times
  - Validate against employee records
  - Auto-create attendance records
- 

### US-902: Bank API Integration

**As a** payroll officer

**I want to** initiate salary transfers via bank API

**So that** payments are automated and reliable

**Acceptance Criteria:**

- Connect to bank API (SWIFT, ACH, local payment gateway)
  - Generate payment file in required format
  - Auto-confirm transfers, update status
  - Handle payment failures and retry logic
-

## US-903: Email Notifications

**As a system**

**I want to** send email notifications for key events

**So that** users stay informed

### Acceptance Criteria:

- Send email when payroll run approved
  - Send pay stub to employee after payroll completion
  - Send alerts for payment failures
  - Customizable email templates per tenant
- 

### Notes

- All datetime fields use ISO 8601 format
- Decimal amounts support 2 decimal places (currency cents/paisa)
- Timezone support: store UTC, display in user's local timezone
- Password reset flow: send reset link via email, expire after 24 hours
- Consider implementing approval workflows for large changes (salary increase, termination)
- Compliance requirements may vary by country (India, Philippines, etc.) – design should support regulatory flexibility