**Salary compensation part brief explanation.**

**1) Core concepts & data**

**a) Designations & salary bands**

Each designation defines the **default basic salary** (editable per employee, but you always have a default):

* HR: 90,000 LKR
* Cleaning Staff: 45,000 LKR
* …etc.

**Tables**

* designations(id, name, default\_basic, epf\_employee\_pct, epf\_employer\_pct, etf\_employer\_pct)
* employees(..., designation\_id, basic\_salary) — if basic\_salary is null, use designation.default\_basic.

**b) Leave types & policies**

We treat leave as **types**, each with behavior:

| **Field** | **Meaning** |
| --- | --- |
| code | e.g., MEDICAL, COMPASSIONATE\_PARENT, ANNUAL, UNPAID |
| name | Human label |
| paid | true means **no deduction** from salary |
| default\_quota\_days | Typical allowance per year/month |
| max\_consecutive\_days | Optional limit |
| auto\_quota\_renewal | ANNUAL or MONTHLY |

**Examples**

* **Medical**: paid=true, default\_quota\_days=14/year
* **Compassionate (parent funeral)**: paid=true, max\_consecutive\_days=7
* **Annual**: paid=true, quota (e.g., 14)
* **Unpaid**: paid=false (deduct from salary via proration)

**Tables**

* leave\_types(id, code, name, paid, default\_quota\_days, max\_consecutive\_days, auto\_quota\_renewal)
* employee\_leaves(id, employee\_id, date\_from, date\_to, type\_id, approved\_by, notes)  
  During payroll, you aggregate monthly usage per type.

**Payroll impact**

* Sum approved leaves in month by type.
* **Paid types** (medical, compassionate, annual): **do not reduce** salary.
* **Unpaid**: reduce by proration (basic \* unpaid\_days / working\_days).

**c) Attendance monthly**

You can still keep a summarized row per employee/month:

* attendance\_monthly(employee\_id, year, month, days\_worked, paid\_leave\_days, unpaid\_leave\_days, overtime\_hours)

Paid/unpaid can be calculated from employee\_leaves + raw attendance; or you write them directly here.

**d) Reimbursements (transport, internet) & Bonuses**

These are **add-ons** for a month. They are not part of recurring “allowances” rules (unless you want them recurring). Two ways:

1. **Recurring** → keep as employee\_adjustments(kind='allowance') (basis flat).
2. **One-off monthly** → employee\_reimbursements(employee\_id, year, month, category, amount, note)  
   Categories: TRANSPORT, INTERNET, etc.

**Bonuses**

* employee\_bonuses(employee\_id, year, month, reason, amount)
* Bulk assign via UI (multi-select employees) writes many rows at once.

**Payroll impact**

* Reimbursements and bonuses are **added** after attendance proration and before tax/deductions (or after—up to your policy; many orgs add before tax if taxable, after if fully non-taxable). Add a taxable flag to each category if needed.

**e) Deductions (statutory & others)**

Keep your comp\_rules + employee\_adjustments model for **recurring** allowances/deductions.

* EPF employee % (deduct)
* Income tax (percentage or progressive)
* Loans/Insurance (flat/month)

**f) Calculation order (per employee per month)**

1. Determine **basic** (employee.basic or designation.default\_basic).
2. Determine **working days** (e.g., 22).
3. Get **unpaid leave days** (from employee\_leaves or attendance\_monthly).
4. **Unpaid deduction** = basic \* (unpaid\_days / working\_days).
5. Compute **overtime** (optional) and **paid leaves** (0 deduction).
6. Base gross = basic - unpaid\_deduction + overtime.
7. Add **recurring allowances** + **monthly reimbursements** + **bonuses**.
8. Gross = base gross + add-ons.
9. Compute **deductions** (EPF, tax, loans…).
10. **Net pay** = Gross – Total deductions.
11. Write payroll\_items breakdown JSON for transparency.

The compassionate/medical leave logic is covered because they’re marked paid=true, hence they **do not** contribute to unpaid days.

**2) API endpoints (minimal)**

* GET /api/payroll/run/:year/:month → fetch run & items
* POST /api/payroll/run { year, month } → recompute
* POST /api/reimbursements/bulk  
  { year, month, items:[{employee\_id, category, amount, note, taxable}] }
* POST /api/bonuses/bulk  
  { year, month, reason, items:[{employee\_id, amount}] }
* POST /api/attendance/monthly → upsert summary rows
* POST /api/leaves → create leave requests/records (HR approval flow optional)
* GET /api/config/leave-types / POST /api/config/leave-types → HR manages leave policies
* GET /api/config/designations / POST /api/config/designations → manage salary bands
* POST /api/comp/adjustments → recurring emp allowances/deductions

Your current payroll.service.js just needs two small additions:

* When computing attendance, **derive unpaid vs paid** from leave records.
* Sum **reimbursements** and **bonuses** for the year/month and add to allowances before deductions.

**3) How monthly pay is “automatic”**

1. HR maintains **designations** (defaults) and per-employee **basic salary overrides** if needed.
2. HR (or managers) record **attendance** and **approved leave** (medical/compassionate unpaid/paid).
3. HR bulk-uploads **reimbursements** (transport/internet) and **bonuses** for the month using the bulk UI.
4. HR presses **Recalculate** → backend computes run and stores results.
5. HR reviews **Net salary summary** and **payslips**; optionally “POST” the run.

**B) Front-end structure (pages & components)**

Use your existing Salary & Compensation header. We’ll add the three new tools and bulk UI.

**Routes**

/earnings

/deductions

/allowances

/overtime-adjustments

/compensation-adjustment

/net-salary-summary

/bulk-actions <-- NEW (bonuses & reimbursements in one place)

/policies <-- NEW (leave types & reimbursement categories)

/designations <-- NEW (salary bands)

**Tiny backend changes (conceptually)**

* **Leave aggregation** inside payroll compute:
  + Query employee\_leaves for the month; group by type.
  + paid=true types → contribute to paid\_leave\_days.
  + paid=false types → contribute to unpaid\_leave\_days.
* **Reimbursements / Bonuses** inside payroll compute:
  + Sum employee\_reimbursements for the month → add to allowances (honor taxable flag if you want to tax them).
  + Sum employee\_bonuses for the month → add to allowances (usually taxable).

These plug into your existing payroll.service.js before deductions.

**UX for HR (scenario)**

1. HR opens **Policies**, creates:
   * LEAVE: MEDICAL (paid, 14d/year), COMP\_PARENT (paid, 7d max), UNPAID (unpaid)
   * REIMB: TRANSPORT (taxable), INTERNET (non-taxable, if you want)
2. HR opens **Designations**, sets:
   * HR→ 90,000; Cleaner→ 45,000; …
3. Managers approve **leaves** during the month (Employee → Medical 2d; Employee → COMP\_PARENT 3d). Both are **paid**.
4. HR uploads **attendance**, logs **overtime** where needed.
5. HR goes to **Bulk actions** → tab **Reimbursements**:
   * Select a group, Category TRANSPORT, Month 2025-08, Amount 4500 → Apply.
6. HR goes to **Bulk actions** → tab **Bonuses**:
   * Select top performers, Reason “Quarterly bonus”, Month 2025-08, Amount per person → Apply.
7. HR opens **Net salary summary** for 2025-08 → clicks **Recalculate** → reviews totals and downloads CSV.