Background Context

In a company, there is data that contains employees' salaries. They're paid with the same rule, which is monthly-based, with regular 8 working hours per day (9AM-5PM), 5 days a week (monday-friday). Their take-home pay will be prorated based on their attendance. Along with that, they can also propose overtime, which is paid at twice the prorated salary for hours taken. They can also submit reimbursement requests which will be included in the payslip.

Objective

Create a scalable payslip generation system that can handle predefined rules for employee attendance, overtime, and reimbursement.

Requirements

- Create 100 fake employees in the database, each prefilled with various salary, username, and password.
- Create 1 fake admin in the database, prefilled with username and password.
- Create an endpoint where admin can add attendance period start & end date for particular payroll.
- Create an endpoint where employees can submit their own attendance.
    - No rules for late or early check-ins or check-outs; check-in at any time that day counts.
    - Submissions on the same day should count as one.
    - Users cannot submit on weekends.
- Create an endpoint where employees can submit overtime.
    - Overtime must be proposed after they are done working.
    - They can submit the number of hours taken for that overtime.
    - Overtime cannot be more than 3 hours per day.
    - Overtime can be taken any day.
- Create an endpoint where employees can submit reimbursements.
    - Employees can attach the amount of money that needs to be reimbursed.
    - Employees can attach a description to that reimbursement.
- Create an endpoint where admin can run payroll (process payments to employees).
    - Once payroll is run, attendance, overtime, and reimbursement records from that period cannot affect the payslip.
    - Payroll for each attendance period can only be run once.
- Create an endpoint where employees can generate a payslip.
    - Payslip contains a breakdown of their attendance and how it affects the salary.
    - Payslip contains a breakdown of their overtime and how much it is multiplied by the salary.
    - Payslip contains a list of reimbursements.
    - Payslip contains the total take-home pay, which is an accumulation of all components.
- Create an endpoint where admin can generate a summary of all employee payslips.
    - The summary contains take-home pay of each employee.
    - The summary contains the total take-home pay of all employees.

Technical Requirements

- Use PostgreSQL as the main database.
- Handle API requests via HTTP using JSON as the data format.
- Implement automated testing (unit tests, integration tests, or both) for all functionality.
- Provide clear documentation covering how-to guides, API usage, and software architecture.
- Submit the code to a public GitHub repository.

Plus Points

- Measure performance scalability of each functionality.
- Ensure that every record is tracable.
    - Each record should include created_at and updated_at timestamps.
    - Track the user who performed each action (created_by, updated_by).
    - Store the IP address of requests for audit purposes.
    - Maintain an audit log table to track significant changes to records.
    - Include request_id in logs for request tracing across services.