MOHAMED AATHIL

30415

EMPLOYEE LEAVE MANAGEMENT SYSTEM

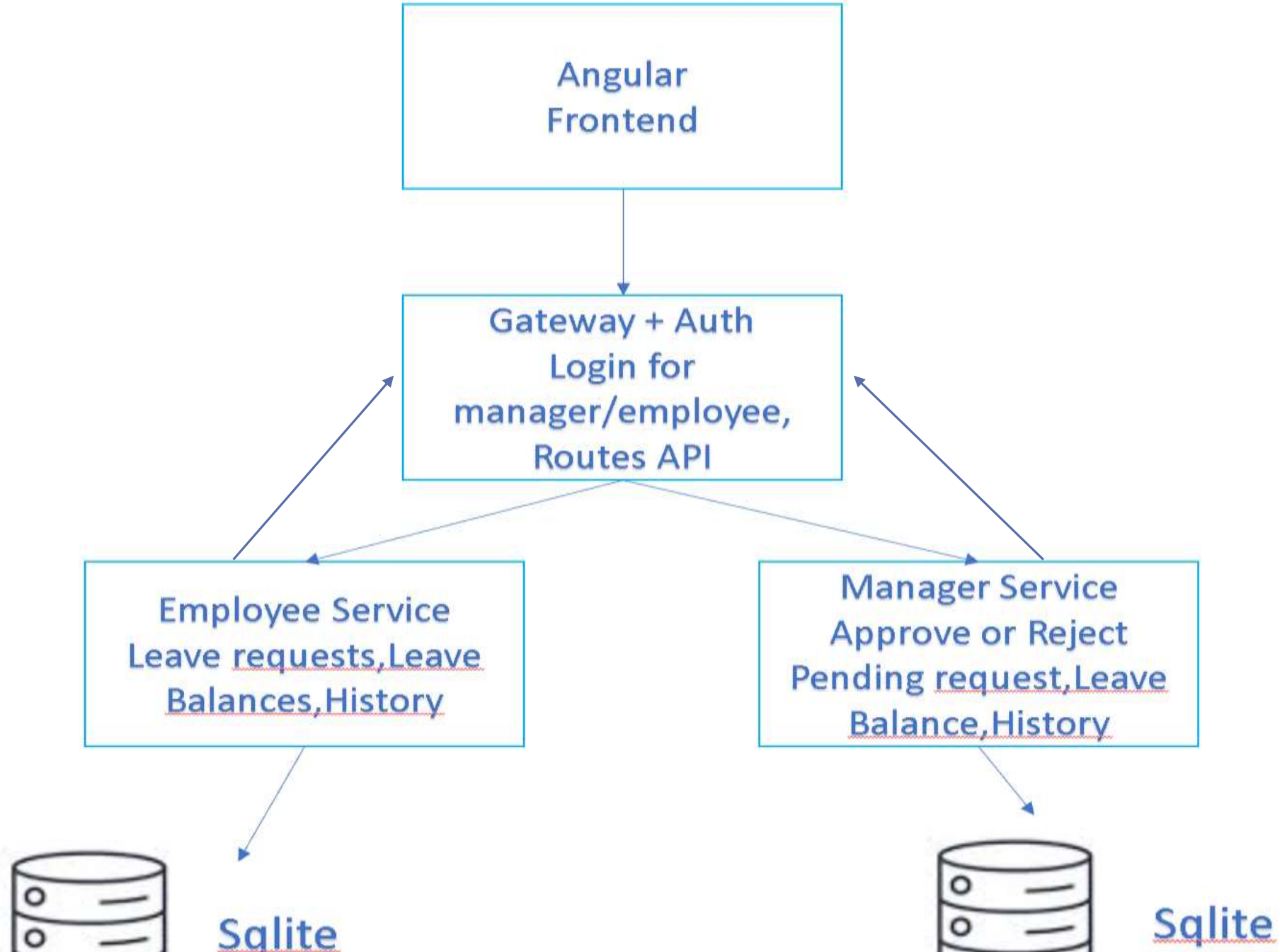
- This employee leave management system provides a platform where employees can submit leave requests, and managers can review and approve or reject those requests efficiently.
- Employee can choose dates , types of leave and can view remaining leaves and history.
- Manager can view the pending request and can either approve or reject with remarks and can see all his employees history and leave balances

Architecture used: Microservices

Services used : Employee , Manager , Gateway

Each service have their own database (Sqlite)

Services communicate over API's through gateway.


Angular
Frontend
Gateway + Auth
Login for
manager/employee,
Routes API
Employee Service
Leave requests,Leave
Balances,History
Manager Service
Approve or Reject
Pending request,Leave
Balance,History
Sqlite
Sqlite

Services & API

Employee Service : Leave Request , Leave Balances, History

POST - /apply_leave → Employee applies for leave (pending by default)

GET - /my_leaves/<id> → Get all leave requests of a specific employee

DELETE - /delete_leave/<id> → Delete leave request (only if status is pending)

Services & API

Manager Service : Approve or Reject Leaves, Leave all employees leave balances, History

GET - /all_leaves → Get all leave requests (with employee details)

GET - /leave_requests → Get only pending leave requests (for approval)

PUT - /update_leave/<id> → Update leave status (approve/reject/pending) + adjust balances

Containerization

Four Containers :

lms-employee

lms-manager

lms-gateway

lms-frontend

volumes:

lms_employee_data

lms_manager_data

Deployment (Docker)

Create Docker images and launch the complete stack with Docker Compose in detached mode

Check service health status, then access the frontend through your browser

Log in with the default admin account to begin setting up employees, managers, and leave details

Data remains safe between restarts because it is stored in persistent volumes