Project: PerformanceTrack – Internal Employee Performance Review & Goal Management System

1. Introduction

The purpose of this document is to provide a detailed Low-Level Design (LLD) for **PerformanceTrack**, a web-based system designed to manage employee performance reviews, goal setting, and appraisal workflows within an organization. The system enables employees to set goals, managers to review performance, and admins to generate appraisal reports—all **without integrating with any external HR or payroll systems**.

It uses a **REST API-based backend architecture** and **Angular/React** for the frontend. Compatible with **Java (Spring Boot)** and **.NET (ASP.NET Core)** frameworks.

Actors / Users

- **Employee:** Sets goals, submits self-assessments, and views performance ratings.
- **Manager:** Reviews employee performance, provides feedback, and approves ratings.
- **Admin:** Configures review cycles, manages users, and generates appraisal reports.

2. Module Overview

- **2.1 User Management**
  Handles registration, authentication, and role-based access.
- **2.2 Goal Setting & Tracking**
  Allows employees to define goals and track progress.
- **2.3 Performance Review Workflow**
  Supports multi-step review and approval process.
- **2.4 Feedback & Rating**
  Captures feedback and assigns performance ratings.
- **2.5 Analytics & Reporting**
  Generates dashboards for performance trends and appraisal metrics.
- **2.6 Notifications & In-App Alerts**
  Provides in-app alerts for review deadlines and feedback updates (no external email/SMS).

3. Architecture Overview

- **Frontend:** Angular or React
- **Backend:** REST API-based architecture

- **Database:** Relational Database (MySQL/PostgreSQL/SQL Server)

4. Module-Wise Design

4.1 User Management Module

Features:

- Register as employee, manager, or admin
- Login and manage profiles

Entities:

- **User**
    - UserID
    - Name
    - Role (Employee/Manager/Admin)
    - Email
    - Department
    - Status (Active/Inactive)

4.2 Goal Setting & Tracking Module

Features:

- Define goals with measurable targets
- Track progress and completion

Entities:

- **Goal**
    - GoalID
    - Title
    - Description
    - AssignedToUserID
    - StartDate
    - EndDate
    - Status (Pending/InProgress/Completed)

4.3 Performance Review Workflow Module

Features:

- Initiate review cycles

- Capture self-assessment and manager feedback

Entities:

- **ReviewCycle**
    - CycleID
    - Title
    - StartDate
    - EndDate
    - Status (Active/Closed)
- **PerformanceReview**
    - ReviewID
    - CycleID
    - UserID
    - SelfAssessment
    - ManagerFeedback
    - Rating
    - ReviewDate

4.4 Feedback & Rating Module

Features:

- Provide qualitative feedback
- Assign numeric ratings

Entities:

- **Feedback**
    - FeedbackID
    - ReviewID
    - GivenByUserID
    - Comments
    - Date

4.5 Analytics & Reporting Module

Features:

- Performance trends by department
- Goal achievement metrics

Entities:

- **Report**
    - ReportID
    - Scope (Department/User/Period)
    - Metrics (AverageRating, GoalCompletionRate, FeedbackCount)
    - GeneratedDate

4.6 Notifications & In-App Alerts Module

Features:

- Notify users about review deadlines and feedback updates

Entities:

- **Notification**
    - NotificationID
    - UserID
    - Type (ReviewReminder/FeedbackUpdate)
    - Message
    - Status (Unread/Read)
    - CreatedDate

5. Deployment Strategy

- **Local:** Angular/React frontend, Spring Boot/.NET Core backend, local DB.
- **Production:** On-prem or private cloud deployment with secure APIs; **no external integrations**.

6. Database Design

Tables:

- User(UserID, Name, Role, Email, Department, Status)
- Goal(GoalID, Title, Description, AssignedToUserID, StartDate, EndDate, Status)
- ReviewCycle(CycleID, Title, StartDate, EndDate, Status)
- PerformanceReview(ReviewID, CycleID, UserID, SelfAssessment, ManagerFeedback, Rating, ReviewDate)
- Feedback(FeedbackID, ReviewID, GivenByUserID, Comments, Date)
- Report(ReportID, Scope, Metrics, GeneratedDate)
- Notification(NotificationID, UserID, Type, Message, Status, CreatedDate)

7. User Interface Design

- **Employee Dashboard:** Set goals, submit self-assessment, view ratings and feedback.
- **Manager Dashboard:** Review employee performance, provide feedback, approve ratings.
- **Admin Dashboard:** Configure review cycles, manage users, generate appraisal reports.

8. Non-Functional Requirements

- **Performance:** Handle **8,000** concurrent users during review cycles.
- **Security:** RBAC, encrypted data, audit logs, internal-only access.
- **Scalability:** Horizontal scaling for review and goal modules; partition by department.
- **Availability: 99.9%** uptime; analytics can degrade gracefully.
- **Maintainability:** Modular architecture, API versioning, schema migration scripts.

9. Assumptions & Constraints

- Entirely **self-contained**: No external systems, APIs, or devices.
- Authentication and notifications are **in-app only**.
- No integration with external HR or payroll systems.
- No dependency on external messaging or email services.