1. **Project Overview**

The Employee Management System is a web application that allows managers to manage employee records, leave schedules, and performance. In addition, admins can post job openings, and potential candidates can view and apply for these jobs. The web application will be built using NodeJS/ExpressJS for the backend and MySQL for the database, while React will be used for the frontend.

**2.0) Objectives**

The objectives of the Employee Management System are as follows:

* To provide a centralized system for managing employee records, leave schedules, and performance.
* To allow managers to easily extract reports based on employee performance, number of hours worked, and other relevant metrics.
* To track the number of leave days due and taken by employees.
* To notify other employees of any approved leave applications and the relevant dates of the leave.
* To remind all employees when their leave dates are approaching.
* To provide a user-friendly interface that is easy to use and navigate.
* To allow admins to post job openings and potential candidates to view and apply for them.

**3.0) Features**

The Employee Management System will include the following features:

* Employee records management: The system will allow managers to add, edit, and delete employee records, including personal information, job details, and performance metrics.
* Leave management: The system will allow managers to manage employee leave schedules, including approving or rejecting leave applications, tracking the number of leave days due and taken, and notifying employees of approved leave applications.
* Performance management: The system will allow managers to track employee performance metrics, including number of hours worked, tasks completed, and other relevant metrics.
* Reporting: The system will allow managers to extract reports based on employee performance, number of hours worked, and other relevant metrics.
* Notifications: The system will send notifications to employees when their leave applications are approved and when their leave dates are approaching.
* User management: The system will provide authentication and authorization for users, with different levels of access based on their roles.
* Job posting: The system will allow admins to post job openings, including the job title, job description, required qualifications, and application deadline.
* Job search: The system will allow potential candidates to search for job openings based on various criteria, such as job title and location.
* Job application: The system will allow potential candidates to apply for job openings by submitting their resume and a cover letter.

**4.0) Architecture**

The Employee Management System will be built using a three-tier architecture, with the following components:

* Front-end: The front-end will be built using React, a modern web framework for building user interfaces. It will provide a user-friendly interface for managing employee records, leave schedules, and performance, as well as searching and applying for job openings. Redux state management is also used to manage the global state of this application.
* Back-end: The back-end will be built using NodeJS/ExpressJS, a popular web framework for building server-side applications. It will provide REST APIs for managing employee records, leave schedules, and performance, as well as job posting, job search, and job application functionalities, as well as authentication and authorization for users.
* Database: The database will be built using MySQL, a popular relational database management system. It will store employee records, leave schedules, performance metrics, and job-related data.

**5.0 Timeline and Deliverables**

The following is a tentative timeline for the project:

* Days 1-2: Requirements gathering and analysis, architecture design, and project planning.
* Days 3-4: Front-end development, including UI design, component development, and integration with the back-end APIs.
* Days 5-6: Back-end development, including API development, authentication and authorization implementation, and integration with the database.
* Days 7-8: Database development, including schema design, data modelling, and implementation.
* Days 9-10: Job posting, job search, and job application development, including API development and integration with the database.
* Days 11-12: Testing and debugging, including unit testing, integration testing, and system testing.
* Days 13-14: Deployment and maintenance, including deployment to a production environment, ongoing maintenance, and support.

The following are the deliverables for the project:

* Project proposal
* Architecture design document
* Front-end codebase and documentation
* Back-end codebase and documentation
* Database schema and documentation
* User manual and installation guide
* Testing and quality assurance reports

**Conclusion**

The Employee Management System is a web application that will provide managers with a centralized system for managing employee records, leave schedules, and performance, while allowing admins to post job openings and potential candidates to view and apply for them. The application will be built using NodeJS/ExpressJS and MySQL for the backend and database, and React for the frontend. The system will include features such as employee records management, leave management, performance management, reporting, notifications, user management, job posting, job search, and job application. The project will be implemented using a three-tier architecture and will be delivered over a period of 14 weeks.