# MiniTask Project – Problem and Solution Document

## 1. Problem Statement

The objective of this project is to build a secure, modular, and maintainable user management system using Spring Boot and PostgreSQL. The system should allow user registration, authentication, and session-based access control with session expiry validation. The project also includes database version management using Liquibase.

## 2. Solution Overview

The project was implemented in structured phases to ensure modularity and maintainability. Each phase built upon the previous one to achieve a fully functional and secure REST-based backend system.

## 3. Implementation Phases

### Phase 1 – Environment and Database Setup

PostgreSQL and pgAdmin were installed. A database named 'minitask' was created. Spring Boot was configured to connect to PostgreSQL via the application.properties file, ensuring successful connection.

### Phase 2 – Project Base Setup

A clean package structure was established under 'com.minitask' with sub-packages for controller, service, repository, entity, dto, and config. Dependencies for PostgreSQL driver were added for database connectivity.

### Phase 3 – Entities and Repositories

Two JPA entities, User and UserSession, were created to represent the users and their active sessions. Corresponding repositories, UserRepository and UserSessionRepository, were implemented by extending JpaRepository.

### Phase 4 – Service Layer Development

Business logic was implemented in the service layer to handle user registration, fetching user details, authentication, session token creation, and session validation logic.

### Phase 5 – REST Controllers

REST APIs were developed for user registration, retrieval, and authentication. DTOs were used to structure request and response bodies cleanly.

### Phase 6 – Authentication and Session Validation

Session-based authentication was implemented where each protected API checks for a valid 'X-Session-Token' header. The token validity was limited to 10 minutes, and expired or invalid tokens trigger a 401 Unauthorized response.

### Phase 7 – Testing

All endpoints were tested via Postman. Scenarios included successful registration, authentication, fetching users, and handling expired or invalid tokens.

### Phase 8 – Liquibase Integration

Liquibase was integrated for database schema management. Auto table creation was disabled, and changelog files were written to handle table creation and constraints in a version-controlled manner.

### Phase 9 – Cleanup and Documentation

Code was refactored for readability and maintainability. A README file was prepared summarizing project setup, endpoints, authentication flow, and Liquibase usage.

## 4. Conclusion

The MiniTask project successfully demonstrates secure session-based authentication using Spring Boot and PostgreSQL. It follows best practices in architecture, modular design, and database version management through Liquibase.