SRS Document – Contact List API

**SOFTWARE REQUIREMENTS SPECIFICATION**

Contact List API – Performance Testing

Version 1

1. **Purpose of the document:**

This document defines the functional and non-functional requirements of the Contact List API Performance Testing. It serves as a reference for developers, testers, and stakeholders to understand the scope, approach, and deliverables of performance testing activities for the Contact List API.

* Project Overview
* Information Architecture
* System Design

1. **Project Overview:**

**2.1 Audience:**

This document is intended for QA Engineers, Developers, Project Managers, and other stakeholders who are involved in ensuring the Contact List API meets performance requirements. It provides a clear understanding of the testing strategy, scenarios, and expected outcomes.

**2.2 Hardware and Hosting:**

The performance tests will be executed on the following environment:  
- OS: Windows 10/11  
- Tool: Apache JMeter  
- Backend: Contact List API Server  
- Database: Hosted on scalable servers  
- Network: Stable internet connection for simulating load conditions

1. **Information Architecture:**

The Contact List API provides endpoints for user management functionalities such as creating users, retrieving profiles, updating user data, authentication, and deletion. Performance testing ensures that all these endpoints can handle expected and peak loads efficiently.

## **3.1 API Modules**

The API includes the following modules/endpoints for performance validation:  
1. POST /users – Create a new user  
2. POST /users/login – Authenticate user  
3. GET /users/me – Retrieve user profile  
4. PATCH /users/me – Update user profile  
5. POST /users/logout – Log out user  
6. DELETE /users/me – Delete user account

## **3.2 Functional Areas Covered**

- User Registration & Authentication  
- Profile Management  
- Session Management (Login/Logout)  
- Load & Stress Handling for APIs  
- Error Handling for invalid scenarios

# **4. System Design (High-Level)**

Frontend: Not applicable (API-level testing)  
Backend: Node.js/Express REST APIs  
Database: NoSQL Database  
Performance Tool: Apache JMeter  
Authentication: JWT for secure access  
Hosting: Cloud-based hosting for API and database servers