

# Software Requirements Specification (SRS) for LinkedIn-Like Web Application using Laravel

## 1 Introduction

### 1.1 Product Overview

This document defines the software requirements for a LinkedIn-like social networking application built using the Laravel framework. The application is designed for job seekers, recruiters, and professionals aiming to build and maintain a professional profile, network with peers, and search or post jobs. The target users range from students to experienced professionals with a basic to intermediate level of technical expertise. The application will be web-based, accessible through major browsers, and optimized for desktops and mobile devices.

## 2 Specific Requirements

### 2.1 External Interface Requirements

#### 2.1.1 User Interfaces

- Clean and responsive UI using Bootstrap/Tailwind.
- Dashboard for user profile management, messaging, job postings, and feeds.
- Modal windows for login, post creation, and messages.
- Forms for registration, login, and profile editing.
- Navigation bar with routes to Home, My Network, Jobs, Messaging, Notifications, and Me.

#### 2.1.2 Hardware Interfaces

Not applicable. The system is web-based and hardware-independent from the user's side.

#### 2.1.3 Software Interfaces

- Laravel Framework (v10.x)
- MySQL Database
- PHP (v8.x)

- Operating System: Linux/Windows server
- APIs: RESTful APIs for internal data interaction and external job APIs integration

#### **2.1.4 Communications Protocols**

- HTTP/HTTPS for web communication
- SMTP for email verification and notifications
- WebSockets for real-time messaging (via Laravel Echo and Pusher)

## **2.2 Software Product Features**

### **F1 User Registration and Authentication** (Essential)

- The system shall allow users to register with email and password.
- Email verification shall be required to activate accounts.

### **F2 Profile Management** (Essential)

- Users shall create and update their professional profiles.
- Profile sections shall include education, experience, skills, and summary.

### **F3 Networking Features** (Important)

- The system shall allow users to connect with other users.
- Users shall receive notifications for connection requests.

### **F4 Job Posting and Search** (Essential)

- Employers shall post job listings.
- Users shall search and apply for jobs.

### **F5 Messaging System** (Important)

- The system shall support one-on-one real-time messaging.
- The system shall store chat history for each user.

### **F6 Feed and Content Sharing** (Desirable)

- Users shall post updates, share articles, and react to posts.

### **F7 Notifications** (Important)

- The system shall notify users of new connections, messages, and job offers.

## **2.3 Software System Attributes**

### **2.3.1 Reliability**

The system should have 99.9% uptime. Laravel's built-in error handling and exception reporting will be used. MTTF should be maximized through unit testing and error logging.

### **2.3.2 Availability**

Daily backup schedules and Laravel queue jobs will ensure availability. Session and queue data will persist through Redis or database drivers.

### **2.3.3 Security**

- Laravel's built-in CSRF protection and hashed passwords using bcrypt.
- JWT for API authentication.
- Role-based access control.
- HTTPS encryption.

### **2.3.4 Maintainability**

Codebase will follow Laravel's MVC architecture. Modular controllers and models ensure separation of concerns and easier updates.

### **2.3.5 Portability**

Code is platform-independent. Hosted on any LAMP/WAMP stack. Uses PHP, a portable language, and Laravel, which is framework-agnostic regarding OS.

### **2.3.6 Performance**

- Up to 1000 simultaneous users during peak time.
- Transaction throughput of 200 req/sec during job application peak.
- Optimized database queries and Redis caching.

## **2.4 Database Requirements**

- Users Table: id, name, email, password, profile data
- Jobs Table: id, title, description, company, location, user\_id
- Connections Table: user\_id, connection\_id, status
- Messages Table: sender\_id, receiver\_id, message, timestamp
- Posts Table: id, content, user\_id, created\_at
- Notifications Table: id, user\_id, type, message, read\_status