

# Phase 2:

# Software Requirements Specification (SRS)

Project: JobLaunch (Web-Based Job Application System)

Submission Deadline: December 7, 2025

## 1. Introduction

### 1.1 Purpose

This document outlines the architecture design and specific requirements of the web application, JobLaunch. The technology is designed to ease the recruitment process by linking job seekers to companies within a simplified, cloud-native setup. The document is intended to provide guidance to the team of developers and supervisors on the implementation process.

### 1.2 Scope

The system is a web-based media intended to handle job posting, applications submission, and status updates. The system is serverless, meaning it separates the frontend (Vercel CDN service) and backend (Vercel Serverless Functions service), and uses Supabase with PostgreSQL for the database and authentication.

### 1.3 Acronyms and Abbreviations

SRS: Software Requirement Specification

JWT: JSON Web Token (used for secure, stateless authentication)

API: Application Programming Interface

RDBMS: Relational database management system

CRUD: Create, Read, Update, Delet operations.

Admin: Responsible for managing users and posts.

Job Seeker: A user who searches and applies for job postings.

Company: An employer who registers and posts job vacancies.

## 2. Overall Description

### 2.1 Product Perspective and Architecture

JobLaunch is designed to operate as a three-tier cloud-hosted application involving Frontend, Serverless Backend, and Managed Database.

Frontend: HTML, CSS, JavaScript, and Tailwind CSS. The files were deployed on Vercel.

Backend: Serverless API functions running on Node.js and hosted on Vercel.

Database: PostgreSQL, which is managed by Supabase, and is securely connected via connection pooling.

### 2.2 User Characteristics

User Roles and Descriptions

- Job Applicant: Person searching for a job.

Functions: Look up jobs, edit profile, apply to jobs, and monitor status.

- Employer: Company representative.

Duties: Posting job openings, selection of applicants, and status updates on applications.

## **2.3. Operating Environment and Constraints**

The code needs to comply with principles of server-less computing, such as statelessness and limited execution time. The design should also exhibit responsiveness on desktop and mobile devices.

# **3. System Requirements**

## **3.1 Functional Requirements (FR)**

### **3.1.1 Authentication & Authorization**

FR-AUTH-01: User registration support with a choice between the "Seeker" and "Employer" role shall be provided.

FR-AUTH-02: All stored passwords should be hashed by standard hashing algorithms before storage.

FR-AUTH-03: After a successful log-in, the system should provide a JSON Web Token (JWT) to preserve and secure API log-ins.

### **3.1.2 Job Management (Employer)**

FR-JOB-01: Employers should be able to develop new jobs with the following details: Title, Description, Salary Range, Location, and Category.

FR-JOB-02: Employers should have the ability to edit and/or delete job postings written by themselves.

FR-JOB-03: The system shall allow Employers to see a list of applicants to each of their job postings.

### **3.1.3 Search & Browse (Seeker)**

FR-SEARCH-01: The system shall support searching and filtering by keywords, company name, and job category.

FR-SEARCH-02: The details of a job should be provided when selected.

### **3.1.4 Application Workflow**

FR-APP-01: The logged-in seekers should have the ability to send an application, optionally along with a message.

FR-APP-02: The system shall not allow a Seeker to apply to the same job more than once.

FR-APP-03: Employers must have the ability to status-update applications to predetermined stages (such as "Pending," "Interview Scheduled," and "Rejected").

FR-APP-04: Applicants shall have access to information on their applications and the status of each.

## **3.2 Non-Functional Requirements**

### **3.2.1 Performance**

NFR-05: Serverless API functions implementing CRUD operations must take less than 1.5 seconds to execute.

NFR-06: The result of job searching should be displayed to the user within 1 second after starting the process.

### **3.2.2 Security**

NFR-07: The system shall implement authorization checks on each secured API route to guarantee users access to and manipulate only data relevant to their role (for example, Employers should not be able to view another Employer's applicants).

NFR-08: The database connection string should be secured using Vercel Environment Variables and should not ever go into source control.

### **3.2.3 Usability**

NFR-09: User Interface shall utilize Tailwind CSS to enable a responsive and clean design, which should work on the desktop and mobile platforms.

NFR-10: Critical forms shall support client-side validation and server-side messages should always be similar.

## **4. Initial Database Setup (Using PostgreSQL)**

The relational database will hold the basic objects and their interconnections:

Users Table: Stores authentication information and role. Jobs Table: holds information on active and inactive jobs.

Applications Table: Relates Users (Seekers) to Jobs and maintains the status of the applications.