Software Requirements Specification (SRS)

1. Introduction

Project Title: Internship & Job Portal

Purpose: To build a backend system that powers an internship and job portal where students can find and apply for opportunities, and recruiters can post and manage listings.

Scope: This backend application will be developed using Java, Spring Boot, and MySQL. It will expose REST APIs for user management, authentication, job and internship listings, applications, and recruiter dashboards.

2. Overall Description

Product Perspective:

This is a new, standalone backend project. It will serve APIs to be consumed by a frontend application or mobile app.

User Classes and Characteristics:

- Student: Can register, login, view job/internship listings, and apply.
- Recruiter: Can post job/internship opportunities and view applicants.
- Admin (optional): Can manage users and listings.

Assumptions and Dependencies:

- The frontend will consume the exposed REST APIs.
- No third-party integrations are required initially.
- JWT authentication will be used.
- File uploads are stored locally (or optionally via database).

3. Functional Requirements

- User registration and login using JWT.
- Role-based access control for Students and Recruiters.
- Recruiters can post, edit, delete job/internship listings.
- Students can browse and apply to jobs/internships.
- Applications will be stored and viewable by recruiters.
- Resume upload support (optional).

- API response in JSON format.

4. Non-Functional Requirements

- Performance: 95% of API responses should be under 2 seconds.
- Scalability: Support for 10,000+ users.
- Security: JWT-based authentication, password hashing using BCrypt.
- Maintainability: Follows standard layered architecture.
- Availability: 99% uptime expected.

5. System Design (High-Level)

Architecture: RESTful service architecture using Spring Boot.

Database: MySQL

Entities:
- User: id, name, email, password, role
- Job: id, title, description, location, type, recruiter_id
- Application: id, user_id, job_id, resume_url, status

6. API Overview

- POST /api/auth/register: Register a new user
- POST /api/auth/login: Login and get JWT token
- GET /api/jobs: Fetch all job/internship listings
- POST /api/jobs: Create a job/internship (recruiter only)
- PUT /api/jobs/{id}: Edit job listing
- DELETE /api/jobs/{id}: Delete job listing
- POST /api/apply/{jobId}: Student applies for job
- GET /api/recruiter/applications: Recruiter views all applications

7. Constraints

- Backend is developed in Java with Spring Boot.
- Database used is MySQL.
- APIs must follow RESTful conventions.
- All communication should be secured using HTTPS.

8. Appendix

Tools and Technologies:

- Java 17
- Spring Boot
- MySQL
- IntelliJ IDEA
- Git & GitHub
- Postman for API testing
- Swagger for API documentation