***Practical - 10***

**Aim: Generate SRS (Software Requirements Specification) Document**

# 

## **Project Title: Online Job Portal**

## **Introduction**

### **1.1 Purpose**

The purpose of this document is to define the requirements for the development of the **Online Job Portal System**. The system is designed to streamline the recruitment process for job seekers and employers by providing an efficient platform for applying to jobs, conducting skill tests, and managing interview processes.

### **1.2 Scope**

The Online Recruitment System will allow users (job seekers) to create profiles, apply for jobs, take skill tests, and attend interviews. Employers will be able to post job listings, assess candidates based on skills, and offer job packages. The system will also automate parts of the recruitment process, such as skill testing and interview scheduling, and reducing manual effort.

### **1.3 Definitions, Acronyms, and Abbreviations**

* **SRS** – Software Requirements Specification
* **UI** – User Interface
* **ATS** – Applicant Tracking System
* **CRUD** – Create, Read, Update, Delete

### **1.4 References**

* IEEE Std 830-1998
* Bootstrap / React Documentation
* Node.js / Django Docs
* REST API Standards

### **1.5 Overview**

This document outlines the comprehensive requirements and design aspects of a **Job Portal System**. It includes both **functional** and **non-functional requirements**, detailing how the system should behave and the standards it must meet.

## **2. Overall Description**

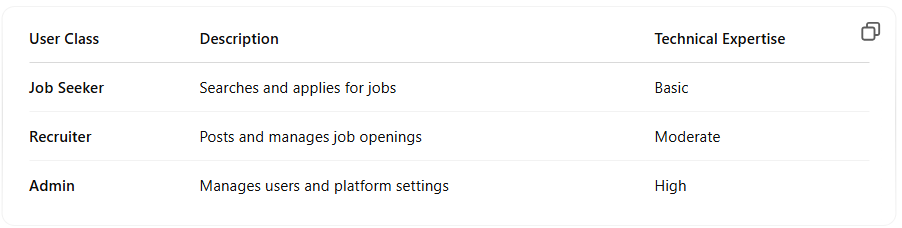
### **2.1 Product Perspective**

The Online Job Portal System is an independent web application that can optionally be integrated with third-party APIs like LinkedIn, Google Jobs, etc. It supports dynamic job listings, real-time notifications, and secure authentication.

### **2.2 Product Functions**

* Job Seekers can register, upload resumes, and apply for jobs
* Recruiters can post, edit, and delete job openings
* Admin manages user roles, reports, and site activity
* Notifications sent for job matches, application status, and recruiter actions

### **2.3 User Classes and Characteristics**



### **2.4 Operating Environment**

* **Frontend**: React.js / Angular / Flutter
* **Backend**: Node.js / Django / ASP.NET
* **Database**: PostgreSQL / MongoDB
* **Hosting**: AWS / Azure / Heroku

### **2.5 Design and Implementation Constraints**

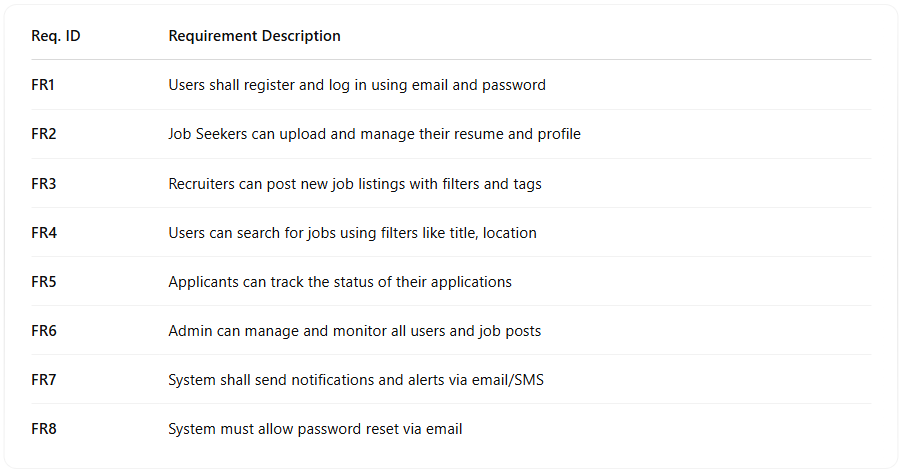
* Should be mobile and desktop compatible
* All user interactions must be secure (HTTPS, OAuth)
* Must comply with GDPR & data protection regulations

### **2.6 Assumptions and Dependencies**

* Users have access to a stable internet connection
* SMTP and Notification APIs are available
* Resume files are PDF or DOC format only

## **Specific Requirements**

### **3.1 Functional Requirements**



### **3.2 Non-Functional Requirements**

**3.2.1 Performance**

* The system should handle up to 10,000 concurrent users without performance degradation.
* Test results should be processed and displayed within 5 seconds after test completion.

**3.2.2 Security**

* **Data Encryption**: All sensitive information (e.g., personal details, Aadhaar number) will be encrypted in transit and at rest.
* **Role-Based Access Control**: Different users (job seekers, HR) will have access to specific functionalities based on their roles.

**3.2.3 Reliability**

* The system should have 99.5% uptime, ensuring that users can access it without interruption.

**3.2.4 Usability**

* The platform will feature a simple, intuitive interface to help users easily navigate through job applications, tests, and interviews.

### **3.3 External Interface Requirements**

#### **3.3.1 User Interfaces**

**Job Seeker Dashboard**

* Displays available jobs, application status, and upcoming tests or interviews.

**Employer Dashboard**

* Displays job listings, applicant status, and interview scheduling.

#### **3.3.2 Hardware Interfaces**

Optional Kiosk support for walk-in application booths

#### **3.3.3 Software Interfaces**

API integration with Firebase, Twilio, Google Jobs, LinkedIn  
Email servers for transactional communication

### **3.4 System Features**

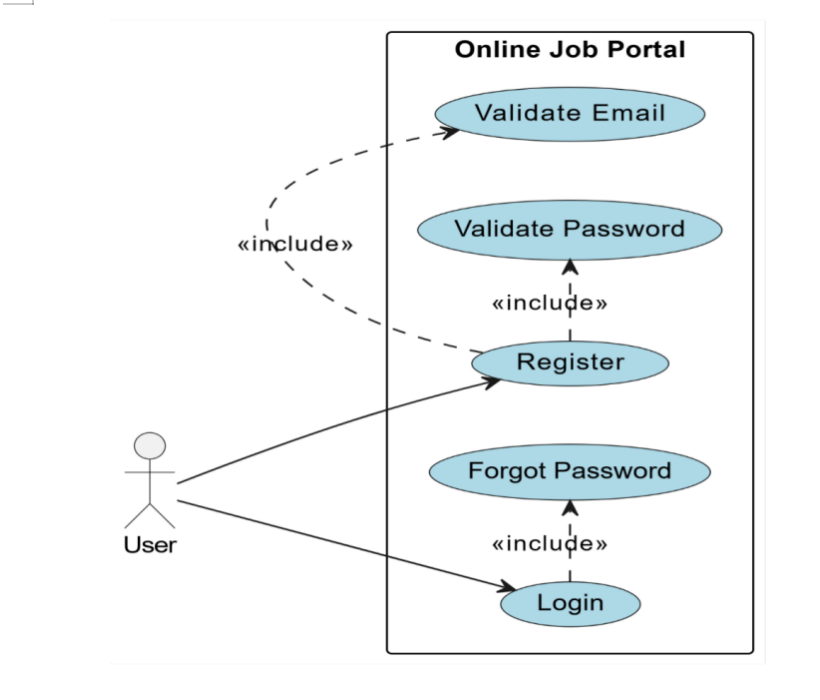
## 

## 

## 

## **Appendices**

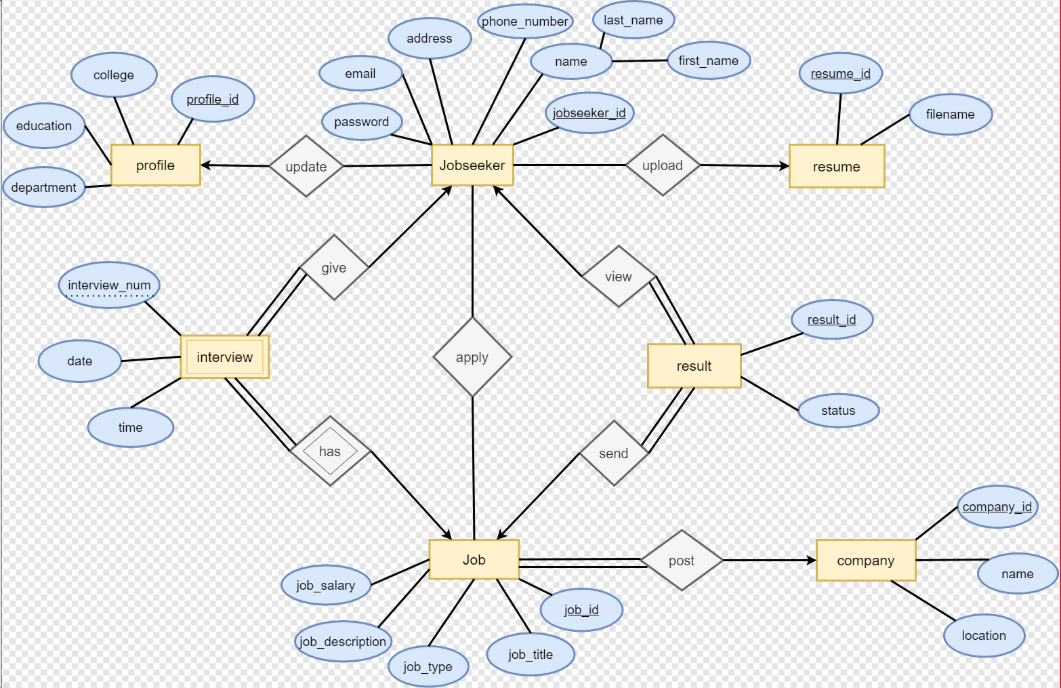
### **Appendix A: Use Case Diagrams**



### 

### 

### **Appendix B: Entity Relationship Diagram (ERD)**



### 

### 

### 

### **Appendix C: Data Flow Diagrams (DFD)**

* **Level - 0**

Request

Employers

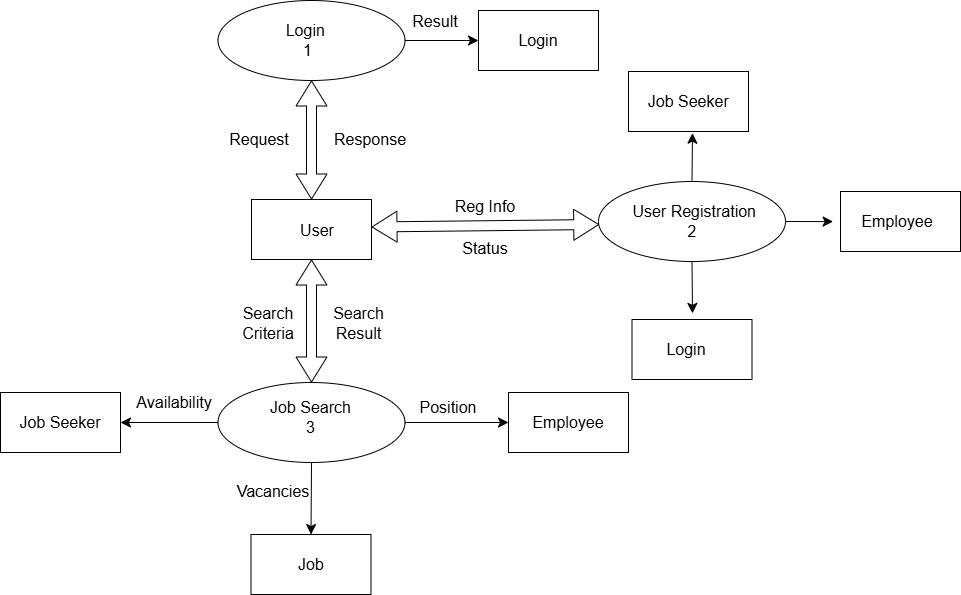
Job Portal

Job Seeker

Response

Admin

* **Level – 1**

****

* **Level - 2**

