Job Portal Project Documentation

1. Project Planning & Management

1.1 Project Overview

The Job Portal is a web-based platform that allows employers to post job listings and collect applications from job seekers. Job seekers can submit their CVs and respond to custom questions set by the employer. The platform facilitates efficient recruitment by enabling seamless interaction between job posters and applicants.

1.2 Objectives

- Provide an intuitive platform for employers to post jobs with custom application questions.
- Allow job seekers to apply for jobs by submitting CVs and answering job-specific questions.
- Ensure a streamlined hiring process through a well-organized dashboard for employers.
- Implement a secure and scalable system to manage job postings and applications.

1.3 Project Scope

- **Users**: Job posters (employers) and job seekers.
- **Core Features**:
  - Job posting with custom application questions.
  - CV submission and question answering by job seekers.
  - Application tracking system for employers.
  - Secure authentication and data protection.
- **Technology Stack**:
  - **Frontend**: React.js
  - **Backend**: .NET
  - **Database**: SQL-based (MSSQL)
  - **Storage**: Cloudinary/AWS for file uploads

1.4 Timeline & Milestones

Milestone	Deadline
Project Planning & Management	03/21/2025
Literature Review	03/21/2025
Requirements Gathering	03/21/2025
System Analysis & Design	03/21/2025
Implementation (Development)	05/09/2025
Testing & Deployment	05/09/2025

1.5 Risk Assessment & Mitigation

- **Security Risks**: Protect sensitive user data using encryption and authentication (JWT, OAuth).
- **Performance Issues**: Optimize database queries and implement caching strategies.
- **User Engagement**: Ensure user-friendly UI/UX with responsive design.

---

2. Requirements Gathering

2.1 Stakeholder Analysis

- **Employers**: Need an efficient way to manage job listings and applications.
- **Job Seekers**: Require a simple process to find and apply for jobs.

2.2 User Stories & Use Cases

- **As a job poster**, I want to create job listings with custom questions so that I can filter applicants effectively.
- **As a job seeker**, I want to upload my CV and answer job-related questions so that I can apply for jobs easily.
- **As an employer**, I want to review applications and filter candidates based on their responses and CVs.

2.3 Functional Requirements

- User authentication (Signup/Login for employers and job seekers).
- Job posting with title, description, custom questions, and deadlines.
- Job application submission with CV upload and question responses.

- Employer dashboard to manage job listings and applications.
- Search and filter functionality for job listings.

2.4 Non-Functional Requirements

- **Security**: Implement secure authentication and data protection.
- **Performance**: Ensure quick response times and efficient database operations.
- **Usability**: Design an intuitive and user-friendly interface.

---

3. System Analysis & Design

3.1 Problem Statement & Objectives

Many existing job portals lack a structured way for employers to collect additional information from applicants beyond a resume. This system aims to bridge this gap by providing a flexible platform where employers can post jobs with custom questions and manage applications effectively.

3.2 Use Case Diagram

Actors:

- Job Seeker
- Employer
- Admin (optional for moderation)

Use Cases:

- Job Seeker: Register, Login, Browse Jobs, Apply for Jobs, Upload CV.
- Employer: Register, Login, Post Jobs, Review Applications.

3.3 Software Architecture

- **Frontend**: React.js or Next.js (for enhanced SEO and performance).
- **Backend**:.NET Core
- **Database**:
    - MSSQL for structured data.
- **Authentication**: JWT for secure access.
- **Storage**: Cloudinary/AWS for CV uploads.

3.4 Database Design & Data Modeling

ER Diagram Overview

- **Users Table**: Contains job seekers and employers.
- **Jobs Table**: Stores job details and application questions.

- **Applications Table**: Holds applications, including CV links and question responses.

3.5 Data Flow & System Behavior

DFD Levels

- **Context Level**: Shows interactions between job seekers, employers, and the system.
- **Level 1**: Illustrates the flow of job postings, applications, and reviews.

3.6 UI/UX Design & Prototyping

- **Wireframes & Mockups**: Designed for key pages (home, job details, application form, employer dashboard).
- **UI Guidelines**: Maintain a clean, modern look with accessibility in mind.