**Full Stack Job Portal with Strapi and Next.js using REST API and Postgres**

**Software Requirements Specification**

Version 1.0



**Group Id: S240298ED1**

**Supervisor Name : Amjad Iqbal Khan**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date (dd/mm/yyyy)** | **Version** | **Description** | **Author** |
| Current date | 1.0 | Full Stack Jobs Portal is a web based project in which we will combine strengths of Next.js, Strapi, and Postgres. By following industry best practices, we aim to ensure our application is both resilient and efficient.  Next.js and Strapi are two powerful tools that developers use to create robust web applications. Next.js, which builds on React components, stands out with features like easy routing through files, rendering content on both the server and client sides, and built-in optimization. On the other hand, Strapi serves as a flexible CMS, allowing developers to create and customize APIs without being tied to specific frontend technologies. | BC200414326 |

**Table of Contents**

* [Scope (of the project)](#scope)
* [Functional Requirements Non Functional requirements](#FRNFR)
* [Use Case Diagram](#UCD)
* [Usage Scenarios](#UCS)
* [Adopted Methodology](#Adopted)
* [Work Plan (Use MS Project to create Schedule/Work Plan)](#Gantt)

**SRS Document**

**Scope of Project:**

Full Stack Job Portal project aims to develop a robust web application using Next.js, Strapi (with Postgres), and REST API for job seekers, companies, and administrators. The key features and scope of this project are as follows:

**1. User Roles and Authentication:**

* Implement user authentication and authorization using Simple JWT.
* Differentiate between user roles: Admin, Content Editor/Manager, and Frontend Users (registered companies and job seekers).

**2. Job Management:**

* Registered companies can add, update, and manage job postings.
* Specify job details including company name, address, expected salary, job type, experience required, and application deadline.
* Jobs are categorized by type (Permanent, Contractual, Full Time, Part Time).

**3. Job Search and Filtering:**

* Users can search jobs based on keywords and location.
* Filter jobs by type, education, experience, and salary range using checkboxes.

**4. Profile Management:**

* Registered users can update their profiles, upload CVs, and view the list of applied jobs.
* Companies can view and manage their posted jobs and see the list of candidates who applied.

**5. Content Management:**

* Content Editors/Managers can review, approve, and publish jobs posted by companies.
* Manage media assets associated with jobs (images, videos).

**Functional and non Functional Requirements:**

**Functional Requirements**

**Super Admin / Administrator**

**1. User Management**

* Ability to manage user accounts (create, update, delete).
* Grant or revoke access permissions for content editors/managers.
* View and manage registered companies.

**2. Job Management**

* Access to all job postings for moderation and approval.
* Ability to approve, reject, or modify job postings.
* View statistics on total jobs, pending jobs, approved jobs, and rejected jobs.

**3. Analytics and Reporting**

* View overall site statistics, including user activity, job trends, and user demographics.
* Generate reports on job statistics (e.g., job distribution by type, salary trends).

**Content Editor / Manager**

**1. Content Creation and Management**

* Create and publish job postings with required details (company name, address, salary, job type, experience).
* Edit and update existing job postings.
* Upload images and media related to job postings.

**2. Job Approval Workflow**

* Submit job postings for approval by the Super Admin.
* Receive notifications on job approval status.

**3. CV Management**

* Access and manage uploaded CVs by applicants.
* View and filter applicants based on job postings.

**4. Analytics**

* View job performance metrics (views, applications, hires).
* Monitor user engagement with job postings.

**Frontend Users (Registered Users/Companies)**

**1. Job Posting**

* Register as a company and post job listings.
* Specify job details (type, location, salary, experience).
* Set expiry date for job postings to prevent applications after the deadline.

**2. Job Search and Filtering**

* Search for jobs based on keywords and location.
* Filter job listings by type, education, experience, and salary range.

**3. Profile Management**

* Update user/company profile information.
* View applied jobs and their statuses (pending, accepted, rejected).
* View history of posted jobs and their application status.

**4. CV Upload**

* Upload and manage CVs for job applications.
* Receive notifications on application status changes.

**Non-Functional Requirements**

**1. Authentication and Authorization**

* Implement JWT-based authentication for user login and access control.
* Ensure role-based access to features and data.

**2. Exception Handling**

* Implement custom error handling for better user experience.
* Provide meaningful error messages and appropriate status codes.

**3. Performance and Scalability**

* Optimize server-side rendering with Next.js for faster page loads.
* Ensure efficient database queries and data caching.

**4. Maintainability:**

* Codebase should follow best practices for maintainability and extensibility, ensuring ease of future updates and modifications.

**5. Security**

* Implement data encryption for sensitive user information.
* Prevent unauthorized access and data breaches.

**6. Deployment and Hosting**

* Deploy the application on Railway or Vercel.

**7. File Hosting Integration**

* Integrate Cloudinary for secure and efficient file uploads (CVs, images).
* Ensure proper configuration and management of uploaded files.

Use Case Diagram(s):

<Provide here the use case diagram of your system>

Usage Scenarios:

<Provide here the usage scenarios of all use cases in table format explaining Use Case title, Use Case Id, Actions, Description, Alternative Paths, Pre and Post Conditions, Author, Exceptions. You are supposed to provide a usage scenario for each of use case shown in use case diagram>

Adopted Methodology

<Write here detail about the adopted methodology for your project. You must choose VU Process Model which is a combination of waterfall and spiral models. More or combining two or more methodologies is explained in SE2 (CS605) handouts. >

Work Plan (Use MS Project to create Schedule/Work Plan)

<Provide Gantt chart of your final project>