1. ABSTRACT :- Recruitment System

2. INTRODUCTION:-

1.1 Problem Statement

1.2 Software Requirement Specification:

Functional Requirements.

Non-Functional Requirements.

Hardware Requirements.

Software Requirements.

3. SOFTWARE:-

2.1 HTML

2.2 CSS

2.3 JSON

2.4 PostgreSQL

4. DIAGRAMS:-

3.1 Use case diagram.

3.2 Activity Diagram.

3.3 Class Diagram.

3.4 Interaction Diagram.

3.5 State Chart Diagram.

3.6 Deployment Diagram.

3.7 Components Diagram.

5. DATA ANALYSIS:-

5.1 Candidate information.

5.2 Application data.

5.3 Interview data.

5.4 Hiring outcomes.

5.5 Job post data.

6. PROJECT MODULES:-

6.1 User management module.

6.2 Job posting module.

6.3 Job application module

6.4 Resume management module.

6.5 Interview management module.

6.6 Admin dashboard module.

7. PROGRAMS:-

7.1. Index.html

7.2 Style.css

7.3 Package.json

8. METHODOLOGY:-

8.1 Requirement analysis.

8.2 System design.

8.3 Development.

8.4 Testing.

8.5 Deployment.

8.6 Maintenance & Updates.

9. RESULT:-

9.1 Conclusion.

1. ABSTRACT:-

Recruitment plays a crucial role in building a strong workforce. This recruitment system aims to streamline the hiring process by automating candidate management, job postings, and interview scheduling. With a user-friendly interface, it improves efficiency for both employers and job seekers. The system leverages data-driven insights to enhance decision-making, ensuring the best talent acquisition. Built using modern technology, it offers scalability and flexibility to meet evolving hiring needs.

2. INTRODUCTION:-

Problem statement :-

Recruitment is a fundamental process for organizations seeking to build a skilled workforce. However, traditional hiring methods often face challenges such as inefficiency, manual errors, and time-consuming procedures. The absence of a centralized system leads to difficulties in managing candidate applications, job postings, and interview scheduling. Additionally, organizations struggle with data-driven decision-making, making talent acquisition less effective.

The need for an automated recruitment system arises to address these issues by streamlining processes, enhancing transparency, and improving overall efficiency. By integrating technology-driven solutions, a recruitment system can optimize hiring workflows, reduce administrative burdens, and provide valuable insights into candidate selection. This system ensures a seamless experience for both recruiters and job seekers, ultimately leading to better hiring decisions and organizational growth.

SRS:-

Functional requirements:

1.User Authentication:

Secure login for recruiters and applicants.

2.Job Posting & Management:

Recruiters can create, edit, and delete job listings.

3.Candidate Application & Tracking:

Applicants can submit resumes, and recruiters can track application status.

4.Resume Parsing & Screening:

Automated analysis of resumes to shortlist candidates based on predefined criteria.

5.Interview Scheduling:

System-driven scheduling and notifications for interviews.

6.Decision-Making Support:

Recruiters can rate and comment on candidate applications.

7.Reporting & Analytics:

Insights into recruitment trends and candidate success rates.

Non-functional requirements:-

These define the system's overall behavior and efficiency:

Security:

Data encryption and role-based access control.

Performance:

Fast response time, optimized database queries, and high availability.

Usability:

Intuitive UI, easy navigation, and accessible design.

Scalability:

Ability to handle an increasing number of users and applications.

Reliability:

Minimal downtime and robust error handling.

Hardware requirements:-

The minimum hardware configuration for smooth system operation:

Server Requirements:

Processor: Intel Xeon or equivalent

RAM: 16GB minimum

Storage: SSD with at least 500GB

Network: High-speed internet connectivity

Client-Side Requirements:

Processor: Intel i5 or above

RAM: 8GB minimum

Storage: 250GB HDD/SSD

Display: Minimum 1080p resolution

Internet: Stable broadband connection

Software requirements:-

The software components required for development and deployment:

Operating System:

Windows Server, Linux (Ubuntu, CentOS)

Database Management System:

MySQL, PostgreSQL, or SQL Server

Frontend Frameworks:

React, Angular, or Vue.js

Backend Technologies:

Node.js, Django, or Spring Boot

Web Server: Apache or Nginx

Software:-

* **HTML (HyperText Markup Language)** – Used to structure the web pages and interface for recruiters and applicants. It provides the foundational framework for content display.
* **CSS (Cascading Style Sheets)** – Enhances the visual appeal of the recruitment system by styling elements, improving responsiveness, and ensuring a user-friendly design.
* **JSON (JavaScript Object Notation)** – A lightweight data format used for storing and exchanging structured information between the front end and back end, making API communication efficient.
* **PostgreSQL** – A powerful, open-source relational database management system for securely storing candidate profiles, job postings, application statuses, and recruiter data. It supports advanced query capabilities and scalability for large data sets.

Diagram:-

1. usecase diagram

This diagram represents the interactions between users and the system.

* **Actors:** Recruiters, Job Applicants, Admin
* **Use Cases:**
  + Recruiters post job vacancies
  + Applicants submit resumes
  + Admin manages user roles
  + System schedules interviews
  + Decision-making on candidate selection

### 2.Activity diagram:-

Illustrates the workflow of the system's processes.

* Applicant submits job application → System stores resume → Recruiter reviews application → Shortlisted candidates receive interview invitations → Hiring decision made

### 3.Class diagram:-

Defines the structure of the system in terms of classes and relationships.

* **Classes:**
  + User (Attributes: ID, Name, Role)
  + Job (Attributes: ID, Title, Description)
  + Application (Attributes: Candidate ID, Job ID, Status)
  + Interview (Attributes: Date, Time, Mode)

### 4. Iteration diagram:-

Shows the communication between objects in the system.

* **Sequence of interactions:**
  + Applicant submits resume → System validates details → Recruiter receives notification → Interview scheduled

### 5. State chart diagram:-

Defines various states an object can have in response to actions.

* **States for a job application:**
  + Submitted → Under Review → Shortlisted → Interview Scheduled → Hired / Rejected

### 6.Deployment diagram:-

Represents the system's physical architecture.

* **Servers:**
  + Web Server → Hosts the recruitment portal
  + Database Server → Stores applicant and job data
  + Cloud Server (optional) → Handles remote access

### 7. Component diagram:-

Shows the system’s modular structure.

* **Components:**
  + User Authentication Module
  + Job Posting Module
  + Resume Parsing Module
  + Interview Scheduling Module