**INTERNAL RECRUITMENT - IJP**

**Project Writeup**

IJP refers to “internal job posting.” When a job vacancy opens, the companies or the hiring teams always try to post it internally, and the main motive to do this is to determine whether there are internal candidates who can fill the position or not. Before any job posting, the hiring team ensures that a Job Description (JD) is prepared after cross-checking the duties that the hired candidates will have to perform with the team in need and the team manager to fill the position with the right candidate.

Once the required data is collected by the hiring team, the job description is prepared, and the hiring team posts the vacant position on the company website or informs the employees about the IJP (Internal Job Posting). To recruit internally, the hiring team first arranges a meeting with the business team and team managers to make sure that they have collected and validated the data for job requirements.

**Project Planning**

1. **Defining Objective:**
   1. Increased workforce satisfaction and retention rates by onboarding existing employees of the organization
2. **Identifying Key Entities:**
   1. **Candidate:** can apply for the job posting available and open in system
   2. **HR (Admin):** can create, read, update and delete job posting
   3. **Job Description:** description about the job, already discussed with business teams
   4. **Employees:** all existing employees’ record in the system
3. **Technology Stack**

|  |  |
| --- | --- |
| **Stack** | **Technologies** |
| **Frontend** | Angular v17, Bootstrap |
| **Backend** | Java, SpringBoot, Hibernate |
| **Version Control** | Git, Github |
| **Database** | RDBMS (MySQL, Postgres, h2) |
| **DevOps** | Docker, Kubernetes |
| **Cloud** | Google Cloud Platform (GCP) |
| **Testing** | Postman, Junit, Selenium, Jasmine, Karma |
| **Project Management Tool** | Jira |

1. **System Design**
   1. **Architecture**
      * **Microservice**: Each module as a separate service
      * **API Gateway:** Central point for API Request
      * **Database:** Centralized / Distributed with proper indexing
      * **Security:** Data encryption, Authentication, Authorization
   2. **Database Schema**: Tables as needed in the system
2. **Development**
   1. **Frontend** 
      * Use of Angular Framework
      * Responsive UI using Bootstrap
      * Authentication and Role based access
   2. **Backend**
      * RESTful APIs for modules
      * Business Logic and Data Validation
   3. **Integration**
      * Seamless integration with frontend and backend
      * Develop APIs for communication between different microservices
      * Data synchronization and consistency checks
3. **Testing**
   1. **Unit Testing**

* Write tests for individual components and functions

**b.** **Integration Testing**

* Test interactions between different modules and services
* Ensure end-to-end functionality

1. **Deployment**
   1. **Environment Setup**

* Configure cloud infrastructure (servers, databases, storage)
  1. **Monitoring and Maintenance**
* Use monitoring tools (e.g., Splunk, Grafana) to track system performance
* Implement logging, performance metrics and distributed tracing

1. **Documentation**
   1. **User Manuals**

* Create detailed documentation for HR managers and employees

1. **Conclusion**

**Project Overview**

1. **Project Title:** Internal Recruitment - IJP
2. **Duration:** 10 Days
3. **Team Size:** 05 full-stack developers
4. **Project Goals:**
5. Develop a comprehensive Internal Job Posting System focusing on Human Capital Management (HCM).
6. Deploy the system as a *Software as a Service (SaaS)* to ensure scalability, accessibility, and maintainability.
7. Implement core HR functionalities including recruitment, employee retention and satisfaction, gathering business requirements etc.
8. **Key Features**
9. **Candidates should be able to register in the system**
10. **HR (Admin) should be able to login in the system.**
11. **HR (Admin) should be able to add job advertisement to the system**
12. **General job position names should be added to the system. For example Software Developer, Software Architect.**
13. **Candidates should be able to be listed. (available Candidates List)**
14. **Job positions should be listed. (available job posting List)**
15. **All active job postings in the system should be listed**
16. **HR (Admin) should be able to modify and close a posting in the system**
17. **System Features**

|  |  |
| --- | --- |
| **Key Feature 1** | **Candidates should be able to register in the system** |
| Trigger | User clicks on “Apply” Link on the home page |
| Pre-condition | All the open jobs should be displayed at home page |
| Post-condition | The complete information about the user is stored in the application database |
| UI Screen | The form containing the following fields –   |  |  |  | | --- | --- | --- | | First Name | Text value |  | | Last Name | Text value |  | | Employee Id | Text value | Verified from DB | | Date of Birth | Date value | Date picker control | | Email | Text value | Verified from DB | | Apply | Button |  | | Cancel | Button |  |   \* Add more fields, if required. |
| Main Flow | The user provides the required details in the form and Clicks “Apply” button. On successfully saving the user’s details, user is forwarded to the home page along with the message “Details have been saved successfully.”.  Once successfully applied, the notification should dispatch to the Candidate manager and HR (Admin) for further action. |
| Data Validations | - All fields are required  - Input Data has to be validated at frontend |
| Business Rule | - Email and Employee ID to be verified from Database  - If there is a previously registered e-mail for selected job, the registration will not take place |
| Exception Flow | In case of any failure in Data Validation or Business Rules, user should be brought back to the registration screen along with the relevant error message. |
| Alternate Flow | When the user selects “Cancel” button, s/he should be taken back to the home Page |

|  |  |
| --- | --- |
| **Key Feature 2** | **HR (Admin) should be able to login in the system** |
| Trigger | User clicks on “Admin Login” Link on the home page |
| Pre-condition | Admin Login link appears on the Home Page |
| Post-condition | The user is successfully login in the system |
| UI Screen | The form containing the following fields –   |  |  |  | | --- | --- | --- | | Email | Text value | Verified from DB | | Password | Text value | Verified from DB | | Login | Button |  | | Cancel | Button |  | |
| Main Flow | The user provides the required details in the form and Clicks “Login” button. On successfully login, user is forwarded to the job posting dashboard page. |
| Data Validations | - All fields are required  - Input Data has to be validated at frontend |
| Business Rule | - Email and Password to be verified from Database |
| Exception Flow | In case of any failure in Data Validation or Business Rules, user should be brought back to the login screen along with the relevant error message. |
| Alternate Flow | When the user selects “Cancel” button, s/he should be taken back to the home Page |

|  |  |
| --- | --- |
| **Key Feature 3** | **HR (Admin) should be able to add job posting to the system** |
| Trigger | User clicks on “Add Job” Link on the Dashboard Page |
| Pre-condition | HR (Admin) should be logged in |
| Post-condition | New job posting is created in the system |
| UI Screen | The form containing the following fields –   |  |  |  | | --- | --- | --- | | Job Id | Text value | Auto populated | | Description | Textarea |  | | Designation | Drop down | Pre-populated | | Location | Drop down | Pre-populated | | Required Skill-set | Checkboxes |  | | Year of Experience | Drop down | Min Experience | | Languages Known | Multiple | Selection | | Salary Range | Range | Min – Max Range | | Post | Button |  | | Cancel | Button |  |   \* Add more fields, if required. |
| Main Flow | The user provides the required details in the form and Clicks “Post” button. On successfully posting a job, user is forwarded to the job posting dashboard page and receive acknowledged message about new post – “New job posted successfully” |
| Data Validations | - All fields are required  - Input Data has to be validated at frontend |
| Business Rule | - Designation and location to be populated from database |
| Exception Flow | In case of any failure in Data Validation or Business Rules, user should be brought back to the Add Job screen along with the relevant error message. |
| Alternate Flow | When the user selects “Cancel” button, s/he should be taken back to the Dashboard Page |