**Payroll Management System**

**Project Writeup**

Employee’s moral is affected by payroll administration and employee is the main resource of any organization. The main aim of this project is to develop a software application for organization through which they can manage financial information of employees. Using this application organization will manage employee leaves, salary, tax payment, check for every month, reports etc. Details are maintained in database for future use

**Project Planning**

1. **Defining Objective:**
   1. The system maintains the information on the salary of every employee of the whole organization & generates pay slips of employees.
2. **Identifying Key Entities:**
   1. **Leaves:** can apply for the job posting available and open in system
   2. **Salary:** description about the job, already discussed with business teams
   3. **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:** Payroll Management System
2. **Duration:** 10 Days
3. **Team Size:** 05 full-stack developers
4. **Project Goals:**
5. Develop a comprehensive Payroll Management 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 defining the salary structure, employee leave management, report generation etc.
8. **Key Features**
9. **Employee Module: Tracks all the information of the employee.**
10. **All types of CRUD Operation on Employee.**
11. **Employee is a role based module where *Admin* can perform each and every operation on data, but the employee will be able to view his/her data. Therefore access level restrictions has to be implemented in this project.**
12. **You are required to develop Salary Service to manage the employees’ salary.**
13. **You are required to develop Leave Service to manage the employees’ leaves.**
14. **Only Admins are able to perform CRUD operations on Salary structure.**
15. **Employee-Salary Reports to be generated by Accountants for stockholders**
16. **All Employees should be able to login in the system.**
17. **Pay slips to be generated for each employee considering the leave policy.**
18. **Employees can apply for a leave and the notification for same should be sent to his/her manager for further action**
19. **System Features**

|  |  |
| --- | --- |
| **Key Feature 1** | **User should be able to login in the system** |
| Trigger | User clicks on “Login” Link on the home page |
| Pre-condition | User should be available in the database |
| 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 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 2** | **Admin User should be able to create new Employee** |
| Trigger | User clicks on “Add Employee” Link on the Dashboard page |
| Pre-condition | All the employees should be displayed at Dashboard page |
| Post-condition | The complete information about the new employee is stored in the database |
| UI Screen | The form containing the following fields –   |  |  |  | | --- | --- | --- | | First Name | Text value |  | | Last Name | Text value |  | | Employee Id | Text value | Auto generated | | Date of Birth | Date value | Date picker control | | Mobile Number | Text value | Verified from DB | | Designation / Role | Drop down | Pre-populated | | Add Employee | Button |  | | Cancel | Button |  |   \* Add more fields, if required. |
| Main Flow | The user provides the required details in the form and Clicks “Add Employee” button. On successfully saving the 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 newly added user for further action. |
| Data Validations | - All fields are required  - Input Data has to be validated at frontend |
| Business Rule | - Mobile number to be verified from Database  - Employee ID is auto generated  - If there is a previously registered mobile number, 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 Dashboard Page |

|  |  |
| --- | --- |
| **Key Feature 3** | **Employee-Salary Reports** |
| Trigger | **User clicks on “Generate Reports” Link on the Dashboard Page** |
| Pre-condition | User (Accountant) should be logged in |
| Post-condition | Generated report for the stock-holders |
| UI Screen | The form containing the following fields –   |  |  |  | | --- | --- | --- | | Search by Month/Year | Drop down | Auto populated | | From | Date Value | Date picker preferred | | To | Date Value | Date picker preferred | | Search by Employee | Drop down | Pre-populated from Database | | Generate | Button |  | | Cancel | Button |  |   \* Add more fields, if required. |
| Main Flow | The user provides the required details in the form and Clicks “Generate” button. On success, the user should be able to download the report. |
| Data Validations | - Input Data has to be validated at frontend |
| Business Rule | - “Search by Employee” field 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 Dashboard 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 |