**EXIT FLOW**

|  |  |  |
| --- | --- | --- |
| **1** | **Name of the Project** | **Exit Flow** |
| ***2*** | **Problem Statement** | To design Bot like application, to Remove or transfer employee details from various HR systems post last working day of employee once the resignation is accepted in the system. The bot should start making necessary changes in relevant HR systems preparing flags to take effect post last working day of the employee eliminating dependency on HR agents to do this manually. |
| **3** | **Objective/ Vision** | Exit Flow is built with an objective to significantly improve the efficiency of the offboarding process. By automating the removal or transfer of employee details, it reduces the time and effort required by HR agents, allowing them to focus on more strategic tasks.  Features of Exit Flow Application with respect to HR:   1. Acknowledge the termination requests 2. Can initiate the termination process by filling the termination form if the organization wants to terminate any employee 3. Add employee details 4. View the list of terminated employees 5. View all the employees 6. Bot should move the terminated employee details to the History table automatically on successful termination   Features of Exit Flow Application with respect to Employees:   1. Initiate Process by filling resignation form 2. Track the status of termination request |
| **4** | **Users of the System** | HRs and Employees of the organization |
| **5.** | **Assumptions** | User is installed with any of the web browsers |
| ***6*.** | **Functional Requirements** | 1. User should login. If not an existing user, then they should register themselves with the platform. While registering, user need to provide the following information, Email, Password and confirm password and wait till his account gets activated by master admin. 2. User should be able to change the password if he/she forgets their password. 3. If the user is HR, they should be redirected to admin dashboard. 4. If the user is an Employee, they should be redirected to user dashboard. 5. Employee should be able to initiate the termination process by filling the resignation form. 6. Employee should be able to track the status of his termination request. 7. HR should be able to initiate the termination process if organization wants to terminate any employee. 8. HR should be able to acknowledge the termination request. 9. HR should be able to add the employee details. 10. HR should be able to view the registered and terminated employee details in a table. 11. On successful termination, terminated employee details should be moved to History table. |
| **7.** | **Non-functional requirements** | 1. App should be responsive to display consistently across multiple device screens. 2. The password should not be saved directly into the database (encryption method should be used). 3. UI should be user friendly. |
| **8.** | **Tools and Technologies to be used** | 1. Frontend – React.js, Material UI 2. Backend – Microservice Architecture, Spring Boot, Java 8, Lombok, Spring Data JPA 3. Database – MySQL 4. Docker Compose 5. Version Control: GIT 6. IDE: Visual Studio Code, IntelliJ 7. Others: Postman, Jacoco |

**Components of the Project**

|  |  |
| --- | --- |
| **Database Models** | **Description** |
| **User**  **(authService)** | userId  email  password  userType  isActive |
| **UserCredentials** | Email  password |
| **Employee** | id  firstname  lastname  email  city  phone  address  dob  doj  account  ifsc  bankName  compensation  position  department  country  state  gender  status  teamId |
| **History** | e\_id  firstname  lastname  email  city  phone  address  dob  doj  account  ifsc  bankName  compensation  position  department  country  state  gender  status  teamId |
| **Manager** | department  name  e\_id |
| **TeamLead** | teamId  name  e\_id |
| **Request** | e\_id  firstname  lastname  department  lastWorkingDay  resignationDate  status |
| **Asset** | assetId  present |

|  |  |
| --- | --- |
| **Services** | **Description** |
| **AuthService** | registerUser  updateUser  authenticateUser  sendOtp  verifyOtp  updatePassword  sendRegistrationConfirmationEmail  sendRegistrationAcceptanceEmail  sendOtpEmail  generateToken |
| **TerminationService** | findAllRequests  addRequest  removeRequest  getRequestByE\_Id  updateRequestByE\_Id  addAsset |
| **UserService** | addEmployee  removeEmployee  getAllEmployees  updateEmployee  getEmployeeById  getEmployeeByemail  addHistory  removeHistory  getHistoryById  getAllHistory  addManager  removeManager  getManagerByDepartment  managerExists  addTeamLead  removeTeamLead  getTeamLeadById  teamLeadExists |

|  |  |
| --- | --- |
| **UI Components** | **Description** |
| **Auth** | User perform register, login and change password using this component. |
| **ChatBot** | Employees can clarify questions about how to proceed with the termination service and get the status of termination service in this component. |
| **Dashboard** | User can view the homepage containing their dashboard which shows pending termination requests for HR; their own termination requests for employees and can route to different pages in the application based on the user type i.e., HR or an employee in this component. |
| **ResignationForm** | Users can apply for the resignation in this component. |
| **EmployeeInfo** | HR can view the details of all the employees and the terminated employees in the form of table in this component. |
| **InputForm** | HR can fill the details of employee in this component. |
| **MasterAdmin** | Master admin can view the user id, email of the users requested for registration into the application and set their user type as HR or employee in this component. |
| **Profile** | Users can view their profile in this component. |
| **Redux** | In this component, the details of the users who are logged in are stored in a redux store. |
| **Status** | HRs can acknowledge each step of termination request while employees can track their status of current step of termination request in this component. |
| **Utils** | This component has basic card structure that are displayed in the home page of users, no request page that is displayed on home page when there are no termination requests, not found page that is displayed when the user tries to route for the non-existing URL. |