**Project Title:   
Smart Employee Management System (SEM)**

**Project Description**

**Smart Employee Management System (SEM)** is a web-based Java application that enables HR departments to manage employee information efficiently. The system allows the creation, update, retrieval, and deletion (CRUD) of employee records, attendance management, department mapping, and more. The backend is powered by **PostgreSQL**, **Core Java** is used for business logic, **JDBC** for database communication, and **Web Services** (REST APIs) are used to expose functionality to external systems. **Selenium** is used for automating UI testing.

**Key Features / Modules**

**1. User Authentication Module**

* Login/logout system with admin and HR roles.
* Role-based access control.
* Password hashing and validation.

**2. Employee Management Module**

* Add/edit/delete employee details.
* Search and filter employees by department, joining date, etc.
* Assign roles and departments.

**3. Attendance Tracking Module**

* Add and update daily attendance.
* Generate attendance reports for employees.
* View history and trends.

**4. Department Management Module**

* Add/edit/delete departments.
* Map employees to departments.
* View department-wise employee list.

**5. RESTful Web Services (Spring Boot or Simple Java)**

* Expose employee data (GET, POST, PUT, DELETE).
* External clients can consume these APIs.
* JSON used as data format.

**6. Automation Testing Module (Selenium)**

* Automate login, employee CRUD operations, and attendance submission.
* Headless browser testing using Selenium WebDriver.
* Test report generation using TestNG or JUnit.

**Technologies Used**

| **Technology** | **Description** |
| --- | --- |
| Core Java | Business logic, object-oriented structure |
| JDBC | Database connectivity between Java app and PostgreSQL |
| PostgreSQL | Backend database to store employee, department, attendance data |
| RESTful Web Services | APIs to integrate with third-party systems (e.g., payroll, reporting) |
| Selenium WebDriver | UI automation testing |
| HTML/CSS/JS | Basic front-end (optional, can use JSP or any lightweight UI) |
| Apache Tomcat | Deployment server (if using JSP/Servlets) |
| Maven/Gradle | Dependency and build management |

**Database Schema (Simplified)**

**Table: employees**

| **Column** | **Data Type** |
| --- | --- |
| emp\_id | SERIAL (PK) |
| name | VARCHAR |
| email | VARCHAR |
| department\_id | INT (FK) |
| role | VARCHAR |
| joining\_date | DATE |

**Table: departments**

| **Column** | **Data Type** |
| --- | --- |
| department\_id | SERIAL (PK) |
| name | VARCHAR |

**Table: attendance**

| **Column** | **Data Type** |
| --- | --- |
| attendance\_id | SERIAL (PK) |
| emp\_id | INT (FK) |
| date | DATE |
| status | VARCHAR |

**Sample REST API Endpoints**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/employees | Get all employees |
| GET | /api/employees/{id} | Get employee by ID |
| POST | /api/employees | Add new employee |
| PUT | /api/employees/{id} | Update employee details |
| DELETE | /api/employees/{id} | Delete employee |

**Selenium Testing Plan**

* **Test Case 1:** Automate login functionality for Admin and HR
* **Test Case 2:** Automate "Add New Employee" form
* **Test Case 3:** Automate search employee
* **Test Case 4:** Automate attendance marking
* **Test Case 5:** Validate web service response using Selenium + REST-assured (optional)