**Employee Portal Technical Design**

Version 1.0

September 21, 2020

# **Overview**

To create an Employee Portal which can help to register new employees in the system and also provide list of available employees in the organization.

# **Business Objective**

As an Employee Portal user, I should be able to register new employees in the system with details of the employee with First Name, Last Name, Gender, Date of Birth and Department.

Additionally, the system should list the details of all the employees when the user requests it.

# **Technical Stack Used**

1. Spring boot with inbuilt logging, web and data-jpa features
2. Spring boot actuator
3. Apache Derby Database (in-memory)
4. Junit5 and Mockito for unit testing
5. Spring Rest API template

# **Components and Approach**

1. Spring Boot main application class – EmployeePortalApplication.java
2. Controller class – EmployeeController.java
3. Service class – EmployeePortalService.java
4. Dao interface – EmployeePortalRepository.java
5. Model Class – Employee.java

## **Implementation**

* API End points
  + /EmployeeService/register – This end point registers new employees with the details provided in the request for the Employee model object
  + /EmployeeService/employees – This end point lists the details of all the employees in sorting in ascending order of first name.
* SLF4j with logback logging technique is used to log details related to the API. This comes inbuilt with latest spring boot version. The logback.xml is added in the project classpath which contains details related to the logging pattern on the console and on the file.
  + RollingFileAppender has been used for standard output and errors generated in the application.
  + It uses size and time-based rolling policy.
* Exception handling is provided with CustomExceptionHandler to handle exceptions and provide appropriate message to the user.
* Profile based application.properties file is provided to run based on the environment selected.
* Inbuilt support for Apachy derby for database connection and table creation
* UUID support for generating primary key for the Employee table.

## **Database Changes**

An EMPLOYEE table has been created to save Employee details.

Structure of the table is as follows:

|  |  |  |
| --- | --- | --- |
| **Column** | **Data Type** | **Size of Column** |
| ID | VARCHAR | 255 |
| FIRST\_NAME | VARCHAR | 30 |
| LAST\_NAME | VARCHAR | 50 |
| GENDER | VARCHAR | 10 |
| DEPARTMENT | VARCHAR | 50 |
| BIRTH\_DATE | TIMESTAMP | - |