**Employee Payroll Management System with Spring Boot: JPA, CRUD Operations, and RESTful API**

**OVERVIEW**

Create a simple payroll service that manages the employees of a company. Store employee objects in a database, and access them (via JPA).

**FUNCTIONAL REQUIREMENTS**

* Create a "Controller" folder inside the "src" directory of the Spring Boot application.
* Create an "ApiController.java" file inside the "Controller" folder.
* Create "Models" folder inside the "src" directory of the Spring Boot application.
* Create " Employee.java" file inside the "Models" folder and define 9 variables: id, name, address, phone\_number, email, job\_title, department, salary, hire\_date.
* Create a "Service" folder inside the "src" directory of the Spring Boot application.
* Inside the "Service" folder, create a "EmployeeService.java" file which handles the business logic for managing the employee entities. This interface is typically annotated with @Service annotation.
* Create a "Repository" folder inside the "src" directory of the Spring Boot application.
* Inside the "Repository" folder, create a "EmployeeRepo.java" this is an interface that extends the JpaRepository interface provided by Spring Data JPA, and defines methods for CRUD (Create, Read, Update, Delete) operations on the employee entity. This interface is typically annotated with @Repository annotation.

**Implement the following API endpoints:**

* POST - "/" --> Adds a new employee record to the database.
* GET - "/{id}" --> Returns the employee information
* GET - "/" --> Returns the list of all employee information.
* GET - "/employee/first-three-characters-of-first-name"-->Returns the first three characters of the first name of all employees.
* GET –“ /employees/hired/{hire\_date}”-->Returns the list of employees hired on the specified date.

**Note:**

Copy and paste it into the **application.properties** file

**spring.jpa.hibernate.ddl-auto=update**

**spring.datasource.url=jdbc:mysql://localhost/song?createDatabaseIfN otExist=true**

**spring.datasource.username=root spring.datasource.password=examly**

**spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver spring.jpa.show-sql= true**

**spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect API endpoint: 8080**

**Platform Guidelines:**

* To run the command use Terminal in the platform.

**Spring Boot:**

* To start/run the application **'mvn spring-boot:run'**