```
                    Date : 27/02/2021
                     Spring Boot 9AM
                       Mr. RAGHU
        -----------------------------------------------
Mini Project Code:
 http://www.mediafire.com/file/4wbtjlgdkok9aux/SpringBoot2WebMvcMySQlC
rud_2702021_RAGHU_9AM.zip/file

Task:
1. Register User Example
   https://www.youtube.com/watch?v=rgG2_T-OB8g
2. Error and Exception Handling
  https://www.youtube.com/c/NareshIT/search?query=error%20raghu


-------------Stage Coding---------------------------------
Stage#1  Register Employee [done]
Stage#2  Display Data
Stage#3  Delete By Id
Stage#4  Edit Page and Update Data
Stage#5  UI BootStrap Design
Stege#6  UI-JQuery Validation
Stage#7  AJAX validation
Stage#8  Error Page Handling
Stage#9  Exception Handling
Stage#10 Log Implementation
 ------------------------------------------------
             Stage#2  Display Data

*) use method findAll() that gets data from DB in
   List<T> Format.

*) Use Model memory at Controller to send data to UI

*) At UI Page use Thymeleaf th:each(forEach)
   to convert List<T> to HTML Table

  <tr> -> th:each="ob:${collectionKey}"
  <td> -> th:text="${ob.variableName}"
-----coding steps----------------------------------------
#a. Define abstract method in Service Interface

--IEmployeeService.java---
  List<Employee> getAllEmployees();
----------------------

#b. Implement method in service Impl

---EmployeeServiceImpl.java--
public List<Employee> getAllEmployees() {
        List<Employee> list = repo.findAll();
        return list;
}
------------------------------

#c. Define method in Controller
```

```
--EmployeeController.java---
@GetMapping("/all")
public String showAllEmps(Model model) {
        List<Employee> list = service.getAllEmployees();
        model.addAttribute("list", list);
        return "EmployeeData";
}
------------------------------
```

#d. create one UI Page For Data Display

```
---EmployeeData.html---
<html xmlns:th="https://www.thymeleaf.org/">
<body>
<h3>EMPLOYEE DATA PAGE</h3>
<table border="1">
        <tr>
                <th>ID</th>
                <th>NAME</th>
                <th>SAL</th>
                <th>DEPT</th>
                <th>ADDR</th>
                <th>HRA</th>
                <th>TA</th>
        </tr>
        <tr th:each="ob:${list}">
                <td th:text="${ob.empId}"></td>
                <td th:text="${ob.empName}"></td>
                <td th:text="${ob.empSal}"></td>
                <td th:text="${ob.empDept}"></td>
                <td th:text="${ob.empAddr}"></td>
                <td th:text="${ob.empHra}"></td>
                <td th:text="${ob.empTa}"></td>
        </tr>
</table>
</body>
</html>
```

```
                *********************
                Stage#3  Delete By Id
                *********************
```

URL-Rewriting :- creating one HyperLink using static path
      and dynamic path is called as URL-Rewriting.

```
ex:
   /employee/delete?id=10
   /employee/delete?id=11
   /employee/delete?id=12
   /employee/delete?id=13
   ..etc
```

In above example : /employee/delete?id= is fixed(static)
   To get Id ${ob.empId} is changed for every row
   (dynamic)

```
Syntax for URL-ReWriting:
  /path/path(key=${expression})

  converted to :  ../path/path?key=val

ex :
   /employee/delete(id=${ob.empId})
  Converted to:
    /employee/delete?id=10
    /employee/delete?id=11
    /employee/delete?id=12
    /employee/delete?id=13
     ..etc
--Full Link---------------------------------
<a th:href="@{___}"> DELETE </a>

<a th:href="@{/employee/delete(id=${ob.empId})}"> DELETE </a>
------------------------------------------------------



-----coding steps--------------------------------------
#a. Define abstract method in Service Interface

  void deleteEmployee(Integer id);

#b. Implement method in service Impl

 public void deleteEmployee(Integer id) {
        repo.deleteById(id);
 }

#c Controller method
        @GetMapping("/delete")
        public String deleteEmp(
                        @RequestParam("id") Integer empId
                        )
        {
                service.deleteEmployee(empId);
                return "redirect:all";
        }

#d. create HyperLink for DELETE using
    URL Rewriting

<a th:href="@{/employee/delete(id=${ob.empId})}"> DELETE </a>
------------------------------------------------------------


Q) What is diff between
   for loop and
   for each loop and for each method

   for loop => start-end -condition
   for each => Array/Collection
   forEach  => Taken Function Interface Input
                what should be added in ForEach method Body
```

```
*)
@Autowired
private  HandlerMapping  hm;
```