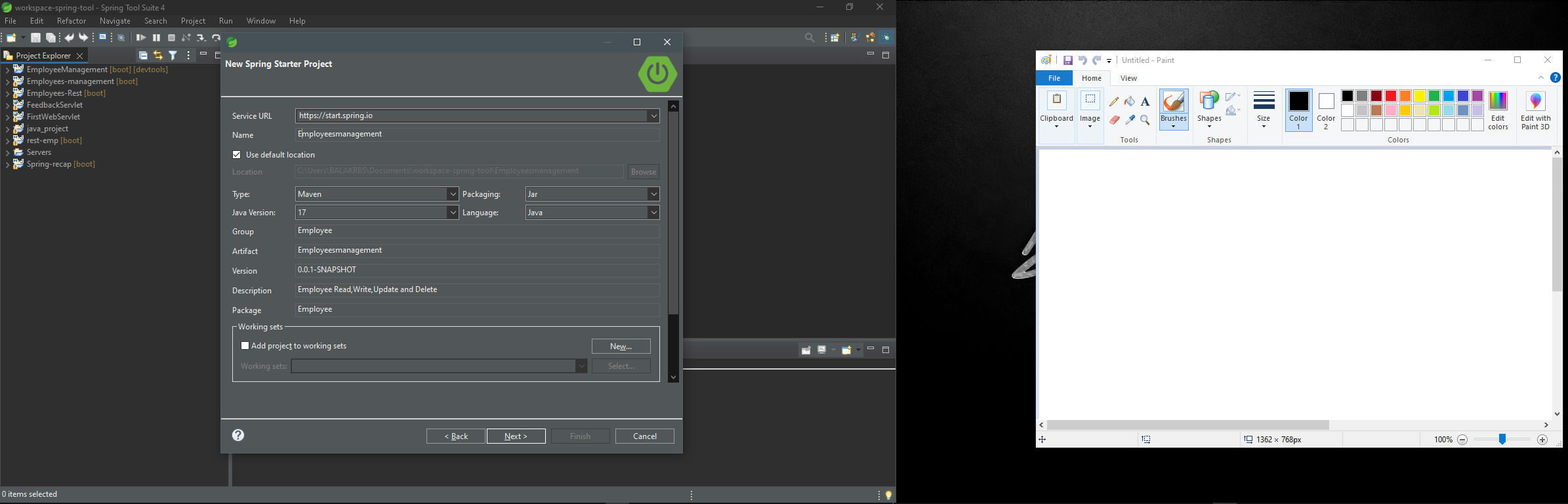
**GUVI FULL STACK JAVA**

Employee Management Application

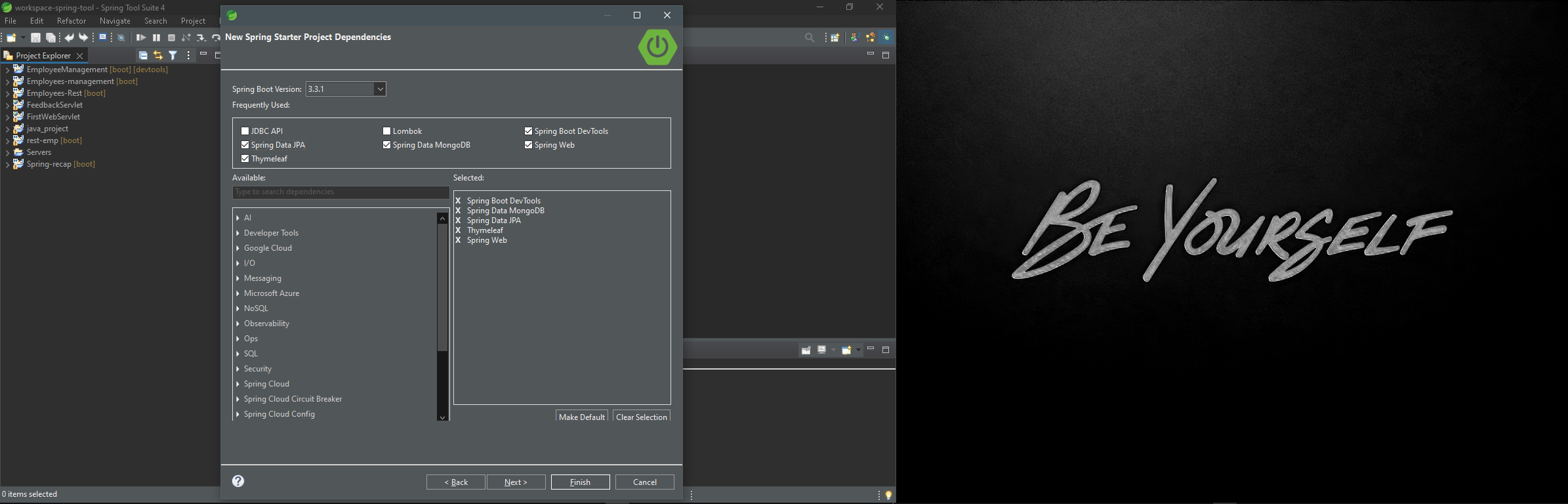
How to Setup-

1. To setup the project application follow the path follow given below:

File > New > Other > Select a wizard>Spring Boot>Spring Starter Project



1. Adding required dependencies for the project:

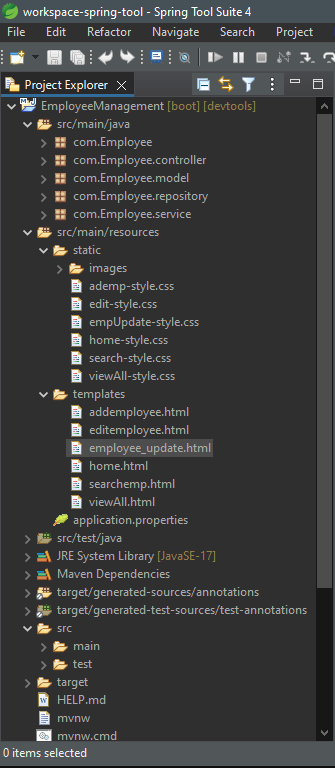


1. Now we have the “Employee Management” Spring boot project.

src/main/java – has the all java files for Back-end

src/main/resources/static – has the all css, images files for Front-end

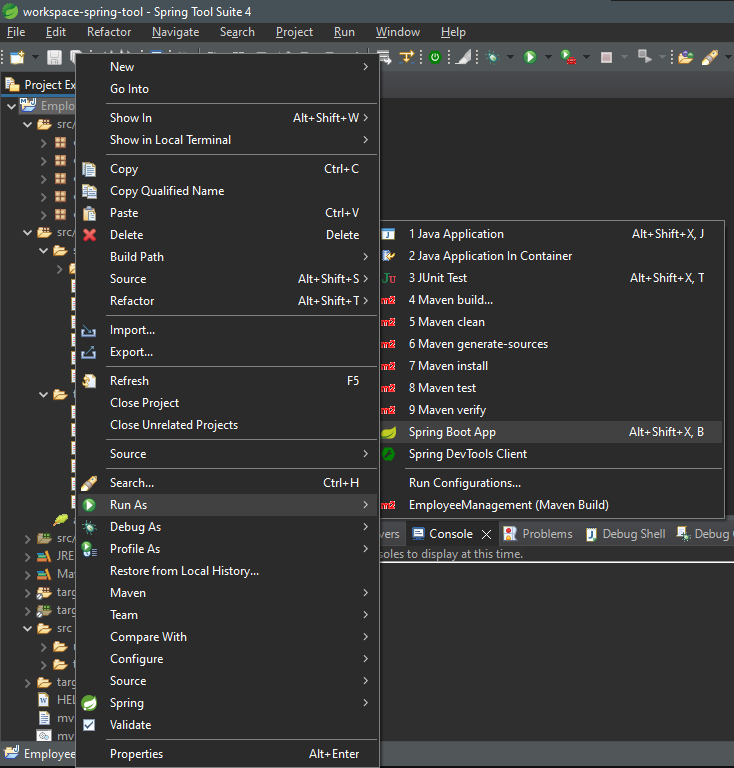
src/main/resources/templates – has the all html files for Front-end

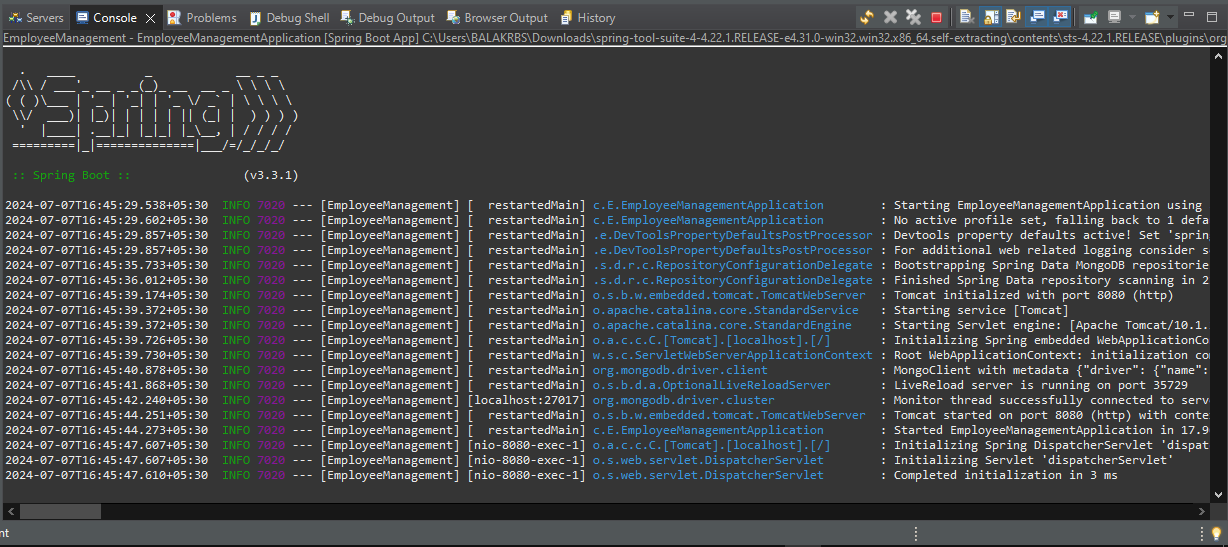


How to Run-

Follow the below path:

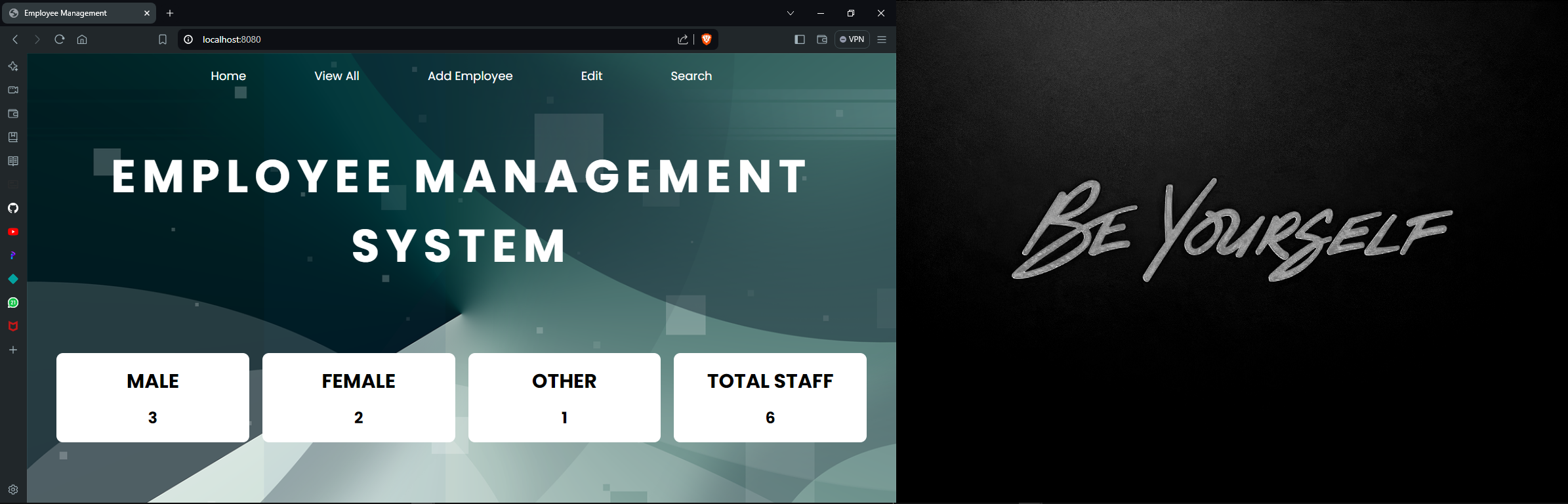
EmployeeManagement >Left click>Run As >Spring boot app



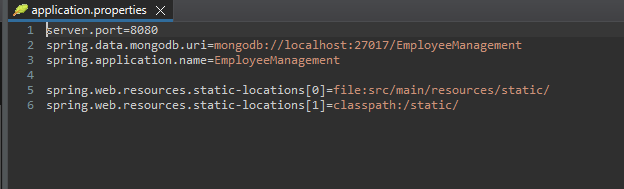


Browser:

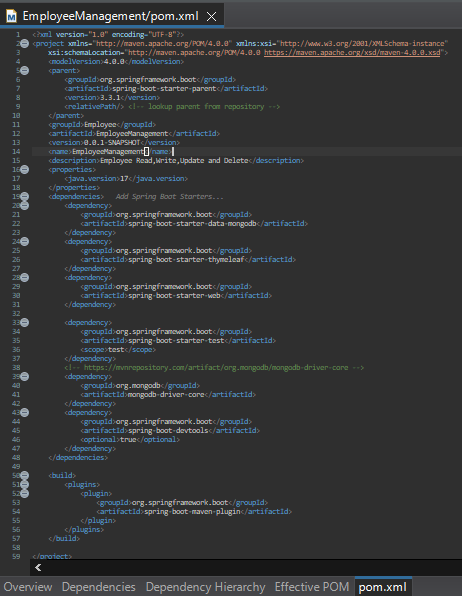
Enter the localhost:8080 in the brower url Tab.



This Application.properties has the server port, database url, application name and other properties.



This is Xml file has the All Dependency of the project.



Com.Employee.Controller

It has the main control of the project and give flow to the website by binding the Front-end and Back-end.

Code:

package com.Employee.controller;

import java.util.Optional;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.GetMapping;

import org.springframework.web.bind.annotation.ModelAttribute;

import org.springframework.web.bind.annotation.PathVariable;

import org.springframework.web.bind.annotation.PostMapping;

import com.Employee.model.Employee;

import com.Employee.service.EmployeeService;

@Controller

public class EmployeeController {

@Autowired

private EmployeeService empService;

//-----------------------Reload Landing page methods----------------

@GetMapping("/")

public String home(Model model) {

//--------------Counting methods and load Home page---------------

model.addAttribute("countT", empService.getTotalCount());

model.addAttribute("countM", empService.getMaleCount());

model.addAttribute("countF", empService.getFemaleCount());

model.addAttribute("countO", empService.getOtherCount());

return "home";

}

//-----------------------navigation forward methods----------------------

@GetMapping("/forwardSearch")

public String forwardSearch(Model model) {

model.addAttribute("employee", new Employee());

return "searchemp";

}

//-----------------------Add employee methods----------------------------

@GetMapping("/forwardAdd")

public String forwardAddemployee(Model model) {

model.addAttribute("employee", new Employee());

return "addemployee";

}

//---------------------------Editing methods---------------------------

@GetMapping("/displayEdit")

public String editEmployee(Model model) {

model.addAttribute("employees",empService.getAllEmployee());

return "editemployee";

}

@GetMapping("/showformforupdate/{id}")

public String showformforupdate(@PathVariable(value="id") String id,Model model) {

Optional<Employee> employee=empService.getEmployeeId(id);

model.addAttribute("employees",employee);

return "employee\_update";

}

@PostMapping("/update")

public String updateEmployee(@ModelAttribute Employee employee,Model model) {

empService.saveEmployee(employee);

return "redirect:/displayEdit";

}

//-------------------------delete methods-------------------------

@GetMapping("/deleteEmployee/{id}")

public String deleteEmployee(@PathVariable(value="id") String id) {

this.empService.deleteEmployeeById(id);

return "redirect:/displayEdit";

}

}