



EMPLOYEE MANAGEMENT

Submitted By: Shwetank Chaudhary



DECEMBER 31, 2023

NEC

INDEX

S.NO.	TOPIC	PAGE NO.
1.	OBJECTIVE/ PROBLEM STATEMENT	2
2.	FOLDER VIEW	4
3.	BACKEND	12
4.	FRONTEND	54
5.	Snapshots of Output	88
6.	Error Log	104
7.	Bibliography	106

OBJECTIVE/ PROBLEM STATEMENT

EMPLOYEE MANAGEMENT CONSOLE WEBSITE

Project includes:

- Home Page (before Sign in):
 - About Us
 - Contact Us
 - Login
- Home Page (after Sign in):
 - About Us
 - Contact Us
 - Log out
 - All Services
 - Eureka
- Services
 - Insert
 - Delete
 - Show All Employees
 - Update
 - Security
 - Eureka

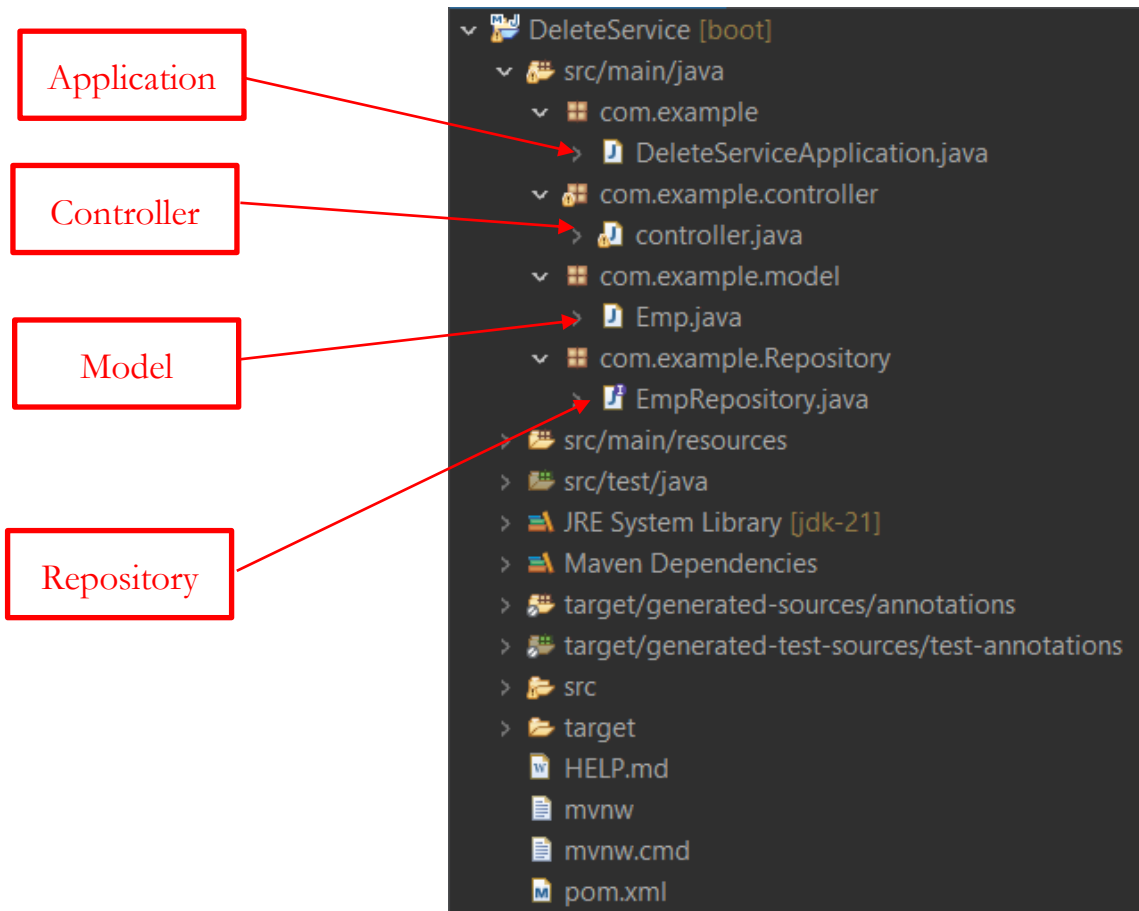
Output Snapshots

Bug Report

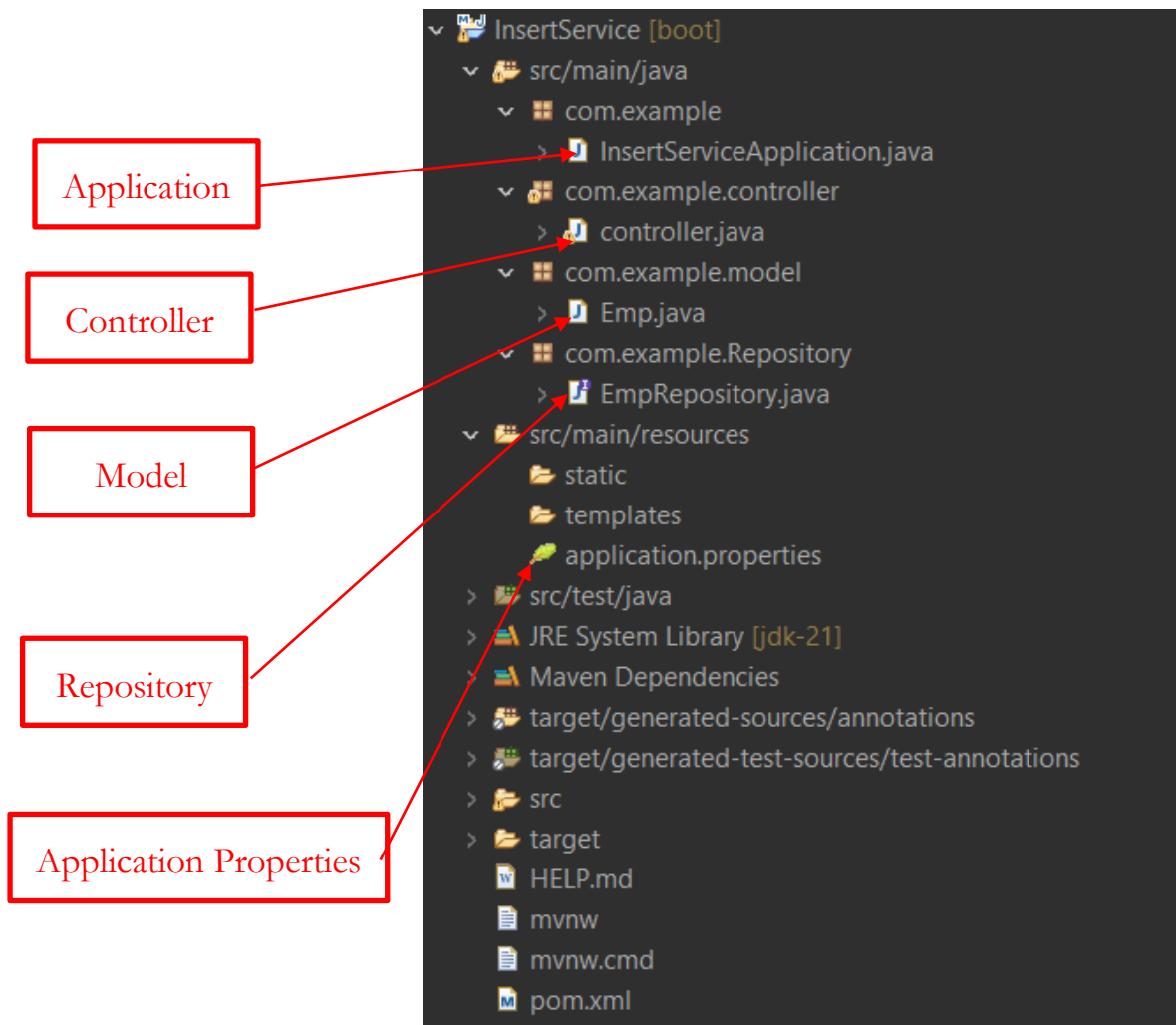
Reference Sites

FOLDER VIEW

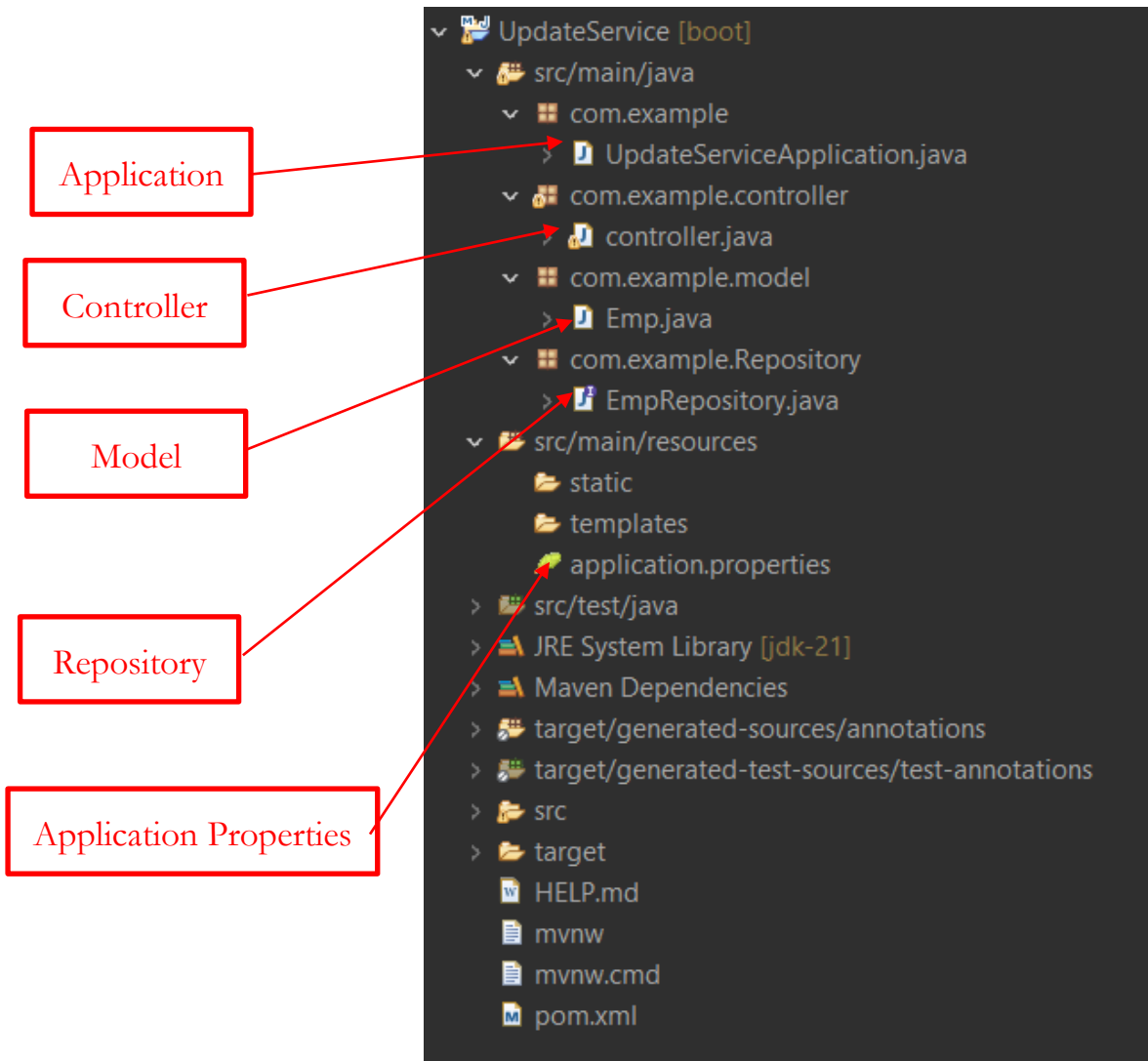
Delete Service



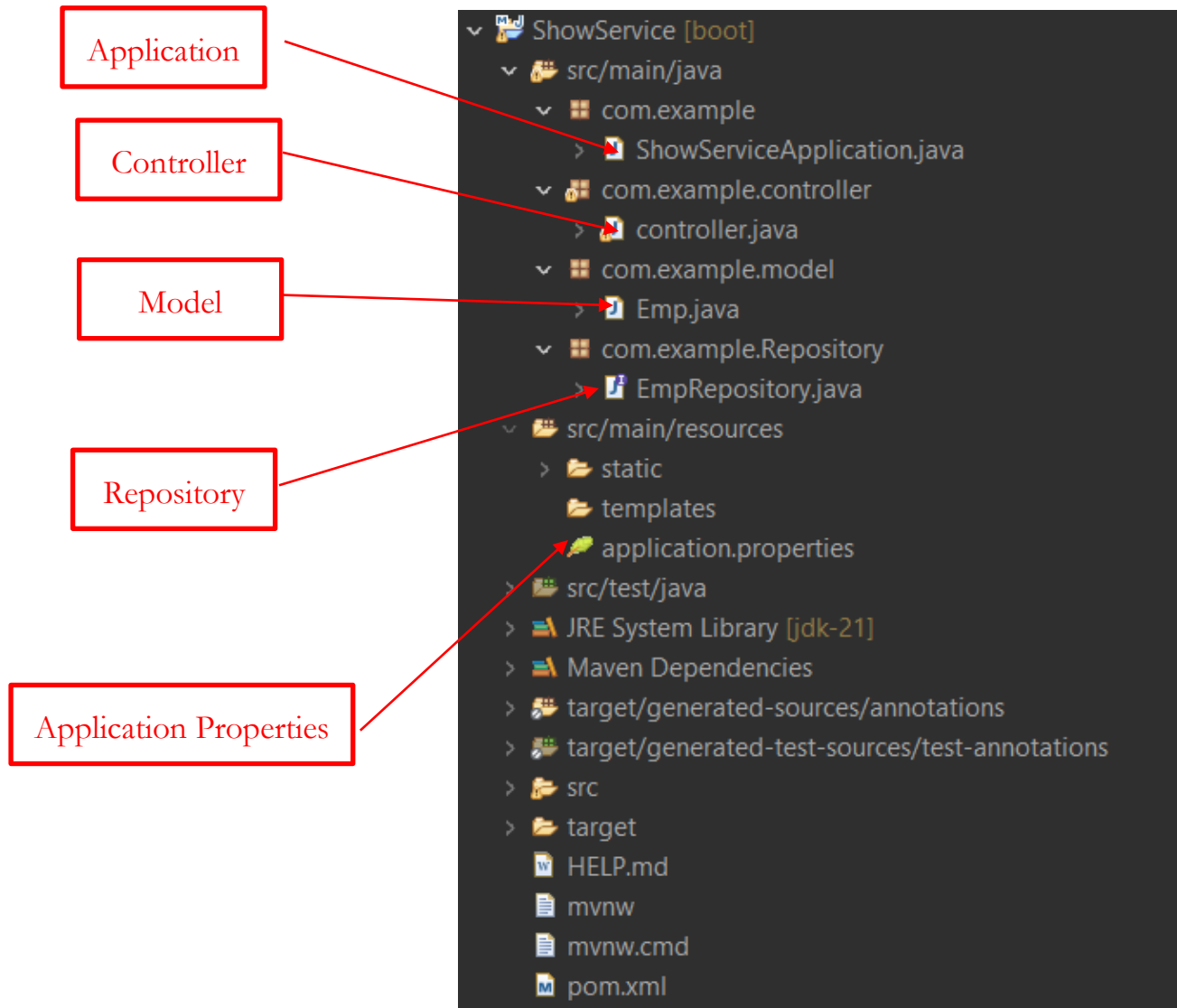
Insert Service



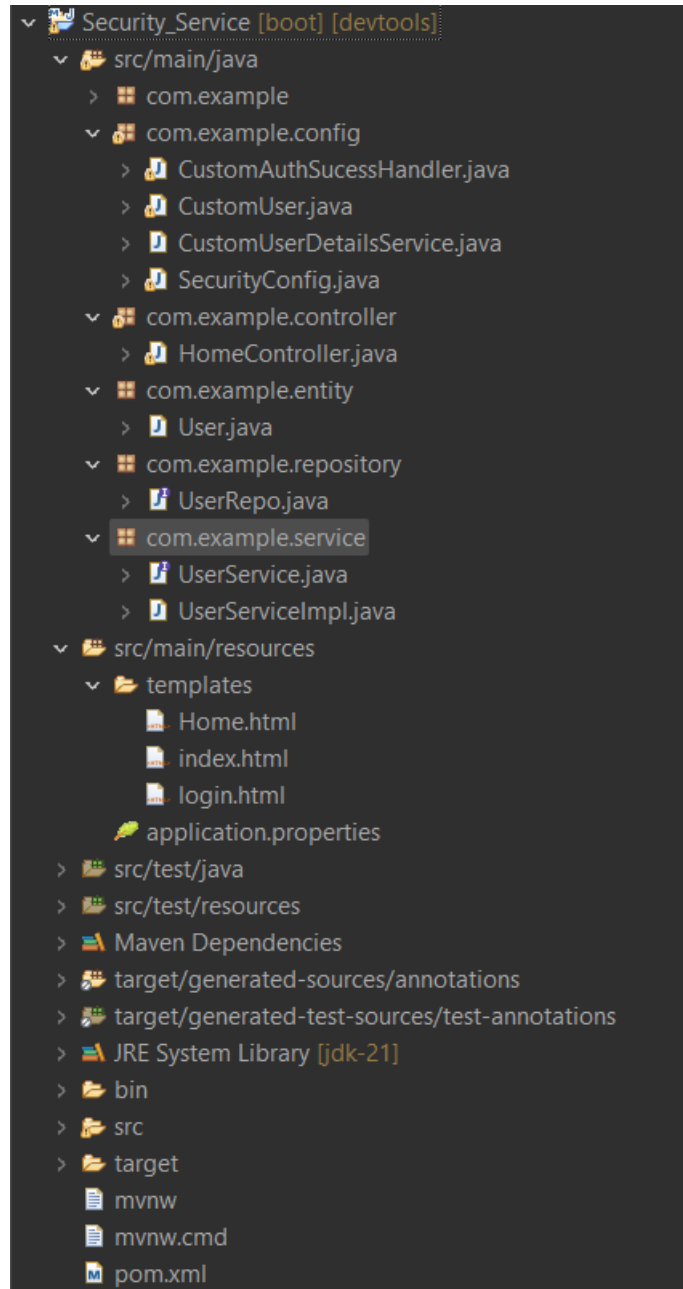
Update Service



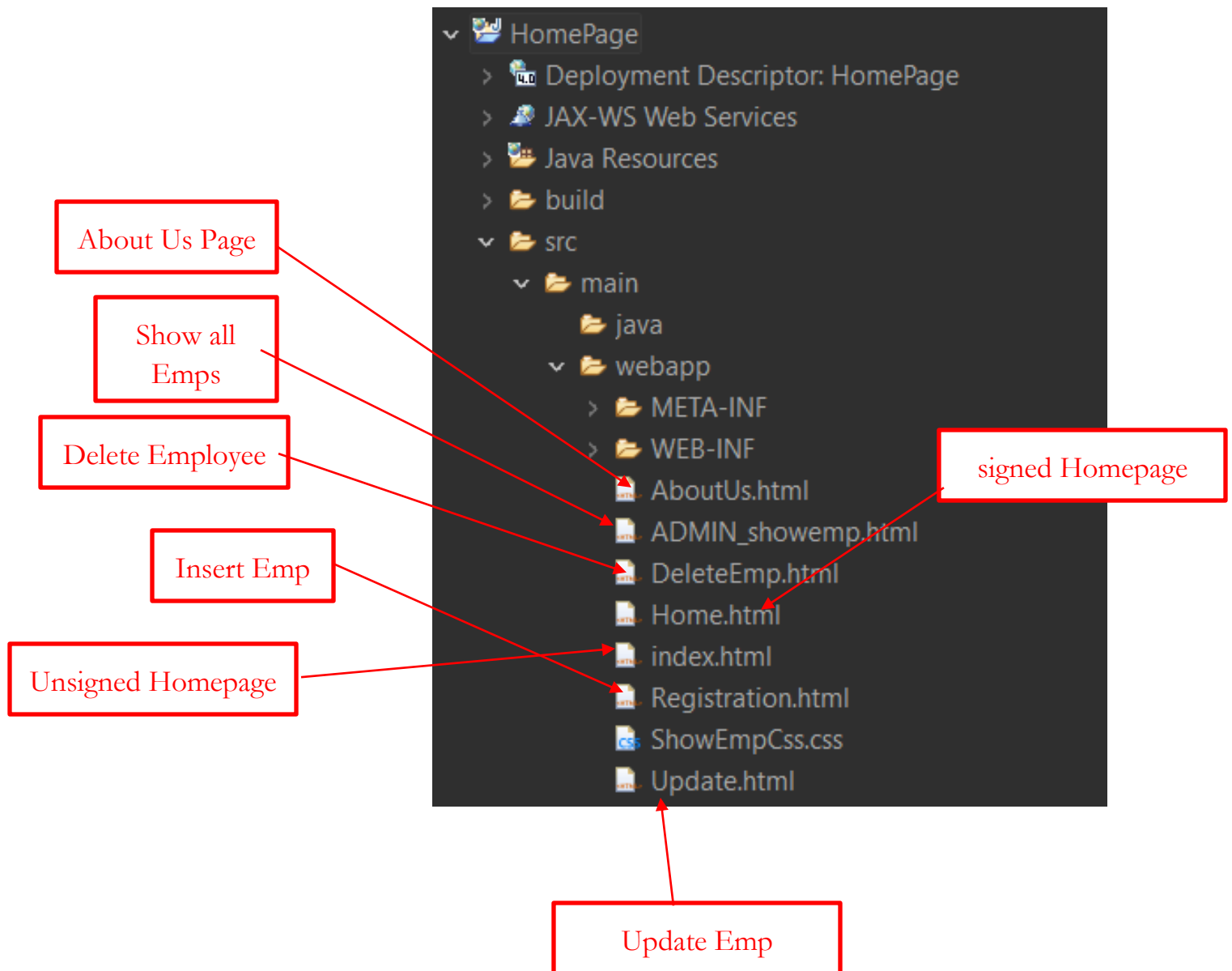
Display Service



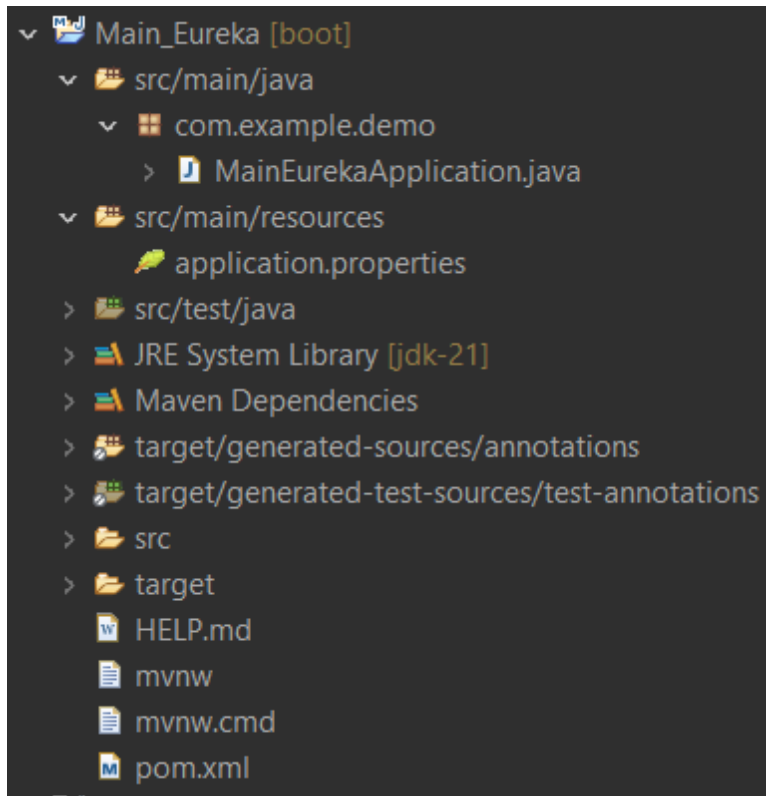
Security Service



HTML PAGES



Eureka



BACKEND CODES

Common Codes For All Services

MODEL

```
package com.example.model;

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Entity
@Table(name = "employees")
public class Emp {

    @Id
    @Column(name = "id")
    private int id;

    @Column(name = "username")
    private String username;

    @Column(name = "first_name")
    private String first_name;
```

```
@Column(name = "last_name")
```

```
private String last_name;
```

```
@Column(name = "phonenummer")
```

```
private String phonenummer;
```

```
@Column(name = "DOB")
```

```
private String DOB;
```

```
@Column(name = "position")
```

```
private String position;
```

```
@Column(name = "department")
```

```
private String department;
```

```
@Column(name = "hire_date")
```

```
private String hire_date;
```

```
@Column(name = "salary")
```

```
private float salary;
```

```
@Column(name = "password")
```

```
private String password;
```

```
@Column(name = "role")
private String role;

public int getId() {
    return id;
}

public void setId(int id) {
    this.id = id;
}

public String getPhonenumber() {
    return phonenumber;
}

public void setPhonenumber(String phonenumber) {
    this.phonenumber = phonenumber;
}

public String getUsername() {
    return username;
}
```



```
public void setUsername(String username) {  
    this.username = username;  
}
```

```
public String getFirst_name() {  
    return first_name;  
}
```

```
public void setFirst_name(String first_name) {  
    this.first_name = first_name;  
}
```

```
public String getLast_name() {  
    return last_name;  
}
```

```
public void setLast_name(String last_name) {  
    this.last_name = last_name;  
}
```

```
public String getDob() {  
    return DOB;  
}
```

```
}
```

```
public void setDob(String DOB) {  
    this.DOB = DOB;  
}
```

```
public String getPosition() {  
    return position;  
}
```

```
public void setPosition(String position) {  
    this.position = position;  
}
```

```
public String getDepartment() {  
    return department;  
}
```

```
public void setDepartment(String department) {  
    this.department = department;  
}
```

```
public String getHire_date() {
```

```
        return hire_date;
    }

    public void setHire_date(String hire_date) {
        this.hire_date = hire_date;
    }

    public float getSalary() {
        return salary;
    }

    public void setSalary(float salary) {
        this.salary = salary;
    }

    public String getPassword() {
        return password;
    }

    public void setPassword(String password) {
        this.password = password;
    }
}
```

```
public String getRole() {  
    return role;  
}
```

```
public void setRole(String role) {  
    this.role = role;  
}
```

```
public Emp(int id, String username, String first_name, String last_name, String  
phonenummer, String DOB, String position,  
String department, String hire_date, float salary, String password,  
String role) {  
    this.id = id;  
    this.username = username;  
    this.first_name = first_name;  
    this.last_name = last_name;  
    this.phonenummer = phonenummer;  
    this.DOB = DOB;  
    this.position = position;  
    this.department = department;  
    this.hire_date = hire_date;  
    this.salary = salary;  
    this.password = password;  
    this.role = role;
```

```
    }  
  
    public Emp() {  
    }  
  
}
```

REPOSITORY

```
package com.example.Repository;  
  
import org.springframework.data.repository.CrudRepository;  
  
import com.example.model.Emp;  
  
public interface EmpRepository extends CrudRepository<Emp, Integer> {  
  
}
```

Insert Service

APPLICATION

```
package com.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@EnableDiscoveryClient
@SpringBootApplication
public class InsertServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(InsertServiceApplication.class, args);
    }

}
```

CONTROLLER

```
package com.example.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.http.HttpStatus;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.security.crypto.password.PasswordEncoder;
import org.springframework.http.*;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.example.Repository.EmpRepository;
import com.example.model.Emp;

@RestController
@RequestMapping("/emp")
@CrossOrigin(origins = "*")
```

```

public class controller {

    @Autowired

    private EmpRepository repo;

    @Bean

    public PasswordEncoder passwordEncoder() {

        return new BCryptPasswordEncoder();

    }


    @PostMapping("/add")

    public ResponseEntity<String> addEmp(@RequestBody Emp emp) {

        try {

            if(repo.existsById(emp.getId())) {

                return

                ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Employee

                Can't be Registered");

            }

            else {

                emp.setPassword(passwordEncoder().encode(emp.getPassword()));

                repo.save(emp);

                return

                ResponseEntity.status(HttpStatus.CREATED).body("Employee Has been

                Registered");

            }

        }

        catch(Exception e) {

```



```
        return  
        ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Employee  
        Can't be Registered");  
    }  
    }  
}
```

APPLICATION PROPERTIES

```
spring.application.name=InsertService  
server.port=8088  
spring.datasource.url= jdbc:mysql://localhost:3306/Project_Emp_management  
spring.datasource.username= root  
spring.datasource.password= 123456  
spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.MySQLDialect  
spring.jpa.hibernate.ddl-auto= validate  
eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

Delete Service

APPLICATION

```
package com.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@EnableDiscoveryClient
@SpringBootApplication
public class DeleteServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(DeleteServiceApplication.class, args);

    }

}
```

CONTROLLER

```
package com.example.controller;
```

```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;

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

import com.example.Repository.EmpRepository;
```

```
@RestController
@RequestMapping("/emp")
@CrossOrigin(origins = "*")
public class controller {

    @Autowired
    private EmpRepository repo;

    @GetMapping("/del/{id}")
```

```

    public ResponseEntity<String> DelEmpPost(@PathVariable("id") int id) {
        if(repo.existsById(id)) {
            repo.deleteById(id);

            return
            ResponseEntity.status(HttpStatus.ACCEPTED).body("Employee Has been
            Deleted");
        }

        return
        ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Employee
        Can't be Deleted");
    }
}

```

APPLICATION PROPERTY

```

spring.application.name=DeleteService
server.port=8091
spring.datasource.url= jdbc:mysql://localhost:3306/Project_Emp_management
spring.datasource.username= root
spring.datasource.password= 123456
spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.MySQLDialect
spring.jpa.hibernate.ddl-auto= validate
eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

```

Show Service

APPLICATION

```
package com.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.client.discovery.EnableDiscoveryClient;

@EnableDiscoveryClient
@SpringBootApplication
public class ShowServiceApplication {

    public static void main(String[] args) {
        SpringApplication.run(ShowServiceApplication.class, args);
    }

}
```

CONTROLLER

```
package com.example.controller;

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

```
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.CrossOrigin;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

import com.example.Repository.EmpRepository;
import com.example.model.Emp;
```

```
@RestController
@RequestMapping("/emp")
@CrossOrigin(origins = "*")
public class controller {

    @Autowired
    private EmpRepository repo;

    @GetMapping("/show")
    // @PreAuthorize("hasAuthority(ADMIN)")
```

```

        public Iterable<Emp> ShowEmp() {
            return repo.findAll();
        }

        @PostMapping("/show")
        public Emp ShowEmpId(@RequestBody int id) {
            Emp e = repo.findById(id).orElse(null);
            return e;
        }

        @GetMapping("/show/{username}")
        public Emp ShowEmpGet(@PathVariable String username) {
            Emp e = repo.findByusername(username);
            return e;
        }
    }
}

```

APPLICATION PROPERTY

```

spring.application.name=ShowService
server.port=8089
spring.datasource.url= jdbc:mysql://localhost:3306/Project_Emp_management
spring.datasource.username= root

```


spring.datasource.password= 123456

spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.MySQLDialect

spring.jpa.hibernate.ddl-auto= validate

eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka

spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver



Update Service

APPLICATION

```
package com.example;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class UpdateServiceApplication {

    public static void main(String[] args) {

        SpringApplication.run(UpdateServiceApplication.class, args);

    }

}
```

CONTROLLER

```
package com.example.controller;

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

```
import org.springframework.context.annotation.Bean;

import org.springframework.http.HttpStatus;

import org.springframework.http.ResponseEntity;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.security.crypto.password.PasswordEncoder;

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

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

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

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

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

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


import com.example.Repository.EmpRepository;

import com.example.model.Emp;
```

```
@RestController

@RequestMapping("/emp")

@CrossOrigin(origins = "*")

public class controller {

    @Autowired

    private EmpRepository repo;
```

@Bean

```
public PasswordEncoder passwordEncoder() {  
    return new BCryptPasswordEncoder();  
}
```

@PostMapping("/update")

```
public ResponseEntity<String> updateEmpPost(@RequestBody Emp emp) {  
    if (repo.existsById(emp.getId())) {  
        Emp existingemp = repo.findById(emp.getId()).orElse(null);  
  
        if (existingemp != null) {  
            if (emp.getFirst_name() != null && !emp.getFirst_name().isEmpty()) {  
                existingemp.setFirst_name(emp.getFirst_name());  
            }  
  
            if (emp.getLast_name() != null && !emp.getLast_name().isEmpty()) {  
                existingemp.setLast_name(emp.getLast_name());  
            }  
  
            if (emp.getDob() != null && !emp.getDob().isEmpty()) {  
                existingemp.setDob(emp.getDob());  
            }  
  
            if (emp.getPosition() != null && !emp.getPosition().isEmpty()) {  
                existingemp.setPosition(emp.getPosition());  
            }  
        }  
    }  
}
```

```

        if (emp.getDepartment() != null && !emp.getDepartment().isEmpty())
        {
            existingemp.setDepartment(emp.getDepartment());
        }
        if (emp.getHire_date() != null && !emp.getHire_date().isEmpty()) {
            existingemp.setHire_date(emp.getHire_date());
        }
        if (emp.getSalary() != 0) {
            existingemp.setSalary(emp.getSalary());
        }
        if (emp.getPassword() != null && !emp.getPassword().isEmpty()) {

existingemp.setPassword(passwordEncoder().encode(emp.getPassword()));

        }
        if (emp.getRole() != null && !emp.getRole().isEmpty()) {
            existingemp.setRole(emp.getRole());
        }
        if (emp.getPhonenumber() != null &&
!emp.getPhonenumber().isEmpty()) {
            existingemp.setPhonenumber(emp.getPhonenumber());
        }

        repo.save(existingemp);

        return
ResponseEntity.status(HttpStatus.ACCEPTED).body("Employee has been updated");

```

```

    }

    return
    ResponseEntity.status(HttpStatus.NOT_FOUND).body("Employee has not been
    found");
    }

    return
    ResponseEntity.status(HttpStatus.INTERNAL_SERVER_ERROR).body("Employee
    cannot be updated");
    }

}

```

APPLICATION PROPERTIES

```

spring.application.name=UpdateService
server.port=9696
spring.datasource.url= jdbc:mysql://localhost:3306/Project_Emp_management
spring.datasource.username= root
spring.datasource.password= 123456
spring.jpa.properties.hibernate.dialect= org.hibernate.dialect.MySQLDialect
spring.jpa.hibernate.ddl-auto= validate

```

```
eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

Security Service

PASSWORD ENCODER

```
package com.example;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

public class demo {

    public static void main(String[] args) {

        System.out.println(new BCryptPasswordEncoder().encode("1234"));

    }

}
```

SECURITY CONFIG

```
package com.example.config;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import org.springframework.security.authentication.dao.DaoAuthenticationProvider;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
```

```
import
org.springframework.security.config.annotation.web.configuration.EnableWebSecurity;

import org.springframework.security.core.userdetails.UserDetailsService;

import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import org.springframework.security.web.SecurityFilterChain;

import
org.springframework.security.web.authentication.AuthenticationFailureHandler;
```

@Configuration

@EnableWebSecurity

```
public class SecurityConfig {
```

```
    @Autowired
```

```
    public CustomAuthSuccessHandler successHandler;
```

```
    @Bean
```

```
    public BCryptPasswordEncoder passwordEncoder() {
```

```
        return new BCryptPasswordEncoder();
```

```
    }
```

```
    @Bean
```

```
    public UserDetailsService getDetailsService() {
```

```
        return new CustomUserDetailsService();
```

```
    }
```


@Bean

```
public DaoAuthenticationProvider getAuthenticationProvider() {  
    DaoAuthenticationProvider daoAuthenticationProvider = new  
    DaoAuthenticationProvider();  
    daoAuthenticationProvider.setUserDetailsService(getDetailsService());  
    daoAuthenticationProvider.setPasswordEncoder(passwordEncoder());  
    return daoAuthenticationProvider;  
}
```

@Bean

```
public SecurityFilterChain filterChain(HttpSecurity http) throws Exception  
{
```

```
    http.csrf().disable()
```

```
    .authorizeHttpRequests().requestMatchers("/user/**").hasRole("USER")  
    .requestMatchers("/emp/**").hasRole("ADMIN")  
    .requestMatchers("/home.html/**").hasRole("ADMIN")  
    .requestMatchers("/**").permitAll().and()  
    .formLogin().loginPage("/signin")  
    .loginProcessingUrl("/userLogin")  
    .successHandler(sucessHandler)  
    .permitAll();
```

```
        return http.build();
    }

}
```

CUSTOM USER DETAIL SERVICE

```
package com.example.config;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.core.userdetails.UserDetails;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.core.userdetails.UsernameNotFoundException;
import org.springframework.stereotype.Component;

import com.example.entity.User;
import com.example.repository.UserRepo;

@Component
public class CustomUserDetailsService implements UserDetailsService {
```

@Autowired

private UserRepo userRepo;

@Override

public UserDetails loadUserByUsername(String username) throws
UsernameNotFoundException {

 User user = userRepo.findByUsername(username);

 System.out.println(user);

 if (user == null) {

 throw new UsernameNotFoundException("user not found");

 } else {

 return new CustomUser(user);

 }

}

}

CUSTOM USER

package com.example.config;

import java.util.Arrays;

```

import java.util.Collection;

import org.springframework.security.core.GrantedAuthority;
import org.springframework.security.core.authority.SimpleGrantedAuthority;
import org.springframework.security.core.userdetails.UserDetails;

import com.example.entity.User;

public class CustomUser implements UserDetails {

    private User user;

    public CustomUser(User user) {
        super();
        this.user = user;
    }

    @Override
    public Collection<? extends GrantedAuthority> getAuthorities() {

        SimpleGrantedAuthority authority = new
SimpleGrantedAuthority(user.getRole());

        return Arrays.asList(authority);
    }
}

```

```
}
```

```
@Override
```

```
public String getPassword() {  
    return user.getPassword();  
}
```

```
@Override
```

```
public String getUsername() {  
    return user.getUsername();  
}
```

```
@Override
```

```
public boolean isAccountNonExpired() {  
    return true;  
}
```

```
@Override
```

```
public boolean isAccountNonLocked() {  
    return true;  
}
```

```
@Override
```

```

        public boolean isCredentialsNonExpired() {
            return true;
        }

        @Override
        public boolean isEnabled() {
            return true;
        }
    }
}

```

CUSTOM AUTH SUCCESS HANDLER

```

package com.example.config;

import java.io.IOException;
import java.security.Principal;
import java.util.Set;

import org.springframework.security.core.Authentication;
import org.springframework.security.core.authority.AuthorityUtils;
import
org.springframework.security.web.authentication.AuthenticationSuccessHandler;

```

```

import org.springframework.stereotype.Component;

import jakarta.servlet.ServletException;
import jakarta.servlet.http.HttpServletRequest;
import jakarta.servlet.http.HttpServletResponse;

@Component

public class CustomAuthSuccessHandler implements AuthenticationSuccessHandler {

    private Principal p;

    @Override

    public void onAuthenticationSuccess(HttpServletRequest request,
    HttpServletResponse response,

        Authentication authentication) throws IOException,
    ServletException {

        Set<String> roles =
    AuthorityUtils.authorityListToSet(authentication.getAuthorities());

        if (roles.contains("ADMIN")) {

            response.sendRedirect("http://localhost:8080/HomePage/Home.html");

        }

        else {

            System.out.println("No redirection");

        }

    }

}

```

```
}  
  
}
```

HOME CONTROLLER

```
package com.example.controller;  
  
import java.security.Principal;  
  
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.PostMapping;  
  
import com.example.entity.User;  
import com.example.repository.UserRepo;  
import com.example.service.UserService;  
import com.fasterxml.jackson.annotation.JsonCreator.Mode;  
  
import jakarta.servlet.http.HttpSession;
```


@Controller

public class HomeController {

 @Autowired

 private UserService userService;

 @Autowired

 private UserRepo userRepo;

 @ModelAttribute

 public void commonUser(Principal p, Model m) {

 if (p != null) {

 String email = p.getName();

 User user = userRepo.findByUsername(email);

 m.addAttribute("user", user);

 }

 }

 @GetMapping("/")

 public String index() {

 return "index";

```
}
```

```
@GetMapping("/signin")
```

```
public String login() {
```

```
    return "login";
```

```
}
```

```
@PostMapping("/saveUser")
```

```
public String saveUser(@ModelAttribute User user, HttpSession session,  
Model m) {
```

```
    // System.out.println(user);
```

```
User u = userService.saveUser(user);
```

```
if (u != null) {
```

```
    // System.out.println("save sucess");
```

```
    session.setAttribute("msg", "Register successfully");
```

```
} else {
```

```
    // System.out.println("error in server");
```

```
    session.setAttribute("msg", "Something wrong server");
```

```
}
```

```
return "redirect:/register";
```

```
}  
  
}
```

USER REPO

```
package com.example.repository;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
  
import com.example.entity.User;  
  
public interface UserRepo extends JpaRepository<User, Integer> {  
  
    public User findByUsername(String email);  
  
}
```

APPLICATION PROPERTY

```
spring.datasource.url=jdbc:mysql://localhost:3306/Project_Emp_management  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.username=root
```

spring.datasource.password=123456

spring.jpa.hibernate.ddl-auto=update

spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect

server.port = 1080

spring.application.name=SecurityService

eureka.client.serviceUrl.defaultZone=http://localhost:8761/eureka

Main Eureka

APPLICATION

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;
import org.springframework.cloud.netflix.eureka.server.EnableEurekaServer;

@EnableEurekaServer
@SpringBootApplication
public class MainEurekaApplication {

    public static void main(String[] args) {

        SpringApplication.run(MainEurekaApplication.class, args);

    }

}
```

APPLICATION PROPERTY

`server.port=8761`

`eureka.client.register-with-eureka=false`

`eureka.client.fetch-registry=false`

FRONTEND CODES

Home Page (Logged Out)

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
"
    integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous"
    type="text/css">
  <style>
    /* Reset some default styles */
    body,
    h1,
    p {
      margin: 0;
      padding: 0;
    }

    /* Set full viewport height for the cover */
    html,
    body {
      height: 100%;
    }

    /* Cover container */
    .cover-container {
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100vh;
      background-size: cover;
      background-position: center;
      position: relative;
    }

    /* Overlay for the cover */
    .overlay {
      position: absolute;
      top: 0;
      left: 0;
      width: 100%;
      height: 100%;
      background: rgba(0, 0, 0, 0.5);
      /* Adjust the alpha value for the desired overlay darkness */
    }

    /* Text on the cover */
```



```

        .cover-text {
            color: #fff;
            /* Text color */
            text-align: center;
            z-index: 1;
        }

        /* Optional: Add more styles as needed */

        .login {
            position: fixed;
            /* Fixed position */
            right: 705px;
            /* Adjust the right position as needed */
            bottom: 245px;
            /* Adjust the bottom position as needed */
            z-index: 999;
            /* Set the z-index to ensure it's above other elements */
            padding: 10px 15px;
            /* Adjust padding as needed */
            background-color: #929AAA;
            /* Button background color */
            color: #fff;
            /* Button text color */
            border: none;
            /* Remove border */
            border-radius: 5px;
            /* Add border radius for rounded corners */
            cursor: pointer;

            box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2), 0 6px 20px 0
rgba(0,0,0,0.19);

            width: 10%;

            text-align: center;
        }
        .login:hover {
            background-color: transparent; /* Change background color on hover */
        }
    </style>
</head>

<body>
    <ul class="nav nav-tabs">
        <li class="nav-item">
            <a class="nav-link active" href="index.html">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="AboutUs.html">About Us</a>
        </li>
        <li class="nav-item">
            <a class="nav-link disabled"
href="Registration.html">Registration</a>
        </li>
        <li class="nav-item dropdown">
            <a class="login" href="http://localhost:1080/Home.html"
role="button" aria-haspopup="true"

```

```

        aria-expanded="false" id="login">LogIn</a>
    </li>
</div>
</ul>
<div class="cover-container"
    style="background-image: url('https://www.pixelstalk.net/wp-
content/uploads/2016/11/Free-Cover-Image.jpg');">
    <div class="overlay"></div>
    <div class="cover-text">
        <h1>SHWETANK CHAUDHARY</h1>
        <p>Employee Management Project for NEC Training.
            Please Login to Access the services.
        </p>
    </div>
</div>

<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
    integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
    crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
    integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
    crossorigin="anonymous"></script>
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
    integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmY1"
    crossorigin="anonymous"></script>
</body>
</html>

```

Home Page (After Login)

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
"
      integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous"
      type="text/css">

  <style>

    body,
    h1,
    p {
      margin: 0;
      padding: 0;
    }

    html,
    body {
      height: 100%;
    }

    .cover-container {
      display: flex;
      flex-direction: column;
      align-items: center;
      justify-content: center;
      height: 100%;
      background-size: cover;
      background-position: center;
      position: relative;
    }

    .overlay {
      position: absolute;
      top: 0;
      left: 0;
      width: 100%;
      height: 100%;
      background: rgba(0, 0, 0, 0.5);
    }

  </style>
</head>

<body>
  <div class="cover-container" style="height: 100%; position: relative;">
    <div class="overlay"></div>
    <div class="content"></div>
  </div>
</body>
</html>
```

```

        .cover-text {
            color: #fff;
            text-align: center;
            z-index: 1;
        }

        #logout {
            position: absolute;
            right: 30px;
            top: 2px;
        }
    </style>
</head>

<body>
    <ul class="nav nav-tabs">

        <li class="nav-item">
            <a class="navbar-brand" href="Home.html">
                
                </a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="AboutUs.html">About Us</a>
            </li>
            <li class="nav-item">
                <a class="nav-link"
href="./Registration.html">Registration</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="ADMIN_showemp.html">Show All
Emps</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="ContactUs.html" >Contact Us</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="http://localhost:8761/"
>Eureka</a>
            </li>

            <li class="nav-item">
                <a href="http://localhost:1080/logout"><button class="btn
btn-danger" id="logout">Logout</button></a>
            </li>
        </ul>
        <div class="cover-container"
            style="background-image:
url('https://images.pexels.com/photos/1408221/pexels-photo-
1408221.jpeg?auto=compress&cs=tinysrgb&w=1260&h=750&dpr=1');">
            <div class="overlay"></div>
            <div class="cover-text">
                <h1>YOU SIGNED IN AS ADMIN</h1>
                <p>You can access the servers through options in the Nav
bar...</p>
            </div>

```

```
</body>  
</html>
```

Registration (Insert Service)

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Registration</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css
"
        integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous"
        type="text/css">
  <style>
    body {
      font-family: 'Roboto', sans-serif;
      margin: 0;
      padding: 0;
      background-color: #edede9;
      color: #333;
    }

    form {
      max-width: 800px;
      margin: 20px auto;
      padding: 20px;
      background-color: #f5ebe0;
      border-radius: 10px;
      box-shadow: 0 0 16px rgba(0, 0, 0, 0.1);
    }

    .name_parent,
    .div2,
    .div3,
    .div4 {
      display: flex;
      justify-content: space-evenly;
      margin-bottom: 15px;
    }

    .page_heading {
      color: #bc6c25;
      text-align: center;
    }
  </style>
</head>

<body>
  <div class="page_heading">
    <h2>Registration</h2>
  </div>
  <div class="name_parent">
    <div class="div2">
      <input type="text" value="Name" />
    </div>
    <div class="div3">
      <input type="text" value="Email" />
    </div>
    <div class="div4">
      <input type="text" value="Phone" />
    </div>
  </div>
  <div class="div5">
    <input type="password" value="Password" />
  </div>
  <div class="div6">
    <input type="password" value="Confirm Password" />
  </div>
  <div class="div7">
    <input type="button" value="Register" />
  </div>
  <div class="div8">
    <input type="button" value="Cancel" />
  </div>
</body>
</html>
```

```

        input,select {
            width: 100%;
            padding: 8px;
            margin-right: 8px;
            margin-bottom: 15px;
            box-sizing: border-box;
            border: 1px solid #ccc;
            border-radius: 4px;
        }

        button {
            background-color: #283618;
            color: #fff;
            padding: 10px 15px;
            border: none;
            border-radius: 3px;
            cursor: pointer;
        }

        button:hover {
            background-color: #606c38;
        }

        .msg {
            text-align: center;
            color: green;
            margin-bottom: 15px;
        }

        #logout {
            position: absolute;
            right: 30px;
            top: 2px;
        }
    </style>
</head>

<body>
    <ul class="nav nav-tabs">

        <li class="nav-item">
            <a class="navbar-brand" href="Home.html">
                
                </a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="AboutUs.html">About Us</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="ADMIN_showemp.html">Show All
Emps</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="http://localhost:8761/"
>Eureka</a>

```

```

        </li>

        <li class="nav-item">
            <a href="http://localhost:1080/logout"><button class="btn btn-danger" id="logout">Logout</button></a>
        </li>
    </ul>
    <h1 class="page_heading">Add Employee</h1>

    <form id="employeeForm">
        <input type="number" id="id" name="id" placeholder="Employee Id"
required autocomplete="off" required>*<br>
        <input type="email" id="username" name="username"
placeholder="Email ID/UserName" required autocomplete="off" required>*<br>
        <input type="text" id="phonenumber" name="phonenumber"
placeholder="Mobile Number" required autocomplete="off" required>*<br>
        <div class="name_parent">
            <input class="f_name" type="text" id="f_name" name="f_name"
placeholder="First Name" required>*
            <input class="l_name" type="text" id="l_name" name="l_name"
placeholder="Last Name" required>*
        </div>

        <div class="div2">
            <input type="text" id="DOB" name="DOB" placeholder="Date Of
Birth" onfocus="(this.type='date')">
            <input type="text" id="position" name="position"
placeholder="Designation" required>*
        </div>

        <div class="div3">
            <input type="text" id="department" name="department"
placeholder="Department Name" required>*
            <input type="text" id="hire_date" name="hire_date"
placeholder="Hired Date" onfocus="(this.type='date')">
            <input type="text" id="salary" name="salary"
placeholder="Salary" required>*
        </div>

        <div class="div4">
            <input type="password" id="password" name="password"
placeholder="Password" required>*
            <input type="text" id="role" name="role" placeholder="Role"
required>*
        </div>

        <button type="button" onclick="addEmployee()">Add
Employee</button>

        <div class="msg" id="msg"></div>
    </form>

    <h2 id="load" class="load"></h2>

    <script>
        function addEmployee() {

            var employeeData = {

```



```

        id: document.getElementById("id").value,
        username: document.getElementById("username").value,
        first_name: document.getElementById("f_name").value,
        last_name: document.getElementById("l_name").value,

        phonenumber: document.getElementById("phonenumber").value,
        DOB: document.getElementById("DOB").value,
        position: document.getElementById("position").value,
        department:
document.getElementById("department").value,
        hire_date:
document.getElementById("hire_date").value,
        salary: document.getElementById("salary").value,
        password: document.getElementById("password").value,
        role: document.getElementById("role").value
    };

    console.log(JSON.stringify(employeeData));
    console.log(typeof(employeeData.DOB));

    var xhr = new XMLHttpRequest();
    xhr.open("POST", "http://localhost:8088/emp/add", true);
    xhr.setRequestHeader("Content-Type", "application/json");

    xhr.onload = function () {
        if (xhr.status === 201) {
            document.getElementById("msg").innerHTML =
"Employee Added Successfully";
        } else {
            document.getElementById("msg").innerHTML =
"Failed to add employee";
        }
    };

    xhr.send(JSON.stringify(employeeData));
    }
</script>
</body>
</html>

```

About Us

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>ABOUT PROJECT</title>
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
"
      integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous"
      type="text/css">

  <style>
    .header {
      padding: 60px;
      text-align: center;
      background: #1abc9c;
      color: white;
      font-size: 30px;
    }

    #logout {
      position: absolute;
      right: 30px;
      top: 2px;
    }

    .content {
      font-family: cursive;
    }
  </style>
</head>

<body>
  <ul class="nav nav-tabs">
    <!-- Logo -->
    <li class="nav-item">
      <a class="navbar-brand" href="Home.html">
        
      </a>
    </li>
    <li class="nav-item-active">
      <a class="nav-link" href="AboutUs.html">About Us</a>
    </li>
    <li class="nav-item">
      <a class="nav-link"
href="./Registration.html">Registration</a>
```

```

        </li>
        <li class="nav-item">
            <a class="nav-link" href="ADMIN_showemp.html">Show All
Emps</a>
        </li>
        <li class="nav-item">
            <a class="nav-link"
href="http://localhost:8761/">Eureka</a>
        </li>

        <li class="nav-item">
            <a href="http://localhost:1080/logout"><button class="btn
btn-danger" id="logout">Logout</button></a>
        </li>
    </ul>
    <div class="header">
        <h1>ABOUT PROJECT</h1>
    </div>
    <div class="content"><br><br>
    <h1><u> This Project was made by SHWETANK CHAUDHARY for NEC
TRAINING.</u></h1><br><br>
    <ul>
        <h2>All the services are only accessble if you sign in as
admin.<br>
        <b>Username: admin &nbsp; password: password</b>
    </h2>
    </ul>
    <ul>
        <h3>
            <ul>Insert Service:
                <h4>
                    <ul>Is in Registration Tab</ul>
                </h4>
            </h3>

            <h3>
            <ul>Delete Service:
                <h4>
                    <ul>Is in Show All Emps Tab -> Employee
Card</ul>
                </h4>
            </h3>

            <h3>
            <ul>Update Service:
                <h4>
                    <ul>Is in Show All Emps Tab -> Employee
Card</ul>
                </h4>
            </h3>

            <h3>
            <ul>Show/Display Service:
                <h4>
                    <ul>Is in Show All Emps Tab </ul>
                </h4>
            </h3>
    </ul>

```

```

        <ul>You can view a single employee card
using search option. Enter id of the employee to their
card.</ul>
    </h5>
</h3>
<h3>
    <ul>Eureka :
        <h4>
            <ul>Shows how many services are
running.</ul>
        </h4>
    </ul>
</h3>
</div>
</body>
</html>

```

Show All Employees (Show Service)

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>All Employee Details</title>
  <link href="ShowEmpCss.css" rel="stylesheet">
</head>

<body>

  <div class="topnav">
    <a class="active" href="Home.html">Home</a>
    <a href="AboutUs.html">About</a>
    <input id="search_employee" type="text" placeholder="Search
Employee By Id">
    <input type="button" onclick="searchEmployees()" value="Search">
  </div>
  <ol id="cardsContainer" style="--length: 5" role="list">
    <h1><strong>EMPLOYEES DETAILS</strong></h1>
    <p></p>
  </ol>
</div>
<script>
var number = 1;
var allEmployees = [];

function fetchempdata() {
  var xhr = new XMLHttpRequest();
  xhr.open("GET", "http://shwetank:8089/emp/show", true);
  xhr.onload = function () {
    if (xhr.status == 200) {
      var emps = JSON.parse(xhr.responseText);
      allEmployees = emps;
      displayData(emps);
    } else {
      console.error("Failed to fetch");
    }
  };

  xhr.send();
}

function searchEmployees() {
  var searchInput = document.getElementById('search_employee').value;
  var filteredEmployees = allEmployees.filter(function (employee) {
    var id = `${employee.id}`;
    return id.includes(searchInput);
  });

  displayData(filteredEmployees);
}
```

```

}

function displayData(emps) {
    var cardsContainer = document.getElementById("cardsContainer");
    cardsContainer.innerHTML = ''; // Clear existing content before
displaying new results

    emps.forEach(function (emp) {
        var card = document.createElement("div");
        card.className = "card";
        card.innerHTML = `
            <li style="--i: ${number++}">
                <h3>${emp.first_name} ${emp.last_name}</h3>
                <p><span class="e_id">ID: </span>${emp.id}</p>
                <p><span class="position">Position: </span>
${emp.position}</p>
                <p><span class="user_name">User Name:</span>
${emp.username}</p>
                <p><b><span class="salary">Salary:</span>
${emp.salary}</b></p>
                <p><span class="phone Number">Phone Number:</span>
${emp.phonenumber}</p>
                <p><span class="Hire_Date">Date of Joining:</span>
${emp.hire_date}</p>
                <p><span class="Role">Role: </span> ${emp.role}</p>
                <form action="DeleteEmp.html">
                    <input type="submit" value="Delete User"
class="delete">
                </form>
                <form action = "Update.html">
                    <input type="submit" value="Update User"
class="delete">
                </form>
            </li>
        `;
        cardsContainer.appendChild(card);
    });

    fetchempdata();
</script>

</body>

</html>

```

Delete Emp (Delete Service)

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css"
"
      integrity="sha384-
Gn5384xqQ1aoWXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm"
crossorigin="anonymous">
  <title>Delete Employee</title>
  <style>
    .msg h4{
      color:red;
    }
    .msg h3{
      color:green;
    }
    #logout {
      position: absolute;
      right: 30px;
      top: 2px;
    }
  </style>
</head>

<body>
  <ul class="nav nav-tabs">
    <li class="nav-item">
      <a class="nav-link active" href="Home.html">Home</a>
    </li>
    <li class="nav-item">
      <a class="nav-link" href="AboutUs.html">About Us</a>
    </li>
    <li class="nav-item">
      <a class="nav-link disabled"
href="Registration.html">Registration</a>
    </li>
    <li class="nav-item">
      <a href="http://localhost:1080/logout"><button class="btn
btn-danger" id="logout">Logout</button></a>
    </li>
  </ul>
  <div class="alert alert-danger" role="alert">
    <h4 class="alert-heading">Deleting Employee...</h4>
    <p>TO Confirm The deletion of the Employee Data.<br>Type the
Employee Id Below</p>
    <hr>
    <div class="input-group mb-3">
      <div class="input-group-prepend">
```

```

        <span class="input-group-text" id="inputGroup-sizing-
default">Employee Id</span>
    </div>
    <input type="text" class="form-control" aria-
label="Default" aria-describedby="inputGroup-sizing-default" id="Del_id">
    </div>
    <button type="submit" class="btn btn-outline-danger"
onclick="del()">Delete</button>
</div>
    <div class="msg" id="msg"></div>
    <script src="https://code.jquery.com/jquery-3.2.1.slim.min.js"
    integrity="sha384-
KJ3o2DKtIkvYIK3UENzmM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5KkN"
    crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js"
    integrity="sha384-
ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibX39j7fakFPskvXusvfa0b4Q"
    crossorigin="anonymous"></script>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js"
    integrity="sha384-
JZR6Spejh4U02d8jOt6vLEHfe/JQGiRRSQQxSfFWpi1MquVdAyjUar5+76PVCmYl"
    crossorigin="anonymous"></script>
    <script>

        function del() {
            var id = document.getElementById("Del_id").value;
            var xhr = new XMLHttpRequest();
            xhr.open("GET", 'http://shwetank:6060/emp/del/'+id, true);

            xhr.onload = function () {
                if (xhr.status == 202) {
                    document.getElementById("msg").innerHTML = "<h3>Employee
Deleted Successfully<h3>";
                } else {
                    document.getElementById("msg").innerHTML = "<h4>Employee
Failed to be Deleted. Kindly REFRESH and REnter the id or Go Back</h4>";
                }
            };

            xhr.send();
        }

    </script>
</body>
</html>

```


Update Employee (Update Service)

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Update Employee</title>
  <style>
    body {
      font-family: 'Roboto', sans-serif;
      margin: 0;
      padding: 0;
      background-color: #edede9;
      color: #333;
    }

    /* Navigation Styles */
    nav {
      background-color: rgb(3, 7, 192);
      overflow: hidden;
    }

    nav a {
      float: left;
      display: block;
      color: #dda15e;
      font-weight: 400;
      text-align: center;
      padding: 14px 16px;
      text-decoration: none;
    }

    nav a:hover {
      background-color: #ddd;
      color: black;
    }

    /* Form Container Styles */
    form {
      max-width: 800px;
      margin: 20px auto;
      padding: 20px;
      background-color: #f5ebe0;
      border-radius: 10px;
      box-shadow: 0 0 16px rgba(0, 0, 0, 0.1);
    }

    /* Sections Styling */
    .name_parent,
    .div2,
    .div3,
    .div4 {
```

```

        display: flex;
        justify-content: space-evenly;
        margin-bottom: 15px;
        /* Added margin for spacing */
    }

    /* Page Heading Styles */
    .page_heading {
        color: #bc6c25;
        text-align: center;
    }

    /* Input Styles */
    input,select {
        width: 100%;
        padding: 8px;
        margin-right: 8px;
        margin-bottom: 15px;
        box-sizing: border-box;
        border: 1px solid #ccc;
        border-radius: 4px;
    }

    /* Button Styles */
    button {
        background-color: #283618;
        color: #fff;
        padding: 10px 15px;
        border: none;
        border-radius: 3px;
        cursor: pointer;
    }

    button:hover {
        background-color: #606c38;
    }

    /* Message Styles */
    .msg {
        text-align: center;
        color: green;
        margin-bottom: 15px;
    }
}
</style>
</head>

<body>
    <h1 class="page_heading">Update Employee</h1>

    <form id="employeeForm">
        <input type="number" id="id" name="id" placeholder="Employee Id"
required autocomplete="off" required>*<br>
        <input type="email" id="username" name="username"
placeholder="Email ID/UserName" required autocomplete="off" required>*<br>
        <input type="text" id="phonenumner" name="phonenumner"
placeholder="Mobile Number" required autocomplete="off" required>*<br>
        <div class="name_parent">

```

```

        <input class="f_name" type="text" id="f_name" name="f_name"
placeholder="First Name" required>*
        <input class="l_name" type="text" id="l_name" name="l_name"
placeholder="Last Name" required>*
    </div>

    <div class="div2">
        <input type="text" id="DOB" name="DOB" placeholder="Date Of
Birth" onfocus="(this.type='date')">
        <input type="text" id="position" name="position"
placeholder="Designation" required>*
    </div>

    <div class="div3">
        <input type="text" id="department" name="department"
placeholder="Department Name" required>*
        <input type="text" id="hire_date" name="hire_date"
placeholder="Hired Date" onfocus="(this.type='date')">
        <input type="text" id="salary" name="salary"
placeholder="Salary" required>*
    </div>

    <div class="div4">
        <input type="password" id="password" name="password"
placeholder="Password" required>*
        <input type="text" id="role" name="role" placeholder="Role"
required>*
    </div>

    <button type="button" onclick="updateEmployee()">Update
Employee</button>

    <div class="msg" id="msg"></div>
</form>

<h2 id="load" class="load"></h2>

<script>
    function updateEmployee() {
        // Create an object with the employee data
        var employeeData = {
            id: document.getElementById("id").value,
            username: document.getElementById("username").value,
            first_name: document.getElementById("f_name").value,
            last_name: document.getElementById("l_name").value,

            phonenumber: document.getElementById("onenumber").value,
            DOB: document.getElementById("DOB").value,
            position: document.getElementById("position").value,
            department:
document.getElementById("department").value,
            hire_date:
document.getElementById("hire_date").value,
            salary: document.getElementById("salary").value,
            password: document.getElementById("password").value,
            role: document.getElementById("role").value
        };
        console.log(employeeData);
    }

```

```

        // Perform AJAX request to add the employee
        var xhr = new XMLHttpRequest();
        xhr.open("POST", "http://localhost:9696/emp/update", true);
        xhr.setRequestHeader("Content-Type", "application/json");

        xhr.onload = function () {
            if (xhr.status === 202) {
                document.getElementById("msg").innerHTML =
"Employee Updated Successfully";
            }
            else if (xhr.status === 500) {
                document.getElementById("msg").innerHTML =
"Employee Not Present.<br><a href='Registration.html'>Register User</a>";
            }
            else {
                document.getElementById("msg").innerHTML =
"Failed to Update employee. Reload and Retry";
            }
        };

        xhr.send(JSON.stringify(employeeData));
    }
</script>
</body>
</html>

```

Contact Us

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="UTF-8">
  <title>Contact Page</title>
  <style>
    *,
    *:before,
    *:after {
      box-sizing: border-box;
      -webkit-font-smoothing: antialiased;
      -moz-osx-font-smoothing: grayscale;
    }

    body {
      background: linear-gradient(to right, #eald6f 0%, #eb466b
100%);
      font-size: 12px;
    }

    body,
    button,
    input {
      font-family: 'Montserrat', sans-serif;
      font-weight: 700;
      letter-spacing: 1.4px;
    }

    .background {
      display: flex;
      min-height: 100vh;
    }

    .container {
      flex: 0 1 700px;
      margin: auto;
      padding: 10px;
    }

    .screen {
      position: relative;
      background: #3e3e3e;
      border-radius: 15px;
    }

    .screen:after {
      content: ' ';
      display: block;
      position: absolute;
      top: 0;
      left: 20px;
```

```

        right: 20px;
        bottom: 0;
        border-radius: 15px;
        box-shadow: 0 20px 40px rgba(0, 0, 0, .4);
        z-index: -1;
    }

    .screen-header {
        display: flex;
        align-items: center;
        padding: 10px 20px;
        background: #4d4d4f;
        border-top-left-radius: 15px;
        border-top-right-radius: 15px;
    }

    .screen-header-left {
        margin-right: auto;
    }

    .screen-header-button {
        display: inline-block;
        width: 8px;
        height: 8px;
        margin-right: 3px;
        border-radius: 8px;
        background: white;
    }

    .screen-header-button.close {
        background: #ed1c6f;
    }

    .screen-header-button.maximize {
        background: #e8e925;
    }

    .screen-header-button.minimize {
        background: #74c54f;
    }

    .screen-header-right {
        display: flex;
    }

    .screen-header-ellipsis {
        width: 3px;
        height: 3px;
        margin-left: 2px;
        border-radius: 8px;
        background: #999;
    }

    .screen-body {
        display: flex;
    }

    .screen-body-item {

```

```

        flex: 1;
        padding: 50px;
    }

    .screen-body-item.left {
        display: flex;
        flex-direction: column;
    }

    .app-title {
        display: flex;
        flex-direction: column;
        position: relative;
        color: #eald6f;
        font-size: 26px;
    }

    .app-title:after {
        content: '';
        display: block;
        position: absolute;
        left: 0;
        bottom: -10px;
        width: 25px;
        height: 4px;
        background: #eald6f;
    }

    .app-contact {
        margin-top: auto;
        font-size: 8px;
        color: #888;
    }

    .app-form-group {
        margin-bottom: 15px;
    }

    .app-form-group.message {
        margin-top: 40px;
    }

    .app-form-group.buttons {
        margin-bottom: 0;
        text-align: right;
    }

    .app-form-control {
        width: 100%;
        padding: 10px 0;
        background: none;
        border: none;
        border-bottom: 1px solid #666;
        color: #ddd;
        font-size: 14px;
        text-transform: uppercase;
        outline: none;
        transition: border-color .2s;
    }

```

```

}

.app-form-control::placeholder {
  color: #666;
}

.app-form-control:focus {
  border-bottom-color: #ddd;
}

.app-form-button {
  background: none;
  border: none;
  color: #eald6f;
  font-size: 14px;
  cursor: pointer;
  outline: none;
}

.app-form-button:hover {
  color: #b9134f;
}

.credits {
  display: flex;
  justify-content: center;
  align-items: center;
  margin-top: 20px;
  color: #ffa4bd;
  font-family: 'Roboto Condensed', sans-serif;
  font-size: 16px;
  font-weight: normal;
}

.credits-link {
  display: flex;
  align-items: center;
  color: #fff;
  font-weight: bold;
  text-decoration: none;
}

.dribbble {
  width: 20px;
  height: 20px;
  margin: 0 5px;
}

@media screen and (max-width: 520px) {
  .screen-body {
    flex-direction: column;
  }

  .screen-body-item.left {
    margin-bottom: 30px;
  }

  .app-title {

```



```

        flex-direction: row;
    }

    .app-title span {
        margin-right: 12px;
    }

    .app-title:after {
        display: none;
    }
}

@media screen and (max-width: 600px) {
    .screen-body {
        padding: 40px;
    }

    .screen-body-item {
        padding: 0;
    }
}
</style>
</head>

<body>
    <div class="background">
        <div class="container">
            <div class="screen">
                <div class="screen-header">
                    <div class="screen-header-left">
                        <div class="screen-header-button
close"></div>
                        <div class="screen-header-button
maximize"></div>
                        <div class="screen-header-button
minimize"></div>
                    </div>
                    <div class="screen-header-right">
                        <div class="screen-header-
ellipsis"></div>
                        <div class="screen-header-
ellipsis"></div>
                        <div class="screen-header-
ellipsis"></div>
                    </div>
                </div>
                <div class="screen-body">
                    <div class="screen-body-item left">
                        <div class="app-title">
                            <span>CONTACT</span>
                            <span>US</span>
                        </div>
                        <div class="app-contact">CONTACT INFO :
+62 81 314 928 595</div>
                    </div>
                    <div class="screen-body-item">
                        <div class="app-form">
                            <div class="app-form-group">

```

```

control" placeholder="NAME" >
control" placeholder="EMAIL">
control" placeholder="CONTACT NO">
message">
control" placeholder="MESSAGE">
buttons">
button">CANCEL</button>
button">SEND</button>
</div>
</div>
</div>
</div>
</div>
<div class="credits">
  <a class="credits-link"
href="http://localhost:8080/HomePage/Home.html">
    Home Page
  </a>
</div>
</div>
</div>
</body>
</html>

```

Login (Security Service)

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Login Page</title>

  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f8f9fa;
      margin: 0;
      padding: 0;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
      text-align: center;
    }

    section {
      width: 100%;
    }

    .container {
      max-width: 400px;
      margin: auto;
    }

    .card {
      border: 1px solid #ced4da;
      border-radius: 8px;
      box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1);
    }

    .card-header {
      background-color: #007bff;
      color: #fff;
      padding: 15px;
      border-bottom: 1px solid #ced4da;
      border-radius: 8px 8px 0 0;
    }

    .card-body {
      padding: 20px;
    }

    label {
      display: block;
      margin-bottom: 5px;
      font-weight: bold;
    }
```

```

    }

    input {
        width: 100%;
        padding: 8px;
        margin-bottom: 15px;
        box-sizing: border-box;
        border: 1px solid #ced4da;
        border-radius: 4px;
    }

    button {
        width: 100%;
        padding: 10px;
        background-color: #007bff;
        color: #fff;
        border: none;
        border-radius: 4px;
        cursor: pointer;
    }

    button:hover {
        background-color: #0056b3;
    }

    #logoutBtn {
        margin-top: 10px;
        background-color: #dc3545;
        margin-left: 50px;
        margin-right: auto;
    }

    #logoutBtn:hover {
        background-color: #c82333;
    }

    .hidden {
        display: none;
    }

    .ed_table {
        background-color: #F49CBB;
        margin-top: auto;
        padding: 10px;
        border: 1px solid #ced4da;
        border-radius: 8px;
        margin: 25px 0;
        font-family: cursive;
        min-width: 500px;
        min-height: 500px;
        box-shadow: 0 0 20px rgba(0, 0, 0, 0.15);
        font-size: large;
        margin-left: auto;
        margin-right: auto;
    }
}

```

```

        #Home{
            position: absolute;
            margin-left: -20px;
            margin-top: 30px;
        }
    </style>
</head>

<body>
    <section>
        <img src =
"https://drive.google.com/uc?id=1iFuzqzRuZ8NSsH9akJaVqbD3UM7I9Yh6"
height="400" width="400"><br>
        <a href="http://localhost:8080/HomePage"
id="Home"><button>Home</button></a>
    </section>
    <section>
        <div class="container p-3">
            <div class="row">
                <div class="col-md-6 offset-md-3">
                    <div class="card" id="loginCard">
                        <div class="card-header text-center fs-
4">Login Page</div>
                        <th:block th:if="{param.error}">
                            <p class="text-center text-danger
fs-5">

                                [{${session.SPRING_SECURITY_LAST_EXCEPTION.message}}]</p>
                        </th:block>
                        <th:block th:if="{param.logout}">
                            <p class="text-center text-success
fs-5">Logout Successfully</p>
                        </th:block>
                        <div class="card-body">
                            <form action="userLogin"
method="post">
                                <div class="mb-3">
                                    <label
for="username">Enter Email</label>
                                    <input type="text"
name="username" class="form-control" id="username" required>
                                </div>
                                <div class="mb-3">
                                    <label
for="password">Enter Password</label>
                                    <input type="password"
name="password" class="form-control" required>
                                </div>
                                <button type="button"
class="btn btn-primary col-md-12"
                                onclick="loginAndFetchData()">Login</button>
                            </form>
                        </div>
                    </div>
                </div>
            </div>
            <div id="employeeDetails" class="hidden"></div>
        </div>
    </section>
</body>
</html>

```

```

<a href="http://localhost:1080/signin?logout">
<button id="logoutBtn"
class="btn btn-danger col-md-12
hidden"

href="http://localhost:1080/signin?logout">Logout</button> </a>
</div>
</div>
</div>
</section>

<script>
function loginAndFetchData() {
    // Get the username from the input field
    var username = document.getElementById("username").value;

    // Perform AJAX request after login to fetch employee
details
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "http://localhost:8089/emp/show/" +
username, true);
    xhr.setRequestHeader("Content-Type", "application/json");

    xhr.onload = function () {
        if (xhr.status === 200) {
            var employeeDetails =
JSON.parse(xhr.responseText);
            if (employeeDetails.role === "ADMIN") {
                window.location.href =
"http://localhost:8080/HomePage/Home.html";
            }
            else{
                displayEmployeeDetails(employeeDetails);
            }

        } else {
            console.error("Failed to fetch employee
details.");
            document.getElementById("employeeDetails").innerHTML = "<p>Error
fetching employee details</p>";

        }
    };

    xhr.send();
}

function displayEmployeeDetails(employeeDetails) {
    // Display employee details on the frontend
    var detailsHtml = `
    <h1>Welcome ${employeeDetails.first_name}
${employeeDetails.last_name}</h1>
    <table class="ed_table">
    <tr>
    <th>ID</th>
    <td>${employeeDetails.id}</td>

```

```

        </tr>
        <tr>
            <th>User Name</th>
            <td>${employeeDetails.username}</td>
        </tr>
        <tr>
            <th>Phone Number</th>
            <td>${employeeDetails.phonenumber}</td>
        </tr>
        <tr>
            <th>Designation</th>
            <td>${employeeDetails.position}</td>
        </tr>
        <tr>
            <th>Department</th>
            <td>${employeeDetails.department}</td>
        </tr>
        <tr>
            <th>Joining Date</th>
            <td>${employeeDetails.hire_date}</td>
        </tr>
        <tr>
            <th>Salary</th>
            <td>${employeeDetails.salary}</td>
        </tr>
        <tr>
            <th>Role</th>
            <td>${employeeDetails.role}</td>
        </tr>
    </table>
`;

    // Update the HTML element with the employee details
    document.getElementById("employeeDetails").innerHTML =
detailsHtml;

    // Hide the login form and show employee details and logout
button

    document.getElementById("loginCard").classList.add("hidden");

    document.getElementById("employeeDetails").classList.remove("hidden");

    document.getElementById("logoutBtn").classList.remove("hidden");
    }

    function logout() {
        // Reset the UI to show the login form and hide employee
details and logout button

        document.getElementById("loginCard").classList.remove("hidden");

        document.getElementById("employeeDetails").classList.add("hidden");

        document.getElementById("logoutBtn").classList.add("hidden");
    }
</script>

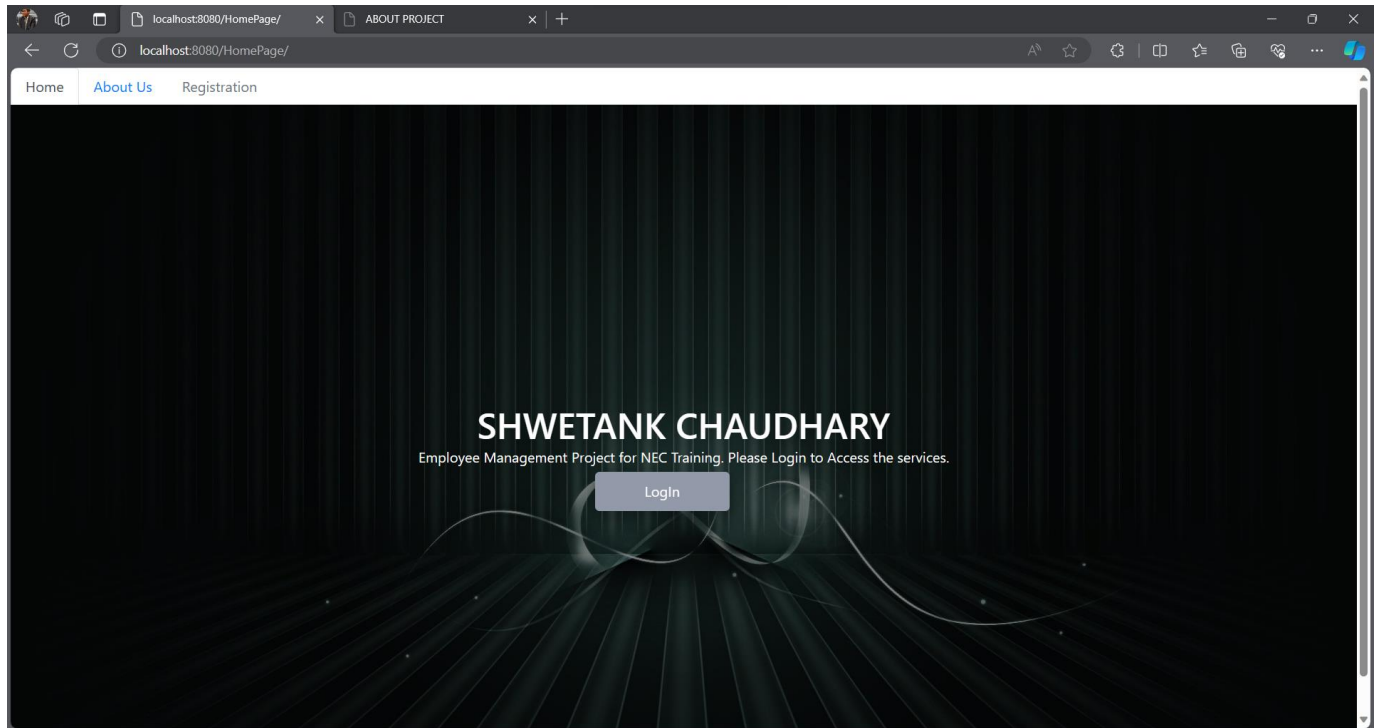
```

```
</body>
```

```
</html>
```


SNAPSHOTS OF OUTPUT

Home Page (Signed Out)



Login Page



Home

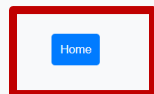
Login Page

Enter Email

Enter Password

Login

Login as User



Redirect to Home Page

Login Page

Enter Email

Pranay@gmail.com

Enter Password

.....

Login

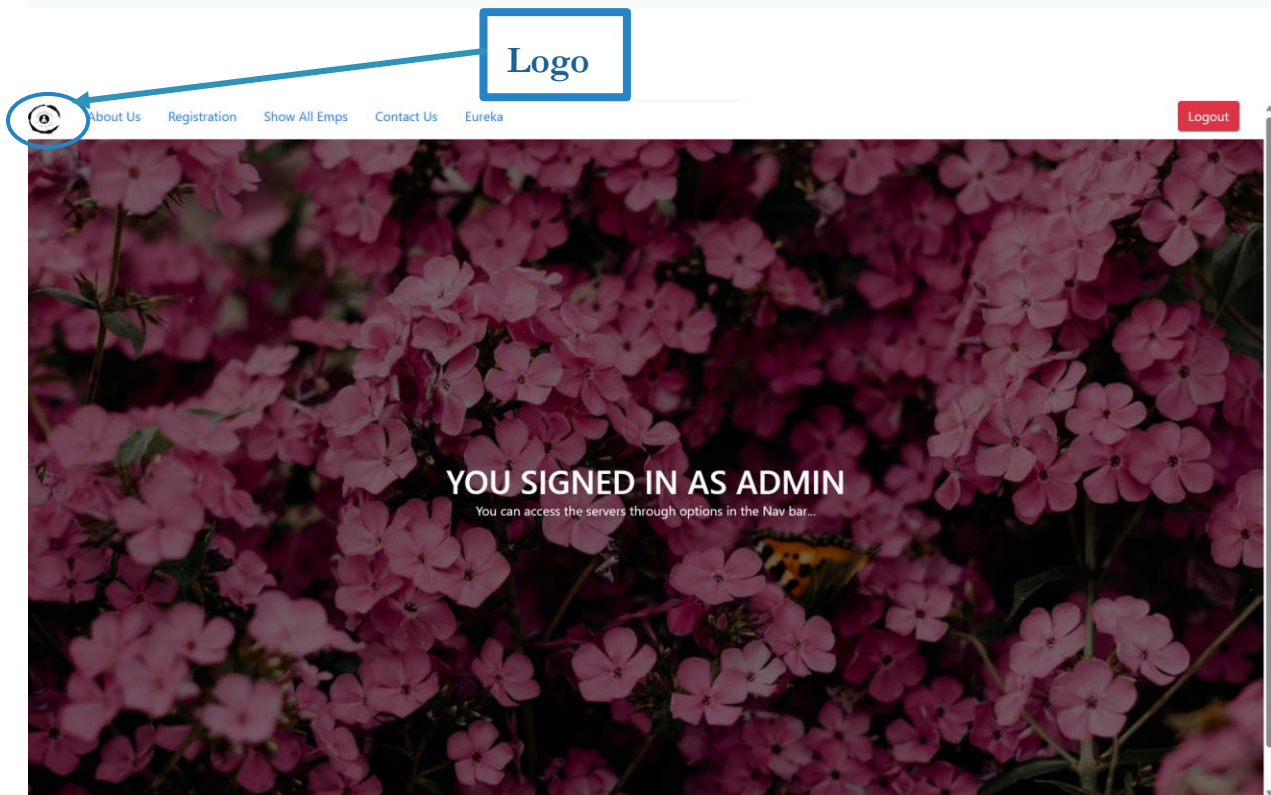
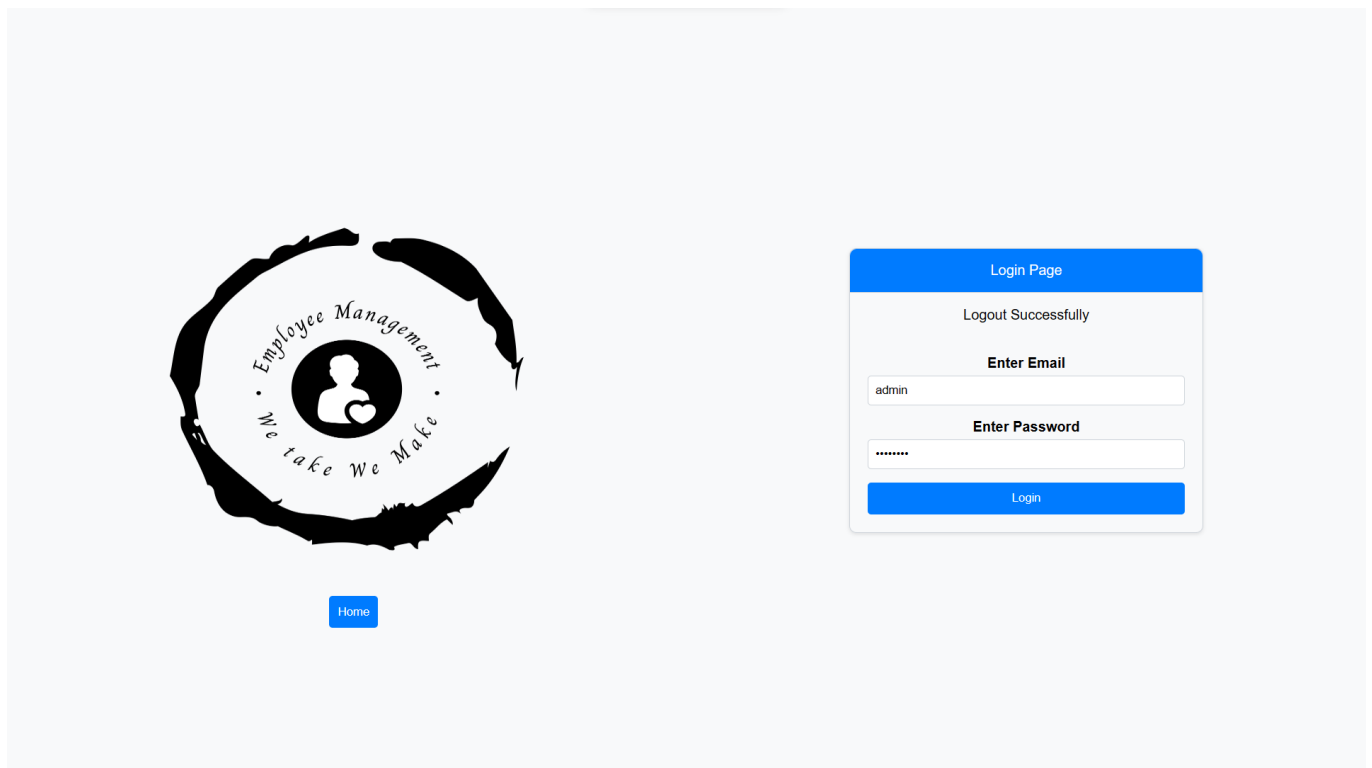


Welcome Pranay Yadav

ID	104
User Name	Pranay@gmail.com
Phone Number	09319673376
Designation	CEO
Department	CSE
Joining Date	2025-10-10
Salary	10000
Role	USER

Logout

Logging in as Admin



About Us

[About Us](#)[Registration](#)[Show All Emps](#)[Eureka](#)[Logout](#)

ABOUT PROJECT

This Project was made by SHWETANK CHAUDHARY for NEC TRAINING.

All the services are only accessible if you sign in as admin.

Username: admin password: password

Insert Service:

Is in Registration Tab

Delete Service:

Is in Show All Emps Tab -> Employee Card

Update Service:

Is in Show All Emps Tab -> Employee Card

Show/Display Service:

Is in Show All Emps Tab

You can view a single employee card using search option. Enter id of the employee to their card.

Eureka :

Shows how many services are running.

Registration

[Home](#) [About](#)

Search 112

No Employee of Id 112



[About Us](#) [Show All Emps](#) [Eureka](#)

[Logout](#)

Add Employee

112

*

Roy@gmail.com

*

09319673376

*

Abhishek

* Roy *

02/13/2001



* Lawyer *

Law

* 01/24/2024



* 100000 *

* USER *

Add Employee

Employee Added Successfully

After Registering

[Home](#) [About](#)

Search 112

1

Abhishek Roy

ID: 112
Position: Lawyer
User Name: Roy@gmail.com
Salary: 100000
Phone Number: 09319673376
Date of Joining: 2024-01-24
Role: USER

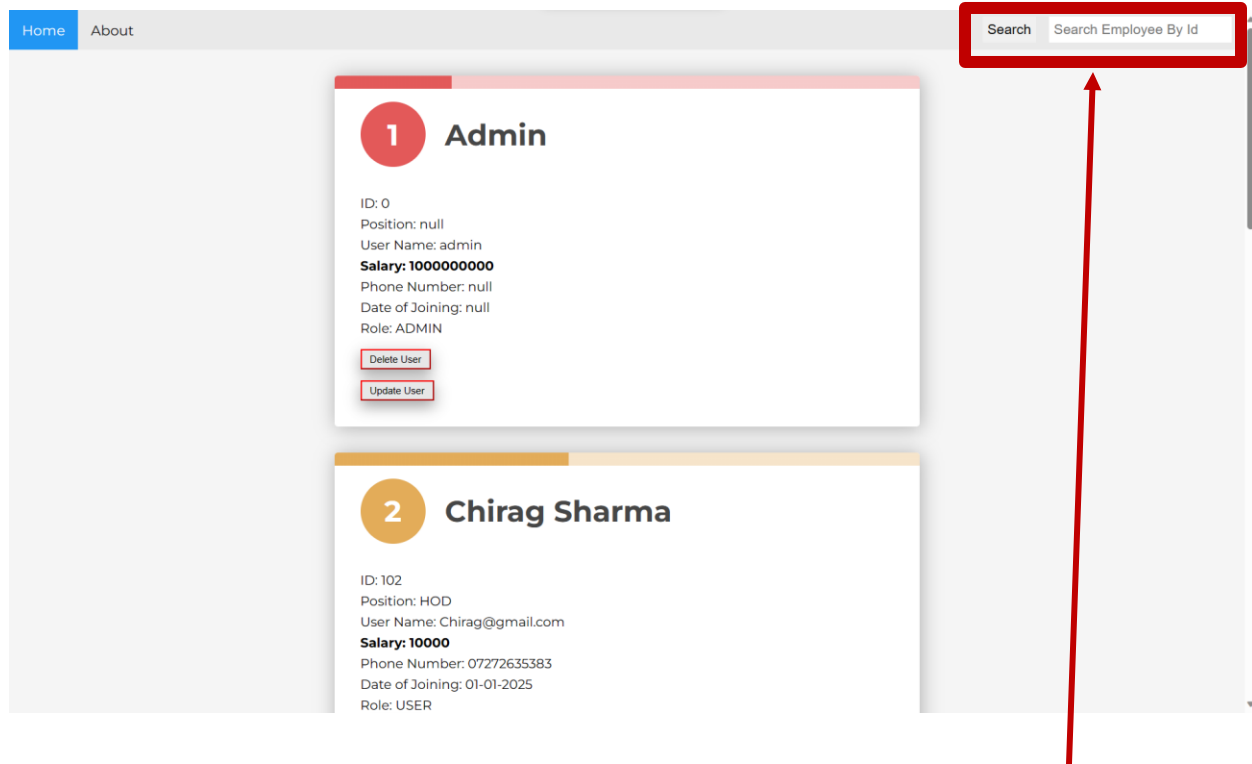
Delete User

Update User

Now The Employee Exists

Search Box. Give Single Employee as output when searched using Employee Id

Show All Employees



Search Box. Give Single Employee as output when searched using Employee Id



Shwetank Chaudhary

ID: 103

Position: CEO

User Name: shwetankchoudhary2@gmail.com

Salary: 10000

Phone Number: 09319673376

Date of Joining: 0015-12-15

Role: USER

Delete User

Update User

Both Delete and Update services are Accessed from here.

Update

3

Shwetank Chaudhary

ID: 103
Position: CEO
User Name: shwetankchoudhary2@gmail.com
Salary: 10000
Phone Number: 09319673376
Date of Joining: 0015-12-15
Role: USER

Delete User

Update User

Only First Name and Phone number will be changed in this example

Update Employee

103

*

Email ID/UserName

*

8527629951

*

Sanjay

* Last Name

mm/dd/yyyy

Designation

Department Name

* Hired Date

Salary

Password

* Role

Update Employee

Employee Updated Successfully



Sanjay Chaudhary

ID: 103

Position: CEO

User Name: shwetankchoudhary2@gmail.com

Salary: 10000

Phone Number: 8527629951

Date of Joining: 0015-12-15

Role: USER

Delete User

Update User

Name and Phone Number has been changed.

Result Grid													Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:	
	id	username	first_name	last_name	phonenumber	DOB	position	department	hire_date	salary	password	role								
	0	admin	Admin							1000000000	\$2a\$10\$49FMypZdFpWYsg...	ADMIN								
	102	Chirag@gmail.com	Chirag	Sharma	07272635383	08-18-2003	HOD	Mechanical Engineer	01-01-2025	10000	\$2a\$10\$9rop4B5KC1rX2yY9...	USER								
▶	103	shwetank@gmail.com	Sanjay	Chaudhary	8527629951		CEO	AIML	2024-01-10	100000	\$2a\$10\$YJix78BI.U2.WyYU...	USER								
	104	Pranay@gmail.com	Pranay	Yadav	09319673376		CEO	CSE	2025-10-10	10000	\$2a\$10\$d6QXMGwQwGNWF...	USER								
	105	Ayush@gmail.com	Devesh	Mani Tripathi	9819963897		CEO	VFX	2025-01-10	10000000	\$2a\$10\$ec0fkWgjfi/tOgJFen...	USER								
	112	Roy@gmail.com	Abhishek	Roy	09319673376		Lawyer	Law	2024-01-24	100000	\$2a\$10\$JFZznv9CDnEkbIko...	USER								
*																				

Delete

1

Sanjay Chaudhary

ID: 103

Position: CEO

User Name: shwetankchoudhary2@gmail.com

Salary: 10000

Phone Number: 8527629951

Date of Joining: 0015-12-15

Role: USER

Delete User

Update User

WHEN ENTERED ID IS WRONG

Home

About Us

Registration

Logout

Deleting Employee...

TO Confirm The deletion of the Employee Data.
Type the Employee Id Below

Employee Id

115

Delete

Employee Failed to be Deleted. Kindly REFRESH and REnter the id or Go Back

WHEN ENTERED ID IS RIGHT

Deleting Employee...

TO Confirm The deletion of the Employee Data.
Type the Employee Id Below

Employee Id 103

Delete

Employee Deleted Successfully

Result Grid													Filter Rows:		Edit:		Export/Import:		Wrap Cell Content:	
	id	username	first_name	last_name	phonenumber	DOB	position	department	hire_date	salary	password	role								
▶	0	admin	Admin		NULL	NULL	NULL	NULL	NULL	1000000000	\$2a\$10\$49FMypZdFpWYsg...	ADMIN								
	102	Chirag@gmail.com	Chirag	Sharma	07272635383	08-18-2003	HOD	Mechanical Engineer	01-01-2025	10000	\$2a\$10\$9rop4B5KC1rX2yY9...	USER								
	104	Pranay@gmail.com	Pranay	Yadav	09319673376		CEO	CSE	2025-10-10	10000	\$2a\$10\$d6QXMgwQwGNWF...	USER								
	105	Ayush@gmail.com	Devesh	Mani Tripathi	9819963897	NULL	CEO	VFX	2025-01-10	10000000	\$2a\$10\$ec0fkWgjfi/tOgJFen...	USER								
	112	Roy@gmail.com	Abhishek	Roy	09319673376	NULL	Lawyer	Law	2024-01-24	100000	\$2a\$10\$JFZznv9CDnEkbIko...	USER								
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL								

No Employee with Id 103 is there, because it has been successfully deleted.

Eureka

DS Replicas

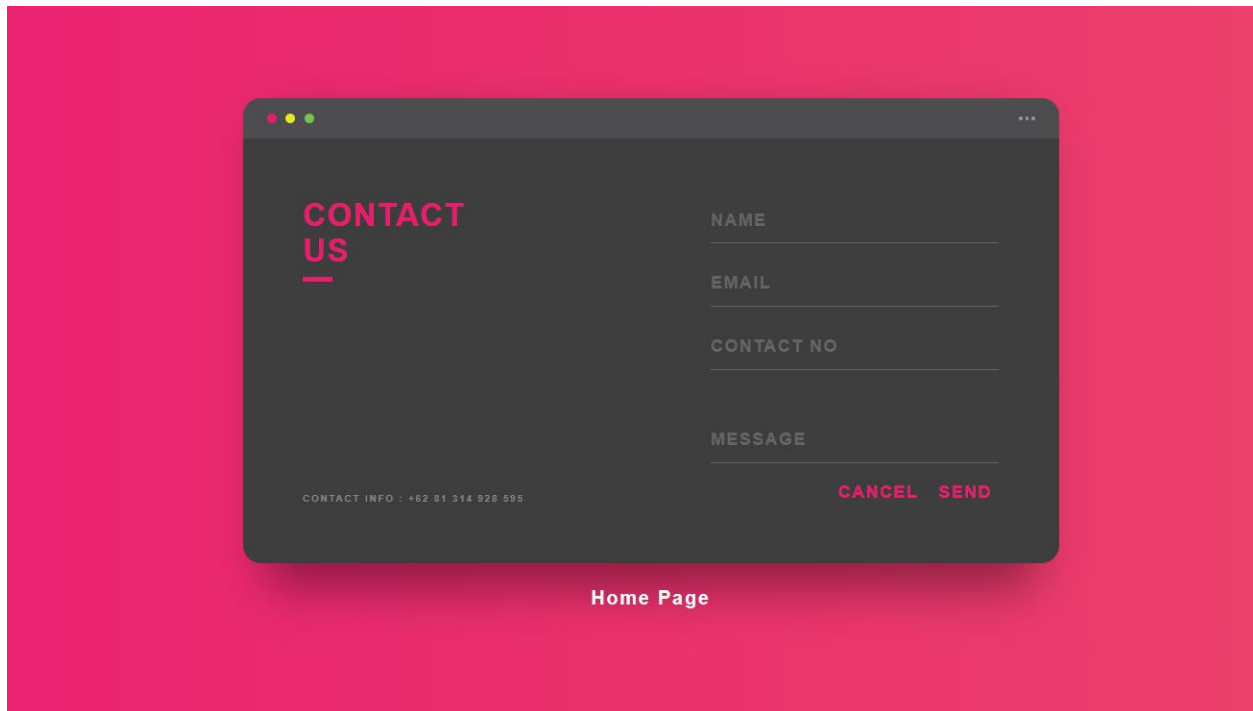
[localhost](#)

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
DELETESERVICE	n/a (1)	(1)	UP (1) - Shwetank:DeleteService:8091
INSERTSERVICE	n/a (1)	(1)	UP (1) - Shwetank:InsertService:8088
SECURITYSERVICE	n/a (1)	(1)	UP (1) - Shwetank:SecurityService:1080
SHOWSERVICE	n/a (1)	(1)	UP (1) - Shwetank:ShowService:8089
UPDATESERVICE	n/a (1)	(1)	UP (1) - Shwetank:UpdateService:9696

General Info

Contact Us



ERROR LOG

Error Encounter		Error Message	Error Solved		
Date	Time		Time	Date	
14/12/202	7:40p.m.	spring.cloud.gateway.routes[0].id is an unkn	7:50p.m.	14/12/2023	
14/12/202	10:00 p.m.	Failed to execute goal [32mo	10:34 p.m.	14/12/2023	
15/12/23	2:37pm	Error creating bean with name 'gatewayRout	9:00 AM	16/12/23	
		The dependencies of some of the beans in the application context form a cycle:			
		<pre> └─>┐ securityConfig(field private org.springframework.security.core.userdetails.UserDetailsService com.example.SecurityConfig.userDetail Ser vice) </pre>			
18/12/23	1:25 PM		9:32 PM	18/12/23	

REFERENCE/ BIBLIOGRAPHY

- <https://stackoverflow.com/questions/15512730/moving-a-text-45-pixel-up-using-css>
- StackOverFlow
- <https://www.geeksforgeeks.org/>
- <https://freefrontend.com/>
- <https://docs.spring.io/spring-security/reference/servlet/authentication/passwords/form.html>
- <https://docs.spring.io/spring-security>
- <https://www.baeldung.com/security-spring>
- <https://www.javatpoint.com/>
- <https://mvnrepository.com/>



THANK YOU