

## Mini Project - 3

Student Name/ID Number:	Chathushi Jayarathna
Unit Number and Title:	ACWD Module 5 – Web Development Foundations
Academic Year:	<b>2022/23</b>
Unit Assessor:	Ms. Arvinder Kaur
Project Title:	Development of a Registration Form in JSP with Database (Not Struts)
Issue Date:	22/09/2022
Submission Date:	19/10/2022
Internal Verifier Name:	
Date:	19/10/2022

### Learner declaration

I certify that the work submitted for this assignment is my own and that research sources are fully acknowledged.

Student signature: **Chathushi**

Date:17/10/2022

## 01) Create the Login System

- Controller

```
1 package abcjobs.controller;
2
3 import java.util.Map;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Controller;
7 import org.springframework.ui.Model;
8 import org.springframework.web.bind.annotation.ModelAttribute;
9 import org.springframework.web.bind.annotation.PostMapping;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RequestMethod;
12 import org.springframework.web.bind.annotation.RequestParam;
13 import org.springframework.web.servlet.ModelAndView;
14
15 import abcjobs.bean.User;
16 import abcjobs.service.UserService;
17
18 @Controller
19 public class MainController {
20     @Autowired
21     private UserService userService;
22
23
24     @RequestMapping("")
25     public ModelAndView signin() {
26         ModelAndView mav = new ModelAndView("home");
27         return mav;
28     }
29
30     @RequestMapping("/login")
31     public ModelAndView home() {
32         ModelAndView mav = new ModelAndView("login");
33         return mav;
34     }
35
36
37     @RequestMapping("/register")
38     public ModelAndView profile() {
39         ModelAndView mav = new ModelAndView("register");
40         return mav;
41     }
42
43     @RequestMapping("/update")
44     public ModelAndView updateprofile() {
45         ModelAndView mav = new ModelAndView("update");
46         return mav;
47     }
48
49     @RequestMapping("/pro")
50     public ModelAndView pro() {
51         ModelAndView mav = new ModelAndView("user-profile");
52         return mav;
53     }
54
55
56     @RequestMapping("/upro")
57     public ModelAndView upro() {
58         ModelAndView mav = new ModelAndView("edit-profile");
59         return mav;
60     }
61 }
```

```

61
62
63
64 @PostMapping("/handle-login")
65 public String checklogin(@ModelAttribute User user, Model model) {
66     User lUser = userService.findUserByEmail(user.getEmail());
67     if (lUser != null && user.getPassword().equals(lUser.getPassword())) {
68         model.addAttribute("message", "Welcome");
69         return "user-profile";
70
71     }else {
72
73     }
74     return "redirect:/login";
75 }
76
77
78
79
80
81 // @RequestMapping("/admin-panel")
82 // public ModelAndView adminhome() {
83 //     List<User> listUser = userService.listAll();
84 //     ModelAndView mav = new ModelAndView("AdminHome");
85 //     mav.addObject("listUser", listUser);
86 //     return mav;
87 // }
88
89 @RequestMapping("/signup")
90 public String adduser(Map<String, Object> model) {
91     User user = new User();
92     model.put("user", user);
93     return "register";
94 }
95
96 @RequestMapping(value = "/save", method = RequestMethod.POST)
97 public String saveUser(@ModelAttribute("user") User user) {
98     userService.save(user);
99     return "thank-you";
100 }

```

```

131
132 @RequestMapping(value = "/updated", method = RequestMethod.POST)
133 public String updateUser( @ModelAttribute("edituser") User edituser) {
134     userService.save(edituser);
135     return "redirect:/view-profile";
136 }
137
138 @RequestMapping("/edit")
139 public ModelAndView edituser(@RequestParam long id) {
140     ModelAndView mav = new ModelAndView("Update");
141     User user = userService.get(id);
142     mav.addObject("useredit", user);
143     return mav;
144 }
145 }
146

```

- JSP

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <script src="https://kit.fontawesome.com/848830b72b.js"
6   crossorigin="anonymous"></script>
7 <link rel="stylesheet"
8   href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
9   beta1/dist/css/bootstrap.min.css" />
10 <head>
11 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
12 <link href="resource/css/style.css" rel="stylesheet" />
13 <link href="resource/css/login.css" rel="stylesheet" />
14
15 <nav class="navbar navbar-expand-lg text-bg-light sticky-top">
16   <div class="container">
17     <a class="navbar-brand fw-bold fs-3"
18       href="<%=request.getContextPath()%>">ABC Jobs</a>
19     <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
20       data-bs-target="#navbarNav" aria-controls="navbarNav"
21       aria-expanded="false" aria-label="Toggle navigation">
22       <span class="navbar-toggler-icon"></span>
23     </button>
24     <div class="nav navbar-nav navbar-right" id="navbarNav">
25       <ul>
26         class="navbar-nav ms-auto me-0 me-lg-auto fw-semibold text-uppercase">
27         <li class="nav-item me-0 me-lg-3"><a class="nav-link"
28           href="<%=request.getContextPath()%>"> Home</a></li>
29         <li class="nav-item me-0 me-lg-3"><a class="nav-link"
30           href="search-user"> People</a></li>
31         <li class="nav-item me-0 me-lg-3"><a class="nav-link"
32           href="jobs"> Jobs</a></li>
33       </ul>
34       <div class="nav-button d-flex flex-column flex-lg-row">
35         <a href="login"
36           class="btn btn-outline-secondary me-0 me-lg-3 mb-3 mb-lg-0">Login</a>
37         <a href="register" class="btn btn-success mb-3 mb-lg-0">Register</a>
38       </div>
39     </div>
40   </div>
41 </nav>
42
43
44 </head>
45 <body>
46   <main>
47
48     <section id="login">
49
50       <div
51         class="hero-image mt-5 mt-md-5 d-flex justify-content-center col-sm-10">
52
53
54       <div class="container mt-5">
55
56         <div class="d-flex justify-content-end ">
57           
```

```

59     <div class="col-lg-6 col-md-8">
60         <div class="login-form">
61             <h2 class="fw-semibold">Login</h2>
62             <br>
63             <!-- Login Form -->
64             <form action="handle-login" method="post"
65                 class="mt-4 d-flex flex-column" id="form">
66                 <div class="form-group">
67                     <input id="loginEmail" class="form-control mb-3" type="text"
68                         placeholder="Email or phone" name="email" required />
69                 </div>
70
71                 <div class="form-group">
72                     <input id="loginPassword" class="form-control" type="password"
73                         placeholder="Password" name="password" required />
74                 </div>
75                 <br>
76                 <div class="forgot-password mt-3 mb-3 text-end">
77                     <a class="text-decoration-none" href="forgot-password">Forgot
78                         Password?</a>
79                 </div>
80
81                 <button class="btn btn-success" type="submit">Login</button>
82             </form>
83             <br> <br>
84             <p class="p-0 mt-3">
85                 Don't have an account? <a
86                     class="text-primary text-decoration-none" href="register">
87                     Register </a>
88             </p>
89         </div>
90     </div>
91 </div>
92 </section>
93 </main>
94
95 </body>
96 <footer class="text-center mt-2">
97     <p>&copy; ABC JOBS. All right reserved</p>
98 </footer>
99
100 <script
101     src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js">
102 </script>
103 </html>

```

- **Service**

```
1 package abcjobs.service;
2
3 import java.util.List;
4
5 import javax.transaction.Transactional;
6
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.stereotype.Service;
9
10 import abcjobs.bean.User;
11 import abcjobs.repository.LoginRepository;
12 import abcjobs.repository.UserRepository;
13
14 @Service
15 @Transactional
16 public class UserService {
17
18     @Autowired
19
20     private LoginRepository loginRepository;
21     public User findUserByEmail(String email) {
22         return loginRepository.findByEmail(email);
23     }
24
25     @Autowired
26     UserRepository repo;
27
28     public void save(User user) {
29         repo.save(user);
30     }
31
32     public List<User> listAll() {
33         return (List<User>) repo.findAll();
34     }
35
36
37     public User get(Long id) {
38         return repo.findById(id).get();
39     }
40
41     public void delete(Long id) {
42         repo.deleteById(id);
43     }
44
45     public List<User> search(String keyword) {
46         return repo.search(keyword);
47     }
48 }
49
50
```

- **Bean**

```
1 package abcjobs.bean;
2
3 import javax.persistence.Entity;
4 import javax.persistence.GeneratedValue;
5 import javax.persistence.GenerationType;
6 import javax.persistence.Id;
7
8 @Entity
9 public class User {
10     @Id
11     @GeneratedValue(strategy = GenerationType.IDENTITY)
12     private long id;
13     private String firstname;
14     private String lastname;
15     private String email;
16     private String password;
17     private String confirmPassword;
18     private String contact;
19     private String education;
20     private String city;
21     private String country;
22     private String workplace;
23     private String position;
24
25     public User() {
26
27     }
28
29     protected User(long id, String firstname, String lastname, String email, String
password, String confirmPassword,
30         String contact, String education, String city, String country, String
workplace, String position) {
31
32         this.id = id;
33         this.firstname = firstname;
34         this.lastname = lastname;
35         this.email = email;
36         this.password = password;
37         this.confirmPassword = confirmPassword;
38         this.contact = contact;
39         this.education = education;
40         this.city = city;
41         this.country = country;
42         this.workplace = workplace;
43         this.position = position;
44     }
45     public long getId() {
46         return id;
47     }
48     public void setId(long id) {
49         this.id = id;
50     }
51     public String getFirstname() {
52         return firstname;
53     }
54     public void setFirstname(String firstname) {
55         this.firstname = firstname;
56     }
57     public String getLastName() {
```

```

58     return lastname;
59 }
60 public void setLastname(String lastname) {
61     this.lastname = lastname;
62 }
63 public String getEmail() {
64     return email;
65 }
66 public void setEmail(String email) {
67     this.email = email;
68 }
69 public String getPassword() {
70     return password;
71 }
72 public void setPassword(String password) {
73     this.password = password;
74 }
75 public String getConfirmpassword() {
76     return confirmpassword;
77 }
78 public void setConfirmpassword(String confirmpassword) {
79     this.confirmpassword = confirmpassword;
80 }
81 public String getContact() {
82     return contact;
83 }
84 public void setContact(String contact) {
85     this.contact = contact;
86 }
87 public String getEducation() {
88     return education;
89 }
90 public void setEducation(String education) {
91     this.education = education;
92 }
93 public String getCity() {
94     return city;
95 }
96 public void setCity(String city) {
97     this.city = city;
98 }
99 public String getCountry() {
100     return country;
101 }
102 public void setCountry(String country) {
103     this.country = country;
104 }
105 public String getWorkplace() {
106     return workplace;
107 }
108 public void setWorkplace(String workplace) {
109     this.workplace = workplace;
110 }
111 public String getPosition() {
112     return position;
113 }
114 public void setPosition(String position) {
115     this.position = position;
116 }
117 }

```

```

118 @Override
119 public String toString() {
120     return "User [id=" + id + ", firstname=" + firstname + ", lastname=" + lastname +
121     ", email=" + email
122     + ", password=" + password + ", confirmpassword=" + confirmpassword + ",
123     contact=" + contact + ", education="
124     + education + ", city=" + city + ", country=" + country + ", workplace=" +
125     workplace + ", position="
126     + position + "]\n";
127 }
128 }

```



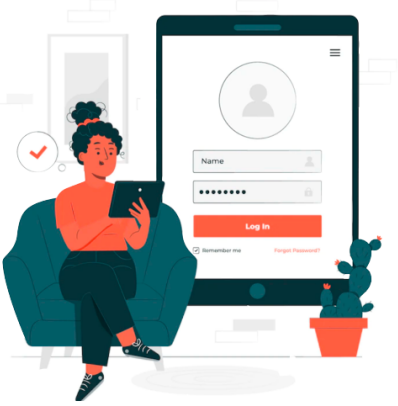
- **Repo**

```
1 package abcjobs.repository;  
2  
3 import org.springframework.data.jpa.repository.JpaRepository;  
4  
5 import abcjobs.bean.User;  
6  
7 public interface LoginRepository extends JpaRepository<User, String> {  
8  
9     User findByEmail(String email);  
10  
11 }  
12
```

- **Login Page**

ABC Jobs

HOMEPEOPLEJOBSLoginRegister



## Login

[Forgot Password?](#)

Login

Don't have an account? [Register](#)

© ABC JOBS. All right reserved

## 02) Create a Registration Page

- Controller

```
1 package abcjobs.controller;
2
3 import java.util.Map;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Controller;
7 import org.springframework.ui.Model;
8 import org.springframework.web.bind.annotation.ModelAttribute;
9 import org.springframework.web.bind.annotation.PostMapping;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RequestMethod;
12 import org.springframework.web.bind.annotation.RequestParam;
13 import org.springframework.web.servlet.ModelAndView;
14
15 import abcjobs.bean.User;
16 import abcjobs.service.UserService;
17
18 @Controller
19 public class MainController {
20     @Autowired
21     private UserService userService;
22
23
24     @RequestMapping("")
25     public ModelAndView signin() {
26         ModelAndView mav = new ModelAndView("home");
27         return mav;
28     }
29
30     @RequestMapping("/login")
31     public ModelAndView home() {
32         ModelAndView mav = new ModelAndView("login");
33         return mav;
34     }
35
36
37     @RequestMapping("/register")
38     public ModelAndView profile() {
39         ModelAndView mav = new ModelAndView("register");
40         return mav;
41     }
42
43     @RequestMapping("/update")
44     public ModelAndView updateprofile() {
45         ModelAndView mav = new ModelAndView("update");
46         return mav;
47     }
48
49     @RequestMapping("/pro")
50     public ModelAndView pro() {
51         ModelAndView mav = new ModelAndView("user-profile");
52         return mav;
53     }
54
55
56     @RequestMapping("/upro")
57     public ModelAndView upro() {
58         ModelAndView mav = new ModelAndView("edit-profile");
59         return mav;
60     }
61 }
```

```

61
62
63
64 @PostMapping("/handle-login")
65 public String checklogin(@ModelAttribute User user, Model model) {
66     User lUser = userService.findUserByEmail(user.getEmail());
67     if (lUser != null && user.getPassword().equals(lUser.getPassword())) {
68         model.addAttribute("message", "Welcome");
69         return "user-profile";
70     }
71     }else {
72     }
73 }
74 return "redirect:/login";
75 }
76
77
78
79
80
81 // @RequestMapping("/admin-panel")
82 // public ModelAndView adminhome() {
83 //     List<User> listUser = userService.listAll();
84 //     ModelAndView mav = new ModelAndView("AdminHome");
85 //     mav.addObject("listUser", listUser);
86 //     return mav;
87 // }
88
89 @RequestMapping("/signup")
90 public String adduser(Map<String, Object> model) {
91     User user = new User();
92     model.put("user", user);
93     return "register";
94 }
95
96 @RequestMapping(value = "/save", method = RequestMethod.POST)
97 public String saveUser(@ModelAttribute("user") User user) {
98     userService.save(user);
99     return "thank-you";
100 }

```

```

131
132 @RequestMapping(value = "/updated", method = RequestMethod.POST)
133 public String updateUser( @ModelAttribute("edituser") User edituser) {
134     userService.save(edituser);
135     return "redirect:/view-profile";
136 }
137
138 @RequestMapping("/edit")
139 public ModelAndView edituser(@RequestParam long id) {
140     ModelAndView mav = new ModelAndView("Update");
141     User user = userService.get(id);
142     mav.addObject("useredit", user);
143     return mav;
144 }
145 }
146

```

- JSP

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <script src="https://kit.fontawesome.com/848830b72b.js" crossorigin="anonymous">
6 </script>
7 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
8   beta1/dist/css/bootstrap.min.css"/>
9 <link href="resource/css/style.css" rel="stylesheet" />
10 <link href="resource/css/register.css" rel="stylesheet" />
11 <head>
12 <nav class="navbar navbar-expand-lg text-bg-light sticky-top">
13   <div class="container">
14     <a class="navbar-brand fw-bold fs-3" href="<%= request.getContextPath() %>">ABC
15     Jobs</a>
16     <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-
17       target="#navbarNav" aria-controls="navbarNav" aria-expanded="false" aria-
18       label="Toggle navigation">
19       <span class="navbar-toggler-icon"></span>
20     </button>
21     <div class="nav navbar-nav navbar-right" id="navbarNav">
22       <ul class="navbar-nav ms-auto me-0 me-lg-auto fw-semibold text-uppercase">
23         <li class="nav-item me-0 me-lg-3">
24           <a class="nav-link" href="<%= request.getContextPath() %>"> Home</a>
25         </li>
26         <li class="nav-item me-0 me-lg-3">
27           <a class="nav-link" href="search-user"> People</a>
28         </li>
29         <li class="nav-item me-0 me-lg-3">
30           <a class="nav-link" href="jobs"> Jobs</a>
31         </li>
32       </ul>
33       <div class="nav-button d-flex flex-column flex-lg-row">
34         <a href="login" class="btn btn-outline-secondary me-0 me-lg-3 mb-3 mb-lg-
35         0">Login</a>
36         <a href="register" class="btn btn-success mb-3 mb-lg-0">Register</a>
37       </div>
38     </div>
39   </nav>
40 </head>
41 <body>
42   <main>
43     <div class="container mt-5 mb-5">
44       <form action="save" method="post" id="form" onsubmit="return
45       checkValidation()">
46       <div class="row justify-content-left">
47         <div class="col-lg-5 col-md-8">
48           <div class="d-flex flex-column regis-wrapper">
49             <div class="heading">
50               <h2>Register</h2>
51             </div>
52           </div>
53         </div>
54       </div>
55     </div>
56   </main>
57 </body>
58 </html>
```

```

53         <!-- First Name -->
54         <div class="form-group">
55             <label for="firstname">First Name</label>
56             <input type="text" name="firstname" class="form-control"
57 id="firstname" required/>
58         </div>
59         <p class="error errorFname">First Name is required</p>
60
61         <!-- Last Name -->
62         <div class="form-group">
63             <label for="lastname">Last Name</label>
64             <input type="text" name="lastname" class="form-control"
65 id="lastname" required/>
66         </div>
67         <p class="error errorLname">Last Name is required</p>
68
69         <!-- Email -->
70         <div class="form-group">
71             <label for="email">Email</label>
72             <input type="text" name="email" class="form-control" id="email"
73 required/>
74         </div>
75         <p class="error errorEmail">Email is required</p>
76
77         <!-- Password -->
78         <div class="form-group">
79             <label for="password">Password</label>
80             <input type="password" name="password" class="form-control"
81 id="password" required/>
82         </div>
83         <p class="error errorPassword">Password is required</p>
84
85         <!-- Password -->
86         <div class="form-group">
87             <label for="confirmpassword">Repeat Password</label>
88             <input type="password" name="confirmpassword" class="form-
89 control" id="confirmpassword" required/>
90         </div>
91         <p class="error errorPassword">Password is required</p>
92
93         <button type="submit" class="btn btn-success mt-
94 4">Register</button>
95
96         <div class="wrapper">
97             <div class="hero-image mt-5 mt-md-5 d-flex justify-content-center col-sm-
98 20">
99                 
101             </div>
102         </div>
103     </form>
104 </div>

```

```

105     </main>
106
107
108
109
110     <script src="js/validator.js"></script>
111
112 </body>
113
114 <footer class="text-center mt-2">
115     <p>&copy; ABC JOBS. All right reserved</p>
116 </footer>
117
118 <script
119 src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0/dist/js/bootstrap.bundle.min.js">
120 </script>
121 </html>

```

- **Service**

```
1 package abcjobs.service;
2
3 import java.util.List;
4
5 import javax.transaction.Transactional;
6
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.stereotype.Service;
9
10 import abcjobs.bean.User;
11 import abcjobs.repository.LoginRepository;
12 import abcjobs.repository.UserRepository;
13
14 @Service
15 @Transactional
16 public class UserService {
17
18     @Autowired
19
20     private LoginRepository loginRepository;
21     public User findUserByEmail(String email) {
22         return loginRepository.findByEmail(email);
23     }
24
25     @Autowired
26     UserRepository repo;
27
28     public void save(User user) {
29         repo.save(user);
30     }
31
32     public List<User> listAll() {
33         return (List<User>) repo.findAll();
34     }
35
36
37     public User get(Long id) {
38         return repo.findById(id).get();
39     }
40
41     public void delete(Long id) {
42         repo.deleteById(id);
43     }
44
45     public List<User> search(String keyword) {
46         return repo.search(keyword);
47     }
48 }
49
50
```

- **Bean**

```
1 package abcjobs.bean;
2
3 import javax.persistence.Entity;
4 import javax.persistence.GeneratedValue;
5 import javax.persistence.GenerationType;
6 import javax.persistence.Id;
7
8 @Entity
9 public class User {
10     @Id
11     @GeneratedValue(strategy = GenerationType.IDENTITY)
12     private long id;
13     private String firstname;
14     private String lastname;
15     private String email;
16     private String password;
17     private String confirmpassword;
18     private String contact;
19     private String education;
20     private String city;
21     private String country;
22     private String workplace;
23     private String position;
24
25     public User() {
26
27     }
28
29     protected User(long id, String firstname, String lastname, String email, String
password, String confirmpassword,
30         String contact, String education, String city, String country, String
workplace, String position) {
31
32         this.id = id;
33         this.firstname = firstname;
34         this.lastname = lastname;
35         this.email = email;
36         this.password = password;
37         this.confirmpassword = confirmpassword;
38         this.contact = contact;
39         this.education = education;
40         this.city = city;
41         this.country = country;
42         this.workplace = workplace;
43         this.position = position;
44     }
45     public long getId() {
46         return id;
47     }
48     public void setId(long id) {
49         this.id = id;
50     }
51     public String getFirstname() {
52         return firstname;
53     }
54     public void setFirstname(String firstname) {
55         this.firstname = firstname;
56     }
57     public String getLastName() {
```

```

58     return lastname;
59 }
60 public void setLastname(String lastname) {
61     this.lastname = lastname;
62 }
63 public String getEmail() {
64     return email;
65 }
66 public void setEmail(String email) {
67     this.email = email;
68 }
69 public String getPassword() {
70     return password;
71 }
72 public void setPassword(String password) {
73     this.password = password;
74 }
75 public String getConfirmpassword() {
76     return confirmpassword;
77 }
78 public void setConfirmpassword(String confirmpassword) {
79     this.confirmpassword = confirmpassword;
80 }
81 public String getContact() {
82     return contact;
83 }
84 public void setContact(String contact) {
85     this.contact = contact;
86 }
87 public String getEducation() {
88     return education;
89 }
90 public void setEducation(String education) {
91     this.education = education;
92 }
93 public String getCity() {
94     return city;
95 }
96 public void setCity(String city) {
97     this.city = city;
98 }
99 public String getCountry() {
100     return country;
101 }
102 public void setCountry(String country) {
103     this.country = country;
104 }
105 public String getWorkplace() {
106     return workplace;
107 }
108 public void setWorkplace(String workplace) {
109     this.workplace = workplace;
110 }
111 public String getPosition() {
112     return position;
113 }
114 public void setPosition(String position) {
115     this.position = position;
116 }
117

```

```

118 @Override
119 public String toString() {
120     return "User [id=" + id + ", firstname=" + firstname + ", lastname=" + lastname +
121     ", email=" + email
122     + ", password=" + password + ", confirmpassword=" + confirmpassword + ",
123     contact=" + contact + ", education="
124     + education + ", city=" + city + ", country=" + country + ", workplace=" +
125     workplace + ", position="
126     + position + "]\n";
127 }
128

```



- **Repo**

```
1 package abcjobs.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.Query;
6 import org.springframework.data.repository.CrudRepository;
7 import org.springframework.data.repository.query.Param;
8
9 import abcjobs.bean.User;
10
11 public interface UserRepository extends CrudRepository<User, Long> {
12
13
14     @Query(value = "SELECT u FROM User u WHERE u.firstname LIKE '%' || :keyword || '%"
15         + " OR u.email LIKE '%' || :keyword || '%"")
16     public List<User> search(@Param("keyword") String keyword);
17
18 }
```

- **Registration Page**

ABC Jobs

HOME PEOPLE JOBS Login Register

Register

First Name

chathushi

Last Name

Dilhari

Email

chathudil77@gmail.com

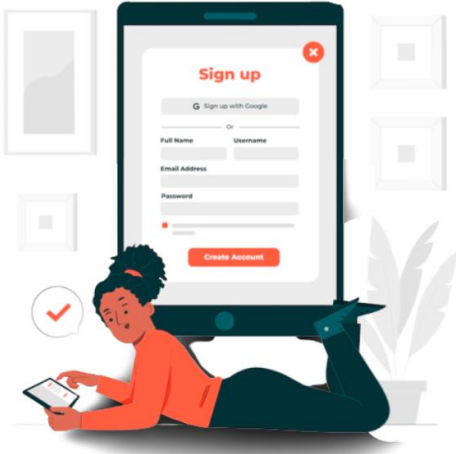
Password

\*\*\*\*\*

Repeat Password

\*\*\*\*\*

Register



The illustration shows a person with dark skin and curly hair, wearing an orange long-sleeved shirt and dark pants, lying on their stomach on a light-colored floor. They are holding a tablet computer. In the background, there is a large, stylized screen displaying a 'Sign up' form. The form has fields for 'Full Name', 'Username', 'Email Address', and 'Password', along with a 'Create Account' button. A red 'X' icon is in the top right corner of the screen. To the left of the person, there is a small circular icon with a checkmark. The background also features some framed pictures on the wall and a potted plant.

© ABC JOBS. All right reserved

### 03)Create an Update Profile Page

- **Controller**

```
1 package abcjobs.controller;
2
3 import java.util.Map;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Controller;
7 import org.springframework.ui.Model;
8 import org.springframework.web.bind.annotation.ModelAttribute;
9 import org.springframework.web.bind.annotation.PostMapping;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RequestMethod;
12 import org.springframework.web.bind.annotation.RequestParam;
13 import org.springframework.web.servlet.ModelAndView;
14
15 import abcjobs.bean.User;
16 import abcjobs.service.UserService;
17
18 @Controller
19 public class MainController {
20     @Autowired
21     private UserService userService;
22
23
24     @RequestMapping("")
25     public ModelAndView signin() {
26         ModelAndView mav = new ModelAndView("home");
27         return mav;
28     }
29
30     @RequestMapping("/login")
31     public ModelAndView home() {
32         ModelAndView mav = new ModelAndView("login");
33         return mav;
34     }
35
36
37     @RequestMapping("/register")
38     public ModelAndView profile() {
39         ModelAndView mav = new ModelAndView("register");
40         return mav;
41     }
42
43     @RequestMapping("/update")
44     public ModelAndView updateprofile() {
45         ModelAndView mav = new ModelAndView("update");
46         return mav;
47     }
48
49     @RequestMapping("/pro")
50     public ModelAndView pro() {
51         ModelAndView mav = new ModelAndView("user-profile");
52         return mav;
53     }
54
55
56     @RequestMapping("/upro")
57     public ModelAndView upro() {
58         ModelAndView mav = new ModelAndView("edit-profile");
59         return mav;
60     }
61 }
```

-

•

```
61
62
63
64 @PostMapping("/handle-login")
65 public String checklogin(@ModelAttribute User user, Model model) {
66     User lUser = userService.findUserByEmail(user.getEmail());
67     if (lUser != null && user.getPassword().equals(lUser.getPassword())) {
68         model.addAttribute("message", "Welcome");
69         return "user-profile";
70     }
71     }else {
72     }
73     }
74     return "redirect:/login";
75 }
76
77
78
79
80
81 // @RequestMapping("/admin-panel")
82 // public ModelAndView adminhome() {
83 //     List<User> listUser = userService.listAll();
84 //     ModelAndView mav = new ModelAndView("AdminHome");
85 //     mav.addObject("listUser", listUser);
86 //     return mav;
87 // }
88
89 @RequestMapping("/signup")
90 public String adduser(Map<String, Object> model) {
91     User user = new User();
92     model.put("user", user);
93     return "register";
94 }
95
96 @RequestMapping(value = "/save", method = RequestMethod.POST)
97 public String saveUser(@ModelAttribute("user") User user) {
98     userService.save(user);
99     return "thank-you";
100 }
```

```
131
132 @RequestMapping(value = "/updated", method = RequestMethod.POST)
133 public String updateUser( @ModelAttribute("edituser") User edituser) {
134     userService.save(edituser);
135     return "redirect:/view-profile";
136 }
137
138 @RequestMapping("/edit")
139 public ModelAndView edituser(@RequestParam long id) {
140     ModelAndView mav = new ModelAndView("Update");
141     User user = userService.get(id);
142     mav.addObject("useredit", user);
143     return mav;
144 }
145 }
146
```

- JSP

```
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <script src="https://kit.fontawesome.com/848830b72b.js"
6   crossorigin="anonymous"></script>
7 <link rel="stylesheet"
8   href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-
9   beta1/dist/css/bootstrap.min.css" />
10 <link href="resource/css/style.css" rel="stylesheet" />
11 <link href="resource/css/user-profile.css" rel="stylesheet" />
12 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
13 <head>
14 <nav class="navbar navbar-expand-lg text-bg-light sticky-top">
15   <div class="container">
16     <a class="navbar-brand fw-bold fs-3"
17       href="<%=request.getContextPath()%>">ABC Jobs</a>
18     <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
19       data-bs-target="#navbarNav" aria-controls="navbarNav"
20       aria-expanded="false" aria-label="Toggle navigation">
21       <span class="navbar-toggler-icon"></span>
22     </button>
23     <div class="nav navbar-nav navbar-right" id="navbarNav">
24       <ul
25         class="navbar-nav ms-auto me-0 me-lg-auto fw-semibold text-uppercase">
26         <li class="nav-item me-0 me-lg-3"><a class="nav-link"
27           href="<%=request.getContextPath()%>"> Home</a></li>
28         <li class="nav-item me-0 me-lg-3"><a class="nav-link"
29           href="search-user"> People</a></li>
30         <li class="nav-item me-0 me-lg-3"><a class="nav-link"
31           href="jobs"> Jobs</a></li>
32         </ul>
33         <div class="nav-button d-flex flex-column flex-lg-row">
34           <a href="login"
35             class="btn btn-outline-secondary me-0 me-lg-3 mb-3 mb-lg-0">Login</a>
36           <a href="register" class="btn btn-success mb-3 mb-lg-0">Register</a>
37         </div>
38       </div>
39     </div>
40 </nav>
41
42 </head>
43 <body>
44
45
46 <main>
47   <section id="user-profile">
48     <div class="container">
49       <div class="d-flex mt-5">
50         <!-- profile -->
51         <main class="me-3 main-profile">
52           <div class="profile">
53             <div class="banner"></div>
54
55             <div class="p-4 pt-5 position-relative">
56               <div class="profile-picture position-absolute">
57                 
```

```

59         </div>
60
61     <div class="d-flex justify-content-between">
62         <h3 class="name"></h3>
63         <!-- Edit -->
64         <div class="d-flex flex-column">
65             <a class="btn btn-success" href="upro">Edit Profile</a> <br>
66
67         </div>
68
69     </div>
70
71 </div>
72
73 <div class="text-center">
74     <h2>
75         <%
76         String name = request.getParameter("email");
77         out.println(" Hi welcome " + name);
78         %>
79     </h2>
80
81     <br>
82 </div>
83 <section class="section about-section gray-bg" id="about">
84     <div class="container">
85         <div class="row align-items-center flex-row-reverse">
86             <div class="col-lg-9">
87                 <div
88                     class="about-text go-to">
89                     <h3 class="dark-color">Full Name</h3>
90
91                 <section class="about">
92                     <div class="d-flex justify-content-between">
93                         <h5>About</h5>
94
95                 </div>
96                 <h6 class="theme-color lead">A Lead UX & UI
97                     designer based in Canada</h6>
98                 <p>
99                     I
100                     <mark>design and develop</mark>
101                     services for customers of all sizes, specializing in
102                     creating stylish, modern websites, web services and
103                     online stores. My passion is to design digital user
104                     experiences through the bold interface and meaningful
105                     interactions.
106                 </p>
107                 </section>
108                 <br> <br>
109                 <div class="col-md-8">
110                     <div class="card mb-3">
111                         <div class="card-body">
112                             <div class="row">
113                                 <div class="col-sm-3">
114                                     <h6 class="mb-0">Full Name</h6>
115                                 </div>
116                                 <div class="col-sm-9 text-secondary">Full
117 Name</div>

```

- **Service**

```
1 package abcjobs.service;
2
3 import java.util.List;
4
5 import javax.transaction.Transactional;
6
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.stereotype.Service;
9
10 import abcjobs.bean.User;
11 import abcjobs.repository.LoginRepository;
12 import abcjobs.repository.UserRepository;
13
14 @Service
15 @Transactional
16 public class UserService {
17
18     @Autowired
19     private LoginRepository loginRepository;
20     public User findUserByEmail(String email) {
21         return loginRepository.findByEmail(email);
22     }
23
24     @Autowired
25     UserRepository repo;
26
27     public void save(User user) {
28         repo.save(user);
29     }
30
31     public List<User> listAll() {
32         return (List<User>) repo.findAll();
33     }
34
35     public User get(Long id) {
36         return repo.findById(id).get();
37     }
38
39     public void delete(Long id) {
40         repo.deleteById(id);
41     }
42
43     public List<User> search(String keyword) {
44         return repo.search(keyword);
45     }
46 }
47
48
49
50
```

- **Bean**

```
1 package abcjobs.bean;
2
3 import javax.persistence.Entity;
4 import javax.persistence.GeneratedValue;
5 import javax.persistence.GenerationType;
6 import javax.persistence.Id;
7
8 @Entity
9 public class User {
10     @Id
11     @GeneratedValue(strategy = GenerationType.IDENTITY)
12     private long id;
13     private String firstname;
14     private String lastname;
15     private String email;
16     private String password;
17     private String confirmPassword;
18     private String contact;
19     private String education;
20     private String city;
21     private String country;
22     private String workplace;
23     private String position;
24
25     public User() {
26
27     }
28
29     protected User(long id, String firstname, String lastname, String email, String
password, String confirmPassword,
30         String contact, String education, String city, String country, String
workplace, String position) {
31
32         this.id = id;
33         this.firstname = firstname;
34         this.lastname = lastname;
35         this.email = email;
36         this.password = password;
37         this.confirmPassword = confirmPassword;
38         this.contact = contact;
39         this.education = education;
40         this.city = city;
41         this.country = country;
42         this.workplace = workplace;
43         this.position = position;
44     }
45     public long getId() {
46         return id;
47     }
48     public void setId(long id) {
49         this.id = id;
50     }
51     public String getFirstname() {
52         return firstname;
53     }
54     public void setFirstname(String firstname) {
55         this.firstname = firstname;
56     }
57     public String getLastname() {
```

```

58     return lastname;
59 }
60 public void setLastname(String lastname) {
61     this.lastname = lastname;
62 }
63 public String getEmail() {
64     return email;
65 }
66 public void setEmail(String email) {
67     this.email = email;
68 }
69 public String getPassword() {
70     return password;
71 }
72 public void setPassword(String password) {
73     this.password = password;
74 }
75 public String getConfirmpassword() {
76     return confirmpassword;
77 }
78 public void setConfirmpassword(String confirmpassword) {
79     this.confirmpassword = confirmpassword;
80 }
81 public String getContact() {
82     return contact;
83 }
84 public void setContact(String contact) {
85     this.contact = contact;
86 }
87 public String getEducation() {
88     return education;
89 }
90 public void setEducation(String education) {
91     this.education = education;
92 }
93 public String getCity() {
94     return city;
95 }
96 public void setCity(String city) {
97     this.city = city;
98 }
99 public String getCountry() {
100     return country;
101 }
102 public void setCountry(String country) {
103     this.country = country;
104 }
105 public String getWorkplace() {
106     return workplace;
107 }
108 public void setWorkplace(String workplace) {
109     this.workplace = workplace;
110 }
111 public String getPosition() {
112     return position;
113 }
114 public void setPosition(String position) {
115     this.position = position;
116 }
117 }

```

```

118 @Override
119 public String toString() {
120     return "User [id=" + id + ", firstname=" + firstname + ", lastname=" + lastname +
121     ", email=" + email
122     + ", password=" + password + ", confirmpassword=" + confirmpassword + ",
123     contact=" + contact + ", education="
124     + education + ", city=" + city + ", country=" + country + ", workplace=" +
125     workplace + ", position="
126     + position + "]\n";
127 }
128 }

```




# Repo

```
1 package abcjobs.repository;
2
3 import java.util.List;
4
5 import org.springframework.data.jpa.repository.Query;
6 import org.springframework.data.repository.CrudRepository;
7 import org.springframework.data.repository.query.Param;
8
9 import abcjobs.bean.User;
10
11 public interface UserRepository extends CrudRepository<User, Long> {
12
13
14     @Query(value = "SELECT u FROM User u WHERE u.firstname LIKE '%' || :keyword || '%"
15             + " OR u.email LIKE '%' || :keyword || '%"")
16     public List<User> search(@Param("keyword") String keyword);
17
18 }
```

# User Profile

ABC Jobs

HOMEPEOPLEJOBSLoginRegister



Hi welcome chathudil77@gmail.com

Full Name

About

A Lead UX & UI designer based in Canada

I design and develop services for customers of all sizes, specializing in creating stylish, modern websites, web services and online stores. My passion is to design digital user experiences through the bold interface and meaningful interactions.

Full Name	Full Name
Email	\$(user.email)
Birthday	DD/MM/YYYY
Mobile	(320) 380-4539
Address	Bay Area, San Francisco, CA
Other	Education, Work, Skills

Edit Profile

## Edit Profile

Edit Profile

Email

chathudil77@gmail.com

Full Name

Chathushi Dilhari

Birthday

07/07/2000

Mobile

(320) 380-4539

Address

Bay Area, San Francisco, CA

Other

Education, Work, Skills

Save Profile

## Source Code



Source Code.zip