

HANDLING USER AUTHENTICATION PROJECT SOURCE CODE

Prepared By: Deepa Tamilselvan

LoginController.java

```
package com.example.Authentication.controllers;
```

```
import org.slf4j.Logger;
```

```
import org.slf4j.LoggerFactory;
```

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

```
import    org.springframework.stereotype.Controller;    import
```

```
org.springframework.ui.ModelMap;                        import
```

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

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

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

```
org.springframework.web.bind.annotation.ResponseBody; import
```

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

```
import com.example.Authentication.entities.User;
```

```
import com.example.Authentication.repositories.UserRepository;
```

```
@RestController public class
```

```
LoginController {
```

```
@Autowired
```

```
    UserRepository userRepository;
```

```
    @GetMapping(value="/")
```

```
    public String showIndexPage(ModelMap model){
```

```
        return "<html>\n"
```

```
            + "<head>\n"
```

```
            + "    <style>\n"
```

```
            + "        .center {\n"
```

```
            + "            text-align: center;\n"
```

```
            + "        }\n"
```

```
            + "    \n"
```

```
            + "    </style>\n"
```

```
            + "</head>\n"
```

```
            + "<body style=\"background-color:lightblue;\">\n"
```

```
            + "    <div class=\"center\">\n"
```

```
            + "        <h1>User Login Page</h1>\n"
```

```

+ "        \n"
+ "        <h2 class=\"hello-title\">Welcome</h2>\n"
+ "        \n"
+ "        <a href=\"/allusers\">View all users</a>\n"
+ "        <br><br>\n"
+ "    <form method=\"get\" action=\"login\">\n"
+ "        <br><h3>Login below:</h3>\n"
+ "        <input type=\"text\" id=\"name\"
name=\"name\" placeholder=\"Name\" required>\n"
+ "        <input type=\"text\" id=\"email\"
name=\"email\" placeholder=\"Email\" required>\n"
+ "        <input type=\"text\" id=\"password\"
name=\"password\" placeholder=\"Password\" required>        \n"
+ "        <input type=\"submit\"
value=\"Enter\" />\n"
+ "    </form>"
+ " </div>\n"
+ "</body>\n"
+ "</html>";
}

```

```
@GetMapping("/login")
```

```
public String showLogin(@RequestParam("name") String name,  
@RequestParam("email") String email, @RequestParam("password") String  
password, ModelMap map) {
```

```
    User u = new User(name,email,password);  
    userRepository.save(u);
```

```
    return "<html>\n"
```

```
        + "<head>\n"
```

```
        + "    <style>\n"
```

```
        + "        .center {\n"
```

```
        + "            text-align: center;\n"
```

```
        + "        }\n"
```

```
        + "    \n"
```

```
        + "    </style>\n"
```

```
        + "</head>\n"
```

```
        + "<body style=\"background-color:lightblue;\">\n"
```

```
        + "    <div class=\"center\">\n"
```

```
        + "        <h1>Logged In</h1>\n"
```

```
        + "    \n"
```

```
        + "        <h2 class=\"hello-title\">Successfully Added Your  
Information</h2>\n"
```

```
        + "    </div>\n"
```

```
        + "</body>\n"
```

```
        + "</html>";
```

```
    }
```

```
    @GetMapping("/allusers") public @ResponseBody
```

```
        String getAllFeedbacks() {
```

```
        // This returns a JSON or XML with the Feedbacks
```

```
        Iterable<User> allUser = userRepository.findAll();
```

```
        return "<html>\n"
```

```
            + "<head>\n"
```

```
                + "    <style>\n"
```

```
                    + "        .center {\n"
```

```
                        + "            text-align: center;\n"
```

```
                    + "        }\n"
```

```
                + "    \n"
```

```
            + "    </style>\n"
```

```
        + "</head>\n"
```

```
        + "<body style=\"background-color:lightblue;\">\n"
```

```
        + "    <div class=\"center\">\n"
        + "<h1>Feedback Table</h1>\n"
    + allUser.toString()

    + "    </div>\n"
    + "</body>\n"
    + "</html>";

}
@PostMapping("/login")

public String submitLogin(@RequestParam String username, @RequestParam
String password){

    //TODO:

    return "Success";

}
}
```

User.java

```
package com.example.Authentication.entities;

import javax.persistence.Entity; import
javax.persistence.GeneratedValue; import
javax.persistence.GenerationType; import
javax.persistence.Id;

@Entity // This tells Hibernate to make a table out of this class
public class User {

    @Id

    @GeneratedValue(strategy=GenerationType.IDENTITY)

    private Integer id;


    private String name;


    private String email;


    private String password;

    public User()

    {
```



```
}
```

```
public User(String name, String email, String password) {
```

```
    this.name = name; this.email = email; this.password =  
    password;
```

```
}
```

```
public String getPassword() { return
```

```
    password;
```

```
}
```

```
public void setPassword(String password) {
```

```
    this.password = password;
```

```
}
```

```
public Integer getId() {
```

```
    return id;
```

```
}
```

```
public void setId(Integer id) {
```

```
    this.id = id;
```

```
}
```

```
public String getName() {
```

```
    return name;
```

```
}
```

```
public void setName(String name) {  
    this.name = name;  
}
```

```
public String getEmail() {  
    return email;  
}
```

```
public void setEmail(String email) {  
    this.email = email;  
}
```

```
@Override
```

```
public String toString() {
```

```
    return "<br><h3>" + name + " [" + id + "]:" + "</h3><h4>email: " + email +  
"</h4><h4>password: " + password + "</h4><br>";  
}
```

```
}
```

UserNotFoundException.java

```
package com.example.Authentication.exceptions;
public class UserNotFoundException extends RuntimeException { private
    static final long serialVersionUID = 1L;
}
```

UserRepository.java

```
package com.example.Authentication.repositories;

import org.springframework.data.repository.CrudRepository;

import com.example.Authentication.entities.User;

public interface UserRepository extends CrudRepository<User, Integer> {

    public User findByName(String name);
}
```

UserService.java

```
package com.example.Authentication.services;
```

```
import java.util.Optional;
import
org.springframework.be
ans.factory.annotation.A
utowired; import
org.springframework.ste
reotype.Service;
```

```
import      com.example.Authentication.entities.User;      import
com.example.Authentication.exceptions.UserNotFoundException;
import com.example.Authentication.repositories.UserRepository;
```

```
@Service
```

```
public class UserService {
```

```
    @Autowired
```

```
    private UserRepository userRepository;
```

```
    public Iterable<User> GetAllUsers()
```

```
{  
    return userRepository.findAll();  
}
```

```
public User GetUserByName(String name) {  
    User foundUser = userRepository.findByName(name); return  
    foundUser;  
}
```

```
public User GetUserById(int id) {  
    Optional<User> foundUser = userRepository.findById(id);
```

//TODO: we need to decide how to handle a "Not Found" condition

```
if (!foundUser.isPresent()) {  
    throw new UserNotFoundException();  
}
```

```
return(foundUser.get());  
}
```

```
public void updateUser(User usertoUpdate) {  
    userRepository.save(usertoUpdate);  
}  
  
}
```

AuthenticationApplication.java

```
package com.example.Authentication;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class AuthenticationApplication {  
  
    public static void main(String[] args) {  
        SpringApplication.run(AuthenticationApplication.class, args);  
    }  
  
}
```

Application.properties

spring.jpa.hibernate.ddl-auto=update

spring.datasource.url=jdbc:mysql://\${MYSQL_HOST:localhost}:3306/phase2

spring.datasource.username=root

spring.datasource.password=root

logging.level.org.springframework.web: DEBUG

server.port=8090

server.error.whitelabel.enabled=false

Error.jsp

<html>

<body>

```
<h2>Page not found</h2>
```

```
</body>
```

```
</html>
```

Greeting.java

```
<html>
```

```
<body>
```

```
<h2>Spring Application</h2>
```

```
</body>
```

```
</html>
```

=====X=====