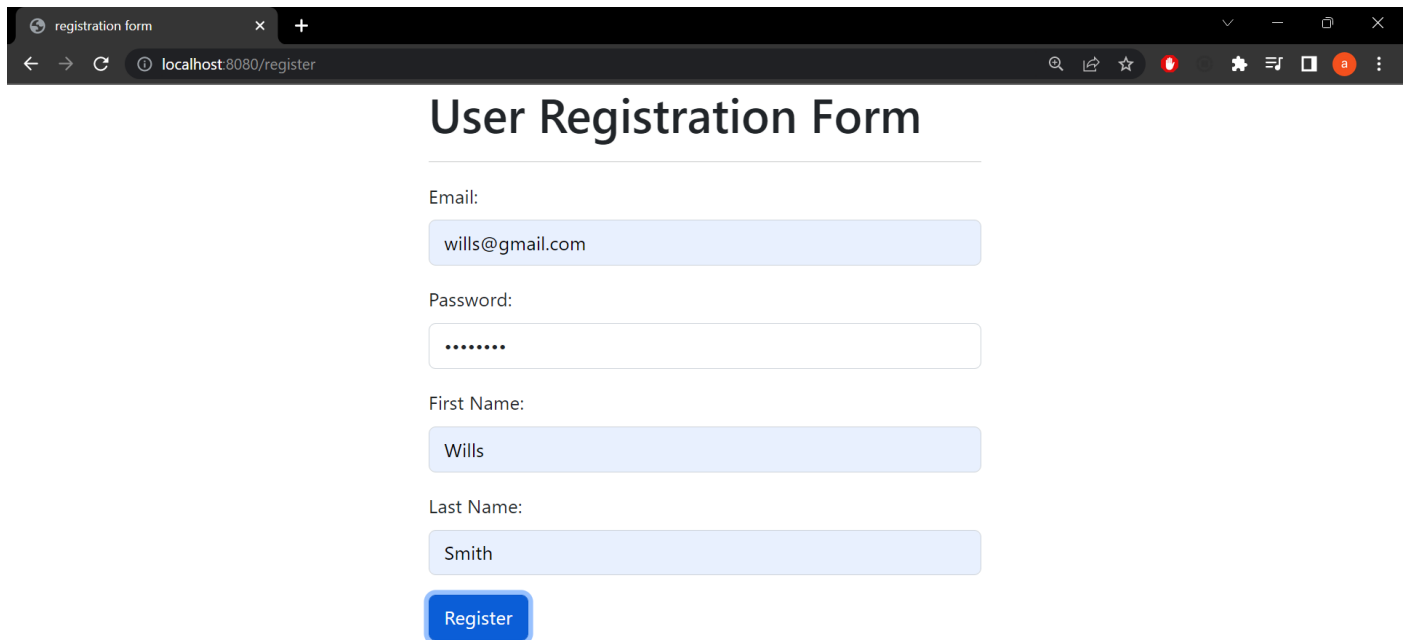
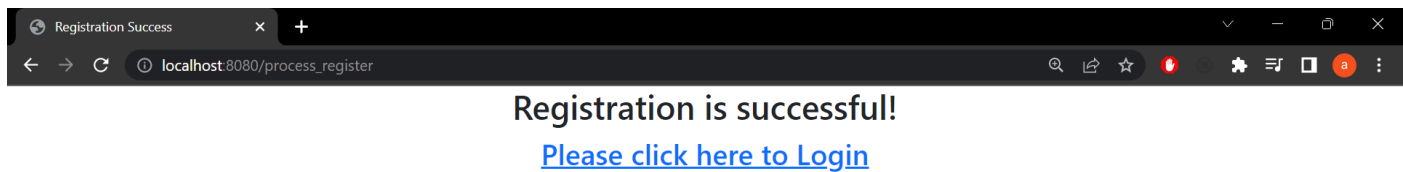


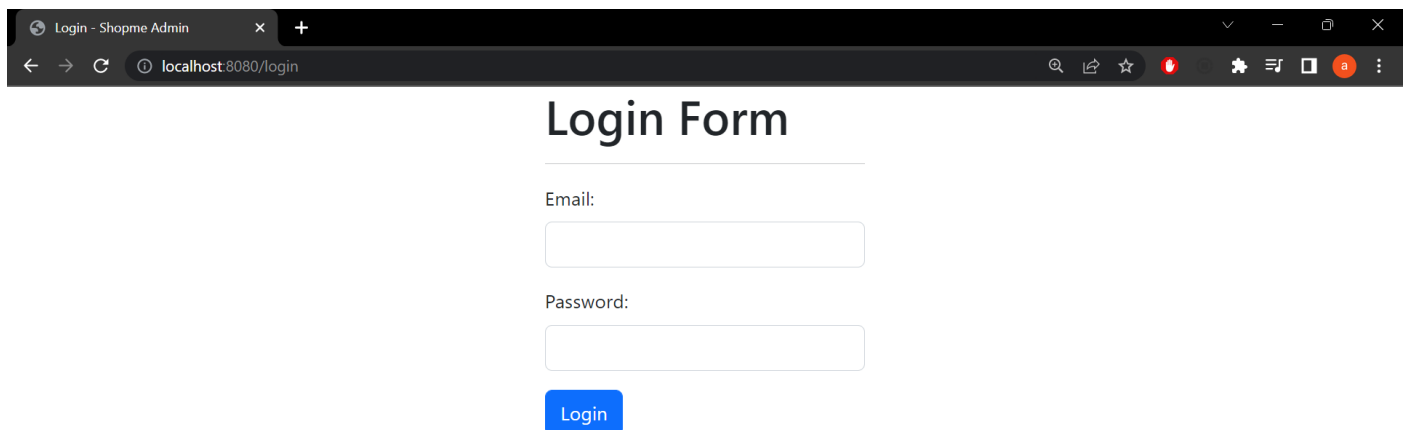
Spring Boot Registration and Login using mysql



The screenshot shows a web browser window with the title "registration form" and the address bar displaying "localhost:8080/register". The page has a dark theme. The main heading is "User Registration Form". Below it, there are four input fields: "Email:" with the value "wills@gmail.com", "Password:" with masked characters ".....", "First Name:" with the value "Wills", and "Last Name:" with the value "Smith". At the bottom of the form is a blue "Register" button.



The screenshot shows a web browser window with the title "Registration Success" and the address bar displaying "localhost:8080/process_register". The page has a dark theme. The main heading is "Registration is successful!". Below it is a blue link that says "Please click here to Login".



The screenshot shows a web browser window with the title "Login - Shopme Admin" and the address bar displaying "localhost:8080/login". The page has a dark theme. The main heading is "Login Form". Below it, there are two input fields: "Email:" and "Password:". At the bottom of the form is a blue "Login" button.

Login - Shopme Admin

localhost:8080/login

Login Form

Email:

wills@gmail.com

Password:

.....

Login

index

localhost:8080/profile

Welcome, Wills Smith

Sign Out

registration form

localhost:8080/process_register

wills@gmail.com already exists!

User Registration Form

Email:

wills@gmail.com

Password:

First Name:

Wills

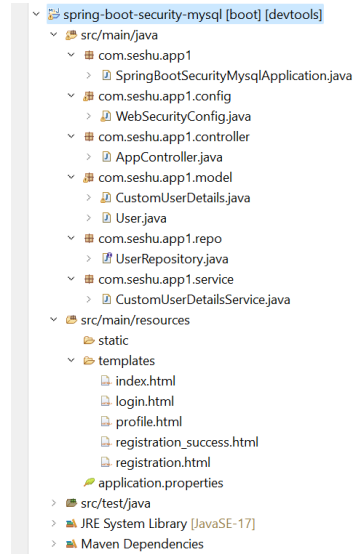
Last Name:

Smith

Register

Create spring boot application ***spring-boot-security-mysql*** with fallowing dependencies.

- X Spring Boot DevTools
- X Spring Data JPA
- X MySQL Driver
- X Spring Security
- X Thymeleaf
- X Spring Web



pom.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>2.7.4</version>
    <relativePath /> <!-- lookup parent from repository -->
  </parent>
  <groupId>com.seshu</groupId>
  <artifactId>spring-boot-security-mysql</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <name>spring-boot-security-mysql</name>
  <description>Demo project for Spring Boot</description>
  <properties>
    <java.version>17</java.version>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-data-jpa</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-security</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-thymeleaf</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
    <dependency>
      <groupId>org.thymeleaf.extras</groupId>
      <artifactId>thymeleaf-extras-
springsecurity5</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-devtools</artifactId>
      <scope>runtime</scope>
    </dependency>
  </dependencies>
</project>
```

```
        <optional>true</optional>
    </dependency>
    <dependency>
        <groupId>mysql</groupId>
        <artifactId>mysql-connector-java</artifactId>
        <scope>runtime</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-test</artifactId>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.springframework.security</groupId>
        <artifactId>spring-security-test</artifactId>
        <scope>test</scope>
    </dependency>
</dependencies>

<build>
    <plugins>
        <plugin>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-maven-plugin</artifactId>
        </plugin>
    </plugins>
</build>

</project>
```

application.properties

```
spring.jpa.hibernate.ddl-auto=update
spring.datasource.url=jdbc:mysql://localhost:3306/adiDB
spring.datasource.username=root
spring.datasource.password=seshu
```

```
package com.seshu.app1.model;

import javax.persistence.*;

@Entity
@Table(name = "users")
public class User {
    @Id
    @GeneratedValue
```

```
private Long id;

private String email;

private String password;

private String firstName;

private String lastName;

public Long getId() {
    return id;
}

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

public String getEmail() {
    return email;
}

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

public String getPassword() {
    return password;
}

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

public String getFirstName() {
    return firstName;
}

public void setFirstName(String firstName) {
    this.firstName = firstName;
}

public String getLastName() {
    return lastName;
}

public void setLastName(String lastName) {
    this.lastName = lastName;
}
```

```
    }

    @Override
    public String toString() {
        return "User [id=" + id + ", email=" + email + ", password="
+ password + ", firstName=" + firstName
        + ", lastName=" + lastName + "]\n";
    }
}
```

```
package com.seshu.app1.repo;

import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;

import com.seshu.app1.model.User;

public interface UserRepository extends JpaRepository<User, Long> {
    @Query("SELECT u FROM User u WHERE u.email = ?1")
    public User findByEmail(String email);
}
```



```
package com.seshu.appl.model;

import java.util.Collection;

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

public class CustomUserDetails implements UserDetails {

    private User user;

    public CustomUserDetails(User user) {
        this.user = user;
    }

    @Override
    public Collection<? extends GrantedAuthority> getAuthorities() {
        return null;
    }

    @Override
    public String getPassword() {
        return user.getPassword();
    }

    @Override
    public String getUsername() {
        return user.getEmail();
    }

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

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

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

    @Override
    public boolean isEnabled() {
```

```
        return true;
    }

    public String getFullName() {
        return user.getFirstName() + " " + user.getLastName();
    }
}
```

```
package com.seshu.app1.service;

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 com.seshu.app1.model.CustomUserDetails;
import com.seshu.app1.model.User;
import com.seshu.app1.repo.UserRepository;

public class CustomUserDetailsService implements UserDetailsService {

    @Autowired
    private UserRepository userRepo;

    @Override
    public UserDetails loadUserByUsername(String username) throws
UsernameNotFoundException {
        User user = userRepo.findByEmail(username);
        if (user == null) {
            throw new UsernameNotFoundException("User not found");
        }
        return new CustomUserDetails(user);
    }
}
```

```
package com.seshu.appl.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

import com.seshu.appl.model.User;
import com.seshu.appl.repo.UserRepository;

@Controller
public class AppController {

    @Autowired
    private UserRepository userRepo;

    @GetMapping("/")
    public String viewHomePage() {
        return "index";
    }

    @GetMapping("/register")
    public String showRegistrationForm(Model model) {
        model.addAttribute("user", new User());

        return "registration";
    }

    @PostMapping("/process_register")
    public String processRegister(User user, Model model) {
        User existedUser = userRepo.findByEmail(user.getEmail());
        String existedEmail = "";
        if(existedUser != null) {
            existedEmail = existedUser.getEmail();
        }

        if (user.getEmail().equals(existedEmail)) {
            model.addAttribute("existedUser", existedEmail);
            return "registration";
        } else {
            BCryptPasswordEncoder passwordEncoder = new
BCryptPasswordEncoder();
            String encodedPassword =
passwordEncoder.encode(user.getPassword());
            user.setPassword(encodedPassword);
        }
    }
}
```

```
        userRepo.save(user);  
        return "registration_success";  
    }  
}  
  
@GetMapping("/login")  
public String viewLoginPage() {  
    return "login";  
}  
  
@GetMapping("/profile")  
public String showProfilePage() {  
    return "profile";  
}  
}
```

```
package com.seshu.appl.config;

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.authentication.builders.Au
thenticationManagerBuilder;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.config.annotation.web.configuration.EnableWe
bSecurity;
import org.springframework.security.config.annotation.web.configuration.WebSecur
ityConfigurerAdapter;
import org.springframework.security.core.userdetails.UserDetailsService;
import org.springframework.security.crypto.bcrypt.BCryptPasswordEncoder;

import com.seshu.appl.service.CustomUserDetailsService;

@Configuration
@EnableWebSecurity
public class WebSecurityConfig extends WebSecurityConfigurerAdapter {

    @Bean
    public UserDetailsService userDetailsService() {
        return new CustomUserDetailsService();
    }

    @Bean
    public BCryptPasswordEncoder passwordEncoder() {
        return new BCryptPasswordEncoder();
    }

    @Bean
    public DaoAuthenticationProvider authenticationProvider() {
        DaoAuthenticationProvider authProvider = new
        DaoAuthenticationProvider();
        authProvider.setUserDetailsService(userDetailsService());
        authProvider.setPasswordEncoder(passwordEncoder());

        return authProvider;
    }
}
```

```
@Override
protected void configure(AuthenticationManagerBuilder auth) throws
Exception {
    auth.authenticationProvider(authenticationProvider());
}

@Override
protected void configure(HttpSecurity http) throws Exception {
    http.authorizeRequests()
        .antMatchers("/profile").authenticated()
        .anyRequest().permitAll()
        .and()
        .formLogin()
            .loginPage("/login")
            .usernameParameter("email")
            .defaultSuccessUrl("/profile")
            .permitAll()
        .and()
        .logout().logoutSuccessUrl("/").permitAll();
}
```

index.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="ISO-8859-1">
  <title>index</title>
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min
.css" rel="stylesheet" integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundl
e.min.js" integrity="sha384-
u10knCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>

</head>
<body>
  <div class="container">
    <a th:href="@{register}">Register</a>
    <a th:href="@{login}">Login</a>
  </div>
</body>
</html>
```

registration.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
    <meta charset="ISO-8859-1">
    <title>registration form</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min
.css" rel="stylesheet"
        integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundl
e.min.js"
        integrity="sha384-
u10knCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>

</head>

<body>
    <div class="container col-5">
        <div th:if="{existedUser != null}" class="alert alert-danger
notification">
            <span th:text="{existedUser}"></span> already exists!
        </div>

        <h1>User Registration Form</h1>
        <hr>
        <form th:action="@{/process_register}" th:object="{user}"
method="post">
            <div class="mb-3">
                <label class="form-label">Email: </label>
                <input type="email" th:field="{email}" class="form-
control" required />
            </div>

            <div class="mb-3">
                <label class="form-label">Password: </label>
                <input type="password" th:field="{password}"
class="form-control" required />
            </div>

            <div class="mb-3">
                <label class="form-label">First Name: </label>
```



```

        <input type="text" th:field="*{firstName}"
class="form-control" required />
    </div>

    <div class="mb-3">
        <label class="form-label">Last Name: </label>

        <input type="text" th:field="*{lastName}"
class="form-control" required />
    </div>

    <div>
        <button type="submit" class="btn btn-
primary">Register</button>
    </div>
</form>
</div>
</body>
</html>

```

registration_success.html

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="ISO-8859-1">
    <title>Registration Success</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min
.css" rel="stylesheet" integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundl
e.min.js" integrity="sha384-
u10knCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>

</head>
<body>
    <div class="container text-center">
        <h3>Registration is successful!</h3>
        <h4><a th:href="@{login}">Please click here to Login</a></h4>
    </div>

</body>
</html>

```


login.html

```

<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0, minimum-scale=1.0">
    <title>Login - Shopme Admin</title>
    <link
href="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min
.css" rel="stylesheet"
    integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlZB6+fzT"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivrivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundl
e.min.js"
    integrity="sha384-
u10knCvWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>
</head>

<body>
    <div class="container col-3">

        <form th:action="@{/login}" method="post">
            <h1>Login Form</h1><hr>
            <div th:if="${param.error}">
                <p class="text-
danger">[[${session.SPRING_SECURITY_LAST_EXCEPTION.message}]]</p>
            </div>

            <div class="mb-3">
                <label class="form-label">Email: </label>
                <input type="email" name="email" class="form-control"
required />
            </div>
            <div class="mb-3">
                <label class="form-label">Password: </label>
                <input type="password" name="password" class="form-
control" required />
            </div>

            <div>
                <input type="submit" value="Login" class="btn btn-
primary" />
            </div>
        </form>
    </div>

```

```
        </form>
    </div>
</body>

</html>
```

profile.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="ISO-8859-1">
    <title>index</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min
.css" rel="stylesheet" integrity="sha384-
iYQeCzEYFbKjA/T2uDLTpkwGzCiq6soy8tYaI1GyVh/UjpbCx/TYkiZhlzB6+fzT"
crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundl
e.min.js" integrity="sha384-
u10knCvxWvY5kfmNBILK2hRnQC3Pr17a+RTT6rIHI7NnikvbZlHgTPOOmMi466C8"
crossorigin="anonymous"></script>
</head>
<body>
    <div class="container">
        <div>
            <form th:action="@{/logout}" method="post">
                <p>
                    Welcome,
[[${#request.userPrincipal.principal.fullName}]]</b>
                </p>
                <input type="submit" value="Sign Out" />
            </form>
        </div>
    </div>
</body>
</html>
```