

```
1 package stanford.studentloginsp15;
2
3 import org.springframework.boot.SpringApplication;
4 import org.springframework.boot.autoconfigure.
5   SpringApplication;
6
7 @SpringBootApplication
8 public class StudentLoginSp15Application {
9
10   public static void main(String[] args) {
11     SpringApplication.run(
12       StudentLoginSp15Application.class, args);
13   }
14 }
```

```
1 package stanford.studentloginsp15.DTO;
2 import jakarta.validation.constraints.Email;
3 import jakarta.validation.constraints.NotEmpty;
4 public class StudentDTO {
5     private Long id;
6     @NotEmpty
7     private String firstName;
8     @NotEmpty
9     private String lastName;
10    @NotEmpty(message = "Email should not be empty")
11    @Email
12    private String email;
13
14    @NotEmpty(message = "Password should not be empty"
15    ")
16    private String password;
17
18    public Long getId() {
19        return id;
20    }
21
22    public void setId(Long id) {
23        this.id = id;
24    }
25
26    public String getFirstName() {
27        return firstName;
28    }
29
30    public void setFirstName(String firstName) {
31        this.firstName = firstName;
32    }
33
34    public String getLastName() {
35        return lastName;
36    }
37
38    public void setLastName(String lastName) {
39        this.lastName = lastName;
40    }
```

```
41     public String getEmail() {
42         return email;
43     }
44
45     public void setEmail(String email) {
46         this.email = email;
47     }
48
49     public String getPassword() {
50         return password;
51     }
52
53     public void setPassword(String password) {
54         this.password = password;
55     }
56 }
57 }
58 }
```

```
1 package stanford.studentloginsp15.model;
2 import jakarta.persistence.*;
3 import lombok.Getter;
4 import lombok.NoArgsConstructor;
5 import lombok.Setter;
6 import lombok.ToString;
7 import java.util.ArrayList;
8 import java.util.List;
9
10 @Entity
11 @Setter
12 @Getter
13 @NoArgsConstructor
14 @ToString
15 @Table(name ="roles")
16 public class Role {
17     @Id
18     @GeneratedValue(strategy = GenerationType.
IDENTITY)
19     Long id;
20     @Column(nullable = false,unique = true)
21     String name;
22     @ManyToMany(mappedBy = "roles")
23     List<Student> students = new ArrayList<>();
24
25     public Role(Long id, String name, List<Student>
students) {
26         this.id = id;
27         this.name = name;
28         this.students = students;
29     }
30 }
31
```

```
1 package stanford.studentloginsp15.model;
2 import jakarta.persistence.*;
3 import lombok.Getter;
4 import lombok.NoArgsConstructor;
5 import lombok.Setter;
6 import lombok.ToString;
7 import java.util.ArrayList;
8 import java.util.List;
9 @Entity
10 @Setter
11 @Getter
12 @NoArgsConstructor
13 @ToString
14 public class Student {
15     @Id
16     @GeneratedValue(strategy=GenerationType.IDENTITY)
17     Long id;
18     @Column(nullable = false)
19     String name;
20     String email;
21     @Column(nullable = false,unique = true)
22     String password;
23     @ManyToMany(fetch=FetchType.EAGER,cascade =
24     CascadeType.ALL)
24         @JoinTable(name="student_roles", joinColumns = {@
25             JoinColumn(name="role_id", referencedColumnName = "id"
26         )})
25         public List<Role> roles = new ArrayList<>();
26
27     public Student(Long id, String name, String email
28     , String password, List<Role> roles) {
29         this.id = id;
30         this.name = name;
31         this.email = email;
32         this.password = password;
33         this.roles = roles;
34     }
35 }
```

```
1 package stanford.studentloginsp15.config;
2
3 import org.springframework.context.annotation.Bean;
4 import org.springframework.context.annotation.
Configuration;
5 import org.springframework.security.config.annotation
.web.builders.HttpSecurity;
6 import org.springframework.security.config.annotation
.web.configuration.EnableWebSecurity;
7 import org.springframework.security.crypto.bcrypt.
BCryptPasswordEncoder;
8 import org.springframework.security.crypto.password.
PasswordEncoder;
9 import org.springframework.security.web.
SecurityFilterChain;
10 import org.springframework.security.web.util.matcher.
AntPathRequestMatcher;
11
12 @Configuration
13 @EnableWebSecurity
14 public class SpringSecurity {
15     @Bean
16     public static PasswordEncoder passwordEncoder() {
17         return new BCryptPasswordEncoder();
18     }
19
20     @Bean
21     public SecurityFilterChain filterChain(
HttpSecurity http) throws Exception {
22
23         http.csrf(csrf -> csrf.disable())
24             .authorizeHttpRequests(auth -> auth.
requestMatchers("/register/**")
25                 .permitAll()
26                 .requestMatchers("/index")
27                 .permitAll()
28                 .requestMatchers("/students")
29                 .hasRole("ADMIN"))
30
31         .formLogin(form -> form.loginPage("/")
login")
```

```
32          .loginProcessingUrl("/login")
33          .defaultSuccessUrl("/students
  ")
34          .permitAll())
35      .logout(logout -> logout.
  logoutRequestMatcher(new AntPathRequestMatcher("/
    logout")))
36          .permitAll());
37      return http.build();
38  }
39
40 }
41
```

```
1 package stanford.studentloginsp15.service;
2
3 import stanford.studentloginsp15.dto.StudentDTO;
4 import stanford.studentloginsp15.model.Student;
5
6 import java.util.List;
7
8 public interface StudentService {
9     void saveStudent(StudentDTO studentDTO);
10    Student findStudentByEmail(String email);
11    List<StudentDTO> findAllStudents();
12 }
13
```

```
1 package stanford.studentloginsp15.service.impl;
2 import java.util.Arrays;
3 import java.util.List;
4 import java.util.stream.Collectors;
5 import org.springframework.security.crypto.password.PasswordEncoder;
6 import org.springframework.stereotype.Service;
7 import stanford.studentloginsp15.dto.StudentDTO;
8 import stanford.studentloginsp15.model.Role;
9 import stanford.studentloginsp15.model.Student;
10 import stanford.studentloginsp15.repository.RoleRepository;
11 import stanford.studentloginsp15.repository.StudentRepository;
12 import stanford.studentloginsp15.service.StudentService;
13
14 @Service
15 public class StudentServiceImpl implements
16     StudentService {
17     private StudentRepository studentRepository;
18     private RoleRepository roleRepository;
19     private PasswordEncoder passwordEncoder;
20
21     public StudentServiceImpl(StudentRepository
22         studentRepository, RoleRepository roleRepository,
23         PasswordEncoder passwordEncoder) {
24         this.studentRepository = studentRepository;
25         this.roleRepository = roleRepository;
26         this.passwordEncoder = passwordEncoder;
27     }
28
29     @Override
30     public void saveStudent(StudentDTO studentDTO){
31         Student student = new Student();
32         student.setName(studentDTO.getFirstName() +
33             " " + studentDTO.getLastName());
34         student.setEmail(studentDTO.getEmail());
35         student.setPassword(passwordEncoder.encode(
36             studentDTO.getPassword()));
37         Role role = roleRepository.findByName("
```

```
32 ROLE_ADMIN");
33         if(role == null){
34             role = checkRoleExist();
35         }
36         student.setRoles(Arrays.asList(role));
37         studentRepository.save(student);
38     }
39
40     private Role checkRoleExist(){
41         Role role = new Role();
42         role.setName("ROLE_ADMIN");
43         return roleRepository.save(role);
44     }
45     @Override
46     public Student findStudentByEmail(String email){
47         return studentRepository.findByEmail(email);
48     }
49
50     @Override
51     public List<StudentDTO> findAllStudents(){
52         List<Student> students = studentRepository.
53         findAll();
54         return students.stream().map((student ->
55             mapToStudentDto(student))).collect(Collectors.toList
56         ());
57     }
58
59     private StudentDTO mapToStudentDto(Student
60     student){
61         StudentDTO studentDTO = new StudentDTO();
62         String[] str = student.getName().split(" ");
63         studentDTO.setFirstName(str[0]);
64         studentDTO.setLastName(str[1]);
65         studentDTO.setEmail(student.getEmail());
66         return studentDTO;
67     }
68 }
```

```
1 package stanford.studentloginsp15.security;
2
3 import org.springframework.security.core.authority.
4 SimpleGrantedAuthority;
5 import org.springframework.security.core.userdetails.
6 UserDetails;
7 import org.springframework.security.core.userdetails.
8 UserDetailsService;
9 import org.springframework.security.core.userdetails.
10 UsernameNotFoundException;
11 import org.springframework.stereotype.Service;
12 import stanford.studentloginsp15.model.Student;
13 import stanford.studentloginsp15.repository.
14 StudentRepository;
15
16 import java.util.stream.Collectors;
17
18 @Service
19 public class CustomStudentDetailsService implements
20 UserDetailsService {
21
22     private StudentRepository studentRepository;
23     public CustomStudentDetailsService(
24         StudentRepository studentRepository) {
25         super();
26         this.studentRepository = studentRepository;
27     }
28
29     @Override
30     public UserDetails loadUserByUsername(String
31         usernameOrEmail) throws UsernameNotFoundException {
32
33         Student student = studentRepository.
34             findByEmail(usernameOrEmail);
35         if(student != null) {
36             return new org.springframework.security.
37                 core.userdetails.User(student.getEmail(),
38                     student.getPassword(),
39                     student.getRoles().stream()
40                         .map((role) -> new
41                             SimpleGrantedAuthority(role.getName())))
42         }
43     }
44 }
```

```
31                               .collect(Collectors.  
32           toList()));  
33       }else {  
34           throw new UsernameNotFoundException("Invalid email or password");  
35       }  
36   }  
37
```

```
1 package stanford.studentloginsp15.controller;
2 import java.util.List;
3 import org.springframework.beans.factory.annotation.Autowired;
4 import org.springframework.stereotype.Controller;
5 import org.springframework.ui.Model;
6 import org.springframework.validation.BindingResult;
7 import org.springframework.web.bind.annotation.GetMapping;
8 import org.springframework.web.bind.annotation.ModelAttribute;
9 import org.springframework.web.bind.annotation.PostMapping;
10 import jakarta.validation.Valid;
11 import stanford.studentloginsp15.dto.StudentDTO;
12 import stanford.studentloginsp15.model.Student;
13 import stanford.studentloginsp15.service.StudentService;
14
15 @Controller
16 public class StudentAuthController {
17     private StudentService studentService;
18
19     @Autowired
20     public StudentAuthController(StudentService studentService) {
21         this.studentService = studentService;
22     }
23     // handler method to handle the home (index.html
24     // is home) page request
25     @GetMapping("/index")
26     public String home() {
27         return "index";
28     }
29     // handler method handles the login request
30     @GetMapping("/login")
31     public String login() {
32         return "login";
33     }
34 }
```

```
35     // handler method to handle the student
36     // registration form request
37     @GetMapping("/register")
38     public String showRegistrationForm(Model model) {
39
40         // create a model object to store form data
41         StudentDTO student = new StudentDTO();
42         model.addAttribute("student", student);
43
44         return "register";
45     }
46
47     // handler method to handle student registration
48     // from submit request
49     @PostMapping("/register/save")
50     public String registration(@Valid @ModelAttribute
51         ("student") StudentDTO studentDto, BindingResult
52         result,
53         Model model) {
54
55         Student existingStudent = studentService.
56         findStudentByEmail(studentDto.getEmail());
57
58         if (existingStudent != null &&
59             existingStudent.getEmail() != null && !
60             existingStudent.getEmail().isEmpty()) {
61             result.rejectValue("email", null, "There
62             is already an account registered with the same email"
63             );
64         }
65
66         if (result.hasErrors()) {
67             model.addAttribute("student", studentDto
68             );
69
70             return "/register";
71         }
72
73         studentService.saveStudent(studentDto);
74         return "redirect:/register?success";
75     }
```

```
66      }
67
68      // handler method is used to handle a list of
69      // students
70      @GetMapping("/students")
71      public String students(Model model) {
72          List<StudentDTO> students = studentService.
73          findAllStudents();
74
75          model.addAttribute("students", students);
76
77      }
78 }
79
```

```
1 package stanford.studentloginsp15.repository;
2
3 import org.springframework.data.jpa.repository.
4 JpaRepository;
5
6 public interface RoleRepository extends JpaRepository
7 <Role, Long> {
8     Role findByName(String name);
9 }
```

```
1 package stanford.studentloginsp15.repository;
2 import stanford.studentloginsp15.dto.StudentDTO;
3 import stanford.studentloginsp15.model.Student;
4
5 import java.util.List;
6
7 public interface StudentService {
8     void saveStudent(StudentDTO studentDTO);
9     Student findStudentByEmail(String email);
10    List<StudentDTO> findAllStudents();
11 }
12
```

```
1 package stanford.studentloginsp15.repository;
2 import org.springframework.data.jpa.repository.
JpaRepository;
3 import stanford.studentloginsp15.model.Student;
4
5 public interface StudentRepository extends
JpaRepository<Student, Long>{
6     Student findByEmail(String email);
7 }
8
```

```
1 spring.datasource.url=jdbc:mysql://localhost:3306/
  studentregisterdb
2 spring.datasource.username= root
3 spring.datasource.password= SzhengSQL123
4 spring.jpa.properties.hibernate.dialect=org.hibernate
  .dialect.MySQLDialect
5 spring.datasource.driver-class-name=com.mysql.cj.jdbc
  .Driver
6 # Hibernate ddl auto (create, create-drop, validate,
  update)
7 spring.jpa.hibernate.ddl-auto= update
8 # show queries on console
9 spring.jpa.show-sql=true
```

```
1 <!DOCTYPE html>
2 <html lang="en"    xmlns:th="http://www.thymeleaf.org">
3   >
4   <head>
5     <meta charset="UTF-8">
6     <title>Student Registration and Login System</title>
7     <link href="https://cdn.jsdelivr.net/npm/
bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="
stylesheet" integrity="sha384-EVSTQN3/
azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTxFspd3yD65VohpuuCOMLA
SjC" crossorigin="anonymous">
8   </head>
9   <body>
10  <nav class="navbar navbar-expand-lg navbar-dark bg-
dark">
11    <div class="container-fluid">
12      <a class="navbar-brand" th:href="@{/index}">
13        Student Registration and Login System</a>
14      <button class="navbar-toggler" type="button"
15        data-bs-toggle="collapse" data-bs-target="#
16        navbarSupportedContent" aria-controls=""
17        navbarSupportedContent" aria-expanded="false" aria-
18        label="Toggle navigation">
19        <span class="navbar-toggler-icon"></span>
20      </button>
21      <div class="collapse navbar-collapse" id=""
22        navbarSupportedContent">
23        <ul class="navbar-nav me-auto mb-2 mb-lg-
0">
24          <li class="nav-item">
25            <a class="nav-link active" aria-
26            current="page" th:href="@{/register}">Register</a>
27          </li>
28        </ul>
29      </div>
30    </div>
31  </nav>
32  <br /><br />
33  <div class="container">
34    <div class="row">
```

```
27      <h1 class="text-center"> Student Registration  
and Login System </h1>  
28  </div>  
29 </div>  
30 </body>  
31 </html>  
32
```

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <meta charset="UTF-8">
5     <title>Student Registration and Login System</title>
6     <link href="https://cdn.jsdelivr.net/npm/
bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-EVSTQN3/
azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpucO0mLA
SjC"
7         crossorigin="anonymous">
8 </head>
9 <body>
10 <nav class="navbar navbar-expand-lg navbar-dark bg-
dark">
11     <div class="container-fluid">
12         <a class="navbar-brand" th:href="@{/index}">
13             Student Registration and Login System</a>
14         <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#
navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
15             <span class="navbar-toggler-icon"></span>
16         </button>
17         <div class="collapse navbar-collapse" id="
navbarSupportedContent">
18             <ul class="navbar-nav me-auto mb-2 mb-lg-
0">
19                 <li class="nav-item">
20                     <a class="nav-link active" aria-
current="page" th:href="@{/register}">Register</a>
21                 </li>
22             </ul>
23         </div>
24     </div>
25     <br /><br />
26     <div class="container">
27         <div class="row">
```

```
28      <div class="col-md-6 offset-md-3">
29
30          <div th:if="${param.error}">
31              <div class="alert alert-danger">
32                  Invalid Email or Password</div>
33          </div>
34          <div th:if="${param.logout}">
35              <div class="alert alert-success"> You
36                  have been logged out.</div>
37          </div>
38
39          <div class="card">
40              <div class="card-header">
41                  <h2 class="text-center">Login
42                      Form</h2>
43              </div>
44              <div class="card-body">
45                  <form method="post"
46                      role="form"
47                      th:action="@{/login}"
48                      class="form-horizontal" >
49                  <div class="form-group mb-3">
50                      <label class="control-
51                          label"> Email</label>
52
53                      <input
54                          type="text"
55                          id="username"
56                          name="username"
57                          class="form-
58                          control"
59                          placeholder=""
60                          Enter email address"
61                          />
62                  </div>
63
64                  <div class="form-group mb-3">
65                      <label class="control-
66                          label"> Password</label>
67
68                      <input
69                          type="password"
70                          id="password"
71                          />
72                  </div>
73
74          </div>
75
76      </div>
77
78  </div>
79
80  </div>
81
82  </div>
83
84  </div>
85
86  </div>
87
88  </div>
89
90  </div>
91
92  </div>
93
94  </div>
95
96  </div>
97
98  </div>
99
100 </div>
```

```
62                               name="password"
63                               class="form-
  control"
64                               placeholder="
Enter password"
65                               />
66                               </div>
67                               <div class="form-group mb-3"
  >
68                               <button type="submit"
  class="btn btn-primary" >Submit</button>
69                               <span> Not registered ?
70                               <a th:href="@{/
register}">Register/Signup here</a>
71                               </span>
72                               </div>
73                               </form>
74                               </div>
75                               </div>
76                               </div>
77                         </div>
78 </div>
79 </body>
80 </html>
81
```

```
1 <!DOCTYPE html>
2 <html lang="en"
3     xmlns:th="http://www.thymeleaf.org">
4 <head>
5     <meta charset="UTF-8">
6     <title>Student Registration and Login System</
7         title>
8     <link href="https://cdn.jsdelivr.net/npm/
9         bootstrap@5.0.2/dist/css/bootstrap.min.css"
10        rel="stylesheet"
11        integrity="sha384-EVSTQN3/
12         azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWspd3yD65VohhpuuC0mLA
13         SjC"
14         crossorigin="anonymous">
15 </head>
16 <body>
17 <nav class="navbar navbar-expand-lg navbar-dark bg-
18         dark">
19     <div class="container-fluid">
20         <a class="navbar-brand" th:href="@{/index}">
21             Student Registration and Login System</a>
22         <button class="navbar-toggler" type="button"
23             data-bs-toggle="collapse" data-bs-target="#
24             navbarSupportedContent" aria-controls=""
25             navbarSupportedContent" aria-expanded="false" aria-
26             label="Toggle navigation">
27             <span class="navbar-toggler-icon"></span>
28         </button>
29         <div class="collapse navbar-collapse" id="
30             navbarSupportedContent">
31             <ul class="navbar-nav me-auto mb-2 mb-lg-
32             0">
33                 <li class="nav-item">
34                     <a class="nav-link active" aria-
35             current="page" th:href="@{/login}">Login</a>
36                 </li>
37             </ul>
38         </div>
39     </div>
40 </nav>
41 <br /><br /><br />
```

```
29 <div class="container">
30     <div class="row col-md-8 offset-md-2">
31         <div class="card">
32             <div class="card-header">
33                 <h2 class="text-center">Registration
34             </h2>
35             <div th:if="${param.success}">
36                 <div class="alert alert-info">
37                     You have successfully registered
38                     our app!
39                 </div>
40             </div>
41             <div class="card-body">
42                 <form
43                     method="post"
44                     role="form"
45                     th:action="@{/register/save}"
46                     th:object="${student}"
47                 >
48                     <div class="form-group mb-3">
49                         <label class="form-label">
50                             First Name</label>
51                         <input
52                             class="form-control"
53                             id="firstName"
54                             name="firstName"
55                             placeholder="Enter
56                             first name"
57                             th:field="*{firstName}
58                             type="text"
59                             />
60                         <p th:errors = "*{firstName}"
61                             class="text-danger"
62                             th:if="#{fields.hasErrors
63                             ('firstName')}"></p>
64                     </div>
65                     <div class="form-group mb-3">
66                         <label class="form-label">
```

```
62 Last Name</label>
63             <input
64                     class="form-control"
65                     id="lastName"
66                     name="lastName"
67                     placeholder="Enter
68                     last name"
69                     type="text"
70             />
71             <p th:errors = "*{lastName}"
72                     class="text-danger"
73                     th:if="#fields.
74                     hasErrors('lastName')}"></p>
75             </div>
76
77             <div class="form-group mb-3">
78                 <label class="form-label">
79                     Email</label>
80                     <input
81                             class="form-control"
82                             id="email"
83                             name="email"
84                             placeholder="Enter
85                             email address"
86                             type="email"
87                     />
88                     <p th:errors = "*{email}"
89                     class="text-danger"
90                     th:if="#fields.
91                     hasErrors('email')}"></p>
92             </div>
93
94             <div class="form-group mb-3">
95                 <label class="form-label">
```

```
94                               name="password"
95                               placeholder="Enter
96                               password"
97                               th:field="*{password}
98                               type="password"
99                               />
100                             <p th:errors = "*{password}"
101                               class="text-danger"
102                               th:if="#{fields.
103                               hasErrors('password')}"></p>
104                               </div>
105                               <div class="form-group">
106                               <button class="btn btn-
107                               primary" type="submit">Register</button>
108                               <span>Already registered? <a
109                               th:href="@{/login}">Login here</a></span>
110                               </div>
111 </body>
112 </html>
113
```

```
1 <!DOCTYPE html>
2 <html lang="en" xmlns:th="http://www.thymeleaf.org">
3 <head>
4     <meta charset="UTF-8">
5     <title>Student Registration and Login System</title>
6     <link href="https://cdn.jsdelivr.net/npm/
bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet"
      integrity="sha384-EVSTQN3/
azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65Vohhp00mLA
SjC" crossorigin="anonymous">
7 </head>
8 <body>
9 <nav class="navbar navbar-expand-lg navbar-dark bg-
dark">
10    <div class="container-fluid">
11        <a class="navbar-brand" th:href="@{/index}">
12            Student Registration and Login System</a>
13        <button class="navbar-toggler" type="button"
14            data-bs-toggle="collapse" data-bs-target="#
15            navbarSupportedContent" aria-controls=""
16            navbarSupportedContent" aria-expanded="false" aria-
17            label="Toggle navigation">
18            <span class="navbar-toggler-icon"></span>
19        </button>
20        <div class="collapse navbar-collapse" id="
21            navbarSupportedContent">
22            <ul class="navbar-nav me-auto mb-2 mb-lg-
23            0">
24                <li class="nav-item">
25                    <a class="nav-link active" aria-
26                    current="page" th:href="@{/logout}">Logout</a>
27                </li>
28            </ul>
29        </div>
30    </div>
31 </nav>
32 <div class="container">
33     <div class="row col-md-10">
34         <h2>List of Registered Students</h2>
35     </div>
```

```
28 <table class="table table-bordered table-hover">
29     <thead class="table-dark">
30         <tr>
31             <th>First Name</th>
32             <th>Last Name</th>
33             <th>Email</th>
34         </tr>
35     </thead>
36     <tbody>
37         <tr th:each = "student : ${students}">
38             <td th:text = "${student.firstName}"></td>
39             <td th:text = "${student.lastName}"></td>
40             <td th:text = "${student.email}"></td>
41         </tr>
42     </tbody>
43 </table>
44 </div>
45 </body>
46 </html>
47
```