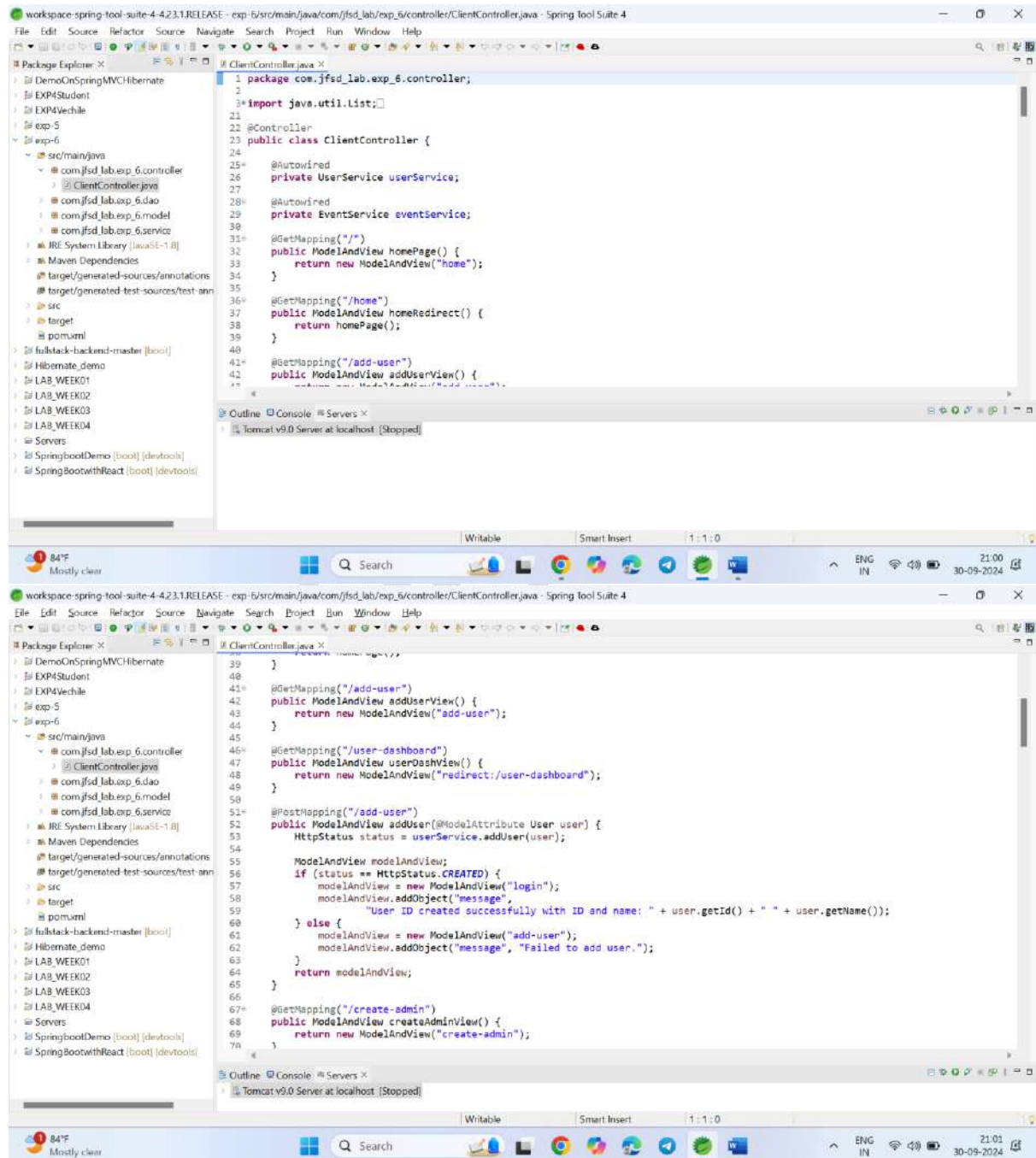


JFSD_LAB_EXPERIMENT_6

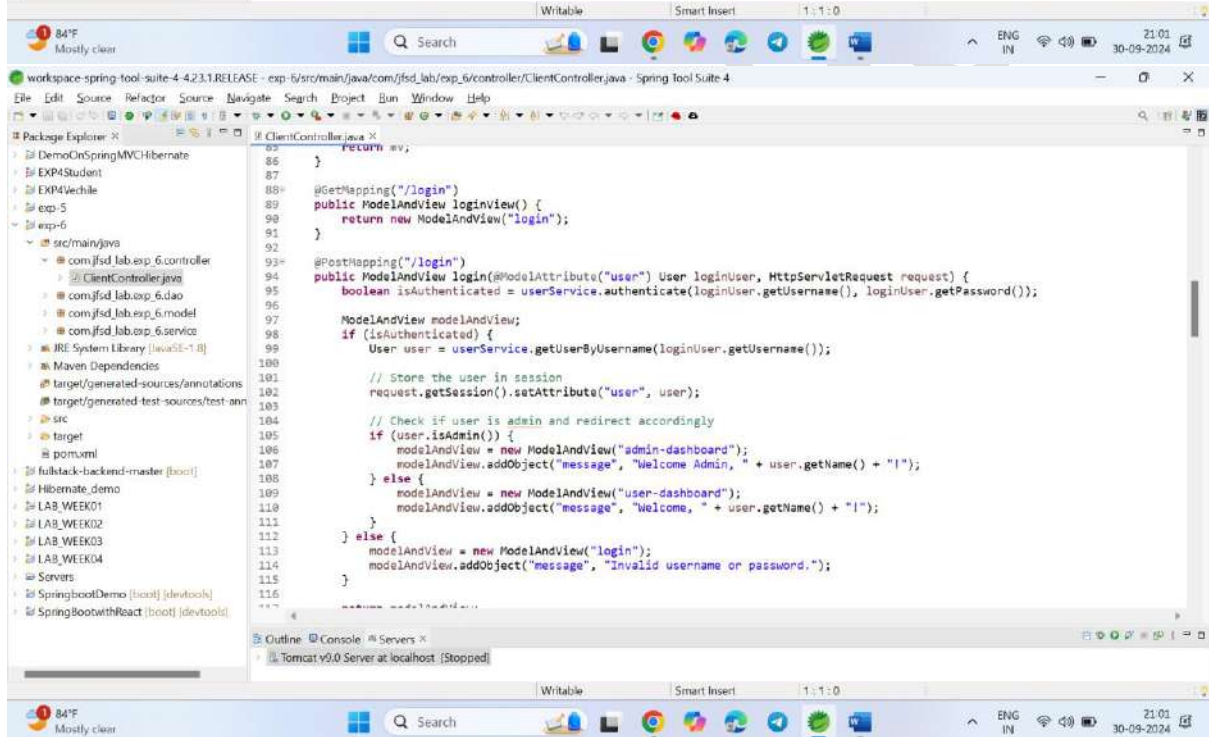
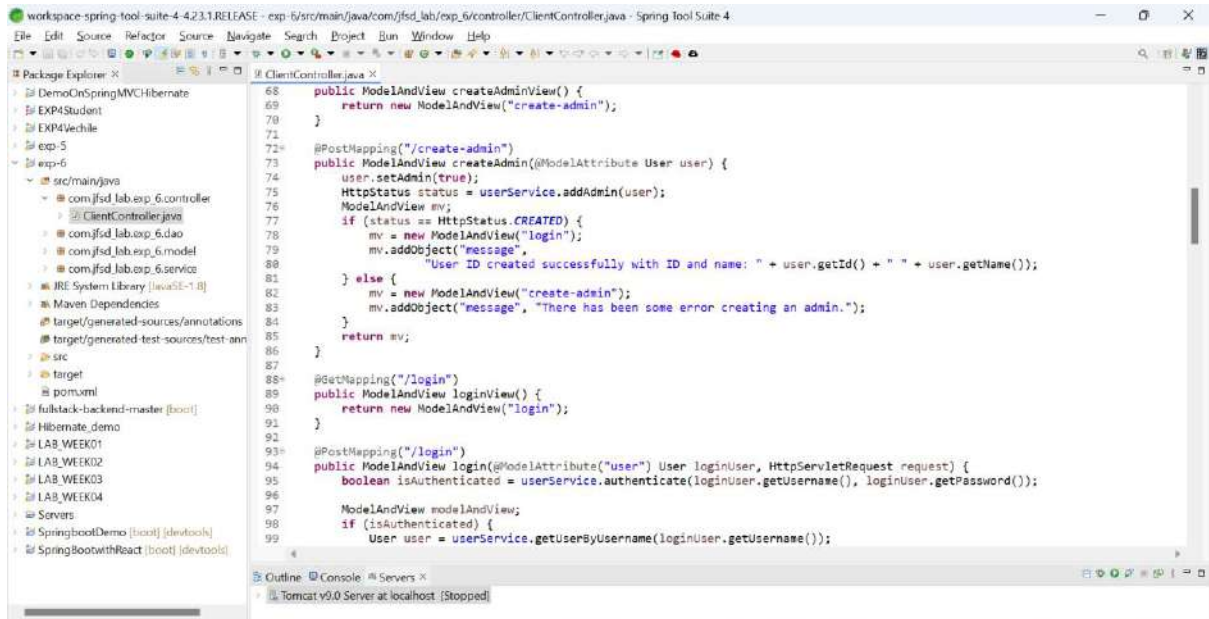
Name : p.uday kiriti

Id : 2200031965

Sec : 15



```
1 package com.jfsd_lab.exp_6.controller;
2
3 import java.util.List;
4
5 @Controller
6 public class ClientController {
7
8     @Autowired
9     private UserService userService;
10
11     @Autowired
12     private EventService eventService;
13
14     @GetMapping("/")
15     public ModelAndView homePage() {
16         return new ModelAndView("home");
17     }
18
19     @GetMapping("/home")
20     public ModelAndView homeRedirect() {
21         return homePage();
22     }
23
24     @GetMapping("/add-user")
25     public ModelAndView addUserView() {
26         // ModelAndView ModelAndView("add-user");
27     }
28
29     @GetMapping("/user-dashboard")
30     public ModelAndView userDashView() {
31         return new ModelAndView("redirect:/user-dashboard");
32     }
33
34     @PostMapping("/add-user")
35     public ModelAndView addUser(@ModelAttribute User user) {
36         HttpStatus status = userService.addUser(user);
37
38         ModelAndView modelAndView;
39         if (status == HttpStatus.CREATED) {
40             modelAndView = new ModelAndView("login");
41             modelAndView.addObject("message",
42                 "User ID created successfully with ID and name: " + user.getId() + " " + user.getName());
43         } else {
44             modelAndView = new ModelAndView("add-user");
45             modelAndView.addObject("message", "Failed to add user.");
46         }
47         return modelAndView;
48     }
49
50     @GetMapping("/create-admin")
51     public ModelAndView createAdminView() {
52         return new ModelAndView("create-admin");
53     }
54 }
```



workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/controller/ClientController.java - Spring Tool Suite 4

```
116  
117     return modelAndView;  
118 }  
119  
120  
121 @GetMapping("/logout")  
122 public ModelAndView logout(HttpSession session) {  
123     // Invalidate the session  
124     session.invalidate();  
125  
126     // Redirect to login page with a message  
127     ModelAndView modelAndView = new ModelAndView("login");  
128     modelAndView.addObject("message", "You have successfully logged out.");  
129     return modelAndView;  
130 }  
131  
132 @GetMapping("/add-event")  
133 public ModelAndView addEventView() {  
134     List<User> allUsers = userService.getAllUsers(); // fetch all users from UserService  
135     ModelAndView mv = new ModelAndView("add-event");  
136     mv.addObject("allUsers", allUsers);  
137     mv.addObject("event", new Event());  
138     return mv;  
139 }  
140  
141 @PostMapping("/add-event")  
142 public ModelAndView addEvent(@ModelAttribute Event event, HttpServletRequest request) {  
143     HttpStatus status = eventService.createEvent(event);  
144     if (status == HttpStatus.CREATED) {  
145         // Redirect to admin dashboard after successful event creation  
146         mv = new ModelAndView("redirect:/admin-dashboard");  
147     }  
148 }
```

Outline Console Servers
Tomcat v9.0 Server at localhost [Stopped]

workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/controller/ClientController.java - Spring Tool Suite 4

```
142     HttpStatus status = eventService.createEvent(event);  
143     ModelAndView mv;  
144     if (status == HttpStatus.CREATED) {  
145         // Redirect to admin dashboard after successful event creation  
146         mv = new ModelAndView("redirect:/admin-dashboard");  
147         request.getSession().setAttribute("message",  
148             "Event created successfully with ID and title: " + event.getId() + " " + event.getTitle());  
149     } else {  
150         mv = new ModelAndView("add-event");  
151         mv.addObject("message", "There has been some error creating an event.");  
152     }  
153     return mv;  
154 }  
155  
156 @GetMapping("/admin-dashboard")  
157 public ModelAndView adminDashView() {  
158     return new ModelAndView("admin-dashboard");  
159 }  
160  
161 @GetMapping("/manage-events")  
162 public ModelAndView manageEventsView() {  
163     ModelAndView mv = new ModelAndView("manage-events");  
164     List<Event> allEvents = eventService.getAllEvents(); // Fetch all events  
165     mv.addObject("allEvents", allEvents); // Add events to the model  
166     return mv;  
167 }  
168  
169 // EDIT Event - GET (load event details for editing)  
170 @GetMapping("/editEvent/{id}")  
171 public ModelAndView editEventView(@PathVariable("id") int eventId) {  
172     Event event = eventService.getEventById(eventId); // Fetch the event by ID  
173     ModelAndView mv = new ModelAndView("edit-event");  
174     mv.addObject("event", event);  
175 }
```

Outline Console Servers
Tomcat v9.0 Server at localhost [Stopped]

workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/controller/ClientController.java - Spring Tool Suite 4

```
169 // EDIT Event - GET (load event details for editing)
170 @GetMapping("/editEvent/{id}")
171 public ModelAndView editEvent(@PathVariable("id") int eventId) {
172     Event event = eventService.getEventById(eventId); // Fetch the event by ID
173     ModelAndView mv = new ModelAndView("edit-event");
174     mv.addObject("allUsers", userService.getAllUsers()); // Add this line
175     mv.addObject("event", event); // Pass the whole event object to the view
176     return mv;
177 }
178
179 @PostMapping("/editEvent/{id}")
180 public ModelAndView editEvent(@PathVariable("id") int eventId, @ModelAttribute Event updatedEvent,
181     HttpServletRequest request) {
182     updatedEvent.setId(eventId);
183     HttpStatus status = eventService.updateEvent(updatedEvent);
184     ModelAndView mv;
185     if (status == HttpStatus.OK) {
186         mv = new ModelAndView("redirect:/manage-events");
187         request.getSession().setAttribute("message", "Event updated successfully: " + updatedEvent.getTitle());
188     } else {
189         mv = new ModelAndView("edit-event");
190         mv.addObject("message", "Error updating the event.");
191         mv.addObject("event", updatedEvent);
192         mv.addObject("allUsers", userService.getAllUsers()); // Add this line
193     }
194     return mv;
195 }
196
197 // DELETE Event
198 @GetMapping("/deleteEvent/{id}")
199 public ModelAndView deleteEvent(@PathVariable("id") int eventId, HttpServletRequest request) {
200     HttpStatus status = eventService.deleteEvent(eventId); // Delete the event by ID
```

Outline Console Servers
Tomcat v9.0 Server at localhost [Stopped]

workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/controller/ClientController.java - Spring Tool Suite 4

```
197 // DELETE Event
198 @GetMapping("/deleteEvent/{id}")
199 public ModelAndView deleteEvent(@PathVariable("id") int eventId, HttpServletRequest request) {
200     HttpStatus status = eventService.deleteEvent(eventId); // Delete the event by ID
201     ModelAndView mv;
202     if (status == HttpStatus.OK) {
203         // Redirect to manage events after successful deletion
204         mv = new ModelAndView("redirect:/manage-events");
205         request.getSession().setAttribute("message", "Event deleted successfully.");
206     } else {
207         mv = new ModelAndView("manage-events");
208         mv.addObject("message", "Error deleting the event.");
209     }
210     return mv;
211 }
212
213 // User Register events view
214 @GetMapping("/register-events")
215 public ModelAndView registerEventView() {
216     ModelAndView mv = new ModelAndView("register-events");
217     List<Event> allEvents = eventService.getAllEvents(); // Fetch all events
218     mv.addObject("allEvents", allEvents); // Add events to the model
219     return mv;
220 }
221
222 @GetMapping("/registerEvent/{eventId}")
223 public ModelAndView registerEvent(@PathVariable("eventId") int eventId, HttpSession session) {
224     ModelAndView mv = new ModelAndView("register-events");
225
226     User user = (User) session.getAttribute("user"); // Get the logged-in user from session
227     if (user != null) {
228         HttpStatus status = eventService.registerUserToEvent(eventService.getEventById(eventId), user);
```

Outline Console Servers
Tomcat v9.0 Server at localhost [Stopped]

workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/controller/ClientController.java - Spring Tool Suite 4

File Edit Source Refactor Source Navigate Search Project Run Window Help

Package Explorer

- src/main/java
 - com.jfsd_lab.exp_6.controller
 - ClientController.java

```
230 mv.addObject("message", "You have successfully registered for the event.");
231 } else if (status == HttpStatus.NOT_FOUND) {
232     mv.addObject("message", "Event not found.");
233 } else {
234     mv.addObject("message", "Error occurred while registering for the event. Please try again.");
235 }
236 } else {
237     mv.addObject("message", "Please login to register for an event.");
238 }
239
240 // Fetch the updated list of events to display after registration attempt
241 mv.addObject("allEvents", eventService.getAllEvents());
242 return mv;
243 }
244
245 @GetMapping("/registered-events")
246 public ModelAndView viewRegisteredEvents(HttpSession session) {
247     User user = (User) session.getAttribute("user"); // Get the logged-in user
248     if (user != null) {
249         List<Event> registeredEvents = eventService.getRegisteredEventsForUser(user); // Fetch events by user
250         ModelAndView mv = new ModelAndView("registered-events"); // Load the JSP view
251         mv.addObject("registeredEvents", registeredEvents); // Pass the registered events to the view
252         return mv;
253     }
254     ModelAndView mv = new ModelAndView("login");
255     mv.addObject("message", "Please log in to view registered events.");
256     return mv;
257 }
258 }
259
260 }
261 }
```

Outline Console Servers

Tomcat v9.0 Server at localhost [Stopped]

workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/dao/EventsDAO_Impl.java - Spring Tool Suite 4

File Edit Source Refactor Source Navigate Search Project Run Window Help

Package Explorer

- src/main/java
 - com.jfsd_lab.exp_6.dao
 - EventsDAO_Impl.java

```
1 package com.jfsd_lab.exp_6.dao;
2
3 import java.util.ArrayList;
4
14 public class EventsDAO_Impl implements EventsDAO {
15
16     private HibernateTemplate ht;
17
18     public HibernateTemplate getHt() {
19         return ht;
20     }
21
22     public void setHt(HibernateTemplate ht) {
23         this.ht = ht;
24     }
25
26     @Override
27     @Transactional
28     public HttpStatus createEvent(Event e) {
29         try {
30             ht.saveOrUpdate(e);
31             return HttpStatus.CREATED;
32         } catch (Exception ex) {
33             System.out.println("Error creating event: " + ex.getMessage());
34             return HttpStatus.INTERNAL_SERVER_ERROR;
35         }
36     }
37
38     @Override
39     @Transactional
40     public HttpStatus registerUserToEvent(Event e, User user) {
41         try {

```

Outline Console Servers

Tomcat v9.0 Server at localhost [Stopped]

workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/dao/EventsDAO_Impl.java - Spring Tool Suite 4

File Edit Source Refactor Source Navigate Search Project Run Window Help

Package Explorer: com.jfsd_lab.exp_6.controller, com.jfsd_lab.exp_6.dao, EventsDAO_Impl.java, EventsDAO.java, UserDAO_Impl.java, UserDAO.java, com.jfsd_lab.exp_6.model, com.jfsd_lab.exp_6.service, JRE System Library [JavaSE-1.8], Maven Dependencies, target/generated-sources/annotation, target/generated-test-sources/test-annotation, src, pom.xml, fullstack-backend-master [boot], Hibernate_demo, LAB_WEEK01, LAB_WEEK02, LAB_WEEK03, LAB_WEEK04, Servers, CoderushofDemos (Front-End-Backend)

```
35 }
36 }
37 }
38 @Override
39 @Transactional
40 public HttpStatus registerUserToEvent(Event e, User user) {
41     try {
42         // Ensure the list of registered users is initialized
43         if (e.getRegisteredUsers() == null) {
44             e.setRegisteredUsers(new ArrayList<>());
45         }
46         // Add the user to the list of registered users
47         e.getRegisteredUsers().add(user);
48         // Save or update the event
49         ht.saveOrUpdate(e);
50
51         return HttpStatus.OK;
52     } catch (Exception ex) {
53         System.out.println("Error registering user to event: " + ex.getMessage());
54         return HttpStatus.INTERNAL_SERVER_ERROR;
55     }
56 }
57
58 @Override
59 public List<User> getUsersRegisteredToEvent(Event e) {
60     try {
61         return ht.loadAll(User.class).stream()
62             .filter(user -> e.getRegisteredUsers().stream()
63                 .anyMatch(registeredUser -> registeredUser.getId() == user.getId()))
64             .collect(Collectors.toList());
65     } catch (Exception ex) {
66         System.out.println("Error fetching users registered to event: " + ex.getMessage());
67         return null;
68     }
69 }
```

Outline Console Servers

Tomcat v9.0 Server at localhost [Stopped]

workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/dao/EventsDAO_Impl.java - Spring Tool Suite 4

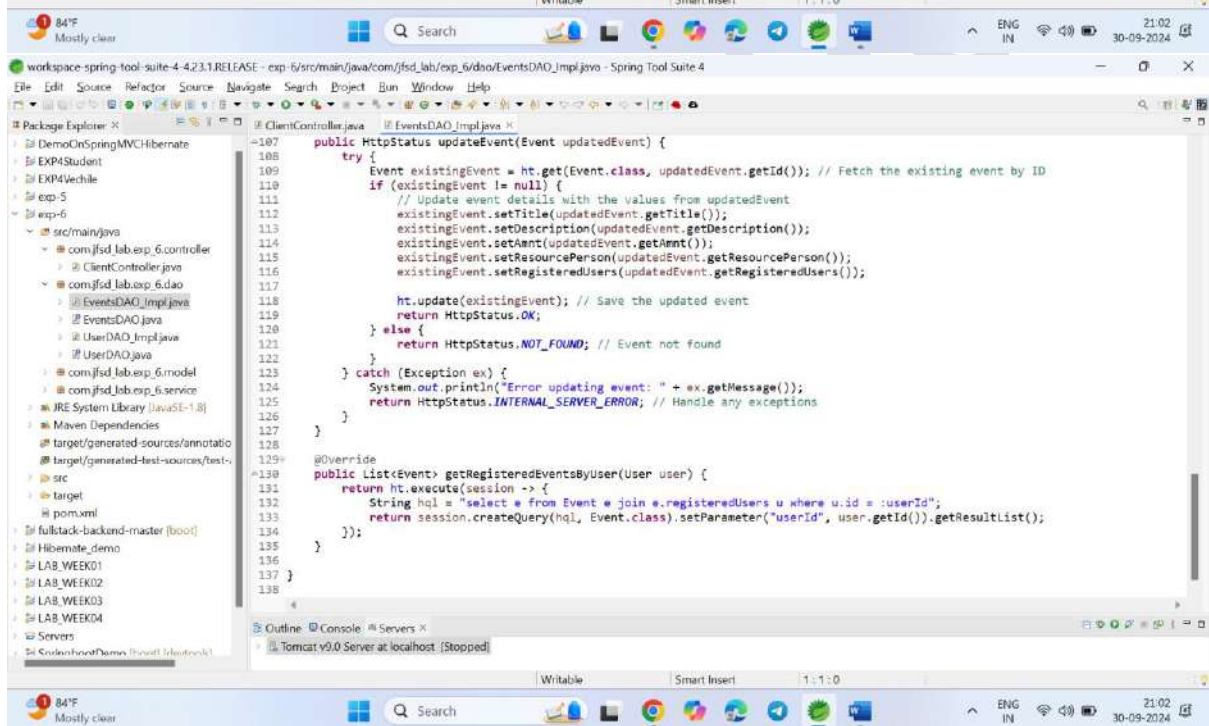
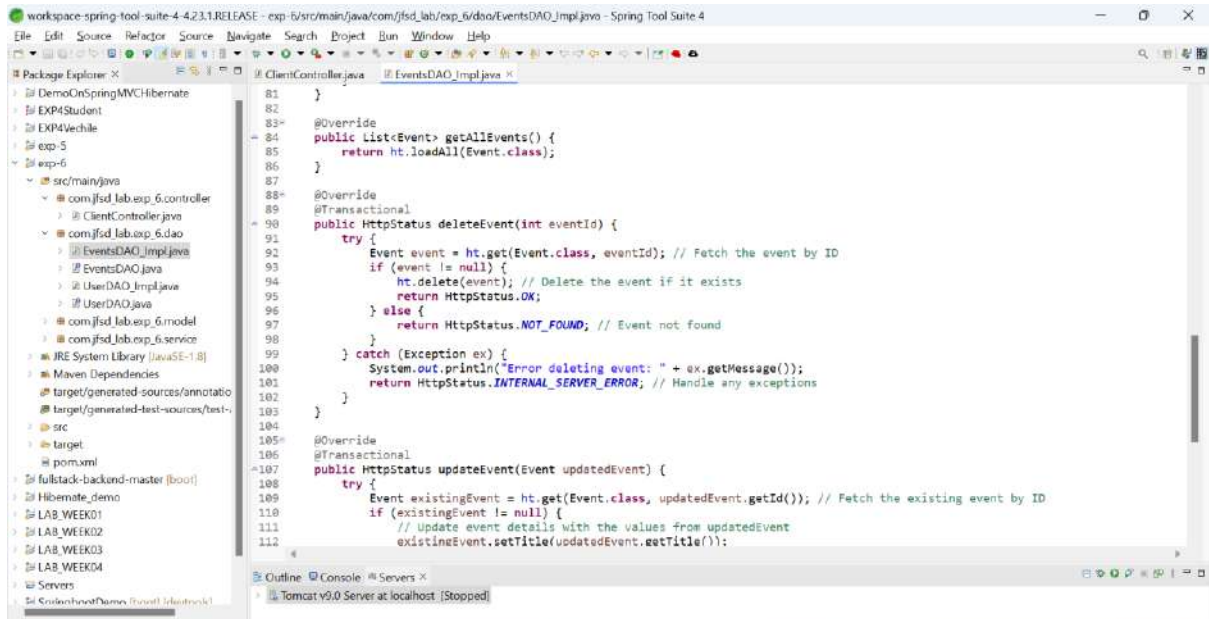
File Edit Source Refactor Source Navigate Search Project Run Window Help

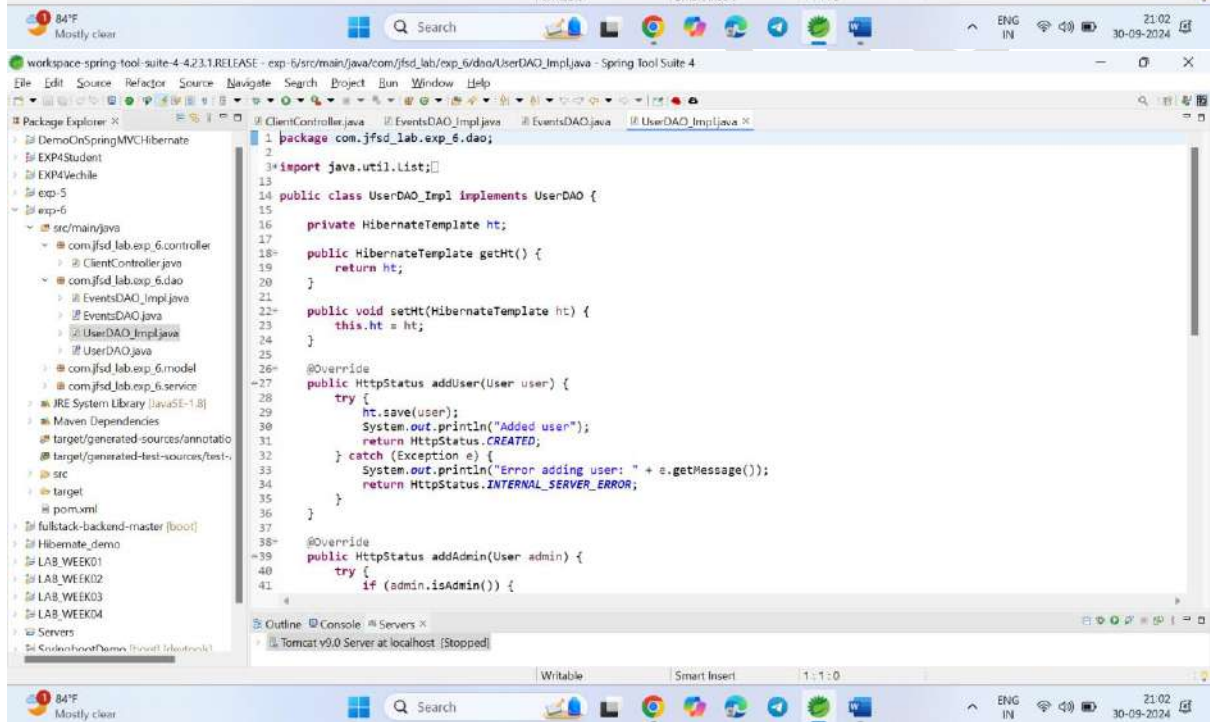
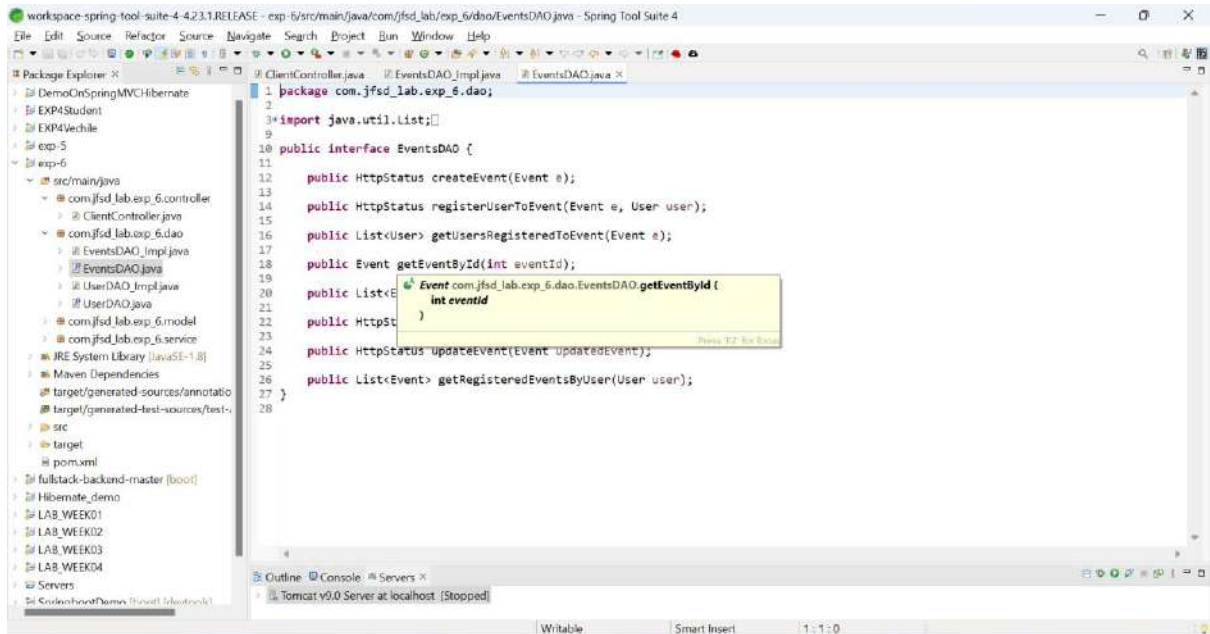
Package Explorer: com.jfsd_lab.exp_6.controller, com.jfsd_lab.exp_6.dao, EventsDAO_Impl.java, EventsDAO.java, UserDAO_Impl.java, UserDAO.java, com.jfsd_lab.exp_6.model, com.jfsd_lab.exp_6.service, JRE System Library [JavaSE-1.8], Maven Dependencies, target/generated-sources/annotation, target/generated-test-sources/test-annotation, src, pom.xml, fullstack-backend-master [boot], Hibernate_demo, LAB_WEEK01, LAB_WEEK02, LAB_WEEK03, LAB_WEEK04, Servers, CoderushofDemos (Front-End-Backend)

```
56 }
57 }
58 }
59 }
60 }
61 @Override
62 public List<User> getUsersRegisteredToEvent(Event e) {
63     try {
64         return ht.loadAll(User.class).stream()
65             .filter(user -> e.getRegisteredUsers().stream()
66                 .anyMatch(registeredUser -> registeredUser.getId() == user.getId()))
67             .collect(Collectors.toList());
68     } catch (Exception ex) {
69         System.out.println("Error fetching users registered to event: " + ex.getMessage());
70         return null;
71     }
72 }
73
74 @Override
75 public Event getEventById(int eventId) {
76     try {
77         return ht.get(Event.class, eventId);
78     } catch (Exception ex) {
79         System.out.println("Error fetching event by ID: " + ex.getMessage());
80         return null;
81     }
82 }
83
84 @Override
85 public List<Event> getAllEvents() {
86     return ht.loadAll(Event.class);
87 }
```

Outline Console Servers

Tomcat v9.0 Server at localhost [Stopped]





workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/dao/UserDAO_Impl.java - Spring Tool Suite 4

Package Explorer: DemoOnSpringMVCHibernate, EXP4Student, EXP4Vehicle, exp-5, exp-6, src/main/java, com.jfsd_lab.exp_6.controller, ClientController.java, com.jfsd_lab.exp_6.dao, EventsDAO_Impl.java, EventsDAO.java, UserDAO_Impl.java, UserDAO.java, com.jfsd_lab.exp_6.model, com.jfsd_lab.exp_6.service, JRE System Library [JavaSE-1.8], Maven Dependencies, target/generated-sources/annotation, target/generated-test-sources/test-annotation, src, target, pom.xml, fullstack-backend-master [boot], Hibernate_demo, LAB_WEEK01, LAB_WEEK02, LAB_WEEK03, LAB_WEEK04, Servers, ControllerTestDemo (Front) [debugging]

```
34 }
35 }
36 }
37 }
38 }
39 @Override
40 public HttpStatus addAdmin(User admin) {
41     try {
42         if (admin.isAdmin()) {
43             ht.save(admin);
44             System.out.println("Added admin");
45             return HttpStatus.CREATED;
46         } else {
47             System.out.println("Check if user is actually admin");
48             return HttpStatus.FORBIDDEN;
49         }
50     } catch (Exception e) {
51         System.out.println("Error adding admin: " + e.getMessage());
52         return HttpStatus.INTERNAL_SERVER_ERROR;
53     }
54 }
55 @Override
56 public List<User> getAllUsers() {
57     try {
58         return ht.loadAll(User.class);
59     } catch (Exception e) {
60         System.out.println("Error fetching users: " + e.getMessage());
61         return null;
62     }
63 }
64 }
65 @Override
```

Outline: Console: Servers: Tomcat v9.0 Server at localhost [Stopped]

workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/dao/UserDAO_Impl.java - Spring Tool Suite 4

Package Explorer: DemoOnSpringMVCHibernate, EXP4Student, EXP4Vehicle, exp-5, exp-6, src/main/java, com.jfsd_lab.exp_6.controller, ClientController.java, com.jfsd_lab.exp_6.dao, EventsDAO_Impl.java, EventsDAO.java, UserDAO_Impl.java, UserDAO.java, com.jfsd_lab.exp_6.model, com.jfsd_lab.exp_6.service, JRE System Library [JavaSE-1.8], Maven Dependencies, target/generated-sources/annotation, target/generated-test-sources/test-annotation, src, target, pom.xml, fullstack-backend-master [boot], Hibernate_demo, LAB_WEEK01, LAB_WEEK02, LAB_WEEK03, LAB_WEEK04, Servers, ControllerTestDemo (Front) [debugging]

```
51 }
52 }
53 }
54 }
55 @Override
56 public List<User> getAllUsers() {
57     try {
58         return ht.loadAll(User.class);
59     } catch (Exception e) {
60         System.out.println("Error fetching users: " + e.getMessage());
61         return null;
62     }
63 }
64 }
65 @Override
66 public User findByUsername(String username) {
67     return ht.execute(new HibernateCallback<User>() {
68         @Override
69         public User doInHibernate(Session session) {
70             // Use TypedQuery instead of deprecated Query<User>
71             TypedQuery<User> query = session.createQuery("FROM User WHERE username = :username", User.class);
72             query.setParameter("username", username);
73             List<User> users = query.getResultList();
74             if (users != null && !users.isEmpty()) {
75                 return users.get(0);
76             }
77             return null; // User not found
78         }
79     });
80 }
81 }
82 }
```

Outline: Console: Servers: Tomcat v9.0 Server at localhost [Stopped]

workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/dto/UserDAO.java - Spring Tool Suite 4

File Edit Source Refactor Source Navigate Search Project Run Window Help

Package Explorer

- com.jfsd_lab.exp_6.controller
 - ClientController.java
- com.jfsd_lab.exp_6.dao
 - EventsDAO.java
 - EventsDAO_Impl.java
 - UserDAO.java
 - UserDAO_Impl.java
- com.jfsd_lab.exp_6.model
- com.jfsd_lab.exp_6.service
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- target/generated-sources/annotation
- target/generated-test-sources/test-annotation
- src
- target
- pom.xml

```
1 package com.jfsd_lab.exp_6.dao;
2
3 import java.util.List;
4
5 public interface UserDAO {
6
7     public HttpStatus addUser(User user);
8
9     public HttpStatus addAdmin(User admin);
10
11     User findByUsername(String username);
12
13     public List<User> getAllUsers();
14 }
15
```

Outline Console Servers

Tomcat v9.0 Server at localhost [Stopped]

workspace-spring-tool-suite-4-4.23.1.RELEASE - exp-6/src/main/java/com/jfsd_lab/exp_6/model/Event.java - Spring Tool Suite 4

File Edit Source Refactor Source Navigate Search Project Run Window Help

Package Explorer

- com.jfsd_lab.exp_6.controller
 - ClientController.java
- com.jfsd_lab.exp_6.dao
 - EventsDAO.java
 - EventsDAO_Impl.java
 - UserDAO.java
 - UserDAO_Impl.java
- com.jfsd_lab.exp_6.model
 - Event.java
 - User.java
- com.jfsd_lab.exp_6.service
- JRE System Library [JavaSE-1.8]
- Maven Dependencies
- target/generated-sources/annotation
- target/generated-test-sources/test-annotation
- src
- target
- pom.xml

```
1 package com.jfsd_lab.exp_6.model;
2
3 import java.util.List;
4
5 @Entity
6 public class Event {
7
8     @Id
9     @GeneratedValue(strategy = GenerationType.IDENTITY)
10     private int id;
11
12     @Column(nullable = false, unique = true)
13     private String title;
14
15     private String description;
16
17     private int amnt;
18
19     // Relationship with the resource person (User)
20     @ManyToOne
21     @JoinColumn(name = "resource_person_id", referencedColumnName = "id")
22     private User resourcePerson;
23
24     // Relationship with the registered users (User)
25     @ManyToMany(fetch = FetchType.EAGER)
26     @JoinTable(name = "event_registered_users", joinColumns = @JoinColumn(name = "event_id"), inverseJoinColumns = @JoinColumn(name = "user_id"))
27     private List<User> registeredUsers;
28
29     // Getters and Setters
30     public int getId() {
31         return id;
32     }
33 }
34
```

Outline Console Servers

Tomcat v9.0 Server at localhost [Stopped]

