

Project: Task Manager Application

Developed by: Karthiga Gujuluva Ravichandran

Email: (karthiga.ravic@hcl.com)

Created on Date: 02/21/2021

Prototype:

The Task Manager Application consists of two modes (One for User and other for Admin) with functionalities in each mode. The user interaction will be through a webpage of link ([Welcome Page](#)) which leads to Welcome Page where the user or admin can register or login. User has privileges to create/update/delete tasks. Admin has privileges to delete users and their tasks. Web Security is implemented in this project. Passwords are encrypted and page is opened only when the privilege is given to the user or admin. The final Application will be developed in various sprints based on this prototype. The prototype explains the functionalities and format of the final project.

Sprint Plan:

Task Manager Application project is divided into **4 sprints**. Each sprint consists of **1 week** of time. The tasks achieved in the sprints are given below:

Sprint 1:

Time allocated for this sprint is **40 hours**. The tasks involved in this sprint are:

Tasks:

1. Development of Welcome screen, Login screen, Register Screen
2. Welcome screen should have 2 options i.e., Registration button and login button.
3. Login screen has 2 inputs namely username and password and a submit button to proceed to Home page.
4. The username and password should be required fields and can accept all characters.
5. Login button searches if user already exists in the database and proceeds to login success screen with Web Security module.
6. Database is created for the application and fed with initial data.
7. Linking all screens and the database through application.properties file.

8. JSP is used to create screens and Service, model, Repository, Entity, Security Configuration files are used to get data from DB, validate and display on the webpage.
9. Users, Tasks, Roles, User Role tables are created to store all data and they are fed with initial data.
10. Development of registration screen that has first name, last name, username, password, email, role, activity, and a register button.
11. Register button leads to login screen. All are required fields.
12. Error messages are thrown in login screen if incorrect data is given in the fields.
13. Save method is created to save the data to table.
14. Find user functionality checks if user already exists in table.
15. Appropriate messages are thrown to the user if save happens or fails.

Sprint 2:

Time allocated for this sprint is **40 hours**. The tasks involved in this sprint are:

Tasks:

1. Home Page and its functionalities are handled here.
2. It has 2 buttons for creating tasks and viewing tasks.
3. Create task leads to create task page and it has fields for entering data in task name, email, severity, start and end dates.
4. The data is saved to table with method in repository and save or error message appear on the Home Page after Save or Error on save.
5. View Tasks Page has all the tasks that has been created by the user and has delete and edit buttons for each task.
6. Clicking delete button should lead to Delete Task Page and it shows all the details of that task.
7. On clicking delete button leads to deletion of that task and error or success message is shown in View Tasks Page.
8. Edit button leads to Update Task Page and it shows the details of that task and it can be edited.
9. Clicking update button leads to updating of that task and error or success message is shown in View Tasks Page.
10. Back to Options Page leads to Home Page.
11. Logout button leads to Login screen.

Sprint 3:

Time allocated for this sprint is **40 hours**. The tasks involved in this sprint are:

Tasks:

1. When admin credentials are given in the login screen, it should lead to Welcome Admin Page.
2. This page shows all the users with their details.
3. Each user in the table has delete button. Admin has rights to delete a user from the table along with all the tasks that has been created by that user.
4. Clicking the delete button leads to deletion of that user and all tasks related to that user.
5. It leads to Delete Another User Screen and has button to delete another user which leads to Welcome Admin Page.
6. Logout leads to Login Page.
7. Web security is used to handle user and admin access to pages.

Sprint 4:

Time allocated for this sprint is **40 hours**. The tasks involved in this sprint are:

Tasks:

1. Logger is used to log all the tasks done by the user.
2. Password encryption is done to store the password credentials using MyUserDetailsService methods.
3. UI alignment, custom exception handling, performance improvement is done in this sprint.
4. All the bugs are fixed in this sprint and testing is done.

Task Manager Application - Link

<https://github.com/Karthiga-web/Phase3EndProject.git> is the repository link for my Task Manager Application in GitHub.

Algorithm:

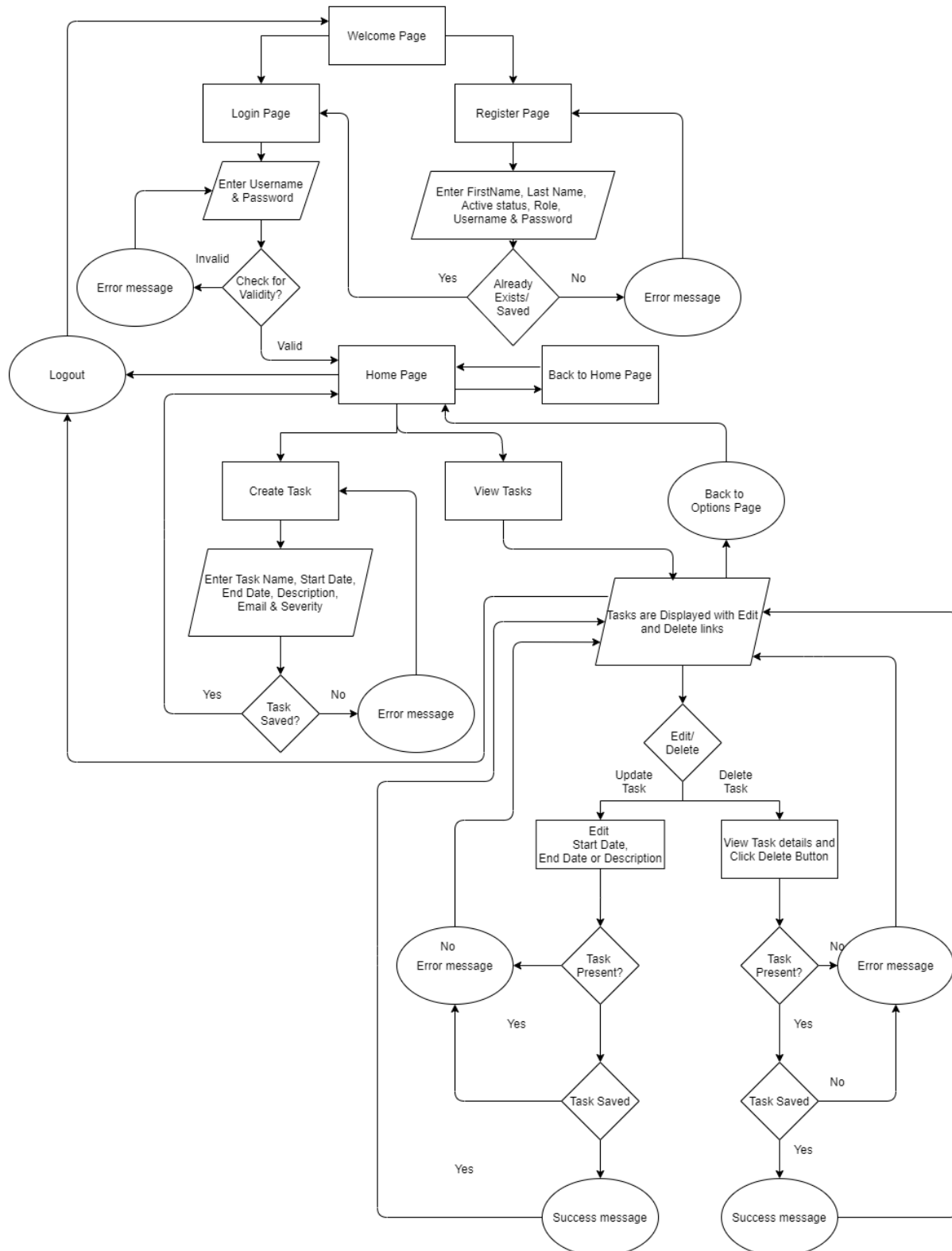
1. The project opens with Welcome Page that has two buttons for Login and Registration.
2. Register Page has fields for first name, last name, username, password, active, role.
3. When all fields are filled with data and register button is clicked, the user is registered, and it leads to login screen with message that user is registered.
4. The login screen has username, password and when the fields are filled with data and login button is clicked, it leads to Home Page.

5. Home Page has Create Task and View Tasks button and Logout link.
6. Create Task button leads to Create Task Page which has Task name, start and end dates, email, description, severity and create task button.
7. When all fields are filled with correct data and button is clicked, task is created.
8. When incorrect data are entered, it leads to error page with exception handler.
9. View Tasks page shows all the tasks for that user with links to edit and delete the tasks created by him.
10. It also has Logout link and Back to Options Page button.
11. Edit button leads to Update the Task page which shows the data of that task and user can edit the data for some fields and update the task with update button. The updated task is saved to Database.
12. Errors in updating is handled by exception handlers and error messages are thrown to the View Tasks Page.
13. Delete Button leads to Delete Page and Task Data are shown for that task, clicking delete button leads to deletion of task and error on deletion leads to error message on view tasks page.
14. Logout link leads to Login Screen with successfully logged out message.
15. Back to Option Page button leads to Home Page.
16. When Admin credentials are given in Login screen, it leads to Welcome Admin Page which shows all the users with their data.
17. Each user has delete button and on clicking the delete button deletes that user and leads to Delete Another User screen that has button to delete another user along with message of users deletion.
18. On clicking that button, it leads to Welcome Admin Page.
19. Logout link leads to Login Screen.
20. If any other links are used while using admin mode through URL change lead to Log Out! Page with no access message and vice versa for user.

Flowchart:

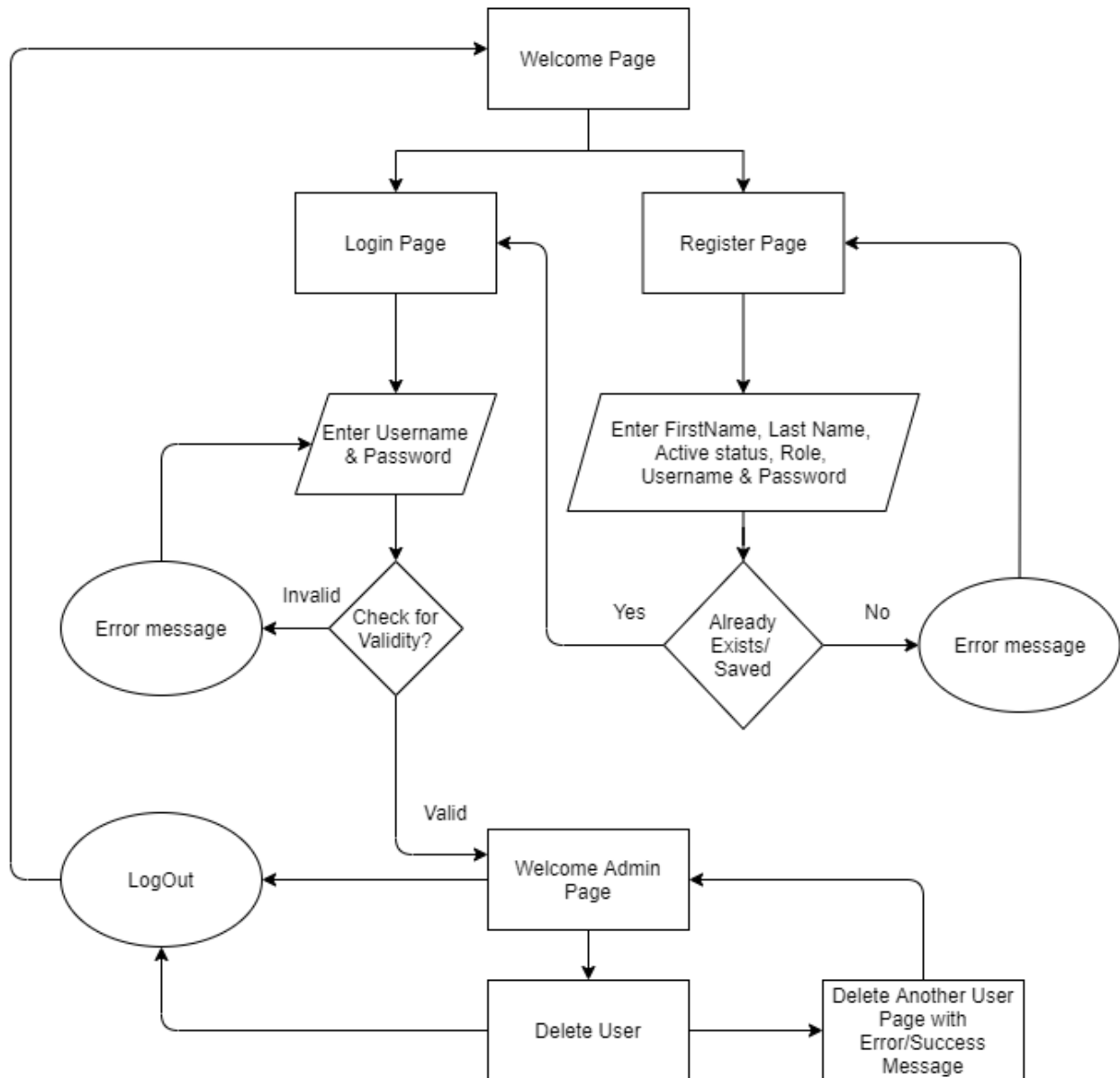
TASK MANAGER APPLICATION

User Mode



Task Manager Application

Admin Mode



Codes:

Java:

```
MyUserDetailsService.java x TaskManagerService.java TaskManagerServiceImpl.java
1 package com.hcl.taskManagerProject.service;
2
3 import org.springframework.beans.factory.annotation.Autowired;
4
5 @Service
6 public class MyUserDetailsService implements UserDetailsService {
7
8     @Autowired
9     private TaskManagerServiceImpl userService;
10
11     @Override
12     @Transactional
13     public UserDetails loadUserByUsername(String username) throws UsernameNotFoundException {
14         Optional<UserEntity> user = userService.findByUserName(username);
15         UserEntity newUser = user.get();
16         List<GrantedAuthority> authorities = getUserAuthority(newUser.getRoles());
17         return buildUserForAuthentication(newUser, authorities);
18     }
19
20     private List<GrantedAuthority> getUserAuthority(Set<Role> userRoles) {
21         Set<GrantedAuthority> roles = new HashSet<GrantedAuthority>();
22         for (Role role : userRoles) {
23             roles.add(new SimpleGrantedAuthority(role.getRole()));
24         }
25         List<GrantedAuthority> grantedAuthorities = new ArrayList<>(roles);
26         return grantedAuthorities;
27     }
28
29     private UserDetails buildUserForAuthentication(UserEntity user, List<GrantedAuthority> authorities) {
30         return new org.springframework.security.core.userdetails.User(user.getUserName(), user.getPassword(),
31             user.getActive(), true, true, true, authorities);
32     }
33 }
```

```
TaskManagerController.java x Role.java x
1 package com.hcl.taskManagerProject.entity;
2
3 import lombok.AllArgsConstructor;
4
5 @Data
6 @Builder
7 @AllArgsConstructor
8 @NoArgsConstructor
9 @Entity
10 @Table(name = "roles")
11 public class Role {
12     @Id
13     @GeneratedValue(strategy = GenerationType.AUTO)
14     @Column(name = "role_id")
15     private int id;
16     @Column(name = "role")
17     private String role;
18 }
```

```

RoleRepository.java x TaskRepository.java UserRepository.java MyUserDetailsSe
1 package com.hcl.taskManagerProject.repository;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5
6
7
8 @Repository
9 public interface RoleRepository extends JpaRepository<Role, Integer> {
10     Role findByRole(String role);
11 }

```

```

SaveExceptionHandler.java x TaskManagerController.java Role.java TaskEntity.java UserEntity.java
1 package com.hcl.taskManagerProject.controller;
2
3 import org.springframework.http.HttpStatus;
4
5
6
7
8
9
10 @ControllerAdvice
11 public class SaveExceptionHandler {
12     @ExceptionHandler(value = {SaveFailedException.class})
13     public ResponseEntity<Object> exception(SaveFailedException exception) {
14         return new ResponseEntity<>("Product not found", HttpStatus.METHOD_FAILURE);
15     }
16 }
17

```

```

SaveFailedException.java x RoleRepository.java TaskRepository.java UserRepository.java
1 package com.hcl.taskManagerProject.exceptions;
2
3 public class SaveFailedException extends RuntimeException {
4     private static final long serialVersionUID = 1L;
5
6     public SaveFailedException() {
7         super("Error occurred on save!");
8     }
9 }
10

```



```
TaskManagerController.java  Role.java  TaskEntity.java X
1 package com.hcl.taskManagerProject.entity;
2
3 import java.time.LocalDate;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20 @Data
21 @NoArgsConstructor
22 @Entity
23 @Table(name = "tasks")
24 public class TaskEntity {
25     @Id
26     @GeneratedValue(strategy = GenerationType.IDENTITY)
27     @Column(name = "taskId")
28     int taskId;
29     @Column(name = "taskName", nullable = false)
30     String taskName;
31     // @NotNull(message = "{task.date.not.null}")
32     @DateTimeFormat(pattern = "yyyy-MM-dd")
33     @Column(name = "startDate")
34     LocalDate startDate;
35     @DateTimeFormat(pattern = "yyyy-MM-dd")
36     @Column(name = "endDate")
37     LocalDate endDate;
38     @Column(name = "description", nullable = false, length = 3200)
39     String description;
40     @Column(name = "email", nullable = false)
41     String email;
42     @Column(name = "severity", nullable = false)
43     String severity;
44     @Column(name = "userId", nullable = false)
45     int userId;
46 }
47
```

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

TaskManagerController.java
1 package com.hcl.taskManagerProject.controller;
2
3 import java.util.List;
4 import java.util.Optional;
5
6 import javax.servlet.http.HttpServletRequest;
7 import javax.servlet.http.HttpServletResponse;
8
9 import org.slf4j.Logger;
10 import org.slf4j.LoggerFactory;
11 import org.springframework.beans.factory.annotation.Autowired;
12 import org.springframework.security.authentication.AnonymousAuthenticationToken;
13 import org.springframework.security.core.Authentication;
14 import org.springframework.security.core.context.SecurityContextHolder;
15 import org.springframework.security.web.authentication.logout.SecurityContextLogoutHandler;
16 import org.springframework.stereotype.Controller;
17 import org.springframework.ui.Model;
18 import org.springframework.ui.ModelMap;
19 import org.springframework.web.bind.annotation.GetMapping;
20 import org.springframework.web.bind.annotation.ModelAttribute;
21 import org.springframework.web.bind.annotation.PathVariable;
22 import org.springframework.web.bind.annotation.PostMapping;
23 import org.springframework.web.bind.annotation.RequestMapping;
24 import org.springframework.web.bind.annotation.RequestMethod;
25 import org.springframework.web.bind.annotation.RequestParam;
26
27 import com.hcl.taskManagerProject.entity.TaskEntity;
28 import com.hcl.taskManagerProject.entity.UserEntity;
29 import com.hcl.taskManagerProject.exceptions.SaveFailedException;
30 import com.hcl.taskManagerProject.service.TaskManagerService;
31
32 @Controller
33 public class TaskManagerController {
34     String currentUserName;
35     int taskId;
36     @Autowired
37     TaskManagerService service;
38
39     Logger logger = LoggerFactory.getLogger(TaskManagerController.class);
40
41     @GetMapping("/")
42     String hello() {
43         logger.info("Mapping to index");
44         return "index";
45     }
46
47     @PostMapping("/decide")
48     public String find(@RequestParam(name = "button") String buttonValue) {
49         logger.info("Finding user clicked which button");
50         if (buttonValue.equals("Login")) {
51             return "login";
52         } else {
53             return "register";
54         }
55     }
56 }
```

TaskManagerController.java

```
54     }
55 }
56
57 @RequestMapping(value = "/login")
58 public String getLoginPage(@RequestParam(name = "error", required = false) String error, Model model,
59     @RequestParam(name = "logout", required = false) String logout) {
60     if (error != null) {
61         model.addAttribute("error", "Invalid Username or Password");
62     }
63     if (logout != null) {
64         model.addAttribute("logout", "You have Successfully Logged Out");
65     }
66     return "login";
67 }
68
69 @RequestMapping(value = "/logout", method = RequestMethod.GET)
70 public String logoutPage(HttpServletRequest request, HttpServletResponse response) {
71     Authentication auth = SecurityContextHolder.getContext().getAuthentication();
72     if (auth != null) {
73         new SecurityContextLogoutHandler().logout(request, response, auth);
74     }
75     return "logout";
76 }
77
78 public int getUserIdMethod() {
79     Authentication authentication = SecurityContextHolder.getContext().getAuthentication();
80     Optional<UserEntity> usercheck = null;
81     if (!(authentication instanceof AnonymousAuthenticationToken)) {
82         String currentUserName = authentication.getName();
83         usercheck = service.findByUserName(currentUserName);
84     }
85     return usercheck.get().getUserId();
86 }
87
88 @RequestMapping(value = "/welcome", method = RequestMethod.GET)
89 String welcome1(ModelMap model) {
90     logger.info("Mapping to index");
91     Authentication authentication = SecurityContextHolder.getContext().getAuthentication();
92     if (authentication.getAuthorities().iterator().next().getAuthority().equalsIgnoreCase("ROLE_ADMIN")) {
93         Iterable<UserEntity> users = service.findAllUsers();
94         model.addAttribute("listUsers", users);
95         return "adminWelcome";
96     }
97     return "welcome";
98 }
99
100 @PostMapping("/register")
101 public String register(@ModelAttribute("user") UserEntity user, ModelMap model) {
102     logger.info("registration page entered");
103     Optional<UserEntity> usercheck = service.findByUserName(user.getUserName());
104     if (usercheck.isPresent()) {
105         model.addAttribute("message", "Username Already Exists!Try a different one!");
106         return "/register";
107     }
108 }
```


TaskManagerController.java x

```
106     return "/register";
107 } else {
108     String save = saveMethod(user);
109     if (save.equals("Saved")) {
110         model.addAttribute("message", "User Information Saved");
111         return "/login";
112     } else {
113         model.addAttribute("message", "Error Occured on Registration. Please Try again!");
114         return "/register";
115     }
116 }
117 }
118
119 public String saveMethod(UserEntity user) {
120     try {
121         service.update(user);
122         return "Saved";
123     } catch (Exception e) {
124         return "Error";
125     }
126 }
127
128 @RequestMapping(value = "/welcome", method = RequestMethod.POST)
129 public String welcome(@RequestParam(name = "button") String buttonValue, ModelMap model) {
130     logger.info("welcome page entered");
131     if (buttonValue.equals("CreateTask")) {
132         return "/createTask";
133     } else {
134         List<TaskEntity> tasks = service.findAllMethod(getUserIdMethod());
135         model.addAttribute("listTasks", tasks);
136         return "/viewTasks";
137     }
138 }
139
140 @PostMapping("/createTask")
141 public String createTask(@ModelAttribute("createTask") TaskEntity task, ModelMap model) {
142     logger.info("create task page entered");
143     task.setUserId(getUserIdMethod());
144     String save = saveTaskMethod(task);
145     if (save.equals("Saved")) {
146         model.addAttribute("message", "Task created");
147         return "/welcome";
148     } else {
149         model.addAttribute("message", "Error Occured on Task Save. Please Try again!");
150         return "/createTask";
151     }
152 }
153
154 public String saveTaskMethod(TaskEntity user) throws SaveFailedException {
155     try {
156         service.update(user);
157         return "Saved";
158     } catch (Exception e) {
159         throw new SaveFailedException();
160     }
161 }
```

```

TaskManagerController.java X
159         throw new ServletException();
160     }
161 }
162
163 @RequestMapping("/viewTasks")
164 public String viewTasks() {
165     logger.info("Welcome Page Entered");
166     return "welcome";
167 }
168
169 @RequestMapping("/deleteAnotherUser")
170 public String deleteAnotherUser(ModelMap model) {
171     logger.info("Admin Welcome Page Entered");
172     Iterable<UserEntity> users = service.findAllUsers();
173     model.addAttribute("listUsers", users);
174     return "adminWelcome";
175 }
176
177 @GetMapping("/updateTask/{id}")
178 public String updateTask(@PathVariable String id, ModelMap model) {
179     logger.info("update task page entered");
180     taskId = Integer.parseInt(id);
181     Optional<TaskEntity> taskEntity = service.findTaskById(taskId);
182     TaskEntity newTaskEntity = taskEntity.get();
183     if (taskEntity.isPresent()) {
184         model.addAttribute("taskId", newTaskEntity.getTaskId());
185         model.addAttribute("taskName", newTaskEntity.getTaskName());
186         model.addAttribute("startDate", newTaskEntity.getStartDate());
187         model.addAttribute("endDate", newTaskEntity.getEndDate());
188         model.addAttribute("description", newTaskEntity.getDescription());
189         model.addAttribute("email", newTaskEntity.getEmail());
190         model.addAttribute("severity", newTaskEntity.getSeverity());
191         model.addAttribute("taskId", newTaskEntity.getTaskId());
192         return "/updateTask";
193     } else {
194         List<TaskEntity> tasks = service.findAllMethod(getUserIdMethod());
195         model.addAttribute("listTasks", tasks);
196         model.addAttribute("message", "Task Not found! Error!");
197         return "/viewTasks";
198     }
199 }
200
201 @PostMapping("/updateDone")
202 public String updateDone(@ModelAttribute("updateDone") TaskEntity task, ModelMap model) {
203     logger.info("updateDone entered");
204     TaskEntity newTaskEntity;
205     Optional<TaskEntity> taskEntity = service.findTaskById(task.getTaskId());
206     newTaskEntity = taskEntity.get();
207     if (taskEntity.isPresent()) {
208         newTaskEntity.setDescription(task.getDescription());
209         newTaskEntity.setEmail(task.getEmail());
210         newTaskEntity.setEndDate(task.getEndDate());
211         newTaskEntity.setSeverity(task.getSeverity());
212         newTaskEntity.setStartDate(task.getStartDate());

```


TaskManagerController.java

```
212         newTaskEntity.setStartDate(task.getStartDate());
213         newTaskEntity.setTaskId(taskId);
214         newTaskEntity.setUserId(getUserIdMethod());
215         try {
216             service.saveTask(newTaskEntity);
217             List<TaskEntity> tasks = service.findAllMethod(getUserIdMethod());
218             model.addAttribute("listTasks", tasks);
219             model.addAttribute("message", "Task Information Updated");
220             return "/viewTasks";
221         } catch (Exception e) {
222             List<TaskEntity> tasks = service.findAllMethod(getUserIdMethod());
223             model.addAttribute("listTasks", tasks);
224             model.addAttribute("message", "Error occurred on Updation!");
225             return "/viewTasks";
226         }
227     } else {
228         List<TaskEntity> tasks = service.findAllMethod(getUserIdMethod());
229         model.addAttribute("listTasks", tasks);
230         model.addAttribute("message", "Task Not found! Error!");
231         return "/viewTasks";
232     }
233 }
234
235 @GetMapping("/deleteUser/{id}")
236 public String deleteUser(@PathVariable String id, ModelMap model) {
237     logger.info("DeleteUserDone entered");
238     int userId;
239     userId = Integer.parseInt(id);
240     try {
241         service.findUserById(userId);
242         Iterable<UserEntity> users = service.findAllUsers();
243         model.addAttribute("listUsers", users);
244         model.addAttribute("message", "User Deleted");
245         return "deleteAnotherUser";
246     } catch (Exception e) {
247         Iterable<UserEntity> users = service.findAllUsers();
248         model.addAttribute("listUsers", users);
249         model.addAttribute("message", "Error occurred on Deletion of User!");
250         return "deleteAnotherUser";
251     }
252 }
253
254 @GetMapping("/deleteTask/{id}")
255 public String deleteTask(@PathVariable String id, ModelMap model) {
256     taskId = Integer.parseInt(id);
257     Optional<TaskEntity> taskEntity = service.findTaskById(taskId);
258     TaskEntity newTaskEntity = taskEntity.get();
259     if (taskEntity.isPresent()) {
260         model.addAttribute("taskId", newTaskEntity.getTaskId());
261         model.addAttribute("taskName", newTaskEntity.getTaskName());
262         model.addAttribute("startDate", newTaskEntity.getStartDate());
263         model.addAttribute("endDate", newTaskEntity.getEndDate());
264         model.addAttribute("description", newTaskEntity.getDescription());
265         model.addAttribute("email", newTaskEntity.getEmail());
```

TaskManagerController.java

```
262     model.addAttribute("startDate", newTaskEntity.getStartDate());
263     model.addAttribute("endDate", newTaskEntity.getEndDate());
264     model.addAttribute("description", newTaskEntity.getDescription());
265     model.addAttribute("email", newTaskEntity.getEmail());
266     model.addAttribute("severity", newTaskEntity.getSeverity());
267     model.addAttribute("taskId", newTaskEntity.getTaskId());
268     return "/deleteTask";
269 } else {
270     List<TaskEntity> tasks = service.findAllMethod(getUserIdMethod());
271     model.addAttribute("listTasks", tasks);
272     model.addAttribute("message", "Task Not found! Error!");
273     return "/viewTasks";
274 }
275 }
276
277 @PostMapping("/deleteDone")
278 public String deleteDone(@ModelAttribute("deleteDone") TaskEntity task, ModelMap model) {
279     logger.info("updateDone entered");
280     TaskEntity newTaskEntity;
281     Optional<TaskEntity> taskEntity = service.findTaskById(task.getTaskId());
282     newTaskEntity = taskEntity.get();
283     if (taskEntity.isPresent()) {
284         newTaskEntity.setDescription(task.getDescription());
285         newTaskEntity.setEmail(task.getEmail());
286         newTaskEntity.setEndDate(task.getEndDate());
287         newTaskEntity.setSeverity(task.getSeverity());
288         newTaskEntity.setStartDate(task.getStartDate());
289         newTaskEntity.setTaskId(taskId);
290         newTaskEntity.setUserId(getUserIdMethod());
291         try {
292             service.deleteTask(newTaskEntity);
293             List<TaskEntity> tasks = service.findAllMethod(getUserIdMethod());
294             model.addAttribute("message", "Task Deleted");
295             model.addAttribute("listTasks", tasks);
296             return "/viewTasks";
297         } catch (Exception e) {
298             List<TaskEntity> tasks = service.findAllMethod(getUserIdMethod());
299             model.addAttribute("listTasks", tasks);
300             model.addAttribute("message", "Error occurred on Deletion!");
301             return "/viewTasks";
302         }
303     } else {
304         List<TaskEntity> tasks = service.findAllMethod(getUserIdMethod());
305         model.addAttribute("listTasks", tasks);
306         model.addAttribute("message", "Task Not found! Error!");
307         return "/viewTasks";
308     }
309 }
310
311 @RequestMapping("/403")
312 public String accessdenied() {
313     return "403";
314 }
315 }
```

TaskManagerService.java X TaskManagerServiceImpl.java

```
1 package com.hcl.taskManagerProject.service;
2
3 import java.util.List;
4
5
6
7
8
9 public interface TaskManagerService {
10
11     boolean existsByUsernameAndPassword(String userName, String password);
12
13     Optional<UserEntity> findByUserName(String userName);
14
15     UserEntity update(UserEntity user);
16
17     TaskEntity update(TaskEntity user);
18
19     List<TaskEntity> findAllMethod(int userId);
20
21     Optional<TaskEntity> findTaskById(int taskId);
22
23     void saveTask(TaskEntity newTaskEntity);
24
25     void deleteTask(TaskEntity newTaskEntity);
26
27     Optional<TaskEntity> findTaskByTaskName(String taskName);
28
29     Iterable<UserEntity> findAllUsers();
30
31     void findUserById(int userId);
32 }
33
```



```

TaskManagerServiceImpl.java X
1 package com.hcl.taskManagerProject.service;
2
3 import java.util.Arrays;
4
23
24 @Service
25 public class TaskManagerServiceImpl implements TaskManagerService {
26     @Autowired
27     UserRepository userRepository;
28
29     @Autowired
30     RoleRepository roleRepository;
31
32     @Autowired
33     TaskRepository taskRepository;
34
35     BCryptPasswordEncoder bCryptPasswordEncoder;
36
37     @Configuration
38     static class BCryptPasswordEncoderContextConfiguration {
39
40         @Bean
41         public BCryptPasswordEncoder employeeService() {
42             return new BCryptPasswordEncoder();
43         }
44     }
45
46     @Autowired
47     public TaskManagerServiceImpl(UserRepository userRepository, RoleRepository roleRepository,
48         BCryptPasswordEncoder bCryptPasswordEncoder) {
49         this.userRepository = userRepository;
50         this.roleRepository = roleRepository;
51         this.bCryptPasswordEncoder = bCryptPasswordEncoder;
52     }
53
54     @Override
55     public boolean existsByUsernameAndPassword(String userName, String password) {
56         return userRepository.existsByUsernameAndPassword(userName, password);
57     }
58
59     @Override
60     public Optional<UserEntity> findByUserName(String userName) {
61         return userRepository.findByUserName(userName);
62     }
63
64     @Override
65     public UserEntity update(UserEntity user) {
66         user.setPassword(bCryptPasswordEncoder.encode(user.getPassword()));
67         user.setActive(user.getActive());
68         user.getRoles().stream().forEach(f -> {
69             Role userRole = roleRepository.findByRole(f.getRole());
70             user.setRoles(new HashSet<Role>(Arrays.asList(userRole)));
71         });
72     }
73

```

```

TaskManagerServiceImpl.java
73     return userRepository.save(user);
74 }
75
76 @Override
77 public TaskEntity update(TaskEntity task) {
78     return taskRepository.save(task);
79 }
80
81 @Override
82 public List<TaskEntity> findAllMethod(int userId) {
83     List<TaskEntity> list = taskRepository.findAll();
84     List<TaskEntity> newList = null;
85     newList = list.stream().filter(employee -> employee.getUserId() == userId).collect(Collectors.toList());
86     return newList;
87 }
88
89 @Override
90 public Optional<TaskEntity> findTaskById(int id) {
91     Optional<TaskEntity> entity = taskRepository.findById(id);
92     return entity;
93 }
94
95 @Override
96 public void saveTask(TaskEntity taskEntity) {
97     taskRepository.save(taskEntity);
98 }
99
100 @Override
101 public void deleteTask(TaskEntity newTaskEntity) {
102     taskRepository.deleteById(newTaskEntity.getTaskId());
103 }
104
105 @Override
106 public Optional<TaskEntity> findTaskByTaskName(String taskName) {
107     return taskRepository.findByTaskName(taskName);
108 }
109
110 @Override
111 public Iterable<UserEntity> findAllUsers() {
112     Iterable<UserEntity> list = userRepository.findAll();
113     return list;
114 }
115
116 @Override
117 public void findUserById(int userId) {
118     userRepository.deleteById(userId);
119     List<TaskEntity> list = taskRepository.findAll();
120     List<TaskEntity> newList = null;
121     newList = list.stream().filter(employee -> employee.getUserId() == userId).collect(Collectors.toList());
122     newList.stream().forEach(task->{
123         taskRepository.deleteById(task.getTaskId());
124     });
125 }
126

```

```

TaskRepository.java
1 package com.hcl.taskManagerProject.repository;
2
3 import java.util.Optional;
4
5
6
7
8
9
10 @Repository
11 public interface TaskRepository extends JpaRepository<TaskEntity, Integer> {
12     Optional<TaskEntity> findByTaskName(String taskName);
13
14     void deleteAllByUserId(int userId);
15 }
16

```

```

TaskManagerController.java  Role.java  TaskEntity.java  UserEntity.java x
1 package com.hcl.taskManagerProject.entity;
2
3 import javax.persistence.CascadeType;
4
5 @Data
6 @Builder
7 @AllArgsConstructor
8 @NoArgsConstructor
9 @Entity
10 @Table(name = "users")
11 public class UserEntity {
12     @Id
13     @GeneratedValue(strategy = GenerationType.IDENTITY)
14     @Column(name = "userId")
15     int userId;
16     @Column(name = "userName", unique=true)
17     String userName;
18     @Column(name = "password", nullable = false)
19     String password;
20     @Column(name = "firstName", nullable = false)
21     String firstName;
22     @Column(name = "lastName", nullable = false)
23     String lastName;
24     @Column(name = "active")
25     Boolean active;
26     @ManyToMany(cascade = CascadeType.MERGE)
27     @JoinTable(name = "user_role", joinColumns = @JoinColumn(name = "userId"), inverseJoinColumns = @JoinColumn(name = "role_id"))
28     Set<Role> roles;
29 }

```

```
UserRepository.java x MyUserService.java TaskManagerService.java TaskManagerService.java
1 package com.hcl.taskManagerProject.repository;
2
3 import java.util.Optional;
4
10 @Repository
11 public interface UserRepository extends CrudRepository<UserEntity, Integer> {
12
13     boolean existsByUsernameAndPassword(String userName, String password);
14
15     Optional<UserEntity> findByUserName(String userName);
16 }
17
```

```
WebSecurityConfig.java X
1 package com.hcl.taskManagerProject.config;
2
3 import org.springframework.beans.factory.annotation.Autowired;
14
15 @Configuration
16 public class WebSecurityConfig extends WebSecurityConfigurerAdapter{
17     @Autowired
18     MyUserDetailsService userDetailsService;
19
20     @Bean(name="passwordEncoder")
21     public PasswordEncoder passwordEncoder(){
22         return new BCryptPasswordEncoder();
23     }
24
25     @Override
26     protected void configure(AuthenticationManagerBuilder auth) throws Exception {
27         auth
28             .userDetailsService(userDetailsService)
29             .passwordEncoder(passwordEncoder());
30     }
31
32     @Override
33     protected void configure(HttpSecurity http) throws Exception {
34         http
35             .authorizeRequests()
36             .antMatchers("/").permitAll()
37             .antMatchers("/login").permitAll()
38             .antMatchers("/decide").permitAll()
39             .antMatchers("/register").permitAll()
40             .antMatchers("/viewTasks").hasRole("USER")
41             .antMatchers("/updateTask").hasRole("USER")
42             .antMatchers("/createTask").hasRole("USER")
43             .antMatchers("/deleteTask").hasRole("USER")
44             .antMatchers("/deleteUser").hasRole("ADMIN")
45             .antMatchers("/adminWelcome").hasRole("ADMIN")
46             .antMatchers("/deleteAnotherUser").hasRole("ADMIN")
47             .antMatchers("/welcome").permitAll()
48             .and().csrf().disable()
49             .formLogin()
50             .loginPage("/login").failureUrl("/login?error=true")
51             .defaultSuccessUrl("/welcome")
52             .usernameParameter("userName")
53             .passwordParameter("password")
54             .and().logout(logout -> logout
55             .logoutRequestMatcher(new AntPathRequestMatcher("/logout"))
56             .logoutSuccessUrl("/"))
57             .invalidateHttpSession(true)
58             .exceptionHandling()
59             .accessDeniedPage("/403");
60     }
61 }
62 }
63
```

JSP:


```
403.jsp x adminWelcome.... deleteTask.jsp index.jsp login.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Log Out!</title>
8 </head>
9 <body>
10 <table>
11 <tr>
12 <td>You don't have access to this Page!</td>
13 </tr>
14 </table>
15 </body>
16 </html>
```

```
deleteAnoth... deleteTask.jsp index.jsp login.jsp register.jsp updateTask.jsp viewTasks.jsp welcome.jsp x »
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4 <!DOCTYPE html>
5 <html>
6 <head>
7 <meta charset="ISO-8859-1">
8 <title>Home Page</title>
9 </head>
10 <body>
11 <div align="center">
12 ${message }
13 <form id="welcome" action="/welcome" method="post">
14 <table style="width: 50%">
15 <tr>
16 <td>Create Task</td>
17 <td><input type="submit" name="button" value="CreateTask" /></td>
18 <td>View Tasks</td>
19 <td><input type="submit" name="button" value="ViewTasks" /></td>
20 </tr>
21 </table>
22 <table>
23 <tr>
24 <td><a href="<c:url value="/Logout"/>">Logout</a></td>
25 </tr>
26 </table>
27 </form>
28 </div>
29 </body>
30 </html>
```

```
deleteAnoth... x deleteTask.jsp index.jsp login.jsp register.jsp updateTask.jsp viewTasks.jsp x welcome.jsp »_3
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c"%>
4 <!DOCTYPE html>
5 <html>
6 <head>
7   <meta charset="ISO-8859-1">
8   <title>View Tasks</title>
9   <style>
10    #ex_table {
11      table-layout: fixed !important;
12    }
13
14    #ex_table thead tr {
15      width: 500px !important;
16      position: fixed !important;
17    }
18
19    #ex_table tbody tr {
20      width: 500px !important;
21    }
22
23    #ex_table th, td {
24      width: 100px;
25    }
26
27    table {
28      font-family: arial, sans-serif;
29      border-collapse: collapse;
30      text-align: center;
31      vertical-align: text-bottom;
32      width: 100%;
33    }
34
35    td, th {
36      border: 1px solid #dddddd;
37      text-align: left;
38      vertical-align: bottom;
39      padding: 8px;
40    }
41
42    tr:nth-child(even) {
43      background-color: #dddddd;
44    }
45  </style>
46 </head>
47 <body>
48   <div align="center">
49     ${message }
50     <form id="viewTasks" action="/viewTasks" method="post">
51       <table>
52         <tr>
53           <th>Task Name</th>
```

```
deleteAnoth... deleteTask.jsp index.jsp login.jsp register.jsp updateTask.jsp viewTasks.jsp x welcome.jsp »
27 <div>
28     font-family: arial, sans-serif;
29     border-collapse: collapse;
30     text-align: center;
31     vertical-align: text-bottom;
32     width: 100%;
33 }
34
35 td, th {
36     border: 1px solid #dddddd;
37     text-align: left;
38     vertical-align: bottom;
39     padding: 8px;
40 }
41
42 tr:nth-child(even) {
43     background-color: #dddddd;
44 }
45 </style>
46 </head>
47 <body>
48     <div align="center">
49         ${message }
50         <form id="viewTasks" action="/viewTasks" method="post">
51             <table>
52                 <tr>
53                     <th>Task Name</th>
54                     <th>Start Date</th>
55                     <th>End Date</th>
56                     <th>Description</th>
57                     <th>Email</th>
58                     <th>Severity</th>
59                     <th>Click to Update</th>
60                     <th>Click to Delete</th>
61                 </tr>
62                 <c:forEach var="task" items="${listTasks}">
63                     <tr>
64                         <td>${task.taskName}</td>
65                         <td>${task.startDate}</td>
66                         <td>${task.endDate}</td>
67                         <td>${task.description}</td>
68                         <td>${task.email}</td>
69                         <td>${task.severity}</td>
70                         <td><a href="updateTask/${task.taskId}">Edit</a></td>
71                         <td><a href="deleteTask/${task.taskId}">Delete</a></td>
72                     </tr>
73                 </c:forEach>
74             </table>
75             <a href="c:url value="/Logout"/>"Logout</a> <input type="submit"
76                 value="Back to Options Page" />
77         </form>
78     </div>
79 </body>
80 </html>
```



```
deleteAnoth... deleteTask.jsp index.jsp login.jsp register.jsp updateTask.jsp x viewTasks.jsp welcome.jsp »
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="ISO-8859-1">
7   <title>Update the task</title>
8 </head>
9 <body>
10   <div align="center">
11     ${message }
12     <form id="updateDone" action="/updateDone" method="post">
13       <table style="width: 50%">
14         <tr>
15           <td>Task ID</td>
16           <td><input type="text" name="taskId" readonly value="${ taskId}"></td>
17         </tr>
18         <tr>
19           <td>Task Name</td>
20           <td><input type="text" name="taskName" readonly
21             value="${ taskName}"></td>
22         </tr>
23         <tr>
24           <td>Start Date</td>
25           <td><input type="date" name="startDate" value="${ startDate}"></td>
26         </tr>
27         <tr>
28           <td>End Date</td>
29           <td><input type="date" name="endDate" value="${ endDate}"></td>
30         </tr>
31         <tr>
32           <td>Description</td>
33           <td><textarea id="description" name="description" rows="4"
34             cols="50">${ description}</textarea></td>
35         </tr>
36         <tr>
37           <td>Email</td>
38           <td><input type="text" name="email" value="${ email}"></td>
39         </tr>
40         <tr>
41           <td>Severity</td>
42           <td><select name="severity" id="severity" readonly value="${ severity}">
43             <option value="${ severity}">${ severity}</option>
44           </select></td>
45         </tr>
46       </table>
47       <input type="submit" value="Update Task" />
48     </form>
49   </div>
50 </body>
</html>
```

```
deleteAnoth... deleteTask.jsp index.jsp login.jsp register.jsp X updateTask.jsp viewTasks.jsp welcome.jsp »_3
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Register Here!</title>
8 </head>
9 <body>
10 <div align="center">
11   ${message }
12   <form id="register" action="/register" method="post">
13     <table style="width: 50%">
14       <tr>
15         <td>First Name</td>
16         <td><input type="text" name="firstName" required /></td>
17       </tr>
18       <tr>
19         <td>Last Name</td>
20         <td><input type="text" name="lastName" required /></td>
21       </tr>
22       <tr>
23         <td>Username</td>
24         <td><input type="text" name="userName" required /></td>
25       </tr>
26       <tr>
27         <td>Password</td>
28         <td><input type="text" name="password" required /></td>
29       </tr>
30       <tr>
31         <td>Active:</td>
32         <td><input type="radio" id="activeYes" name="active"
33           value="true"> <label for="yes">YES</label> <br></td>
34         <td><input type="radio" id="activeNo" name="active"
35           value="false"> <label for="no">NO</label> <br></td>
36       </tr>
37       <tr>
38         <td>Role</td>
39         <td><input type="radio" id="user" name="roles" value="1"><label
40           for="user">User</label></td>
41         <td><input type="radio" id="admin" name="roles" value="2"><label
42           for="admin">Admin</label></td>
43       </tr>
44     </table>
45     <input type="submit" value="Register" />
46   </form>
47 </div>
48 </body>
49 </html>
```

```
deleteAnoth... X deleteTask.jsp index.jsp login.jsp X register.jsp updateTask.jsp viewTasks.jsp welcome.jsp »_3
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4 <!DOCTYPE html>
5 <html>
6 <head>
7 <meta charset="ISO-8859-1">
8 <title>Login Here!</title>
9 </head>
10 <body>
11 <div align="center">
12   ${message }
13   <form id="Login" action="/login" method="post">
14     <table style="width: 50%">
15       <tr>
16         <td>UserName</td>
17         <td><input type="text" name="userName" required /></td>
18       </tr>
19       <tr>
20         <td>Password</td>
21         <td><input type="text" name="password" required /></td>
22       </tr>
23       <c:if test="${param.error ne null}">
24         <div>Invalid user name and password.</div>
25       </c:if>
26     </table>
27     <input type="submit" value="Login" />
28   </form>
29 </div>
30 </body>
31 </html>
```

```
deleteAnoth... x deleteTask.jsp index.jsp x login.jsp register.jsp updateTask.jsp viewTasks.jsp welcome.jsp »_3
1  <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2    pageEncoding="ISO-8859-1"%>
3  <!DOCTYPE html>
4  <html>
5  <head>
6    <meta charset="ISO-8859-1">
7    <title>Welcome Page</title>
8  </head>
9  <body>
10     <div align="center">
11       <form id="index" action="/decide"
12         method="post">
13         <table style="width: 50%">
14           <tr>
15             <td>Login</td>
16             <td><input type="submit" name = "button" value="Login" /></td>
17             <td>Register</td>
18             <td><input type="submit" name = "button" value="Registration" /></td>
19           </tr>
20         </table>
21       </form>
22     </div>
23 </body>
24 </html>
```

```
deleteAnoth... deleteTask.jsp x index.jsp login.jsp register.jsp updateTask.jsp viewTasks.jsp welcome.jsp »_3
1 1<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6   <meta charset="ISO-8859-1">
7   <title>Delete the task</title>
8 </head>
9 <body>
10   <div align="center">
11     ${message }
12     <form id="deleteDone" action="/deleteDone" method="post">
13       <table style="width: 50%">
14         <tr>
15           <td>Task Id:</td>
16           <td><input type="text" name="taskId" readonly value="${ taskId}"></td>
17         </tr>
18         <tr>
19           <td>Task Name:</td>
20           <td><input type="text" name="taskName" readonly
21             value="${ taskName}"></td>
22         </tr>
23         <tr>
24           <td>Start Date:</td>
25           <td><input type="date" name="startDate" readonly
26             value="${ startDate}"></td>
27         </tr>
28         <tr>
29           <td>End Date:</td>
30           <td><input type="date" name="endDate" readonly
31             value="${ endDate}"></td>
32         </tr>
33         <tr>
34           <td>Description:</td>
35           <td><textarea id="description" name="description" rows="4"
36             cols="50" readonly>${ description}</textarea></td>
37         </tr>
38         <tr>
39           <td>Email:</td>
40           <td><input type="text" name="email" readonly value="${ email}"></td>
41         </tr>
42         <tr>
43           <td>Severity:</td>
44           <td><select name="severity" id="severity" readonly value="${ severity}">
45             <option value="${ severity}">${ severity}</option>
46           </select></td>
47         </tr>
48       </table>
49       <input type="submit" value="Delete Task" />
50     </form>
51   </div>
52 </body>
53 </html>
```

```
deleteAnoth... x deleteTask.jsp index.jsp login.jsp register.jsp updateTask.jsp viewTasks.jsp welcome.jsp »_3
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <!DOCTYPE html>
4 <html>
5 <head>
6 <meta charset="ISO-8859-1">
7 <title>Delete Another User?</title>
8 </head>
9 <body>
10 <form id="deleteAnotherUser" action="/deleteAnotherUser" method="post">
11 <div align="center">
12   ${message } <br></div>
13 <input type="submit" value="Click Here to Delete Another User" />
14 </div>
15 </form>
16 </body>
17 </html>
```

```
adminWelcome.jsp x deleteTask.jsp index.jsp login.jsp register.jsp updateTask.jsp
1 <%@ page language="java" contentType="text/html; charset=ISO-8859-1"
2   pageEncoding="ISO-8859-1"%>
3 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
4 <!DOCTYPE html>
5 <html>
6 <head>
7 <meta charset="ISO-8859-1">
8 <title>Home Page</title>
9 </head>
10 <body>
11 <div align="center">
12   ${message }
13   <form id="welcome" action="/welcome" method="post">
14     <table style="width: 50%">
15       <tr>
16         <td>Create Task</td>
17         <td><input type="submit" name="button" value="CreateTask" /></td>
18         <td>View Tasks</td>
19         <td><input type="submit" name="button" value="ViewTasks" /></td>
20       </tr>
21     </table>
22     <table>
23       <tr>
24         <td><a href="<c:url value="/Logout"/>">Logout</a></td>
25       </tr>
26     </table>
27   </form>
28 </div>
29 </body>
30 </html>
```

Pom:


```
47         <optional>true</optional>
48     </dependency>
49 <dependency>
50     <groupId>javax.servlet</groupId>
51     <artifactId>jstl</artifactId>
52 </dependency>
53 <dependency>
54     <groupId>com.h2database</groupId>
55     <artifactId>h2</artifactId>
56     <scope>runtime</scope>
57 </dependency>
58 <dependency>
59     <groupId>mysql</groupId>
60     <artifactId>mysql-connector-java</artifactId>
61     <scope>runtime</scope>
62 </dependency>
63 <dependency>
64     <groupId>org.projectlombok</groupId>
65     <artifactId>lombok</artifactId>
66     <optional>true</optional>
67 </dependency>
68 <dependency>
69     <groupId>org.springframework.boot</groupId>
70     <artifactId>spring-boot-starter-test</artifactId>
71     <scope>test</scope>
72 </dependency>
73 <dependency>
74     <groupId>org.springframework.security</groupId>
75     <artifactId>spring-security-test</artifactId>
76     <scope>test</scope>
77 </dependency>
78 </dependencies>
79
80 <build>
81     <plugins>
82         <plugin>
83             <groupId>org.springframework.boot</groupId>
84             <artifactId>spring-boot-maven-plugin</artifactId>
85             <configuration>
86                 <excludes>
87                     <exclude>
88                         <groupId>org.projectlombok</groupId>
89                         <artifactId>lombok</artifactId>
90                     </exclude>
91                 </excludes>
92             </configuration>
93         </plugin>
94     </plugins>
95 </build>
96
97 </project>
98
```

```
taskManagerProject/pom.xml X
4  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5  <modelVersion>4.0.0</modelVersion>
6  <parent>
7    <groupId>org.springframework.boot</groupId>
8    <artifactId>spring-boot-starter-parent</artifactId>
9    <version>2.4.2</version>
10   <relativePath /> <!-- lookup parent from repository -->
11 </parent>
12 <groupId>com.hcl</groupId>
13 <artifactId>taskManagerProject</artifactId>
14 <version>0.0.1-SNAPSHOT</version>
15 <name>taskManagerProject</name>
16 <description>Task Manager project with Spring Boot</description>
17 <properties>
18   <java.version>1.8</java.version>
19 </properties>
20 <dependencies>
21   <dependency>
22     <groupId>org.springframework.boot</groupId>
23     <artifactId>spring-boot-starter-data-jpa</artifactId>
24   </dependency>
25   <dependency>
26     <groupId>org.springframework.boot</groupId>
27     <artifactId>spring-boot-starter-security</artifactId>
28   </dependency>
29   <dependency>
30     <groupId>org.springframework.boot</groupId>
31     <artifactId>spring-boot-starter-web</artifactId>
32   </dependency>
33   <dependency>
34     <groupId>org.springframework.boot</groupId>
35     <artifactId>spring-boot-starter-tomcat</artifactId>
36     <scope>provided</scope>
37   </dependency>
38   <dependency>
39     <groupId>org.apache.tomcat.embed</groupId>
40     <artifactId>tomcat-embed-jasper</artifactId>
41     <scope>provided</scope>
42   </dependency>
43   <dependency>
44     <groupId>org.springframework.boot</groupId>
45     <artifactId>spring-boot-devtools</artifactId>
46     <scope>runtime</scope>
47     <optional>true</optional>
48   </dependency>
49   <dependency>
50     <groupId>javax.servlet</groupId>
51     <artifactId>jstl</artifactId>
52   </dependency>
53   <dependency>
54     <groupId>com.h2database</groupId>
55     <artifactId>h2</artifactId>
```

WorkBench:

usersuser_roletasksroles

Limit to 1000 rows

1 • SELECT * FROM sourashtracclub.users;

Result Grid

user_id	active	first_name	last_name	password	user_name
9	1	Karthiga	Ravichandran	\$2a\$10\$3TrJmmqeN.Jl0baCS6vLOM1gEjKBJAhu092tAYAjrXkK.4Rl7...	k
10	1	Karthiga	Ravichandran	\$2a\$10\$UmlQZs4Lhoj6az2CGWsf0emFaFogWj.6GQQmyPn.b5sLZ...	a
15	1	Ram	Kumar	\$2a\$10\$uvq744M.K9Z5If6byvf1.xYqmXXoxRCZFgrOzTtUFOZN2V...	ram
NULL	NULL	NULL	NULL	NULL	NULL

Form EditorField TypesQuery Stats

users 3 ×

Navigator

SCHEMAS

Filter objects

- drinks
- ecommerce
- employees
- karthiga
- learnjspervlet
- myweb
- sakila
- sourashtracclub
 - hibernate_sequence
 - roles
 - tasks
 - user_role
 - users
- Views
- Stored Procedures
- Functions
- spring
- sys
- task

AdministrationSchemas

Information

Table: tasks

Columns:

- task_id int AI PK
- description varchar(3200)
- email varchar(255)
- end_date date
- severity varchar(255)
- start_date date
- task_name varchar(255)

Object InfoSession

usersuser_roletasksroles

Limit to 1000 rows

1 • SELECT * FROM sourashtracclub.users;

Result Grid

user_id	active	first_name	last_name	password	user_name
9	1	Karthiga	Ravichandran	\$2a\$10\$3TrJmmqeN.Jl0baCS6vLOM1gEjKBJAhu092tAYAjrXkK.4Rl7...	k
10	1	Karthiga	Ravichandran	\$2a\$10\$UmlQZs4Lhoj6az2CGWsf0emFaFogWj.6GQQmyPn.b5sLZ...	a
14	1	Ravi	Chari	\$2a\$10\$2Vztys7uRSRgSpXSMIdetHnJEtZBJXE65/g1JfGdEd.TvQ...	r
15	1	Ram	Kumar	\$2a\$10\$uvq744M.K9Z5If6byvf1.xYqmXXoxRCZFgrOzTtUFOZN2V...	ram
NULL	NULL	NULL	NULL	NULL	NULL

Form EditorField TypesQuery Stats

users 3 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
18	22:08:31	SELECT * FROM sourashtracclub.users LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Navigation

SCHEMAS

Filter objects

drinks

ecommerce

employees

karthiga

learnjspervlet

myweb

sakila

sourashtclub

hibernate_sequence

roles

tasks

user_role

users

Views

Stored Procedures

Functions

spring

sys

task

Administration

Schemas

Information

Table: tasks

Columns:

task_idint: A1 PK

descriptionvarchar(3200)

emailvarchar(255)

end_date date

severity varchar(255)

start_date date

task_name varchar(255)

user_id int

Object Info

Session

Navigation

SCHEMAS

Filter objects

drinks

ecommerce

employees

karthiga

learnjspervlet

myweb

sakila

sourashtclub

hibernate_sequence

roles

tasks

user_role

users

Views

Stored Procedures

Functions

spring

sys

task

Administration

Schemas

Information

Table: tasks

Columns:

task_idint: A1 PK

descriptionvarchar(3200)

emailvarchar(255)

end_date date

severity varchar(255)

start_date date

task_name varchar(255)

user_id int

Object Info

Session

users

user_role

tasks

roles

Limit to 1000 rows

1

SELECT * FROM sourashtclub.user_role;

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content: 15

	user_id	role_id
10	1	
14	1	
15	1	
9	2	
NULL	NULL	

user_role 3 x

Apply

Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
18	22:08:31	SELECT * FROM sourashtclub.users LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

users

user_role

tasks

roles

Limit to 1000 rows

1

SELECT * FROM sourashtclub.tasks;

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content: 15

	task_id	description	email	end_date	severity	start_date	task_name	user_id
11	rdth		karthiga.ravic@hd.com	2021-02-04	Medium	2021-02-02	1	10
12	updated task des...		grikarthikagr@gmail.com	2021-02-18	High	2021-02-04	2	10
17	Task 4 created		karthiga.ravic@hd.com	2021-02-04	Low	2021-02-02	Task4	10
18	zCx		asda	2021-02-11	Medium	2021-02-02	1	15
19	zsdzdc		azdx	2021-02-11	Medium	2021-02-01	2	15
NULL	NULL		NULL	NULL	NULL	NULL	NULL	NULL

tasks 1 x

Apply

Revert

Output

Action Output

#	Time	Action	Message	Duration / Fetch
18	22:08:31	SELECT * FROM sourashtclub.users LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

Navigation: users, user_role, tasks, roles

Limit to 1000 rows

1 • `SELECT * FROM sourashtraclub.tasks;`

Result Grid

task_id	description	email	end_date	severity	start_date	task_name	user_id
11	rdth	karthiga.ravic@ihd.com	2021-02-04	Medium	2021-02-02	1	10
12	updated task des...	grkarthikagr@gmail.com	2021-02-18	High	2021-02-04	2	10
17	Task 4 created	karthiga.ravic@ihd.com	2021-02-04	Low	2021-02-02	Task4	10
18	zCx	asda	2021-02-11	Medium	2021-02-02	1	15
19	zdczdc	azdx	2021-02-11	Medium	2021-02-01	2	15
20	xcf	xfcb	2021-02-04	Low	2021-02-01	svf	14
21	xfgv	dxfg	2021-02-11	Medium	2021-02-08	drq	14

Table: tasks

Columns:

- task_id: int AI PK
- description: varchar(3200)
- email: varchar(255)
- end_date: date
- severity: varchar(255)
- start_date: date
- task_name: varchar(255)
- user_id: int

Object Info Session

Navigation: users, user_role, tasks, roles

Limit to 1000 rows

1 • `SELECT * FROM sourashtraclub.roles;`

Result Grid

role_id	role
1	ROLE_USER
2	ROLE_ADMIN

Table: roles

Columns:

- role_id: int PK
- role: varchar(255)

Information

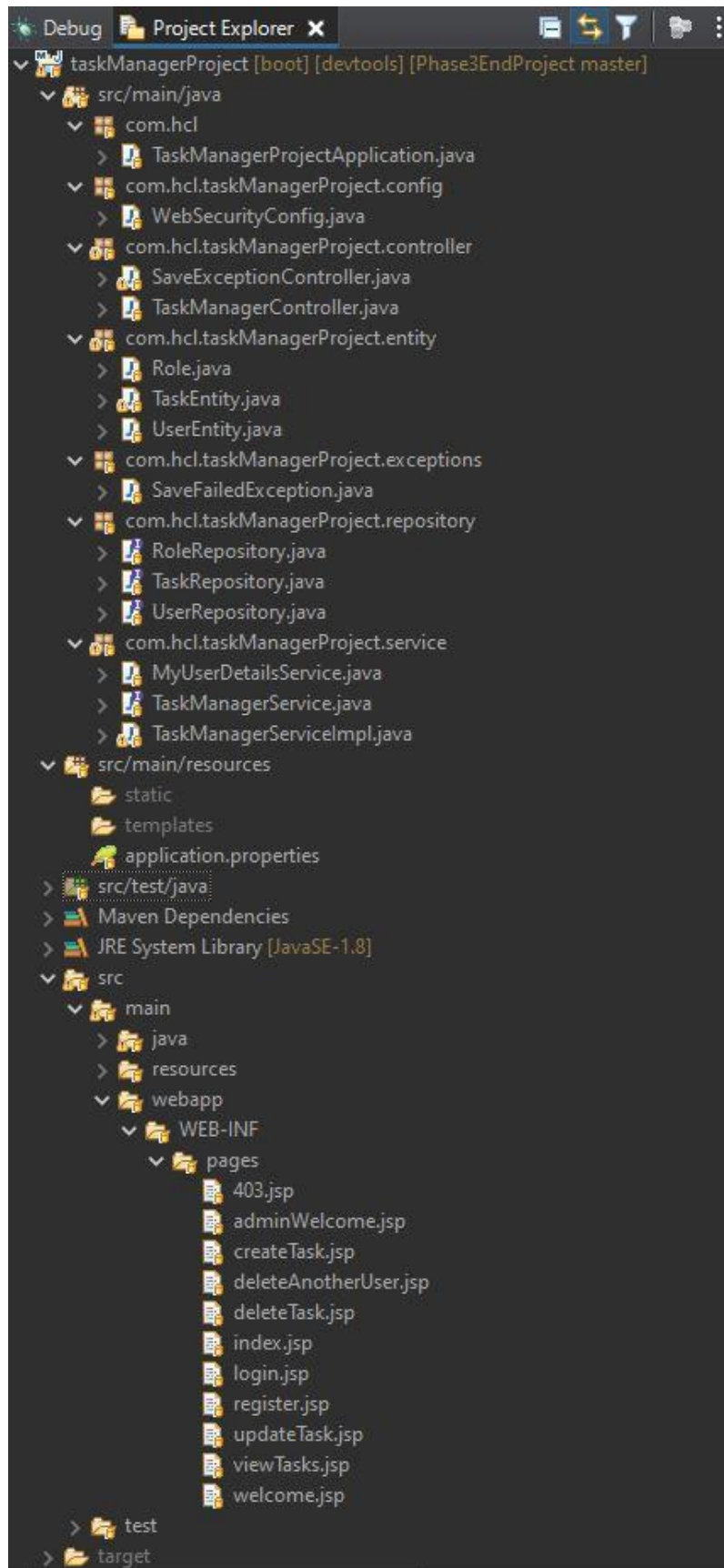
Application Properties:

```

application.properties
1 spring.mvc.view.prefix = /WEB-INF/pages/
2 spring.mvc.view.suffix = .jsp
3 spring.mvc.static-path-pattern=/resources/**
4 spring.jpa.hibernate.ddl-auto=update
5 spring.datasource.url= jdbc:mysql://localhost:3306/sourashtraclub?createDatabaseIfNotExist=true
6 spring.datasource.username=root
7 spring.datasource.password=root
8 spring.jpa.properties.hibernate.format_sql=true
9 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
10 logging.level.org.springframework.web: DEBUG
11 server.port=8082
12 spring.datasource.initialization-mode=always

```

Folder Structure:



Outputs:

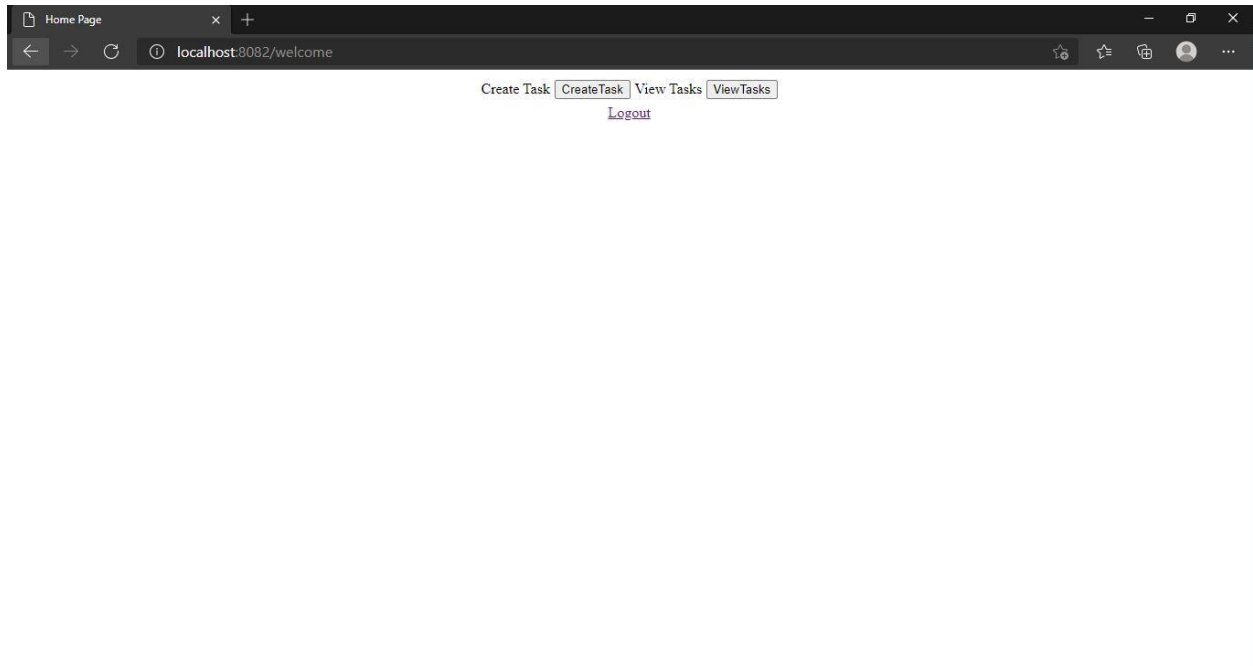
WelcomePage



LoginPageWithCorrectCredentials



HomePageForUser



CreateTaskPage

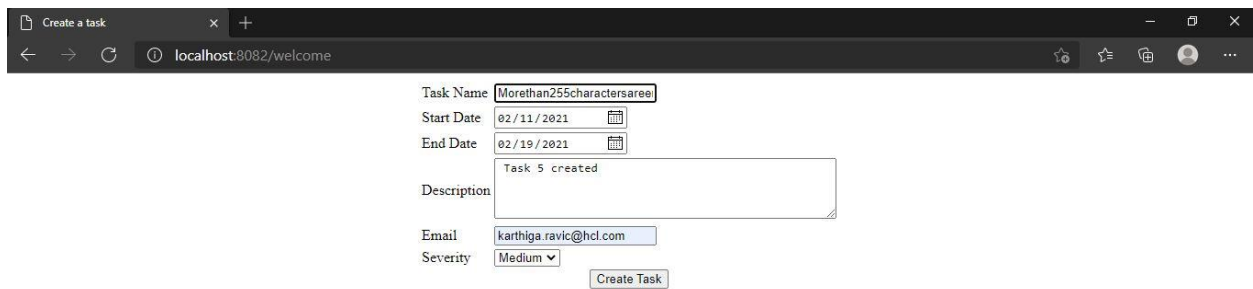
A screenshot of the 'Create a task' page. The form contains the following fields and controls:

- Task Name:
- Start Date: with a calendar icon
- End Date: with a calendar icon
- Description:
- Email:
- Severity: with a dropdown arrow
- Create Task:

HomePageWithMessage



CreateTaskPageWithIncorrectValues



ErrorOccuredOnSaveException



ViewTasksPage

A screenshot of a web browser window showing the 'View Tasks' page. The address bar displays 'localhost:8082/welcome'. The page contains a table with 8 columns: Task Name, Start Date, End Date, Description, Email, Severity, Click to Update, and Click to Delete. There are four rows of task data. Below the table, there are two links: 'Logout' and 'Back to Options Page'.

Task Name	Start Date	End Date	Description	Email	Severity	Click to Update	Click to Delete
1	2021-02-02	2021-02-04	rdth	karthiga.ravic@hcl.com	Medium	Edit	Delete
2	2021-02-03	2021-02-04	er	grkarthikagr@gmail.com	High	Edit	Delete
sdf	2021-02-02	2021-02-19	sxg	karthiga.ravic@hcl.com	Medium	Edit	Delete
Task4	2021-02-02	2021-02-04	Task 4 created	karthiga.ravic@hcl.com	Low	Edit	Delete

[Logout](#) [Back to Options Page](#)

UpdateTaskPage

Update the task

localhost:8082/updateTask/12

Task ID

12

Task Name

2

Start Date

02/04/2021

End Date

02/18/2021

Description

updated task description, dates

Email

grkarthikagr@gmail.com

Severity

High

Update Task

ViewTasksPage2

View Tasks

localhost:8082/updateDone

Task Information Updated

Task Name	Start Date	End Date	Description	Email	Severity	Click to Update	Click to Delete
1	2021-02-02	2021-02-04	rdth	karthiga.ravic@hcl.com	Medium	Edit	Delete
2	2021-02-04	2021-02-18	updated task description, dates	grkarthikagr@gmail.com	High	Edit	Delete
sdf	2021-02-02	2021-02-19	sxg	karthiga.ravic@hcl.com	Medium	Edit	Delete
Task4	2021-02-02	2021-02-04	Task 4 created	karthiga.ravic@hcl.com	Low	Edit	Delete

[Logout](#) [Back to Options Page](#)

DeleteTaskPage

Delete the task

localhost:8082/deleteTask/15

Task Id:

15

Task Name

sdf

Start Date

02/02/2021

End Date

02/19/2021

Description

sxg

Email

karthiga.ravic@hcl.com

Severity

Medium

Delete Task

ViewTasksPageWithDeletedSuccessMessage

View Tasks

localhost:8082/deleteDone

Task Deleted

Task Name	Start Date	End Date	Description	Email	Severity	Click to Update	Click to Delete
1	2021-02-02	2021-02-04	rdth	karthiga.ravic@hcl.com	Medium	Edit	Delete
2	2021-02-04	2021-02-18	updated task description, dates	grkarthikagr@gmail.com	High	Edit	Delete
Task4	2021-02-02	2021-02-04	Task 4 created	karthiga.ravic@hcl.com	Low	Edit	Delete

[Logout](#)
[Back to Options Page](#)

UpdateTaskWithIncorrectValues

Update the task

localhost:8082/updateTask/17

Task ID

17

Task Name

Task4

Start Date

02/02/2021

End Date

02/04/2021

Description

Task 4 created

Email

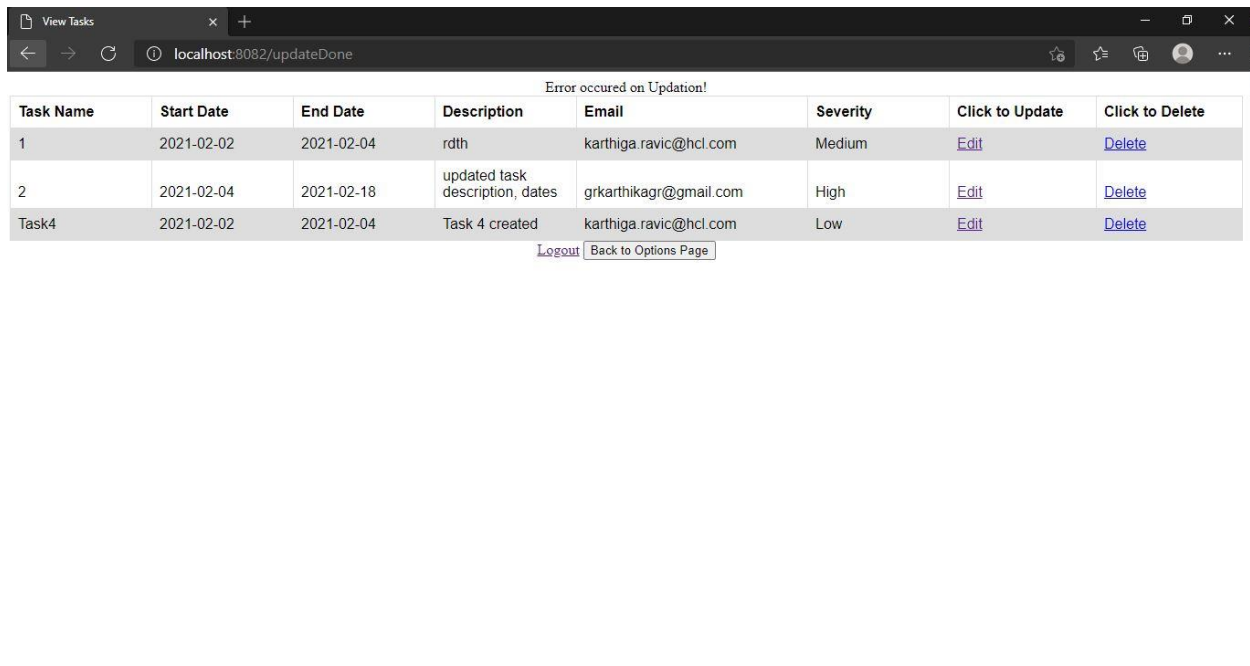
karthiga.ravic@hcl.com4444

Severity

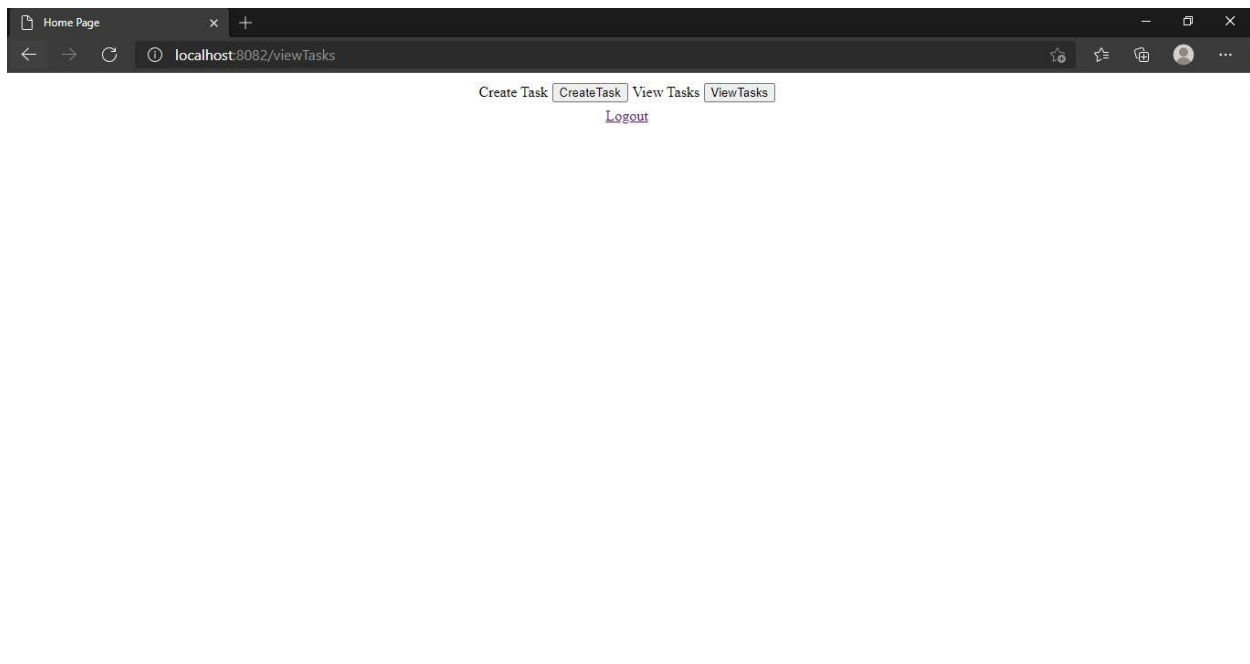
Low

Update Task

ViewTasksPageWithDeleteFailPage



BackToOptionsPage



LogoutToWelcomePage



RegisterPage

A screenshot of a web browser window showing the 'Register Here!' page. The address bar shows 'localhost:8082/decide'. The form contains the following fields and options:

- First Name: Ravi
- Last Name: Chan
- Username: r
- Password: r
- Active: ☒ YES ☐ NO
- Role: ☒ User ☐ Admin

A 'Register' button is located below the form.

RegistrationSuccessPage

Browser window showing a successful registration page. The address bar displays `localhost:8082/register`. The page title is "Login Here!". The form displays "User Information Saved" with the following details:

- UserName:
- Password:
- Login button

SameUsernamePage

Browser window showing a registration page. The address bar displays `localhost:8082/decide`. The page title is "Register Here!". The form displays the following fields and options:

- First Name:
- Last Name:
- Username:
- Password:
- Active: ☒ YES ☐ NO
- Role: ☐ User ☒ Admin
- Register button

UsernameAlreadyExists

Register Here!

localhost:8082/register

Username Already Exists/Try a different one!

First Name

Last Name

Username

Password

Active: ☒ YES ☐ NO

Role ☒ User ☐ Admin

LoginPageWithAdminCredentials

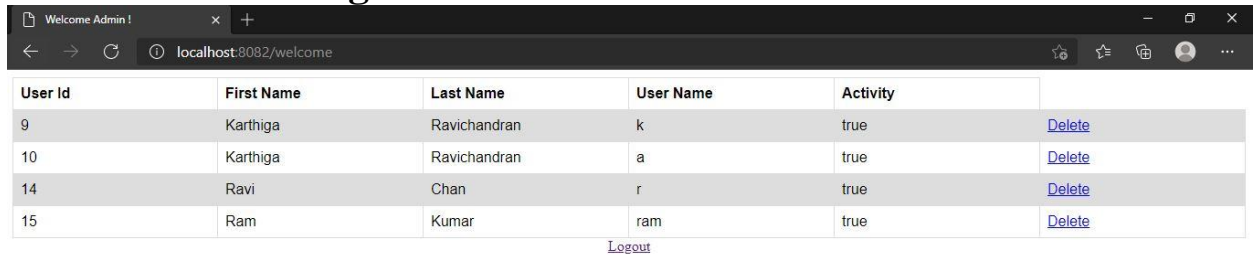
Login Here!

localhost:8082/decide

UserName:

Password:

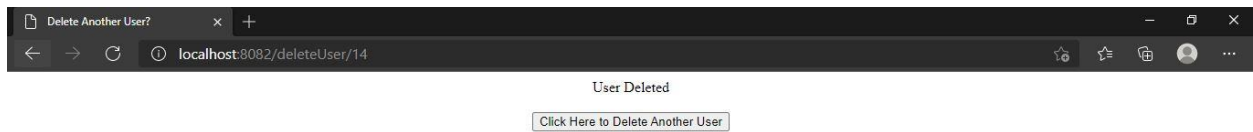
WelcomeAdminPage



User Id	First Name	Last Name	User Name	Activity	
9	Karthiga	Ravichandran	k	true	Delete
10	Karthiga	Ravichandran	a	true	Delete
14	Ravi	Chan	r	true	Delete
15	Ram	Kumar	ram	true	Delete

[Logout](#)

DeletedRavi1chanUserWithTasks



WelcomeAdminPageAfterDeletingOneUser

Welcome Admin !localhost:8082/deleteAnotherUser

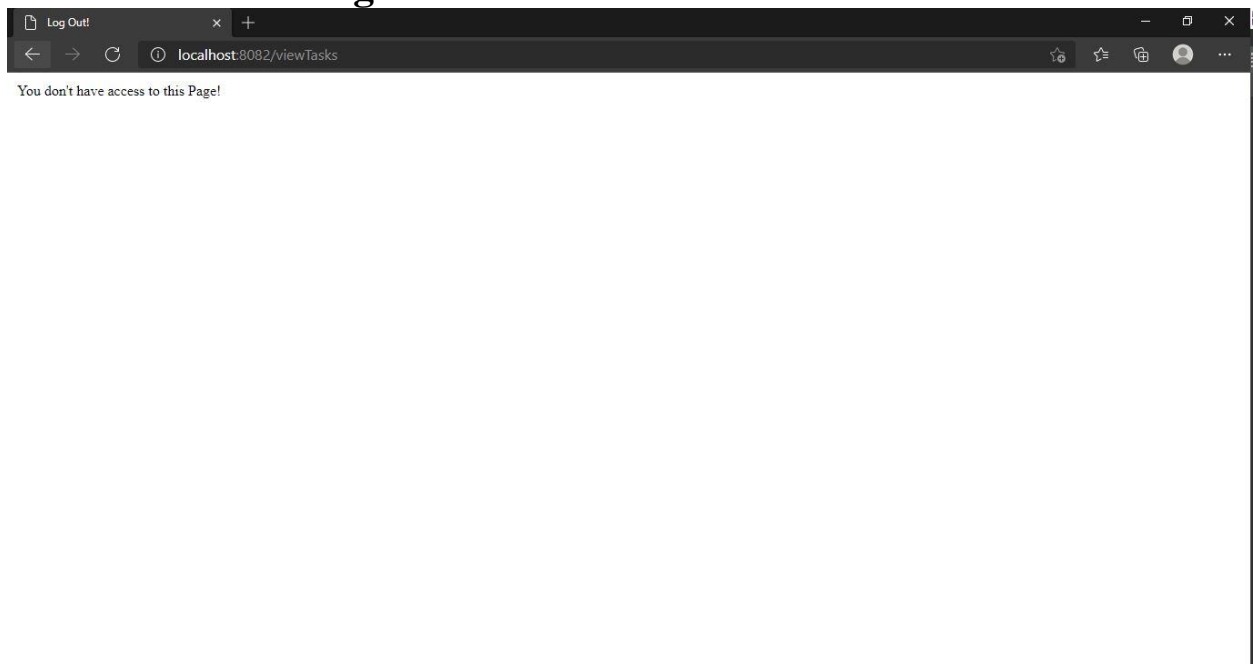
User Id	First Name	Last Name	User Name	Activity	
9	Karthiga	Ravichandran	k	true	Delete
10	Karthiga	Ravichandran	a	true	Delete
15	Ram	Kumar	ram	true	Delete

Logout

LogoutPageForAdmin



AdminNoAccessPage



UserNoAccessPage



Java Concepts Involved in the project:

1. Classes
2. Exception Handling
3. Packages
4. Spring Boot Framework
5. Logger – slf4j Logger and LoggerFactory
6. Web Security
7. Password Encryption
8. Service-Model-Controller-Repository-JSP

Conclusion:

Task Manager Application does not crash for incorrect inputs or any discrepancies caused by the user or admin. It has better performance as it uses Service, repository, Controller, jsp and logger to log data. It encrypts the password and uses Web security to provide access to pages for users.