

CRUD Web Application with Spring Boot, MySQL and Thymeleaf

- MVC (Model-View-Controller) is a popular architecture pattern used in software development to separate an application's concerns into three interconnected components:
 - Model
 - View
 - Controller
- The Model represents the data and logic of the application
- The View represents the user interface
- The Controller handles user input and manages the interaction between the Model and View.
- This separation of concerns allows for better code organization, maintainability, and scalability.
- Spring Boot is a popular Java framework that provides a simplified way of building and deploying web applications.
- Features
 - auto-configuration
 - embedded web server (**Tomcat** server)
 - robust set of libraries to build web applications faster
- Spring Boot follows the MVC architecture pattern
 - Model is represented by domain objects or entities
 - View is represented by templates or user interfaces
 - Controller is represented by the Spring Framework Controller
- Spring Boot provides several features to make MVC development easier
 - annotations-based configuration
 - data binding
 - error handling.
- Tools Used
 - **Java** - language
 - **Spring Boot** – Java Web Framework (follows MVC architecture)
 - **Thymeleaf** – to render HTML pages
 - **MySQL** – Database
 - **IntelliJ** - Ide

View Users

Airbnb Users

localhost:8080

Airbnb Users

User ID	Name	Email	Phone	Action
1	Abhishek Patil	abhijpatil2003@gmail.com	09632714472	<button>Delete</button>
3	Amogh	amogh@gmail.com	9898989898	<button>Delete</button>
52	Aniruddha	aniruddha@gmail.com	9696969696	<button>Delete</button>
53	Pranav Raju	pranav@gmail.com	9215263987	<button>Delete</button>
54	Manav	manav@gmail.com	9345697832	<button>Delete</button>

Add User

Add User

Add User Record

localhost:8080/add-user

Add User Record

Name:

Enter name

Email:

Enter email

Phone:

Enter phone

Submit

View Users

MvcApplication.java

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

User.java

```
1 package dev.abhishek.mvc.models;
2
3 import jakarta.persistence.Entity;
4 import jakarta.persistence.GeneratedValue;
5 import jakarta.persistence.GenerationType;
6 import jakarta.persistence.Id;
7 import Lombok.ALLArgsConstructor;
8 import Lombok.Data;
9 import Lombok.NoArgsConstructor;
10
11 @Entity
12 @Data
13 @ALLArgsConstructor
14 @NoArgsConstructor
15 public class User{
16     @Id
17     @GeneratedValue(strategy= GenerationType.AUTO)
18     private Integer userid;
19     private String name;
20     private String email;
21     private String phone;
22 }
23
```

application.properties

```
1 spring.jpa.hibernate.ddl-auto=update
2 spring.datasource.url=jdbc:mysql://${MYSQL_HOST:localhost}:3306/mvc
3 spring.datasource.username=root
4 spring.datasource.password=abhi-patil
5 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
```

UserRepository.java

```
1 package dev.abhishek.mvc.repositories;
2
3 import org.springframework.data.repository.CrudRepository;
4 import dev.abhishek.mvc.models.User;
5 public interface UserRepository extends CrudRepository<User, Integer> {
6
7 }
8
```

MainController.java

```
1 package dev.abhishek.mvc.controllers;
2
3 import dev.abhishek.mvc.models.User;
4 import dev.abhishek.mvc.repositories.UserRepository;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Controller;
7 import org.springframework.ui.Model;
8 import org.springframework.web.bind.annotation.*;
9 import java.util.List;
10
11 @Controller
12 public class MainController {
13     @Autowired
14     private UserRepository userRepository;
15
16     @GetMapping(path="/")
17     public String viewUser(Model model){
18         List<User> users = (List<User>) userRepository.findAll();
19         model.addAttribute("users", users);
20         return "view-user";
21     }
22
23     @GetMapping(path="/add-user")
24     public String returnAddUserPage(Model model) {
25         User user = new User();
26         model.addAttribute("user", user);
27         return "add-user";
28     }
29
30     @PostMapping(path="/add-user")
31     public String addNewUser(@ModelAttribute("user") User user) {
32         userRepository.save(user);
33         return "redirect:/add-user";
34     }
35
36     @PostMapping("/delete-user")
37     public String deleteUser(@RequestParam("id") Integer id, Model model) {
38         userRepository.deleteById(id);
39         model.addAttribute("users", userRepository.findAll());
40         return "redirect:/";
41     }
42 }
```

add-user.html

```
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.w3.org/1999/xhtml" Lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Add User Record</title>
6      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
7      <style>
8          body {
9              background-color: #f8f9fa;
10          }
11          .container {
12              max-width: 500px;
13              margin: 0 auto;
14              margin-top: 30px;
15              background-color: #ffffff;
16              padding: 20px;
17              border-radius: 10px;
18              box-shadow: 0px 0px 10px rgba(0,0,0,0.1);
19          }
20          h1 {
21              text-align: center;
22              font-weight: bold;
23              color: #007bff;
24          }
25          label {
26              font-weight: bold;
27          }
28          button[type="submit"] {
29              margin-top: 20px;
30              background-color: #007bff;
31              border-color: #007bff;
32          }
33          button[type="submit"]:hover {
34              background-color: #0069d9;
35              border-color: #0062cc;
36          }
37          a {
38              color: #28a745 !important;
39              margin-right: 20px;
40              border: 1px solid #28a745;
41              padding: 5px 10px;
42              border-radius: 5px;
43          }
44          a:hover {
45              color: #ffffff !important;
46              background-color: #28a745;
47              border-color: #28a745;
48              text-decoration: none;
49          }
50      </style>
51  </head>
52  <body>
53      <div class="container">
54          <h1>Add User Record</h1>
55          <form action="#" th:action="@{/add-user}" method="POST" th:object="${user}">
56              <div class="form-group">
57                  <label>Name:</label>
58                  <input type="text" th:field = "${name}" class="form-control" id="name" placeholder="Enter name">
59              </div>
60              <div class="form-group">
61                  <label>Email:</label>
62                  <input type="email" th:field = "${email}" class="form-control" id="email" placeholder="Enter email">
63              </div>
64              <div class="form-group">
65                  <label>Phone:</label>
66                  <input type="text" th:field = "${phone}" class="form-control" id="phone" placeholder="Enter phone">
67              </div>
68              <button type="submit" class="btn btn-primary btn-block">Submit</button>
69          </form>
70          <div class="text-center mt-3">
71              <a href="/">View Users</a>
72          </div>
73      </div>
74  </body>
75  </html>
```

view-user.html (part-1)



```
1  <!DOCTYPE html>
2  <html lang="en" xmlns:th="http://www.w3.org/1999/xhtml">
3  <head>
4    <meta charset="UTF-8">
5    <title>Airbnb Users</title>
6    <style>
7      body {
8        font-family: Arial, sans-serif;
9        background-color: #f8f9fa;
10     }
11
12     h1 {
13       text-align: center;
14       font-size: 2.5rem;
15       margin-top: 50px;
16       margin-bottom: 30px;
17     }
18
19     table {
20       margin: 0 auto;
21       border-collapse: collapse;
22       background-color: #fff;
23       box-shadow: 0 0 20px rgba(0,0,0,0.2);
24     }
25
26     th, td {
27       padding: 10px 20px;
28       text-align: center;
29     }
30
31     th {
32       background-color: #007bff;
33       color: #fff;
34     }
35
36     tbody tr:nth-child(2n) {
37       background-color: #f2f2f2;
38     }
39
40     .button {
41       display: inline-block;
42       background-color: #007bff;
43       color: #fff;
44       padding: 10px 20px;
45       margin-top: 20px;
46       border-radius: 5px;
47       text-decoration: none;
48       transition: background-color 0.3s ease;
49     }
```

view-user.html (part-2)

```
1      .button:hover {
2          background-color: #0056b3;
3      }
4
5      .delete-button {
6          display: inline-block;
7          background-color: #dc3545;
8          color: #fff;
9          padding: 5px 10px;
10         margin-left: 10px;
11         border-radius: 5px;
12         text-decoration: none;
13         transition: background-color 0.3s ease;
14     }
15
16     .delete-button:hover {
17         background-color: #c82333;
18     }
19 </style>
20 </head>
21 <body>
22 <h1>Airbnb Users</h1>
23 <table border="1" cellpadding="0" cellspacing="0">
24     <thead>
25     <tr>
26         <th>User ID</th>
27         <th>Name</th>
28         <th>Email</th>
29         <th>Phone</th>
30         <th>Action</th>
31     </tr>
32     </thead>
33     <tbody>
34     <tr th:each="user : ${users}">
35         <td th:text="${user.userid}"></td>
36         <td th:text="${user.name}"></td>
37         <td th:text="${user.email}"></td>
38         <td th:text="${user.phone}"></td>
39         <td>
40             <form method="post" th:action="@{/delete-user}">
41                 <input type="hidden" name="id" th:value="${user.userid}" />
42                 <button type="submit" class="button">Delete</button>
43             </form>
44         </td>
45     </tr>
46     </tbody>
47 </table>
48 <div style="text-align: center;">
49     <a href="/add-user" class="button">Add User</a>
50 </div>
51 </body>
52 </html>
```