

Validation in Spring Boot

Validation is like a quality check for data. Just like a teacher marks your answers right or wrong, validation checks if the information people give to your program is correct and follows the rules.

Spring Boot provides various mechanisms for validation, including annotations, custom validators, error handling and group validation.

Validation Annotations

@NotNull: Ensures a field is not null.

@NotBlank: Enforces non-nullity and requires at least one non-whitespace character.

@Min(value): Checks if a numeric field is greater than or equal to the specified minimum value.

@Size(min, max): Validates if a string or collection size is within a specific range.

@Email: Ensures a field contains a valid email address format.

@Digits(integer, fraction): Validates that a numeric field has a specified number of integer and fraction digits.

@Past and **@Future** : Checks that a date or time field is in the past and future respectively.

PersonForm.java

```
package com.example.demo;
```

```
import java.util.Date;
```

```
import jakarta.validation.constraints.*;
```

```
public class PersonForm {
```

```
    @NotNull
```

```
    @Size(min=3,max=30)
```

```
    private String name;
```

```
    @NotNull
```

```
    @Min(18)
```

```
    private int age;
```

```
    @NotNull
```

```
    @Size(min=10,max=10)
```

```
    @Digits(message="Enter valid",integer=10,fraction=0)
```

```
    private String phone;
```

```
@NotBlank
@email(message="Enter Valid address")
private String email;

@NotNull
@Past
private Date date;

public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public int getAge() {
    return age;
}

public void setAge(int age) {
    this.age = age;
}

public String toString()
{
    return name+" "+age+" "+phone;
}

public String getPhone() {
```

```
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public Date getDate() {
        return date;
    }
    public void setDate(Date date) {
        this.date = date;
    }
}
```

ValidationApplication.java

```
package com.example.demo;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication

public class ValidationApplication {

    public static void main(String[] args) {
        SpringApplication.run(ValidationApplication.class, args);
    }
}
```

WebController.java

```
package com.example.demo;

import org.springframework.stereotype.Controller;
import org.springframework.validation.BindingResult;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;
import
org.springframework.web.servlet.config.annotation.WebMvcConfigurer;

import jakarta.validation.Valid;

import org.springframework.web.servlet.config.annotation.*;
import org.springframework.ui.Model;

@Controller
public class WebController implements WebMvcConfigurer {

    @GetMapping("/")
    public String showForm(PersonForm personForm, Model model)
    {
        model.addAttribute("personForm", personForm);
        return "form";
    }

    @PostMapping("/")
    public String checkPersonInfo(@Valid PersonForm
personForm, BindingResult bindingResult, Model model)
    {
        model.addAttribute("personForm", personForm);
        if(bindingResult.hasErrors())
        {
            return "form";
        }
        return "results";
    }
}
```

Form.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
    <form action="#" th:action="@{/}" th:object="${personForm}"
method="post">
        <table>
            <tr>
                <td>Name:</td>
                <td><input type="text" th:field="*{name}" /></td>
                <td
                    th:if="${#fields.hasErrors('name')}"
th:errors="*{name}">Name Error</td>
            </tr>

            <tr>
                <td>Age:</td>
                <td><input type="text" th:field="*{age}" /></td>
                <td
                    th:if="${#fields.hasErrors('age')}"
th:errors="*{age}">Age Error</td>
            </tr>

            <tr>
                <td>Phone no:</td>
                <td><input type="text" th:field="*{phone}" /></td>
                <td
                    th:if="${#fields.hasErrors('phone')}"
th:errors="*{phone}">Phone Error</td>
            </tr>

            <tr>
                <td>Email:</td>
                <td><input type="text" th:field="*{email}" /></td>
                <td
                    th:if="${#fields.hasErrors('email')}"
th:errors="*{email}">Email Error</td>
            </tr>
        </table>
    </form>
</body>
</html>
```

```
<tr>

    <tr>
        <td>Date:</td>
        <td><input type="text" th:field="*{date}" /></td>
        <td
            th:if="${#fields.hasErrors('date')}"
th:errors="*{date}">Date Error</td>
    </tr>

    <tr>
        <td><button type="submit">Submit</button></td>
    </tr>

</table>
</form>
</body>
</html>
```


Result.html

```
<!DOCTYPE html>

<html>

<head>

<meta charset="ISO-8859-1">

<title>Insert title here</title>

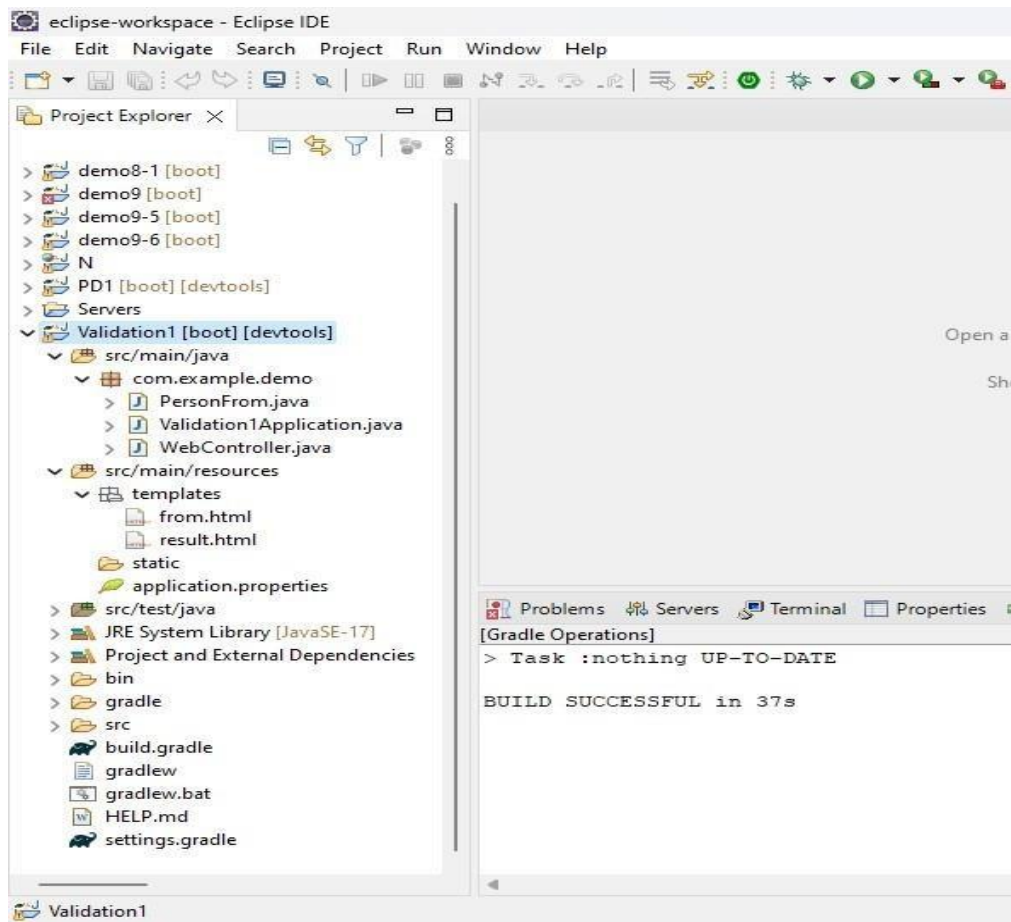
</head>

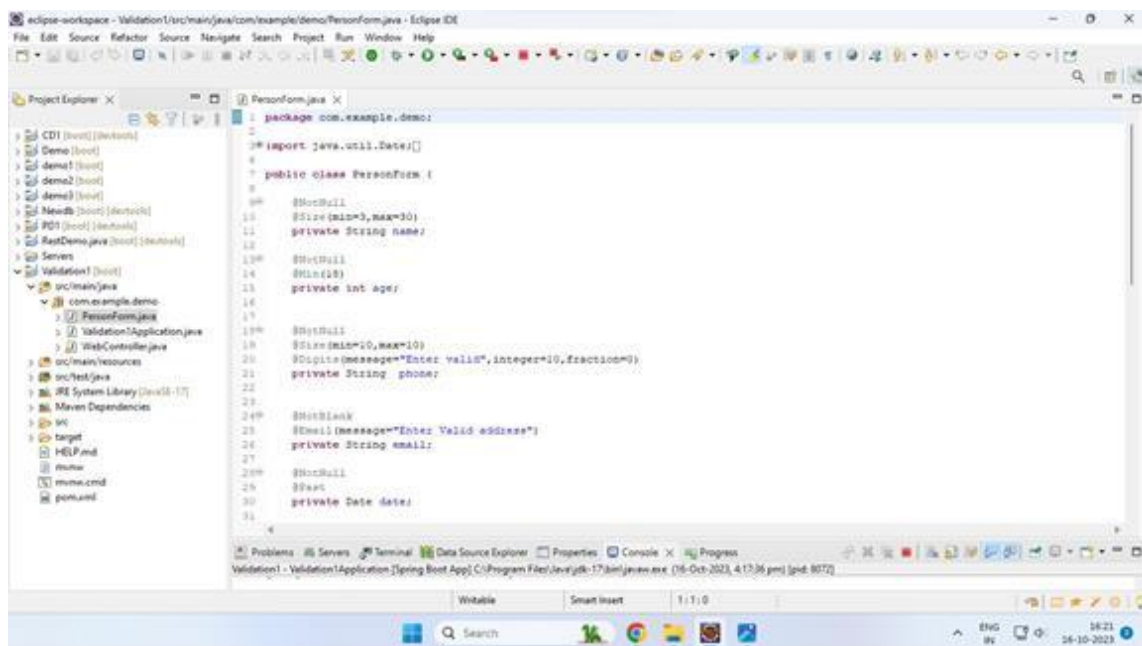
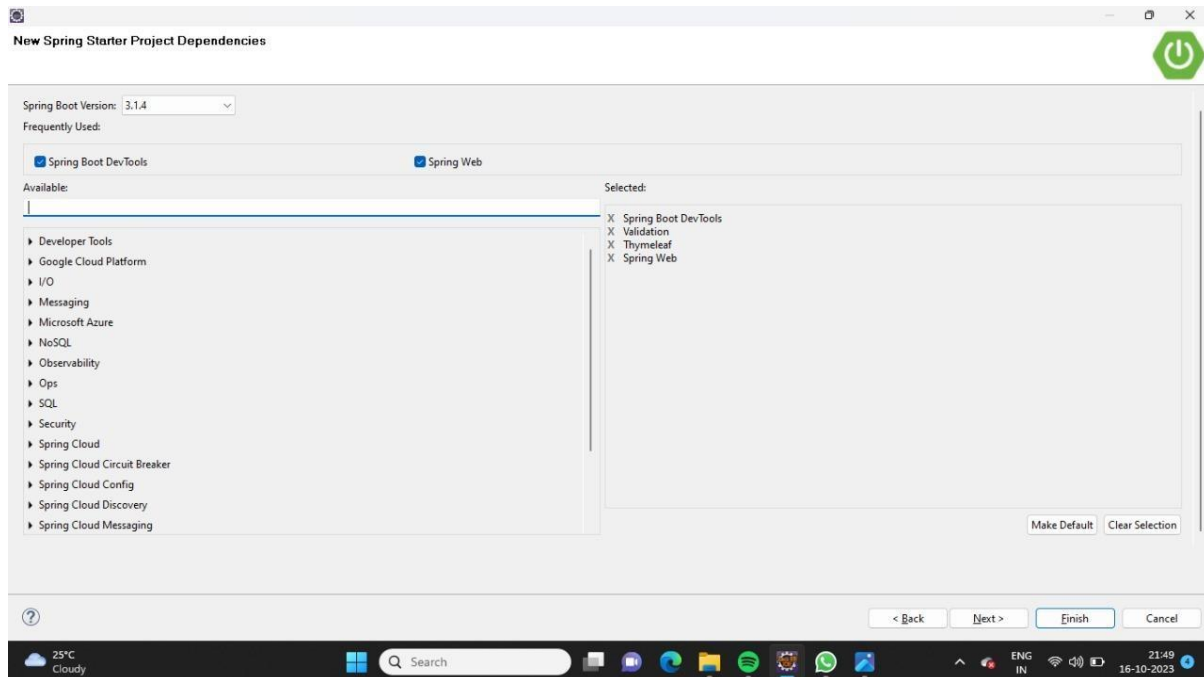
<body> Welcome

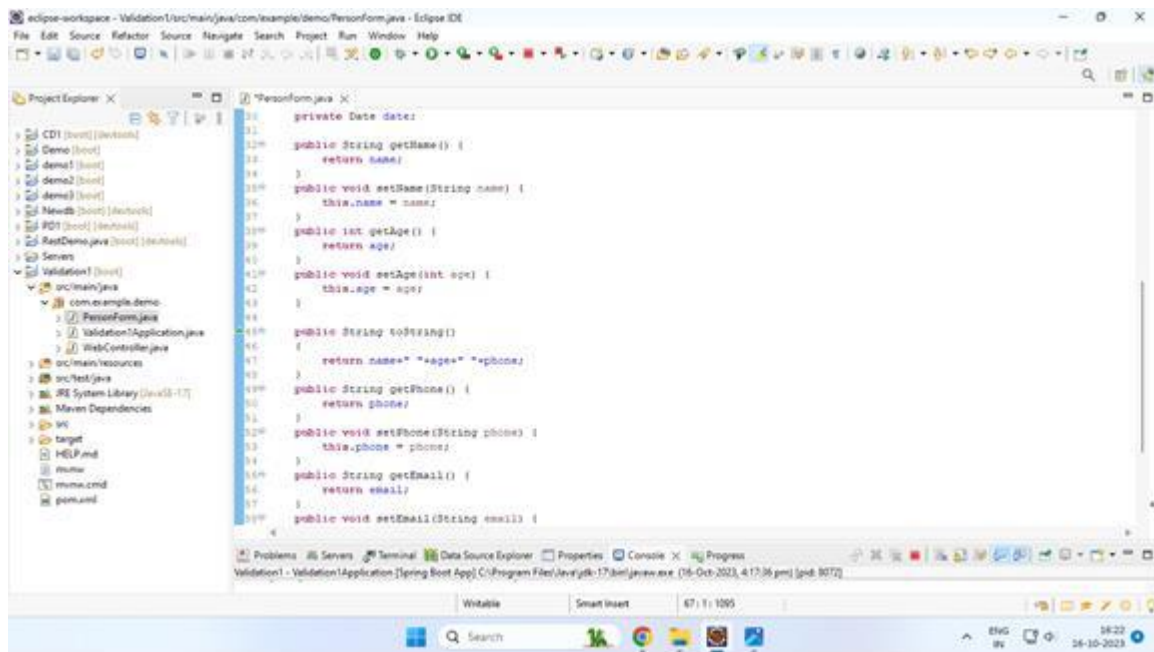
    <p th:text="'Name:' + ${personForm.name}" />
    <p th:text="'age:' + ${personForm.age}" />
    <p th:text="'phone no:' + ${personForm.phone}" />
    <p th:text="'email:' + ${personForm.email}" />
    <p th:text="'dob:' + ${personForm.date}" />

</body>

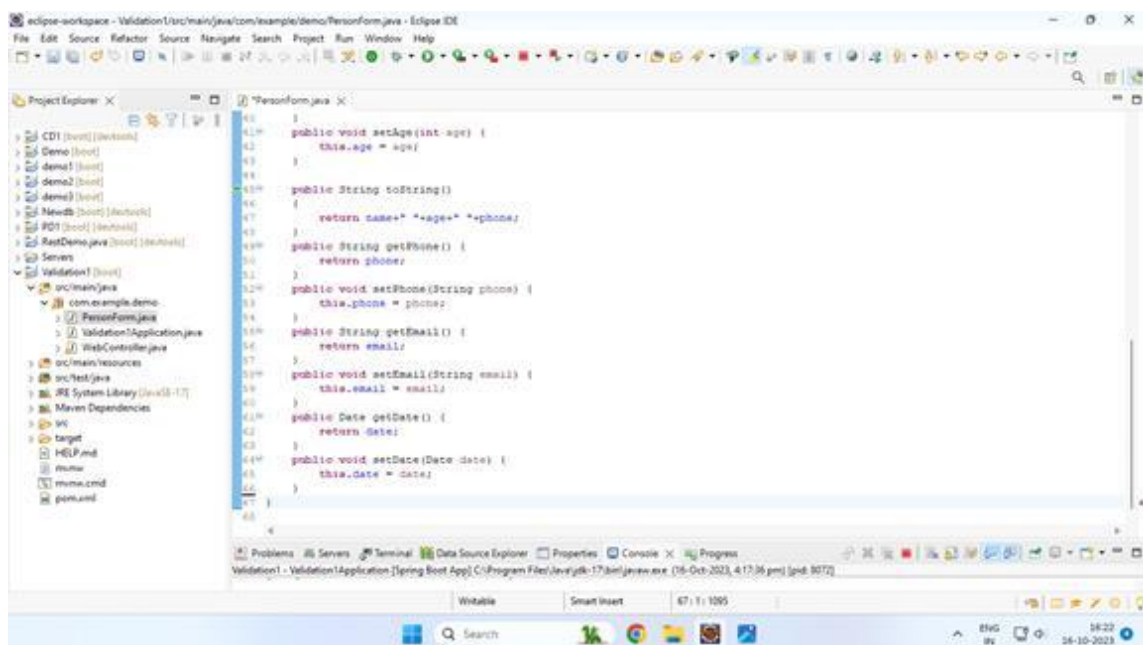
</html>
```



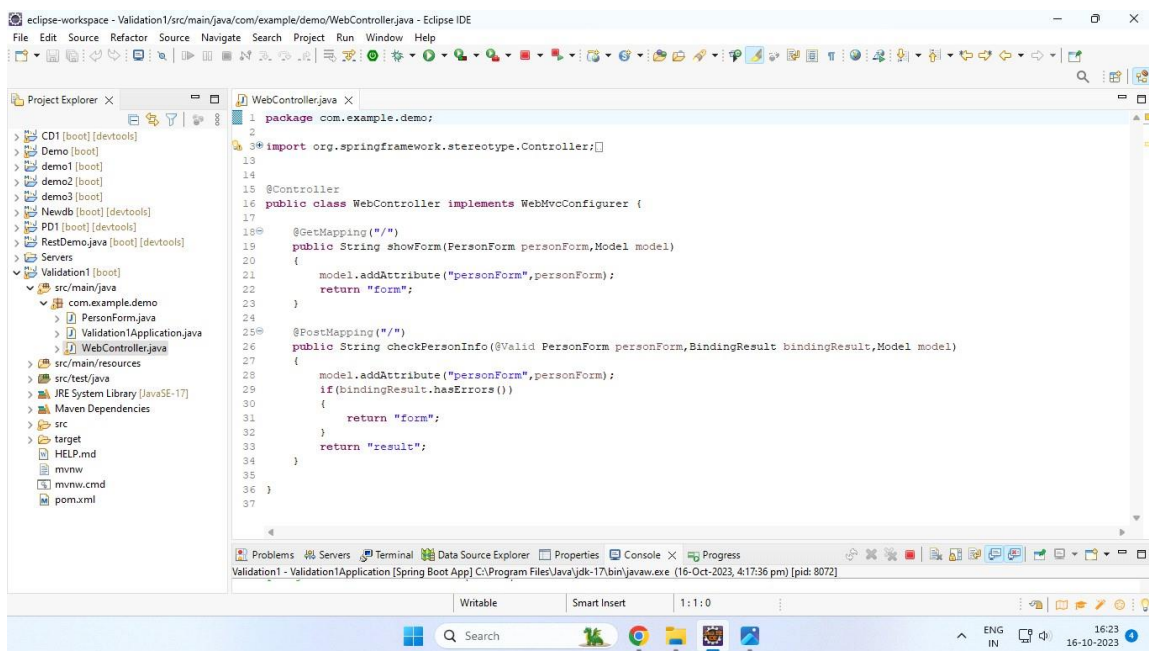
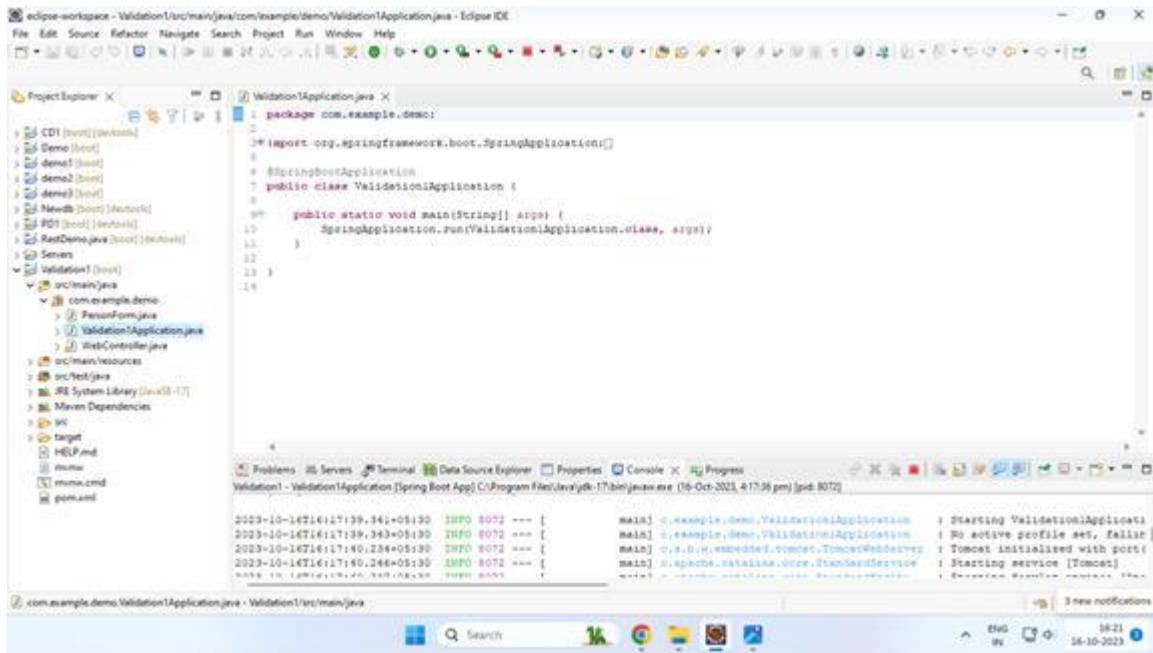




```
11 private Date date;
12
13 public String getName() {
14     return name;
15 }
16
17 public void setName(String name) {
18     this.name = name;
19 }
20
21 public int getAge() {
22     return age;
23 }
24
25 public void setAge(int age) {
26     this.age = age;
27 }
28
29 public String toString() {
30     return name + " " + age + " " + phone;
31 }
32
33 public String getPhone() {
34     return phone;
35 }
36
37 public void setPhone(String phone) {
38     this.phone = phone;
39 }
40
41 public String getEmail() {
42     return email;
43 }
44
45 public void setEmail(String email) {
```



```
46 }
47
48 public void setAge(int age) {
49     this.age = age;
50 }
51
52 public String toString() {
53     return name + " " + age + " " + phone;
54 }
55
56 public String getPhone() {
57     return phone;
58 }
59
60 public void setPhone(String phone) {
61     this.phone = phone;
62 }
63
64 public String getEmail() {
65     return email;
66 }
67
68 public void setEmail(String email) {
69     this.email = email;
70 }
71
72 public Date getDate() {
73     return date;
74 }
75
76 public void setDate(Date date) {
77     this.date = date;
78 }
79 }
```



```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3 <head>
4 <meta charset="ISO-8859-1">
5 <title>Insert title here</title>
6 </head>
7 <body>
8 <form action="#" th:action="@{"/}" th:object="${personForm}" method="post">
9 <table>
10 <tr>
11 <td>Name:</td>
12 <td><input type="text" th:field="${name}" /></td>
13 <td><th:if="${fields.hasErrors('name')}" th:errors="${name}">Name Error</td>
14 </tr>
15 <tr>
16 <td>Age:</td>
17 <td><input type="text" th:field="${age}" /></td>
18 <td><th:if="${fields.hasErrors('age')}" th:errors="${age}">Age Error</td>
19 </tr>
20 <tr>
21 <td>Phone no:</td>
22 <td><input type="text" th:field="${phone}" /></td>
23 <td><th:if="${fields.hasErrors('phone')}" th:errors="${phone}">Phone Error</td>
24 </tr>
25 <tr>
26 <td>Email:</td>
27 <td><input type="text" th:field="${email}" /></td>
28 <td><th:if="${fields.hasErrors('email')}" th:errors="${email}">Email Error</td>
29 </tr>
30 </table>
31 <td><button type="submit">Submit</button></td>
32 </tr>
33 </table>
34 </body>
35 </html>
```

```
20 </tr>
21 <tr>
22 <td>Phone no:</td>
23 <td><input type="text" th:field="${phone}" /></td>
24 <td><th:if="${fields.hasErrors('phone')}" th:errors="${phone}">Phone Error</td>
25 </tr>
26 <tr>
27 <td>Email:</td>
28 <td><input type="text" th:field="${email}" /></td>
29 <td><th:if="${fields.hasErrors('email')}" th:errors="${email}">Email Error</td>
30 </tr>
31 <tr>
32 <td>Date:</td>
33 <td><input type="text" th:field="${date}" /></td>
34 <td><th:if="${fields.hasErrors('date')}" th:errors="${date}">Date Error</td>
35 </tr>
36 <tr>
37 <td><button type="submit">Submit</button></td>
38 </tr>
39 </table>
40 </body>
41 </html>
```