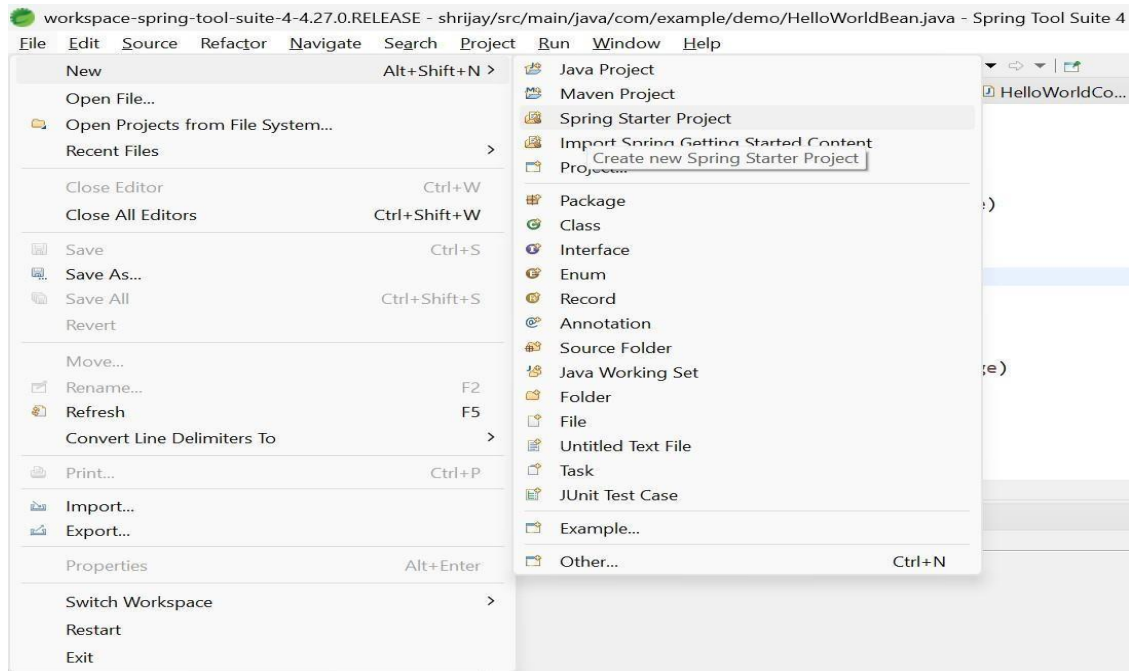


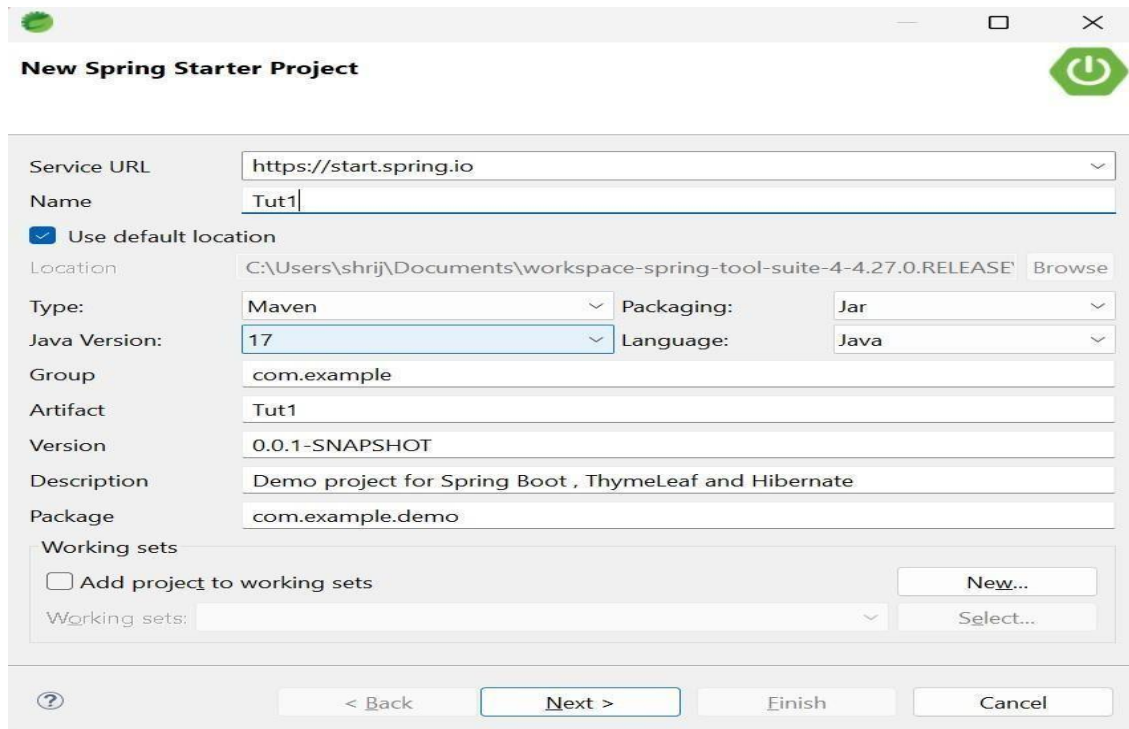
Spring Boot and RESTful Web Services

Steps to create a spring boot project :

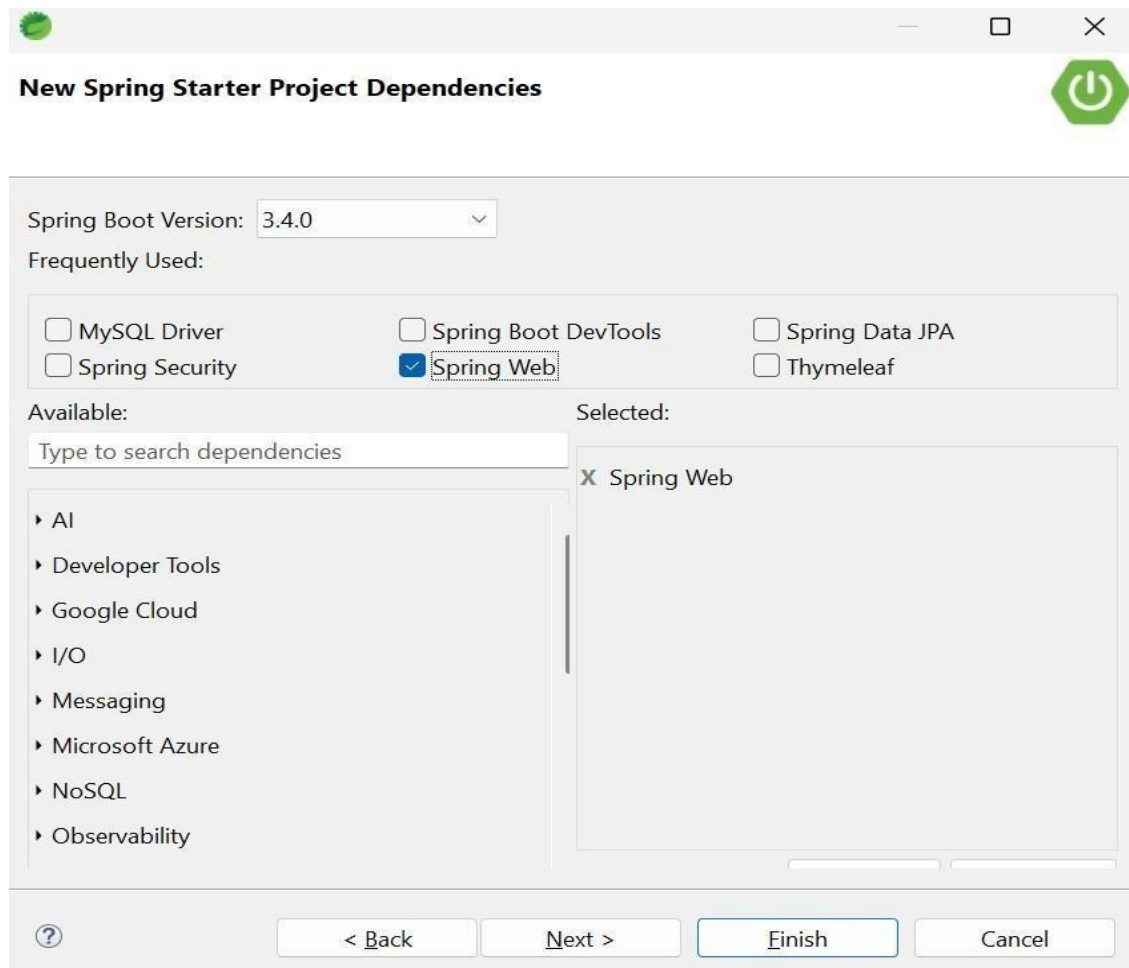
New File ⑦ Spring Starter Project



Project name ⑦next



Select spring web ⑦finish



To run the application right click on main class run as then run as java application.

Q.1) Write a program to create a simple Spring Boot application that prints a message.

HelloController.java

```
package com.example.demo;

import org.springframework.web.bind.annotation.RequestMapping; import
org.springframework.web.bind.annotation.RestController;

@RestController

public class HelloController {

    @RequestMapping("/")

    public String hello()

    {

        return "shrijay is here !";

    }

}
```

Tut1Application.java package

```
com.example.demo;

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

@SpringBootApplication

public class Tut1Application {

    public static void main(String[] args) {

        SpringApplication.run(Tut1Application.class, args);

    }

}
```

Output :



Q.2) Write a program to demonstrate RESTful Web Services with spring boot.

HelloWorldBean.java package com.example.demo;

```
public class HelloWorldBean {  
    public String message;    public  
    HelloWorldBean(String message)  
    {  
        this.message=message;  
    }  
    public String getMessage()  
    {  
        return message;  
    }  
    public void setMessage(String message)  
    {  
        this.message = message;  
    }  
    @Override  
    public String toString()  
    {  
        return String.format ("HelloWorldBean [message=%s]", message);  
    }  
}
```

HelloWorldController.java package

com.example.demo;

```
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RestController;
```

```

@RestController public class
HelloWorldController {

    @GetMapping(path="/hello-world")

public String helloWorld()

    {

        return "Batman is here!";

    }

    @GetMapping(path="/hello-world-bean")

public HelloWorldBean helloWorldBean()

    {

        return new HelloWorldBean("How are you fella? xD"); }

}

```

ShrijayApplication.java

```

package com.example.demo;

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

@SpringBootApplication

public class ShrijayApplication {

    public static void main(String[] args) {

        SpringApplication.run(ShrijayApplication.class, args);

    }

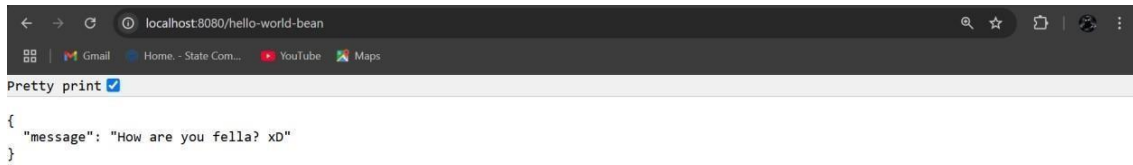
}

```

Output :



Batman is here!



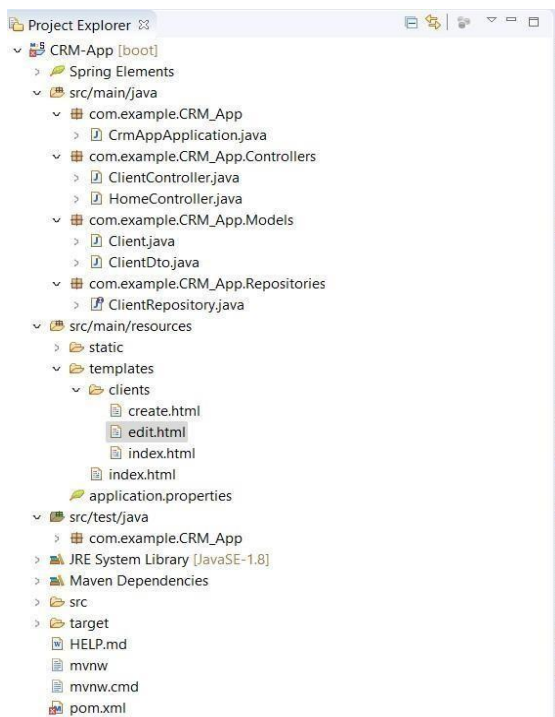
Q.3) Write a program to demonstrate database access with spring boot.

1) Go To spring initializr For Creating Springboot Project :

The screenshot shows the Spring Initializr web application at <https://start.spring.io>. The interface is divided into several sections:

- Project:** Includes radio buttons for **Gradle - Groovy**, **Gradle - Kotlin**, and **Maven** (selected).
- Language:** Includes radio buttons for **Java** (selected), **Kotlin**, and **Groovy**.
- Spring Boot:** Includes radio buttons for **3.4.1 (SNAPSHOT)**, **3.4.0** (selected), **3.3.7 (SNAPSHOT)**, and **3.3.6**.
- Project Metadata:** Includes input fields for **Group** (com.example), **Artifact** (demo), **Name** (demo), **Description** (Demo project for Spring Boot), and **Package name** (com.example.demo). It also has a **Packaging** section with **Jar** (selected) and **War** options, and a **Java** version section with **23**, **21**, and **17** (selected).
- Dependencies:** A list of dependencies with checkboxes and labels: **Spring Web** (WEB), **Spring Data JPA** (SQL), **Thymeleaf** (TEMPLATE ENGINES), **PostgreSQL Driver** (SQL), and **Validation** (JIO). A button **ADD DEPENDENCIES... CTRL + B** is present.
- Buttons:** At the bottom, there are buttons for **GENERATE CTRL + G**, **EXPLORE CTRL + SPACE**, and an ellipsis button.

• Project Explorer



2) Create a database and table (Mysql / Posgress)

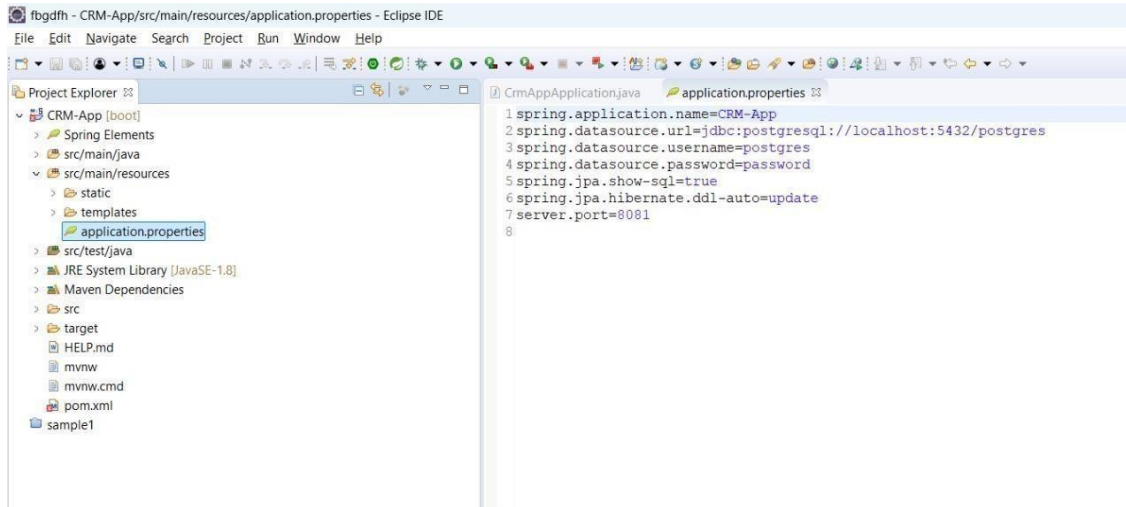
CREATE TABLE IF NOT EXISTS
clients (id SERIAL PRIMARY
KEY, address
VARCHAR(255), create_at
TIMESTAMP(6),


```

email VARCHAR(255) NOT NULL UNIQUE,
first_name
VARCHAR(255), last_name
VARCHAR(255), phone
VARCHAR(255),
status VARCHAR(255)
);

```

3) Application.properties



4) Pom.xml

```

5) <?xml version="1.0" encoding="UTF-8"?>
6) <project
  xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-
    instance"
7)   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
    4.0.0.xsd">
8)   <modelVersion>4.0.0</modelVersion>
9)   <parent>
10)     <groupId>org.springframework.boot</groupId>
11)     <artifactId>spring-boot-starter-
    parent</artifactId>
12)     <version>3.4.0</version>
13)     <relativePath/> <!-- lookup parent from
    repository
    -->
14)   </parent>
15)   <groupId>com.example</groupId>

```

```

16)      <artifactId>CRM-App</artifactId>
17)      <version>0.0.1-SNAPSHOT</version>
18)      <name>CRM-App</name>
19)      <description>Demo project for Spring
        Boot</description>
20)      <url/>
21)      <licenses>
22)      <license/>
23)      </licenses>
24)      <developers>
25)      <developer/>
26)      </developers>
27)      <scm>
28)      <connection/>
29)      <developerConnection/>
30)      <tag/>
31)      <url/>
32)      </scm>
33)      <properties>
34)      <java.version>17</java.version>
35)      </properties>
36)      <dependencies>
37)      <dependency>
38)      <groupId>org.springframework.boot</groupId>
39)      <artifactId>spring-boot-starter-
        datajpa</artifactId>
40)      </dependency>
41)      <dependency>
42)      <groupId>org.springframework.boot</groupId>
43)      <artifactId>spring-boot-
        starterthymeleaf</artifactId>
44)      </dependency>
45)      <dependency>
46)      <groupId>org.springframework.boot</groupId>
47)      <artifactId>spring-boot-
        startervalidation</artifactId>
48)      </dependency>
49)      <dependency>

```

```

50)         <groupId>org.springframework.boot</groupId>
51)         <artifactId>spring-boot-
starterweb</artifactId>
52)     </dependency>
53)
54)     <dependency>
55)     <groupId>org.postgresql</groupId>
56)     <artifactId>postgresql</artifactId>
57)     <scope>runtime</scope>
58)     </dependency>
59)     <dependency>
60)     <groupId>org.springframework.boot</groupId>
61)     <artifactId>spring-boot-
startertest</artifactId>
62)     <scope>test</scope>
63)     </dependency>
64) </dependencies>
65)
66)     <build>
67)     <plugins>
68)     <plugin>
69) <groupId>org.springframework.boot</groupId>
70)     <artifactId>spring-boot-
mavenplugin</artifactId>
71)     </plugin>
72)     </plugins>
73)     </build>
74)
75) </project>

```

5) Entity

```

package com.example.CRM_App.Models;
import java.util.Date;

```

```

import jakarta.persistence.Column;
import jakarta.persistence.Entity;
import
jakarta.persistence.GeneratedValue;

```

```

import
jakarta.persistence.GenerationType;
import jakarta.persistence.Id;
import jakarta.persistence.Table;

@Entity
@Table(name="clients")
public class Client {

    @Id

    @GeneratedValue(strategy=GenerationType.IDENTITY)
    private int id;    private String firstName;
        private String
lastName;

    @Column(unique=true,nullable=false)
    private String email; private String
phone; private String address;
    private String status;
        @Column(name =
"createAt") private Date
createAt;    public int getId()
{        return id;
    }
    public void setId(int id) {
        this.id
        = id;    }
        public String getFirstName() {
            return firstName;
        }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
    public String getLastName() {
        return lastName;
    }
}

```

```

        public void setLastName(String lastName) {
            this.lastName = lastName;
        }
    public String getEmail() {
        return
        email;    }
        public void setEmail(String email) {
            this.email = email;
        }
    public String getPhone() {
        return
        phone;    }
        public void setPhone(String phone) {
            this.phone = phone;
        }
    public String getAddress() {
        return
        address;    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getStatus() {
        return
        status;    }
        public void setStatus(String status) {
            this.status = status;
        }
        public Date getCreateAt() {
            return createAt;
        }
    public void setCreateAt(Date createAt) {
        this.createAt = createAt;
    }

}

```

B) Repository

```
package com.example.CRM_App.Repositories; import
org.springframework.data.jpa.repository.JpaRepository;

import com.example.CRM_App.Models.Client;

public interface ClientRepository extends
JpaRepository<Client,Integer> {

public Client findByEmail(String email);
}
```

C) Controller-1 package

```
com.example.CRM_App.Controllers;

import java.util.Date;
import java.util.List;

import
org.springframework.beans.factory.annotation.Autowired;
import org.springframework.data.domain.Sort; import
org.springframework.stereotype.Controller; import
org.springframework.web.bind.annotation.GetMapping;
import
org.springframework.web.bind.annotation.ModelAttribute;
import
org.springframework.web.bind.annotation.PostMapping;
import
org.springframework.web.bind.annotation.RequestMapping;
import
org.springframework.web.bind.annotation.RequestParam;

import com.example.CRM_App.Models.Client;
import
com.example.CRM_App.Models.ClientDto;
import com.example.CRM_App.Repositories.ClientRepository;
```

```

import jakarta.validation.Valid;

import org.springframework.ui.Model; import
org.springframework.validation.BindingResult;
import
org.springframework.validation.FieldError;
@Controller
@RequestMapping("/clients")
public class ClientController {

    @Autowired
    private ClientRepository clientRepo;

    @GetMapping({"/", "/"})
    public String getClient(Model model) {
        List<Client> clients =
clientRepo.findAll(Sort.by(Sort.Direction.DESC,
"id"));        model.addAttribute("clients",
clients);        return "clients/index";
    }

    @GetMapping("/create")    public
String createClient(Model model) {
    ClientDto clientDto = new ClientDto();
    model.addAttribute("clientDto", clientDto);
    return "clients/create";
}

    @PostMapping("/create")
    public String createClient(
@Valid @ModelAttribute ClientDto clientDto,
BindingResult result
    ) {

        if (clientRepo.findByEmail(clientDto.getEmail()) != null)
        {
            result.addError(            new
FieldError("clientDto", "email", clientDto.getEmail(),
false, null, null, "Email address is already used")

```

```
);  
}
```

```
        if (result.hasErrors()) {  
            return "clients/create";  
        }
```

```
        Client client = new Client();  
        client.setFirstName(clientDto.getFirstName());  
        client.setLastName(clientDto.getLastName());  
        client.setEmail(clientDto.getEmail());  
        client.setPhone(clientDto.getPhone());  
        client.setAddress(clientDto.getAddress());  
        client.setStatus(clientDto.getStatus());  
        client.setCreateAt(new Date());
```

```
        clientRepo.save(client);
```

```
        return "redirect:/clients";  
    }
```

```
@GetMapping("/edit")
```

```
public String editClient(Model model, @RequestParam int id) {  
    Client client = clientRepo.findById(id).orElse(null);
```

```
        if (client == null) {  
            return "redirect:/clients";  
        }
```

```
        ClientDto clientDto = new ClientDto();  
        clientDto.setFirstName(client.getFirstName());  
        clientDto.setLastName(client.getLastName());  
        clientDto.setEmail(client.getEmail());  
        clientDto.setPhone(client.getPhone());  
        clientDto.setAddress(client.getAddress());  
        clientDto.setStatus(client.getStatus());
```



```

model.addAttribute("client", client);
model.addAttribute("clientDto", clientDto);

return "clients/edit";
}

@PostMapping("/edit")
public String editClient(
    Model model,
    @RequestParam int id,
    @Valid @ModelAttribute ClientDto clientDto,
    BindingResult result
) {

    if (result.hasErrors()) {
        return "clients/edit";
    }

    Client client = clientRepo.findById(id).orElse(null);

    if (client == null) {
        return "redirect:/clients";
    }

    client.setFirstName(clientDto.getFirstName());
    client.setLastName(clientDto.getLastName());
    client.setEmail(clientDto.getEmail());
    client.setPhone(clientDto.getPhone());
    client.setAddress(clientDto.getAddress());
    client.setStatus(clientDto.getStatus());

    clientRepo.save(client);

    return "redirect:/clients";
}

@GetMapping("/delete")

```

```

        public String deleteClient(@RequestParam int id) {
            Client client = clientRepo.findById(id).orElse(null);

            if (client != null) {
                clientRepo.delete(client);
            }

            return "redirect:/clients";
        }
    }

```

C) Controller-2 package

```
com.example.CRM_App.Controllers;
```

```
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;
```

```
@Controller
public class HomeController {
```

```
    @GetMapping("/")
    public String home()
    {
        return "index";
    }
}

```

D) validation package

```
com.example.CRM_App.Models;
import
jakarta.validation.constraints.Email;
import
jakarta.validation.constraints.NotEmpty;
public class ClientDto {
    @NotEmpty(message = "First Name is required")
    private String firstName;
```

```

    @NotEmpty(message = "Last Name is required")
    private String lastName;

    @Email(message = "Valid email is required")
    @NotEmpty(message = "Email is required")
    private String email;

    private String phone;
    private String address;

    @NotEmpty(message = "Status is required")
    private String status;

    // Getters and Setters
    public String
    getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getLastName() {
        return lastName;
    }

    public void setLastName(String lastName) {
        this.lastName = lastName;
    }

    public String getEmail() {
        return email;
    }

```

```

    public void setEmail(String email) {
        this.email = email;
    }

    public String getPhone() {
        return phone;
    }

    public void setPhone(String phone) {
        this.phone = phone;
    }

    public String getAddress() {
return address;
    }

    public void setAddress(String address) {
        this.address = address;
    }

    public String getStatus() {
return status;
    }

    public void setStatus(String status) {
        this.status = status;
    }
}

```

E) Template File: Ensure client/index.html exists in the src/main/resources/templates/ directory.

1. [src/main/resources/templates/index.html](#)

```

<!doctype html>
<html lang="en">
<head>

  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOMLASjC
" crossorigin="anonymous">
  <link rel="stylesheet" href="style.css">
  <title>Clients Manager</title>
</head>
<body>
  <div class="container my-3">
    <h1 class="text-center mb-3" style="color:white;font-size:60px">Welcome to our
Website</h1>
    <center><a class="btn btn-primary" href="/clients">Clients</a></center>
  </div>

  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.
js" integrity="sha384-
MrcW6ZMFYIzclA8NI+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
  </body>
</html>

```

2.src/main/resources/templates/clients/index.html

```

<!doctype html>
<html lang="en" xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <!-- Bootstrap CSS -->
  <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-

```

```

EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOMLASjC
" crossorigin="anonymous">
  <title>Clients Manager</title>
</head>
<body>
  <div class="container my-4">
    <h1 class="text-center mb-3">List of Clients</h1>
    <a class="btn btn-primary" href="/clients/create">New Clients</a>

    <table class="table">
      <thead>
        <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Email</th>
          <th>Phone</th>
          <th>Address</th>
          <th>Status</th>
          <th>Created At</th>
          <th>Action</th>
        </tr>
      </thead>
      <tbody class="table-group-divider">
        <tr th:each="Client : ${clients}">
          <td th:text="${Client.id}"></td>
          <td th:text="${Client.firstName + ' ' + Client.lastName}"></td>
          <td th:text="${Client.email}"></td>
          <td th:text="${Client.phone}"></td>
          <td th:text="${Client.address}"></td>
          <td th:text="${Client.status}"></td>
          <td th:text="${Client.createAt != null ? #dates.format(Client.createAt,
'yyyyMM-dd HH:mm:ss') : 'N/A'}"></td>
          <td style="white-space:nowrap">
            <a class="btn btn-primary btn-sm"
th:href="@{/clients/edit(id=${Client.id})}">E
dit</a>
            <a class="btn btn-danger btn-sm" th:href="@{/clients/delete(id=${Client.id})}"
onclick="return confirm('Are you sure?')">Delete</a>
          </td>
        </tr>
      </tbody>
    </table>
  </div>

```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.
js" integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
</body>
</html>

```

3.src/main/resources/templates/clients/create.html

```

<!doctype html>
<html lang="en">

<head>

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.
min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWFspD3yD65VohhpUuCOMLASjC
" crossorigin="anonymous">

<title>Clients Manager</title>
</head>

<body>
<div class="container my-4">
<div class="row">
<div class="col-md-8 mx-auto rounded border p-4">
<h2 class="text-center mb-5">New Client</h2>
<form method="post" th:object="${clientDto}">
<div class="row mb-3">
<label class="col-sm-4 col-form-label">First Name</label>
<div class="col-sm-8">
<input class="form-control" th:field="${clientDto.firstName}"
/>
<p th:if="${#fields.hasErrors('firstName')}"
th:errorclass="text-danger"
th:errors="${clientDto.firstName}"></p>
</div>
</div>

```

```

<div class="row mb-3">
  <label class="col-sm-4 col-form-label">Last Name</label>
  <div class="col-sm-8">
    <input class="form-control" th:field="${clientDto.lastName}">
    <p th:if="${#fields.hasErrors('lastName')}}" th:errorclass="text-danger"
th:errors="${clientDto.lastName}">
      </p>
    </div>
  </div>

```

```

<div class="row mb-3">
  <label class="col-sm-4 col-form-label">Email</label>
  <div class="col-sm-8">
    <input class="form-control" th:field="${clientDto.email}">
    <p th:if="${#fields.hasErrors('email')}}" th:errorclass="text-danger"
th:errors="${clientDto.email}"></p>
  </div>
</div>

```

```

<div class="row mb-3">
  <label class="col-sm-4 col-form-label">Phone</label>
  <div class="col-sm-8">
    <input class="form-control" th:field="${clientDto.phone}">
    <p th:if="${#fields.hasErrors('phone')}}" th:errorclass="text-danger"
th:errors="${clientDto.phone}"></p>
  </div>
</div>

```

```

<div class="row mb-3">
  <label class="col-sm-4 col-form-label">Address</label>
  <div class="col-sm-8">
    <input class="form-control" th:field="${clientDto.address}">
    <p th:if="${#fields.hasErrors('address')}}" th:errorclass="text-danger"
th:errors="${clientDto.address}">
      </p>
    </div>
  </div>

```

```

<div class="row mb-3">
  <label class="col-sm-4 col-form-label">Status</label>
  <div class="col-sm-8">
    <select class="form-select" th:field="${clientDto.status}">

```



```

        <option value="New">New</option>
        <option value="Permanent">Permanent</option>
        <option value="Lead">Lead</option>
        <option value="Occasional">Occasional</option>
        <option value="Inactive">Inactive</option>
    </select>
    <p th:if="{#fields.hasErrors('status')}" th:errorclass="text-danger"
th:errors="{clientDto.status}"></p>
</div>
</div>
<div class="row">
    <div class="offset-sm-4 col-sm-4 d-grid">
        <button type="submit" class="btn btn-primary">Submit</button>
    </div>
    <div class="col-sm-4 d-grid">
        <a class="btn btn-outline-primary" href="/clients">Cancel</a>
    </div>
</div>

```

```

</form>

```

```

</div>
</div>
</div>

```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.
js" integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
</body>

```

```

</html>

```

<4.src/main/resources/templates/clients/edit.html>

```

<!doctype html>
<html lang="en">

```

```

<head>

```

```

<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- Bootstrap CSS -->
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.
min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC
" crossorigin="anonymous">

<title>Clients Manager</title>
</head>

<body>
<div class="container my-4">
  <div class="row">
    <div class="col-md-8 mx-auto rounded border p-4">
      <h2 class="text-center mb-5">Edit Client</h2>
      <form method="post" th:object="${clientDto}">

        <div class="row mb-3">
          <label class="col-sm-4 col-form-label">Client Id</label>
          <div class="col-sm-8">
            <input class="form-control-plaintext" readonly th:value="${client.id}" />
          </div>
        </div>

        <div class="row mb-3">
          <label class="col-sm-4 col-form-label">First Name</label>
          <div class="col-sm-8">
            <input class="form-control" th:field="${clientDto.firstName}" required />
            <p th:if="${#fields.hasErrors('firstName')}" th:errorclass="text-danger"
th:errors="${clientDto.firstName}"></p>
          </div>
        </div>

        <div class="row mb-3">
          <label class="col-sm-4 col-form-label">Last Name</label>
          <div class="col-sm-8">
            <input class="form-control" th:field="${clientDto.lastName}"
required>
            <p th:if="${#fields.hasErrors('lastName')}"
th:errorclass="text-danger" th:errors="${clientDto.lastName}">
          </p>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>

```

```

    </div>
</div>

<div class="row mb-3">
    <label class="col-sm-4 col-form-label">Email</label>
    <div class="col-sm-8">
        <input class="form-control" th:field="${clientDto.email}"
required>
        <p th:if="${#fields.hasErrors('email')}"
th:errorclass="text-danger" th:errors="${clientDto.email}"></p>
    </div>
</div>

<div class="row mb-3">
    <label class="col-sm-4 col-form-label">Phone</label>
    <div class="col-sm-8">
        <input class="form-control" th:field="${clientDto.phone}"
required>
        <p th:if="${#fields.hasErrors('phone')}"
th:errorclass="text-danger" th:errors="${clientDto.phone}"></p>
    </div>
</div>
<div class="row mb-3">
    <label class="col-sm-4 col-form-label">Address</label>
    <div class="col-sm-8">
        <input class="form-control" th:field="${clientDto.address}"
required>
        <p th:if="${#fields.hasErrors('address')}"
th:errorclass="text-danger" th:errors="${clientDto.address}">
    </p>
    </div>
</div>

<div class="row mb-3">
    <label class="col-sm-4 col-form-label">Status</label>
    <div class="col-sm-8">
        <select class="form-select" th:field="${clientDto.status}">
            <option value="New">New</option>
            <option value="Permanent">Permanent</option>
            <option value="Lead">Lead</option>
            <option value="Occasional">Occasional</option>
            <option value="Inactive">Inactive</option>
        </select>
    </div>
</div>

```

```

        <p th:if="{#fields.hasErrors('status')}}" th:errorclass="text-danger"
th:errors="{clientDto.status}"></p>
    </div>
</div>

<div class="row mb-3">
    <label class="col-sm-4 col-form-label">Created At</label>
    <div class="col-sm-8">
        <input
            class="form-control-plaintext"
            th:value="{client.CreateAt}" required/>
        </div>
    </div>
    <div class="row">
        <div class="offset-sm-4 col-sm-4 d-grid">
            <button type="submit" class="btn btn-primary">Update</button>
        </div>
        <div class="col-sm-4 d-grid">
            <a class="btn btn-outline-primary" href="/clients">Cancel</a>
        </div>
    </div>

</form>

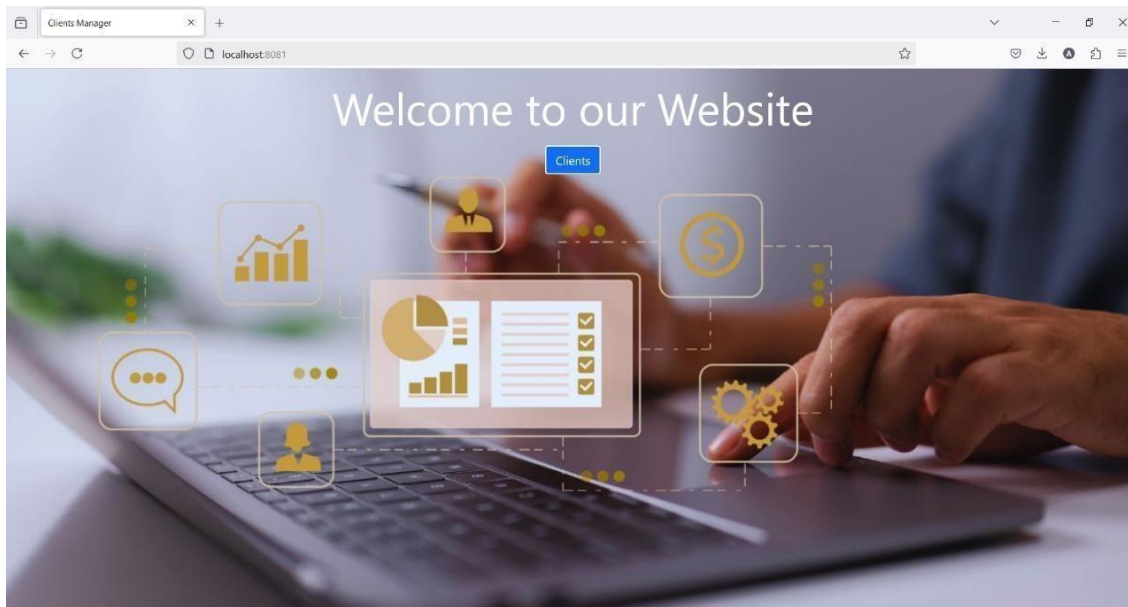
</div>
</div>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.
js" integrity="sha384-
MrcW6ZMFYIzclA8NI+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous"></script>
</body>

</html>

```

† Output:-



clients

List of Clients

[New Clients](#)

ID	Name	Email	Phone	Address	Status	Created At	Action
15	Raj Singh	raj6465416@gmail.com	46451354689	hostel	Inactive	2024-12-08 15:00:39	Edit Delete
14	ashish dgdh	ashish73737447@gmail.com	yfyvjy	gykgkj	Permanent	2024-12-08 14:56:24	Edit Delete
13	harsh patil	harsh25@gmail.com	9856418521	dombevili	New	2024-12-08 14:08:11	Edit Delete
10	Patricia Thomas	patricia.thomas@example.com	012-345-6789	101 Fir Rd	Active	2023-12-07 00:00:00	Edit Delete
9	James Taylor	james.taylor@example.com	901-234-5678	890 Chestnut Blvd	Active	2023-12-06 00:00:00	Edit Delete
7	David Martin	david.martin@example.com	789-012-3456	234 Elm St	Active	2023-12-02 00:00:00	Edit Delete
6	Lisa Davis	lisa.davis@example.com	678-901-2345	987 Spruce Ct	Active	2023-12-04 00:00:00	Edit Delete
5	Robert Brown	robert.brown@example.com	567-890-1234	654 Cedar Ln	Inactive	2023-11-29 00:00:00	Edit Delete
4	Susan Wilson	susan.wilson@example.com	456-789-0123	321 Birch Blvd	Active	2023-12-05 00:00:00	Edit Delete
3	Michael Jones	michael.jones@example.com	345-678-9012	789 Pine Rd	Active	2023-12-03 00:00:00	Edit Delete
2	Jane Smith	jane.smith@example.com	234-567-8901	456 Oak Ave	Inactive	2023-11-25 00:00:00	Edit Delete
1	John Doe	john.doe@example.com	123-456-7890	123 Maple St	Active	2023-12-01 00:00:00	Edit Delete

Clients Manager

localhost:8081/clients/create

New Client

First Name

Last Name

Email

Phone

Address

Status

Clients Manager

localhost:8081/clients/edit?id=15

Edit Client

Client Id

First Name

Last Name

Email

Phone

Address

Status

Created At

Clients Manager

localhost:8081/clients

List of Clients

ID	Name	Email	Phone	Address	Status	Created At	Action
15	Raj Singh	raj6465416@gmail.com	46451354689	hostel	Inactive	2024-12-08 15:00:39	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
14	ashish dghdh	ashish73737447@gmail.com	yfyjyj	gykgkj	Permanent	2024-12-08 14:56:24	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
13	harsh patil	harsh25@gmail.com	9856418521	dombevili	New	2024-12-08 14:08:11	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
10	Patricia Thomas	patricia.thomas@example.com			ive	2023-12-07 00:00:00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
9	James Taylor	james.taylor@example.com			ive	2023-12-06 00:00:00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
7	David Martin	david.martin@example.com			ive	2023-12-02 00:00:00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
6	Lisa Davis	lisa.davis@example.com	678-901-2345	987 Spruce Ct	Active	2023-12-04 00:00:00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
5	Robert Brown	robert.brown@example.com	567-890-1234	654 Cedar Ln	Inactive	2023-11-29 00:00:00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
4	Susan Wilson	susan.wilson@example.com	456-789-0123	321 Birch Blvd	Active	2023-12-05 00:00:00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
3	Michael Jones	michael.jones@example.com	345-678-9012	789 Pine Rd	Active	2023-12-03 00:00:00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
2	Jane Smith	jane.smith@example.com	234-567-8901	456 Oak Ave	Inactive	2023-11-25 00:00:00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>
1	John Doe	john.doe@example.com	123-456-7890	123 Maple St	Active	2023-12-01 00:00:00	<input type="button" value="Edit"/> <input type="button" value="Delete"/>

localhost:8081/clients/edit?id=15

localhost:8081

Are you sure?