**N**N**A**a**M**m**E**e **-**: S**A**u**D**m**IT**it **Y**K**A**um**K**a**U**r **MAR JHA BR**R**A**ol**N**l N**C**o**H**: 2**-**20**C**1**S**3**E**31**R**6**E**9**G**00**IO**57**NAL SE**B**C**ra**T**n**I**c**O**h**N**: CS**-** E**A** R

LAB NO – 001 D

1. **Create a REST controller in Spring Boot that handles a GET request to return a welcome message.**

**Use the endpoint /api/welcome. Test it using Postman. Requirements:**

**Define a @RestController**

**Use @GetMapping("/api/welcome")**

**Return a simple message like "Hello Mr Are Good!** import org.springframework.web.bind.annotation.GetMapping; import org.springframework.web.bind.annotation.RestController;

@RestController

public class WelcomeController {

@GetMapping("/api/welcome") public String welcome() {

return "Welcome to Spring Boot REST API!";

}

}

1. **Create a controller in Spring Boot that passes a user's name and email to the view.**

**Display the name and email on the view page using expression language.**

**Requirements:**

**Use @Controller and Model**

**Pas data like model.addAttribute("name", "Muskan")**

**Show it using ${name} in the view**

import org.springframework.stereotype.Controller; import org.springframework.ui.Model;

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

@Controller

public class UserController {

@GetMapping("/user")

public String user(Model model) { model.addAttribute("name", "Ankit"); model.addAttribute("email", "[ankit@example.com](mailto:ankit@example.com)"); return "user";

}

}

<!DOCTYPE html>

<html xmlns:th="[http://www.thymeleaf.org](http://www.thymeleaf.org/)">

<head>

<title>User Info</title>

</head>

<body>

<h1>User Information</h1>

<p>Name: <span th:text="${name}"></span></p>

<p>Email: <span th:text="${email}"></span></p>

</body>

</html>



1. **Create a form view that accepts name and email, and submits to a controller using**

**@PostMapping.**

**Show the submitted data back in another view page.**

**Requirements:**

**Create a User model class**

**Use @ModelAttribute or @RequestParam in controller Show submitted data in the result page**

public class User { private String name; private String email;

// Getters and Setters

}

import org.springframework.stereotype.Controller; import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.\*;

@Controller

public class FormController {

@GetMapping("/form")

public String showForm(Model model) { model.addAttribute("user", new User()); return "form";

}

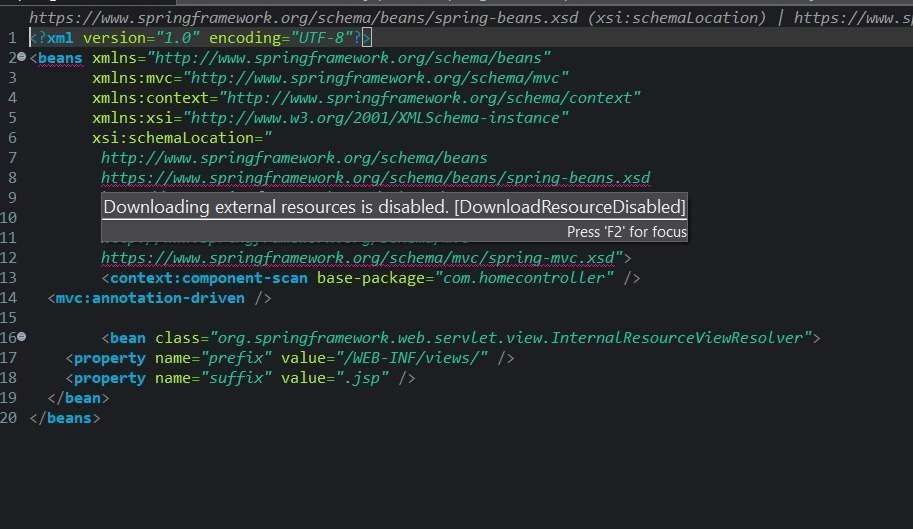
@PostMapping("/submit")

public String submitForm(@ModelAttribute User user, Model model) { model.addAttribute("name", user.getName()); model.addAttribute("email", user.getEmail());

return "result";

}

}



<!DOCTYPE html>

<html xmlns:th="[http://www.thymeleaf.org](http://www.thymeleaf.org/)">

<head>

<title>Submit Form</title>

</head>

<body>

<h1>Submit Your Info</h1>

<form th:action="@{/submit}" th:object="${user}" method="post"> Name: <input type="text" th:field="\*{name}" /><br/>

Email: <input type="email" th:field="\*{email}" /><br/>

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

</form>

</body>

</html>

1. **Create a REST endpoint that accepts a User object (name G email) in JSON format using @RequestBody.**

**Return the same object with a custom message. Requirements:**

**Create a POST endpoint: /api/user**

Use @RequestBody User user Test it in Postman.

public class User { private String name; private String email;

// Getters and Setters

}

import org.springframework.web.bind.annotation.\*;

@RestController

public class UserRestController {

@PostMapping("/api/user")

public String createUser(@RequestBody User user) {

return "User received: Name - " + user.getName() + ", Email - " + user.getEmail();

}

}

{

"name": "Ankit",

"email": "[ankit@example.com](mailto:ankit@example.com)"

}

