# Home Delivery Application

# Question Scenario

The ***Home Delivery Application*** is built using Spring Boot with Thymeleaf*.* The application is used to deliver the products to customers safely to their addresses. The application consists of 4 entities; Delivery Partner, Product, User and User Address.

Admin wants to assign the delivery of user ***Robert* to** a delivery person whose name is ***James.*** He wants the order to be delivered between ***5:00 PM*** to ***8:00 PM.*** For that, he needs to check the ***availability*** of ***James*** for the required duration. His availability is from ***4:00 PM*** to ***9:00 PM.***

Since ***James*** is available, assign the delivery to him and change his availability to ***8:00 PM*** to ***01:00 AM*** (Since his total duration of availability is ***5 hours***).

If ***James*** is not available, the admin has to assign the delivery to others. For that, he needs to check whether or not the location of others is the same as ***Robert’s*** delivery address, also their availability. If he finds any delivery person with required criteria, assign the delivery to him and change his ***availability***. If he finds multiple delivery guys with required criteria, assign the delivery to the person with the highest ***ratings***.

If there are no delivery guys available for requirement, throw an exception with the message saying,

**Sorry...! James is already occupied with other deliveries. We are unable to assign delivery of your product to other delivery persons also right now.**

# Objectives

1. ***Database*** layer with required code exists in the application. Complete the logic of methods present in ***Presentation (Controller)*** and ***Business (Service)*** layers. You are not allowed to change the method signature of ***Presentation*** and ***Business*** layers

1. We have defined the below method in the ***Presentation*** Layer. You must complete the body of it.

@PostMapping("/assign/delivery")

**public** ModelAndView **assignDelivery**(@RequestParam("userId") Long userId, @RequestParam("deliveryPartnerId") Long deliveryPartnerId,

@RequestParam("userAddress") String userAddress, @RequestParam("requiredFromDate") String requiredFromDate, @RequestParam("requiredToDate") String requiredToDate,

@RequestParam("requiredFromTime") String requiredFromTime,

@RequestParam("requiredToTime") String requiredToTime, @ModelAttribute User user, Model model, BindingResult result, RedirectAttributes redirectAttributes) **throws** Exception {

1. We have defined below methods in ***Business***Layer. You must complete the body of them.

**public** AssignDeliveryResponseDTO **assignDelivery** (AssignDeliveryRequestDTO deliveryDTO) **throws** Exception

In this method, check whether or not User, User address, selected Delivery Partner are present in the database. If proceed then proceed, else throw the exception.

private boolean **checkAvailability**(UserAddress userpresentAddress, AssignDeliveryRequestDTO deliveryDTO, DeliveryPartner deliveryPartner) {

In this helper method, check the availability of the delivery partner.

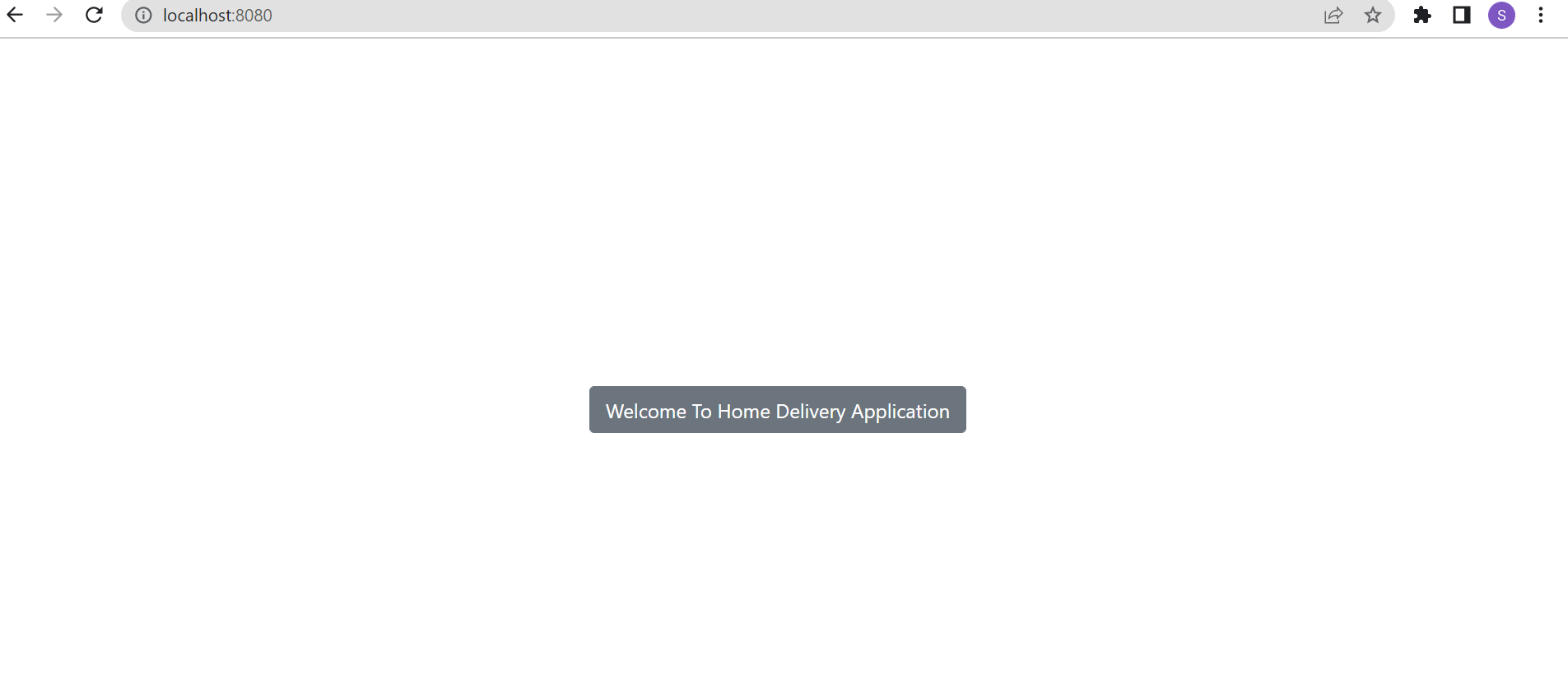
private void **assignDelivery**(User user, DeliveryPartner deliveryPartner, AssignDeliveryRequestDTO deliveryDTO,

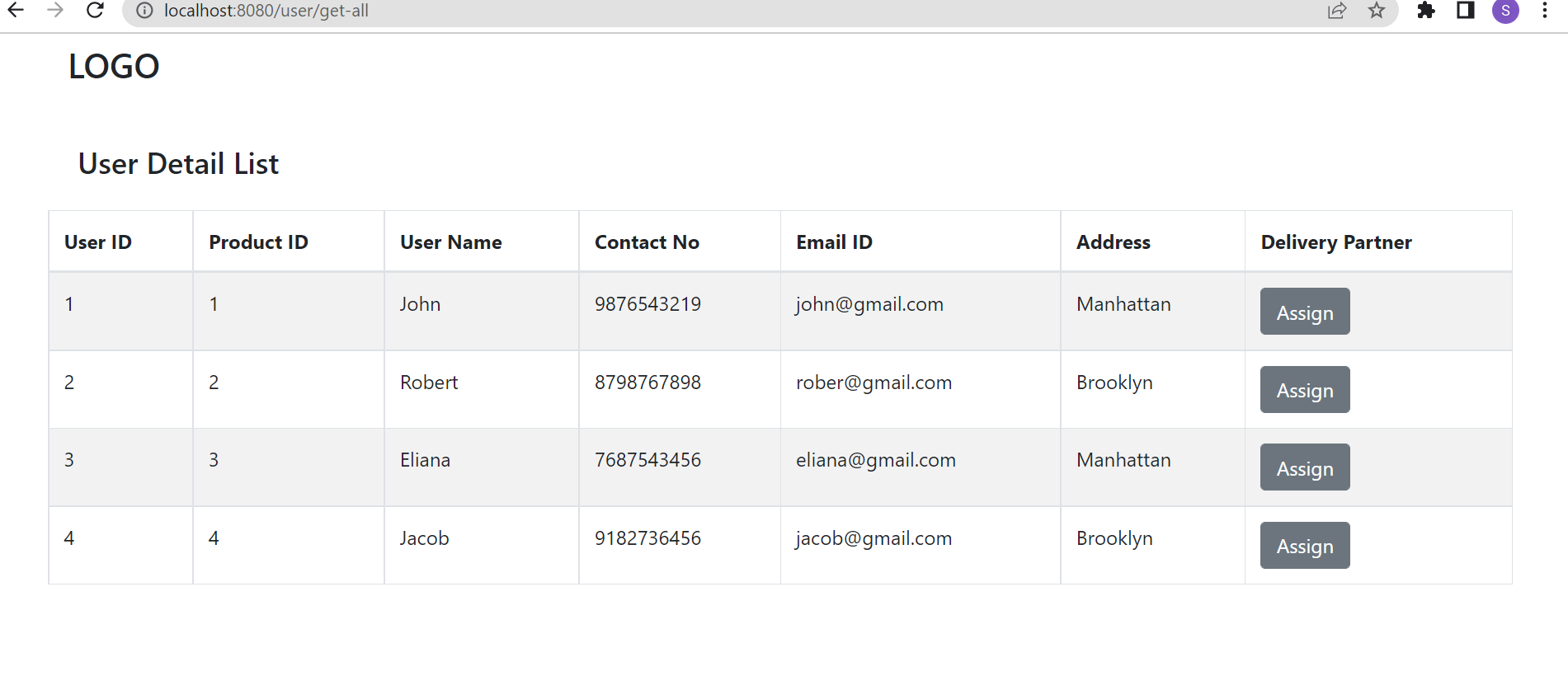
AssignDeliveryResponseDTO responseDTO) {

In this helper method, update the availability of delivery partner and do mapping between user and delivery partner.

# Sample Screens

1. **Home Page Screen:** This page contains the welcome home page button, on click of it details of list of users whose product need to be delivered will be displayed. Admin can click on the specified user to assign a delivery partner to him.

****

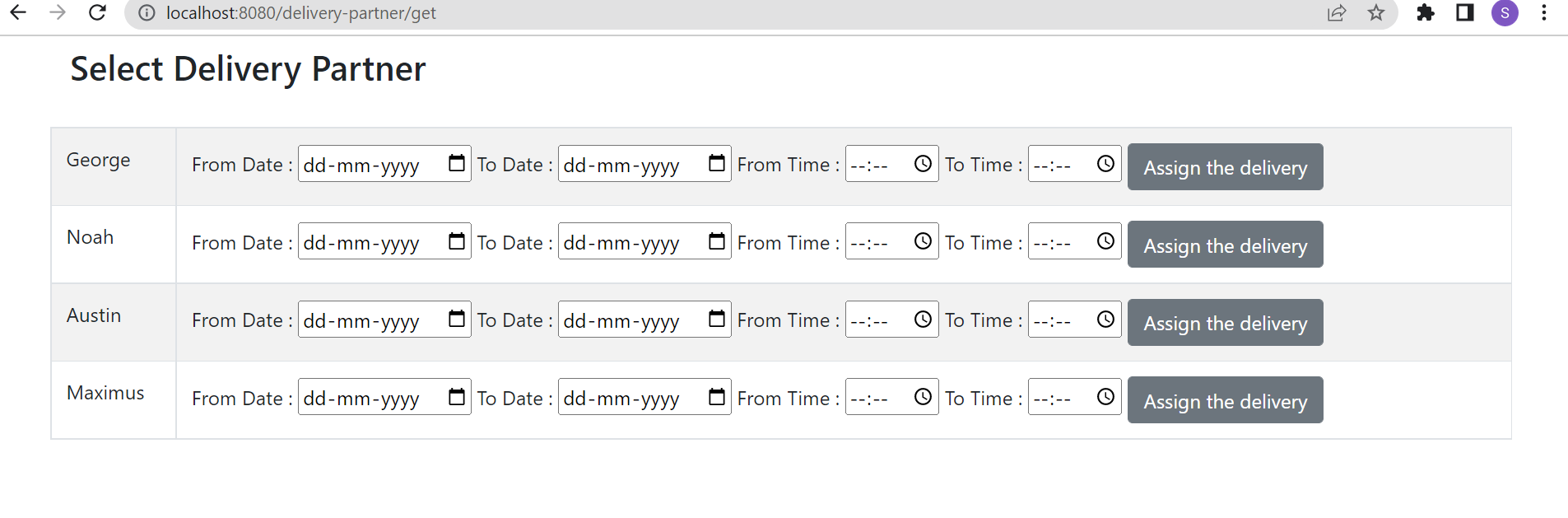


1. **Assign delivery to delivery partner:** When admin clicks on ***Assign*** button, details of all the delivery partners will be displayed. Admin can select the ***RequiredDateFrom***, ***RequiredDateTo***, ***RequiredTimeFrom*** and ***RequiredTimeTo.*** As soon as he clicks on ***Assign*** button, there will be 3 cases,

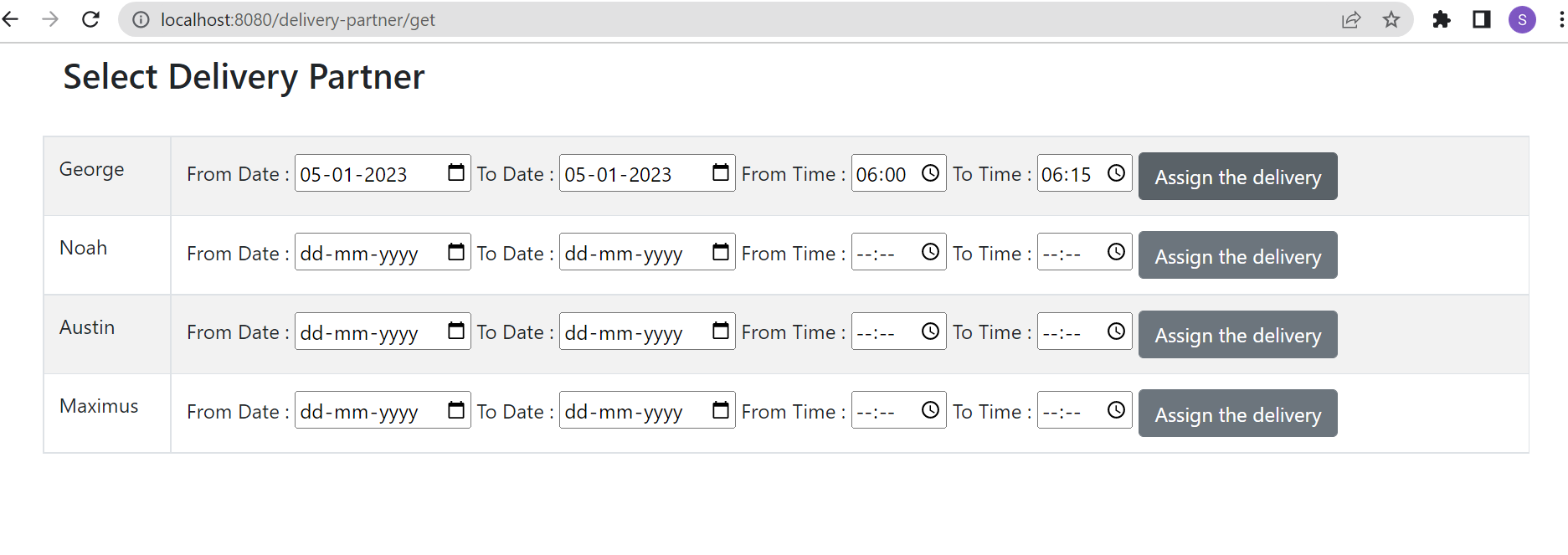
* If the selected delivery partner is available, success message will be displayed.
* If the selected delivery partner is not available, other delivery partners will be checked for given criteria, if anyone is found a success message will be displayed.
* If no one is available, an error message will be displayed.

**Note ->** Assigning the delivery refers, mapping between user and delivery partner.

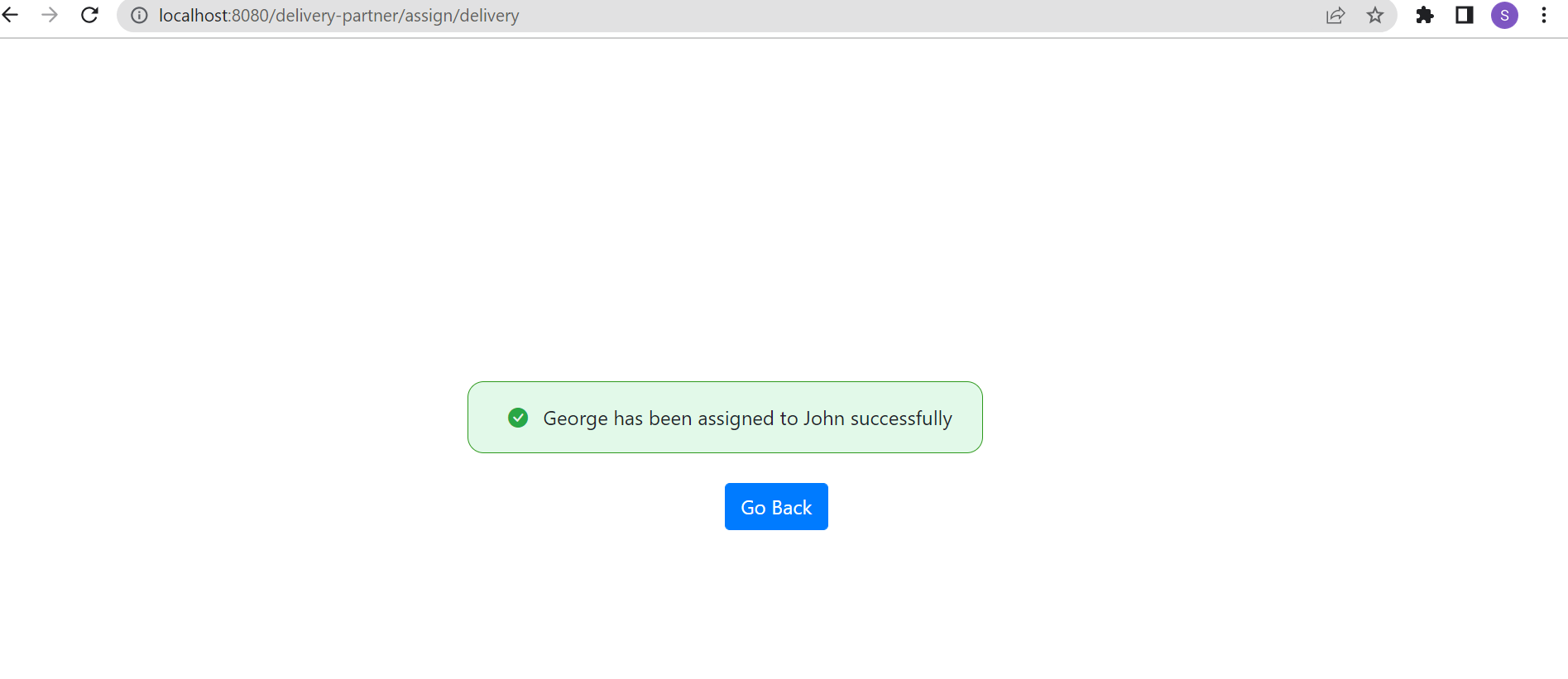
**Displaying all delivery partners ->**



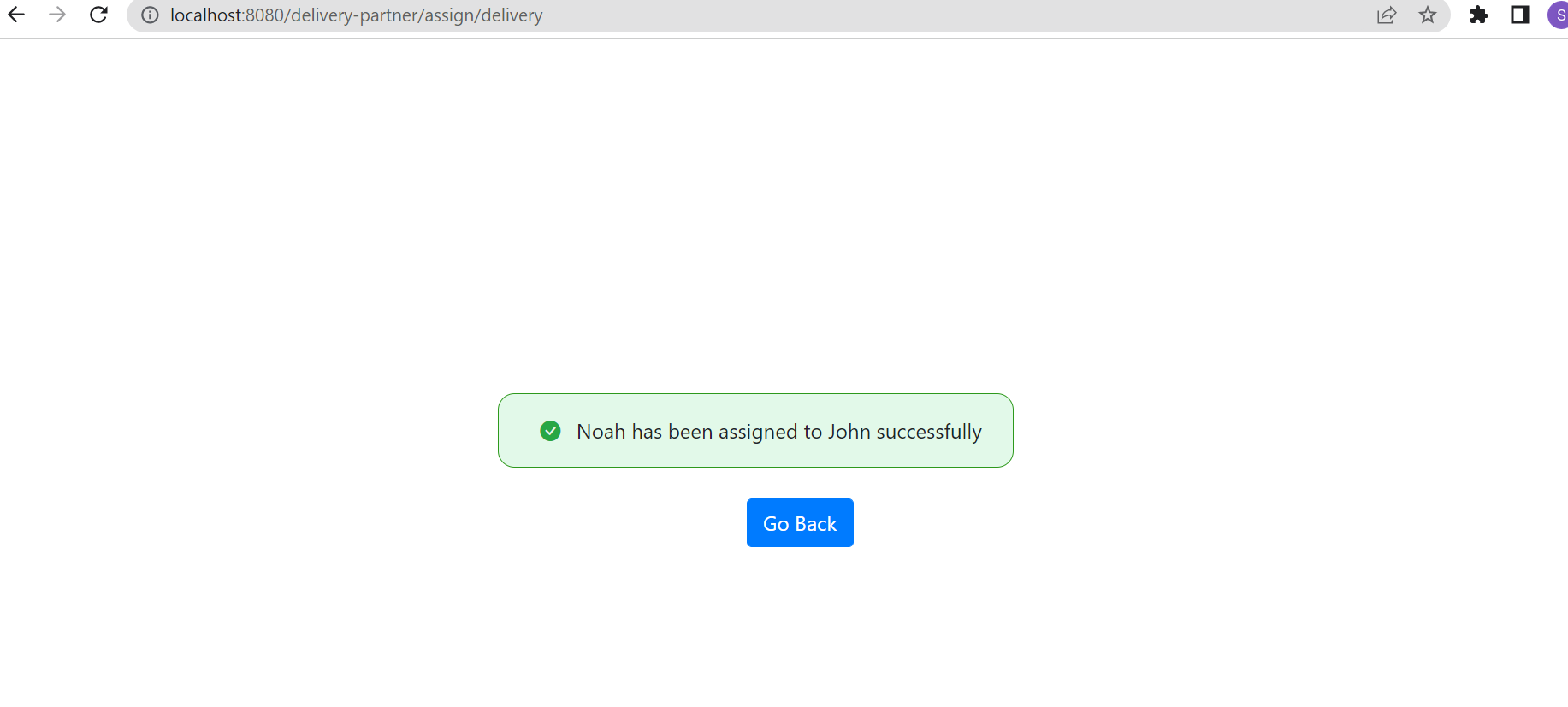
**Admin wants to assign the delivery to George ->**



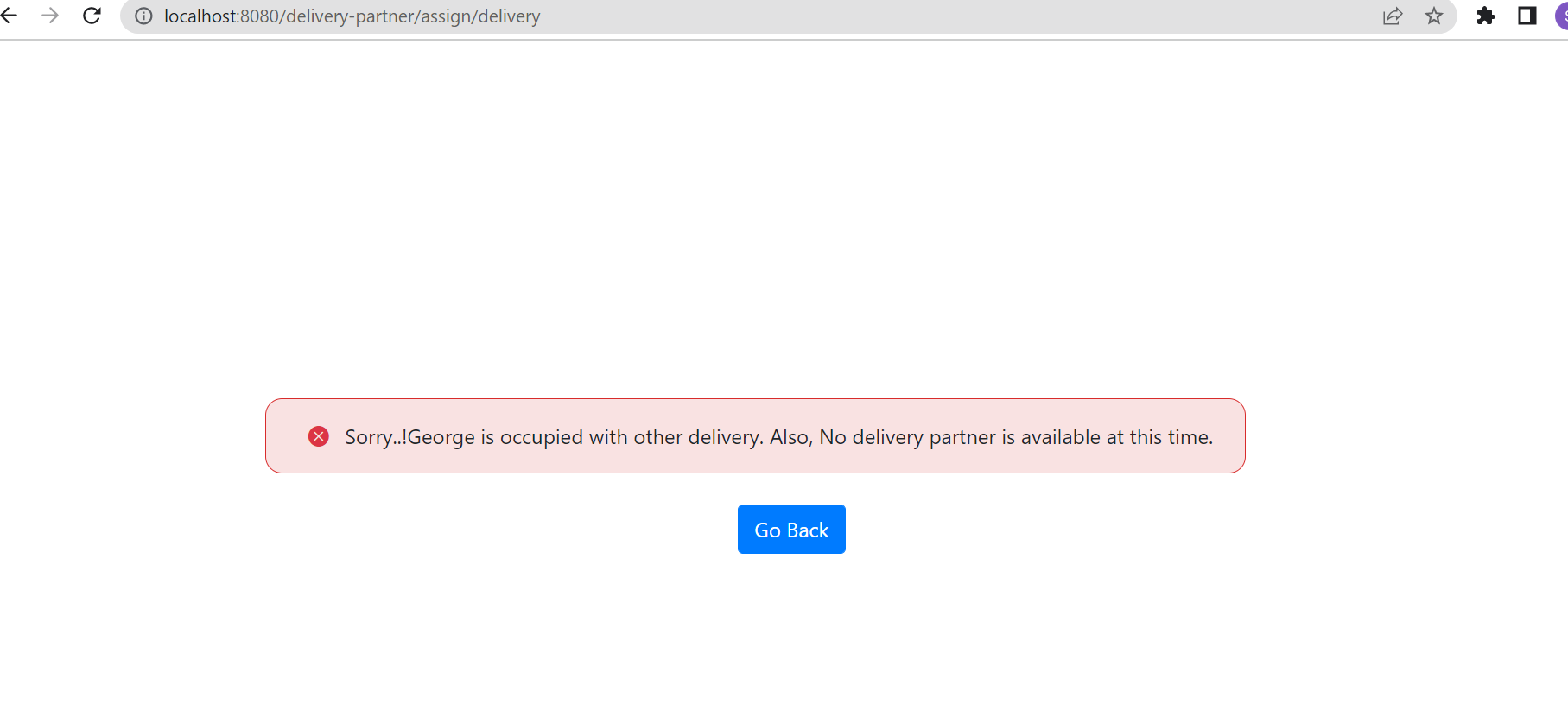
When we hover on Delivery partner’s name, his availability will be displayed as tooltip.

**Case 1: Displaying the success message, as George is available ->**  


**Case 2: When George is not available, condition has been checked for other delivery partners, Since Noah is available, delivery has been assigned to him.**



**Case 3: If no one is available, error message is displayed as shown below**

****

**Instructions**

You are not allowed to make any changes in below files.

1. application.properties
2. pom.xml
3. com.example.demo.controller.UserController.java
4. com.example.demo.controller.UserService.java

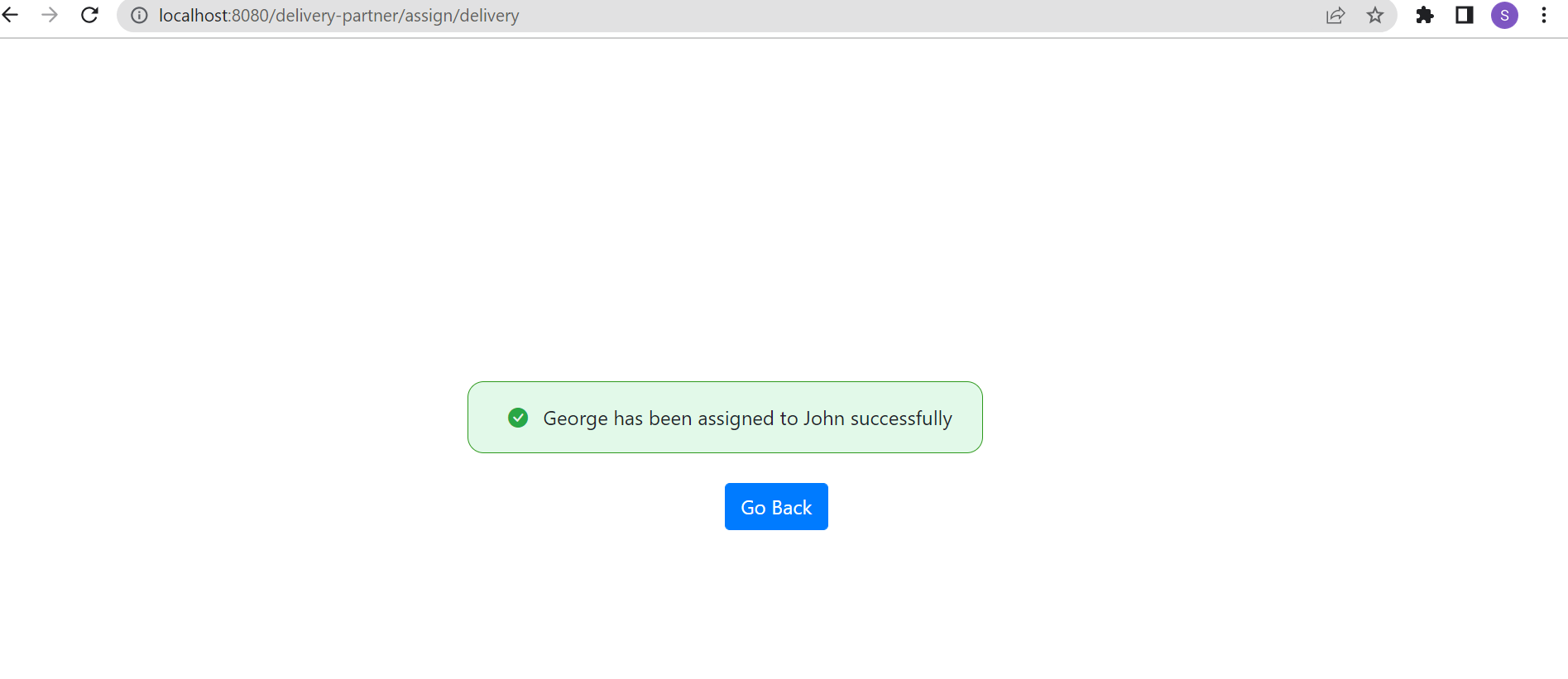
You are not allowed to make any changes in the files present in below packages.

1. com.example.demo.pojo
2. com.example.demo.dto
3. com.example.demo.exception
4. com.example.demo.repository
5. com.example.demo.util

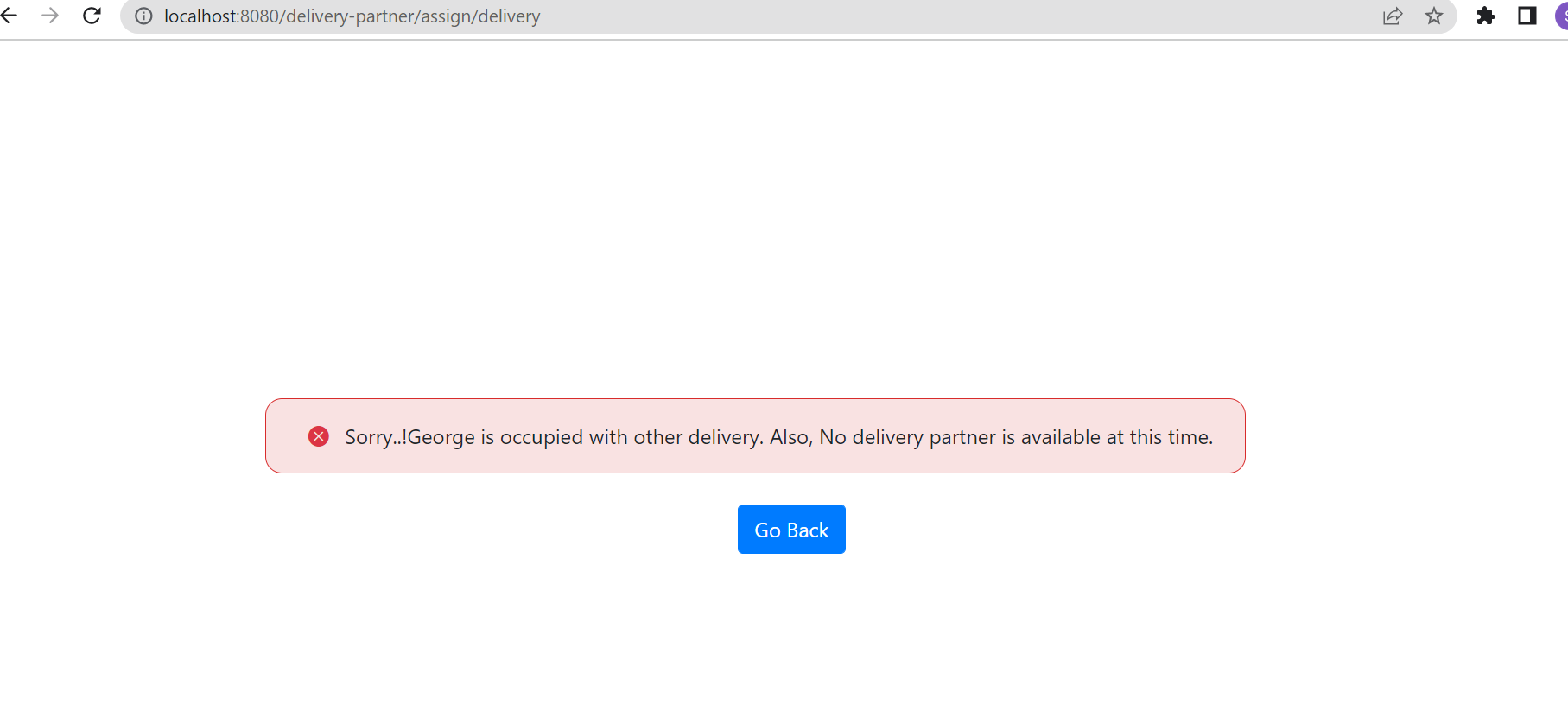
You are not allowed to insert new data to the database.

# Output

**If you find the any delivery guy with required criteria, the final UI image should be as below,**



**If you did not find the any delivery guy with required criteria, error message should be displayed as below**

****

# Build, Deploy and Run

1. Load the project in the IDE
2. Update the maven dependency
3. mvn life cycle to build & deploy
4. Run the project

# Project Files

You can only work upon the below files. You are allowed to create custom exception classes and controller advice class.

1. DeliveryPartnerController.java
2. DeliveryPartnerService.java