**Exercise 4: Online Bookstore - Processing Request Body and Form Data**

**Business Scenario:**

In this exercise, we are tasked with creating endpoints to handle customer registration by processing both JSON request bodies and form data submissions. These functionalities will enable our online bookstore to accept and manage customer registrations effectively.

**Objective:**

1. Implement a POST endpoint to handle customer registration by accepting a JSON request body.
2. Implement an endpoint to process form data for customer registrations.

**1. Request Body Handling:**

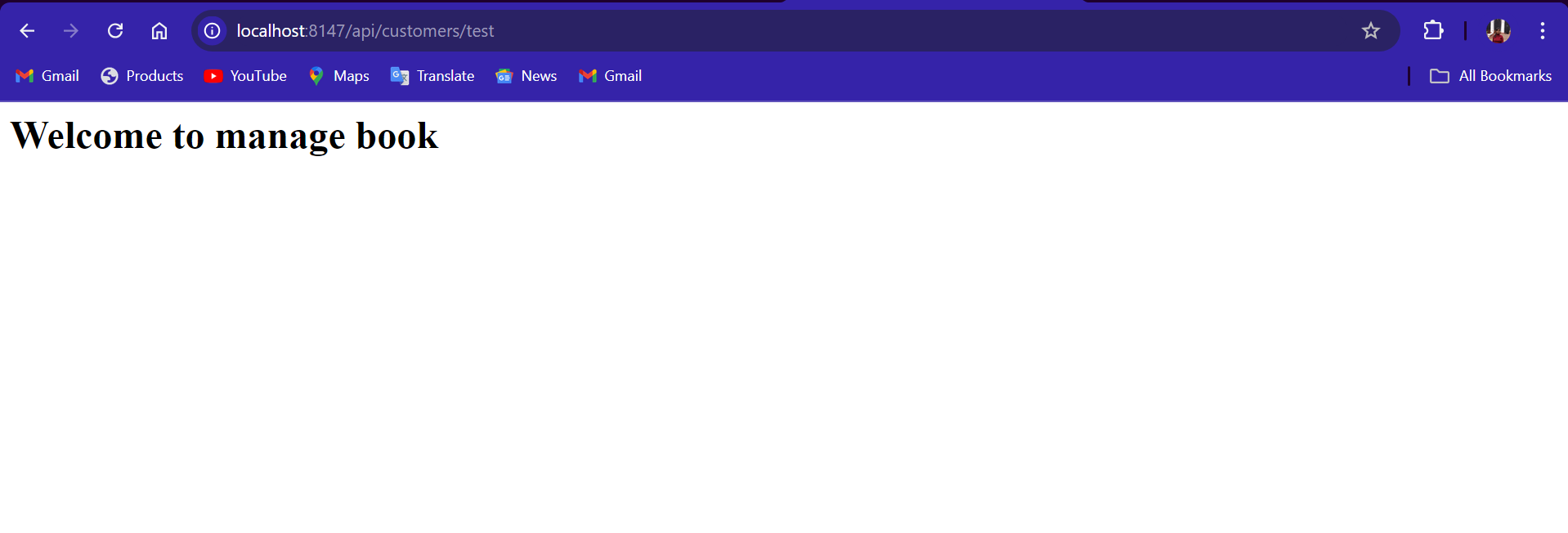
* **Purpose:** To create a new customer by accepting data in JSON format.
* **Implementation Details:**
  + **Endpoint:** /api/customers/save1
  + **HTTP Method:** POST
  + **Expected Request Body:** A JSON object representing the customer's details, including name, emailid, and password.
  + **Functionality:**
    - The JSON request body is automatically mapped to the Customer entity by Spring's @RequestBody annotation.
    - The customer details are then saved to the database using the CustomerService class.
  + **Return Value:**
    - Returns a success message upon successful registration.
    - Redirects to the registration confirmation page (or could return the saved customer object depending on the application flow).

**2. Form Data Handling:**

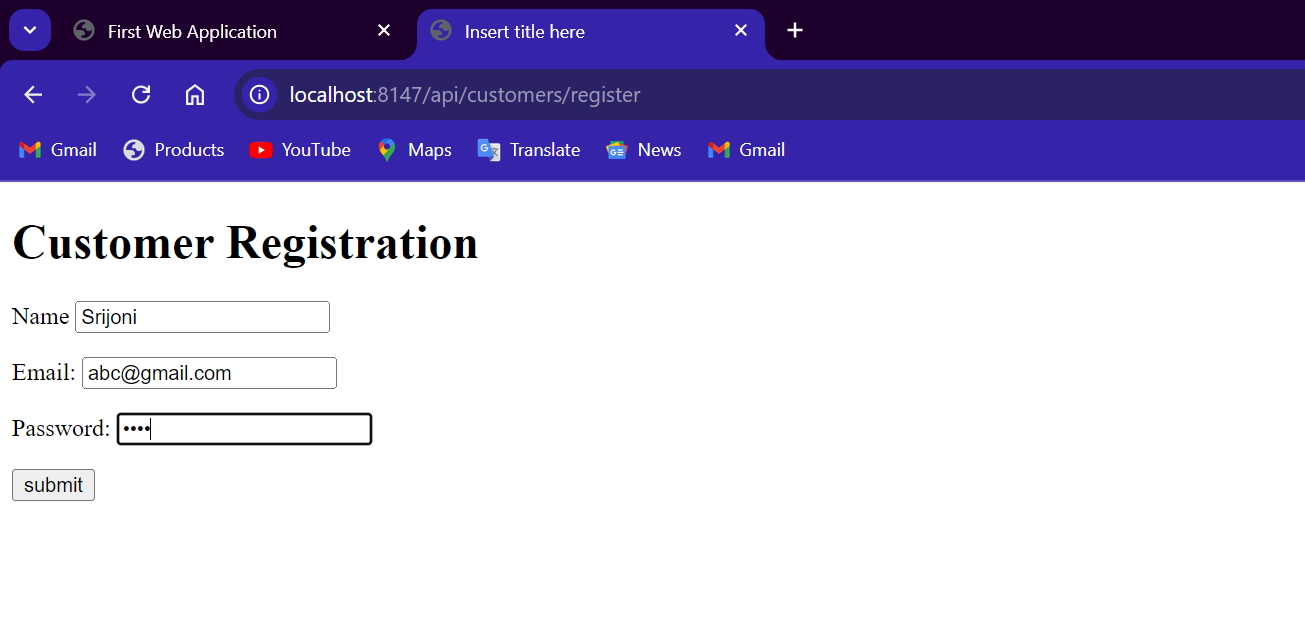
* **Purpose:** To process customer registration data submitted via an HTML form.
* **Implementation Details:**
  + **Endpoint:** /api/customers/save
  + **HTTP Method:** POST
  + **Expected Form Data:** Form fields representing name, emailid, and password.
  + **Functionality:**
    - The form data is bound to the Customer entity using the @ModelAttribute annotation.
    - The CustomerService saves the customer details to the database.
    - This method is typically used in scenarios where the user submits data via an HTML form on the frontend.
  + **Return Value:**
    - Returns the view name for the registration confirmation page (JSP page in this case).

**Testing :**

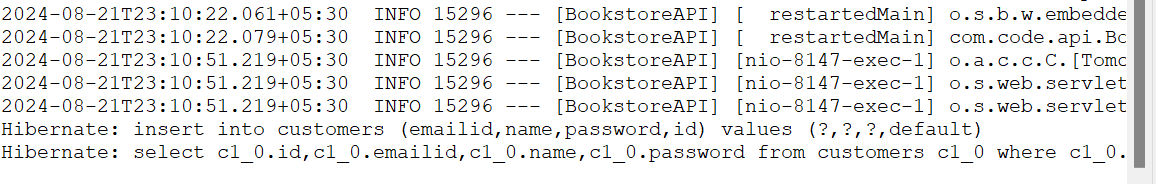
In chrome typed this URL to test the app: <http://localhost:8147/api/customers/test>



In chrome typed this URL to register new customer: <http://localhost:8147/api/customers/register>



After clicking on submit it gets save and in the IDE this output is displayed after the operations:



This documentation summarizes the approach taken to handle customer registrations via JSON request bodies and form data in the Online Bookstore project. The provided code file contains the necessary implementations to achieve these functionalities.