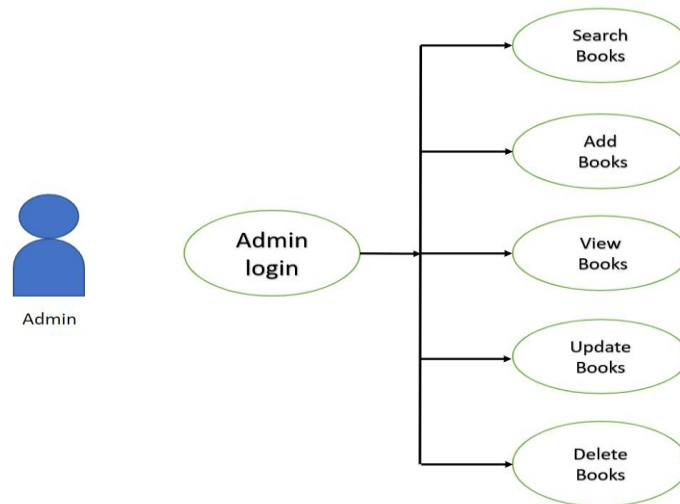


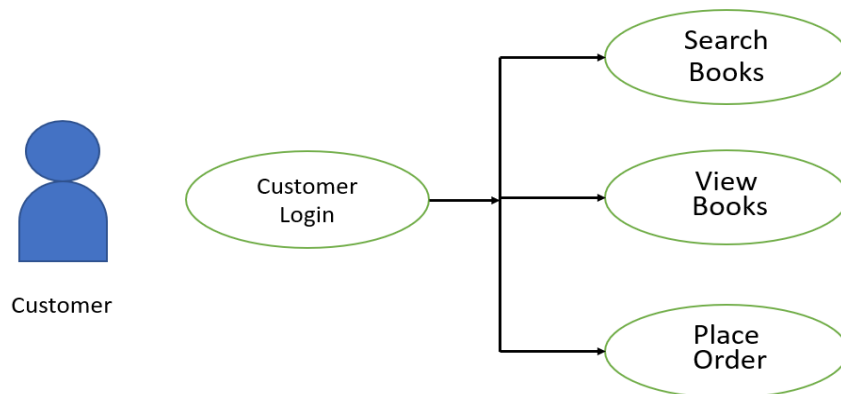
Online Book Store

- ❖ The Online Book Store System provides
Microservices Architecture based software solution.
- ❖ This Online Book store had two stack holders
 - Admin
 - Customer
- ❖ Here Different Microservices are combined together
to perform as a unified application.
- ❖ All the microservices are communicating with each
using Feign Client.

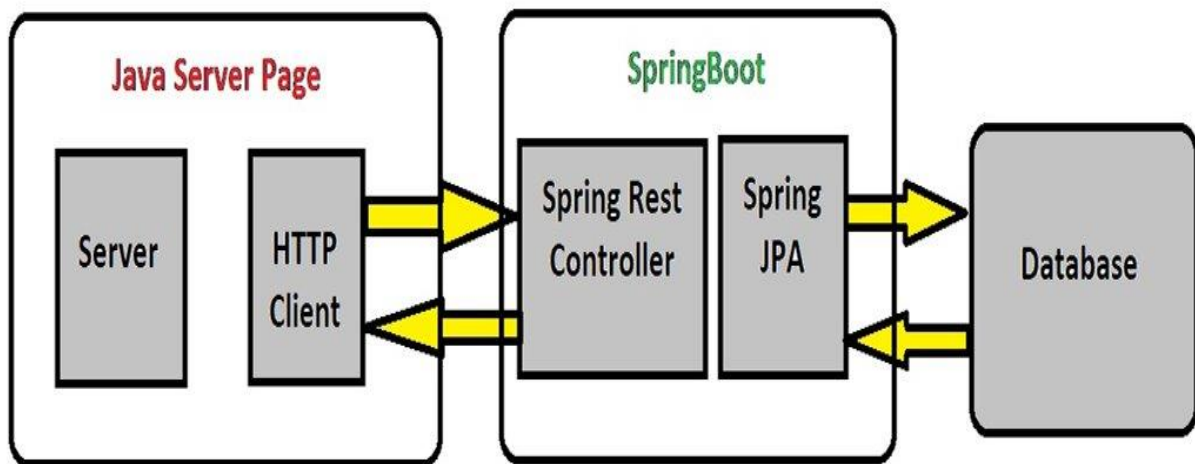
Architecture Diagram for Admin



Architecture Diagram for Customer



Client-Server Architecture



Technologies Used

- ❖ **Front End:** Java Server Page
- ❖ **Backend:** Java, **Spring Boot**, **Restful-Services**, **Hibernate**, **Spring-DataJpa**, **Ehcache**, **EurekaServer** , **JWT**, **Slf4j**.
- ❖ **Database:** MySQL - database
- ❖ **Tools:** **Spring Tool Suite**, **Swagger**, **Postman**.

Working Microservices

- ❖ Authorization microservice
- ❖ Inventory Microservice
- ❖ Order Microservice
- ❖ Netflix Eureka Server

Authorization Microservices

- ❖ Authentication Microservices will perform operations like:
 - Login
 - Logout
- ❖ Authentication will provide the JWT token after user's validation.
- ❖ Providing the authorization based on user's login.
- ❖ The microservice is running in port 8008 with servlet path “/authorization”.

Inventory Microservices

❖ Inventory Microservices will perform operations like:

- View Books
- Add Book
- Update Books
- Delete Books
- Search Book

❖ This microservices interacts with the database to store and manipulate the data.

❖ Here EhCaching is used at service layer.

❖ The microservice will run on port 8082 with servlet path “/inv-ms”.



/inv-ms/v3/api-docs

Explore

Servers

http://localhost:8082/inv-ms - Generated server url

book-controller

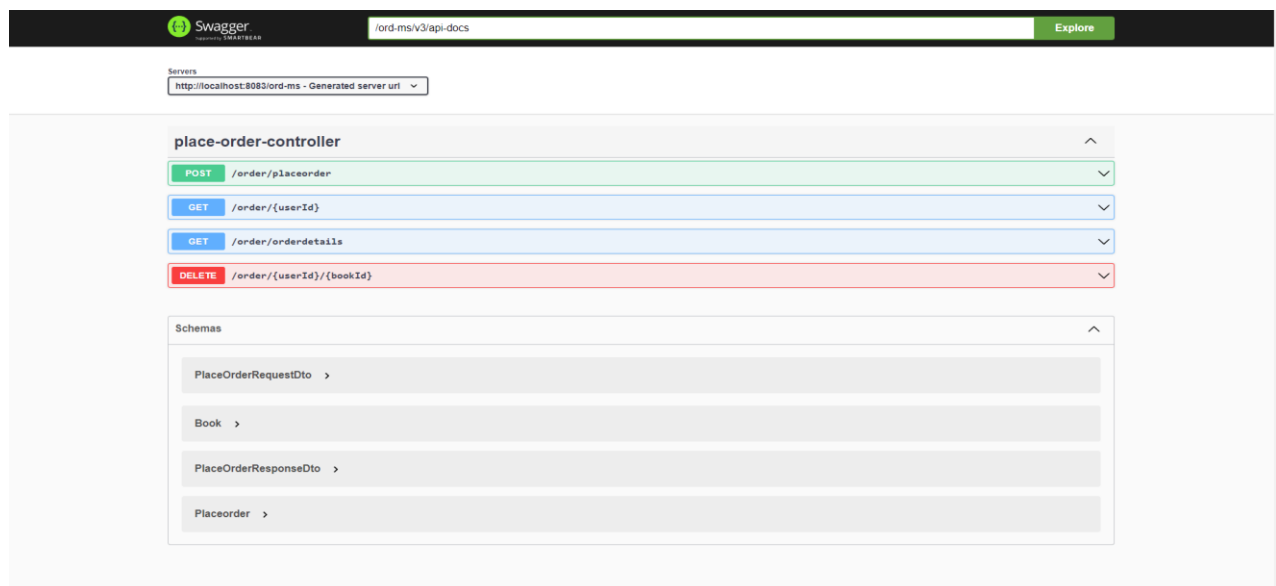
POST	/book/update	⌵
POST	/book/add	⌵
GET	/book	⌵
GET	/book/searchBookByName/{searchBook}	⌵
GET	/book/searchBookById/{bookId}	⌵
DELETE	/book/delete/{bookId}	⌵

Schemas

Book >


Order Microservices

- ❖ Order Microservices will perform operations like:
 - View Orders
 - Place Orders
- ❖ This microservice will run on port 8083 with servlet path “/ord-ms”.
- ❖ It interacts with Inventory Microservices for Placing Book Order using Feign Client.



Netflix Eureka Server

This Microservice acts as a service which registers all the microservices running on different ports.

HOME LAST 1000 SINCE STARTUP

System Status

Environment	N/A	Current time	2022-05-02T22:40:19 +0530
Data center	N/A	Uptime	00:31
		Lease expiration enabled	true
		Renews threshold	6
		Renews (last min)	8

DS Replicas

localhost

Instances currently registered with Eureka

Application	AMIs	Availability Zones	Status
AUTHORIZATION-SERVICE	n/a (1)	(1)	UP (1) - LTIN273761.cts.com:authorization-service:8008
INVENTORY-MS	n/a (1)	(1)	UP (1) - LTIN273761.cts.com:inventory-ms:8082
ORDER-MS	n/a (1)	(1)	UP (1) - LTIN273761.cts.com:order-ms:8083

General Info

Name	Value
total-avail-memory	128mb
num-of-cpus	8
current-memory-usage	55mb (42%)
server-up-time	00:31
registered-replicas	http://localhost:8761/eureka/
unavailable-replicas	http://localhost:8761/eureka/
available-replicas	

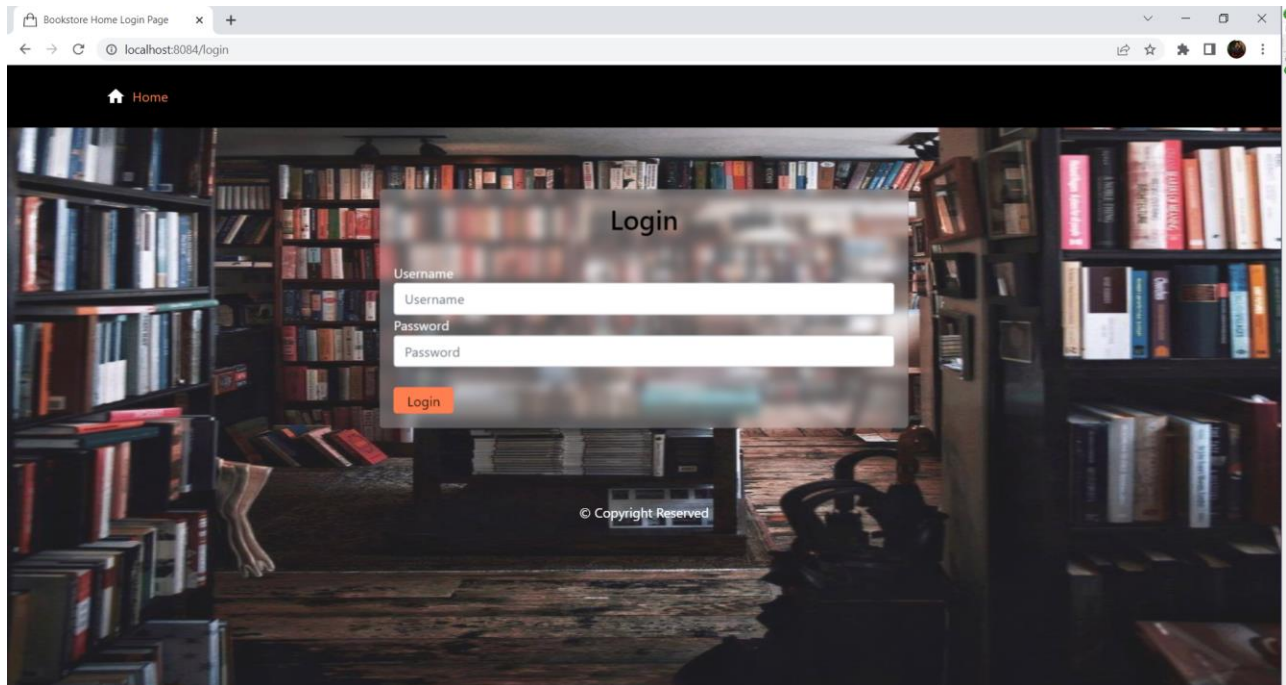
Instance Info

Client Book Portal

- ❖ The Book Portal acts as a client interface through which the user or admin can perform all operations.
- ❖ This Client interacts with Inventory, Order & Authorization Microservice are interacting with each other using Feign Client.
- ❖ This Client will run on port 8084

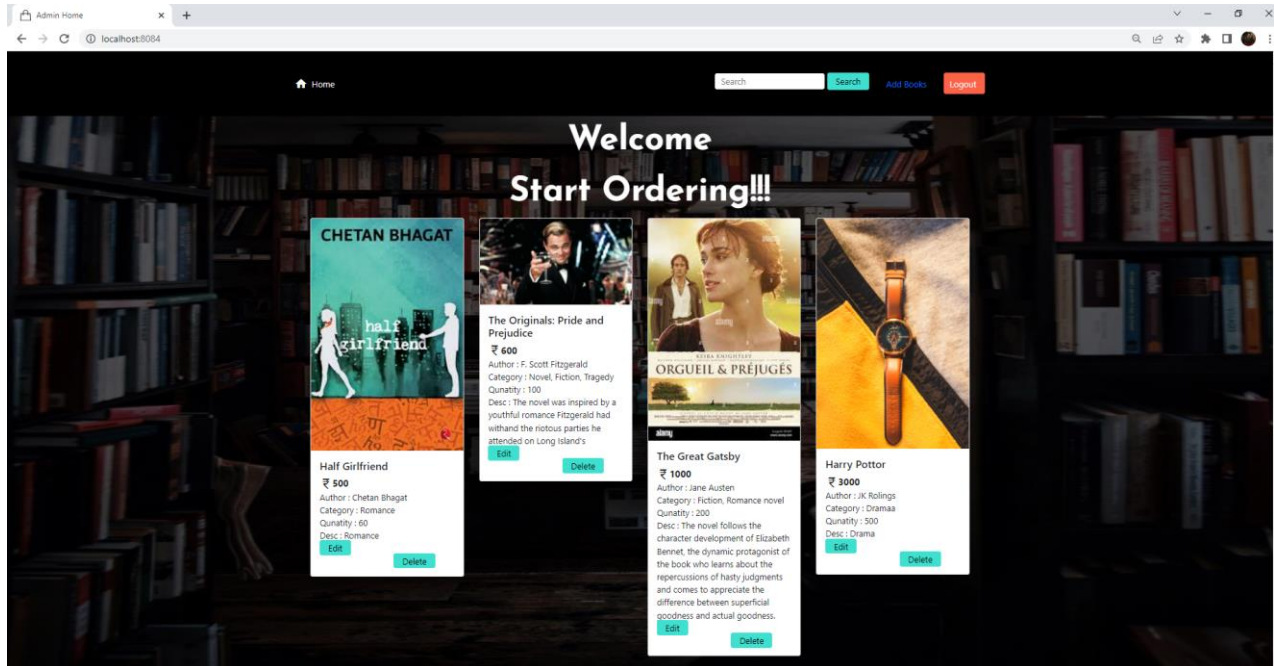
User Interface of Online BookStore Application

1.Login(<http://localhost:8084/login>)

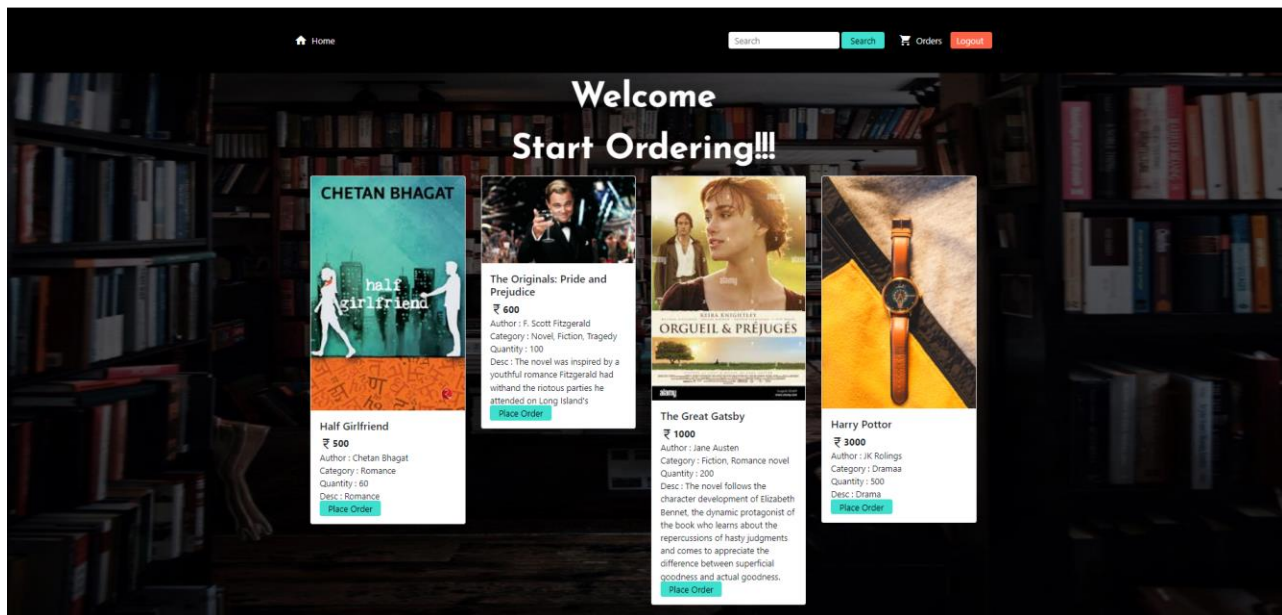


The user's name and password are input which is validating against the credentials present in the mysql database.

2. Admin Dashboard



3. Customer Dashboard



-----End-----

