RETURN ORDER MANAGEMENT SYSTEM

* Project Members
* Gaurav Raj
* Darsh Jain
* Shrey Choubey
* Ritansh Bangre
* Ashish Singh
* Project Overview
* A leading Supply chain Management Organization wants to automate the return orders, by classifying them to repair or replacement. Repair is for all main or integral part of their product. Replacement is for accessories.
* Project Details
* Microservices
  1. Authentication

The Authentication microservice is used to create JWT tokens. It used for authorization of other microservices.

* 1. Component Processing

The intent of this microservice is to determine the component processing detail. It interacts with packaging and delivery microservice to get the consolidated cost for the processing and expected date of delivery. It also interacts with payment microservice to process the payment from credit card.

* 1. Packaging and Delivery

The Packaging and Delivery microservice should get the component type and count to determine the packaging and delivery charge.

* 1. Payment

The Payment microservice gets the credit card details of the user and the total processing charge. Deducts the processing charge from the current amount and returns the balance amount.

* 1. Return Order Portal

The Return Order Portal local MVC app consists of all jsp pages which are connected to various other microservices.

* Port Number (Before deploying on Elastic Beanstalk)
  1. Authentication – 8081
  2. Packaging and Delivery – 8084
  3. Payment – 8085
  4. Component Processing – 8083
  5. Return Order Portal – 8082
* Port Number (After deploying on Elastic Beanstalk)
  1. Authentication – 5000
  2. Packaging and Delivery – 5000
  3. Payment – 5000
  4. Component Processing – 5000
  5. Return Order Portal – 8082
* Amazon Web Services(AWS) – Elastic Beanstalk
  + All the Microservices are deployed on AWS Elastic Beanstalk.
  + AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.
  + For more information about AWS Beanstalk, refer the following link :- <https://aws.amazon.com/elasticbeanstalk/>
* AWS Elastic Beanstalk deployed microservices links
  + Authentication Microservice :-

<http://orderjwtauthentication-env.eba-8y3qevhm.us-east-2.elasticbeanstalk.com/>

* + Packaging and Delivery Microservice :-

<http://orderpackaginganddelivery-env.eba-jwqfv6x4.us-east-2.elasticbeanstalk.com/>

* + Payment Microservice :-

http://orderpay-env.eba-k38wafyn.us-east-2.elasticbeanstalk.com/

* + Component Processing Microservice :-

<http://ordercomponentprocess-env.eba-dmzamjzm.us-east-2.elasticbeanstalk.com/>

* + Return Order Portal :-

<http://localhost:8082/>login

(These services must be consumed from an MVC app running in a local environment)

* Database
  + All the microservices are independently deployed.
  + An In-memory database (H2 database) has been used in the application.

1. **Components:**

Tables

1. PROCESS\_REQUEST
2. PROCESS\_RESPONSE
3. **Payments:**

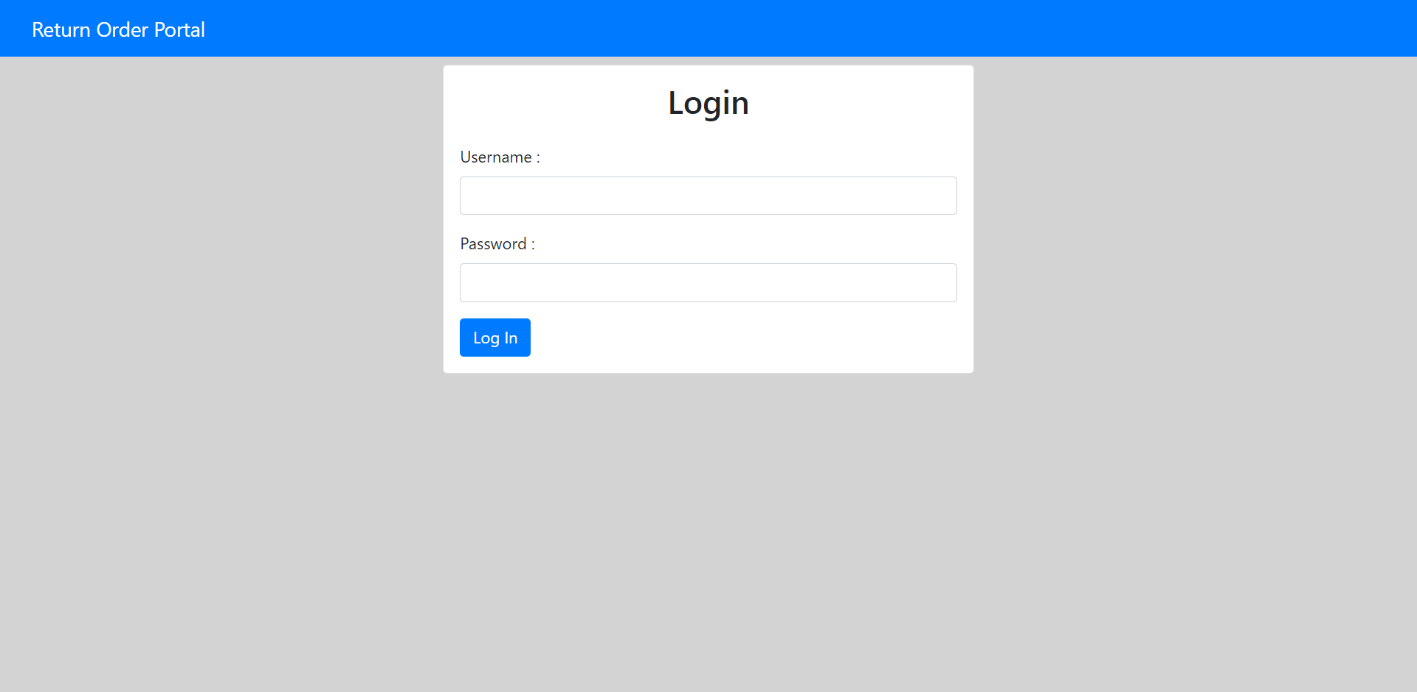
Table

1. CREDIT\_CARD\_DETAILS
2. **Jwt Authentication:**

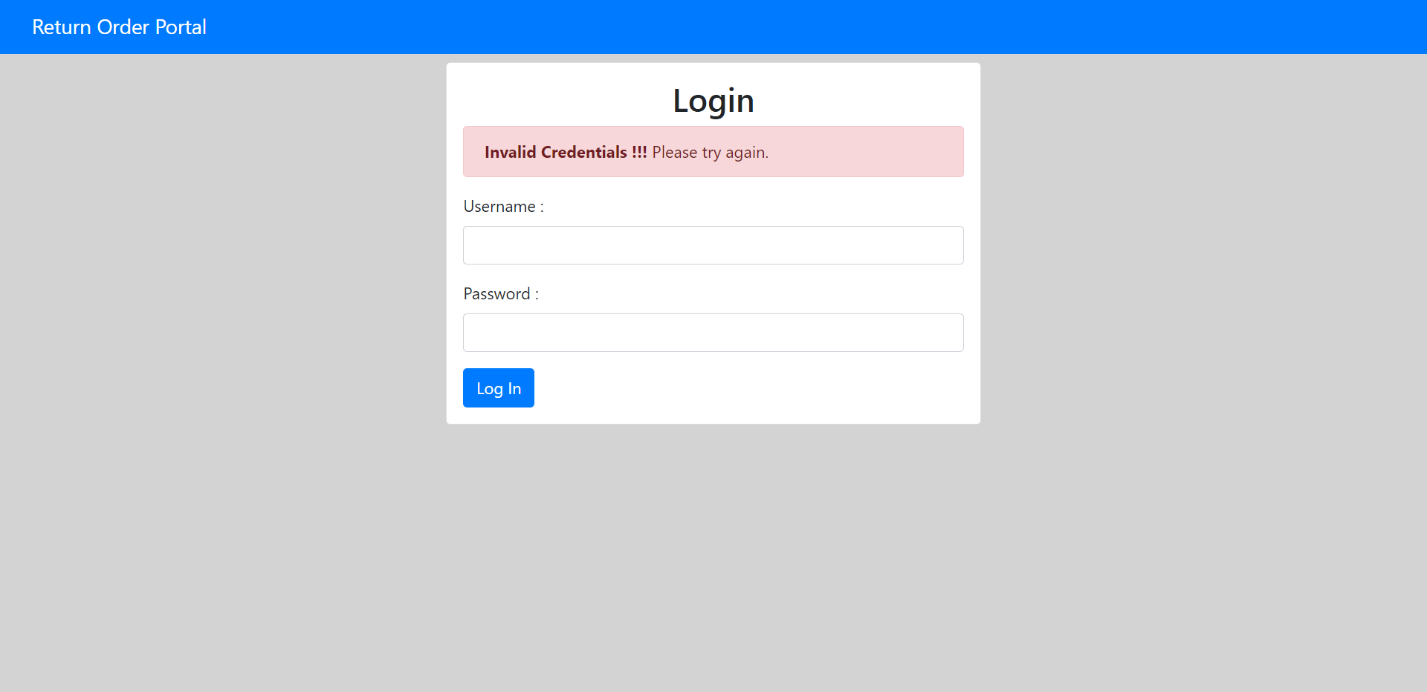
Table

1. USER

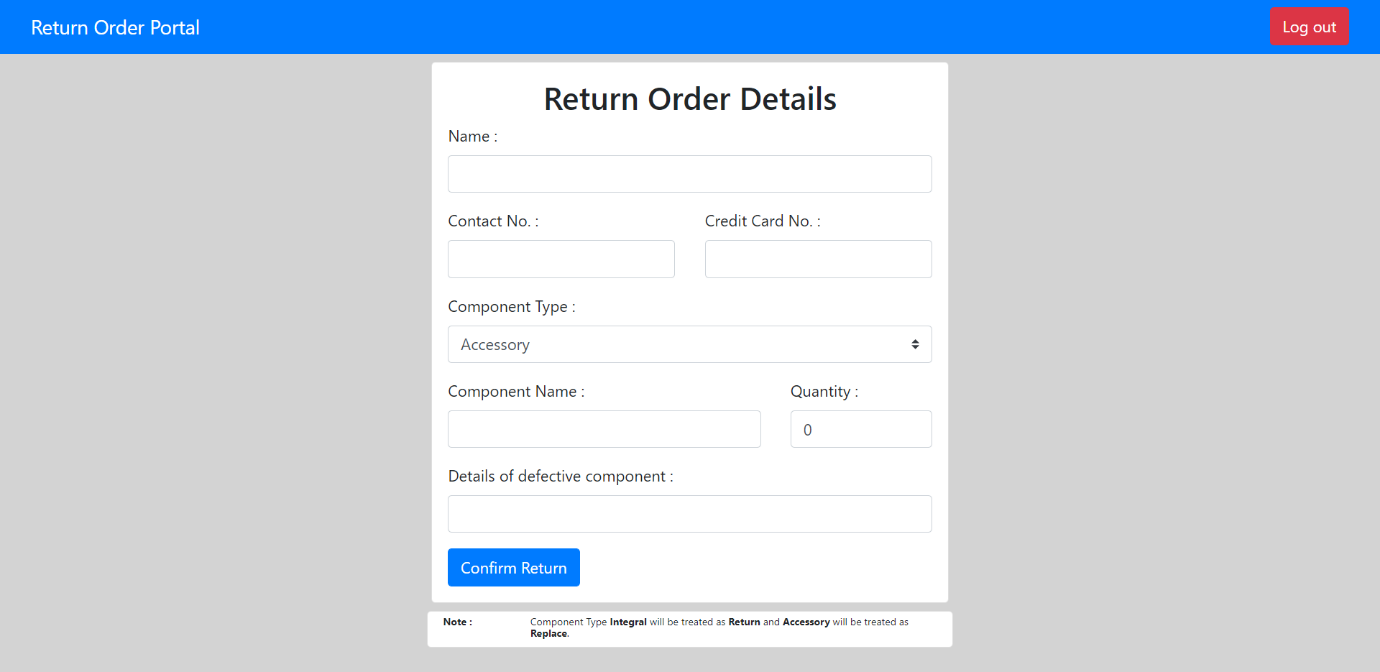
* PMD
* PMD is a source code analyzer. It finds common programming flaws like unused variables, empty catch blocks, unnecessary object creation, and so forth. It supports [Salesforce.com](http://Salesforce.com) Apex, Java, JavaScript, XML, XSL.
* Additionally it includes CPD, the copy-paste-detector. CPD finds duplicated code in Java, C, C++, C#, PHP, Ruby, Fortran, JavaScript, Matlab, Swift.
* Valid Credentials
  + Login Credentials (provided in USER table)
    - Username: Gaurav
    - Password: Gaurav123
    - Username: Darsh
    - Password: Darsh123
    - Username: Shrey
    - Password: Shrey123
    - Username: Ritansh
    - Password: Ritansh123
    - Username: Ashish
    - Password: Ashish123
  + Credit Card Number (provided in CREDIT\_CARD\_DETAILS table)
    - Card no: 123456789123
    - Card no: 987654321987
* Screen Layouts
  + - *Login Page*



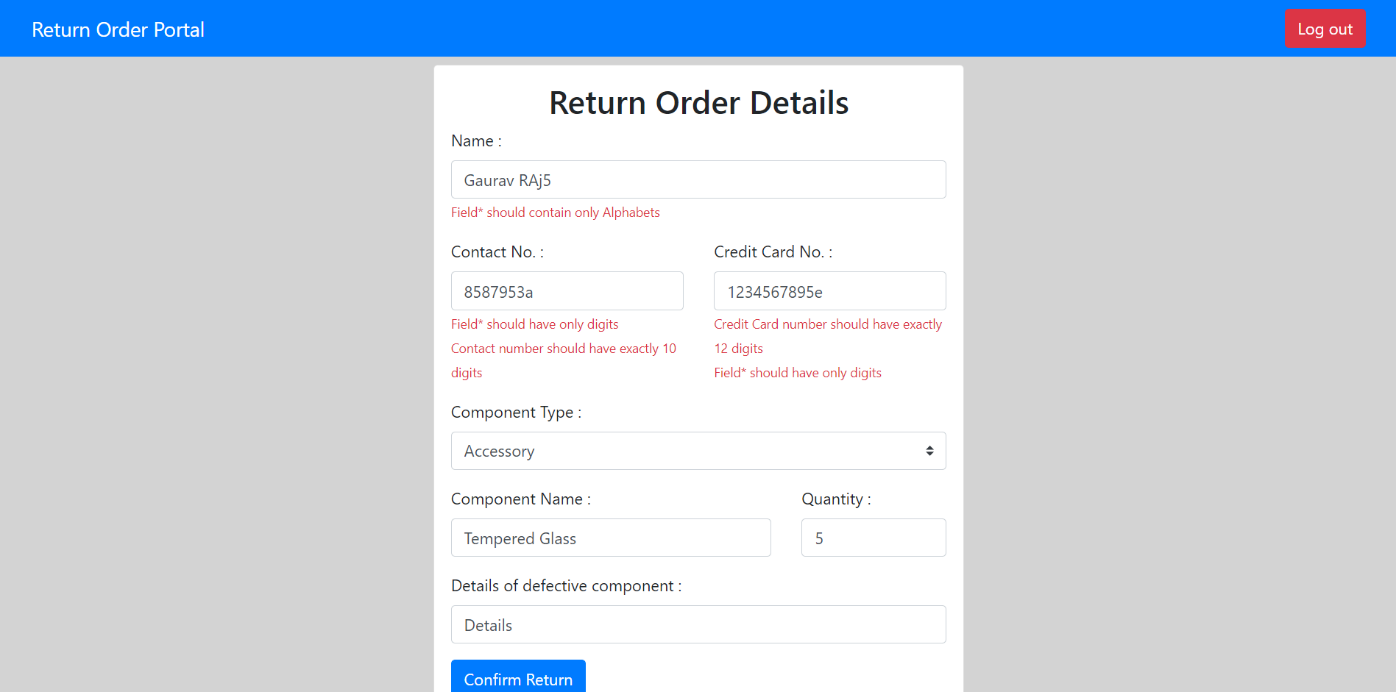
* + - *Login Page – Credential validation*



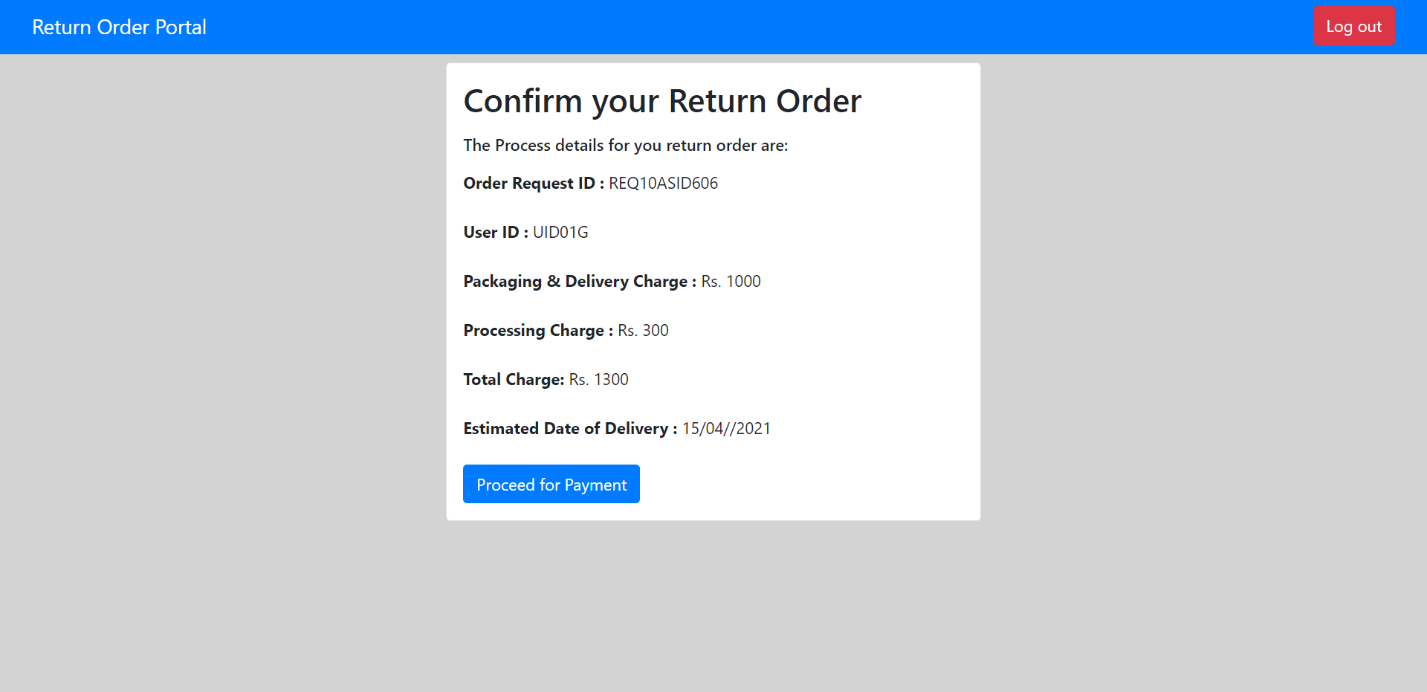
* + - *Home Page*



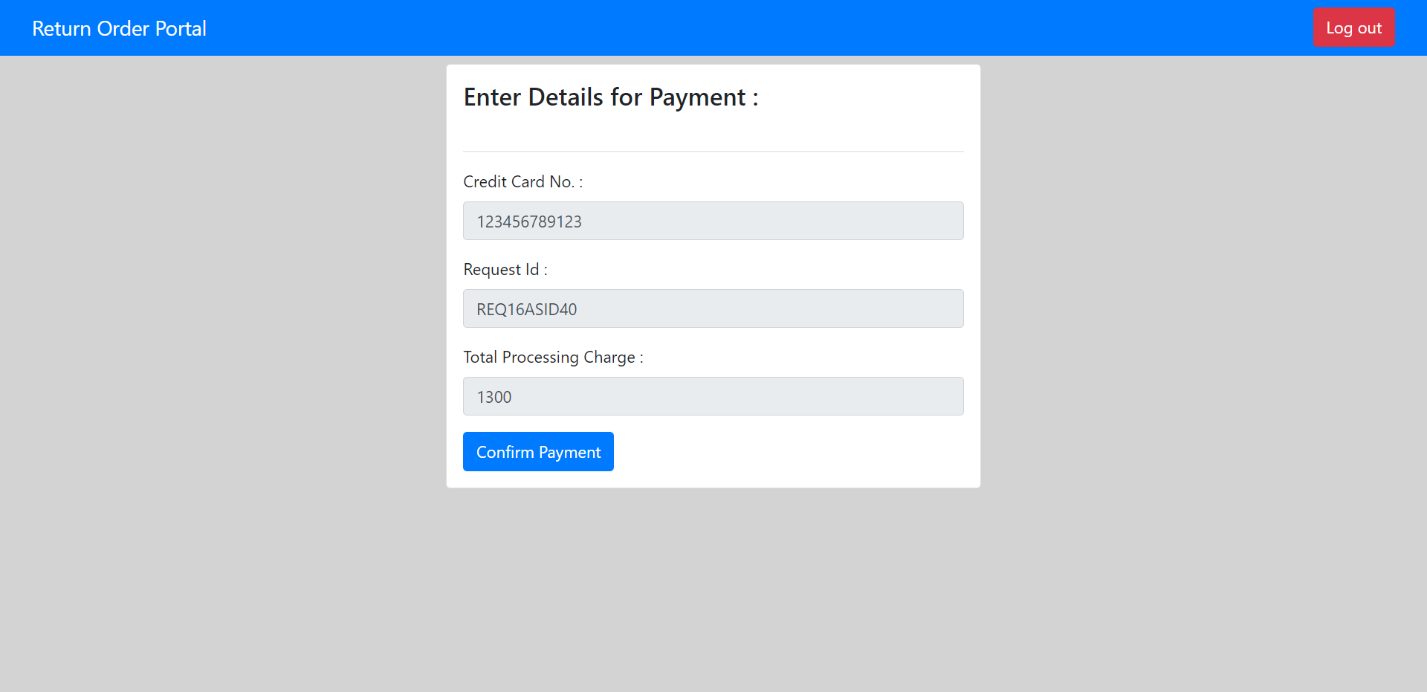
* + - *Home Page – Form Validation*



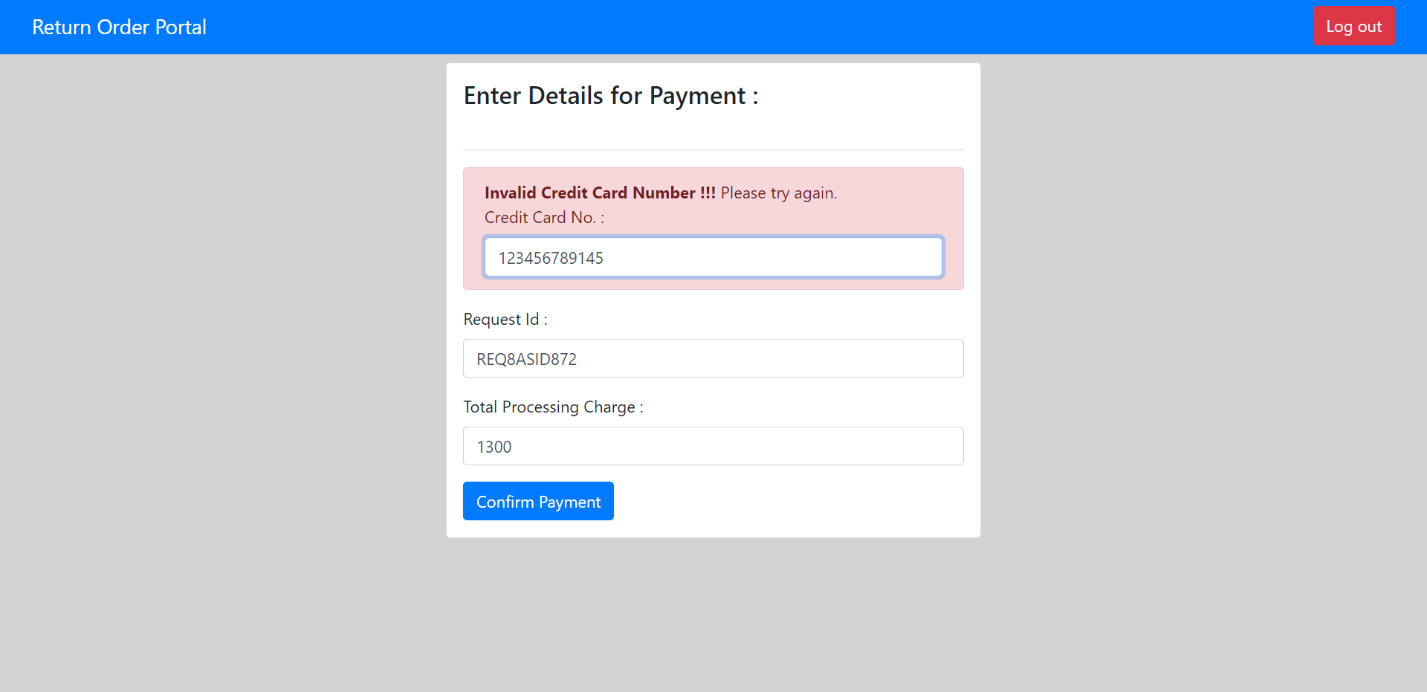
* + - *Process Order*



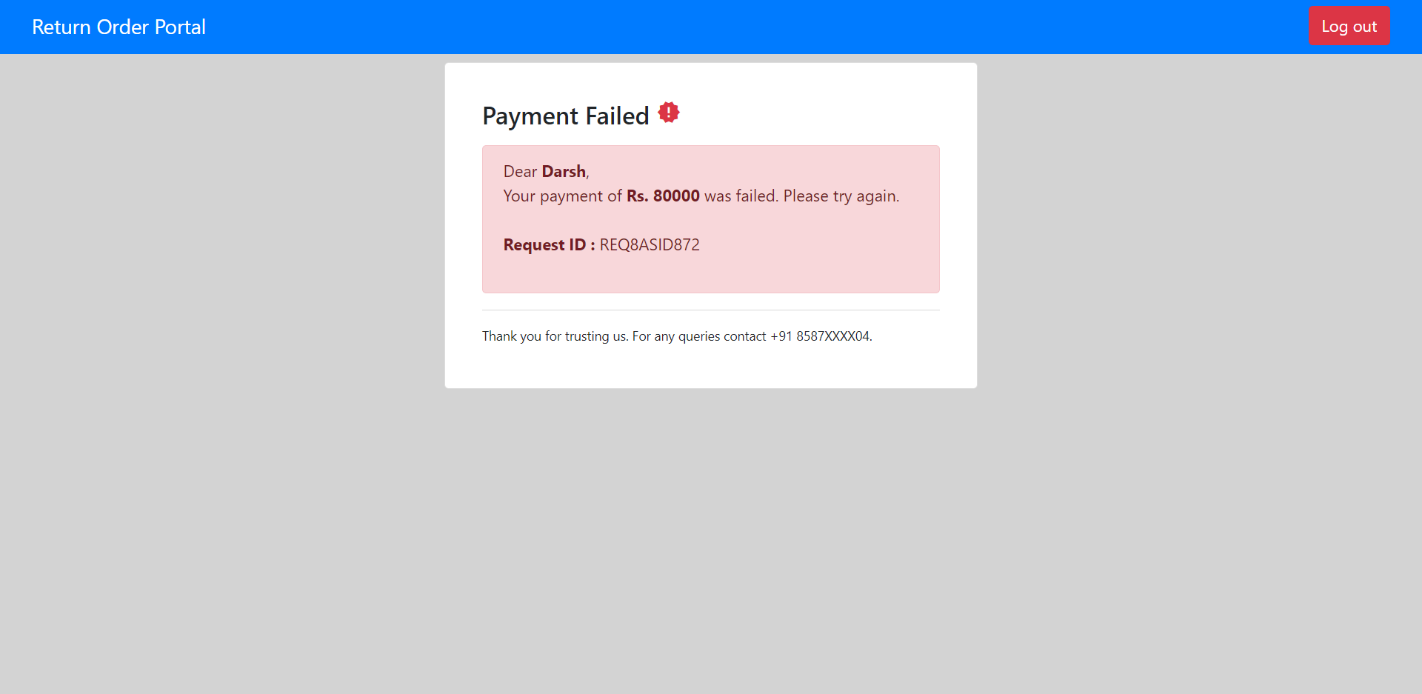
* + - *Process Payment*



* + - *Process Payment – Card Validation*



* + - *Process Payment – Failure*



* + - *Process Payment – Success*

