***RETURN ORDER MANAGEMENT SYSTEM***

***Project Members:***

* SAHITHYA PAPAREDDY (896850)
* ROHIT JAIN (896858)
* MANEESHA PUTLURU (896815)
* SHUVAM KUMAR DUTTA (897011)

***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:***

* ***Authorization-***
  + The Authorization microservice is used to create JWT tokens. It used for authorization of other microservices.
* ***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.
* ***Packaging and Delivery-***
  + The Packaging and Delivery microservice should get the component type and count to determine the packaging and delivery charge.
* ***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.
* ***Return Order Portal-***
  + The Return Order Portal microservice consists of all jsp pages which are connected to various other microservices.

***Port Number:***

* + Authorization – 8084
  + Packaging and Delivery – 8083
  + Payment – 8082
  + Component Processing – 8081
  + Return Order Portal – 8080

***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:***

* Component Processing Microservice

<http://component.eba-9mnvvicj.ap-south-1.elasticbeanstalk.com/>

* Packaging and Delivery Microservice

[http://packaging-delivery.eba-9mnvvicj.ap-south-1.elasticbeanstalk.com/](%20http://packaging-delivery.eba-9mnvvicj.ap-south-1.elasticbeanstalk.com/)

* Payment Microservice

<http://payment.eba-9mnvvicj.ap-south-1.elasticbeanstalk.com/>

* Authorization Microservice

<http://authentication.eba-9mnvvicj.ap-south-1.elasticbeanstalk.com/>

* Return Order Portal

<http://localhost:8080/>

NOTE:(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.

H2 Console links:

1. <http://localhost:8080/h2-console>

* Tables
* MYUSER
* PROCESS\_REQUEST
* PROCESS\_RESPONSE
* USER\_DETAILS

1. <http://localhost:8081/h2-console>

* Tables
* PAYMENT
* PROCESS\_REQUEST
* PROCESS\_RESPONSE

1. <http://localhost:8082/h2-console>

* Tables
* CREDIT\_CARD

***EclEmma Code Coverage:***

[EclEmma](http://www.eclemma.org/) is a great Java code coverage tool that has an Eclipse plugin.

It’s very simple and intuitive and has all you would expect from a code coverage tool. With it, you can:

* See code coverage for a java application that you’ve run (and potentially merge multiple run instances)
* See code coverage for JUnit tests and maven tests
* See which classes have which amount of coverage.