****

**Faculty of Contemporary Sciences and Technologies,**

**Skopje**

**Software Engineering**

**(Sem. 5, 2024/2025)**

Final Project:

***PressLogic, getting notifications before time runs out of your stock, is an easy way to manage stock in e-commerce platforms.***

**Student(s)**: **Mentor**:

1. 130690, Altin Ejupi Prof. Nuhi Besimi
2. 130903Rrezak Lamaxhema
3. 130340Ayşegül Havva Aksoy

January 2025, Skopje

Table of Contents

[Project Outline 3](#_Toc189087355)

[Overview 3](#_Toc189087356)

[Programming languages We will use 4](#_Toc189087357)

[Team Members 4](#_Toc189087358)

[Users story 5](#_Toc189087359)

[Class Diagram 5](#_Toc189087360)

[Use Case Diagram 6](#_Toc189087361)

[Software Architecture 7](#_Toc189087362)

[1. Front End: 8](#_Toc189087363)

[2. Processes: 8](#_Toc189087364)

[3. Back End: 8](#_Toc189087365)

[Unit Testing 9](#_Toc189087366)

[Student Work 12](#_Toc189087367)

# Project Outline

## Overview

Our project is about e-commerce platforms that will be used to simplify inventory management and stock management. It will help businesses keep track of their stock and automatically get email notifications when products run low on stock.

Our customers will be e-commerce businesses and platforms that need a better way to handle their inventory. Whether it’s an online store or a warehouse with thousands of products, this platform ensures they never run out of stock. The platform will have mostly those 2 primary functions:

* 1. **Automatically** sends an alerting **email** ofstock level when inventory is low, reducing manual work, of counting sales and inbounds.
  2. **Keeps all data in one place**, making it easy to track and manage inventory in real time.

The main **users** will be inventory managers and warehouse staff at e-commerce companies.

Since we're working on this project using an Agile methodology, we'll be adaptable and work as a team to deal with many changes. This is particularly crucial because inventory systems are frequently unexpected!   
 1. **Altin** will concentrate on database construction, backend development, and the fundamental logic of the system.  
 2. **Rezak and Ayşegül** will be in charge of frontend development, making sure that Javascript is implemented and the user interface is simple to use.   
  
There will be six primary sections to the platform:   
1. **The frontend experience** for users is known as the user interface (HTML, CSS).   
2. **Inventory Management**: SQL-based tools for monitoring and controlling stock   
3. **Automated emailing**: A system that uses Laravel and PHP to track inventory and place new orders as necessary   
4. **Database:** Backend inventory for every data

5. **Reporting** – Insights and analytics for users

We estimate the project will take about **81 hours** to finish, including development, testing, and documentation, normally the times tell the total from all of us.

## Programming languages We will use

1. Frontend (JavaScript, HTML, CSS).
2. Backend (PHP, Laravel)
3. Database (MySQL)

This project is just a small step to the beginning of more complex and helpful software! It’s designed as a stepping stone for more advanced systems we can build in the future, potentially helping more businesses solve inventory challenges.

## Team Members

1. Altin Ejupi (ae30690@seeu.edu.mk): Backend developer

2. Rrezak Lamaxhema (rl30903@seeu.edu.mk): Frontend developer

3. Ayşegül Havva Aksoy (aa30340@seeu.edu.mk): Frontend developer

# Users stories

An e-commerce businessman comes to us and tells us about the problem he has with stock and supply management when the demand increases. Those are his words: “ ***Hey PressLogic, I have huge problems with my supply chain and my inventory management, when I have products on my website near the end, or the last 2 ones left it does not show on a website that is low of stock or that is out of stock when I receive a lot of orders some of them get dialed. The reason they get delayed is that most of my products come from China and a customer will have an extra 7 days at least before our new stock arrives. Because of those delays and not good management of inventory, we are getting complaints daily and canceled orders because of this which affects overall sales and company profile. We want to have a solution for this and a faster one so our problem is solved once and forever, I would like to have a notification system when my products on the website are being sold and they are near to sold out I would like to have notification how much I have left, so I can order before they sold from China this way I think I will not have a delay. Hope I am clear and let me know how much it will cost and how fast you guys can make it***”.

We just replied to our client: “ ***Thank you for reaching us, we will make the best solution for you***”.

# Class Diagram

The class diagram below represents the system’s structure for managing inventory and sending notifications.

* **User Class**: Handles user details (userID, email) with methods like login().
* **Product Class**: Manages product data (productID, quantity, threshold) and checks stock levels.
* **Order Class**: Tracks orders (orderID, productID) with methods like placeOrder().

It illustrates how the system's components interact to streamline inventory management.

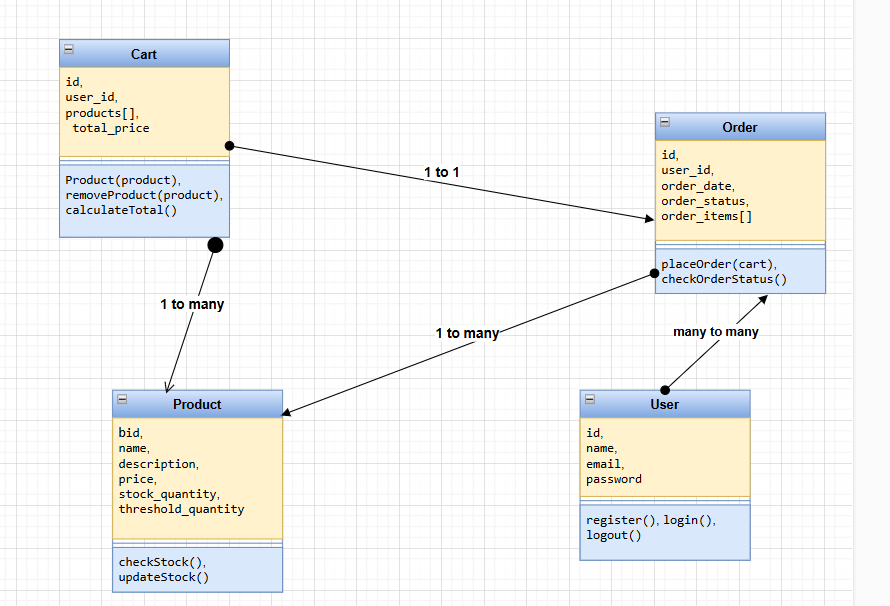


Figure 1-Class diagram

# Use Case Diagram

This use case diagram represents the process of managing product selection and stock availability in an e-commerce system. The user or owner interacts with the system to perform three main actions: selecting a product, selecting its quantity, and adding it to the cart.

1. **Product Out of Stock**:
   * If the selected product's quantity exceeds the available stock, the system generates an **email notification** to inform the user about the unavailability.
2. **Product in Stock**:
   * If the selected product's quantity is available, the item is successfully added to the **shopping cart**, allowing the user to proceed with the purchase.

This diagram highlights the system's capability to manage stock effectively while ensuring user notifications for a smooth shopping experience.

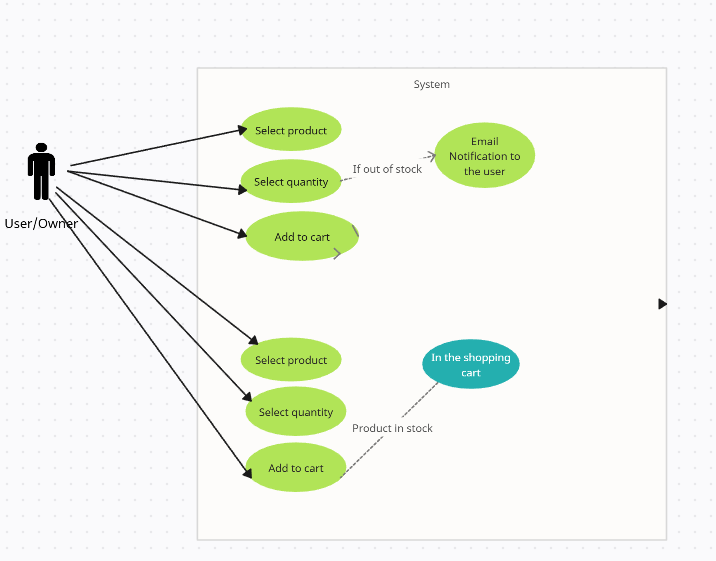


Figure 2 Use Case Diagram

# Software Architecture

This software architecture diagram illustrates the workflow of an inventory and order system.

1. Front End: The customer selects products, places them in a cart, and initiates orders. The business owner receives notifications via email, keeping them informed of stock updates.
2. Processes: Products are temporarily saved locally in the cart. Once the order is placed, the details are saved to the orders database. Simultaneously, the PHP Mailer sends an email notification to the business owner for any stock thresholds reached.
3. Back End: The backend handles stock updates and checks. If the stock level falls below a predefined threshold (e.g., 20 items), a notification is pushed. The system updates the database by deducting the purchased quantity, while Java code manages these backend operations.

This architecture ensures efficient order processing, inventory management, and proactive notifications.

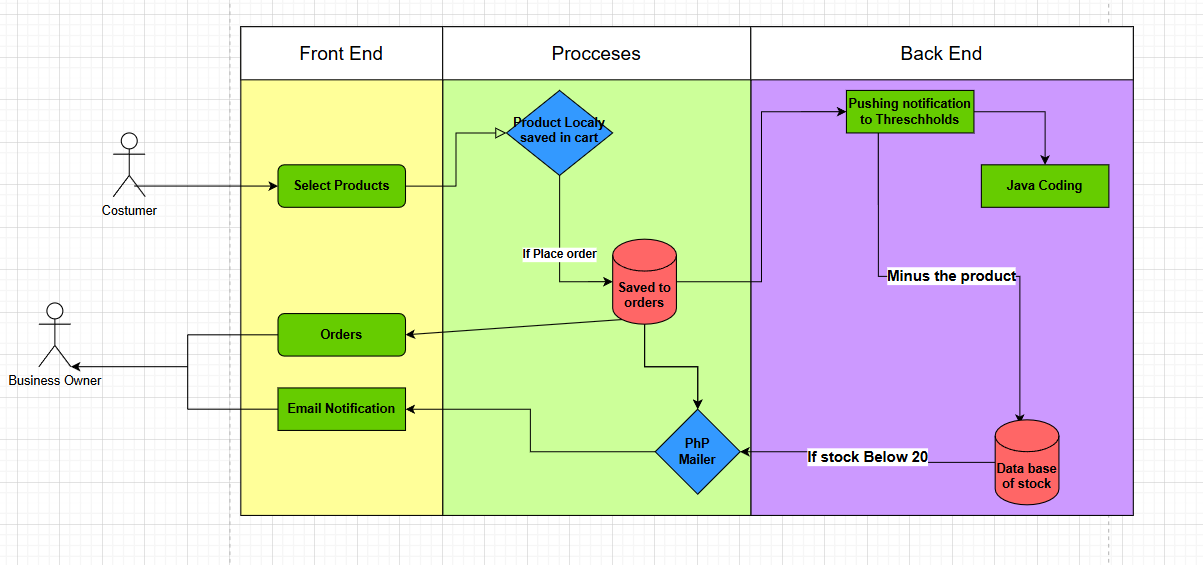


Figure 3-Software Architecture

# Unit Testing

For this project, we did a lot of testing and how the system would react in different situations, some of them not testing that the system works perfectly in every situation will be shown below:

* 1. First Testing is when the threshold is triggered by only one product, we have 26 Cheetos, we will buy 6 and the threshold is 20, we will see if the system will work and notify us.

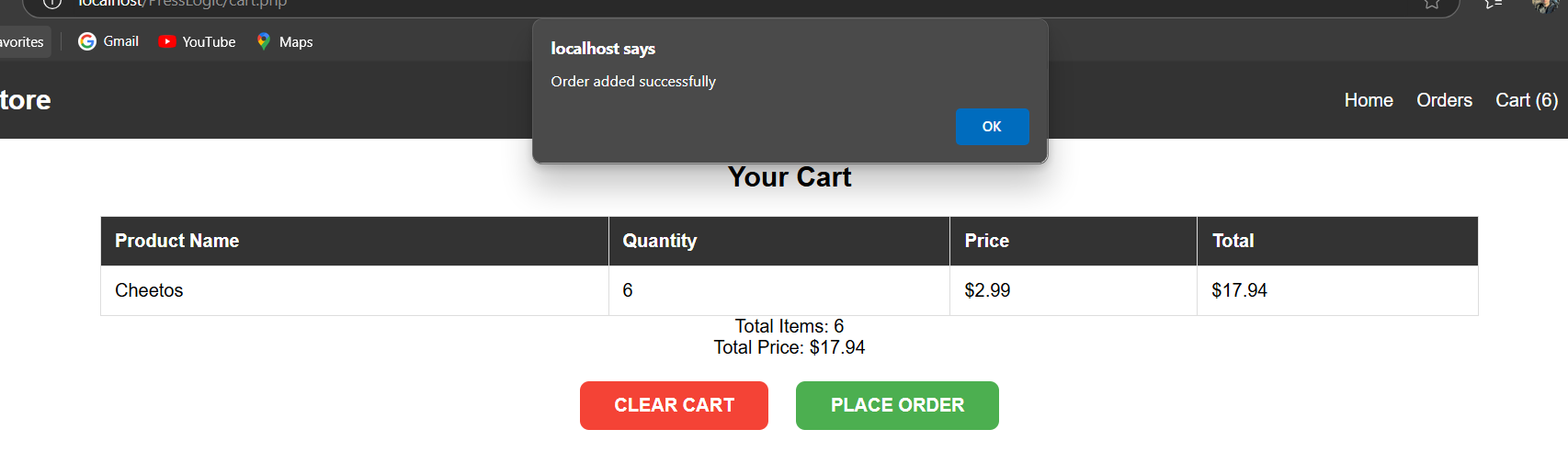
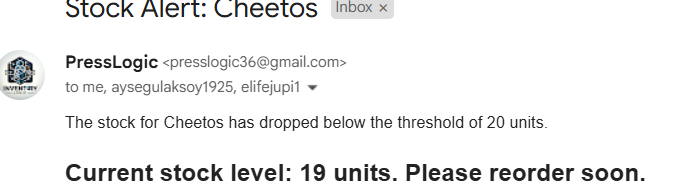


Figure 4-Email after stock dropped below 20

Figure 5-Number of cheetos in tock (DB)

Figure 6-Order placed

* 1. When 3 products trigger the threshold, we will see if the system push the email for 3 of the products.

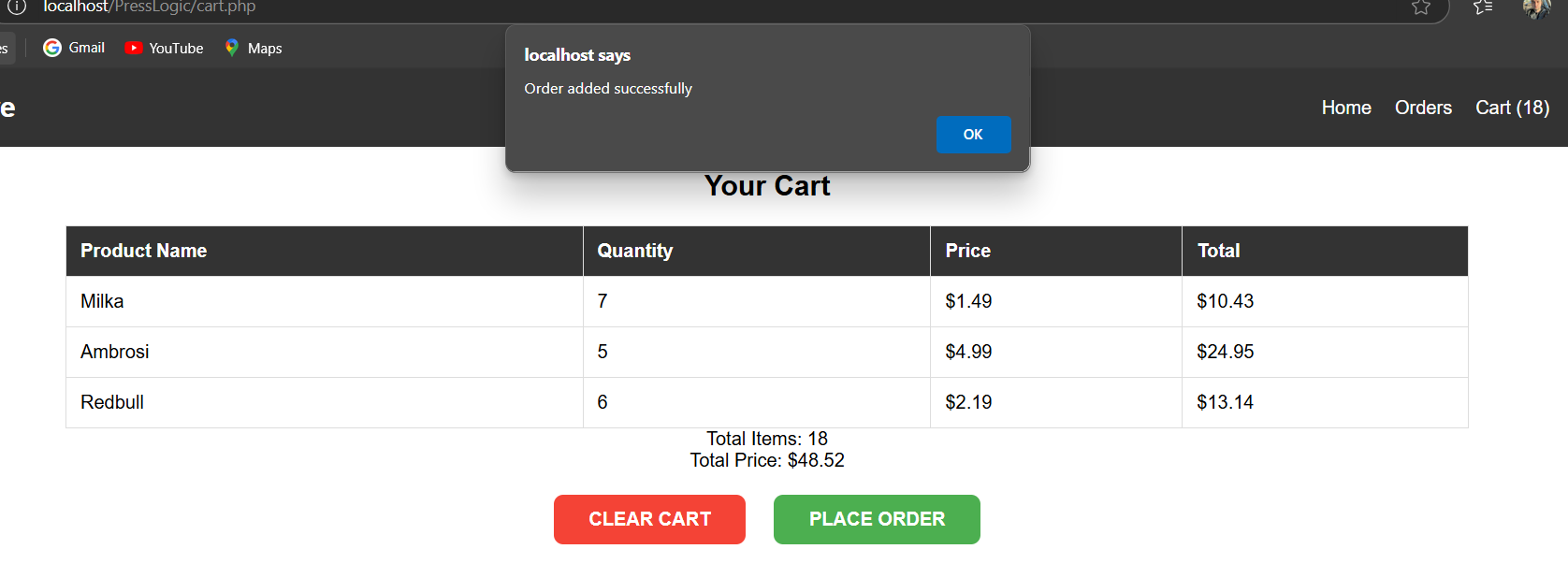
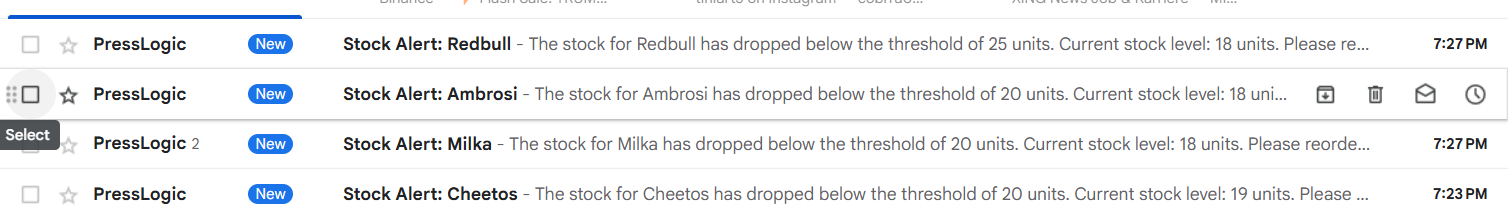


Figure 7-Email of all trggered products come

Figure 8-Order added with sucess

Figure 9-pushing 3 products to trigger the system

3. Other unit testing is what will happen when we have one product which already passed the threshold level and also is close to 0 , if a user will place such a order, how will that be translated in in databases.



Figure 10-When stock is 2

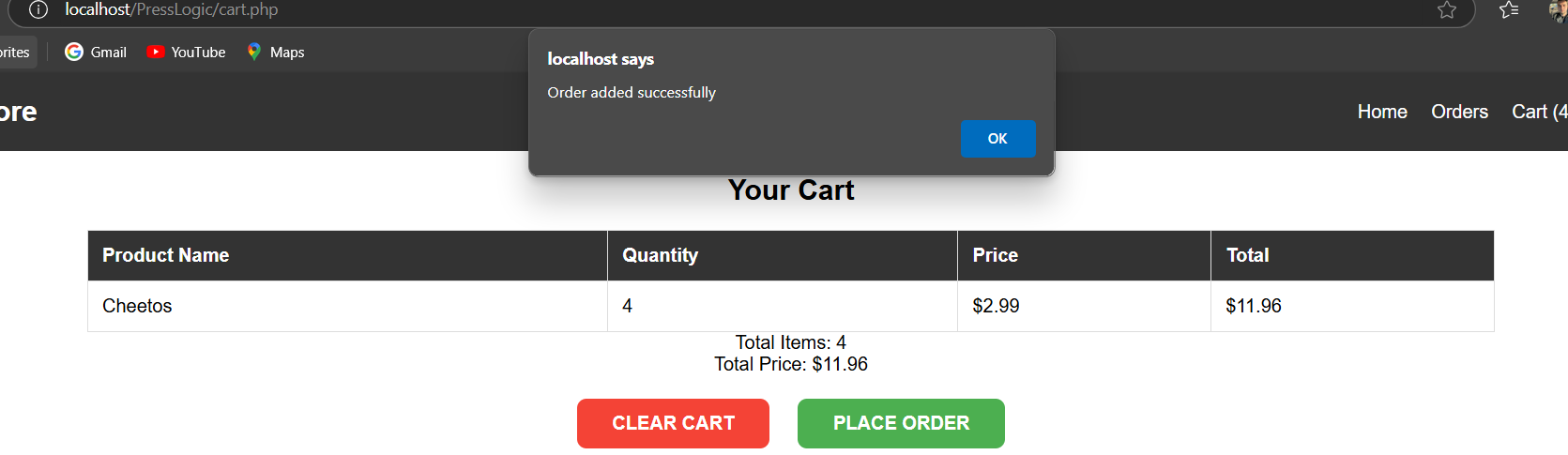


Figure 11-order placed with success

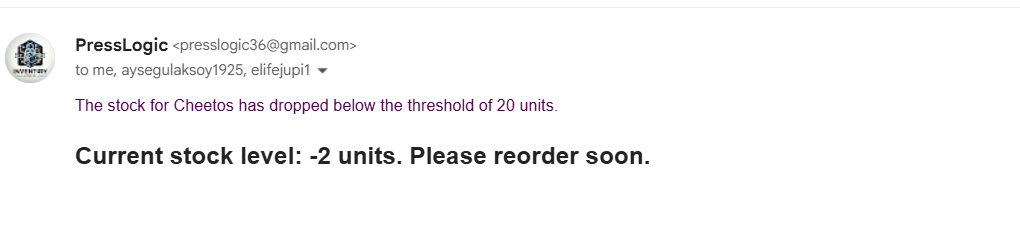


Figure 12-when stock hits below 0 the order is made but level is in -2

4. We will see now how the system will react when the stock lands exactly at threshold number 20, when it does not pass, will the trigger push the email system?



Figure 13-stock is 21

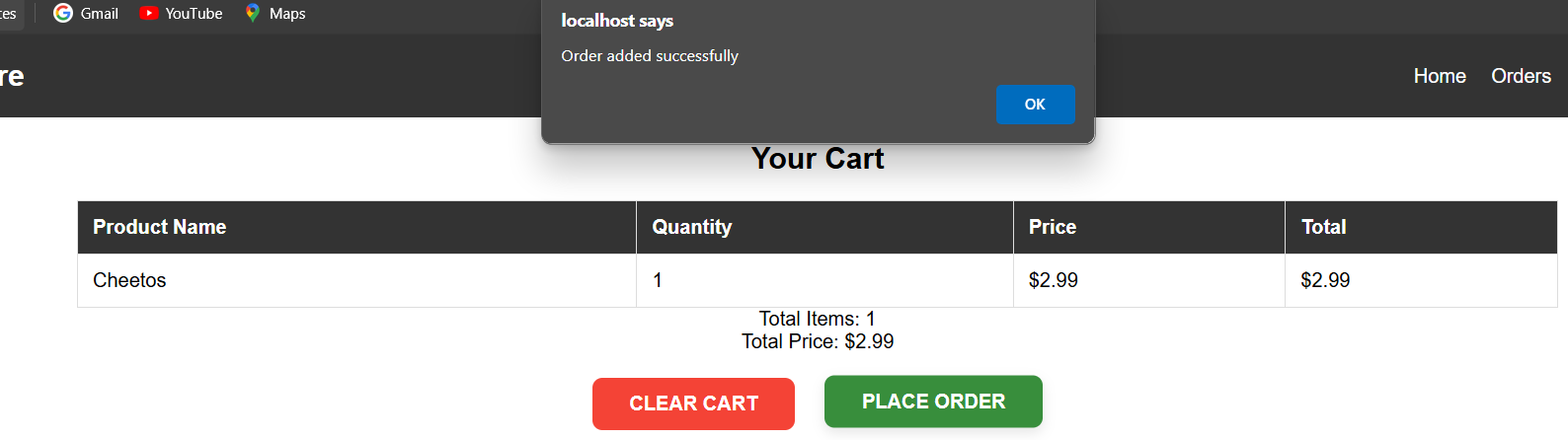


Figure 14-the order is placed

When the trigger or the stock is at breakpoint, we do not get an email therefore only if the threshold is passed or below we got the alerting email.

In this project, we conducted several unit tests to ensure the system behaves as expected in various scenarios:

1. **Single Product Threshold Trigger**: Tested if the system correctly sends a notification when a single product's stock drops below the threshold.
2. **Multiple Products Threshold Trigger**: Verified that the system can handle multiple products simultaneously triggering the threshold and sending notifications for each.
3. **Critical Low Stock Scenario**: Examined how the system handles orders for products already below the threshold and close to zero, ensuring proper database updates.
4. **Exact Threshold Edge Case**: Checked if the system refrains from sending a notification when the stock level matches, but does not go below, the threshold.

These tests confirm the system reliably sends notifications only when stock drops below the defined threshold, ensuring efficient and accurate alerting.

# Student Work

1. 130903Rrezak Lamaxhema- Did the frontend and basic part of the project like files (index.php, logic.php, orders.php) which was the first skeleton for this project.

2. 130340Ayşegül Havva Aksoy – Did the frontend part and also the design for all elements and photos on it. Besides this took care of creating the local storage of products ( synic\_cart, cart.php. ee.css, logic.css).

3. 130690, Altin Ejupi-Did the logical part of the project, the JavaScript code, and the Database part, also synchronizing the data with php connection with html. ( Logic.jss, Press.sql, Documentation of the project)