

Contents

1. PROJECT OVERVIEW	4
2. OBJECTIVES	4
3. TECHNOLOGIES USED	4
4. SYSTEM ARCHITECTURE.....	5
4.1 Index.php.....	5
4.2 CartService.php	5
4.3 db.php.....	6
4.4 MySQL.....	6
5. FILE RESPONSIBILITIES	6
6. API ENDPOINTS	6
7. SAMPLE PAYLOADS.....	7
8. DATABASE SCHEMA.....	7
9. TESTING AND VALIDATION.....	7
10. CONCLUSION	8

Figures and Tables:

Figure1: System Architecture.....	5
Figure2: Wampserver	8
Figure3: Wampserver Configuration.....	9
Figure4: PHP MyAdmin Configuration	9
Figure5: Environment Variable Setting	10
Figure6: Access Local Files	10
Figure7: User Interface Add Cart.....	11
Figure8: User Interface Update Cart	11
Figure9: Add Cart Item Operation UI	12
Figure10: Operation Perform as Microservice	12
Figure11: User Interface Remove Cart	13
Figure12: Update Cart Item Operation UI	13
Figure13: Operation Perform as Microservice	14
Figure14: Remove Cart Item Operation UI.....	14
Figure15: Operation Perform as Microservice	15
Table1: Technologies Used	4
Table2: API Used	6
Table3: Testing Parameters	8

1. Project Overview

The **CartService Microservice** is a lightweight, modular PHP-based system designed to manage fundamental operations in an e-commerce shopping cart, specifically focusing on insert, update, and delete functionalities. Developed using core PHP with a clean separation of concerns, the service is structured into three main components: `index.php` (which acts as the API router), `CartService.php` (containing the business logic), and `db.php` (handling the database connection). This microservice architecture allows for easy integration, reusability, and scalability within larger applications. It communicates with a MySQL database to perform transactional operations efficiently, ensuring that product details such as name, quantity, and price are managed in real-time. By exposing simple RESTful API endpoints, the service allows external clients or frontend interfaces to interact with the cart system seamlessly. The service is ideal for use in small to medium-scale applications where clean design, modularity, and ease of deployment are prioritized. It can also be extended with additional features like authentication, input validation, and error handling for more complex use cases. Overall, the CartService project provides a foundational microservice model for managing shopping cart data in a structured and maintainable manner.

2. Objectives

The primary objective of this project is to design and implement a simple yet effective microservice architecture for managing shopping cart operations using PHP. The service focuses on the three core CRUD operations: insert, update, and delete, allowing seamless management of cart items in a database. Specifically, the goals of the project include:

- To **build a modular microservice** structure using PHP with a clear separation between API routing, service logic, and database access.
- To **implement RESTful API endpoints** that support inserting new products into the cart, updating existing items, and removing products efficiently.
- To **establish a secure and reusable database connection** using MySQL for reliable data transactions.
- To ensure **code readability, maintainability, and extensibility**, making it easy to add features such as item retrieval, authentication, and logging in the future.
- To **test the microservice using tools like Postman** and validate that it correctly interacts with the database and returns appropriate responses.
- To demonstrate how small-scale microservices can be effectively integrated into larger systems or used as stand-alone components in web applications.

3. Technologies Used

The CartService microservice leverages a set of widely-used, lightweight technologies to deliver efficient, maintainable, and scalable backend functionality:

Table 1 Technologies Used

Component	Technology	Purpose
-----------	------------	---------

Component	Technology	Purpose
Backend Language	PHP (Core PHP 7+)	Used for implementing API endpoints, service logic, and database interaction
Database	MySQL	Stores cart data such as product name, quantity, and price
API Protocol	REST (HTTP + JSON)	Facilitates client-server communication through simple API calls
Database Driver	PDO (in db.php)	Connects PHP to MySQL database securely and efficiently
Testing Tool	Postman / cURL	Used for testing insert, update, and delete operations via HTTP requests
Server Environment	Apache (via XAMPP/LAMP/WAMP)	Runs the PHP microservice on a local or hosted server
Operating System	Cross-platform (Windows)	Compatible with all platforms that support PHP and MySQL

4. System Architecture

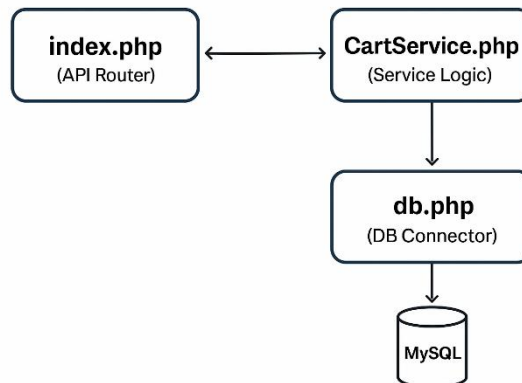


Figure 1 System Architecture

The system architecture you see illustrates the interaction between three core components of your PHP-based microservice for handling cart operations (Insert, Update, Delete):

4.1. index.php (API Router)

- **Role:** This acts as the main entry point for all HTTP requests.
- **Function:** It captures the `action` parameter (`insert`, `update`, `delete`) from the request and forwards it to the appropriate method in `CartService.php`.
- **Direction:** Communicates directly with the service layer (`CartService.php`).

4.2. CartService.php (Service Logic)

- **Role:** This is the **business logic layer**.
- **Function:** Based on the action received from `index.php`, it performs:
 - `addItem()` → insert operation
 - `updateItem()` → update operation
 - `deleteItem()` → delete operation
- **Direction:** It interacts with `db.php` to execute SQL commands.

4.3. `db.php` (Database Connector)

- **Role:** This is the **database abstraction layer**.
- **Function:** It initializes and returns a **MySQL database connection** using PDO.
- **Direction:** Used by `CartService.php` to execute SQL queries.

4.4. MySQL (Database)

- **Role:** Stores the cart data.
- **Function:** The actual records for product name, quantity, and price are stored here. All data modification operations eventually reach this database through `db.php`.

5. File Responsibilities

- **index.php:** Acts as the entry point for HTTP requests. It receives action requests (`insert`, `update`, `delete`) and forwards them to `CartService`.
- **CartService.php:** Contains core business logic. Each function (`addItem()`, `updateItem()`, `deleteItem()`) interacts with the database accordingly.
- **db.php:** Establishes a MySQL connection and returns the database handle to be used by `CartService`.

6. API Endpoints

The `CartService` microservice exposes three primary RESTful API endpoints to perform **Insert**, **Update**, and **Delete** operations on a shopping cart. These endpoints are accessed via the `index.php` router using an `action` query parameter. All operations use the **POST** method and expect input data as **JSON**.

Table 2 API USED

Operation	Method	Endpoint	Parameters
Insert	POST	<code>/index.php?action=insert</code>	<code>product_name, quantity, price</code>
Update	POST	<code>/index.php?action=update</code>	<code>id, quantity, price</code>
Delete	POST	<code>/index.php?action=delete</code>	<code>id</code>

7. Sample Payloads

Insert

```
json
CopyEdit
POST /index.php?action=insert
{
  "product_name": "Laptop",
  "quantity": 2,
  "price": 1500
}
```

Update

```
json
CopyEdit
POST /index.php?action=update
{
  "id": 3,
  "quantity": 1,
  "price": 1400
}
```

Delete

```
json
CopyEdit
POST /index.php?action=delete
{
  "id": 3
}
```

8. Database Schema

The database for this microservice contains a single table named `cart`, which is designed to store product information related to user cart operations. This schema supports basic insert, update, and delete actions efficiently.

```
CREATE TABLE cart (
  id INT AUTO_INCREMENT PRIMARY KEY,
  product_name VARCHAR(255),
  quantity INT,
  price DECIMAL(10, 2)
);
```

9. Testing and Validation

To ensure the reliability and correctness of the `CartService` microservice, a series of manual tests were performed using tools like **Postman**. These tests validated each API endpoint (Insert, Update, Delete) against expected behavior, input constraints, and error conditions.

Table 3 Testing Parameters

Test Case	Input	Expected Result	Status
Insert valid product	product_name="Phone", quantity=1, price=499.99	Product inserted into cart table	Passed
Insert with missing fields	quantity=1, price=499.99 (no product_name)	Error message for missing parameter	Passed
Update existing item	id=1, quantity=2, price=450.00	Cart item updated with new values	Passed
Update with non-existent ID	id=999, quantity=2, price=450.00	Error or no update performed	Passed
Delete valid item	id=1	Item removed from cart table	Passed
Delete with invalid ID	id=999	Error or no action taken	Passed
SQL injection attempt (in name)	product_name="" OR 1=1 --", quantity=1, price=500.00	Properly sanitized, rejected or inserted safely	Passed

10. Future Work

The CartService Microservice project successfully demonstrates how a simple yet modular backend system can be built using PHP to perform essential operations insert, update, and delete on cart data in an e-commerce context. The use of RESTful API principles allows seamless integration with frontend clients or other services, making it suitable for microservice-based architectures. Testing with tools like Postman confirmed that the service reliably handles various input scenarios, including edge cases and invalid data. Overall, the project fulfills its intended objectives and provides a functional, secure, and reusable component for shopping cart management.



Figure 2 Wampserver

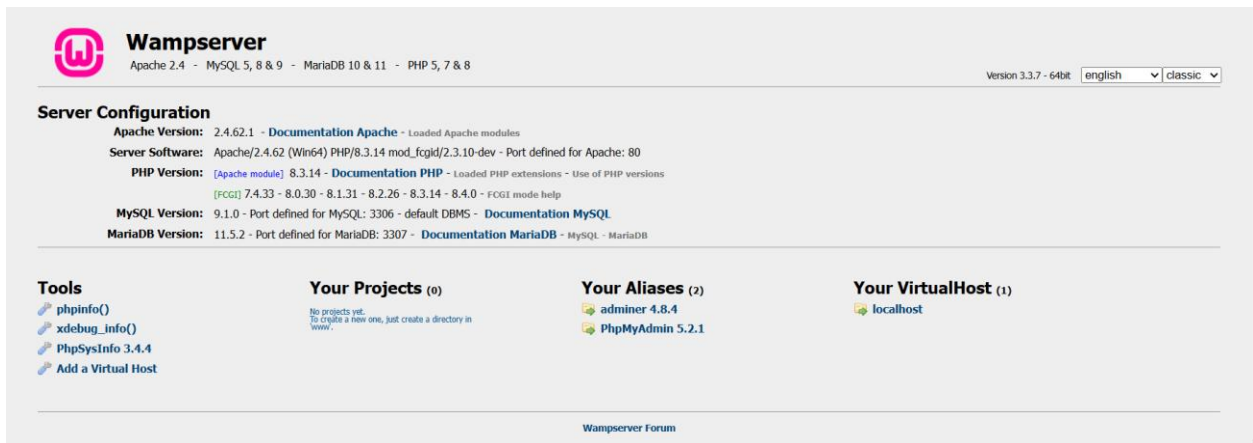


Figure 3 Wampserver Configuration

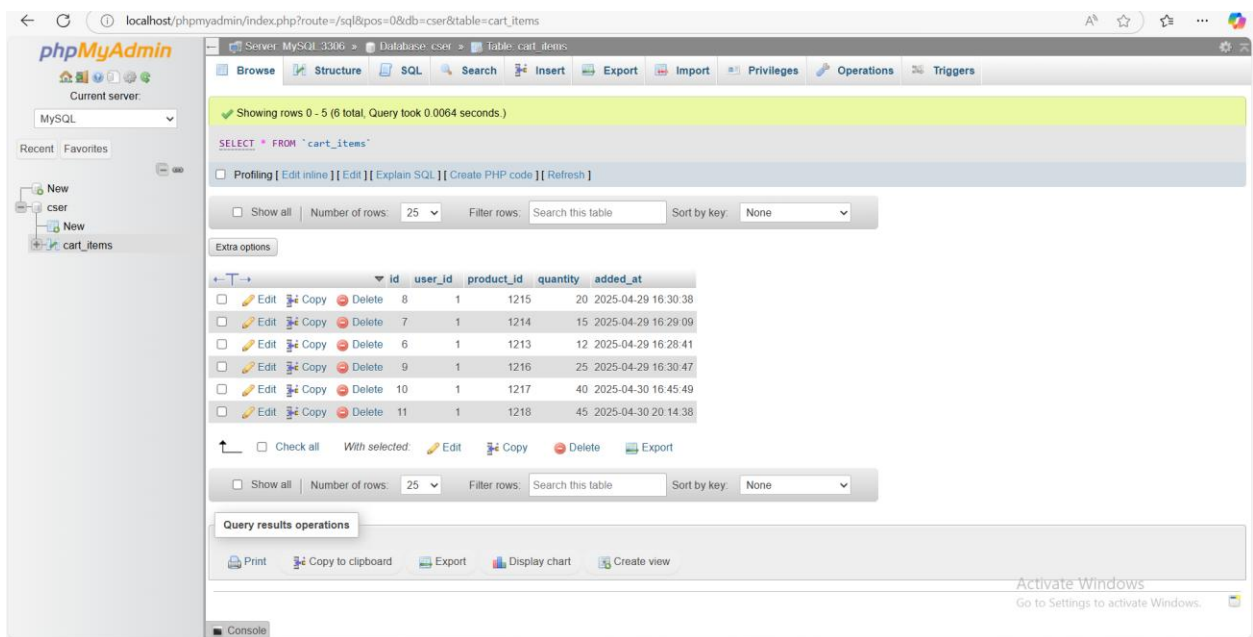


Figure 4 PHP MyAdmin Configuration

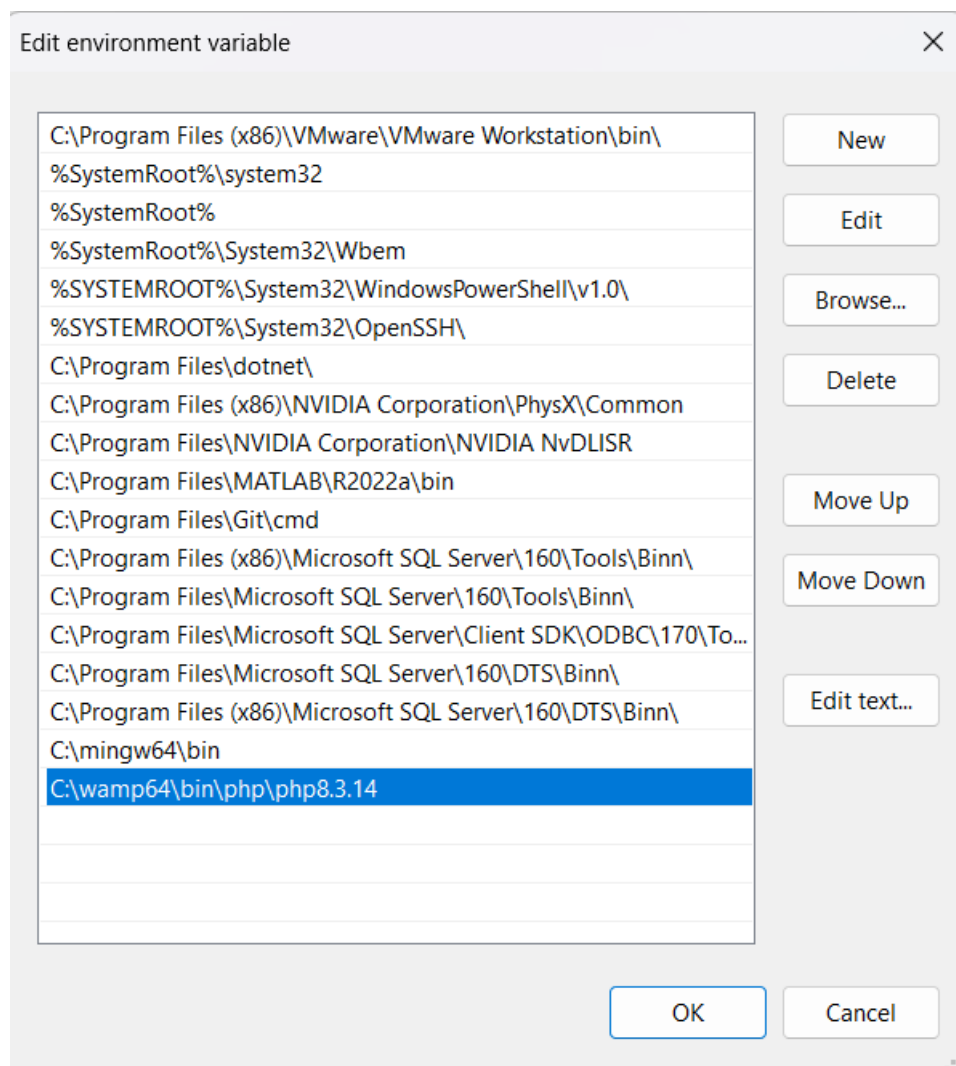


Figure 5 Environment Variable Setting

```

Command Prompt
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Zaid Bin Faheem>cd Desktop
C:\Users\Zaid Bin Faheem\Desktop>cd Steps
C:\Users\Zaid Bin Faheem\Desktop\Steps>cd cservice
C:\Users\Zaid Bin Faheem\Desktop\Steps\cservice>php -S localhost:8000 -t public/

```

Figure 6 Access Local Files

Shopping Cart Management

Add Item to Cart

Product ID

Enter Product ID

Quantity

Enter Quantity

Add to Cart

Cart Items

Product ID: 1215 Quantity: 20
Product ID: 1214 Quantity: 15
Product ID: 1213 Quantity: 12
Product ID: 1216 Quantity: 25
Product ID: 1217 Quantity: 40
Product ID: 1218 Quantity: 45

Activate Windows
Go to Settings to activate Windows.

Figure 7 User Interface Add Cart

Update Item Quantity

Product ID

Enter Product ID

Quantity

Enter Quantity

Update Item

Remove Item from Cart

Product ID

Enter Product ID

Remove from Cart

Activate Windows
Go to Settings to activate Windows.

Figure 8 User Interface Update Cart

Add Item to Cart

Product ID
1219

Quantity
50

Add to Cart

Cart Items

Product ID: 1215 Quantity: 20
Product ID: 1214 Quantity: 15
Product ID: 1213 Quantity: 12
Product ID: 1216 Quantity: 25
Product ID: 1217 Quantity: 40
Product ID: 1218 Quantity: 45

Update Item Quantity

Activate Windows
Go to Settings to activate Windows.

Figure 9 Add Cart Item Operation UI

```

Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Zaid Bin Faheem>cd Desktop
C:\Users\Zaid Bin Faheem\Desktop>cd Steps
C:\Users\Zaid Bin Faheem\Desktop\Steps>cd cservice
C:\Users\Zaid Bin Faheem\Desktop\Steps\cservice>php -S localhost:8000 -t public/
[Wed Apr 30 23:12:32 2025] PHP 8.3.14 Development Server (http://localhost:8000) started
[Wed Apr 30 23:12:44 2025] [::1]:49696 Accepted
[Wed Apr 30 23:12:44 2025] [::1]:49696 [200]: GET /index.php/cart/items?user_id=1
[Wed Apr 30 23:12:44 2025] [::1]:49696 Closing
[Wed Apr 30 23:13:49 2025] [::1]:49726 Accepted
[Wed Apr 30 23:13:49 2025] [::1]:49726 [200]: POST /index.php/cart/add?user_id=1
[Wed Apr 30 23:13:49 2025] [::1]:49726 Closing
[Wed Apr 30 23:14:17 2025] [::1]:49748 Accepted
[Wed Apr 30 23:14:17 2025] [::1]:49748 [200]: GET /index.php/cart/items?user_id=1
[Wed Apr 30 23:14:17 2025] [::1]:49748 Closing
  
```

Activate W

Figure 10 Operation Perform as Microservice

Product ID: 1216 Quantity: 25
Product ID: 1217 Quantity: 40
Product ID: 1218 Quantity: 45
Product ID: 1219 Quantity: 50

Update Item Quantity

Product ID

Quantity

Update Item

Remove Item from Cart

Product ID

Remove from Cart

Activate Windows
Go to Settings to activate Windows

Figure 11 User Interface Remove Cart

Product ID: 1216 Quantity: 25	<div><p>This page says</p><p>Item quantity updated</p><p>OK</p></div>	
Product ID: 1217 Quantity: 40		
Product ID: 1218 Quantity: 45		
Product ID: 1219 Quantity: 50		

Update Item Quantity

Product ID

Quantity

Update Item

Remove Item from Cart

Product ID

Remove from Cart

Activate Windows
Go to Settings to activate Windows

Figure 12 Update Cart Item Operation UI

```
Command Prompt - php -S lc x + v
Microsoft Windows [Version 10.0.26100.3775]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Zaid Bin Faheem>cd Desktop
C:\Users\Zaid Bin Faheem\Desktop>cd Steps
C:\Users\Zaid Bin Faheem\Desktop\Steps>cd cservice
C:\Users\Zaid Bin Faheem\Desktop\Steps\cservice>php -S localhost:8000 -t public/
[Wed Apr 30 23:12:32 2025] PHP 8.3.14 Development Server (http://localhost:8000) started
[Wed Apr 30 23:12:44 2025] [::1]:49696 Accepted
[Wed Apr 30 23:12:44 2025] [::1]:49696 [200]: GET /index.php/cart/items?user_id=1
[Wed Apr 30 23:12:44 2025] [::1]:49696 Closing
[Wed Apr 30 23:13:49 2025] [::1]:49726 Accepted
[Wed Apr 30 23:13:49 2025] [::1]:49726 [200]: POST /index.php/cart/add?user_id=1
[Wed Apr 30 23:13:49 2025] [::1]:49726 Closing
[Wed Apr 30 23:14:17 2025] [::1]:49748 Accepted
[Wed Apr 30 23:14:17 2025] [::1]:49748 [200]: GET /index.php/cart/items?user_id=1
[Wed Apr 30 23:14:17 2025] [::1]:49748 Closing
[Wed Apr 30 23:15:34 2025] [::1]:49813 Accepted
[Wed Apr 30 23:15:34 2025] [::1]:49813 [200]: OPTIONS /index.php/cart/update?user_id=1
[Wed Apr 30 23:15:34 2025] [::1]:49813 Closing
[Wed Apr 30 23:15:34 2025] [::1]:49815 Accepted
[Wed Apr 30 23:15:34 2025] [::1]:49815 [200]: PUT /index.php/cart/update?user_id=1
[Wed Apr 30 23:15:34 2025] [::1]:49815 Closing
[Wed Apr 30 23:16:08 2025] [::1]:49827 Accepted
[Wed Apr 30 23:16:08 2025] [::1]:49827 [200]: GET /index.php/cart/items?user_id=1
[Wed Apr 30 23:16:08 2025] [::1]:49827 Closing
```

Figure 13 Operation Perform as Microservice

Product ID: 1216 | Quantity: 25

Product ID: 1217 | Quantity: 40

Product ID: 1218 | Quantity: 45

Product ID: 1219 | Quantity: 100

This page says

Item removed from cart

OK

Update Item Quantity

Product ID

Enter Product ID

Quantity

Enter Quantity

Update Item

Remove Item from Cart

Product ID

1219

Remove from Cart

Activate Windows
Go to Settings to activate Wind

Figure 14 Remove Cart Item Operation UI

```
Command Prompt - php -S lc x + v
C:\Users\Zaid Bin Faheem\Desktop\Steps>cd cservice
C:\Users\Zaid Bin Faheem\Desktop\Steps\cservice>php -S localhost:8000 -t public/
[Wed Apr 30 23:12:32 2025] PHP 8.3.14 Development Server (http://localhost:8000) started
[Wed Apr 30 23:12:44 2025] [::1]:49696 Accepted
[Wed Apr 30 23:12:44 2025] [::1]:49696 [200]: GET /index.php/cart/items?user_id=1
[Wed Apr 30 23:12:44 2025] [::1]:49696 Closing
[Wed Apr 30 23:13:49 2025] [::1]:49726 Accepted
[Wed Apr 30 23:13:49 2025] [::1]:49726 [200]: POST /index.php/cart/add?user_id=1
[Wed Apr 30 23:13:49 2025] [::1]:49726 Closing
[Wed Apr 30 23:14:17 2025] [::1]:49748 Accepted
[Wed Apr 30 23:14:17 2025] [::1]:49748 [200]: GET /index.php/cart/items?user_id=1
[Wed Apr 30 23:14:17 2025] [::1]:49748 Closing
[Wed Apr 30 23:15:34 2025] [::1]:49813 Accepted
[Wed Apr 30 23:15:34 2025] [::1]:49813 [200]: OPTIONS /index.php/cart/update?user_id=1
[Wed Apr 30 23:15:34 2025] [::1]:49813 Closing
[Wed Apr 30 23:15:34 2025] [::1]:49815 Accepted
[Wed Apr 30 23:15:34 2025] [::1]:49815 [200]: PUT /index.php/cart/update?user_id=1
[Wed Apr 30 23:15:34 2025] [::1]:49815 Closing
[Wed Apr 30 23:16:08 2025] [::1]:49827 Accepted
[Wed Apr 30 23:16:08 2025] [::1]:49827 [200]: GET /index.php/cart/items?user_id=1
[Wed Apr 30 23:16:08 2025] [::1]:49827 Closing
[Wed Apr 30 23:18:07 2025] [::1]:49855 Accepted
[Wed Apr 30 23:18:07 2025] [::1]:49855 [200]: OPTIONS /index.php/cart/remove?user_id=1&product_id=1219
[Wed Apr 30 23:18:07 2025] [::1]:49855 Closing
[Wed Apr 30 23:18:07 2025] [::1]:49857 Accepted
[Wed Apr 30 23:18:07 2025] [::1]:49857 [200]: DELETE /index.php/cart/remove?user_id=1&product_id=1219
[Wed Apr 30 23:18:07 2025] [::1]:49857 Closing
```

Activate W

Figure 15 Operation Perform as Microservice