1. Implementation

1.1. Project Structure

The microservice is organized into a separate directory named Product_Microservice, distinct from the main website's source code. This enforces physical separation of concerns.

1.2. Microservice Logic

Unlike the Layered Architecture in Lab 3 where logic was split across Controllers and Repositories, this Microservice encapsulates the necessary logic in a single entry point api.php to handle HTTP requests efficiently.

Key responsibilities of this file:

1. **Database Connection:** Connects securely to the webbandocu database.
2. **CORS Handling:** Allows external applications to request data.
3. **Routing:** Detects request methods (GET, POST) to trigger appropriate actions.

Code Snippet (Core Logic):

```
// Handling GET Request for Product List or Detail

if ($method == 'GET') {

    if (isset($_GET['id'])) {

        // Fetch specific product by ID

        $id = intval($_GET['id']);

        $sql = "SELECT * FROM products WHERE item_id = $id";

        // ... execute query and return JSON

    } else {

        // Fetch all products
        $sql = "SELECT item_id, item_name, item_price, item_quantity FROM products";

        // ... execute query and return JSON array

    }

}
```

2. Test Results (Isolation Testing)

To verify that the service operates independently, we bypassed the main web interface and interacted directly with the Microservice via **Port 5001**.

Startup Command:

```
php -S localhost:5001 api.php
```

2.1. Test: Get All Products (GET)

- **Endpoint:** http://localhost:5001/api.php
- **Description:** Retrieves the full catalog of second-hand items.
- **Response Status:** 200 OK

2.2. Test: Get Product Details (GET)

- **Endpoint:** http://localhost:5001/api.php?id=1
- **Description:** Retrieves details for a specific product (e.g., ID 1).
- **Response Status:** 200 OK

2.3. Test: Create New Product (POST)

- **Endpoint:** http://localhost:5001/api.php
- **Description:** Adds a new item to the inventory.
- **Payload (JSON):**

```
{
    "name": "Vintage Camera",

    "price": 1500000,

    "quantity": 1

}
```

- **Response Status:** 201 Created