1. Implementation

1.1. Architecture Overview

The Gateway runs independently on **Port 5000**. It intercepts all incoming traffic.

- Client sends request to localhost:5000 (Gateway).
- Gateway verifies the Security Token.
- If valid, Gateway uses cURL to call localhost:5001 (Product Service).
- Gateway returns the response to the Client.

1.2. Gateway Logic (**gateway.php**)

The gateway logic is divided into three main layers:

1. **Security Layer:** Checks for the presence of a valid Authorization header. We simulated two roles:
    - Bearer valid-user-token: Allows read-only access (GET).
    - Bearer admin-token: Allows write access (POST). If the token is missing or invalid, the Gateway returns HTTP 401.
2. **Routing Layer:** Identifies the destination service. In this implementation, it constructs the target URL pointing to localhost:5001 while preserving query parameters (e.g., ?id=1).
3. **Proxy Layer:** Uses curl_exec to forward the client's request (Method, Headers, Body) to the backend and returns the backend's response to the client. It also handles HTTP 503 errors if the backend service is offline.

2. Test Results

We verified the Gateway's functionality using **Postman** through four specific scenarios:

2.1. Security Test: Unauthorized Access

- **Scenario:** Client sends a request without an Authentication Token.
- **Request:** GET http://localhost:5000

- **Result:** The Gateway blocked the request and returned **HTTP 401 Unauthorized** with the message: *"Unauthorized access. Invalid or missing token."*

2.2. Routing Test: Authorized Access (Success)

- **Scenario:** Client sends a valid user token.
- **Request:** GET http://localhost:5000 with Header Authorization: Bearer valid-user-token.
- **Result:** The Gateway successfully forwarded the request to Port 5001. The response was **HTTP 200 OK**, containing the JSON list of products retrieved from the backend database.

2.3. Authorization Test: Forbidden Action

- **Scenario:** A regular User tries to perform an Admin action (Creating a product).
- **Request:** POST http://localhost:5000 with Header Authorization: Bearer valid-user-token.
- **Result:** The Gateway inspected the method and token, determining the user lacked permissions. It returned **HTTP 403 Forbidden** with the message: *"Forbidden. Only Admins can create products."*

2.4. Resilience Test: Service Unavailable

- **Scenario:** The Backend Service (Port 5001) was manually stopped to simulate a crash.
- **Request:** GET http://localhost:5000 with a valid token.
- **Result:** The cURL connection failed. Instead of crashing, the Gateway handled the exception and returned **HTTP 503 Service Unavailable** with the message: *"Service Unavailable. Backend is down."*