## horizontal line



Bakery Software Management

System

**──────────────────**

كريم عبد المنعم احمد

عبدالرحمن محمد عبدالنبي

احمد محمد نبيل

باسم بركة الله

# Overview

This project proposes the development and documentation of a robust set of APIs (Application Programming Interfaces) to streamline the operations of online shops. These APIs will empower developers to integrate functionalities like product management, order processing, and customer interaction into their e-commerce platforms.

# Goals

1. Develop a comprehensive set of RESTful APIs for managing online shops
2. Functionalities like product management, order processing, and customer interaction into their e-commerce platforms
3. Improve efficiency and scalability of online shop operations through automation and seamless integration (APIs can automate tasks such as product updates and order processing, which can save time and resources. They can also be easily scaled to accommodate a growing business.)

# Specifications

**Functional Specifications:**

* **API Type:** RESTful web services
* **Authentication:** JSON Web Token (JWT) based authentication
* **Available Endpoints:**
  + **Products:**
    - GET /products: Retrieve a list of all products.
    - GET /products/{id}: Retrieve details of a specific product by its ID.
    - (Optional) POST /products: Create a new product (if inventory management is included).
    - (Optional) PUT /products/{id}: Update an existing product.
    - (Optional) DELETE /products/{id}: Delete a product (if inventory management is included).
  + **Orders:**
    - GET /orders: Retrieve a list of all orders (may require additional filters based on user or status).
    - GET /orders/{id}: Retrieve details of a specific order by its ID.
    - POST /orders: Place a new order, specifying products and quantities.
  + **Customers (Optional):**
    - GET /customers: Retrieve a list of all customers (may require authentication for privacy reasons).
    - GET /customers/{id}: Retrieve details of a specific customer by their ID (may require authentication for privacy reasons).
    - (Optional) POST /customers: Create a new customer account.
  + **Inventory (Optional):**
    - GET /inventory: Retrieve information on current inventory levels.
    - (Optional) PUT /inventory/{id}: Update inventory levels for a specific product.
* **Non-Functional Specifications:**
* **Security:**
  + Secure data transmission using HTTPS.
  + Proper authorization practices to restrict access to sensitive data.
  + Regular security audits and vulnerability assessments.
* **Performance:**
  + APIs should be responsive and handle requests efficiently.
  + Implement caching mechanisms to improve performance for frequently accessed data.
* **Scalability:**
  + The APIs should be able to handle a growing number of users and requests.
* **Documentation:**
  + Comprehensive API documentation with clear instructions, usage examples (including code snippets in different programming languages if applicable).
  + Separate error handling documentation detailing error codes and troubleshooting steps.
* **Versioning:**
  + Implement a versioning system for the APIs and documentation to track changes and ensure compatibility.

**Additional Considerations:**

* **Error Handling:** Define a consistent error handling approach using HTTP status codes and clear error messages.
* **Testing:** Implement a comprehensive testing strategy to ensure the APIs function correctly.
* **Deployment:** Choose a suitable deployment environment for the APIs that is secure and reliable.
* **Monitoring:** Monitor the APIs for performance and errors to identify and address any issues promptly.