Assessment 3

# Task 1

The system architecture for Jim's Fresh Fruit Click & Collect eStore should be scalable, secure, and cost-effective to meet the needs of the small business. This architecture will follow a client-server model and use a multi-tier architecture to ensure flexibility, security, and scalability. The architecture will involve several key components, including clients, servers, databases, and external services.

## Client Layer (Frontend)

The client layer consists of devices used by customers, staff, and administrators to access the eStore. This layer includes web browsers and mobile apps through which users interact with the system.

Devices:

Desktop PCs, laptops, tablets, smartphones.

User Interfaces:

A responsive web application or mobile app (Android/iOS) built using frameworks such as ReactJS or Angular for a seamless, cross-device experience.

Technology:

HTML5, CSS3, JavaScript, and a front-end framework (ReactJS, Angular).

Role:

Users (customers, staff, and admin) interact with the system to browse products, place orders, manage inventory, and generate reports.

Justification:

A responsive web app ensures compatibility with various devices, improving the user experience for both customers and staff. ReactJS or Angular are popular choices due to their component-based architecture, which allows reusability and enhances performance.

## Application Layer (Backend)

The application layer is responsible for processing customer orders, managing inventory, and generating reports. This layer is hosted on an application server that runs the business logic.

Technology:

Node.js or Django (Python) for the backend API.

Business Logic:

This handles order management, payment processing, customer notifications, and inventory tracking.

Role:

Receives client requests (e.g., placing an order, updating inventory), processes them, and sends back the results to the client.

Justification:

Node.js allows for asynchronous, event-driven architecture, making it suitable for high-performance web applications with real-time capabilities. Django offers a "batteries-included" framework, which includes security, scalability, and rapid development tools. Python is widely used and trusted for building scalable applications.

## Database Layer

The database layer is where all the data is stored and managed, including customer information, product inventory, order details, and payment records.

Database Management System (DBMS):

MySQL or PostgreSQL for relational data storage.

Role:

Store and manage customer orders, product catalogue, inventory, and payment information.

Justification:

MySQL or PostgreSQL are both open-source, relational databases that offer scalability, security, and robustness. They are suitable for handling structured data such as orders, products, and user accounts.

## Security and Authentication Layer

This layer provides secure authentication and data protection for the entire system.

Authentication:

OAuth 2.0 for secure login (via Google, Facebook, or custom accounts).

Data Security:

SSL/TLS encryption for secure communication between clients and the server.

Role:

Ensures that sensitive data, such as customer information and payment details, are securely transmitted and stored.

Justification:

OAuth 2.0 provides a standardized, secure protocol for user authentication. SSL/TLS encryption ensures secure data transmission over the internet, protecting sensitive information like payment details.

## Payment Gateway Integration

The system will integrate with an external payment gateway for processing payments.

Payment Gateway:

Stripe, PayPal, or Square.

Role:

Manages secure payment transactions, ensuring that Jim's Fresh Fruit does not need to store or process payment information directly.

Justification:

Stripe, PayPal, or Square are secure, widely used payment gateways that offer easy integration with eCommerce systems and comply with PCI-DSS security standards.

## Hosting and Infrastructure

The application, database, and other backend services will be hosted in a cloud environment, ensuring scalability and high availability.

Cloud Provider:

AWS (Amazon Web Services) or Microsoft Azure.

Components:

**Web Server:** Nginx or Apache for serving web pages and static content.

**Application Server**: Hosts the business logic (Node.js or Django application).

**Database Server:** MySQL or PostgreSQL server to manage data.

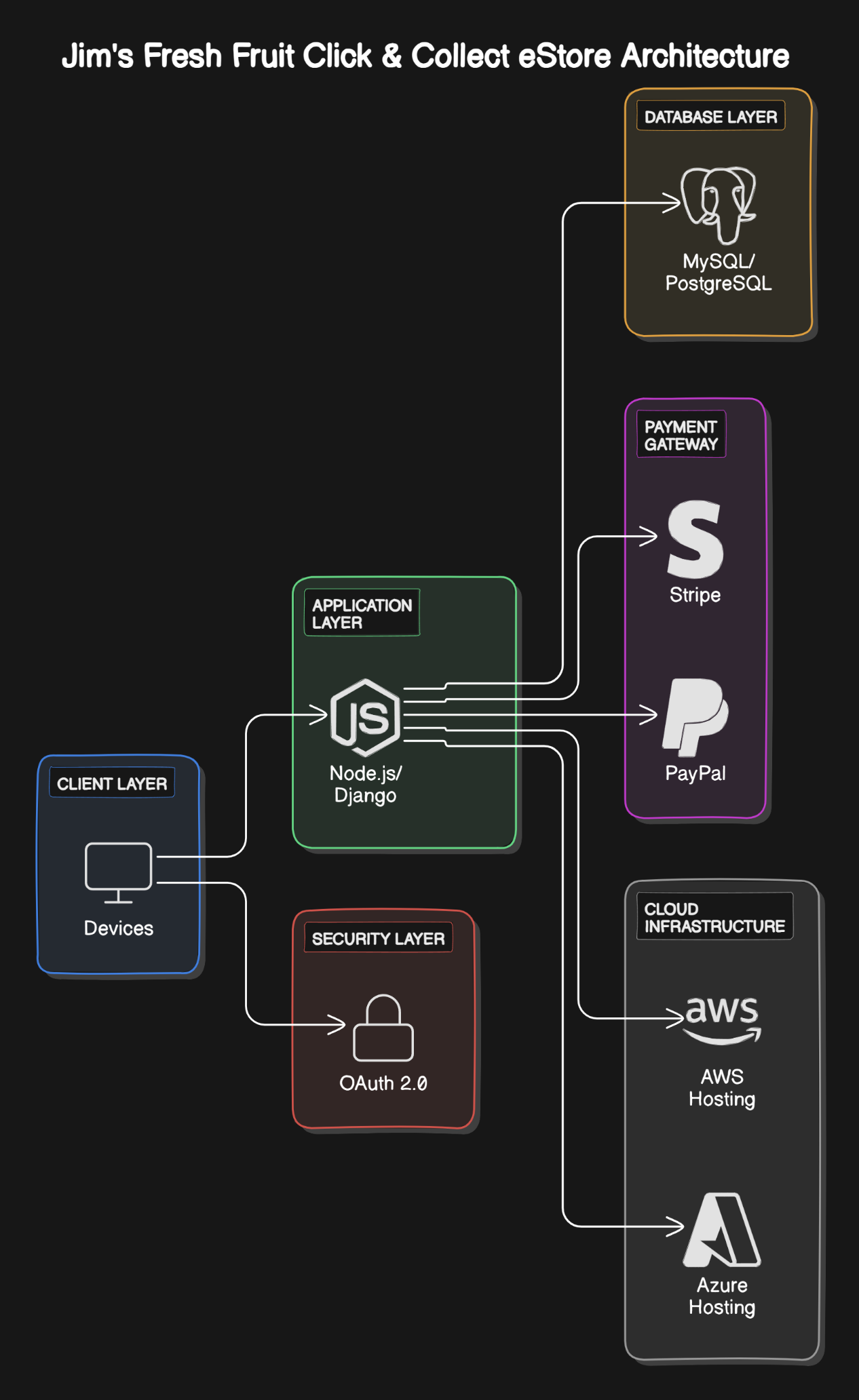
**Storage:** Cloud storage for media files (images of fruits, order receipts, etc.).

**Load Balancer**: AWS Elastic Load Balancing to distribute traffic evenly across multiple servers.

Justification:

AWS or Azure provide highly scalable and reliable cloud services, ensuring that the system can manage increased traffic during peak hours (e.g., promotions or holiday seasons). Cloud storage ensures durability and easy access to stored files, and load balancing enhances the system's ability to manage high traffic volumes.

## Architecture Diagram



Client Layer:

Devices (Web browser on desktops, laptops, tablets, mobile phones) Access Jim's Fresh Fruit Click & Collect eStore using web technologies (HTML5, CSS3, JavaScript).

Application Layer:

Application Server (Node.js/Django). Processes business logic (order processing, inventory management).

Database Layer:

Database Server (MySQL/PostgreSQL). Stores structured data (customer info, product catalogue, orders).

Payment Gateway:

Stripe/PayPal. Secure payment processing integrated with the application layer.

Security Layer:

OAuth 2.0 for user authentication. SSL/TLS for encrypted data transmission.

Cloud Infrastructure:

AWS/Azure hosting services, including web, application, and database servers. Load balancing, auto-scaling, and cloud storage.

Conclusion:

This architecture design ensures that Jim’s Fresh Fruit Click & Collect eStore is scalable, secure, and user-friendly. By leveraging modern cloud infrastructure, open-source databases, and secure authentication/payment protocols, the system will be robust and capable of managing the demands of both customers and staff. The choice of AWS or Azure ensures that the infrastructure can scale as the business grows, while the use of modern frontend technologies ensures a smooth customer experience.

# Task 2

For the Jim’s Fresh Fruit Click & Collect eStore, I will perform use case realization by creating a sequence diagram for the "Place Order" use case. This use case involves a customer selecting fruits, adding them to the cart, proceeding to checkout, making a payment, and receiving an order confirmation.

## Use Case: Place Order

Actors:

Customer, System (eStore), Payment Gateway

Primary Flow:

1. The customer browses the product catalogue.
2. The customer adds fruits to the shopping cart.
3. The customer proceeds to checkout.
4. The system displays the order summary and total cost.
5. The customer confirms the order and makes a payment.
6. The system processes the payment through a payment gateway.
7. The system sends an order confirmation to the customer.

## Sequence Diagram

1. Customer interacts with the eStore system to browse the product catalogue and selects fruits to add to their shopping cart.
2. eStore system updates the shopping cart and displays the selected items to the customer.
3. Once ready, the Customer initiates the checkout process.
4. eStore system displays the order summary, including the total cost, for the customer to review.
5. The Customer confirms the order and proceeds to make a payment.
6. The eStore system sends the payment details to the Payment Gateway (e.g., Stripe or PayPal).
7. The Payment Gateway processes the payment and sends a confirmation back to the eStore system.
8. The eStore system updates the order status to "confirmed" and sends an order confirmation to the Customer (via email or system notification).
9. The order details are stored in the Database for tracking and inventory management.

