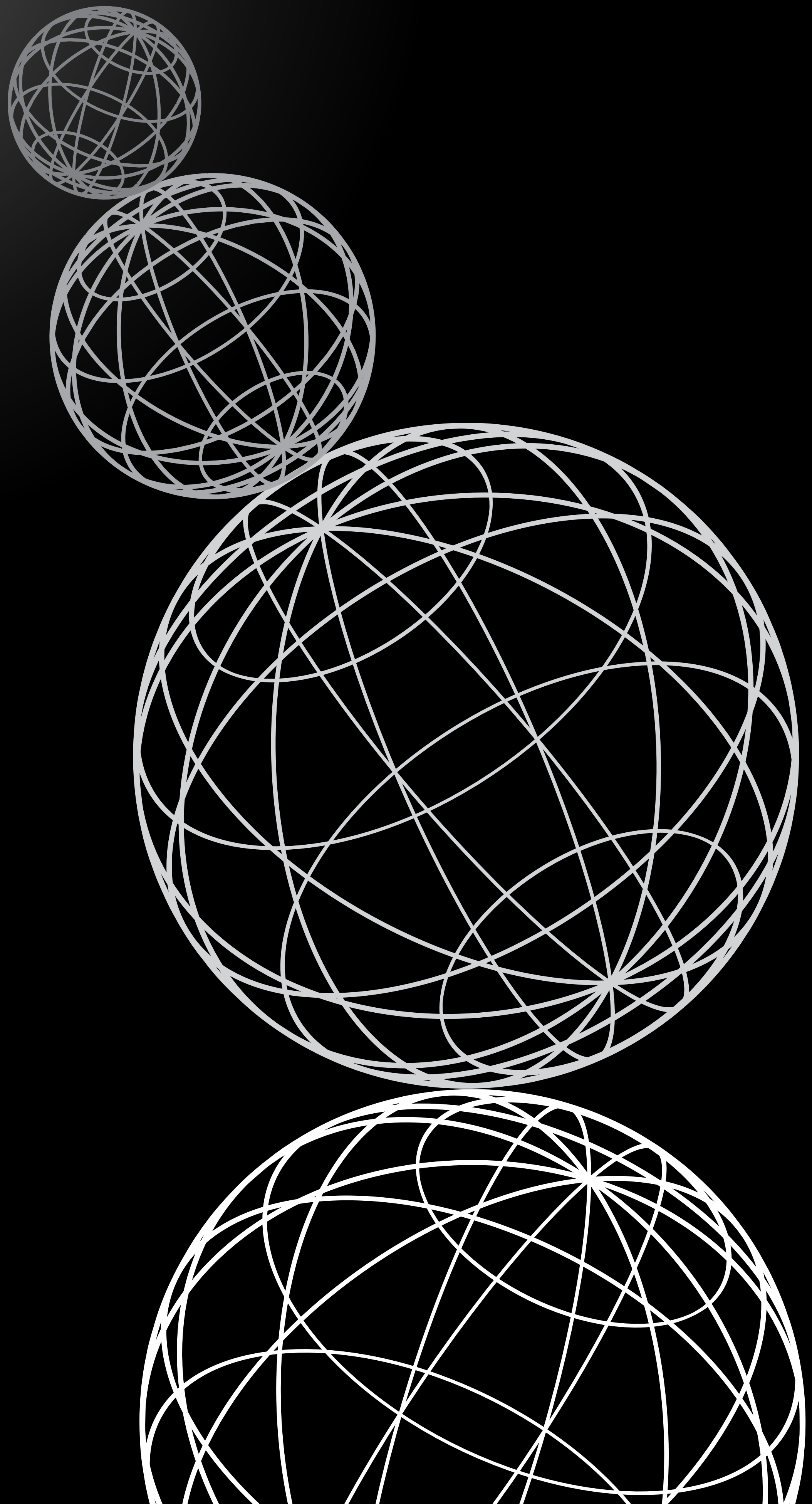
INVENTRA - INTELLIGENT INVENTORY MANAGEMENT SYSTEM

MILESTONE-2

PRESENTED BY
RISHWANA SIRAJUTHEEN

WHAT IS INVENTRA?

- Inventra is an intelligent inventory management system
- Designed to automate inventory tracking and control
- Reduces manual errors and stock inconsistencies
- Follows industry-standard backend architecture



PROBLEM STATEMENT

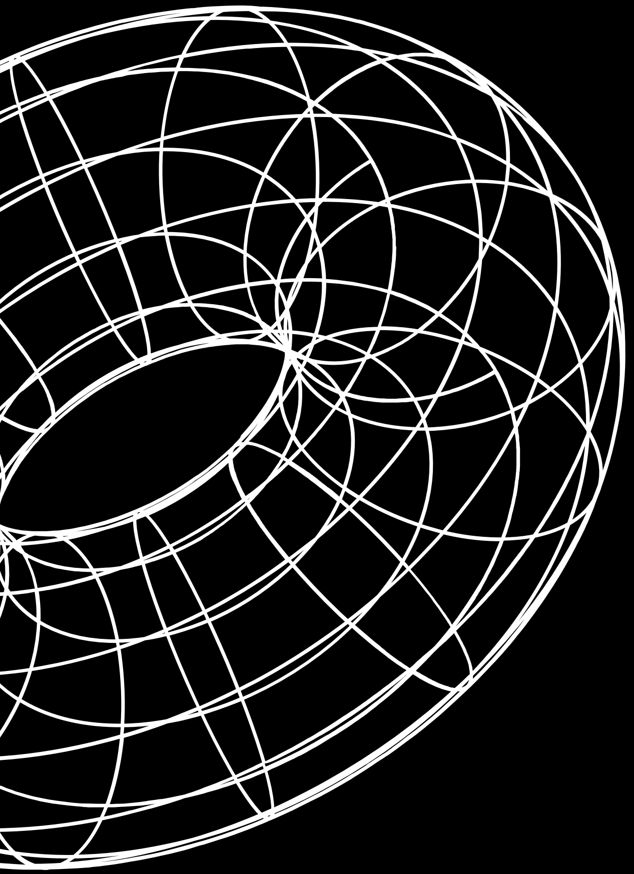
Problems in Traditional Inventory Systems

- Manual stock updates
- Duplicate product entries
- Stock mismatch and losses
- No proper access control
- Lack of audit and tracking

OBJECTIVES OF INVENTRA

System Objectives

- Secure system access using authentication
- Accurate stock tracking in real time
- Prevent duplicate and invalid operations
- Maintain transaction history
- Provide alerts and reports



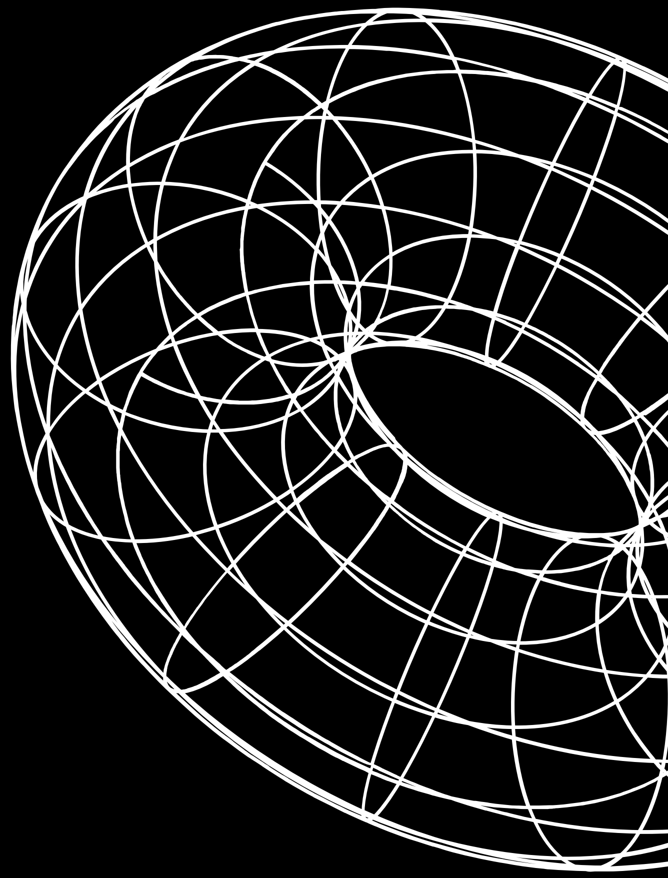
SYSTEM ARCHITECTURE

Layered Architecture

- Presentation Layer (UI)
- Controller Layer
- Service Layer (Business Logic)
- Repository Layer
- Database Layer

Flow:

UI → Controller → Service → Repository → Database



MODULE OVERVIEW

Major Modules

- Authentication & Authorization
- Product & Inventory Management
- Inventory Alerts
- Transaction History
- Reports & Analytics

MODULE 1: AUTHENTICATION

Authentication & Authorization

- User registration and login
- Encrypted password storage
- JWT-based token authentication
- Role-based access control
- Secure password recovery

Objective

- Secure access to the system
- Validate user identity
- Control access using roles

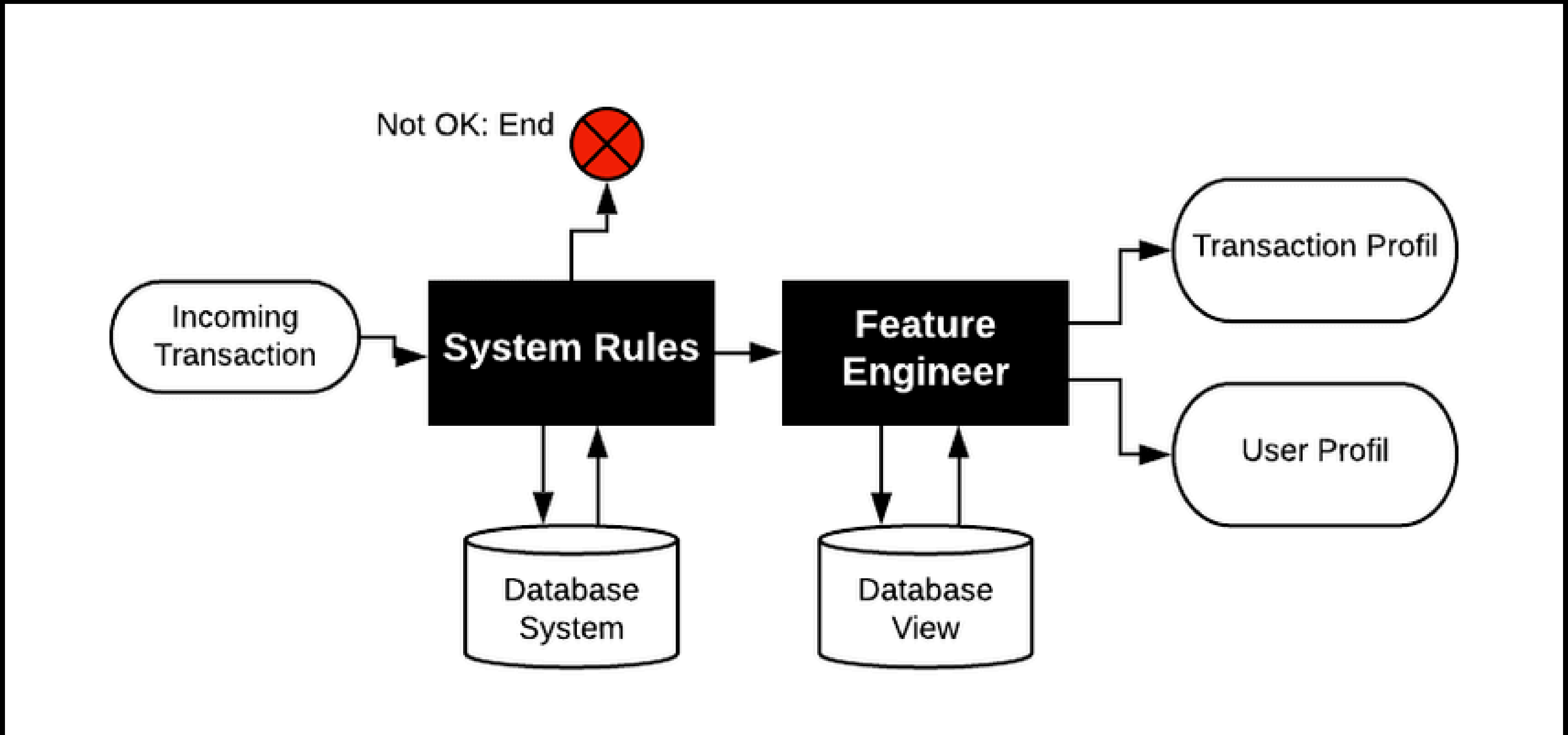
WHY AUTHENTICATION IS REQUIRED

Problems Without Authentication

- Unauthorized access
- Data manipulation
- No user accountability

Solution

- Login-based access
- Role-based control
- Token-based authorization



CLASSES USED IN MODULE 1

Architecture Flow

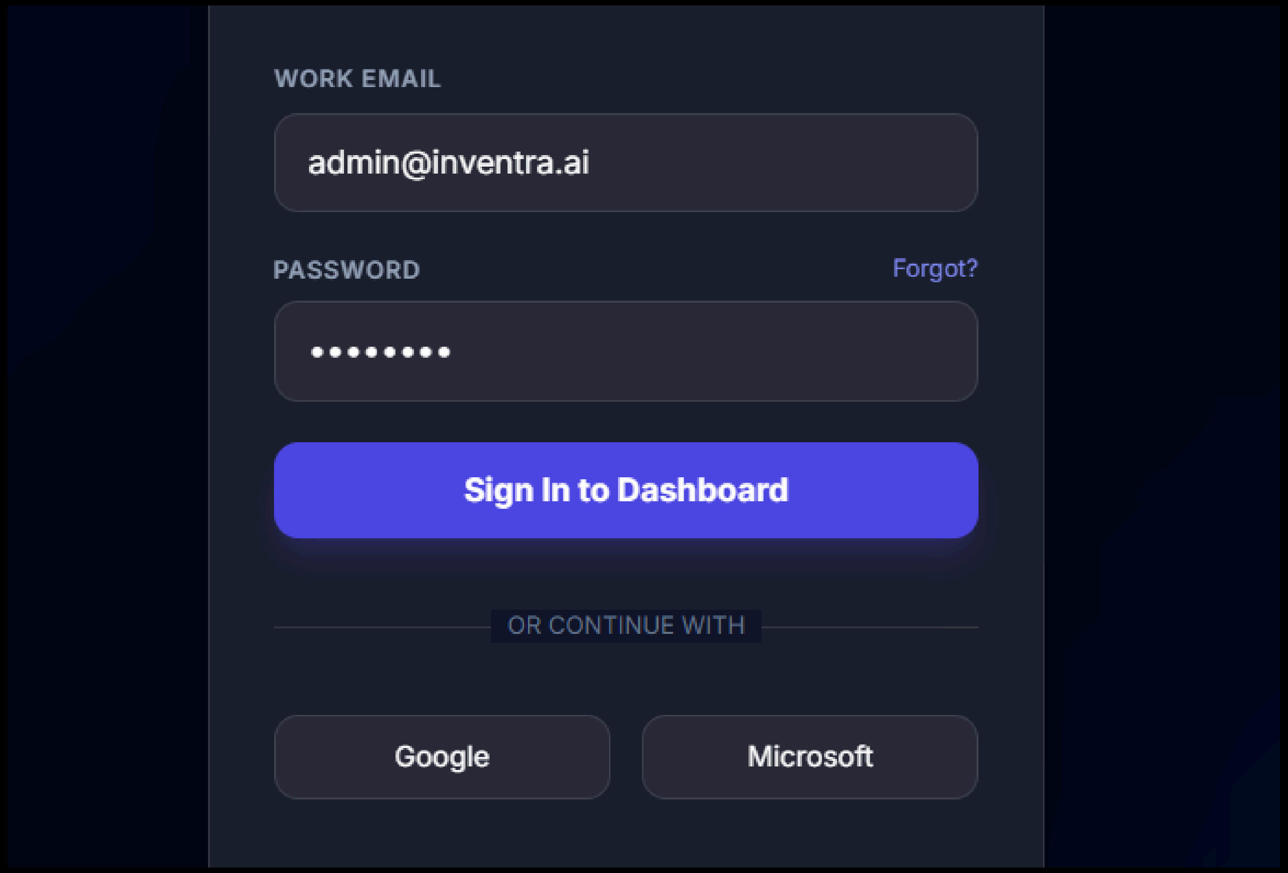
UI → AuthController → AuthService → UserRepository → Database

Explain

- Controller handles requests
- Service applies business rules
- Repository interacts with database

Core Classes

- AuthController
- AuthService
- User
- UserRepository
- JWTUtility
- EmailService



WORK EMAIL

admin@inventra.ai

PASSWORD

••••••••

STEP-BY-STEP TECHNICAL EXPLANATION

User Interface (UI)

- User enters username & password
- Sends request to backend

AuthController

- Receives request
- Does no validation
- Forwards data to service layer

AuthService (CORE LOGIC)

- Verifies user existence
- Matches encrypted password
- Generates JWT Token

UserRepository

- Communicates with database
- Fetches user record
- Saves or updates user data

Database

- Stores encrypted passwords
- Stores user roles
- Never stores plain credentials

JWT Token

- Generated after successful login
- Sent to UI
- Used for all future requests

SIGN-UP (REGISTER USER) – LOGIC

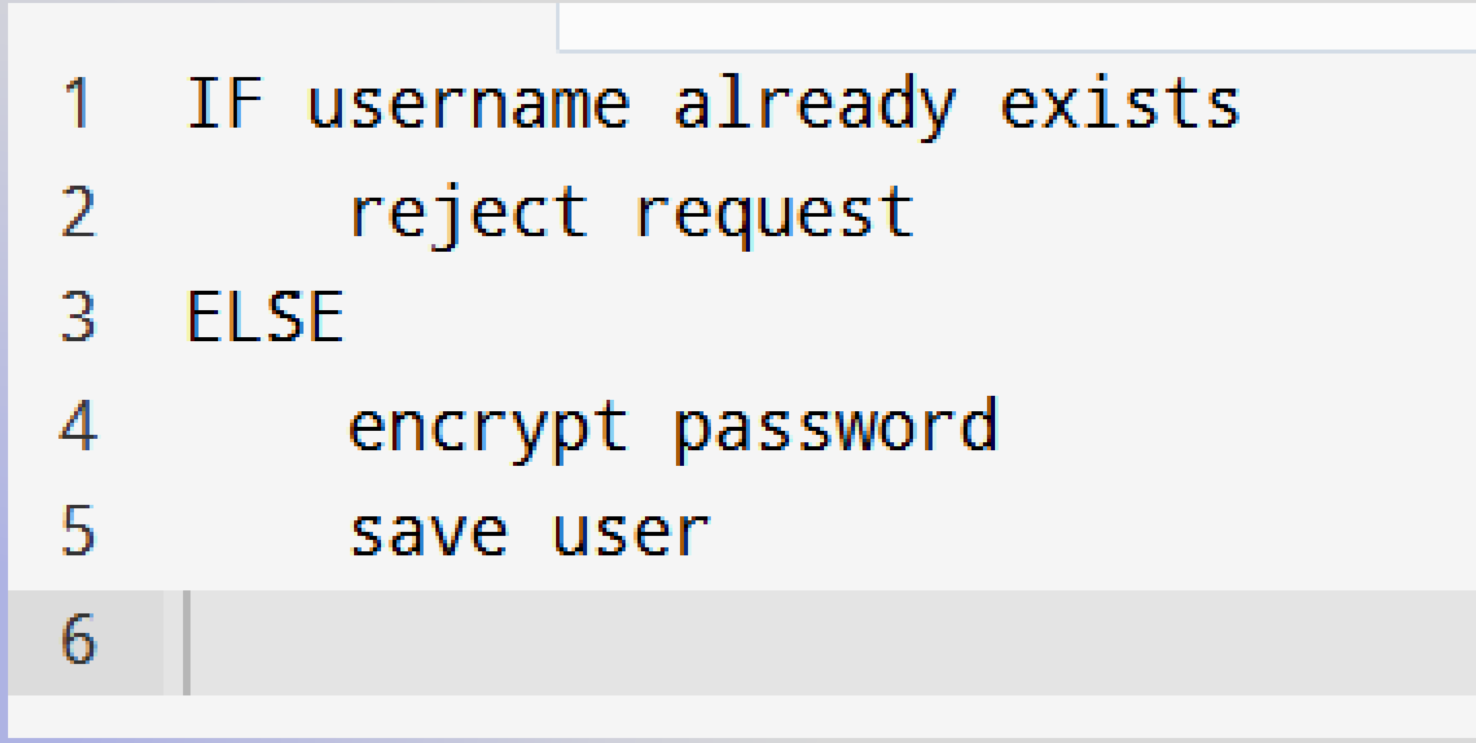
Purpose

- Create new user securely
- Prevent duplicate users

Key Explanation

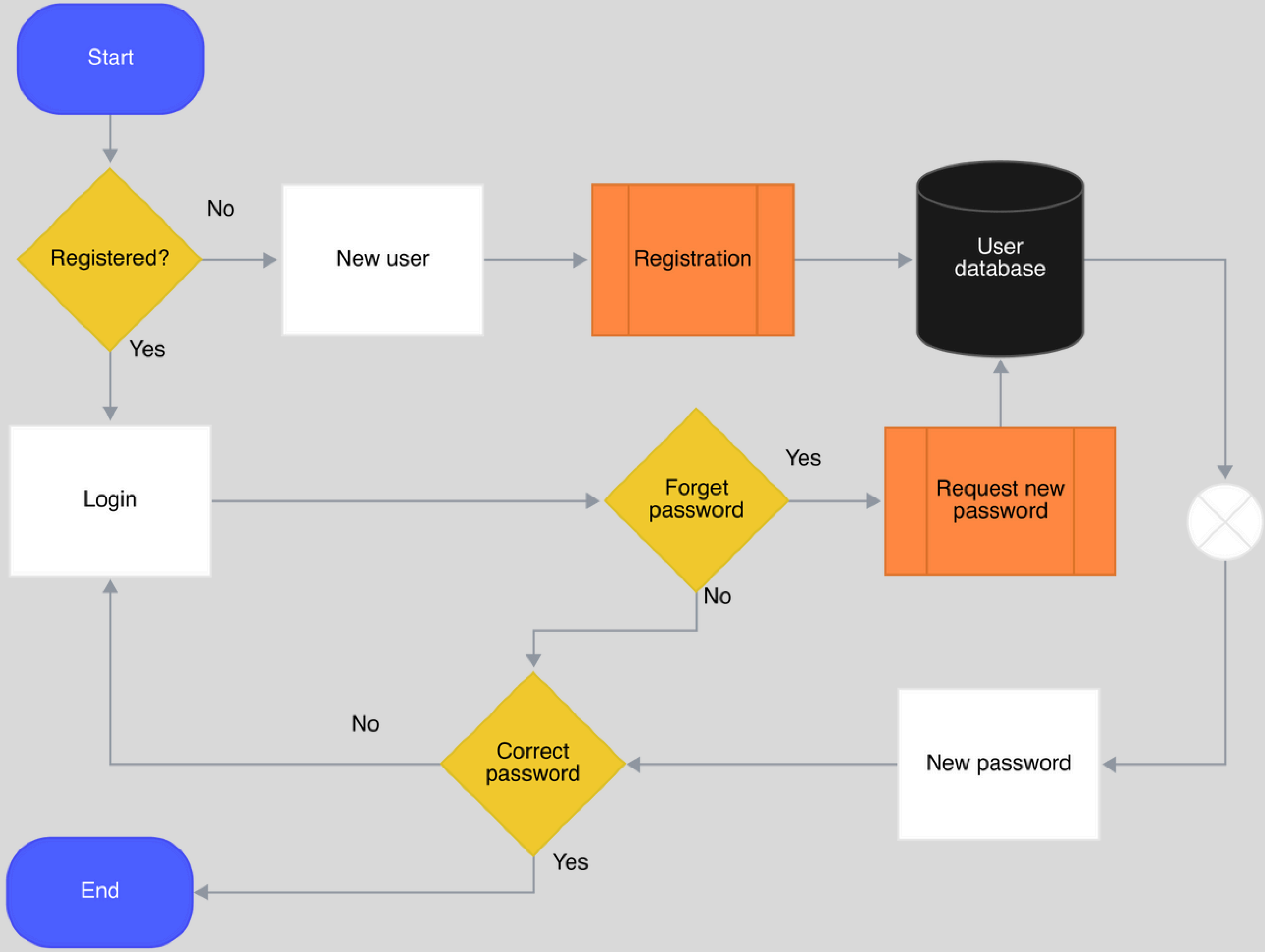
- Username uniqueness enforced
- Password encryption before storage

Pseudocode



```
IF username already exists
    reject request
ELSE
    encrypt password
    save user
```

SIGN-UP FLOW



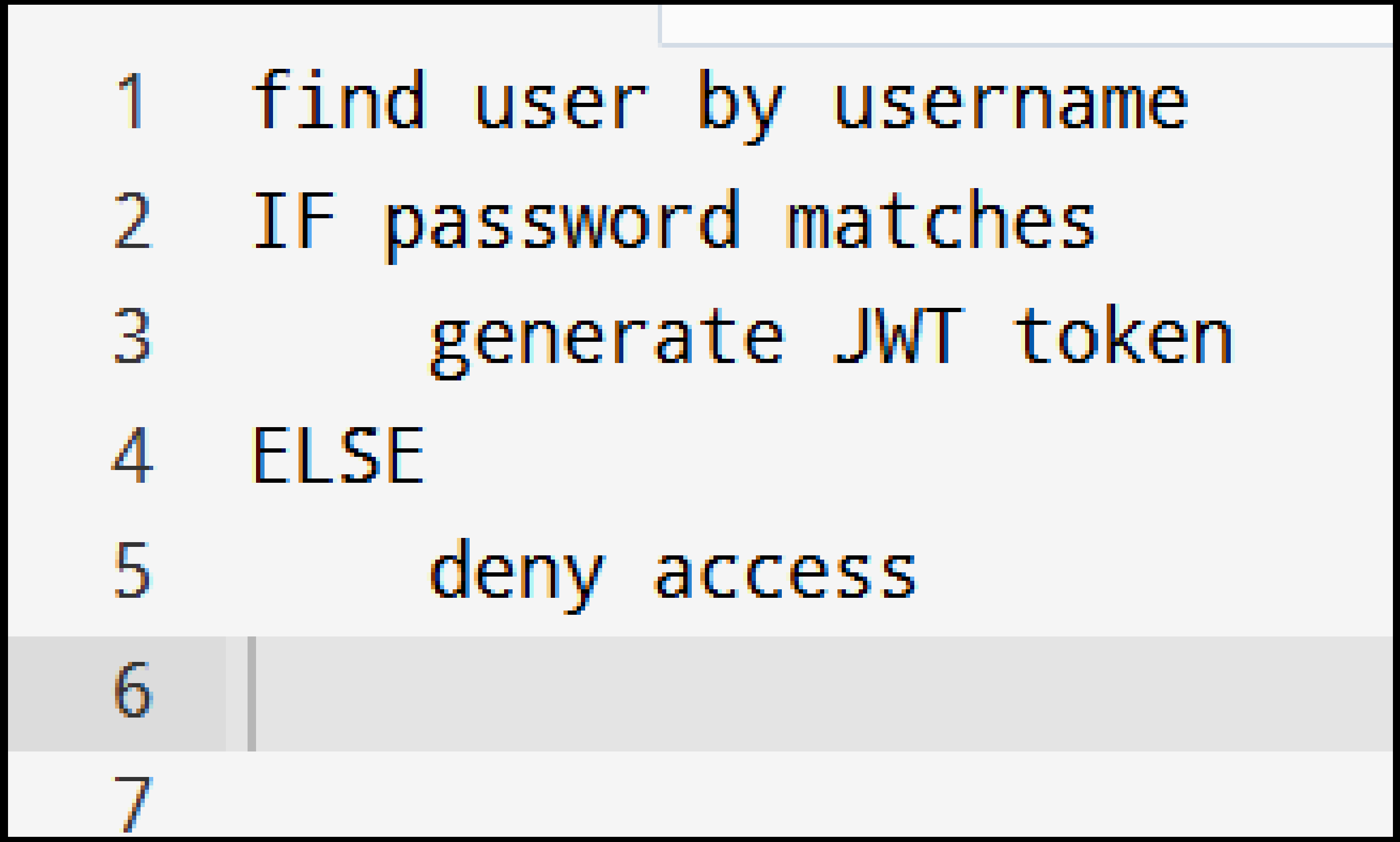
UI → Controller → Service → Repository → DB

SIGN-IN (LOGIN) & AUTHENTICATION

Purpose

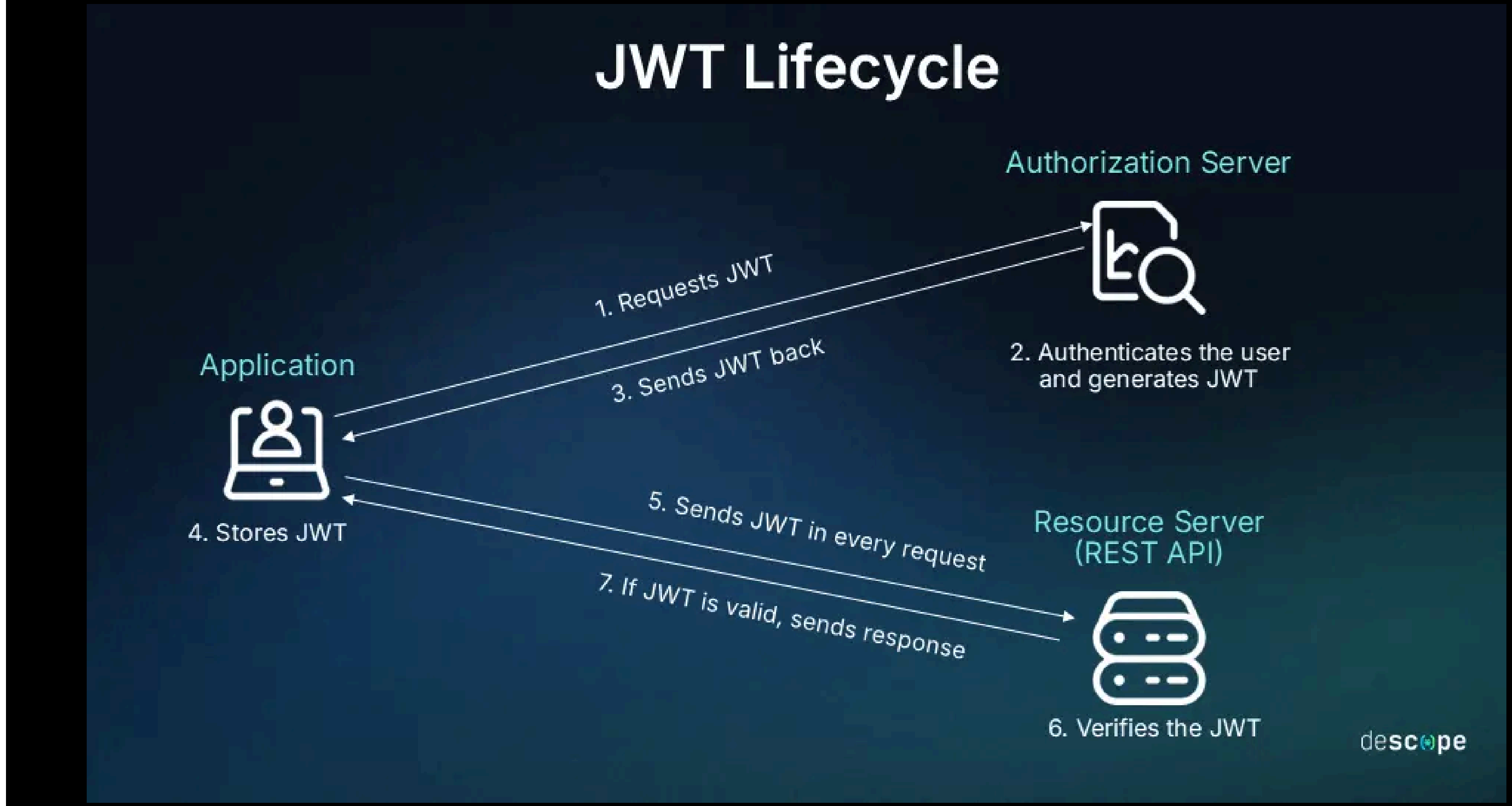
- Validate credentials
- Generate access token

Pseudocode



```
find user by username
IF password matches
    generate JWT token
ELSE
    deny access
```

JWT TOKEN – CORE SECURITY CONCEPT


JWT Lifecycle
Authorization Server
1. Requests JWT
2. Authenticates the user and generates JWT
Application
3. Sends JWT back
4. Stores JWT
5. Sends JWT in every request
Resource Server (REST API)
7. If JWT is valid, sends response
6. Verifies the JWT
descope

Why JWT?

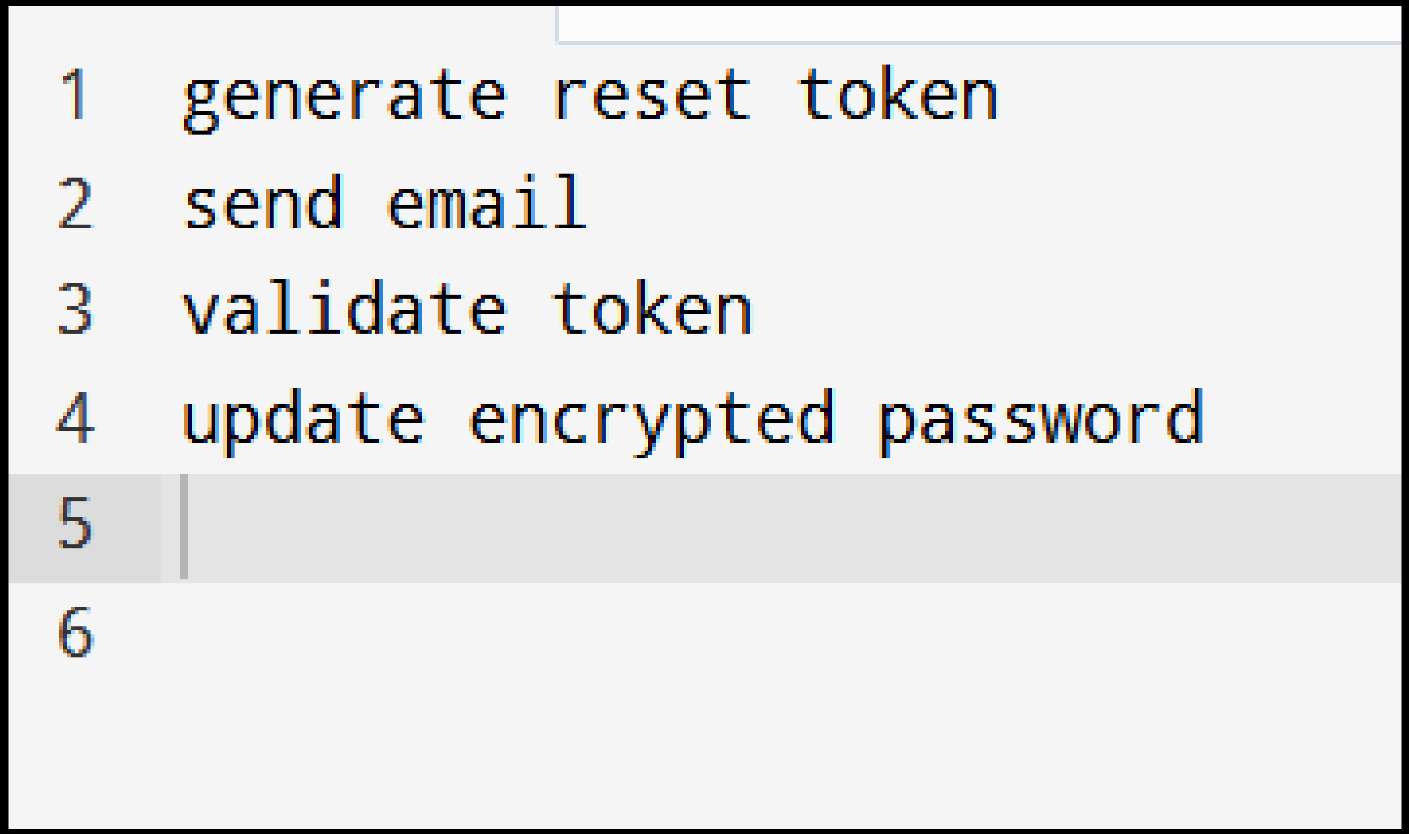
- Stateless authentication
- Scalable
- Secure API access

FORGOT & RESET PASSWORD (SECURITY)

Purpose

- Recover access safely
- Avoid password exposure

Pseudocode



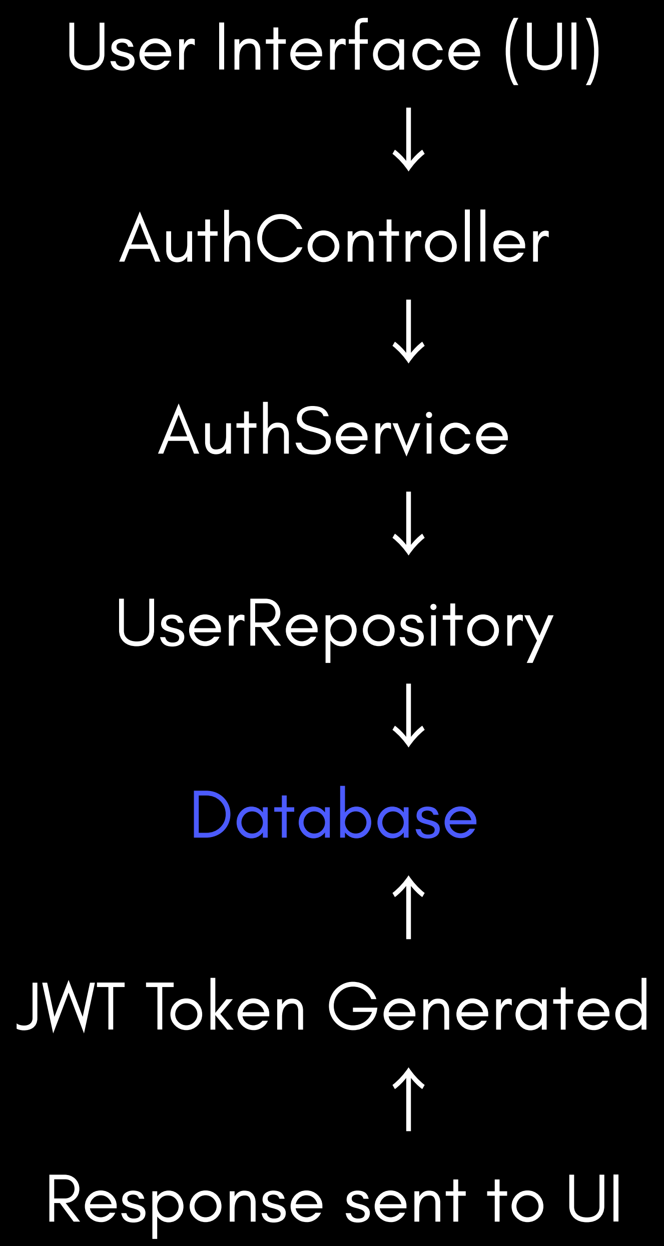
```
generate reset token
send email
validate token
update encrypted password


```

MODULE 1 – SUMMARY & STRENGTHS

Key Strengths

- Secure authentication
- Encrypted password storage
- Token-based authorization
- Industry-standard architecture

FLOW



User Interface (UI)
↓
AuthController
↓
AuthService
↓
UserRepository
↓
Database
↑
JWT Token Generated
↑
Response sent to UI

MODULE 2 – PRODUCT & INVENTORY MANAGEMENT

Objective

- Manage products
- Track stock accurately
- Enforce business rules
- Prevent inventory inconsistency

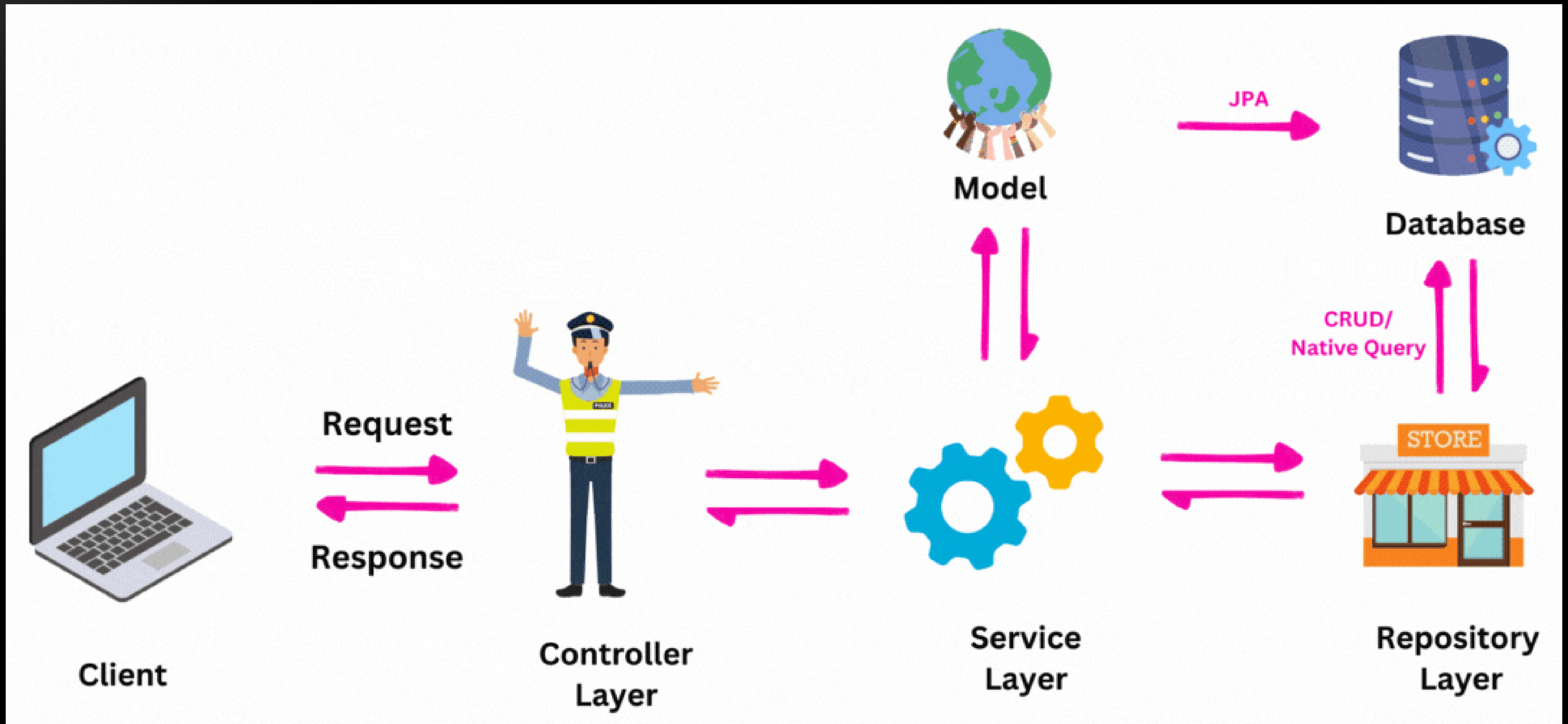
Problems Without Proper Inventory Logic

- Duplicate products
- Negative stock
- Manual errors
- No tracking or accountability

Solution

- Centralized stock control
- Validation before update
- Automatic logging and alerts

ARCHITECTURE



Architecture Flow

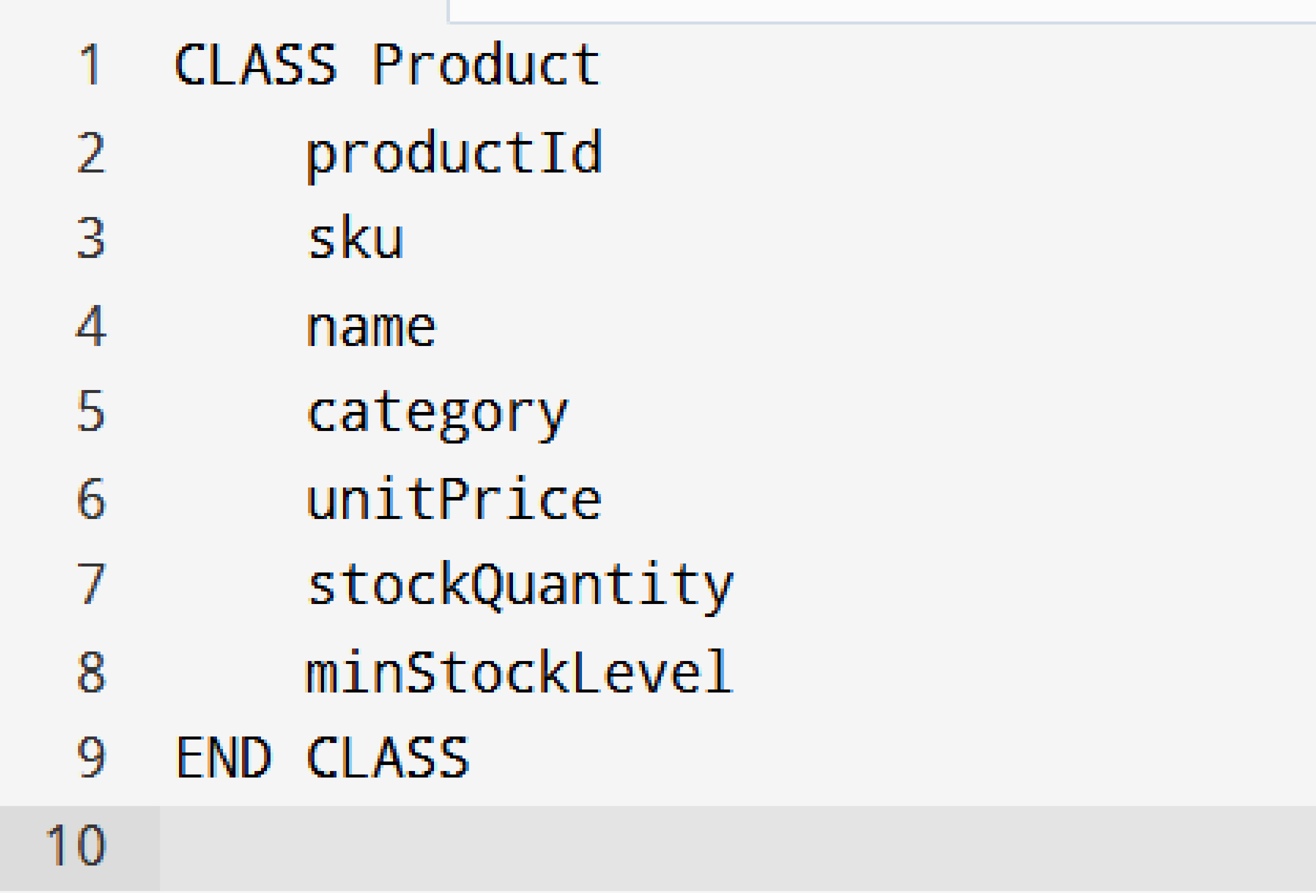
UI → ProductController → ProductService → ProductRepository → Database

PRODUCT ENTITY DESIGN

Purpose

- SKU ensures uniqueness
- stockQuantity is dynamic
- minStockLevel triggers alerts

Product Class (Entity)



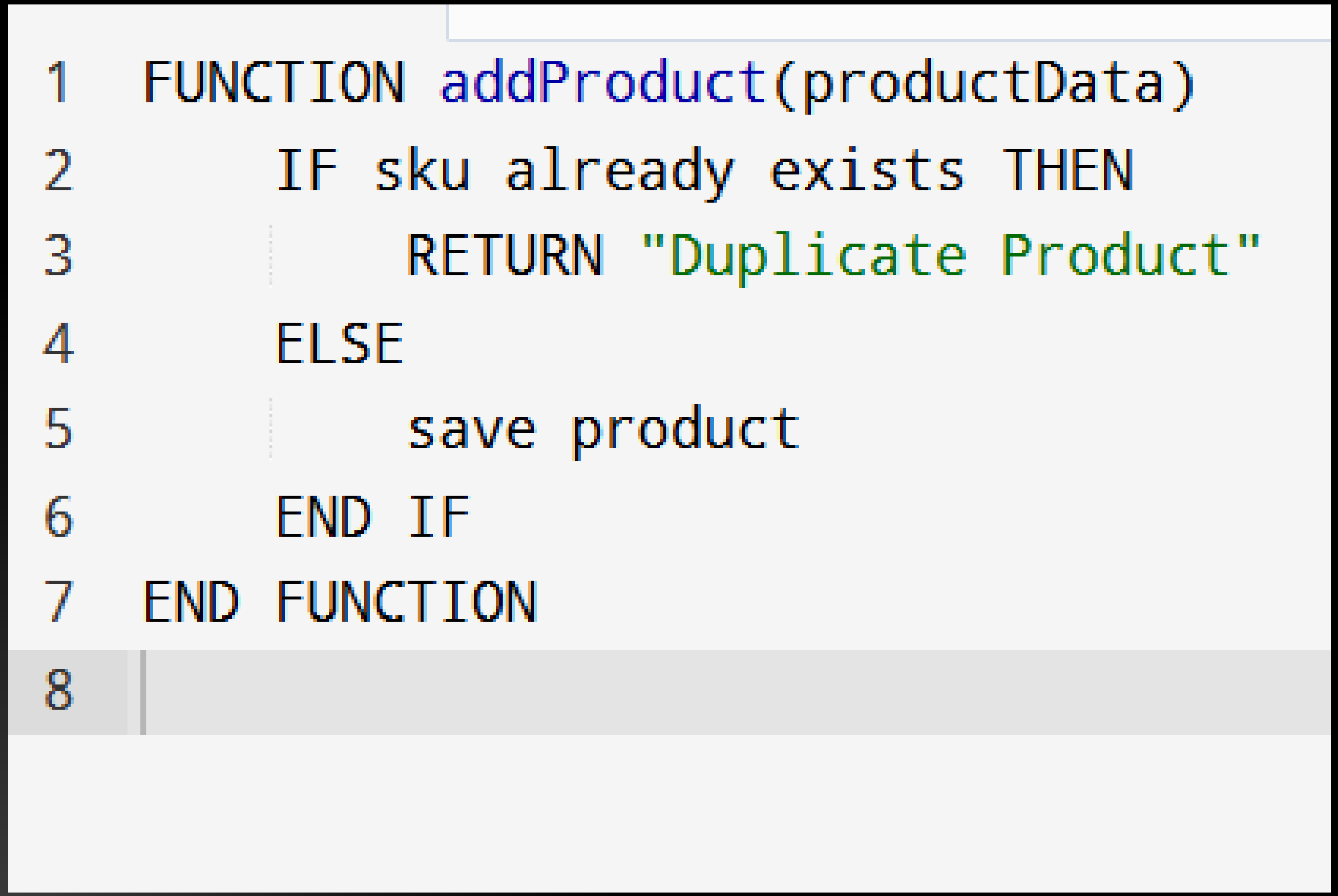
```
CLASS Product
    productId
    sku
    name
    category
    unitPrice
    stockQuantity
    minStockLevel
END CLASS
```

ADD PRODUCT – BUSINESS LOGIC

Purpose

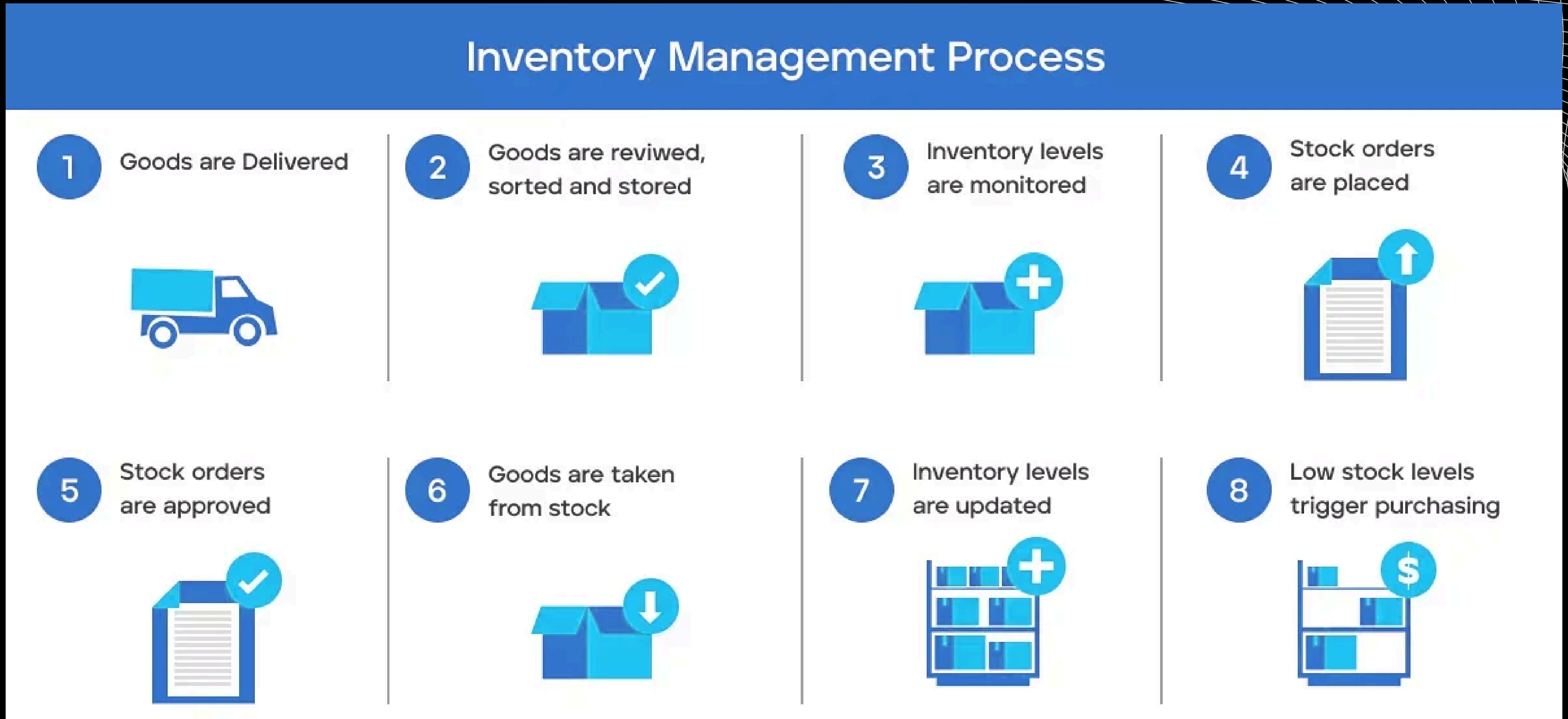
- Add new product safely
- Avoid duplicates

Pseudocode



```
FUNCTION addProduct(productData)
    IF sku already exists THEN
        RETURN "Duplicate Product"
    ELSE
        save product
    END IF
END FUNCTION
```

STOCK IN OPERATION



Inventory Management Process

1 Goods are Delivered

2 Goods are reviwed, sorted and stored

3 Inventory levels are monitored

4 Stock orders are placed

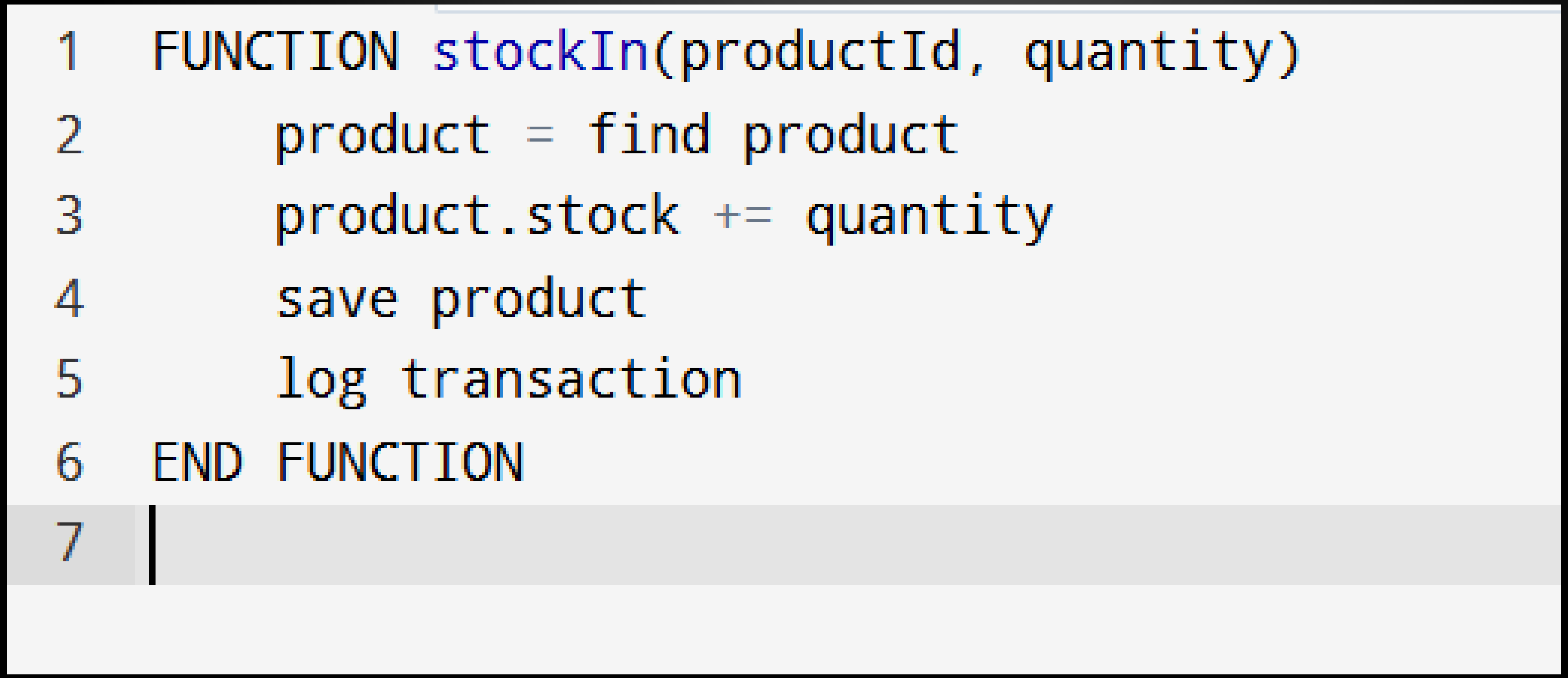
5 Stock orders are approved

6 Goods are taken from stock

7 Inventory levels are updated

8 Low stock levels trigger purchasing

Pseudocode



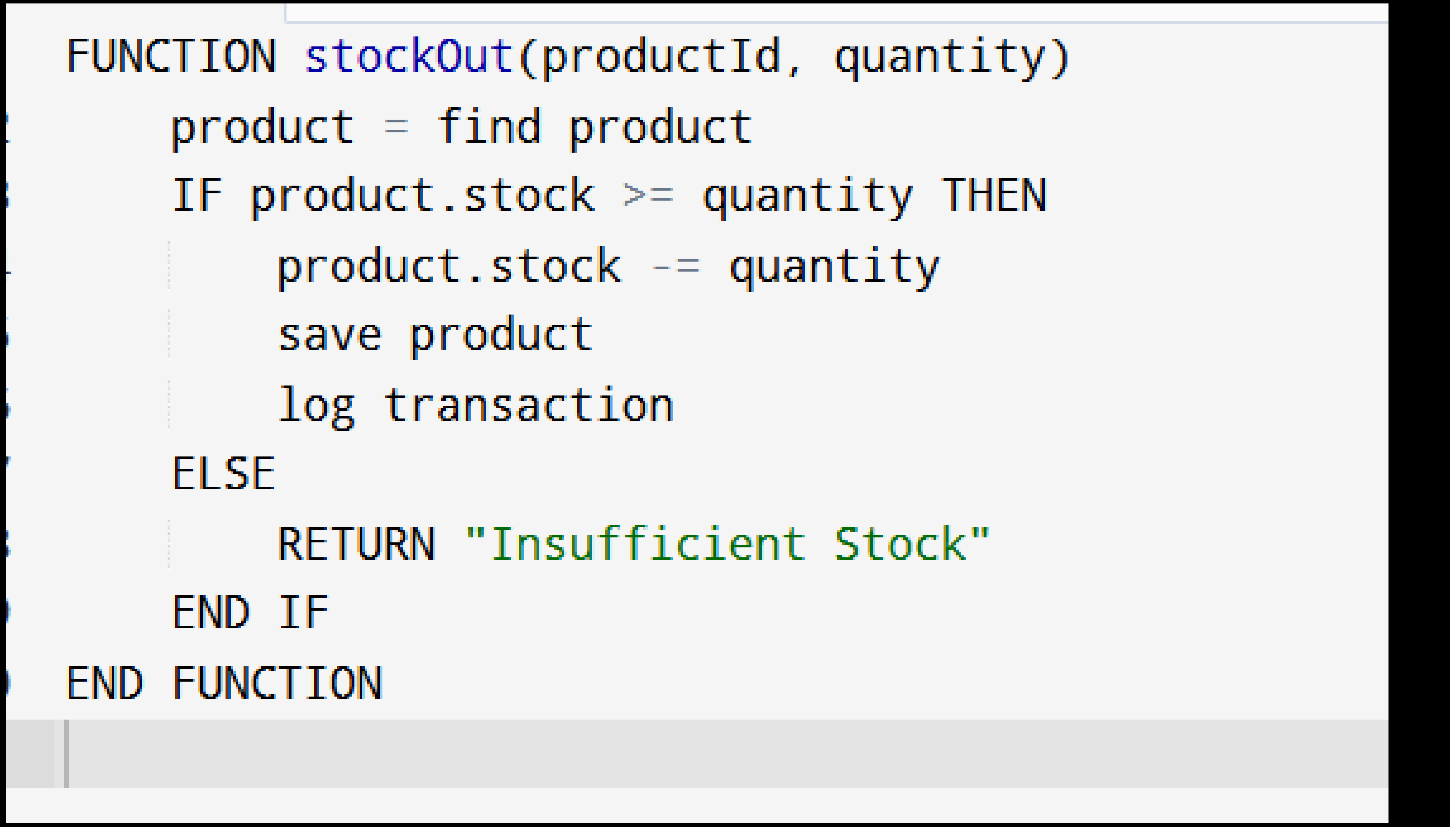
```
FUNCTION stockIn(productId, quantity)
    product = find product
    product.stock += quantity
    save product
    log transaction
END FUNCTION
```

Explanation

- Used during purchase or restocking
- No restriction on upper limit

STOCK OUT OPERATION

Pseudocode



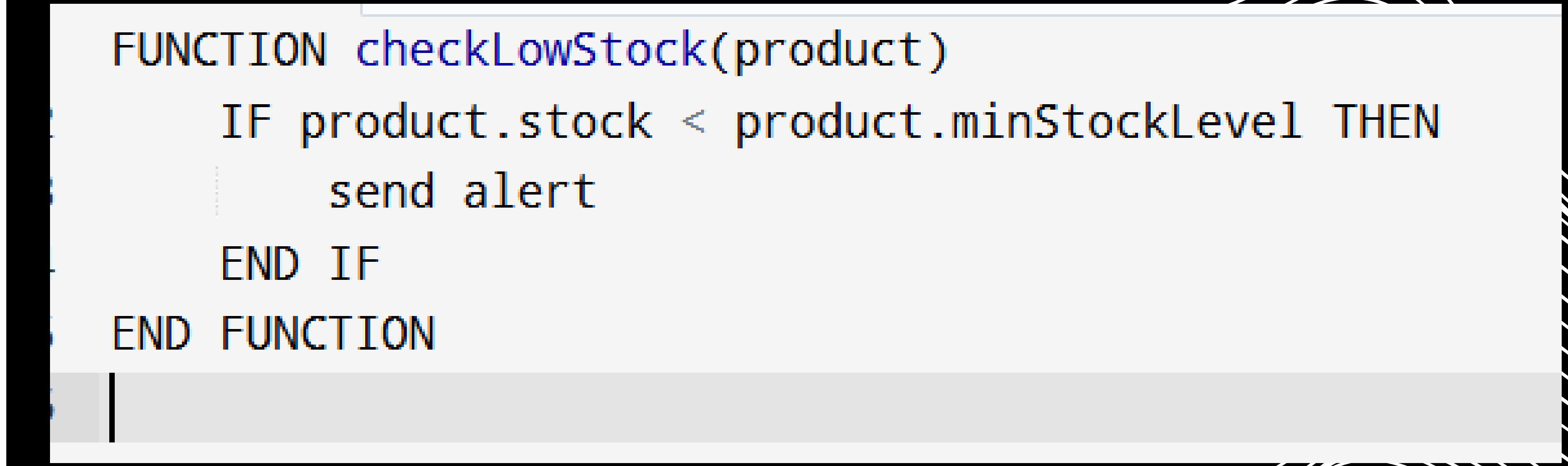
```
FUNCTION stockOut(productId, quantity)
    product = find product
    IF product.stock >= quantity THEN
        product.stock -= quantity
        save product
        log transaction
    ELSE
        RETURN "Insufficient Stock"
    END IF
END FUNCTION
```

ALERT TRIGGERING

Purpose

- Prevents stock-out
- Reduces manual monitoring
- Improves efficiency

Pseudocode



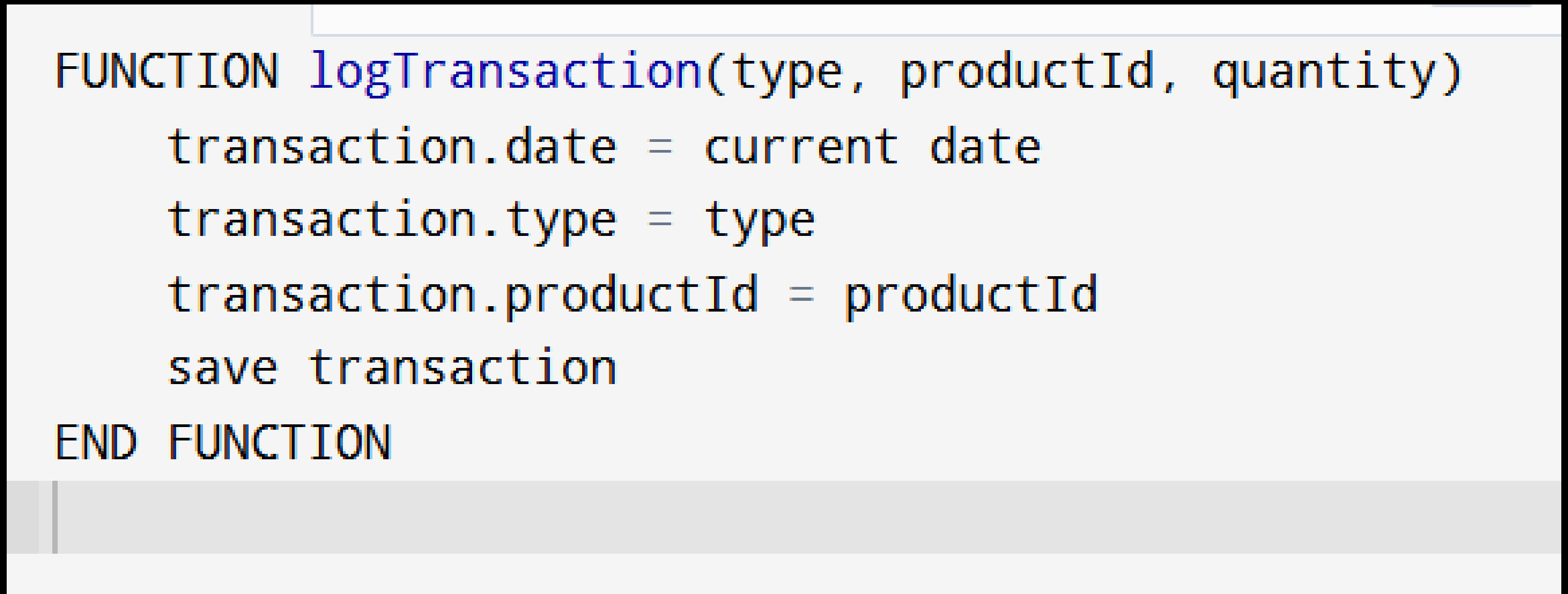
```
FUNCTION checkLowStock(product)
    IF product.stock < product.minStockLevel THEN
        send alert
    END IF
END FUNCTION
```

TRANSACTION LOGGING

Purpose

- Full audit trail
- Transparency
- Report generation

Pseudocode

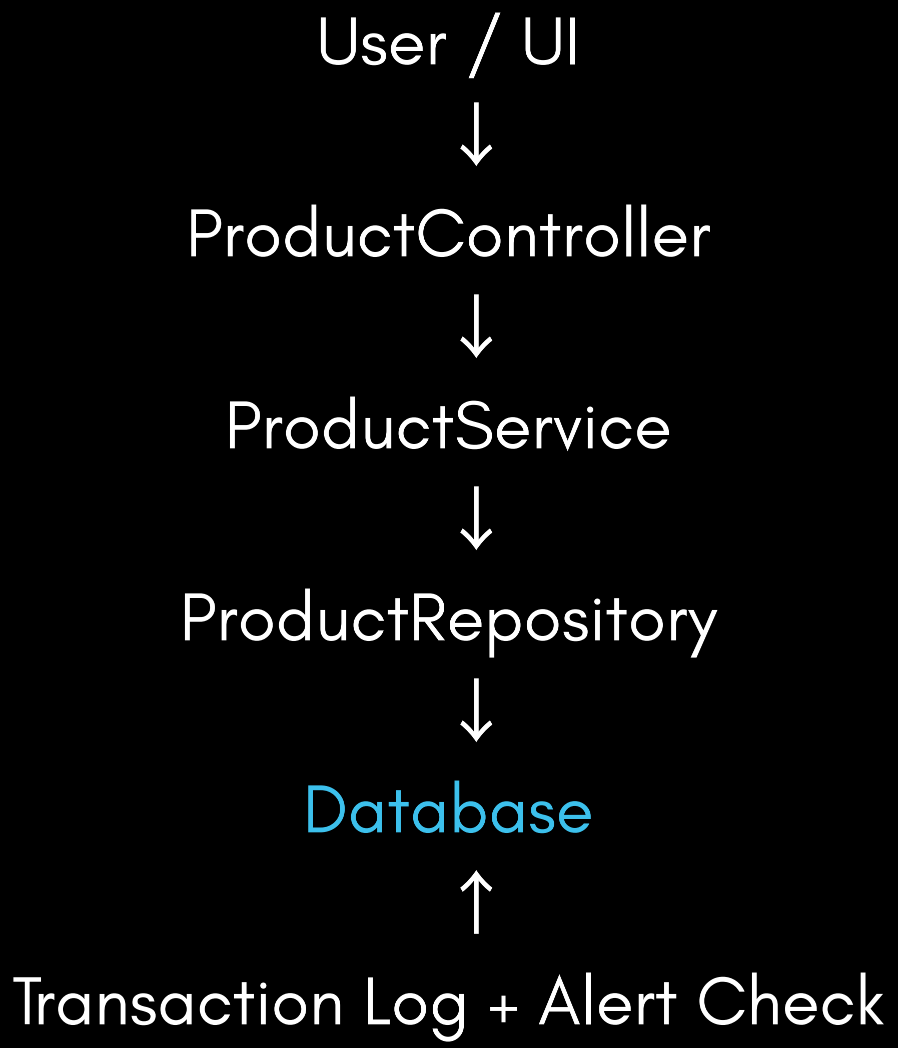


```
FUNCTION logTransaction(type, productId, quantity)
    transaction.date = current date
    transaction.type = type
    transaction.productId = productId
    save transaction
END FUNCTION
```

SUMMARY & STRENGTHS

Key Strengths

- Centralized business logic
- Validation before updates
- Automatic alerts
- Complete audit trail



User / UI
↓
ProductController
↓
ProductService
↓
ProductRepository
↓
Database
↑
Transaction Log + Alert Check

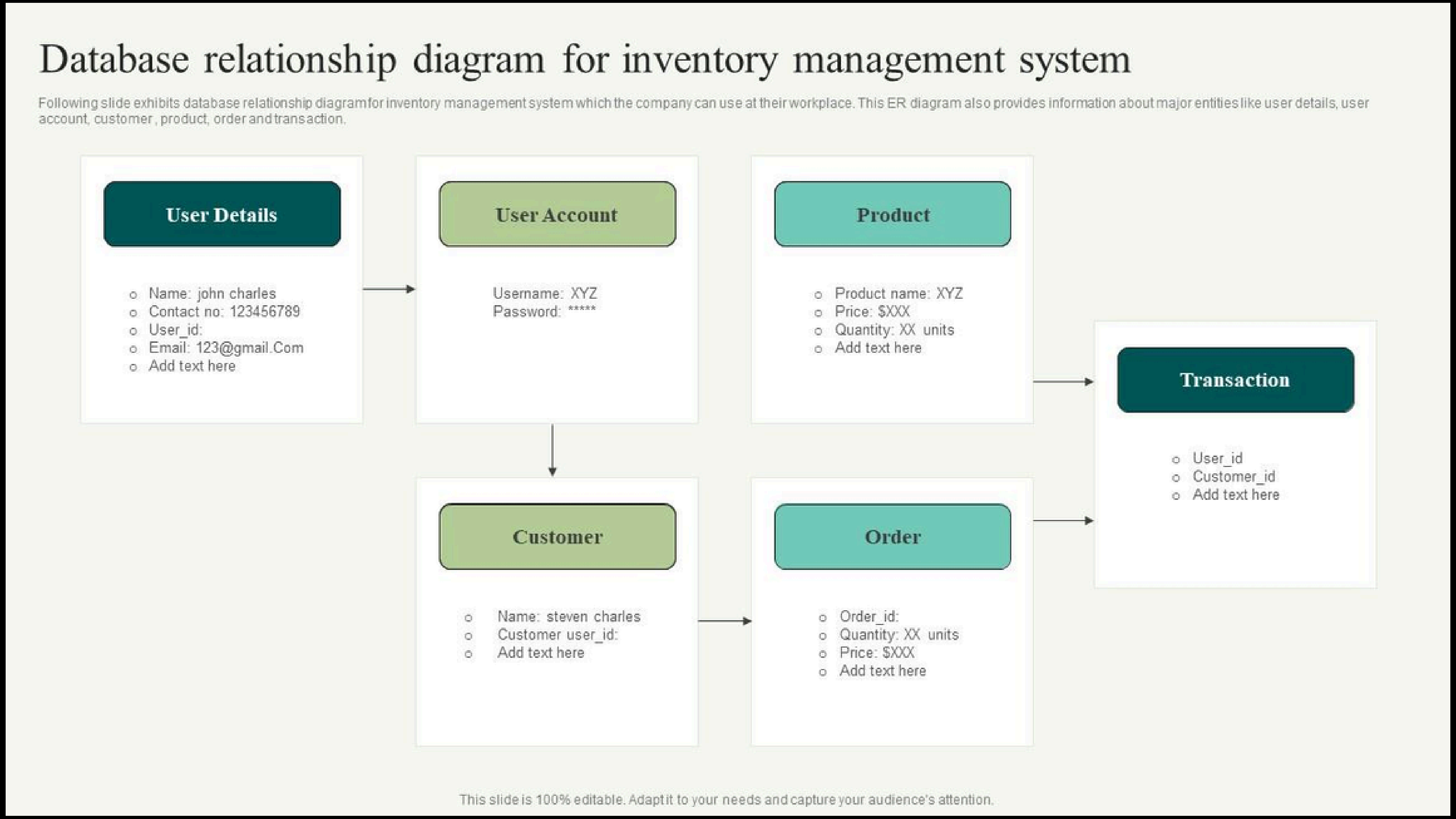
INVENTRA – DATABASE SYSTEM

DATABASE TYPE

Relational Database Management System (RDBMS)

WHY A DATABASE IS REQUIRED

- Persistent data storage
- Secure user information
- Accurate stock tracking
- Transaction history & auditing
- Report generation



Database relationship diagram for inventory management system

Following slide exhibits database relationship diagram for inventory management system which the company can use at their workplace. This ER diagram also provides information about major entities like user details, user account, customer, product, order and transaction.

User Details

- Name: john charles
- Contact no: 123456789
- User_id:
- Email: 123@gmail.Com
- Add text here

User Account

Username: XYZ
Password: *****

Product

- Product name: XYZ
- Price: $XXX
- Quantity: XX units
- Add text here

Transaction

- User_id
- Customer_id
- Add text here

Customer

- Name: steven charles
- Customer user_id:
- Add text here

Order

- Order_id:
- Quantity: XX units
- Price: $XXX
- Add text here

This slide is 100% editable. Adapt it to your needs and capture your audience's attention.

USER TABLE (Authentication Module)

Field	Description
user_id (PK)	Unique user ID
username	Unique login name
password	Encrypted password
email	User email
role	Admin / Staff
reset_token	Password reset token

PRODUCT TABLE (Inventory Module)

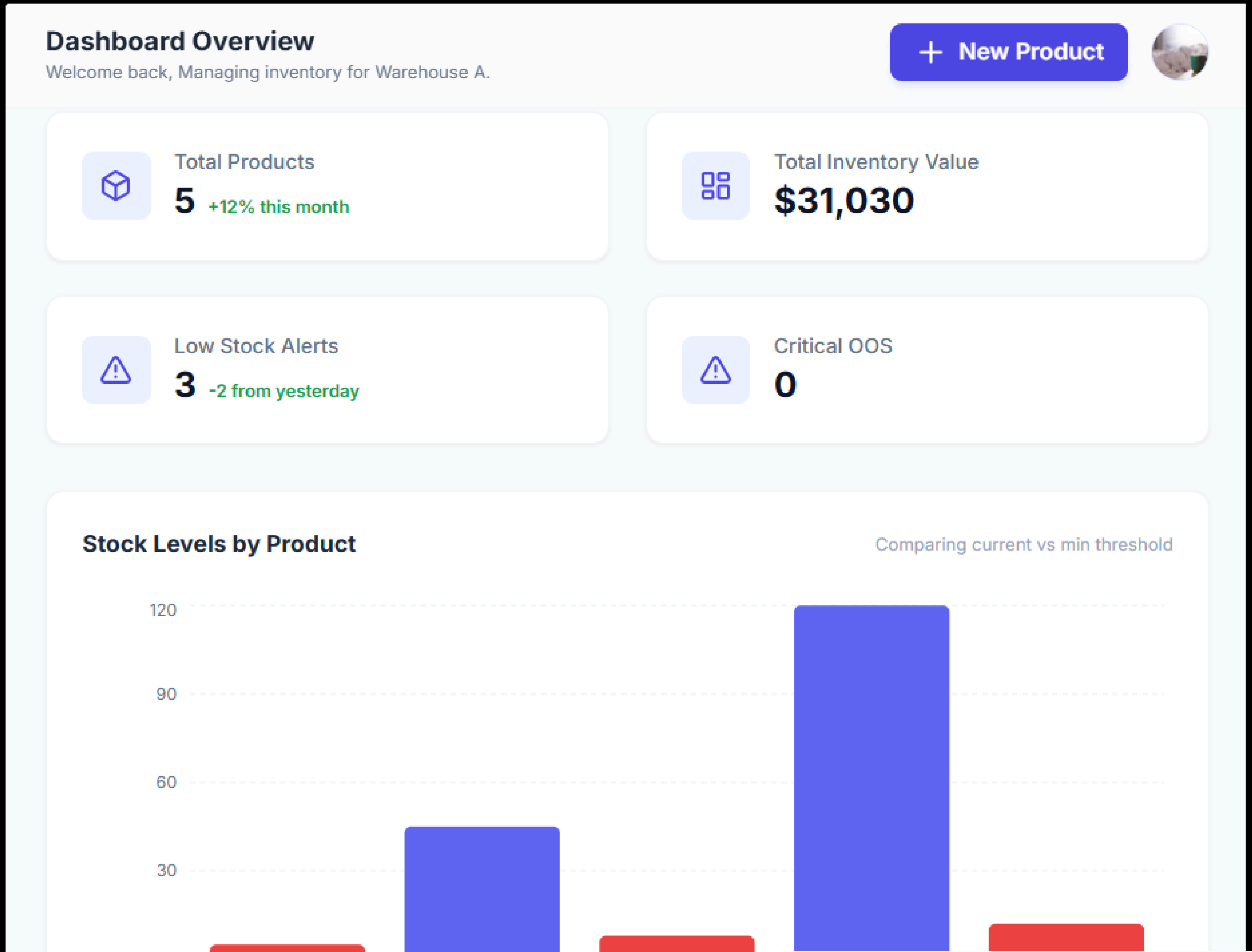
Field	Description
product_id (PK)	Unique product ID
sku	Unique product code
name	Product name
category	Product category
unit_price	Price per unit
stock_quantity	Current stock
min_stock_level	Alert threshold

TRANSACTION TABLE (Audit & History)

Field	Description
transaction_id (PK)	Unique transaction ID
product_id (FK)	Related product
transaction_type	STOCK_IN / STOCK_OUT
quantity	Quantity changed
transaction_date	Date & time
performed_by	User ID

ALERT TABLE (Optional – Alerts Module)

Field	Description
alert_id (PK)	Alert ID
product_id (FK)	Related product
alert_type	LOW_STOCK
alert_status	Active / Resolved
created_at	Timestamp



Dashboard Overview

Welcome back, Managing inventory for Warehouse A.

Total Products
5 +12% this month

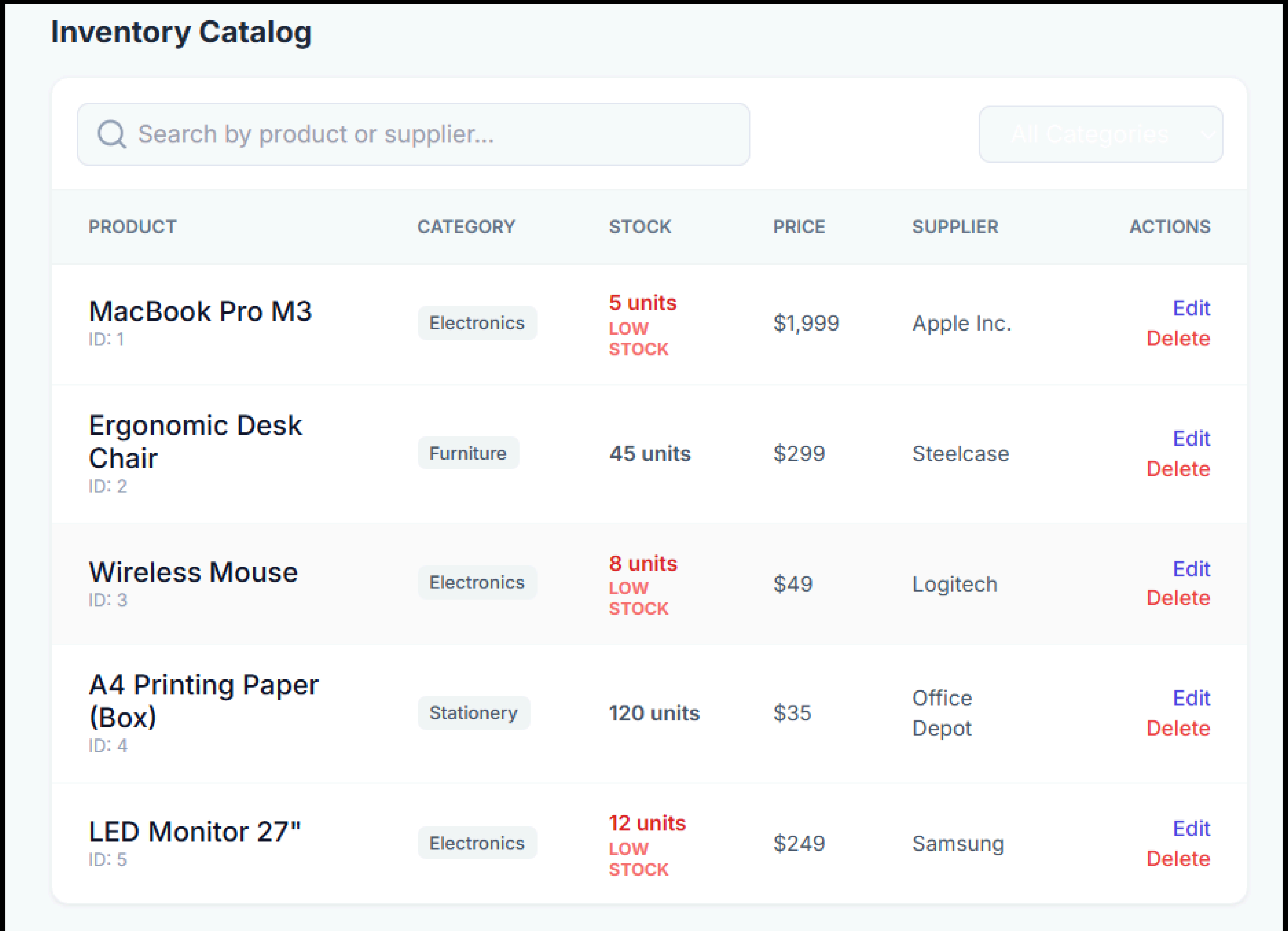
Total Inventory Value
$31,030

Low Stock Alerts
3 -2 from yesterday

Critical OOS
0

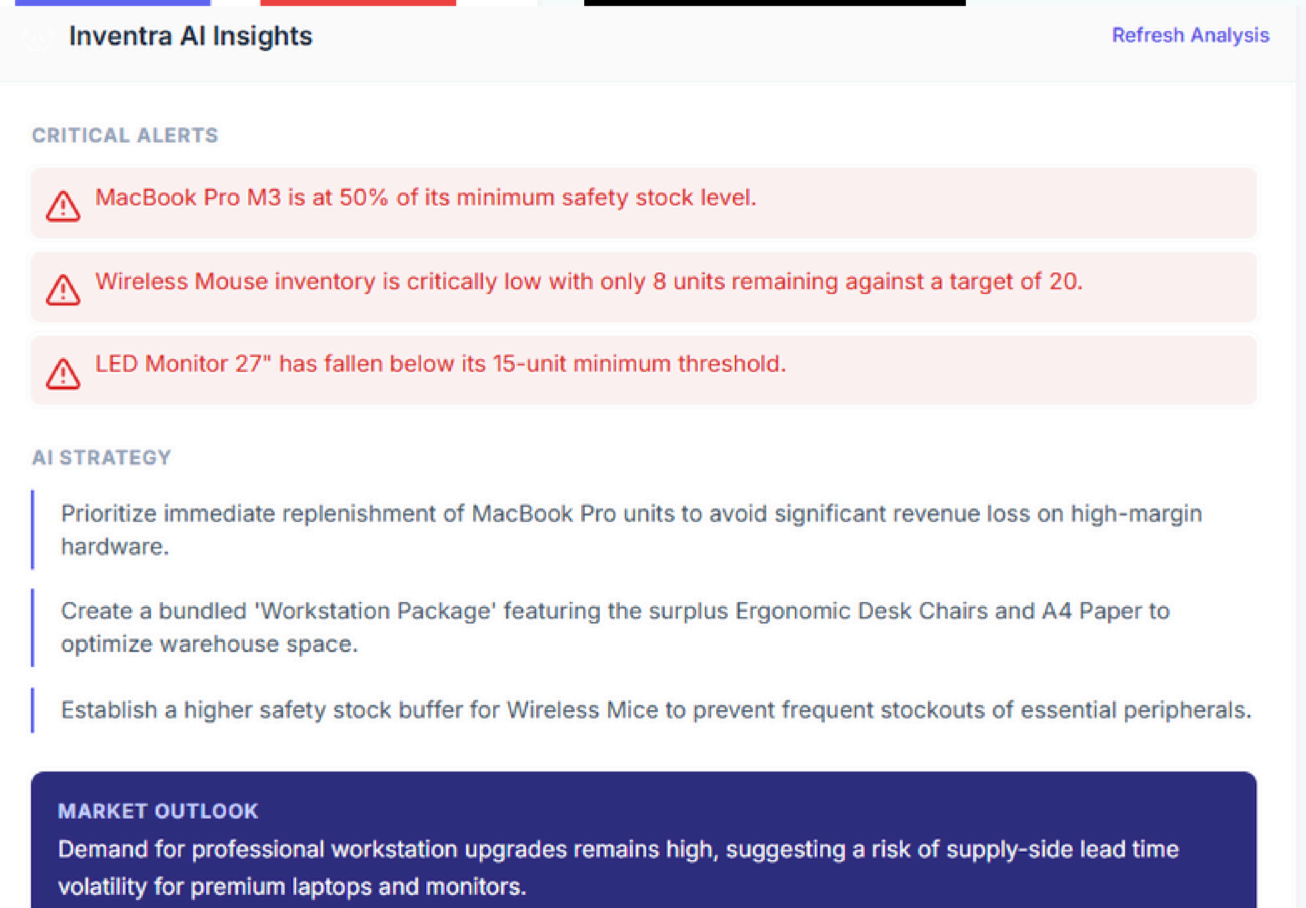
Stock Levels by Product

Comparing current vs min threshold



Inventory Catalog

PRODUCT	CATEGORY	STOCK	PRICE	SUPPLIER	ACTIONS
MacBook Pro M3 ID: 1	Electronics	5 units LOW STOCK	$1,999	Apple Inc.	Edit Delete
Ergonomic Desk Chair ID: 2	Furniture	45 units	$299	Steelcase	Edit Delete
Wireless Mouse ID: 3	Electronics	8 units LOW STOCK	$49	Logitech	Edit Delete
A4 Printing Paper (Box) ID: 4	Stationery	120 units	$35	Office Depot	Edit Delete
LED Monitor 27" ID: 5	Electronics	12 units LOW STOCK	$249	Samsung	Edit Delete



Inventra AI Insights

CRITICAL ALERTS

⚠ MacBook Pro M3 is at 50% of its minimum safety stock level.

⚠ Wireless Mouse inventory is critically low with only 8 units remaining against a target of 20.

⚠ LED Monitor 27" has fallen below its 15-unit minimum threshold.

AI STRATEGY

Prioritize immediate replenishment of MacBook Pro units to avoid significant revenue loss on high-margin hardware.

Create a bundled 'Workstation Package' featuring the surplus Ergonomic Desk Chairs and A4 Paper to optimize warehouse space.

Establish a higher safety stock buffer for Wireless Mice to prevent frequent stockouts of essential peripherals.

MARKET OUTLOOK

Demand for professional workstation upgrades remains high, suggesting a risk of supply-side lead time volatility for premium laptops and monitors.

DATABASE RELATIONSHIPS

User (1) ———— (M) Transactions
Product (1) ———— (M) Transactions
Product (1) ———— (M) Alerts

DATABASE SUPPORTS BUSINESS LOGIC

1. Reduce stock in products table
2. Insert record in transactions table
3. Check min_stock_level
4. Insert alert if required

DATABASE SECURITY & RELIABILITY

- Encrypted passwords
- Unique constraints (username, SKU)
- Transactions ensure atomicity
- Prevents partial updates

DATABASE SYSTEM – SUMMARY

Strengths

- Structured relational design
- Secure and consistent data storage
- Full audit trail
- Scalable for enterprise use

Thankyou