

# Inventory Management

Presented by Aman Kumar  
Roll - 22020107008

# Problem Statement

Efficient inventory management is crucial for businesses to track stock levels, reduce wastage, and meet customer demands. However, small and medium-sized businesses (SMBs) often lack access to sophisticated inventory management tools due to their high cost and complexity. This results in manual processes prone to errors, inefficiencies, and delays, negatively impacting productivity and profitability.

Existing inventory management solutions are either too costly, overly complex, or lack the specific features required by SMBs. Businesses need a lightweight, user-friendly, and cost-effective software solution to track and manage inventory effectively.

# Proposed Solution

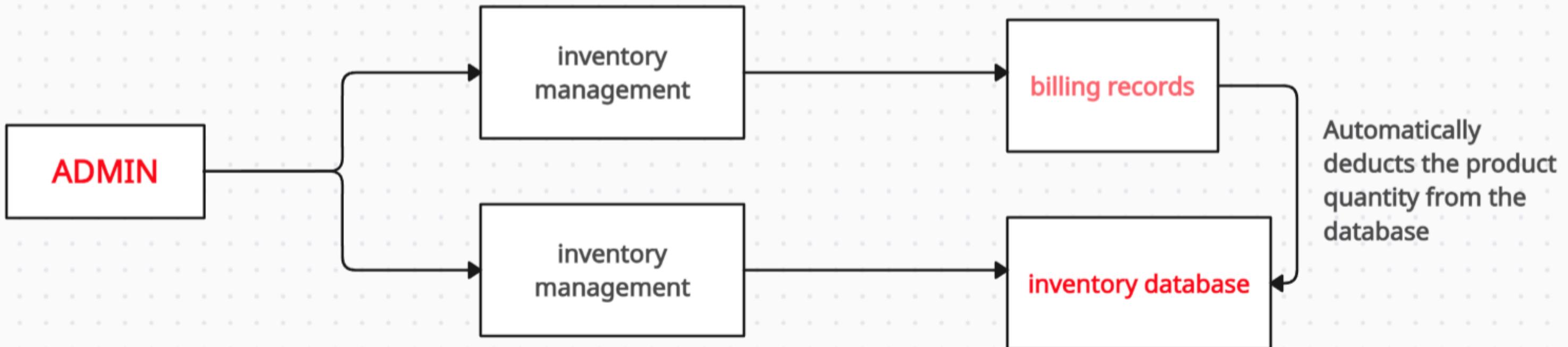
A lightweight inventory management system designed specifically for SMBs that:

- **Is modular:** Core inventory features can be enhanced with optional modules (e.g., billing, supplier management).
- **Provides real-time updates:** Tracks stock changes and generates alerts dynamically.
- **Is scalable:** Accommodates growth in the business and inventory size.
- **Is accessible:** Operates on the cloud or locally, depending on user preferences.



# High-level Workflow

- Admin logs into the system.
- Adds or modifies inventory data.
- Generates bills during sales, which:
- Automatically adjusts inventory levels in the database.
- Records the transaction in the billing system.
- The inventory and billing modules synchronize data in real-time.
- Reports and alerts are generated for better monitoring and decision-making.



# Tech Stack

## **MongoDB (Database):**

- MongoDB is used to store all the data, like stock details and bills, in a structured way.
- It is a NoSQL database that works well with JavaScript, making it easy to save and retrieve data.

## **Express.js (Backend Framework):**

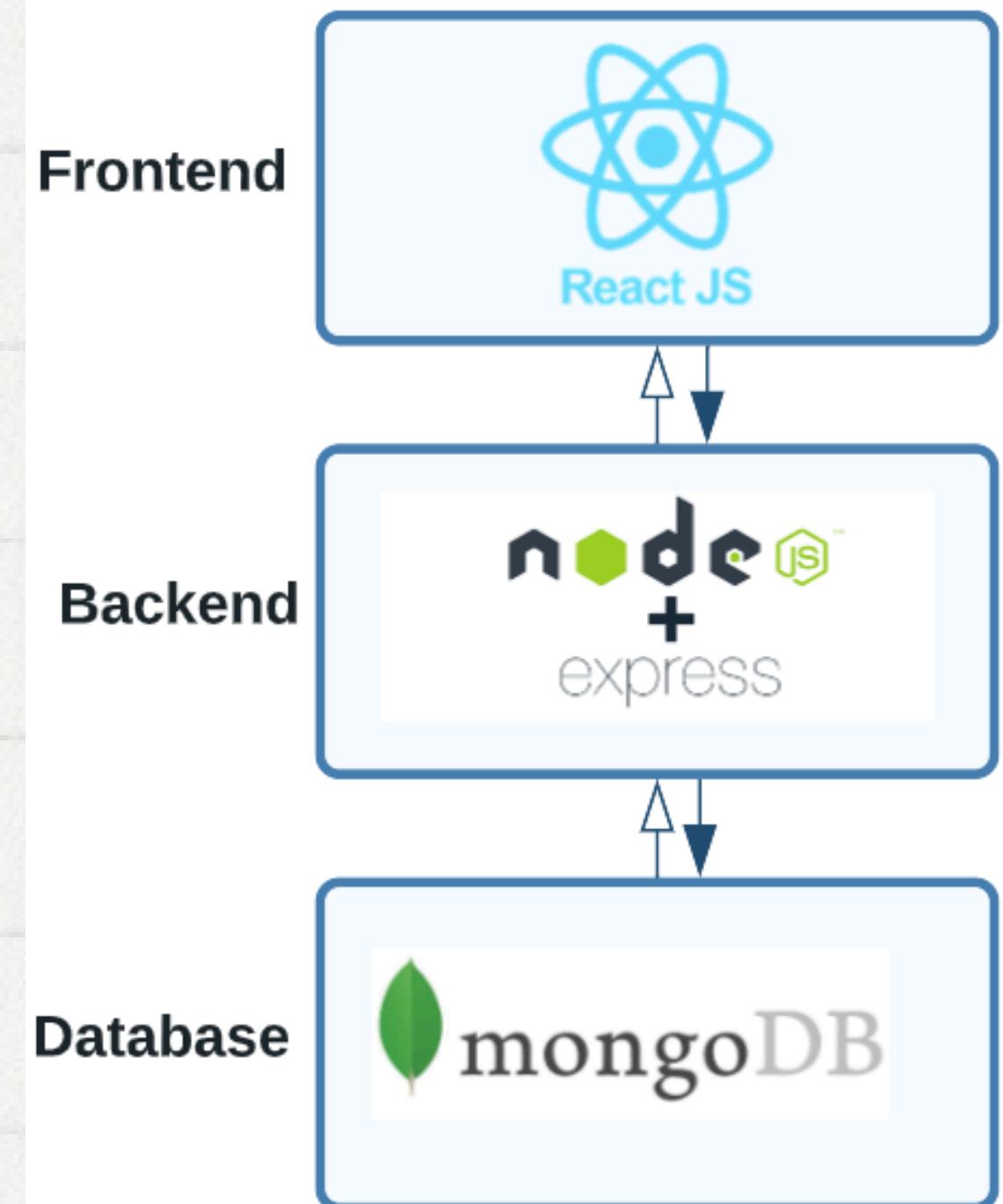
- Express handles the logic and routes for the server.
- It helps connect the frontend to the database through APIs.

## **React.js (Frontend Framework):**

- React is used to create the user interface where shopkeepers can view and manage stocks and bills.
- It provides a smooth, dynamic, and interactive experience.

## **Node.js (Runtime Environment):**

- Node.js runs the backend code and allows us to use JavaScript for server-side development.
- It connects all the parts of the application seamlessly.



# Low-level Design

## Adding or Modifying Inventory Input:

- Admin enters product details (e.g., product name, quantity, price, etc.).

## Processing:

- Frontend validates the input (e.g., no negative quantities).
- Data is sent to the backend API.

## Database Update:

- Backend verifies the product ID.

### 1. If the product exists:

- Updates the stock level or price in the inventory database.

### 2. If the product is new:

- Adds a new record to the inventory database.

## Feedback to Admin:

- The admin is notified of the successful addition/modification with a message on the UI.

# Generating a Bill

## Input:

- Admin selects products being sold and enters the quantity for each.

## Processing:

- Frontend calculates the total bill (price × quantity for each product).
- Sends the selected products and quantities to the backend API.

## Backend Workflow:

### 1. For each product:

- Fetches the product details from the inventory database.
- Verifies that the requested quantity is available in stock.
- Deducts the quantity from the stock.

### 2. Creates a new entry in the billing records database with:

- List of products sold, quantities, total price, and timestamp.

## Feedback to Admin:

- A printable bill is generated and displayed on the UI.

# Working demo

## Welcome to Shop Manager

Total Products: 4

Total Bills: 4

Total Categories: 5

[View Stocks](#) [Add Stock](#) [Create Bills](#) [View Bills](#)

## Welcome to Shop Manager

Total Products: 4

Total Categories: 5

[View Stocks](#) [Add Stock](#) [Create Bills](#) [View Bills](#)

### Add a New Product

Product Name \*: Guava

Price \*: 500

Category \*: Food

Stock \*: 250

Description: Guava Truck arrived on 27th December 2022

[Save](#) [Cancel](#)

## Create Bill

Customer Name: Rakesh

Select a Product: Guava

Quantity: 10

Price: 150

[Add Product](#)

**Total Amount: 1500**

[Submit Bill](#)