Technical Report: Shoes Inventory Management Application

Introduction

The Shoes Inventory Management Application is a sophisticated system designed to streamline the management of shoe inventories across multiple stores. Built with React on the client-side, Node.js on the server-side with Express and GraphQL for the API, and using socket.io for real-time updates, this application facilitates seamless user interaction and efficient inventory monitoring.

Application Overview

Features

1. **Real-time Updates:**
   - Upon completing a sale, the UI updates dynamically without requiring a page reload. Updated inventory data is pushed from the server to all connected clients using socket.io, ensuring immediate visibility of changes.

2. **Client-Side Routes:**
   - Landing Page: Displays a list of stores as cards. Upon selecting a store, the store's inventory list is shown. Users can initiate a purchase by specifying the number of units they wish to buy.

   - Monitor Stock: Provides a consolidated view of all stores' inventory statuses in a single table. Filters by store name or shoe model can be applied to refine the displayed data.

   - Reports: Presents graphical insights including donut, bar, and pie charts:
     - Top shoes sold across all stores (including least sold).
     - Top performing stores based on sales metrics.
     - Top shoes sold by each store.

Workflow Example: Sales Update

- Scenario: User initiates a purchase of 5 units of a specific shoe from Store A.
- Action:
  - User selects Store A from the landing page and navigates to the shoe inventory.

- User inputs "5" units in the purchase quantity field or uses buttons to adjust the quantity.
- Upon completing the purchase, the server updates the sold shoe inventory and logs the transaction in the order history using socket.io for real-time updates.
- All connected clients are notified in real-time and display the updated inventory for all stores without refreshing the page.

Technical Architecture

Frontend (React)

- Components and State Management:
- Components are organised to support the three main routes: Landing Page, Monitor Stock, and Reports.
- State management ensures that components reflect real-time data changes using React's state and props mechanism.
- UI components include interactive elements such as input fields, buttons for quantity adjustment, and dynamic data visualisation for reports.

Backend (Node.js with Express and GraphQL)

- Server Technology:
- Implemented in TypeScript on Node.js with Express for robust backend functionality.
- The API is built with GraphQL, providing a flexible and efficient way to query and manipulate data.
- socket.io facilitates real-time communication between server and clients, enabling instant updates of inventory changes without polling.

Real-time Updates

- Socket.io Integration:

- Utilises socket.io to establish WebSocket connections between clients and the server.
- Enables bidirectional communication for immediate propagation of inventory changes to connected clients upon successful transaction completion.

GraphQL API

- Overview:
  - GraphQL is used to construct a flexible and efficient API for the backend.
  - Enables clients to request exactly the data they need, reducing over-fetching and improving performance.
  - Facilitates complex queries and mutations required for managing shoe inventories and sales transactions.

Implementation Details

Database Schema

- Tables:
  - Shoes: Stores shoe information including model, quantity, and store association.
  - Stores: Contains store details.
  - OrderHistory: Logs sales transactions including shoe model, quantity sold, store ID, and timestamp.

Client-Side Logic

- Routing and Navigation:
  - React Router manages navigation between different routes and components.
  - Private routes ensure access control for authenticated users.

Server-Side Logic

- Socket.io Integration:
  - Establishes and manages WebSocket connections for real-time updates of inventory and sales data.
  - Updates are broadcasted to all connected clients upon changes to ensure synchronized views.

Monitoring Stock Route

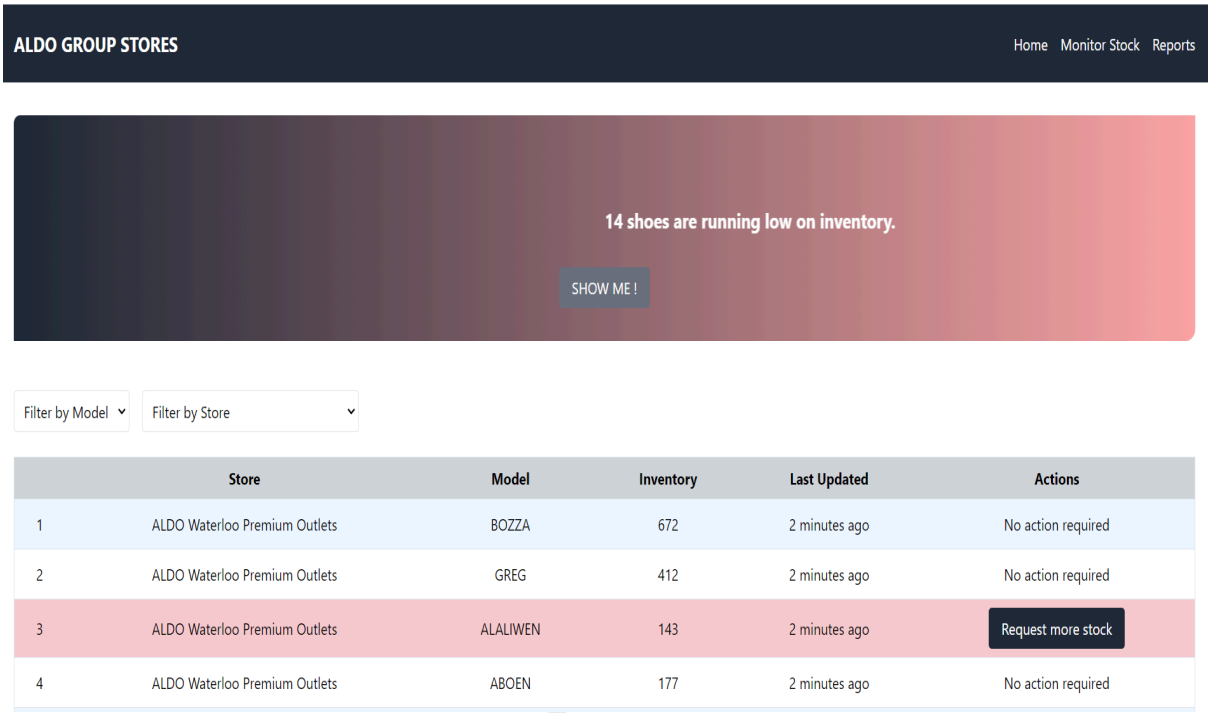
- Overview:
  - Displays a unified table showing inventory status for all stores.
  - Allows filtering by store name or shoe model to narrow down displayed data.
  - Supports real-time updates via socket.io, ensuring immediate visibility of inventory changes across all connected clients.

Conclusion

The Shoes Inventory Management Application leverages modern technologies including React, Node.js with Express, GraphQL, and socket.io to provide a robust and efficient solution for managing shoe inventories across multiple stores. By enabling real-time updates via WebSocket connections, intuitive user interfaces, and comprehensive reporting capabilities, the application enhances operational efficiency and empowers businesses with actionable insights for optimising sales and inventory management strategies.

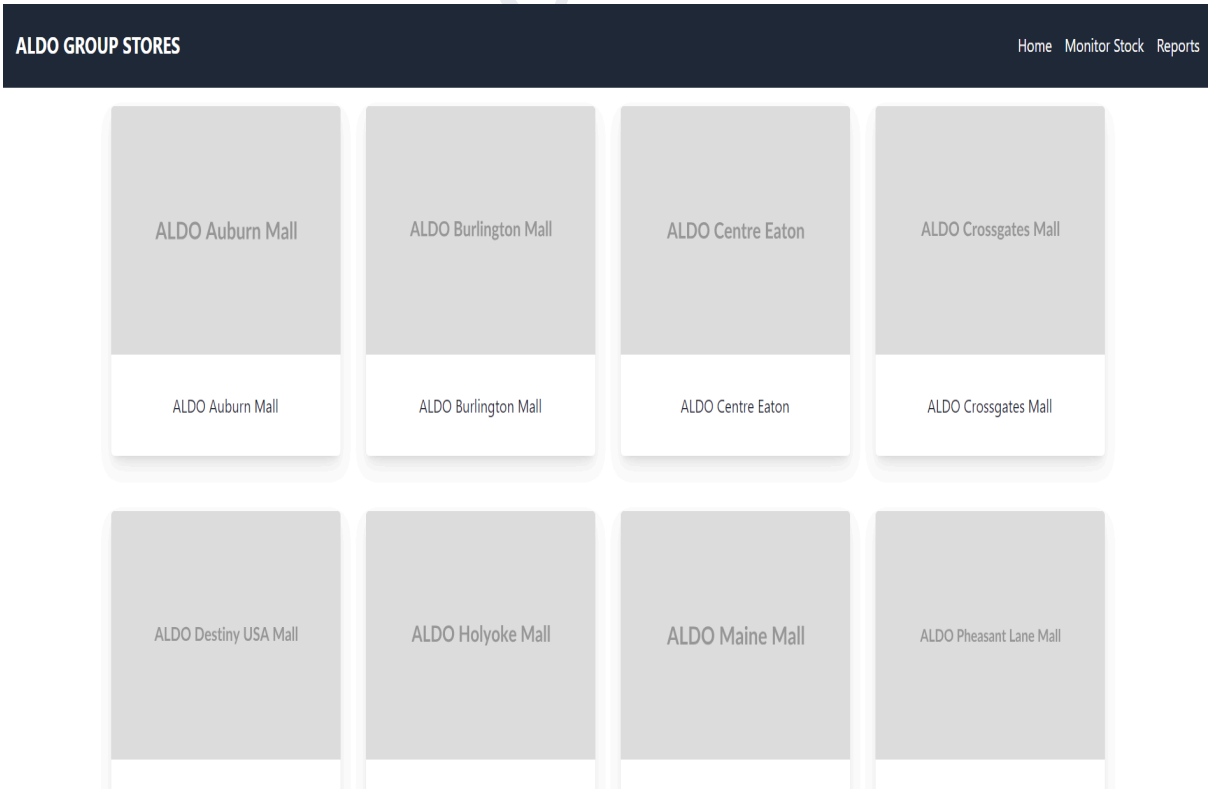
Appendix

Shows a consolidated table of all stores' inventory statuses with live updates



	Store	Model	Inventory	Last Updated	Actions
1	ALDO Waterloo Premium Outlets	BOZZA	672	2 minutes ago	No action required
2	ALDO Waterloo Premium Outlets	GREG	412	2 minutes ago	No action required
3	ALDO Waterloo Premium Outlets	ALALIWEN	143	2 minutes ago	Request more stock
4	ALDO Waterloo Premium Outlets	ABOEN	177	2 minutes ago	No action required

Displays a list of stores as cards



Displays the selected store's shoe inventory