🧾 Fizzy Crave – Inventory Management System

Program: BCA (AI & DS), SOET
Mentor: Aman Jatain

👨‍💻 Team Members

Name	Role	Roll No
Akanksha Kumari	Project Manager	2401201162
Preeti Chaurasiya	UI/UX Designer	2401730262
Payal Tanwar	Frontend Developer	2401201191
Khushi Chauhan	Backend Developer	2401201147

---

1. Introduction

In today's rapidly evolving digital economy, businesses—especially in the **food and beverage (F&B)** industry—face increasing challenges in managing inventory effectively. Manual methods are often time-consuming, error-prone, and difficult to scale. Recognizing these pain points, **Fizzy Crave** was developed as a smart, web-based inventory management system to optimize stock control, track orders, and streamline day-to-day operations.

This project reflects a real-world solution that helps small-scale businesses digitize their inventory process, automate product tracking, reduce waste, and enhance customer satisfaction through prompt stock updates and secure transactions.

---

2. Project Objectives

The key goals of Fizzy Crave include:

- ✅ Eliminating manual errors in stock management
- ✅ Providing a user-friendly interface for both administrators and customers
- ✅ Enabling real-time stock tracking and updates
- ✅ Alerting users about low-stock items to prevent shortages
- ✅ Supporting category-wise product browsing and online ordering
- ✅ Integrating payment processing via third-party gateways (e.g., PayPal)
- ✅ Generating inventory and sales reports for data-driven decision-making

- ✅ Building a responsive, scalable, and secure system

These objectives are aligned with modern inventory best practices that ensure operational efficiency and business growth.

---

3. System Architecture

Fizzy Crave is structured on a **3-tier architecture**, ensuring separation of concerns and smooth scalability:

```
scss
CopyEdit
Presentation Layer (Frontend)   →   HTML, CSS, JS (Browser UI)
Application Layer (Backend)     →   Node.js, Express.js (Business Logic &
Routing)
Data Layer (Database)           →   MongoDB (Inventory, Users, Orders)
```

📌 Key Modules:

- **User Module** – Registration, login, profile
- **Product Module** – Add/edit/delete products by admin
- **Cart & Checkout Module** – Order and payment processing
- **Inventory Module** – Real-time tracking, stock status
- **Report Module** – Inventory and sales data visualization

This modular approach enables fast maintenance and future enhancements.

---

4. Frontend Implementation

The frontend was developed using the latest versions of:

- **HTML5** – For semantic structure
- **CSS3** – For layout, theming, and responsiveness
- **JavaScript** – For dynamic rendering and validation

📱 Key Features:

- Clean homepage with dynamic categories (Snacks, Drinks, Dairy, etc.)
- Product cards with images, prices, and quantity indicators
- Fully responsive layout using Flexbox/Grid
- Cart page for reviewing and modifying orders
- Real-time alerts for low-stock items
- Forms for user registration and inventory entry
- Client-side validation and session control

UX Enhancements:

- Tooltips, hover effects, and modal dialogs for interactivity
- Visual stock indicators (e.g., red for low, green for available)
- Smooth navigation via buttons and links
- Mobile-first design tested across screen sizes

---

5. Backend Implementation

The backend is the brain of the system, built with:

- **Node.js** – Non-blocking, event-driven JavaScript runtime
- **Express.js** – Lightweight web framework for route handling and middleware
- **MongoDB** – NoSQL document-based database for flexible schema design

🔐 Core Functionalities:

- User authentication (password hashing, JWT-based sessions)
- Product CRUD operations (Create, Read, Update, Delete)
- Inventory quantity updates after checkout
- Admin panel access and verification
- API error handling, input validation, and secure access control
- Payment routing logic for third-party integration (PayPal)

The codebase follows RESTful API design principles, ensuring ease of integration and testing.

---

6. Dialogflow Integration

❌ **Not applicable** – This project does not involve a chatbot or NLP interaction.
However, future versions may integrate AI or ML-based modules for **predictive inventory management**, demand forecasting, or voice-enabled input.

---

7. Sample Interaction

➤ Customer Flow:

1. Visits **fizzycrave.onrender.com**
2. Registers or logs in
3. Browses through product categories like *Drinks*, *Snacks*, *Desserts*
4. Adds desired items to cart
5. Proceeds to checkout → redirected to **PayPal**

6. Order confirmed → Inventory updated in real-time

➤ Admin Flow:

1. Logs in to dashboard
2. Views current stock, orders, and users
3. Adds new inventory or updates quantities
4. Generates reports and downloads CSV/JSON files
5. Receives alerts for low inventory

---

8. UI/UX Design Considerations

Design Philosophy: Simple, intuitive, and clean.

Key UX Features:

- **Responsive design**: Works seamlessly on mobile, tablet, and desktop
- **Color coding**: Red for low stock, green for available, gray for inactive
- **Font hierarchy**: Large bold headings, medium-sized content for readability
- **Error & success messages**: Clearly defined for user feedback
- **Security prompts**: Warnings before deleting products or logging out
- **Favicon, title, and meta tags** for brand identity

Tools used for prototyping: **Figma** (UI wireframes), **Canva** (icons & design assets)

---

9. Deployment

The system is hosted on **Render.com**, with CI/CD linked to GitHub for automatic updates.

Hosting Details:

- **Frontend & Backend**: Hosted as a full-stack app on Render
- **Database**: MongoDB Atlas (cloud-hosted)
- **Version Control**: Git & GitHub
- **Environment Variables**: Secured using `.env` file
- **Domain**: mailto:https://fizzycrave.onrender.com

Deployed system is tested for:

- Load performance
- HTTPS encryption
- Form security (e.g., XSS, SQLi prevention)

---

10. Conclusion

Fizzy Crave represents a practical, real-world application of inventory management tailored to the needs of small businesses in the F&B sector. By transitioning from manual tracking to digital automation, businesses can save time, reduce costs, and make informed decisions.

The project fulfills its core objectives by offering a complete inventory management solution—with real-time updates, user interaction, and responsive UI—all within a secure and scalable framework.

✅ Future Enhancements:

- 📱 Mobile app (React Native)
- 🔍 Barcode scanning for products
- 📈 AI-based demand forecasting
- 📊 Advanced analytics dashboard
- 🧾 Invoice and billing system
- 📡 SMS/email notifications for stock alerts