



SweetsNSweets is a full-stack sweet shop management system built using the MERN stack. It allows users to browse and purchase sweets, while administrators can manage inventory through a secure, role-based dashboard.

The project demonstrates real-world full-stack development practices including authentication, authorization, REST APIs, and a modern responsive frontend.

Features

User Features

- User registration and login with JWT authentication\
- Browse all available sweets\
- Search sweets by name\
- Purchase sweets (disabled automatically when stock is zero)\
- Real-time stock updates after purchase

Admin Features

- Admin-only access control\
 - Add new sweets to inventory\
 - Update existing sweet details\
 - Delete sweets from inventory\
 - Secure protected routes and APIs
-

Tech Stack

Frontend

- React (Vite)\
- Tailwind CSS\
- React Router\
- Axios\
- Lazy loading with React.lazy & Suspense

Backend

- Node.js\
- Express.js\
- MongoDB & Mongoose\
- JWT Authentication\

- bcryptjs for password hashing
-

📁 Project Structure

Backend

```
backend/
└── src/
    ├── controllers/
    ├── models/
    ├── routes/
    ├── middlewares/
    └── utils/
    └── server.js
    └── package.json
```

Frontend

```
frontend/
└── src/
    ├── api/
    ├── components/
    ├── pages/
    ├── App.jsx
    └── main.jsx
    └── package.json
```

🔒 Authentication & Authorization

- JWT-based authentication\
 - Tokens stored securely in localStorage\
 - Role-based access (user, admin)\
 - Admin routes and actions protected on both frontend and backend
-

📝 API Endpoints

Auth

- POST /api/v1/auth/register\
- POST /api/v1/auth/login

Sweets (Protected)

- GET /api/v1/sweets\
- GET /api/v1/sweets/search\
- POST /api/v1/sweets (Admin)\
- PUT /api/v1/sweets/:id (Admin)\
- DELETE /api/v1/sweets/:id (Admin)

Inventory

- POST /api/v1/sweets/:id/purchase\
 - POST /api/v1/sweets/:id/restock (Admin)
-

Environment Variables

Create a `.env` file in the backend directory:

```
PORT=5000
MONGO_URI=your_mongodb_connection_string
JWT_SECRET=your_jwt_secret
```

► Running the Project Locally

1 Clone the repository

```
git clone https://github.com/your-username/SweetsNSweets.git
cd SweetsNSweets
```

2 Backend Setup

```
cd backend
npm install
npm run dev
```

3 Frontend Setup

```
cd frontend
npm install
npm run dev
```

Frontend runs at:

<http://localhost:5173>

Backend runs at:

<http://localhost:5000>

UI Highlights

- Clean and modern dashboard\
- Responsive design using Tailwind CSS\
- Admin edit modal for updating sweets\
- Lazy-loaded pages with loading indicators

Learning Outcomes

- Full-stack MERN architecture\
- Secure authentication & authorization\
- RESTful API design\
- Role-based UI rendering\
- State management and API integration\
- Professional project structure

Future Enhancements

- Pagination and filtering\
- Order history for users\
- Admin analytics dashboard\
- Image upload for sweets\
- Payment gateway integration

Author

Ayush Gupta

Full-Stack Developer

Built as a learning and portfolio project

Application Screenshots

Login Page

The screenshot shows a web browser window with a dark theme. The address bar displays "localhost:5173/login". The main content area is a light gray box titled "Sign in to your account". It contains instructions "Enter your credentials to continue". There are two input fields: "Email address" with the value "test@gmail.com" and "Password" with several dots. Below the fields is a blue "Sign In" button. At the bottom of the box, there is a link "Don't have an account? [Create one](#)". The browser's toolbar at the top includes various tabs and icons.

Register Page

The screenshot shows a web browser window with a dark theme. The address bar displays "localhost:5173/register". The main content area is a light gray box titled "Create an account". It contains instructions "Fill in the details below to register". There are three input fields: "Full name" with the value "John Doe", "Email address" with the value "name@example.com", and "Password" with the placeholder "Create a strong password". Below the fields is a blue "Register" button. At the bottom of the box, there is a link "Already have an account? [Sign in](#)". The browser's toolbar at the top includes various tabs and icons.

Dashboard (Admin View)

The screenshot shows the 'Available Sweets' section of the sweetsNsweets dashboard. It displays five sweet items in cards:

- Motichoor Laddu**: Festival Special, ₹600, 98 in stock. Buttons: Purchase, Edit, Delete.
- Kaju Katli**: Dry Fruiti, ₹650, 49 in stock. Buttons: Purchase, Edit, Delete.
- Gulab Jamun**: Milk Based, ₹300, 99 in stock. Buttons: Purchase, Edit, Delete.
- Rajbhog**: Milk Based, ₹400, 99 in stock. Buttons: Purchase, Edit, Delete.

A search bar at the top right says "Search sweets...". The dashboard header includes "SweetsNSweets", "Dashboard", "Admin", and "Logout".

✏️ Edit Sweet Modal

The screenshot shows the 'Update Sweet' modal dialog. It contains fields for modifying sweet details:

- Sweet Name: Motichoor Laddu
- Category: Festival Special
- Price: 600 (dropdown) and 98 (input field)

Buttons at the bottom are "Cancel" and "Update Sweet". The background is blurred to focus on the modal.

🔍 Search & Filter

Screenshot of the SweetsNSweets user dashboard. The URL is localhost:5173/dashboard. The page title is "SweetsNSweets". The top navigation bar includes "Dashboard", "Admin", and "Logout". A search bar contains the text "kaju". Below the search bar, a product card for "Kaju Katli" is displayed, categorized as "Dry Fru". The price is ₹650, and it has 49 in stock. Buttons for "Purchase", "Edit", and "Delete" are present.

+ Add Sweet (Admin)

Screenshot of the SweetsNSweets admin panel. The URL is localhost:5173/admin. The page title is "SweetsNSweets". The top navigation bar includes "Dashboard", "Admin", and "Logout". A section titled "Admin Panel" is shown, with the sub-instruction "Add new sweets to the store inventory". A form titled "Add New Sweet" is displayed, containing fields for "Sweet Name", "Category", "Price (₹)", and "Quantity". A large blue "Add Sweet" button is at the bottom.

👤 Dashboard (User View)

