

Naveena Sweets & Bakery

A modern, high-performance e-commerce web application for Naveena Sweets & Bakery, showcasing traditional Indian sweets, savory snacks, and freshly baked goods. Built with React and Vite, this application offers a seamless shopping experience with optimized WebP images and beautiful UI/UX.

✨ Features

🛒 Product Catalog

Browse through extensive categories including:

- **Sweet Items** (/sweet) - Traditional Indian sweets and gift boxes
- **Ghee Sweets** (/ghee) - Classic ghee-based delicacies
- **Milk Sweets** (/milk) - Rich milk-based sweets
- **Savouries** (/sav) - Savory snack varieties
- **Namkeen** (/namk) - Traditional namkeen items
- **Cakes** (/cake) - Regular cakes and pastries
- **Cool Cakes** (/cool-cakes) - Refrigerated specialty cakes
- **Cakes & Pastries** (/cakes-pastries) - Assorted baked goods
- **Buns, Breads & Donuts** (/buns-breads-donuts) - Fresh bakery items
- **Biscuits, Pies & Boxed Items** (/biscuits-pies-boxed) - Packaged baked goods

🎯 Core Features

- **Shopping Cart System** - Full-featured cart with Context API state management
- **Product Detail Pages** - Dynamic routing /product/:category/:id
- **Image Optimization** - All images pre-optimized to WebP format (800w)
- **Image Preloading** - Critical images preloaded for instant display
- **Lazy Loading** - React Lazy Load Image Component for performance
- **Responsive Design** - Mobile-first design with Tailwind CSS
- **Smooth Animations** - Framer Motion powered transitions
- **Interactive Carousels** - Swiper.js based product showcases
- **Customer Favorites Section** - Curated popular items
- **Scroll to Top** - Automatic scroll on route changes

Tech Stack

Core

- React 19.1.1 - UI framework
- Vite 7.1.2 - Build tool and dev server
- React Router DOM 7.9.1 - Client-side routing

Styling

- Tailwind CSS 4.1.14 - Utility-first CSS framework
- Styled Components 6.1.19 - CSS-in-JS library
- PostCSS & Autoprefixer - CSS processing

UI/UX Libraries

- Framer Motion 12.23.24 - Animation library
- Swiper 12.0.2 - Touch slider component
- Lucide React 0.544.0 - Icon library
- React Icons 5.5.0 - Additional icon set
- React Lazy Load Image Component 1.6.3 - Image lazy loading

Utilities

- Class Variance Authority 0.7.1 - Component variants
- clsx 2.1.1 - Conditional className utility
- Tailwind Merge 3.3.1 - Merge Tailwind classes
- Intersection Observer 0.12.2 - Viewport detection polyfill

Development

- ESLint 9.33.0 - Code linting
- @vitejs/plugin-react 5.0.0 - React support for Vite

Getting Started

Prerequisites

- Node.js 18+ (specified in `.nvmrc`)
- npm or yarn package manager

Installation

1. Clone the repository

```
git clone https://github.com/zaheer037/Naveena-Sweets-Bakery.git  
cd Naveena-Sweets-Bakery
```

2. Install dependencies

```
npm install
```

3. Start development server

```
npm run dev
```

Opens on <http://localhost:5173>

Available Scripts

Command	Description
<code>npm run dev</code>	Start Vite dev server with HMR
<code>npm run build</code>	Build for production to <code>dist/</code>
<code>npm run preview</code>	Preview production build locally
<code>npm run lint</code>	Run ESLint on all files
<code>npm run optimize</code>	Analyze images in <code>src/assets/</code>
<code>npm run convert-images</code>	Convert images to WebP format

Project Structure

```
Naveena-Sweets-Bakery/
├── src/
│   ├── Components/          # Reusable UI components
│   │   ├── About.jsx
│   │   ├── Cakes.jsx
│   │   ├── Carousel.jsx
│   │   ├── CoolCakes.jsx
│   │   ├── CustomerFavs.jsx
│   │   ├── Footer.jsx
│   │   ├── GheeSweets.jsx
│   │   ├── Header.jsx
│   │   ├── MilkSweets.jsx
│   │   ├── NamekeenItems.jsx
│   │   ├── OptimizedImage.jsx
│   │   ├── ProductCard.jsx
│   │   ├── Saviours.jsx
│   │   ├── StyledInput.jsx
│   │   └── SweetItems.jsx
│   └── ui/                  # Reusable UI primitives
│       ├── alert.jsx
│       ├── animated-add-to-cart-button.jsx
│       └── toast.jsx
├── Pages/                 # Route components
│   ├── Home.jsx
│   ├── Sweets.jsx
│   ├── Saviours.jsx
│   ├── Gheesweets.jsx
│   ├── MilkSweets.jsx
│   ├── Namkeen.jsx
│   ├── Cakes.jsx
│   ├── CoolCakesPage.jsx
│   ├── CakesPastriesPage.jsx
│   ├── BunsBreadsDonutsPage.jsx
│   ├── BiscuitsPiesBoxedPage.jsx
│   └── ProductDetailPage.jsx
├── Context/               # React Context
│   └── CartContext.jsx    # Shopping cart state
├── Layout/                # Layout wrapper
│   └── MainLayout.jsx
├── Routing/               # Route definitions
│   └── AppRouting.jsx     # All route configurations
├── data/                  # Static data
│   └── allProducts.js    # 700+ lines product catalog
├── assets/                # Static images (WebP 800w)
├── utils/                 # Helper functions
│   └── imagePreloader.js
├── lib/                   # Shared utilities
│   └── utils.js
├── styles/                # Additional CSS
│   └── ImageOptimization.css
├── App.jsx                 # Root component
├── App.css
├── main.jsx                # Entry point
└── index.css                # Global styles
├── public/
│   └── _redirects           # Netlify redirects
├── dist/                   # Production build output
├── convert-new-images.js  # Image conversion utility
├── optimize-images.js     # Image optimization utility
├── index.html              # HTML template
├── vite.config.js          # Vite configuration
├── tailwind.config.js      # Tailwind configuration
├── postcss.config.js       # PostCSS configuration
├── eslint.config.js        # ESLint configuration
├── staticwebapp.config.json # Azure Static Web Apps config
└── package.json             # Node version
└── .nvmrc                  # Node version
```

Configuration

Vite (`vite.config.js`)

- `Path alias: "@" → "./src"`
- `Plugin: @vitejs/plugin-react`

Routing (`src/Routing/AppRouting.jsx`)

All routes include automatic scroll-to-top on navigation:

- `/` - Home
- `/sweet` - Sweet items
- `/ghee` - Ghee sweets
- `/milk` - Milk sweets
- `/sav` - Saviours/snacks
- `/namk` - Namkeen
- `/cake` - Cakes
- `/cool-cakes` - Cool cakes
- `/cakes-pastries` - Cakes & pastries
- `/buns-breads-donuts` - Buns, breads & donuts
- `/biscuits-pies-boxed` - Biscuits, pies & boxed items
- `/product/:category/:id` - Product details

State Management

- **CartContext** - Global cart state using React Context API
- Provides cart operations to all components

Image Optimization

Current Setup

- All product images pre-converted to **WebP format**
- Standard width: **800w** for optimal quality/size balance
- Images stored in `src/assets/`

Optimization Scripts

```
# Analyze existing images  
npm run optimize  
  
# Convert new images to WebP  
npm run convert-images
```

Critical Image Preloading

Implemented in `App.jsx` using `utils/imagePreloader.js` for instant hero/banner image display.

Deployment

Azure Static Web Apps

Configured via `staticwebapp.config.json` with SPA routing support. All routes redirect to `index.html` with 200 status.

Build Process

```
npm run build
```

- Output: `dist/` directory
- Assets: Hashed filenames for cache busting
- Index: `dist/index.html`

Hosting Platforms Supported

- Azure Static Web Apps

Design & UX

Key Design Principles

- **Heritage Brand Identity** - "Crafting Sweet Moments Since 1995"
- **Performance First** - WebP images, lazy loading, code splitting
- **Mobile Responsive** - Tailwind breakpoints for all devices
- **Smooth Interactions** - Framer Motion animations
- **User-Friendly Navigation** - Clear category organization

UI Components

- Custom styled components with CVA variants
- Reusable animated buttons
- Toast notifications for user feedback
- Optimized image component with lazy loading
- Product cards with hover effects

Contributing

1. Fork the repository
2. Create feature branch (`git checkout -b feature/AmazingFeature`)
3. Commit changes (`git commit -m 'Add AmazingFeature'`)
4. Push to branch (`git push origin feature/AmazingFeature`)
5. Open Pull Request

License

This project is proprietary and confidential.

Made with ❤ by Zaheer