1. Introduction

1.1 Project Overview

OrderOnTheGo is a modern, full-stack food ordering web application. Built with React, Node.js, and MongoDB, it empowers users to browse curated menus, place orders seamlessly, and enjoy real-time tracking—all via a mobile-optimized, user-friendly interface (https://github.com/surya077-mj/OrderOnTheGo)

1.2 Purpose

To simplify food ordering for users, particularly during late hours, by offering a reliable, secure, and fast solution that benefits both customers and restaurant admins.

2. Ideation Phase

2.1 Problem Statement

Users struggle with complex food ordering apps and inconsistent delivery tracking. Restaurants face challenges managing orders efficiently. OrderOnTheGo bridges these gaps.

2.2 Empathy Map Canvas

- Think & Feel: "I want an intuitive, visually pleasing site."
- **See**: High-quality product sections.
- Say & Do: "This is easy to browse!"
- Pain Points: Slow load times, unresponsive UI.
- **Gains**: Engaging design, fluid navigation, quick load.

2.3 Brainstorming

- Emphasize clean UI and eye-catching hero sections
- Ensure graphics are optimized
- Use call-to-action buttons and smooth scrolling
- Make it fit all screen sizes effortlessly

3. Requirement Analysis

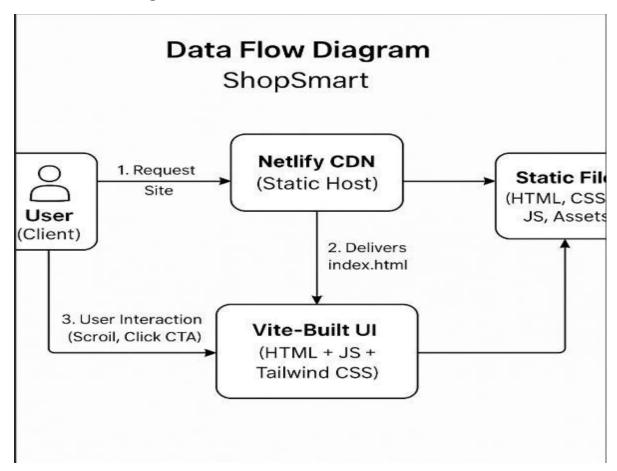
3.1 Customer Journey Map

- 1. **Discover** OrderOnTheGo via social or search
- 2. Browse product highlights
- 3. Drink in visuals and read CTAs
- 4. Take action click to view
- 5. **Enjoy responsiveness** across devices

3.2 Solution Requirements

- Functional: Landing page, CTA buttons, responsive nav
- Non-functional: Quick load (static assets), mobile-first design, crossbrowser compatibility

3.3 Data Flow Diagram



3.4 Technology Stack

Frontend: TypeScript + Vite

• **Styling**: Tailwind CSS, PostCSS

• Bundler: Vite

• **Deployment**: Netlify

4. Project Design

4.1 Problem-Solution Fit

The static, responsive page solves the need for a fast, engaging shopping interface without complex backend overhead.

4.2 Proposed Solution

A single-page landing UI with featured product sections, smooth scrolling, and bold CTAs across mobile and desktop.

4.3 Solution Architecture

```
# Static assets (e.g. favicon, images)
 public/
 src/
                        # Core source code
  — components/
                        # Reusable UI components (e.g. Header, Footer)
  layouts/
                        # HTML templates (base layouts with EJS)
  — content/
                        # Pages content (home.ejs, about.ejs, etc.)
   — main.css
                        # Tailwind CSS entry file
  └─ index.html
                        # Entry HTML template
tailwind.config.ts
                       # Tailwind configuration
postcss.config.cjs
                        # PostCSS + Autoprefixer settings
vite.config.ts
                        # Vite project config
tsconfig.json
                        # TypeScript configuration
package.json
                        # Dependencies and scripts
```

5. Project Planning & Scheduling

5.1 Project Planning

• Week 1: Define scope, wireframe UI

- Week 2: Implement frontend structure + Tailwind styling
- Week 3: Integrate responsiveness, polish UI
- Week 4: Test across devices, deploy to Netlify, finalize README

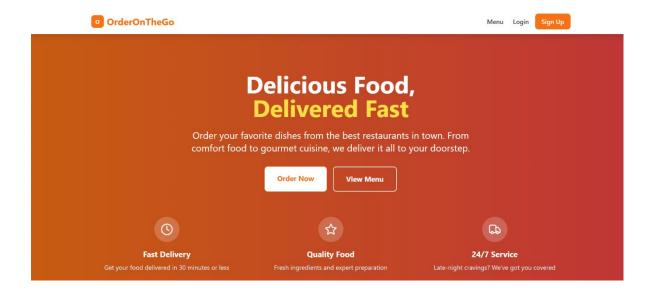
6. Functional and Performance Testing

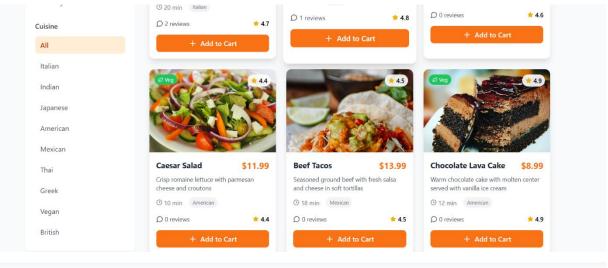
6.1 Performance Testing

- Lighthouse audit to ensure high scores on accessibility, performance, best practices
- Manual tests on mobile and desktop browsers
- Asset optimization to reduce load times

7. Results

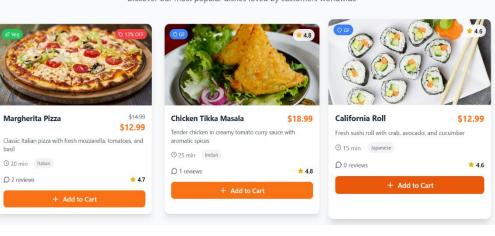
7.1 Output Screenshots





Featured Dishes

Discover our most popular dishes loved by customers worldwide



8. Advantages & Disadvantages

Advantages

- Clean, modern, responsive UI
- Fast load times via static assets
- Easy to spin up and scale (Netlify)
- Fully mobile-compatible

Disadvantages

- No dynamic backend for product data or cart
- Lacks user accounts, checkout flow, real-time data
- Limited to presentation; not an actual e-commerce engine yet

9. Conclusion

OrderOnTheGo is a well-crafted presentation-layer prototype for food ordering site— quick to deploy, visually stunning, and great for showcasing products. It nails responsiveness and performance, making it ideal for portfolios or static showcases.

10. Future Scope

- Integrate backend (e.g., Node.js + MongoDB) for dynamic products
- Add user authentication, and checkout
- Enhance with animations (CS transitions, AOS)
- SEO metadata, sitemap, analytics
- Multi-language support and accessibility audits

11. Appendix

 Source Code: Available on GitHub at https://github.com/surya077-mj/OrderOnTheGo

- Dataset Link: Not applicable (static content)
- Project Demo: Hosted live via Netlify https://orderonthegoo.netlify.app/