**Project Documentation**

**Overview**

This project is my implementation of the *Frontend Developer Design Implementation Test* by The Tonic Technologies. The goal was to recreate one screen from the provided Figma design using **Next.js** while maintaining design accuracy, clean code, and a clear development structure.

**Screen Selection**

I selected screen one because it showcases more UI elements such as navigation, hero section, feature blocks, etc, which would put me in a better position to showcase my implementation ability and dynamic approach to problem solving

**Tools Used**

Framework: Next.js 14 (App Router)

Styling: Tailwind CSS

Icons: Lucide React

Deployment: Netlify [[Link](https://amazing-torte-ac0ebe.netlify.app/)]

Version Control: GitHub [[Link](https://github.com/Caleb-1023/ttt-frontend-developer-test)]

Supporting Packages: ShadCN, React Fast Marquee

**Folder and File Structure**

I structured the application by analyzing the webpage layout and segmenting it into logical sections, which informed the breakdown of components. The project follows a modular architecture with a dedicated components directory. Typically, I organize components by page or route, alongside a shared folder that contains reusable elements utilized across multiple parts of the application. For this project, the components directory consists of two primary subfolders: home, which contains components specific to the selected screen, and shared, which houses common UI elements such as the navigation bar and footer.

**Development Process**

I started by inspecting the Figma layers to understand spacing and component hierarchy. I used Tailwind’s responsive classes to ensure a consistent layout across screen sizes. I also utilized ShadCN components to streamline development and reduce the need to rebuild common interface elements from scratch.

I appreciate the consistency in the design, which made development quicker as I was able to pick specifications quickly.

**Coding Style and Consistency**

I ensured that each component maintained minimal and focused functionality to enhance readability and maintainability. While I structured the components to accept props for potential extensibility, I intentionally limited their complexity given the project’s scope, the low likelihood of future modifications, and the time constraints of the assessment.

**Improvements**  
I implemented responsive design principles to ensure optimal display across various screen sizes, utilizing flexible layouts and stacking components where necessary. Additionally, I incorporated subtle animations to enhance user experience and provide a sense of interactivity and visual fluidity throughout the interface.

**Challenges**

I encountered an issue with certain images appearing blurry during implementation. To address this, I temporarily replaced the affected assets with their SVG equivalents, which maintained visual clarity across different screen resolutions. This served as an efficient short-term solution given the time constraints, though a more robust debugging process would be required in a full production environment.

**Deployment**

The project was deployed on Netlify using the GitHub repository for continuous integration.

**Live link:** <https://amazing-torte-ac0ebe.netlify.app/>

**GitHub repo:** <https://github.com/Caleb-1023/ttt-frontend-developer-test>