🧩 Frontend Task (Figma → React/Next JS)

📝 Overview

Welcome to Zavisoft's Frontend Implementation Task. You are about to translate design into reality by building two pages from Figma with accuracy and attention to detail.
Consider this document your blueprint — read it thoroughly before you begin.

🎨 Figma Design

MUST - Read the comments sections from the figma design, fetch from the api endpoints where mentioned, other than that, keep everything else static, take images and other stuff from the design file

https://www.figma.com/design/tL4VI3H0y0XE5N7aqvvIE2/Frontend-task?node-id=0-1&t=FUXy47B5mpww4mMg-1

🧪 API Details (Products Only)

The Platzi Fake Store API is available at;
https://fakeapi.platzi.com/en/rest/products/
https://fakeapi.platzi.com/en/rest/categories/

✅ Core Requirements

- See the Technology Stack table and choose according to your preference.
- Implement the UI to match the Figma design (layout, spacing, typography, colors, interactions), read each section's comments.
- Integrate the API for product listing(Landing page), product details(Product Detail page) and the product categories list(products categories list).
- Ensure responsive behavior for mobile, and desktop.

🛠️ Technology Stack

Category	Tools
Framework	React / Next.js
State Management	Redux / Context API
Data Fetching	RTK Query / Axios
Language	JavaScript / TypeScript
Styling	Tailwind CSS
Component Library	Any (optional)

📌 Important Notes

- Focus on UI fidelity and code quality.
- Include loading, empty, and error states for API requests.
- Make proper and meaningful commit history
- You may use AI for this task
- Even if you complete 70/80% of the task, do submit, we are lenient on marking and try our best to consider you for the next round.
- We care about how you think and implement a task - do you know your code?

💯 Mark Distribution => Total 100

1. Design ==> Web(35) and Responsive(15)
2. Codebase ==> Clean component architecture(5) + meaningful commit history(5)
3. Proper data flow logic ==> State management(10) + Clean data fetching architecture(10)
4. Proper UI management ==> Loading, Empty, and Error states UI(20)

📤 What You Must Submit

1. GitHub Repo — Frontend repo with meaningful commits, clean, modular, documented; include a README with overview, setup, tech stack, live URL, and notes.
2. Live Deployment — Working frontend URL.

🌟Bonus (Optional)

- Implement the Cart Page design and its functionality, keep the states local, no need for API endpoint (20)
- Add minimal but meaningful animations/transitions where appropriate(10)

Submission Deadline ⇒ 11:59 PM 23/02/26 Monday

Submission Link ⇒ https://docs.google.com/forms/d/e/1FAIpQLScov8sJFVsQWxC26P1Vkh2a14TZH_Y_EOnX23mCRqugd2FARg/viewform?usp=header