# Technical Assignment

**Company:** Whzan

**Job:** Senior Front-End Developer

---

# Take-Home Assignment: Catalog Explorer

## Context

You're building a catalog explorer for a marketplace platform. Shoppers need to browse product catalogs, search and filter items, view detailed product information, and save items for later review.

**Business Context:**

- The platform serves thousands of products across multiple categories
- Mobile traffic accounts for 70% of usage
- Users frequently share product searches and filtered views with others
- The application must serve users with diverse abilities and assistive technologies
- Network conditions vary significantly across our user base

Build this application from scratch using React and your choice of JavaScript or TypeScript.

### Technical Constraints

**Required:**

- React
- JavaScript or TypeScript (your choice)
- Vite or Webpack for build tooling
- You may use any styling approach and utility libraries as needed

**API Requirements:**

- You must implement your own REST API (mock server, stub, JSON files, or local service)
- Do not use third-party APIs for product data

### Example Data Model

Design your own data structures. For reference, here's an example of what product data might look like:

```
{
  "id": "itm_123",
  "name": "Wireless Headphones",
  "price": 129.99,
  "currency": "USD",
  "rating": 4.6,
  "reviewCount": 87,
  "tags": ["audio", "wireless"],
  "inStock": true,
  "imageUrl": "https://...",
  "updatedAt": "2025-01-15T12:30:00Z"
}
```

Detailed product views may include additional information like descriptions, specifications, and image galleries.

# Requirements

## Functional Requirements

Your application must allow users to:

1. **Browse Products**
   - View a collection of products with key information (name, price, image, availability, etc.)
   - Search for products by text
   - Filter products by at least one attribute (e.g., category, tag, availability)
   - Sort products by at least one criterion (e.g., price, rating)
2. **View Product Details**
   - Access detailed information about a specific product
   - Navigate between the catalog view and detail view
3. **Save Items**
   - Mark products as favorites or add them to a watchlist
   - View their saved items
   - Saved items should persist across browser sessions

## Non-Functional Requirements

Your solution must demonstrate:

1. **Accessibility**
   - The application must be usable by all users, including those using assistive technologies (screen readers, keyboard navigation, etc.)
   - Consider users with visual, motor, and cognitive accessibility needs
2. **Mobile Experience**
   - Given that 70% of traffic comes from mobile devices, the application must provide an excellent experience across all screen sizes
   - Performance and usability on mobile devices is critical
3. **Reliability**
   - The application must handle network failures and errors gracefully
   - Users should never be left in a confusing or broken state
   - Data operations should be dependable and predictable
4. **Performance & Scalability**
   - The solution must work efficiently with catalogs containing thousands of products
   - Consider the performance implications of your architectural decisions
5. **Usability**
   - Users should always understand what's happening in the application
   - Interactions should feel responsive and provide appropriate feedback
   - The experience should be intuitive and efficient
6. **Shareability**

- Users need to be able to share specific catalog views (searches, filters, sorts) with others
- When someone receives a shared link, they should see the same view as the person who shared it

7. **Maintainability**
   - The codebase should be structured so that other developers can easily understand, modify, and extend it
   - Consider how your code will be maintained over time by a team
8. **Deployability**
   - The application must be ready to deploy to different environments (development, staging, production)
   - Environment-specific configuration should be managed appropriately

---

## Deliverables

### 1. GitHub Repository

A single repository containing:

- All source code
- Development and production build configurations
- A **README.md** with:
  - Prerequisites and setup instructions
  - How to run the application locally
  - How to build for production
  - How to run the production build locally
  - Any other information needed to evaluate your solution

### 2. SOLUTION.md Document

A written document explaining your approach:

**Required Topics:**

- **Requirements Analysis:** How did you interpret the requirements? What assumptions did you make?
- **Architecture Decisions:** Why did you structure your code this way? What alternatives did you consider?
- **Non-Functional Requirements:** How does your solution address each of the non-functional requirements (accessibility, performance, reliability, etc.)?
- **Trade-offs:** What trade-offs did you make? What would you do differently with more time or resources?
- **Technology Choices:** Why did you choose JavaScript or TypeScript? What influenced your other technology decisions?

### 3. Video Demo (5-10 minutes)

Using Loom or similar screen recording tool:

- **Demonstrate the working application** (2-3 minutes)
  - Show the core user flows
  - Demonstrate how it works on different screen sizes
  - Show how you addressed key requirements (accessibility, error handling, etc.)
- **Code walkthrough** (3-5 minutes)
  - Explain your architecture and code organization

- Highlight interesting technical implementations
- Discuss how your code supports the non-functional requirements

- **Reflection** (1-2 minutes)
  - Key decisions and why you made them
  - What you're proud of
  - What you would improve given more time

---

## Evaluation Criteria

You will be evaluated on:

1. How well your solution meets the functional requirements
2. How effectively you addressed the non-functional requirements
3. The quality of your architecture and code organization
4. Your ability to make and justify technical decisions
5. How clearly you communicate your reasoning and trade-offs

---

## Time Estimate

This assignment is designed to take approximately **4 hours**, but you may take as much time as you need.

---

## Questions?

If any requirements are unclear, document your assumptions in your SOLUTION.md and explain how those assumptions influenced your decisions. This is part of what we want to see – how you handle ambiguity and make reasoned choices.