

Front-End Technical Assessment

Role: Front-End Developer

Expected Duration: 6 hours

AI Usage: Required

Exercise 1: Build a Responsive UI Component

Create a Personal Creative Portfolio from scratch, composed of:

- Backend: API built with Strapi v4+ to manage content
- Frontend: SPA built with React v18+ consuming the Strapi API
- Responsive and mobile-first design. Freedom to choose styling tools

Requirements:

- Fully responsive (mobile-first).
- Follow clean-code and best architecture practices.
- Include component structure, styling, and one unit test.
- Explain how AI contributed to your solution.

Exercise 2: API Integration + Error Handling

Integrate with <https://fakestoreapi.com/products> and:

- Create a service to fetch the list of products.
- Implement detailed error handling (5xx, 4xx, network).
- Build a simple page listing product titles (use Strapi for content like in exercise 1).
- Document AI assistance in the design of the solution.

Deliverables Exercises 1 & 2

- GitHub repository (public or private – if private, grant access to: anfvlandiaer) including full backend and frontend source code with descriptive commits
- Complete README.md including installation instructions, technical decisions, an 'AI Usage' section, and optional screenshots or demo video
- Functional application: both Strapi and React projects must run locally; deployment is not required

Exercise 3: Architecture Reasoning (AI-assisted)

Design a scalable front-end architecture using microfrontends and DevOps practices. Your answer must include:

- Explanation of chosen microfrontend strategy.
- CI/CD pipeline design.

- Scalability, maintainability, and performance considerations.
- A screenshot or textual transcription of the AI prompts used.

Exercise 4: AI-Assisted Code Refactoring

Refactor the following code using AI into modern, production-ready quality:

```
function getUser(d){ return  
fetch("https://jsonplaceholder.typicode.com/users/"+d).then(x=>x.json()).then(j=>console  
.log(j)) }
```

Requirements:

- - Use AI to refactor the code.
- - Explain which prompts were used.
- - Deliver the final clean version of the code.