# Competency-Based Task

## Task Overview

We would like you to build a small app that demonstrates your front-end development skills. This can be a simple static page, a multi-page layout, or a minimal interactive application. Your project should consume an API that provides data for the page.

We would like to see your app use an API that provides a basic endpoint and some content/data for your page/component. We have chosen 'Mock Shop' https://mock.shop/ - a public endpoint that provides a mock Shopify storefront.

The API requires no authentication.

### Deliverables

We are interested in seeing how you approach common front-end tasks. Below are some key features we’d like to see in your build:

1. **Reusable Component:**
   * Implement a reusable interactive component of your choice, such as an accordion, carousel, or call-to-action (CTA) card.
2. **Responsive CTA Grid:**
   * Display a grid of at least **8 items** retrieved from the API.
   * The grid should be fully responsive, spanning the width of the page and adjusting layout as the viewport size decreases.
   * Example of a collapsing CTA grid: [Example](http://d.pr/v/BB7EDL)
   * Include **sorting functionality** to allow users to order products by price. Bonus points for additional filters (e.g., by color or collection).

### Suggestions

We’d like to see well-structured and maintainable code. Here are some key areas we will evaluate:

* **Project Organization:**
  + Clear and logical file/folder structure.
* **Code Quality & Best Practices:**
  + Strong understanding of **ES6+ syntax**.
  + Clean and modular TypeScript.
* **Styling & Responsiveness:**
  + Use a CSS methodology of your choice (**CSS Modules, Styled Components, Tailwind, or plain CSS**).
  + Implement smooth and subtle UI animations (e.g., button hover states, expanding accordions). Nothing too over the top required here.
* **Performance Considerations:**
  + Minimal, efficient, and well-optimized code.
* **Optional Extras (if time allows):**
  + Unit tests or functional tests (not required, but good to see).
  + Git version control with a visible commit history that shows how your solution evolved.

### Submission

Please provide your submission in a GitHub or Bitbucket repository so we can clone locally and run the application to get an overall taste of your development process.

**Please understand this doesn’t have to be 100% complete, and we would appreciate it if you didn't spend more than 7 hours on it.** We're just looking to see some examples of your capability.

Comments throughout your code are useful if you think there’s anything particularly complex. If you manage to get through everything – that's fantastic. When you're happy with your build – please send us a link or invite the relevant contacts in UNRVLED to your repository. We will follow-up with a meeting where we will allow you to walk us through your application and explain your approaches and any decisions you made.

If you have any questions about this, please email the UNRVLD team and we'll be happy to chat further.

Thank you and good luck