Group assignment: Webshop

**Description**

In this assignment, you will develop a small e-commerce application in Vite. Users will be able to view a list of products, view details for a specific product, and manage a shopping cart. The application will be designed in Figma before implementation.

**⚠️**[Please read this information before starting the project](https://chasacademy.instructure.com/courses/471/pages/grupparbetet-alla-svar-pa-ett-stalle)**⚠️**

You are expected to participate in group reconciliations throughout the project. See the schedule [hereLinks to an external site.](https://docs.google.com/spreadsheets/d/1bDxmZAoOo4u-hXxQt-HjMYPKEy06UBKhU6gPGe464bE/edit?usp=sharing" \t "_blank)

**Requirements for passing**

**⚠️ Hygiene requirements**

* The application will be built with Vite.
* Routing should be handled withreact-router-dom.
* Design the application in Figma and create wireframes and mockups.
* The project will be version managed with Git and available on GitHub.
* Responsive design that works on both mobile and desktop.

**Basic functionality**

**1. Product list**

* Get products from [this APILinks to an external site.](https://fakestoreapi.com/docs)
* Show a list of products.

**2. Product page**

* View detailed information about a product.
* Implement navigation to the product page via react-router-dom.

**3. Shopping cart**

* Users should be able to add products to the shopping cart and "complete a purchase" (no payment implementation needed).

**4. Order confirmation**

* Show a simple order confirmation after a completed purchase (no payment implementation needed).

**Code structure and syntax**

* The application should be divided into separate components and files.
* All components should be functional and written using arrow-function syntax.
* Components should be exported at export defaultthe bottom of the file.
* Prettier should be used to ensure consistent code formatting.

**For higher grades**

To receive a higher grade, all requirements for passing must be met, and in addition:

**Code quality**

* Very well-structured and semantic code.
* No unnecessary comments in the code.

**Extended functionality (for VG)**

* **Filtering** : Implement filtering of products based on category.
* **Shopping cart storage** : Save your shopping cart localStorageso it doesn't get lost when reloading.
* **Change quantity in shopping cart:** Implement the ability to change the quantity of a product in the shopping cart.

**Assessment criteria**

**Design and structure**

* Whether wireframes and mockups match the final application.
* Whether design decisions (color, typography, UX) are justified.

**Functionality**

* If the user can navigate between pages and add products to the shopping cart correctly.
* If the application works smoothly and without bugs.

**Code quality**

* If the code is well-organized, semantic, and easy to read.
* If the code structure and component division are well thought out.

**Submission**

* Submit your solution as a **GitHub repository** with:
  + All code for the project
* A Figma file with wireframes and mockups is attached in Canvas
* **Presentation** (max 20 min/group):
  + Demo of the application
  + Justifications for design and coding decisions
  + Description of the work process and role distribution

If you are unable to attend the presentation, a written report must be submitted covering the above points (except the demo).

그룹 과제: 웹숍

**설명**

이 과제에서는 Vite를 사용하여 작은 전자상거래 애플리케이션을 개발합니다. 사용자는 제품 목록을 보고, 특정 제품에 대한 세부 정보를 보고, 쇼핑 카트를 관리할 수 있어야 합니다. 해당 애플리케이션은 구현하기 전에 Figma에서 설계되어야 합니다.

**⚠️**[프로젝트를 시작하기 전에 이 정보를 꼭 읽어주세요](https://chasacademy.instructure.com/courses/471/pages/grupparbetet-alla-svar-pa-ett-stalle)**⚠️**

프로젝트 진행 중에는 그룹 화해에 참여해야 합니다. [여기에서](https://docs.google.com/spreadsheets/d/1bDxmZAoOo4u-hXxQt-HjMYPKEy06UBKhU6gPGe464bE/edit?usp=sharing) 일정을 확인하세요[외부 사이트로의 링크.](https://docs.google.com/spreadsheets/d/1bDxmZAoOo4u-hXxQt-HjMYPKEy06UBKhU6gPGe464bE/edit?usp=sharing)

**합격을 위한 요구사항**

**⚠️ 위생 요구 사항**

* 해당 애플리케이션은 Vite를 사용하여 구축됩니다.
* 라우팅은 다음과 같이 처리되어야 합니다.react-router-dom.
* Figma에서 애플리케이션을 디자인하고 와이어프레임과 모형을 만듭니다.
* 이 프로젝트는 Git을 사용하여 버전 관리되며 GitHub에서 제공됩니다.
* 모바일과 데스크톱 모두에서 작동하는 반응형 디자인입니다.

**기본 기능**

**1. 제품 목록**

* [이 API](https://fakestoreapi.com/docs) 에서 제품을 가져옵니다[외부 사이트로의 링크.](https://fakestoreapi.com/docs)
* 제품 목록을 보여줍니다.

**2. 제품 페이지**

* 제품에 대한 자세한 정보를 확인하세요.
* . 을 통해 제품 페이지로의 탐색을 구현합니다 react-router-dom.

**3. 쇼핑 카트**

* 사용자는 쇼핑 카트에 제품을 추가하고 "구매를 완료"할 수 있어야 합니다(결제 구현이 필요하지 않음).

**4. 주문 확인**

* 구매가 완료된 후 간단한 주문 확인을 표시합니다(결제 구현이 필요하지 않음).

**코드 구조 및 구문**

* 신청서는 별도의 구성 요소와 파일로 나누어야 합니다.
* 모든 구성 요소는 기능적이어야 하며 화살표 함수 구문을 사용하여 작성되어야 합니다.
* 구성 요소는 파일 맨 아래 에 내보내야 합니다 .export default
* 일관된 코드 형식을 보장하려면 Prettier를 사용해야 합니다.

**더 높은 학년을 위해**

더 높은 학점을 받으려면 합격에 필요한 모든 요건을 충족해야 하며, 추가로 다음 사항을 충족해야 합니다.

**코드 품질**

* 매우 잘 구조화되고 의미론적인 코드입니다.
* 코드에 불필요한 주석이 없어야 합니다.

**확장된 기능(VG용)**

* **필터링** : 카테고리에 따라 제품 필터링을 구현합니다.
* **카트 보관** : 다시 로드할 때 카트가 localStorage분실되지 않도록 보관하세요.
* **장바구니의 수량 변경:** 장바구니에 있는 제품의 수량을 변경하는 기능을 구현합니다.

**평가 기준**

**디자인과 구조**

* 와이어프레임과 모형이 최종 애플리케이션과 일치하는지 여부.
* 디자인 결정(색상, 타이포그래피, UX)이 정당한지 여부.

**기능성**

* 사용자가 페이지 간에 탐색하고 장바구니에 상품을 올바르게 추가할 수 있는지 여부
* 해당 애플리케이션이 버그 없이 원활하게 작동한다면.

**코드 품질**

* 코드가 잘 구성되어 있고, 의미가 명확하며, 읽기 쉬운 경우.
* 코드 구조와 구성요소 구분이 잘 고려되어 있다면.

**제출**

* **다음을 사용하여 GitHub 저장소** 로 솔루션을 제출하세요 .
  + 프로젝트의 모든 코드
* 와이어프레임과 모형이 포함된 Figma 파일이 Canvas에 첨부되어 있습니다.
* **프레젠테이션** (최대 20분/그룹):
  + 응용 프로그램 데모
  + 디자인 및 코딩 결정에 대한 정당성
  + 업무 프로세스 및 역할 분배에 대한 설명

프레젠테이션에 참석할 수 없는 경우, 위 사항(데모 제외)을 포함하는 서면 보고서를 제출해야 합니다.