

NB :

- **Seulement l'exécution du code est notée**
- **Toutes les méthodes de récupération des données doivent être implémentées dans le service ProductService.**

Travail demandé :

- 1- Créer la classe décrivant le modèle de données « Product » suivante :

```
export class Product{  
  
    id ! : number  
    name !: string  
    description ! : string  
    price ! : number  
    available!: Boolean  
    likes !: number []  
}
```

- 2- Au niveau du composant **NavBarComponent**, ajoutez 2 liens “Ajouter un produit” et “Consulter les produits”. **(1 pt)**
- 3- En cliquant sur le lien « Consulter les produits », le composant **ProductsComponent** est affiché. **(0.5)**
- 4- En cliquant sur le lien « Ajouter un produit », le composant **AddProductComponent** est chargée. **(0.5 pt)**
- 5- Si un utilisateur tape un path non existant, le composant **NotFoundComponent** est chargé. **(0.5 pt)**
- 6- Au niveau du composant **AddProductComponent**, ajoutez un formulaire d’ajout d’un produit **(3 pts)**

- 7- Affichez **un message adéquat pour** chaque contrainte non respectée au niveau du formulaire sachant que :
- Les champs **name** et **description** sont obligatoires (1 pt)
 - Le champ **price** est strictement positif (0.5 pt)
 - Le bouton est désactivé tant que le formulaire est invalide. (1 pt)
- 8- Une fois les contraintes sont respectées et en cliquant sur le bouton « Ajouter », le produit est ajouté dans la liste des produits avec une valeur de l'attribut « **available** » égale à true. (3 pts)
- 9- Au niveau du composant **ProductsComponent**, récupérer la liste des produits à partir du service et afficher uniquement les produits disponibles (valeur de l'attribut **available** égal à true). (3 pts)
- 10- Ajouter devant chaque produit un bouton « **Like** ». En cliquant sur le bouton « **Like** », l'id de l'utilisateur est ajoutée dans l'attribut likes du produit sélectionné. Dans notre cas, vous pouvez insérer une valeur de votre choix comme id de l'utilisateur et cette valeur ne peut pas se répéter plusieurs fois (un utilisateur ne peut pas aimer le même produit plusieurs fois) (3 pts)
- 11- Afficher devant chaque produit le nombre de "likes" (3 pts)

Amélioration de la visibilité d'Esprit grâce à GitHub et au programme GitHub Education

Contexte

Afin d'accroître la visibilité d'Esprit, nous souhaitons exploiter pleinement les avantages offerts par GitHub.

De plus, nous tirerons parti des opportunités proposées par le programme GitHub Education for Students.

Evaluation Criteria (2 Points)

1. **Hébergement (Facultatif)** : Github education for students (Heroku, DigitalOcean, Namecheap, etc) + **Repo public**

2. Description du projet

Lorsque vous créez ou mettez à jour un dépôt sur GitHub, vous avez la possibilité d'ajouter une **description** du projet (qui apparaît juste en dessous du nom du dépôt). Il est important d'y inclure des mots-clés concis mais informatifs pour aider les autres utilisateurs à comprendre immédiatement de quoi il s'agit.

Exemple :

- Briefly state that the project was created as part of a university course or program. Include the course name and its relevance to the project.
- Example: This project, **Task Manager Pro**, was developed as part of the coursework for **CS503 - Full-Stack Development** at [XYZ University](#).

3. Utilisation de "topics" (étiquettes de projet)

GitHub permet d'ajouter des **"topics"** à un dépôt, qui sont des mots-clés permettant de mieux classer le projet et de le rendre plus facilement trouvable. Ces topics apparaissent sur la page du projet et aident à définir son domaine d'application.

Exemples de topics :

- python
- machine-learning
- data-analysis
- web-development

- deep-learning
- artificial-intelligence
- data-visualization

Pourquoi c'est important :

- Les topics aident les utilisateurs à trouver des projets spécifiques à leurs intérêts.
- Ils permettent aux projets de se lier à d'autres projets similaires ou pertinents, renforçant ainsi la visibilité.
- L'utilisation appropriée de ces tags peut également améliorer le référencement du projet dans les moteurs de recherche externes (Google, par exemple).

Edit repository details [Close]

Description

Short description of this repository

Website

Enter a valid URL

Topics (separate with spaces)

Include in the home page

- ☒ Releases
- ☒ Packages
- ☒ Deployments

Cancel Save changes

4. Mots-clés dans le README

Le fichier `README.md` est essentiel pour expliquer le projet et attirer des contributeurs. L'insertion de mots-clés pertinents dans le **README** (en particulier dans les sections "Description", "Installation", "Usage", etc.) permet non seulement de mieux décrire le projet, mais aussi de le rendre plus facilement trouvable via les moteurs de recherche.

Conseils pour intégrer des mots-clés dans le README :

- Utiliser des mots-clés pertinents dans le **titre du projet** et dans les **sections principales** (par exemple, "Features", "Installation", "Technologies utilisées").
- Mentionner des **technologies spécifiques** utilisées dans le projet (par exemple : Python, React, TensorFlow, Kubernetes, etc.).
- Si le projet est un outil ou une bibliothèque, inclure des mots-clés liés à son utilisation (par exemple : data processing, API, GUI, automation).

Exemple de structure :

Project Title

Overview

Features

Tech Stack

Frontend

Backend

Other Tools

Directory Structure

Getting Started

Acknowledgments

Edit Preview

```
1  # Project Title
2  ## Overview
3  ## Features
4  ## Tech Stack
5  ### Frontend
6  ### Backend
7  ### Other Tools
8  ## Directory Structure
9  ## Getting Started
10 ## Acknowledgments
```

Exemple :

Task Manager Pro A full-stack task management application built with the MERN stack.

Overview

This project was developed as part of the coursework for **CS503 - Full-Stack Development** at [XYZ University](<https://xyzuniversity.edu>). It explores real-time task management with a focus on responsive design and collaborative functionality. ---

Features

- Real-time task updates using **WebSockets**
- User authentication with **JWT**
- Intuitive UI built with **React.js** and **Material-UI** ---

Acknowledgments

This project was completed under the guidance of [Professor Jane Doe](<mailto:jane.doe@xyzuniversity.edu>) at XYZ University.

It was also presented at the XYZ University Tech Fair 2024. ---

Exigence : Utiliser le mot-clé **“Esprit School of Engineering”** pour faire référence à Esprit.

Grille d'évaluation (2 points)

Critère	Points
Hébergement	0,5 (facultatif)
Description du projet	0,5
Utilisation de "topics"	0,5
Mots-clés dans le README	0,5 ou 1 si pas hébergement