

Examen en cours de formation #1

Date de rendu : **9 Janvier 2026**

Date de la soutenance orale: **15 Janvier 2026**

Domaines

CP1 : Installer et configurer son environnement de travail en fonction du projet

CP2 : Maquetter des interfaces

CP3 : Réaliser des interfaces statiques

CP4 : Développer la partie dynamique Front-End d'une application Web

Compétences évaluées

Module	Compétences visées
Réalisation de maquette	Respect des normes et bonnes pratiques UI / UX Cohérence visuelle
Intégration de maquette	<ul style="list-style-type: none">- Mobile first- Responsive design
Accessibilité (RGAA)	Bonnes pratiques ARIA, contrastes, navigation clavier
SEO	<ul style="list-style-type: none">- Structure sémantique- Titres hiérarchisés, métadonnée- Respect des bonnes pratiques SEO
Langages client (HTML, CSS, JS)	Conformité W3C

JavaScript avancé	<ul style="list-style-type: none">- Syntaxe JavaScript moderne- Contacter une API distante
-------------------	---

Résumé du projet

Vous devez réaliser une **mini-boutique web** composée de **3 pages** :

1. Page d'accueil (Home)
2. Liste des produits
3. Détail d'un produit

Le projet se déroule en **deux étapes obligatoires** :

1. Création d'une **maquette** (Figma ou équivalent)
2. **Intégration complète** de cette maquette dans une application HTML/Javascript ou **React**

Le site doit être **responsive**, accessible, performant et conforme aux standards (W3C, RGAA, SEO).

Soutenance orale du projet

Le projet fera l'objet d'une présentation orale le 15 janvier 2026. Lors de cette soutenance, vous présenterez la maquette, l'application réalisée, ainsi que vos choix techniques en termes d'accessibilité, de performance et de SEO.

À propos des spécifications

Les spécifications détaillées servent à guider la création de la maquette.

Elles indiquent les sections à prévoir, la structure attendue et les éléments fonctionnels qui doivent apparaître dans l'interface.

La maquette doit donc reprendre l'intégralité des éléments décrits, et servira de référence pour l'intégration dans l'application HTML/Javascript ou React.

Spécifications → Maquette → Intégration HTML/Javascript ou React

Voir Annexe 1 : Spécifications fonctionnelles

Sujet

L'entreprise GreenMarket souhaite promouvoir la consommation responsable en proposant une boutique en ligne dédiée à des produits éthiques et durables :

- Cosmétiques naturels
- Vêtements recyclés
- Accessoires zéro déchet

Elle souhaite disposer d'une application web responsive permettant aux utilisateurs de :

- Accéder à la page d'accueil
- Consulter un catalogue de produits
- Afficher le détail d'un produit

Le projet doit respecter :

- Les normes W3C, RGAA,
- Les bonnes pratiques d'éco-conception, de SEO et d'accessibilité

Livrables attendus

1. Maquette de l'interface

Réalisée sur Figma, Adobe XD ou équivalent.

Pages obligatoires :

- Page d'accueil (Home)
- Liste des produits
- Détail d'un produit

Critères :

- Ergonomie
- Responsive design
- Accessibilité (contraste, navigation clavier, ARIA)
- Cohérence graphique avec l'identité de GreenMarket

2. Application web fonctionnelle — Front-End HTML/JAVASCRIPT React

Intégration fidèle de la maquette dans votre application, respectant la structure, les styles et le responsive.

Pages obligatoires :

- Page d'accueil (Home)
- Page liste des produits (Products)
- Page détail d'un produit (Product Detail)

Données API :

L'API disponible pour la récupération des articles est : <https://fakestoreapi.com/docs>

Endpoints :

- **GET** /products
- **GET** /products/:id

Important :

- Il est possible que l'API renvoie des articles différents à chaque requête ou bien hors sujet avec le thème de GreenMarket (Cela sera pris en compte lors de la correction)
- Aucune fonctionnalité de panier n'est demandée dans ce projet.

Attendus techniques

Intégration & développement

- Capacité à reproduire fidèlement la maquette
- Respect des espacements, proportions, couleurs, typographies
- Mise en page responsive
- Développement d'une interface cohérente et fonctionnelle

Application HTML/JS ou React

- Appels API (fetch ou axios)
- Gestion du chargement et des erreurs
-

Accessibilité

- Attributs ARIA pertinents
- Navigation clavier
- Contrastes suffisants
- Texte alternatif pour les images

SEO

- Structure sémantique correcte
- Titres hiérarchisés
- Balises meta
- Bonnes pratiques SEO (performance, lisibilité, structure)

Performance

- Assets optimisés
- Bonne structure du code
- Score Lighthouse recommandé : supérieur à 80

Obligation : Ajouter les métriques Lighthouse dans le repository

Chaque élève doit intégrer **dans le dépôt Git** un dossier contenant les résultats de Lighthouse.

Exemples acceptés :

- Rapport Lighthouse exporté (JSON, HTML ou PDF)
- Captures d'écran des scores (Performance, Accessibilité, Best Practices, SEO)

Dossier recommandé :
/docs/lighthouse/

Contraintes

- Normes : W3C, RGAA
- Responsive obligatoire

Livrables finaux

- Maquette complète (Figma)
- Application HTML/js ou React fonctionnelle
- Code source bien structuré
- Rapport technique (SEO, accessibilité, choix techniques)
- Dossier Lighthouse obligatoire

1. Modalités de rendu (obligatoires)

Dépôt GitHub

Le rendu devra obligatoirement être effectué sous la forme d'un **repository GitHub** contenant :

- Le code source complet du projet

- La maquette Figma (lien dans un fichier [README.md](#))
- Le dossier Lighthouse : [/docs/lighthouse/](#)
- Le rapport technique (SEO, accessibilité, choix techniques)
- Un README clair contenant :
 - o Une description du projet
 - o Les instructions d'installation
 - o Les instructions de lancement
 - o Le lien vers la version déployée
 - o Les technologies utilisées

Déploiement

L'application doit être **déployée en ligne** via l'une des plateformes suivantes :

- **Vercel** (recommandé pour React)
- **Netlify**
- GitHub Pages (HTML/CSS/JS uniquement)

Le lien du déploiement doit être indiqué dans le [README.md](#).

L'application doit être fonctionnelle au moment de la soutenance.

2. Modalités d'évaluation

Critères éliminatoires

Les points suivants peuvent entraîner une pénalité sévère ou une non-validation :

- Projet non rendu
- Dépôt GitHub manquant ou incomplet
- Pages obligatoires absentes
- Absence des rapports Lighthouse
- Non-participation à la soutenance

Critères de qualité attendus

- Code commenté, clair et structuré
- Respect strict des normes W3C
- Score Lighthouse ≥ 80 dans au moins 3 catégories

- Aucun *alt* manquant
- Navigation complète au clavier

3. Consignes officielles de la soutenance

Format

- Durée totale : **25 minutes**
 - o 15 minutes de présentation
 - o 10 minutes de questions du jury

Contenu attendu dans la soutenance

Vous devrez présenter :

- Votre maquette (structure, choix graphiques, cohérence UI/UX)
- Votre application (démonstration en direct)
- L'architecture et les choix techniques
Votre démarche d'accessibilité (ARIA, clavier, contrastes)
- Votre démarche SEO
- Vos scores Lighthouse et comment vous les avez améliorés

Aucun autre support n'est accepté que :

- la maquette
- le site en ligne
- le dépôt GitHub
(sauf si autorisation spéciale)

Évaluation orale

L'oral permet d'évaluer :

- votre compréhension du projet
- votre autonomie
- votre capacité à expliquer vos choix
votre maîtrise technique

4. Conditions officielles du déroulement de l'examen

Travail individuel

Ce projet est **strictement individuel**, Toute similarité suspecte entraînera :

- une convocation
- une possible invalidation de l'examen

Technologies autorisées

- HTML / CSS / JS
- React

Pas d'autres frameworks (Vue, Angular, Svelte) sauf validation préalable.

Aides autorisées

- Documentation officielle
- Exercices et TP réalisés en cours.
- Ressources W3C / RGAA / MDN
- Vous devez être capable d'expliquer chaque ligne de votre code.