

Construire une mini-application météo

Domaine : Développement d'applications / Services numériques / Interfaces Web

Compétences professionnelles mobilisées

Développer des solutions logicielles

- Lire, analyser, modifier un code existant
- Implémenter des fonctionnalités basées sur des API Web
- Utiliser des outils et des modèles de programmation adaptés (HTML/CSS/JS)
- Respecter les bonnes pratiques de lisibilité et organisation du code

Déployer, tester, maintenir un service numérique

- Intégrer un service REST externe (API)
- Tester et valider le fonctionnement d'une application Web
- Gérer les erreurs de service et proposer des comportements alternatifs
- Vérifier la compatibilité multi-supports (desktop / mobile)

Communiquer dans un contexte professionnel

- Documenter ses choix techniques
 - Présenter une solution logicielle à un client ou formateur
-

Contexte de l'activité

Vous travaillez au sein d'une petite équipe responsable d'un **service Web d'information météorologique**.

Le développeur principal vous confie un **prototype fonctionnel mais non mis en forme**.

Votre rôle est de transformer ce prototype en **solution présentable à un client** (interface + lisibilité + robustesse).

Le code existant est minimal : il affiche la météo d'une **ville fixe** via l'API Open-Meteo.

Votre mission est d'en faire **une interface agréable, claire et lisible**.

Livrables attendus

1. **Une page HTML fonctionnelle**
2. **Un fichier CSS** avec un design personnel
3. **Un fichier JS** qui réutilise le code fourni
4. **Un court document expliquant :**

- les choix de mise en page
 - les ajouts personnels
 - les difficultés rencontrées
-

Travail demandé

1) Analyse du code fourni

Vous devez :

- lire l'ensemble des fichiers HTML / CSS / JS fournis
 - identifier les sections importantes (structure, API, ...)
 - rédiger un court **rapport d'analyse (5 à 10 lignes)**
 - rôle du fichier HTML
 - rôle du JS (API, affichage, prévisions)
 - données renvoyées par l'API
 - points faibles / points d'amélioration possibles
-

2) Création d'une interface utilisateur

Vous devez créer une **mise en forme complète (CSS)** en respectant les contraintes suivantes :

Obligatoire :

- structurer clairement :
 - nom de la ville
 - météo du jour
 - icône
 - description
 - vent
 - prévisions 4 jours
- rendre la page **responsive**
- choisir une **charte graphique cohérente** (couleurs, polices, espacement)
- assurer la lisibilité sur mobile

Recommandé :

- utiliser flexbox ou grid
- créer une interface moderne ou minimaliste
- ajouter transitions / ombres / encadrements simples

Aucun framework CSS (Bootstrap, Tailwind) n'est autorisé. L'objectif est de manipuler **HTML + CSS pur.**

3) Améliorations techniques

Vous devez ajouter **au minimum 3 améliorations** parmi cette liste :

Améliorations possibles

- Améliorer la gestion des erreurs (si l'API échoue → afficher un message clair)
- Mettre les températures en couleur (bleu froid, rouge chaud)
- Ajouter un message "Chargement..." pendant la récupération des données
- Ajouter une animation d'apparition
- Ajouter une icône dans la prévision pour chaque jour
- Ajouter l'humidité ou le ressenti (si disponible dans l'API)
- Ajouter un mode sombre / clair
- Ajouter la date complète ("Lundi 12 février") pour la météo du jour

Vous devez expliquer au minimum **pourquoi** trois améliorations ont été choisies.

4) Tests & validation

Vous devez vérifier :

Fonctionnel :

- l'affichage de la météo actuelle
- l'affichage des prévisions
- le fallback en cas de code météo non reconnu
- les améliorations ajoutées

Technique :

- aucune erreur JS dans la console
- la page fonctionne sous Chrome, Firefox et un navigateur mobile
- la page reste utilisable sans CSS (critère minimal d'accessibilité)

Qualité :

- nommage correct des classes / IDs
 - code commenté si nécessaire
 - indentation propre
-

5) Communication & documentation

Vous devez fournir à la fin :

Un mini-fichier (2 pages maximum) contenant :

- les objectifs des modifications réalisées

- les choix graphiques
- les choix techniques (améliorations ajoutées)
- un screenshot de la version finale
- les éventuelles difficultés rencontrées

Ce dossier sert d'entraînement à la rédaction professionnelle BTS CIEL.

Les fichiers HTML, CSS et JS

Évaluation (barème)

Critère	Points
Analyse du code et compréhension	2 pts
Mise en page HTML claire	2 pts
Design CSS propre, lisible, responsive	7 pts
Améliorations techniques (3 min.)	4 pts
Qualité du code JS / organisation	2 pts
Document explicatif	3 pts
Total	20 pts
