

Sujet d'examen : WeatherNow - Dashboard Météo

Durée totale : 6 heures

Technologies : HTML, CSS, JavaScript, React

Données externes : API OpenWeatherMap

Objectif global : Concevoir puis développer une application web complète, moderne et responsive, affichant la météo d'une ville saisie par l'utilisateur.

Consignes générales (obligatoires pour les 2 parties)

- *Le site doit être responsive (mobile first, puis adaptation desktop)*
- *UX et UI soignées : lisibilité, espacement, hiérarchie visuelle*
- *Code clair, commenté et structuré*
- *Nom de l'application à afficher sur l'interface : WeatherNow*
- *L'application doit être en français (labels, messages d'erreur...)*

Partie 1 - Version HTML / CSS / JavaScript (3h)

Dossier attendu : /static-version/

Version autonome contenant tous les fichiers nécessaires à l'exécution

Fonctionnalités requises

Page d'accueil — éléments à afficher

1. En-tête
 - Nom de l'application : WeatherNow
2. Zone de recherche
 - Champ de saisie pour taper une ville
 - Bouton “Rechercher”
3. Menu avec au minimum :
 - Onglet Accueil
 - Onglet Historique
4. Historique des recherches
 - Sauvegarde automatique en LocalStorage

Comportement

- Utilisation de fetch() pour appeler l'API OpenWeatherMap

- Affichage des données météo de la ville saisie
- Gestion des erreurs obligatoire
 - Ville inconnue → message d'erreur visible et clair
 - Empêcher la recherche si le champ est vide

Exigences UX/UI

- Mise en page professionnelle (Flexbox ou CSS Grid)
- Couleurs cohérentes, contrastes lisibles
- Police web moderne (Roboto, Inter ou équivalent)
- Design en carte pour les informations météo
- Organisation claire des pages via le menu

API à utiliser :

https://api.openweathermap.org/data/2.5/weather?q={ville}&units=metric&appid={clé_API}

Partie 2 - Version React (3h)

Dossier attendu : /react-version/

Design et fonctionnalités identiques à la partie 1

Code découpé en composants

Contraintes techniques

- Projet créé via Vite ou Create React App
- Utilisation obligatoire des hooks : useState, useEffect
- Structure minimale des composants :

App

```

  └── SearchBar
  └── WeatherCard
    └── HistoryList

```

Exigences fonctionnelles

- Appel API identique à la version statique
- Gestion des erreurs (ville introuvable)
- Gestion de l'historique via LocalStorage

Structure du code

- Variables d'état pour : ville saisie, données météo, erreur éventuelle, historique
- Fonction dédiée à l'appel de l'API
- Aucune duplication de code inutile

Livrables finaux

Projet compressé en ZIP comportant les dossiers suivants :

Weathernow/

 |— static-version/

 └— react-version/

Chaque dossier doit être fonctionnel sans installation supplémentaire.

Conclusion

Le but est de démontrer :

- La capacité à concevoir une interface web moderne
- La compréhension du DOM (partie 1)
- La maîtrise des composants React et des hooks (partie 2)
- L'intégration d'une API réelle