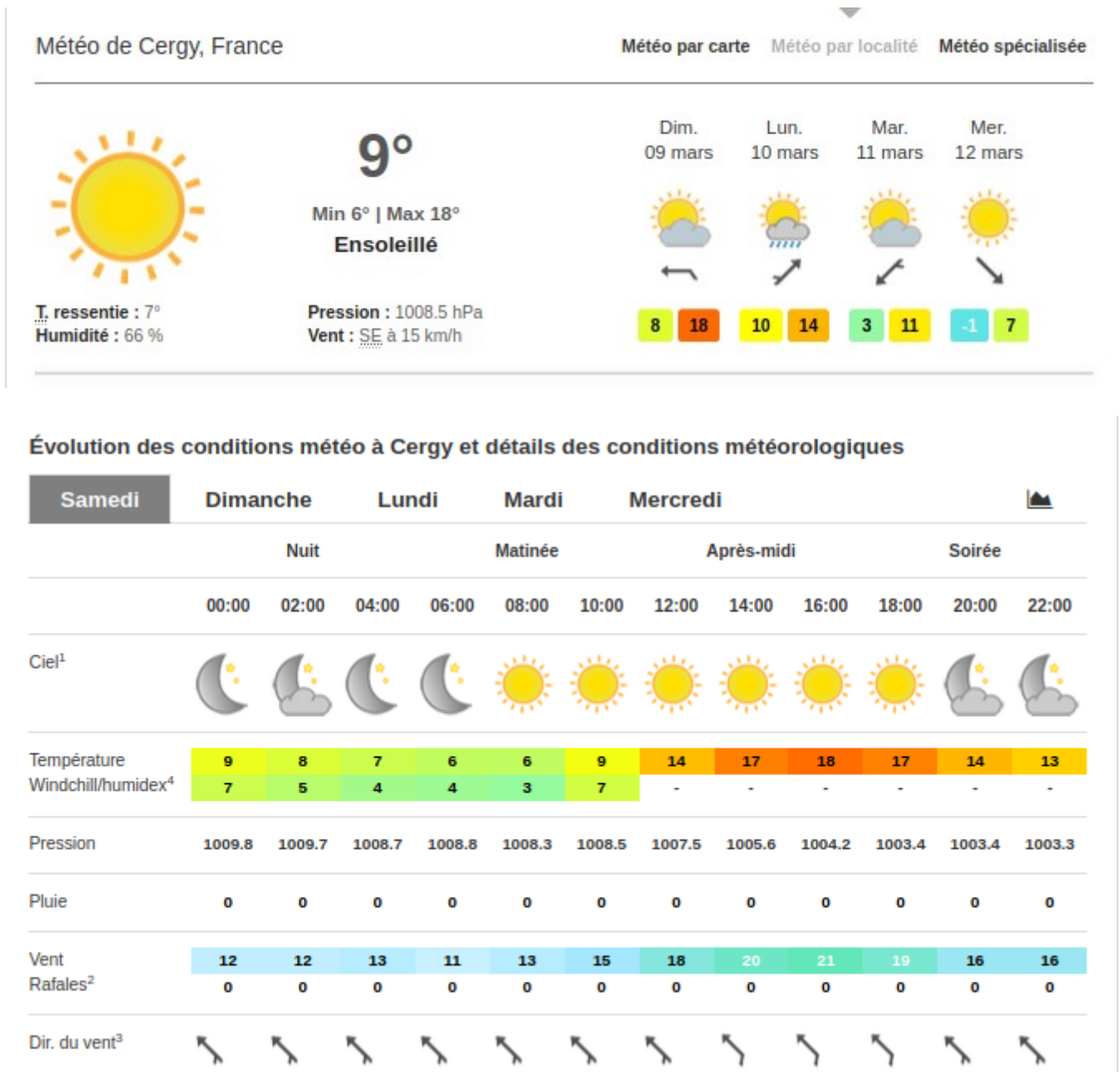


# Projet UE Web "Prévisions Météo & Climat"

L2 I - S4 - 2024-2025 - v. 1.01 du 2 avr. 2025



## Objectif

Le but du projet est de mettre en œuvre l'ensemble des éléments techniques de l'UE « Développement Web » de L2-I : HTML 5 / CSS 3 / PHP 8 dans le cadre d'une réalisation en binôme. Le site réalisé s'appuiera sur plusieurs API<sup>1</sup>.

<sup>1</sup> API : Application Programming Interface.

## Contexte

De nombreuses activités nécessitent de connaître les prévisions météorologiques à quelques jours, en particulier : les métiers du bâtiment (BTP), l'agriculture, les sports en extérieur, de nombreuses situations de la vie quotidienne, etc.

Le contexte du projet est un site web orienté sur les prévisions météo en France métropolitaine. L'utilisateur sélectionne sa région, son département puis sa ville et le serveur web lui affiche la météo du jour ainsi que les prévisions sur les jours suivants. Pour cela vous vous appuyerez sur les ressources disponibles en ligne (cf. Quelques sites d'API qui peuvent être utilisées).

## Travail à réaliser et fonctionnement du site

### Première partie : prise en main des formats d'échanges JSON et XML des API Web

De nombreuses solutions d'API (*Application Programming Interface*) web de type REST s'appuient sur le format d'échange JSON (*JavaScript Object Notation*) : c'est le cas par exemple de l'API "APOD" (*Astronomy Picture of The Day*) proposée par la NASA (*National Aeronautics and Space Administration*) : vous pouvez tester le lien suivant (avec Firefox) :

[https://api.nasa.gov/planetary/apod?api\\_key=DEMO\\_KEY&date=2025-03-15](https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY&date=2025-03-15)

- cliquez sur l'élément « url » pour visualiser l'image associée
- cliquez sur l'onglet « Raw Data » pour visualiser le flux JSON (non mis en forme par le navigateur).

Afin de prendre en main ce type d'échange, il est demandé de construire une page « *developper* » ou « *tech* » (ou tout autre nom adapté que vous choisirez !) dont le lien sera dans le footer de votre site de projet et de placer l'image du jour du service « APOD » et son texte explicatif sur cette page (inutile d'utiliser l'image en HD !).

Remarque : la clef « DEMO\_KEY » ne permet qu'un nombre réduit de requêtes : il sera nécessaire d'activer une clef valide sur le site de la NASA.

N.B. : Pour le site « apod », parfois l'image du jour est une vidéo<sup>2</sup> : vous devrez alors utiliser la balise HTML 5 « video » pour insérer la vidéo dans votre page (ou à défaut utiliser le paramètre « &thumbs=True » dans l'URL) une troisième situation existe avec « media-type:other » : vous placerez alors une image par défaut dans ce cas.

Toujours dans le même objectif de prise en main des API et des flux, ajoutez une autre fonctionnalité sur cette même page « tech » de votre site du projet afin de mettre en œuvre une API utilisant cette fois un flux XML (*eXtended Markup Language*) pour la réponse : vous pouvez identifier la position

<sup>2</sup> Il existe également un cas particulier pour lequel le media-type est video mais sans un fichier vidéo (ex. [https://api.nasa.gov/planetary/apod?api\\_key=DEMO\\_KEY&date=2025-03-02](https://api.nasa.gov/planetary/apod?api_key=DEMO_KEY&date=2025-03-02)) vous pourrez alors utiliser la balise <iframe> si vous souhaitez prendre en compte cette situation.

géographique approximative de votre visiteur via son adresse IP : cf. <https://www.geoplugin.com/webservices/xml> ex. : <http://www.geoplugin.net/xml.gp?ip=193.54.115.192> : ajoutez cette information sur la page « tech » de votre site.

Enfin deux autres situations sont proposées toujours dans le même objectif de maîtrise des requêtes à une API :

<https://ipinfo.io/193.54.115.192/geo> ; pour une extraction simple d'informations depuis un flux JSON : l'adresse IP dans l'URL est à remplacer par celle de l'internaute pour lui indiquer une estimation de sa position géographique.

<https://api.whatismyip.com/ip-address-lookup.php?key=KEY&input=193.54.115.235&output=xml> ; pour une extraction simple d'informations depuis un flux XML (dans ce dernier exemple, KEY est à remplacer par une valeur de clef valide, à activer sur le site correspondant).

**N.B.** : cette première partie technique, indispensable au projet, sera validée lors du premier point d'avancement.

Vous placerez donc ces fonctionnalités dans une page « tech » spécifique (à part) de votre site afin de conserver ces exemples et qu'ils puissent être validés lors du premier point d'avancement ainsi qu'au moment de livrer le projet.

## Deuxième partie : Météo d'une ville

Les données statiques d'entrée (région / département / ville) seront issues de fichiers CSV (cf. Annexes). Les informations météorologiques (données dynamiques) sont à récupérer sur un site dédié (cf. Annexes) et à mettre en forme.

L'utilisateur du site aura la possibilité de :

- Sélectionner sa région sur une carte interactive à l'aide des tags HTML map, usemap et area (cf. <https://developer.mozilla.org/fr/docs/Web/HTML/Element/map>)
- Sélectionner son département : vous devrez définir l'ergonomie la plus adaptée
- Sélectionner sa ville dans une liste déroulante



*Illustration 1: carte administrative des régions (France) - source :  
IGN*

La page doit afficher les prévisions météo du jour sur la ville sélectionnée ainsi que la prévisions sur les jours suivants

Les échanges se feront impérativement entre votre serveur Web et l’API (éventuellement plusieurs à agréger, cf. Extensions possibles). Les réponses utiliseront principalement le format JSON mais vous devrez également exploiter le format XML afin de vous permettre de maîtriser les 2 principaux flux d’échanges des API disponibles dans le contexte du Web (cf. Première partie : prise en main des formats d’échanges JSON et XML des API Web)

## Troisième partie : stockage

Votre solution devra en outre réaliser un stockage côté serveur, sur fichier, des villes consultées par l’ensemble des utilisateurs du site. Vous utiliserez impérativement un format CSV<sup>3</sup> pour votre stockage.

Afin d’améliorer l’ergonomie et l’expérience utilisateur<sup>4</sup>, vous ajouterez le stockage côté client de la dernière ville consultée par l’internaute via un mécanisme de **cookie** (ville consultée, date de la consultation, etc.)

Enfin, vous prévoirez la génération **côté serveur** d’un histogramme des villes les plus consultées sur le site. Vous afficherez ce graphique dans une rubrique « statistiques » de votre site.

<sup>3</sup> CSV : Comma Separated Values

<sup>4</sup> UX : user experience.

## Quatrième partie : compléments TD

### a Image aléatoire

Sélectionnez dynamiquement dans un dossier "photos" une image au hasard (sur le thème du projet !) à afficher sur la page d'accueil de votre site (comme cela se pratique sur de nombreux sites<sup>5</sup>), par exemple dans la zone « aside » (le rafraîchissement de la page entraînera l'affichage aléatoire d'une autre image). Vous utiliserez les balises <figure> et <figcaption> avec le nom du fichier par exemple.

N.B. : le script PHP ne connaît pas à l'avance le contenu du dossier « photos », il faut donc explorer dynamiquement ce dossier et il est conseillé d'utiliser des tableaux PHP pour traiter ce problème.

### b cookie

Reprenez l'exercice permettant le choix de la charte graphique (mode jour / mode nuit) en mettant en place un cookie afin de permettre une mémorisation du choix de la charte graphique (celle-ci étant conservée sur les pages du site et lors d'une visite ultérieure depuis le même navigateur).

Le cookie doit donc être mis en place ou modifié via un lien (sur une image) permettant de modifier son choix, associé à la méthode « set\_cookie ».

La relecture du cookie si il existe doit être considéré lors d'une visite ultérieure (vous pouvez fermer puis rouvrir votre navigateur pour simuler cette situation).

Si la valeur du cookie est erronée, supprimez-le.

N.B. : le cookie ne doit être valable que pour votre espace (ex. dans « /~login/ » dans le cas d'un hébergement partagé) et non pour la totalité du serveur puisque votre serveur peut héberger plusieurs sites simultanément.

## Recommandations fonctionnelles

L'utilisateur pourra personnaliser son affichage du résultat : ex. n'afficher que les informations générales ou au contraire obtenir plus de détails sur la météo d'une ville.

Ses préférences ainsi que la dernière ville consultée seront sauvegardées sur son navigateur.

Une partie « statistiques » permettra de visualiser sous forme graphique l'utilisation du site : notamment le nombre total de visiteurs ainsi que les villes les plus fréquemment consultées.

Le site obtenu devra être esthétique et ergonomique.

Vous n'oublierez pas de choisir un nom à votre site et donc un logo et un favicon !

## Recommandations générales de gestion de projet

- Définir les pages les plus importantes de votre site en terme de fonctionnalités et organiser la répartition des tâches au sein de votre binôme-projet.

---

5 cf. par exemple : <https://www.cyu.fr/en>

- Définir la charte graphique et l'ergonomie du site ainsi que les maquettes (*mockup*) des principales pages.
- Définir le planning avec les principales échéances et évaluer les risques majeurs. Vous n'oublierez pas de prévoir les phases de validations intermédiaires (points d'avancement) afin de vous assurer de la conformité avec la demande en fonction de vos choix.

Le but est donc de concevoir un site web fonctionnel pour lequel il vous appartient d'imaginer des cas d'utilisation permettant de répondre au besoin exprimé.

## Résultat attendu

Le projet devra s'intégrer dans une architecture web 3 tiers (navigateur / serveur Web / APIs), la solution côté client devra être valide aussi bien HTML5 que CSS3 (au niveaux syntaxique, validateurs du W3C, XML et structurelle) et la solution côté serveur devra s'appuyer sur PHP 8. L'organisation pour la phase de développement est libre mais le résultat devra être disponible **régulièrement** sur au moins un des comptes des membres de l'équipe projet. Vous veillerez à la validation de l'accessibilité et également des performances de vos pages (cf. détails lors du TD #10).

## Extensions possibles

Vous pouvez prévoir d'exploiter d'autres sources de données afin que votre site constitue une agrégation de ces différentes sources (alerte météo, vigicrue, etc.) ou en lien avec les problématiques de changement climatique.

## Livrables

- Le site devra être en ligne sur Internet (à priori même URL que pour les TD donc à terme le site des TD sera déplacé dans un sous-dossier « td » de votre hébergement).
- Les codes PHP du projets et autres fichiers du site dans leur arborescence fonctionnelle => la simple copie de l'arborescence du site (après extraction) doit donc être opérationnelle.
- La documentation du code au format HTML (générée à partir des commentaires phpdoc).
- Un mini-rapport de quelques pages (entre 5 et 10 pages hors annexes éventuelles, au format odt ou à défaut docx ainsi qu'une version pdf : les sections et sous-sections doivent être automatiquement numérotées) permettant de faire une synthèse du projet aussi bien au niveau technique qu'organisationnel (gestion de projet). Ce document devra en particulier intégrer les points suivants (liste non exhaustive) : les membres de l'équipe-projet, la répartition des tâches et l'organisation (diagramme de Gantt), l'analyse du sujet à l'aide d'un diagramme de cas d'utilisations, les particularités et les choix techniques, le plan du site, les « *mockup* » des pages, etc. Le rapport doit être pensé comme un support technique professionnel et ne doit pas être un manuel utilisateur (il n'y a pas de manuel utilisateur d'un site Web !)
- Un fichier *markdown* « readme.md » contenant les informations principales (noms des auteurs du projet, les URL du site sur Internet, etc.).

- L'ensemble sera rendu sous la forme d'une archive nommée « grp\_prj\_###\_X\_Y.zip » , avec ###, le numéro (y compris la lettre du groupe TD) de votre groupe projet et X, Y les NOMS - prénoms des membres du binôme projet.

## Planning

1. Un premier point d'avancement est prévu en semaine 13 (séance TD #10) afin de pouvoir s'assurer de la conformité du cahier des charges avec la demande et valider la première partie (cf. Première partie : prise en main des formats d'échanges JSON et XML des API Web).
2. Des points d'avancement sont prévu à chaque séance de TD à partir du TD #10.
3. Une « *release* » intermédiaire est prévue lors de la dernière séance de TD (TD #12 ; semaine 15) avec démonstration de l'avancement du projet<sup>6</sup>. Lors de ce dernier point d'avancement, à titre indicatif, au moins 50 % du projet devra être opérationnel.
4. Les livrables du projet sont à rendre en fin de semaine 16 (au plus tard, le dimanche 20 avril 2025 ).
5. Les soutenances auront lieu en semaines 18 (dates prévisionnelles : lundi 28 avril 2025 et mardi 29 avril 2025).

NB : à **chaque étape**, le travail attendu doit être disponible en ligne.

## Critères d'évaluation

- Niveau de réalisation (pourcentage de couverture du sujet).
- Originalité de la solution.
- Maîtrise des éléments techniques de l'UE (HTML 5, CSS 3, PHP 8).
- Ergonomie et esthétique du résultat.
- Qualité technique sur le navigateur (« *client-side* ») : absence de bug et validation des pages (HTML / CSS / XML).
- Qualité technique de l'ensemble des scripts PHP (« *server-side* ») et génération de la documentation (au format HTML) des fonctions utilisées (cf. doxygen).
- Niveau technique spécifique au projet (solution technique, affichage graphique des résultats, etc.).
- Vidéo **scénarisée** de la démonstration de la réalisation de 3 min (180 sec) max en étant vigilant à la répartition du temps de parole de chaque membre du binôme, à rendre avant le dimanche 20 avril 2025
- Soutenance avec support visuel de quelques diapositives (5 *slides* maximum) de 5 min max.

---

<sup>6</sup> Les projets qui n'auraient pas fait l'objet de points d'avancement aux dates fixées ne pourront pas participer à la soutenance.

## Annexes

### Quelques sites d'API qui peuvent être utilisées

- <https://www.weatherapi.com/>
- <https://www.prevision-meteo.ch/>
- <https://api.open-meteo.com/>
- <https://openweathermap.org/>
- <https://api.tutiempo.net/>
- <https://api.meteo-concept.com/>
- <https://www.infoclimat.fr/>
- <https://portail-api.meteofrance.fr/>

### Quelques sites de ressources qui peuvent être utilisées

- <https://www.data.gouv.fr/en/datasets/listes-des-communes-par-r-gions-d-partements-circonscriptions/>
- [https://www.data.gouv.fr/fr/datasets/regions-departements-villes-et-villages-de-france-et-doutre-mer/#\\_](https://www.data.gouv.fr/fr/datasets/regions-departements-villes-et-villages-de-france-et-doutre-mer/#_)
- <https://www.data.gouv.fr/fr/datasets/nomenclature-code-des-nouvelles-regions-2016/>
- <https://www.data.gouv.fr/es/datasets/base-officielle-des-codes-postaux/>
- <https://www.insee.fr/fr/information/3363419#titre-bloc-26>
- <http://education.ign.fr/ressources/fonds-de-cartes#Fonds1>

### Ressources pour les *mockup*

- <https://www.figma.com/>
- <https://wireframe.cc/>
- <https://www.canva.com/>
- <https://mockitt.com/>

### Solution d'hébergement possible

- <https://www.alwaysdata.com/fr/>