

Cahier des charges du projet Web PréviZen



Cursus : Licence 2ème année, UE Développement Web

Réalisé par : Albrun Mathis & Khelil Imène

Enseignant :

M. Lemaire

Année universitaire : 2024 - 2025

1. Introduction.....	2
1.1. Contextualisation.....	3
1.2. Objectifs et analyse des besoins.....	3
2. Architecture et fonctionnalités.....	4
2.1. Organisation technique.....	4
2.2. Architecture globale.....	4
2.3. Approfondissement des fonctionnalités clés.....	5
3. Stockage et statistiques.....	6
3.1. Affichage des diverses statistiques.....	7
3.2. Gestion des cookies.....	7
4. Qualité d'usage et évolutivité du site.....	7
4.1. Conformité technique.....	8
4.2. Expérience utilisateur.....	8
4.3. Gestion du projet.....	8
4.4. Perspectives.....	8
5. Conclusion.....	9
6. Annexe.....	10

1. Introduction

1.1. Contextualisation

La météo est un élément omniprésent dans notre quotidien. Allant de la recherche de la température actuelle en passant par la prévision des conditions pour effectuer une activité en extérieur, nous avons tous déjà eu affaire à de nombreuses applications et sites Internet consacrés à la météo. C'est donc dans ce contexte que vient s'inscrire notre site internet : PreviZen. Réalisé dans un cadre universitaire, en seconde année de Licence Informatique à CY Cergy Paris Université, ce projet "Prévisions Météo & Climat" a pour but de déployer un site internet fonctionnel permettant aux internautes de consulter les données météo en fonction des zones géographiques sélectionnées (région, département, ville).

1.2. Objectifs et analyse des besoins

La réalisation d'un tel site internet de prévision météo poursuit plusieurs objectifs pédagogiques, techniques et fonctionnels. En effet, cela nous pousse à mettre en œuvre différentes compétences acquises au cours du semestre dans l'UE Développement Web. Afin de parvenir à la création de notre site, nous avons utilisé divers langages comme de l'HTML5 (HyperText Markup Language) et CSS3, permettant de confectionner nos pages de manière pertinentes et fonctionnelles. Le langage PHP8 joue également en rôle central au sein de ce projet en assurant le dynamisme de l'ensemble de notre site coté serveur et dans la communication entre les différentes interfaces de programmation d'application (API). En effet, ces interfaces donnent accès le stockage de divers types de données, ici des données météo, et sont accessibles via des requêtes HTTP de type GET. Suite à cela, les API retournent les informations souhaitées sous forme de flux JSON ou XML, des formes exploitables pour un langage tel que le PHP. Dans le cadre d'un site météo, ces récupérations de données sont essentielles afin de garantir la précision des informations données aux utilisateurs. On retrouvera également au sein de notre projet du JavaScript, nous permettant une certaine complémentarité avec le PHP central. PreviZen garantit plusieurs accès à la météo comme notamment via une page de météo local, de météo des plages ou encore de météo des neiges. Il est également possible de consulter les prévisions climatiques à travers l'ensemble du

territoire métropolitain par le biais d'une carte interactive. Pour cela, il suffit de cliquer sur une région, de choisir le bon département et de renseigner sa ville pour bénéficier d'une météo sur 7 jours. La gestion des divers éléments de ces pages et de la structure globale du site a nécessité quelques adaptations techniques afin de garantir une expérience confortable. En effet, l'accessibilité et l'ergonomie ont été les maîtres mots durant la période de réalisation de PreviZen, au même titre que la responsivité. 2. Architecture et fonctionnalités

2.1. Organisation technique

Afin de mener à bien notre projet, nous avons porté une attention particulière à la conception technique de notre arborescence de fichiers. En nous basant sur une analyse du sujet¹, nous avons fait le choix d'opter pour une organisation facilitant le débogage et la maintenance du site, tout en permettant la localisation rapide des potentielles sources d'erreurs. Ainsi, cinq répertoires se sont naturellement dégagés :

- *data/* : Ce dossier nous permet de regrouper les fichiers de stockage de données pour les régions, les départements ou encore les communes. Ces derniers sont essentiels pour permettre le choix dynamique des prévisions météo. On retrouvera également le stockage du nombre de visites de notre site, une donnée importante afin de mesurer l'efficacité de notre projet.
- *include/* : Ce répertoire est au cœur de notre site. En contenant l'ensemble des fichiers de traitement PHP, ce répertoire garantit la gestion des flux JSON ou XML ainsi que l'accès aux structures sémantiques des blocs partagés comme le header ou le footer de nos pages web.
- *js/* : Ce dossier contient les scripts JavaScript des cartes dynamiques présentes dans les sections de météo de neiges et des plages. Ces scripts s'appuient sur la bibliothèque OpenLayers pour proposer une cartographie interactive fluide et informative, renforçant l'immersion de l'utilisateur dans les pages *mer.php* et *neige.php*. Ainsi, les principales stations balnéaires et

¹ NB: voir Annexe, figure 2
Mars 2025

montagnardes sont facilement localisables, donnant accès à certaines données météo par le biais de popups.

- *style/* : Accueillant les feuilles de styles CSS adaptées au style de jour et de nuit, il permet d'appliquer une charte graphique adaptative selon les préférences de l'utilisateur.
- *images/* : Ce répertoire regroupe l'ensemble des éléments graphiques de notre site comme les logos, les icones ou encore les images.

Chaque dossier a une responsabilité bien définie, mais les fichiers qu'ils contiennent sont utilisés de manière transversale dans les différentes pages du site.

2.2. Architecture globale

L'architecture de PreviZen repose sur une série de pages structurées autour de cas d'usage bien identifiés. Chaque page a été conçue pour répondre à un besoin spécifique des utilisateurs, tout en s'intégrant harmonieusement dans la navigation globale du site. La page d'accueil *index.php* en est le point d'entrée. Elle propose une vue d'ensemble intuitive avec un accès rapide à la météo locale, un aperçu visuel de la France via une carte interactive, et une sélection dynamique par région, département et ville. L'objectif de cette page est de guider l'utilisateur dès son arrivée vers la section la plus pertinente.

La page *local.php* est dédiée à l'affichage des prévisions météo détaillées en fonction de la géolocalisation ou par sélection manuelle. L'utilisateur y accède via une sélection hiérarchique intuitive, alimentée dynamiquement par des fichiers CSV internes. Une fois la ville sélectionnée, la page affiche les données actuelles (température réelle et ressentie, vent, pictogramme), ainsi qu'un découpage par tranches horaires et une prévision complète sur 7 jours. Toutes ces données sont récupérées en temps réel depuis l'API *WeatherAPI*, puis traitées en PHP pour garantir un rendu rapide, clair et accessible. Afin compléter ces données météo, nous proposons une page *mer.php*² dédiée à la météo des plages. Après avoir sélectionné une zone côtière (Manche, Atlantique ou Méditerranée), l'utilisateur accède aux principales stations balnéaires de cette zone. Il peut y consulter la température de l'air et de l'eau, l'état

² NB: voir Annexes, figure 4

de la mer, le vent et l'indice UV. De plus, l'intégration d'une carte interactive basée sur *OpenLayers* permet une visualisation rapide des conditions venteuses, via des cercles colorés indiquant la force du vent. Dans un fonctionnement similaire, le service de visualisation des conditions neigeuses assuré par *neige.php* est organisé par massif montagneux (Alpes, Pyrénées, Jura, etc.). En effet, cette page fournit des données comme la hauteur de neige prévue ou la météo globale. Là encore, une carte interactive permet de localiser les principales stations françaises d'un coup d'œil et d'accéder directement à leurs prévisions. Enfin, la page *air.php* propose une analyse de la qualité de l'air en inscrivant les quantités de certains gazs nocifs. En utilisant le flux JSON de *WeatherAPI*, elle affiche en temps réel les niveaux de polluants (PM2.5, PM10, CO, NO2, O3) pour les principales métropoles de l'Hexagone. Elle s'inscrit dans une volonté d'élargir le champ d'information climatique, en allant au-delà des simples prévisions météo.

L'ensemble de ces pages partage une structure commune (entête, menu, pied de page), intégrée via des fichiers inclus (*header.inc.php*, *footer.inc.php*), assurant cohérence visuelle et facilité de maintenance.

2.3. Approfondissement des fonctionnalités clés

Notre site intègre également plusieurs fonctionnalités avancées visant à améliorer l'accessibilité, l'ergonomie et l'expérience utilisateur. Le mode jour/nuit est sélectionnable manuellement via un bouton, et mémorisé à l'aide de cookies ou via un paramètre GET, ce qui permet à l'utilisateur de conserver son thème préféré ou de tester le contraste sans accepter les cookies. De plus, les cartes interactives jouent un rôle central dans la navigation du site. La carte de France métropolitaine permet à l'utilisateur de sélectionner visuellement une région : chaque forme vectorielle (`<path>`) est cliquable via une balise `<a>` et redirige dynamiquement vers les départements associés. Ces deux approches combinent accessibilité, ergonomie et compatibilité, tout en garantissant une navigation fluide sur l'ensemble des supports. Côtés données, le site exploite plusieurs API : *WeatherAPI* (JSON) pour les prévisions, *NASA APOD* (JSON) pour l'image du jour, *GeoPlugin* (XML) pour la géolocalisation, et *WhatIsMyIP* (XML) pour détecter l'adresse IP. En complément, la page *neige.php* affiche pour chaque massif montagneux (Alpes, Jura, Vosges, etc.) une image sélectionnée aléatoirement parmi un ensemble propre à

ce massif, ce qui apporte une touche visuelle dynamique et renouvelée à chaque visite.

3. Stockage et statistiques

3.1. Affichage des diverses statistiques

Afin de suivre l'utilisation du site PreviZen et d'améliorer l'expérience utilisateur, un système de statistiques a été intégré à la page *statistiques.php*. Dès qu'un utilisateur consulte une ville, cette information est stockée côté serveur et la dernière ville est également conservée côté client via un cookie. Ces données sont ensuite exploitées pour générer deux graphiques dynamiques grâce à la bibliothèque Chart.js, pilotée par un fichier JavaScript *stats.js*. Ce script JavaScript utilise `fetch()` pour interroger en AJAX un point d'accès *stats_data.php*, qui renvoie les données statistiques au format JSON. Ainsi, on crée aisément un diagramme circulaire représentant la répartition des consultations par ville, tandis qu'une courbe d'évolution temporelle des visites quotidiennes est affichée et mise à jour à chaque visite. Les clés des objets sont utilisées comme labels (noms de villes ou dates), et les valeurs comme données numériques. Ce traitement dissocié permet de charger les statistiques sans ralentir pour autant l'affichage initial de la page. Ainsi, cette section combine stockage côté serveur, cookie de personnalisation, rendu graphique interactif, et traitement client, offrant une vue claire et évolutive de l'usage du site.

3.2. Gestion des cookies

L'utilisation de plusieurs cookies permet également l'amélioration de l'expérience utilisateur tout en respectant la confidentialité des données. Premièrement, un cookie nommé `theme` est mis en place pour mémoriser le choix de style (jour ou nuit), garantissant une accessibilité optimale même après rechargement ou navigation entre pages. Ce paramètre peut aussi être contrôlé via l'URL (`?style=...`), ce qui permet de tester le rendu visuel sans stocker nécessairement de donnée. Ensuite, lorsqu'une ville est choisie dans le formulaire hiérarchique, un cookie `ville` est créé pour proposer automatiquement la météo de cette commune lors de prochaines visites. Un second cookie `derniere_consultation` conserve la dernière ville affichée ainsi que la

date de consultation, offrant un rappel contextuel en page d'accueil. Enfin, un cookie `last_city` est défini à partir de l'adresse IP pour enrichir les statistiques d'accès de manière totalement anonyme.

4. Qualité d'usage et évolutivité du site

4.1. Conformité technique

L'accessibilité a été un axe transversal dans l'ensemble de l'implémentation du site PreviZen, quel que soit le type de page (accueil, météo locale, plage, neige, pollution ou statistiques). L'ensemble des pratiques du web moderne sont mises en oeuvre: balisage HTML sémantique, attributs alt systématiquement renseignés pour toutes les images (via une fonction PHP dédiée), et utilisation d'une palette de couleurs à contraste élevé, déclinée dans deux feuilles de style (jour et nuit). Le changement de thème est accessible à tout moment via un bouton dédié, avec conservation des paramètres GET et sans rechargement brutal. Des outils comme le validateur HTML/CSS du W3C ou Wave ont permis de tester régulièrement la conformité de notre site aux standards d'accessibilité.

4.2. Expérience utilisateur

Côté utilisateur, PreviZen a été conçu pour garantir une navigation fluide, homogène et fonctionnelle sur l'ensemble des pages. La structuration hiérarchique des sélections permet un accès rapide et logique aux données météorologiques locales. La page d'accueil regroupe les principales fonctionnalités via une interface claire et centralisée. Chaque aspect de l'ergonomie a été optimisé pour renforcer l'expérience utilisateur tout en assurant une cohérence visuelle et technique.

4.3. Gestion du projet

La gestion du projet PreviZen s'est appuyée sur une planification claire et collaborative, matérialisée par un diagramme de Gantt précis. Chaque membre du binôme a été impliqué dans des tâches spécifiques selon ses compétences³ tout en conservant des phases de travail en commun pour garantir la cohérence globale. La répartition des responsabilités a permis une progression fluide, avec des chevauchements judicieux pour assurer le relais technique et éviter les blocages. Cette organisation a facilité le respect du calendrier initial, tout en laissant de la place à des ajustements au fil du développement.

³ NB: voir Annexe, figure1.

4.4. Perspectives

Sur le plan technique, le développement de PreviZen nous a permis de renforcer des compétences clés en structuration modulaire du code PHP, en intégration d'API REST externes (notamment WeatherAPI), ainsi qu'en gestion avancée des cookies pour améliorer l'expérience utilisateur. L'architecture du projet repose sur un modèle trois tiers, avec une séparation claire entre l'interface utilisateur, la logique serveur et les services distants, garantissant stabilité, lisibilité et évolutivité du code. L'organisation via des fichiers `include/` a facilité la maintenance et la réutilisation des composants.

Nous avons également veillé à rendre l'interface accessible et cohérente sur toutes les pages, en intégrant des mécanismes de personnalisation tels que le choix du thème ou la mémorisation de la dernière ville consultée. Le système de statistiques, basé sur des fichiers CSV, offre une première approche du suivi utilisateur tout en respectant la confidentialité.

À terme, plusieurs pistes d'amélioration sont envisagées, comme la persistance des données en base de données, l'ajout de notifications météo ciblées, l'ouverture aux prévisions internationales ou encore la création d'une interface mobile optimisée. Ces évolutions viendraient enrichir un socle déjà fonctionnel et techniquement solide.

5. Conclusion

Le projet PreviZen a constitué une mise en application concrète et exigeante des compétences acquises en UE Développement Web, à travers une approche complète alliant technique, ergonomie et pertinence fonctionnelle. L'utilisation conjointe de technologies web modernes (HTML5, CSS3, PHP8, JavaScript) et de services distants (API météo, géolocalisation, cookies, statistiques) a permis de concevoir un site structuré, intuitif et centré sur l'utilisateur. Grâce à une architecture modulaire et évolutive, le site répond efficacement à une diversité de besoins concrets, qu'il s'agisse de météo locale, côtière, montagnarde ou de qualité de l'air.

PreviZen se distingue par une structure cohérente, une interface accessible et des fonctionnalités utiles, tout en intégrant des mécanismes simples de personnalisation et de mémorisation. Ce projet, à la fois abouti et fonctionnel, constitue une base solide pour d'éventuelles évolutions futures, telles que l'ajout d'alertes météo, la connexion à une base de données ou l'élargissement aux prévisions internationales. Il reflète une mise en œuvre rigoureuse des acquis du semestre, avec une attention portée à la clarté, à la maintenabilité et à l'expérience utilisateur.

6. Annexe

Figure 1: Diagramme de GANTT modélisant la gestion du projet

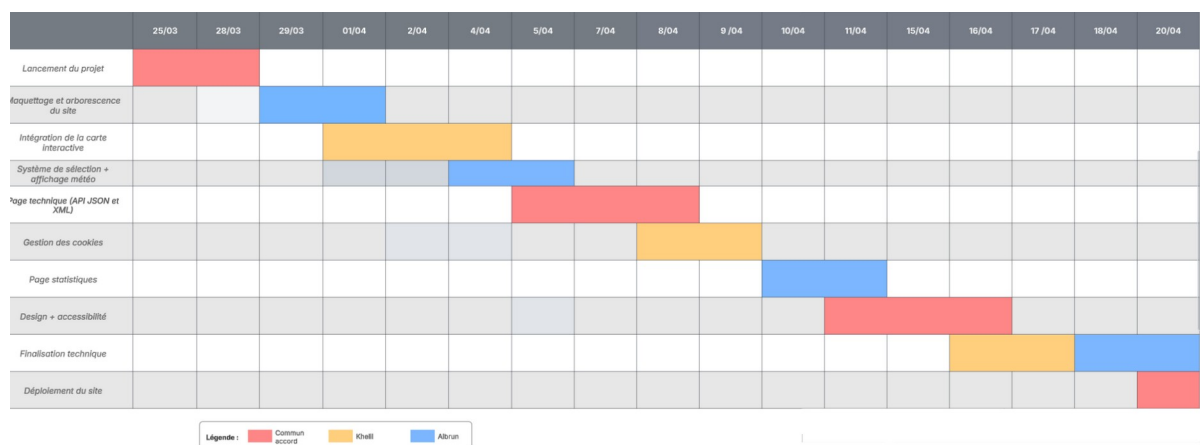
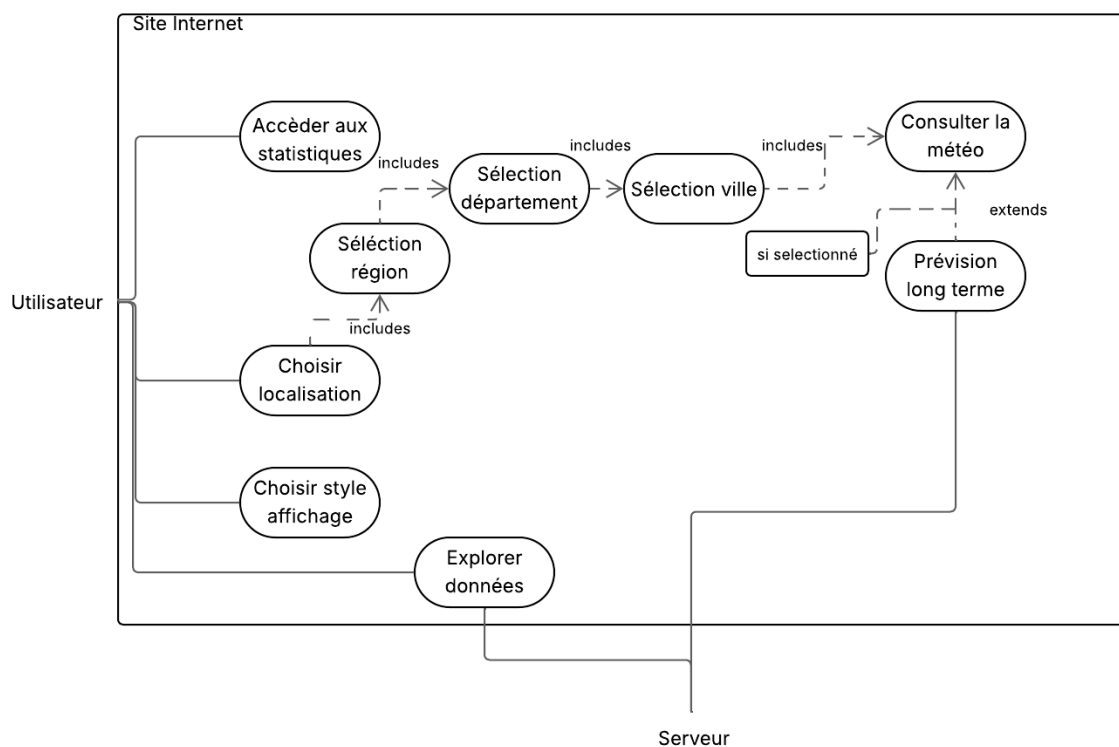


Figure 2: Diagramme de GANTT modélisant la gestion du projet



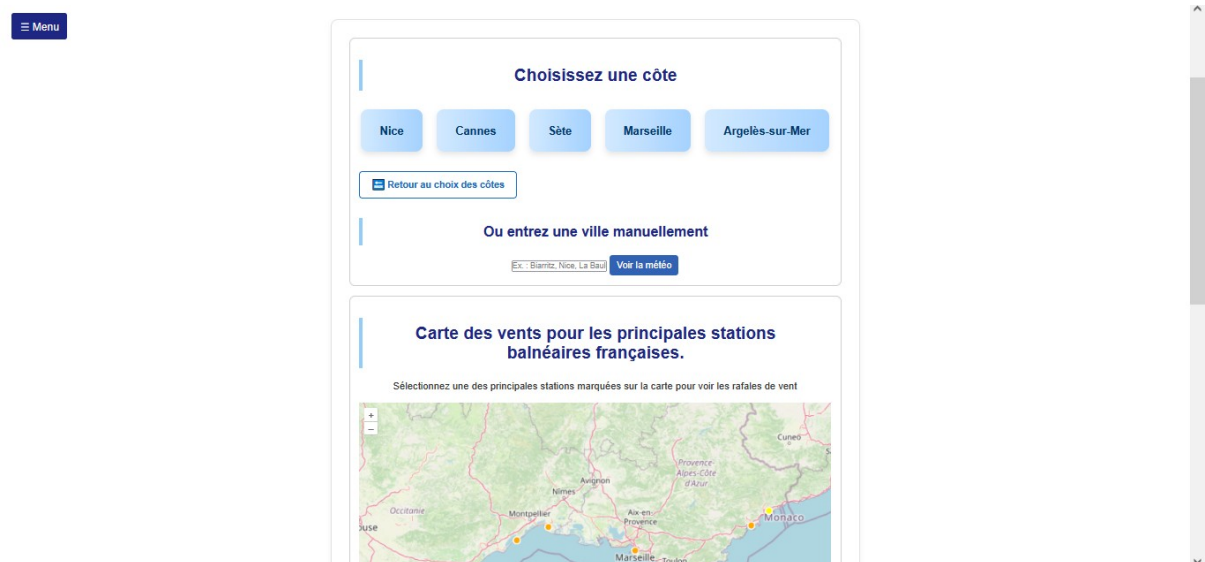


Figure 3 : Mockup de la page index.php, section carte interactive

Figure 4 : Mockup de la page mer.php, section carte dynamique

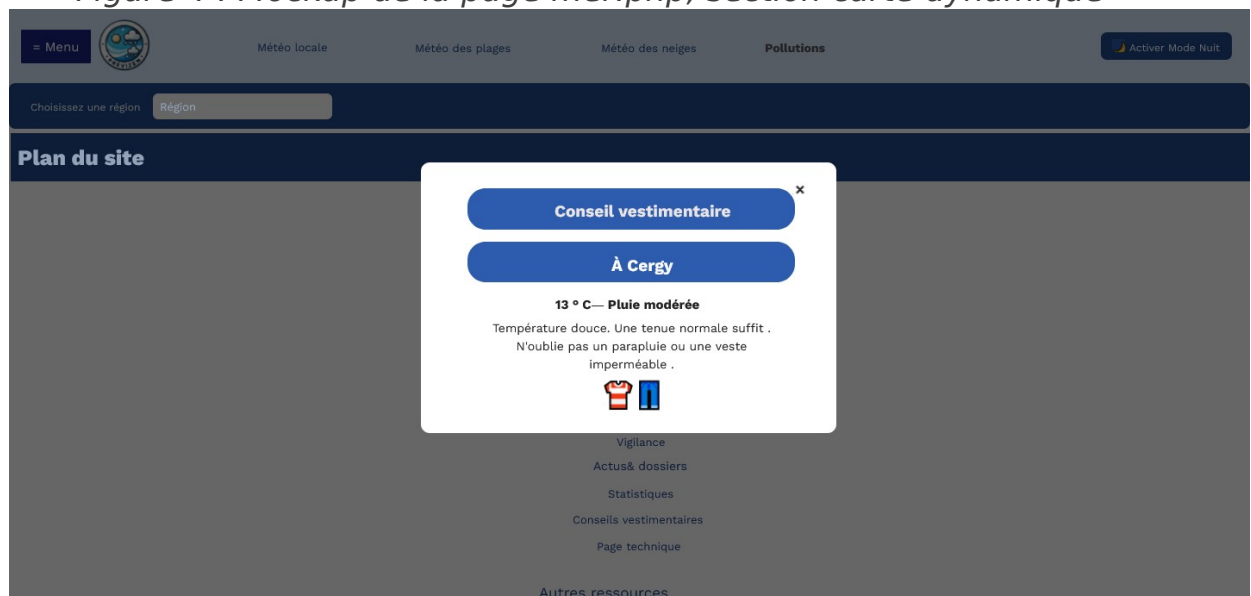


Figure 5 : Mockup de la page neige.php, section carte dynamique

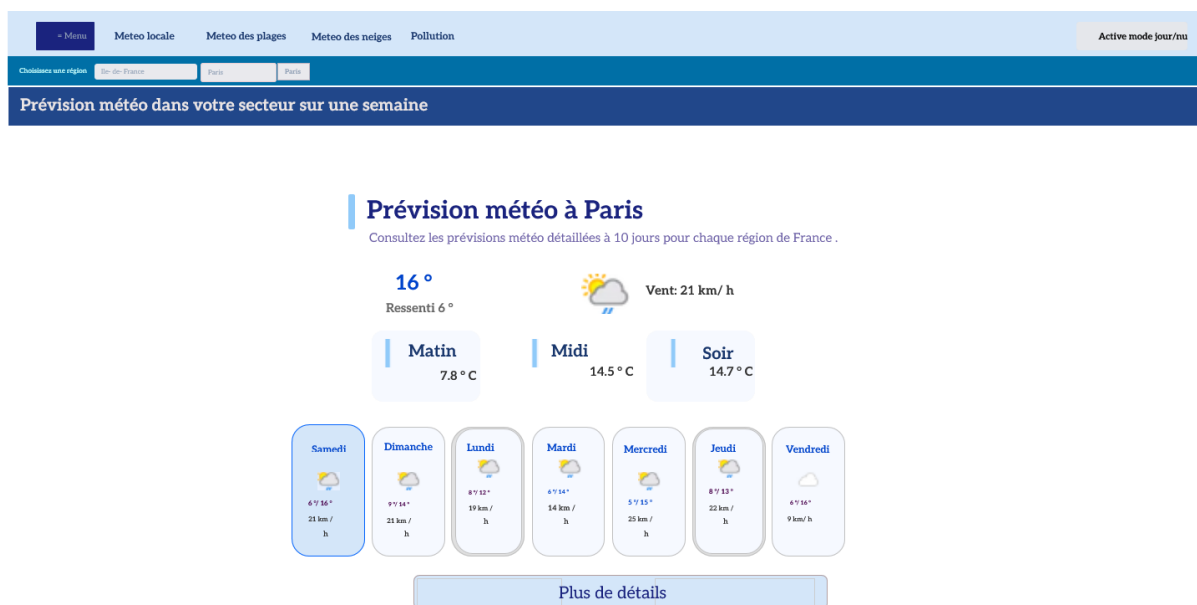


Figure 6 : Mockup de la page actus.php

Figure 7 : Mockup de la page mer.php



Figure 8 : Mockup du pop up de conseils vestimentaires

Figure 9 : Mockup de la page principale pour la prevision sur 7 jours

Figure 10 : Mockup de la section a propos