

DOSSIER PROJET

ANNEXE : ECFs Frontend

CORALIE DUBREUIL

Sommaire

1. Ecf 1 : PokéAPI

1. Liste des compétences du référentiel couvertes par le projet.....p.3
2. Cahier des charges du projet.....p.3
3. Présentation du rendu visuel du projet.....p.5
4. Présentation technique du projet.....p.8

2. Ecf 2 : Route 66

1. Liste des compétences du référentiel couvertes par le projet.....p.12
2. Cahier des charges du projet.....p.12
3. Présentation du rendu visuel du projet.....p.15
4. Présentation technique du projet.....p.18

Ecf 1 : PokéAPI

1. Liste des compétences du référentiel couvertes par le projet

Activité type : Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Compétences professionnelles :

1. Maquetter une application
2. Réaliser une interface utilisateur web statique et adaptable
3. Développer une interface utilisateur web dynamique

2. Cahier des charges du projet

Contexte du projet

Il a fallu participer à la réalisation du front-end d'une application web « responsive » destinée à visualiser les monstres Pokémon et leurs principales caractéristiques en exploitant dynamiquement les données issues de la base de données PokéAPI (<https://pokeapi.co/>).

Spécifications fonctionnelles

Le site devait comporter 2 pages web.

1. Une page statique listant au minimum 6 Pokémon de mon choix issu de la base de données PokéAPI.

Chaque Pokémon était présenté succinctement avec les caractéristiques suivantes :

- Son image

- Son nom
- Un lien, paramétré avec l'ID du Pokémon, redirigeant vers la page dynamique.

2. Une page dynamique présentant les caractéristiques détaillées d'un Pokémon donné (paramétrage de la page grâce à l'ID du Pokémon passé en paramètre de la querystring de l'URL de la page).

Cette page affichait les caractéristiques du pokémon en AJAX ou Fetch depuis l'API HTTP <https://pokeapi.co/api/v2/pokemon/{id}/>.

Les caractéristiques du Pokémon comprenait au minimum :

- Son ID
- Son nom
- Sa hauteur
- Son poids
- Son espèce d'appartenance
- Son image

Charte graphique

La charte graphique devait respecter les contraintes suivantes :

- Une palette couleur harmonieuse limitée à 4 couleurs (hors blanc, noir et gris).
- Un bandeau supérieur incluant le titre « Pokémon front-end »
- Une zone d'affichage principale unique contextuelle
- Un pied de page incluant le logo PokéAPI avec un lien nommé « powered by PokéAPI » vers le site officiel de l'API Pokémon.

3. Présentation du rendu visuel du projet

Le projet consistait donc à faire 2 pages web, l'une statique comportant une équipe de 6 pokémons et l'autre construite dynamiquement affichant le pokémon sélectionné dans la page statique avec plus de détails. Ces 2 pages devaient être responsive c'est-à-dire s'adapter à toutes les tailles d'écran et être construite à partir de la version pour téléphone portable en premier autrement dit en mobile-first.

J'ai pour cela, d'abord réalisé la maquette de la version mobile du projet grâce à un outils de maquettage en ligne gratuit : moqups. Cette maquette sert surtout à montrer le chemin pour aller de la page statique à la page dynamique. Voir l'image ci-dessous.

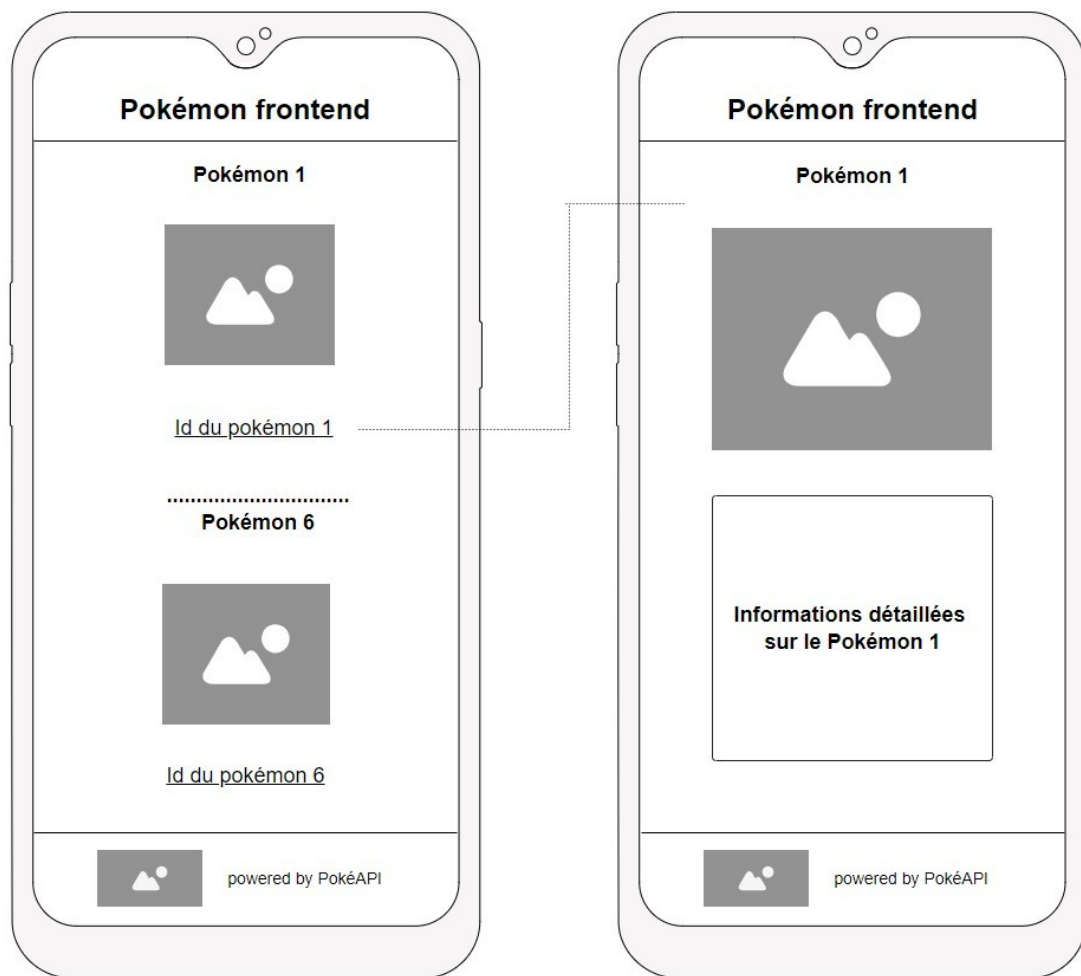
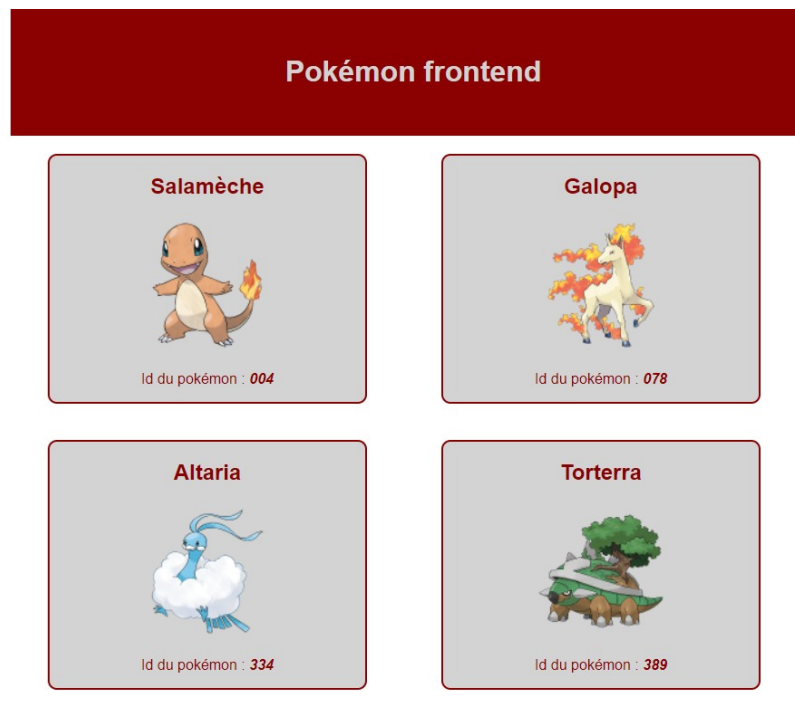


Figure 1: Maquette du projet pokéAPI

J'ai ensuite réalisé la page statique avec l'équipe de 6 pokémons en responsive :

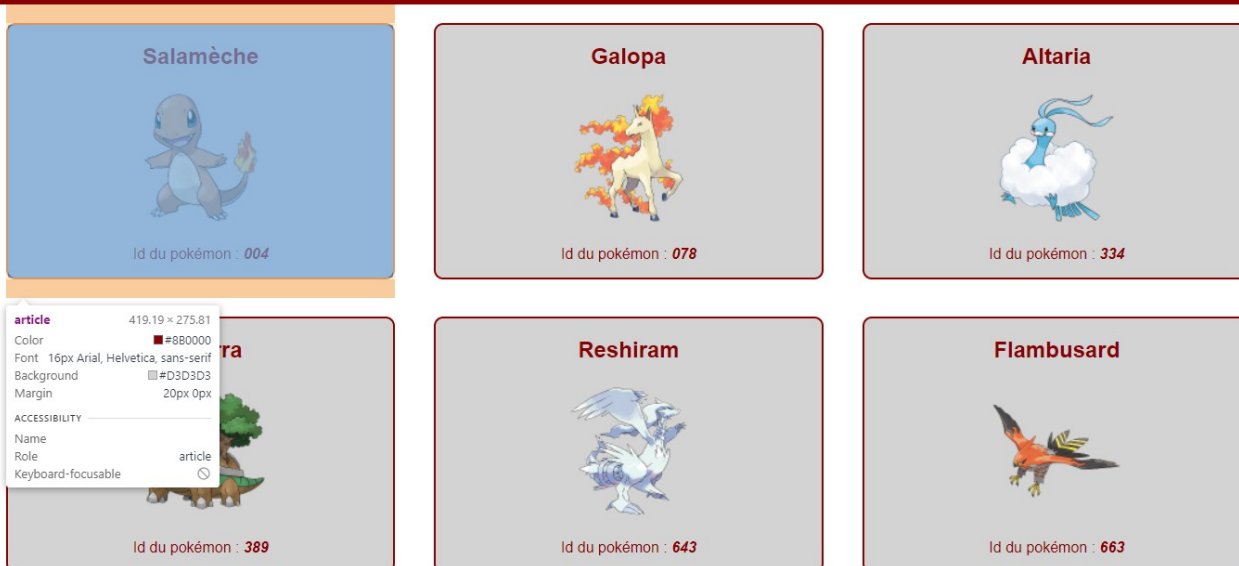


Version mobile



Version tablette

Pokémon frontend



Version écran d'ordinateur

Comme on le voit sur les images, les cadres de pokémons s'alignent différemment en fonction de la taille de l'écran, 1 par ligne pour la version mobile, puis 2 pour les tablettes et enfin 3 pour les écrans d'ordinateurs.

La page dynamique accessible depuis l'id du pokémon correspondant sur la page principale est entièrement construite en JavaScript à partir de la base de données JSON PokéAPI, voici le résultat visuel pour le premier pokémon de la liste en version mobile :

Pokémon frontend

Charmander



Id du pokémon : 4

Hauteur du pokémon : 6

Poids du pokémon : 85

Espèce du pokémon : charmander



PokéAPI

powered by

Exemple de page dynamique

L'url ou adresse de la page dynamique est écrite dans la page statique, l'id variant en fonction du pokémon sélectionné :

```
Desktop/cours/DWWW/ecf1/html/pokemon_info.html?id=4
```

4.Présentation

technique du projet


```

<!DOCTYPE html>
<html Lang="fr">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="../css/style.css" />
    <title>ECF1 : PokemonAPI</title>
  </head>
  <body>
    <header>
      <a href="index.html" class="entete"><h1>Pokémon frontend</h1></a>
    </header>
    <main>
      <section class="flex">
        <article>
          <h2>Salamèche</h2>
          
          <p>Id du pokémon : <a href="pokemon_info.html?id=4"> 004</a></p>
        </article>

```

Figure 2: Code HTML de la page statique

La page statique a été construite en HTML5. Les éléments les plus importants présents sur cette capture d'écran sont :

- le lien présent dans la balise « head » vers le fichier css (qui permet de modifier le style de la page),
- l'attribut « class » avec sa valeur « flex » attribué à la section dans la balise « main » qui va permettre ensuite de réaliser en css le côté responsive de la page,
- les informations du premier pokémon qui sur cette page, sont écrits dans le corps même de la page.

Sur l'image suivant, j'ai choisi de montrer le code en css des « media query » c'est-à-dire la partie qui permet l'adaptation de la page en fonction des formats d'écrans.

```

/*****Média query*****/
@media only screen and (min-width: 481px) and (max-width: 1024px) {
  .flex {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-around;
  }
  .flex article {
    width: 40%;
  }
}

@media only screen and (min-width: 1025px) {
  .flex {
    display: flex;
    flex-wrap: wrap;
    justify-content: space-around;
  }
  .flex article {
    width: 30%;
  }
}

```

Figure 3: Code CSS des media query

Sur l'image suivante, on voit le code HTML de la page dynamique, elle est identique à la page statique à 2 exceptions près :

- cette fois-ci les informations du pokémons ne sont pas codées dans la page même, seule les balises entourant ces informations sont présentes et elles sont vides.
- Il y a une balise supplémentaire juste avant la balise de fermeture du « body » s'appelant « script » qui permet de faire le lien avec le fichier JavaScript qui va permettre de remplir les balises vides quand on affichera la page à partir de celle statique.

```

<!DOCTYPE html>
<html Lang="fr">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="../../css/style.css" />
    <title>ECF1 : PokemonAPI</title>
  </head>
  <body>
    <header>
      <a href="index.html" class="entete"><h1>Pokémon frontend</h1></a>
    </header>
    <main>
      <section>
        <article>
          <h2 id="name"></h2>
          <img src="" alt="image du pokemon" id="img" />
          <p id="id"></p>
          <p id="height"></p>
          <p id="weight"></p>
          <p id="species"></p>
        </article>
      </section>
    </main>
    <footer>
      
      <a href="https://pokeapi.co/" class="entete">powered by PokéAPI</a>
    </footer>
    <script src="../../js/script.js"></script>
  </body>
</html>

```

Figure 4: Code HTML de la page dynamique

```

//Récupération de l'Id du pokémon\\
let params = new URL(document.location).searchParams;
let id = parseInt(params.get("id"));
//Récupération des informations du pokémon dans l'API et affichage des données du pokémon dans la page HTML\\
let url = "https://pokeapi.co/api/v2/pokemon/" + id;
function xhrListener() {
  let pokemon = JSON.parse(xhr.responseText);
  //Nom du pokémon\\
  let name = pokemon.name;
  let pokename = document.getElementById("name");
  pokename.textContent = name;
  //Image du pokémon\\
  let pokeimg = document.getElementById("img");
  pokeimg.src =
    "https://raw.githubusercontent.com/PokeAPI/sprites/master/sprites/pokemon/other/official-artwork/" +
    id +
    ".png";
  //Id du pokémon\\
  let pokeid = document.getElementById("id");
  pokeid.textContent = "Id du pokémon : " + id;
  //Hauteur du pokémon\\
  let height = pokemon.height;
  let pokeheight = document.getElementById("height");
  pokeheight.textContent = "Hauteur du pokémon : " + height;
  //Poids du pokémon\\
  let weight = pokemon.weight;
  let pokeweight = document.getElementById("weight");
  pokeweight.textContent = "Poids du pokémon : " + weight;
  //Espèce du pokémon\\
  let species = pokemon.species.name;
  let pokespecies = document.getElementById("species");
  pokespecies.textContent = "Espèce du pokémon : " + species;
}
//Requête Ajax\\
const xhr = new XMLHttpRequest();
xhr.onload = xhrListener;
xhr.open("get", url, true);
const result = xhr.send();

```

Figure 5: Code JavaScript du projet PokéAPI

Le code JavaScript est construit de la façon suivante :

- D'abord, je récupère l'id du pokémon souhaité depuis l'url de la page et grâce à la fonction « parseInt » on force le langage à reconnaître l'Id comme un nombre entier.
- Ensuite j'ajoute cette variable id (nombre entier) à l'adresse de l'API Pokémons pour obtenir les informations du pokémon correspondant grâce à une requête AJAX. J'ai ainsi pu choisir et ajouter les informations qui m'intéressait aux balises vides correspondantes.

Ecf 2 : Route 66

1. Liste des compétences du référentiel couvertes par le projet

Activité type : Développer la partie front-end d'une application web ou web mobile en intégrant les recommandations de sécurité

Compétences professionnelles :

4. Réaliser une interface utilisateur avec une solution de gestion de contenu ou e-commerce.

2. Cahier des charges du projet

Contexte du projet

Il a fallu réaliser le site vitrine pour le compte de l'association « Route 66 » de collectionneurs de « vieilles mécaniques américaines ».

Cette association souhaitait pouvoir rédiger et publier, à destination du grand public, des articles techniques et des photos sur les vieux véhicules américains.

Charte graphique

La charte graphique devait respecter les contraintes suivantes :

- Le thème devait être sobre, simple, en harmonie (couleurs et polices) avec le style « old school » et uniforme sur l'ensemble des pages publiques.
- Un bandeau supérieur devait être présent sur toutes les pages. Il incluait :
 - le nom de l'association
 - le logo de l'association
 - un lien de retour vers la page précédente si nécessaire

Spécifications fonctionnelles du « front office »

Le site devait comporter les pages web publiques suivantes :

- Une page d'accueil comprenant :
 - 2 photos titrées redirigeant respectivement vers les pages suivantes :
 - «Vieilles mécaniques»
 - «Bielles et pistons»
 - une zone de saisie texte permettant d'effectuer une recherche parmi les articles et photos publiés incluant leurs titres et leurs mots clés.
- Une page titrée «Vieilles mécaniques» listant les photos publiées sur les différents véhicules. Ces photos seront listées par ordre chronologique décroissant de publication.
- Une page titrée «Bielles et pistons» listant les articles techniques sur les différents véhicules. Ces articles seront listés par ordre chronologique décroissant de publication.
- Une page titrée «Votre recherche» listant les articles et photos correspondant à la recherche. Ces résultats seront listés par ordre chronologique décroissant de publication.
- Une page contextuelle présentant le détail de l'article ou de la photo à afficher.

Chaque article technique ou photo publié devra comporter un titre, un descriptif et au moins un mot clé.

Lors de l'affichage détaillé, chaque article technique ou photo devait faire mention du copyright (auteur de l'article ou de la photo et date de rédaction ou de prise de vue).

Bien que francophone, l'association «Route 66» est totalement tournée vers la culture nord américaine. Le site devait être donc rédigé en français mais devait proposer une traduction automatique en anglais grâce à la technologie Google Translate.

Spécifications fonctionnelles du «back office »

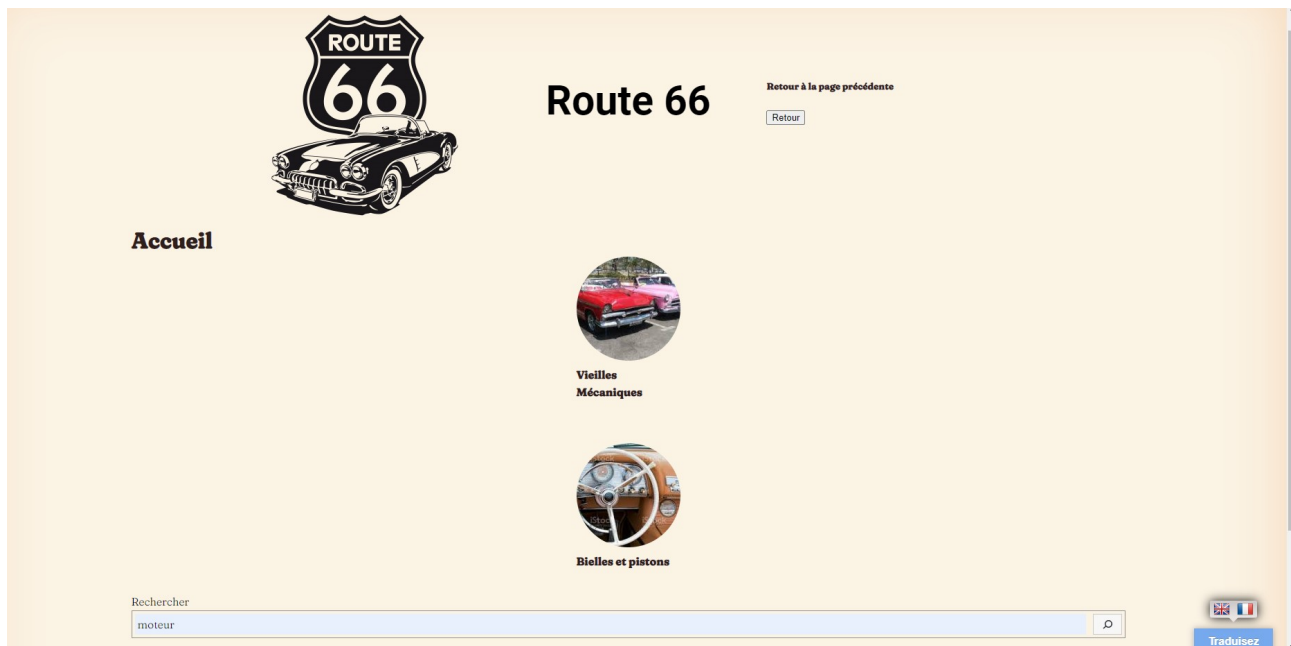
Le «back-office» devait permettre à un «référént communication» de l'association, pour unique fonctionnalité, la rédaction et la publication des articles et des photos. Un «référént

informatique» de l'association devait pouvoir se connecter afin de modifier les différents paramètres du site.

Jeu de test

Afin de tester le site, il était demandé de définir les comptes utilisateurs des référents communication et informatique. Par ailleurs, au moins 3 photos et 3 articles fictifs devait être publiés.

3. Présentation du rendu visuel du projet



Page d'accueil du site route 66



Page Vieilles mécaniques



Page Bielles et pistons

Voiture américaine noire



Photo d'une voiture noire

Photo prise le 24/01/2006 par La Mort

Exemple d'une présentation de voiture photo

Article 3

Titre de l'article technique 3

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque diam nisl, dignissim a ligula nec, lobortis molestie sem. Ut non ex eget elit imperdiet congue. Praesent blandit pulvinar tortor aliquet ultrices. In non auctor est. Phasellus varius nisi sed lorem tincidunt elementum. Aliquam dignissim enim sed dui fringilla, eget facilisis sapien vehicula. Quisque nisi purus, lobortis eget vestibulum vitae, iaculis quis leo. Suspendisse finibus velit et risus tempor tristique vel eu sem. Nulla dapibus arcu in dictum tempor. Duis a viverra nibh. Nulla facilisi. Etiam vel luctus quam. Aenean est eros, cursus eu auctor eu, feugiat mattis libero. Ut viverra rhoncus scelerisque. Quisque a molestie ante. Vestibulum pretium vehicula enim.

Sed vitae vulputate dui. Quisque ut rutrum sapien. Vestibulum in dolor eget purus mollis fermentum eget eu elit. Maecenas a nunc ut lectus consectetur ultricies id sed nunc. In vitae eleifend sapien. Aliquam scelerisque ornare porta. Pellentesque id justo lacus. Aenean tincidunt rutrum ligula in tincidunt. Cras ex est, interdum vel massa id, vulputate aliquet enim. Donec eleifend pretium est vitae pellentesque. In finibus lectus nec consequat scelerisque. Maecenas est ipsum, blandit in efficitur ut, cursus quis lacus. Vivamus tristique fermentum sem. In venenatis tellus maximus enim lacinia, in consectetur massa elementum. Sed auctor nec mauris ut porta. Nullam a rhoncus purus.

Duis faucibus malesuada turpis eget imperdiet. Curabitur malesuada est est, quis ornare nunc dictum a. Curabitur bibendum dolor ac convallis fringilla. Pellentesque vehicula facilisis nisi nec lobortis. Sed bibendum lectus vitae augue convallis, ut mattis orci sodales. Aliquam sollicitudin, purus ut ullamcorper pharetra, lacus lacus maximus est, malesuada ornare magna ex et sem. Suspendisse eu libero vel tellus imperdiet pretium id sed tellus. Sed at augue sed sem gravida pulvinar suscipit in libero. Curabitur tempus enim at felis portitor, a eleifend purus gravida. Sed sed enim commodo, volutpat quam non, finibus diam.

Maecenas libero massa, cursus in rhoncus sit amet, sagittis non est. Phasellus eget metus laoreet sem dapibus aliquam ut ut diam. In tincidunt tincidunt nisl, eleifend posuere leo malesuada sed. Nullam dignissim gravida ipsum id cursus. Etiam et magna id odio tempus aliquet vitae vitae ante. Praesent pharetra a urna eget aliquet. Donec feugiat nunc bibendum lectus lacinia, a convallis nunc dictum. Nulla commodo venenatis magna vitae ornare. Praesent vitae mollis leo, quis mollis risus. Donec viverra scelerisque maximus.

Vivamus sit amet ipsum ligula. Donec imperdiet at erat id imperdiet. Phasellus ut scelerisque elit, dictum sagittis ligula. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean sodales felis nec placerat scelerisque. Quisque eu tellus sapien. Sed sit amet nunc quis est tincidunt molestie nec non nulla. Proin sodales mollis arcu, sit amet ornare sem aliquam in. Pellentesque facilisis sed purus at molestie. Fusce massa lacus, malesuada at nulla ac, fermentum dictum nunc. Cras mollis vehicula justo at molestie.

Article rédigé le 14/04/1959 par Maman

Exemple d'un faux-article

Ces captures d'écran montre que comme demandé dans le cahier des charges, le site vitrine réalisé avec Wordpress utilise un thème « old school ». A partir de la page d'accueil, on peut soit consulter les photos dans la page « Vieilles mécaniques » soit consulté des articles techniques dans la page « Bielles et pistons » soit effectuer une recherche dans la base de mot-clé du site (en particulier dans les étiquettes attachées aux articles et photos). Il y a également présent comme demandé un bouton retour dans l'en-tête du site pour revenir à la page précédente ainsi qu'un bouton traduire (qui permet de passer le site du

français à l'anglais et vice-versa). Ces éléments sont visibles sur la capture d'écran de la page d'accueil.



Article 3

Titre de l'article technique 3 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque diam nisl, dignissim a ligula nec, lobortis molestie sem. Ut non ex eget elit imperdiet congue. Praesent blandit pulvinar tortor aliquet ultrices. In non auctor est. Phasellus varius nisi sed lorem tincidunt elementum. Aliquam dignissim enim sed dui fringilla, eget facilisis [...]

Published on 30 Jan 2022 — Last update 30 Jan 2022

Article 2

Titre de l'article technique 2 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quisque diam nisl, dignissim a ligula nec, lobortis molestie sem. Ut non ex eget elit imperdiet congue. Praesent blandit pulvinar tortor aliquet ultrices. In non auctor est. Phasellus varius nisi sed lorem tincidunt elementum. Aliquam dignissim enim sed dui fringilla, eget facilisis [...]

Published on 30 Jan 2022 — Last update 30 Jan 2022



Photo 3

Voiture américaine noire Photo prise le 24/01/2006 par La Mort

Published on 30 Jan 2022 — Last update 30 Jan 2022



Exemple des résultats d'une recherche

4. Présentation technique du projet

| <input type="checkbox"/> Extension | Description |
|--|--|
| <input type="checkbox"/> Back Button Widget Settings Go Pro Désactiver | Back button widget for WordPress. Version 1.5.3 Par Algoritmika Ltd Afficher les détails |
| <input type="checkbox"/> Elementor Réglages Désactiver Passer à la version Pro | Le constructeur de sites web Elementor a tout pour plaire : constructeur de page par glisser-déposer, conception parfaite au pixel près, modification adaptée au mobile, et bien plus encore. Commencez dès maintenant ! Version 3.7.2 Par Elementor.com Afficher les détails Docs & FAQs Tutoriels vidéo |
| <input type="checkbox"/> Elementor Header & Footer Builder Réglages Désactiver | Cette puissante extension permet de créer un en-tête et un pied de page personnalisés avec Elementor et de les afficher à certains endroits. Vous pouvez également créer des blocs Elementor personnalisés et les placer n'importe où sur le site web avec un code court. Version 1.6.12 Par Brainstorm Force, Nikhil Chavan Afficher les détails |
| <input type="checkbox"/> Google Language Translator Désactiver Settings | The MOST SIMPLE Google Translator plugin. This plugin adds Google Translator to your website by using a single shortcode, [google-translator]. Settings include: layout style, hide/show specific languages, hide/show Google toolbar, and hide/show Google branding. Add the shortcode to pages, posts, and widgets. Version 6.0.15 Par Translate AI Multilingual Solutions Afficher les détails |
| <input type="checkbox"/> WP Extended Search Désactiver Réglages | Extend search functionality to search in selected post meta, taxonomies, post types, and all authors. Version 2.1.1 Par Sum17 Afficher les détails Docs More Plugins |
| <input type="checkbox"/> Extension | Description |

Liste des extensions utilisées

Pour réaliser ce site j'ai utilisé les extensions suivantes :

- Back button widget : qui permet de créer le bouton retour présent dans l'en-tête de toutes les pages.
- Elementor et son extension : Elementor Header & Footer Builder : qui m'ont permis de placer le bouton retour dans l'en-tête.
- Google Language Translator : qui m'ont permis de créer le bouton de traduction français/anglais.
- WP Extended Search : qui m'a permis d'ajouter les étiquettes dans les paramètres de la barre de recherche.

| <input type="checkbox"/> Identifiant | Nom | E-mail | Rôle |
|--|-----|--|----------------------------------|
| <input type="checkbox"/>  RefCom | — | cyasama+refcom@gmail.com | Auteur / autrice |
| <input type="checkbox"/>  RefInfo | — | cyasama+refinfo@gmail.com | Administrateur / administratrice |
| <input type="checkbox"/>  Route66admin | — | cyasama@gmail.com | Administrateur / administratrice |
| <input type="checkbox"/> Identifiant | Nom | E-mail | Rôle |

Liste des comptes créés

J'ai également créer deux comptes supplémentaires en plus du compte principal administrateur comme demandé par le cahier des charges. Le premier avec les mêmes droits d'administration et le deuxième uniquement pour créer des articles.

