



Trouve ton artisan !

Avec la région
Auvergne-Rhône-Alpes



Sommaire

1 Présentation

Introduction P.3.

Outils de création P.4.

2 Etapes de créations

Prototypage P.5-7.

Développement du site P.8-9-10.

Design du site P.11-12

Versionning du site P.13.

3 Contrôle qualité

Validation HTML P.14.

Validation CSS P.15.

4 Hébergement & Liens Utiles P.16.

5 Conclusion P.17.

Présentation du projet

Introduction

Dans le cadre de ma formation, je présente aujourd'hui la création de la partie Front-end du site "trouve ton artisan", plateforme dédiée aux artisans de la région.

L'objectif de cette plateforme sera de permettre aux particuliers de trouver un artisan et de lui demander facilement des renseignements, prestations ou encore tarifs via un formulaire de contact.

Le site se devra d'être responsif, accessible au plus grand nombre et conforme à la norme WCAG21 (<https://www.w3.org/TR/WCAG21/>).

La sécurité sera un point important à mettre en place.

Le site devra aussi s'intégrer à l'environnement numérique de la région Auvergne Rhône-Alpes trouvable ici :

<https://www.auvergnerhonealpes.fr/contenus/ladministration-regionale> .

Outils de création

- VSCode

Editeur de texte facile d'accès avec une prise en main rapide et intuitive

- Framework ReactJS

React est une bibliothèque Javascript libre, maintenue par Meta ainsi que par une communauté de développeurs individuels et d'entreprises depuis 2013.

- Figma

Figma est un éditeur de graphiques vectoriels et un outil de prototypage. Il sera utilisé ici comme outil de prototypage, c'est à dire qu'il servira à la création des prototypes à présenter en amont de la création du site.

- Git & GitHub

Git et GitHub seront utilisés pour la gestion du versionning et du déploiement des branches secondaires vers la branche principale du projet.

- Bootstrap

Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web. C'est un ensemble qui contient des codes HTML et CSS.

Etapes de création

Prototypage

Le prototypage de l'application "Trouve ton artisan" à ici été réalisé grâce à l'application Figma.

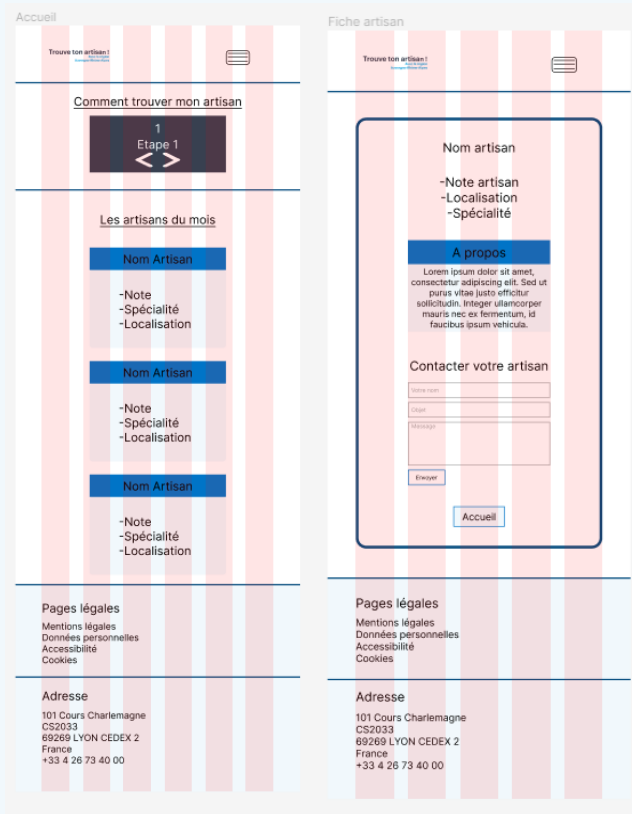


Le but du prototypage est de créer une version préliminaire, faisant office de présentation du projet en amont du développement, à présenter au client afin de modifier avec lui si nécessaire la mise en forme, l'expérience utilisateurs et les aspects dits "graphiques" du site en devenir.

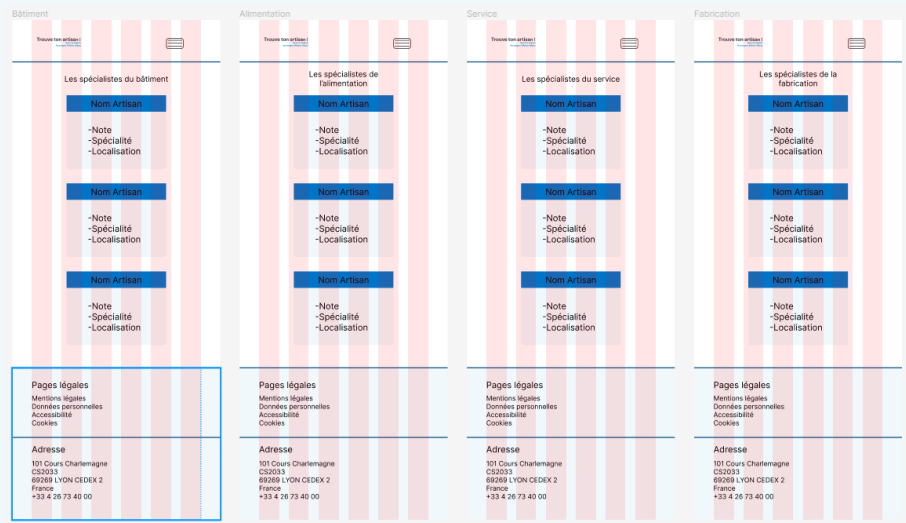
Les premiers prototypes créés, dans un but responsif, ont été les prototypes mobiles avec accueil, fiche artisan, et 4 pages "Bâtiments", "Service", "Alimentation", "Fabrication".



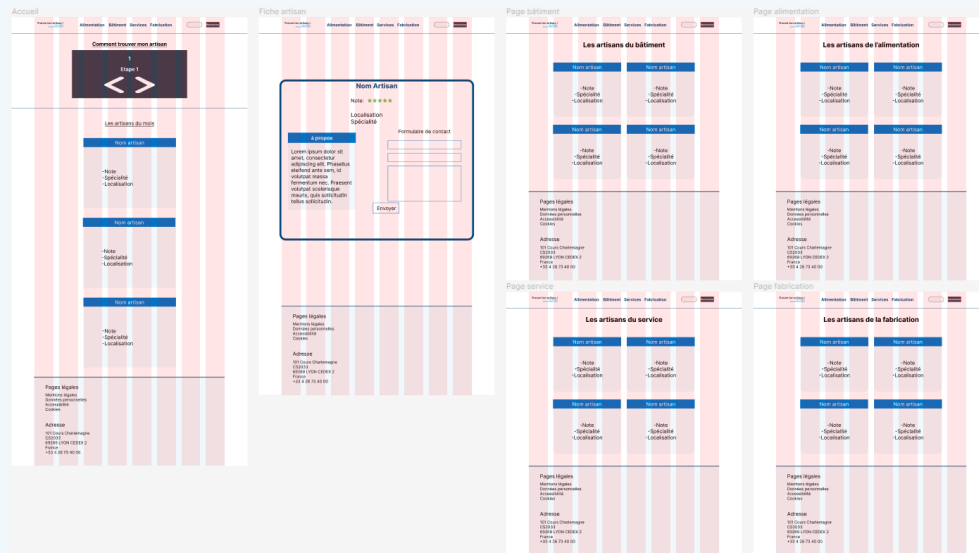
Pages d'accueil et Artisan



Pages référencent les artisans par catégories



La version “tablette” se situant en taille entre “mobile” et “desktop” à été créé sur les mêmes bases que la version mobile mais adapté au format tablette pour une expérience utilisateur fluide et dynamique.



Dans la construction globale, on trouvera, de la même manière que dans les versions mobiles, un page Accueil, une page Artisan, et les quatre pages “Bâtiment”, “Service”, “Alimentation”, et “Fabrication”.

La dernière étape du prototypage a été la création de la version “desktop”, donc version écran d’ordinateur.

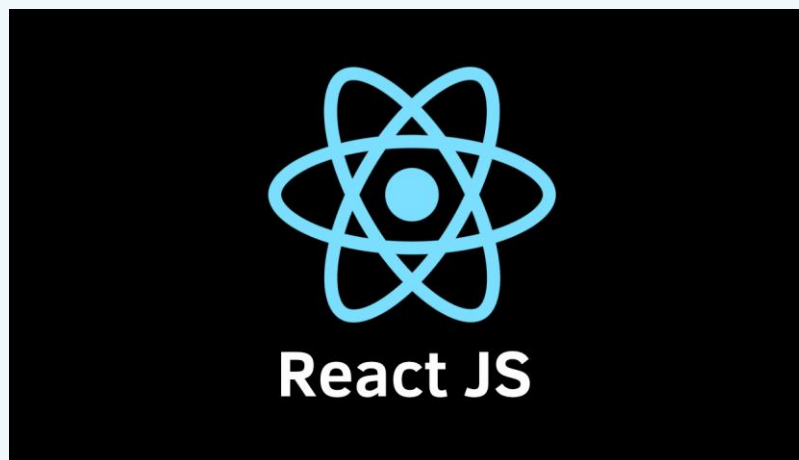


Cette dernière est entièrement ré adaptés à la navigation sur ordinateur, mais conserve tout de même toutes les fonctionnalités présentes sur les versions mobiles et tablettes.

Développement du site

Développement du code de la partie Front-End:

Le développement de l'application "Trouve ton artisan" a entièrement été créé avec le Framework ReactJS.



ReactJs est une librairie orientée composants. L'API exposée va essentiellement servir à créer des composants graphiques. Le principe de React est de créer un arbre de composants et de l'associer à un élément du DOM.

React utilise le langage JSX, très proche de Javascript, dans lequel on peut inclure un code proche du html pour définir les rendus des composants React.

Pour fonctionner, le code JSX est compilé en son équivalent JS.

Dépendances utilisées :

```
"react": "^18.3.1",  
"react-dom": "^18.3.1",
```

Ces dépendances sont des packages NPM.

React est une bibliothèque JS permettant de créer des composants réutilisables qui peuvent être combinés pour former des interfaces utilisateurs complètes.

React-dom est un package qui fournit des méthodes spécifiques pour manipuler le DOM dans le contexte de React. Il est utilisé pour rendre les composants React dans le navigateur web.

```
"react-router-dom": "^6.23.0",
```

Package NPM fournit des outils de routage pour les applications web React.

Il permet de gérer la navigation dans une application en associant des composants React à des URL spécifiques. Cela permet de créer des SPA (Single Page Application) où les composants sont chargés dynamiquement en fonction de l'URL.

```
"react-scripts": "5.0.1",
```

Ce package est un outil de ligne de commande fourni par CRA (Create React App).

Il permet la gestion de la compilation et du développement en temps réel, le démarrage d'un serveur de développement intégré, la gestion des dépendances et des scripts, etc...



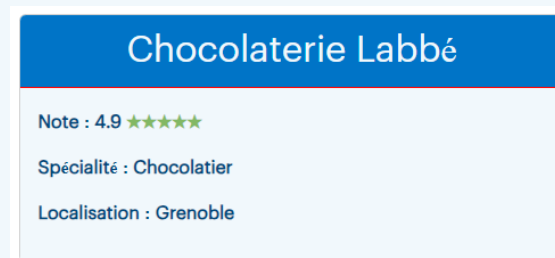
```
"@emailjs/browser": "^4.3.3",
```

Ce package est une bibliothèque Javascript qui permet l'envoi d'e-mails directement depuis le navigateur web côté client, sans devoir passer par un serveur d'envoi d'e-mails côté Back-End

Design du site :

Pour ce qui est de l’aspect “esthétique” du site, un mélange a été fait entre la librairie Bootstrap et l’utilisation de feuilles de styles (stylesheet) écrites en langage CSS3.

Exemple d’une utilisation concrète de bootstrap sur le site “Trouve ton artisan”:



Bootstrap permet de créer rapidement assez personnalisables grâce à des blocs de codes prêts à l’emploi comme ci-dessous:

```
<div className="card topCard">
  <div className="card-header">
    <h2>{artisan.name}</h2>
  </div>
  <div className="card-body">
    <p>Note : {artisan.note} {Array.from({ length: Math.round(parseFloat(artisan.note)) },
    ([, index] => <span key={index}>&#9733</span>)}</p>
    <p>Spécialité : {artisan.specialty}</p>
    <p>Localisation : {artisan.location}</p>
  </div>
</div>
```

Ici la mise en forme va se faire grâce aux className présents à l’intérieurs des balises.

Par exemple, `<div className="card-header">`, cette div contiendra le “card-header”, donc l’en-tête de la carte.

Pour le reste du site, le CSS a été codé de sorte qu’il soit “mobile-first”, donc d’abord adapté à une version mobile du site.



Pour ensuite adapter le responsif mobile aux autres formats, le paramètre suivant a été intégré en y ajoutant des valeurs correspondant aux formats

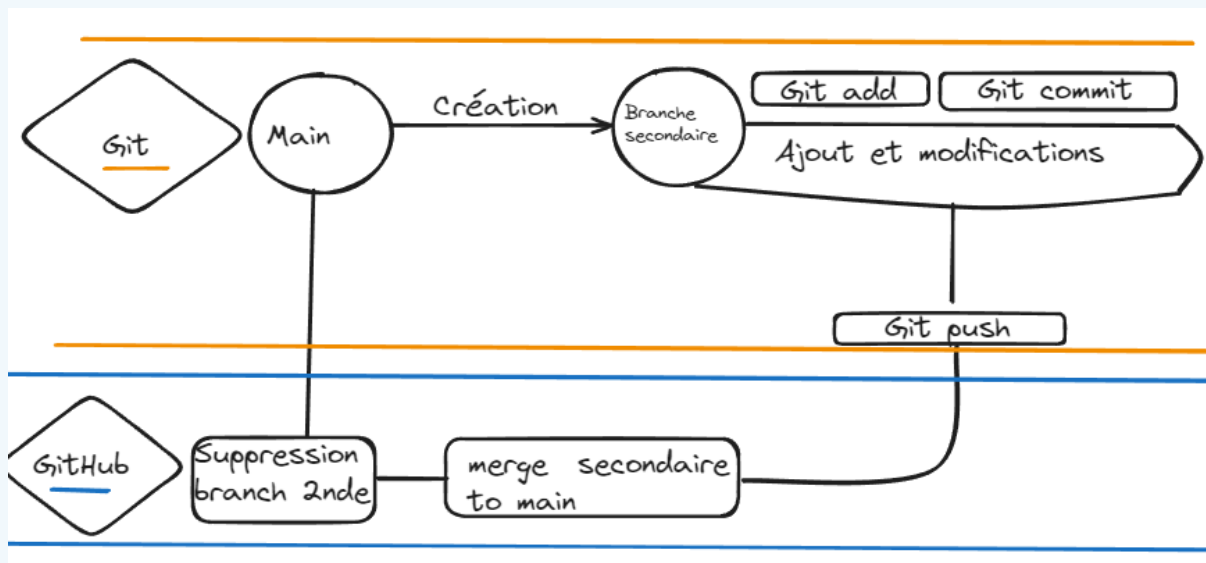
```
@media (min-width: 768px){
```

tablettes et desktop:

```
}
```

Versionning du site

Pour ce qui est du versionning du site, Git et GitHub ont été utilisés en respectant un workflow logique et sécurisé.



En complément de ce schéma, il est intéressant d'ajouter qu'à chaque création d'une branche secondaire, une "issue" a été créer sur GitHub.

Celle-ci est reliée à la branche secondaire attenante et sera donc supprimée au même moment que sa branche lors du merdge vers la branche principale

Contrôles qualités

Validation W3C

Dans l'optique de garantir une bonne écriture générale du code, celui-ci a été examiné à l'aide du validateur proposé par le W3C.

The screenshot shows the Nu Html Checker interface. The 'Checker input' section contains a text area with the following HTML code:

```
<html lang="fr"><head>
  <meta charset="utf-8">
  <link rel="icon" href="/img/favicon-32.png">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="theme-color" content="#000000">
  <meta name="description" content="Trouve ton artisan, le site qui référence les artisans de la région Auvergne Rhône Alpe">
  <link rel="shortcut icon" href="/img/favicon.png">
  <meta name="keywords" content="Artisans, Fabricants, alimentaire, liste, recherche, répertoire, région, Auvergne, Rhône Alpe">
  <link rel="apple-touch-icon" href="/img/favicon-32.png">
<!--
manifest.json provides metadata used when your web app is installed on a
user's mobile device or desktop. See https://developers.google.com/web/fundamentals/web-app-manifest/
-->
<link rel="manifest" href="/manifest.json">
<!--Bootstrap-->
```

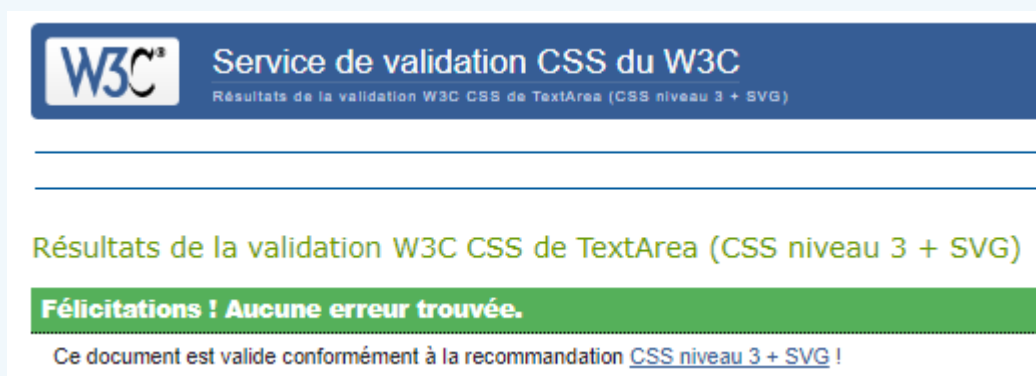
Below the code, there is a 'Check' button. The 'Message Filtering' section shows a single error message:

1. **Error** Start tag seen without seeing a doctype first. Expected `<!DOCTYPE html>`.
From line 1, column 1, to line 1, column 16
`<html lang="fr"><head>`

Celui-ci ne révèle aucune erreur si ce n'est une seule inhérente à l'utilisatin du framework ReactJs

Validation CSS

Toujours dans un but de propreté dans la rédaction du code de l'app, l'ensemble du langage CSS a été examiné au près du validateur W3C dédié :



Aucune erreur n'a été remonté.



Hébergement & Liens utiles

Dans un souci d'économie et de simplicité, l'application a été hébergée directement sur la plateforme www.CodeSandbox.io à l'adresse URL suivante : <https://fcz78j-3000.csb.app/>

Le lien du Repository GitHub est trouvable ici :
<https://github.com/Frostbowl/trouve-ton-artisan.git>

Le lien Figma est ici :
<https://www.figma.com/design/azS7nOuKppOel0wv6Bv9ev/Trouve-ton-artisan?node-id=1-3&t=JocLugMwGJqmJb2K-0>



Conclusion

En conclusion, le projet “Trouve ton artisan” a été un exercice enrichissant qui m’a permis de mettre en pratique les compétences acquises au cours du second cycle de ma formation.

L’objectif principal était de créer une plateforme web intuitive et accessible, facilitant la mise en relation entre les artisans de la région Auvergne Rhône-Alpes et les particuliers à la recherche de leurs services.

Grâce aux outils comme ReactJS, Bootstrap et Figma, le projet a pu aboutir à une solution responsive et adaptée aux demandes d’un client pour l’instant fictif.

Je tiens à remercier Mr Orluk Alain, mon mentor pour sa bienveillance et sa pédagogie tout au long de la réalisation de ce second bilan.