



MINISTÈRE CHARGÉ
DE L'EMPLOI

DOSSIER PROFESSIONNEL (DP)

| | | |
|-------------------------|---|------------------------------------|
| <i>Nom de naissance</i> | - | Lambert-Conchon |
| <i>Nom d'usage</i> | - | Lambert |
| <i>Prénom</i> | - | Sophie |
| <i>Adresse</i> | - | 6 rue Emile Lhote 35400 Saint-Malo |

Titre professionnel visé

Certificat de compétences professionnelles développer la partie front end d'une application web ou web mobile en intégrant les recommandations de sécurité

MODALITÉ D'ACCÈS :

- Parcours de formation

DOSSIER PROFESSIONNEL (DP)

Présentation du dossier

Le dossier professionnel (DP) constitue un élément du système de validation du titre professionnel.

Ce titre est délivré par le Ministère chargé de l'emploi.

Le DP appartient au candidat. Il le conserve, l'actualise durant son parcours et le présente obligatoirement à chaque session d'examen.

Pour rédiger le DP, le candidat peut être aidé par un formateur ou par un accompagnateur VAE.

Il est consulté par le jury au moment de la session d'examen.

Pour prendre sa décision, le jury dispose :

1. des résultats de la mise en situation professionnelle complétés, éventuellement, du questionnaire professionnel ou de l'entretien professionnel ou de l'entretien technique ou du questionnement à partir de productions.
2. du **Dossier Professionnel (DP)** dans lequel le candidat a consigné les preuves de sa pratique professionnelle.
3. des résultats des évaluations passées en cours de formation lorsque le candidat évalué est issu d'un parcours de formation
4. de l'entretien final (dans le cadre de la session titre).

[Arrêté du 22 décembre 2015, relatif aux conditions de délivrance des titres professionnels du ministère chargé de l'Emploi]

Ce dossier comporte :

- pour chaque activité-type du titre visé, un à trois exemples de pratique professionnelle ;
- un tableau à renseigner si le candidat souhaite porter à la connaissance du jury la détention d'un titre, d'un diplôme, d'un certificat de qualification professionnelle (CQP) ou des attestations de formation ;
- une déclaration sur l'honneur à compléter et à signer ;
- des documents illustrant la pratique professionnelle du candidat (facultatif)
- des annexes, si nécessaire.

DOSSIER PROFESSIONNEL (DP)

Pour compléter ce dossier, le candidat dispose d'un site web en accès libre sur le site.

 <http://travail-emploi.gouv.fr/titres-professionnels>

Sommaire

Exemples de pratique professionnelle

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

5

- Crédation d'une maquette Figma et intégration du site statique (mon portfolio) p.
- Crédation et réalisation d'un site Wordpress avec Divi et Woocommerce(Les savons de Malo, boutique en ligne de savons artisanaux) p.
- Réalisation d'un site de type Allociné en React (Film Flow) p

Titres, diplômes, CQP, attestations de formation (facultatif)

Déclaration sur l'honneur

Documents illustrant la pratique professionnelle (facultatif)

Annexes (Si le RC le prévoit)

EXEMPLES DE PRATIQUE PROFESSIONNELLE

DOSSIER PROFESSIONNEL (DP)

Activité-type 1

Création d'une maquette Figma et intégration du site statique

Exemple n°1 -

Mon portfolio (<https://sophie.fronthub.fr>)

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

1. LE CONTEXTE

Pour cet exercice vous allez créer votre **portfolio**.

- Ce travail sera d'abord effectué sous forme de **maquette avec Figma**.
- Une fois la démarche du parcours utilisateur et la maquette validée, vous basculez sur son **intégration**
- Ce site sera situé à la racine de votre nom de domaine.

2. CRÉATION ET CONCEPTION DE LA MAQUETTE

- Je détermine les informations importantes qui doivent apparaître sur le site : (Présentation personnelle, contacts et liens, photo, compétences techniques, études et certifications, expériences professionnelles / CV)

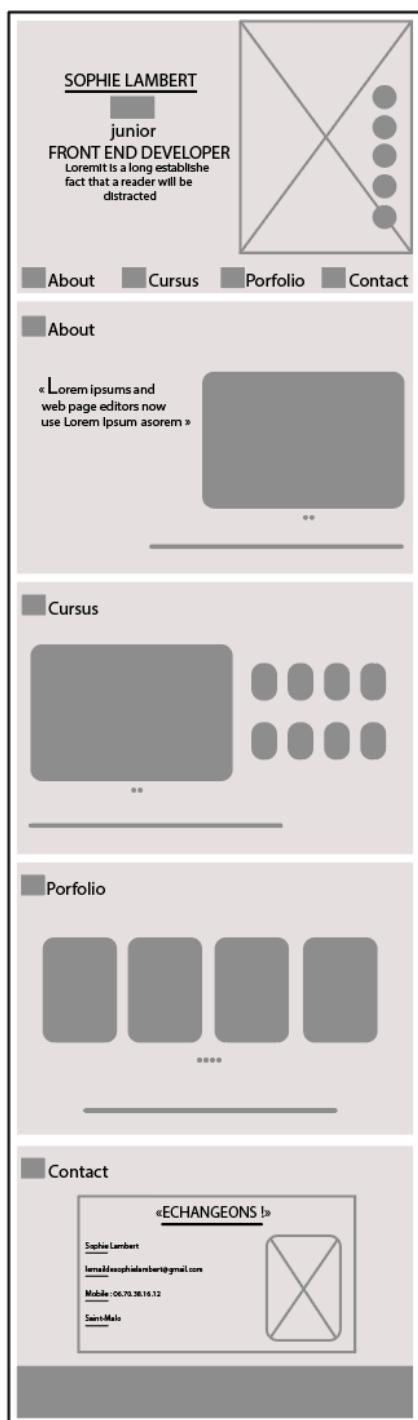
Dans un second temps il sera nécessaire d'ajouter Contributions open sources, références et témoignages et éventuellement Blog ou articles de blog je laisse donc cette possibilité.

- J'écris mes contenus (j'essaye d'être pertinente, d'utiliser des mots clefs, d' utiliser des titres h1,h2,h3 et des textes courts pour faciliter la lecture et l'indexation par les moteurs de recherche.
- Je réfléchie à l'ui et l'ux : je définis quel parcours utilisateur je souhaite, quel design, quelles informations je veux mettre en avant, quelles fonctions je peux/veux mettre en place sur le site .
et quelles animations.
 - Je décide de commencer par la création d'un Wireframe (maquette fonctionnelle) afin de m'aider à concevoir la maquette.

DOSSIER PROFESSIONNEL (DP)

WIREFRAME PORTFOLIO

DESKTOP



MOBILE



DOSSIER PROFESSIONNEL (DP)

- Je construis l'univers graphique et je réalise et/ou je réunis toutes les pièces nécessaires à la réalisation (photo, création graphique, icônes, cv etc...), ensuite je réalise une charte graphique et le traitement des images et des icônes.

Pour cela j'utilise Photoshop 1 illustrator mais aussi les site suivants

- SVG Backgrounds (<https://www.svgbackgrounds.com>)
- Unsplash (<https://unsplash.com>)
- Pexels (<https://www.pexels.com>)
- Pixabay (<https://pixabay.com>)
- Freepik (<https://www.freepik.com>)
- Flaticon (<https://www.flaticon.com>)
- Icons8 (<https://icons8.com>)
- The nounproject (<https://thenounproject.com>)

DOSSIER PROFESSIONNEL (DP)

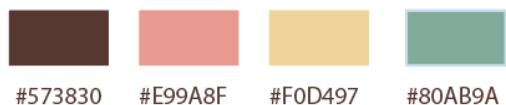
Charte Graphique

PORTFOLIO

Ambiance graphique / Picto



Gamme de couleurs



Typographies / Fonts /Graisses

Nunito sans serif

abcdefghijklmnopqrstuvwxyz light

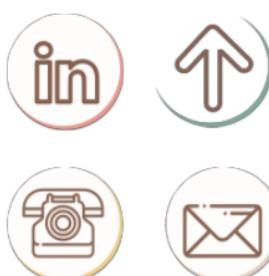
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

abcdefghijklmnopqrstuvwxyz regular
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

abcdefghijklmnopqrstuvwxyz bold
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

Londrina Shadow

abcdefghijklmnopqrstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789



DOSSIER PROFESSIONNEL (DP)

- Je réalise ensuite la maquette avec des versions desktop et responsive.
 - Je crée une frame aux dimensions, mac book pro 16 pouces qui sera la base de mon travail 1728px J'aurais également la version iphone 13 pro 360px version responsive



DOSSIER PROFESSIONNEL (DP)

The screenshot displays a professional portfolio website for Sophie Lambert, a Front End developer. The layout is clean and modern, featuring a header with her name and title, a bio quote, and a circular profile picture. Below the header are navigation links for About, Cursus, Portfolio, and Contact. The 'About' section includes a quote from Charles de Montesquieu and a decorative sunburst graphic. The 'Cursus' section lists her skills in Adobe Photoshop, Adobe Illustrator, Keynote, and Figma, along with experience in WordPress, CSS, HTML, and Visual studio code. It also details her education, including a 2012-2013 CCP1 developer web developer module (Front end) at Institut Haute Théâtre Sébastien Deneuville and a 2000-2001 BTP menuisier à part-time, CAP couture fauteuil, Médiathèque de Bourgogne et généraux Moulins et François Grasauvel ESAMCO Paris. The 'Portfolio' section shows two examples: 'LJ CRÈME CRÉPÉE' and 'ESPRIT NORDIC CHARPENTE & MENUISERIE'. The 'Contact' section features a large input field for messages. A sidebar on the right provides a mobile preview of the site, showing its responsive design and a dimension of 323 x 845 pixels.

mac-portfolio-sophie

Sophie Lambert
FRONT END developer

« La vie est comme un arc-en-ciel : il faut de la pluie et du soleil pour en voir les couleurs. »

About Cursus Portfolio Contact

About

« Qui aime à s'instruire, n'est jamais oisif »
Charles de Montesquieu.

Mes compétences

- Adobe Photoshop
- Adobe Illustrator
- Keynote
- Figma

WordPress CSS HTML Visual studio code Java script

Mes formations

2012-2013 CCP1 : développeur web/ web mobile (Front end). Institut Haute Théâtre Sébastien Deneuville. (suivie le COP 2 développeur back end)

2000-2001 BTP menuisier à part-time, CAP couture fauteuil Médiathèque de Bourgogne et généraux Moulins et François Grasauvel ESAMCO Paris.

2009-2010 BEP pré à partie, CAP couture fauteuil Médiathèque de Bourgogne et généraux Moulins et François Grasauvel GRETNA Lorient.

1995-1999 Baccalauréat Sciences option SVT. Moulins lycée Le Maine.

Portfolio

LJ CRÈME CRÉPÉE

ESPRIT NORDIC CHARPENTE & MENUISERIE

Contact

NE PAS CLIQUER

iphon...

323 x 845

Mes compétences

- Adobe Photoshop
- Adobe Illustrator
- Keynote
- Figma

Mes formations

2012-2013 CCP1 : développeur web/ web mobile (Front end). Institut Haute Théâtre Sébastien Deneuville. (suivie le COP 2 développeur back end)

2000-2001 BTP menuisier à part-time, CAP couture fauteuil Médiathèque de Bourgogne et généraux Moulins et François Grasauvel ESAMCO Paris.

2009-2010 BEP pré à partie, CAP couture fauteuil Médiathèque de Bourgogne et généraux Moulins et François Grasauvel GRETNA Lorient.

1995-1999 Baccalauréat Sciences option SVT. Moulins lycée Le Maine.

MES FORMATIONS

2012-2013 CCP1 : développeur web/ web mobile (Front end). Institut Haute Théâtre Sébastien Deneuville. (suivie le COP 2 développeur back end)

2000-2001 BTP menuisier à part-time, CAP couture fauteuil Médiathèque de Bourgogne et généraux Moulins et François Grasauvel ESAMCO Paris.

2009-2010 BEP pré à partie, CAP couture fauteuil Médiathèque de Bourgogne et généraux Moulins et François Grasauvel GRETNA Lorient.

1995-1999 Baccalauréat Sciences option SVT. Moulins lycée Le Maine.

ESPRIT NORDIC CHARPENTE & MENUISERIE

FRANCIS

DOSSIER PROFESSIONNEL (DP)

- Enfin je réalise le prototype en ajoutant les liens internes pour simuler la navigation.
 - je simule les liens ou interactions de navigation ainsi que les transitions
 - Je test ma maquette à chaque étape en versions desktop et en version responsive.

<https://youtu.be/-wZ1mJS959c>

Lien de la vidéo démo de ma maquette desktop

<https://youtube.com/shorts/xCjcmq3yQo>

Lien de la vidéo démo de ma maquette mobile

3. L'INTÉGRATION

- Je créer la structure HTML

La structure global

DOSSIER PROFESSIONNEL (DP)

```
1  <!DOCTYPE html>
2  <html lang="fr">
3
4  <head></head>
5
6  <body >
7      <div id="topbar">      </div>
8
9      <header id="header" itemscope itemtype="https://schema.org/Person" >
10         <div class="entete" > </div>
11         <div class="icons-bar" ></div>
12     </header>
13
14     <nav>
15         <div class="menu-items">
16             <a href="#about"></a>
17         </div>
18
19         <div class="menu-items">
20             <a href="#cursus"></a>
21         </div>
22     te
23         <div class="menu-items">
24             <a href="#portfolio"></a>
25         </div>
26
27         <div class="menu-items">
28             <a href="#contact"></a>
29         </div>
30     </nav>
31
32
33     <section id="about">
34         <h3 > </h3>
35         <article ></article>
36     </section>
37
38     <section id="cursus">
39         <h3> </h3>
40         <article ></article>
41     </section>
42
43
44     <section id="portfolio">
45         <h3> </h3>
46         <div></div>
47     </section>
48
49     <section id="contact">
50         <h3> </h3>
51         <div ></div>
52     </section>
53
54 </body>
55
56 <footer></footer>
57
```

DOSSIER PROFESSIONNEL (DP)

Mon portfolio est divisé en 4 sections, la balise <section> est utilisée pour définir une section spécifique de la page, identifiée par l'attribut id= " nom de la section ".

À l'intérieur de la balise <section>, il y a un en-tête de niveau 3 <h3> .

Ces 4 sections correspondent au 4 boutons de la barre de navigation.

J'ai pris soin d'utiliser les bonnes balises tel que header, nav, article, section, ol, blockquote et d'utiliser également la bonne sémantique pour les titres h1,h2,h3,h4.

Comment j'ai organisé mon menu Stylisé

J'ai utilisé des divs avec des class spécifiques pour créer des éléments de menu stylisés.

J'ai utilisé les balise <a> pour créer des liens vers les sections spécifiques de la page en utilisant href.

J'ai utilisé la balise pour afficher les icône du menu et la balise pour afficher un texte associé à chaque icône.

L'utilisation de checkbox pour le menu burger et le menu latéral d'icônes

Dans mon code, j'ai utilisé les checkboxes pour créer des éléments interactifs qui permettent d'afficher ou de masquer du contenu lorsqu'ils sont cochés ou décochés. Voici comment cela fonctionne :

J'ai créé les checkboxes avec la balise <input> et l'attribut type défini sur "checkbox".

Exemple : <input type="checkbox" id="toggle">

J'ai associé les checkboxes à des labels avec de l'attribut for et l'id correspondant de la checkbox.

Exemple : <label for="toggle">≡</label>

Lorsqu'un label est cliqué, la checkbox correspondante est cochée ou décochée, ce qui modifie son état.

Pour afficher ou masquer du contenu en fonction de l'état de la checkbox,j'ai utilisé des sélecteurs CSS et des règles CSS.

Dans mon code HTML , les checkboxes sont utilisées pour créer un menu burger pour la barre de navigation supérieure. Lorsque la checkbox est cochée, une classe CSS est ajoutée à la div parente, ce qui modifie l'apparence du menu et

DOSSIER PROFESSIONNEL (DP)

affiche les options de navigation. Lorsqu'elle est décochée, la classe est supprimée, masquant ainsi le menu.

J'ai utilisé le même mécanisme pour les checkboxes associées aux icônes de mail et de téléphone. Lorsque la checkbox est cochée, la div contenant les informations de contact est affichée. Lorsqu'elle est décochée, la div est masquée.

j'utilise les checkboxes pour contrôler l'affichage ou le masquage de contenu en fonction de leur état (coché ou décoché). Cela me permet d'ajouter des fonctionnalités interactives à la page HTML sans utiliser javaScript.

Améliorer le référencement (SEO) de mon Portfolio grâce à des attributs

Je réalise des schémas de données grâce au site schema.org :
(<https://schema.org>)

En utilisant des attributs ensemble, je décris un élément HTML en le reliant à un schéma de données.

Je définis son type ('itemtype') et j'identifie ses propriétés ('itemprop'). j'utilise itemscope pour 'alerter' sur l'importance des informations itemtype et itemprop.

J'ai utilisé cette technique plusieurs fois

- Pour me décrire itemtype="<http://schema.org/Person>" & itemprop="description"
- Pour décrire mes compétences , mes expériences pro et mes formations.
- Pour décrire mes projets

- Ecrire le fichier CSS desktop & responsive à l'aide des médias query

J'ai séparé mes déclarations CSS en plusieurs parties bien distincts et j'ai repris ce classement dans les media query.

- général, body,h1,h2,h3,h4, header, icone barre, navbar, menu, section about etc...

Général

- J'ai fait une réinitialisation CSS (* { box-sizing: border-box; margin: 0; padding: 0; }) J'ai défini que tous les éléments du document doivent avoir une marge et un padding à 0 par défaut.

DOSSIER PROFESSIONNEL (DP)

- J'ai configuré le comportement de défilement en douceur pour la page HTML.
- J'ai spécifié que les éléments "strong" doivent utiliser la police de caractères "Nunito" sans empattement.
- J'ai défini que les liens non visités doivent avoir une décoration de texte supprimée.

Body

- J'ai défini la couleur de fond de la une page et la couleur du texte
- j'ai choisi la police Nunito pour tout le corps de page
- J'ai défini un curseur personnalisé à partir d'une image (que je n'arrive pas à faire fonctionner)

Je définis le style des éléments h1,h2,h3 et h4 ainsi que des pseudo éléments qui leurs sont rattachés.

Header

- J'ai défini la hauteur de l'en-tête à 60px.
- J'ai configuré la couleur de fond de l'en-tête
- J'ai spécifié que l'en-tête doit être fixé en haut de la page lors du défilement.
- J'ai utilisé les flexbox pour les éléments de l'en-tête afin de les aligner horizontalement.
- J'ai ajouté une marge en haut et en bas pour l'en-tête.

Icon bar

- J'ai utilisé les flexbox pour les icônes de la barre de navigation afin de les aligner horizontalement.
- J'ai défini une marge à gauche et à droite pour les icônes de la barre de navigation.
- J'ai spécifié une transition de couleur de 0,3 seconde pour les icônes de la barre de navigation lorsqu'elles sont survolées.

J'ai stylisé les boutons mail et phone

Barre de navigation

- J'ai défini la hauteur de la barre de navigation à 60px.
- J'ai spécifié la couleur de fond de la barre de navigation comme #573B30.

DOSSIER PROFESSIONNEL (DP)

- J'ai utilisé un conteneur flex pour les éléments du menu afin de les aligner horizontalement.
- J'ai ajouté une marge à gauche pour les éléments du menu.

Je stylise ensuite chacune des sections chacun des éléments des sections

Je stylise le footer

La partie responsive du site

Dans le même ordre que précédemment je vais écrire le code nécessaire pour que le site soit parfaitement responsive en utilisant un seul breakpoint à 950px.

- Je vais faire disparaître certains élément (notamment les élément de décoration placé en background)
- Je vais placer des éléments en colonne et non plus en ligne.

4. LA REVUE DE CODE

J'ai ensuite testé mon code en le passant dans plusieurs outils.
Cela m'a permis de corriger des erreurs .

J'ai testé la vitesse d'affichage avec page speed
J'ai testé si mon code CSS était correct et les performance du site sur validator W3C
J'ai testé l'accessibilité sur wave
J'ai testé les liens sur site24X7

bien sûr, suite à cela, j'ai réalisé toutes les corrections nécessaires.

En conclusion :

Avec du recul et à ce stade de mon apprentissage, je trouve que mon code est assez bien organisé, l'utilisation des balises et des schémas est également assez efficace. Je pense pouvoir améliorer encore mon portfolio en retirant les barres de scroll et en remplaçant pour les deux premières section par des carousel, je remplacerais également la barre de scroll des projet par une présentation de cartes mieux dimensionnées et placés en wrap.

De plus je modifierais toutes mes police en rem et j'essayerais d'uniformiser les unités de mesure en pourcentage , px ou vh et vw. Je vérifierais la bonne utilisation des positions relative et absolue.

DOSSIER PROFESSIONNEL (DP)

Il me reste également à mettre en place l'option copier/coller pour le mail et le CV et également l'accès direct au téléphone sur mobile et à la boîte mail sur mail .

2. Précisez les moyens utilisés :

5. LES RESSOURCES UTILISÉES

L'inspiration & Figma

- <https://www.hostinger.fr/tutoriels/portfolio-developpeur-web>(Portfolio de devlopeur Web (exemples pour s'inspirer)
- chaîne youtube JessUI Design Jessica trocmé (figma)
<https://www.youtube.com/channel/UCxYGJ6GQpu--u8s9ovWX57Q>

HTML & CSS

- cours HTML / CSS graphicart <https://grafikart.fr/>
- développer mozilla.org .
- w3school <https://www.w3schools.com/>
- PDF css-flexbox poster (annexes)
- PDF CSS ressource (annexes)
- PDF HTML ressource (annexes)
- VSC mon portfolio (CSS + HTML)

La revue de code

- Pour tester un la vitesse d'affichage
<https://pagespeed.web.dev/>
- Test CSS !
<https://jigsaw.w3.org/css-validator/>
- L'accessibilité

DOSSIER PROFESSIONNEL (DP)

<https://wave.webaim.org/>

- Les performances <https://validator.w3.org/>
- vérifier les liens du site <https://www.site24x7.com/fr/tools/verificateur-de-liens.html>

3. Avec qui avez-vous travaillé ?

J'ai travaillé seule, mais je me suis tournée vers mon formateur Nicolas 'Mo' Le Beuzit lorsque j'étais vraiment bloquée sur certains points. (notamment pour le débogage) et c'est ainsi que j'ai appris à travailler avec l'inspecteur.

4. Contexte

Nom de l'entreprise, organisme ou association

Formation IMTS 1ère année

Période d'exercice ➔ Du : 08 Nov 2022 au : 09 Nov 2022

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Activité-type 2

Création et réalisation d'un site Wordpress avec Divi et Woocommerce

Exemple n° 1 - Les savons de Malo, boutique en ligne de savons artisanaux

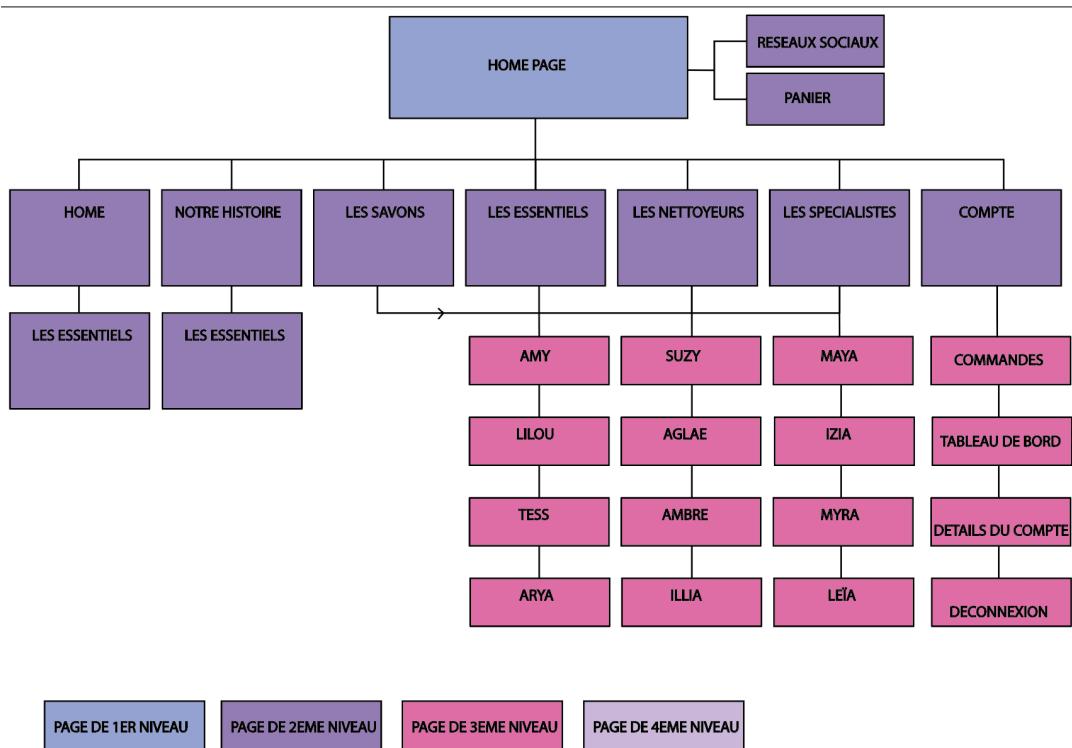
1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

1. LE CONTEXTE

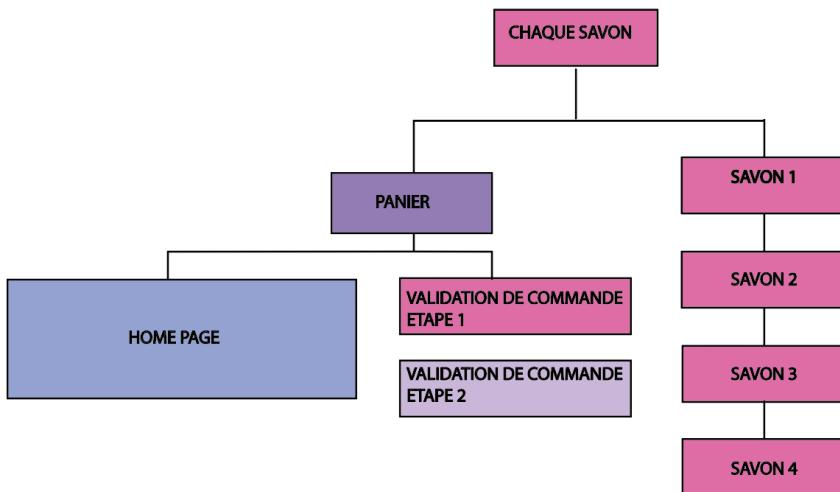
Je suis à l'origine d'un projet de création de savons. Mon objectif est de créer un site vitrine qui présente l'entreprise "Les Savons de Malo" et de mettre en avant l'ensemble des produits disponibles à la vente.

2. JE CRÉE L'ARBORESCENCE DU SITE

J'ai essayé de faire en sorte que chaque page amène à la boutique et que des produits soient proposés aux moments opportuns.



DOSSIER PROFESSIONNEL (DP)



3. JE CRÉE MES CONTENUS

Je définis le concept et l'identité de la marque Les Savons de Malo “.

J'effectue une veille concurrentielle pour identifier les besoins du site, le demande et la concurrence et également pour inspiration.

Je crée ma liste de produits avec la description des chacun et les prix.

Je créer un contenu attrayant (texte, photos, vidéos) sans oublier la notion de SEO dans mes textes et mes titres.

Des Call to action sont présents par des boutons visibles avec des textes et des icônes chaleureux et attrayants.



DOSSIER PROFESSIONNEL (DP)

4. JE CRÉE LA CHARTE GRAPHIQUE

DOSSIER PROFESSIONNEL (DP)

Charte Graphique



LES SAVONS DE MALO

LES SAVONS DE MALO



#056F96 #89BAD6 #C8DDEC #FFFFFF

yanonekaffeesatz

abcdefghijklmnoprstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

light

abcdefghijklmnoprstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

regular

abcdefghijklmnoprstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

bold

American Typewriter

abcdefghijklmnoprstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

light

abcdefghijklmnoprstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

regular

abcdefghijklmnoprstuvwxyz
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0123456789

bold

En ce qui concerne cette charte graphique, elle avait été travaillée au préalable pour un projet de création de savons " les Savons de Malo ". Le projet présenté n'est pas si fictif que ça ...

DOSSIER PROFESSIONNEL (DP)

Le logo à été créé par Thibaut Leclercq graphiste/ illustrateur de talent :
<https://thibgraphic.com>.

5. J'INSTALLE WORDPRESS

3.1 : Je choisi un hébergeur et un plan d'hébergement

C'est Nicolas qui s'est occupé de cette partie car il a pris en charge un domaine commun à tous.

Nicolas a choisi O2 Switch pour des raisons de prix et de fiabilité.

Le choix de l'hébergeur est important pour la sécurité.

Il a opté pour un plan d'hébergement avec un certificat SSL.

En exécutant le site en HTTPS (protocole de transfert hypertexte sécurisé), nous assurons la sécurité, ainsi qu'un avantage pour le référencement SEO, car Google a déclaré que c'était un facteur de classement.

3.2 : J'enregistre un nom de domaine

3.3 : J'installe wordpress

- La dernière version a été choisie pour des raisons évidentes de sécurité.
- Je suis attentive à la sélection des mots de passe, qui doivent être soigneusement choisis et vraiment fiables.
- Cela peut être facilement effectué par l'hébergeur ou manuellement en suivant les instructions fournies par WordPress.

6. JE STYLISE LE SITE

4.1 : Je met en place mon environnement de travaille dans wordpress

je commence par désinstaller tous les plugins qui ne me seront pas utiles.

Ensuite j'installe ceux que je souhaite ajouter (toujours la dernière version pour une question de sécurité)

- **Divi** (Nous avons choisi Divi car c' est un plugin très répandu et sécuritaire , il a une interface plutôt agréable et il offre beaucoup de flexibilité. De plus Divi est bien pensé pour les sites responsives et il intègre parfaitement WooCommerce. Divi a des options

DOSSIER PROFESSIONNEL (DP)

pour le référencement avec lesquelles on peut configurer facilement les balises H1, les urls etc...)

- **WooCommerce** (plateforme de e-commerce sécurisé)
- **SecuPReSS Free - Sécurité WordPress** (assure la sécurité du site)
- **UpdraftPlus - Sauvegarde / Restauration** (assure des sauvegardes de façon gratuite , et automatiser. De plus la sauvegarde peut être envoyée à un service tier (par exemple google drive)
- **WP - Optimize - Nettoyer, compresser, mettre en cache** (De la même team que UpdraftPlus,c'est une extension d'optimisation de Wordpress. Elle peut nettoyer la base de donnée, automatiser la compression des images, optimiser le cache...)
- **Yoast SEO pour optimiser le SEO** (c'est une extension d'optimisation SEO qui peut tester les pages et leur pertinence en fonction des mots clés que l'on choisit)

ensuite, je configure mes paramètres dans réglages (URL du site, visibilité, page d'accueil temporaire ..)

DOSSIER PROFESSIONNEL (DP)

| Action | Description | Autres options |
|--|---|-------------------------------|
| Extension | Mises à jour automatiques | |
| SecuPress Free — Sécurité WordPress | Plus qu'une extension, la garantie d'un site Web protégé par des experts. Réglages Docs Support Désactiver | Activer les mises à jour auto |
| UpdraftPlus - Sauvegarde/Restauration | Sauvegarde et restauration : sauvegarder localement, ou sur Amazon S3, Dropbox, Google Drive, Rackspace, (S)FTP, WebDAV & e-mail, en planification automatique. Support Premium/Pro Réglages Désactiver Visite guidée | Activer les mises à jour auto |
| WooCommerce | Une boîte à outils eCommerce qui vous aide à tout vendre. Magnifiquement. Réglages Désactiver | Activer les mises à jour auto |
| WP-Optimize - Nettoyer, compresser, mettre en cache. | WP-Optimize makes your site fast and efficient. It cleans the database, compresses images and caches pages. Fast sites attract more traffic and users. Optimiser Réglages Premium Désactiver | Activer les mises à jour auto |
| Yoast SEO | La première solution SEO tout-en-un pour WordPress, y compris l'analyse des pages de contenu, les plans de site XML et bien plus encore. Passer à la version premium Finissez votre first-time configuration FAQ Réglages Désactiver | Activer les mises à jour auto |
| Extension | Mises à jour automatiques | |

4.2 : Je choisi un template DIVI

- Toujours la dernière version pour la sécurité
- Le choix du template n'était pas obligatoire, mais j'en ai trouvé un qui m'inspirait . Finalement j'ai peut-être perdu plus de temps à modifier le template que si j'étais partie de zéro.

4.3 : Je configure les éléments essentiels au site

- Les polices, les tailles de typographie
- Les coloris du site (on peut créer une petite gamme de couleur)
- Les boutons
- peut déjà rentrer nos éléments graphiques (logo...)

DOSSIER PROFESSIONNEL (DP)



4.4 : Je crée le menu et la page Home

- Tout le style est à mettre en place par l'onglet apparence. Comme vu ci dessus, c'est de là que l'on peut styliser le menu, les paramètres généraux, le pied de page, les boutons, le style en version responsive et également le WooCommerce.

4.5 : Je crée le reste des pages

- Wordpress fonctionne avec un éditeur de page en module et en bloc.
- Les éléments sont préconstruits mais on peut les déplacer, ajouter, supprimer et personnaliser.
(les titres, les formulaires, des images, des boutons , des zones de textes etc...)
- L'éditeur de page fonctionne en glisser déposer, c'est donc extrêmement simple d'utilisation et rapide !
- C'est extrêmement pratique d'avoir l'aperçu en direct , cela fait gagner beaucoup de temps, en effet on peut voir les modifications en instantané.

4.6 : Je crée la partie e-commerce du site

- Je commence par faire une configuration de WooCommerce pour définir les paramètres de base (la configuration des devises, des méthodes de paiements, les options d'expéditions etc)

DOSSIER PROFESSIONNEL (DP)

- J'ajoute les produits , un par un sur le site par l'onglet 'produit => ajouter un produit)
Une description très complète est proposée (des images, les prix , les variantes etc ...
J'ai pu indiquer que mes savons pouvaient être vendus en 100 g ou en 1kg.
- Le gestionnaire de commande se gère dans 'Commandes' à partir du tableau de bord
. (On peut consulter les nouvelles commandes, les traiter, suivre leur statut etc...)
- J'ai souhaité modifier l'apparence de ma partie WooCommerce mais j'ai modifié directement les fichiers du thème principal, je suis maintenant assez frileuse pour faire une mise à jour..
J'aurais dû utiliser un plugin dédié ou utiliser un thème enfant ce qui aurait été plus propre et plus simple pour les mises à jour .

7. JE TEST ET J'OPTIMISE LE SITE

Ensuite grâce à mes plugins j'ai pu vérifier l'optimisation , la sécurité et optimiser le SEO.

8. LE SITE (LA NAVIGATION, L'INTERFACE)

Lien vers la démo du site " Les savons de Malo "

<https://youtu.be/n8Lete3MF6w>

vous pourrez y voir

- Un design cohérent avec les produit et l'ADN des savons de Malo
- Un site ergonomique et convivial
- Un site marchand qui fonctionne

2. Précisez les moyens utilisés :

J'ai regardé les tutos Wordpress de :

- Graphikart
- Tuto youtube : formation DIVI : créer un site internet sur WordPress de A à Z (débutant)

DOSSIER PROFESSIONNEL (DP)

https://www.youtube.com/watch?v=duX_QGToCNE

3. Avec qui avez-vous travaillé ?

J'ai travaillé seule, mais je me suis tournée vers mon formateur Nicolas 'Mo' Le Beuzit lorsque j'étais vraiment bloquée sur certains points, notamment pour la prise en main de l'interface de wordpress, certains bug de construction et la modification du code dans les fichiers du site)

4. Contexte

Nom de l'entreprise, organisme ou association ▶

Formation IMTS 1ère année

Période d'exercice ▶ Du : 12/12/22 au : 23/12/22

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Activité-type 3 Réalisation d'un site de type Allociné en React

Exemple n° 1 - Film flow

1. Décrivez les tâches ou opérations que vous avez effectuées, et dans quelles conditions :

1.LE CONTEXTE

A l'aide de ce cahier des charges et de votre kanban personnel,
vous aurez pour mission de créer l'application FilmFlow.

Ce projet aura lieu les 4 prochaines semaines. (du 27/02 au 17/03).

Un processus de demande de support a été défini (tickets).

En pièce jointe, vous trouverez:

- Le cahier des charges
- Le process demande de support
- Lien vers le wireframe

Vous recevrez par e-mail votre lien d'invitation vers votre kanban personnel.

Vous devrez présenter ce travail vendredi 10/03 au CDA. Pour cela, vous aurez au préalable créé un support ou vous expliquez votre projet. Ce temps de présentation durera 15 minutes. Le jeudi 9/03 sera dédié à la préparation de ce support. Le projet devra donc être achevé mercredi 8/03.

Lien vers le kanban support :

<https://trello.com/invite/b/nY5pU6Kt/ATTI171ec923dfb8c8693ef7f1bf731267acEBAB7575/ticket>

DOSSIER PROFESSIONNEL (DP)

2.POUR UN BON DÉPART

- Lire attentivement le cahier des charges
Prendre connaissance du wireframe, de la charte graphique, des fonctionnalités attendu
- Prendre connaissance du kanban et ses fonctionnalités
- Lire la documentation TV MAZE, et repérer les URL qui vont m'être utiles pour chacune des requêtes à effectuer.
- Mettre en place mon environnement de travail
 - J'avais déjà télécharger Node.js au préalable
 - Par le terminal je créer le fichier avec le nom du projet
 - Je lance le serveur de développement avec la commande 'npm start' et cela ouvre automatiquement l'application REACT dans mon navigateur par défaut (chrome)
cela permet de voir les modification en direct sur le navigateur

3.PRÉPARATION DES DOSSIERS DE COMPOSANTS DANS MON EDITEUR DE CODE (VSC).

- Pour cela je m'appuie sur le kanban dans lesquelles les tâches sont répertoriées, cela me donne les indices pour savoir quels sont les composants attendus.
- Préparer les pages index.html, index.css, main.jsx et App.jsx

4.JE SORS TOUTES MES FONCTIONNALITÉES AVEC DES REQUÈTES

A- Main.jsx

main.jsx (que j'aurais pu appeler index) importe plusieurs modules React tels que **React, ReactDOM, BrowserRouter, Route, Routes, Home.jsx, Resultats.jsx, NavBar.jsx, Serie.jsx, Acteurs.jsx, Favoris.jsx, SearchBar.jsx** et **App**.

DOSSIER PROFESSIONNEL (DP)

J'utilise la méthode `createRoot()` de `ReactDOM` pour créer un point de départ de rendu de l'application React et je le passe en élément du DOM avec l'ID `root`.

Avec la méthode `render()`, je rends le composant `<App />` sous `<React.StrictMode>`.

Cela me garantit que l'application suit les bonnes pratiques de React et de détecter des erreurs potentielles dès le développement.

J'utilise `BrowserRouter` pour mettre en place la navigation de l'application en utilisant les composants `Route` et `Routes` pour définir les itinéraires de navigation et les pages à afficher pour chaque itinéraire. Les pages incluses sont `Home`, `Réultats`, `Série`, `Acteurs`, et `Favoris`.

Je crée un composant `NavBar.jsx` pour la barre de navigation de l'application, ainsi qu'un composant `SearchBar.jsx` pour permettre aux utilisateurs de rechercher des contenus dans l'application.

Ce fichier `main.jsx` sert de point d'entrée pour l'application React et met en place la structure générale de l'application en important et en rendant les composants nécessaires pour la navigation et l'affichage des pages.

```
src > main.jsx
 1  import React from 'react'
 2  import ReactDOM from 'react-dom/client'
 3  import { BrowserRouter, Route, Routes } from 'react-router-dom'
 4  import { Home } from './Pages/Home/Home'
 5  import { Resultats } from './Pages/Resultats/Resultats'
 6
 7
 8
 9  import './index.css'
10  import { NavBar } from './Component/NavBar/NavBar'
11  import { Serie } from './Pages/Serie/Serie'
12  import { Acteurs } from './Pages/Acteurs/Acteurs'
13  import { Favoris } from './Pages/Favoris/Favoris'
14  import { SearchBar } from './Component/SearchBar/SearchBar'
15  import { App } from './App'
16
17
18  ReactDOM.createRoot(document.getElementById('root')).render(
19    <React.StrictMode>
20      |   <App />
21      |   </React.StrictMode>,
22    )
23
24
25
```

DOSSIER PROFESSIONNEL (DP)

B- App.jsx

Je crée le fichier **App.jsx**. Ce composant joue un rôle central dans l'application, il définit la structure de base de l'application et gère les fonctionnalités suivantes:

- Importation des bibliothèques et des composants nécessaires :
 - La bibliothèque **react-router-dom** pour gérer les routes de l'application.
 - Les pages de l'application (**Home**, **Resultats**, **Serie**, **Acteurs**, **Favoris**).
 - Le composant **Modal** de la bibliothèque **react-modal** pour afficher une boîte de dialogue modale. (Définition du style personnalisé pour la boîte de dialogue modale, ces déclaration CSS ne devraient pas être dans ce fichier mais dans le fichier index.css)
- Je lie la boîte de dialogue modale à l'élément racine de l'application
- Je définis le composant **App**

je définis le composant **App** avec une fonction fléchée en utilisant **useState** pour gérer l'état de la boîte de dialogue modale.

 - L'état **isOpen** est initialisé à false, ce qui signifie que la boîte de dialogue modale est initialement fermée.
 - Deux fonctions, **openModal** et **closeModal**, sont définies pour ouvrir et fermer la boîte de dialogue modale en mettant à jour l'état **isOpen**.
- Le return
 - J'utilise le composant **BrowserRouter** pour gérer les routes de l'application.
 - Le composant **NavBar** affiche une barre de navigation, avec les fonctions **openModal** et **closeModal** passées en tant que **props**.
 - Le composant **Modal** affiche la boîte de dialogue modale lorsqu'elle est ouverte. Il contient le composant **SearchBar**.
 - Je définis les différentes routes de l'application à l'aide des composants **Routes** et **Route** de **react-router-dom**. Chaque route correspond à un chemin d'URL spécifique et affiche le composant correspondant lorsque l'URL correspond à ce chemin.

Pour résumer tout cela, le fichier **App.jsx** joue un rôle clé dans la mise en place des routes de l'application, l'affichage de la barre de navigation, la gestion de la boîte de dialogue modale et l'affichage des différents composants en fonction de l'URL actuelle.

DOSSIER PROFESSIONNEL (DP)

J'y importe les composants nécessaires, y définit la structure de base de l'application et y pilote la navigation entre les pages.

DOSSIER PROFESSIONNEL (DP)

```
src > App.jsx > ...
1  import { useState } from 'react';
2  import Modal from 'react-modal';
3  import { BrowserRouter, Route, Routes } from 'react-router-dom';
4  import { Acteurs } from './Pages/Acteurs/Acteurs';
5  import { Favoris } from './Pages/Favoris/Favoris';
6  import { Home } from './Pages/Home/Home';
7  import { Resultats } from './Pages/Resultats/Resultats';
8  import { Serie } from './Pages/Serie/Serie';
9
10 //import components
11 import { NavBar } from './Component/NavBar/NavBar';
12 import { SearchBar } from './Component/SearchBar/SearchBar';
13
14
15
16 //style personnalisé pour la boîte de dialogue modale
17 const customStyles = {
18   content: {
19     top: '50%',
20     left: '50%',
21     right: 'auto',
22     bottom: 'auto',
23     marginRight: '-50%',
24     transform: 'translate(-50%, -50%)',
25     borderRadius: '15px',
26     backgroundColor: 'rgb(78, 73, 73)',
27   },
28
29   input: {
30     backgroundColor: 'rgb(100, 92, 92)',
31   },
32
33
34   ReactModal__Overlay: {
35
36     backGroundColor: 'rgb(71, 92, 92, 0.83)'
37   }
38
39
40 }
41
42
43 // Make sure to bind modal to your appElement (https://reactcommunity.org/react-modal/accessibility/)
44 // fonction Modal.setAppElement pour lier la boîte de dialogue modale à l'élément racine de l'application.
45 Modal.setAppElement('#root');
46
47 //composant App qui contient la barre de navigation et la boîte de dialogue modale.
48 export const App = () => {
49
50
51   //utilise useState pour stocker l'état de la boîte de dialogue modale et les fonctions openModal
52   // et closeModal pour ouvrir et fermer la boîte de dialogue modale.
53   const [isOpen, setIsOpen] = useState(false);
54
55
56   function openModal() {
57     setIsOpen(true);
58   }
59
60   function closeModal() {
61     setIsOpen(false);
62   }
63
64   return (
65     <BrowserRouter>
66       <NavBar openModal={openModal} closeModal={closeModal} />
67
68       <Modal
69         isOpen={isOpen}
70         onRequestClose={closeModal}
71         style={customStyles}
72         contentLabel="Example Modal"
73       >
74         <SearchBar openModal={openModal} closeModal={closeModal} />
75       </Modal>
76
77       <Routes>
78
79         <Route path='/' element={<Home />} />
80         <Route path='/resultats/' element={<Resultats />} />
81         <Route path='/serie/:id' element={<Serie />} />
82         <Route path='/acteurs/:id' element={<Acteurs />} />
83         <Route path='/favoris' element={<Favoris />} />
84
85       </Routes>
86     </BrowserRouter>
87   // React Router pour définir les différentes routes de l'application et pour afficher les différentes pages en
88   // fonction de l'URL actuelle.
89
90 }
```

DOSSIER PROFESSIONNEL (DP)

Il faut préciser que Le fichier **main.jsx** sert de point d'entrée principal de l'application, il définit la structure globale de l'application, importe les différents modules et composants. Une fois les modules et composants importés, le fichier **main.jsx** utilise la méthode **ReactDOM.createRoot()** pour créer un point de départ de rendu dans l'application.

Il indique où l'application sera rendue dans le navigateur.

Dans le fichier HTML il y a la balise `<div id="root"></div>`.

Le fichier **main.jsx** utilise la méthode **ReactDOM.createRoot()** pour indiquer que c'est dans cette balise avec l'ID "root" que l'application doit être rendue.

(c'est le premier fichier qui sera exécuté lors du démarrage de l'application)

Le fichier **App.jsx**, définit la structure de base de mon application et contient les composants et la logique pour la navigation, comme les composants **BrowserRouter**, **Routes** et **Route**. Le fichier **App.jsx** est le composant racine de l'application.

Ce que je retiens le mieux est que le fichier **main.jsx** est un point d'entrée de l'application et de la configuration alors que le fichier **App.jsx** est le composant principal de l'application. Le fichier **main.jsx** utilise le composant **App** pour rendre l'application dans l'élément DOM avec l'ID **root**.

C- Home.jsx ⇒ SearchBar.jsx

Dans le fichier **Home.jsx** je définis le composant **Home** qui est exporté pour être utilisé dans d'autres parties de l'application. Dans ce fichier j'importe plusieurs éléments React tels que **useEffect** (un hook de React), **BiFontSize**, **Carroussel.jsx**, **NavBar.jsx** et **SearchBar.jsx**.

J'utilise la fonction **useEffect** pour initialiser le stockage local de la liste des favoris. Si la liste n'existe pas déjà dans le stockage local, elle est initialisée avec un tableau vide.

Dans le **return**, le composant **SearchBar.jsx** et le composant **Carroussel.jsx** sont rendus. Le composant **SearchBar.jsx** permet de rechercher du contenu dans l'application et le composant **Carroussel.jsx** est utilisé pour afficher un diapo d'affiche des dernières séries sorties.

La page **Home.jsx** est une page 'd'accueil' dans laquelle j'affiche une barre de recherche et un carrousel. C'est aussi ici que j'initialise le stockage local de la liste des favoris si elle n'existe pas encore.

DOSSIER PROFESSIONNEL (DP)

D-SerachBar.jsx

SearchBar est responsable de l'affichage de barre de recherche.

Voici son fonctionnement :

J'importe certains modules et composants nécessaires :

- un fichier CSS pour les styles
- le composant **FiSearch** des icônes React (loupe)
- le hook **useNavigate** de "react-router-dom" pour la navigation.

J'utilise **useEffect** et **useState** pour gérer les effets et l'état.

Le composant **SearchBar** est défini comme une fonction qui prend deux paramètres (**openModal** et **closeModal**) et retourne du JSX pour l'affichage.

État et gestion des événements :

J'utilise le hook **useState** pour définir l'état **query**, qui contient la valeur de la recherche. Lorsque la valeur de l'input de recherche change, la fonction **handleInputChange** est appelée et met à jour l'état **query** avec la nouvelle valeur.

Soumission du formulaire :

Lorsque le formulaire est soumis (lorsque le bouton de soumission est cliqué), la fonction **handleSubmit** est appelée. Elle empêche le comportement par défaut du formulaire, puis utilise le hook **useNavigate** pour effectuer la navigation vers la page de résultats de la recherche en passant la valeur de **query** dans l'état du navigateur. Ensuite, la fonction **closeModal** est appelée pour fermer la modale de recherche.

Rendu :

Le composant renvoie du JSX qui représente la structure du composant de la barre de recherche. Il s'agit d'un formulaire contenant un champ de texte (input) lié à l'état **query**, qui est mis à jour via la fonction **handleInputChange** lorsqu'il y a des changements. Le formulaire a également un bouton de soumission avec une icône de recherche. Lorsque le formulaire est soumis, la fonction **handleSubmit** est appelée.

C'est ainsi que fonctionne le composant **SearchBar**

DOSSIER PROFESSIONNEL (DP)

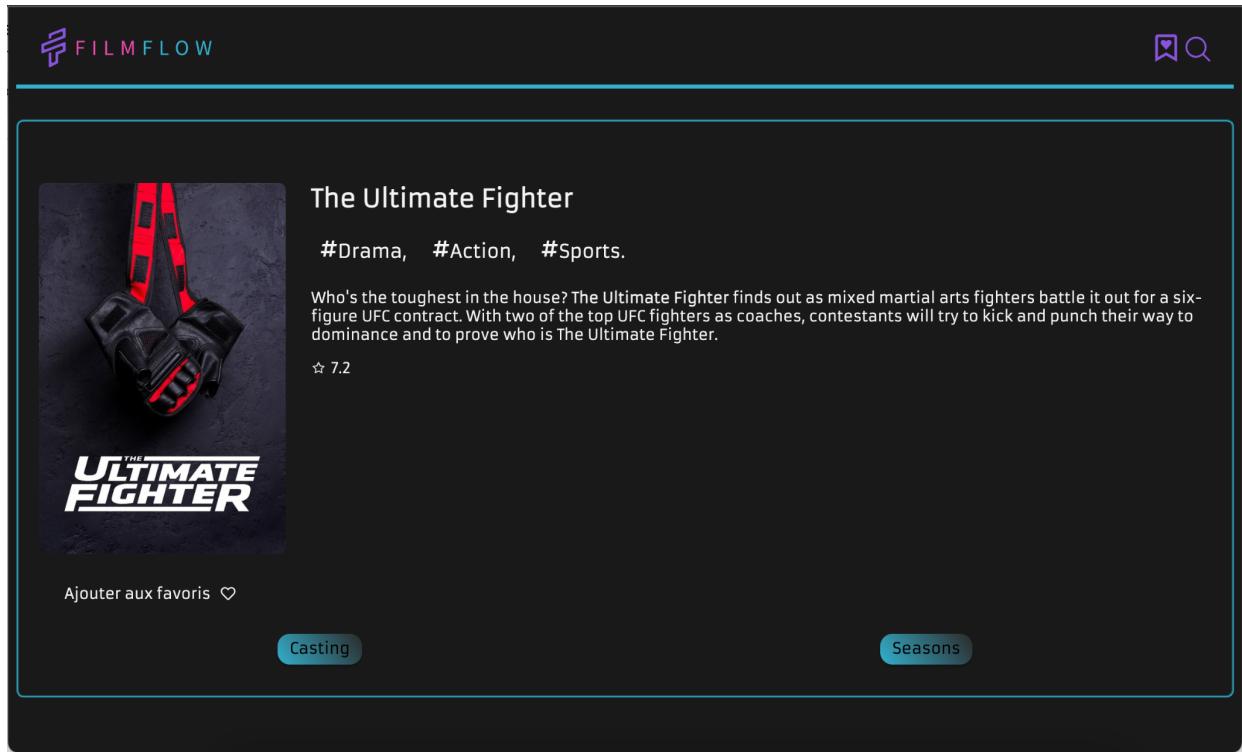
Il affiche une barre de recherche et permet à l'utilisateur de saisir une requête.

E-NavBar.jsx

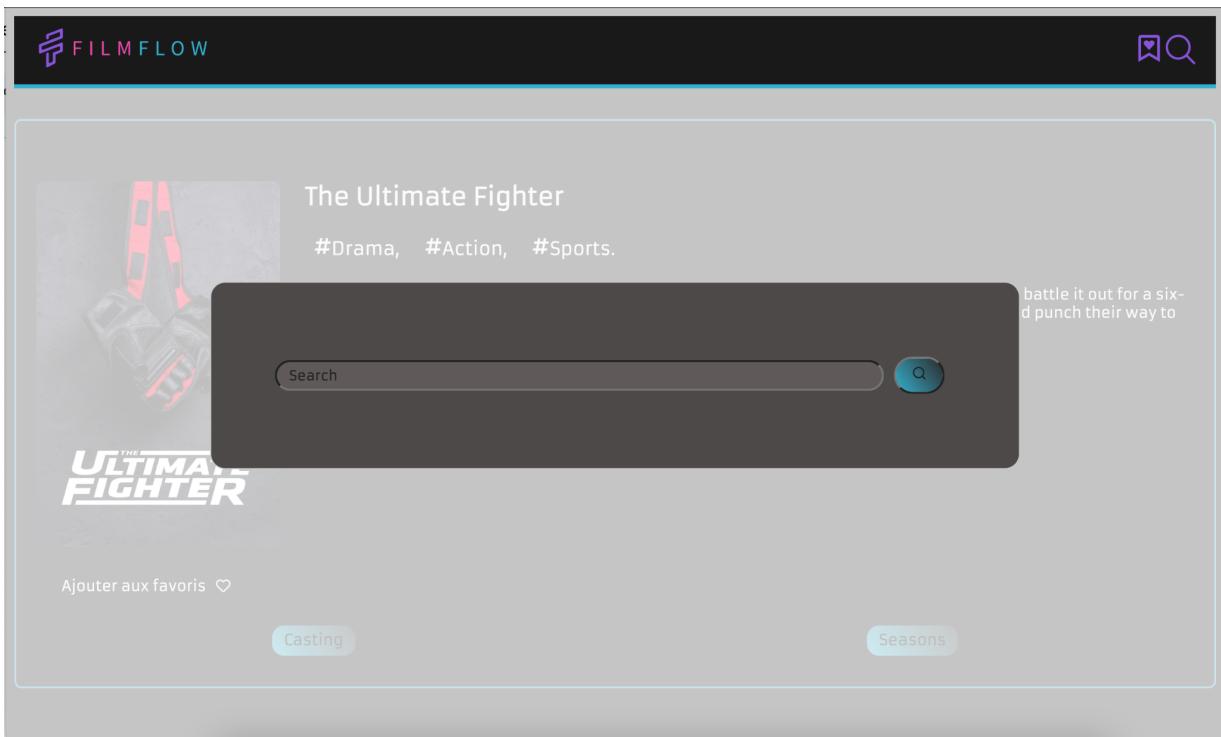
Le composant **NavBar.jsx** crée une barre de navigation pour le site web. Dans ce composant j'utilise la bibliothèque React et les outils **React Router** et **React Icons**. C'est avec un état que je gère l'ouverture et la fermeture d'une barre de recherche.

La NavBar récupère l'emplacement actuel de l'utilisateur avec **useLocation** de React Router. Ensuite on utilise cet emplacement pour déterminer si l'utilisateur se trouve sur la page d'accueil ou non. Ensuite, on utilise **useState** pour gérer l'état de la barre de recherche.

Dans le composant **NavBar.jsx**, il y a une barre de navigation avec un lien vers la page d'accueil et deux icônes. L'icône du cœur est un lien vers la page de favoris et l'icône de la loupe ouvre la barre de recherche en cliquant dessus. (**SearchBar.jsx**)



DOSSIER PROFESSIONNEL (DP)



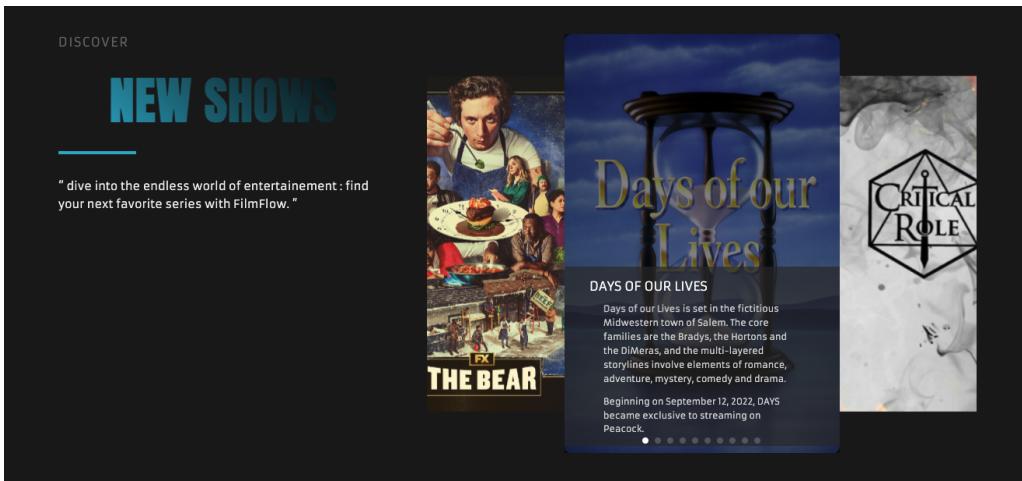
Lorsque la barre de recherche est ouverte, alors un formulaire de recherche s'ouvre avec une zone de texte et un bouton de soumission. Le formulaire de recherche utilise un autre composant **SearchBar.jsx**.

Lorsque l'utilisateur soumet une recherche, la **NavBar.jsx** appelle une fonction **handleSearchSubmit** qui reçoit la requête de recherche en paramètre. Cette fonction est utilisée pour effectuer une recherche et afficher les résultats sur une autre page. (**Resultats.jsx**)

La NavBar gère également un arrière-plan flou lorsque la barre de recherche est ouverte pour mettre en évidence la barre de recherche et rendre le reste de la page moins visible.

F- Carroussel.jsx

DOSSIER PROFESSIONNEL (DP)



Le carrousel a été créé à partir d'un tuto :

(Light Code :<https://www.youtube.com/watch?v=gGAizkIH-BQ>)

Le composant **Carrousel.jsx** utilise les bibliothèques **Swiper** (pour créer un carrousel d'images) et **react-router-dom** pour la navigation.

Le composant récupère les 10 premiers programmes TV du jour aux États-Unis depuis l'**API TVMaze** et les affiche dans le carrousel. Le code contient également des styles CSS personnalisés pour le carrousel.

** J'explique en détail un carrousel en JS dans mon rapport de stage **

Je choisi donc de passer rapidement sur celui-ci d'autant plus qu'il fait appel à un plugin React

G- Resultats.jsx ⇒ ListeResult.jsx

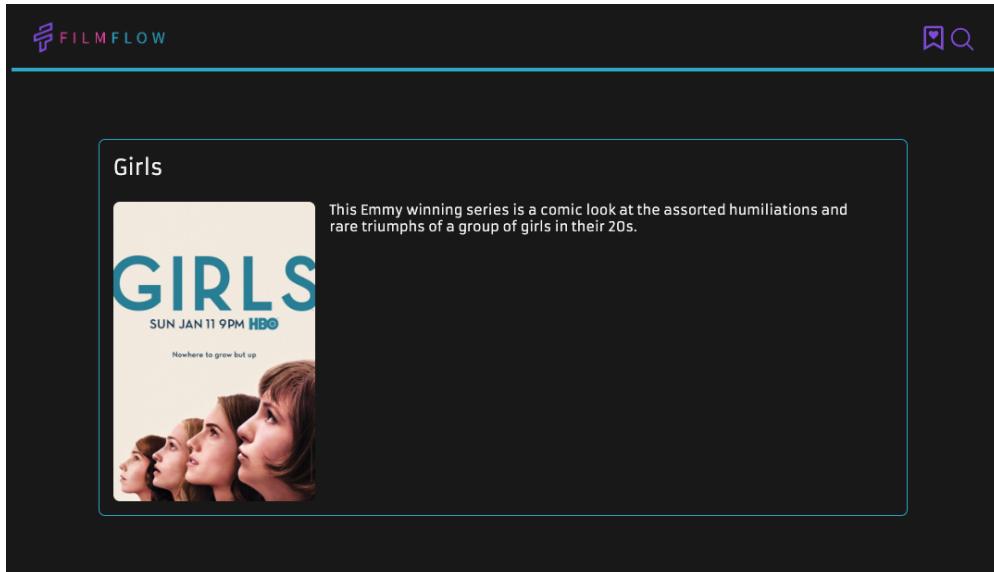
Dans **Resultat.jsx** j'utilise plusieurs éléments, notamment la bibliothèque **react-router-dom** pour la **navigation**, les hook **useState** et **useEffect** pour la **gestion du state** et des **effets de bord**.

Dans **Resultat.jsx** je récupère un paramètre "**query**" passé à la **route** grâce au hook **useLocation**, j'effectue une **requête API** avec la méthode **fetch** pour récupérer une liste de séries TV depuis l'**API TVMaze**, je stock le résultat dans le **state "shows"**, et affiche chaque élément de "**shows**" sous forme de **liens** vers une **page de détails** grâce au composant "**Link**". J'utilise également un composant appelé "**ListeResult.jsx**"

DOSSIER PROFESSIONNEL (DP)

pour afficher les informations de chaque série. (J'ai choisi de récupérer les 10 dernières séries américaines).

H- ListeResult.jsx⇒ Serie.jsx



"**ListeResult.jsx**" prend en entrée une **prop "valeur"** qui est un **objet** contenant les informations sur une série TV.

Grâce à composant je peux afficher les informations d'une série TV

- son titre (récupéré à partir de "valeur.show.name")
- une image (récupérée à partir de "valeur.show.image")
- un résumé (récupéré à partir de "valeur.show.summary")

Si l'image n'est pas disponible, une image de remplacement stockée dans "`../../assets/image-not-found.jpeg`" est affichée.

J'utilise également la propriété "**dangerouslySetInnerHTML**" pour afficher le résumé de la série, ce qui me permet d'interpréter le contenu HTML dans la propriété "**valeur.show.summary**".

(La propriété "**dangerouslySetInnerHTML**" est une propriété spéciale en React qui permet d'injecter du code HTML directement dans un composant, en utilisant une syntaxe similaire à celle utilisée dans les fichiers HTML.

Cette propriété doit être utilisée avec précaution car elle peut présenter des risques de sécurité en permettant l'injection de code malveillant dans l'application. d'où, le nom "**dangerouslySetInnerHTML**".

DOSSIER PROFESSIONNEL (DP)

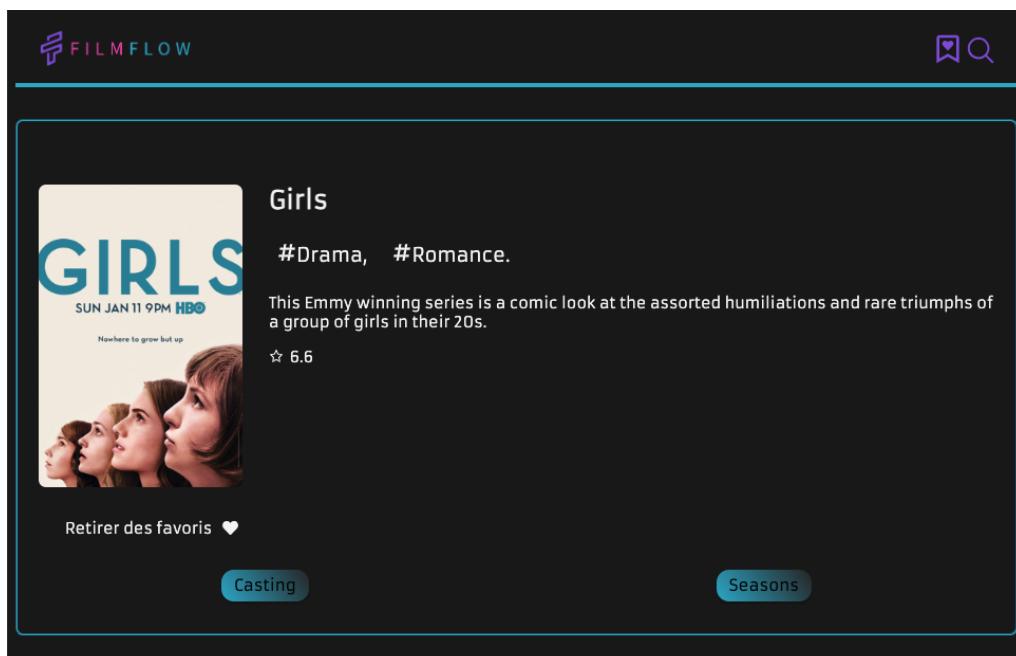
(Cette propriété ne doit être utilisée que dans des cas spécifiques et avec des données de confiance. Dans la plupart des cas, il est préférable d'utiliser des composants React pour afficher des données dans l'application.)

I- Serie.jsx ⇒ Casting.jsx

⇒ Saison.jsx

⇒ DetailsSerie.jsx

⇒ Season.jsx



C'est avec le composant serie.jsx que j'affiche les détails d'une série de télévision. J'utilise l'API de **TMaze** pour récupérer les informations de la série en fonction de son ID. J'affiche également une icône pour ajouter ou retirer la série des favoris, ainsi que des boutons pour afficher le casting de la série ou les saisons de la série. J'utilise également la bibliothèque **React-Toastify** pour afficher des messages de notification lorsqu'une série est ajoutée ou retirée des favoris. La liste des favoris est stockée dans le **localStorage** du navigateur, et est mise à jour à chaque modification de la liste.

J'utilise plusieurs hooks React :

- "useParams" permet de récupérer les paramètres de l'URL, ici l'ID de la série à afficher.

DOSSIER PROFESSIONNEL (DP)

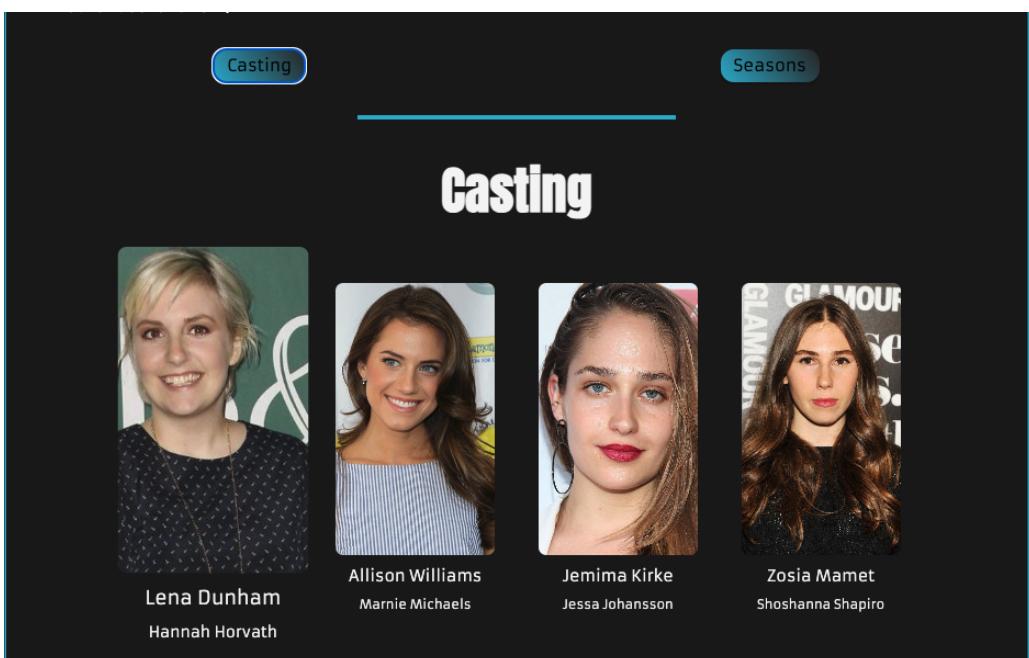
- "**useState**" permet de définir des états locaux dans le composant, utilisés pour stocker des données telles que les informations sur la série ou la liste des favoris.
- "**useEffect**" permet d'effectuer des actions en réponse à des événements tels que le chargement du composant ou la modification d'un état.

Le composant **Serie.jsx** appelle d'autres composants enfants tels que "**Casting.jsx**", "**Seasons.jsx**" et "**DetailsSerie.jsx**" pour afficher des informations supplémentaires sur la série.

J'utilise également la librairie "**react-icons**" pour afficher des icônes, ainsi que la librairie "**react-toastify**" pour afficher des messages d'information à l'utilisateur.

En résumé, avec ce composant j'affiche les détails d'une série télévisée et permet à l'utilisateur de l'ajouter ou de la retirer de ses favoris, ainsi que d'accéder à des informations supplémentaires sur la série.

I- Casting.jsx



Casting.jsx, est responsable de l'affichage des informations de casting pour une série spécifique.

DOSSIER PROFESSIONNEL (DP)

Voici son fonctionnement :

J'importe les modules et composants nécessaires:

- le composant **Link** de "react-router-dom" pour la création de liens,
- le composant **Acteurs** de la page "**Acteurs**"
- un fichier CSS pour les styles.

J'utilise **useEffect** et **useState** du module "react" pour gérer les effets et l'état respectivement.

Le composant **Casting** est défini comme une fonction qui prend un paramètre **idSerie** et retourne du JSX pour l'affichage.

État et requête API :

J'utilise le hook **useState** pour définir l'état **showCast**, qui stocke les informations de **casting**.

J'utilise également le hook **useEffect** pour effectuer une requête API vers l'URL [https://api.tvmaze.com/shows/\\${idSerie}/cast](https://api.tvmaze.com/shows/${idSerie}/cast) chaque fois que l'identifiant de série (**idSerie**) change.

Je convertis les résultats de la requête en JSON et je les stock dans l'état **showCast** à l'aide de la fonction **setShowCasting**.

Return et Affichage :

Le composant renvoie du JSX qui représente la structure du composant de casting.

une ligne horizontale (**<hr>**)

un titre "Casting" (**<h1>**).

une liste (****) contenant des éléments de casting

Pour chaque personne dans **showCast**, j'utilise une balise **<Link>** pour créer un lien vers la page des acteurs correspondante en utilisant l'identifiant de la personne (**person.person.id**).

À l'intérieur du lien, j'utilise une div avec la classe **acteurCard**, contenant une image, le nom de la personne (**person.person.name**), et le nom du personnage (**person.character.name**).

Je définis L'image de l'acteur en utilisant une logique de condition ternaire pour vérifier si une image originale (**person.person.image?.original**) est disponible. Sinon, je vérifie si

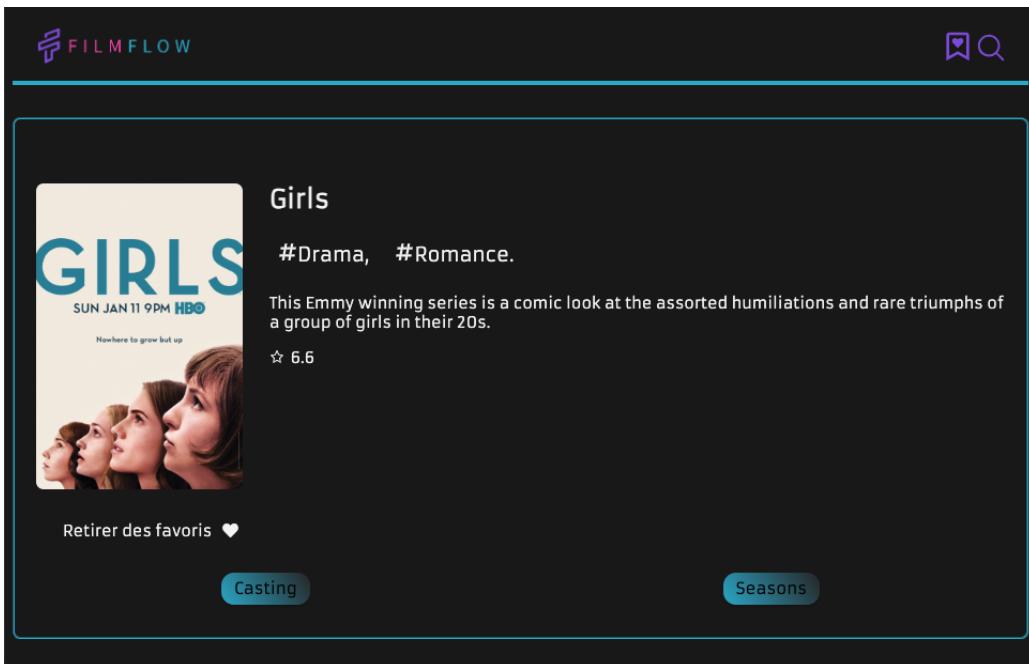
DOSSIER PROFESSIONNEL (DP)

une image de taille moyenne (`person.person.image?.medium`) est disponible. Si aucune des deux n'est disponible, j'utilise une image par défaut (`NotFound`).

J'affiche les noms de la personne et du personnage dans des balises `<p>` avec des classes de style correspondantes.

C'est ainsi que fonctionne le composant Casting. J'effectue une requête API pour obtenir les informations de casting d'une série spécifique et les afficher dans une liste avec des liens vers les pages des acteurs correspondantes.

J- DetailsSerie.jsx



Avec `DetailsSerie.jsx` j'affiche les détails d'une série TV .

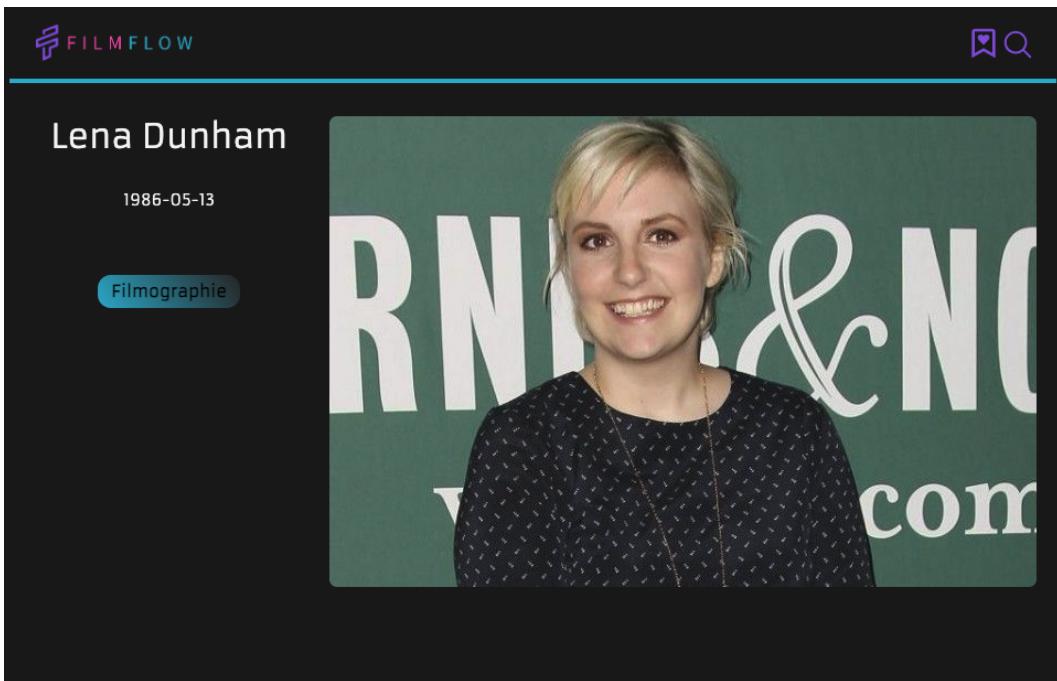
Je commence par afficher l'image de la série en utilisant la propriété "`src`" et en utilisant la logique suivante : si l'image "original" existe, l'afficher, sinon, afficher l'image "medium". S'il n'y a pas d'image, une image de remplacement est affichée.

Ensuite, j'affiche le titre de la série et les genres associés, les genres sont affichés avec l'icône "hashtag" pour chaque genre. Le résumé de la série est également affiché, avec "`dangerouslySetInnerHTML`" pour interpréter les balises HTML (comme vu précédemment).

Enfin, j'affiche la note de la série sous forme d'étoiles, avec la note moyenne de la série à côté.

DOSSIER PROFESSIONNEL (DP)

K- Acteurs.jsx ⇒ ListeFilmographie.js



Ici j'utilise les hooks `useEffect` et `useState` importés de **React** pour gérer l'effet de chargement initial et l'état. Également `useParams` et la fonctionnalité `Link` pour gérer les paramètres d'URL et la navigation.

J'importe également `ListeFilmographie.jsx`.

À l'intérieur du composant voici comment j'ai organisé les choses :

Je récupère les paramètres d'URL avec le hook `useParams`, et l'ID de l'acteur est extrait.

Deux états sont définis : `infoActeur` pour stocker les informations de l'acteur et `filmo` pour indiquer si la filmographie doit être affichée.

J'utilise le hook `useEffect` pour effectuer une requête à l'API **TVMaze** et mettre à jour l'état `infoActeur` avec les informations de l'acteur.

DOSSIER PROFESSIONNEL (DP)

Lorsque le bouton "Filmographie" est cliqué, l'état `filmo` est mis à jour pour afficher le composant `ListeFilmographie`.

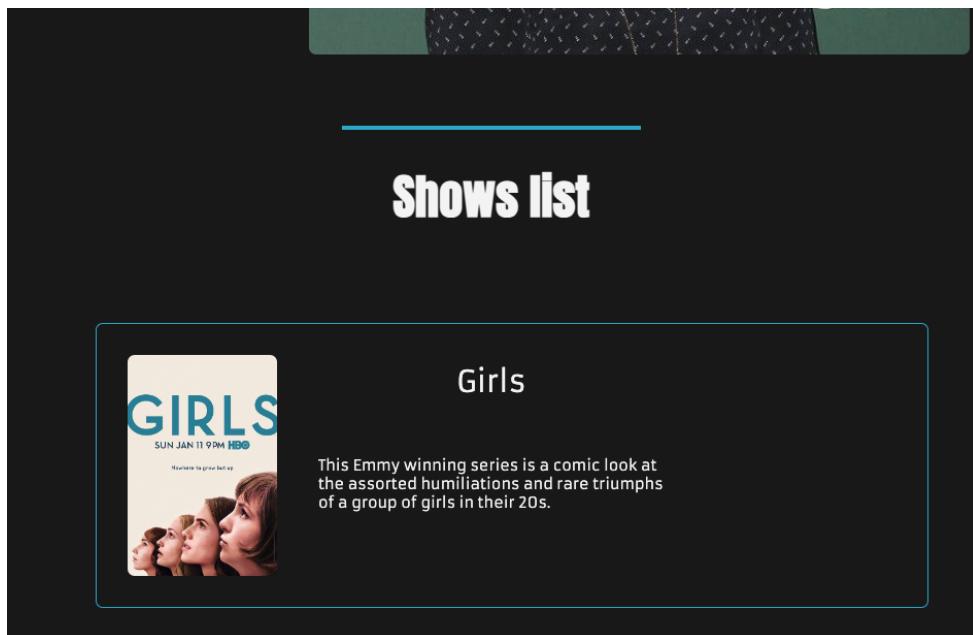
Le composant `Acteur.jsx` affiche

- le nom,
- la date de naissance
- la date de décès de l'acteur,
- Une image de l'acteur. Si l'état `filmo` est true, le composant `ListeFilmographie.jsx` est également affiché.

L- DétailsActeurs.jsx

J'ai créer ce composant pensant qu'il me serait nécessaire au départ (car indiqué dans le Trello) mais finalement il ne m'a pas été utile car les détails sont finalement dans le composant `Acteur.jsx`.

M- ListeFilmographie.jsx

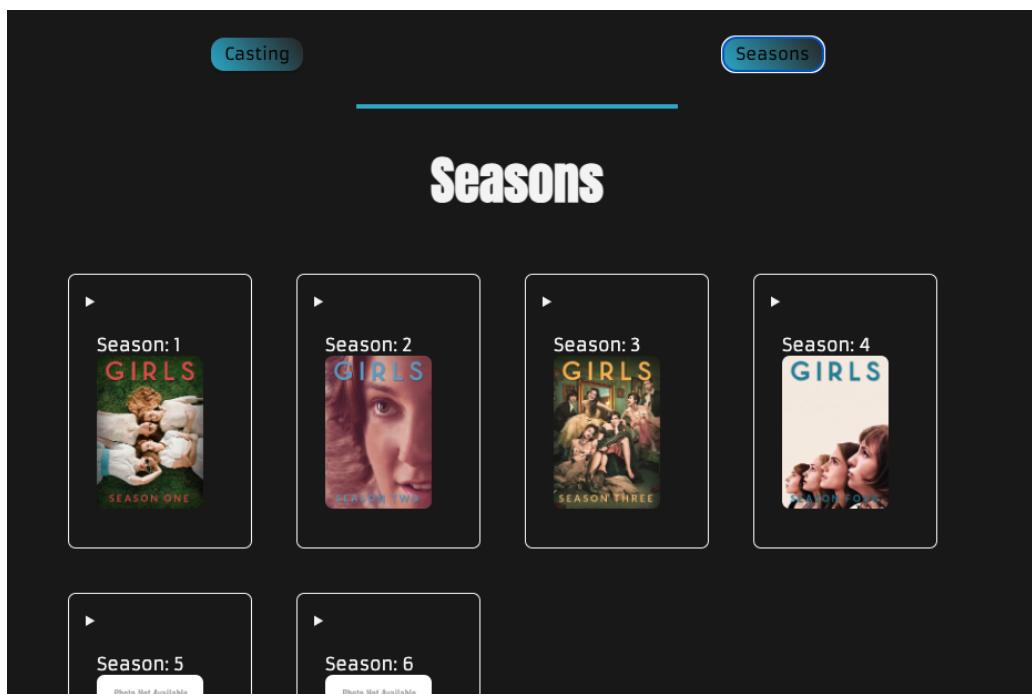


Le composant `ListeFilmographie.jsx` est conçu pour recevoir la valeur "`idSerie`" et j'utilise le hook `useEffect` pour appeler l'API `TVMaze` et récupérer une liste de séries associés à l'ID de la série fournie. La liste est stockée dans l'état "`filmographie`" du

DOSSIER PROFESSIONNEL (DP)

composant. Avec ce composant j'affiche ensuite la liste des séries en utilisant la méthode **map** pour parcourir l'état filmographie et générer une liste d'éléments sur l'interface de l'application. Chaque élément contient une image, le nom et un résumé de la série. Les liens dans chaque élément redirigent vers une page détaillée de chaque série.

N- Saisons.jsx



Avec ce composant **Saison.jsx** j'affiche la liste des saisons de la série sélectionnée..

Dans ce composant j'utilise les hooks **useState** et **useEffect** pour gérer l'état de la liste des saisons et l'appel à l'API **TVMaze** pour récupérer ces informations.

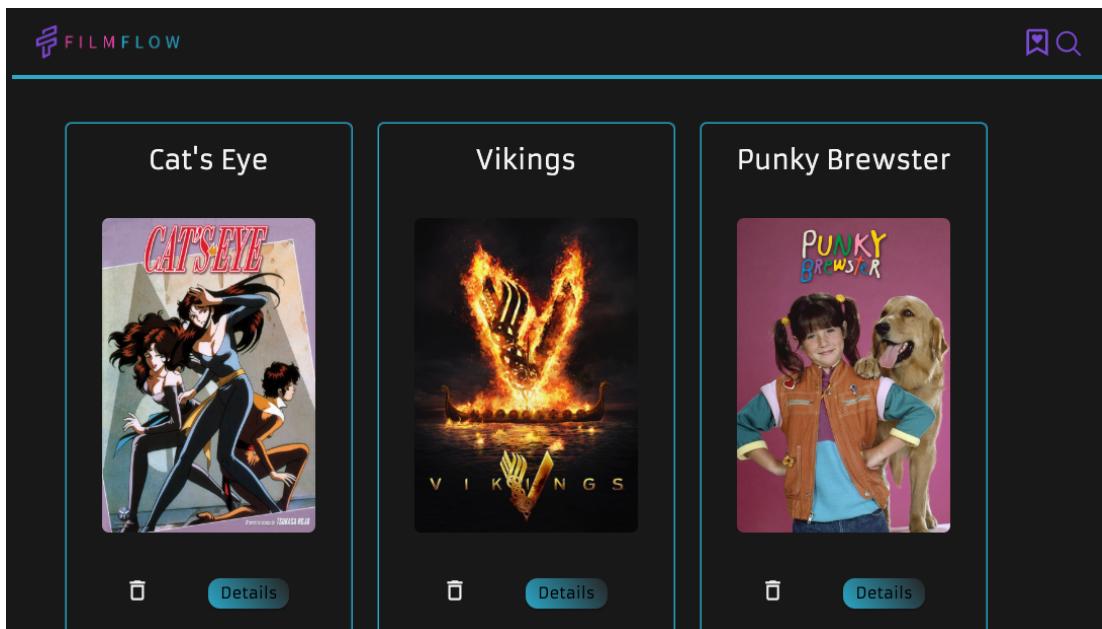
J'affiche une liste de saisons, chacune avec une image, le numéro de saison, le nom de la saison et un résumé. Le résumé est affiché dans un élément "**details**" qui peut être ouvert ou fermé en cliquant sur le nom de la saison. (comme les FAQ bien souvent sur les sites web).

Dans ce composant j'utilise **useState** pour gérer l'état de la saison active, c'est-à-dire la saison dont le résumé est actuellement ouvert. Comme demandé dans le cahier des

DOSSIER PROFESSIONNEL (DP)

charges il s'agit d'un accordéon obtenu avec la fonction `toggleSummary` qui ouvre et ferme les saisons en fonction de l'interaction de l'utilisateur.

O- Favoris.jsx ⇒ DetailsSerie.jsx



Dans ce composant `Favoris.jsx` j'affiche la liste des séries favorites stockées dans le local storage.

Lorsque le composant se charge, il récupère la liste des favoris stockée dans le local storage, puis crée **un tableau de promesses** pour récupérer les données de chaque série favorite en appelant l'API TVMaze. Une fois que toutes les promesses sont résolues, il met à jour le `state` avec les données récupérées, ce qui déclenche le rendu de la liste des favoris.

Pour chaque série, le composant créé :

- l'affichage d'un lien vers la page de détails de la série,
- l'affichage du nom de la série
- l'affichage de l'image de la série (si elle existe)
- l'affichage d'un bouton de suppression pour retirer la série de la liste des favoris.

Le composant utilise également **le hook `useEffect`** pour effectuer une action lors du chargement du composant, et **le hook `useState`** pour stocker la liste des séries favorites.

DOSSIER PROFESSIONNEL (DP)

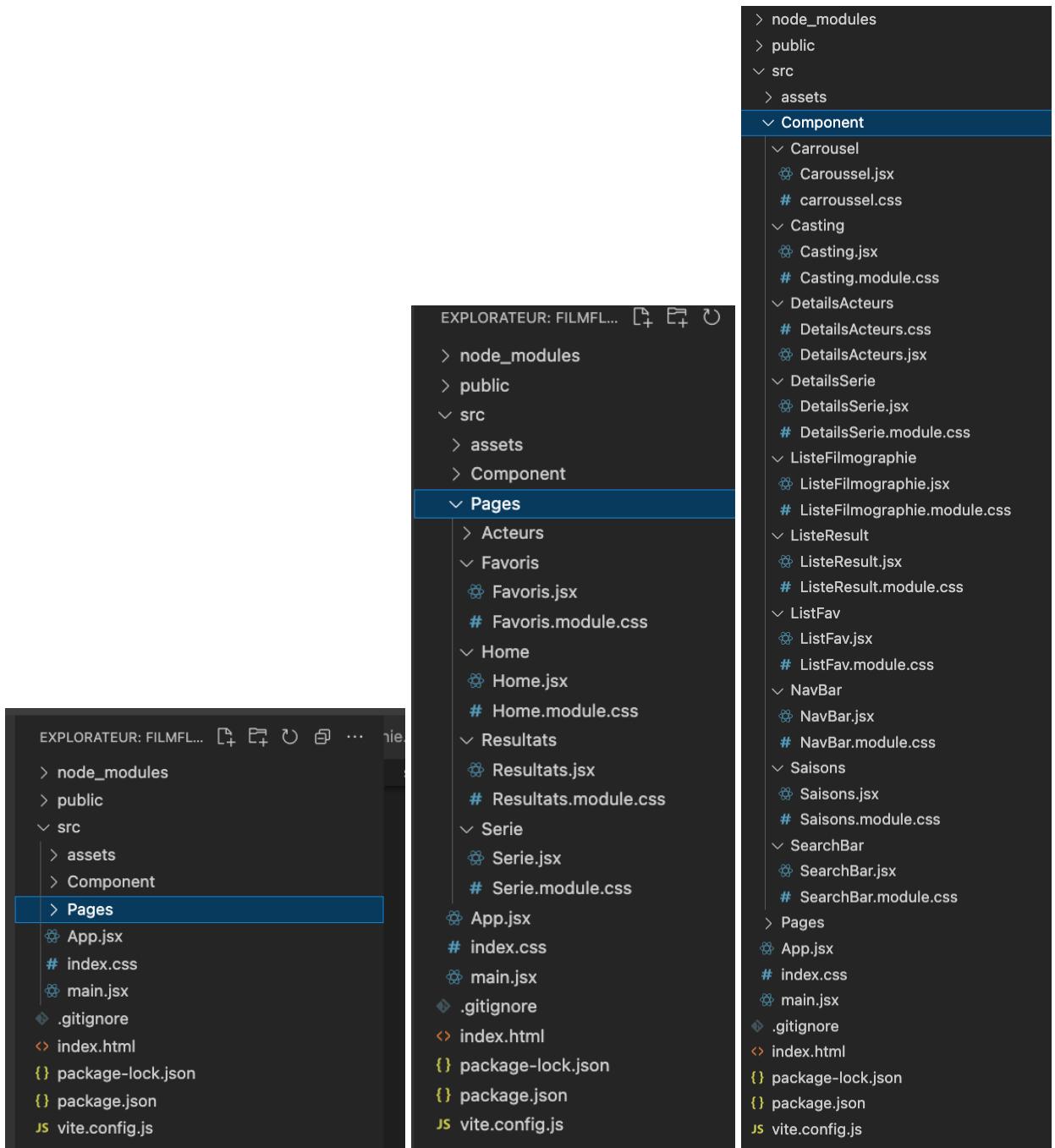
O-ListFav.jsx

J'avais prévu un composant ListFav.jsx car préconisé dans le suivis de projet mais j'ai finalement intégrer ce contenu dans Favoris.jsx

Je vous ai présenté l'ensemble des composants et des pages de mon projet FilmFlow. Cela donne une idée de la logique et du fonctionnement de ce projet.

Vous trouverez ci-dessous le lien de la démo Film Flow sur Youtube
<https://youtu.be/1ZxDmyhl-oQ>

DOSSIER PROFESSIONNEL (DP)



5.COMMENT J'ORGANISE MON CSS DANS CE PROJET REACT

Dans le projet FilmFlow, j'ai utilisé une feuille de style appelée index.css qui englobe les styles qui affectent l'ensemble du projet. Cette feuille de style contient des règles CSS générales qui impactent la typographie, la couleur de fond, les balises H1, le header, le footer, ainsi que des classes réutilisables dans toute l'application. En d'autres termes, il s'agit d'une feuille de style globale.

DOSSIER PROFESSIONNEL (DP)

De plus, j'ai attribué à chaque composant sa propre feuille de style distincte. J'ai placé ces feuilles de style dans le même dossier que les fichiers jsx correspondants. Cette approche permet d'éviter les conflits de noms de classe et offre une meilleure organisation du code.

Je ne vais pas entrer dans les détails du CSS ici, car j'ai déjà beaucoup développé cette partie dans mon rapport de stage. Ce n'est pas l'objectif principal de cette présentation.

2. Précisez les moyens utilisés :

6.LES RESSOURCES QUI M'ONT PERMIS DE CONSTRUIRE CE PROJET :

La bibliothèque REACT

<https://fr.reactjs.org/>

NOS COURS REACT-JS Mis en page par MARVIN MORIN (élève de ma classe) qui a gentillement partagé ses cours mis en forme sur le WEB.

<https://marvin.fronthub.fr/React-Cours-Pierre-Castrec/index.html>

Le tuto APPRENDRE REACT.JS en 1 heure de (Vldev, youtube)

<https://www.youtube.com/watch?v=h2a0cSC1Vz8>

Le tuto du carrousel déjà mentionné précédemment (light code, youtube)

<https://www.youtube.com/watch?v=gGAizkIH-BQ>

7.LES PLUGINS UTILISÉS:

- react-router-dom (bibliothèque qui permet de gérer les routes dans une application React)

DOSSIER PROFESSIONNEL (DP)

- swiper (bibliothèque JavaScript utilisable avec différents framework dont React pour créer des carrousels et des galeries d'images)
- react-icôns (bibliothèque d'icônes pour React)
- react-dom (bibliothèque qui fournit des méthodes pour interagir avec le DOM (Document Object Model) dans React)
- react-toastify (bibliothèque de notifications toast pour React)
- react (bibliothèque principale de React)
- vite (outil de build)
- react-modal (bibliothèque qui permet de créer des modales dans une application React.)

3. Avec qui avez-vous travaillé ?

4. Contexte

Nom de l'entreprise, organisme ou association ▶

IMTS Front end 1ère année.

Période d'exercice ▶ Du : 20/02/23 au : 08/03/23

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

| Intitulé | Autorité ou organisme | Date |
|--------------|----------------------------------|---|
| Cliquez ici. | Cliquez ici pour taper du texte. | Cliquez ici pour sélectionner une date. |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

DOSSIER PROFESSIONNEL (DP)

Déclaration sur l'honneur

Je soussigné(e) Sophie Lambert-conchon ,
déclare sur l'honneur que les renseignements fournis dans ce dossier sont exacts et
que je suis l'auteur(e) des réalisations jointes.

Fait à Saint-Malo le 20/06/2023
pour faire valoir ce que de droit.

Signature :Sophie Lambert-Conchon

DOSSIER PROFESSIONNEL (DP)

Documents illustrant la pratique professionnelle

(facultatif)

Intitulé

Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)