

DOSSIER PROFESSIONNEL (DP)

Nom de naissance ▶ Russell
Nom d'usage ▶ Russell
Prénom ▶ Charlotte
Adresse ▶ 5, Kernanvel, 22160, Carnoët, Côtes-d'Armor

Titre professionnel visé

Développeur web et web mobile

MODALITÉ D'ACCÈS :

- ☒ Parcours de formation
- ☐ Validation des Acquis de l'Expérience (VAE)

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.

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

p. 5

► CP 1 Maquetter une application

p. 5-12

► CP 2 Réaliser une interface utilisateur web statique et adaptable

p. 13-17

► CP 3 Développer une interface utilisateur web dynamique

p. 18-21

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

p. 22

► CP 5 - Créer une base de données

p. 22-24

► CP 6 - Développer les composants d'accès aux données

p. 25-27

► CP 7 - Développer la partie back-end d'une application web ou web mobile

p. 28-30

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

p. 31

Déclaration sur l'honneur

p. 32

Documents illustrant la pratique professionnelle *(facultatif)*

p. 33

Annexes *(Si le RC le prévoit)*

p. 34

EXEMPLES DE PRATIQUE PROFESSIONNELLE

Activité-type 1

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

Exemple n°1 ► CP 1 - Maquetter une application

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

Lors du lancement d'un nouveau site web, nous utilisons des maquettes pour établir la structure initiale du site avant de passer aux étapes de conception et de développement. Nous faisons cela avec des schémas appelés wireframes. Fondamentalement, elles créent une représentation visuelle du site web, permettant au client d'apporter des modifications nécessaires dès le début.

Pendant la formation, nous avons reçu un exercice pour concevoir un wireframe pour un site de recettes. Nous devons créer la page d'accueil et une page de recettes et les deux pages devaient être réactives dans les versions mobile, tablette et desktop.

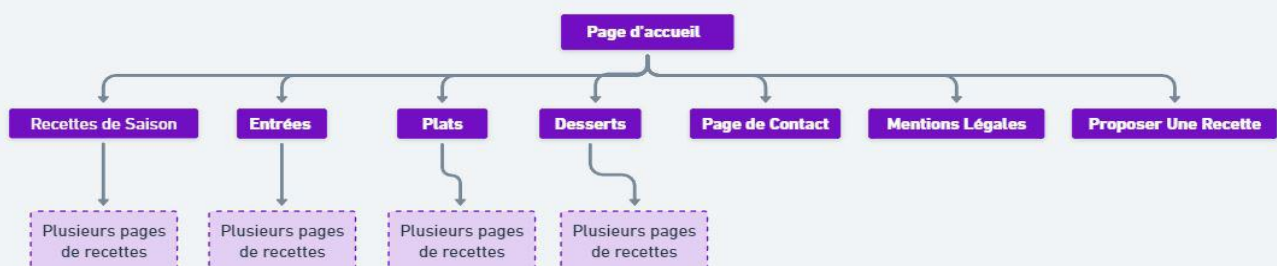
J'ai d'abord commencé par lire entièrement la maquette et analyser le contenu requis. Ensuite, j'ai élaboré des user-stories pour une meilleure compréhension de ce qui a été demandé, et des priorités avant de commencer.

User-Stories

En tant que	Je veux	Afin de (si besoin/nécessaire)
Visiteur	Je veux pouvoir accéder à la liste des ingrédients d'une recette	Afin de réaliser celle-ci dans ma cuisine
Visiteur	Je veux pouvoir cliquer sur le logo pour revenir à la page d'accueil	Afin de naviguer facilement sur la site
Visiteur	Je veux avoir un menu de navigation pour accéder aux pages principales	Afin de naviguer facilement sur la site
Visiteur	Je veux pouvoir lire un édito court qui change hebdomadaire	Afin de découvrir des informations sur les recettes
Visiteur	Je veux pouvoir accéder, via un lien, une recette qui change au rafraîchissement de la page	Afin de découvrir de nouvelles recettes
Visiteur	Je veux voir une liste des 5 recettes les plus récentes	Afin de connaître les dernières recettes ajoutées au site
Visiteur	Je veux voir une liste des 5 recettes les mieux notées	Afin de découvrir les recettes les plus populaires
Visiteur	Je veux voir des liens dans le footer pour accéder aux pages de contact et aux mentions légales	Afin d'accéder à des informations essentielles
Visiteur	Je veux voir un lien dans le footer vers un formulaire pour proposer une recette	Afin de contribuer au contenu du site
Visiteur	Je veux voir la liste des ingrédients séparée du reste de la recette	Afin de voir clairement les ingrédients nécessaires
Visiteur	Je veux voir les étapes de la recette identifiée par un numéro	Afin de suivre facilement la préparation

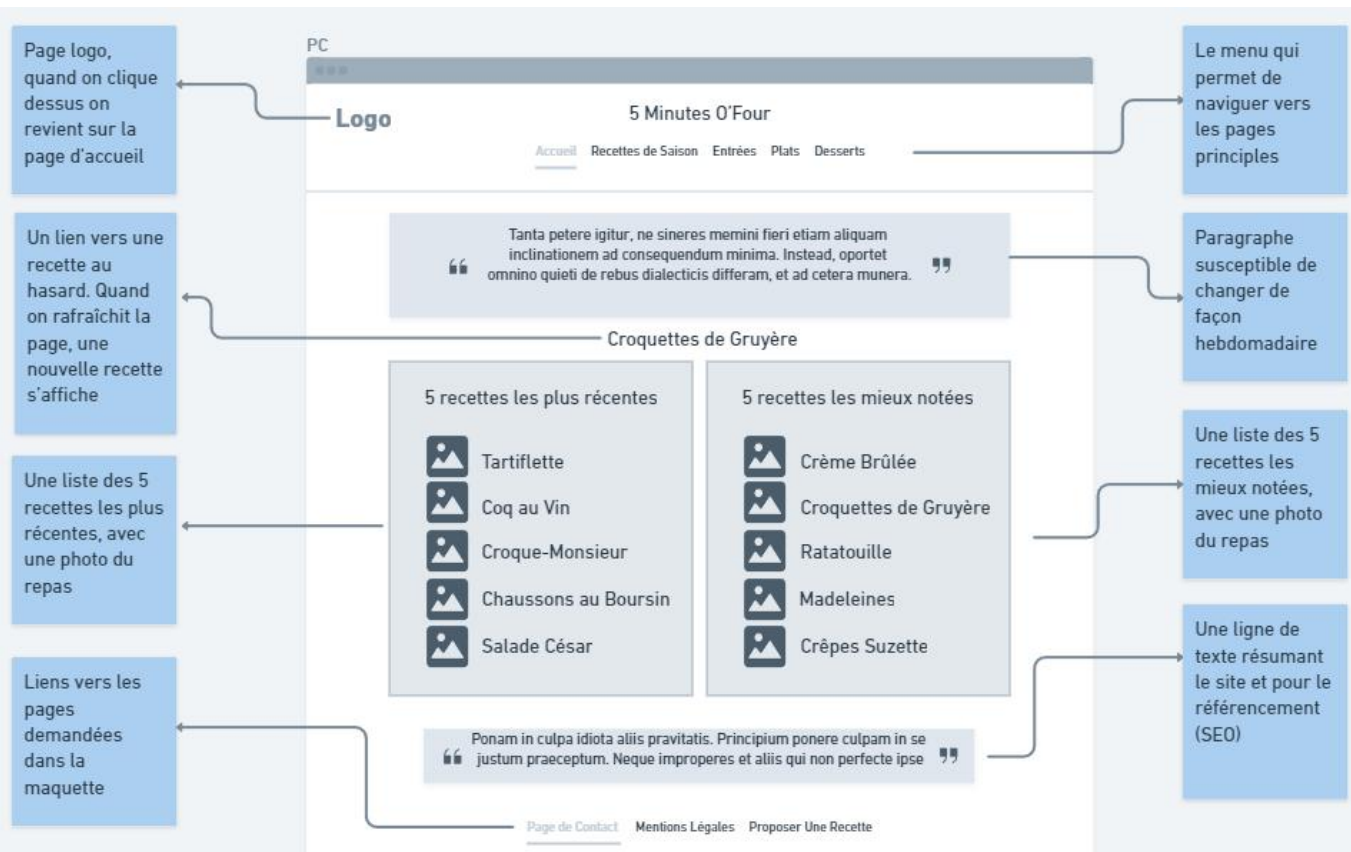
J'ai ensuite utilisé ces informations pour créer une mind map représentant l'arborescence des pages, en identifiant les pages principales. Cette mind map a servi de guide pour élaborer des wireframes.

Arborescence des Pages



Page d'accueil - Mode Desktop

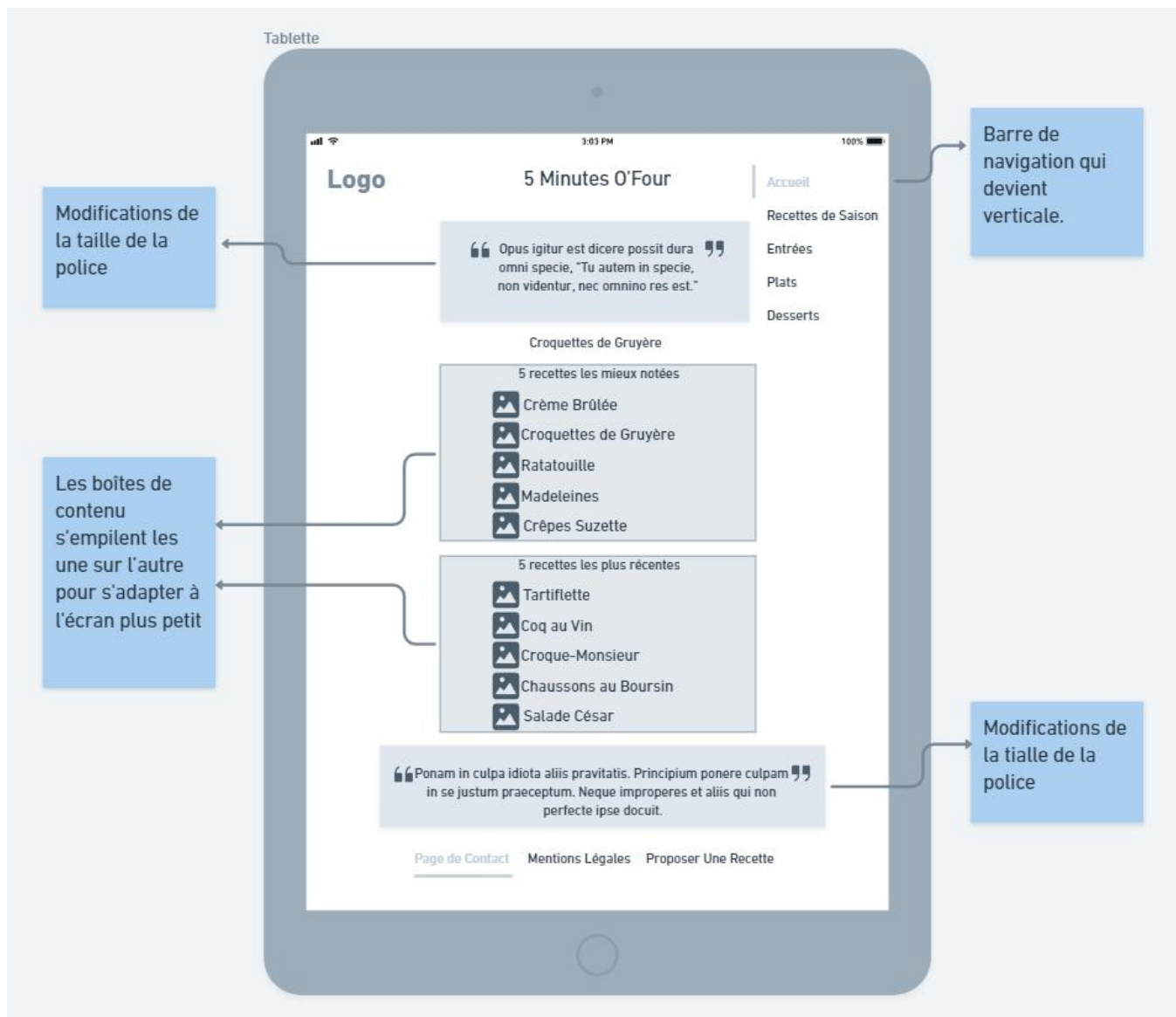
Sur la page d'accueil, j'avais une liste de fonctions attendues s'agissait d'un petit paragraphe qui change chaque semaine, un lien vers une recette aléatoire qui change lorsque la page est actualisée, puis une liste des cinq recettes les plus récentes et des cinq recettes mieux notées.



DOSSIER PROFESSIONNEL (DP)

Page d'accueil - Mode Tablette

En mode tablette, j'ai changé la barre de navigation pour qu'elle soit verticale afin qu'il soit plus facile de sélectionner la page web avec le pouce droit. Pour s'adapter au plus petit écran j'ai modifié la taille de la police et des images. J'ai ensuite empilé les boîtes de contenu. En production, je finirai par compléter cela en utilisant des requêtes multimédias.



DOSSIER PROFESSIONNEL (DP)

Page d'accueil - Mode Mobile

En mode mobile, j'ai à nouveau modifié la taille de la police et des images pour les rendre plus petites. J'ai également changé le menu en un menu déroulant, sur la droite pour permettre une utilisation facile.



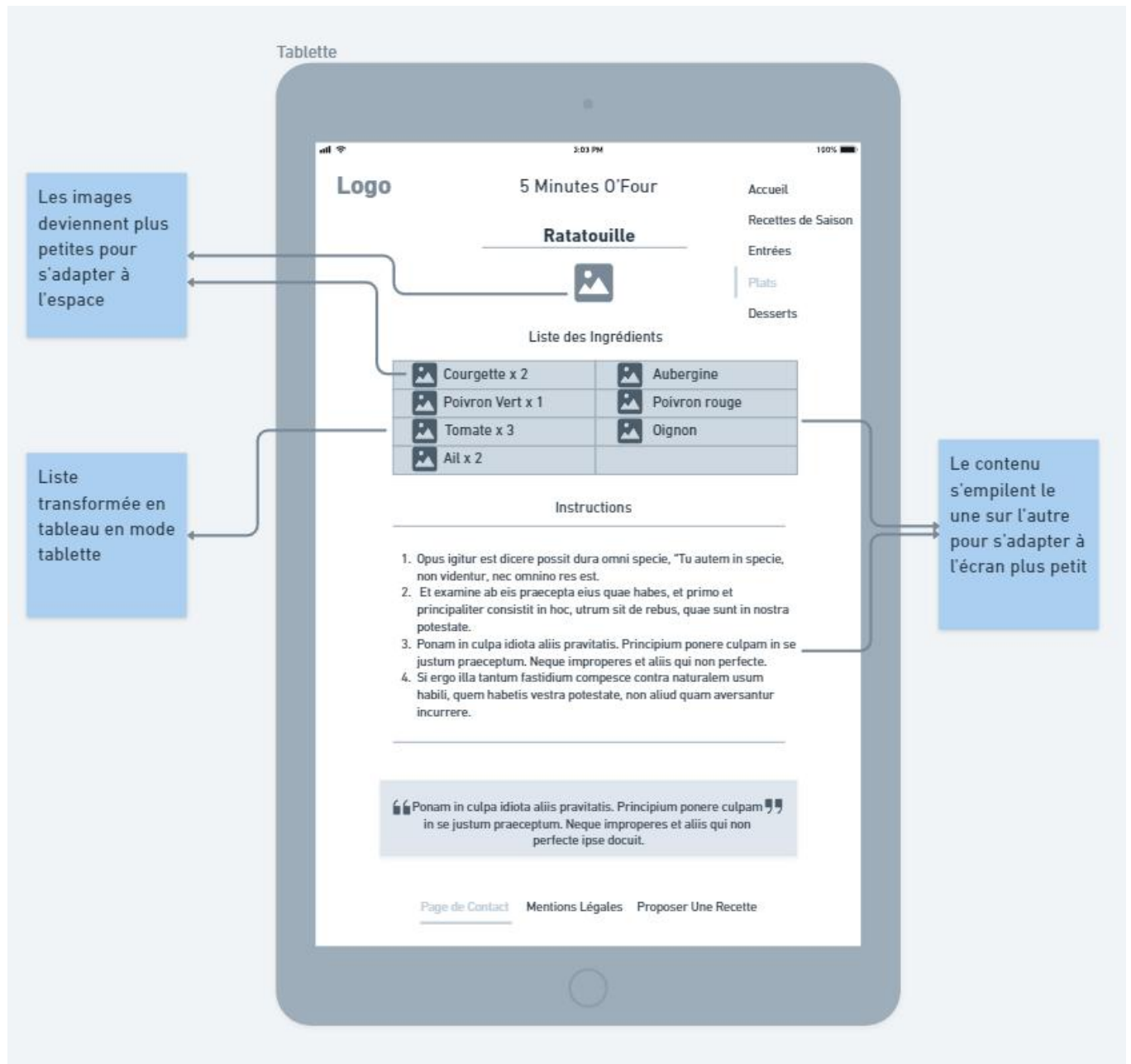
Page de Recette - Mode Desktop

Sur la page des recettes les fonctions attendues étaient une liste d'ingrédients séparée du reste de la recette et la recette en étapes numérotées.



Page de Recette - Mode Tablette

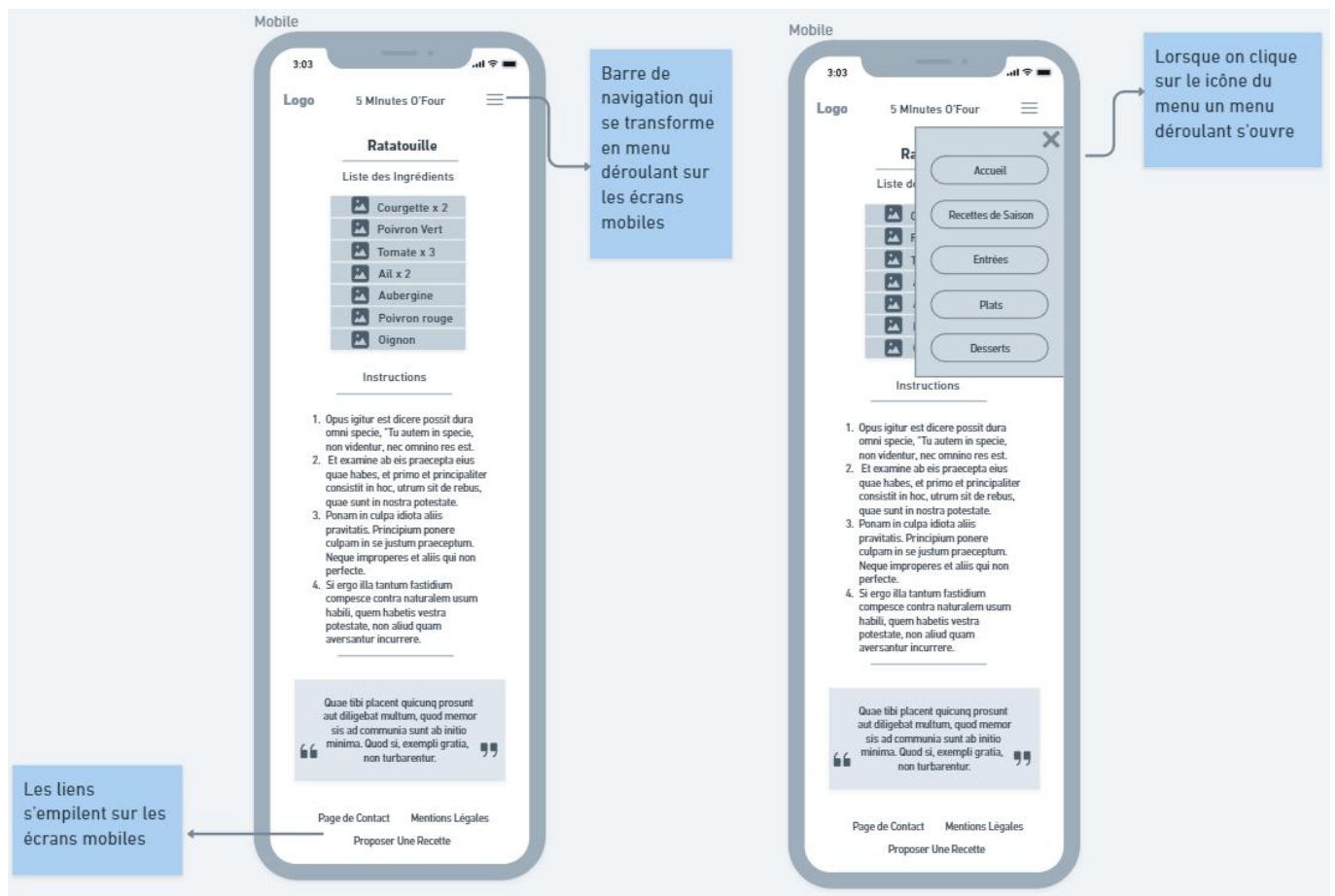
Sur la page de la tablette, j'ai empilé les boîtes de contenu et ajouté un fond gris aux ingrédients pour les rendre plus clairs. J'ai également réduit la taille de la police et la taille de l'image. Comme la page d'accueil, la barre de navigation s'empile sur le côté droit.



DOSSIER PROFESSIONNEL (DP)

Page de Recette - Mode Mobile

En mode mobile, la barre de navigation est à nouveau devenue un menu déroulant. La liste des ingrédients a également été modifiée en une liste d'une seule ligne, avec les ajustements habituels de la taille de l'image et de la taille de la police.



DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Pour cet exercice j'ai utilisé whimsical.com comme outil. Il existe d'autres outils tels que diagrams.net, mockflow.com et wireframe.cc, mais j'ai trouvé whimsical correspondait le mieux à mes besoins. Ses raccourcis clavier et sa grande variété d'outils spécialement adaptés aux wireframes ont rendu le travail beaucoup plus rapide et facile.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul

4. Contexte

Nom de l'entreprise, organisme ou association ► *O'clock*

Chantier, atelier, service ► Projet réalisé lors d'un jour de vielle

Période d'exercice ► Du : *10/08/2023* au : *13/08/2023*

5. Informations complémentaires (facultatif)

<https://whimsical.com/wireframe-MS9h7B1RMNt6R4MkQoSmuw>

Activité-type 1

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

Exemple n° 2 ▶ CP 2 - Réaliser une interface utilisateur web statique et adaptable

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

Pendant la formation, nous avons réalisé un projet de box-model. Ils nous ont donné des maquettes d'images à suivre et les textes dans un fichier. Nous avons terminé les bases (header, contenu de l'article et footer) en classe et le reste était des devoirs.



Liens HTML

```
<link rel="stylesheet" href="../../css/reset.css">
<link rel="stylesheet" href="../../css/styles.css">
<link rel="preconnect" href="https://fonts.bunny.net">
<link href="https://fonts.bunny.net/css?family=oswald:400,500,600" rel="stylesheet" />
```

- Stylesheet reset.css: utilisée pour éviter les problèmes de compatibilité entre navigateurs en réinitialisant les styles par défaut.
- Stylesheet styles.css: le fichier où sont définies les règles de style pour façonner l'apparence du site.
- Preconnect - fonts.bunny.net : utilisé pour améliorer les performances du site en établissant une pré connexion avec le serveur de polices.
- Lien pour la police : rends la police disponible avec des poids spécifiques.

J'ai utilisé une bordure pour les lignes, en ajustant la couleur et la largeur en conséquence.

```
h1 {
  border-top: 2px solid #A02F2F;
  border-bottom: 2px solid #A02F2F;
  width: 95%;
}
.paragraph-intro {
  border-top: 1px solid;
  padding-top: 15px;
}
```

Pour la zone de contenu, j'ai utilisé un div pour structurer et regrouper le contenu. J'ai attribué un classname "container" à ce div pour permettre l'application facile des styles dans le CSS.

```
<div class="container">
```

Les Styles CSS Pour le Div "Container"

```
div.container {
  width: 80%;
  background-color: white;
  margin: 10px auto;
  text-align: center;
  border: 1px solid;
}
```

width: définit la largeur du conteneur lui-même.
margin: 10px en haut et en bas de l'élément. Auto à gauche et à droite

DOSSIER PROFESSIONNEL (DP)

Pour la liste des principaux acteurs j'ai utilisé un ul (unordered List) avec des liens href. Dans les liens les attributs target="_blank" et rel="noopener" sont utilisés. L'attribut target="_blank" est utilisé pour ouvrir le lien dans un nouvel onglet du navigateur et l'attribut rel="noopener" est utilisé pour empêcher que cette nouvelle page accède à la fenêtre parente, renforçant la sécurité de la navigation.

Les Styles CSS Pour La Liste

```
section li {
  color: ■ #8E0606;
  background-color: □ #EEEEEE;
  text-align: center;
  text-transform: uppercase;
  border-radius: 20px;
  padding: 5px;
  margin: 10px auto;
  width: 70%;
}
section li:hover {
  color: □ #EEEEEE;
  background-color: ■ #8E0606;
}

<ul>
<!-- _blank attribute opens the page in a separate window -->
<!-- noopener attribute for added security. It stops the newly opened page from having access to the parent page -->
<li><a href="https://fr.wikipedia.org/wiki/Simon_Astier" target="_blank" rel="noopener">Simon Astier</a></li>
<li><a href="https://fr.wikipedia.org/wiki/Alban_Lenoir" target="_blank" rel="noopener">Alban Lenoir</a></li>
<li><a href="https://fr.wikipedia.org/wiki/S%C3%A9bastien_Lalanne" target="_blank" rel="noopener">Sébastien Lalanne</a></li>
<li><a href="https://fr.wikipedia.org/wiki/G%C3%A9rard_Darier" target="_blank" rel="noopener">Gérard Darier</a></li>
<li><a href="https://fr.wikipedia.org/wiki/Agn%C3%A8s_Boury" target="_blank" rel="noopener">Agnès Boury</a></li>
<li><a href="https://fr.wikipedia.org/wiki/Fran%C3%A7ois_Podetti" target="_blank" rel="noopener">François Podetti</a></li>
<li><a href="https://fr.wikipedia.org/wiki/Arnaud_Joyet" target="_blank" rel="noopener">Arnaud Joyet</a></li>
<li><a href="https://fr.wikipedia.org/wiki/Philippe_No%C3%ABl" target="_blank" rel="noopener">Philippe Noël</a></li>
<li><a href="https://fr.wikipedia.org/wiki/%C3%89tienne_Fague" target="_blank" rel="noopener">Étienne Fague</a></li>
<li><a href="https://fr.wikipedia.org/wiki/Aurore_Pourtteyron" target="_blank" rel="noopener">Aurore Pourteyron</a></li>
</ul>
```

Pour les images du bas, j'ai utilisé flexbox pour aligner les images au centre et dans une colonne.

```
<div class="gallery">
  <article class="gallery picture">
    
    <h3>Prêt à l'action</h3>
    <p>La fine équipe du début de l'aventure</p>
  </article>
  <article class="gallery picture">
    
    <h3>Prêt à l'action</h3>
    <p>La fine équipe du début de l'aventure</p>
  </article>
  <article class="gallery picture">
    
    <h3>Prêt à l'action</h3>
    <p>La fine équipe du début de l'aventure</p>
  </article>
</div>

.gallery {
  display: flex;
  justify-content: center;
  flex-direction: column;
  align-items: center;
  padding: 0;
}
.picture {
  border: 2px solid □ #EEEEEE;
  margin-bottom: 15px;
  background-color: □ white;
}
.img-gallery {
  height: 110px;
}
```

Display: flex: est utilisée pour activer le modèle de disposition flexbox, ce qui facilite le positionnement dynamique des éléments en utilisant des propriétés flexbox box. Dans cet exemple, j'ai appliqué justify-content: center, qui positionne les éléments "gallery" au centre, flex-direction: column, qui organise les éléments dans une disposition en colonne et align-items: center, qui aligne les éléments au centre sur l'axe transversal.

Responsivité

J'ai ajouté "name="viewport" content="width=device-width, initial-scale=1" à la balise méta dans le code HTML. Cela permet de contrôler la façon dont la page est affichée sur les appareils mobiles et tablettes et garantit la réactivité du site en indiquant au navigateur de prendre en compte la largeur de l'appareil et d'appliquer une première échelle. `<meta name="viewport" content="width=device-width, initial-scale=1">`

Ensuite, j'ai incorporé des media queries afin de rendre le site réactif sur mobile et tablette. Ces règles spécifiques me permettent d'ajuster les styles et la mise en page du site en fonction de la taille de l'écran.

Media Queries Tablette

```
/* Tablet */
@media (max-width: 768px) {
  div.container {
    width: 80%;
  }
  h1 {
    font-size: 40px;
  }
  h2 {
    font-size: 24px;
  }
  h3 {
    font-size: 20px;
  }
  article {
    max-width: 90%;
  }
}
```

Ces media queries ciblent des écrans ayant une largeur maximale de 768 pixels. Elles modifient la largeur du conteneur div avec la classe 'container', restreignent la largeur de l'article, et ajustent la taille de la police.

Media Queries Mobile

```
/* Mobile */
@media (max-width: 480px) {
  div.container {
    width: 90%;
  }
  header img {
    height: 115px;
  }
  h1 {
    font-size: 30px;
  }
  h2 {
    font-size: 20px;
  }
  h3 {
    font-size: 18px;
  }
  article {
    max-width: 100%;
  }
  main {
    margin: 15px;
  }
  footer {
    padding: 20px 0;
  }
  section li {
    font-size: 12px;
  }
  .img-gallery {
    height: 80px;
  }
}
```

Ces medias queries ciblent des écrans avec une largeur maximale de 480 pixels. En raison de l'espace plus restreint sur ces écrans, plusieurs ajustements supplémentaires sont nécessaires. Cette fois, j'ai agrandi le conteneur pour éviter une compression excessive du contenu, modifié la hauteur de l'image de le header pour éviter qu'elle ne soit trop grande, ajusté les tailles de police et élargi la largeur de l'article pour occuper tout l'écran.

DOSSIER PROFESSIONNEL (DP)

← Résultat final sur tablette

Résultat final sur mobile →



Hero Corp La série

Présentation



Hero Corp est une série télévisée française fantastique et humoristique, produite par CALT, créée par Simon Astier et Alban Lenoir, réalisée par Simon Astier et écrite par Arnaud Joyet et Simon Astier.

Les deux premières saisons ont été diffusées du 25 octobre 2008 au 29 janvier 2010 sur Comédie+ et rediffusées sur France 4 et Game One. La troisième saison a été diffusée sur France 4 du 21 octobre 2013 au 6 décembre 2013. La quatrième saison a été, quant à elle, diffusée du 19 décembre 2014 au 23 janvier 2015 sur France 4 et à partir du 9 février 2015 sur Game One. Une cinquième et dernière saison a été diffusée sur France 4 du 24 mai 2017 au 7 juin 2017.

Synopsis

À la suite d'une guerre dans les années 1980, il est décidé de créer l'agence Hero Corp, une organisation regroupant tous les super-héros afin de maintenir un climat de paix. Cette agence possède plusieurs sites secrets sur la planète et dans le département de la Lozère se trouvent les retraités, mis au rancart, démissionnaires, démasqués, pas-formés. Coupés du monde, ils peuvent retrouver une vie calme et paisible. Vingt ans après, ce calme paisible vole en éclats lorsque réapparaît The Lord.

Face au retour de The Lord, le plus grand super-villain de l'Histoire, le village est démun. Selon une prédiction, John est l'unique solution face à ce danger que Hero Corp préfère garder sous silence.

John arrive dans un village isolé pour aller enterrer sa tante. Il se rend compte que les habitants cachent quelque chose et qu'ils n'ont pas l'air décidés à le laisser partir.

Principaux acteurs

SIMON ASTIER

ALBAN LENOIR

SÉBASTIEN LALANNE

GERARD DARIER

AGNÈS BOURY

FRANÇOIS PODETTI

ARNAUD JOYET

PHILIPPE NOËL

ÉTIENNE FAGUE

AUORE POURTEYRON

Quelques images de la série



Prêt à l'action

La fine équipe du début de l'aventure



Prêt à l'action

La fine équipe du début de l'aventure



Prêt à l'action

La fine équipe du début de l'aventure

Présentation de Hero Corp, la série © Hero Corp



Hero Corp La série

Présentation



Hero Corp est une série télévisée française fantastique et humoristique, produite par CALT, créée par Simon Astier et Alban Lenoir, réalisée par Simon Astier et écrite par Arnaud Joyet et Simon Astier.

Les deux premières saisons ont été diffusées du 25 octobre 2008 au 29 janvier 2010 sur Comédie+ et rediffusées sur France 4 et Game One. La troisième saison a été diffusée sur France 4 du 21 octobre 2013 au 6 décembre 2013. La quatrième saison a été, quant à elle, diffusée du 19 décembre 2014 au 23 janvier 2015 sur France 4 et à partir du 9 février 2015 sur Game One. Une cinquième et dernière saison a été diffusée sur France 4 du 24 mai 2017 au 7 juin 2017.

Synopsis

À la suite d'une guerre dans les années 1980, il est décidé de créer l'agence Hero Corp, une organisation regroupant tous les super-héros afin de maintenir un climat de paix. Cette agence possède plusieurs sites secrets sur la planète et dans le département de la Lozère se trouvent les retraités, mis au rancart, démissionnaires, démasqués, pas-formés. Coupés du monde, ils peuvent retrouver une vie calme et paisible. Vingt ans après, ce calme paisible vole en éclats lorsque réapparaît The Lord.

Face au retour de The Lord, le plus grand super-villain de l'Histoire, le village est démun. Selon une prédiction, John est l'unique solution face à ce danger que Hero Corp préfère garder sous silence.

John arrive dans un village isolé pour aller enterrer sa tante. Il se rend compte que les habitants cachent quelque chose et qu'ils n'ont pas l'air décidés à le laisser partir.

Principaux acteurs

SIMON ASTIER

ALBAN LENOIR

SÉBASTIEN LALANNE

GERARD DARIER

AGNÈS BOURY

FRANÇOIS PODETTI

ARNAUD JOYET

PHILIPPE NOËL

ÉTIENNE FAGUE

AUORE POURTEYRON

Quelques images de la série



Prêt à l'action

La fine équipe du début de l'aventure



Prêt à l'action

La fine équipe du début de l'aventure



Prêt à l'action

La fine équipe du début de l'aventure

Présentation de Hero Corp, la série © Hero Corp

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

Pour cet exercice j'ai utilisé vscode et devtools dans google chrome.

3. Avec qui avez-vous travaillé ?

J'ai travaillé en classe, en suivant le formateur pour une petite partie de ce projet, le reste a été réalisé par moi-même, seule. Après la fin de la formation, j'ai ajouté des media-queries, pour rendre le site responsive sur mobile et tablette.

4. Contexte

Nom de l'entreprise, organisme ou association ► *O'clock*

Chantier, atelier, service ► Projet réalisé pendant saison 1 de la formation.

Période d'exercice ► Du : *04/05/2023* au : *13/11/23*

5. Informations complémentaires (facultatif)

<https://github.com/Charlirussell/Box-Model>

Activité-type 1

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

Exemple n° 3 ► CP 3 - Développer une interface utilisateur web dynamique

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

Pendant la formation, nous avons reçu la tâche de créer un soundboard en utilisant JavaScript. On nous a fourni une liste d'instructions ainsi que des fichiers HTML et CSS, ainsi que deux fichiers JavaScript avec des commentaires TODO indiquant les étapes à suivre.

J'ai d'abord assuré de la bonne initialisation de l'app en ajoutant `document.addEventListener("DOMContentLoaded", app.init);`. Cela garantit que la fonction "init" sera appelée qu'une fois le DOM est prêt et quand la structure du document HTML sera entièrement chargée.

Ensuite, j'ai ajouté les `querySelector` pour associer chaque propriété de "soundboard" à son élément correspondant.

```
/** Method for selecting the different pad elements in the DOM */
definePads: function() {
  // TODO 2
  // Query selectors that link the properties to their elements
  soundboard.themeElement = document.querySelector('.pad-1');
  soundboard.leviosaElement = document.querySelector('.pad-2');
  soundboard.dobyElement = document.querySelector('.pad-3');
  soundboard.griffindorElement = document.querySelector('.pad-4');
  soundboard.trollElement = document.querySelector('.pad-5');
  soundboard.silenceElement = document.querySelector('.pad-6');
},
```

Il s'agissait ensuite de créer les "eventListeners" pour chaque élément "pad". Lorsqu'un "pad" est cliqué, le event listener exécute le "click handler" approprié.

```
<main class="pads">
  <div class="pad-1">&#9834; Hedwig's Theme &#9834;</div>
  <div class="pad-2">&#9834; Leviosa &#9834;</div>
  <div class="pad-3">&#9834; Doby &#9834;</div>
  <div class="pad-4">&#9834; Griffindor &#9834;</div>
  <div class="pad-5">&#9834; Troll &#9834;</div>
  <div class="pad-6">&#9834; Silence &#9834;</div>
</main>
```

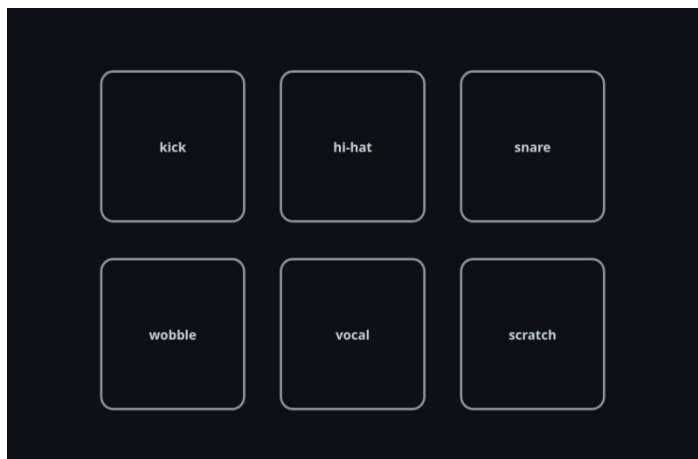
```
/* Method allowing you to add an event listener per pad */
attachEvents: function() {
  // TODO 3
  // Event listeners for each element
  soundboard.themeElement.addEventListener('click', soundboard.handleThemeClick);
  soundboard.leviosaElement.addEventListener('click', soundboard.handleLeviosaClick);
  soundboard.dobyElement.addEventListener('click', soundboard.handleDobyClick);
  soundboard.griffindorElement.addEventListener('click', soundboard.handleGriffindorClick);
  soundboard.trollElement.addEventListener('click', soundboard.handleTrollClick);
  soundboard.silenceElement.addEventListener('click', soundboard.handleSilenceClick);
  //Global event listener for the keyboard function
  document.addEventListener('keydown', soundboard.handleKeyboard);
},
```

La dernière étape consistait à créer un gestionnaire qui contrôle les pads depuis le clavier. Pour ce faire, j'ai ajouté la fonction "handleKeyboard". Ensuite, dans la méthode "attachEvents", j'ai créé un écouteur d'événement global qui écoute l'événement "keydown" et exécute la méthode `soundboard.handleKeyboard`.

```
// Handle executed when any key is pressed
handleKeyboard: function(event) {
  //if (event.key === 'a') { console.log('a key')}
  if (event.key === 'q') { soundboard.themeElement.click(); }
  if (event.key === 'w') { soundboard.leviosaElement.click(); }
  if (event.key === 'e') { soundboard.dobyElement.click(); }
  if (event.key === 'r') { soundboard.griffindorElement.click(); }
  if (event.key === 't') { soundboard.trollElement.click(); }
  if (event.key === 'y') { soundboard.silenceElement.click(); }
},
```

DOSSIER PROFESSIONNEL (DP)

Après, j'ai décidé de personnaliser le soundboard, en créant un thème Harry Potter.
Le style CSS original donné par o'clock ressemblait à ceci.

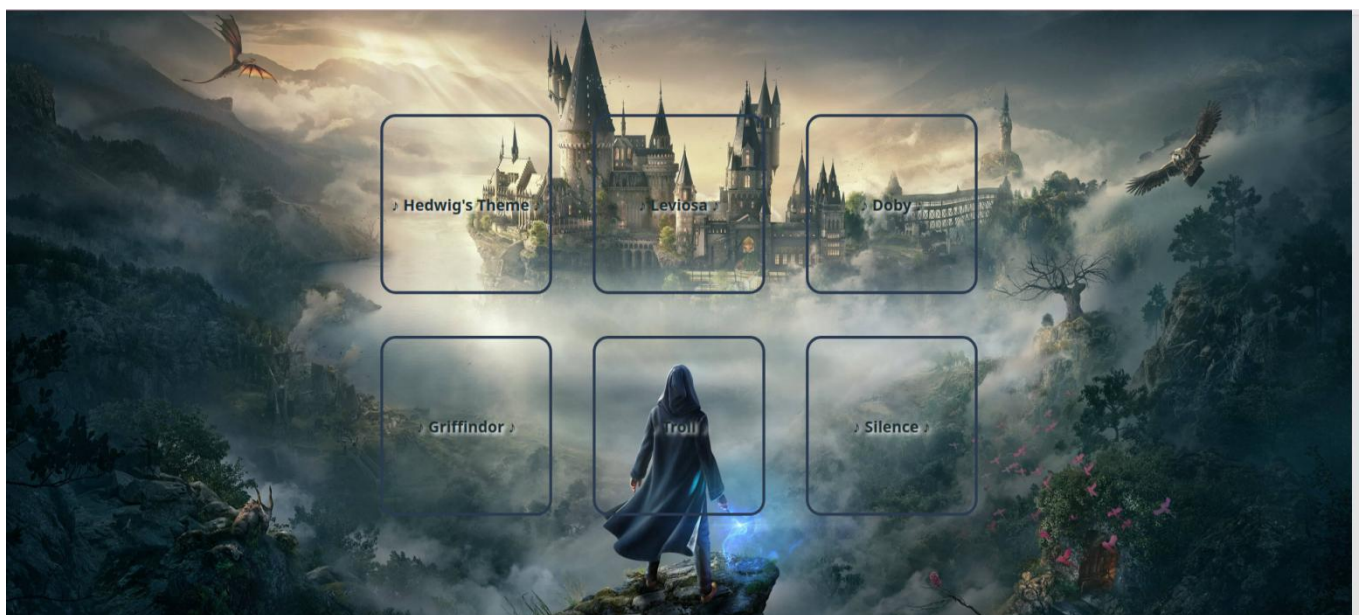


← Au survol, les pads ressemblaient à ceci.

Et quand on clique dessus, ils ressemblent à ça. →



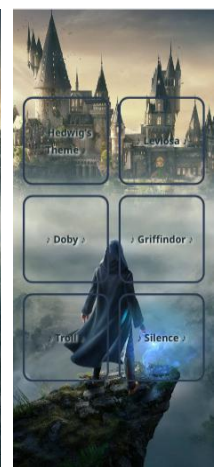
Ceci est la version finale de mon soundboard.



Le pad en survol et le pad lorsqu'on clique dessus.



Le soundboard est également réactif sur mobile et tablette.



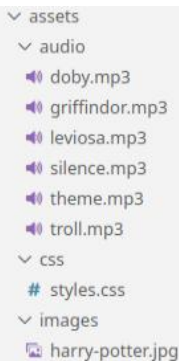
DOSSIER PROFESSIONNEL (DP)

Pour le CSS, j'ai modifié les couleurs pour celles qui semblaient aller le mieux. Il faudrait encore un peu de réglage, car je trouve qu'ils sont un peu trop sombres.

Je choisis ensuite une image libre de droits et l'ajoute dans un dossier images, que j'ai ajouté au dossier "assets", en le liant dans le CSS balise "body" avec une URL vers son dossier de destination. Background-attachment: fixed arrête le défilement de l'image et background-position: center la maintient fixe au centre. Ces règles maintiennent l'image fixe pour toutes les tailles d'écran.

```
body {  
  color-scheme: dark;  
  background-image: url('../images/harry-potter.jpg');  
  background-size: cover;  
  background-attachment: fixed;  
  background-position: center;  
}
```

Pour l'audio, j'ai trouvé six clips audio de Harry Potter libres de droits et je les ai ajoutés dans un dossier audio encore dans le dossier des "assets".



Après, j'ai modifié les liens dans la fonction "preloadSamples" pour créer un lien correct vers les nouveaux clips audio.

```
/* Method to initialize and preload the different audio samples to play */  
preloadSamples: function() {  
  soundboard.themeAudio = new Audio('assets/audio/theme.mp3');  
  soundboard.leviosaAudio = new Audio('assets/audio/leviosa.mp3');  
  soundboard.dobyAudio = new Audio('assets/audio/doby.mp3');  
  soundboard.griffindorAudio = new Audio('assets/audio/griffindor.mp3');  
  soundboard.trollAudio = new Audio('assets/audio/troll.mp3');  
  soundboard.silenceAudio = new Audio('assets/audio/silence.mp3');  
},
```

```
.pads > div {  
  border: .2rem solid #2b3b51;  
  color: #1c2e2f;  
  text-shadow: 3px 3px 5px rgba(255, 255, 255, 1);  
  border-radius: 1rem;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  cursor: pointer;  
  aspect-ratio: 1;  
  font-weight: 600;  
  user-select: none;  
}  
  
.pads > div:hover {  
  color: #45b1f8;  
  text-shadow: 3px 3px 5px rgb(15, 10, 32);  
  border-color: #C0C0C0;  
}  
  
.pads > div:active {  
  border-color: transparent;  
  background-color: #9dddbb;  
  color: #1c2e2f;  
  text-shadow: 3px 3px 5px rgba(255, 255, 255, 1);  
  box-shadow: 0 0 2rem 0px #92dbfc;  
}
```

2. Précisez les moyens utilisés :

J'ai utilisé vscode pour ce projet. J'ai également utilisé myinstants.com pour les clips audio libres de droits et wallpapers.com pour l'image.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul sur ce projet

DOSSIER PROFESSIONNEL (DP)

4. Contexte

Nom de l'entreprise, organisme ou association ► *O'clock*

Chantier, atelier, service ► Saison 7 de la formation

Période d'exercice ► Du : 25/07/2023 au : 06/08/2023

5. Informations complémentaires (facultatif)

<https://github.com/Charlirussell/Soundboard>

Activité-type 2

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

Exemple n° 1 CP 5 - Créer une base de données

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

Une base de données sert à stocker, organiser et gérer des données. Pendant ma formation j'ai utilisé Adminer pour créer et gérer des bases de données. J'ai également exécuté des requêtes SQL pour la création de tables, l'insertion, récupération, modification et la suppression des données.

La première étape de la création d'une base de données est Le Modèle Conceptuel de Données (MCD). C'est la première étape car il permet de visualiser et de définir la structure générale des données. Le MCD est un schéma qui sert à organiser notre BDD pour éviter des erreurs, redondances et illogismes. Dans un MCD il y a des entités, propriétés et les associations. Nous définissons les relations entre deux entités en utilisant les cardinalités. Les cardinalités servent à déterminer le nombre d'occurrences qu'une entité peut avoir dans une relation avec une autre entité. Les types différents des cardinalités sont :

One-to-One: Chaque élément d'une entité peut être associé à un seul élément dans une autre entité, et vice versa.

One-to-Many: Un élément d'une entité peut être associé à plusieurs éléments dans une autre entité, mais chaque élément de cette autre entité est lié à un seul élément dans la première entité.

Many-to-Many: Chaque élément dans une entité peut être associé à plusieurs éléments dans une autre entité, et vice versa. Cette relation crée une table intermédiaire pour gérer les correspondances multiples.

Nous créons le MCD en utilisant Mocodo.

Ensuite, nous créons le dictionnaire de données (DDD). Il permet de définir les caractéristiques spécifiques de chaque champ, comme le type, les spécificités et la description. Cela permet de traduire les données en SQL ce qui nous permet de générer la base de données dans adminer. Pour chaque champ défini dans le dictionnaire, nous créons les requêtes SQL correspondantes pour définir la structure de la table dans la base de données.

Une fois que le DDD est complet et traduit en instructions SQL, nous pouvons exécuter ces requêtes SQL dans Adminer, qui génère les tables avec les structures définies. Une fois cette étape achevée, le Modèle Logique de Données (MLD) peut être visualisé dans la section "Database schema" de Adminer. Le MLD est formé en utilisant des clés étrangères pour établir des liens entre les différentes tables de la base de données.

DOSSIER PROFESSIONNEL (DP)

Les commandes MySQL pour créer une base de données appelée "videogame". Avec deux tables nommées "platform" and "videogame".

```
DROP TABLE IF EXISTS `videogame`.`videogame` ;

CREATE TABLE IF NOT EXISTS `videogame`.`videogame` (
  `id` INT UNSIGNED NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(64) NOT NULL,
  `editor` VARCHAR(64) NULL,
  `release_date` DATE NULL,
  `platform_id` INT UNSIGNED NOT NULL,
  PRIMARY KEY (`id`),
  INDEX `fk_videogame_platform_idx` (`platform_id` ASC))
ENGINE = InnoDB;

SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;

-- Data for table `videogame`.`platform`
-----|
START TRANSACTION;
USE `videogame`;
INSERT INTO `videogame`.`platform` (`id`, `name`) VALUES (1, 'PC');
INSERT INTO `videogame`.`platform` (`id`, `name`) VALUES (2, 'MegaDrive');
INSERT INTO `videogame`.`platform` (`id`, `name`) VALUES (3, 'SNES');
INSERT INTO `videogame`.`platform` (`id`, `name`) VALUES (4, 'PlayStation');

COMMIT;

-- Data for table `videogame`.`videogame`
-----
START TRANSACTION;
USE `videogame`;
INSERT INTO `videogame`.`videogame` (`id`, `name`, `editor`, `release_date`, `platform_id`) VALUES (1, 'Theme Hospital', 'Bullfrog', '1997-09-10', 1);
INSERT INTO `videogame`.`videogame` (`id`, `name`, `editor`, `release_date`, `platform_id`) VALUES (2, 'NBA Jam', 'Midway', '1993-04-02', 2);
INSERT INTO `videogame`.`videogame` (`id`, `name`, `editor`, `release_date`, `platform_id`) VALUES (3, 'Day of Tentacle', 'LucasArts', '1993-06-20', 1);
INSERT INTO `videogame`.`videogame` (`id`, `name`, `editor`, `release_date`, `platform_id`) VALUES (4, 'Castlevania', 'Konami', '1986-04-06', 3);

COMMIT;
```



Le MLD de cette table.

DOSSIER PROFESSIONNEL (DP)

La base de données du videogame vue dans adminer.

■	Table	Engine [?]	Collation [?]	Data Length [?]	Index Length [?]	Data Free [?]	Auto Increment [?]	Rows [?]	Comment [?]
<input type="checkbox"/>	platform	InnoDB	utf8mb4_general_ci	16,384	0	0	5	~ 4	
<input type="checkbox"/>	videogame	InnoDB	utf8mb4_general_ci	16,384	16,384	0	5	~ 4	
	2 in total	InnoDB	utf8mb4_general_ci	32,768	16,384	0			

■	Modify	Id	name	■	Modify	Id	name	editor	release_date	platform_Id
<input type="checkbox"/>	edit	1	PC	<input type="checkbox"/>	edit	1	Theme Hospital	Bullfrog	1997-09-10	1
<input type="checkbox"/>	edit	2	MegaDrive	<input type="checkbox"/>	edit	2	NBA Jam	Midway	1993-04-02	2
<input type="checkbox"/>	edit	3	SNES	<input type="checkbox"/>	edit	3	Day of Tentacle	LucasArts	1993-06-20	1
<input type="checkbox"/>	edit	4	PlayStation	<input type="checkbox"/>	edit	4	Castlevania	Konami	1986-04-06	3

2. Précisez les moyens utilisés :

Mocodo pour créer le MCD. Le langage mySQL et adminer pour créer la base de données.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul.

4. Contexte

Nom de l'entreprise, organisme ou association ► *O'clock*

Chantier, atelier, service ► Projet réalisé pendant le formation.

Période d'exercice ► Du : 21/06/23 au : 22/06/23

5. Informations complémentaires (facultatif)

Activité-type 2

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

Exemple n° 2 CP 6 - Développer les composants d'accès aux données

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

Le développement des composants d'accès aux données consiste à créer des modules logiciels qui facilitent la communication avec une base de données.

Il faut d'abord se connecter à la base de données. Nous faisons cela en créant deux fichiers appelés config.ini.dist et config.ini dans le dossier app.

```
DB_HOST=
DB_NAME=
DB_USERNAME=
DB_PASSWORD=
```

config.ini.dist est utilisé pour fournir des exemples de configuration et des valeurs par défaut.

```
DB_HOST=localhost
DB_NAME=oshop
DB_USERNAME=oshop
DB_PASSWORD=oshop
```

Nous copions et collons les données de config.ini.dist dans notre fichier config.ini. Après, on ajoute les informations de la base de données, en faisant attention à ne pas utiliser d'espaces

```
app > .htaccess
1 deny from all
```

Ensuite, nous ajoutons un fichier appelé .htaccess aussi dans le dossier app. À l'intérieur de ce dossier, nous écrivons simplement "deny from all". Cela restreindra l'accès à tout contenu présent dans ce répertoire via le serveur web et protégera les informations de la base de données contre la visibilité.

Ensuite nous créons une classe Database pour gérer la connexion à la base de données. Il récupère les données de config.ini, analyse le fichier et renvoie un tableau associatif.

```
class Database
{
    /** @var PDO */
    private $dbh;
    private static $_instance;
    private function __construct()
    {
        // Retrieve the data from the file config.ini
        // The function parse_ini_file parses the file and returns an associative array
        $configData = parse_ini_file(__DIR__ . '/../config.ini');

        try {
            $this->dbh = new PDO(
                "mysql:host={$configData['DB_HOST']};dbname={$configData['DB_NAME']};charset=utf8",
                $configData['DB_USERNAME'],
                $configData['DB_PASSWORD'],
                array(PDO::ATTR_ERRMODE => PDO::ERRMODE_WARNING) // Show SQL errors on screen
            );
        } catch (\Exception $exception) {
            echo 'Erreur de connexion...<br>';
            echo $exception->getMessage() . '<br>';
            echo '<pre>';
            echo $exception->getTraceAsString();
            echo '</pre>';
            exit;
        }
    }
    // the unique method you need to use
    public static function getPDO()
    {
        // If no instance => create one
        if (empty(self::$_instance)) {
            self::$_instance = new Database();
        }
        return self::$_instance->dbh;
    }
}
```

DOSSIER PROFESSIONNEL (DP)

```
public function productsByCategoryId($idCategory)
{
    $sql = "SELECT * FROM `product` WHERE `category_id` = ".$idCategory;
    // version with Joiner
    /*
    $sql = "SELECT `product`.*, `category`.`name` AS `category_name`
    FROM `product`
    INNER JOIN `category` ON `product`.`category_id` = `category`.`id`
    WHERE `category_id`=".$idCategory;
    */
    /*
    Product {#14 ▼
      -description: "Proinde concepta rabie saeviore, quam desperatio inc
      -picture: "assets/images/produits/1-kiss.jpg"
      -price: "40.00"
      -rate: "4"
      -status: "1"
      -brand_id: "2"
      -category_id: "1"
      -type_id: "7"
      #id: "1"
      #name: "Kissing"
      #created_at: "2018-10-17 11:00:00"
      #updated_at: null
      +"category_name": "Détente"
    }
    */

    // TODO : créer PDO
    /** @var PDO $pdo */
    $pdo = Database::getPDO();

    // TODO : query la requete
    $pdoStatement = $pdo->query($sql);
    // ? est ce que je récupère 1 résultat ou plusieurs ?
    $results = $pdoStatement->fetchAll(PDO::FETCH_CLASS, Product::class);
    // dump($results);

    // TODO renvoyer les résultats
    return $results;
}
```

Nous pouvons désormais récupérer des informations de la base de données à l'aide de fonctions. Nous utilisons une requête SQL pour récupérer les données requises. PDO est utilisé pour exécuter des requêtes SQL.

Nous pouvons utiliser des jointures comme "INNER JOIN" pour récupérer des informations supplémentaires entre les tables, telles que le nom de la catégorie associée à chaque produit.

2. Précisez les moyens utilisés :

c et adminer pour ce projet.

3. Avec qui avez-vous travaillé ?

J'ai travaillé seul.

4. Contexte

Nom de l'entreprise, organisme ou association ► *O'clock*

Chantier, atelier, service ► Projet réalisé pendant le formation.

Période d'exercice ► Du : 07/07/23 au : 19/07/23

5. Informations complémentaires (facultatif)

Activité-type 2

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

Exemple n° 3 CP 7 - Développer la partie back-end d'une application web ou web mobile

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

Pendant le projet O'shop, nous avons utilisé altorouter pour listant toutes les routes et endpoints de l'application. Dans le index.php nous créons les routes.

```
$router->map(  
    // 1. la méthode HTTP qui est requise  
    // il existe plusieurs méthode HTTP : GET, POST, PATCH, PUT, DELETE (OPTIONS, HEAD ...)  
    // ? https://developer.mozilla.org/fr/docs/Web/HTTP/Methods  
    'GET',  
    // 2. la route : /, /home, /categories  
    '/',  
    // 3. la cible sur laquelle on doit aller  
    // pour nous la cible c'est le controller et la méthode du controller, idem que notre tableau  
    [  
        // le nom du controller  
        "controller" => HomeController::class,  
        // le nom de la méthode  
        "method" => "home"  
    ],  
    // 4. le nom UNIQUE de la route  
    // on verra plus tard comment s'en servir  
    'home'  
);
```

Nous définissons la méthode HTTP requise. Il peut s'agir de l'une des listes ci-dessous:

GET: Pour rechercher ou récupérer des données.

POST: Pour soumettre des données pour traitement au serveur. Par exemple, l'envoi d'un formulaire.

PATCH: Pour mettre à jour des parties spécifiques d'une ressource. PUT: Pour mettre à jour une ressource existante, remplaçant complètement la ressource existante par celle fournie.

DELETE: Pour supprimer une ressource.

Ensuite nous définissons le chemin URL de la route. Nous le dirigeons ensuite vers le contrôleur et la méthode appropriés pour gérer la demande entrante lorsqu'elle correspond à ce chemin. Enfin, nous nommons cette route pour la rendre facilement identifiable pour une utilisation ultérieure dans l'application.

```
class HomeController extends CoreController
{
    public function home()
    {
        //die(__FILE__ . " : " . __LINE__);

        // TODO aller chercher en BDD les 5 categories pour la page d'accueil
        // 1. une instance du Model
        $categoryModel = new Category();
        // 2. on choisit la méthode/requete
        $categoriesForHome = $categoryModel->getAllForHome();
        // dump($categoriesForHome);
        // TODO : afficher notre template
        // 1. le nom du template
        $viewName = 'index';
        // 2. les informations spécifiques
        $viewData = [
            "categories" => $categoriesForHome
        ];
        // 3. on require les templates grace à la méthode show
        $this->show($viewName, $viewData);
    }

    public function legalMentions()
    {
        //die("Voici la page des mentions légales");
        // TODO : afficher notre template
        // 1. le nom du template
        $viewName = 'legalMentions';
        // 2. les informations spécifiques
        $viewData = []; // pour l'instant on a rien
        // 3. on require les templates grace à la méthode show
        $this->show($viewName, $viewData);
    }
}
```

Nous créons un fichier .htaccess qui permet de rediriger toutes les requêtes vers index.php.

```
# on active la réécriture d'URL
# pour l'utiliser et faire une redirection
RewriteEngine On

# dynamically setup base URI
# ici on met dans la variable $1 toutes les informations après ../public/
# ex : ../public/home => $1 = home
RewriteCond %{REQUEST_URI}::$1 ^(/.+)/(.*):.$2$
RewriteRule ^(.*) - [E=BASE_URI:%1]

# redirect every request to index.php
# and give the relative URL in "page" GET param

# Si le dossier n'existe pas : -d
RewriteCond %{REQUEST_FILENAME} !-d
# Si le fichier n'existe pas : -f
RewriteCond %{REQUEST_FILENAME} !-f

# alors on réécrire/rediriger vers index.php
# lors de la réécriture, on utilise le $1 pour fournir les informations après le ../public/
# ex : ../public/home => $1 = home
# on réécrit : index.php?page=/home
RewriteRule ^(.*)$ index.php?page=/$1 [QSA,L]
```

Pour la sécurité, nous avons à nouveau utilisé .htaccess avec "deny from all" écrit à l'intérieur pour restreindre l'accès à tout contenu présent dans app via le serveur web et protégera les informations contre la visibilité.

DOSSIER PROFESSIONNEL (DP)

2. Précisez les moyens utilisés :

J'ai travaillé avec notre équipe de projet.

3. Avec qui avez-vous travaillé ?

J'ai travaillé avec notre équipe de projet

4. Contexte

Nom de l'entreprise, organisme ou association ► *O'clock*

Chantier, atelier, service ► Projet réalisé pendant le formation.

Période d'exercice ► Du : *04/05/2023* au : *11/10/23*

5. Informations complémentaires (facultatif)

DOSSIER PROFESSIONNEL (DP)

Titres, diplômes, CQP, attestations de formation

(facultatif)

Intitulé	Autorité ou organisme	Date
Diplôme de Niveau 3 - Utilisation de Gaz: Installation et Entretien	Eastleigh College - Royaume-Uni	2014

Déclaration sur l'honneur

Je soussigné(e) [prénom et nom] Charlotte RUSSELL ,
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 Carnoët le 19/11/23

pour faire valoir ce que de droit.

Signature :



Documents illustrant la pratique professionnelle

(facultatif)

Intitulé
Cliquez ici pour taper du texte.

DOSSIER PROFESSIONNEL (DP)

ANNEXES

(Si le RC le prévoit)