420-3P2-HU Programmation d’interfaces web  
ÉVALUATION CERTIFICATIVE FINALE (60%)

# Modalités de l’évaluation

* Il s’agit d’un travail individuel
* La durée de l’évaluation est de 4h.
* Vous devez créer une copie personnelle du projet fourni dans GitLab (fork) et ajoutez l’enseignante à votre projet personnel pour qu’elle puisse voir votre code.
* Vous devez soumettre votre code (et ce fichier) dans GitLab à l’aide d’un ou de plusieurs archivages (*commits*).
* Le dernier archivage (commit) et la remise sur LÉA doivent être complétés avant de quitter la salle de classe.
* Vous devez déployer votre site avant le 16 décembre 23h55.
* Ne pas oublier de mentionner la source du code qui n’est pas le vôtre dans les commentaires. Vous avez le droit de réutiliser vos vieux exercices et laboratoires (l’indiquer en commentaire).
* N’hésitez pas à utiliser les méthodes de débogage que nous avons utilisées en classe pendant votre évaluation.
* Attention : un site qui n’est pas déployé n’est pas accepté. **Le déploiement fait partie de la remise.**
* La grille d’évaluation (évaluation complète) est à la fin de ce document.
* Gardez du temps pour l’audit (au moins 30 min) : c’est une façon facile de me montrer que vous comprenez ce qu’il faut faire, même si vous manquez de temps pour tout faire/corriger.
* Penser à me montrer ce que vous savez faire ! Même si vous ne réussissez pas à implémenter complètement ou parfaitement une fonctionnalité, vous pouvez me démontrer que vous êtes capables d’utiliser JavaScript et/ou JQuery pour aller chercher des valeurs et modifier le DOM.

Bonne réussite et bonnes vacances !

Au plaisir de vous revoir cet hiver 😊

# Exigences de la mini-application réalisée

L’objectif de cette évaluation est de créer une liste de lutins pour votre client « Le Pôle de l’emploi » qui permet aux lutins de trouver un poste à leur hauteur. Pour ce faire vous devrez répondre aux exigences du client (décrites ci-dessous) et respecter les maquettes fournies.

1. Voici, ci-contre, la hiérarchie du site à partir de laquelle vous devez construire le menu de navigation. (*Consulter aussi les maquettes*)
2. L’interface réalisée contient **deux** pages HTML : la page « Nos lutins » et une autre page « Ajouter un lutin ». Les autres pages ne seront pas réalisées dans le cadre de cette évaluation.
3. Sur la page « Nos lutins » on affiche une liste de lutins. On peut y chercher un lutin par son nom ainsi que modifier ou supprimer un lutin de la liste.
4. Sur la page « Ajouter un lutin » on crée un nouveau lutin **qui est ensuite affiché sur la page « Nos lutins »**.
5. Pour réaliser le travail vous devez corriger et compléter le code fourni.
6. Vous ne pouvez pas modifier l’emplacement des fichiers

### Liste des lutins (index.html)

1. Au chargement de la page, des lutins sont affichés dans la liste à partir des données disponibles à cet endroit : <https://5dd00936d5f1a500149a8305.mockapi.io/heros/lutins> (appel asynchrone).

Une image contenant texte

Description générée automatiquement

1. La photo des lutins dépend de leur plus grande qualité. Si leur qualité est la **curiosité**, leur photo sera **curiosite.jpg**
2. Le bouton « Ajouter » est un lien vers la page **ajouter.html**
3. Le bouton « Modifier » permet de modifier le lutin dans une boîte de dialogue. Seuls le nom et le id du lutin ne sont pas modifiables. L’interface fournit une rétroaction sous forme de **messages d’erreurs placés sous les champs qui ne respectent pas les conditions** imposées. Il est vérifié que :
   1. Les années d’expérience sont entre 1 et 45 ans
   2. Le poste souhaité ne contient pas de chiffres
   3. Le salaire demandé est :
      1. Au moins 20
      2. Au maximum 4x le nombre d’années d’expérience (*ex. si un lutin a 20 ans d’expérience, il devrait demander au maximum 80 cadeaux/année*).

Une image contenant texte

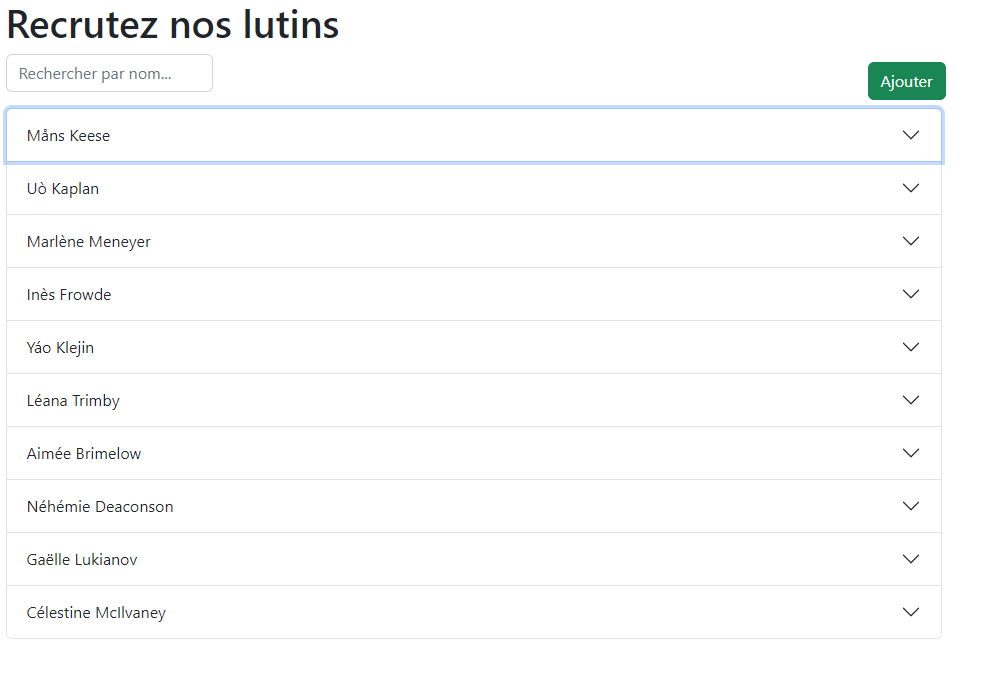
Description générée automatiquement

1. Lorsqu’on supprime un lutin de la liste un message de confirmation est affiché pour qu’on puisse confirmer la suppression.

Une image contenant texte

Description générée automatiquement

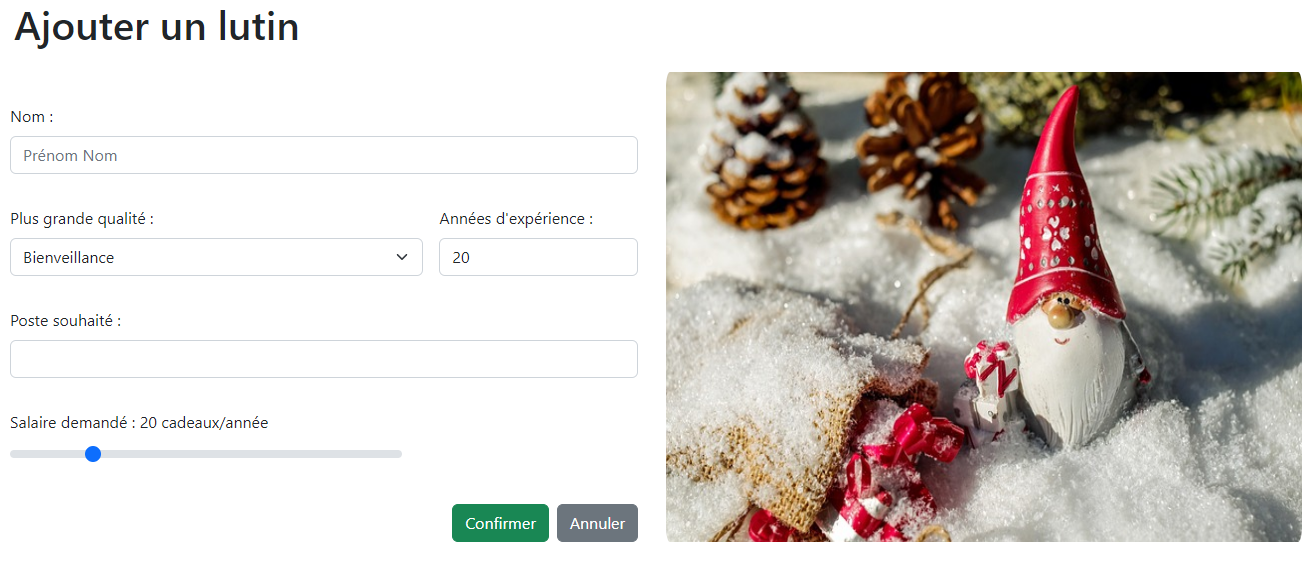
1. Lorsqu’on entre du texte dans le champ de recherche, seuls les lutins dont le nom contient ce texte sont affichés

Une image contenant texte

Description générée automatiquement

### formulaire pour ajouter un lutin (ajouter.html)

Pour ajouter un lutin à la liste, on doit soumettre le formulaire suivant :



Le formulaire s’adapte à la taille de l’écran

Une image contenant texte

Description générée automatiquement

1. L’interface fournit une rétroaction sous forme de **messages d’erreurs placés sous les champs qui ne respectent pas les conditions** imposées. Il est vérifié que :
   1. Les années d’expérience sont entre 1 et 50 ans
   2. Le poste souhaité ne contient pas de chiffres
   3. Le salaire demandé est :
      1. Au moins 20
      2. Au maximum 4x le nombre d’années d’expérience (*ex. si un lutin a 20 ans d’expérience, il devrait demander au maximum 80 cadeaux/année*).

Lorsqu’on clique sur le bouton « Ajouter » du formulaire, on affiche la liste des lutins comprenant le nouveau lutin si les données sont valides.

***Important : vous n’avez pas à garder les modifications faites sur la page de la liste (ex. les modifications et les suppressions); vous devez seulement envoyer votre nouveau lutin à la liste.***

Vous serez évaluées et évalués sur votre capacité à recréer la maquette et à implémenter les fonctionnalités demandées, mais aussi sur la qualité de votre code et le respect des normes et principes vus en classe (voir la grille de correction.)

Je vous conseille fortement de travailler dans cet ordre :

* HTML et mise en forme de base avec Bootstrap (maximum 45 min)
* JavaScript (maximum 2h)
* Audit et révision du code (revoir toutes les exigences; compléter la grille de la page suivante)
* Vérification de l’organisation et de la documentation du code CSS et JavaScript
* Mise en forme finale avec classes Bootstrap et CSS
* Déploiement et remise

# AUDIT

* Important : faire un archivage avant de faire le contrôle de la qualité **pour que je puisse évaluer votre audit**. Utiliser un message descriptif (ex. « Version évaluée lors de l’audit »)
* Remplir la grille d’audit (vous êtes notés sur la qualité de votre audit.) C’est normal de trouver des erreurs lors de l’audit.
* Corrigez votre code et votre interface et améliorez votre note 😊
* Vous pouvez utiliser les outils vus en classe pour faciliter votre audit (ex. [Wave](https://wave.webaim.org/extension/))

| **Aspect** | **Critère** | **OK ou décrire ce qu’il y a à corriger** |
| --- | --- | --- |
| **Typographie** | La taille du texte reflète la hiérarchie de l'information | Ok |
| **Couleurs** | Les couleurs ne sont pas la seule source de rétroaction (accessibilité) | Ok |
| Les contrastes sont conformes aux normes établies (WCAG) | OK |
| Utilisation uniforme | Ok |
| **Formulaires** | Toutes les données sont validées | Manques de validation et le peu qui est présent est dans le html |
| Les utilisatrices et les utilisateurs sont informés des restrictions | Les utilisateurs ne peuvent pas s’y rendre pour commencer |
| Les entrées obligatoires sont clairement indiquées (guidage et traitement des erreurs) | Non présent |
| Des messages d'erreurs clairs et précis, situés près de l'erreur, sont affichés au besoin | Non présent |
| **Navigation** | Les liens fonctionnent | Seul un lien fonctionne |
| L'internaute a un moyen de savoir où il est dans le site | Ok |
| La navigation est identique partout sur le site | Ok |
| **Respect des spécifications[[1]](#footnote-1)** | Toutes les fonctionnalités demandées ont été implémentées. | * L’appel asynchrone n’est pas fait. Sans ceci, la quasi-totalité de l’appli non-fonctionelle. * Le changement de photo ne s’effectue pas et ne peux pas être visuellement vérifié * La modification du lutin n’est pas accessible, mais quelques validations sont présentes en html. Ne modifie pas d’objet JS * Le nom du lutin n’est pas présent dans le message pour supprimer, non plus le message lui-même. * Les noms de lutins n’apparaissent pas dans la recherche; la liste de lutin elle-même est absente. * Un formulaire brut d’ajout de lutin est présent en HTML mais ne retourne aucun objet JS * Messages d’erreurs absent puisque l’interface est absente. |
| **Qualité du code** | Respect de toutes les normes du cours (documentation, nomenclature, etc.) Utiliser l’inspection de code de l’IDE et la liste des notes de cours (<https://vlevasseur.gitbook.io/420-3p2-interfaces-web/controle-de-la-qualite/qualite-du-code> ) | Sauf pour quelques variables non utilisés, l’inspection du code ne relève aucune erreur |

# Remise en classe :

1. Laissez votre site accessible jusqu’à ce que je remette les notes.
2. Faire un dernier commit, si ce n’est pas déjà fait.
3. Partager le projet avec l’enseignante, si ce n’est pas déjà fait.
4. Soyez fier/fière de vous ! Vous êtes maintenant capable de faire de « vrais » sites interactifs 😊

# Déploiement :

1. Déployer votre interface et remettre l’url dans le formulaire suivant si ce n’est pas déjà fait : <https://forms.gle/9eDyZ3Wmrgr5gAAT8> (devoir S15 - hébergement)

*Note : Je devrais pouvoir accéder au travail sans spécifier la page html. (Ex. https://listecadeaux.com ou https://listecadeaux.com/final mais pas https://listecadeaux.com/liste.html)*

1. Laissez votre site accessible jusqu’à ce que je remette les notes.
2. Soyez fier/fière de vous ! Vous êtes maintenant capable de mettre de « vrais » sites en ligne😊

| Critère et pondération | Critères de performance évalué(s) | Précisions | Excellent | Très bien | Bien | Insuffisant | Non observé | Commentaires | Note |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Réalisation d’une interface conforme aux normes et spécifications  (00ST.4)  **40 pts** | Utilisation appropriée du langage de balisage (00ST.4.1) et intégration correcte des images (00ST.4.3) | Utilisation appropriée du langage de balisage (HTML) selon toutes les normes vues en classe (accessibilité, utilisation de balises sémantiques, hiérarchie de l'information, validation adéquate des données des formulaires, nom des fichiers, etc.), conformément aux documents de conception. | Respect de toutes les normes vues en classe.  Respect de toutes les spécifications. Aucune erreur de syntaxe. | Quelques modifications mineures sont nécessaires pour rencontrer toutes les normes et toutes les spécifications. Aucune erreur de syntaxe. | La majorité des normes vues en classes sont globalement respectées. L’interface est conforme aux documents de conception dans son ensemble. | Problèmes significatifs dans la hiérarchie de l’information **OU** absence de validation dans les formulaires **OU** plusieurs spécifications ont été ignorées. | Présence de plusieurs problèmes majeurs (ex. plusieurs erreurs de syntaxe) OU non-respect global des normes vues en classe OU non-respect global des spécifications. |  | 15 |
| 15 | 12 | 9 | 6 | 0 |
| Création et utilisation appropriées des feuilles de styles (00ST.4.2) | Création et utilisation appropriées des feuilles de styles (CSS). La mise en forme est effectuée par le biais de feuille(s) de styles et/ou des classes Bootstrap. Tout le code est adéquatement indenté, organisé et commenté. Respect des normes vues en classe (ex. accessibilité, homogénéité), des maquettes et des instructions. Utilisation de sélecteurs appropriés. Les noms des classes créées sont descriptifs. | Mise en forme homogène qui met de l'avant la hiérarchie de l’information. Respect de toutes les normes vues en classe. Utilisation appropriée des classes. Respect des maquettes et de toutes les instructions. | Mise en forme homogène qui met de l'avant la hiérarchie de l’information. Quelques modifications mineures sont nécessaires pour rencontrer toutes les normes et exigences. | Respect de la plupart des normes vues en classes. Plusieurs modifications sont nécessaires pour rencontrer toutes les normes. Présence de code inutile ou redondant. | La mise en forme n'a pas été ajoutée de façon appropriée **(ex. présence de balise style dans le HTML)** OU aucune classe utilisée. | La mise en forme nécessite plusieurs modifications pour être homogène OU pour mettre de l'avant la hiérarchie de l'information. |  | 15 |
| 15 | 12 | 9 | 6 | 0 |
| Adaptation de l’interface en fonction du format d’affichage et de la résolution (00ST.4.4) | Site réactif. Respect des normes présentées en classe et des documents de conception (maquette). Utilisation adéquate de Bootstrap. Utilisation appropriée des propriétés d’affichage, de positionnement, d’espacement pour répondre aux spécifications. | Utilisation adéquate et efficace de Bootstrap. Site réactif répondant à toutes les normes vues en classe et correspondant à toutes les spécifications. | Quelques modifications mineures sont nécessaires pour rencontrer toutes les normes et toutes les spécifications. Utilisation de Bootstrap. | La plupart des spécifications sont respectées. Interface réactive, mais la réactivité pourrait être améliorée. Bootstrap peut ne pas être utilisé, ou utilisé de façon minimale. | De nombreuses modifications sont nécessaires pour rencontrer les spécifications. L’interface est peu réactive et peut être **difficilement utilisable à certaines résolutions**. | Le site n’est pas réactif. Non-respect global des spécifications. |  | 10 |
| 10 | 8 | 6 | 4 | 0 |
| Gestion adéquate des événements selon les normes et spécifications  (00ST.6)  **40 pts** | Manipulation adéquate des objets du modèle DOM (00ST.6.1) | Utilisation adéquate de JavaScript et de JQuery pour implémenter les fonctionnalités demandées. Manipulation adéquate des données au format JSON. Sélection, création et modification des objets à l'aide de méthodes appropriées. Utilisation de sélecteurs adéquats. Code concis (pas de répétition inutile; utilisation appropriée des paramètres). Respect des normes vues en classe (ex. utilisation de fichiers séparés, utilisation de JQuery, nomenclature des variables, des fonctions et des fichiers, documentation adéquate de toutes les méthodes et fonctions.) | Sélection, création et modification des objets à l’aide de méthodes JavaScript ou JQuery appropriées. Respect de toutes les normes vues en classe. Respect de toutes les spécifications. Code concis. Parcours, création et modification des objets JavaScript de façon efficace. | Sélection, création et modification des objets à l’aide de méthodes JavaScript ou JQuery généralement appropriées. Respect généralisé des normes vues en classe. Quelques modifications mineures sont nécessaires pour respecter toutes les normes et les spécifications. | Sélection, création et modification des objets à l’aide de méthodes JavaScript ou JQuery généralement appropriées. Présence de quelques d’erreurs mineures d’exécution qui n’empêchent pas l’utilisation de l’interface OU quelques modifications mineures permettent de corriger le code. Respect global des spécifications. | Non-respect de plusieurs spécifications OU le code est dans le HTML OU JQuery n'est pas utilisé. | Présence de plusieurs erreurs qui rendent difficile l’utilisation de l’interface lors de la sélection, la création ou la modification OU non-respect global des spécifications. |  | 20 |
| 20 | 16 | 12 | 8 | 0 |
| Programmation correcte des interactions entre l’interface Web et l’utilisatrice ou l’utilisateur (00ST.6.2) et programmation et intégration adéquates d’animations ou de widgets (00ST.6.3) | Validation adéquate des données entrées par l’utilisatrice ou l’utilisateur (validation de ce qui ne peut être fait dans le HTML.) Respect des spécifications. Gestion adéquate des évènements : soumission du formulaire, détection de données erronées, mise à jour lors de l'entrée de données, clic, etc. Les évènements sont correctement identifiés et les méthodes appropriées sont lancées. Des messages d'erreurs adéquats et placés près de l'erreur sont fournis au besoin à l'utilisateur. Utilisation de JQuery et de Bootstrap. | Tous les évènements nécessaires au respect des documents de conception sont correctement identifiés et gérés. Des messages d'erreurs adéquats placés près de l'erreur sont fournis au besoin à l'utilisateur. Utilisation adéquate de JQuery et Bootstrap. | La plupart des évènements sont correctement identifiés et gérés, mais pas tous (au plus quelques modifications mineures sont requises pour respecter toutes les spécifications; l'interface est utilisable). Présence de messages d'erreur au besoin. Utilisation de JQuery et Bootstrap. | La plupart des évènements sont correctement identifiés et gérés, mais pas tous (quelques fonctionnalités mineures peuvent être affectées). Respect global des spécifications. | Plusieurs évènements sont ignorés OU **absence de validation et de messages d'erreur**. | Présence d'erreurs qui rendent difficile l’utilisation de l’interface OU non-respect global des spécifications. |  | 20 |
| 20 | 16 | 12 | 8 | 0 |
| Déploiement et tests adéquats selon les normes et spécifications  (00ST.7 et 00ST.8)  **10 pts** | Contrôler la qualité de l’application. (00ST.7) et participer au déploiement de l’application chez un hébergeur Web. (00ST.8) | Réalisation d’une revue de code rigoureuse. Réalisation d’un audit ergonomique à l’aide d’une grille fournie. Déploiement à l’endroit spécifié et redirection vers https. Corrections appropriées suite à l’audit. | La revue de code et l’audit ont été faits de façon rigoureuse et méticuleuse. La majorité des problèmes soulevés pendant l’audit ont été corrigés. L’application a été déployée selon les exigences. | La revue de code et l’audit ont été faits en grande partie. L’application déployée est fonctionnelle. Tous les problèmes majeurs soulevés à l’audit ou à la revue de code ont été corrigés. | La revue de code et l’audit ont été faits en grande partie. L’application déployée est fonctionnelle. Peu de correctifs ont été apportés ou la revue de code ou l’audit manque de rigueur. | Un des éléments n’a pas été réalisé : déploiement fonctionnel, revue de code, audit ergonomique. | L’application déployée n’est pas fonctionnelle, la revue de code et l’audit ergonomique n’ont pas été remis. |  | 10 |
| 10 | 8 | 6 | 4 | 0 |
| Documentation rigoureuse et justification de ses choix  (00ST.1, 00ST2.2 et 00ST.9)  **10 pts** | Analyser le projet de développement de l’application (00ST.1), préparer l’environnement de développement informatique (00ST.2) et rédiger la documentation (00ST.9) | L’ensemble du code est adéquatement documenté. Toutes les fonctions sont adéquatement documentées (rôle, paramètres, type de retour). Identification adéquate des tâches à effectuer lors de l’analyse des documents de conception. Notation claire du travail effectué par l’utilisation de messages descriptifs lors des archivages (*commit*). | L’ensemble du code est adéquatement documenté. Toutes les fonctions sont adéquatement documentées (rôle, paramètres, type de retour). Identification adéquate des tâches à effectuer lors de l’analyse des documents de conception. Notation claire du travail effectué par l’utilisation de messages descriptifs lors des archivages (*commit*). | Quelques modifications mineures sont nécessaires pour rencontrer les exigences de ce critère. | Plusieurs modifications mineures sont nécessaires pour rencontrer les exigences de ce critère. Le code est documenté, mais pas toujours adéquatement. Quelques tâches n’ont pas été identifiées. Les messages des archivages gagneraient à être plus descriptifs. | Faible documentation du code OU absence d’un livrable (archivage) OU les tâches à effectuer n’ont pas été identifiées. | Absence de plus d’un élément (documentation, tâches à effectuer, livrables). |  | 10 |
| 10 | 8 | 6 | 4 | 0 |
| **Total** | | | | | | | | | 100 |

**Commentaires**

1. Important : permet d’évaluer votre compréhension des exigences [↑](#footnote-ref-1)