ONLINE
FORMAPRO

Développer une interface
Utilisateur web dynamique

MovieQuiz

Travail : individuel

Début projet : 05/06
Fin du projet : 16/06

MovieQuiz

CONSIGNE

L'objectif de ce projet est de créer un jeu interactif en JavaScript, à réaliser individuellement en 7 jours. Le jeu permettra aux joueurs de deviner des films à partir d'images grises, et de vérifier leurs réponses en comparant avec les réponses attendues.

PRODUCTION ATTENDUE

1.Un jeu fonctionnel en JavaScript qui respecte les spécifications fournies dans l'énoncé.
2.Une page de préloader qui s'affiche au démarrage du jeu pour assurer une expérience fluide.
3.Une page d'accueil avec des vignettes d'images de films, où les vignettes sont grises si le film correspondant n'a pas encore été trouvé, et en couleur avec le nom du film s'il a été trouvé.
4.Lorsque le joueur clique sur une image grise, une nouvelle fenêtre s'ouvre pour afficher l'image en grand format.
5.Dans la fenêtre de l'image en grand, un champ de saisie est fourni où le joueur peut entrer sa réponse.
6.Après avoir cliqué sur "Soumettre", la réponse donnée par le joueur est comparée à celle attendue, extraite à partir d'un fichier JSON contenant les réponses pour chaque film.
7.Si la réponse est correcte, un message de succès apparaît à l'utilisateur, et après 3 secondes, il est redirigé vers la page d'accueil. Sur cette page, l'image correspondante est affichée en couleur.
8.Si la réponse est incorrecte, un message d'avertissement apparaît à l'utilisateur, et le champ de saisie est réinitialisé pour lui permettre de retenter sa chance.
9.La progression du joueur est sauvegardée en utilisant le localStorage à chaque fois qu'une bonne réponse est donnée.
10. Lors qu'un utilisateur clique sur l'image d'un film ou série déjà trouvée, alors la page détail de celui-ci s'ouvrira. Les données seront issues de l'API TMBD (https://developer.themoviedb.org/docs)
11. Une version en ligne et un accès au repository gitHub.
12. Un code CSS fait à l'aide de SASS.
13. Utilisation de la bibliothèque Swiper recommandée.

MovieQuiz

LES ÉTAPES DU PROJET

1.Mise en place du préloader : Créer une page de chargement qui s'affiche au démarrage du jeu pour assurer une expérience fluide.

2.Page d'accueil avec vignettes d'images : Après le préloader, afficher une page d'accueil contenant plusieurs vignettes d'images de films. Les vignettes doivent être grisées si le film correspondant n'a pas encore été trouvé, sinon elles doivent s'afficher en couleur avec le nom du film.

3.Ouverture de l'image sélectionnée : Lorsque le joueur clique sur une image grisée, une nouvelle fenêtre s'ouvre pour afficher l'image en grand format.

4.Champ de saisie de la réponse : Dans la fenêtre de l'image en grand, fournir un champ de saisie où le joueur peut entrer sa réponse.

5.Comparaison de la réponse : Après avoir cliqué sur "Soumettre", comparer la réponse donnée par le joueur à celle attendue. La réponse attendue sera extraite d'un fichier JSON contenant les réponses pour chaque film.

6.Affichage des résultats : Si la réponse est correcte, afficher un message de succès à l'utilisateur et le rediriger vers la page d'accueil après 3 secondes. Sur la page d'accueil, l'image correspondante sera alors affichée en couleur.

7.Gestion des réponses incorrectes : Si la réponse est incorrecte, afficher un message d'avertissement à l'utilisateur et réinitialiser le champ de saisie pour lui permettre de retenter sa chance.

8. Ouverture de la page détail d’un film ou d’une série au clic sur un film ou série déjà trouvée. Se connecter à l’API TMDB, récupérer les informations nécessaires sur le film ou la série, le synopsis, les acteurs et les afficher.

MovieQuiz

OBJECTIFS D'APPRENTISSAGE :

- Maîtriser les bases du développement web en utilisant JavaScript.
- Manipuler et afficher des images en utilisant le DOM (Document Object Model).
- Gérer les événements utilisateur (clic, soumission de formulaire, etc.) avec JavaScript.
- Charger des données à partir d'un fichier JSON.
- Comparer les réponses fournies par les utilisateurs avec les réponses attendues.
- Utiliser le localStorage pour sauvegarder la progression du joueur.
- Se connecter et récupérer les informations issues d’une API

COMPÉTENCES DÉVELOPPÉES :

- Programmation en JavaScript.
- Manipulation du DOM.
- Gestion des événements.
- Traitement des données JSON.
- Gestion du stockage local avec localStorage.
- Traitement des données issues d’une API

MovieQuiz

ANNEXES – STRUCTURE FICHIER JSON

{ "movies": { "movie": [{ "title": "Movie Title", "picture": "src link" }, { "title": "Movie Title", "picture": "src link" }, { "title": "Movie Title", "picture": "src link" }, { "title": "Movie Title", "picture": "src link" }, { "title": "Movie Title", "picture": "src link" }] } }


```json
{
  "movies": {
    "movie": [
      {
        "title": "Movie Title",
        "picture": "src link"
      },
      {
        "title": "Movie Title",
        "picture": "src link"
      },
      {
        "title": "Movie Title",
        "picture": "src link"
      },
      {
        "title": "Movie Title",
        "picture": "src link"
      },
      {
        "title": "Movie Title",
        "picture": "src link"
      }
    ]
  }
}
```