

Projet individuel : Développer un jeu de Memory avec HTML5 – CSS3 – JavaScript

1. Objectif du projet

Vous devez réaliser individuellement une application Web complète proposant un jeu de Memory, associée à un système d'inscription, d'authentification et de récupération de mot de passe, le tout en utilisant exclusivement HTML5, CSS3 et JavaScript natif. Aucun framework, aucune bibliothèque externe, aucun back-end et aucun serveur ne doivent être utilisés. Toutes les données doivent être stockées dans le localStorage du navigateur.

2. Périmètre fonctionnel global

2.1 Gestion d'utilisateur

Vous devez permettre à l'utilisateur de :

- créer un compte (nom + email + mot de passe)
- se connecter
- se déconnecter
- récupérer son mot de passe via une page dédiée
- sauvegarder ses préférences de jeu
- conserver un historique de ses meilleurs scores

2.2 Jeu de Memory

L'application doit permettre de :

- choisir un thème de jeu
- choisir la dimension du plateau
- afficher un plateau généré dynamiquement
- retourner et comparer des cartes
- compter le nombre de coups
- détecter la fin de partie
- enregistrer automatiquement les meilleurs scores
- afficher les 10 meilleurs scores du joueur connecté

3. Architecture HTML attendue

Vous devez structurer l'application en plusieurs fichiers HTML :

- index.html
- login.html
- register.html
- forgot-password.html
- reset-password.html
- game.html
- scores.html

4. Exigences CSS

- Mise en forme cohérente et responsive
- Utilisation de Flexbox ou CSS Grid
- Gestion des cartes recto/verso via transitions et ou transformations
- Media queries pour mobile / tablette / desktop pour le responsive-design

5. Exigences JavaScript

5.1 Authentification & profil

- création, connexion, déconnexion
- récupération et redéfinition du mot de passe
- stockage des préférences utilisateur

5.2 Logique du Memory

- génération dynamique du plateau
- mélange aléatoire des cartes
- gestion des clics
- comparaison des paires
- verrouillage temporaire
- comptage des coups
- détection de fin de partie
- sauvegarde des scores

6. Contraintes techniques

- Aucun framework JS ou CSS
- Aucun CDN externe
- Aucune base de données : uniquement localStorage
- Projet exécutable localement via index.html

7. Livrables attendus

7.1 Dépôt GitHub privé

Vous devez créer un dépôt GitHub privé contenant l'ensemble du code source.

Contraintes :

- Le dépôt doit être privé.
- Vous devez ajouter @ssy-sdv comme collaborateur.
- Commits progressifs et nommés clairement.
- Structure propre du dépôt.

7.2 README complet et professionnel

Le README doit inclure :

- Titre du projet
- Description générale
- Instructions d'installation et d'exécution

- Organisation du code
- Fonctionnalités implémentées / non implémentées
- Captures d'écran (optionnel)
- Limites et pistes d'amélioration
- Version / changelog

7.3 Projet exécutable

Le projet doit fonctionner en ouvrant index.html.

7.4 Exemple de structure recommandée

```
/
├── index.html
├── login.html
├── register.html
├── forgot-password.html
├── reset-password.html
├── game.html
├── scores.html
├── css/
│   └── style.css
├── js/
│   ├── auth.js
│   ├── game.js
│   ├── storage.js
│   └── utils.js
└── images/
```

8. Critères d'évaluation

- Qualité HTML : sémantique, propreté, structure
- Qualité CSS : esthétique, responsive, transitions
- Qualité JS : structure, commentaires, logique robuste
- Fonctionnalités : jeu jouable, compte utilisable, scores enregistrés
- UX : fluidité, cohérence, navigation intuitive