Épreuve finale – Applications Web

Le présent travail a pour objectif de mobiliser l'ensemble des notions abordées au cours de la session dans le cadre du cours d'**applications web**. À travers le développement d'un **ePortfolio interactif** en React, vous mettrez en pratique vos compétences en conception d'**interfaces web dynamiques**, en intégration de données via des **API externes**, ainsi qu'en structuration et **organisation de projets**.

Le développement devra être réalisé de manière **incrémentale**, selon une **approche itérative** favorisant l'amélioration continue du produit. Vous serez tenu de présenter, tout au long du projet, des **démonstrations partielles** obligatoires à votre enseignant. Ces démonstrations permettront de **recueillir des commentaires ciblés**, de **valider vos choix techniques et fonctionnels**, et de vous aider à **recentrer votre travail** si nécessaire.

L'épreuve se conclura par une **démonstration finale** portant sur l'ensemble du site développé, laquelle constituera une **condition essentielle à l'évaluation de votre projet**.

Exigences générales

Le projet consistera à réaliser un site web complet avec React, entièrement personnalisé, reflétant votre identité professionnelle, et démontrant votre capacité à intégrer plusieurs fonctionnalités réparties sur différentes pages.

Le site devra respecter les critères suivants :

- Être **développé entièrement par vous**, sans recourir à des modèles préexistants ni à des générateurs de site.
- Être **soigné sur le plan visuel** et convivial pour l'utilisateur :
  - L'interface doit être claire, attrayante et cohérente.
  - L'utilisation de bibliothèques CSS comme Tailwind, Bootstrap ou autres est permise et encouragée.
- Intégrer au moins **trois pages** principales, décrites à la section suivante, comportant chacune les éléments suivants :
  - Une en-tête (Header)
  - Une barre de navigation (NavBar)
  - Un pied de page (Footer)

Fonctionnalité complémentaire obligatoire

Une fonctionnalité supplémentaire, laissée à votre choix, devra être intégrée afin de vous permettre d'explorer une nouvelle compétence. Voici quelques exemples inspirants :

- Intégrer un **carrousel d'images** pour présenter vos réalisations, projets ou loisirs, tout en explorant les animations CSS.
- Ajouter un **thème sombre / clair** pour expérimenter avec la gestion du style conditionnel et améliorer l'expérience utilisateur.
- Ajouter une **barre de recherche** dans la section Projets permettant :

    - de filtrer par technologie utilisée ;
    - de saisir une technologie pour vérifier si elle est associée à un ou plusieurs de vos projets.
- Autres idées : des suggestions supplémentaires seront proposées en classe.

> La fonctionnalité choisie devra être validée par l'enseignant avant son intégration dans le projet.

Les 3 pages obligatoires

1. Le ePortfolio

Cette page constituera le cœur de votre site. Elle devra être divisée en trois sections distinctes, chacune ayant une fonction bien précise.

I. Présentation personnelle

Cette section a pour but de mettre en lumière qui vous êtes. Elle vous permet de vous présenter de manière personnelle et authentique. Vous avez la liberté de choisir son organisation, en vous assurant que les éléments suivants y figurent :

- Votre nom
- Une photo ou un avatar
- Quelques traits de personnalité
- Vos intérêts professionnels, passions, loisirs, etc.

À vous d'opter pour une mise en page à votre image, qui soit à la fois informative et attrayante. La section pourrait par exemple se présenter sous forme d'une galerie, si vous êtes un artiste et souhaitez mettre de l'avant vos œuvres.

II. Présentation professionnelle

Cette section devra mettre en valeur votre parcours académique et professionnel. Elle peut inclure les éléments suivants :

- Votre cheminement scolaire et vos formations
- Vos emplois et stages
- Vos activités parascolaires ou compétitions (ex. : concours de robotique)
- Vos implications bénévoles
- La mention de personnes de référence, si pertinent, pour appuyer certaines expériences

Vous pouvez opter pour une présentation classique (listes, tableaux, etc.) ou choisir une approche plus visuelle, comme une ligne du temps, des icônes, ou une mise en page originale qui reflète votre style.

L'essentiel est de rendre cette section claire, structurée et représentative de votre parcours.

III. Projets informatiques

Cette section mettra en valeur les projets que vous avez réalisés, que ce soit dans le cadre de vos cours ou à titre personnel. Vous êtes libre de choisir le rendu visuel (grille, cartes, colonnes, etc.), mais chaque projet devra impérativement inclure les éléments suivants :

- Le titre du projet
- Une courte description
- La liste des technologies utilisées, classées par type (ex. : Java → langage, VS Code → IDE, React → bibliothèque/framework)
- Images ou captures d'écran si disponibles
- Lien vers GitHub ou toute autre plateforme de code, si applicable

La présentation doit être claire et agréable à consulter. L'utilisation de cartes, badges ou icônes est fortement encouragée pour structurer l'affichage.

> Gardez en tête que les éléments affichés dans cette section devront également être collectés via le formulaire pour permettre l'ajout dynamique de nouveaux projets.

---

2. Le formulaire d'ajout de projet

Un formulaire permettant l'ajout de nouveaux projets devra etre mis en place. Celui-ci devra inclure un champ pour chacun des éléments requis dans la section **projets informatiques** (titre, description, technologies, image, lien GitHub, etc.).

Le formulaire doit être simple et agréable à utiliser, pensez à y intégrer des mécanisme facilitant la saisie tel que des menus déroulants pour séléctionner le type de technologie ou autre. Enfin, à la soumission du formulaire, l'ajout dynamique à la section projets devra se faire.

> À la soumission, les informations saisies devront être ajoutées dynamiquement à la section projets informatiques.

3. API externe d'un champ d'intérêt

Cette page mettra en œuvre l'utilisation d'une API publique libre, en intégrant deux fonctionnalités principales : le défilement d'éléments et la recherche ciblée.

Fonctionnalité attendue

- **Défilement d'éléments** : la page devra permettre de naviguer à travers les éléments de l'API à l'aide de boutons "précédent" / "suivant", en affichant pour chaque élément des informations concises.
- Exemple : une carte affichant le nom et l'image de différents Pokémon. La navigation doit être fluide et continue dans les deux sens, sans interruption.
- **Recherche d'un élément** : l'utilisateur doit pouvoir effectuer une recherche selon différents critères, tels que le nom, l'identifiant ou une catégorie/caractéristique spécifique.
    - Exemple : rechercher "pikachu" et afficher sa fiche complète (image, poids, types, attaques, etc.).

> L'interface doit être intuitive et claire, avec des instructions visibles sur la façon d'effectuer une recherche.

Intégration d'une API publique libre

Vous devez choisir une API liée à un **centre d'intérêt personnel** (ex. météo, cinéma, voitures, pokémon, etc.).

- Une liste d'APIs recommandées est disponible ici : https://github.com/public-apis/public-apis.
- Le **choix de l'API** ainsi que les **données affichées** devront être **validés par l'enseignant**.

> La **pertinence des informations à afficher** et le style de présentation dépendront de l'API choisie. Il est de votre responsabilité d'adapter l'affichage et l'expérience utilisateur en conséquence.

Fonctionnalité bonus (facultative)

Vous pouvez ajouter une **page d'accueil** incluant un **mini système d'enregistrement**, par exemple en demandant à l'utilisateur de saisir son nom et son adresse courriel.

Une fois cette étape complétée, l'**accès aux autres pages du site devra être conditionnel à cette inscription**. Autrement dit, tant qu'un utilisateur ne s'est pas enregistré via la page d'accueil, il ne pourra pas accéder aux pages internes (ex. : `/interet` ou `/projets`).

Cette fonctionnalité peut nécessiter l'apprentissage ou l'exploration de notions supplémentaires (stockage local, redirection conditionnelle, logique d'accès, etc.).

> Assurez-vous que toutes les fonctionnalités principales du projet soient terminées, fonctionnelles et validées avant de vous lancer dans cette extension.

Les contraintes techniques

Le travail devra démarrer à partir d'un projet React vide (`npm create vite@latest`), sans template préconstruit. Le code source devra être versionné sur un **dépôt GitHub**, partagé avec l'enseignant **dès la première démonstration**.

Le projet sera évalué notamment sur les aspects techniques suivants :

- Organisation logique du projet
  - Les composants doivent être bien séparés et réutilisables, selon leur rôle (présentation, logique, formulaire, etc.)
  - La structure des fichiers et dossiers doit être claire et cohérente (ex. : components/, pages/, assets/, etc.).
- Utilisation appropriée des hooks React
  - Usage pertinent de `useState`, `useEffect` et, au besoin, d'autres hooks (`useRef`, `useMemo`, etc.)
  - Gestion propre et efficace de l'état et des effets secondaires.
- Qualité du code
  - Nommage cohérent et significatif des variables et fonctions.
  - Respect des bonnes pratiques enseignées : lisibilité, encapsulation, réutilisabilité, clarté des responsabilités, etc.

> L'ensemble de votre code devra être compréhensible, propre et maintenable en tout temps, car vous serez amené à le présenter et le modifier en direct lors des démonstrations.

Flux de travail et déroulement des démonstrations

Le développement de votre projet s'échelonnera sur **trois semaines (15 heures de cours)**. Afin d'assurer un accompagnement régulier et structuré, des **démonstrations partielles obligatoires** auront lieu chaque semaine, pendant les séances prévues en classe.

Fonctionnement général

- Chaque démo vous permettra d'obtenir des **rétroactions concrètes** pour ajuster ou bonifier votre travail.
- Les démonstrations sont **obligatoires** : une **absence non justifiée** ou le **non-respect des attentes** formulées lors de la démo précédente pourra entraîner une **pénalité sur la note finale**.
- Un **push sur GitHub** (accompagné de commits clairs) est attendu **chaque semaine**, afin de rendre visibles vos avancements et permettre un suivi continu du projet.
- Lors de chaque démonstration, vous devrez être en mesure de :
    - Expliquer clairement votre démarche
    - Naviguer dans votre code de façon fluide
    - Apporter des modifications en direct, si nécessaire
      Le **recours à du code provenant de sources externes est strictement interdit** pour les fonctionnalités de base du site (formulaires, navigation, affichage de données, etc.) ; toute autre ressource externe utilisée doit être **clairement référencée**.

Le suivi et le maintient du dépôt GitHub

Un suivi rigoureux de votre progression est attendu, pour ce faire :

- Vous devrez **organiser votre travail** dans votre fichier `README.md` afin de faciliter la communication de vos avancements lors des démos.
- Une **section "commentaires et suivis"** devra être maintenue à jour dans votre fichier `README.md` afin de **documenter les rétroactions de l'enseignant** à chaque étape.
- Vous devez effectuer des **commits réguliers** sur GitHub, avec des messages clairs et significatifs, permettant de retracer l'évolution de votre projet semaine après semaine.

> Le contenu des démonstrations partielles sera **personnalisé** selon la direction que vous choisirez pour votre projet. L'enseignant adaptera, dès la première démo, ses attentes en fonction de vos choix (API sélectionnée, structure du contenu, ordre de développement, etc.).

Attentes de la première démonstration

Les éléments suivants devront être prêts dès la première démonstration :

- Un **dépôt GitHub actif**, avec le projet initialisé et l'enseignant ajouté comme collaborateur
- Le **lien du dépôt soumis** dans la boîte de remise prévue à cet effet sur Moodle
- Le **choix de l'API** pour la page **champ d'intérêt**, accompagné d'une description claire des fonctionnalités prévues :
    - Explications sur le fonctionnement de la recherche
    - Quelles informations seront affichées lors du défilement
- La structure de base du projet mise en place :
    - Navigation fonctionnelle entre les pages principales

    - Composants principaux créés (même avec du contenu temporaire, comme un simple `h1`)
- Un plan de travail pour les semaines à venir, indiquant l'ordre dans lequel vous prévoyez développer les différentes fonctionnalités

---

Grille d'évaluation

Les critères d'évaluation	**pondération**
Assiduité et rigueur du travail	**20 %**
- Présence et qualité des démos partielles	
- Propreté du dépôt et du README	
- Suivi du projet à l'aide de **commits réguliers**, accompagnés de **messages clairs et significatifs**	
Fonctionnalités et rendu visuel	**50 %**
- Présentation, projets, formulaire, API	
- Fonctionnement global et navigation	
- Mise en page (CSS / librairies)	
Qualité du code et respect des notions	**30 %**
- Bonne utilisation de React et des hooks	
- Séparation des composants, propreté	
- Bonne encapsulation et logique claire	

> **Important :** sans démonstration finale, **AUCUNE NOTE ne sera attribuée** au projet, ce qui entraînera un **échec automatique** de l'évaluation. Aucun projet ne sera corrigé en l'absence de cette présentation.