

TP 1 :

1- Sur le bureau, Créer un dossier « Front-end »

2- Réaliser une page web suivante nommée « ser1tp1.html » :



Tableau des Balises HTML

HTML

Liste des balises HTML et leur utilisation courante

Balise	Fonction de la Balise	Attribut	Fonction de l'Attribut
<p>	Définit un paragraphe	class	Applique des styles CSS
		id	Identifiant unique
		style	Définit un style CSS inline
<a>	Crée un lien hypertexte	href	Lien URL
		target	Détermine où ouvrir le lien
		title	Texte d'info bulle

3- Ajouter les autres balises que vous connaissez à ce tableau et enregistrez votre travail.

TP2 :

1- Dans une page web nommée « ser1tp2.html », Réaliser les deux listes suivantes :

Liste à puces imbriquées

- Front-end
 - HTML
 - CSS
 - JavaScript
 - React
 - Vue
 - Angular
- Back-end
 - Node.js
 - PHP
 - Python
 - Django
 - Flask

Liste Numérotée Imbriquée

1. Front-end
 1. HTML
 2. CSS
 3. JavaScript
 1. React
 2. Vue
 3. Angular
2. Back-end
 1. Node.js
 2. PHP
 3. Python
 1. Django
 2. Flask

2- Réaliser la page web « ser2tp1.html »:

Inscription à un Événement

Informations Personnelles	Coordonnées
Nom : <input type="text"/>	Adresse e-mail : <input type="text"/>
Prénom : <input type="text"/>	Numéro de téléphone : <input type="text"/>
Date de naissance : <input type="text" value="jj/mm/aaaa"/>	Adresse : <input type="text"/>
Genre : <input type="text" value="Masculin"/>	Ville : <input type="text"/>

Soumettre

Réinitialiser

3- Réaliser la page web « ser2tp2.html »:

Choix de Niveau d'Études

Niveau d'études : <input type="radio"/> Licence <input type="radio"/> Master <input type="radio"/> Doctorat
Options : <input type="checkbox"/> Demande de bourse <input type="checkbox"/> Demande de logement <input type="checkbox"/> Demande de stage

Soumettre

Réinitialiser

1- En utilisant HTML et CSS, réaliser une page web(CV) sous le nom « ser3tp1.html » :

Nom Prénom

🏠 123 Rue de Exemple, Ville, Code Postal
☎ +33 6 12 34 56 78
✉ exemple@email.com
🔗 linkedin.com/in/exemple

Profil

Passionné(e) par [votre domaine], je possède une expérience significative dans [vos spécialités]. Mon objectif est de [votre objectif professionnel].

Expérience Professionnelle

Poste - Entreprise

Janvier 2020 - Présent

- Responsabilité 1
- Responsabilité 2
- Responsabilité 3

Poste Précédent - Ancienne Entreprise

Janvier 2018 - Décembre 2019

- Responsabilité 1
- Responsabilité 2
- Responsabilité 3

Éducation

Diplôme - Établissement

Année d'obtention

Compétences

Compétence 1

Compétence 2

Compétence 3

Compétence 4

Compétence 5

TP 1 :

1- Créer une page web interactive « ser4tp1.html » où les étudiants peuvent tester leurs connaissances avec un quiz simple. Cette page contiendra des questions à choix multiple (QCM) et utilisera du HTML et CSS, avec une touche de JavaScript pour gérer les réponses.

Quiz de Connaissances

1. Quelle est la capitale de la France ?

- ☐ Londres
- ☒ Paris
- ☐ Berlin
- ☐ Madrid

2. Quel est le plus grand océan du monde ?

- ☒ Atlantique
- ☐ Indien
- ☐ Pacifique
- ☐ Arctique

Soumettre

Vous avez obtenu 1 sur 2 !

TP2 :

1- A partir de la page web précédente, créer une nouvelle page web « ser4tp2.html » en Ajoutant plus de questions.

2- Afficher des messages en fonction du score

(par exemple, "Bien joué !" ou "Essayez encore !").

3- Enregistrer les scores dans le stockage local pour un quiz futur.

TP 1 :

Créez un formulaire avec deux champs de saisie pour entrer deux nombres, affichez la somme des deux nombres.

Calculatrice de Somme

12	15	Calculer la Somme
----	----	-------------------

La somme est : 27

TP 2 :

Créez un formulaire de connexion avec un champ pour le nom d'utilisateur et un champ pour le mot de passe. Vérifiez que les champs ne sont pas vides et affichez un message d'erreur si l'un d'eux est vide. La bordure de la zone non remplie doit être en rouge.

Connexion

MAHDAD	Mot de passe	Se connecter
--------	--------------	--------------

Remplir tous les champs.

Dans le cas contraire, affiche un message « Succès » dans une boîte de dialogue

TP 3:

Créez un formulaire avec un champ pour le mot de passe. Lorsque l'utilisateur saisit son mot de passe, vérifiez la longueur et affichez un message indiquant si le mot de passe est "faible", "moyen", ou "fort".

Vérification du Mot de Passe

Mot de passe : Moyen

TP 1:

Créez un formulaire avec deux champs de saisie : un pour la taille (en mètres) et un pour le poids (en kilogrammes). Lorsque l'utilisateur soumet le formulaire, affichez son IMC, et donner la catégorie.

(Insuffisance pondérale : $IMC < 18.5$; Poids normal : $18.5 \leq IMC < 24.9$;

Surpoids : $25 \leq IMC < 29.9$; Obésité : $IMC \geq 30$)

Calcul de l'IMC

Calculer l'IMC

Votre IMC est : 28.41 (Surpoids)

TP 2:

Créez un formulaire avec deux champs : un pour le montant en euros et un pour le taux de change, Lorsque l'utilisateur soumet le formulaire, affichez le montant converti en dirhams.

Conversion Euro en Dirham

Convertir

Montant converti : 107000.00 MAD

TP 3:

Créez un formulaire avec une liste de cases à cocher pour sélectionner plusieurs hobbies. Lorsque l'utilisateur soumet le formulaire, affichez les hobbies sélectionnés.

Sélectionnez vos Hobbies

☒ Lecture

☐ Musique

☐ Cuisine

☐ Photographie

☒ Jeux vidéo

☒ Sport

☒ Voyage

☐ Art

☐ Jardinage

☐ Danse

Soumettre

Hobbies sélectionnés : Lecture, Sport, Voyage, Jeux vidéo

TP 1:

un formulaire avec un champ pour saisir une adresse e-mail. Lorsque l'utilisateur soumet le formulaire, vérifiez si l'adresse e-mail est valide et affichez un message approprié.

Validation d'E-mail

Valider

L'e-mail est valide.

TP 2:

Créer une application de liste de tâches qui inclut la date et l'heure.

Liste de Tâches

Ajouter

cours Js (Due: 19/09/2024 23:19:00)

✓

🗑

Match de Foot (Due: 27/09/2024 02:26:00)

✓

🗑

TP 3:

Créez la calculatrice suivante :

7*9

7

8

9

/

4

5

6

*

1

2

3

-

C

0

=

+

TP 1:

Créez un formulaire qui compte le nombre de mots, de caractères et de phrases dans un texte que l'utilisateur saisit.

Compteur de Mots

salut les étudiants, comment allez vous, cava? avez vous réaliser votre travail?

Compter

Nombre de mots : 12

Nombre de caractères : 80

Nombre de phrases : 2

TP 2:

Créez un formulaire qui compte le nombre total de "J'aime" et de "Je n'aime pas".

Publication

J'aime

Je n'aime pas

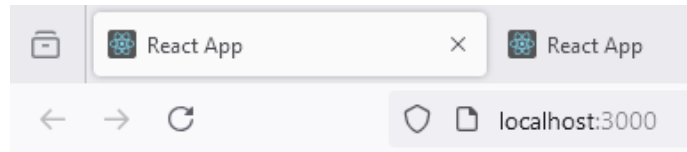
Totals

J'aime: 5

Je n'aime pas: 3

TP 1 :

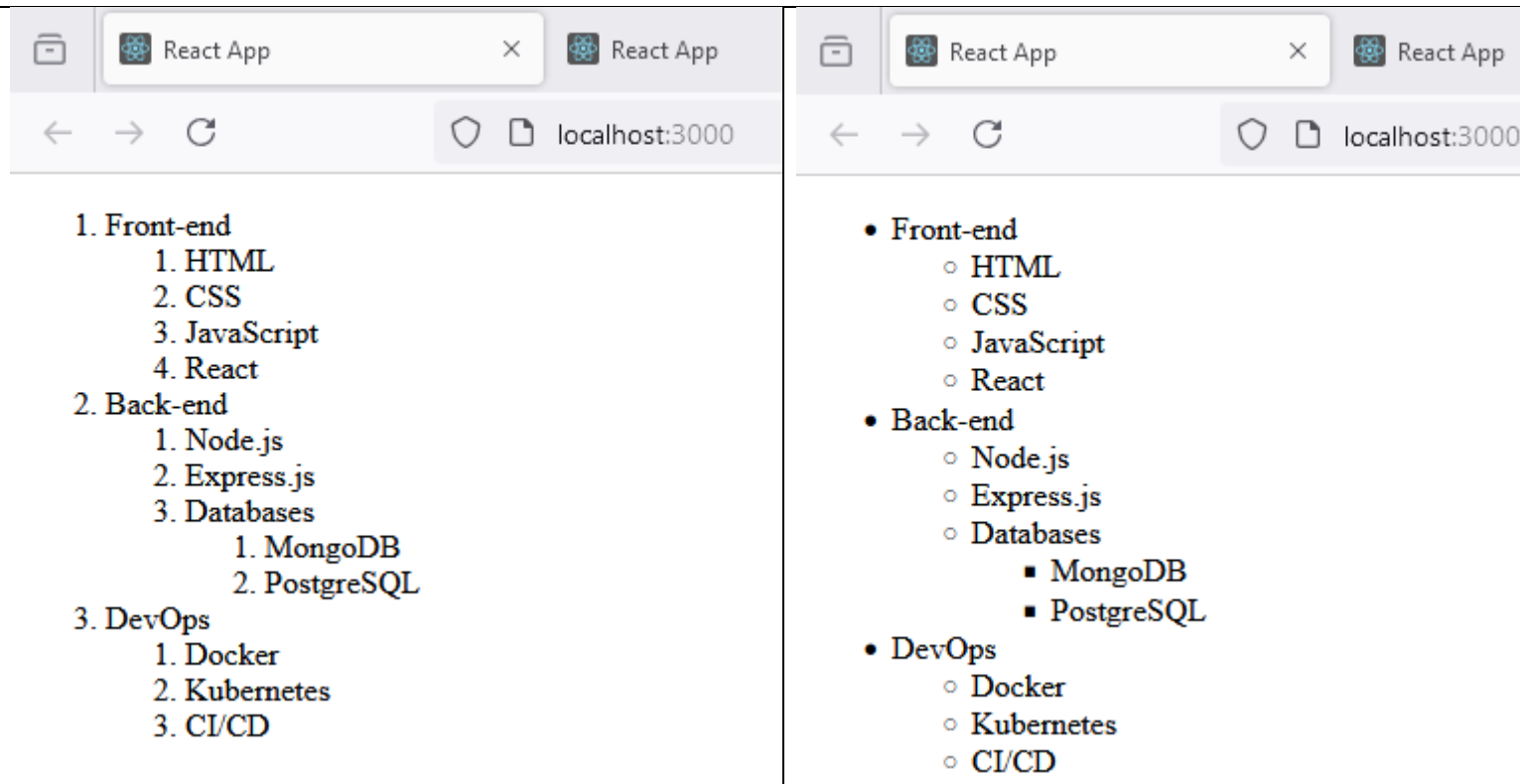
- 1- Installer le NodeJs et le Npm, puis créer votre première application React
- 2- Réaliser une page qui affiche « Hello, World »



Hello world!

TP 2 :

Réaliser une page suivante :



TP 3 :

Réaliser une page suivante :

Login

Email:

Password:

Login

TP 1 :

En utilisant JSX, réaliser la page web suivante :

Login

Email:

Password:

Login

TP 2 :

En utilisant JSX, réaliser la page web suivante :

Inscription à l'OFPPT

Nom Complet:

Email:

Téléphone:

Mot de passe:

Filière:

Diplôme Obtenu:

S'inscrire

Exercice 1 :

Convertissez les composants suivants écrits en JSX en utilisant *React.createElement* :

```
const Greeting = () => {
  return <h1>Bienvenue à l'OFPPT!</h1>;
};
```

```
const TodoList = () => {
  const todos = ['Apprendre React',
    'Pratiquer les exercices', 'Construire un projet'];
  return (
    <ul> {
      todos.map(todo => <li
        key={todo}>{todo}</li>)}
    </ul>
  );
};
```

```
const UserProfile = ({ name, age }) => {
  return (
    <div>
      <h2>{name}</h2>
      <p>Âge: {age}</p>
    </div>
  );
};
```

Exercice 2 :

Convertissez les composants suivants en JSX

```
const App = () => {
  return React.createElement(
    'div',
    null,
    'Bienvenue à l'OFPPT!');
};
```

```
const App = () => {
  return React.createElement(
    'div',
    null,
    React.createElement(
      'h1',
      null,
      'Mon Application'),
    React.createElement(
      'p',
      null,
      'Ceci est un paragraphe.'));
};
```

Exercice 1 :**Convertissez le composant suivant en JSX**

```
const ShoppingCart = () => {  
  return React.createElement(  
    'div',  
    null,  
    React.createElement(  
      'h2',  
      null,  
      'Mon Panier'),  
    React.createElement(  
      'ul',  
      null,  
        React.createElement(  
          'li',  
          null,  
          'Produit 1'),  
        React.createElement(  
          'li',  
          null,  
          'Produit 2'),  
        React.createElement(  
          'li',  
          null,  
          'Produit 3')  
        )  
      )  
  );  
};
```

Exercice 2 :**Réaliser la page suivante :**

Somme de Nombres (JSX)

Premier nombre	Deuxième nombre	Calculer
----------------	-----------------	----------

Somme de Nombres (Sans JSX)

Premier nombre	Deuxième nombre	Calculer
----------------	-----------------	----------



ROYAUME DU MAROC

مكتب التكوين المهني وإنعاش الشغل
Office de la Formation Professionnelle et de la Promotion du Travail

INSTITUT SPECIALISE DE TECHNOLOGIE APPLIQUEE DRIOUCH

Série N°13

Formation : DD 201

Prof : SAID MAHDAD

Module : développer en Front-end

Nom et prénom :

TP 1 :

Créer la page suivante qui contient 5 composants:



À propos de l'OFPPT

L'OFPPT est l'acteur principal de la formation professionnelle au Maroc. Il offre une large gamme de formations adaptées aux besoins du marché du travail pour aider les jeunes à développer leurs compétences et à améliorer leur employabilité.

Nos Filières

- Informatique
- Génie Civil
- Gestion
- Industrie
- Tourisme
- Artisanat

Pourquoi choisir la formation professionnelle ?

La formation professionnelle vous permet d'acquérir des compétences pratiques, d'être prêt pour le marché du travail, et de bénéficier d'une grande diversité de filières adaptées aux besoins actuels de l'économie.

Témoignages

"La formation m'a permis de décrocher un emploi stable."

- Fatima

"Grâce à l'OFPPT, j'ai acquis des compétences précieuses pour le marché du travail."

- Yassine

TP 2 :

Créer la page suivante :

Cours de React

[Introduction](#) [Composants](#) [Props](#) [État](#) [Hooks](#) [Routing](#)

Introduction à React

React est une bibliothèque JavaScript pour construire des interfaces utilisateur.

Les Composants

Les composants sont les blocs de construction de l'interface utilisateur en React.

Les Props

Les props (propriétés) sont des données que l'on passe aux composants.

© 2024 Cours de React

TP 1 :

Créez un composant SumCalculator qui prend trois nombres en tant que props (num1, num2, num3) et affiche leur somme dans un composant enfant DisplaySum.

Calculateur de Somme

La somme des trois nombres est : 30

TP 2 :

Créez un composant MaxCalculator qui prend trois nombres (num1, num2, num3) et affiche le maximum parmi ces trois dans un composant enfant DisplayMax.

Calculateur de Maximum

Le maximum est : 15

TP 3 :

Créez un composant VowelCounter qui prend une chaîne de caractères (text) et affiche le nombre de voyelles dans un composant enfant DisplayVowelCount.

Compteur de Voyelles

Le nombre de voyelles dans la chaîne est : 6

TP 4 :

Créez un composant UnitConverter qui prend une distance en kilomètres (kilometers) et affiche sa conversion en miles dans un composant enfant DisplayMiles.

Convertisseur d'Unités

La distance en miles est : 6.21

TP 5 :

Créez un composant PositiveChecker qui prend un nombre (number) et affiche si ce nombre est positif, négatif ou zéro dans un composant enfant DisplaySign.

Vérificateur de Signe

Le nombre -5 est négatif.

TP 1 :

Créez un composant fonctionnel sous le nom Book qui reçoit un objet book en props et affiche les détails du livre (titre, auteur et année de publication)

Détails du Livre

Titre : React In Action

Auteur : Mark Tielens Thomas

Année de publication : 2016

TP 2 :

Créez un composant de classe sous le nom Student qui reçoit un objet student en props et affiche les informations de l'étudiant (nom, âge, filière).

Détails de l'Étudiant

Nom : ABOU GHAZI

Âge : 22

Filière : Informatique

TP 3 :

Créez un composant fonctionnel StudentList qui reçoit un tableau students en props et affiche une liste d'étudiants. Chaque étudiant doit afficher son prénom et son âge.

Liste des Étudiants

- Prénom : Mohamed

Âge : 22 ans

- Prénom : Khadija

Âge : 24 ans

- Prénom : Ayman

Âge : 23 ans

TP 4 :

Créez un composant de classe sous le nom BookList qui reçoit un tableau books en props et affiche la liste des livres. Chaque livre doit afficher son titre et son auteur.

Liste des Livres

- Titre : Le Petit Prince

Auteur : Antoine de Saint-Exupéry

- Titre : 1984

Auteur : George Orwell

- Titre : L'Étranger

Auteur : Albert Camus

TP 1 :

Créez deux composants (de classe et fonctionnel) qui réalisent la somme de deux nombres saisis au clavier

Calcul de la Somme de Deux Nombres
Somme : 6**TP 2 :**

Créez un composant fonctionnel qui réalise la somme de deux nombres saisis au clavier après avoir cliqué un bouton

Calcul de la Somme de Deux Nombres

Somme : 6**TP 4 :**

Créez le composant qui affiche le formulaire suivant :

Formulaire de Contact

Nom

SAID MAHDAD

Email

SAID.MAHDAD@GMAIL.COM

Message

le message que je suis en train d'écrire

Soumettre

Données en temps réel :

Nom : SAID MAHDAD

Email : SAID.MAHDAD@GMAIL.COM

Message : le message que je suis en train d'écrire

Merci pour votre message, SAID MAHDAD !

Exercice 1 :

1- Quelle commande est utilisée pour créer un nouveau projet React ?

- ☐ npm create react-app my-app
- ☐ npx create-react-app my-app
- ☐ npm start react-app my-app
- ☐ npm install react-app my-app

2- Qu'est-ce que JSX ?

- ☐ Une bibliothèque JavaScript
- ☐ Une syntaxe de JavaScript permettant d'écrire des balises HTML dans le code JavaScript
- ☐ Un framework pour les interfaces utilisateur
- ☐ Une fonction JavaScript permettant d'intégrer CSS dans React

3- Quel mot-clé est utilisé pour créer un composant de classe en React ?

- ☐ constructor
- ☐ class
- ☐ function
- ☐ component

4- Comment appelle-t-on une fonction qui renvoie du JSX ?

- ☐ Un composant de classe
- ☐ Une fonction utilitaire
- ☐ Un composant fonctionnel
- ☐ Une méthode de classe

5- Comment passe-t-on des données d'un parent à un composant enfant ?

- ☐ state
- ☐ context
- ☐ props
- ☐ setState

6- Lequel des éléments suivants est utilisé pour définir les props d'un composant fonctionnel en React ?

- ☐ this.props
- ☐ state
- ☐ Les paramètres de la fonction
- ☐ render

7- Lequel des suivants est un avantage des composants fonctionnels sur les composants de classe ?

- ☐ Ils sont plus complexes
- ☐ Ils sont plus lents à exécuter
- ☐ Ils nécessitent moins de code
- ☐ Ils ne supportent pas les hooks

8- Quelle est la fonction principale de render() dans un composant de classe ?

- ☐ Gérer les événements utilisateur
- ☐ Déclencher un cycle de vie
- ☐ Retourner du JSX pour afficher l'interface utilisateur
- ☐ Charger les données externes

9- Quelle méthode est utilisée pour retourner du contenu dans un composant de classe ?

- ☐ render()
- ☐ show()
- ☐ display()
- ☐ return()

10. Quel est le rôle principal des props dans React ?

- ☐ Gérer l'état interne d'un composant
- ☐ Ajouter des événements de clics
- ☐ Passer des données du parent à l'enfant
- ☐ Gérer les appels d'API

Exercice 2 :

1- Décrivez le processus d'installation de React et de création d'une nouvelle application React.

.....

.....

.....

.....

.....

.....

.....

2- Qu'est-ce que JSX, et pourquoi est-il utilisé dans React ? Donnez un exemple de JSX.

.....

.....

.....

.....

.....

.....

.....

3- Quelle est la différence entre un composant de classe et un composant fonctionnel dans React ? Donnez un exemple de chaque.

4- Comment passe-t-on un objet comme prop à un composant enfant en React ? Montrez-le avec un exemple.

5- Expliquez comment on peut passer un tableau comme prop et comment il est utilisé dans le composant enfant. Donnez un exemple.

Exercice 3 :

1. Créez un composant fonctionnel appelé `WelcomeMessage` qui prend un prop `name` et affiche "Bienvenue, [name]".
2. Créez un composant de classe appelé `UserCard` qui prend deux props : `name` et `age`, et qui affiche un petit paragraphe contenant ces informations.
3. Créez un composant fonctionnel appelé `FruitList` qui prend un tableau de fruits en prop et qui affiche chaque fruit dans une liste non ordonnée (``).
4. Créez un composant de classe appelé `ProductCard` qui prend trois props d'un objet `Produit` : `name`, `price`, et `description`, et qui affiche ces informations dans une carte de produit stylée avec des balises HTML comme `<h2>`, `<p>`, etc.
5. Créez un composant fonctionnel appelé `GreetingList` qui prend un tableau de noms en prop et affiche un message de bienvenue pour chaque nom sous forme de liste d'éléments `<h3>`.

Projet : Application de Publication avec Réactions et Commentaires (Similaire à Facebook)

Objectif :

Créer une application React simple qui permet de publier du texte, d'ajouter des réactions ("J'aime", "Je n'aime pas") et de gérer les commentaires pour chaque publication. Le design de l'application doit être basique mais agréable, avec un affichage centré et des fonctionnalités interactives.

Fonctionnalités requises :

1. Zone de saisie de la publication :

- L'utilisateur doit pouvoir saisir un texte dans une zone de texte pour créer une publication.
- Après avoir cliqué sur le bouton "Publier", la publication doit s'afficher en dessous de la zone de saisie.
- La zone de saisie doit rester active pour permettre d'ajouter d'autres publications sans supprimer les précédentes.

2. Affichage de la publication :

- Chaque publication doit être affichée avec un texte aligné à gauche, et les publications doivent apparaître sous la zone de saisie.
- Les publications doivent inclure trois boutons :
 - **"J'aime"** : Incrémente le nombre de "J'aime".
 - **"Je n'aime pas"** : Incrémente le nombre de "Je n'aime pas".
 - **"Commenter"** : Permet d'afficher une zone de texte pour écrire des commentaires.

3. Commentaires :

- En cliquant sur "Commenter", une zone de saisie doit apparaître pour permettre à l'utilisateur de saisir un commentaire.
- Les commentaires doivent être publiés après avoir cliqué sur le bouton "Ajouter commentaire".
- Les commentaires doivent être affichés en dessous de la publication .

Technologies :

- **React.js** : Utiliser React pour la gestion des composants et des états.
- **CSS** : Utiliser un fichier CSS pour styliser l'application.

Captures d'écran :

1)-



2)-

Écrivez votre publication ici...

Publier

c'est ma 1ere publication

J'aime (0)

Je n'aime pas (0)

Commenter

3)-

Écrivez votre publication ici...

Publier

c'est ma 1ere publication

J'aime (6)

Je n'aime pas (3)

Commenter

c'est mon 1er commentaire

c'est mon 2eme commentaire

Ajouter commentaire

Projet : Application de Gestion de Tâches (Todo List)

Objectif :

Créer une application React qui permet de gérer des tâches (todo list) où l'utilisateur peut ajouter, marquer comme terminée, supprimer, et filtrer des tâches. L'interface sera simple mais interactive, avec des actions pour organiser les tâches quotidiennes.

Fonctionnalités Requises :

1. Ajouter une tâche :

- L'utilisateur doit pouvoir saisir une tâche dans une zone de texte.
- Un bouton "**Ajouter**" permet d'ajouter cette tâche à la liste.
- Une fois ajoutée, la tâche est affichée dans une liste avec un bouton pour la marquer comme terminée ou la supprimer.

2. Marquer une tâche comme terminée :

- Chaque tâche de la liste doit avoir un bouton "**Terminer**" qui, lorsqu'il est cliqué, change l'apparence de la tâche (par exemple, texte barré ou couleur différente) pour indiquer qu'elle est terminée.

3. Supprimer une tâche :

- Chaque tâche doit également avoir un bouton "**Supprimer**" pour retirer la tâche de la liste.

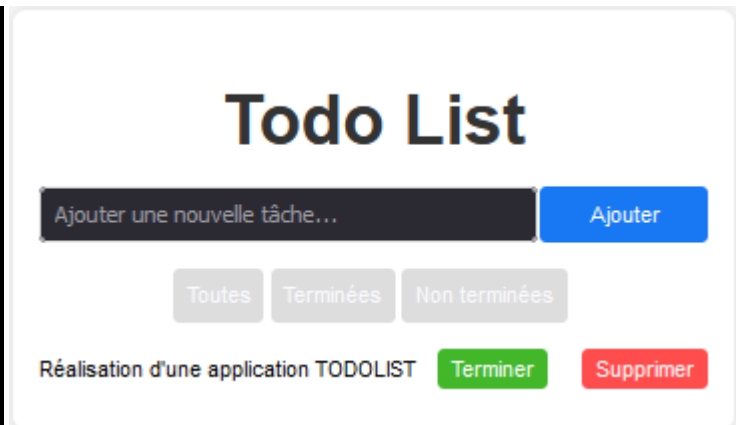
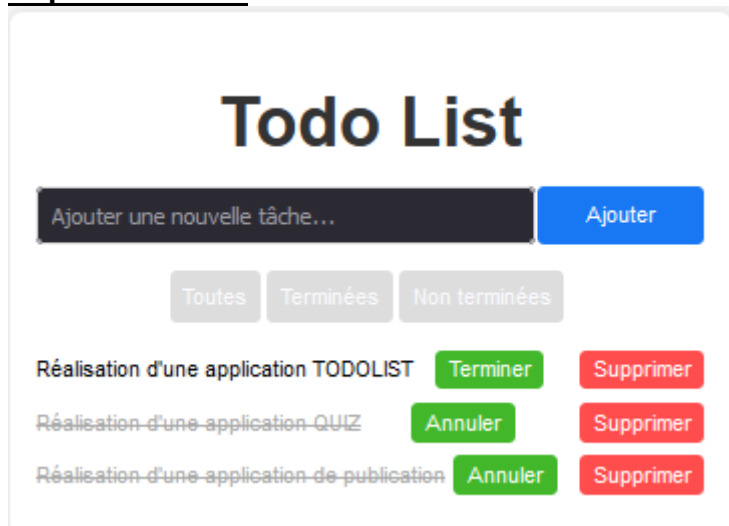
4. Filtrer les tâches :

- Ajouter des boutons de filtrage pour afficher :
 - Toutes les tâches.
 - Les tâches terminées.
 - Les tâches non terminées.

Technologies à utiliser :

- **React.js** : Utilisation des composants et des hooks pour gérer les tâches.
- **CSS** : Styliser l'application pour rendre l'interface agréable.

Captures d'écran



Fonctionnalités supplémentaires (optionnel) :

- **Édition d'une tâche** : Permettre à l'utilisateur de modifier une tâche existante.
- **Priorités** : Ajouter une fonctionnalité pour assigner une priorité (basse, moyenne, haute) à chaque tâche et les organiser en fonction de la priorité.
- **Affichage des statistiques** : Montrer combien de tâches qui sont terminées et combien restent à faire.

TP 1 :

Créer un composant qui affiche un compteur. Le bouton "Incrémenter" augmente le compteur de 1 à chaque clic

Compteur : 5

Incrémenter

TP 2 :

Créer un composant qui permet à l'utilisateur de saisir du texte dans un champ input et affiche ce texte sous l'input en temps réel.

Cv etudiants?

Texte saisi : Cv etudiants?

TP 3 :

Créer un composant avec deux boutons, "Rouge" et "Bleu". Chaque bouton change la couleur de fond de la page en conséquence

Rouge

Bleu

TP 4 :

Créer un composant avec un bouton qui change son état entre "Visible" et "Invisible", ce qui permet de cacher ou montrer un texte

Cacher

Montrer

Le texte est visible.

TP 5 :

Créer un compteur qui permet de l'incrémenter et de le réinitialiser à zéro via deux boutons séparés.

Compteur : 0

Incrémenter

Réinitialiser

TP 1 :

Créer un formulaire avec deux champs : "Nom" et "Email". Affiche un message d'erreur si l'un des champs est vide lorsque l'utilisateur clique sur "Soumettre".

Tous les champs doivent être remplis.

TP 2 :

Créer un formulaire avec une liste déroulante. L'utilisateur peut choisir une couleur, et la couleur de fond de la page change en fonction de l'option sélectionnée.

Couleur choisie : white

TP 3 :

Créer une simple application de to-do-list où l'utilisateur peut ajouter des tâches. Chaque tâche saisie s'affiche dans une liste.

- useState
- useEffect

TP 4 :

Créer un composant où l'utilisateur saisit son année de naissance et le composant affiche l'âge calculé en fonction de l'année actuelle.

Votre âge est : 35

TP 5 :

Crée un composant qui permet à l'utilisateur de choisir une langue dans un menu déroulant (par exemple : "Français", "Anglais", "Espagnol") et qui affiche un message de bienvenue dans la langue sélectionnée

Bienvenido

TP 1 :

Créer un composant qui affiche "Bienvenue!" une seule fois, au moment où le composant se charge.

TP 2 :

Créer un compteur avec un bouton pour augmenter la valeur. Utilise useEffect pour afficher la valeur du compteur dans la console chaque fois qu'elle change.

TP 3 :

Créer un compteur avec un bouton pour augmenter la valeur. Utilise useEffect pour changer le titre de la page à chaque fois que le compteur est incrémenté



TP 4 :

Créer un composant qui récupère des données depuis une API au chargement et les affiche.

```
1 import React, { useState, useEffect } from 'react';
2 function DataFetcher() {
3   const [data, setData] = useState(null);
4   useEffect(() => {
5     fetch('https://api.example.com/data')
6       .then(response => response.json())
7       .then(data => setData(data));
8   }, []);
9   return (
10    <div>
11      <p>Données : {JSON.stringify(data)}</p>
12    </div>
13  );
14 }
15 export default DataFetcher;
```

TP 5 :

Crée un composant qui charge des données depuis un fichier JSON externe et affiche chaque élément dans une liste.

```
1 import React, { useState, useEffect } from 'react';
2 function DataList() {
3   const [data, setData] = useState([]);
4   useEffect(() => {
5     fetch('/path/to/data.json')
6       .then(response => response.json())
7       .then(data => setData(data))
8       .catch(error => console.error('Erreur de chargement des données', error));
9   }, []);
10   return (
11     <ul>
12       {data.map((item, index) => (
13         <li key={index}>{item.name}</li>
14       ))}
15     </ul>
16   );
17 }
18 export default DataList;
```


Projet : Application de Quiz sur React avec JSON

Objectif :

Créer une application React qui propose une série de questions à choix multiples à l'utilisateur. Les questions seront stockées dans un fichier JSON et chargées dynamiquement dans l'application. À la fin du quiz, un score basé sur les réponses fournies sera affiché à l'utilisateur. Le design sera basique mais agréable, avec des fonctionnalités interactives.

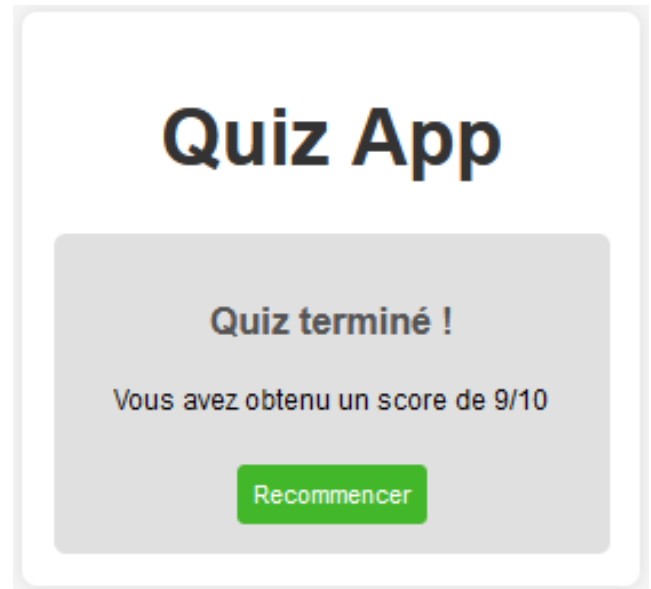
Fonctionnalités Requises :

- Affichage des questions :**
 - Chaque question doit être affichée avec des réponses à choix multiples (au moins 4 réponses par question).
 - L'utilisateur doit pouvoir sélectionner une réponse pour chaque question.
 - Les questions seront chargées depuis un fichier JSON local (pas d'API).
- Calcul du score :**
 - Lorsque l'utilisateur a terminé le quiz, son score est calculé en fonction des réponses correctes fournies.
 - Le score final doit être affiché avec un message récapitulatif.
 - Un bouton "**Recommencer**" permet de relancer le quiz avec une réinitialisation du score.

Technologies à utiliser :

- React.js** : Gestion des composants et des états pour afficher les questions et gérer les réponses.
- JavaScript** : Pour la logique de vérification des réponses et du calcul du score.
- CSS** : Pour styliser l'application (mise en page, couleurs, animations de survol).
- JSON** : Le quiz doit utiliser un fichier *json* qui contient les questions, les options de réponse, et la bonne réponse.

Captures d'écran :



Fonctionnalités supplémentaires (optionnel) :

- Chronomètre** : Limiter le temps pour répondre à chaque question ou pour l'ensemble du quiz.
- Indicateur de progression** : Afficher la progression du quiz, par exemple "Question 3 sur 10".
- Feedback sur les réponses** : Après avoir soumis une réponse, afficher si elle est correcte ou non avant de passer à la suivante.

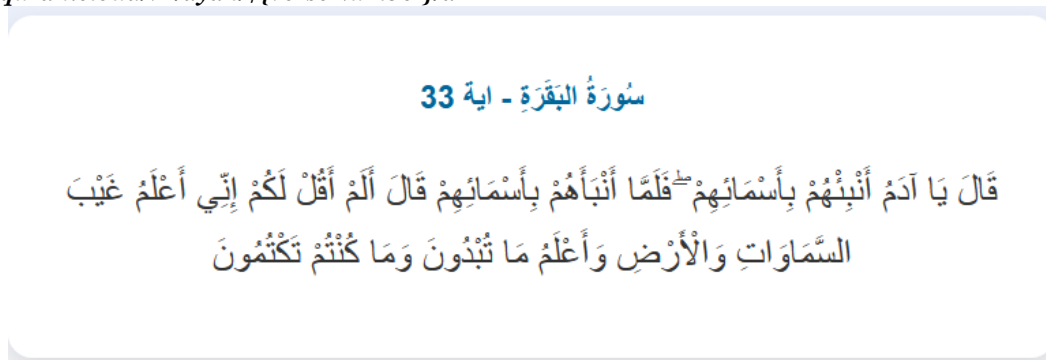
TP 1 :

Créer un composant qui affiche les données météo de la ville de Driouch

**TP 2 :**

Créer un composant qui retourne des verset de Quran à partir de l'API :

API : [https://api.alquran.cloud/v1/ayah/\\${verseNumber}/ar](https://api.alquran.cloud/v1/ayah/${verseNumber}/ar)

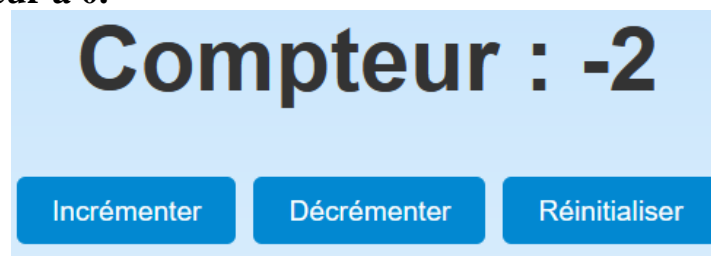
**TP 3 :**

Créez un composant React qui affiche un compteur avec trois boutons pour :

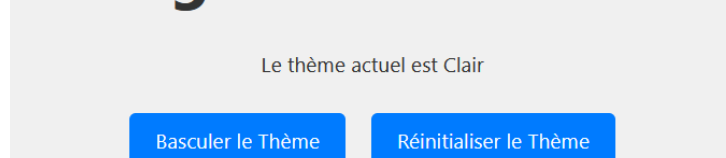
Incrémenter le compteur de 1.

Décrémenter le compteur de 1.

Réinitialiser le compteur à 0.

**TP 4 :**

Utilisez useReducer pour basculer entre un thème clair et un thème sombre, Ajoutez une option pour réinitialiser le thème à l'état par défaut (clair).

Changement de Thème

Projet : Application de Gestion de Tâches (Todo List)

Objectif :

Créer une application React qui permet de gérer des tâches (todo list) où l'utilisateur peut ajouter, marquer comme terminée, supprimer, et filtrer des tâches. L'interface sera simple mais interactive, avec des actions pour organiser les tâches quotidiennes.

Fonctionnalités Requises :

5. **Ajouter une tâche :**

- L'utilisateur doit pouvoir saisir une tâche dans une zone de texte.
- Un bouton "**Ajouter**" permet d'ajouter cette tâche à la liste.
- Une fois ajoutée, la tâche est affichée dans une liste avec un bouton pour la marquer comme terminée ou la supprimer.

6. **Marquer une tâche comme terminée :**

- Chaque tâche de la liste doit avoir un bouton "**Terminer**" qui, lorsqu'il est cliqué, change l'apparence de la tâche (par exemple, texte barré ou couleur différente) pour indiquer qu'elle est terminée.

7. **Supprimer une tâche :**

- Chaque tâche doit également avoir un bouton "**Supprimer**" pour retirer la tâche de la liste.

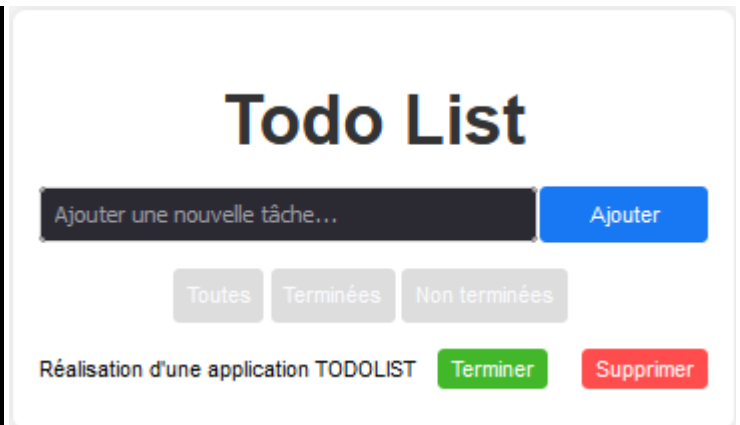
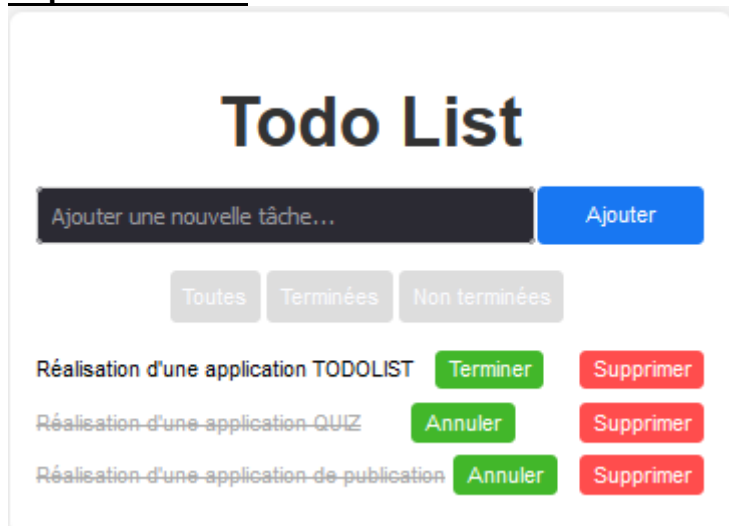
8. **Filtrer les tâches :**

- Ajouter des boutons de filtrage pour afficher :
 - Toutes les tâches.
 - Les tâches terminées.
 - Les tâches non terminées.

Technologies à utiliser :

- **React.js** : Utilisation des composants et des hooks pour gérer les tâches.
- **CSS** : Styliser l'application pour rendre l'interface agréable.

Captures d'écran



Fonctionnalités supplémentaires (optionnel) :

- **Édition d'une tâche** : Permettre à l'utilisateur de modifier une tâche existante.
- **Priorités** : Ajouter une fonctionnalité pour assigner une priorité (basse, moyenne, haute) à chaque tâche et les organiser en fonction de la priorité.
- **Affichage des statistiques** : Montrer combien de tâches qui sont terminées et combien restent à faire.

TP 1 :

En utilisant Usereducer créer le composant suivant :

Système de Vote



Total de votes : 3

Pourcentage "Pour" : 66.67%

Pourcentage "Contre" : 33.33%

Vote "Pour"

Vote "Contre"

Réinitialiser les Votes

TP 2 :

Créer un composant qui retourne les données Météo suivantes(en utilisant Axios)

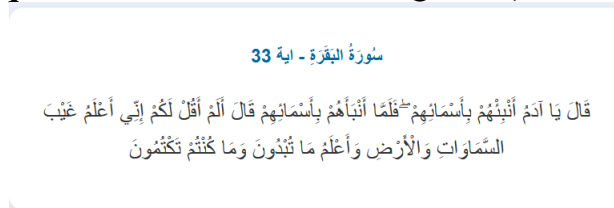
Météo actuelle :

- Température : 22.6°C
- Vitesse du vent : 3.7 km/h

API : [https://api.open-meteo.com/v1/forecast?latitude=\\${10}&longitude=\\${10}¤t_weather=true](https://api.open-meteo.com/v1/forecast?latitude=${10}&longitude=${10}¤t_weather=true)

TP 3 :

Créer un composant React qui affiche les versets du Quran(en utilisant Axios)



سُورَةُ الْبَقَرَةِ - آيَة 33

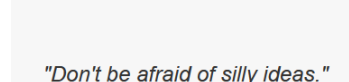
قَالَ يَا آدَمُ أَنْبِئْهُمْ بِأَسْمَائِهِمْ طَلَمَا أَنْبَأْتُهُمْ بِأَسْمَائِهِمْ قَالَ أَلَمْ أَقُلْ لَكُمْ إِنِّي أَعْلَمُ الْغَيْبِ السَّمَاوَاتِ وَالْأَرْضِ وَأَعْلَمُ مَا تُنْبِئُونَ وَمَا كُنْتُمْ تَكْتُمُونَ

API : [https://api.alquran.cloud/v1/ayah/\\${verseNumber}/ar](https://api.alquran.cloud/v1/ayah/${verseNumber}/ar)

TP 4 :

Créer un composant React qui affiche le conseil du jour(en utilisant Axios)

Conseil du Jour



"Don't be afraid of silly ideas."

API : <https://api.adviceslip.com/advice>

TP 5 :

Créer un composant React qui affiche les information des pays(en utilisant Axios)

Morocco

Capitale : Rabat

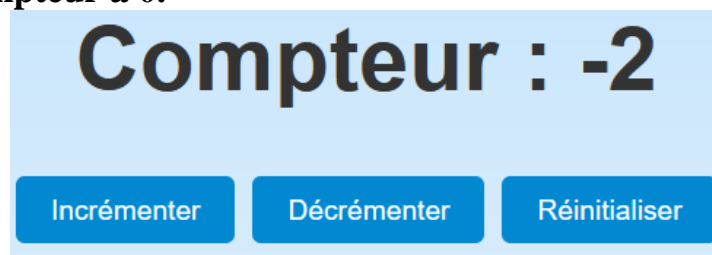


API : <https://restcountries.com/v3.1/all>

TP 1 :

Contexte : développer une application simple en utilisant React et Redux pour gérer l'état d'un compteur, avec trois boutons pour :

- Incrémenter le compteur de 1.
- Décrémenter le compteur de 1.
- Réinitialiser le compteur à 0.



TP 2 :

Contexte : développer une application simple en utilisant React et Redux pour gérer un message. Cette application permettra d'afficher, de modifier et de réinitialiser ce message.

Objectifs :

1. Afficher le message actuel à l'écran.
2. Modifier le message en saisissant un texte dans un champ d'entrée.
3. Effacer le message en réinitialisant son contenu à une chaîne vide.

TP 1 :**Description :**

Créez une application pour gérer une liste d'utilisateurs avec les fonctionnalités suivantes :

- Ajouter un utilisateur
- Supprimer un utilisateur
- Afficher les détails d'un utilisateur

Gestion des Utilisateurs

<input type="text" value="Nom"/>	<input type="text" value="Email"/>	<input type="button" value="Ajouter Utilisateur"/>
----------------------------------	------------------------------------	--

Étapes :

1. Créez un **reducer** qui gère une liste d'utilisateurs
2. Implémentez trois actions : `ADD_USER`, `REMOVE_USER`, et `SELECT_USER`.
3. Affichez les utilisateurs et leurs détails dans des composants connectés à Redux.

Liste des Utilisateurs

- MAHDAD (MAHDAD.SAID@GMAIL.COM)

Détails de l'Utilisateur

Nom: MAHDAD

Email: MAHDAD.SAID@GMAIL.COM

TP 2 :**Description:**

Développez une application où les utilisateurs peuvent :

- Ajouter des articles au panier
- Supprimer des articles du panier
- Afficher le total des articles

Étapes :

1. Créez un **reducer** pour gérer le panier.
2. Configurez un **store** et connectez-le à un composant React.
3. Gérez les actions `ADD_TO_CART` et `REMOVE_FROM_CART`.

Panier d'Achat

<input type="text" value="Nom de l'article"/>	<input type="text" value="Prix de l'article"/>	<input type="button" value="Ajouter au Panier"/>
---	--	--

Articles dans le Panier

- pomme - \$5.00
- orange - \$3.00

Total : \$8.00