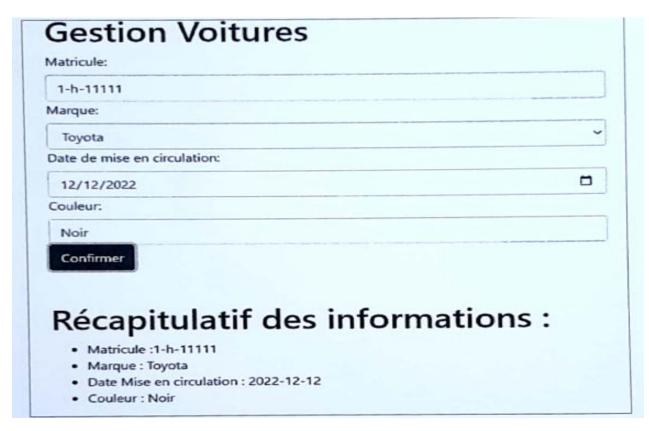
## Exercice 1:

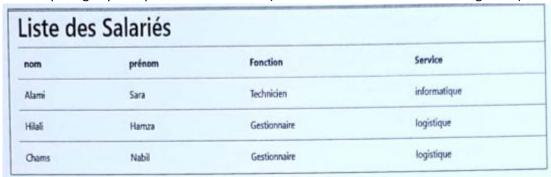
1. Dans le composant principal SalarieContext.js , initialiser la variable d'état salaries en appelant l'api du backend suivant :

```
Méthode HTTP
URL de l'api
                           http://localhost:8000/salaries
Resultat
                           1
                              "_id" : "s1",
                             "nomsal" : "Alami",
                             "prenomsal" : "Sara",
                             "fonction" : "Technicien",
                              "service" :{
                                 "codeser" : "1",
                                 "nomser" : "informatique"
                                      }
                            },
```

- 2. Dans le composant app.js , créer la liste des routes en utilisant React-Router-Dom
  - Composant1.js
  - Composant2 .js
- 3. Réaliser un composant composant1.js permettant à l'utilisateur de saisir les informations d'une voiture, lors du clic sur le bouton confirmer on doit afficher le récapitulatif des informations sur la même page comme suit :



4. Dans le composant2.js, afficher les informations des salariés stockées dans la variable d'état salaries partagée par le provider du context (l'utilisation de useContext est obligatoire)



5. Dans le composant2.js, ajouter l'option rechercher par service



## Exercice 2:

Soit le fichier data.js contenant une liste des experts comme suit :

```
const expertsData = [
    "id": 1,
     "nom_complet": "DUPONT Jean",
    "événements": [
         "thème": "Développement Web",
         "date_debut": "2024-03-02", "date_fin": "2024-03-15",
         "description": "Introduction au développement web moderne",
         "cout journalier": 500,
         "durée": 13
         "thème": "Applications Mobiles",
         "date debut": "2024-04-01",
         "date_fin": "2024-04-15",
         "description": "Conception et développement d'applications mobiles",
         "cout_journalier": 700,
         "durée": 14
     ]
  },
export default expertsData;
```

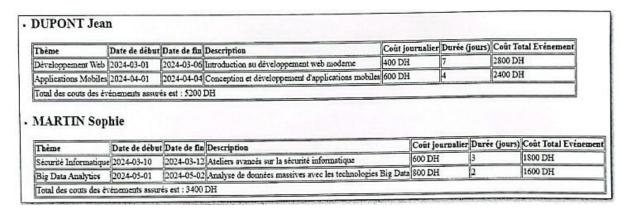
1. Créer le composant Evenements.js qui reçoit la liste d'événements du fichier expertData.js, d'un expert donné en tant que props et les affiche dans une table HTML, en incluant le coût total pour chaque événement dans chaque ligne, et en affichant le coût total de l'ensemble des événements en bas de la table sous la forme suivante :

Thème	Date de début	Date de fin	Description	Coût journalier	Durée (jours)	Coût Total Evénement
Développement Web	2024-03-01	2024-03-06	Introduction au développement web moderne	400 DH	7	2800 DH
Applications Mobiles	2024-04-01	2024-04-04	Conception et développement d'applications mobiles	600 DH	4	2400 DH

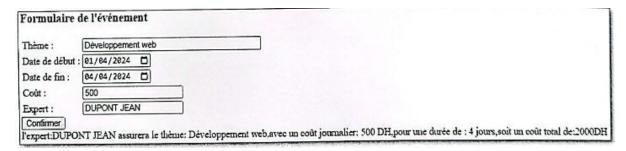
2. Créer le composant Expert.js qui reçoit un expert en tant que props et qui l'affiche sous forme d'une balise () (nom complet de l'expert et sa liste des événements) sous la forme suivante :

Thème	Date de début	Date de fin	Description	Cout journalier	Durée (jours)	Coût Total Evénemen
Développement Web	-			400 DH	-	2800 DH
Applications Mobiles		2024-04-04	Conception et développement d'applications mobiles	600 DH	4	2400 DH

3. Créer le composant Experts1 qui importe la constante « expertsData » du fichier data.js et affiche la liste des experts sous forme d'une liste à puce () en utilisant les composants précédents



4. Créer un composant nommé Formulaire.js permettant de saisir le thème de l'événement ,sa date de début, sa date de fin, son coût et un expert. Le clic sur le bouton confirmer affiche les informations saisies de la manière suivante :



5. En appelant l'api du Backend suivant :

Méthode HTTP	Get
URL de l'api	http://localhost:8000/experts2
Résultat	[,

```
"id": 1,
  "nom_complet": "DUPONT Jean",
  "événements": [
      "thème": "Développement Web",
      "date_debut": "2024-03-01",
      "date_fin": "2024-03-06"
      "description": "Introduction au développement web moderne",
      "cout_journalier": 400,
      "durée": 7
      "thème": "Applications Mobiles",
      "date_debut": "2024-04-01",
      "date_fin": "2024-04-04",
      "description": "Conception et développement d'applications mobiles",
      "cout_journalier": 600,
"durée": 4
},
```

Créer le composant Experts2 qui permet de récupérer les données de l'api et l'afficher sous forme d'une liste à puces ()

6. Créer le composant App.js incluant les chemins et les routes vers les composants : Formulaire.js (/formulaire),Experts1.js (/Experts1) et Experts2.js (/Experts2)