

Voici la version corrigée de votre application, où l'affichage de la liste des équipements est fait dans un **tableau Bootstrap**. J'ai également ajouté un design avec Bootstrap pour que l'application soit visuellement attrayante.

---

## Étape 1 : Installer Bootstrap

### 1. Ajoutez Bootstrap à votre projet :

- Installez Bootstrap via npm :

```
npm install bootstrap
```

- Importez Bootstrap dans `src/index.js` :

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

---

## Étape 2 : Code corrigé et amélioré

Voici le code complet corrigé avec Bootstrap pour afficher la liste dans un tableau.

### App.js

```
import React, { useState } from 'react';
import FormEquip from './components/FormEquip';
import ListEquip from './components/ListEquip';

function App() {
  const [equipments, setEquipments] = useState([]);

  // Ajouter un équipement
  const handleAddEquipment = (equipment) => {
    setEquipments([...equipments, equipment]);
  };

  // Marquer un équipement comme "A réformer"
  const handleReformEquipment = (index) => {
    const updatedEquipments = [...equipments];
    updatedEquipments[index].isReform = !updatedEquipments[index].isReform;
    setEquipments(updatedEquipments);
  };

  // Supprimer un équipement
  const handleDeleteEquipment = (index) => {
    setEquipments(equipments.filter((_, i) => i !== index));
  };
}
```

```
return (
  <div className="container mt-4">
    <h1 className="text-center mb-4">Test React: Gérer les équipements</h1>
    <FormEquip onAdd={handleAddEquipment} />
    <ListEquip
      equipments={equipments}
      onReform={handleReformEquipment}
      onDelete={handleDeleteEquipment}
    />
  </div>
);
}

export default App;
```

---

### FormEquip.js

```
import React, { useState } from 'react';

function FormEquip({ onAdd }) {
  const [name, setName] = useState('');
  const [date, setDate] = useState('');

  const handleSubmit = (e) => {
    e.preventDefault();
    if (!name) {
      alert('Le nom de l'équipement est obligatoire.');
```

```
      return;
    }
    onAdd({ name, date, isReform: false });
    setName('');
    setDate('');
  };

  return (
    <form onSubmit={handleSubmit} className="mb-4">
      <div className="mb-3">
        <label className="form-label">Nom de l'équipement :</label>
        <input
          type="text"
          className="form-control"
          value={name}
          onChange={(e) => setName(e.target.value)}
          required
        />
      </div>
      <div className="mb-3">
        <label className="form-label">Date d'achat :</label>
        <input
          type="date"
```

```

        className="form-control"
        value={date}
        onChange={(e) => setDate(e.target.value)}
      />
    </div>
    <button type="submit" className="btn btn-primary">
      Ajouter
    </button>
  </form>
);
}

export default FormEquip;

```

### ListEquip.js

```

import React from 'react';

function ListEquip({ equipments, onReform, onDelete }) {
  if (equipments.length === 0) {
    return <p className="text-center">Aucun équipement ajouté pour le moment.</p>;
  }

  return (
    <table className="table table-bordered table-striped">
      <thead className="table-dark">
        <tr>
          <th>#</th>
          <th>Nom de l'équipement</th>
          <th>Date d'achat</th>
          <th>Statut</th>
          <th>Actions</th>
        </tr>
      </thead>
      <tbody>
        {equipments.map((equip, index) => (
          <tr key={index} className={equip.isReform ? 'table-warning' : ''}>
            <td>{index + 1}</td>
            <td>{equip.name}</td>
            <td>{equip.date || 'Pas de date'}</td>
            <td>{equip.isReform ? 'À réformer' : 'Actif'}</td>
            <td>
              <button
                className={`btn btn-${equip.isReform ? 'secondary' : 'warning'}
me-2` }
                onClick={() => onReform(index)}
              >
                {equip.isReform ? 'Annuler' : 'À réformer'}
              </button>
              <button className="btn btn-danger" onClick={() => onDelete(index)}>

```

```
                Supprimer
            </button>
        </td>
    </tr>
</tbody>
</table>
);
}

export default ListEquip;
```

---

## Étape 3 : Ajouter un style Bootstrap

Bootstrap s'occupe principalement de la mise en page. Voici comment les composants sont stylés automatiquement :

1. Les **formulaires** utilisent les classes `form-label`, `form-control`, et `btn`.
2. Les **tableaux** sont stylés avec `table`, `table-bordered`, et `table-striped`.
3. Les **boutons** utilisent `btn-primary`, `btn-warning`, et `btn-danger`.

---

## Aperçu du design

1. **Formulaire en haut** pour ajouter des équipements.
2. **Tableau centré** listant tous les équipements avec :
  - Numéro d'ordre.
  - Nom.
  - Date.
  - Statut (Actif ou À réformer).
  - Boutons pour "À réformer" et "Supprimer".
3. **Couleurs différenciées** :
  - Les équipements marqués "À réformer" sont en **jaune clair** grâce à la classe `table-warning`.

---

Si vous rencontrez des erreurs ou souhaitez ajouter d'autres fonctionnalités, dites-moi! 😊