

Cours MERN - Semaine 7

Maîtriser les Hooks Essentiels React

`useState, useEffect et useReducer`

Abdelweheb GUEDDES & Mohamed Ben Jazia / Ecole Polytechnique Sousse

8 novembre 2025

Table des matières

1 Objectifs Pédagogiques	2
2 Partie 1 : Concepts Essentiels (30 min)	2
2.1 Hook 1 : <code>useState</code> - Ajouter de la mémoire	2
2.1.1 L'Immutabilité : La Règle d'Or	2
2.2 Hook 2 : <code>useEffect</code> - Les effets de bord	3
2.3 Hook 3 : <code>useReducer</code> - État complexe	3
3 Partie 2 : Projets Pratiques (2h30)	5
3.1 Projet 1 : Gestionnaire de Tâches avec <code>useReducer</code> (45 min)	5
3.1.1 Étape 1 : Créer le Reducer	5
3.1.2 Étape 2 : Le Composant Principal	6
3.2 Projet 2 : Liste de Courses avec Timer (45 min)	11
3.2.1 Code Complet	11
3.3 Projet 3 : Blog Interactif avec Tri et Recherche (60 min)	18
3.3.1 Code Complet de l'Application	18
4 Récapitatif et Concepts Clés	26
5 Travail à Rendre	27
6 Ressources	27
6.1 Documentation Officielle	27
6.2 Outils	28

1 Objectifs Pédagogiques

L'objectif central de cette séance est de maîtriser les trois hooks fondamentaux de React à travers des projets pratiques concrets.

À la fin de cette séance, vous serez capable de :

- Utiliser useState pour gérer l'état local simple et complexe
- Maîtriser useEffect pour les effets de bord (API, timers, localStorage)
- Implémenter useReducer pour organiser la logique d'état complexe
- Appliquer l'immutabilité correctement dans toutes les mises à jour d'état
- Créer des applications interactives complètes avec persistance de données

2 Partie 1 : Concepts Essentiels (30 min)

2.1 Hook 1 : useState - Ajouter de la mémoire

useState permet d'ajouter de l'état à un composant fonctionnel. React conserve cet état entre les rendus.

Syntaxe de base :

```
1 const [ state , setState ] = useState ( initialValue ) ;
2
3 // Exemples
4 const [ count , setCount ] = useState ( 0 ) ;
5 const [ name , setName ] = useState ( '' ) ;
6 const [ items , setItems ] = useState ( [ ] ) ;
7 const [ user , setUser ] = useState ( { name : '' , age : 0 } ) ;
```

2.1.1 L'Immutabilité : La Règle d'Or

En React, l'état ne doit **jamais** être modifié directement.

INCORRECT - Mutation directe :

```
1 // Objets
2 user . name = " Bob " ;
3 setUser ( user ) ; // React ne détecte pas le changement
4
5 // Tableaux
6 items . push ( newItem ) ;
7 setItems ( items ) ; // React ne détecte pas le changement
```

CORRECT - Immutabilité :

```
1 // Objets
2 setUser ( { ... user , name : " Bob " } ) ; // Nouvel objet
3
4 // Tableaux
5 setItems ( [ ... items , newItem ] ) ; // Nouveau tableau
```

Opérations immutables sur les tableaux :

Opération	Méthode immutable
Ajouter	[...arr, newItem]
Supprimer	arr.filter((_,i) => i !== index)
Modifier	arr.map((item,i) => i === index ? newItem : item)

2.2 Hook 2 : useEffect - Les effets de bord

`useEffect` permet d'exécuter du code après le rendu pour synchroniser avec des systèmes externes.

Syntaxe :

```

1  useEffect(() => {
2      // Code à exécuter
3
4      return () => {
5          // Nettoyage (optionnel)
6      };
7 }, [ dependencies ]);
```

Les trois cas d'usage :

1. Une seule fois au montage : `useEffect(() => {...}, [])`
2. Quand une valeur change : `useEffect(() => {...}, [value])`
3. À chaque rendu : `useEffect(() => {...})` (rarement utilisé)

Exemples courants :

- Charger des données depuis une API
- Créer un timer ou intervalle
- Sauvegarder dans localStorage
- S'abonner à des événements

2.3 Hook 3 : useReducer - État complexe

Quand l'état devient complexe (plusieurs sous-valeurs liées), `useReducer` offre une meilleure organisation.

Syntaxe :

```

1  const [state, dispatch] = useReducer(reducer, initialState);
2
3 // Fonction reducer
4 function reducer(state, action) {
5     switch (action.type) {
6         case 'ACTION_NAME':
7             return {...state, /* modifications */};
```

```
8     default :
9         return state ;
10    }
11 }
12
13 // Utilisation
14 dispatch({ type: 'ACTION_NAME', payload: value });
```

Quand utiliser useReducer ?

- Plusieurs sous-valeurs d'état liées
- Logique de mise à jour complexe
- Transitions d'état multiples

3 Partie 2 : Projets Pratiques (2h30)

3.1 Projet 1 : Gestionnaire de Tâches avec useReducer (45 min)

Objectif : Créer une Todo List complète utilisant `useReducer` pour gérer l'état.

Fonctionnalités :

- Ajouter des tâches
- Marquer comme complétées
- Supprimer des tâches
- Filtrer (toutes / actives / complétées)
- Persistance avec localStorage

3.1.1 Étape 1 : Crée le Reducer

```

1  export const initialState = {
2      todos: [] ,
3      filter: 'all' // 'all', 'active', 'completed'
4  };
5
6  export function todoReducer(state, action) {
7      switch (action.type) {
8          case 'ADD_TODO':
9              return {
10                  ...state,
11                  todos: [
12                      {
13                          id: Date.now(),
14                          text: action.payload,
15                          completed: false,
16                          createdAt: new Date().toISOString()
17                      },
18                      ...state.todos
19                  ]
20              };
21
22          case 'TOGGLE_TODO':
23              return {
24                  ...state,
25                  todos: state.todos.map(todo =>
26                      todo.id === action.payload
27                          ? {...todo, completed: !todo.completed}
28                          : todo
29                  )
30              };
31
32          case 'DELETE_TODO':
33              return {
34                  ...state,
35                  todos: state.todos.filter(todo =>

```

```

36             todo.id !== action.payload
37         )
38     };
39
40     case 'SET_FILTER':
41         return {
42             ...state,
43             filter: action.payload
44         };
45
46     case 'LOAD_TODOS':
47         return {
48             ...state,
49             todos: action.payload
50         };
51
52     default:
53         return state;
54     }
55 }
```

Listing 1 – src/reducers/todoReducer.js

3.1.2 Étape 2 : Le Composant Principal

```

1 import { useReducer, useState, useEffect } from 'react';
2 import { todoReducer, initialState } from './reducers/todoReducer';
3
4 function App() {
5     const [state, dispatch] = useReducer(todoReducer, initialState);
6     const [input, setInput] = useState('');
7
8     // Charger depuis localStorage au montage
9     useEffect(() => {
10         const saved = localStorage.getItem('todos');
11         if (saved) {
12             dispatch({
13                 type: 'LOAD_TODOS',
14                 payload: JSON.parse(saved)
15             });
16         }
17     }, []);
18
19     // Sauvegarder dans localStorage à chaque modification
20     useEffect(() => {
21         if (state.todos.length > 0) {
22             localStorage.setItem('todos', JSON.stringify(state.todos))
23         }
24     }, [state.todos]);
25 }
```

```
26 const addTodo = () => {
27   if (input.trim()) {
28     dispatch({ type: 'ADD_TODO', payload: input });
29     setInput('');
30   }
31 };
32
33 // Filtrer les todos
34 const filteredTodos = state.todos.filter(todo => {
35   if (state.filter === 'active') return !todo.completed;
36   if (state.filter === 'completed') return todo.completed;
37   return true;
38 }) ;
39
40 return (
41   <div style={{
42     maxWidth: '600px',
43     margin: '50px auto',
44     padding: '20px',
45     fontFamily: 'Arial, sans-serif'
46   }>
47     <h1 style={{ textAlign: 'center', color: '#2c3e50' }}>
48       Ma Todo List
49     </h1>
50
51   /* Formulaire d'ajout */
52   <div style={{
53     display: 'flex',
54     gap: '10px',
55     marginBottom: '20px'
56   }>
57     <input
58       type="text"
59       value={input}
60       onChange={(e) => setInput(e.target.value)}
61       onKeyPress={(e) => e.key === 'Enter' && addTodo()}
62     >
63       placeholder="Nouvelle tâche..."
64       style={{
65         flex: 1,
66         padding: '12px',
67         fontSize: '16px',
68         border: '2px solid #ddd',
69         borderRadius: '5px'
70       }}
71     />
72     <button
73       onClick={addTodo}
74       style={{
75         padding: '12px 24px',
76         backgroundColor: '#27ae60',
```

```
76          color: 'white',
77          border: 'none',
78          borderRadius: '5px',
79          cursor: 'pointer',
80          fontSize: '16px',
81          fontWeight: 'bold'
82      } }
83      >
84          Ajouter
85      </button>
86  </div>

87  {/* Filtres */}
88  <div style={{{
89      display: 'flex',
90      gap: '10px',
91      marginBottom: '20px',
92      justifyContent: 'center'
93  }}}>
94      {[ 'all', 'active', 'completed' ].map( filter => (
95          <button
96              key={filter}
97              onClick={() => dispatch({
98                  type: 'SET_FILTER',
99                  payload: filter
100             })}
101             style={{
102                 padding: '8px 16px',
103                 backgroundColor: state.filter === filter
104                     ? '#3498db'
105                     : '#ecf0f1',
106                 color: state.filter === filter
107                     ? 'white'
108                     : '#2c3e50',
109                 border: 'none',
110                 borderRadius: '5px',
111                 cursor: 'pointer',
112                 fontWeight: state.filter === filter
113                     ? 'bold'
114                     : 'normal'
115             }}
116         >
117             { filter === 'all' ? 'Toutes' :
118                 filter === 'active' ? 'Actives' :
119                     'Complété
120             es' }
121         </button>
122     ))}
123  </div>

124  {/* Liste des tâches */}
125  <div>
```

```
126     { filteredTodos.length === 0 ? (
127         <p style={{{
128             textAlign: 'center',
129             color: '#95a5a6',
130             padding: '40px'
131         }}>
132             Aucune tâche à afficher
133         </p>
134     ) : (
135         filteredTodos.map(todo => (
136             <div
137                 key={todo.id}
138                 style={{
139                     display: 'flex',
140                     alignItems: 'center',
141                     padding: '15px',
142                     marginBottom: '10px',
143                     backgroundColor: todo.completed
144                     ? '#d5f4e6'
145                     : 'white',
146                     border: '1px solid #ddd',
147                     borderRadius: '5px'
148                 }}
149             >
150                 <input
151                     type="checkbox"
152                     checked={todo.completed}
153                     onChange={() => dispatch({
154                         type: 'TOGGLE_TODO',
155                         payload: todo.id
156                     })}
157                     style={{
158                         marginRight: '15px',
159                         width: '20px',
160                         height: '20px',
161                         cursor: 'pointer'
162                     }}
163                 />
164                 <span style={{
165                     flex: 1,
166                     fontSize: '16px',
167                     textDecoration: todo.completed
168                     ? 'line-through'
169                     : 'none',
170                     color: todo.completed
171                     ? '#95a5a6'
172                     : '#2c3e50'
173                 }}>
174                     {todo.text}
175                 </span>
176                 <button>
```

```

177         onClick={() => dispatch({
178             type: 'DELETE_TODO',
179             payload: todo.id
180         })}
181         style={{
182             padding: '6px 12px',
183             backgroundColor: '#e74c3c',
184             color: 'white',
185             border: 'none',
186             borderRadius: '3px',
187             cursor: 'pointer'
188         }}
189     >
190         Supprimer
191     </button>
192 </div>
193 )
194 )
195 </div>
196
197 /* Statistiques */
198 <div style={{
199     marginTop: '20px',
200     textAlign: 'center',
201     padding: '15px',
202     backgroundColor: '#ecf0f1',
203     borderRadius: '5px'
204 }}>
205     <strong>Total :</strong> {state.todos.length} | 
206     <strong> Complétées :</strong> {
207         state.todos.filter(t => t.completed).length
208     }
209 </div>
210 </div>
211 );
212 }
213
214 export default App;

```

Listing 2 – src/App.jsx

Points clés de ce projet :

- **useReducer** centralise toute la logique d'état
- **useEffect** gère la persistance localStorage
- **useState** pour l'input du formulaire
- **Immutabilité** respectée dans tous les cas du reducer

3.2 Projet 2 : Liste de Courses avec Timer (45 min)

Objectif : Créer une liste de courses avec un timer Pomodoro intégré.

Compétences :

- useState avec différents types
- useEffect avec nettoyage (timer)
- Persistance localStorage
- Gestion de formulaire

3.2.1 Code Complet

```

1 import { useState, useEffect } from 'react';
2
3 function ShoppingListApp() {
4     // Etats pour la liste
5     const [items, setItems] = useState([]);
6     const [inputValue, setInputValue] = useState('');
7     const [inputPrice, setInputPrice] = useState('');
8
9     // Etats pour le timer Pomodoro
10    const [minutes, setMinutes] = useState(25);
11    const [seconds, setSeconds] = useState(0);
12    const [isActive, setIsActive] = useState(false);
13
14    // Charger la liste depuis localStorage
15    useEffect(() => {
16        const saved = localStorage.getItem('shopping-list');
17        if (saved) {
18            setItems(JSON.parse(saved));
19        }
20    }, []);
21
22    // Sauvegarder la liste
23    useEffect(() => {
24        localStorage.setItem('shopping-list', JSON.stringify(items));
25    }, [items]);
26
27    // Timer Pomodoro
28    useEffect(() => {
29        let interval = null;
30
31        if (isActive) {
32            interval = setInterval(() => {
33                if (seconds === 0) {
34                    if (minutes === 0) {
35                        setIsActive(false);
36                        alert('Temps de course écouté !');
37                    } else {
38                        setMinutes(minutes - 1);
39                        setSeconds(59);
40                    }
41                } else {
42                    setSeconds(seconds - 1);
43                }
44            }, 1000);
45        }
46    }, [isActive]);
47
48    // Ajouter un élément à la liste
49    const handleAddItem = () => {
50        const newItem = {
51            name: inputValue,
52            price: inputPrice
53        };
54        setItems([newItem, ...items]);
55        setInputValue('');
56        setInputPrice('');
57    }
58
59    // Supprimer un élément de la liste
60    const handleDeleteItem = (index) => {
61        const newItems = [...items];
62        newItems.splice(index, 1);
63        setItems(newItems);
64    }
65
66    // Calculer le total de la liste
67    const calculateTotal = () => {
68        const total = items.reduce((acc, item) => acc + item.price, 0);
69        return total;
70    }
71
72    // Sauvegarder la liste dans localStorage
73    useEffect(() => {
74        localStorage.setItem('shopping-list', JSON.stringify(items));
75    }, [items]);
76}

```

```

40         }
41     } else {
42         setSeconds(seconds - 1);
43     }
44 }, 1000);
45 }

46 // Nettoyage : arrêter l'intervalle
47 return () => {
48     if (interval) clearInterval(interval);
49 };
50 , [isActive, minutes, seconds]);

52 const addItem = () => {
53     if (inputValue.trim()) {
54         setItems([
55             ...items,
56             {
57                 id: Date.now(),
58                 name: inputValue,
59                 price: inputPrice ? parseFloat(inputPrice) : 0,
60                 bought: false
61             }
62         ]);
63         setInputValue('');
64         setInputPrice('');
65     }
66 };
67 }

68 const toggleBought = (id) => {
69     setItems(items.map(item =>
70         item.id === id
71             ? {...item, bought: !item.bought}
72             : item
73     )));
74 };
75 }

76 const deleteItem = (id) => {
77     setItems(items.filter(item => item.id !== id));
78 };
79 }

80 const resetTimer = () => {
81     setIsActive(false);
82     setMinutes(25);
83     setSeconds(0);
84 };
85 }

86 const totalPrice = items.reduce((sum, item) => sum + item.price, 0);
87 const boughtCount = items.filter(item => item.bought).length;
88

```

```

90     return (
91       <div style={{
92         maxWidth: '700px',
93         margin: '50px auto',
94         padding: '20px'
95       }>
96         <h1 style={{ textAlign: 'center' }}>
97           Liste de Courses
98         </h1>
99
100        {/* Timer Pomodoro */}
101        <div style={{
102          backgroundColor: '#34495e',
103          color: 'white',
104          padding: '20px',
105          borderRadius: '10px',
106          marginBottom: '30px',
107          textAlign: 'center'
108        }>
109          <h3>Timer de Course</h3>
110          <div style={{
111            fontSize: '48px',
112            fontWeight: 'bold',
113            margin: '20px 0',
114            color: isActive ? '#e74c3c' : 'white'
115          }>
116            {String(minutes).padStart(2, '0')}: {String(seconds).padStart(2, '0')}
117          </div>
118          <div style={{ display: 'flex', gap: '10px',
119            justifyContent: 'center' }}>
120            <button
121              onClick={() => setIsActive(!isActive)}
122              style={{
123                padding: '10px 20px',
124                backgroundColor: isActive ? '#f39c12' : '#27ae60',
125                color: 'white',
126                border: 'none',
127                borderRadius: '5px',
128                cursor: 'pointer',
129                fontSize: '16px'
130              }}
131            >
132              {isActive ? 'Pause' : 'Démarrer'}
133            </button>
134            <button
135              onClick={resetTimer}
136              style={{
137                padding: '10px 20px',
138                backgroundColor: '#95a5a6',

```

```

139             color: 'white',
140             border: 'none',
141             borderRadius: '5px',
142             cursor: 'pointer',
143             fontSize: '16px'
144         )}
145     >
146         Reset
147         </button>
148     </div>
149 </div>
150
151     /* Formulaire d'ajout */
152 <div style={{{
153     backgroundColor: '#ecf0f1',
154     padding: '20px',
155     borderRadius: '10px',
156     marginBottom: '20px'
157 }}>
158     <div style={{ display: 'flex', gap: '10px',
159     marginBottom: '10px' }}>
160         <input
161             type="text"
162             value={inputValue}
163             onChange={(e) => setInputValue(e.target.value)
164             }
165             onKeyPress={(e) => e.key === 'Enter' &&
166             addItem() }
167             placeholder="Nom de l'article..."
168             style={{
169                 flex: 2,
170                 padding: '10px',
171                 fontSize: '16px',
172                 border: '1px solid #bdc3c7',
173                 borderRadius: '5px'
174             }}
175         />
176         <input
177             type="number"
178             value={inputPrice}
179             onChange={(e) => setInputPrice(e.target.value)
180             }
181             onKeyPress={(e) => e.key === 'Enter' &&
182             addItem() }
183             placeholder="Prix..."
184             step="0.01"
185             style={{
186                 flex: 1,
187                 padding: '10px',
188                 fontSize: '16px',
189                 border: '1px solid #bdc3c7',
190             }}
191     </div>
192 </div>
193
194     <div style={{ display: 'flex', gap: '10px' }}>
195         <input
196             type="text"
197             value={inputCategory}
198             onChange={(e) => setInputCategory(e.target.value)
199             }
200             onKeyPress={(e) => e.key === 'Enter' &&
201             addItem() }
202             placeholder="Catégorie..."
203             style={{
204                 flex: 2,
205                 padding: '10px',
206                 fontSize: '16px',
207                 border: '1px solid #bdc3c7',
208             }}
209         />
210         <input
211             type="number"
212             value={inputQuantity}
213             onChange={(e) => setInputQuantity(e.target.value)
214             }
215             onKeyPress={(e) => e.key === 'Enter' &&
216             addItem() }
217             placeholder="Quantité..."
218             style={{
219                 flex: 1,
220                 padding: '10px',
221                 fontSize: '16px',
222                 border: '1px solid #bdc3c7',
223             }}
224     </div>
225 </div>
226
227     <div style={{ display: 'flex', gap: '10px' }}>
228         <input
229             type="text"
230             value={inputBrand}
231             onChange={(e) => setInputBrand(e.target.value)
232             }
233             onKeyPress={(e) => e.key === 'Enter' &&
234             addItem() }
235             placeholder="Marque..."
236             style={{
237                 flex: 2,
238                 padding: '10px',
239                 fontSize: '16px',
240                 border: '1px solid #bdc3c7',
241             }}
242         />
243         <input
244             type="number"
245             value={inputStock}
246             onChange={(e) => setInputStock(e.target.value)
247             }
248             onKeyPress={(e) => e.key === 'Enter' &&
249             addItem() }
250             placeholder="Stock..."
251             style={{
252                 flex: 1,
253                 padding: '10px',
254                 fontSize: '16px',
255                 border: '1px solid #bdc3c7',
256             }}
257     </div>
258 </div>
259
260     <div style={{ display: 'flex', gap: '10px' }}>
261         <input
262             type="text"
263             value={inputColor}
264             onChange={(e) => setInputColor(e.target.value)
265             }
266             onKeyPress={(e) => e.key === 'Enter' &&
267             addItem() }
268             placeholder="Couleur..."
269             style={{
270                 flex: 2,
271                 padding: '10px',
272                 fontSize: '16px',
273                 border: '1px solid #bdc3c7',
274             }}
275         />
276         <input
277             type="number"
278             value={inputWeight}
279             onChange={(e) => setInputWeight(e.target.value)
280             }
281             onKeyPress={(e) => e.key === 'Enter' &&
282             addItem() }
283             placeholder="Poids..."
284             style={{
285                 flex: 1,
286                 padding: '10px',
287                 fontSize: '16px',
288                 border: '1px solid #bdc3c7',
289             }}
290     </div>
291 </div>
292
293     <div style={{ display: 'flex', gap: '10px' }}>
294         <input
295             type="text"
296             value={inputSize}
297             onChange={(e) => setInputSize(e.target.value)
298             }
299             onKeyPress={(e) => e.key === 'Enter' &&
300             addItem() }
301             placeholder="Taille..."
302             style={{
303                 flex: 2,
304                 padding: '10px',
305                 fontSize: '16px',
306                 border: '1px solid #bdc3c7',
307             }}
308         />
309         <input
310             type="number"
311             value={inputPriceUnit}
312             onChange={(e) => setInputPriceUnit(e.target.value)
313             }
314             onKeyPress={(e) => e.key === 'Enter' &&
315             addItem() }
316             placeholder="Prix unitaire..."
317             style={{
318                 flex: 1,
319                 padding: '10px',
320                 fontSize: '16px',
321                 border: '1px solid #bdc3c7',
322             }}
323     </div>
324 </div>
325
326     <div style={{ display: 'flex', gap: '10px' }}>
327         <input
328             type="text"
329             value={inputDescription}
330             onChange={(e) => setInputDescription(e.target.value)
331             }
332             onKeyPress={(e) => e.key === 'Enter' &&
333             addItem() }
334             placeholder="Description..."
335             style={{
336                 flex: 2,
337                 padding: '10px',
338                 fontSize: '16px',
339                 border: '1px solid #bdc3c7',
340             }}
341         />
342         <input
343             type="number"
344             value={inputStockUnit}
345             onChange={(e) => setInputStockUnit(e.target.value)
346             }
347             onKeyPress={(e) => e.key === 'Enter' &&
348             addItem() }
349             placeholder="Stock unitaire..."
350             style={{
351                 flex: 1,
352                 padding: '10px',
353                 fontSize: '16px',
354                 border: '1px solid #bdc3c7',
355             }}
356     </div>
357 </div>
358
359     <div style={{ display: 'flex', gap: '10px' }}>
360         <input
361             type="text"
362             value={inputImage}
363             onChange={(e) => setInputImage(e.target.value)
364             }
365             onKeyPress={(e) => e.key === 'Enter' &&
366             addItem() }
367             placeholder="Image..."
368             style={{
369                 flex: 2,
370                 padding: '10px',
371                 fontSize: '16px',
372                 border: '1px solid #bdc3c7',
373             }}
374         />
375         <input
376             type="number"
377             value={inputStockUnit}
378             onChange={(e) => setInputStockUnit(e.target.value)
379             }
380             onKeyPress={(e) => e.key === 'Enter' &&
381             addItem() }
382             placeholder="Stock unitaire..."
383             style={{
384                 flex: 1,
385                 padding: '10px',
386                 fontSize: '16px',
387                 border: '1px solid #bdc3c7',
388             }}
389     </div>
390 </div>
391
392     <div style={{ display: 'flex', gap: '10px' }}>
393         <input
394             type="text"
395             value={inputImage}
396             onChange={(e) => setInputImage(e.target.value)
397             }
398             onKeyPress={(e) => e.key === 'Enter' &&
399             addItem() }
400             placeholder="Image..."
401             style={{
402                 flex: 2,
403                 padding: '10px',
404                 fontSize: '16px',
405                 border: '1px solid #bdc3c7',
406             }}
407         />
408         <input
409             type="number"
410             value={inputStockUnit}
411             onChange={(e) => setInputStockUnit(e.target.value)
412             }
413             onKeyPress={(e) => e.key === 'Enter' &&
414             addItem() }
415             placeholder="Stock unitaire..."
416             style={{
417                 flex: 1,
418                 padding: '10px',
419                 fontSize: '16px',
420                 border: '1px solid #bdc3c7',
421             }}
422     </div>
423 </div>
424
425     <div style={{ display: 'flex', gap: '10px' }}>
426         <input
427             type="text"
428             value={inputImage}
429             onChange={(e) => setInputImage(e.target.value)
430             }
431             onKeyPress={(e) => e.key === 'Enter' &&
432             addItem() }
433             placeholder="Image..."
434             style={{
435                 flex: 2,
436                 padding: '10px',
437                 fontSize: '16px',
438                 border: '1px solid #bdc3c7',
439             }}
440         />
441         <input
442             type="number"
443             value={inputStockUnit}
444             onChange={(e) => setInputStockUnit(e.target.value)
445             }
446             onKeyPress={(e) => e.key === 'Enter' &&
447             addItem() }
448             placeholder="Stock unitaire..."
449             style={{
450                 flex: 1,
451                 padding: '10px',
452                 fontSize: '16px',
453                 border: '1px solid #bdc3c7',
454             }}
455     </div>
456 </div>
457
458     <div style={{ display: 'flex', gap: '10px' }}>
459         <input
460             type="text"
461             value={inputImage}
462             onChange={(e) => setInputImage(e.target.value)
463             }
464             onKeyPress={(e) => e.key === 'Enter' &&
465             addItem() }
466             placeholder="Image..."
467             style={{
468                 flex: 2,
469                 padding: '10px',
470                 fontSize: '16px',
471                 border: '1px solid #bdc3c7',
472             }}
473         />
474         <input
475             type="number"
476             value={inputStockUnit}
477             onChange={(e) => setInputStockUnit(e.target.value)
478             }
479             onKeyPress={(e) => e.key === 'Enter' &&
480             addItem() }
481             placeholder="Stock unitaire..."
482             style={{
483                 flex: 1,
484                 padding: '10px',
485                 fontSize: '16px',
486                 border: '1px solid #bdc3c7',
487             }}
488     </div>
489 </div>
490
491     <div style={{ display: 'flex', gap: '10px' }}>
492         <input
493             type="text"
494             value={inputImage}
495             onChange={(e) => setInputImage(e.target.value)
496             }
497             onKeyPress={(e) => e.key === 'Enter' &&
498             addItem() }
499             placeholder="Image..."
500             style={{
501                 flex: 2,
502                 padding: '10px',
503                 fontSize: '16px',
504                 border: '1px solid #bdc3c7',
505             }}
506         />
507         <input
508             type="number"
509             value={inputStockUnit}
510             onChange={(e) => setInputStockUnit(e.target.value)
511             }
512             onKeyPress={(e) => e.key === 'Enter' &&
513             addItem() }
514             placeholder="Stock unitaire..."
515             style={{
516                 flex: 1,
517                 padding: '10px',
518                 fontSize: '16px',
519                 border: '1px solid #bdc3c7',
520             }}
521     </div>
522 </div>
523
524     <div style={{ display: 'flex', gap: '10px' }}>
525         <input
526             type="text"
527             value={inputImage}
528             onChange={(e) => setInputImage(e.target.value)
529             }
530             onKeyPress={(e) => e.key === 'Enter' &&
531             addItem() }
532             placeholder="Image..."
533             style={{
534                 flex: 2,
535                 padding: '10px',
536                 fontSize: '16px',
537                 border: '1px solid #bdc3c7',
538             }}
539         />
540         <input
541             type="number"
542             value={inputStockUnit}
543             onChange={(e) => setInputStockUnit(e.target.value)
544             }
545             onKeyPress={(e) => e.key === 'Enter' &&
546             addItem() }
547             placeholder="Stock unitaire..."
548             style={{
549                 flex: 1,
550                 padding: '10px',
551                 fontSize: '16px',
552                 border: '1px solid #bdc3c7',
553             }}
554     </div>
555 </div>
556
557     <div style={{ display: 'flex', gap: '10px' }}>
558         <input
559             type="text"
560             value={inputImage}
561             onChange={(e) => setInputImage(e.target.value)
562             }
563             onKeyPress={(e) => e.key === 'Enter' &&
564             addItem() }
565             placeholder="Image..."
566             style={{
567                 flex: 2,
568                 padding: '10px',
569                 fontSize: '16px',
570                 border: '1px solid #bdc3c7',
571             }}
572         />
573         <input
574             type="number"
575             value={inputStockUnit}
576             onChange={(e) => setInputStockUnit(e.target.value)
577             }
578             onKeyPress={(e) => e.key === 'Enter' &&
579             addItem() }
580             placeholder="Stock unitaire..."
581             style={{
582                 flex: 1,
583                 padding: '10px',
584                 fontSize: '16px',
585                 border: '1px solid #bdc3c7',
586             }}
587     </div>
588 </div>
589
590     <div style={{ display: 'flex', gap: '10px' }}>
591         <input
592             type="text"
593             value={inputImage}
594             onChange={(e) => setInputImage(e.target.value)
595             }
596             onKeyPress={(e) => e.key === 'Enter' &&
597             addItem() }
598             placeholder="Image..."
599             style={{
600                 flex: 2,
601                 padding: '10px',
602                 fontSize: '16px',
603                 border: '1px solid #bdc3c7',
604             }}
605         />
606         <input
607             type="number"
608             value={inputStockUnit}
609             onChange={(e) => setInputStockUnit(e.target.value)
610             }
611             onKeyPress={(e) => e.key === 'Enter' &&
612             addItem() }
613             placeholder="Stock unitaire..."
614             style={{
615                 flex: 1,
616                 padding: '10px',
617                 fontSize: '16px',
618                 border: '1px solid #bdc3c7',
619             }}
620     </div>
621 </div>
622
623     <div style={{ display: 'flex', gap: '10px' }}>
624         <input
625             type="text"
626             value={inputImage}
627             onChange={(e) => setInputImage(e.target.value)
628             }
629             onKeyPress={(e) => e.key === 'Enter' &&
630             addItem() }
631             placeholder="Image..."
632             style={{
633                 flex: 2,
634                 padding: '10px',
635                 fontSize: '16px',
636                 border: '1px solid #bdc3c7',
637             }}
638         />
639         <input
640             type="number"
641             value={inputStockUnit}
642             onChange={(e) => setInputStockUnit(e.target.value)
643             }
644             onKeyPress={(e) => e.key === 'Enter' &&
645             addItem() }
646             placeholder="Stock unitaire..."
647             style={{
648                 flex: 1,
649                 padding: '10px',
650                 fontSize: '16px',
651                 border: '1px solid #bdc3c7',
652             }}
653     </div>
654 </div>
655
656     <div style={{ display: 'flex', gap: '10px' }}>
657         <input
658             type="text"
659             value={inputImage}
660             onChange={(e) => setInputImage(e.target.value)
661             }
662             onKeyPress={(e) => e.key === 'Enter' &&
663             addItem() }
664             placeholder="Image..."
665             style={{
666                 flex: 2,
667                 padding: '10px',
668                 fontSize: '16px',
669                 border: '1px solid #bdc3c7',
670             }}
671         />
672         <input
673             type="number"
674             value={inputStockUnit}
675             onChange={(e) => setInputStockUnit(e.target.value)
676             }
677             onKeyPress={(e) => e.key === 'Enter' &&
678             addItem() }
679             placeholder="Stock unitaire..."
680             style={{
681                 flex: 1,
682                 padding: '10px',
683                 fontSize: '16px',
684                 border: '1px solid #bdc3c7',
685             }}
686     </div>
687 </div>
688
689     <div style={{ display: 'flex', gap: '10px' }}>
690         <input
691             type="text"
692             value={inputImage}
693             onChange={(e) => setInputImage(e.target.value)
694             }
695             onKeyPress={(e) => e.key === 'Enter' &&
696             addItem() }
697             placeholder="Image..."
698             style={{
699                 flex: 2,
700                 padding: '10px',
701                 fontSize: '16px',
702                 border: '1px solid #bdc3c7',
703             }}
704         />
705         <input
706             type="number"
707             value={inputStockUnit}
708             onChange={(e) => setInputStockUnit(e.target.value)
709             }
710             onKeyPress={(e) => e.key === 'Enter' &&
711             addItem() }
712             placeholder="Stock unitaire..."
713             style={{
714                 flex: 1,
715                 padding: '10px',
716                 fontSize: '16px',
717                 border: '1px solid #bdc3c7',
718             }}
719     </div>
720 </div>
721
722     <div style={{ display: 'flex', gap: '10px' }}>
723         <input
724             type="text"
725             value={inputImage}
726             onChange={(e) => setInputImage(e.target.value)
727             }
728             onKeyPress={(e) => e.key === 'Enter' &&
729             addItem() }
730             placeholder="Image..."
731             style={{
732                 flex: 2,
733                 padding: '10px',
734                 fontSize: '16px',
735                 border: '1px solid #bdc3c7',
736             }}
737         />
738         <input
739             type="number"
740             value={inputStockUnit}
741             onChange={(e) => setInputStockUnit(e.target.value)
742             }
743             onKeyPress={(e) => e.key === 'Enter' &&
744             addItem() }
745             placeholder="Stock unitaire..."
746             style={{
747                 flex: 1,
748                 padding: '10px',
749                 fontSize: '16px',
750                 border: '1px solid #bdc3c7',
751             }}
752     </div>
753 </div>
754
755     <div style={{ display: 'flex', gap: '10px' }}>
756         <input
757             type="text"
758             value={inputImage}
759             onChange={(e) => setInputImage(e.target.value)
760             }
761             onKeyPress={(e) => e.key === 'Enter' &&
762             addItem() }
763             placeholder="Image..."
764             style={{
765                 flex: 2,
766                 padding: '10px',
767                 fontSize: '16px',
768                 border: '1px solid #bdc3c7',
769             }}
770         />
771         <input
772             type="number"
773             value={inputStockUnit}
774             onChange={(e) => setInputStockUnit(e.target.value)
775             }
776             onKeyPress={(e) => e.key === 'Enter' &&
777             addItem() }
778             placeholder="Stock unitaire..."
779             style={{
780                 flex: 1,
781                 padding: '10px',
782                 fontSize: '16px',
783                 border: '1px solid #bdc3c7',
784             }}
785     </div>
786 </div>
787
788     <div style={{ display: 'flex', gap: '10px' }}>
789         <input
790             type="text"
791             value={inputImage}
792             onChange={(e) => setInputImage(e.target.value)
793             }
794             onKeyPress={(e) => e.key === 'Enter' &&
795             addItem() }
796             placeholder="Image..."
797             style={{
798                 flex: 2,
799                 padding: '10px',
800                 fontSize: '16px',
801                 border: '1px solid #bdc3c7',
802             }}
803         />
804         <input
805             type="number"
806             value={inputStockUnit}
807             onChange={(e) => setInputStockUnit(e.target.value)
808             }
809             onKeyPress={(e) => e.key === 'Enter' &&
810             addItem() }
811             placeholder="Stock unitaire..."
812             style={{
813                 flex: 1,
814                 padding: '10px',
815                 fontSize: '16px',
816                 border: '1px solid #bdc3c7',
817             }}
818     </div>
819 </div>
820
821     <div style={{ display: 'flex', gap: '10px' }}>
822         <input
823             type="text"
824             value={inputImage}
825             onChange={(e) => setInputImage(e.target.value)
826             }
827             onKeyPress={(e) => e.key === 'Enter' &&
828             addItem() }
829             placeholder="Image..."
830             style={{
831                 flex: 2,
832                 padding: '10px',
833                 fontSize: '16px',
834                 border: '1px solid #bdc3c7',
835             }}
836         />
837         <input
838             type="number"
839             value={inputStockUnit}
840             onChange={(e) => setInputStockUnit(e.target.value)
841             }
842             onKeyPress={(e) => e.key === 'Enter' &&
843             addItem() }
844             placeholder="Stock unitaire..."
845             style={{
846                 flex: 1,
847                 padding: '10px',
848                 fontSize: '16px',
849                 border: '1px solid #bdc3c7',
850             }}
851     </div>
852 </div>
853
854     <div style={{ display: 'flex', gap: '10px' }}>
855         <input
856             type="text"
857             value={inputImage}
858             onChange={(e) => setInputImage(e.target.value)
859             }
860             onKeyPress={(e) => e.key === 'Enter' &&
861             addItem() }
862             placeholder="Image..."
863             style={{
864                 flex: 2,
865                 padding: '10px',
866                 fontSize: '16px',
867                 border: '1px solid #bdc3c7',
868             }}
869         />
870         <input
871             type="number"
872             value={inputStockUnit}
873             onChange={(e) => setInputStockUnit(e.target.value)
874             }
875             onKeyPress={(e) => e.key === 'Enter' &&
876             addItem() }
877             placeholder="Stock unitaire..."
878             style={{
879                 flex: 1,
880                 padding: '10px',
881                 fontSize: '16px',
882                 border: '1px solid #bdc3c7',
883             }}
884     </div>
885 </div>
886
887     <div style={{ display: 'flex', gap: '10px' }}>
888         <input
889             type="text"
890             value={inputImage}
891             onChange={(e) => setInputImage(e.target.value)
892             }
893             onKeyPress={(e) => e.key === 'Enter' &&
894             addItem() }
895             placeholder="Image..."
896             style={{
897                 flex: 2,
898                 padding: '10px',
899                 fontSize: '16px',
900                 border: '1px solid #bdc3c7',
901             }}
902         />
903         <input
904             type="number"
905             value={inputStockUnit}
906             onChange={(e) => setInputStockUnit(e.target.value)
907             }
908             onKeyPress={(e) => e.key === 'Enter' &&
909             addItem() }
910             placeholder="Stock unitaire..."
911             style={{
912                 flex: 1,
913                 padding: '10px',
914                 fontSize: '16px',
915                 border: '1px solid #bdc3c7',
916             }}
917     </div>
918 </div>
919
920     <div style={{ display: 'flex', gap: '10px' }}>
921         <input
922             type="text"
923             value={inputImage}
924             onChange={(e) => setInputImage(e.target.value)
925             }
926             onKeyPress={(e) => e.key === 'Enter' &&
927             addItem() }
928             placeholder="Image..."
929             style={{
930                 flex: 2,
931                 padding: '10px',
932                 fontSize: '16px',
933                 border: '1px solid #bdc3c7',
934             }}
935         />
936         <input
937             type="number"
938             value={inputStockUnit}
939             onChange={(e) => setInputStockUnit(e.target.value)
940             }
941             onKeyPress={(e) => e.key === 'Enter' &&
942             addItem() }
943             placeholder="Stock unitaire..."
944             style={{
945                 flex: 1,
946                 padding: '10px',
947                 fontSize: '16px',
948                 border: '1px solid #bdc3c7',
949             }}
950     </div>
951 </div>
952
953     <div style={{ display: 'flex', gap: '10px' }}>
954         <input
955             type="text"
956             value={inputImage}
957             onChange={(e) => setInputImage(e.target.value)
958             }
959             onKeyPress={(e) => e.key === 'Enter' &&
960             addItem() }
961             placeholder="Image..."
962             style={{
963                 flex: 2,
964                 padding: '10px',
965                 fontSize: '16px',
966                 border: '1px solid #bdc3c7',
967             }}
968         />
969         <input
970             type="number"
971             value={inputStockUnit}
972             onChange={(e) => setInputStockUnit(e.target.value)
973             }
974             onKeyPress={(e) => e.key === 'Enter' &&
975             addItem() }
976             placeholder="Stock unitaire..."
977             style={{
978                 flex: 1,
979                 padding: '10px',
980                 fontSize: '16px',
981                 border: '1px solid #bdc3c7',
982             }}
983     </div>
984 </div>
985
986     <div style={{ display: 'flex', gap: '10px' }}>
987         <input
988             type="text"
989             value={inputImage}
990             onChange={(e) => setInputImage(e.target.value)
991             }
992             onKeyPress={(e) => e.key === 'Enter' &&
993             addItem() }
994             placeholder="Image..."
995             style={{
996                 flex: 2,
997                 padding: '10px',
998                 fontSize: '16px',
999                 border: '1px solid #bdc3c7',
1000            }}
1001        />
1002        <input
1003            type="number"
1004            value={inputStockUnit}
1005            onChange={(e) => setInputStockUnit(e.target.value)
1006            }
1007            onKeyPress={(e) => e.key === 'Enter' &&
1008            addItem() }
1009            placeholder="Stock unitaire..."
1010            style={{
1011                flex: 1,
1012                padding: '10px',
1013                fontSize: '16px',
1014                border: '1px solid #bdc3c7',
1015            }}
1016    </div>
1017 </div>
1018
1019     <div style={{ display: 'flex', gap: '10px' }}>
1020         <input
1021             type="text"
1022             value={inputImage}
1023             onChange={(e) => setInputImage(e.target.value)
1024             }
1025             onKeyPress={(e) => e.key === 'Enter' &&
1026             addItem() }
1027             placeholder="Image..."
1028             style={{
1029                 flex: 2,
1030                 padding: '10px',
1031                 fontSize: '16px',
1032                 border: '1px solid #bdc3c7',
1033             }}
1034         />
1035         <input
1036             type="number"
1037             value={inputStockUnit}
1038             onChange={(e) => setInputStockUnit(e.target.value)
1039             }
1040             onKeyPress={(e) => e.key === 'Enter' &&
1041             addItem() }
1042             placeholder="Stock unitaire..."
1043             style={{
1044                 flex: 1,
1045                 padding: '10px',
1046                 fontSize: '16px',
1047                 border: '1px solid #bdc3c7',
1048             }}
1049     </div>
1050 </div>
1051
1052     <div style={{ display: 'flex', gap: '10px' }}>
1053         <input
1054             type="text"
1055             value={inputImage}
1056             onChange={(e) => setInputImage(e.target.value)
1057             }
1058             onKeyPress={(e) => e.key === 'Enter' &&
1059             addItem() }
1060             placeholder="Image..."
1061             style={{
1062                 flex: 2,
1063                 padding: '10px',
1064                 fontSize: '16px',
1065                 border: '1px solid #bdc3c7',
1066             }}
1067         />
1068         <input
1069             type="number"
1070             value={inputStockUnit}
1071             onChange={(e) => setInputStockUnit(e.target.value)
1072             }
1073             onKeyPress={(e) => e.key === 'Enter' &&
1074             addItem() }
1075             placeholder="Stock unitaire..."
1076             style={{
1077                 flex: 1,
1078                 padding: '10px',
1079                 fontSize: '16px',
1080                 border: '1px solid #bdc3c7',
1081             }}
1082     </div>
1083 </div>
1084
1085     <div style={{ display: 'flex', gap: '10px' }}>
1086         <input
1087             type="text"
1088             value={inputImage}
1089             onChange={(e) => setInputImage(e.target.value)
1090             }
1091             onKeyPress={(e) => e.key === 'Enter' &&
1092             addItem() }
1093             placeholder="Image..."
1094             style={{
1095                 flex: 2,
1096                 padding: '10px',
1097                 fontSize: '16px',
1098                 border: '1px solid #bdc3c7',
1099            }}
1100        />
1101        <input
1102            type="number"
1103            value={inputStockUnit}
1104            onChange={(e) => setInputStockUnit(e.target.value)
1105            }
1106            onKeyPress={(e) => e.key === 'Enter' &&
1107            addItem() }
1108            placeholder="Stock unitaire..."
1109            style={{
1110                flex: 1,
1111                padding: '10px',
1112                fontSize: '16px',
1113                border: '1px solid #bdc3c7',
1114            }}
1115    </div>
1116 </div>
1117
1118     <div style={{ display: 'flex', gap: '10px' }}>
1119         <input
1120             type="text"
1121             value={inputImage}
1122             onChange={(e) => setInputImage(e.target.value)
1123             }
1124             onKeyPress={(e) => e.key === 'Enter' &&
1125             addItem() }
1126             placeholder="Image..."
1127             style={{
1128                 flex: 2,
1129                 padding: '10px',
1130                 fontSize: '16px',
1131                 border: '1px solid #bdc3c7',
1132             }}
1133         />
1134         <input
1135             type="number"
1136             value={inputStockUnit}
1137             onChange={(e) => setInputStockUnit(e.target.value)
1138             }
1139             onKeyPress={(e) => e.key === 'Enter' &&
1140             addItem() }
1141             placeholder="Stock unitaire..."
1142             style={{
1143                 flex: 1,
1144                 padding: '10px',
1145                 fontSize: '16px',
1146                 border: '1px solid #bdc3c7',
1147             }}
1148     </div>
1149 </div>
1150
1151     <div style={{ display: 'flex', gap: '10px' }}>
1152         <input
1153             type="text"
1154             value={inputImage}
1155             onChange={(e) => setInputImage(e.target.value)
1156             }
1157             onKeyPress={(e) => e.key === 'Enter' &&
1158             addItem() }
1159             placeholder="Image..."
1160             style={{
1161                 flex: 2,
1162                 padding: '10px',
1163                 fontSize: '16
```

```
185             borderRadius: '5px'
186         )}
187     />
188     <button
189         onClick={addItem}
190         style={{
191             padding: '10px 20px',
192             backgroundColor: '#3498db',
193             color: 'white',
194             border: 'none',
195             borderRadius: '5px',
196             cursor: 'pointer',
197             fontSize: '16px'
198         }}
199     >
200         Ajouter
201     </button>
202     </div>
203 </div>
204
205     /* Liste des articles */
206 <div>
207     {items.map(item => (
208         <div
209             key={item.id}
210             style={{
211                 display: 'flex',
212                 alignItems: 'center',
213                 padding: '15px',
214                 marginBottom: '10px',
215                 backgroundColor: item.bought ? '#d5f4e6'
216                 : 'white',
217                 border: '1px solid #ddd',
218                 borderRadius: '5px'
219             })
220         >
221             <input
222                 type="checkbox"
223                 checked={item.bought}
224                 onChange={() => toggleBought(item.id)}
225                 style={{
226                     marginRight: '15px',
227                     width: '20px',
228                     height: '20px'
229                 }}
230             />
231             <span style={{
232                 flex: 1,
233                 textDecoration: item.bought ? 'line-
234                 through' : 'none',
235                 color: item.bought ? '#95a5a6' : '#2c3e50
236             }}>
```

```

    ,
    fontSize: '16px'
  }}>
  {item.name}
</span>
<span style={{
  fontWeight: 'bold',
  marginRight: '15px',
  color: '#27ae60'
}}>
  {item.price.toFixed(2)}DT
</span>
<button
  onClick={() => deleteItem(item.id)}
  style={{
    padding: '6px 12px',
    backgroundColor: '#e74c3c',
    color: 'white',
    border: 'none',
    borderRadius: '3px',
    cursor: 'pointer'
  }}
>
  Supprimer
</button>
</div>
))}

</div>

/* Statistiques */
<div style={{
  marginTop: '20px',
  padding: '15px',
  backgroundColor: '#3498db',
  color: 'white',
  borderRadius: '5px',
  display: 'flex',
  justifyContent: 'space-between'
}}>
<div>
  <strong>Total :</strong> {items.length} articles
| 
  <strong> Achetés :</strong> {boughtCount}
</div>
<div>
  <strong>Prix total :</strong> {totalPrice.toFixed
(2)}DT
</div>
</div>
</div>
);

```

```
282 }  
283  
284 export default ShoppingListApp;
```

Listing 3 – src/ShoppingListApp.jsx

Concepts utilisés dans ce projet :

1. **useState multiple** : Pour gérer plusieurs états indépendants
2. **useEffect avec cleanup** : Pour le timer (important !)
3. **useEffect avec dépendances** : Pour localStorage
4. **Calculs dérivés** : totalPrice et boughtCount calculés à la volée
5. **Inputs contrôlés** : text et number

3.3 Projet 3 : Blog Interactif avec Tri et Recherche (60 min)

Objectif : Reprendre le blog de la semaine précédente et le rendre complètement interactif.

Nouvelles fonctionnalités :

- Ajouter des articles via formulaire
- Liker / Disliker
- Recherche par titre et contenu
- Tri par likes ou date
- Filtrer par auteur
- Supprimer des articles
- Persistance complète

3.3.1 Code Complet de l'Application

```

1 import { useState, useEffect } from 'react';
2
3 function BlogApp() {
4     // Etats principaux
5     const [articles, setArticles] = useState(() => {
6         const saved = localStorage.getItem('blog-articles');
7         if (saved) return JSON.parse(saved);
8         return [
9             {
10                 id: 1,
11                 title: "Introduction à React",
12                 author: "Alice",
13                 content: "React est une bibliothèque JavaScript...",
14                 likes: 5,
15                 createdAt: new Date().toISOString()
16             },
17             {
18                 id: 2,
19                 title: "Les Hooks en détail",
20                 author: "Bob",
21                 content: "Les hooks révolutionnent React...",
22                 likes: 3,
23                 createdAt: new Date().toISOString()
24             }
25         ];
26     });
27
28     // Etats de filtrage et tri
29     const [searchTerm, setSearchTerm] = useState('');
30     const [filterAuthor, setFilterAuthor] = useState('');
31     const [sortBy, setSortBy] = useState('date'); // 'date' ou 'likes'
32
33     // Etats du formulaire

```

```
34  const [showForm, setShowForm] = useState(false);
35  const [title, setTitle] = useState('');
36  const [author, setAuthor] = useState('');
37  const [content, setContent] = useState('');

38
39 // Sauvegarder dans localStorage
40 useEffect(() => {
41   localStorage.setItem('blog-articles', JSON.stringify(articles));
42 }, [articles]);

43
44 // Fonctions de gestion
45 const addArticle = () => {
46   if (!title.trim() || !author.trim() || !content.trim()) {
47     alert('Tous les champs sont obligatoires');
48     return;
49   }

50   const newArticle = {
51     id: Date.now(),
52     title,
53     author,
54     content,
55     likes: 0,
56     createdAt: new Date().toISOString()
57   };

58   setArticles([newArticle, ...articles]);
59   setTitle('');
60   setAuthor('');
61   setContent('');
62   setShowForm(false);
63 };

64
65 const handleLike = (id) => {
66   setArticles(articles.map(article =>
67     article.id === id
68       ? {...article, likes: article.likes + 1}
69       : article
70   ));
71 };
72

73
74 const handleDelete = (id) => {
75   if (window.confirm('Supprimer cet article ?')) {
76     setArticles(articles.filter(article => article.id !== id));
77   }
78 };
79

80
81 // Filtrage et tri
82 let filteredArticles = articles;
```

```
83 // Recherche
84 if (searchTerm) {
85     filteredArticles = filteredArticles.filter(article =>
86         article.title.toLowerCase().includes(searchTerm.toLowerCase()) ||
87         article.content.toLowerCase().includes(searchTerm.toLowerCase())
88     );
89 }
90
91 // Filtre par auteur
92 if (filterAuthor) {
93     filteredArticles = filteredArticles.filter(article =>
94         article.author === filterAuthor
95     );
96 }
97
98 // Tri
99 filteredArticles = [...filteredArticles].sort((a, b) => {
100     if (sortBy === 'likes') {
101         return b.likes - a.likes;
102     }
103     return new Date(b.createdAt) - new Date(a.createdAt);
104 });
105
106
107 // Liste unique des auteurs
108 const authors = [...new Set(articles.map(a => a.author))];
109
110 return (
111     <div style={{
112         maxWidth: '900px',
113         margin: '0 auto',
114         padding: '20px'
115     }>
116         <h1 style={{ textAlign: 'center', color: '#2c3e50' }}>
117             Mon Blog React Interactif
118         </h1>
119
120         {/* Barre d'actions */}
121         <div style={{
122             backgroundColor: '#ecf0f1',
123             padding: '20px',
124             borderRadius: '10px',
125             marginBottom: '20px'
126         }>
127             <div style={{
128                 display: 'grid',
129                 gridTemplateColumns: '1fr 1fr 1fr auto',
130                 gap: '10px',
131                 marginBottom: '15px'
```

```

132     })>
133         {/* Recherche */}
134         <input
135             type="text"
136             value={searchTerm}
137             onChange={(e) => setSearchTerm(e.target.value)}
138         )}
139         placeholder="Rechercher..."
140         style={{
141             padding: '10px',
142             border: '1px solid #bdc3c7',
143             borderRadius: '5px',
144             fontSize: '14px'
145         }}
146     />
147     {/* Filtre auteur */}
148     <select
149         value={filterAuthor}
150         onChange={(e) => setFilterAuthor(e.target.value)}
151     )}
152         style={{
153             padding: '10px',
154             border: '1px solid #bdc3c7',
155             borderRadius: '5px',
156             fontSize: '14px'
157         }}
158         >
159             <option value="">Tous les auteurs</option>
160             {authors.map(author => (
161                 <option key={author} value={author}>
162                     {author}
163                 </option>
164             )))
165             </select>
166             {/* Tri */}
167             <select
168                 value={sortBy}
169                 onChange={(e) => setSortBy(e.target.value)}
170                 style={{
171                     padding: '10px',
172                     border: '1px solid #bdc3c7',
173                     borderRadius: '5px',
174                     fontSize: '14px'
175                 }}
176             >
177                 <option value="date">Plus récents</option>
178                 <option value="likes">Plus likés</option>
179             </select>
180

```

```

181     {/* Bouton ajouter */}
182     <button
183         onClick={() => setShowForm (!showForm)}
184         style={{
185             padding: '10px 20px',
186             backgroundColor: '#27ae60',
187             color: 'white',
188             border: 'none',
189             borderRadius: '5px',
190             cursor: 'pointer',
191             fontWeight: 'bold'
192         }}
193     >
194         {showForm ? 'Annuler' : '+ Nouvel Article'}
195     </button>
196 </div>
197
198     <div style={{ fontSize: '14px', color: '#7f8c8d' }}>
199         {filteredArticles.length} article(s) affiché(s)
200     sur {articles.length}
201     </div>
202 </div>
203
204     {/* Formulaire d'ajout */}
205     {showForm && (
206         <div style={{
207             backgroundColor: '#fff',
208             padding: '20px',
209             borderRadius: '10px',
210             marginBottom: '20px',
211             border: '2px solid #27ae60'
212         }>
213             <h3>Nouvel Article</h3>
214             <input
215                 type="text"
216                 value={title}
217                 onChange={(e) => setTitle(e.target.value)}
218                 placeholder="Titre..."
219                 style={{
220                     width: '100%',
221                     padding: '10px',
222                     marginBottom: '10px',
223                     border: '1px solid #ddd',
224                     borderRadius: '5px',
225                     fontSize: '16px'
226                 }}
227             />
228             <input
229                 type="text"
230                 value={author}
231                 onChange={(e) => setAuthor(e.target.value)}>

```

```
231         placeholder="Auteur..."  
232         style={{  
233             width: '100%',  
234             padding: '10px',  
235             marginBottom: '10px',  
236             border: '1px solid #ddd',  
237             borderRadius: '5px',  
238             fontSize: '16px'  
239         }}  
240     />  
241     <textarea  
242         value={content}  
243         onChange={(e) => setContent(e.target.value)}  
244         placeholder="Contenu..."  
245         rows="5"  
246         style={{  
247             width: '100%',  
248             padding: '10px',  
249             marginBottom: '10px',  
250             border: '1px solid #ddd',  
251             borderRadius: '5px',  
252             fontSize: '14px',  
253             fontFamily: 'inherit'  
254         }}  
255     />  
256     <button  
257         onClick={addArticle}  
258         style={{  
259             padding: '12px 24px',  
260             backgroundColor: '#27ae60',  
261             color: 'white',  
262             border: 'none',  
263             borderRadius: '5px',  
264             cursor: 'pointer',  
265             fontSize: '16px',  
266             fontWeight: 'bold'  
267         }}  
268     >  
269         Publier  
270     </button>  
271 </div>  
272 )}  
273  
274 /* Liste des articles */  
275 <div>  
276     {filteredArticles.length === 0 ? (  
277         <p style={{ textAlign: 'center', color: '#95a5a6'  
278         , padding: '40px' }}>  
279             Aucun article trouvé  
280         </p>  
281     ) : (
```

```

281         filteredArticles.map(article => (
282             <article
283                 key={article.id}
284                 style={{
285                     backgroundColor: 'white',
286                     padding: '20px',
287                     marginBottom: '20px',
288                     borderRadius: '10px',
289                     border: '1px solid #ddd'
290                 }}
291             >
292                 <h2 style={{ color: '#2c3e50', marginTop: '0' }}>
293                     {article.title}
294                 </h2>
295                 <p style={{ color: '#7f8c8d', fontSize: '14px', marginBottom: '15px' }}>
296                     Par {article.author} | {
297                         new Date(article.createdAt).toLocaleDateString('fr-FR')
298                     }
299                 </p>
300                 <p style={{ lineHeight: '1.6', color: '#34495e' }}>
301                     {article.content}
302                 </p>
303                 <div style={{
304                     display: 'flex',
305                     alignItems: 'center',
306                     gap: '10px',
307                     marginTop: '15px',
308                     paddingTop: '15px',
309                     borderTop: '1px solid #ecf0f1'
310                 }}>
311                     <button
312                         onClick={() => handleLike(article.id)}
313                         style={{
314                             padding: '8px 16px',
315                             backgroundColor: '#3498db',
316                             color: 'white',
317                             border: 'none',
318                             borderRadius: '5px',
319                             cursor: 'pointer',
320                             display: 'flex',
321                             alignItems: 'center',
322                             gap: '5px'
323                         }}
324                 >
325             </div>
326         </article>
327     )

```

```

328 >           { article.likes }
329 </button>
330 <button
331     onClick={() => handleDelete(
332         article.id)}
333     style={{
334         padding: '8px 16px',
335         backgroundColor: '#e74c3c',
336         color: 'white',
337         border: 'none',
338         borderRadius: '5px',
339         cursor: 'pointer',
340         marginLeft: 'auto'
341     }}>
342     > Supprimer
343     </button>
344     </div>
345     </article>
346   ))
347   )
348 </div>
349 </div>
350 ) ;
351 }
352 }
353
354 export default BlogApp;

```

Listing 4 – src/BlogApp.jsx - Version complète

Récapitulatif des techniques utilisées :

- **useState avec initialisation lazy** : Charge depuis localStorage au montage
- **useEffect pour persistance** : Sauvegarde automatique
- **Filtrage multiple** : Recherche + auteur + tri
- **Immutabilité stricte** : map(), filter(), spread operator
- **Affichage conditionnel** : Formulaire et message vide
- **Calculs dérivés** : Liste d'auteurs unique avec Set

4 Récapitulatif et Concepts Clés

Les 10 Règles d'Or des Hooks

1. **Toujours au top level** : Jamais dans des conditions, boucles ou fonctions
2. **Immutabilité stricte** : Toujours créer de nouveaux objets/tableaux
3. **Forme fonctionnelle** : `useState(prev => ...)` pour updates multiples
4. **Tableau de dépendances** : Inclure toutes les valeurs utilisées
5. **Cleanup obligatoire** : Pour timers, événements, abonnements
6. **Initialisation lazy** : `useState(() => ...)` pour calculs coûteux
7. **useReducer pour complexité** : Quand plusieurs états liés
8. **localStorage** : `useEffect` pour persistance
9. **Inputs contrôlés** : `value + onChange` toujours liés
10. **Clés stables** : Utiliser des IDs uniques, pas les index

Patterns à retenir :

```

1 // Ajouter à un tableau
2 setItems ([... items , newItem]) ;

3
4 // Supprimer d'un tableau
5 setItems(items . filter(item => item . id !== id)) ;

6
7 // Modifier dans un tableau
8 setItems(items . map(item =>
9     item . id === id ? {... item , prop: value} : item
10));
11
12 // Modifier un objet
13 setUser ({... user , name: "Bob"});

14
15 // Timer avec cleanup
16 useEffect(() => {
17     const timer = setInterval(() => {...} , 1000);
18     return () => clearInterval(timer);
19 } , [deps]) ;

20
21 // localStorage
22 useEffect(() => {
23     localStorage .setItem('key' , JSON .stringify(data));
24 } , [data]);

```

5 Travail à Rendre

Projet Final

Créer une application complète "Gestionnaire de Projets" combinant les trois hooks.

Fonctionnalités obligatoires :

1. useReducer : Gérer les projets (ajouter, modifier statut, supprimer)
2. useEffect : Persistance localStorage + timer par projet
3. useState : Formulaires, filtres, recherche
4. Chaque projet a : titre, description, statut (todo/doing/-done), deadline
5. Filtrer par statut
6. Trier par deadline
7. Recherche dans titre/description
8. Timer Pomodoro par projet
9. Statistiques visuelles (nombre par statut)

Livrables :

- Code source complet
- Captures d'écran de toutes les fonctionnalités
- Explication du reducer (tous les cas)
- Explication de chaque useEffect
- README avec instructions

Échéance : La veille de la prochaine séance à 23h59

Ce travail compte pour 20% de la note finale.

6 Ressources

6.1 Documentation Officielle

- React Documentation : <https://react.dev>
- useState : <https://react.dev/reference/react/useState>
- useEffect : <https://react.dev/reference/react/useEffect>
- useReducer : <https://react.dev/reference/react/useReducer>
- Rules of Hooks : <https://react.dev/reference/rules-of-hooks>

6.2 Outils

- ESLint Plugin React Hooks (détection des erreurs)
- React DevTools (inspection des états)