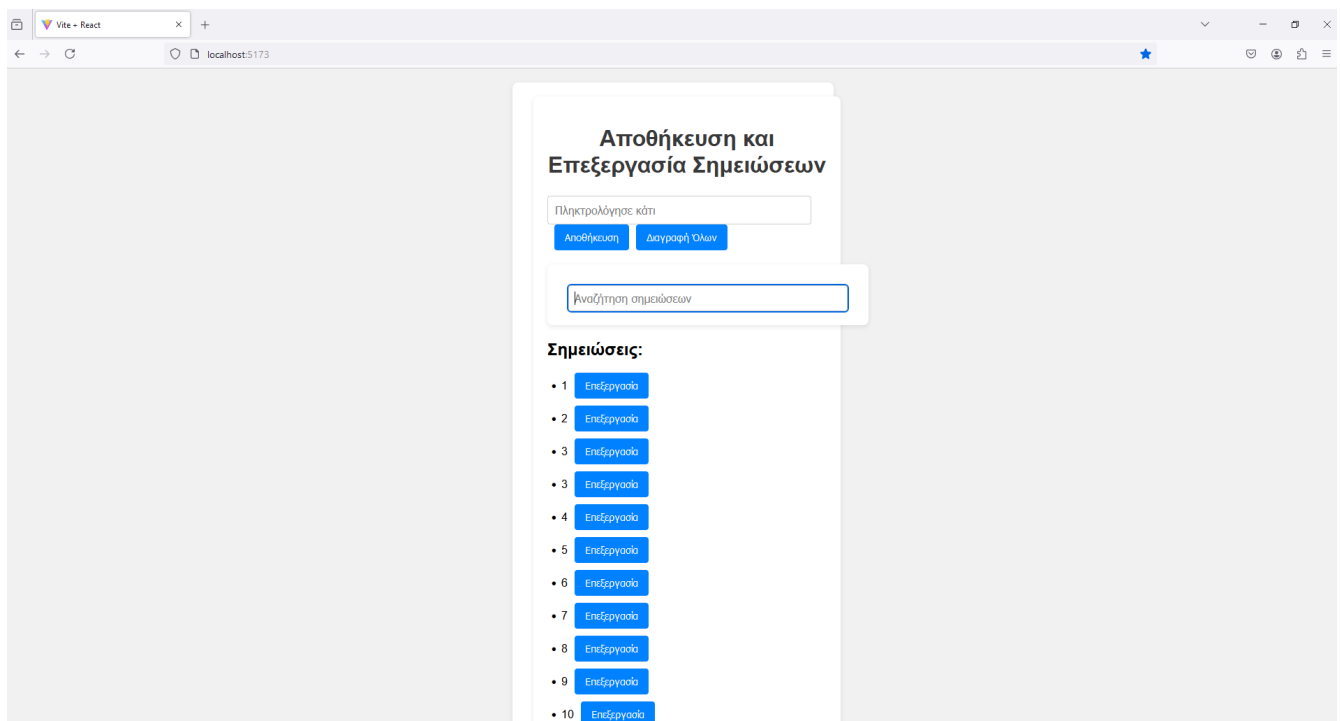
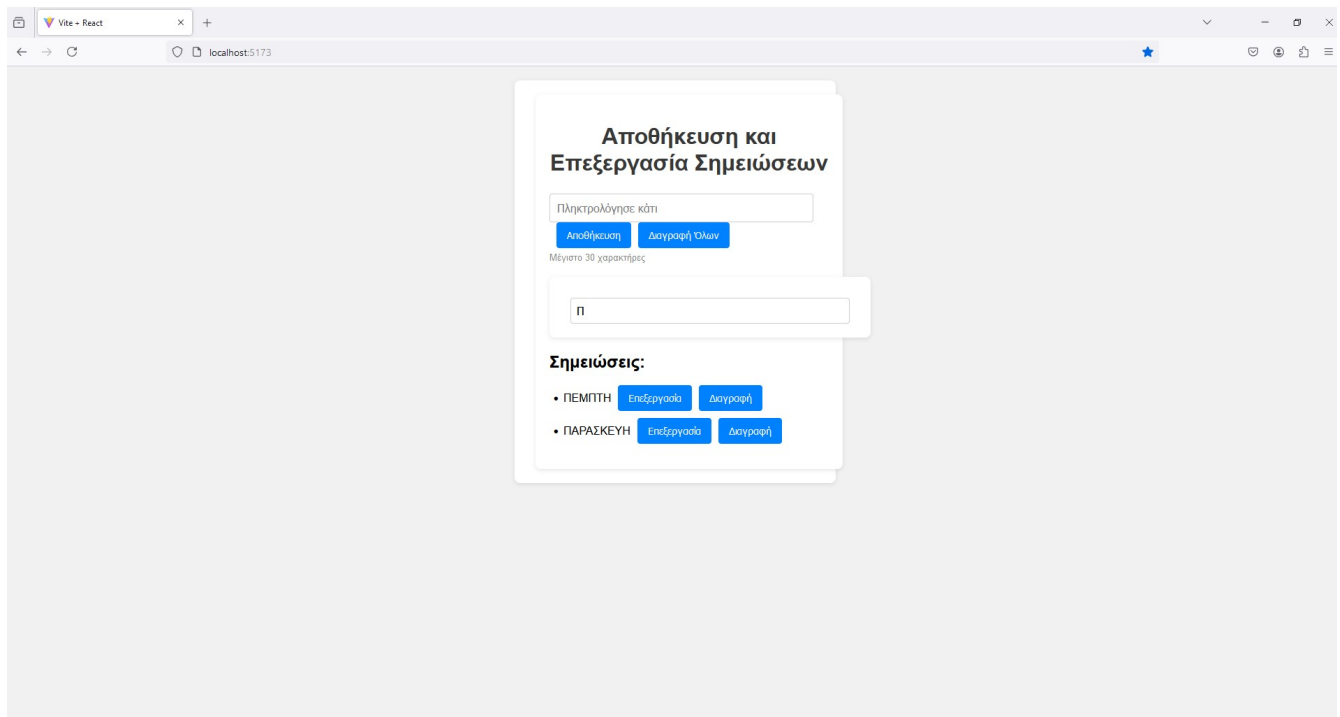


Note keeper με αναζήτηση και επεξεργασία.

App.jsx



App.jsx ΜΕ ΣΧΟΛΙΑ

import { useState, useEffect } from 'react'; // Εισαγωγή hooks από τη React

```
function App() {
  // Καταστάσεις για input, λίστα, αναζήτηση και επεξεργασία
  const [input, setInput] = useState(""); // Κείμενο εισόδου για νέα σημείωση
  const [dataList, setDataList] = useState([]); // Λίστα σημειώσεων
  const [searchTerm, setSearchTerm] = useState(""); // Όρος αναζήτησης
  const [editId, setEditId] = useState(null); // Το ID της σημείωσης που επεξεργαζόμαστε
  const [editText, setEditText] = useState(""); // Το νέο κείμενο κατά την επεξεργασία

  // Φόρτωσε τις σημειώσεις από το localStorage όταν ξεκινήσει η εφαρμογή
  useEffect(() => {
    const stored = localStorage.getItem('myDataList');
    if (stored) {
      setDataList(JSON.parse(stored)); // Αν υπάρχουν, αποθήκευσέ τις στο state
    }
  }, []);

  // Αποθηκεύει μια νέα λίστα στο state και στο localStorage
  const saveDataList = (newList) => {
    setDataList(newList);
    localStorage.setItem('myDataList', JSON.stringify(newList));
  };

  // Προσθήκη νέας σημείωσης (με όριο 30 χαρακτήρες)
  const handleAdd = () => {
    const trimmed = input.trim();
    if (trimmed === "" || trimmed.length > 30) return; // Απορρίπτει άδειες ή πολύ μεγάλες τιμές
    const newItem = { id: Date.now(), text: trimmed }; // Δημιουργία νέου αντικειμένου
    saveDataList([...dataList, newItem]); // Προσθήκη στην υπάρχουσα λίστα
    setInput(""); // Καθαρισμός πεδίου εισαγωγής
  };

  // Έναρξη επεξεργασίας: αποθηκεύει ID και τρέχον κείμενο
  const startEdit = (id, currentText) => {
    setEditId(id);
    setEditText(currentText);
  };

  // Αποθήκευση αλλαγών κατά την επεξεργασία
  const saveEdit = () => {
    const trimmed = editText.trim();
    if (trimmed === "" || trimmed.length > 30) return; // Έλεγχος ορίου και κενού
    const newList = dataList.map(item =>
      item.id === editId ? { ...item, text: trimmed } : item
    );
    saveDataList(newList); // Ενημέρωση λίστας
    setEditId(null);
    setEditText(""); // Τέλος επεξεργασίας
  };

  // Ακύρωση επεξεργασίας
  const cancelEdit = () => {

```

```

    setEditId(null);
    setEditText("");
  };

  // Διαγραφή όλων των σημειώσεων
  const handleClear = () => {
    setDataList([]);
    localStorage.removeItem('myDataList');
  };

  // Διαγραφή μόνο μίας σημείωσης
  const handleDelete = (id) => {
    const newList = dataList.filter(item => item.id !== id); // Φιλτράρει τη λίστα
    saveDataList(newList);
  };

  // Φιλτράρισμα σημειώσεων σύμφωνα με την αναζήτηση
  const filteredList = dataList.filter(item =>
    item.text.toLowerCase().includes(searchTerm.toLowerCase())
  );

  // JSX απόδοση της εφαρμογής
  return (
    <div style={{ padding: '20px' }}>
      <h1>Αποθήκευση και Επεξεργασία Σημειώσεων</h1>

      {/* Πεδίο προσθήκης σημείωσης */}
      <input
        type="text"
        value={input}
        onChange={e => setInput(e.target.value)} // Ενημέρωση input state
        placeholder="Πληκτρολόγησε κάτι"
        onKeyDown={e => e.key === 'Enter' && handleAdd()} // Enter = προσθήκη
        maxLength={30} // Μέγιστο 30 χαρακτήρες
      />
      <button onClick={handleAdd} style={{ marginLeft: '10px' }}>
        Αποθήκευση
      </button>
      <button onClick={handleClear} style={{ marginLeft: '10px' }}>
        Διαγραφή Όλων
      </button>
      {/* Υπόδειξη ορίου χαρακτήρων */}
      <small style={{ display: 'block', marginTop: '5px', color: 'gray' }}>
        Μέγιστο 30 χαρακτήρες
      </small>

      {/* Αναζήτηση σημειώσεων */}
      <div style={{ marginTop: '20px' }}>
        <input
          type="text"
          value={searchTerm}
          onChange={e => setSearchTerm(e.target.value)} // Ενημέρωση όρου αναζήτησης
          placeholder="Αναζήτηση σημειώσεων"
          style={{ width: '100%', padding: '8px' }}
        />
      </div>
    </div>
  );

```

```

    /* Εμφάνιση σημειώσεων */
    <h2>Σημειώσεις:</h2>
    {filteredList.length === 0 ? (
      <p>Δεν υπάρχουν σημειώσεις.</p>
    ) : (
      <ul>
        {filteredList.map(item => (
          <li key={item.id} style={{ marginBottom: '10px' }}>
            /* Αν η σημείωση είναι σε κατάσταση επεξεργασίας */
            {editId === item.id ? (
              <>
                <input
                  type="text"
                  value={editText}
                  onChange={e => setEditText(e.target.value)} // Ενημέρωση νέου κειμένου
                  maxLength={30}
                />
                <button onClick={saveEdit} style={{ marginLeft: '5px' }}>
                  Αποθήκευση
                </button>
                <button onClick={cancelEdit} style={{ marginLeft: '5px' }}>
                  Άκυρο
                </button>
              </>
            ) : (
              <>
                {item.text}
                <button
                  onClick={() => startEdit(item.id, item.text)} // Εκκίνηση επεξεργασίας
                  style={{ marginLeft: '10px' }}
                >
                  Επεξεργασία
                </button>
                <button
                  onClick={() => handleDelete(item.id)} // Διαγραφή μόνο αυτής
                  style={{ marginLeft: '10px' }}
                >
                  Διαγραφή
                </button>
              </>
            )}
          </li>
        )]}
      </ul>
    )}
  </div>
);
}

```

```
export default App; // Εξαγωγή του component
```

main.jsx

```
import React from 'react';
```

```
import ReactDOM from 'react-dom/client';
import App from './App';
import './index.css';

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Αποθήκευση Δεδομένων</title>
  </head>
  <body>
    <div id="root"></div> <!-- Εδώ τοποθετείται το React app -->
  </body>
</html>
```

index.css

```
body {
  font-family: Arial, sans-serif;
  background-color: #f0f0f0;
  margin: 0;
  padding: 20px;
  display: flex;
  justify-content: center;
}

div {
  background: white;
  padding: 20px 30px;
  border-radius: 8px;
  box-shadow: 0 2px 6px rgba(0,0,0,0.1);
  max-width: 400px;
  width: 100%;
}
```

```
h1 {
  color: #333;
  text-align: center;
  margin-bottom: 15px;
}

input {
  padding: 10px;
  font-size: 16px;
  margin-top: 10px;
  width: calc(100% - 22px);
  border: 1px solid #ccc;
  border-radius: 4px;
  box-sizing: border-box;
}

button {
  margin-left: 10px;
  padding: 10px 15px;
  font-size: 14px;
  cursor: pointer;
  background-color: #007bff;
  border: none;
  border-radius: 4px;
  color: white;
  transition: background-color 0.3s ease;
}

button:hover {
  background-color: #0056b3;
}

p {
  margin-top: 20px;
  text-align: center;
  color: #666;
}

ul {
  margin-top: 15px;
  padding-left: 20px;
}

li {
  margin-bottom: 8px;
  font-size: 16px;
}
```

```
}
```

App.jsx ΧΩΡΙΣ ΣΧΟΛΙΑ

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

function App() {
  const [input, setInput] = useState('');
  const [dataList, setDataList] = useState([]);
  const [searchTerm, setSearchTerm] = useState('');
  const [editId, setEditId] = useState(null);
  const [editText, setEditText] = useState('');

  useEffect(() => {
    const stored = localStorage.getItem('myDataList');
    if (stored) {
      setDataList(JSON.parse(stored));
    }
  }, []);

  const saveDataList = (newList) => {
    setDataList(newList);
    localStorage.setItem('myDataList', JSON.stringify(newList));
  };

  const handleAdd = () => {
    const trimmed = input.trim();
    if (trimmed === '' || trimmed.length > 30) return;
    const newItem = { id: Date.now(), text: trimmed };
    saveDataList([...dataList, newItem]);
    setInput('');
  };

  const startEdit = (id, currentText) => {
    setEditId(id);
    setEditText(currentText);
  };

  const saveEdit = () => {
    const trimmed = editText.trim();
    if (trimmed === '' || trimmed.length > 30) return;
    const newList = dataList.map(item =>
      item.id === editId ? { ...item, text: trimmed } : item
    );
    saveDataList(newList);
  };
}
```

```

    setEditId(null);
    setEditText('');
};

const cancelEdit = () => {
    setEditId(null);
    setEditText('');
};

const handleClear = () => {
    setDataList([]);
    localStorage.removeItem('myDataList');
};

const handleDelete = (id) => {
    const newList = dataList.filter(item => item.id !== id);
    saveDataList(newList);
};

const filteredList = dataList.filter(item =>
    item.text.toLowerCase().includes(searchTerm.toLowerCase())
);

return (
    <div style={{ padding: '20px' }}>
        <h1>Αποθήκευση και Επεξεργασία Σημειώσεων</h1>

        { /* Input προσθήκης */ }
        <input
            type="text"
            value={input}
            onChange={e => setInput(e.target.value)}
            placeholder="Πληκτρολόγησε κάτι"
            onKeyDown={e => e.key === 'Enter' && handleAdd()}
            maxLength={30}
        />
        <button onClick={handleAdd} style={{ marginLeft: '10px' }}>
            Αποθήκευση
        </button>
        <button onClick={handleClear} style={{ marginLeft: '10px' }}>
            Διαγραφή Όλων
        </button>
        <small style={{ display: 'block', marginTop: '5px', color: 'gray' }}>
            Μέγιστο 30 χαρακτήρες
        </small>

        { /* Αναζήτηση */ }

```

```

<div style={{ marginTop: '20px' }}>
  <input
    type="text"
    value={searchTerm}
    onChange={e => setSearchTerm(e.target.value)}
    placeholder="Αναζήτηση σημειώσεων"
    style={{ width: '100%', padding: '8px' }}
  />
</div>

{/* Λίστα σημειώσεων */}
<h2>Σημειώσεις:</h2>
{filteredList.length === 0 ? (
  <p>Δεν υπάρχουν σημειώσεις.</p>
) : (
  <ul>
    {filteredList.map(item => (
      <li key={item.id} style={{ marginBottom: '10px' }}>
        {editId === item.id ? (
          <>
            <input
              type="text"
              value={editText}
              onChange={e => setEditText(e.target.value)}
              maxLength={30}
            />
            <button onClick={saveEdit} style={{ marginLeft: '5px' }}>
              Αποθήκευση
            </button>
            <button onClick={cancelEdit} style={{ marginLeft: '5px' }}>
              Άκυρο
            </button>
          </>
        ) : (
          <>
            {item.text}
            <button
              onClick={() => startEdit(item.id, item.text)}
              style={{ marginLeft: '10px' }}
            >
              Επεξεργασία
            </button>
            <button
              onClick={() => handleDelete(item.id)}
              style={{ marginLeft: '10px' }}
            >
              Διαγραφή

```

```
        </button>
      </>
    )}
  </li>
  ))}
</ul>
)}
</div>
);
}

export default App;
```