

```

import React, { useState, useEffect } from "react";
import "./App.css";

const ROWS = 10;
const COLS = 10;

const App = () => {
  const [cells, setCells] = useState(
    Array.from({ length: ROWS }, () => Array(COLS).fill(""))
  );
  const [boldCells, setBoldCells] = useState(new Set());
  const [dragging, setDragging] = useState(false);
  const [selectedCells, setSelectedCells] = useState(new Set());
  const [dependencies, setDependencies] = useState({});

  useEffect(() => {
    const handleMouseUp = () => setDragging(false);
    document.addEventListener("mouseup", handleMouseUp);
    return () => document.removeEventListener("mouseup", handleMouseUp);
  }, []);

  const handleChange = (row, col, value) => {
    const newCells = [...cells];
    newCells[row][col] = value;
    setCells(newCells);
    updateDependentCells(row, col, newCells);
  };

  const toggleBold = (row, col) => {
    const newBoldCells = new Set(boldCells);
    const cellKey = `${row}-${col}`;
    newBoldCells.has(cellKey) ? newBoldCells.delete(cellKey) :
newBoldCells.add(cellKey);
    setBoldCells(newBoldCells);
  };

  const handleMouseDown = (row, col) => {
    setDragging(true);
    setSelectedCells(new Set([`${row}-${col}`]));
  };

  const handleMouseEnter = (row, col) => {
    if (dragging) {
      setSelectedCells((prev) => new Set(prev).add(`${row}-${col}`));
    }
  };

  return (
    <div className="spreadsheet-container">
      <h1 className="title">Google Sheets Clone</h1>
      <table className="spreadsheet">
        <tbody>
          {cells.map((row, rowIndex) => (
            <tr key={rowIndex}>

```

```

    {row.map((cell, colIndex) => (
      <td
        key={colIndex}
        onMouseDown={() => handleMouseDown(rowIndex, colIndex)}
        onMouseEnter={() => handleMouseEnter(rowIndex, colIndex)}
        className={selectedCells.has(`${rowIndex}-${colIndex}`) ? "selected" :
""}
      >
        <input
          type="text"
          value={cell}
          onChange={(e) => handleChange(rowIndex, colIndex, e.target.value)}
          className={boldCells.has(`${rowIndex}-${colIndex}`) ? "bold" : ""}
        />
        <button className="bold-btn" onClick={() => toggleBold(rowIndex,
colIndex)}>B</button>
      </td>
    ))}
  </tr>
))}
</tbody>
</table>
</div>
);
};

export default App;

```