<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Camping Checklist</title>

<!-- Tailwind CSS CDN -->

<script src="https://cdn.tailwindcss.com"></script>

<!-- React and ReactDOM -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/react/18.2.0/umd/react.production.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/react-dom/18.2.0/umd/react-dom.production.min.js"></script>

<!-- Babel for JSX -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/babel-standalone/7.23.2/babel.min.js"></script>

<!-- jsPDF for PDF export -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/jspdf/2.5.1/jspdf.umd.min.js"></script>

</head>

<body class="bg-gray-100 min-h-screen flex items-center justify-center">

<div id="root"></div>

<script type="text/babel">

// Initial data from the Excel content

const initialData = [

{ category: "帳內設備", items: [

{ item: "帳篷 營釘 營繩 營槌 打氣", responsible: "自己顧自己" },

{ item: "睡垫", responsible: "自己顧自己" },

{ item: "鋁箔防潮墊 地布", responsible: "自己顧自己" },

{ item: "防曬防雨天幕 營釘", responsible: "樹窿" },

{ item: "睡袋或棉被", responsible: "自己顧自己" },

{ item: "枕頭", responsible: "自己顧自己" },

{ item: "客廳帳內野餐垫", responsible: "自己顧自己" },

{ item: "行動電源 充電線", responsible: "自己顧自己" }

]},

{ category: "帳外設備", items: [

{ item: "露營桌椅", responsible: "自己顧自己" },

{ item: "照明設備 裝飾燈串 餐具", responsible: "自己顧自己" },

{ item: "爐具", responsible: "Annaball" },

{ item: "飲用水", responsible: "" },

{ item: "蚊香（電）或實體", responsible: "樹窿" },

{ item: "平底pan", responsible: "Annaball、樹窿" },

{ item: "噴槍", responsible: "Annaball" },

{ item: "Gas", responsible: "" }

]},

{ category: "個人用品", items: [

{ item: "盒裝紙巾", responsible: "Annaball" },

{ item: "濕紙巾", responsible: "" },

{ item: "沐浴乳 洗髮精 洗面奶 牙刷 牙膏 牙線", responsible: "自己顧自己" },

{ item: "洗碗精 菜瓜布", responsible: "Annaball" },

{ item: "防曮乳 防蚊液 帽子 太陽眼鏡", responsible: "自己顧自己" },

{ item: "毛巾 換洗衣物", responsible: "自己顧自己" },

{ item: "垃圾袋", responsible: "" },

{ item: "拖鞋", responsible: "自己顧自己" },

{ item: "早餐食材（雞蛋、牛角包、牛奶、牛油）", responsible: "Annaball" },

{ item: "電扇", responsible: "" }

]},

{ category: "食物", items: [

{ item: "煮水壺", responsible: "Annaball" },

{ item: "飲料", responsible: "" },

{ item: "咖啡包 茶包", responsible: "" }

]},

{ category: "狗狗用品", items: [

{ item: "床 毛氈 蚊怕水 光圈", responsible: "自己顧自己" }

]},

{ category: "其他用品", items: [

{ item: "緊急護理包：感冒 外傷 蚊蟲叮咬藥 OK蹦", responsible: "" },

{ item: "酒 醒酒器 酒杯", responsible: "" }

]}

];

// Main App component

function App() {

const [checklist, setChecklist] = React.useState(() => {

// Load from localStorage if available

const saved = localStorage.getItem('campingChecklist');

return saved ? JSON.parse(saved) : initialData;

});

// Save to localStorage whenever checklist changes

React.useEffect(() => {

localStorage.setItem('campingChecklist', JSON.stringify(checklist));

}, [checklist]);

// Handle input change

const handleInputChange = (categoryIndex, itemIndex, value) => {

const newChecklist = [...checklist];

newChecklist[categoryIndex].items[itemIndex].responsible = value;

setChecklist(newChecklist);

};

// Generate PDF

const generatePDF = () => {

const { jsPDF } = window.jspdf;

const doc = new jsPDF();

doc.setFontSize(16);

doc.text("Camping Checklist", 10, 10);

let y = 20;

checklist.forEach((category) => {

doc.setFontSize(14);

doc.text(category.category, 10, y);

y += 10;

category.items.forEach((item) => {

doc.setFontSize(12);

doc.text(`- ${item.item}: ${item.responsible || '未分配'}`, 15, y);

y += 8;

});

y += 5;

});

doc.save("camping\_checklist.pdf");

};

return (

<div className="max-w-4xl mx-auto p-6 bg-white rounded-lg shadow-lg">

<h1 className="text-2xl font-bold mb-4 text-center">露營清單 (Camping Checklist)</h1>

<button

onClick={generatePDF}

className="mb-4 px-4 py-2 bg-blue-500 text-white rounded hover:bg-blue-600"

>

匯出為 PDF

</button>

{checklist.map((category, catIndex) => (

<div key={catIndex} className="mb-6">

<h2 className="text-xl font-semibold mb-2">{category.category}</h2>

<table className="w-full border-collapse">

<thead>

<tr className="bg-gray-200">

<th className="border p-2 text-left">項目</th>

<th className="border p-2 text-left">負責人</th>

</tr>

</thead>

<tbody>

{category.items.map((item, itemIndex) => (

<tr key={itemIndex} className="hover:bg-gray-50">

<td className="border p-2">{item.item}</td>

<td className="border p-2">

<input

type="text"

value={item.responsible}

onChange={(e) => handleInputChange(catIndex, itemIndex, e.target.value)}

className="w-full p-1 border rounded"

placeholder="輸入負責人"

/>

</td>

</tr>

))}

</tbody>

</table>

</div>

))}

</div>

);

}

// Render the app

const root = ReactDOM.createRoot(document.getElementById('root'));

root.render(<App />);

</script>

</body>

</html>