

Hier ist der komplette, voll funktionsfähige Code für die Web-Anwendung. Er beinhaltet die Benutzeroberfläche, die Logik für das Wechseln der Tabs, das Hinzufügen/Löschen von Einträgen, die automatische lokale Speicherung (Local Storage) und eine Export-Funktion (als CSV-Datei für Excel).

Du kannst den gesamten Code kopieren, in einen einfachen Texteditor (wie Notepad) einfügen und als sommerfest.html speichern. Danach öffnest du die Datei einfach per Doppelklick in einem beliebigen Webbrowser.

HTML

```
<!DOCTYPE html>
<html lang="de-AT">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My Sommerfest Planer </title>
  <style>
    :root {
      --primary: #c8102e; /* Österreichisches Rot für den sportlichen Look */
      --primary-hover: #9c0c24;
      --secondary: #2c3e50;
      --bg-color: #f4f7f6;
      --border-radius: 8px;
    }
    body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      background-color: var(--bg-color);
      color: #333;
      margin: 0;
      padding: 20px;
    }
    .container {
      max-width: 1000px;
      margin: 0 auto;
      background: #fff;
      padding: 20px 30px;
      border-radius: var(--border-radius);
      box-shadow: 0 4px 10px rgba(0,0,0,0.1);
    }
    h1 {
      text-align: center;
      color: var(--primary);
      border-bottom: 2px solid var(--primary);
```

```
padding-bottom: 10px;
margin-bottom: 25px;
}
.tabs {
display: flex;
gap: 10px;
margin-bottom: 20px;
flex-wrap: wrap;
}
.tab-btn {
flex: 1;
padding: 12px;
border: none;
background: #e0e0e0;
cursor: pointer;
font-size: 16px;
font-weight: bold;
border-radius: var(--border-radius);
transition: background 0.3s;
color: #333;
}
.tab-btn.active {
background: var(--primary);
color: white;
}
.tab-content {
display: none;
animation: fadeIn 0.3s;
}
.tab-content.active {
display: block;
}
@keyframes fadeIn {
from { opacity: 0; }
to { opacity: 1; }
}
.form-group {
display: flex;
gap: 10px;
margin-bottom: 20px;
flex-wrap: wrap;
background: #f8f9fa;
padding: 15px;
border-radius: var(--border-radius);
border: 1px solid #ddd;
}
input, select, button {
padding: 10px;
```

```
border: 1px solid #ccc;
border-radius: 4px;
font-size: 14px;
}
input, select {
flex: 1;
min-width: 150px;
}
button.add-btn {
background-color: var(--secondary);
color: white;
border: none;
cursor: pointer;
font-weight: bold;
min-width: 120px;
}
button.add-btn:hover {
background-color: #1a252f;
}
table {
width: 100%;
border-collapse: collapse;
margin-top: 10px;
}
th, td {
text-align: left;
padding: 12px;
border-bottom: 1px solid #eee;
}
th {
background-color: #f8f9fa;
color: var(--secondary);
}
tr:hover {
background-color: #fdfdfd;
}
.action-btn {
background: #e74c3c;
color: white;
border: none;
padding: 6px 10px;
border-radius: 4px;
cursor: pointer;
font-size: 12px;
}
.action-btn:hover {
background: #c0392b;
}
```

```

.task-done {
  text-decoration: line-through;
  color: #7f8c8d;
}

.checkbox-custom {
  width: 18px;
  height: 18px;
  cursor: pointer;
}

.export-section {
  text-align: center;
  margin-top: 40px;
  padding-top: 20px;
  border-top: 1px solid #eee;
}

.export-btn {
  background-color: #27ae60;
  color: white;
  border: none;
  padding: 12px 24px;
  font-size: 16px;
  border-radius: 4px;
  cursor: pointer;
  font-weight: bold;
}

.export-btn:hover {
  background-color: #2ecc71;
}
</style>
</head>
<body>

```

```

<div class="container">
  <h1> Vereins-Spinnerfest-Plan </h1>

```

```

<div class="tabs">
  <button class="tab-btn active"
onclick="switchTab('schichtplan')"> Heller-Schichtplan </button>
  <button class="tab-btn" onclick="switchTab('aufgaben')"> Aufgabenliste </button>
  <button class="tab-btn" onclick="switchTab('material')"> Materialbestand </button>
</div>

```

```

<!-- TAB 1: SCHICHTPLAN -->
<div id="schichtplan" class="tab-content active">
  <h2> Heller-Schichtplan </h2>
  <div class="form-group">

```

```

        <input type="text" id="shiftTime" placeholder="Zeit (z.B. 14:00 - 16:00)" required>
        <input type="text" id="shiftRole" placeholder="Station (z.B. Bierausschank, Kassa)"
required>
        <input type="text" id="shiftName" placeholder="Name des Helfers" required>
        <button class="add-btn" onclick="addShift()">Eintragen </button>
    </div>
    <table>
    <thead>
    <tr>
    <th>Zeitraum </th>
    <th>Station / Aufgabe </th>
    <th>Helfer </th>
    <th style="width: 80px;">Aktion </th>
    </tr>
    </thead>
    <tbody id="shiftTableBody">
    </tbody>
    </table>
</div>

```

```

<!-- TAB 2: AUFGABENLISTE -->
<div id="aufgaben" class="tab-content">
    <h2>Aufgabenliste </h2>
    <div class="form-group">
        <input type="text" id="taskDesc" placeholder="Aufgabe (z.B. Biertischgarnituren aufstellen)"
required>
        <input type="text" id="taskAssignee" placeholder="Zuständig (optional)">
        <button class="add-btn" onclick="addTask()">Hinzufügen </button>
    </div>
    <table>
    <thead>
    <tr>
    <th style="width: 50px;">Zuständig </th>
    <th>Aufgabe </th>
    <th>Zuständig </th>
    <th style="width: 80px;">Aktion </th>
    </tr>
    </thead>
    <tbody id="taskTableBody">
    </tbody>
    </table>
</div>

```

```

<!-- TAB 3: MATERIALBESTAND -->
<div id="material" class="tab-content">
  <h2>Materialbestand </h2>
  <div class="form-group">
    <input type="text" id="materialName" placeholder="Artikel (z.B. Spritzer-Krügerl,
Servietten)" required>
    <input type="number" id="materialQty" placeholder="Anzahl / Menge" required min="1">
    <button class="add-btn" onclick="addMaterial()">Hinzufügen </button>
  </div>
  <table>
    <thead>
      <tr>
        <th>Artikel </th>
        <th>Menge </th>
        <th style="width: 80px;">Aktion </th>
      </tr>
    </thead>
    <tbody id="materialTableBody">
    </tbody>
  </table>
</div>

```

```

<!-- EXPORT SEKTION -->
<div class="export-section">
  <button class="export-btn" onclick="exportToCSV()">Alle Daten als CSV
exportieren </button>
</div>
</div>

```

```

<script>
  // --- DATENSPEICHERUNG (Local Storage) ---
  let appData = {
    shifts: [],
    tasks: [],
    materials: []
  };

  // Daten beim Start laden
  function loadData() {
    const savedData = localStorage.getItem('sommerfestData');
    if (savedData) {
      appData = JSON.parse(savedData);
    }
    renderAll();
  }

```

```

// Daten speichern
function saveData() {
  localStorage.setItem('sommerfestData', JSON.stringify(appData));
  renderAll();
}

```

```

// --- TAB STEUERUNG ---
function switchTab(tabId) {
  document.querySelectorAll('.tab-content').forEach(tab => {
    tab.classList.remove('active');
  });
  document.querySelectorAll('.tab-btn').forEach(btn => {
    btn.classList.remove('active');
  });
  document.getElementById(tabId).classList.add('active');
  event.currentTarget.classList.add('active');
}

```

```

// --- SCHICHTPLAN LOGIK ---
function addShift() {
  const time = document.getElementById('shiftTime').value.trim();
  const role = document.getElementById('shiftRole').value.trim();
  const name = document.getElementById('shiftName').value.trim();

  if (time && role && name) {
    appData.shifts.push({ id: Date.now(), time, role, name });
    document.getElementById('shiftTime').value = "";
    document.getElementById('shiftRole').value = "";
    document.getElementById('shiftName').value = "";
    saveData();
  } else {
    alert('Bitte alle Felder für den Schichtplan ausfüllen.');
```

```

  }
}

function deleteShift(id) {
  appData.shifts = appData.shifts.filter(shift => shift.id !== id);
  saveData();
}

```

```

function renderShifts() {
  const tbody = document.getElementById('shiftTableBody');
  tbody.innerHTML = "";
  appData.shifts.forEach(shift => {
    const tr = document.createElement("tr");
    tr.innerHTML = `
      <td>${shift.time}</td>

```

```

        <td>${shift.role}</td>
        <td>${shift.name}</td>
        <td><button class="action-btn" onclick="deleteShift(${shift.id})">Löschen</button></td>
    `;
    tbody.appendChild(tr);
  });
}

```

```

// --- AUFGABEN LOGIK ---
function addTask() {
  const desc = document.getElementById('taskDesc').value.trim();
  const assignee = document.getElementById('taskAssignee').value.trim() || 'Alle';

```

```

  if (desc) {
    appData.tasks.push({ id: Date.now(), desc, assignee, done: false });
    document.getElementById('taskDesc').value = '';
    document.getElementById('taskAssignee').value = '';
    saveData();
  } else {
    alert('Bitte eine Aufgabenbeschreibung eingeben.');
```

```

  }
}

function toggleTask(id) {
  const task = appData.tasks.find(t => t.id === id);
  if (task) {
    task.done = !task.done;
    saveData();
  }
}

```

```

function deleteTask(id) {
  appData.tasks = appData.tasks.filter(task => task.id !== id);
  saveData();
}

```

```

function renderTasks() {
  const tbody = document.getElementById('taskTableBody');
  tbody.innerHTML = '';
  appData.tasks.forEach(task => {
    const tr = document.createElement('tr');
    const textClass = task.done ? 'task-done' : '';
    tr.innerHTML = `
      <td><input type="checkbox" class="checkbox-custom" ${task.done ? 'checked' : ''}
      onclick="toggleTask(${task.id})"></td>
      <td class="${textClass}">${task.desc}</td>
      <td class="${textClass}">${task.assignee}</td>
      <td><button class="action-btn" onclick="deleteTask(${task.id})">Löschen</button></td>
    `;

```

```

    `;
    tbody.appendChild(tr);
  });
}

// --- MATERIAL LOGIK ---
function addMaterial() {
  const name = document.getElementById('materialName').value.trim();
  const qty = document.getElementById('materialQty').value.trim();

  if (name && qty) {
    appData.materials.push({ id: Date.now(), name, qty });
    document.getElementById('materialName').value = "";
    document.getElementById('materialQty').value = "";
    saveData();
  } else {
    alert('Bitte Artikel und Menge eingeben.');
```

```

  }
}

function deleteMaterial(id) {
  appData.materials = appData.materials.filter(m => m.id !== id);
  saveData();
}
```

```

function renderMaterials() {
  const tbody = document.getElementById('materialTableBody');
  tbody.innerHTML = "";
  appData.materials.forEach(item => {
    const tr = document.createElement("tr");
    tr.innerHTML = `
      <td>${item.name}</td>
      <td>${item.qty}</td>
      <td><button class="action-btn"
onclick="deleteMaterial(${item.id})">Löschen</button></td>
    `;
    tbody.appendChild(tr);
  });
}
```

```

// Alle Bereiche neu zeichnen
function renderAll() {
  renderShifts();
  renderTasks();
  renderMaterials();
}
```

```

// --- CSV EXPORT LOGIK ---
```

```

function exportToCSV() {
    let csvContent = "\uFEFF"; // UTF-8 BOM für Excel

    // Schichtplan Export
    csvContent += "--- SCHICHTPLAN ---\n";
    csvContent += "Zeitraum;Station;Helfer\n";
    appData.shifts.forEach(s => {
        csvContent += `${s.time}";"${s.role}";"${s.name}"\n`;
    });
    csvContent += "\n";

    // Aufgaben Export
    csvContent += "--- AUFGABENLISTE ---\n";
    csvContent += "Status;Aufgabe;Zustaendig\n";
    appData.tasks.forEach(t => {
        const status = t.done ? "Erledigt" : "Offen";
        csvContent += `${status}";"${t.desc}";"${t.assignee}"\n`;
    });
    csvContent += "\n";

    // Material Export
    csvContent += "--- MATERIALBESTAND ---\n";
    csvContent += "Artikel;Menge\n";
    appData.materials.forEach(m => {
        csvContent += `${m.name}";"${m.qty}"\n`;
    });

    // Download Trigger
    const blob = new Blob([csvContent], { type: 'text/csv;charset=utf-8;' });
    const link = document.createElement("a");
    const url = URL.createObjectURL(blob);
    link.setAttribute("href", url);
    link.setAttribute("download", "Sommerfest_Planung.csv");
    document.body.appendChild(link);
    link.click();
    document.body.removeChild(link);
}

// App initialisieren
window.onload = loadData;

</script>
</body>
</html>

```