

Hier ist die komplett lauffähige Web-Anwendung in einer einzigen HTML-Datei. Sie nutzt den localStorage des Browsers zur automatischen Speicherung der Daten und beinhaltet österreichische Vereinsbegriffe (wie Schank, Grillerei, Kassa oder Bosna) als vorbefüllte Beispieldaten.

Speichere den folgenden Code einfach als sommerfest-planer.html ab und öffne die Datei per Doppelklick in einem beliebigen Browser:

```
HTML
<!DOCTYPE html>
<html lang="de-AT">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title> Sommerfest-Planer | Sportverein Österreich </title>
</style>
:root {
  --primary: #c8102e;
  --primary-dark: #9e0b22;
  --bg: #f4f5f7;
  --card: #ffffff;
  --text: #1f2937;
  --muted: #6b7280;
  --border: #e5e7eb;
  --success: #16a34a;
  --warning: #ca8a04;
  --danger: #dc2626;
}
* { box-sizing: border-box; font-family: system-ui, -apple-system, sans-serif; }
body { margin: 0; background: var(--bg); color: var(--text); padding-bottom: 40px; }
header {
  background: var(--primary);
  color: white;
  padding: 1rem 2rem;
  display: flex;
  justify-content: space-between;
  align-items: center;
  flex-wrap: wrap;
  gap: 1rem;
  box-shadow: 0 2px 6px rgba(0,0,0,0.15);
}
header h1 { margin: 0; font-size: 1.4rem; display: flex; align-items: center; gap: 0.6rem; }
.header-actions { display: flex; gap: 0.5rem; flex-wrap: wrap; }
button {
```

```

    cursor: pointer;
    border: none;
    border-radius: 6px;
    padding: 0.55rem 0.95rem;
    font-weight: 600;
    font-size: 0.9rem;
    transition: background 0.15s, opacity 0.15s;
  }

  .btn-primary { background: var(--primary); color: white; }
  .btn-primary:hover { background: var(--primary-dark); }
  .btn-light { background: white; color: var(--primary); }
  .btn-light:hover { background: #f3f4f6; }
  .btn-danger { background: var(--danger); color: white; padding: 0.35rem 0.65rem; font-size: 0.8rem; }
  .btn-sm { padding: 0.35rem 0.65rem; font-size: 0.8rem; background: #e5e7eb; color: var(--text); }
  .btn-sm:hover { background: #d1d5db; }
  .container { max-width: 1150px; margin: 1.5rem auto; padding: 0 1rem; }
  .tabs { display: flex; gap: 0.5rem; margin-bottom: 1.5rem; border-bottom: 2px solid var(--border); padding-bottom: 0.5rem; flex-wrap: wrap; }
  .tab-btn { background: transparent; color: var(--muted); font-size: 1rem; padding: 0.5rem 1rem; }
  .tab-btn.active { background: var(--primary); color: white; }
  .card { background: var(--card); border-radius: 8px; padding: 1.25rem; box-shadow: 0 1px 3px rgba(0,0,0,0.08); margin-bottom: 1.5rem; }
  .card h2 { margin-top: 0; font-size: 1.2rem; border-bottom: 1px solid var(--border); padding-bottom: 0.6rem; }
  .form-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(170px, 1fr)); gap: 0.75rem; margin-bottom: 1rem; align-items: end; }
  label { display: block; font-size: 0.8rem; font-weight: 600; color: var(--muted); margin-bottom: 0.25rem; }
  input, select {
    width: 100%;
    padding: 0.5rem;
    border: 1px solid var(--border);
    border-radius: 6px;
    font-size: 0.9rem;
  }

  table { width: 100%; border-collapse: collapse; margin-top: 0.75rem; font-size: 0.92rem; }
  th, td { text-align: left; padding: 0.65rem; border-bottom: 1px solid var(--border); }
  th { background: #f9fafb; color: var(--muted); font-size: 0.8rem; text-transform: uppercase; }
  .badge { padding: 0.2rem 0.5rem; border-radius: 99px; font-size: 0.75rem; font-weight: 600; display: inline-block; }
  .badge-ok { background: #d1ecf1; color: var(--success); }
  .badge-warn { background: #fff3cd; color: var(--warning); }
  .hidden { display: none; }
  .done-row td { text-decoration: line-through; color: var(--muted); }
  .summary-grid { display: grid; grid-template-columns: repeat(auto-fit, minmax(220px, 1fr)); gap: 1rem; margin-bottom: 1.5rem; }
  .stat-box { background: white; padding: 1rem; border-radius: 8px; border-left: 4px solid

```

```

var(--primary); box-shadow: 0 1px 2px rgba(0,0,0,0.06); }
.stat-box span { font-size: 0.8rem; color: var(--muted); display: block; }
.stat-box strong { font-size: 1.5rem; }
@media print {
  header .header-actions, .tabs, form, .no-print { display: none !important; }
  .hidden { display: block !important; }
  body { background: white; }
  .card { box-shadow: none; border: 1px solid #ccc; page-break-inside: avoid; }
}
</style>
</head>
<body>

```

```

<header>
  <h1>  Vereins Sommerfest-Plan </h1>
  <div class="header-actions">
    <button class="btn-light" onclick="exportCSV()">  CSV Export </button>
    <button class="btn-light" onclick="exportJSON()">  Backup JSON </button>
    <button class="btn-light" onclick="document.getElementById('importFile').click()">  Backup
    </button>
    <input type="file" id="importFile" class="hidden" accept=".json" onchange="importJSON(event)">
    <button class="btn-light" onclick="window.print()">  Drucken </button>
    <button class="btn-danger" onclick="resetData()">  Reset </button>
  </div>
</header>

```

```

<div class="container">
  <div class="summary-grid" id="stats"></div>

```

```

<div class="tabs">
  <button class="tab-btn active" onclick="switchTab('shifts', this)">  Helfer-Schichtplan </button>
  <button class="tab-btn" onclick="switchTab('tasks', this)">  Aufgabenliste </button>
  <button class="tab-btn" onclick="switchTab('inventory', this)">  Material & Bestand </button>
</div>

```

```

<!-- SCHICHTPLAN -->
<section id="tab-shifts" class="card">
  <h2> Helfer-Schichtplan </h2>
  <form onsubmit="addShift(event)" class="form-grid">
    <div>
      <label> Station / Bereich </label>
      <select id="shiftStation" required>
        <option value="Schank / Bierwagen"> Schank / Bierwagen </option>

```

```

        <option value="Grillerei & Bosna"> Grillerei & Bosna </option>
        <option value="Kassa & Wertmarken"> Kassa & Wertmarken </option>
        <option value="Kuchen & Kaffee"> Kuchen & Kaffee </option>
        <option value="Auf-/Abbau & Platzdienst"> Auf-/Abbau & Platzdienst </option>
    </select>
</div>
<div>
    <label> Ihrzeit / Von - Bis </label>
    <input type="text" id="shiftTime" placeholder="z.B. 14:00 - 18:00" required>
</div>
<div>
    <label> Name Helfer/K </label>
    <input type="text" id="shiftHelper" placeholder="Vor- und Nachname" required>
</div>
<div>
    <label> Telefon / Kontakt </label>
    <input type="text" id="shiftPhone" placeholder="+43 664 ..." >
</div>
<div>
    <button type="submit" class="btn-primary" style="width:100%"> Schicht eintragen </button>
</div>
</form>

```

```

<table>
  <thead>
    <tr>
      <th> Station </th>
      <th> Zeitraum </th>
      <th> Helfer/K </th>
      <th> Kontakt </th>
      <th class="no-print"> Aktion </th>
    </tr>
  </thead>
  <tbody id="shiftsBody"></tbody>
</table>
</section>

```

```

<!-- AUFGABENLISTE -->
<section id="tab-tasks" class="card hidden">
  <h2> Aufgabenliste (To-Do) </h2>
  <form onsubmit="addTask(event)" class="form-grid">
    <div>
      <label> Aufgabe </label>

```

```

        <input type="text" id="taskTitle" placeholder="z.B. AKM-Meldung abschicken" required>
    </div>
    <div>
        <label> Zuständig </label>
        <input type="text" id="taskOwner" placeholder="z.B. Obmann / Kassier" required>
    </div>
    <div>
        <label> fällig bis </label>
        <input type="date" id="taskDue" required>
    </div>
    <div>
        <button type="submit" class="btn-primary" style="width:100%"> Aufgabe
        hinzufügen </button>
    </div>
</form>

```

```

<table>
  <thead>
    <tr>
      <th> Status </th>
      <th> Aufgabe </th>
      <th> Zuständig </th>
      <th> Frist </th>
      <th class="no-print"> Aktion </th>
    </tr>
  </thead>
  <tbody id="tasksBody"></tbody>
</table>
</section>

```

```

<!-- MATERIALBESTAND -->
<section id="tab-inventory" class="card hidden">
  <h2> Materialbestand & Einkauf </h2>
  <form onsubmit="addInventory(event)" class="form-grid">
    <div>
      <label> Artikel / Material </label>
      <input type="text" id="invItem" placeholder="z.B. Bierfässer 50L / Garnituren" required>
    </div>
    <div>
      <label> Kategorie </label>
      <select id="invCategory">
        <option value="Getränke"> Getränke </option>
        <option value="Speisen"> Speisen </option>
      </select>
    </div>
  </form>

```

```

    <option value="Equipment / Zelt">Equipment / Zelt </option>
    <option value="Sonstiges">sonstige </option>
  </select>
</div>
<div>
  <label>Flüssigkeit-Bestand </label>
  <input type="number" id="invCurrent" min="0" value="0" required>
</div>
<div>
  <label>Zoll-Bestand </label>
  <input type="number" id="invTarget" min="1" value="10" required>
</div>
<div>
  <label>Einheit </label>
  <input type="text" id="invUnit" placeholder="Stk / Kisten / Fass" required>
</div>
<div>
  <button type="submit" class="btn-primary" style="width:100%"> Artikel anlegen </button>
</div>
</form>

```

```

<table>
  <thead>
    <tr>
      <th>Artikel </th>
      <th>Kategorie </th>
      <th>Stk / Soll </th>
      <th>Status </th>
      <th class="no-print">Bestand ändern </th>
    </tr>
  </thead>
  <tbody id="inventoryBody"></tbody>
</table>
</section>
</div>

```

```

<script>
const STORAGE_KEY = 'sv_sommerfest_data_v1';

const defaultData = {
  shifts: [
    { id: 1, station: 'Schank / Bierwagen', time: '11:00 - 15:00', helper: 'Markus Gruber', phone: '0664 1234567' },
    { id: 2, station: 'Grillerei & Bosna', time: '11:30 - 15:30', helper: 'Stefan Hofer', phone: '0676

```

```

9876543' },
  { id: 3, station: 'Kassa & Wertmarken', time: '11:00 - 15:00', helper: 'Sabine Leitner', phone: '0650
1122334' }
],
tasks: [
  { id: 1, title: 'Veranstaltung bei der Gemeinde anmelden', owner: 'Obmann', due: '2026-06-01',
done: true },
  { id: 2, title: 'Bierwagen & Durchlaufkühler bei Brauerei reservieren', owner: 'Platzwart', due:
'2026-06-15', done: false },
  { id: 3, title: 'Wechselgeld für Kassa von der Bank holen', owner: 'Kassier', due: '2026-07-02',
done: false }
],
inventory: [
  { id: 1, item: 'Märzenbier 50L Fass', category: 'Getränke', current: 4, target: 12, unit: 'Fass' },
  { id: 2, item: 'Bierzeltgarnituren', category: 'Equipment / Zelt', current: 30, target: 30, unit: 'Sets' },
  { id: 3, item: 'Bratwürstel & Käsekrainer', category: 'Speisen', current: 0, target: 250, unit:
'Portionen' }
]
};

```

```
let state = loadData();
```

```

function loadData() {
  const raw = localStorage.getItem(STORAGE_KEY);
  if (!raw) return JSON.parse(JSON.stringify(defaultData));
  try {
    return JSON.parse(raw);
  } catch (e) {
    return JSON.parse(JSON.stringify(defaultData));
  }
}

```

```

function saveData() {
  localStorage.setItem(STORAGE_KEY, JSON.stringify(state));
  renderAll();
}

```

```

function switchTab(tab, btn) {
  document.querySelectorAll('section.card').forEach(s => s.classList.add('hidden'));
  document.getElementById('tab-' + tab).classList.remove('hidden');
  document.querySelectorAll('tab-btn').forEach(b => b.classList.remove('active'));
  btn.classList.add('active');
}

```

```

function escapeHTML(str) {
  return String(str).replace(/[\&<>"]/g, m => ({
    '&': '&', '<': '<', '>': '>', '"': '"', "'": "'"
  }[m]));
}

```

```
}
```

```
// --- SCHICHTEN ---
```

```
function addShift(e) {  
  e.preventDefault();  
  state.shifts.push({  
    id: Date.now(),  
    station: document.getElementById('shiftStation').value,  
    time: document.getElementById('shiftTime').value.trim(),  
    helper: document.getElementById('shiftHelper').value.trim(),  
    phone: document.getElementById('shiftPhone').value.trim()  
  });  
  e.target.reset();  
  saveData();  
}
```

```
function deleteShift(id) {  
  state.shifts = state.shifts.filter(s => s.id !== id);  
  saveData();  
}
```

```
// --- AUFGABEN ---
```

```
function addTask(e) {  
  e.preventDefault();  
  state.tasks.push({  
    id: Date.now(),  
    title: document.getElementById('taskTitle').value.trim(),  
    owner: document.getElementById('taskOwner').value.trim(),  
    due: document.getElementById('taskDue').value,  
    done: false  
  });  
  e.target.reset();  
  saveData();  
}
```

```
function toggleTask(id) {  
  const t = state.tasks.find(x => x.id === id);  
  if (t) t.done = !t.done;  
  saveData();  
}
```

```
function deleteTask(id) {  
  state.tasks = state.tasks.filter(x => x.id !== id);  
  saveData();  
}
```

```
// --- MATERIAL ---
```

```
function addInventory(e) {
```

```

    e.preventDefault();
    state.inventory.push({
      id: Date.now(),
      item: document.getElementById('invItem').value.trim(),
      category: document.getElementById('invCategory').value,
      current: Number(document.getElementById('invCurrent').value),
      target: Number(document.getElementById('invTarget').value),
      unit: document.getElementById('invUnit').value.trim()
    });
    e.target.reset();
    saveData();
  }

```

```

function adjustStock(id, delta) {
  const item = state.inventory.find(i => i.id === id);
  if (item) {
    item.current = Math.max(0, item.current + delta);
    saveData();
  }
}

```

```

function deleteInventory(id) {
  state.inventory = state.inventory.filter(i => i.id !== id);
  saveData();
}

```

```

// --- RENDER ---
function renderAll() {
  // Statistik
  const openTasks = state.tasks.filter(t => !t.done).length;
  const missingItems = state.inventory.filter(i => i.current < i.target).length;
  document.getElementById('stats').innerHTML = `
    <div class="stat-box"><span>Eingeteilte
Schichten</span><strong>${state.shifts.length}</strong></div>
    <div class="stat-box"><span>Offene Aufgaben</span><strong>${openTasks} /
${state.tasks.length}</strong></div>
    <div class="stat-box"><span>Artikel
nachzubestellen</span><strong>${missingItems}</strong></div>
  `;
}

```

```

// Schichten
const shiftsBody = document.getElementById('shiftsBody');
shiftsBody.innerHTML = state.shifts.map(s => `
  <tr>
    <td><strong>${escapeHTML(s.station)}</strong></td>
    <td>${escapeHTML(s.time)}</td>
    <td>${escapeHTML(s.helper)}</td>
    <td>${escapeHTML(s.phone || '-')}</td>
  </tr>
`);

```

```

        <td class="no-print"><button class="btn-danger"
onclick="deleteShift(${s.id})">Löschen</button></td>
    </tr>
    `).join("");

// Aufgaben
const tasksBody = document.getElementById('tasksBody');
tasksBody.innerHTML = state.tasks.map(t => `
    <tr class="${t.done ? 'done-row' : ''}">
        <td><input type="checkbox" ${t.done ? 'checked' : ''} onchange="toggleTask(${t.id})"
style="width:18px;height:18px;cursor:pointer;"></td>
        <td>${escapeHTML(t.title)}</td>
        <td>${escapeHTML(t.owner)}</td>
        <td>${escapeHTML(t.due)}</td>
        <td class="no-print"><button class="btn-danger"
onclick="deleteTask(${t.id})">Löschen</button></td>
    </tr>
    `).join("");

// Material
const invBody = document.getElementById('inventoryBody');
invBody.innerHTML = state.inventory.map(i => {
    const ok = i.current >= i.target;
    const diff = Math.max(0, i.target - i.current);
    return `
        <tr>
            <td><strong>${escapeHTML(i.item)}</strong></td>
            <td>${escapeHTML(i.category)}</td>
            <td>${i.current} / ${i.target} ${escapeHTML(i.unit)}</td>
            <td>
                <span class="badge ${ok ? 'badge-ok' : 'badge-warn'}">
                    ${ok ? 'Komplett' : 'Fehlt: ' + diff + ' ' + escapeHTML(i.unit)}
                </span>
            </td>
            <td class="no-print" style="display:flex;gap:0.35rem;">
                <button class="btn-sm" onclick="adjustStock(${i.id}, -1)">-1</button>
                <button class="btn-sm" onclick="adjustStock(${i.id}, 1)">+1</button>
                <button class="btn-danger" onclick="deleteInventory(${i.id})">X</button>
            </td>
        </tr>
        `;
    }).join("");
}

// --- EXPORT & IMPORT ---
function exportCSV() {
    let csv = `uFEFFBereich;Eintrag 1;Eintrag 2;Eintrag 3;Status/Details\n`;
    state.shifts.forEach(s => {

```

```

    csv += `Schichtplan;${s.station}";"${s.time}";"${s.helper}";"Tel: ${s.phone}"\n`;
  });
  state.tasks.forEach(t => {
    csv += `Aufgabe;${t.title}";"${t.owner}";"Frist: ${t.due}";"${t.done ? 'Erledigt' : 'Offen'}"\n`;
  });
  state.inventory.forEach(i => {
    csv += `Material;${i.item}";"${i.category}";"Ist: ${i.current} / Soll: ${i.target}";"${i.unit}"\n`;
  });

```

```

const blob = new Blob([csv], { type: 'text/csv;charset=utf-8;' });
const a = document.createElement('a');
a.href = URL.createObjectURL(blob);
a.download = 'sommerfest-export.csv';
a.click();
}

```

```

function exportJSON() {
  const blob = new Blob([JSON.stringify(state, null, 2)], { type: 'application/json' });
  const a = document.createElement('a');
  a.href = URL.createObjectURL(blob);
  a.download = 'sommerfest-backup.json';
  a.click();
}

```

```

function importJSON(event) {
  const file = event.target.files[0];
  if (!file) return;
  const reader = new FileReader();
  reader.onload = function(e) {
    try {
      const imported = JSON.parse(e.target.result);
      if (imported.shifts && imported.tasks && imported.inventory) {
        state = imported;
        saveData();
        alert('Backup erfolgreich geladen!');
      } else {
        alert('Ungültiges Dateiformat.');
```

```

      } catch (err) {
        alert('Fehler beim Lesen der Datei.');
```

```
state = loadData();
renderAll();
}
```

```
renderAll();
</script>
</body>
</html>
```

Enthaltene Funktionen:

1. **Helfer-Schichtplan:** Einteilung nach typischen Feststationen (Schank, Grillerei, Kassa etc.) mit Zeitfenster und Kontaktdaten.
2. **Aufgabenliste (To-Do):** Zuweisung an Funktionäre (Obmann, Kassier, Platzwart), Fälligkeitsdatum und Abhak-Funktion.
3. **Materialbestand:** Soll-/Ist-Vergleich mit automatischer Berechnung der noch benötigten Einkaufsmengen sowie Schnellbuttons (+1/-1).
4. **Speicher- & Export-Logik:**
 - Speichert jede Änderung sofort im Browser (localStorage).
 - **CSV-Export** (inklusive UTF-8 BOM für direkte Darstellung von Umlauten in Excel).
 - **JSON-Backup** zum Weitergeben des Planungsstands an andere Vorstandsmitglieder.
 - **Druckansicht** (blendet Buttons und Formulare beim Ausdrucken automatisch aus).