

```

import wixData from 'wix-data';
import wixUsers from 'wix-users';
import wixLocation from 'wix-location';
import { sendAdminMail } from 'backend/emailSender.js'; // Backend-Mail-Funktion importieren

// === Einstellungen ===
const PREISE = { einzel: 25, paar: 30, familien: 35, firmen: 105 };
const DANKES_SEITEN = {
  einzel: 'https://deine-domain.de/danke-einzel',
  paar: 'https://deine-domain.de/danke-paar',
  familien: 'https://deine-domain.de/danke-familie',
  firmen: 'https://deine-domain.de/danke-firma',
  default: 'https://deine-domain.de/danke'
};
const MAX_KIDS = 5;

// === IDs ===
const IDS = {
  checks: {
    einzel: '#einzelmitgliedcheck',
    paar: '#paarmitgliedcheck',
    familien: '#familienmitgliedcheck',
    firmen: '#firmenmitgliedcheck',
  },
  sections: {
    adult: '#sectionAdult',
    famZahl: '#sectionFamZahl',
    firma: '#sectionFirma',
    kids: ['#sectionKid1', '#sectionKid2', '#sectionKid3', '#sectionKid4', '#sectionKid5'],
  },
  kidsCount: '#anzahlkinder',
  msg: '#fehlermeldung',
  status: '#statusInfo'
};

let mitgliedTyp = '';
let busy = false;

// === Hilfsfunktionen ===
const getEI = id => { try { return $w(id); } catch { return null; } };

```

```

const isShown = el => el && !el.hidden && !el.collapsed;
const safeShow = el => { if (el?.expand) el.expand(); if (el?.show) el.show(); };
const safeHide = el => { if (el?.collapse) el.collapse(); if (el?.hide) el.hide(); };

function updateTextElement(id, text, forceShow = true) {
  const el = getEl(id);
  if (!el) { console.warn(`⚠ Element ${id} nicht gefunden`); return; }
  if ('text' in el) el.text = text || "";
  if (forceShow) { if (el?.expand) el.expand(); if (el?.show) el.show(); }
  else if (!text) { if (el?.collapse) el.collapse(); if (el?.hide) el.hide(); }
}

const showStatus = t => updateTextElement(IDS.status, t, true);
const showErrorSafe = t => updateTextElement(IDS.msg, t, !!t);

function collapseAll() {
  safeHide(getEl(IDS.sections.adult));
  safeHide(getEl(IDS.sections.famZahl));
  safeHide(getEl(IDS.sections.firma));
  IDS.sections.kids.forEach(id => safeHide(getEl(id)));
}

function handleKidsSections() {
  const raw = getEl(IDS.kidsCount)?.value ?? '0';
  const n = Math.max(0, Math.min(MAX_KIDS, parseInt(raw, 10) || 0));
  IDS.sections.kids.forEach(id => safeHide(getEl(id)));
  for (let i = 0; i < n; i++) safeShow(getEl(IDS.sections.kids[i]));
}

function applyTypeImmediately(type) {
  collapseAll();
  if (!type) return checkFormValid();
  if (type === 'paar') {
    safeShow(getEl(IDS.sections.adult));
  } else if (type === 'familien') {
    safeShow(getEl(IDS.sections.adult));
    safeShow(getEl(IDS.sections.famZahl));
    handleKidsSections();
  } else if (type === 'firmen') {
    safeShow(getEl(IDS.sections.firma));
  }
  checkFormValid();
}

```

```

function getCheckedType() {
  if (getEl(IDS.checks.einzel)?.checked) return 'einzel';
  if (getEl(IDS.checks.paar)?.checked) return 'paar';
  if (getEl(IDS.checks.familien)?.checked) return 'familien';
  if (getEl(IDS.checks.firmen)?.checked) return 'firmen';
  return "";
}

function setCheckedType(type) {
  const set = (id, on) => { const el = getEl(id); if (el && typeof el.checked === 'boolean') el.checked = on; };
  set(IDS.checks.einzel, type === 'einzel');
  set(IDS.checks.paar, type === 'paar');
  set(IDS.checks.familien, type === 'familien');
  set(IDS.checks.firmen, type === 'firmen');
}

function onTypeCheckboxChange(id) {
  const clicked = getEl(id);
  if (!clicked) return;
  if (clicked.checked) {
    const newType = {
      [IDS.checks.einzel]: 'einzel',
      [IDS.checks.paar]: 'paar',
      [IDS.checks.familien]: 'familien',
      [IDS.checks.firmen]: 'firmen'
    };
    setCheckedType(newType);
    mitgliedTyp = newType;
  } else {
    mitgliedTyp = getCheckedType();
  }
  applyTypImmediately(mitgliedTyp);
}

function checkFormValid() {
  let ok = true;
  const fehlende = [];
  if (!mitgliedTyp) { fehlende.push('Mitgliedstyp auswählen'); ok = false; }
  const check = (id, label) => {
    const el = getEl(id);
    if (el && !isShown(el) && !el.value) fehlende.push(label || id);
  }
}

```

```

};

check('#vornamehost', 'Vorname');
check('#nachnamehost', 'Nachname');
check('#emailhost', 'E-Mail');
check('#telehost', 'Telefon');
check('#geburstaghost', 'Geburtstag');
check('#adressealle', 'Adresse');
check('#plzalle', 'PLZ');
check('#ortalle', 'Ort');
check('#landalle', 'Land');
check('#passwordInput', 'Passwort');

if (mitgliedTyp === 'paar' || mitgliedTyp === 'familien') {
  check('#vornameadult', 'Vorname Partner');
  check('#nachnameadult', 'Nachname Partner');
  check('#geburstagadult', 'Geburtstag Partner');
}

if (mitgliedTyp === 'familien') {
  const n = Math.max(0, Math.min(MAX_KIDS, parseInt(getEl(IDS.kidsCount)?.value || '0', 10) || 0));
  for (let i=1; i<=n; i++) {
    check(`#vornamekid${i}`, `Vorname Kind ${i}`);
    check(`#nachnamekid${i}`, `Nachname Kind ${i}`);
    check(`#geburstagkid${i}`, `Geburtstag Kind ${i}`);
  }
}

if (mitgliedTyp === 'firmen') check('#firmaname', 'Firmenname');

ok = ok && fehlende.length === 0;
if (fehlende.length) {
  showErrorSafe(`Fehlend: ${fehlende.join(', ')}`);
  showStatus('⚠ Bitte alle Pflichtfelder ausfüllen!');
} else {
  showErrorSafe('');
  showStatus('✅ Alles ausgefüllt – kann gespeichert werden');
}

return ok;
}

async function getNextMitgliedNummer() {

```

```

const res = await wixData.query('MemberData')
  .descending('mitgliedNummer')
  .limit(1)
  .find();
let maxNum = 0;
if (res.items.length > 0) {
  maxNum = parseInt(res.items[0].mitgliedNummer, 10) || 0;
}
return maxNum + 1;
}

function getMitNr({ num, typ, isHost=false, kidIndex=null, isAdultPartner=false }) {
  if (typ==='einzeln') return `${num}*`;
  if (typ==='paar') return isHost ? `${num} Pa*` : `${num} Pa`;
  if (typ==='familien'){
    if (isHost) return `${num} Fa*`;
    if (isAdultPartner) return `${num} Fa`;
    const labels = ['a','b','c','d','e'];
    return `${num} Fa ${labels[(kidIndex||1)-1]}`;
  }
  if (typ==='firmen') return `${num} Fi*`;
  return `${num}`;
}

$w.onReady(() => {
  collapseAll();
  showStatus('Bitte Formular ausfüllen.');
```

```

  showErrorSafe("");

  if (getEl(IDS.checks.einzeln)) $w(IDS.checks.einzeln).onChange(() =>
onTypeCheckboxChange(IDS.checks.einzeln));
  if (getEl(IDS.checks.paar)) $w(IDS.checks.paar).onChange(() =>
onTypeCheckboxChange(IDS.checks.paar));
  if (getEl(IDS.checks.familien)) $w(IDS.checks.familien).onChange(() =>
onTypeCheckboxChange(IDS.checks.familien));
  if (getEl(IDS.checks.firmen)) $w(IDS.checks.firmen).onChange(() =>
onTypeCheckboxChange(IDS.checks.firmen));

  const kc = getEl(IDS.kidsCount);
  if (kc?.onInput) kc.onInput(() => { if (mitgliedTyp==='familien') { handleKidsSections(); checkFormValid(); } });

```

```

if (kc?.onChange) kc.onChange(() => { if (mitgliedTyp==='familien') { handleKidsSections(); checkFormValid(); }
});

const inputs = [
  '#vornamehost', '#nachnamehost', '#emailhost', '#telehost', '#geburstaghost',
  '#adressealle', '#plzalle', '#ortalle', '#landalle', '#passwordInput',
  '#vornameadult', '#nachnameadult', '#geburstagadult', '#firmaname'
];
for (let i=1; i<=MAX_KIDS; i++) {
  inputs.push(`#vornamekid${i}`, `#nachnamekid${i}`, `#geburstagkid${i}`);
}
inputs.forEach(id => {
  const el = getEl(id);
  if (el?.onInput) el.onInput(checkFormValid);
  if (el?.onChange) el.onChange(checkFormValid);
});

const pre = getCheckedType();
if (pre) {
  mitgliedTyp = pre;
  applyTypelmmmediately(pre);
}

if (getEl('#btnSpeichern')) {
  $w('#btnSpeichern').onClick(() => {
    console.log('💡 Button-Handler ausgelöst');
    btnSpeichern_click();
  });
}
});

export async function btnSpeichern_click() {
  console.log("💡 Speichern-Button geklickt");
  showStatus("⌛ Button wurde gedrückt – prüfe Eingaben ...");

  if (busy) {
    showStatus("⚠️ Speichern läuft bereits – bitte warten");
    return;
  }
}

```

```

if (!checkFormValid()) {
  console.warn("⊖ Pflichtfelder fehlen!");
  return;
}

busy = true;
showStatus('💾 Speichern... Benutzer wird angelegt.');
```



```

try {
  const email = $w('#emailhost').value;
  const pass = $w('#passwordInput').value;
  await wixUsers.register(email, pass);
  await wixUsers.login(email, pass);
  const loginId = wixUsers.currentUser.id;

  const neuNummer = await getNextMitgliedNummer();
  const preis = PREISE[mitgliedTyp];
  const payId = (mitgliedTyp === 'familien') ? `familyId_${neuNummer}` : `paymentId_${neuNummer}`;

  const baseAddr = {
    adresse: $w('#adressealle').value,
    plz: $w('#plzalle').value,
    ort: $w('#ortalle').value,
    land: $w('#landalle').value,
  };

  const host = {
    vorname: $w('#vornamehost').value,
    nachname: $w('#nachnamehost').value,
    email,
    telefon: $w('#telehost').value,
    geburstag: $w('#geburstaghost').value,
    ...baseAddr
  };

  const rows = [];
  rows.push({
    ...host,
    mitgliedTyp,
    gesperrt: true,
    preis,
  });
}

```

```

    mitgliedNummer: getMitNr({ num: neuNummer, typ: mitgliedTyp, isHost: true }),
    host: true,
    loginId: loginId,
    paymentId: payId,
    firmenname: mitgliedTyp === 'firmen' ? $w('#firmaname').value : undefined,
  });

  if (mitgliedTyp === 'paar') {
    rows.push({
      vorname: $w('#vornameadult').value,
      nachname: $w('#nachnameadult').value,
      email: $w('#emailadult').value,
      geburtstag: $w('#geburtstagadult').value,
      ...baseAddr,
      mitgliedTyp,
      gesperrt: true,
      preis,
      mitgliedsnummer: getMitNr({ num: neuNummer, typ: 'paar' }),
      host: false,
      loginId: loginId,
      paymentId: `paymentId_${neuNummer}`,
    });
  }

  if (mitgliedTyp === 'familien') {
    rows.push({
      vorname: $w('#vornameadult').value,
      nachname: $w('#nachnameadult').value,
      email: $w('#emailadult').value,
      geburtstag: $w('#geburtstagadult').value,
      ...baseAddr,
      mitgliedTyp,
      gesperrt: true,
      preis,
      mitgliedsnummer: getMitNr({ num: neuNummer, typ: 'familien', isAdultPartner: true }),
      host: false,
      loginId: loginId,
      paymentId: `familyId_${neuNummer}`,
    });
  }

```

```

const n = Math.max(
  0,
  Math.min(MAX_KIDS, parseInt($w(IDS.kidsCount).value || '0', 10) || 0)
);
for (let i = 1; i <= n; i++) {
  rows.push({
    vorname: $w(`#vornamekid${i}`).value,
    nachname: $w(`#nachnamekid${i}`).value,
    email: $w(`#emailkid${i}`).value,
    geburstag: $w(`#geburstagkid${i}`).value,
    ...baseAddr,
    mitgliedTyp,
    gesperrt: true,
    preis,
    mitgliedsnummer: getMitNr({ num: neuNummer, typ: 'familien', kidIndex: i }),
    host: false,
    loginId: loginId,
    paymentId: `familyId_${neuNummer}`,
  });
}
}

for (const r of rows) {
  await wixData.insert('MemberData', r);
}
await wixData.insert('PaymentBase', {
  userId: loginId,
  paymentId: payId,
  betrag: preis,
});

showStatus('✅ Erfolgreich gespeichert. Weiterleitung in 5 Sekunden...');
setTimeout(() => {
  const url = DANKES_SEITEN[mitgliedTyp] || DANKES_SEITEN.default;
  wixLocation.to(`${url}?loginId=${encodeURIComponent(loginId)}`);
}, 5000);
} catch (e) {
  console.error(e);
  showStatus('❌ Fehler: ${e.message || 'Unbekannter Fehler'}');
} finally {

```

```
    busy = false;  
  }  
}
```