```
<!doctype html>
<html lang="ja">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>漢方日記</title>

  <!-- iOS ホーム画面用（最低限） -->
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-mobile-web-app-title" content="漢方日記">
  <meta name="theme-color" content="#ffffff">

  <style>
    :root { --bg:#f6f7f8; --card:#fff; --text:#111; --muted:#666; --line:#e6e6e6; }
    body { margin:0; font-family: -apple-system, BlinkMacSystemFont, "Hiragino Sans", "Noto
Sans JP", Segoe UI, Roboto, sans-serif; color:var(--text); background:var(--bg); }
    header { position: sticky; top:0; background:rgba(246,247,248,.92); backdrop-filter:
blur(10px); border-bottom:1px solid var(--line); padding:12px 14px; }
    header .row { display:flex; align-items:center; justify-content:space-between; gap:10px; }
    h1 { margin:0; font-size:16px; font-weight:700; }
    .sub { font-size:12px; color:var(--muted); margin-top:3px; }
    main { padding: 12px 12px 90px; max-width: 720px; margin:0 auto; }
    .card { background:var(--card); border:1px solid var(--line); border-radius:14px;
padding:12px; margin:10px 0; box-shadow: 0 1px 0 rgba(0,0,0,.02); }
    .card h2 { margin:0 0 10px; font-size:14px; }
    .pillbar { display:flex; flex-wrap:wrap; gap:8px; }
    .pill { border:1px solid var(--line); background:#fff; padding:10px 12px;
border-radius:999px; font-size:13px; user-select:none; cursor:pointer; }
    .pill.active { border-color:#111; }
    .pill.ghost { color:var(--muted); border-style:dashed; }
    .grid { display:grid; grid-template-columns: 1fr 1fr; gap:10px; }
    label { display:block; font-size:12px; color:var(--muted); margin-bottom:6px; }
    input[type="number"], input[type="text"], textarea, select {
      width:100%; box-sizing:border-box; padding:10px 12px; border-radius:12px; border:1px
solid var(--line); background:#fff; font-size:14px;
    }
    textarea { min-height: 110px; resize: vertical; }
    .row { display:flex; gap:10px; align-items:center; }
    .row > * { flex:1; }
    .btn {
      border:1px solid var(--line); background:#fff; padding:10px 12px; border-radius:12px;
      font-size:14px; cursor:pointer;
    }
    .btn.primary { border-color:#111; }
    .btn.danger { border-color:#c62828; color:#c62828; }
    .muted { color:var(--muted); font-size:12px; }
    .divider { height:1px; background:var(--line); margin:12px 0; }
    /* 追記ボタン固定 */
    .fab {
      position: fixed; left: 12px; right: 12px; bottom: 14px;
      max-width: 720px; margin: 0 auto;
      display:flex; gap:10px;
    }
    .fab .btn { flex:1; padding:14px 12px; border-radius:14px; }
    .log { border:1px solid var(--line); border-radius:12px; padding:10px 12px; background:#fff;
margin-top:8px; }
    .loghead { display:flex; justify-content:space-between; align-items:center; gap:10px; }
    .logtime { font-size:12px; color:var(--muted); }
    .logtext { white-space:pre-wrap; margin-top:6px; font-size:14px; }
    .tiny { font-size:11px; color:var(--muted); }
    .tag { display:inline-block; font-size:11px; border:1px solid var(--line); padding:2px 8px;
border-radius:999px; color:var(--muted); }
  </style>
</head>
<body>
<header>
  <div class="row">
    <div>
      <h1>漢方日記</h1>
      <div class="sub" id="todayLabel"></div>
    </div>
    <button class="btn primary" id="btnSave">保存</button>
  </div>
</header>

<main>
  <div class="card">
    <h2>状態（1つだけ）</h2>
    <div class="pillbar" id="conditionBar"></div>
    <div class="muted" style="margin-top:8px;">良好/普通の日はここだけでもOK。</div>
  </div>

  <div class="card">
    <h2>症状（複数）</h2>
    <div class="pillbar" id="symptomBar"></div>
    <div class="muted" style="margin-top:8px;">やや不調のときに使う想定。選ばなくても保存
OK。</div>
  </div>

  <div class="card">
    <h2>体温</h2>
    <div class="grid">
      <div>
        <label>体温（℃）</label>
        <input type="number" inputmode="decimal" step="0.1" placeholder="36.5"
id="tempInput" />
      </div>
      <div>
        <label>生理</label>
        <select id="menstrualSelect">
          <option value="">未選択</option>
          <option value="pre">前</option>
          <option value="in">中</option>
          <option value="post">後</option>
        </select>
      </div>
    </div>
  </div>

  <div class="card">
    <div class="row" style="align-items:flex-end;">
      <div>
        <h2 style="margin:0;">今日服用した漢方</h2>
        <div class="tiny">「服用中」だけが基本表示。必要なら休薬中も表示。</div>
      </div>
      <button class="btn" id="togglePaused">休薬中を表示</button>
    </div>

    <div class="divider"></div>

    <div class="muted">服用中</div>
    <div class="pillbar" id="kActiveBar" style="margin-top:8px;"></div>

    <div id="pausedWrap" style="display:none; margin-top:12px;">
      <div class="muted">休薬中</div>
      <div class="pillbar" id="kPausedBar" style="margin-top:8px;"></div>
    </div>

    <div class="divider"></div>

    <div class="row">
      <div>
        <label>漢方を追加（将来の処方もここで増やせる）</label>
        <input type="text" id="kNewName" placeholder="例：当帰芍薬散" />
      </div>
      <button class="btn primary" id="btnAddKampo">追加</button>
    </div>
    <div class="muted" style="margin-top:8px;">
      追加した漢方は「服用中」で登録されます（あとで休薬中/終了に変更できます）。
    </div>
  </div>
  <div class="card">
    <h2>体調日記（その日のまとめ：長文OK）</h2>
    <textarea id="summaryText" placeholder="夜にまとめを書いてもOK。空でもOK。
"></textarea>
  </div>

  <div class="card">
    <div class="row" style="align-items:center;">
      <h2 style="margin:0; flex:1;">追記ログ（変化があったら足す）</h2>
      <span class="tag" id="logCount">0件</span>
    </div>
    <div class="muted" style="margin-top:6px;">下の「＋追記」でいつでも追加できます。</div>
    <div id="logList" style="margin-top:10px;"></div>
  </div>

  <div class="card">
    <h2>漢方の管理（ステータス変更）</h2>
    <div class="muted">服用中 / 休薬中 / 終了（履歴保持のため削除はしない設計）</div>
    <div id="kManageList" style="margin-top:10px;"></div>
  </div>

  <div class="card">
    <h2>データ</h2>
    <div class="row">
      <button class="btn" id="btnExport">エクスポート</button>
      <button class="btn" id="btnImport">インポート</button>
    </div>
    <div class="muted" style="margin-top:8px;">
      自分用なので端末に保存（localStorage）しています。機種変更が不安なら、たまにエクス
ポートして保存してね。
    </div>
  </div>
</main>

<div class="fab">
  <button class="btn primary" id="btnAppend">＋追記する</button>
  <button class="btn" id="btnTodayClear" title="今日の入力だけ初期化（記録自体は残る）">今
日をリセット</button>
</div>

<script>
(() => {
  const TZ = "Asia/Tokyo";
  const LS_KEY = "kampo_diary_v01";

  const CONDITIONS = [
    { id:"good", label:"🌿 良好", score:"G" },
    { id:"normal", label:"◯ 普通", score:"O" },
    { id:"unwell", label:"△ やや不調", score:"D" }
  ];

  const SYMPTOMS = [
    { id:"sym_dizzy", label:"めまい" },
    { id:"sym_lightheaded", label:"ふらつき" },
    { id:"sym_eye_strain", label:"眼精疲労" },
    { id:"sym_headache", label:"頭痛" },
    { id:"sym_sleep", label:"睡眠" },
    { id:"sym_gi", label:"胃腸" },
    { id:"sym_menstrual", label:"生理" }
  ];

  const DEFAULT_KAMPO = [
    { id:"k_tontanto", name:"温胆湯", status:"active" },
    { id:"k_kangen", name:"還元清血飲", status:"active" },
    { id:"k_saikosokan", name:"柴胡疏肝湯", status:"active" },
    { id:"k_kogiku", name:"小菊地黄丸", status:"paused" }
  ];

  const $ = (id) => document.getElementById(id);

  function jstDateParts(d = new Date()) {
    // JSTの年月日を安定して作る
    const fmt = new Intl.DateTimeFormat("ja-JP", { timeZone: TZ, year:"numeric",
month:"2-digit", day:"2-digit", weekday:"short" });
    const parts = fmt.formatToParts(d);
    const map = {};
    for (const p of parts) map[p.type] = p.value;
    return {
      y: map.year,
      m: map.month,
      d: map.day,
      wd: map.weekday,
      iso: `${map.year}-${map.month}-${map.day}`
    };
  }

  function nowJSTIso() {
    // JST表記のISOっぽい文字列（オフセット +09:00 固定）
    const d = new Date();
    const fmt = new Intl.DateTimeFormat("sv-SE", {
      timeZone: TZ, year:"numeric", month:"2-digit", day:"2-digit",
      hour:"2-digit", minute:"2-digit", second:"2-digit"
    });
    const s = fmt.format(d).replace(" ", "T");
    return `${s}+09:00`;
  }
  function uid(prefix) {
    return `${prefix}_${Math.random().toString(16).slice(2)}_${Date.now()}`;
  }

  function load() {
    const raw = localStorage.getItem(LS_KEY);
    if (!raw) {
      const init = {
        app_version:"0.1",
        master: { symptoms: SYMPTOMS, kampo: DEFAULT_KAMPO },
        records: {}
      };
      localStorage.setItem(LS_KEY, JSON.stringify(init));
      return init;
    }
    try { return JSON.parse(raw); } catch {
      alert("保存データが壊れているかも。エクスポートがあればインポートしてね。");
      return { app_version:"0.1", master:{ symptoms: SYMPTOMS, kampo: DEFAULT_KAMPO
}, records:{} };
    }
  }

  function save(db) {
    localStorage.setItem(LS_KEY, JSON.stringify(db));
  }

  function getRecord(db, dateIso) {
    if (!db.records[dateIso]) {
      db.records[dateIso] = {
        id: `rec_${dateIso}`,
        date: dateIso,
        timezone: TZ,
        condition_state: "normal",
        condition_score: "O",
        symptom_ids: [],
        menstrual_phase: "",
        temperature_c: "",
        kampo_taken_ids: [],
        daily_summary_text: "",
        append_logs: [],
        created_at: nowJSTIso(),
        updated_at: nowJSTIso()
      };
    }
    return db.records[dateIso];
  }
```

```
function setUpdated(rec) { rec.updated_at = nowJSTIso(); }

// ---- UI render ----
const today = jstDateParts();
$("todayLabel").textContent = `${today.iso}（${today.wd}）`;

const db = load();
const rec = getRecord(db, today.iso);

// Condition pills
function renderCondition() {
  const bar = $("conditionBar");
  bar.innerHTML = "";
  CONDITIONS.forEach(c => {
    const b = document.createElement("button");
    b.className = "pill" + (rec.condition_state === c.id ? " active" : "");
    b.textContent = c.label;
    b.onclick = () => {
      rec.condition_state = c.id;
      rec.condition_score = c.score;
      setUpdated(rec);
      autoSave();
      renderCondition();
    };
    bar.appendChild(b);
  });
}

// Symptoms
function renderSymptoms() {
  const bar = $("symptomBar");
  bar.innerHTML = "";
  SYMPTOMS.forEach(s => {
    const b = document.createElement("button");
    const on = rec.symptom_ids.includes(s.id);
    b.className = "pill" + (on ? " active" : "");
    b.textContent = s.label;
    b.onclick = () => {
      const idx = rec.symptom_ids.indexOf(s.id);
      if (idx >= 0) rec.symptom_ids.splice(idx, 1);
      else rec.symptom_ids.push(s.id);
      setUpdated(rec);
      autoSave();
      renderSymptoms();
    };
    bar.appendChild(b);
  });
}
```

```
function kampoByStatus(status) {
  return db.master.kampo.filter(k => k.status === status);
}

function renderKampoTaken() {
  // Active
  const activeBar = $("kActiveBar");
  activeBar.innerHTML = "";
  kampoByStatus("active").forEach(k => {
    const b = document.createElement("button");
    const on = rec.kampo_taken_ids.includes(k.id);
    b.className = "pill" + (on ? " active" : "");
    b.textContent = k.name;
    b.onclick = () => {
      const idx = rec.kampo_taken_ids.indexOf(k.id);
      if (idx >= 0) rec.kampo_taken_ids.splice(idx, 1);
      else rec.kampo_taken_ids.push(k.id);
      setUpdated(rec);
      autoSave();
      renderKampoTaken();
    };
    activeBar.appendChild(b);
  });

  // Paused (optional)
  const pausedBar = $("kPausedBar");
  pausedBar.innerHTML = "";
  kampoByStatus("paused").forEach(k => {
    const b = document.createElement("button");
    const on = rec.kampo_taken_ids.includes(k.id);
    b.className = "pill ghost" + (on ? " active" : "");
    b.textContent = k.name;
    b.onclick = () => {
      const idx = rec.kampo_taken_ids.indexOf(k.id);
      if (idx >= 0) rec.kampo_taken_ids.splice(idx, 1);
      else rec.kampo_taken_ids.push(k.id);
      setUpdated(rec);
      autoSave();
      renderKampoTaken();
    };
    pausedBar.appendChild(b);
  });
}

function renderLogs() {
  const list = $("logList");
  list.innerHTML = "";
```

```
    const logs = rec.append_logs.slice().sort((a,b)=> (a.created_at < b.created_at ? 1 : -1));
    $("logCount").textContent = `${logs.length}件`;
    logs.forEach(l => {
      const wrap = document.createElement("div");
      wrap.className = "log";
      const head = document.createElement("div");
      head.className = "loghead";
      const time = document.createElement("div");
      time.className = "logtime";
      // HH:MMだけ表示
      const hhmm = l.created_at.slice(11,16);
      time.textContent = hhmm;
      const del = document.createElement("button");
      del.className = "btn danger";
      del.style.padding = "6px 10px";
      del.textContent = "削除";
      del.onclick = () => {
        rec.append_logs = rec.append_logs.filter(x => x.id !== l.id);
        setUpdated(rec);
        autoSave();
        renderLogs();
      };
      head.appendChild(time);
      head.appendChild(del);

      const text = document.createElement("div");
      text.className = "logtext";
      text.textContent = l.text;

      wrap.appendChild(head);
      wrap.appendChild(text);
      list.appendChild(wrap);
    });
  }

  function renderManage() {
    const host = $("kManageList");
    host.innerHTML = "";
    db.master.kampo.forEach(k => {
      const row = document.createElement("div");
      row.className = "log";
      const head = document.createElement("div");
      head.className = "loghead";

      const left = document.createElement("div");
      left.innerHTML = `<div style="font-weight:600">${k.name}</div><div class="tiny">状態:
${k.status}</div>`;
```

```
    const sel = document.createElement("select");
    sel.innerHTML = `
      <option value="active">服用中</option>
      <option value="paused">休薬中</option>
      <option value="ended">終了</option>
    `;
    sel.value = k.status;
    sel.onchange = () => {
      k.status = sel.value;
      // endedになったら「今日服用」にも残ってる可能性があるので外す（履歴は残す）
      if (k.status === "ended") {
        rec.kampo_taken_ids = rec.kampo_taken_ids.filter(id => id !== k.id);
      }
      setUpdated(rec);
      autoSave();
      renderKampoTaken();
      renderManage();
    };

    head.appendChild(left);
    head.appendChild(sel);
    row.appendChild(head);
    host.appendChild(row);
  });
}

function autoSave() {
  save(db);
}

// ---- bind inputs ----
$("tempInput").value = rec.temperature_c ?? "";
$("tempInput").addEventListener("input", () => {
  rec.temperature_c = $("tempInput").value;
  setUpdated(rec);
  autoSave();
});

$("menstrualSelect").value = rec.menstrual_phase ?? "";
$("menstrualSelect").addEventListener("change", () => {
  rec.menstrual_phase = $("menstrualSelect").value;
  setUpdated(rec);
  autoSave();
});

$("summaryText").value = rec.daily_summary_text ?? "";
$("summaryText").addEventListener("input", () => {
  rec.daily_summary_text = $("summaryText").value;
```

```
    setUpdated(rec);
    autoSave();
  });

  $("btnSave").onclick = () => {
    setUpdated(rec);
    autoSave();
    alert("保存しました（端末内）");
  };

  $("togglePaused").onclick = () => {
    const w = $("pausedWrap");
    w.style.display = (w.style.display === "none") ? "block" : "none";
    $("togglePaused").textContent = (w.style.display === "none") ? "休薬中を表示" : "休薬中を
隠す";
  };

  $("btnAddKampo").onclick = () => {
    const name = $("kNewName").value.trim();
    if (!name) return;
    // 既存重複チェック（同名）
    const exists = db.master.kampo.some(k => k.name === name);
    if (exists) {
      alert("同じ名前がすでにあります");
      return;
    }
    db.master.kampo.push({ id: uid("k"), name, status:"active" });
    $("kNewName").value = "";
    setUpdated(rec);
    autoSave();
    renderKampoTaken();
    renderManage();
  };

  $("btnAppend").onclick = () => {
    const text = prompt("追記内容を入力（短文でも長文でもOK）");
    if (text === null) return;
    const t = text.trim();
    if (!t) return;
    rec.append_logs.push({ id: uid("ap"), created_at: nowJSTIso(), text: t });
    setUpdated(rec);
    autoSave();
    renderLogs();
  };

  $("btnTodayClear").onclick = () => {
    if (!confirm("今日の入力を初期化します（追記ログも含む）。よろしい？")) return;
    db.records[today.iso] = {
```

```
    id: `rec_${today.iso}`,
    date: today.iso,
    timezone: TZ,
    condition_state: "normal",
    condition_score: "O",
    symptom_ids: [],
    menstrual_phase: "",
    temperature_c: "",
    kampo_taken_ids: [],
    daily_summary_text: "",
    append_logs: [],
    created_at: nowJSTIso(),
    updated_at: nowJSTIso()
  };
  const fresh = getRecord(db, today.iso);
  // rec参照の差し替えが必要なのでページ再読み込みが簡単
  save(db);
  location.reload();
};

$("btnExport").onclick = async () => {
  const data = JSON.stringify(db, null, 2);
  try {
    await navigator.clipboard.writeText(data);
    alert("データをクリップボードにコピーしました。\nメモ等に貼り付けて保存してください。");
  } catch {
    // fallback: download
    const blob = new Blob([data], { type:"application/json" });
    const url = URL.createObjectURL(blob);
    const a = document.createElement("a");
    a.href = url;
    a.download = `kampo-diary-export-${today.iso}.json`;
    a.click();
    URL.revokeObjectURL(url);
  }
};

$("btnImport").onclick = async () => {
  const text = prompt("エクスポートしたJSONを貼り付けてください");
  if (text === null) return;
  try {
    const obj = JSON.parse(text);
    if (!obj || !obj.master || !obj.records) throw new Error("invalid");
    localStorage.setItem(LS_KEY, JSON.stringify(obj));
    alert("インポートしました。再読み込みします。");
    location.reload();
  } catch {
    alert("JSONが不正です（貼り付けミスの可能性）");
    }
  };

  // initial render
  renderCondition();
  renderSymptoms();
  renderKampoTaken();
  renderLogs();
  renderManage();
})();
</script>
</body>
</html>
```