

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>AllDrive Tow & Trailer Manager</title>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <style>
    :root {
      font-family: system-ui, -apple-system, BlinkMacSystemFont, "Segoe UI", sans-serif;
      line-height: 1.5;
    }
    body {
      margin: 0;
      background: #f3f4f6;
      color: #111827;
    }
    header {
      background: #111827;
      color: #f9fafb;
      padding: 1rem 1.25rem;
      display: flex;
      flex-direction: column;
      gap: 0.25rem;
    }
    header h1 {
      margin: 0;
      font-size: 1.3rem;
      font-weight: 600;
    }
    header p {
      margin: 0;
      font-size: 0.9rem;
      color: #d1d5db;
    }
    .nav {
      display: flex;
      flex-wrap: wrap;
      gap: 0.5rem;
      margin-top: 0.75rem;
    }
    .nav button {
      border: none;
      padding: 0.4rem 0.8rem;
      border-radius: 999px;
    }
  </style>
</head>
<body>
  <header>
    <h1>AllDrive Tow & Trailer Manager</h1>
    <p>Your Trusted Partner for Heavy Haul Solutions</p>
  </header>
  <div>
    <h2>Our Services</h2>
    <ul>
      <li>Heavy Haulage</li>
      <li>Specialized Trailers</li>
      <li>Custom Solutions</li>
      <li>Logistics Support</li>
    </ul>
  </div>
  <div>
    <h2>About Us</h2>
    <p>At AllDrive, we specialize in providing top-notch heavy haulage services. Our team of experts has years of experience in the industry, and we are committed to providing you with the best possible service. We offer a wide range of specialized trailers, including flatbeds, lowboys, and rollbacks, to handle even the most challenging loads. Our custom solutions division is also highly regarded, as we can design and build trailers to meet your specific needs. We understand that logistics is a critical part of any heavy haul operation, which is why we offer comprehensive support throughout the entire process. From planning and scheduling to delivery and beyond, we are here to help you every step of the way. Contact us today to learn more about our services and how we can help you achieve your goals. We look forward to working with you!</p>
  </div>
  <div>
    <h2>Contact Us</h2>
    <form>
      <input type="text" placeholder="Name" required>
      <input type="email" placeholder="Email" required>
      <input type="text" placeholder="Subject" required>
      <input type="text" placeholder="Message" required>
      <input type="submit" value="Send Message" style="background-color: #111827; color: white; border: none; border-radius: 999px; padding: 0.5rem; width: 100%; height: 1.5em;">
    </form>
  </div>
</body>

```

```
background: #374151;
color: #f9fafb;
font-size: 0.85rem;
cursor: pointer;
}
.nav button.active {
background: #10b981;
color: #022c22;
font-weight: 600;
}
main {
max-width: 900px;
margin: 1rem auto 2rem;
padding: 0 0.75rem;
}
.panel {
display: none;
background: #ffffff;
border-radius: 0.75rem;
padding: 1rem;
box-shadow: 0 10px 25px rgba(15, 23, 42, 0.08);
}
.panel.active {
display: block;
}
h2 {
margin-top: 0;
font-size: 1.2rem;
}
h3 {
margin-top: 1.25rem;
font-size: 1rem;
}
form {
display: grid;
gap: 0.75rem;
margin-bottom: 1rem;
}
label {
font-size: 0.85rem;
font-weight: 500;
color: #374151;
}
input, select, textarea {
```

```
margin-top: 0.25rem;
width: 100%;
padding: 0.45rem 0.55rem;
border-radius: 0.4rem;
border: 1px solid #d1d5db;
font-size: 0.9rem;
box-sizing: border-box;
}
textarea {
  resize: vertical;
  min-height: 60px;
}
.row {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
  gap: 0.75rem;
}
.btn {
  display: inline-flex;
  align-items: center;
  justify-content: center;
  gap: 0.25rem;
  padding: 0.4rem 0.8rem;
  border-radius: 999px;
  border: none;
  cursor: pointer;
  font-size: 0.9rem;
}
.btn-primary {
  background: #10b981;
  color: #022c22;
  font-weight: 600;
}
.btn-secondary {
  background: #e5e7eb;
  color: #111827;
}
.btn-danger {
  background: #ef4444;
  color: #f9fafb;
}
.list {
  margin-top: 0.5rem;
  border-top: 1px solid #e5e7eb;
```

```
padding-top: 0.5rem;
max-height: 250px;
overflow-y: auto;
}
.card {
border-radius: 0.5rem;
border: 1px solid #e5e7eb;
padding: 0.5rem 0.65rem;
margin-bottom: 0.5rem;
background: #f9fafb;
}
.card-header {
display: flex;
justify-content: space-between;
align-items: center;
font-size: 0.9rem;
font-weight: 600;
margin-bottom: 0.25rem;
}
.badge {
font-size: 0.7rem;
padding: 0.1rem 0.4rem;
border-radius: 999px;
background: #e5e7eb;
color: #374151;
}
.small {
font-size: 0.8rem;
color: #6b7280;
}
.status {
padding: 0.5rem 0.75rem;
border-radius: 0.5rem;
margin-top: 0.75rem;
font-size: 0.9rem;
}
.status.safe {
background: #ecfdf5;
color: #166534;
border: 1px solid #bbf7d0;
}
.status.warn {
background: #ffffeb;
color: #92400e;
```

```
border: 1px solid #facc15;
}
.status.danger {
background: #fef2f2;
color: #b91c1c;
border: 1px solid #fecaca;
}
.muted {
font-size: 0.8rem;
color: #9ca3af;
}
.inline-controls {
display: flex;
gap: 0.35rem;
flex-wrap: wrap;
margin-top: 0.25rem;
}
.export-box {
margin-top: 0.75rem;
padding: 0.75rem;
border-radius: 0.5rem;
border: 1px dashed #d1d5db;
background: #f9fafb;
}
@media (max-width: 600px) {
header {
padding: 0.75rem 0.75rem 0.5rem;
}
.nav {
gap: 0.35rem;
}
.nav button {
font-size: 0.8rem;
padding: 0.35rem 0.6rem;
}
}
</style>
</head>
<body>
<header>
<h1>AllDrive Tow & Trailer Manager</h1>
<p>Track tow vehicles, trailers, maintenance, and towing safety — all in one place.</p>
<div class="nav">
<button data-panel="vehicles" class="active">Vehicles</button>
```

```
<button data-panel="trailers">Trailers</button>
<button data-panel="towcalc">Tow Rating Calculator</button>
<button data-panel="diagnostics">Trailer Diagnostics</button>
<button data-panel="maintenance">Maintenance Log</button>
<button data-panel="data">Export / Import</button>
</div>
</header>

<main>
  <!-- VEHICLES PANEL -->
  <section id="panel-vehicles" class="panel active">
    <h2>Tow Vehicles</h2>
    <p class="muted">Add your trucks/SUVs with tow ratings so you can calculate safe loads and track maintenance.</p>
    <form id="vehicle-form">
      <div class="row">
        <div>
          <label>Nickname / Identifier
            <input type="text" id="veh-name" placeholder="e.g. 2015 F-150 Work Truck" required>
          </label>
        </div>
        <div>
          <label>Year
            <input type="number" id="veh-year" placeholder="2015">
          </label>
        </div>
        <div>
          <label>Make
            <input type="text" id="veh-make" placeholder="Ford">
          </label>
        </div>
        <div>
          <label>Model
            <input type="text" id="veh-model" placeholder="F-150">
          </label>
        </div>
      </div>
      <div class="row">
        <div>
          <label>Max Tow Rating (lbs)
            <input type="number" id="veh-tow" placeholder="e.g. 9000">
          </label>
        </div>
      </div>
    </form>
  </section>
</main>
```

```
<label>Curb Weight (lbs) <span class="muted">(optional)</span>
<input type="number" id="veh-curb" placeholder="e.g. 5200">
</label>
</div>
</div>
<label>Notes
<textarea id="veh-notes" placeholder="Mods, gearing, transmission, etc."></textarea>
</label>
<button type="submit" class="btn btn-primary">Save Vehicle</button>
<input type="hidden" id="veh-id">
</form>

<h3>Saved Vehicles</h3>
<div id="vehicle-list" class="list"></div>
</section>

<!-- TRAILERS PANEL -->
<section id="panel-trailers" class="panel">
<h2>Trailers</h2>
<p class="muted">Store details about each trailer for towing calculations and maintenance tracking.</p>
<form id="trailer-form">
<div class="row">
<div>
<label>Nickname / Identifier
<input type="text" id="tr-name" placeholder="e.g. 20ft Car Hauler" required>
</label>
</div>
<div>
<label>Type
<input type="text" id="tr-type" placeholder="Utility, Boat, Enclosed, etc.">
</label>
</div>
</div>
<div class="row">
<div>
<label>GVWR (lbs)
<input type="number" id="tr-gvwr" placeholder="e.g. 7000">
</label>
</div>
<div>
<label>Empty Weight (lbs)
<input type="number" id="tr-empty" placeholder="e.g. 2000">
</label>
```

```

</div>
<div>
  <label>Axles
    <input type="number" id="tr-axles" placeholder="1 or 2">
  </label>
</div>
</div>
<label>Notes
  <textarea id="tr-notes" placeholder="Tire size, brake type, special info."></textarea>
</label>
<button type="submit" class="btn btn-primary">Save Trailer</button>
<input type="hidden" id="tr-id">
</form>

<h3>Saved Trailers</h3>
<div id="trailer-list" class="list"></div>
</section>

<!-- TOW CALCULATOR PANEL -->
<section id="panel-towcalc" class="panel">
  <h2>Tow Rating Calculator</h2>
  <p class="muted">Choose a tow vehicle and trailer, add cargo/passenger weight, and get a quick safety check.</p>
  <form id="tow-form">
    <div class="row">
      <div>
        <label>Tow Vehicle
          <select id="tow-veh-select" required>
            <option value="">Select vehicle</option>
          </select>
        </label>
      </div>
      <div>
        <label>Trailer
          <select id="tow-tr-select" required>
            <option value="">Select trailer</option>
          </select>
        </label>
      </div>
    </div>
    <div class="row">
      <div>
        <label>Cargo on Trailer (lbs)
          <input type="number" id="tow-cargo" placeholder="e.g. 3500">
        </label>
      </div>
    </div>
  </form>
</section>

```

```

</label>
</div>
<div>
  <label>Passengers & Gear in Vehicle (lbs)
    <input type="number" id="tow-passengers" placeholder="e.g. 600">
  </label>
</div>
</div>
<button type="submit" class="btn btn-primary">Calculate</button>
</form>
<div id="tow-result"></div>
</section>


<section id="panel-diagnostics" class="panel">
  <h2>Trailer Diagnostics</h2>
  <p class="muted">Pick the symptom and get likely causes and checks. This is not a full replacement for a real inspection, but it helps you or your customer know where to look.</p>
  <form id="diag-form">
    <div class="row">
      <div>
        <label>Trailer
          <select id="diag-tr-select">
            <option value="">Select trailer (optional)</option>
          </select>
        </label>
      </div>
      <div>
        <label>Symptom
          <select id="diag-symptom" required>
            <option value="">Select symptom</option>
            <option value="lights-out">Lights not working / intermittent</option>
            <option value="one-light-out">One light out</option>
            <option value="brakes-weak">Brakes weak or not grabbing</option>
            <option value="wheel-hot">Wheel/hub running hot</option>
            <option value="swaying">Trailer swaying at speed</option>
            <option value="no-brakes-power">No power to electric brakes</option>
            <option value="bouncing">Trailer bouncing / harsh ride</option>
            <option value="frame-crack">Visible frame crack or rust</option>
          </select>
        </label>
      </div>
    </div>
    <button type="submit" class="btn btn-primary">Get Diagnosis</button>
  </form>
</section>

```

```
</form>
<div id="diag-result"></div>
</section>

<!-- MAINTENANCE PANEL -->
<section id="panel-maintenance" class="panel">
  <h2>Maintenance Log</h2>
  <p class="muted">
    Log work done on vehicles and trailers. Entries are stored in this browser.
    You can export them and send to your mechanic (or to AllDrive if you're on a paid plan).
  </p>
  <form id="maint-form">
    <div class="row">
      <div>
        <label>Type
          <select id="maint-type" required>
            <option value="vehicle">Vehicle</option>
            <option value="trailer">Trailer</option>
          </select>
        </label>
      </div>
      <div>
        <label>Item
          <select id="maint-item-select" required>
            <option value="">Select item</option>
          </select>
        </label>
      </div>
    </div>
    <div class="row">
      <div>
        <label>Date
          <input type="date" id="maint-date" required>
        </label>
      </div>
      <div>
        <label>Odometer / Approx. Miles <span class="muted">(optional)</span>
          <input type="number" id="maint-odo" placeholder="e.g. 124500">
        </label>
      </div>
    </div>
    <div class="row">
      <div>
        <label>Work Type
```

```

        <input type="text" id="maint-work" placeholder="e.g. Bearing service, brake job"
required>
        </label>
    </div>
</div>
<label>Notes
    <textarea id="maint-notes" placeholder="Parts used, brands, torque specs,
etc."></textarea>
</label>
<button type="submit" class="btn btn-primary">Add Entry</button>
<input type="hidden" id="maint-id">
</form>

<h3>Logged Maintenance</h3>
<div id="maint-list" class="list"></div>
</section>

<!-- DATA PANEL -->
<section id="panel-data" class="panel">
    <h2>Export / Import Data</h2>
    <p class="muted">
        Export your vehicles, trailers, and maintenance logs as JSON. This is useful for backups or
        sending to your mechanic
        if you're paying for a managed maintenance plan.
    </p>
    <div class="export-box">
        <button id="export-btn" class="btn btn-secondary">Export All Data</button>
        <textarea id="export-output" placeholder="Exported JSON will appear here...">
            readonly</textarea>
    </div>
    <h3>Import Data</h3>
    <p class="muted">
        Paste JSON exported from this app and click Import. This will <strong>replace</strong>
        your current data.
    </p>
    <textarea id="import-input" placeholder="Paste JSON here..."></textarea>
    <div class="inline-controls">
        <button id="import-btn" class="btn btn-primary">Import Data</button>
        <button id="clear-btn" class="btn btn-danger">Clear All Local Data</button>
    </div>
</section>
</main>

<script>
```

```

// Simple localStorage keys
const LS_VEHICLES = "td_app_vehicles";
const LS_TRAILERS = "td_app_trailers";
const LS_MAINT = "td_app_maint";

let vehicles = [];
let trailers = [];
let maintLogs = [];

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

function saveAll() {
    localStorage.setItem(LS_VEHICLES, JSON.stringify(vehicles));
    localStorage.setItem(LS_TRAILERS, JSON.stringify(trailers));
    localStorage.setItem(LS_MAINT, JSON.stringify(maintLogs));
    renderAll();
}

function loadAll() {
    vehicles = JSON.parse(localStorage.getItem(LS_VEHICLES)) || "[]";
    trailers = JSON.parse(localStorage.getItem(LS_TRAILERS)) || "[]";
    maintLogs = JSON.parse(localStorage.getItem(LS_MAINT)) || "[]";
    renderAll();
}

// NAV
document.querySelectorAll(".nav button").forEach(btn => {
    btn.addEventListener("click", () => {
        document.querySelectorAll(".nav button").forEach(b => b.classList.remove("active"));
        document.querySelectorAll(".panel").forEach(p => p.classList.remove("active"));
        btn.classList.add("active");
        const panelId = "panel-" + btn.dataset.panel;
        $(panelId).classList.add("active");
        // refresh selects when needed
        if (panelId === "panel-towcalc" || panelId === "panel-diagnostics" || panelId ===
        "panel-maintenance") {
            renderSelects();
        }
    });
});

// RENDER FUNCTIONS
function renderVehicles() {

```

```

const list = $("vehicle-list");
list.innerHTML = "";
if (vehicles.length === 0) {
    list.innerHTML = '<p class="muted">No vehicles saved yet.</p>';
    return;
}
vehicles.forEach(v => {
    const card = document.createElement("div");
    card.className = "card";
    card.innerHTML = `
        <div class="card-header">
            <span>${v.name}</span>
            <span class="badge">Vehicle</span>
        </div>
        <div class="small">
            ${v.year || ""} ${v.make || ""} ${v.model || ""}<br>
            Tow Rating: ${v.towRating ? v.towRating + " lbs" : "Not set"}
        </div>
        ${v.notes ? `<div class="small" style="margin-top:0.25rem;">${v.notes}</div>` : ""}
        <div class="inline-controls">
            <button class="btn btn-secondary btn-sm-edit">Edit</button>
            <button class="btn btn-danger btn-sm-delete">Delete</button>
        </div>
    `;
    card.querySelector(".btn-sm-edit").addEventListener("click", () => {
        $("veh-id").value = v.id;
        $("veh-name").value = v.name;
        $("veh-year").value = v.year || "";
        $("veh-make").value = v.make || "";
        $("veh-model").value = v.model || "";
        $("veh-tow").value = v.towRating || "";
        $("veh-curb").value = v.curbWeight || "";
        $("veh-notes").value = v.notes || "";
        window.scrollTo({ top: 0, behavior: "smooth" });
    });
    card.querySelector(".btn-sm-delete").addEventListener("click", () => {
        if (confirm("Delete this vehicle?")) {
            vehicles = vehicles.filter(x => x.id !== v.id);
            saveAll();
        }
    });
    list.appendChild(card);
});
}

```

```

function renderTrailers() {
  const list = $("trailer-list");
  list.innerHTML = "";
  if (trailers.length === 0) {
    list.innerHTML = '<p class="muted">No trailers saved yet.</p>';
    return;
  }
  trailers.forEach(t => {
    const card = document.createElement("div");
    card.className = "card";
    card.innerHTML = `
      <div class="card-header">
        <span>${t.name}</span>
        <span class="badge">Trailer</span>
      </div>
      <div class="small">
        ${t.type || ""}<br>
        GVWR: ${t.gvwr ? t.gvwr + " lbs" : "Not set"} | Empty: ${t.emptyWeight ? t.emptyWeight +
        " lbs" : "Not set"}
      </div>
      ${t.notes ? `<div class="small" style="margin-top:0.25rem;">${t.notes}</div>` : ""}
      <div class="inline-controls">
        <button class="btn btn-secondary btn-sm-edit">Edit</button>
        <button class="btn btn-danger btn-sm-delete">Delete</button>
      </div>
    `;
    card.querySelector(".btn-sm-edit").addEventListener("click", () => {
      $("tr-id").value = t.id;
      $("tr-name").value = t.name;
      $("tr-type").value = t.type || "";
      $("tr-gvwr").value = t.gvwr || "";
      $("tr-empty").value = t.emptyWeight || "";
      $("tr-axles").value = t.axles || "";
      $("tr-notes").value = t.notes || "";
      window.scrollTo({ top: 0, behavior: "smooth" });
    });
    card.querySelector(".btn-sm-delete").addEventListener("click", () => {
      if (confirm("Delete this trailer?")) {
        trailers = trailers.filter(x => x.id !== t.id);
        saveAll();
      }
    });
    list.appendChild(card);
  })
}

```

```

    });
}

function renderMaint() {
  const list = $("maint-list");
  list.innerHTML = "";
  if (maintLogs.length === 0) {
    list.innerHTML = '<p class="muted">No maintenance entries yet.</p>';
    return;
  }
  // sort by date desc
  const sorted = [...maintLogs].sort((a, b) => (b.date || "").localeCompare(a.date || ""));
  sorted.forEach(m => {
    const itemName = m.type === "vehicle"
      ? (vehicles.find(v => v.id === m.itemId)?.name || "Vehicle")
      : (trailers.find(t => t.id === m.itemId)?.name || "Trailer");
    const card = document.createElement("div");
    card.className = "card";
    card.innerHTML =
      `<div class="card-header">
        <span>${itemName}</span>
        <span class="badge">${m.type === "vehicle" ? "Vehicle" : "Trailer"}</span>
      </div>
      <div class="small">
        <strong>${m.workType}</strong> on ${m.date || "Unknown date"}
        ${m.odo ? ` | Odo/Miles: ${m.odo}` : ""}
      </div>
      ${m.notes ? `<div class="small" style="margin-top:0.25rem;">${m.notes}</div>` : ""}
      <div class="inline-controls">
        <button class="btn btn-secondary btn-sm-edit">Edit</button>
        <button class="btn btn-danger btn-sm-delete">Delete</button>
      </div>
    `;
    card.querySelector(".btn-sm-edit").addEventListener("click", () => {
      $("maint-id").value = m.id;
      $("maint-type").value = m.type;
      updateMaintItemSelect();
      $("maint-item-select").value = m.itemId;
      $("maint-date").value = m.date || "";
      $("maint-odo").value = m.odo || "";
      $("maint-work").value = m.workType || "";
      $("maint-notes").value = m.notes || "";
      window.scrollTo({ top: 0, behavior: "smooth" });
    });
  });
}

```

```

    card.querySelector(".btn-sm-delete").addEventListener("click", () => {
      if (confirm("Delete this entry?")) {
        maintLogs = maintLogs.filter(x => x.id !== m.id);
        saveAll();
      }
    });
    list.appendChild(card);
  });

function renderSelects() {
  // Tow calculator
  const vehSel = $("tow-veh-select");
  const trSel = $("tow-tr-select");
  vehSel.innerHTML = '<option value="">Select vehicle</option>';
  trSel.innerHTML = '<option value="">Select trailer</option>';
  vehicles.forEach(v => {
    const opt = document.createElement("option");
    opt.value = v.id;
    opt.textContent = v.name;
    vehSel.appendChild(opt);
  });
  trailers.forEach(t => {
    const opt = document.createElement("option");
    opt.value = t.id;
    opt.textContent = t.name;
    trSel.appendChild(opt);
  });

  // Diagnostics trailer select
  const diagTr = $("diag-tr-select");
  diagTr.innerHTML = '<option value="">Select trailer (optional)</option>';
  trailers.forEach(t => {
    const opt = document.createElement("option");
    opt.value = t.id;
    opt.textContent = t.name;
    diagTr.appendChild(opt);
  });

  // Maintenance item select
  updateMaintItemSelect();
}

function updateMaintItemSelect() {

```

```

const type = $("maint-type").value;
const sel = $("maint-item-select");
sel.innerHTML = '<option value="">Select item</option>';
const src = type === "vehicle" ? vehicles : trailers;
src.forEach(item => {
  const opt = document.createElement("option");
  opt.value = item.id;
  opt.textContent = item.name;
  sel.appendChild(opt);
});
}

function renderAll() {
  renderVehicles();
  renderTrailers();
  renderMaint();
  renderSelects();
}

// FORMS

// Vehicles
$("vehicle-form").addEventListener("submit", (e) => {
  e.preventDefault();
  const id = $("veh-id").value || crypto.randomUUID();
  const existingIdx = vehicles.findIndex(v => v.id === id);
  const v = {
    id,
    name: $("veh-name").value.trim(),
    year: $("veh-year").value || null,
    make: $("veh-make").value.trim() || null,
    model: $("veh-model").value.trim() || null,
    towRating: $("veh-tow").value ? Number($("veh-tow").value) : null,
    curbWeight: $("veh-curb").value ? Number($("veh-curb").value) : null,
    notes: $("veh-notes").value.trim() || null
  };
  if (!v.name) {
    alert("Vehicle name is required.");
    return;
  }
  if (existingIdx >= 0) {
    vehicles[existingIdx] = v;
  } else {
    vehicles.push(v);
  }
}

```

```

        }
        $("vehicle-form").reset();
        $("veh-id").value = "";
        saveAll();
    });

// Trailers
$("trailer-form").addEventListener("submit", (e) => {
    e.preventDefault();
    const id = $("tr-id").value || crypto.randomUUID();
    const existingIdx = trailers.findIndex(t => t.id === id);
    const t = {
        id,
        name: $("tr-name").value.trim(),
        type: $("tr-type").value.trim() || null,
        gvwr: $("tr-gvwr").value ? Number($("tr-gvwr").value) : null,
        emptyWeight: $("tr-empty").value ? Number($("tr-empty").value) : null,
        axles: $("tr-axles").value ? Number($("tr-axles").value) : null,
        notes: $("tr-notes").value.trim() || null
    };
    if (!t.name) {
        alert("Trailer name is required.");
        return;
    }
    if (existingIdx >= 0) {
        trailers[existingIdx] = t;
    } else {
        trailers.push(t);
    }
    $("trailer-form").reset();
    $("tr-id").value = "";
    saveAll();
});

// Tow Calculator
$("tow-form").addEventListener("submit", (e) => {
    e.preventDefault();
    const vehId = $("tow-veh-select").value;
    const trId = $("tow-tr-select").value;
    if (!vehId || !trId) {
        alert("Please select both a vehicle and a trailer.");
        return;
    }
    const cargo = Number($("tow-cargo").value || 0);

```

```

const passengers = Number($("#tow-passengers").value || 0);
const veh = vehicles.find(v => v.id === vehId);
const tr = trailers.find(t => t.id === trId);

const towRating = veh?.towRating || 0;
const trailerBase = tr?.emptyWeight || 0;
const totalTrailerWeight = trailerBase + cargo;
const combinedApprox = totalTrailerWeight + passengers + (veh?.curbWeight || 0);

let msg = "";
let cls = "";
if (!towRating) {
    msg = "This vehicle has no tow rating set. Please look up the factory tow rating and add it for better accuracy.";
    cls = "warn";
} else {
    const ratio = totalTrailerWeight / towRating;
    if (ratio <= 0.85) {
        msg = `Approx. trailer weight is about ${Math.round(totalTrailerWeight)} lbs vs tow rating of ${towRating} lbs. This is generally within a safe range if the vehicle, hitch, and trailer are set up correctly. Always confirm with factory specs.`;
        cls = "safe";
    } else if (ratio <= 1.0) {
        msg = `Approx. trailer weight is about ${Math.round(totalTrailerWeight)} lbs vs tow rating of ${towRating} lbs. This is getting close to the limit. Double-check factory specs, tongue weight, and setup before towing.`;
        cls = "warn";
    } else {
        msg = `Approx. trailer weight is about ${Math.round(totalTrailerWeight)} lbs vs tow rating of ${towRating} lbs. This appears to exceed the rated capacity. You should not tow this as-is. Consider a lighter load or a more capable tow vehicle.`;
        cls = "danger";
    }
    msg += `\n\nEstimated combined weight (vehicle + trailer + passengers/gear): ~${Math.round(combinedApprox)} lbs. This is an estimate only. Always rely on actual manufacturer ratings, CAT scale weights, and local laws.`;
}

$("#tow-result").innerHTML =
<div class="status ${cls}">
<strong>Result:</strong><br>
<span>${msg.replace(/\n/g, "<br>")}</span>
</div>
';

```

```

});
```

```

// Diagnostics
$("diag-form").addEventListener("submit", (e) => {
  e.preventDefault();
  const symptom = $("diag-symptom").value;
  if (!symptom) {
    alert("Select a symptom.");
    return;
  }
  let title = "";
  let body = "";
  switch (symptom) {
    case "lights-out":
      title = "Most or All Trailer Lights Not Working";
      body = `
        • Check the ground: Make sure the trailer plug ground wire is clean, tight, and connected to bare metal.<br>
        • Inspect the 7-pin/4-pin connector for corrosion, bent pins, or damage.<br>
        • Verify the tow vehicle's fuse for trailer lights is good.<br>
        • Look for broken or chafed wires along the tongue and frame.`;
      break;
    case "one-light-out":
      title = "Single Light Out";
      body = `
        • Check the bulb or LED unit itself.<br>
        • Inspect the pigtail connector at that light for corrosion.<br>
        • Confirm that light's ground is solid to the trailer frame.<br>
        • Use a test light or meter at the light to confirm power and ground.`;
      break;
    case "brakes-weak":
      title = "Trailer Brakes Weak or Not Grabbing";
      body = `
        • Check the brake controller gain setting inside the tow vehicle.<br>
        • Verify the trailer brake wiring at the axle and junction box.<br>
        • Inspect magnets/shoes on drum brakes or pads on disc setups.<br>
        • Look for loose or out-of-adjustment drum brakes.`;
      break;
    case "wheel-hot":
      title = "Wheel / Hub Running Hot";
      body = `
        • Likely causes: over-tight bearings, failing bearings, dragging brakes.<br>
        • Jack up that wheel and spin it: listen for grinding or roughness.<br>
        • Check for grease leaking out, or metal shavings around the hub.<br>
      
```

```

    • Pull the hub, inspect bearings and races, repack/replace as needed. `;
break;
case "swaying":
title = "Trailer Swaying at Speed";
body = `
    • Check tongue weight: too light is a common cause (aim ~10–15% of trailer weight).<br>
    • Inspect tire pressures and tire condition on both trailer and tow vehicle.<br>
    • Verify correct weight distribution and load placement over axles.<br>
    • Confirm the trailer is level when hitched. Consider a weight-distributing hitch.`;
break;
case "no-brakes-power":
title = "No Power to Electric Brakes";
body = `
    • Check the trailer brake fuse and wiring from brake controller to rear plug.<br>
    • Verify blue brake wire has power when controller is activated (use test light/meter).<br>
    • Inspect the junction box on the trailer tongue for loose or broken connections.<br>
    • Make sure the breakaway switch and battery (if equipped) are working and wired
correctly.`;
break;
case "bouncing":
title = "Trailer Bouncing / Harsh Ride";
body = `
    • Confirm tires are not overinflated beyond their rated pressure for the load.<br>
    • Inspect leaf springs, equalizers, and shackles for wear or binding.<br>
    • Check that load is properly distributed; an empty or very light trailer can bounce
more.<br>
    • Verify shocks, if equipped, are in good condition.`;
break;
case "frame-crack":
title = "Frame Cracks or Rust";
body = `
    • Any crack, severe rust, or deformation in the frame needs attention before towing.<br>
    • Clean the area to inspect how deep the damage goes.<br>
    • Avoid towing until a qualified welder repairs and reinforces the affected area.<br>
    • After repair, inspect regularly for new cracks or movement.`;
break;
default:
title = "No Data";
body = "Select a known symptom.";
}
$("diag-result").innerHTML =
<div class="status warn">
<strong>${title}</strong><br>
<span class="small">${body}</span><br><br>
```

```

<span class="muted">This is a guideline only. For serious issues, get a hands-on
inspection by a qualified technician.</span>
</div>
`;
});

// Maintenance
$("maint-type").addEventListener("change", updateMaintItemSelect);

$("maint-form").addEventListener("submit", (e) => {
  e.preventDefault();
  const type = $("maint-type").value;
  const itemId = $("maint-item-select").value;
  if (!itemId) {
    alert("Select a vehicle or trailer for this entry.");
    return;
  }
  const id = $("maint-id").value || crypto.randomUUID();
  const existingIdx = maintLogs.findIndex(m => m.id === id);
  const m = {
    id,
    type,
    itemId,
    date: $("maint-date").value,
    odo: $("maint-odo").value ? Number($("maint-odo").value) : null,
    workType: $("maint-work").value.trim(),
    notes: $("maint-notes").value.trim() || null
  };
  if (!m.workType) {
    alert("Work type is required.");
    return;
  }
  if (existingIdx >= 0) {
    maintLogs[existingIdx] = m;
  } else {
    maintLogs.push(m);
  }
  $("maint-form").reset();
  $("maint-id").value = "";
  saveAll();
});

// Export / Import
$("export-btn").addEventListener("click", () => {

```

```

const payload = {
  exportedAt: new Date().toISOString(),
  vehicles,
  trailers,
  maintenance: maintLogs
};
$("export-output").value = JSON.stringify(payload, null, 2);
$("export-output").focus();
$("export-output").select();
});

$("import-btn").addEventListener("click", () => {
  if (!confirm("Importing will replace your current data. Continue?")) return;
  try {
    const data = JSON.parse($(".import-input").value || "{}");
    vehicles = Array.isArray(data.vehicles) ? data.vehicles : [];
    trailers = Array.isArray(data.trailers) ? data.trailers : [];
    maintLogs = Array.isArray(data.maintenance) ? data.maintenance : [];
    saveAll();
    alert("Import complete.");
  } catch (err) {
    alert("Import failed. Check that the JSON is valid.");
  }
});

$("clear-btn").addEventListener("click", () => {
  if (!confirm("This will clear all vehicles, trailers, and maintenance logs stored in this browser. Continue?")) return;
  vehicles = [];
  trailers = [];
  maintLogs = [];
  saveAll();
});

// Init
loadAll();
</script>
</body>
</html>

```