Js:

```
// (optional) remove after confirming once

console.log('JS alive');


// === Apps Script endpoint (must end with /exec) ===

const endpoint =
'https://script.google.com/macros/s/AKfycbwJAUHbDMbGxEMwOSPRn5ZsauEE9a5z7NWJOcPJZf2SzKVzNe4pnVnBvdCShvJXxRVY/exec';

console.log('Endpoint:', endpoint);


// === Reveal service boxes with subtle slide-in effect (kept from your code) ===

window.addEventListener('scroll', () => {

  const boxes = document.querySelectorAll('.service-box');

  boxes.forEach((box, i) => {

    const boxTop = box.getBoundingClientRect().top;

    const triggerPoint = window.innerHeight - 50;

    if (boxTop < triggerPoint) {

      setTimeout(() => {

        box.style.opacity = '1';

        box.style.transform = 'translateY(0)';

      }, i * 100);

    }

  });

});


// === Hover details helpers (kept from your code) ===

function showDetails(serviceBox) {

  const details = serviceBox.querySelector('.service-details');

  if (details) details.style.display = 'block';

}

function hideDetails(serviceBox) {

  const details = serviceBox.querySelector('.service-details');

  if (details) details.style.display = 'none';

}
```

```
// ===== One-form wiring to match your contact.html =====
(function () {
  if (!window.fetch) return;

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

  // Elements from your contact.html
  const form        = $('contactForm');
  const btnSubmit    = $('submitBtn');
  const statusMsg    = $('form-msg');         // <p id="form-msg">
  const btnInquiry   = $('btn-inquiry');
  const btnEstimate  = $('btn-estimate');
  const modeInput    = $('mode');           // <input id="mode" name="type" value="inquiry">

  const estimateFields = $('estimateFields');  // container to show/hide
  const serviceField   = $('serviceField');

  const messageLabel = $('messageLabel');
  const message      = $('message');
  const formNote     = $('formNote');
  const serviceSelect= $('service');

  // Normalize field names so the Sheet columns are consistent
  function normalizePayload(raw) {
    const p = { ...raw };

    // Ensure we send a type that is either 'inquiry' or 'estimate'
    p.type = (p.type || (modeInput ? modeInput.value : 'inquiry')).toLowerCase();

    // preferred contact: accept either `preferred` or `contactMethod`
    if (!p.preferred && p.contactMethod) p.preferred = p.contactMethod;

    // Map message/details depending on mode
    if (p.type === 'estimate') {
```

```
    if (!p.details && p.message) p.details = p.message;
  } else {
    if (!p.message && p.details) p.message = p.details;
  }

  // Add source path for context
  p.source = window.location.pathname;
  return p;
}

async function sendToSheet(payload) {
  try {
    if (btnSubmit) btnSubmit.disabled = true;
    if (statusMsg) statusMsg.textContent = '';

    await fetch(endpoint, {
      method: 'POST',
      mode: 'no-cors', // opaque; treat as success if no exception
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(payload),
    });

    if (statusMsg) {
      statusMsg.textContent = 'Thanks! We received your request.';
      statusMsg.style.color = 'green';
    }

    if (form) form.reset();
    // preserve current mode value after reset
    if (modeInput && payload.type) modeInput.value = payload.type;

  } catch (err) {
    console.error(err);
    if (statusMsg) {
      statusMsg.textContent = 'Sorry, something went wrong. Please try again.';
      statusMsg.style.color = 'red';
    }
  } finally {
    if (btnSubmit) btnSubmit.disabled = false;
  }
}

// Mode UI toggles
function setMode(mode) {
  if (modeInput) modeInput.value = mode;

  const isEstimate = mode === 'estimate';
  if (btnEstimate) btnEstimate.classList.toggle('active', isEstimate);
  if (btnInquiry)  btnInquiry.classList.toggle('active', !isEstimate);

  if (estimateFields) estimateFields.classList.toggle('hidden', !isEstimate);
  if (serviceField)   serviceField.classList.toggle('hidden', !isEstimate);

  if (messageLabel) messageLabel.textContent = isEstimate ? 'Project Details' : 'Message';
  if (message) {
    message.placeholder = isEstimate
      ? 'Tell us about your project scope and any details we should know.'
      : 'How can we help?';
  }
  if (btnSubmit) btnSubmit.textContent = isEstimate ? 'Request Estimate' : 'Send Message';
  if (formNote) {
    formNote.textContent = isEstimate
      ? 'We’ll follow up to confirm any details for your estimate.'
      : 'We typically reply within 1 business day.';
  }
}

// Initialize mode from URL (?mode=estimate&service=…)
```

```
function prefillFromURL() {
  let startMode = 'inquiry';
  const params = new URLSearchParams(window.location.search);

  const urlMode = (params.get('mode') || '').toLowerCase();
  if (urlMode === 'estimate') startMode = 'estimate';

  const svc = params.get('service');
  if (svc && serviceSelect) {
    const options = Array.from(serviceSelect.options).map(o => o.value.toLowerCase());
    if (!options.includes(svc.toLowerCase())) {
      const opt = document.createElement('option');
      opt.value = svc; opt.textContent = svc;
      serviceSelect.appendChild(opt);
    }
    serviceSelect.value = svc;
    startMode = 'estimate';
  }

  setMode(startMode);
}

// Wire buttons
if (btnInquiry)  btnInquiry.addEventListener('click', () => setMode('inquiry'));
if (btnEstimate) btnEstimate.addEventListener('click', () => setMode('estimate'));

// Submit handler for the single form
if (form) {
  form.addEventListener('submit', async (e) => {
    e.preventDefault();
    const raw = Object.fromEntries(new FormData(form).entries());

    // Honeypot: bots fill this; humans shouldn't
    if ((raw._hp || '').trim() !== '') return;
```

```
   const payload = normalizePayload(raw);
   console.log('Submitting:', payload);
   await sendToSheet(payload);
  });
 }

 // Run after DOM is ready
 if (document.readyState === 'loading') {
  document.addEventListener('DOMContentLoaded', prefillFromURL);
 } else {
  prefillFromURL();
 }
})();
```

Apps Script:

```
// === CONFIG (fill after we confirm save) ===
const SPREADSHEET_ID = 'REPLACE_ME';
const SHEET_NAME = 'Sheet1';

function doGet(e) {
 if (e && e.parameter && e.parameter.test === '1') {
  const sh = SpreadsheetApp.openById(SPREADSHEET_ID).getSheetByName(SHEET_NAME);
  sh.appendRow([new Date(), 'GET test', 'ok']);
  return ContentService.createTextOutput('OK+WROTE')
   .setMimeType(ContentService.MimeType.TEXT);
 }
 return ContentService.createTextOutput('OK')
  .setMimeType(ContentService.MimeType.TEXT);
}

function parseRequest(e) {
 let data = {};
```

```
  if (e && e.parameter) data = { ...e.parameter };
  if (e && e.postData && typeof e.postData.contents === 'string') {
    const raw = e.postData.contents;
    const ctype = (e.postData.type || '').toLowerCase();
    if (ctype.includes('application/json')) {
      try { data = { ...data, ...JSON.parse(raw) }; } catch (_){}
    }
    if (!Object.keys(data).length || ctype.includes('application/x-www-form-urlencoded')) {
      try { data = { ...data, ...Utilities.parseQueryString(raw) }; } catch (_){}
    }
  }
  return data;
}

function doPost(e) {
  const d = parseRequest(e);
  if ((d._hp || '').toString().trim() !== '') {
    return ContentService.createTextOutput(JSON.stringify({ ok: true, skipped: true }))
      .setMimeType(ContentService.MimeType.JSON);
  }
  const sh = SpreadsheetApp.openById(SPREADSHEET_ID).getSheetByName(SHEET_NAME);
  sh.appendRow([
    new Date(),
    (d.type||''), (d.name||''), (d.email||''), (d.phone||''),
    (d.address||''), (d.zip||''), (d.preferred||d.contactMethod||''),
    (d.service||''), (d.timeline||''), (d.budget||''),
    (d.message||d.details||''), (d.source||'')
  ]);
  return ContentService.createTextOutput(JSON.stringify({ ok: true }))
    .setMimeType(ContentService.MimeType.JSON);
```

Contact.cs

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
  <title>Contact / Estimates | Hometown Projects LLC</title>
  <link rel="stylesheet" href="style.css" />
  <style>
    .contact-section { padding: 2rem 1rem 3rem; max-width: 1000px; margin: 0 auto; }
    .contact-section h1 { color: #8B2C2C; text-align: center; margin-bottom: 1rem; }

    .mode-switch { display: flex; gap: .75rem; justify-content: center; margin: 1rem auto 2rem; }
    .mode-btn {
      background: #eee; border: 2px solid #ccc; color: #333;
      padding: .6rem 1rem; border-radius: 8px; cursor: pointer; font-weight: 700;
      transition: transform .15s ease, box-shadow .15s ease, border-color .15s ease;
    }
    .mode-btn.active { background: #8B2C2C; color: #fff; border-color: #8B2C2C; box-shadow: 0 6px 16px rgba(0,0,0,.12); }

    .contact-container { display: grid; grid-template-columns: 1.2fr .8fr; gap: 2rem; }
    @media (max-width: 900px) { .contact-container { grid-template-columns: 1fr; } }

    .card {
      background: #fff; border: 2px solid #8B2C2C; border-radius: 10px; padding: 1.25rem;
      box-shadow: 0 6px 16px rgba(0,0,0,.08);
    }

    form label { display: block; font-weight: 700; margin-top: .9rem; }
    form input, form textarea, form select {
      width: 100%; padding: .6rem .7rem; border: 1px solid #ccc; border-radius: 6px; margin-top: .35rem; font-size: 1rem;
    }
    .row { display: grid; grid-template-columns: 1fr 1fr; gap: 1rem; }
    @media (max-width: 600px) { .row { grid-template-columns: 1fr; } }

    .submit-btn {
      margin-top: 1.1rem; background: #8B2C2C; color: #fff; border: none;
      padding: .7rem 1.25rem; border-radius: 8px; font-weight: 700; cursor: pointer;
      transition: background .2s ease, transform .2s ease, box-shadow .2s ease;
      box-shadow: 0 4px 12px rgba(0,0,0,.15);
    }
    .submit-btn:hover { background: #9f4747; transform: translateY(-2px); }

    .contact-info h3 { color: #8B2C2C; margin-top: .25rem; }
    .muted { color: #666; font-size: .95rem; }

    .hidden { display: none; }
  </style>
</head>
<body>
  <header>
    <a href="index.html" class="logo-container">
      <img src="logo.png" alt="Hometown Projects LLC logo" class="logo" />
    </a>
    <nav>
      <ul>
        <li><a href="index.html">Home</a></li>
        <li><a href="contact.html" aria-current="page">Contact Us</a></li>
        <li><a href="about.html">About Us</a></li>
        <li><a href="services.html">Services</a></li>
      </ul>
    </nav>
  </header>

  <main class="contact-section">
    <h1>Contact Us</h1>

    <div class="mode-switch">
      <button class="mode-btn active" id="btn-inquiry" type="button">General Inquiry</button>
      <button class="mode-btn" id="btn-estimate" type="button">Request an Estimate</button>
    </div>
```

```html
<div class="contact-container">
 <!-- FORM CARD -->
 <form id="contactForm" class="card" novalidate>
  <!-- single hidden mode/type field used by JS -->
  <input type="hidden" id="mode" name="type" value="inquiry">

  <!-- Common fields -->
  <div class="row">
   <div>
    <label for="name">Name</label>
    <input type="text" id="name" name="name" autocomplete="name" required>
   </div>
   <div>
    <label for="email">Email</label>
    <input type="email" id="email" name="email" autocomplete="email" required>
   </div>
  </div>

  <div class="row">
   <div>
    <label for="phone">Phone <span class="muted">(optional)</span></label>
    <input type="tel" id="phone" name="phone" autocomplete="tel">
   </div>
   <div id="serviceField" class="hidden">
    <label for="service">Service</label>
    <select id="service" name="service">
     <option value="">Select a service...</option>
     <option>Lawn Care</option>
     <option>Landscaping</option>
     <option>Tree Trimming</option>
     <option>Snow Removal</option>
     <option>Gutter Cleaning</option>
     <option>Mulching</option>
    </select>
  </div>
</div>

<!-- Estimate-only fields -->
<div id="estimateFields" class="hidden">
  <div class="row">
    <div>
      <label for="address">Address</label>
      <input type="text" id="address" name="address" autocomplete="street-address">
    </div>
    <div>
      <label for="zip">ZIP</label>
      <input type="text" id="zip" name="zip" inputmode="numeric" pattern="[0-9]*">
    </div>
  </div>

  <div class="row">
    <div>
      <label for="contactMethod">Preferred Contact</label>
      <select id="contactMethod" name="contactMethod">
        <option>Email</option>
        <option>Phone</option>
        <option>Text</option>
      </select>
    </div>
  </div>
</div>

<!-- Details -->
<label for="message" id="messageLabel">Message</label>
<textarea id="message" name="message" rows="6" placeholder="How can we help?" required></textarea>

<!-- Honeypot (hidden) -->
```

```
        <input type="text" name="_hp" autocomplete="off" tabindex="-1" style="position:absolute;left:-9999px;height:0;width:0;opacity:0;">

        <button class="submit-btn" type="submit" id="submitBtn">Send Message</button>
        <p class="muted" id="formNote">We typically reply within 1 business day.</p>
        <p class="muted" id="form-msg" aria-live="polite"></p>
      </form>

      <!-- INFO CARD -->
      <div class="contact-info card">
        <h3>Get in Touch</h3>
        <p>Email: info@hometownprojects.com</p>
        <p>Phone: (317) 515-2905</p>
        <p>Location: Indianapolis, IN</p>
        <hr style="border:none;height:2px;background:#8B2C2C;border-radius:999px;margin:1rem 0;">
        <p class="muted">
          For quick estimates, pick <strong>Request an Estimate</strong> and tell us your service, address, and how you’d like to be contacted.
          If you clicked from a service card, we’ll pre-fill the service for you.
        </p>
      </div>
    </div>
  </main>

  <footer>
    <p>© 2025 Hometown Projects LLC. All rights reserved.</p>
  </footer>

  <!-- Load page JS after HTML -->
  <script src="script.js?v=13"></script>
</body>
</html>
```