// === NEW Chrome Skiptrace Extension MVP ===

{

  "manifest\_version": 3,

  "name": "ANV Capital Skiptracer",

  "version": "1.0",

  "description": "Automates skiptracing of leads from your Google Sheet and updates contact info.",

  "permissions": [

    "tabs",

    "scripting",

    "storage",

    "identity"

  ],

  "host\_permissions": [

    "https://truepeoplesearch.com/\*",

    "https://fastpeoplesearch.com/\*",

    "https://whitepages.com/\*",

    "https://\*.beenverified.com/\*",

    "https://script.google.com/macros/\*"

  ],

  "background": {

    "service\_worker": "background.js"

  },

  "action": {

    "default\_popup": "popup.html",

    "default\_icon": {

      "16": "icon.png",

      "48": "icon.png",

      "128": "icon.png"

    }

  },

  "icons": {

    "16": "icon.png",

    "48": "icon.png",

    "128": "icon.png"

  }

}

[📝 Notes

* manifest\_version: 3 is required by all current Chrome extensions.
* host\_permissions: Be sure to include all the domains you intend to scrape. You can add or remove from this list.
* identity: Included in case we extend OAuth or authenticated requests later.
* icon.png: Just drop a basic icon in the same folder with this filename.]

// === popup.html ===

<!DOCTYPE html>

<html>

  <head><title>ANV Skiptracer</title></head>

  <body>

    <h3>Skiptrace Leads</h3>

    <button id="start">Start Skiptracing</button>

    <script src="popup.js"></script>

  </body>

</html>

// === popup.js ===

document.getElementById("start").addEventListener("click", async () => {

  chrome.runtime.sendMessage({ action: "startSkiptrace" });

});

// === background.js ===

chrome.runtime.onMessage.addListener((msg, sender, sendResponse) => {

  if (msg.action === "startSkiptrace") {

    // Fetch rows from Sheet (placeholder function below)

    fetchRowsNeedingSkiptrace().then(rows => {

      rows.forEach((row, index) => {

        setTimeout(() => {

          const searchUrl = `https://www.truepeoplesearch.com/results?name=${encodeURIComponent(row.owner)}&citystatezip=${encodeURIComponent(row.zip)}`;

          chrome.tabs.create({ url: searchUrl, active: false });

        }, index \* 15000); // 15s delay per lead

      });

    });

  }

});

async function fetchRowsNeedingSkiptrace() {

  // Replace with actual Apps Script endpoint later

  const res = await fetch("https://your-apps-script-webapp-url/fetchUnskiptraced");

  const json = await res.json();

  return json.leads; // Each: { rowId, owner, zip }

}

// === content.js ===

setTimeout(() => {

  try {

    const result = document.querySelector(".result-row");

    const name = result.querySelector(".h4 span")?.textContent;

    const phone = result.querySelector(".result-phone")?.textContent;

    const email = ""; // TruePeopleSearch doesn't list emails

    // Parse Row ID from tab title (or use storage method)

    const rowId = "PLACEHOLDER\_ROW\_ID";

    fetch("https://your-apps-script-webapp-url", {

      method: "POST",

      contentType: "application/json",

      body: JSON.stringify({

        rowId,

        name,

        phone,

        email,

        source: "TruePeopleSearch"

      })

    });

  } catch (err) {

    console.warn("Scrape failed", err);

  }

}, 8000);

// === sheets.js (optional OAuth helper or data relay) ===

// Leave blank for now; used for more advanced API usage or auth later.