

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1" />
  <title>Voice-Activated Expense Entry</title>
  <style>
    body {
      font-family: Roboto, sans-serif;
      padding: 20px;
      max-width: 480px;
      margin: auto;
    }
    h2 {
      margin-bottom: 0.5rem;
    }
    p.instructions {
      font-size: 0.9rem;
      margin-bottom: 1rem;
      color: #555;
    }
    button {
      display: block;
      width: 100%;
      background: #1976d2;
      color: white;
      font-size: 1rem;
      padding: 12px;
      border: none;
      border-radius: 8px;
      cursor: pointer;
      margin-top: 12px;
    }
    button:disabled {
      background: #aaa;
      cursor: not-allowed;
    }
    input,
    select {
      width: 100%;
      padding: 10px;
      font-size: 1rem;
      margin-top: 8px;
      border: 1px solid #ccc;
      border-radius: 6px;
      box-sizing: border-box;
    }
    #status {
```

```

    margin-top: 12px;
    font-size: 0.9rem;
    color: #333;
    min-height: 1.2rem;
  }
</style>
</head>
<body>
  <h2>Voice-Activated Expense</h2>
  <p class="instructions">
    Tap "Start Recording" and say:<br />
    <strong>
      "Date: June 6, Category: Food, Amount: 150, Note: Chapati"
    </strong>
  </p>

  <button id="startBtn">🎤 Start Recording</button>
  <div id="status"></div>

  <label for="inputDate">Date</label>
  <input type="date" id="inputDate" />

  <label for="inputCategory">Category</label>
  <select id="inputCategory">
    <option value="Food">Food</option>
    <option value="Rent">Rent</option>
    <option value="Transport">Transport</option>
    <option value="Utilities">Utilities</option>
    <option value="Misc">Misc</option>
  </select>

  <label for="inputAmount">Amount</label>
  <input type="number" id="inputAmount" placeholder="e.g. 150" />

  <label for="inputNote">Note (optional)</label>
  <input type="text" id="inputNote" placeholder="e.g. Chapati" />

  <button id="submitBtn" disabled>💾 Save to Sheet</button>

  <script>
    // —— Replace this with your Apps Script Web App URL ——
    const WEB_APP_URL =
"https://script.google.com/macros/s/YOUR_DEPLOY_ID_HERE/exec";
    //

```

---

let recognition;

```

const statusDiv = document.getElementById("status");
const startBtn = document.getElementById("startBtn");
const submitBtn = document.getElementById("submitBtn");

if (!("webkitSpeechRecognition" in window) && !("SpeechRecognition" in window)) {
  statusDiv.innerText =
    "⚠ Web Speech API not supported. Use Chrome on Android or Safari on iOS in Desktop Site mode.";
  startBtn.disabled = true;
} else {
  const SpeechRecognition = window.SpeechRecognition ||
window.webkitSpeechRecognition;
  recognition = new SpeechRecognition();
  recognition.lang = "en-KE"; // adjust locale if needed
  recognition.interimResults = false;
  recognition.maxAlternatives = 1;

  startBtn.onclick = () => {
    statusDiv.innerText = "Listening...";
    recognition.start();
  };

  recognition.onresult = (event) => {
    const transcript = event.results[0][0].transcript;
    statusDiv.innerText = 'Heard: ' + transcript + ' ';
    parseTranscript(transcript);
    submitBtn.disabled = false;
  };

  recognition.onerror = (e) => {
    statusDiv.innerText = "Error: " + e.error;
  };
}

function parseTranscript(text) {
  // 1. Date
  const dateMatch = text.match(/date[:\s]*([a-zA-Z0-9,\s]+)/i);
  if (dateMatch) {
    const spoken = dateMatch[1].trim();
    const dt = new Date(spoken);
    if (!isNaN(dt)) {
      const y = dt.getFullYear();
      const m = String(dt.getMonth() + 1).padStart(2, "0");
      const d = String(dt.getDate()).padStart(2, "0");
      document.getElementById("inputDate").value = `${y}-${m}-${d}`;
    }
  }
  // 2. Category

```

```

const catMatch = text.match(/category[:\s]*([a-zA-Z]+)/i);
if (catMatch) {
  const cat = catMatch[1].trim().toLowerCase();
  const sel = document.getElementById("inputCategory");
  for (let i = 0; i < sel.options.length; i++) {
    if (sel.options[i].value.toLowerCase() === cat) {
      sel.selectedIndex = i;
      break;
    }
  }
}
// 3. Amount
const amtMatch = text.match(/amount[:\s]*([0-9]+)/i);
if (amtMatch) {
  document.getElementById("inputAmount").value = amtMatch[1];
}
// 4. Note
const noteMatch = text.match(/note[:\s]*(.+)$/i);
if (noteMatch) {
  document.getElementById("inputNote").value = noteMatch[1].trim();
}
}

submitBtn.onclick = () => {
  const d = encodeURIComponent(document.getElementById("inputDate").value);
  const c = encodeURIComponent(document.getElementById("inputCategory").value);
  const a = encodeURIComponent(document.getElementById("inputAmount").value);
  const n = encodeURIComponent(document.getElementById("inputNote").value);

  if (!d || !c || !a) {
    statusDiv.innerText = "⚠️ Date, Category & Amount required.";
    return;
  }
  statusDiv.innerText = "Saving...";
  submitBtn.disabled = true;

  fetch(`${WEB_APP_URL}?date=${d}&category=${c}&amount=${a}&note=${n}`)
    .then(r => r.text())
    .then(txt => {
      statusDiv.innerText = txt.trim() === "SUCCESS"
        ? "✅ Saved to sheet!"
        : "⚠️ Server: " + txt;
    })
    .catch(err => {
      statusDiv.innerText = "❌ Error: " + err.message;
    });
};
</script>

```

```
</body>  
</html>
```