

Exercise 9 — AI Meeting Minutes Generator (Google Sheets + Gemini)

The screenshot shows a Google Sheets spreadsheet on the left and a side panel titled "AI Meeting Minutes Generator" on the right. In the spreadsheet, cell A2 contains the text "Today we discussed the Q4 marketing campaign. Sarah will handle the new social media plan, John will redesign the landing page by next week, and we agreed to review budget allocations on Friday." This text is highlighted with a blue border. The side panel has a title "AI Meeting Minutes Generator" and a close button. Below the title is a text input area with the same text as cell A2. A blue button labeled "Summarize" is at the bottom of the panel.

	A	B	C
1	Meeting Transcr	Summary	Key Points
2	Today we discussed the Q4 marketing campaign. Sarah will handle the new social media plan, John will redesign the landing page by next week, and we agreed to review budget allocations on Friday.		
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

Meetings are important — but summarizing them can be tedious.

What if your meeting notes could **write themselves**?

In this project, you'll build an **AI Meeting Minutes Generator** right inside **Google Sheets**, powered by **Gemini 2.5 Flash** and **Google Apps Script**.

This exercise shows how to take raw meeting transcripts (from Zoom, Google Meet, Teams, etc.) and automatically generate:

- ☒ Concise meeting summaries
- ☒ Key discussion points
- ☒ Action items with owners and due dates
- ☒ Follow-up questions

All formatted directly in your spreadsheet.

What You'll Learn

By the end of this exercise, you'll understand how to:

- Connect **Gemini** to **Google Sheets** via Apps Script REST API
 - Process long text transcripts safely
 - Parse AI responses in JSON format
 - Display results neatly across multiple columns
 - Build a sidebar UI for interactive AI summaries
-

Step 1 — Create Your Sheet & Script

1. Open a new Google Sheet.
2. Go to **Extensions** → **Apps Script**.
3. Create three files:
 - `Code.gs`
 - `Sidebar.html`
 - `appscript.json`

You'll paste the full code from this tutorial into those files.

Step 2 — The Apps Script (`Code.gs`)

This file handles:

- The custom **menu** (“AI Tools”)
- The **Gemini API call** using `UrlFetchApp`
- A **robust JSON parser** to clean up model output
- The **meeting summarization** logic
- Sample data insertion

Here’s the complete version, including all recent fixes:

```
/**
 * Exercise 9 – AI Meeting Minutes Generator
 * Author: Laurence “Lars” Svekis
 * Google Sheets + Gemini REST API
 */

const GEMINI_API_KEY = ''; // or set in Script Properties
const GEMINI_MODEL = 'gemini-2.5-flash';
const GEMINI_BASE =
'https://generativelanguage.googleapis.com/v1beta';
const DEFAULT_TEMP = 0.3;
const DEFAULT_TOKENS = 1024;
const COOLDOWN_MS = 150;

// === MENU ===
function onOpen() {
  SpreadsheetApp.getUi()
    .createMenu('🧠 AI Tools')
    .addItem('Summarize Meeting', 'summarizeMeeting')
    .addItem('Open Sidebar', 'showSidebar')
    .addItem('Insert Sample Data', 'insertSampleMeetings')
    .addToUi();
}

function showSidebar() {
  try {
```

```

    const html = HtmlService.createHtmlOutputFromFile('Sidebar')
        .setTitle('AI Meeting Minutes Generator')
        .setWidth(400);
    SpreadsheetApp.getUi().showSidebar(html);
} catch (e) {
    SpreadsheetApp.getUi().alert('❌ Failed to open sidebar: ' +
e.message);
}
}

function getApiKey() {
    return
PropertiesService.getScriptProperties().getProperty('GEMINI_API_KEY')
|| GEMINI_API_KEY;
}

function callGemini(prompt, tokens = DEFAULT_TOKENS) {
    const apiKey = getApiKey();
    if (!apiKey) return '❌ Missing API key';
    const payload = {
        contents: [{ role: "user", parts: [{ text: prompt }] }],
        generationConfig: { temperature: DEFAULT_TEMP, maxOutputTokens:
tokens, topP: 0.9, topK: 40 }
    };
    const url =
`${GEMINI_BASE}/models/${encodeURIComponent(GEMINI_MODEL)}:generateCon
tent?key=${encodeURIComponent(apiKey)}`;
    const res = UrlFetchApp.fetch(url, { method: 'post', contentType:
'application/json', payload: JSON.stringify(payload),
muteHttpExceptions: true });
    const data = JSON.parse(res.getContentText());
    return data?.candidates?.[0]?.content?.parts?.[0]?.text || '⚠️ No
response';
}

function safeJson(raw) {
    try {

```

```

    let s = String(row).trim().replace(/^```json/i,
    '').replace(/```$/i, '').trim();
    return JSON.parse(s);
  } catch (e) { return null; }
}

function summarizeMeeting() {
  const sheet = SpreadsheetApp.getActiveSheet();
  const range = sheet.getActiveRange();
  const values = range.getValues();
  const out = [];

  for (const row of values) {
    const transcript = String(row[0] || '').trim();
    if (!transcript) { out.push(['']); continue; }

    const prompt = [
      "You are an AI meeting assistant. Summarize the following",
      "meeting transcript into JSON.",
      "Output only JSON with these keys:",
      "{ \"summary\": string, \"key_points\": string[],",
      "\"action_items\": [{\"owner\": string, \"task\": string, \"due\":",
      "string}], \"follow_ups\": string[] }",
      "",
      transcript
    ].join('\n');

    const raw = callGemini(prompt, 1536);
    const parsed = safeJson(raw);
    if (!parsed) {
      out.push(['⚠️ Could not parse JSON', raw, '', '', '']);
    } else {
      const summary = parsed.summary || '';
      const points = (parsed.key_points || []).join('\n• ');
      const actions = (parsed.action_items || []).map(a => `${a.owner}
|| 'N/A' - ${a.task || ''} (${a.due || 'no date'})`).join('\n');
      const follow = (parsed.follow_ups || []).join('\n');
      out.push([summary, points, actions, follow]);
    }
  }
}

```

```

    }
    Utilities.sleep(COOLDOWN_MS);
  }

  const startCol = range.getColumn() + 1;
  sheet.getRange(range.getRow(), startCol, out.length,
4).setValues(out);
  SpreadsheetApp.getActive().toast('✅ Meeting summaries generated');
}

function insertSampleMeetings() {
  const data = [
    ['Meeting Transcript', 'Summary', 'Key Points', 'Action Items',
'Follow Ups'],
    ['Today we discussed the Q4 marketing campaign. Sarah will handle
the new social media plan, John will redesign the landing page by next
week, and we agreed to review budget allocations on Friday.', '', '',
'', ''],
    ['Engineering sync: Maria confirmed API integration done, Alex
testing new endpoints, and QA will start regression testing tomorrow.
Decision: Launch postponed to next Monday.', '', '', '', ''],
    ['Finance update: Reviewed Q3 numbers, expenses within limits.
Next: finalize vendor payments by Thursday and prepare Q4 forecast.',
'', '', '', '']
  ];
  const sheet = SpreadsheetApp.getActiveSheet();
  sheet.clearContents();
  sheet.getRange(1, 1, data.length, data[0].length).setValues(data);
  SpreadsheetApp.getActive().toast('✅ Sample meeting data inserted
(A-E)');
}

```

Step 3 — Sidebar.html

A simple interface where users can paste transcripts and generate summaries interactively.

```
<!DOCTYPE html>
```

```

<html>
  <head>
    <meta charset="utf-8">
    <style>
      body { font-family: system-ui; padding: 1rem; }
      textarea { width: 100%; height: 180px; }
      button { margin-top: 8px; background: #1a73e8; color: #fff;
border: none; padding: 8px 12px; border-radius: 6px; cursor: pointer;
}
      pre { background: #f1f3f4; padding: 10px; border-radius: 8px;
white-space: pre-wrap; }
    </style>
  </head>
  <body>
    <h2>📝 AI Meeting Minutes Generator</h2>
    <p>Paste a transcript below:</p>
    <textarea id="text"></textarea>
    <button onclick="summarize()">Summarize</button>
    <pre id="out"></pre>

    <script>
      function summarize() {
        const t = document.getElementById('text').value;
        document.getElementById('out').textContent = ' ⌚
Summarizing...';
        google.script.run
          .withSuccessHandler(r =>
document.getElementById('out').textContent = r)
          .withFailureHandler(e =>
document.getElementById('out').textContent = e.message)
          .callGemini(t, 1024);
      }
    </script>
  </body>
</html>

```

Step 4 — Manifest (appsscript.json)

```
{
  "timeZone": "America/Toronto",
  "exceptionLogging": "STACKDRIVER",
  "runtimeVersion": "V8",
  "oauthScopes": [
    "https://www.googleapis.com/auth/spreadsheets",
    "https://www.googleapis.com/auth/script.container.ui",
    "https://www.googleapis.com/auth/script.external_request"
  ]
}
```

Step 5 — Test the Project

1. In your Sheet, go to **AI Tools** → **Insert Sample Data**
You'll get three sample meeting transcripts in column **A**.
 2. Select column **A** and choose **AI Tools** → **Summarize Meeting**.
 3. Gemini will fill columns **B–E** automatically with:
 - Summary
 - Key Points
 - Action Items
 - Follow Ups
 4. Try **AI Tools** → **Open Sidebar** to test custom transcripts.
-

Example Output

Meeting Transcript	Summary	Key Points	Action Items	Follow Ups
--------------------	---------	------------	--------------	------------

Discussed Q4 marketing campaign...	Reviewed campaign progress, assigned key responsibilities.	New social media plan Landing page redesign Budget review	Sarah — social media plan (next week) John — landing page redesign (next week)	Review budget allocations Friday
------------------------------------	--	---	---	-------------------------------------



How It Works

- Each transcript is turned into a **structured JSON** by Gemini.
 - The script safely parses this JSON and outputs text into 4 columns.
 - The sidebar version allows quick manual summarization for testing.
 - You can expand this to save summaries in Docs or send email recaps automatically.
-



Custom Ideas to Try

- Add a “Generate Google Doc” button that saves formatted minutes.
 - Auto-detect the meeting title based on the first line.
 - Add timestamps or AI confidence scoring for each section.
 - Extend the script to support multiple Sheets for different teams.
-



What You’ve Built

- ✓ A functional AI-powered meeting assistant inside Google Sheets
- ✓ Real-time text summarization with Gemini
- ✓ A structured workflow for productivity automation

This project is part of the **AI + Apps Script Learning Series** — practical exercises that teach how to integrate Gemini with everyday Google Workspace tools.