

AI TASKS IN GOOGLE SHEETS



Build an AI-Powered Google Sheet with One Click

 Build an AI-Powered Google Sheet with One Click	1
Automatically Populate Your Spreadsheet with Demo Data for All Gemini Exercises	2
 What This Demo Setup Script Does	3
✓ 1. A topic for idea-generation tools (A1)	3
✓ 2. Realistic sample feedback items in A2:A10	3
✓ 3. Meeting notes for summary generation (D2:D5)	3
✓ 4. Email drafting sample inputs (F2:H2)	3

✓ 5. Natural-language → formula prompt (F6)	3
✓ 6. A topic for header suggestion (A13)	3
 The Complete Setup Script	3
 How to Use the Demo Sheet	5
 Why This Is Powerful for Beginners	5
 Test Function: testAllGeminiSheets()	6
 How to run it	10
 Gemini + Google Sheets Toolkit — 5 New AI Exercises	11
♦ Exercise 1 — Categorize Text in A2:A into B2:B	11
What it does	11
How to use it	11
Code	11
♦ Exercise 2 — Summarize Notes in A2:A into B2:B	12
What it does	12
How to use it	12
Code	12
♦ Exercise 3 — Extract Keywords/Tags from A2:A into B2:B	13
What it does	13
How to use it	13
Code	13
♦ Exercise 4 — Suggest Column Headers Based on a Topic	14
What it does	14
How to use it	14
Code	14
♦ Exercise 5 — Generate a Formula from a Natural-Language Description	15
What it does	15
How to use it	15
Code	16
 Optional: Add These to Your “Gemini AI” Menu	16

Automatically Populate Your Spreadsheet with Demo Data for All Gemini Exercises

With *one click*, it builds a ready-to-use spreadsheet packed with realistic sample data for:

- Categorizing text
- Extracting keywords
- Generating summaries
- Translating content
- Creating TODO lists from meeting notes
- Writing email drafts
- Generating column headers

- Converting natural-language descriptions into formulas
- And more!

Combined with your **Gemini AI menu**, this creates a complete playground for learning how AI can enhance spreadsheets.



What This Demo Setup Script Does

When you run `setupGeminiSheetsDemo()`, your Google Sheet is instantly filled with structured sample content designed to work perfectly with all your Gemini-powered functions. Here's what it populates:

✓ 1. A topic for idea-generation tools (A1)

Used by:

- `sheets_topicIdeas()`
- `sheets_topicIdeasPerRow()`

✓ 2. Realistic sample feedback items in A2:A10

Perfect for:

- Categorizing
- Summarizing
- Keyword extraction
- Translating

Examples include:

“User reports slow loading times on dashboard”

“Request to add dark mode theme”

“Feedback: onboarding process is confusing”

✓ 3. Meeting notes for summary generation (D2:D5)

Used by:

- `sheets_meetingSummaryFromRange()`

✓ 4. Email drafting sample inputs (F2:H2)

Used by:

- `sheets_emailDraftFromInputs()`

✓ 5. Natural-language → formula prompt (F6)

Used by:

- `sheets_generateFormulaFromDescription()`

Example:

“Sum all values in C2:C where B2:B equals ‘Paid’”

✓ 6. A topic for header suggestion (A13)

Used by:

- `sheets_suggestHeadersFromTopic()`



The Complete Setup Script

Copy/paste into **SheetsExamples.gs** or any new Apps Script file:

```
function setupGeminiSheetsDemo() {
  const sheet = getActiveSheetOrError_();
  sheet.clear();

  // Title topic
  sheet.getRange("A1")
    .setValue("Project management improvements")
    .setBackground("#d9ead3")
    .setFontWeight("bold");

  sheet.getRange("B1").setValue("Gemini Output");

  // Sample items for A2:A10
  const sampleList = [
    ["User reports slow loading times on dashboard"],
    ["Request to add dark mode theme to UI"],
    ["Billing issue: invoice shows incorrect amount"],
    ["Customer unable to reset password via mobile app"],
    ["Feedback: onboarding process is confusing"],
    ["Bug: notifications not appearing for new messages"],
    ["Idea: implement weekly analytics summary email"],
    ["Question: how to export data in CSV format"],
    ["Support needed: user cannot access shared folder"]
  ];
  sheet.getRange(2, 1, sampleList.length, 1)
    .setValues(sampleList)
    .setBackground("#fff2cc");

  // Meeting notes
  sheet.getRange("D1").setValue("Meeting
Notes").setFontWeight("bold");
  sheet.getRange("D2").setValue("Improve load time by optimizing
database queries.");
  sheet.getRange("D3").setValue("Add dark mode as a Q3 UI
enhancement.");
  sheet.getRange("D4").setValue("Fix invoice calculation bug.");
  sheet.getRange("D5").setValue("Simplify onboarding documentation.");

  // Email drafting inputs
  sheet.getRange("F1").setValue("Topic:");
  sheet.getRange("G1").setValue("Recipient:");
  sheet.getRange("H1").setValue("Context:");
  sheet.getRange("F2").setValue("Delay in project timeline");
```

```

sheet.getRange("G2").setValue("Project Manager");
sheet.getRange("H2").setValue("We need an update on expected testing
completion.");

// Natural language → formula example
sheet.getRange("F5").setValue("Natural Language → Sheets
Formula").setFontWeight("bold");
sheet.getRange("F6").setValue("Sum all values in C2:C where B2:B
equals 'Paid'");

// Header suggestion topic
sheet.getRange("A12").setValue("Topic for header generation:");
sheet.getRange("A13")
    .setValue("Customer support ticket tracking sheet")
    .setBackground("#cfe2f3");

// Helper notes
sheet.getRange("A15").setValue(
    "Demo Ready: Use the Gemini AI menu to test idea generation,
    categorization, translation, summarization, keywords, headers,
    formulas, email drafting, and meeting summaries."
);

sheet.autoResizeColumns(1, 8);
}

```



How to Use the Demo Sheet

1. Open your Google Sheet
2. Go to **Extensions** → **Apps Script**
3. Add the demo setup function & save
4. Run `setupGeminiSheetsDemo()` once
5. Reload the Sheet
6. Open your **Gemini AI** menu and try any function

Your spreadsheet becomes an *instant AI sandbox* where all Gemini examples work right away.



Why This Is Powerful for Beginners

Many people struggle when learning Gemini in Sheets because:

- They don't know what data to use
- Example sheets are inconsistent
- They can't reproduce instructor demos
- Setup takes longer than the actual lesson

This function solves all of that.

With one click, beginners get:

- ✓ Clean, structured sample data
- ✓ Realistic examples
- ✓ A predictable testing environment
- ✓ A polished starting point for tutorials
- ✓ A consistent template for workshops or classrooms

Whether you're teaching AI automation or onboarding teams to Apps Script, this demo sheet makes everything smoother.

Below is a **testAllGeminiSheets()** function that:

- Calls `setupGeminiSheetsDemo()`
- Sets up a sample formula for the "explain formula" tool
- Temporarily tweaks A1 where needed (for TODOs, tutor, formula)
- Calls every Gemini function once
- Uses `Logger.log()` so you can see progress in **View** → **Logs**

⚠ Note: This will fire *many* Gemini requests in one go. On the free tier you might hit rate/usage limits if you run it repeatedly.



Test Function: **testAllGeminiSheets()**

Add this to **SheetsExamples.gs** (in the same project as your other functions):

```
/**
 * Run a full end-to-end test of all Gemini Sheets examples.
 *
 * REQUIREMENTS:
 * - SharedHelpers.gs, SheetsExamples.gs, SheetsMenu.gs are in the
project.
 * - setupGeminiSheetsDemo() exists and populates the sheet with
sample data.
 *
 * This will:
 * 1. Reset and populate demo data.
 * 2. Create a sample formula and set it as the active cell.
 * 3. Call every Gemini-powered function once.
 */
```

```

* Check:
* - The sheet for updated values.
* - Dialogs/popups for explanations, where applicable.
* - Logs (View → Logs) for progress messages.
*/
function testAllGeminiSheets() {
  const sheet = getActiveSheetOrError_();
  const ss = SpreadsheetApp.getActiveSpreadsheet();
  ss.setActiveSheet(sheet);

  Logger.log('=== Step 1: Setup demo data ===');
  setupGeminiSheetsDemo();

  // -----
  // Prepare a sample formula for sheets_explainActiveFormula
  // -----
  Logger.log('Preparing sample formula in E2 for
explainActiveFormula...');
  sheet.getRange('E1')
    .setValue('Sample formula for testing')
    .setFontWeight('bold');
  sheet.getRange('E2').setFormula('=SUM(1, 2, 3)');
  sheet.setActiveSelection(sheet.getRange('E2'));

  Logger.log('Running sheets_explainActiveFormula()...');
  sheets_explainActiveFormula();

  // -----
  // Idea generators - use topic in A1
  // -----
  Logger.log('Running sheets_topicIdeas()...');
  sheets_topicIdeas();

  Logger.log('Running sheets_topicIdeasPerRow()...');
  sheets_topicIdeasPerRow();

  // -----
  // Translate A2:A → B2:B
  // (You may get a language prompt dialog)
  // -----
  Logger.log('Running sheets_translateAtoB()...');
  sheets_translateAtoB();

  // -----

```

```

// TODOs from notes in A1
// Temporarily override A1 with a note-style summary.
// -----
Logger.log('Running sheets_todosFromNotes()...');
const originalA1_forTODO = sheet.getRange('A1').getValue();
sheet
  .getRange('A1')
  .setValue(
    'Discuss project delays, dark mode feature request, and recent
billing issues with the team.'
  );
sheets_todosFromNotes();
sheet.getRange('A1').setValue(originalA1_forTODO);

// -----
// Email draft from A1/B1/C1 → D1
// Demo data already in F2:H2, but our function expects A1/B1/C1?
// If your implementation uses F2:H2, adjust this section as needed.
// Here we copy F2:H2 into A1:C1 temporarily.
// -----
Logger.log('Running sheets_emailDraftFromInputs()...');
const f2 = sheet.getRange('F2').getValue();
const g2 = sheet.getRange('G2').getValue();
const h2 = sheet.getRange('H2').getValue();

const originalA1_forEmail = sheet.getRange('A1').getValue();
const originalB1_forEmail = sheet.getRange('B1').getValue();
const originalC1_forEmail = sheet.getRange('C1').getValue();

sheet.getRange('A1').setValue(f2);
sheet.getRange('B1').setValue(g2);
sheet.getRange('C1').setValue(h2);

sheets_emailDraftFromInputs();

// Restore original A1-C1
sheet.getRange('A1').setValue(originalA1_forEmail);
sheet.getRange('B1').setValue(originalB1_forEmail);
sheet.getRange('C1').setValue(originalC1_forEmail);

// -----
// Meeting summary from A2:A10 → B1
// Uses the feedback items set up in the demo.
// -----

```



```

Logger.log('Running sheets_meetingSummaryFromRange()...');
sheets_meetingSummaryFromRange();

// -----
// Tutor answer from A1 question → B1
// Temporarily set A1 to a question.
// -----
Logger.log('Running sheets_tutorAnswerFromQuestion()...');
const originalA1_forTutor = sheet.getRange('A1').getValue();
sheet
  .getRange('A1')
  .setValue('What is cloud computing and why is it useful?');
sheets_tutorAnswerFromQuestion();
sheet.getRange('A1').setValue(originalA1_forTutor);

// -----
// Categorize A2:A → B2:B
// -----
Logger.log('Running sheets_categorizeAtoB()...');
sheets_categorizeAtoB();

// -----
// Summarize notes A2:A → B2:B
// -----
Logger.log('Running sheets_summarizeNotesAtoB()...');
sheets_summarizeNotesAtoB();

// -----
// Keywords from A2:A → B2:B
// -----
Logger.log('Running sheets_keywordsFromAtoB()...');
sheets_keywordsFromAtoB();

// -----
// Suggest headers from topic in A13 → row 2
// We copy the demo topic from A13 into A1 temporarily
// because sheets_suggestHeadersFromTopic() expects A1.
// -----
Logger.log('Running sheets_suggestHeadersFromTopic()...');
const originalA1_forHeaders = sheet.getRange('A1').getValue();
const topicForHeaders = sheet.getRange('A13').getValue();
sheet.getRange('A1').setValue(topicForHeaders);
sheets_suggestHeadersFromTopic();
sheet.getRange('A1').setValue(originalA1_forHeaders);

```

```
// -----
// Generate formula from description in A1 → B1
// Temporarily write a formula description into A1.
// -----
Logger.log('Running sheets_generateFormulaFromDescription()...');
const originalA1_forFormula = sheet.getRange('A1').getValue();
sheet
  .getRange('A1')
  .setValue('Sum all values in C2:C where B2:B equals "Paid".');
sheets_generateFormulaFromDescription();
sheet.getRange('A1').setValue(originalA1_forFormula);

Logger.log('=== Done: testAllGeminiSheets completed. Check the Sheet
and dialogs for results. ===');
}
```



How to run it

1. Open your **Google Sheet** with the Gemini toolkit attached.
2. Go to **Extensions** → **Apps Script**.
3. Make sure these exist in the project:
 - SharedHelpers.gs
 - SheetsExamples.gs (with all your functions)
 - SheetsMenu.gs
 - setupGeminiSheetsDemo()
4. Paste the testAllGeminiSheets() function into SheetsExamples.gs (or a new file).
5. Click **Run** → **testAllGeminiSheets** and approve permissions if prompted.
6. Watch:
 - Popups for explanations and responses
 - The sheet filling with AI-generated outputs
 - Logs in **View** → **Logs** for progress



Gemini + Google Sheets Toolkit — 5

New AI Exercises

These 5 new functions make Google Sheets even more useful with Gemini:

1. Categorize text in A2:A → category labels in B2:B
2. Summarize long notes in A2:A → short summaries in B2:B
3. Extract keywords/tags from A2:A → tags in B2:B
4. Suggest column headers for a dataset based on a topic
5. Generate a Google Sheets formula from a plain-language description

All of them use the existing `callGemini(prompt)` helper.

♦ Exercise 1 — Categorize Text in A2:A into B2:B

What it does

Reads each row in **A2:A** (e.g., “Bug report”, “Feature request”, “Billing issue”) and asks Gemini to assign a **simple category label** into **B2:B**.

Perfect for:

- Support tickets
- Feedback logs
- Idea lists
- Survey responses

How to use it

1. In your Sheet, put text labels in **A2:A** (e.g., short descriptions of items).
2. Run `sheets_categorizeAtoB()` from the editor or via your Gemini menu.
3. Column B will be filled with categories like “Bug”, “Idea”, “Billing”, etc.

Code

```
/**
 * Categorize each text entry in A2:A into a simple label in B2:B.
 */
function sheets_categorizeAtoB() {
  const sheet = getActiveSheetOrError_();
  const range = sheet.getRange('A2:A');
  const values = range.getValues();

  const output = [];

  values.forEach(row => {
    const text = (row[0] || '').toString().trim();
    if (!text) {
      output.push(['']);
      return;
    }
  });

  const prompt =
```

```

        'You are a simple text categorizer. Read the item below and
return ONE short ' +
        'category label (like "Bug", "Idea", "Question", "Billing",
"Feedback", etc.). ' +
        'Return ONLY the category word or phrase.\n\n' +
        text;

    const category = callGemini(prompt).trim();
    output.push([category]);
  });

  sheet.getRange(2, 2, output.length, 1).setValues(output);
}

```

◆ Exercise 2 — Summarize Notes in A2:A into B2:B

What it does

Summarizes each row's text in **A2:A** into a **very short summary** in **B2:B** (1–2 sentences).
Great for:

- Meeting notes
- Call summaries
- Ticket descriptions
- Long comments

How to use it

1. Put your notes or descriptions in **A2:A**.
2. Run `sheets_summarizeNotesAtoB()`.
3. Column B will contain a short, readable summary for each row.

Code

```

/**
 * Summarize each note in A2:A into a short 1-2 sentence summary in
B2:B.
 */
function sheets_summarizeNotesAtoB() {
  const sheet = getActiveSheetOrError_();
  const range = sheet.getRange('A2:A');
  const values = range.getValues();

  const output = [];

  values.forEach(row => {
    const text = (row[0] || '').toString().trim();
    if (!text) {
      output.push(['']);
    }
  });
}

```

```

        return;
    }

    const prompt =
        'Summarize the following note into 1-2 sentences that capture
the main point, ' +
        'using clear, simple language:\n\n' +
        text;

    const summary = callGemini(prompt).trim();
    output.push([summary]);
});

sheet.getRange(2, 2, output.length, 1).setValues(output);
}

```

◆ Exercise 3 — Extract Keywords/Tags from A2:A into B2:B

What it does

Reads each text in **A2:A** and returns **comma-separated keywords/tags** in **B2:B**.

Useful for:

- Tagging content
- Blog ideas
- Categorizing feedback
- Creating search labels

How to use it

1. Put your raw text in **A2:A** (e.g., “User cannot reset password on mobile app”).
2. Run `sheets_keywordsFromAtoB()`.
3. Column B will show keywords like: auth, password reset, mobile app.

Code

```

/**
 * Extract keywords/tags from A2:A into B2:B (comma-separated).
 */
function sheets_keywordsFromAtoB() {
    const sheet = getActiveSheetOrError_();
    const range = sheet.getRange('A2:A');
    const values = range.getValues();

    const output = [];

    values.forEach(row => {
        const text = (row[0] || '').toString().trim();

```

```

    if (!text) {
      output.push(['']);
      return;
    }

    const prompt =
      'Read the text below and return 3–7 short keywords or tags that capture the main ideas. ' +
      'Return them as a single comma-separated line (no explanations):\n\n' +
      text;

    const tags = callGemini(prompt).trim();
    output.push([tags]);
  });

  sheet.getRange(2, 2, output.length, 1).setValues(output);
}

```

♦ Exercise 4 — Suggest Column Headers Based on a Topic

What it does

Uses Gemini to suggest **column headers** for a dataset based on a topic in **A1**.

It writes the headers into **row 2**: A2, B2, C2, D2 (or more, depending on output length).

Great for:

- Starting a new tracker
- Designing logging sheets
- Planning data structures for a project

How to use it

1. In **A1**, write a topic like:
 - “Content calendar for social media”
 - “Bug tracking for web app”
 - “Student progress tracker”
2. Run `sheets_suggestHeadersFromTopic()`.
3. Headers appear in **row 2** (A2, B2, C2, ...).

Code

```

/**
 * Suggest column headers based on a topic in A1.
 * Writes headers into row 2 (A2, B2, C2, ...).
 */
function sheets_suggestHeadersFromTopic() {

```



```

const sheet = getActiveSheetOrError_();
const topic = (sheet.getRange('A1').getValue() ||
'').toString().trim();

if (!topic) {
  sheet.getRange('A2').setValue('Enter a topic in A1 first.');
```

return;

}

```

const prompt =
  'You are designing a table in a spreadsheet. Based on the topic
below, ' +
  'suggest 4–8 concise column headers. Return them as a single line
separated by commas.\n\n' +
  topic;

const headersLine = callGemini(prompt).trim();
const headers = headersLine
  .split(',')
  .map(h => h.trim())
  .filter(Boolean);

if (headers.length === 0) {
  sheet.getRange('A2').setValue('No headers generated.');
```

return;

}

```

// Write headers into row 2 starting at column A
sheet.getRange(2, 1, 1, headers.length).setValues([headers]);
}
```

♦ Exercise 5 — Generate a Formula from a Natural-Language Description

What it does

Takes a plain-language description in **A1** (e.g., “sum all values in column C where column B is 'Paid'”) and asks Gemini to suggest a **Google Sheets formula** in **B1**.

Perfect for:

- Beginners learning formulas
- Quick prototyping
- Helping non-technical users

How to use it

1. In **A1**, type what you want, for example:

- “Count how many rows in A2:A have the word 'error'”
 - “Sum values in C2:C where B2:B equals 'Paid'”
2. Run `sheets_generateFormulaFromDescription()`.
 3. B1 will contain a suggested formula like:
`=SUMIF(B2:B, "Paid", C2:C)`

Code

```
/**
 * Generate a Google Sheets formula from a natural-language
 * description in A1.
 * Writes the suggested formula into B1.
 */
function sheets_generateFormulaFromDescription() {
  const sheet = getActiveSheetOrError_();
  const description = (sheet.getRange('A1').getValue() ||
  '').toString().trim();

  if (!description) {
    sheet
      .getRange('B1')
      .setValue('Enter a formula description in A1 first. ');
    return;
  }

  const prompt =
    'You are an expert in Google Sheets formulas. Read the description' +
    'below and ' +
    'return ONE valid Google Sheets formula that does what is' +
    'requested. ' +
    'Return ONLY the formula, starting with = and with no' +
    'explanation:\n\n' +
    description;

  const formula = callGemini(prompt).trim();
  sheet.getRange('B1').setValue(formula);
}
```



Optional: Add These to Your “Gemini AI” Menu

If you already have a `SheetsMenu.gs` with `onOpen()`, just add these lines into the menu chain:

```
function onOpen() {
  const ui = SpreadsheetApp.getUi();
```

```

ui.createMenu('Gemini AI')
  .addItem('Idea: A1 → A2 (block)', 'sheets_topicIdeas')
  .addItem('Ideas: A1 → A3:A7 (rows)', 'sheets_topicIdeasPerRow')
  .addItem('Explain active formula', 'sheets_explainActiveFormula')
  .addSeparator()
  .addItem('Translate A2:A → B2:B', 'sheets_translateAtoB')
  .addItem('TODOs from notes in A1', 'sheets_todosFromNotes')
  .addSeparator()
  .addItem('Email draft from A1/B1/C1',
'sheets_emailDraftFromInputs')
  .addItem('Meeting summary A2:A10 → B1',
'sheets_meetingSummaryFromRange')
  .addItem('Tutor answer from A1 → B1',
'sheets_tutorAnswerFromQuestion')
  .addSeparator()
  // New ones:
  .addItem('Categorize A2:A → B2:B', 'sheets_categorizeAtoB')
  .addItem('Summarize notes A2:A → B2:B',
'sheets_summarizeNotesAtoB')
  .addItem('Keywords from A2:A → B2:B', 'sheets_keywordsFromAtoB')
  .addItem('Suggest headers from topic in A1',
'sheets_suggestHeadersFromTopic')
  .addItem('Generate formula from description in A1',
'sheets_generateFormulaFromDescription')
  .addToUi();
}

```