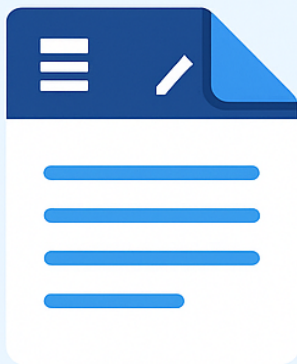


Gemini + Google Workspace Toolkit

Gemini + Google Workspace Toolkit

Google Docs & Google Sheets AI Helpers
with Custom Menus



- Summarize text
 - Generate ideas
 - Translate text
 - Create quiz
 - Suggest comment
- Summarise comment

Gemini AI



Google Docs & Google Sheets AI Helpers with Custom Menus

Github <https://github.com/lsvakis/20-Gemini-Exercises>

This toolkit lets you add **Gemini-powered AI features** directly into:

- **Google Docs** – summarize, rewrite, outline, generate quizzes, suggest comments

- **Google Sheets** – custom `=GEMINI()` function, ideas, translations, summaries, emails, tutor answers

All with:

- A shared `SharedHelpers.gs` file
- A **Docs** script file
- A **Sheets** script file
- Optional **custom UI menus** so users can run tools from a menu instead of the script editor



1. Project Structure

You'll create **two separate container-bound projects**:

A) Google Docs toolkit (Doc-bound)

Files:

- `SharedHelpers.gs`
- `DocsExamples.gs`
- `DocsMenu.gs` (for the custom menu)

B) Google Sheets toolkit (Sheet-bound)

Files:

- `SharedHelpers.gs`
- `SheetsExamples.gs`
- `SheetsMenu.gs` (for the custom menu)

The same `SharedHelpers.gs` code can be reused in both projects.



2. Shared Helpers (used by both Docs & Sheets)

Create a file named **`SharedHelpers.gs`** and paste:

```
/**
 * Shared Gemini Helpers
 *
 * Used by both DocsExamples.gs and SheetsExamples.gs.
 *
 * SETUP:
 * 1. In the Apps Script editor, open Project Settings → Script
properties.
 * 2. Add a property:
 *     Key:  GEMINI_API_KEY
```

```

*      Value: your Gemini API key
* 3. Save.
*/

const GEMINI_MODEL_ID = 'gemini-2.5-flash';

/**
 * Get Gemini API key from Script Properties.
 * Throws a clear error if missing.
 */
function getGeminiApiKey_() {
  const scriptProps = PropertiesService.getScriptProperties();
  const key = scriptProps.getProperty('GEMINI_API_KEY');
  if (!key) {
    throw new Error(
      'GEMINI_API_KEY is not set.\n' +
      'In the Script Editor, open Project Settings → Script'
      'properties and add it.'
    );
  }
  return key;
}

/**
 * Core helper: send a text prompt to Gemini and return a text
 * response.
 *
 * @param {string} prompt
 * @returns {string} Plain text response from Gemini.
 */
function callGemini(prompt) {
  const apiKey = getGeminiApiKey_();

  const url =
    'https://generativelanguage.googleapis.com/v1beta/models/' +
    GEMINI_MODEL_ID +
    ':generateContent?key=' +
    apiKey;

  const payload = {
    contents: [{ parts: [{ text: prompt }] }]
  };

  const res = UrlFetchApp.fetch(url, {

```

```

    method: 'post',
    contentType: 'application/json',
    payload: JSON.stringify(payload),
    muteHttpExceptions: true
  });

  const text = res.getContentText();
  const json = JSON.parse(text);

  try {
    return json.candidates[0].content.parts[0].text;
  } catch (e) {
    throw new Error('Unexpected Gemini response: ' + text);
  }
}

/**
 * Helper: Get active sheet or throw a clear error if this isn't bound
to a Sheet.
 */
function getActiveSheetOrError_() {
  const ss = SpreadsheetApp.getActiveSpreadsheet();
  if (!ss) {
    throw new Error(
      'No active spreadsheet found.\n' +
      'Make sure this script is bound to a Google Sheet (Extensions
→ Apps Script).'
    );
  }
  return ss.getActiveSheet() || ss.getSheets()[0];
}

```

Set your API key once

In the Apps Script editor (for each project):

1. Click **Project Settings**
2. Under **Script properties**, click **Add script property**
3. Add:
 - Key: GEMINI_API_KEY
 - Value: your Gemini API key
4. Save.



3. Google Docs Toolkit (DocsExamples.gs)

Create a file named **DocsExamples.gs** and paste:

```
/**
 * Gemini + Google Docs Examples
 *
 * REQUIREMENT:
 * - This project must be bound to a Google Doc (open a Doc →
Extensions → Apps Script).
 * - SharedHelpers.gs must be in the same project (for callGemini()).
 */

/**
 * EXAMPLE 1 - Summarize selected text
 */
function docs_summarizeSelection() {
  const doc = DocumentApp.getActiveDocument();
  const sel = doc.getSelection();

  if (!sel) {
    DocumentApp.getUi().alert('Select some text first.');
```

return;

}

let text = '';

sel.getRangeElements().forEach(el => {

const element = el.getElement();

if (element.editAsText) {

text += element.asText().getText() + '\n';

}

});

if (!text.trim()) {

DocumentApp.getUi().alert('Selected text is empty.');

return;

}

const summary = callGemini('Summarize this text in a short paragraph:\n\n' + text);

DocumentApp.getUi().alert(summary);

}

```
/**
 * EXAMPLE 2 - Rewrite selected text in simpler language
```

```

*/
function docs_rewriteSimpler() {
  const doc = DocumentApp.getActiveDocument();
  const sel = doc.getSelection();

  if (!sel) {
    DocumentApp.getUi().alert('Select some text first.');
```

return;

```
  }

  let text = '';
  sel.getRangeElements().forEach(el => {
    const element = el.getElement();
    if (element.editAsText) {
      text += element.asText().getText() + '\n';
    }
  });

  if (!text.trim()) {
    DocumentApp.getUi().alert('Selected text is empty.');
```

return;

```
  }

  const simpler = callGemini(
    'Rewrite the following text at a Grade 6 reading level:\n\n' +
text
  );
  DocumentApp.getUi().alert(simpler);
}

/**
 * EXAMPLE 3 - Generate outline from a topic (prompt dialog)
 */
function docs_generateOutlineFromTopic() {
  const ui = DocumentApp.getUi();
  const response = ui.prompt(
    'Outline Generator',
    'Enter a topic:',
    ui.ButtonSet.OK_CANCEL
  );

  if (response.getSelectedButton() !== ui.Button.OK) return;

  const topic = response.getResponseText().trim();

```

```

if (!topic) {
  ui.alert('Please enter a topic.');
```

return;

```

}

const outline = callGemini(
  'Create a short bullet-point outline for this topic:\n\n' + topic
);

const doc = DocumentApp.getActiveDocument();
const body = doc.getBody();
body.appendParagraph('\nOutline for: ' + topic).setBold(true);
body.appendParagraph(outline);
}

/**
 * EXAMPLE 4 - Create quiz questions from selected text
 */
function docs_quizFromSelection() {
  const doc = DocumentApp.getActiveDocument();
  const sel = doc.getSelection();

  if (!sel) {
    DocumentApp.getUi().alert('Select some text first.');
```

return;

```

  }

  let text = '';
  sel.getRangeElements().forEach(el => {
    const element = el.getElement();
    if (element.editAsText) {
      text += element.asText().getText() + '\n';
    }
  });

  if (!text.trim()) {
    DocumentApp.getUi().alert('Selected text is empty.');
```

return;

```

  }

  const quiz = callGemini(
    'Based on the text below, create 5 multiple choice questions with
4 options each and mark the correct answer:\n\n' +
    text

```

```

);

const body = doc.getBody();
body.appendParagraph('\nGenerated Quiz:').setBold(true);
body.appendParagraph(quiz);
}

/**
 * EXAMPLE 5 – Suggest a constructive comment for selected text
 */
function docs_suggestComment() {
  const doc = DocumentApp.getActiveDocument();
  const sel = doc.getSelection();

  if (!sel) {
    DocumentApp.getUi().alert('Select some text first.');
```

return;

```
  }

  let text = '';
  sel.getRangeElements().forEach(el => {
    const element = el.getElement();
    if (element.editAsText) {
      text += element.asText().getText() + '\n';
    }
  });

  if (!text.trim()) {
    DocumentApp.getUi().alert('Selected text is empty.');
```

return;

```
  }

  const comment = callGemini(
    'You are a helpful writing coach. Suggest one constructive comment
to improve this text:\n\n' +
    text
  );

  DocumentApp.getUi().alert('Suggested comment:\n\n' + comment);
}
```

What each Docs function does (for your guide)

- docs_summarizeSelection()
 - Summarizes highlighted text into a short paragraph.

- `docs_rewriteSimpler()`
Rewrites selected text at a Grade 6 reading level (accessibility / clarity).
- `docs_generateOutlineFromTopic()`
Prompts for a topic and inserts a bullet-point outline into the Doc.
- `docs_quizFromSelection()`
Generates 5 multiple-choice questions from selected text, with answers.
- `docs_suggestComment()`
Acts as a writing coach and suggests one constructive comment.



4. Google Sheets Toolkit (SheetsExamples.gs)

Create a file named **SheetsExamples.gs** and paste:

```
/**
 * Gemini + Google Sheets Examples
 *
 * REQUIREMENT:
 * - This project must be bound to a Google Sheet (open a Sheet →
Extensions → Apps Script).
 * - SharedHelpers.gs must be in the same project (for callGemini()
and getActiveSheetOrError_()).
 */

/**
 * CUSTOM FUNCTION – Use directly in Sheets:
 * =GEMINI("Write a haiku about coding.")
 *
 * @customfunction
 */
function GEMINI(prompt) {
  if (typeof prompt !== 'string') {
    return 'Prompt must be a string.';
  }
  return callGemini(prompt);
}

/**
 * EXAMPLE 1 – A1 topic → A2 ideas (single cell)
 */
function sheets_topicIdeas() {
  const sheet = getActiveSheetOrError_();
  const topic = sheet.getRange('A1').getValue();
```

```

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

```

    }

    const ideas = callGemini('Give me 5 creative ideas about: ' +
topic);
    sheet.getRange('A2').setValue(ideas);
  }

/**
 * EXAMPLE 2 - A1 topic → A3:A7 (one idea per row)
 */
function sheets_topicIdeasPerRow() {
  const sheet = getActiveSheetOrError_();
  const topic = sheet.getRange('A1').getValue();

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

```

    return;
  }

  const response = callGemini(
    'Give me 5 short bullet-point ideas (one per line) about: ' +
topic
  );

  const lines = response
    .split(/\r?\n/)
    .filter(line => line.trim() !== '')
    .map(line => [line.replace(/^-\+\s*/, '').trim()]);

  if (lines.length === 0) {
    sheet.getRange('A3').setValue('No ideas generated.');
```

```

    return;
  }

  sheet.getRange(3, 1, lines.length, 1).setValues(lines);
}

/**
 * EXAMPLE 3 - Explain formula in active cell
 */
function sheets_explainActiveFormula() {

```

```

const sheet = getActiveSheetOrError_();
const cell = sheet.getActiveCell();
const formula = cell.getFormula();

if (!formula) {
  SpreadsheetApp.getUi().alert('Active cell does not contain a
formula.');
```

return;

```

}

const explanation = callGemini(
  'Explain what this Google Sheets formula does in simple
terms:\n\n' +
  formula
);
SpreadsheetApp.getUi().alert(explanation);
}

/**
 * EXAMPLE 4 - Translate A2:A → B2:B into chosen language
 */
function sheets_translateAtoB() {
  const sheet = getActiveSheetOrError_();
  const ui = SpreadsheetApp.getUi();

  const langPrompt = ui.prompt(
    'Translation',
    'Translate to which language? (e.g., Spanish, French)',
    ui.ButtonSet.OK_CANCEL
  );

  if (langPrompt.getSelectedButton() !== ui.Button.OK) return;

  const targetLang = langPrompt.getResponseText().trim();
  if (!targetLang) {
    ui.alert('Please enter a valid language.');
```

return;

```

  }

  const values = sheet.getRange('A2:A').getValues();
  const out = [];

  values.forEach(row => {
    const text = row[0];
```

```

    if (!text) {
      out.push(['']);
      return;
    }
    const translated = callGemini(
      'Translate the following text into ' + targetLang + ':\n\n' +
text
    );
    out.push([translated]);
  });

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

/**
 * EXAMPLE 5 – Extract TODOs from meeting notes in A1 → B2:B
 */
function sheets_todosFromNotes() {
  const sheet = getActiveSheetOrError_();
  const notes = sheet.getRange('A1').getValue();

  if (!notes) {
    sheet.getRange('B1').setValue('Enter meeting notes in A1 first. ');
    return;
  }

  const result = callGemini(
    'From these meeting notes, extract a list of clear action items. '
+
    'Return one item per line:\n\n' +
    notes
  );

  const lines = result
    .split(/\r?\n/)
    .filter(line => line.trim() !== '')
    .map(line => [line.replace(/^-\+\s*/, '').trim()]);

  if (lines.length === 0) {
    sheet.getRange('B2').setValue('No action items found. ');
    return;
  }

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

```

```

}

/**
 * EXAMPLE 6 – Email draft from topic + recipient (A1, B1, C1 → D1)
 */
function sheets_emailDraftFromInputs() {
  const sheet = getActiveSheetOrError_();
  const topic = sheet.getRange('A1').getValue();
  const recipient = sheet.getRange('B1').getValue();
  const context = sheet.getRange('C1').getValue();

  if (!topic || !recipient) {
    sheet
      .getRange('D1')
      .setValue('Please enter a topic in A1 and recipient name in B1.');
```

B1.');

```

    return;
  }

  const draft = callGemini(
    'Write a polite email to ' +
      recipient +
      ' about: ' +
      topic +
      '.\n\nExtra context (optional):\n' +
      (context || '(none)')
  );

  sheet.getRange('D1').setValue(draft);
}

/**
 * EXAMPLE 7 – Meeting summary from A2:A10 → B1
 */
function sheets_meetingSummaryFromRange() {
  const sheet = getActiveSheetOrError_();
  const values = sheet.getRange('A2:A10').getValues();

  const notes = values
    .map(row => row[0])
    .filter(Boolean)
    .map(line => '- ' + line)
    .join('\n');
```

```

    if (!notes) {
        sheet.getRange('B1').setValue('Add meeting notes in A2:A10
first.');
```

```

        return;
    }

    const summary = callGemini(
        'Summarize these meeting notes into a short paragraph and 3 bullet
points:\n\n' +
        notes
    );

    sheet.getRange('B1').setValue(summary);
}

/**
 * EXAMPLE 8 – Tutor answer from A1 question → B1
 */
function sheets_tutorAnswerFromQuestion() {
    const sheet = getActiveSheetOrError_();
    const question = sheet.getRange('A1').getValue();

    if (!question) {
        sheet
            .getRange('B1')
            .setValue('Enter a question in A1, e.g. "What is a JavaScript
closure?");
        return;
    }

    const answer = callGemini(
        'You are a patient tutor. Answer this question step by step in
simple language:\n\n' +
        question
    );

    sheet.getRange('B1').setValue(answer);
}

```



What each Sheets function does (for your guide)

- GEMINI(prompt) – custom function you can use in any cell.
- sheets_topicIdeas() – A1 topic → A2 text block of ideas.

- `sheets_topicIdeasPerRow()` – A1 topic → A3:A7 one idea per row.
- `sheets_explainActiveFormula()` – explains the active cell formula.
- `sheets_translateAtoB()` – translates column A (A2:A) into B (B2:B).
- `sheets_todosFromNotes()` – turns meeting notes in A1 into TODOs in B2:B.
- `sheets_emailDraftFromInputs()` – A1/B1/C1 → email draft in D1.
- `sheets_meetingSummaryFromRange()` – A2:A10 → summary in B1.
- `sheets_tutorAnswerFromQuestion()` – A1 question → B1 explanation.



5. Add Custom UI Menus

Now, let's make this **usable without opening the script editor**.

A) Docs Menu (`DocsMenu.gs`)

Create a new file in the **Doc-bound** project called **`DocsMenu.gs`**:

```
/**
 * Add a custom "Gemini AI" menu to Google Docs.
 */
function onOpen() {
  const ui = DocumentApp.getUi();
  ui.createMenu('Gemini AI')
    .addItem('Summarize selection', 'docs_summarizeSelection')
    .addItem('Rewrite simpler (Grade 6)', 'docs_rewriteSimpler')
    .addItem('Generate outline from topic',
'docs_generateOutlineFromTopic')
    .addItem('Create quiz from selection', 'docs_quizFromSelection')
    .addItem('Suggest writing comment', 'docs_suggestComment')
    .addToUi();
}

/**
 * Optional: Manual refresh if needed.
 */
function showGeminiMenu() {
  onOpen();
}
```

How to use:

- Reload the Doc (or run `onOpen()` once from the editor).

- You'll see a **"Gemini AI"** menu in the Docs toolbar.
- Highlight text and choose an action (Summarize, Rewrite, Quiz, etc.).

B) Sheets Menu (SheetsMenu.gs)

Create a new file in the **Sheet-bound** project called **SheetsMenu.gs**:

```
/**
 * Add a custom "Gemini AI" menu to Google Sheets.
 */
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')
    .addToUi();
}

/**
 * Optional: Manual refresh if menu doesn't appear immediately.
 */
function showGeminiMenu() {
  onOpen();
}
```

How to use:

- Reload the Sheet (or run `onOpen()` once).
- A **"Gemini AI"** menu appears.
- Fill the relevant cells (e.g., A1/B1/C1) and choose the matching menu item.



6. How Beginners Use the Toolkit (Docs & Sheets)

For Docs users

1. Open a Doc → use the **Gemini AI** menu.
2. Highlight text → pick:
 - Summarize selection
 - Rewrite simpler
 - Create quiz
 - Suggest comment

Or choose “Generate outline from topic” and type a topic in the dialog.

For Sheets users

1. Fill in cells:
 - A1 topics
 - A2:A10 phrases or notes
 - A1/B1/C1 for email, etc.
2. Use the **Gemini AI** menu:
 - Idea generation
 - Translation
 - Meeting summaries
 - TODO extraction
 - Tutor answers
3. Or use =GEMINI("your prompt") directly in a cell.