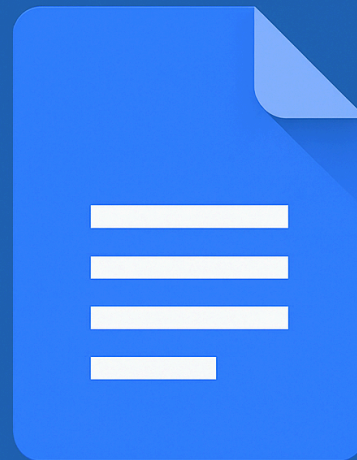


Google Docs Gemini AI Toolkit

+ Setup Script

- Summarize selected text
- Rewrite text at Grade 6 level
- Generate outlines from topics
- Create multiple-choice quizzes
- Provide constructive feedback



Gemini just made Google Docs a whole lot smarter. In this upgrade, you'll unlock **five new AI-powered tools** that supercharge your writing workflow—expanding content, rewriting tone, generating titles, extracting bullet-point insights, and even proofreading your text. With a clean Apps Script file and a custom “Gemini AI” menu, any Google Doc becomes your own intelligent writing assistant.

 Script: setupGeminiDocsDemo()	2
How to use it	7
 5 More Gemini AI Exercises for Google Docs (With Custom Menu)	8
 Prerequisites	8
 Recap: Existing 5 Gemini Functions for Docs	10
 New Exercise 6 — Expand Selected Text With More Detail	13
What this does	13
Code	14
How to use it	14
 New Exercise 7 — Change Tone to More Professional	15
What this does	15
Code	15
How to use it	16
 New Exercise 8 — Generate Title Ideas From Selection	16
What this does	16

Code	16
How to use it	17
🔥 New Exercise 9 — Convert Selection to Key Bullet Points	17
What this does	17
Code	18
How to use it	19
🔥 New Exercise 10 — Proofread & Suggest Improvements	19
What this does	19
Code	19
How to use it	20
🧩 Add / Update the Custom “Gemini AI” Menu	20

Below is a function you can add to your **Docs project** (same one with `SharedHelpers.gs`, `DocsExamples.gs`, `DocsMenu.gs`).

It will:

- Clear the current document
- Insert headings + sample text for each exercise
- Add small instructions about which menu action to try



Script: `setupGeminiDocsDemo()`

```
/**
 * Setup the current Google Doc with sample content
 * for all Gemini AI examples.
 *
 * Run this once in a fresh Doc:
 * - Open Extensions → Apps Script
 * - Add this function to DocsExamples.gs (or a new file)
 * - Run setupGeminiDocsDemo()
 */
function setupGeminiDocsDemo() {
  const doc = DocumentApp.getActiveDocument();
  const body = doc.getBody();

  // Clear existing content
```

```

body.clear();

// Title
body.appendParagraph('Gemini AI – Google Docs Demo')
    .setHeading(DocumentApp.ParagraphHeading.TITLE);

body.appendParagraph(
    'This document is pre-filled with sample content for testing the  

"Gemini AI" menu tools. ' +
    'Select the indicated sections and run the matching Gemini AI  

action from the menu.'
).setSpacingAfter(18);

// Section 1 – Summarize
body.appendParagraph('1. Summarize Selected Text')
    .setHeading(DocumentApp.ParagraphHeading.HEADING1);
body.appendParagraph(
    'Try this with: Gemini AI → "Summarize selection". ' +
    'Highlight the paragraph below first.'
).setItalic(true);

body.appendParagraph(
    'JavaScript is one of the most widely-used programming languages  

in the world, ' +
    'powering both small interactive widgets and large-scale web  

applications. ' +
    'Because it runs in the browser, JavaScript allows developers to  

build ' +
    'interfaces that respond instantly to user actions, without  

needing to reload ' +
    'the entire page from the server.'
).setSpacingAfter(18);

// Section 2 – Rewrite simpler
body.appendParagraph('2. Rewrite Text in Simpler Language')
    .setHeading(DocumentApp.ParagraphHeading.HEADING1);
body.appendParagraph(
    'Try this with: Gemini AI → "Rewrite simpler (Grade 6)". ' +
    'Highlight the paragraph below first.'
).setItalic(true);

body.appendParagraph(
    'The institution seeks to foster interdisciplinary collaboration  

by encouraging ' +

```

```

    'departments to adopt a more holistic approach to curriculum
design. ' +
    'This involves integrating perspectives from multiple fields,
aligning learning ' +
    'outcomes with real-world challenges, and continuously evaluating
the impact on students.'
    ).setSpacingAfter(18);

// Section 3 – Outline from topic
body.appendParagraph('3. Generate Outline From Topic')
    .setHeading(DocumentApp.ParagraphHeading.Heading1);
body.appendParagraph(
    'Try this with: Gemini AI → "Generate outline from topic". ' +
    'When prompted, enter a topic such as "Intro to Google Apps
Script".'
    ).setItalic(true);
body.appendParagraph(
    'Use this section to test how Gemini breaks a topic into sections
and subpoints. ' +
    'You can repeat the command with different topics to compare
outlines.'
    ).setSpacingAfter(18);

// Section 4 – Quiz from selection
body.appendParagraph('4. Create Quiz From Selected Text')
    .setHeading(DocumentApp.ParagraphHeading.Heading1);
body.appendParagraph(
    'Try this with: Gemini AI → "Create quiz from selection". ' +
    'Highlight the paragraph below first.'
    ).setItalic(true);

body.appendParagraph(
    'Cloud computing allows organizations to access computing
resources—such as servers, ' +
    'storage, databases, and software—over the internet instead of
maintaining physical ' +
    'hardware on-site. This model can reduce upfront costs, make it
easier to scale up ' +
    'or down based on demand, and enable teams to collaborate from
different locations.'
    ).setSpacingAfter(18);

// Section 5 – Suggest comment
body.appendParagraph('5. Suggest Writing Comment')

```

```

        .setHeading(DocumentApp.ParagraphHeading.Heading1);
body.appendParagraph(
    'Try this with: Gemini AI → "Suggest writing comment". ' +
    'Highlight the paragraph below first.'
).setItalic(true);

body.appendParagraph(
    'This paragraph is kind of messy and maybe a bit too casual for a
formal document, ' +
    'because it mixes different ideas without much structure and
doesn\'t really explain ' +
    'why any of them are important, which might confuse some readers.'
).setSpacingAfter(18);

// Section 6 – Expand selection
body.appendParagraph('6. Expand Selection With More Detail')
    .setHeading(DocumentApp.ParagraphHeading.Heading1);
body.appendParagraph(
    'Try this with: Gemini AI → "Expand selection (more detail)". ' +
    'Highlight the paragraph below first.'
).setItalic(true);

body.appendParagraph(
    'Regular practice is one of the most effective ways to learn
programming. ' +
    'Short, focused sessions can be more helpful than long, infrequent
study marathons.'
).setSpacingAfter(18);

// Section 7 – Professional tone
body.appendParagraph('7. Rewrite in Professional Tone')
    .setHeading(DocumentApp.ParagraphHeading.Heading1);
body.appendParagraph(
    'Try this with: Gemini AI → "Rewrite in professional tone". ' +
    'Highlight the paragraph below first.'
).setItalic(true);

body.appendParagraph(
    'Hey, I just threw this together really fast, so it\'s not super
polished. ' +
    'We can clean it up later if we need to, but for now it should be
fine.'
).setSpacingAfter(18);

```

```

// Section 8 – Title ideas
body.appendParagraph('8. Generate Title Ideas')
  .setHeading(DocumentApp.ParagraphHeading.Heading1);
body.appendParagraph(
  'Try this with: Gemini AI → "Generate title ideas". ' +
  'Highlight the paragraph below first.'
).setItalic(true);

body.appendParagraph(
  'This article explains how teams can use automation and AI tools
to streamline ' +
  'repetitive tasks, reduce errors, and free up time for more
creative and strategic work.'
).setSpacingAfter(18);

// Section 9 – Bullet takeaways
body.appendParagraph('9. Key Bullet Takeaways From Selection')
  .setHeading(DocumentApp.ParagraphHeading.Heading1);
body.appendParagraph(
  'Try this with: Gemini AI → "Key bullet takeaways from selection".
' +
  'Highlight the paragraph below first.'
).setItalic(true);

body.appendParagraph(
  'When planning a course, it is important to define clear learning
outcomes, align ' +
  'assessments with those outcomes, and provide opportunities for
students to practice ' +
  'new skills. Regular feedback, accessible resources, and varied
teaching strategies ' +
  'can significantly improve student engagement and success.'
).setSpacingAfter(18);

// Section 10 – Proofreading
body.appendParagraph('10. Proofread & Suggest Improvements')
  .setHeading(DocumentApp.ParagraphHeading.Heading1);
body.appendParagraph(
  'Try this with: Gemini AI → "Proofread & suggest improvements". '
+
  'Highlight the paragraph below first.'
).setItalic(true);

body.appendParagraph(

```

```

    'This sentence have a few problem and it doesn\'t really flow that
    well, ' +
    'because the grammar are off and the ideas isn\'t totally clear,
    so it kind of ' +
    'makes the reader stop and think what is actually being said.'
  ).setSpacingAfter(24);

  // Footer note
  body.appendParagraph(
    'Tip: Experiment with different texts in each section to see how
    Gemini responds. ' +
    'You can overwrite these samples with your own real content.'
  ).setItalic(true);
}

```

How to use it

1. Open the Google Doc you want to use as your **demo document**.
2. Go to **Extensions** → **Apps Script**.
3. Paste this function into your existing project (e.g., in `DocsExamples.gs` or a new file).
4. Click **Run** → **setupGeminiDocsDemo** and authorize if prompted.
5. Return to the Doc: you'll see the demo content ready to test with your **Gemini AI** menu.

5 More Gemini AI Exercises for Google Docs (With Custom Menu)

In a previous toolkit, we built 5 Gemini-powered tools for Google Docs:

1. Summarize selected text
2. Rewrite text in simpler language
3. Generate an outline from a topic
4. Create multiple-choice quizzes
5. Suggest constructive comments

In this guide, we'll add **5 more useful AI actions**:

6. Expand selected text with more detail
7. Change the tone to “more professional”
8. Generate catchy title ideas
9. Turn selected text into bullet-point key takeaways
10. Proofread selected text and suggest improvements

You'll also get a **custom “Gemini AI” menu** in Google Docs so you can run everything with just a click.

Prerequisites

- A Google Doc (any document)
- A Gemini API key
- A bound Apps Script project
 - In your Doc: **Extensions** → **Apps Script**

You should already have a `SharedHelpers.gs` file with something like:

```
const GEMINI_MODEL_ID = 'gemini-2.5-flash';
```



```

function getGeminiApiKey_() {
  const props = PropertiesService.getScriptProperties();
  const key = props.getProperty('GEMINI_API_KEY');
  if (!key) throw new Error('GEMINI_API_KEY is not set in Script
Properties.');
```

```

  return key;
}

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)
  });

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

```

And in **Project Settings** → **Script properties** you've added:

- Key: GEMINI_API_KEY
- Value: your Gemini key

Recap: Existing 5 Gemini Functions for Docs

If you already have these in `DocsExamples.gs`, you can skip this section.
If you want a single complete file, keep them in.

```
/**
 * 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);
}
```

New Exercise 6 — Expand Selected Text With More Detail

What this does

You select a paragraph → Gemini **expands it** with more explanation, examples, or detail. Great for:

- Turning notes into fuller paragraphs
- Expanding a rough idea into a better draft
- Adding context for readers

Code

```
function docs_expandSelectionDetail() {
  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 expanded = callGemini(
    'Expand the following text with more detail, explanations, and one
or two examples. ' +
    'Keep it clear and easy to read:\n\n' +
    text
  );

  DocumentApp.getUi().alert(expanded);
}
```

How to use it

1. Highlight a short paragraph or bullet list.

2. In the menu (we'll add it soon): **Gemini AI** → **Expand selection (more detail)**
 3. A dialog shows the expanded version – copy/paste it where you want.
-

New Exercise 7 — Change Tone to More Professional

What this does

Takes your selected text and **rephrases it in a more professional tone**, while keeping the meaning.

Good for:

- Emails
- Reports
- Proposals
- Course announcements

Code

```
function docs_rewriteProfessionalTone() {
  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 rewritten = callGemini(
      'Rewrite the following text in a more professional, polite tone. '
+      'Keep the meaning the same:\n\n' +
      text
    );

    DocumentApp.getUi().alert(rewritten);
  }

```

How to use it

1. Select your informal or rough text (e.g., a quick email draft).
2. Run **Gemini AI** → **Rewrite in professional tone**.
3. Paste the improved version back into the Doc.

New Exercise 8 — Generate Title Ideas From Selection

What this does

Reads your selected text and suggests **several possible titles**.

Useful for:

- Blog posts
- Lessons / modules
- Reports
- Slide decks

Code

```

function docs_generateTitleIdeas() {
  const doc = DocumentApp.getActiveDocument();
  const sel = doc.getSelection();

```



```

if (!sel) {
  DocumentApp.getUi().alert(
    'Select some text first (e.g., an intro or main section).';
  );
  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 ideas = callGemini(
  'Based on the text below, suggest 5 short, catchy title ideas. ' +
  'Each title should be on its own line:\n\n' +
  text
);

DocumentApp.getUi().alert('Suggested titles:\n\n' + ideas);
}

```

How to use it

1. Select a paragraph that represents your content (intro or summary).
2. Run **Gemini AI** → **Generate title ideas**.
3. Choose your favourite title from the list.

New Exercise 9 — Convert Selection to Key Bullet Points

What this does

Get more Apps Script Content at <https://basescripts.com/> by Laurence Svekis

Turns a chunk of text into **key bullet-point takeaways**.

Great for:

- Study notes
- Slide preparation
- Executive summaries
- Lesson highlights

Code

```
function docs_bulletSummaryFromSelection() {
  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 bullets = callGemini(
    'Read the text below and extract 5-10 key bullet-point takeaways.
  ' +
    'Return them as a simple bullet list:\n\n' +
    text
  );

  const body = doc.getBody();
  body.appendParagraph('\nKey Takeaways:').setBold(true);
  body.appendParagraph(bullets);
}
```

}

How to use it

1. Select a section of text.
2. Run **Gemini AI** → **Key bullet takeaways**.
3. The bullet list is **inserted into your document** under the heading “Key Takeaways”.

New Exercise 10 — Proofread & Suggest Improvements

What this does

Reads your selected text and returns **a list of issues + suggestions**.

It doesn't blindly rewrite everything. Instead, it behaves more like a reviewer.

Code

```
function docs_proofreadSelection() {
  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 feedback = callGemini(
  'You are a careful editor. Read the text below and list any
  grammar, clarity, or tone issues. ' +
  'For each issue, suggest an improved version. Format your answer
  as a numbered list:\n\n' +
  text
);

DocumentApp.getUi().alert('Proofreading suggestions:\n\n' +
feedback);
}

```

How to use it

1. Highlight a paragraph or two.
2. Run **Gemini AI** → **Proofread & suggest improvements**.
3. Use the suggestions to fix your text.

Add / Update the Custom “Gemini AI” Menu

To make all 10 functions easy to use, add this `onOpen()` function in a file like `DocsMenu.gs` (in the same project):

```

function onOpen() {
  const ui = DocumentApp.getUi();
  ui.createMenu('Gemini AI')
    // Original 5
    .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')
    .addSeparator()
    // New 5
    .addItem('Expand selection (more detail)',
'docs_expandSelectionDetail')

```

```
.addItem('Rewrite in professional tone',  
'docs_rewriteProfessionalTone')  
.addItem('Generate title ideas', 'docs_generateTitleIdeas')  
.addItem('Key bullet takeaways from selection',  
'docs_bulletSummaryFromSelection')  
.addItem('Proofread & suggest improvements',  
'docs_proofreadSelection')  
.addToUi();  
}
```

If the menu doesn't appear immediately, either:

- Reload the Doc, or
- Run `onOpen()` once from the Apps Script editor.