

Apps Script + Gemini Mastery — Issue #2

 Apps Script + Gemini Mastery — Issue #2	1
AI Content Generator for Google Docs	1
 What You Will Build in Issue #2	1
 Learning Objectives	2
 EXERCISE — Build the AI Content Generator	2
STEP 1 — Add New Menu Items	2
STEP 2 — Create the Sidebar UI for Content Generation	2
STEP 3 — Add a Function to Load the Sidebar	3
STEP 4 — Add the Content Generation Logic	3
STEP 5 — Add a Utility Function to Insert AI Output into the Doc	4
 TESTING	5
 WHAT YOU BUILT IN ISSUE #2	5
 Coming Next: Issue #3 – AI Data Analyst for Google Sheets	5
 Project Structure	6
① Code.gs	6
② GeminiHelpers.gs  This is the helper you were missing	7
③ Analyzer.gs (Issue #1 logic)	9
④ InsertHelpers.gs (shared for Issue #2 and future)	10
⑤ Issue #2 logic – add to GeminiHelpers.gs or a new ContentGenerator.gs	11
⑥ SetupLessonDoc.gs (optional, from earlier)	12
⑦ Sidebar.html (Issue #1 UI)	15
⑧ ContentSidebar.html (Issue #2 UI)	15

<https://github.com/lsvekis/apps-script-gemini-repo>

AI Content Generator for Google Docs

Write blog posts, lesson plans, summaries, outlines & formatted documents using Gemini — all from a custom Apps Script menu.

What You Will Build in Issue #2

A full **AI Content Generator** inside Google Docs with menu items like:

AI Tools

- Generate Blog Post
- Generate Lesson Outline
- Generate Summary
- Expand Highlights into Full Content

Each command opens a sidebar with a simple form where the user enters:

- A topic
- A style (optional)
- A length (optional)

Gemini generates formatted content directly inside the document.

This is a **writer's copilot** for Google Workspace — perfect for educators, creators, content teams, and Docs power users.

Learning Objectives

By completing this issue, you will learn:

- ✓ How to build **HTML forms** in a sidebar
- ✓ How to pass user input to Apps Script
- ✓ How to generate structured prompts for Gemini
- ✓ How to insert AI output directly into the Doc
- ✓ How to extend Apps Script menus with more powerful AI functionality
- ✓ How to build an AI-powered writing assistant

EXERCISE — Build the AI Content Generator

STEP 1 — Add New Menu Items

Add these to your existing `onOpen()` function:

```
function onOpen() {  
  DocumentApp.getUi()  
    .createMenu("AI Tools")  
    .addItem("Analyze Selected Text", "showAnalyzeSidebar")  
    .addItem("Summarize Document", "summarizeDocument")  
    .addItem("Rewrite for Clarity", "rewriteForClarity")  
    .addSeparator()  
    .addItem("AI Content Generator", "showContentGenerator")  
    .addToUi();  
}
```

STEP 2 — Create the Sidebar UI for Content Generation

Create a new file: **ContentSidebar.html**

```
<div style="font-family: Arial; padding: 14px;">  
  <h2>AI Content Generator</h2>  
  
  <label>Topic</label><br/>  
  <input id="topic" style="width: 100%; margin-bottom: 10px;" />
```

```

    <label>Style (optional, e.g., academic, friendly,
professional)</label><br/>
    <input id="style" style="width: 100%; margin-bottom: 10px;" />

    <label>Length (optional, e.g., 200 words, 2 paragraphs)</label><br/>
    <input id="length" style="width: 100%; margin-bottom: 10px;" />

    <button onclick="generate()" style="margin-top:
10px;">Generate</button>

    <pre id="output" style="white-space: pre-wrap; margin-top:
20px;"></pre>

    <script>
    function generate() {
        const topic = document.getElementById("topic").value;
        const style = document.getElementById("style").value;
        const length = document.getElementById("length").value;

        google.script.run.withSuccessHandler(function(result) {
            document.getElementById("output").textContent =
result.message;
            if (result.inserted) {
                document.getElementById("output").textContent += "\n\n✓
Inserted into document.";
            }
        }).generateContent(topic, style, length);
    }
    </script>
</div>

```

STEP 3 — Add a Function to Load the Sidebar

In Code.gs or a UI file:

```

function showContentGenerator() {
    const html = HtmlService.createHtmlOutputFromFile("ContentSidebar")
        .setTitle("AI Content Generator");
    DocumentApp.getUi().showSidebar(html);
}

```

STEP 4 — Add the Content Generation Logic

Add to `GeminiHelpers.gs` or a new file:

```

function generateContent(topic, style, length) {
  if (!topic || topic.trim() === "")
    return { message: "Please enter a topic.", inserted: false };

  const prompt = `
Write content based on the topic below.

Topic: ${topic}

Requirements:
- Style: ${style || "default writing style"}
- Length: ${length || "medium length"}
- Format using headings, paragraphs, and bullet points if appropriate.
- Be clear, helpful, and well-structured.

Begin:
`;

  const response = callGemini(prompt, "");
  insertTextIntoDocument(response);

  return {
    message: "AI content generated successfully.",
    inserted: true
  };
}

```

STEP 5 — Add a Utility Function to Insert AI Output into the Doc

Create a new file: **InsertHelpers.gs**

```

function insertTextIntoDocument(text) {
  const body = DocumentApp.getActiveDocument().getBody();

  body.appendParagraph("");
  body.appendParagraph("=== AI Generated Content ===")
    .setHeading(DocumentApp.ParagraphHeading.HEADING2);

  const paragraphs = text.split("\n");

  paragraphs.forEach(line => {
    if (line.trim() === "") {
      body.appendParagraph("");
    } else {

```

```
        body.appendParagraph(line);  
    }  
    });  
}
```



TESTING

1. Reload the Google Doc
2. Open **AI Tools** → **AI Content Generator**
3. Enter:
 - Topic: *The benefits of AI in education*
 - Style: *friendly and professional*
 - Length: *300 words*
4. Click **Generate**
5. Watch the output appear in your Doc under "=== **AI Generated Content** ==="



WHAT YOU BUILT IN ISSUE #2

You now have a working **AI Content Generator** that:

- ✓ Accepts user input
- ✓ Builds a structured prompt
- ✓ Sends it to Gemini
- ✓ Inserts formatted content directly into the document
- ✓ Provides a reusable framework for future content tools

This framework will be used again in Issue #3 and Issue #4 to build:

- AI-powered Sheets tools
- RAG systems
- Automated workflows
- AI assistants inside Workspace



Coming Next: Issue #3 – AI Data Analyst for Google Sheets

Ask Gemini questions about selected data ranges and get:

- Summaries

- Anomalies
- Trends
- Pivot suggestions
- Chart recommendations
- Plain-English explanations

Full code, UI, prompts, and setup included.

Below is a complete Apps Script setup for:

-  **Issue #1** – AI Document Analyzer (sidebar)
-  **Issue #2** – AI Content Generator (sidebar + inserts into Doc)
-  Includes the **Gemini helper file** and insert helpers

Project Structure

Create these files in your Apps Script project:

- `Code.gs`
- `GeminiHelpers.gs`
- `Analyzer.gs`
- `InsertHelpers.gs`
- *(optional)* `SetupLessonDoc.gs`
- `Sidebar.html`
- `ContentSidebar.html`

Code.gs

```
function onOpen() {
  DocumentApp.getUi()
    .createMenu("AI Tools")
    .addItem("Analyze Selected Text", "showAnalyzeSidebar")
    .addItem("Summarize Document", "summarizeDocument")
    .addItem("Rewrite for Clarity", "rewriteForClarity")
    .addSeparator()
    .addItem("AI Content Generator", "showContentGenerator")
}
```

```

        .addSeparator()
        .addItem("Setup Lesson Doc (Issue #1)", "setupLessonDoc") //
optional
        .addUi();
    }

// Sidebar for Issue #1 (AI Document Analyzer)
function showAnalyzeSidebar() {
    const html = HtmlService.createHtmlOutputFromFile("Sidebar")
        .setTitle("AI Text Analyzer");
    DocumentApp.getUi().showSidebar(html);
}

// Sidebar for Issue #2 (AI Content Generator)
function showContentGenerator() {
    const html = HtmlService.createHtmlOutputFromFile("ContentSidebar")
        .setTitle("AI Content Generator");
    DocumentApp.getUi().showSidebar(html);
}

```

2 GeminiHelpers.gs *This is the helper you were missing*

```

//  Replace with your real Gemini API key
const GEMINI_API_KEY = "YOUR_API_KEY_HERE";
const GEMINI_MODEL    = "gemini-1.5-flash"; // works with v1
generateContent

/**
 * Generic helper to call Gemini's generateContent endpoint.
 * @param {string} prompt - Instruction for the model
 * @param {string} text    - Optional content/body to include
 * @returns {string}      - Model response or error message
 */
function callGemini(prompt, text) {
    const url =
`https://generativelanguage.googleapis.com/v1/models/${GEMINI_MODEL}:g
enerateContent?key=${GEMINI_API_KEY}`;

    const payload = {
        contents: [{
            parts: [{
                text: `${prompt}${text ? "\n\n--\n\n" + text : ""}`
            }]
        }]
    }
}

```

```

    }]
  };

const options = {
  method: "post",
  contentType: "application/json",
  payload: JSON.stringify(payload),
  muteHttpExceptions: true
};

try {
  const res = UrlFetchApp.fetch(url, options);
  const status = res.getResponseCode();
  const body = res.getContentText();

  Logger.log("Gemini status: " + status);
  Logger.log("Gemini body: " + body);

  const json = JSON.parse(body);

  // Successful candidate
  if (json.candidates &&
      json.candidates[0] &&
      json.candidates[0].content &&
      json.candidates[0].content.parts &&
      json.candidates[0].content.parts[0].text) {
    return json.candidates[0].content.parts[0].text;
  }

  // API returned an error
  if (json.error) {
    return "Gemini API error: " + json.error.message +
      (json.error.status ? " (" + json.error.status + ")" :
  "");
  }

  // Unexpected shape
  return "Unexpected Gemini response:\n" + body;

} catch (e) {
  Logger.log("Exception calling Gemini: " + e);
  return "Exception calling Gemini: " + e;
}
}

```

```

/**
 * Quick test helper – run from Script Editor to validate setup.
 */
function testGemini() {
  const result = callGemini(
    "Summarize this in one sentence:",
    "Google Apps Script lets you automate tasks across Google
Workspace."
  );
  Logger.log("Test result:\n" + result);
}

```

3 Analyzer.gs (Issue #1 logic)

```

/**
 * Called from Sidebar.html → analyzeSelectedText button
 * Returns analysis string to display in the sidebar.
 */
function analyzeSelectedText() {
  const doc = DocumentApp.getActiveDocument();
  const selection = doc.getSelection();

  if (!selection) return "No text selected.";

  let text = "";

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

  const prompt = `
You are an expert writing assistant.

Analyze the following text and provide:

1. Key insights (bullet points)
2. A concise summary (3–5 bullet points)
3. Suggested improvements
4. Overall sentiment
5. A clearer rewrite

```

```

Respond in a clean, readable format.
`;

    return callGemini(prompt, text);
}

/**
 * Summarize the entire document. Called from Sidebar.html.
 */
function summarizeDocument() {
    const bodyText =
DocumentApp.getActiveDocument().getBody().getText();
    const prompt = "Summarize the following document in 5-7 bullet
points:";
    return callGemini(prompt, bodyText);
}

/**
 * Rewrite the entire document for clarity. Called from Sidebar.html.
 */
function rewriteForClarity() {
    const bodyText =
DocumentApp.getActiveDocument().getBody().getText();
    const prompt = `
Rewrite the document for clarity, conciseness, and readability.
Preserve the original meaning but improve structure and flow.
`;
    return callGemini(prompt, bodyText);
}

```

4 InsertHelpers.gs (shared for Issue #2 and future)

```

/**
 * Append AI output to the end of the document,
 * with a heading marker above it.
 */
function insertTextIntoDocument(text) {
    const body = DocumentApp.getActiveDocument().getBody();

    body.appendParagraph("");
    body.appendParagraph("=== AI Generated Content ===")
        .setHeading(DocumentApp.ParagraphHeading.HEADING2);
}

```

```

const paragraphs = text.split("\n");

paragraphs.forEach(line => {
  if (line.trim() === "") {
    body.appendParagraph("");
  } else {
    body.appendParagraph(line);
  }
});
}

```

5 Issue #2 logic – add to **GeminiHelpers.gs** or a new **ContentGenerator.gs**

```

/**
 * Generate AI content based on topic, style, and length from the
 * sidebar.
 * Returns an object consumed by ContentSidebar.html.
 */
function generateContent(topic, style, length) {
  if (!topic || topic.trim() === "") {
    return { message: "Please enter a topic.", inserted: false };
  }

  const prompt = `
You are an expert content writer.

Write content based on the topic below.

Topic: ${topic}

Requirements:
- Style: ${style || "neutral, informative, and clear"}
- Length: ${length || "medium length"}
- Use headings, paragraphs, and bullet points when appropriate.
- Be structured, helpful, and easy to scan.

Begin the final content directly.
`;

  const response = callGemini(prompt, "");
  insertTextIntoDocument(response);

```

```
    return {  
      message: "AI content generated successfully.",  
      inserted: true  
    };  
  }  
}
```

6 SetupLessonDoc.gs (optional, from earlier)

```
function setupLessonDoc() {  
  const doc = DocumentApp.getActiveDocument();  
  const body = doc.getBody();  
  
  // Clear existing content  
  body.clear();  
  
  // Title  
  body.appendParagraph("Apps Script + Gemini Mastery – Issue #1")  
    .setHeading(DocumentApp.ParagraphHeading.HEADING1);  
  
  body.appendParagraph("Your First AI-Powered Script: Build a Smart  
Document Analyzer")  
    .setHeading(DocumentApp.ParagraphHeading.HEADING2);  
  
  body.appendParagraph(  
    "This document was generated by the setup script. Use it as your  
working file " +  
    "to test the AI Tools menu connected to Gemini."  
  );  
  
  body.appendParagraph("");  
  
  // Section: What You Will Build  
  body.appendParagraph("What You Will Build")  
    .setHeading(DocumentApp.ParagraphHeading.HEADING2);  
  
  body.appendParagraph(  
    "In this lesson, you will build an AI-powered Document Analyzer  
for Google Docs. " +  
    "You will be able to select text and then run AI Tools → Analyze  
Selected Text " +  
    "to get insights, summaries, and rewrite suggestions from Gemini."  
  );  
  
  // Section: Instructions
```

```

body.appendParagraph("How to Use This Doc with the AI Tools Menu")
    .setHeading(DocumentApp.ParagraphHeading.Heading2);

body.appendListItem("Select any paragraph or section of text
below.")
    .setGlyphType(DocumentApp.GlyphType.BULLET);
body.appendListItem("Open the menu: AI Tools → Analyze Selected
Text.")
    .setGlyphType(DocumentApp.GlyphType.BULLET);
body.appendListItem("Review the output in the sidebar.")
    .setGlyphType(DocumentApp.GlyphType.BULLET);
body.appendListItem("Experiment with Summarize Document and Rewrite
for Clarity.")
    .setGlyphType(DocumentApp.GlyphType.BULLET);

body.appendParagraph("");

// Sample content
body.appendParagraph("Sample Article for AI Analysis")
    .setHeading(DocumentApp.ParagraphHeading.Heading2);

const sampleParagraphs = [
    "Most teams struggle to turn raw information into clear, usable
knowledge. " +
    "Documents grow longer, feedback gets buried in email threads,
and important " +
    "decisions are scattered across chats and comments.",

    "AI-powered tools like Gemini can help by quickly summarizing long
documents, " +
    "highlighting key decisions, and suggesting better ways to
express ideas. " +
    "Instead of manually rewriting paragraphs, you can focus on
thinking and " +
    "strategy while the AI handles structure and phrasing.",

    "However, AI only becomes truly powerful when it is integrated
into your existing workflow. " +
    "That is why Google Apps Script is such a valuable layer: it
lets you bring Gemini " +
    "directly into Docs, Sheets, Slides, and Gmail, automating the
tasks you already do " +
    "every day.",

```

```

    "In this lesson, you will experiment with an AI Document Analyzer
    that you can customize. " +
    "Try selecting different sections of this article and see how
    the AI changes its " +
    "summary, insights, and suggestions based on what you
    highlight."
  ];

  sampleParagraphs.forEach(text => body.appendParagraph(text));

  body.appendParagraph("");

  body.appendParagraph("Longer Sample Section")
    .setHeading(DocumentApp.ParagraphHeading.HEADING3);

  body.appendParagraph(
    "Imagine you are responsible for reviewing weekly project reports
    from multiple " +
    "teams. Each report is several pages long, combining metrics,
    narrative updates, " +
    "risks, and next steps. Reading every word is unrealistic, but
    skipping sections " +
    "means missing important context. With an embedded AI assistant,
    you can highlight " +
    "a section and instantly ask for a concise summary, a list of
    risks, or a clearer " +
    "rewrite that you can paste into your own executive update."
  );

  body.appendParagraph(
    "This is the type of workflow enhancement that Apps Script +
    Gemini makes possible. " +
    "As you test the AI Tools menu on this document, think about the
    documents you " +
    "work with every day: course outlines, project charters, meeting
    notes, or client " +
    "proposals. Any place where you read, summarize, and rewrite text
    is an opportunity " +
    "to add intelligence with just a few lines of code."
  );

  body.appendParagraph("");
  body.appendParagraph("✅ Setup complete. You can now experiment with
  the AI Tools menu on this content.");

```

```
}
```

7 Sidebar.html (Issue #1 UI)

```
<div style="font-family: Arial; padding: 10px;">
  <h2>AI Document Analyzer</h2>

  <button onclick="run('analyzeSelectedText')">Analyze Selected
Text</button>
  <button onclick="run('summarizeDocument')" style="margin-left:
4px;">Summarize Document</button>
  <button onclick="run('rewriteForClarity')" style="margin-left:
4px;">Rewrite for Clarity</button>

  <pre id="output" style="white-space: pre-wrap; margin-top:
20px;"></pre>

  <script>
    function run(fn) {
      document.getElementById("output").textContent = "Running " + fn
+ "...";
      google.script.run
        .withSuccessHandler(function(result) {
          document.getElementById("output").textContent = result;
        })
        [fn]();
    }
  </script>
</div>
```

8 ContentSidebar.html (Issue #2 UI)

```
<div style="font-family: Arial; padding: 14px;">
  <h2>AI Content Generator</h2>

  <label>Topic</label><br/>
  <input id="topic" style="width: 100%; margin-bottom: 10px;" />

  <label>Style (optional, e.g., academic, friendly,
professional)</label><br/>
  <input id="style" style="width: 100%; margin-bottom: 10px;" />

  <label>Length (optional, e.g., 200 words, 2 paragraphs)</label><br/>
  <input id="length" style="width: 100%; margin-bottom: 10px;" />
```

```
<button onclick="generate()" style="margin-top:
10px;">Generate</button>

<pre id="output" style="white-space: pre-wrap; margin-top:
20px;"></pre>

<script>
  function generate() {
    const topic = document.getElementById("topic").value;
    const style = document.getElementById("style").value;
    const length = document.getElementById("length").value;

    document.getElementById("output").textContent = "Generating
content...";

    google.script.run
      .withSuccessHandler(function(result) {
        document.getElementById("output").textContent =
result.message +
        (result.inserted ? "\n\n✓ Inserted into document." : "");
      })
      .generateContent(topic, style, length);
  }
</script>
</div>
```