**Task 2:**

BUILD AN AI-POWERED WEBSITE USING A NO-CODE TOOL \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Internship Task Report**

**Submitted by: Kartik Chabhadiya**

**Internship at: Future Intern**

**August 2025**

## Abstract

* This project automates the process of generating and publishing SEO-optimized blog posts to WordPress using Google Apps Script, Gemini API, and Google Sheets. The automation fetches blog titles from a Google Sheet, generates full HTML blog content, and posts it directly to WordPress via the Post-by-Email feature.

## Introduction

* Manually writing, formatting, and publishing blog posts can be time-consuming. This task addresses the challenge by automating blog content creation and publishing. It uses Gemini API for content generation, Google Apps Script for automation, and Google Sheets as a control panel to manage blog titles and status.

## Tools & Technologies

|  |  |
| --- | --- |
| Tool | Purpose |
| Google Apps Script | Automation and integration between Google Sheets and APIs |
| Gemini API | AI-powered blog content generation |
| Google Sheets | Storage of blog titles, statuses, and generated content |
| WordPress (Post by Email) | Automatic publishing of generated blogs |

## Methodology

* The following steps were taken to complete the task:

1. Created a Google Sheet with columns:Title, Status, and Content.

2. Developed a Google Apps Script to read pending titles from the sheet.

3 . Connected the script to Gemini API for generating HTML blog content.

4. Used MailApp service to send the generated blog to WordPress via Post-by-Email.

5. Updated the sheet status to 'Published' after posting.

6. Ensured professional formatting in generated content.

## Code Implementation

* This Script is written in Java Script.

function postBlogsToWordPress() {

  var sheet = SpreadsheetApp.getActiveSpreadsheet().getActiveSheet();

  var data = sheet.getDataRange().getValues();

  var GEMINI\_API\_KEY = "YOUR\_API\_KYE"; // Replace with your Gemini API key

  var WP\_EMAIL = "YOUR\_EMAIL\_GENRATED\_BY\_WORDPRESS"; // Replace with your WordPress Post-by-Email address

  for (var i = 1; i < data.length; i++) {

    var title = data[i][0];

    var status = data[i][1];

    if (status.toLowerCase() === "pending") {

      var content = generateBlogContentGemini(GEMINI\_API\_KEY, title);

      // Send blog to WordPress

      MailApp.sendEmail({

        to: WP\_EMAIL,

        subject: title,

        htmlBody: content // Send HTML for proper formatting

      });

      // Update sheet

      sheet.getRange(i + 1, 2).setValue("Published");

      sheet.getRange(i + 1, 3).setValue(content);

      Utilities.sleep(3000); // Prevent rate limiting

      break; // Process one blog per run

    }

  }

}

function generateBlogContentGemini(apiKey, title) {

  var prompt = `

Write a 1000-word SEO-optimized blog post for the title: "${title}".

Use only clean HTML formatting (no emojis, no hashtags, no unnecessary special characters).

Use <h2> for main sections and <h3> for sub-sections.

Include relevant links for tools, websites, or references if needed.

Do not insert images, logos, or large banners.

Provide a meta description at the top in a <p><strong>Meta Description:</strong> ...</p> format.

Ensure the style looks professional and ready for a blog.

`;

  var url = "https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash:generateContent?key=" + apiKey;

  var payload = {

    contents: [

      { parts: [{ text: prompt }] }

    ]

  };

  var options = {

    method: "post",

    contentType: "application/json",

    payload: JSON.stringify(payload),

    muteHttpExceptions: true

  };

  var response = UrlFetchApp.fetch(url, options);

  var json = JSON.parse(response.getContentText());

  return json.candidates && json.candidates.length > 0

    ? json.candidates[0].content.parts[0].text.trim()

    : "<p>Error: No content generated.</p>";

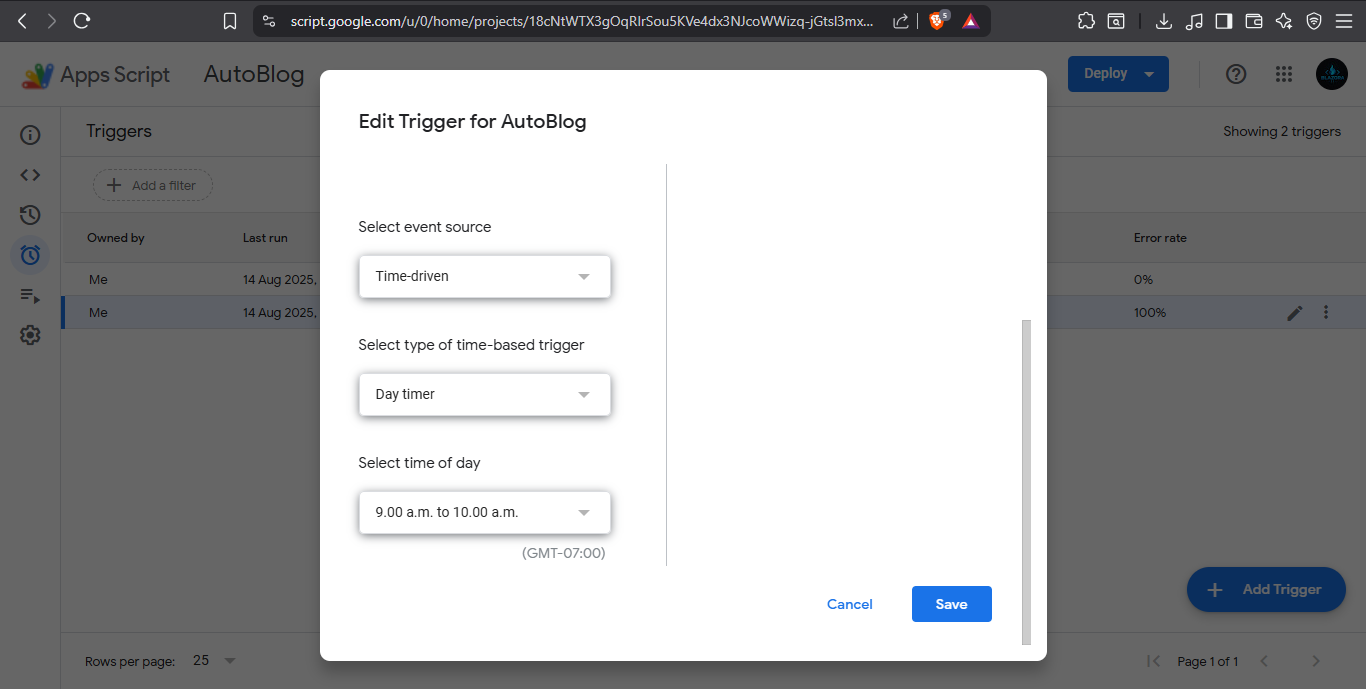
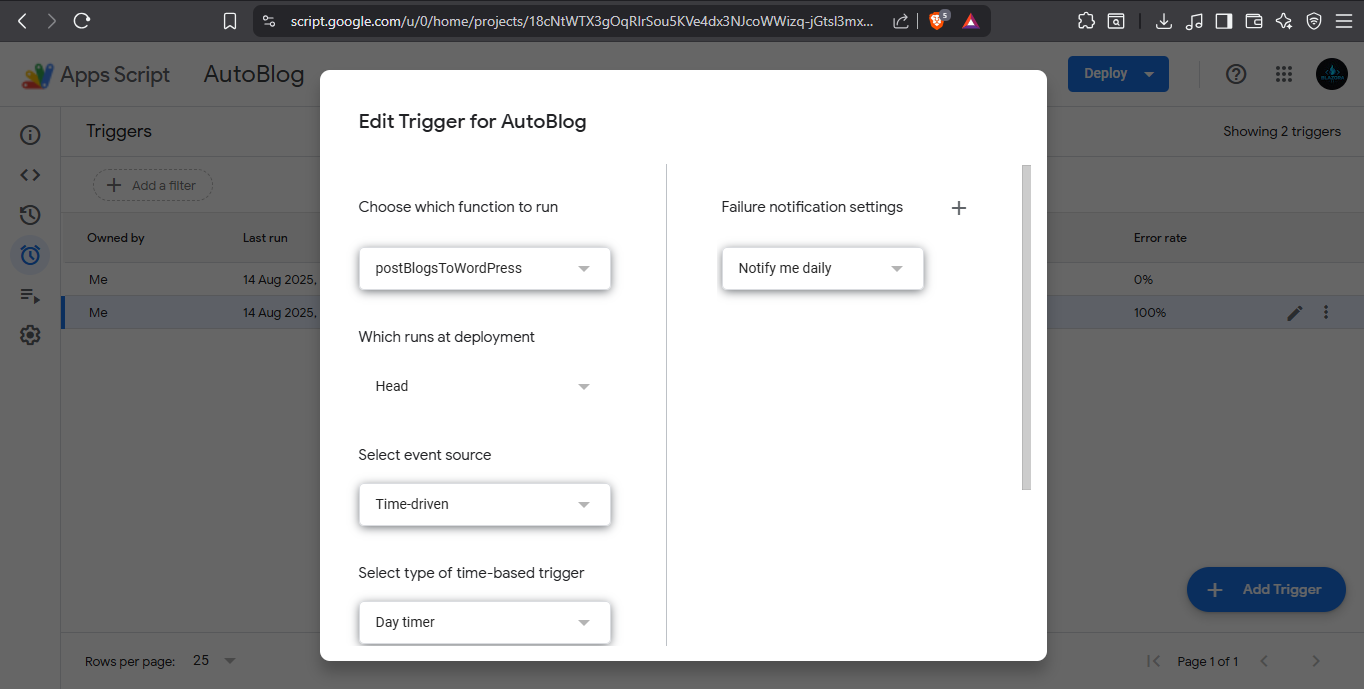
}

## Output

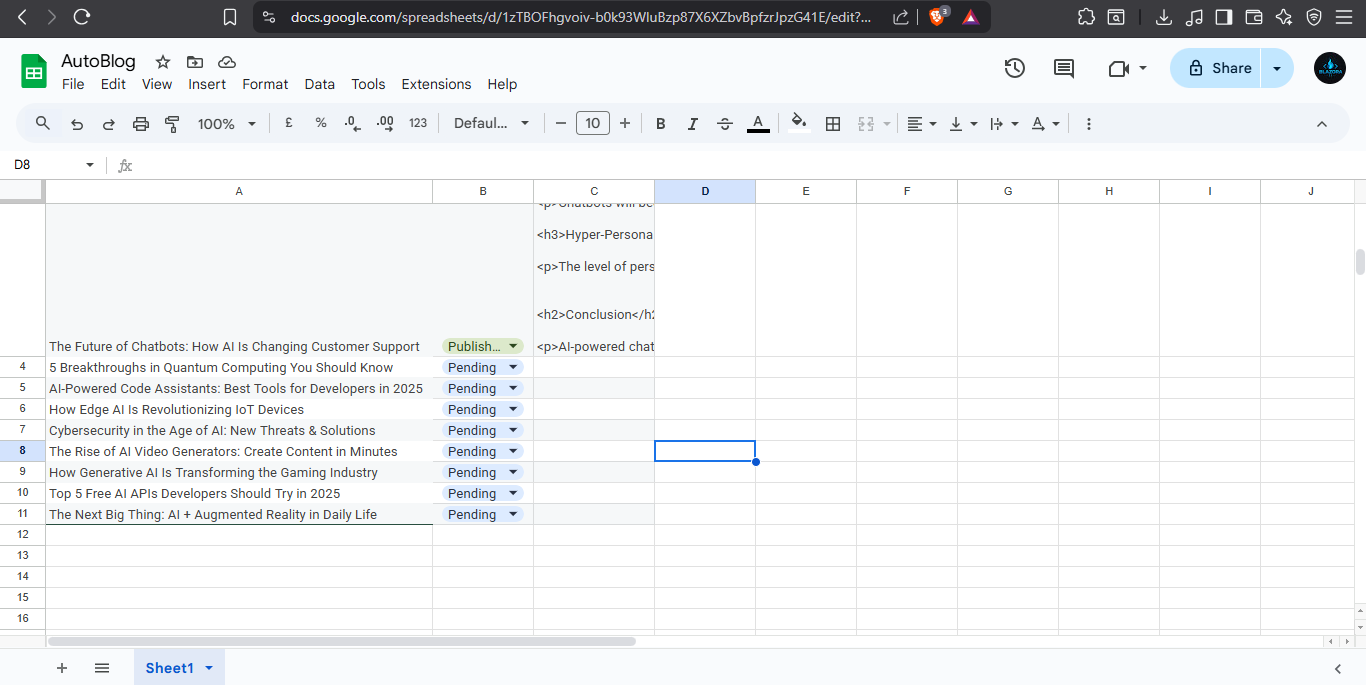
* The following screenshots show the system in action: - Google Sheet before and after running the script - Apps Script editor with code - Word Press blog post published automatically

Steps To Get Output :

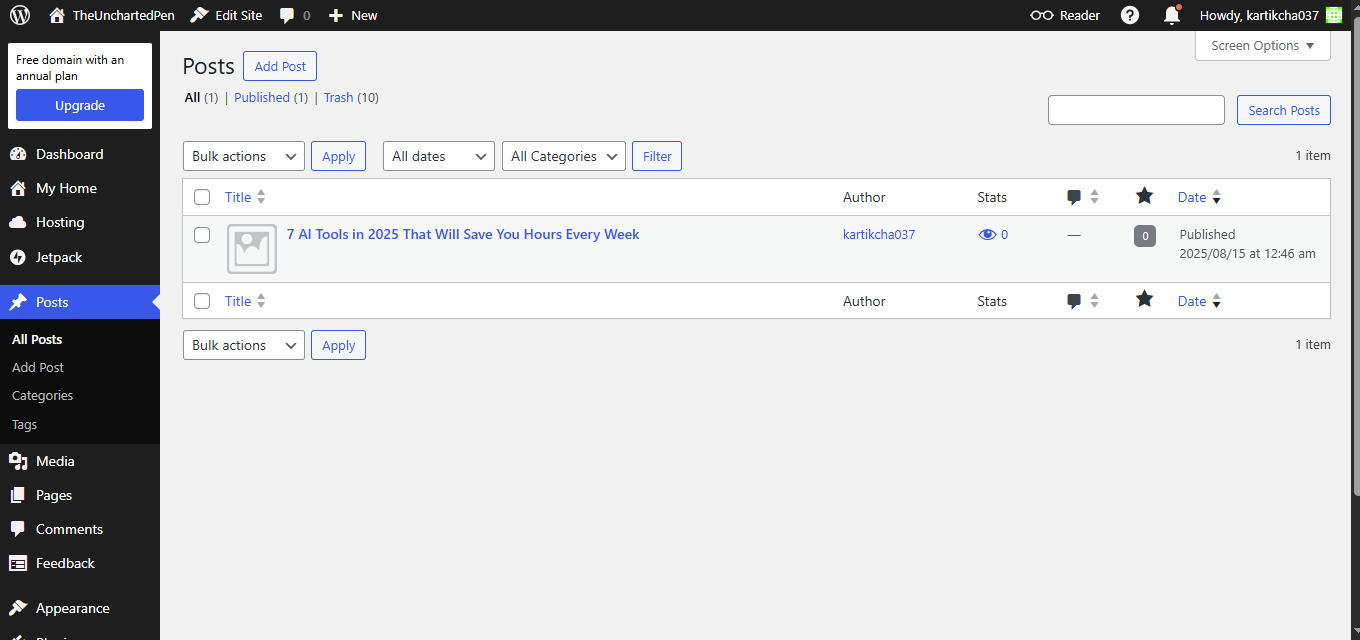
* Step1:Set Trigger in Google Script APP (Only One time when you first write your Script)

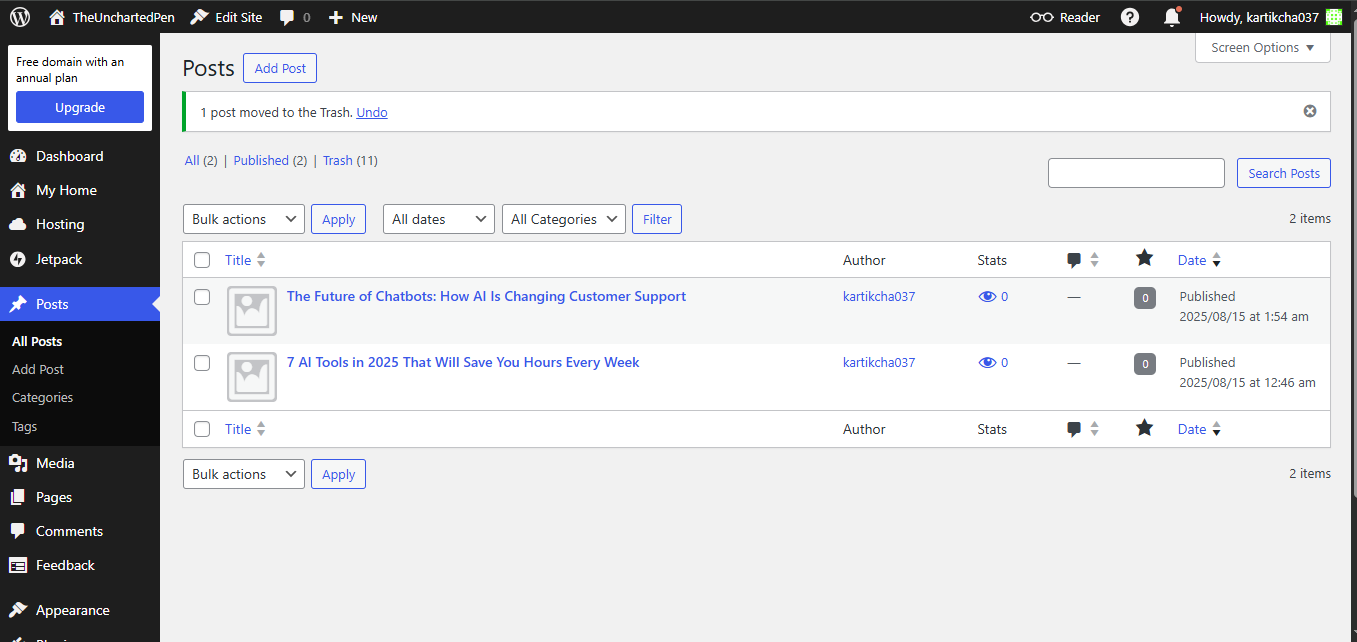


* Step 2:Paste your Topic of Blog in Google Sheet (Add your topices when Your all blog post published which entered in your Script)



* Step 3: Run JS script in Google Script App (Run every time you want to run it automatically trigger at a time that you select in first step and no need to run script every time)
* Step 4:check on your word press account.





## Conclusion

* This automation successfully reduces the manual effort required to publish blogs. It demonstrates practical integration between AI content generation and publishing platforms. Future improvements could include adding image fetching, multi-post scheduling, and error logging.

## References

* Google Apps Script Documentation: https://developers.google.com/apps-script
* Gemini API Documentation: https://ai.google.dev
* Word Press Post by Email:https://wordpress.com/support/post-by-email/