

Automating Quiz Generation in Google Forms Using Google Apps Script and CSV Data

Abstract

This paper describes a systematic approach to automating the creation of quizzes in Google Forms by leveraging Google Apps Script and a CSV file stored in Google Drive. The process eliminates the need for manual data entry, thereby saving time and reducing human error. The methodology outlines how a CSV file containing multiple-choice questions is parsed and converted into a fully functional quiz within Google Forms.

1. Introduction

Educational institutions and instructors often require multiple quizzes for assessments. Manually entering questions into Google Forms is a repetitive and time-consuming task. To address this challenge, Google Apps Script can be employed to automate quiz generation directly from structured CSV files. This paper demonstrates the step-by-step process of uploading a CSV file to Google Drive, integrating it with a Google Apps Script project, and generating a Google Form quiz.

2. Materials and Tools

The following tools are required for this process:

- **Google Drive:** To store the CSV file containing quiz data.
 - **CSV File:** A structured file containing questions, options, and correct answers.
 - **Google Apps Script:** A scripting environment provided by Google for automation.
 - **Google Forms:** The target platform for quiz creation.
-

3. Methodology

The methodology is divided into four primary stages:

3.1 Uploading the CSV File

1. Prepare a CSV file with structured columns (e.g., *Question*, *Option A*, *Option B*, *Option C*, *Option D*, *Correct Answer*).

2. Save the file under a recognizable name such as mcq1.csv.
 3. Upload the CSV file to **Google Drive**.
-

3.2 Creating the Apps Script Project

1. Open **Google Apps Script** from script.google.com.
 2. Create a **new project** and assign a relevant name (e.g., "MCQ Quiz Automation").
 3. Inside the editor, paste the provided code for quiz generation.
 - The code searches Google Drive for mcq1.csv.
 - It reads and parses the file into an array of questions and answers.
 - It creates a new Google Form set as a quiz.
-

3.3 Running the Script

1. Save the script.
 2. From the **Run** menu, execute the function (e.g., createQuizFormFromCSV()).
 3. Grant necessary permissions when prompted (Google requires authorization for Drive and Forms access).
 4. The script parses the CSV, looping through each row to:
 - Insert the question text.
 - Provide multiple-choice or checkbox options.
 - Mark correct answers.
 - Assign points.
-

3.4 Generating the Quiz

- Once the script completes, a new Google Form quiz titled "**MCQ Quiz from CSV**" is automatically created.
 - The quiz is preloaded with all questions, options, and correct answers.
 - Instructors can share the quiz directly with students or further customize it.
-

4. Results

The automation process results in a fully functional Google Form quiz generated directly from CSV data. This eliminates manual question entry and ensures consistency across multiple quizzes. The approach is scalable for large sets of questions (e.g., 50–100 items), significantly reducing instructor workload.

Step 1: CSV File Preparation

1. Create a CSV file with the following **header columns**:

Question, Type, Option1, Option2, Option3, Option4, CorrectAnswer, Points, Required, ShuffleOptions, Description, HelpText, ScaleMin, ScaleMax, GridRows, GridCols, MediaUrl

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Question	Type	Option1	Option2	Option3	Option4	CorrectAnswer	Points	Required	ShuffleOptions	Description	HelpText	ScaleMin	ScaleMax	GridRows	GridCols	MediaUrl
2	What is 2+2?	MCQ	2	3	4	5	4	5	YES	YES	Simple math question	Pick the right answer	NO	NO	NO	NO	NO
3	Select prime numbers	CHECKBOX	2	3	4	5	2,3	10	YES	NO	Multiple correct possible	Select all correct	NO	NO	NO	NO	NO
4	What is your name?	TEXT	NO	NO	NO	NO	NO	NO	YES	NO	Enter your first name	Required field	NO	NO	NO	NO	NO
5	Explain your career goals	PARAGRAPH	NO	NO	NO	NO	NO	NO	NO	NO	Describe your goals briefly	Optional	NO	NO	NO	NO	NO
6	Rate your satisfaction	SCALE	NO	NO	NO	NO	NO	5	YES	NO	1 = Poor, 5 = Excellent	Give honest rating	1	5	NO	NO	NO
7	Match rows and columns	GRID	NO	NO	NO	NO	NO	NO	NO	NO	Match items properly	Grid practice	NO	NO	Apple Banana	Red Yellow	NO
8	Pick multiple per row	CHECKBOX_GRID	NO	NO	NO	NO	NO	NO	NO	NO	Can pick multiple per row	Advanced grid	NO	NO	Q1 Q2 Q3	Yes No Maybe	NO
9	Pick your birth date	DATE	NO	NO	NO	NO	NO	NO	YES	NO	Enter DOB	Date only	NO	NO	NO	NO	NO
10	Pick a time	TIME	NO	NO	NO	NO	NO	NO	NO	NO	Enter preferred time	Time only	NO	NO	NO	NO	NO
11	Look at this image	IMAGE	NO	NO	NO	NO	NO	NO	NO	NO	Sample image question	View carefully	NO	NO	NO	NO	https://upload.wikimedia.org/wikipedia/commons/4/47/PNG_transparency_demonstration_1.png
12	Watch this video	VIDEO	NO	NO	NO	NO	NO	NO	NO	NO	YouTube Video Example	Watch carefully	NO	NO	NO	NO	https://www.youtube.com/watch?v=dQw4w9WgXcQ
13	General Knowledge SECTION	SECTION	NO	NO	NO	NO	NO	NO	NO	NO	Start of GK Section		NO	NO	NO	NO	NO
14	Next Page	PAGE_BREAK	NO	NO	NO	NO	NO	NO	NO	NO	Move to next page		NO	NO	NO	NO	NO
15																	

2. Each row in the CSV defines one form element. Examples include:

- **MCQ** (Multiple Choice)
- **CHECKBOX** (Checkbox)
- **TEXT / PARAGRAPH** (Short or long answer)
- **SCALE** (Rating scale)
- **GRID / CHECKBOX_GRID** (Matrix questions)
- **DATE / TIME** (Calendar inputs)
- **IMAGE / VIDEO** (Media content)
- **SECTION / PAGE_BREAK** (Form structuring elements)

3. Save the file with a name such as mcq1.csv and upload it to **Google Drive**.

Step 2: Apps Script Setup

1. Open script.google.com and create a new project.



apps script



AI Mode **All** Videos Images Short videos News Shopping More ▾ Tools ▾



Google for Developers
<https://developers.google.com/apps-script>

Apps Script | Google for Developers

23 Jul 2025 — Apps Script is a cloud-based JavaScript platform powered by Google Drive that lets you integrate with and automate tasks across Google products.

Script Projects

Create a container-bound project from Google Docs, Sheets, or ...

The Apps Script Dashboard

You can use the Apps Script dashboard to view and manage ...

Google Apps Script overview

You write code in modern JavaScript and have access to ...

Code Samples

Apps Script samples overview · Quickstart. Quickstart samples ...

Reference overview

Apps Script services provide ways for your script to access data on ...

[More results from google.com »](#)

Start a new Scripting



Google Workspace

Home

Apps Script

All products ▾

Resources ▾

Search

Language ▾

Apps Script

Overview

Guides

Reference

Samples

Support

Filter

Home
Developer products
Get started
Build with AI
Try it now

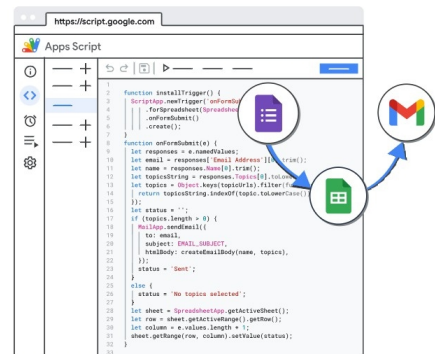
Google Workspace apps
Admin console
Cloud Search
Gmail
Google Calendar
Google Chat
Google Classroom
Google Docs
Google Drive
Google Forms
Google Keep
Google Meet
Google Sheets

Automate & extend Google Workspace with simple code.

Apps Script is a cloud-based JavaScript platform powered by Google Drive that lets you integrate with and automate tasks across Google products.

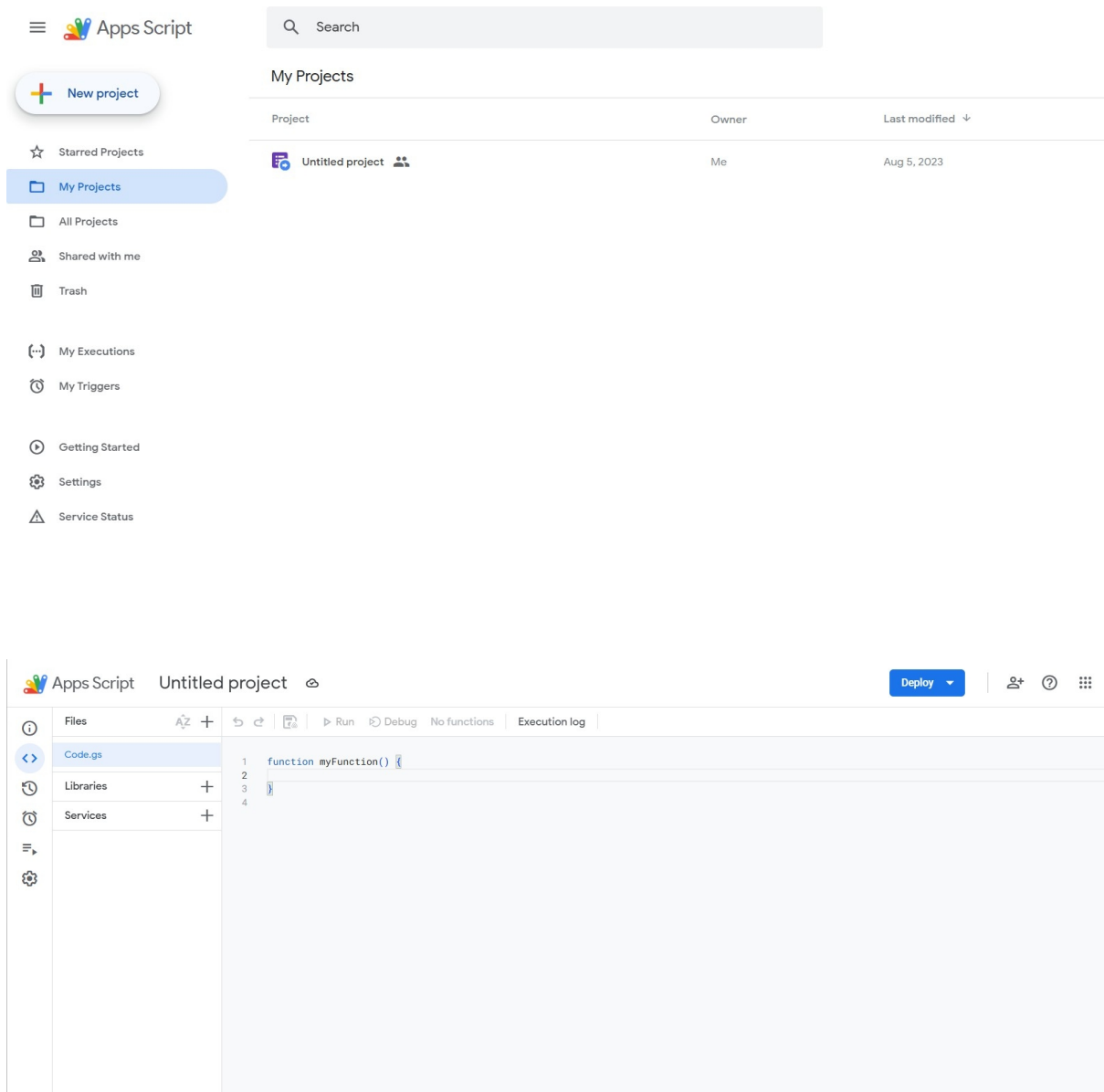
[Start scripting](#)

[What can Apps Script do?](#)



Develop high quality solutions with ease

New Project



2. Paste the Apps Script code that:

- Reads mcq1.csv from Google Drive.
- Parses the CSV into an array.
- Iterates through each row to create the corresponding form item.

The code link is here on [GitHub code](#)

Apps Script Untitled project Unsaved changes Deploy + ? ☰

Files A-Z +

Code.gs

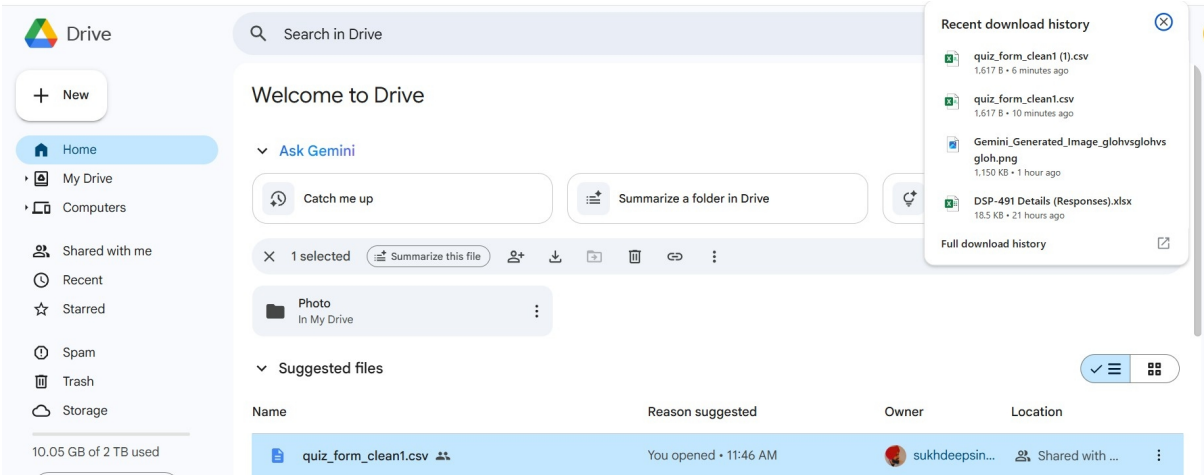
Libraries +

Services +

```
1 function createQuizFormFromCSV() {
2   // Load CSV file from Google Drive
3   var file = DriveApp.getFilesByName("quiz_form_clean2.csv").next();
4   var csvData = Utilities.parseCsv(file.getBlob().getDataAsString());
5
6   // Create Google Form
7   var form = FormApp.create("MCQ Quiz from CSV");
8   form.setIsQuiz(true);
9
10  // Define types that support quiz points
11  var pointSupportedTypes = ["MCQ", "CHECKBOX", "TEXT", "PARAGRAPH", "SCALE"];
12
13  // Loop through rows (skip header)
14  for (var i = 1; i < csvData.length; i++) {
15    var row = csvData[i];
16    if (row.length < 2) continue; // Skip incomplete rows
17
18    var questionText = row[0];
19    var type = row[1] ? row[1].toUpperCase() : "";
20    var options = [row[2], row[3], row[4], row[5]];
21    var correct = row[6];
22    var points = row[7] && row[7] !== "NO" ? parseInt(row[7], 10) : null;
23    var required = row[8] && row[8].toUpperCase() === "YES";
24    var shuffle = row[9] && row[9].toUpperCase() === "YES";
25    var description = row[10] && row[10] !== "NO" ? row[10] : "";
26    var helpText = row[11] && row[11] !== "NO" ? row[11] : "";
27    var scaleMin = row[12] && row[12] !== "NO" ? parseInt(row[12], 10) : 1;
28    var scaleMax = row[13] && row[13] !== "NO" ? parseInt(row[13], 10) : 5;
29    var gridRows = row[14] && row[14] !== "NO" ? row[14].split(",") : [];
30    var gridCols = row[15] && row[15] !== "NO" ? row[15].split(",") : [];
31    var mediaUrl = row[16] && row[16] !== "NO" ? row[16] : "";
32
33    if (!questionText) continue;
```

Step 3: CSV Parsing

1. The script retrieves the CSV file: which is we will upload on google drive like this



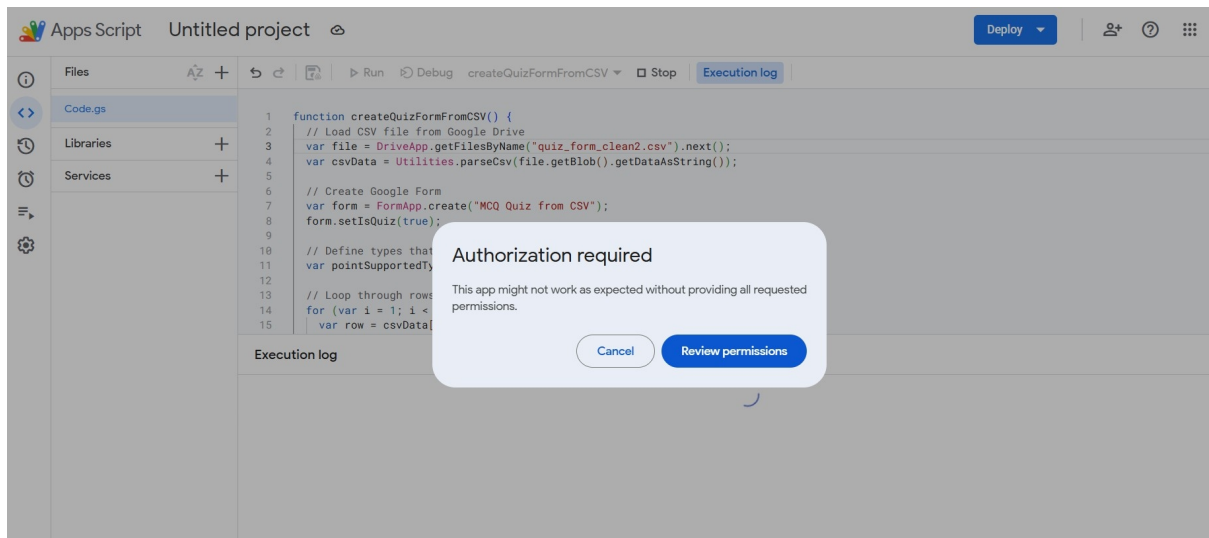
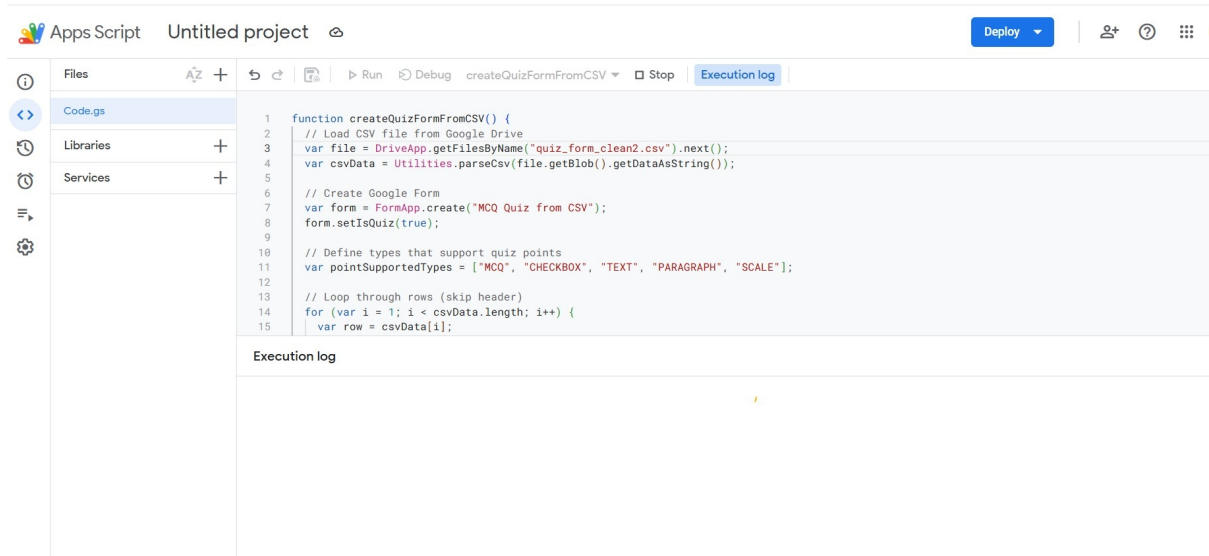
quiz_form_clean2.csv																	
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Question	Type	Option1	Option2	Option3	Option4	CorrectAnswer	Points	Required	ShuffleOptions	Description	HelpText	ScaleMin	ScaleMax	GridRows	GridCols	MediaUrl
2	What is 2+2?	MCQ					5	4	5 YES	YES	Simple math question Pick the right answer		NO	NO	NO	NO	NO
3	Select prime number	CHECKBOX	2	3	4	5	2,3	4	10 YES	NO	Multiple correct post Select all correct		NO	NO	NO	NO	NO
4	What is your name?	TEXT	NO	NO	NO	NO	NO	NO	YES	NO	Enter your first name Required field		NO	NO	NO	NO	NO
5	Explain your career	PARAGRAPH	NO	NO	NO	NO	NO	NO	NO	NO	Describe your goals Optional		NO	NO	NO	NO	NO
6	Rate your satisfaction	SCALE	NO	NO	NO	NO	NO	NO	5 YES	NO	1 = Poor 5 = Excellent One handed rating		1	5	NO	NO	NO
7	Match rows and col	GRID	NO	NO	NO	NO	NO	NO	NO	NO	Match items properly Grid practice		NO	NO	Apple/Banana	Red/Yellow	NO
8	Pick multiple per row	CHECKBOX_GRID	NO	NO	NO	NO	NO	NO	NO	NO	Can pick multiple per Advanced grid		NO	NO	01/02/03	Yes/No/Maybe	NO
9	Pick your birth date	DATE	NO	NO	NO	NO	NO	NO	YES	NO	Enter DOB	Date only	NO	NO	NO	NO	NO
10	Pick a time	TIME	NO	NO	NO	NO	NO	NO	NO	NO	Enter preferred time Time only		NO	NO	NO	NO	NO
11	Look at this image	IMAGE	NO	NO	NO	NO	NO	NO	NO	NO	Sample Image Ques View carefully		NO	NO	NO	NO	https://upload.wikimedia.org/wikipedia/commons/4/4a/Red_banana.jpg
12	Watch this video	VIDEO	NO	NO	NO	NO	NO	NO	NO	NO	YouTube Video Ques Watch carefully		NO	NO	NO	NO	https://www.youtube.com/watch?v=...
13	General Knowledge	SECTION	NO	NO	NO	NO	NO	NO	NO	NO	Start of GK Section		NO	NO	NO	NO	NO
14	Next Page	PAGE_BREAK	NO	NO	NO	NO	NO	NO	NO	NO	Move to next page		NO	NO	NO	NO	NO

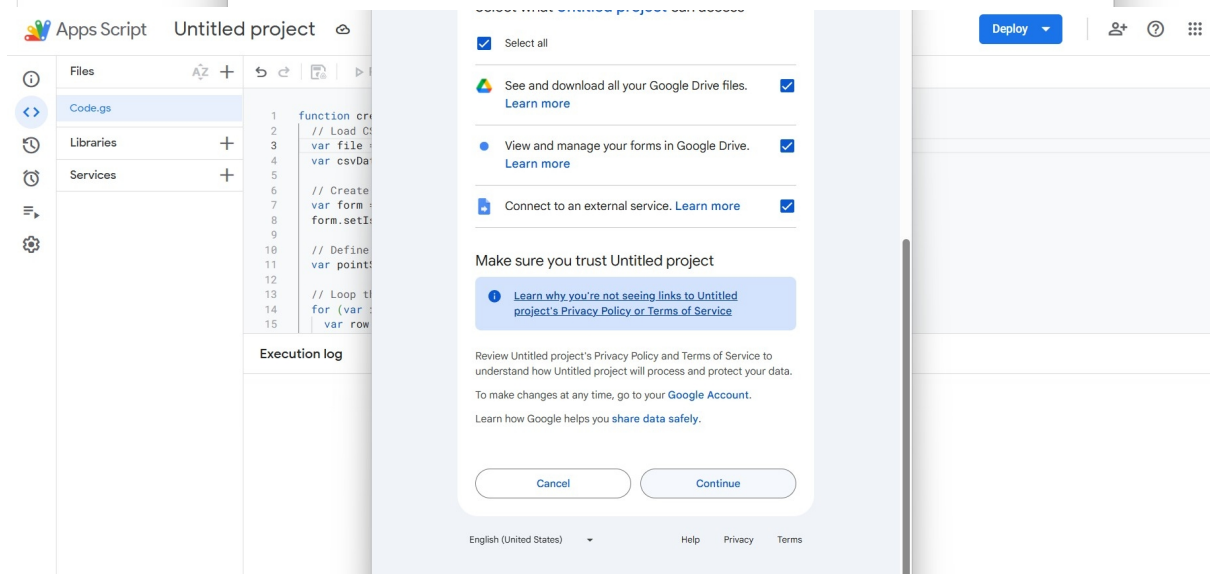
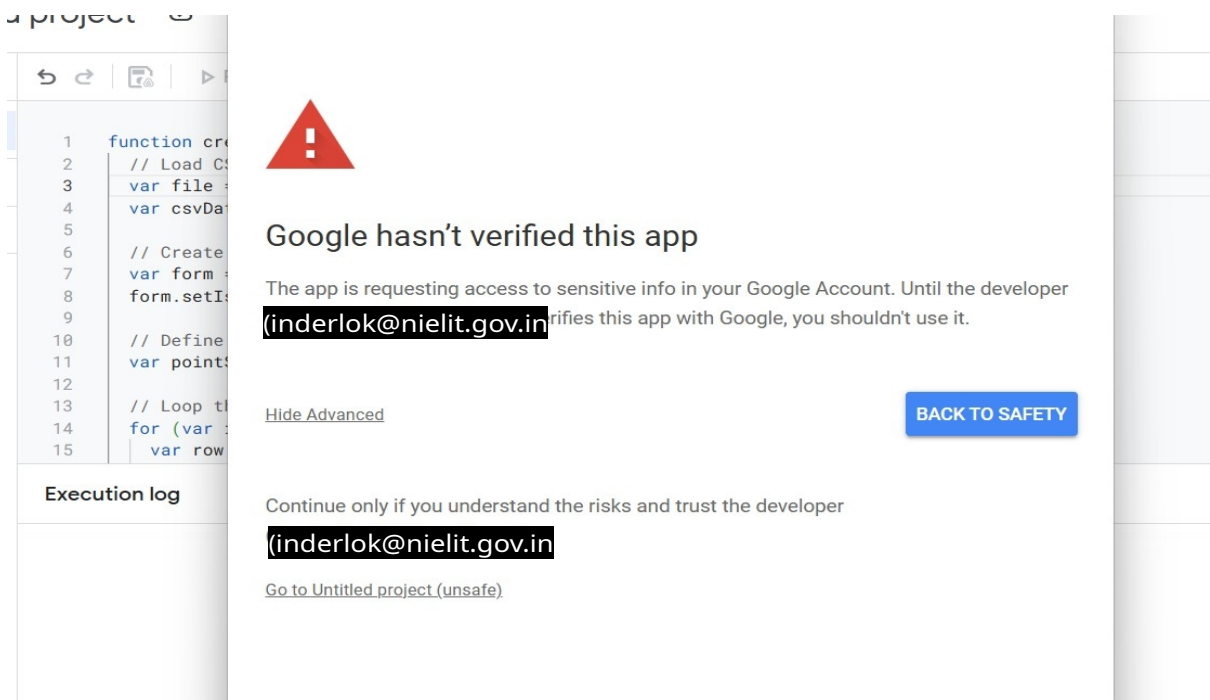
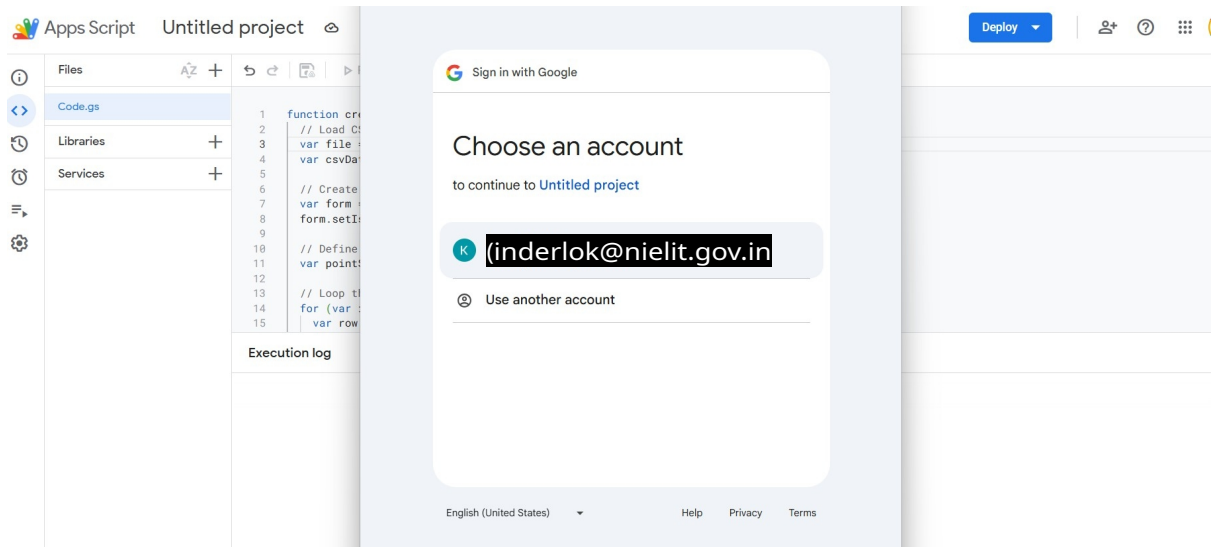
```
var file = DriveApp.getFilesByName("mcq1.csv").next();
var csvData = Utilities.parseCsv(file.getBlob().getDataAsString());
```

2. csvData is a 2D array, where:

- csvData[0] = header row.
- csvData[1] onward = questions and elements.

Go to apps Script and save then run the code this will ask your permissions allow all the permissions





After excuting the script we will get the form link

Execution log			×
11:58:24 AM	Notice	Execution started	
11:58:40 AM	Info	✅ Form created: https://docs.google.com/forms/d/1obPDVd-Us0mMsPywtFQsb4z_00p-2665-Q1d_t2eU7o/edit	
11:58:40 AM	Notice	Execution completed	

Step 4: Google Form Creation

1. A new Google Form is generated:

```
var form = FormApp.create("MCQ Quiz from CSV");  
form.setIsQuiz(true);
```

2. This form will be populated based on CSV rows
-

Step 5: Adding Questions & Elements

The script loops through each CSV row and interprets **Type** to decide what to add:

- **MCQ**: Creates a multiple-choice item with options from Option1-Option4.
- **CHECKBOX**: Creates a checkbox item with multiple correct answers (separated by ; in CorrectAnswer).
- **TEXT / PARAGRAPH**: Creates open-ended input fields.
- **SCALE**: Uses ScaleMin and ScaleMax to set the numeric range.
- **GRID / CHECKBOX_GRID**: Splits GridRows and GridCols into separate options using |.
- **DATE / TIME**: Adds date or time input questions.
- **IMAGE / VIDEO**: Inserts media into the form using the MediaUrl column.
- **SECTION / PAGE_BREAK**: Adds form navigation and structuring.

Additional column behavior:

- **CorrectAnswer**: Marks correct responses for quiz scoring.
- **Points**: Allocates points per question.
- **Required**: Sets whether a question must be answered.

- **ShuffleOptions:** Randomizes option order if set to YES.
 - **Description & HelpText:** Provide supporting instructions.
-

Step 6: Execution

1. Save and run the function `createQuizFormFromCSV()`.
 2. On first run, grant authorization to access Google Drive and Google Forms.
 3. The script generates a **Google Form** with all questions, media, and structure as defined in the CSV.
-

Step 7: Output

- The resulting Google Form will:
- Contain all specified question types.
- Include media (images/videos) where required.
- Apply quiz settings (points, correct answers, required fields).
- Respect structural elements (sections, page breaks).

This ensures the form is a **one-to-one reflection of the CSV design**.

MCQ Quiz from CSV

Published

QuestionsResponsesSettings

Total points: 20

Section 1 of 2

MCQ Quiz from CSV

B

I

U

↺

↻

Form description

What is 2+2? *
Simple math question
Pick the right answer

☐

2

☐

3

☐

4

☐

5

Select prime numbers *
Multiple correct possible
Select all correct

☐

2

☐

3

MCQ Quiz from CSV

Published

QuestionsResponsesSettings

Total points: 20

☐

6

What is your name? *
Enter your first name
Required field

Short answer text

Explain your career goals
Describe your goals briefly
Optional

Long answer text

Rate your satisfaction *
1 = Poor, 5 = Excellent
Give honest rating

1

☐

2

☐

3

☐

4

☐

5

☐

MCQ Quiz from CSV

Published

QuestionsResponsesSettings

Total points: 20

Match items properly
Grid practice

Apple

Red

☐

Yellow

☐

Banana

Red

☐

Yellow

☐

Pick multiple per row
Can pick multiple per row
Advanced grid

Q1

Yes

☐

No

☐

Maybe

☐

Q2

Yes

☐

No

☐

Maybe

☐

Q3

Yes

☐

No

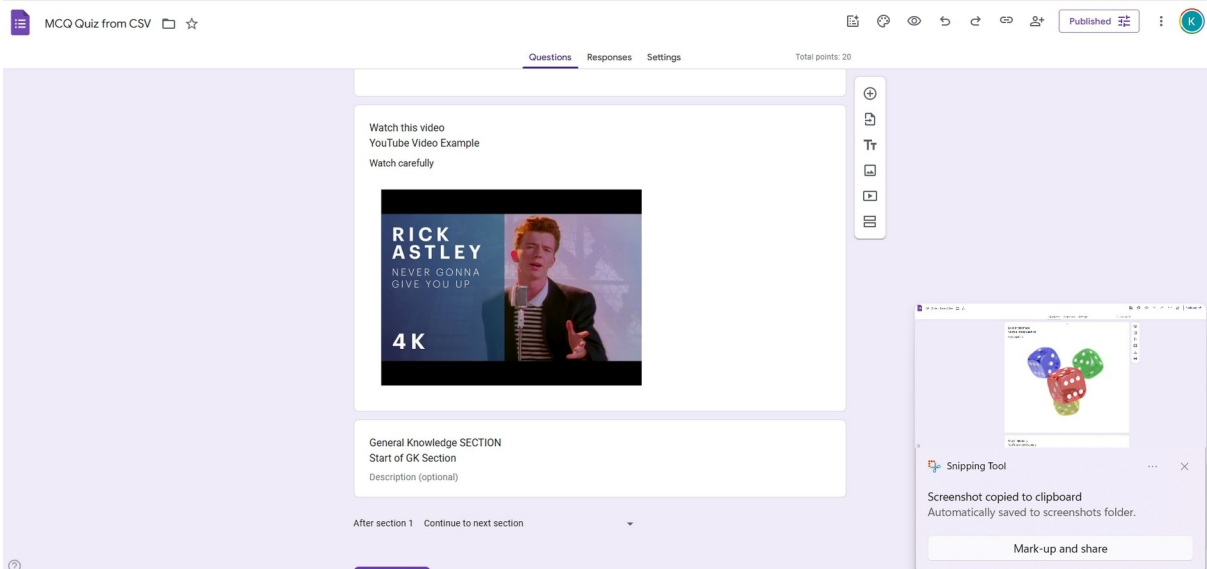
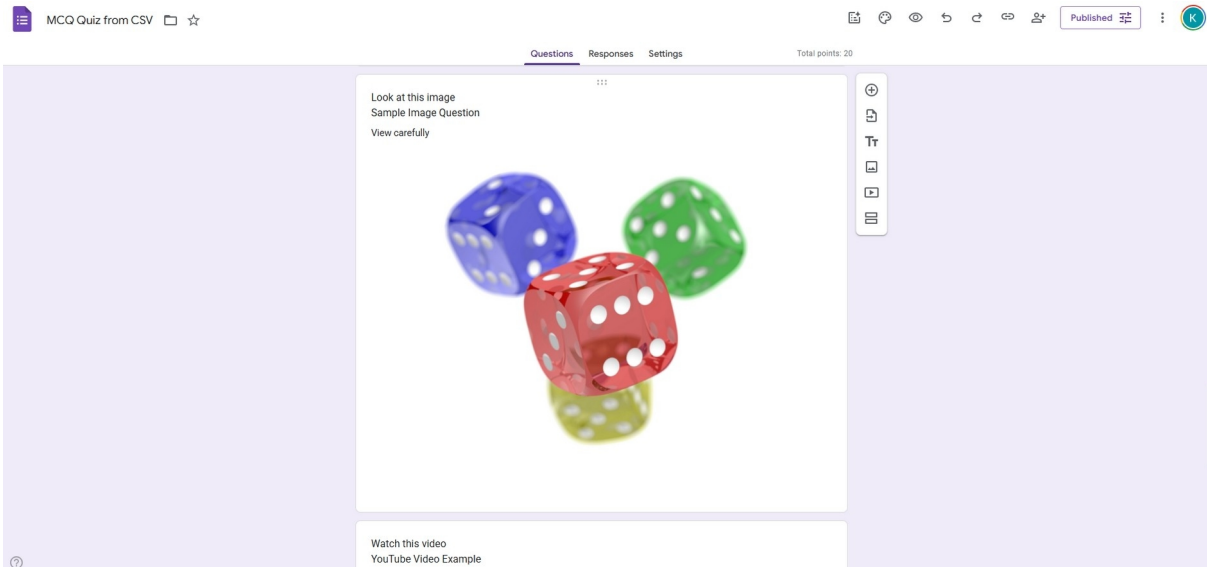
☐

Maybe

☐

Pick your birth date *
Enter DOB
Date only

Month, day, year



Working of this script

Step 1: Load the CSV file

```
var file = DriveApp.getFilesByName("mcq1.csv").next(); var csvData
= Utilities.parseCsv(file.getBlob().getDataAsString());
```

- Finds the file named **mcq1.csv** in Google Drive.
- Reads its content as text, then parses it into a **2D array** (csvData), where each row is a question with details.

Step 2: Create a Google Form

```
var form = FormApp.create("MCQ Quiz from CSV");
form.setIsQuiz(true);
```

- Creates a new Google Form titled **"MCQ Quiz from CSV"**.
 - Sets the form as a **quiz**, so you can assign points and correct answers.
-

Step 3: Define question types that allow points

```
var pointSupportedTypes = ["MCQ", "CHECKBOX", "TEXT", "PARAGRAPH",
"SCALE"];
```

- Only these types will support **quiz points**.
 - For other types (like image, video, section), points are not directly supported.
-

Step 4: Loop through CSV rows

```
for (var i = 1; i < csvData.length; i++) {
  var row = csvData[i];
  ...
}
```

- Skips the header row (i=1).
 - Reads each row of the CSV → every row defines **one question**.
-

Step 5: Extract question data

Each row contains:

```
var questionText = row[0]; // The question
var type = row[1];         // Question type (MCQ, CHECKBOX, etc.)
var options = [row[2], row[3], row[4], row[5]]; // Up to 4 options
var correct = row[6];      // Correct answer var
points = row[7];           // Points for question var
required = row[8];         // Required? YES/NO var shuffle
= row[9];                 // Shuffle options? YES/NO var
description = row[10];     // Extra description var
helpText = row[11];       // Help text (hint) var
scaleMin = row[12];       // For scale-type questions
var scaleMax = row[13];
var gridRows = row[14];   // For grid-type questions
var gridCols = row[15];
var mediaUrl = row[16];   // Image/Video link
```

Step 6: Add question based on type

```
switch (type) { case
"MCQ": ... case
"CHECKBOX": ... case
"TEXT": ...
    case "PARAGRAPH": ...
case "SCALE": ...
case "GRID": ...
    case "CHECKBOX_GRID": ...
case "DATE": ... case
"TIME": ... case "IMAGE":
... case "VIDEO": ...
case "SECTION": ... case
"PAGE_BREAK": ... }
```

- **MCQ (Multiple Choice)** → Creates choices, marks correct answer.
 - **CHECKBOX** → Creates multiple correct answers.
 - **TEXT / PARAGRAPH** → Creates short/long answer input.
 - **SCALE** → Rating scale (like 1-5).
 - **GRID / CHECKBOX_GRID** → Rows & columns (like a matrix).
 - **DATE / TIME** → Date or time picker.
 - **IMAGE / VIDEO** → Fetches from a URL and inserts.
 - **SECTION / PAGE_BREAK** → Adds form sections.
-

Step 7: Apply common properties

```
if (item) {
    if (description) item.setTitle(item.getTitle() + "\n" + description);
    if (helpText && item.setHelpText) item.setHelpText(helpText);

    if (points != null) {
        if (pointSupportedTypes.includes(type) && item.setPoints) {
            item.setPoints(points);
        }
    }


    if (required && item.setRequired) item.setRequired(true);
    if (shuffle && item.setShuffleOptions) item.setShuffleOptions(true);
}
```

- Adds **description** under the question title.

- Adds **help text** if provided.
 - Sets **points** (only if supported).
 - Marks as **required** if needed.
 - Shuffles options if YES.
-

Step 8: Final log

```
Logger.log(" Form created: " + form.getEditUrl());
```



- Prints the **edit link** of the created Google Form.

The apps script

```
function createQuizFormFromCSV() {  
  
    // Load CSV file from Google Drive  
  
    var file = DriveApp.getFilesByName("mcq1.csv").next();  
  
    var csvData = Utilities.parseCsv(file.getBlob().getDataAsString());  
  
  
    // Create Google Form  
  
    var form = FormApp.create("MCQ Quiz from CSV");  
  
    form.setIsQuiz(true);  
  
  
    // Define types that support quiz points  
  
    var pointSupportedTypes = ["MCQ", "CHECKBOX", "TEXT", "PARAGRAPH", "SCALE"];  
  
  
    // Loop through rows (skip header)  
  
    for (var i = 1; i < csvData.length; i++) {  
  
        var row = csvData[i];  
  
        if (row.length < 2) continue; // Skip incomplete rows
```

```

var questionText = row[0];

var type = row[1] ? row[1].toUpperCase() : "";

var options = [row[2], row[3], row[4], row[5]];

var correct = row[6];

var points = row[7] && row[7] !== "NO" ? parseInt(row[7], 10) : null;

var required = row[8] && row[8].toUpperCase() === "YES";

var shuffle = row[9] && row[9].toUpperCase() === "YES";

var description = row[10] && row[10] !== "NO" ? row[10] : "";

var helpText = row[11] && row[11] !== "NO" ? row[11] : "";

var scaleMin = row[12] && row[12] !== "NO" ? parseInt(row[12], 10) : 1;

var scaleMax = row[13] && row[13] !== "NO" ? parseInt(row[13], 10) : 5;

var gridRows = row[14] && row[14] !== "NO" ? row[14].split("|") : [];

var gridCols = row[15] && row[15] !== "NO" ? row[15].split("|") : [];

var mediaUrl = row[16] && row[16] !== "NO" ? row[16] : "";

if (!questionText) continue;

var item = null;

switch (type) {
  case "MCQ":
    var mcqOpts = options.filter(o => o && o.toUpperCase() !== "NO");
    if (mcqOpts.length === 0) continue;
    item = form.addMultipleChoiceItem().setTitle(questionText);
    var mcqChoices = mcqOpts.map(opt => item.createChoice(opt, opt === correct));
    item.setChoices(mcqChoices);

```



```
break;
```

```
case "CHECKBOX":
```

```
    var cbOpts = options.filter(o => o && o.toUpperCase() !== "NO");
```

```
    if (cbOpts.length === 0) continue;
```

```
    item = form.addCheckboxItem().setTitle(questionText);
```

```
    var cbChoices = cbOpts.map(opt =>
```

```
        item.createChoice(opt, correct && correct.split(";").includes(opt))
```

```
    );
```

```
    item.setChoices(cbChoices);
```

```
    break;
```

```
case "TEXT":
```

```
    item = form.addTextItem().setTitle(questionText);
```

```
    break;
```

```
case "PARAGRAPH":
```

```
    item = form.addParagraphTextItem().setTitle(questionText);
```

```
    break;
```

```
case "SCALE":
```

```
    item = form.addScaleItem()
```

```
        .setTitle(questionText)
```

```
        .setBounds(scaleMin, scaleMax);
```

```
    break;
```

```

case "GRID":

    if (gridRows.length && gridCols.length) {

        var gridTitle = questionText + (points ? " (Points: " + points + ")" : "");

        item =
form.addItem().setTitle(gridTitle).setRows(gridRows).setColumns(gridCols);

    }

    break;


case "CHECKBOX_GRID":

    if (gridRows.length && gridCols.length) {

        var gridTitleCB = questionText + (points ? " (Points: " + points + ")" : "");

        item =
form.addCheckboxGridItem().setTitle(gridTitleCB).setRows(gridRows).setColumns(gridC
ols);

    }

    break;


case "DATE":

    item = form.addDateItem().setTitle(questionText + (points ? " (Points: " + points +
)" : ""));

    break;


case "TIME":

    item = form.addTimeItem().setTitle(questionText + (points ? " (Points: " + points + ")"
: ""));

    break;


case "IMAGE":

```

```

if (mediaUrl) {
    try {
        var imgResponse = UrlFetchApp.fetch(mediaUrl, { muteHttpExceptions: true });
        if (imgResponse.getResponseCode() === 200) {
            var imgBlob = imgResponse.getBlob();

            item = form.addItem().setTitle(questionText + (points ? " (Points: " + points
+ ")" : ""));

            item.setImage(imgBlob);
        } else {
            Logger.log("⚠ Image not found for: " + questionText + " (" + mediaUrl + ")");
        }
    } catch (e) {
        Logger.log("❌ Error fetching image for: " + questionText + " -> " + e);
    }
}

break;

```

```

case "VIDEO":
    if (mediaUrl) {
        try {
            if (mediaUrl.includes("youtube.com") || mediaUrl.includes("youtu.be")) {
                item = form.addVideoItem().setTitle(questionText + (points ? " (Points: " + points
+ ")" : ""));

                item.setVideoUrl(mediaUrl);
            } else {
                Logger.log("⚠ Invalid video URL for: " + questionText + " (" + mediaUrl + ")");
            }
        }
    }

```

```

    } catch (e) {
        Logger.log("✗ Error adding video for: " + questionText + " -> " + e);
    }
}

break;

case "SECTION":
    item = form.addSectionHeaderItem().setTitle(questionText);
    break;

case "PAGE_BREAK":
    item = form.addPageBreakItem().setTitle(questionText);
    break;

default:
    Logger.log("Unknown type: " + type + " for question: " + questionText);
    continue;
}

// Apply common properties
if (item) {
    if (description) item.setTitle(item.getTitle() + "\n" + description);
    if (helpText && item.setHelpText) item.setHelpText(helpText);

    // Only apply points where supported
    if (points !== null) {

```

```
if (pointSupportedTypes.includes(type) && item.setPoints) {  
    item.setPoints(points);  
}  
  
// else: already appended to title for unsupported  
}  
  
if (required && item.setRequired) item.setRequired(true);  
if (shuffle && item.setShuffleOptions) item.setShuffleOptions(true);  
}  
}  
  
Logger.log("✅ Form created: " + form.getEditUrl());  
}
```