



[Get ALL AI Automation + Agent Templates >>](#)

Prompts & Links - This AI System Clones Instagram Reels Used by \$10K/Month Creators (100% Automatic)

(Follow the instructions in this video: <https://youtu.be/DriwtqB-4zI>)

Templates:

AI Automation Template for Instant Setup

<https://fabimarkl.com/automation-templates/#reels-clone>

Google Sheet Template (must follow YT tutorial for correct setup):

https://docs.google.com/spreadsheets/d/1K2EMQ-hEyDhYPH_A8QT6Sq4tuSuiXnqFVIX1r54Lxj4/copy

Links:

Free Trial for N8N Account:

<https://n8n.io/>

Download Telegram App:

<https://telegram.org/>

Get API Key for OpenAI:

<https://platform.openai.com/>

Setup Perplexity Account:

<https://www.perplexity.ai/>

Captions App Account:

<https://www.captions.ai/>

Get Instagram Video Scraper:

<https://rapidapi.com/premium-apis-oanor/api/instagram-scraper21>

Setup Cloudinary Account

https://cloudinary.com/users/register_free

Setup JSON2Video Account:

<https://json2video.com/>

[Click Here to Get AI Agent Template for Instant Setup >>](#)

Setup Make.com Account:

<https://www.make.com/>

Setup Upload Posts Account (TikTok)

<https://www.upload-post.com/>

**n8n and [make.com](https://www.make.com/) links are affiliate links that will give me a small percentage of every sale, but have no effect on the price you pay or their free trial.*

AI Agent/Automation Setup:

Get Post ID Function:

```
const input = $input.first().json.message.text; // or whatever
your Telegram node outputs
const match =
input.match(/(?:instagram\.com\/(?:reel|p|tv)\/)([^\?#&]+)/);

if (!match || !match[1]) {
  throw new Error("No valid Instagram post code found.");
}

return [{ code: match[1] }];
```

Instagram Downloader:

Name: x-rapidapi-host

Value: instagram-scraper21.p.rapidapi.com

Name: x-rapidapi-key

Value: YOUR_API_KEY

HTTP Request Upload Image

https://api.cloudinary.com/v1_1/INSERT_CLOUDINARY_ID/image/upload

OpenAI Extract Text from Image:

```
what is the main text in this image that was just analysed: {{
$json.choices[0].message.content }}
```

#Only return that main text as your output. Do not use any ""

#only consider the text at the top according to the image description above. Do ignore any other text.

Filter Audio File

```
const mpd = $('Get
Cover').first().json.data.post.dash_info.video_dash_manifest;

// Match the first <AdaptationSet ... contentType="audio"> ...
<BaseURL>...</BaseURL>
const audioMatch =
mpd.match(/<AdaptationSet[^>]*contentType="audio"[^>]*>[\s\S]*?<BaseURL>(.*?)<\BaseURL>/);

if (!audioMatch || !audioMatch[1]) {
  throw new Error("Audio BaseURL not found in DASH manifest.");
}

const audioUrl = audioMatch[1].replace(/&/g, "&"); // decode
HTML entities
return [{ audioUrl }];
```

Generate Template ID

```
function generateRandomCode(length = 20) {
  const chars =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
  let result = '';
  for (let i = 0; i < length; i++) {
    result += chars[Math.floor(Math.random() * chars.length)];
  }
  return result;
}

return [
  {
    json: {
      code: generateRandomCode()
    }
  }
];
```

Perplexity:

<https://api.perplexity.ai/chat/completions>

Name: Authorization

Value: Bearer YOUR_API_KEY

Name: Content-Type

Value: application/json

```
{
  "model": "sonar-reasoning",
  "messages": [
    {
      "role": "user",
      "content": "Suggest a content idea different from this video
script: '{{ $('Transcribe').item.json.text }}'. It should be in
the same niche and on the exact same topic or content idea but
offer fresh value. You must pick one idea from your research that
matches the topic idea of the video script exactly but is also
different and unique from it so it would stand out on social
media. Example: if the video script contains a list of tools, your
topic must also be a list of tools in that video script topic but
slightly different, maybe different tools etc. If the video's
script is about a plan, strategies, or whatever, you must also
make your topic about that. So you must maintain the nature of the
topic in the video script. You absolutely must be specific as the
original video script. You can't just mention generic tools or
strategies if the original video script contains specific tools.
Etc. That is the level of accuracy and perfect matching of the
video script original topic. Make sure it appeals to a broad
audience like the example."
    }
  ]
}
```

CleanUp Function:

```
// Step 1: Pull raw input
let raw = $input.first().json.choices[0].message.content;

// Step 2: Forcefully remove anything between <think> and </think>
let cleaned = raw.replace(/<think>(.\n)*?</think>/gi,
'').trim();

// Optional cleanup: remove leading/trailing blank lines
cleaned = cleaned.replace(/^\s+|\s+$/g, '');

// Done
```

[Click Here to Get AI Automation Template for Instant Setup >>](#)

```
return [  
  {  
    json: {  
      cleanedResponse: cleaned  
    }  
  }  
];
```

OpenAI Generate Content:

You are rewriting a Instagram Reel video script, caption, and overlay – not inventing a new one. You must follow this format and obey these rules strictly.

CONTEXT:

Here is the content idea to use:

```
{{ $json.cleanedResponse }}
```

STEP 1: Rewrite the original video script BELOW using the new topic/context above but maintain as stubbornly as possible the original script structure and style:

Original script:

```
{{ $('Save Post Info').item.json['Video Script Template'] }}
```



DO NOT CHANGE the original structure or style but

This includes:

- Numbered list
- Sentence breaks
- "I" or first-person narration
- Colloquial/informal tone (like "you're gonna wanna...")



You MUST keep:

- first person narration of the original script at all costs
- MUST be under 700 characters (yes "Characters" not wordcount) this is an absolute MUST, no more than 700 characters!!! But never change the structure or narration style of the original script. It must be an exact imitation.



You MAY change:

- Tool names
- Use cases

- Descriptions
- Niche-specific keywords

#Rule: never use any characters like "" in your generated video script as this will yield syntax errors.

STEP 2: Rewrite the caption text using the new topic.

Keep:

- Same structure and tone
- Same use of #hashtags but space between each hashtag
- Similar sentence count and layout

Caption:

```
{{ $('Save Post Info').item.json['Captions Template'] }}
```

STEP 3: Rewrite the text overlay (short version for the thumbnail or first screen)

Keep:

- EXACT Same length format, case, structure
- Do NOT invent new words unless absolutely necessary

Overlay:

```
{{ $('Extract Text').first().json.message.content }}
```

FINAL OUTPUT FORMAT (no markdown formatting):

Text Overlay: [REWRITTEN TEXT OVERLAY]

Video Script: [REWRITTEN SCRIPT]

Caption Text: [REWRITTEN CAPTION TEXT]

DO NOT return any explanations. Only return the rewritten sections.

Split Function

```
const input = $input.first().json.choices[0].message.content;
```

```
// More flexible splitting regex that tolerates single or multiple
newlines
const textOverlayMatch = input.match(/Text
Overlay:\s*(.+?) (?:\n|\n)/s);
const videoScriptMatch = input.match(/Video
Script:\s*(.+?) (?:\n|\n)Caption Text:/s);
const captionTextMatch = input.match(/Caption Text:\s*(.+)/s);

return [
  {
    json: {
      textOverlay: textOverlayMatch ? textOverlayMatch[1].trim() :
null,
      videoScript: videoScriptMatch ? videoScriptMatch[1].trim() :
null,
      captionText: captionTextMatch ? captionTextMatch[1].trim() :
null,
    }
  }
];
```

Generate Video ID

```
function generateRandomCode(length = 20) {
  const chars =
'ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789';
  let result = '';
  for (let i = 0; i < length; i++) {
    result += chars[Math.floor(Math.random() * chars.length)];
  }
  return result;
}

return [
  {
    json: {
      code: generateRandomCode()
    }
  }
];
```

Get Speaker List:

<https://api.captions.ai/api/creator/list>

Name: Content-Type
Value: application/json
Name: x-api-key
Value: YOUR_API_KEY

Generate Ai Clone Video

<https://api.captions.ai/api/creator/submit>

```
{
  "script": "{{ $('Save New Post').item.json.Script }}",
  "creatorName": "{{ $json.supportedCreators[0] }}",
  "resolution": "fhd"
}
```

Get Video URL:

<https://api.captions.ai/api/creator/poll>

```
{
  "operationId": "{{ $json["operationId"] }}"
}
```

JSON2Video Credentials:

```
{
  "headers": {
    "x-api-key": "INSERT_YOUR_API"
  }
}
```

Add Text to Video (<https://api.json2video.com/v2/movies>):

```
{
  "id": "qbaasr7s",
  "resolution": "instagram-story",
  "quality": "high",
  "scenes": [
    {
      "id": "qyjh9lwj",
      "comment": "Scene 1",
      "elements": []
    }
  ],
}
```

```

"elements": [
  {
    "id": "q6dznczcv",
    "type": "video",
    "src": "{{ $json.url }}",
    "resize": "cover"
  },
  {
    "id": "top-text",
    "type": "text",
    "text": "{{ $('Split').first().json.textOverlay }}",
    "settings": {
      "font-family": "Arial",
      "font-size": "60px",
      "color": "#000000",
      "background-color": "#FFFFFF",
      "horizontal-position": "center",
      "vertical-position": "top",
      "margin-top": "100px",
      "word-break": "break-word",
      "overflow-wrap": "break-word",
      "font-weight": "bold",
      "text-align": "center",
      "width": "80%",
      "padding": "0px",
      "line-height": "1.2",
      "margin": "50px 0 0 0",
      "padding": "0 10px 0 10px",
      "border-radius": "15px"
    }
  },
  {
    "id": "q41n9kxp",
    "type": "subtitles",
    "settings": {
      "style": "classic",
      "position": "bottom",
      "font-family": "Arial",
      "font-size": "100px",
      "word-color": "#ffd700",
      "shadow-color": "#260B1B",
      "line-color": "#F1E7F4",
      "shadow-offset": 0,
      "box-color": "#260B1B",
      "outline-color": "#000000",
      "outline-width": 8
    }
  },

```

```
      "language": "en"
    }
  ]
}
```

Get video:

[https://api.json2video.com/v2/movies?id={{ \\$json.project }}](https://api.json2video.com/v2/movies?id={{ $json.project }})

Telegram Response:

Here is the final version of your video post:

Caption Text:

```
{{ $('Split').item.json.captionText }}
```

```
{{ $json['Video URL'] }}
```

Make Webook in n8n

Name: Content-Type

Value: application/json

TikTok Post Module (upload-post.com):

<https://api.upload-post.com/api/upload>

Authorization

Apikey YOUR_API_KEY

Write Date in Sheet:

```
{{formatDate(now; "  YYYY-MM-DD hh:mm A")}}
```

Make.com Publisher Webhook Response:

```
{
  "platform": "Facebook",
  "postTitle": "{{1.PostTitle}}",
  "ChatID": "{{1.`Chat ID`}}",
  "publishTime": "{{formatDate(now; "YYYY-MM-DD HH:mm:ss")}}"
}
```

Telegram Response

The video "{{ \$json.body.postTitle }}" was successfully published to {{ \$json.body.platform }} at {{ \$json.body.publishTime }}.

Cost Breakdown (~ 30 Second Videos)

Tool	30 videos/mo	60 videos/mo	90 videos/mo	150 videos/mo
CaptionsAI API	\$45.00	\$90.00	\$135.00	\$225.00
n8n	\$24.00	\$24.00	\$24.00	\$24.00
n8n (selfhosted)	\$5.00	\$5.00	\$5.00	\$5.00
OpenAI	\$0.30	\$0.60	\$0.90	\$1.50
Cloudinary	\$0.00	\$0.00	\$0.00	\$0.00
Perplexity	\$0.30	\$0.60	\$0.90	\$1.50
Make.com	\$0.00	\$0.00	\$10.59	\$10.59
JSON2Video	\$2.10	\$4.20	\$6.30	\$10.50
Instagram Scraper	\$0.00	\$0.00	\$0.00	\$9.99
Upload-Post.com (TikTok)	\$14.00	\$14.00	\$14.00	\$14.00
Cost per video (n8n webapp)	\$2.86	\$2.22	\$2.13	\$1.91
TOTAL (n8n webapp)	\$85.70	\$133.40	\$191.69	\$287.09
Cost per video (n8n selfhosted)	\$2.23	\$1.91	\$1.91	\$1.79
TOTAL (n8n selfhosted)	\$66.70	\$114.40	\$172.69	\$268.09
Time saved (12 min/video)	6 hours	12 hours	18 hours	30 hours

Budget Version Cost Breakdown (~ 20 Sec. Video + Manual TikTok Posting)

Tool	30 videos/mo	60 videos/mo	90 videos/mo	150 videos/mo
CaptionsAI API	\$30.00	\$60.00	\$90.00	\$150.00
n8n	\$24.00	\$24.00	\$24.00	\$24.00
n8n (selfhosted)	\$5.00	\$5.00	\$5.00	\$5.00
OpenAI	\$0.30	\$0.60	\$0.90	\$1.50
Cloudinary	\$0.00	\$0.00	\$0.00	\$0.00
Perplexity	\$0.30	\$0.60	\$0.90	\$1.50
Make.com	\$0.00	\$0.00	\$10.59	\$10.59
JSON2Video	\$2.10	\$4.20	\$6.30	\$10.50
Instagram Scraper	\$0.00	\$0.00	\$0.00	\$9.99
TikTok (Manual Posting)	\$0.00	\$0.00	\$0.00	\$0.00
Cost per video (n8n webapp)	\$1.89	\$1.49	\$1.47	\$1.39
TOTAL (n8n webapp)	\$56.70	\$89.40	\$132.69	\$208.09
Cost per video (n8n selfhosted)	\$1.26	\$1.17	\$1.26	\$1.19
TOTAL (n8n selfhosted)	\$37.70	\$70.40	\$113.69	\$179.09
Time saved (12 min/video)	6 hours	12 hours	18 hours	30 hours