🎬 Complete Instructional Video Script

Title: Running Ollama Locally and Accessing It from Google Colab via Pinggy (Mini Model)

1️⃣ Introduction (30 seconds)

“In this video, I’ll show you how to run Ollama on a local PC and then access it from Google Colab using a Pinggy tunnel.

We’ll keep this lightweight by using a **mini model**, which is ideal for demos and teaching.

I’ll walk through installation, configuration, tunnelling, and finally calling Ollama from Colab.”

2️⃣ Installing Ollama Locally (2 minutes)

“First, we need Ollama running locally on the PC.”

On screen: Browser

- Go to **ollama.com**
- Download and install Ollama for your operating system

On screen: Terminal

```
ollama --version
```

“This confirms Ollama is installed correctly.”

3️⃣ Pulling and Testing the Mini Model (2 minutes)

“Next, I’ll pull a lightweight mini model and confirm Ollama works locally.”

```
ollama pull phi3:mini
```

```
ollama run phi3:mini
```

“At this point, Ollama is working locally, but it’s only accessible from this machine.”

4️⃣ Critical Configuration Step – Required for Tunnelling (1 minute)

“Before we expose Ollama through a tunnel, there’s one important configuration step.

By default, Ollama only listens on localhost, which means tunnelling tools like Pinggy cannot reach it.

We need to tell Ollama to listen on all network interfaces.”

Linux / macOS

```
export OLLAMA_HOST=0.0.0.0:11434
```

```
ollama serve
```

Windows (PowerShell)

```
setx OLLAMA_HOST "0.0.0.0:11434"
```

“After setting this, restart Ollama so the change takes effect.”

Verification Step

```
curl http://<your-ip>:11434/api/tags
```

“If this works using your machine’s IP address, then tunnelling will work as well.”

5️⃣ Exposing Ollama Using Pinggy (3 minutes)

“Now we’ll expose Ollama using a Pinggy tunnel so it can be accessed remotely.”

On screen: Terminal

```
ssh -p 443 -R0:localhost:11434 a.pinggy.io
```

“Pinggy generates a temporary public URL that forwards traffic to my local Ollama instance.”

Pause and clearly show the Pinggy URL on screen.

Quick Tunnel Test

```
curl https://<pinggy-url>/api/tags
```

“If this returns the model list, the tunnel is working.”

6️⃣ Calling Ollama from Google Colab (4 minutes)

“Now I’ll switch to Google Colab and call Ollama through the tunnel.”

On screen: Google Colab

```
import requests

url = "https://<pinggy-url>/api/generate"
payload = {
  "model": "phi3:mini",
  "prompt": "Explain interest rates in one short paragraph.",
  "stream": False
}
```

```
response = requests.post(url, json=payload)

print(response.json()["response"])
```

“This request is going from Colab, through Pinggy, to my local PC, and into Ollama using the mini model.”

7️⃣ Common Issues and Notes (1 minute)

“A few important things to be aware of:

- Pinggy URLs are temporary and can expire
- Ollama must be listening on 0.0.0.0, not just localhost
- If requests time out, restart Pinggy first, then Ollama
- Mini models are ideal for demos and teaching because they respond quickly”

8️⃣ Wrap-Up (30 seconds)

“You now have Ollama running locally with a mini model and accessible from Google Colab using a tunnel.

This allows you to demonstrate local inference using your own hardware.

Thanks for watching.”

✅ End of Script