Peppo AI Engineering Internship – Technical Challenge

Objective:
Demonstrate your ability to build and deploy an **AI-integrated application** in the cloud.
We want to see how you handle **model integration, prompt engineering, cloud deployment, and code quality**.

Problem Statement

Build a minimal AI video generation web app that:

1. Takes a **user prompt** as input (text box).
2. Generates a **short video (5–10 seconds)** based on the prompt using **any AI API or open-source model** you prefer (e.g., Pika Labs, Runway, Stability AI, OpenAI's image→video pipelines, etc.).
3. Displays the generated video in the browser.
4. Deploys the app to a **public cloud** (Render, Railway, AWS, GCP, Azure, or Vercel).
5. Shares both:
   - ○ **Live app link** (publicly accessible).
   - ○ **Code repository** (GitHub/Bitbucket).

Requirements

- **Frontend:** Can be minimal — HTML/CSS/JS or React/Vue/Svelte.
- **Backend:** Python (FastAPI/Flask) or Node.js.
- **Deployment:** Public URL required (no localhost demos).

- **README:** Clear setup instructions, dependencies, and API key configuration.
- Use **.env** or secure methods for API keys (do not commit keys to GitHub).

Time Limit

- You have **24 hours** from receiving this problem to submit.
- Partial submissions are acceptable but will be scored lower.

Evaluation Rubric (100 points)

Criteria	Description	Weight
Functionality	App works end-to-end: takes prompt → generates → displays video.	**40**
Deployment	Public cloud deployment is stable and loads without errors.	**20**
Code Quality	Readable, well-structured, commented, and follows good practices.	**15**
Documentation	README is clear, concise, and explains setup/deployment.	**10**
Innovation	Any enhancements: extra features, better UI, caching, prompt improvements, RAG-based context injection, etc.	**10**
Security	Proper handling of API keys and credentials.	**5**

Submission Format

1. `Submit your app here - https://forms.gle/Y6nvT7Cz9JE2UGR38`

Pro Tips

- Keep it **minimal but functional** — don’t over-engineer.

- Focus on **getting something live** rather than perfect.

- If APIs are slow/limited, mock with a placeholder video + logs but explain how you’d handle production.

- Bonus points for **video streaming optimizations** or **RAG-style prompt improvements**.