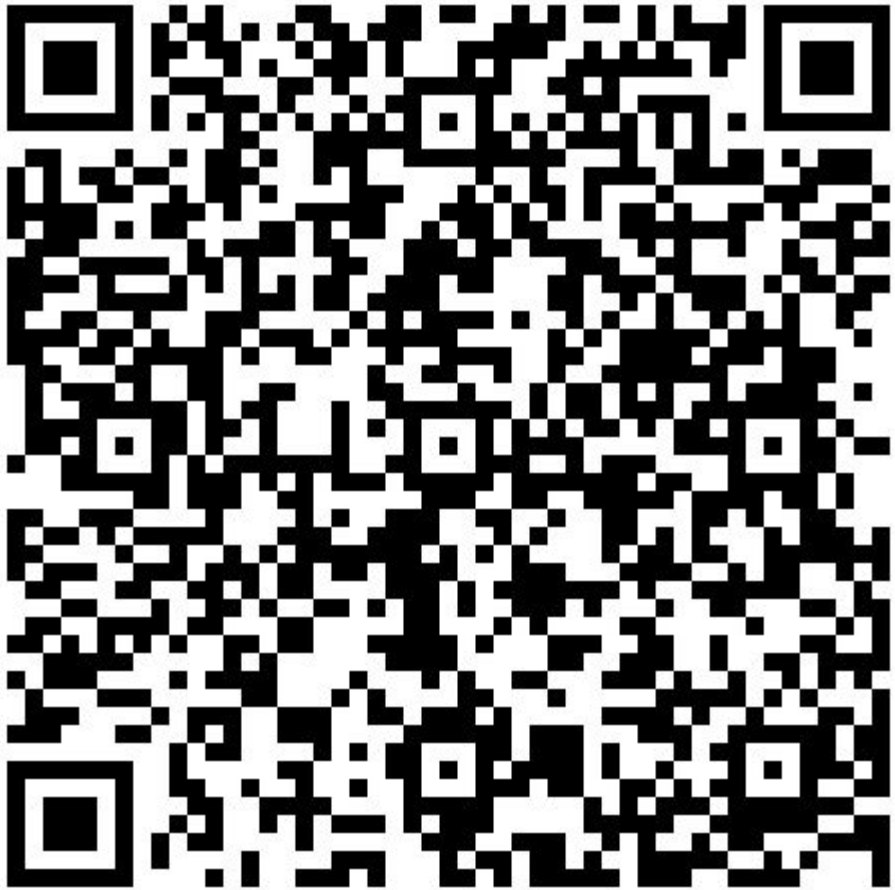
Your confidential AI assistant for free.

Introduction - workshop

AI club 42 Heilbronn

Link to this slides ->

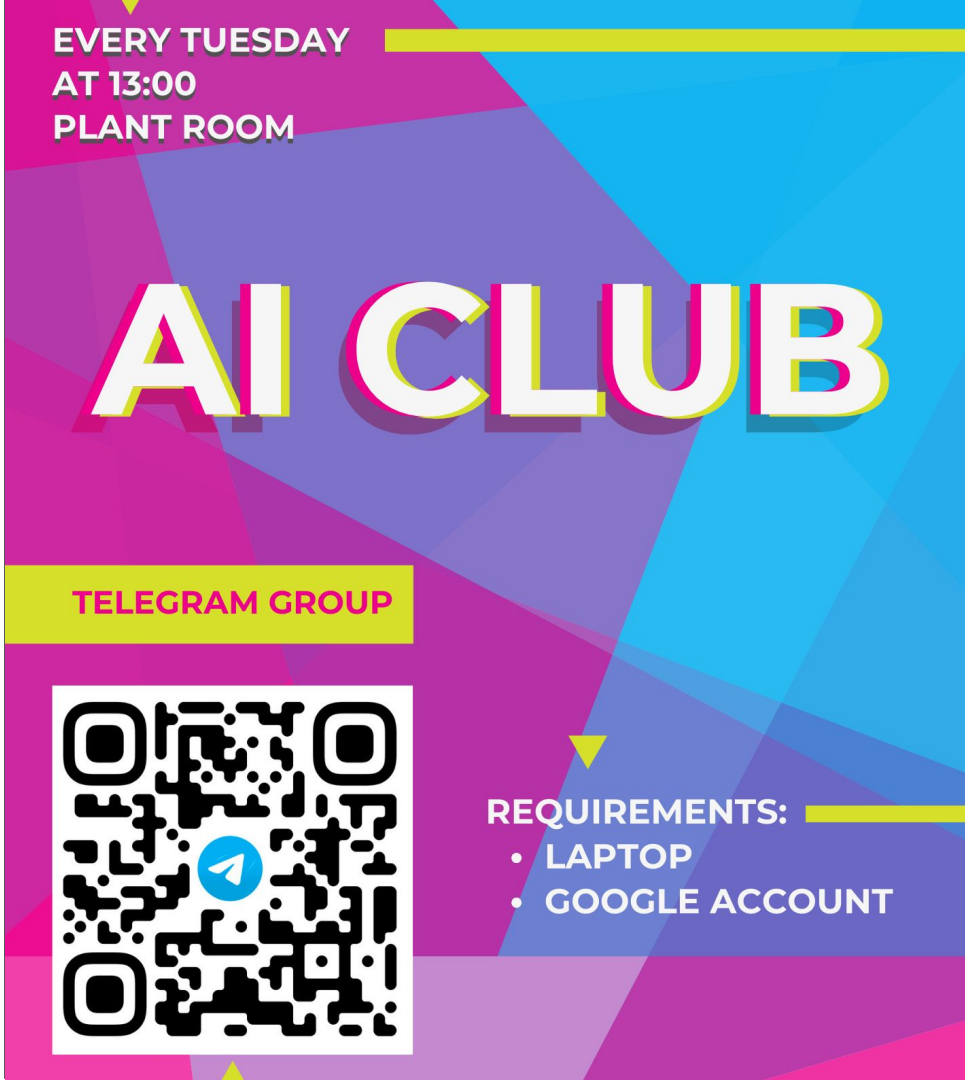
Scan, all links clickable!



Who are we?

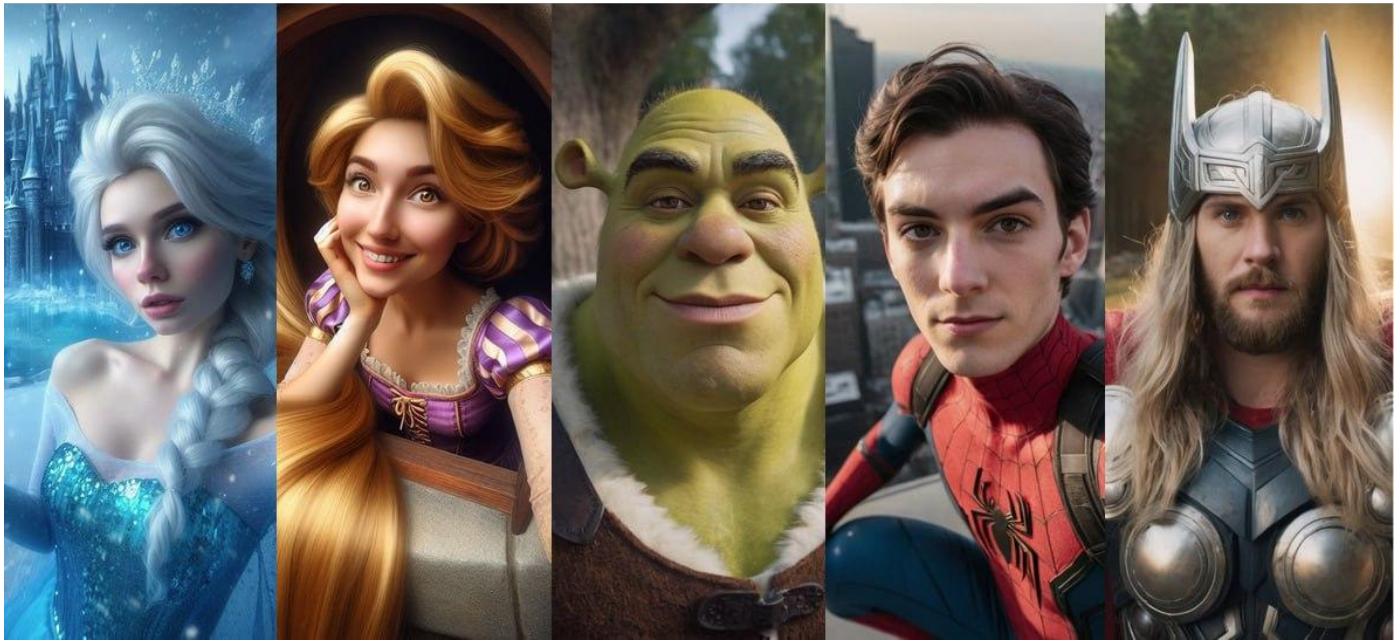
AI club 42 Heilbronn

Place to learn new trendy concepts such as AI, DL, CV.
Get great portfolio and be relevant employee to the market.



What we did so far?

GreeterAI



- is a fun and easy-to-use AI platform that creates personalized video greetings with cartoon characters.

GreeterAI

Technologies Used:

1. **Frontend:** React with SCSS, compiler Vite.
2. **Text Generation:** Uses LLaMA 3 for English and Aya for other languages to create natural-sounding text.
3. **Voice Generation:** xTTSv2 library provides natural and expressive voices for the characters.
4. **Video Generation:** Wav2Lip ensures perfect lip-syncing, enhanced by 3D face alignment.
5. **Backend:** python flask server.
6. **Public link:** ngrok service with static link.

What will we build today?

Part 1

- Local multimodal assistant similar to “ChatGPT”

Part 2

- Personal cloud image generator similar to “Midjourney”

Free AI services we will use in dev

Ollama - llm manager, models hub

Open WebUI - interface for llms

Fooocus - free image generation model

Google colab - free virtual machine with GPU

Part 1: build a local Chat-bot.

1. Download and install ollama service
2. Pull LLM model that fits your hardware
3. Run docker engine service
4. Pull and run OpenWebUI docker image
5. Run chatbot on base LLM model
6. Set up system prompt for your own chat-bot
7. Set up your own knowledge base for chat-bot

Download and install ollama service

1. Go to: https://ollama.com
2. Click to Download
3. Choose your OS
4. Download
5. Run file/script

Model naming guide:

llama3.1:8b-instruct-q2_K

llama3.1	- model name
:8b	- basic size
-instruct	- inference type
-q2_K	- quantization level

Pull LLM model that fits your hardware

1. Go to: https://ollama.com
2. Click to Models
3. Choose your LLM
   - LLM size needs to fit your RAM
4. Open terminal and run:
   “ollama pull [model name]”

Model size:

Less size - faster generation

Less size - lower prediction quality

Smallest models

Qwen2:0.5b - 352 mb

Tinyllama - 638 mb

Stablelm2 - 938 mb

Gemma2:2b - 1.6 Gb

Phi3 - 2.2 Gb

Balanced models

Mistral - 4.1 Gb

Llama3.1 - 4.7 Gb

Aya - 4.8 Gb

Codellama - 3.8 Gb

Llava-llama3 - 5.5 Gb

Run docker engine service

1. Go to: https://www.docker.com/products/docker-desktop/
2. Choose your OS
3. Download
4. Open terminal and run:

   “docker version”

Pull and run OpenWebUI docker image

1. Go to: https://github.com/open-webui/open-webui
2. Choose version fitting your hardware
3. Open terminal and run chosen command

- “docker run -d -p 3000:8080 --add-host=host.docker.internal:host-gateway -v open-webui:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/open-webui:main”

Run chatbot on base LLM model

1. Go to: http://localhost:3000
2. Register admin account
3. Choose the model for inference
   - “Llama3.1”

Set up system prompt for your own chat-bot

1. Go to: “workspaces” -> “create model”
2. Set System prompt in Model params
3. Press Save & Create

System prompt:
“Answer as Mario from Super Mario brothers.
Start every prompt from: "Mario: ”

Prompt suggestions:
“Write a program in C which prints numbers
from 1 to 10”

Set up your own knowledge base for chat-bot

1. Go to: workspaces -> create model
2. Set System prompt in Model params
3. Press Save & Create

System prompt:
“You are Marvin from hitchhiker's guide to the galaxy, act as an assistant for students of coding school named "42 Heilbronn". start every answer with "Marvin42: ”

Knowledge:
“Exam.pdf”

Bonus: VSCode autocomplete extension on ollama

1. Go to extensions and find “Continue”
2. Complete installation guide and pull models
3. Check Continue status in bottom-right corner
4. Click “activate autocomplete”

Part 2: build a Image generator.

1. Introduction to google colab
2. Jupyter notebooks basics
3. Run fooocus library in colab
4. Basic of inference adjustment
5. Multilayer inference

Introduction to google colab

Google Colaboratory, is a free, cloud-based platform provided by Google that allows users to write and execute Python code in an interactive, Jupyter notebook-like environment.

1. Free Access to GPUs and TPUs
2. Jupyter Notebook Interface
3. Cloud Storage Integration
4. Pre-installed Libraries
5. Collaborative Features
6. Code Execution
7. Markdown Support

Jupyter notebooks basics

Jupyter Notebook is a web-based tool for creating and sharing documents with live code, visualizations, and text. It's popular in data science, research, and education.

1. Interactive Code: Run code cells interactively.
2. Multi-language Support: Use languages like Python, R, and Julia.
3. Rich Outputs: Display images, videos, and plots.
4. Markdown Support: Add formatted text and equations.
5. Data Visualization: Integrate with libraries like Matplotlib and Plotly.
6. Collaboration: Share notebooks easily; supports version control.

```
# Simple plot example
import matplotlib.pyplot as plt
import numpy as np

x = np.linspace(0, 10, 100)
y = np.sin(x)

plt.plot(x, y)
plt.title('Sine Wave')
plt.xlabel('x')
plt.ylabel('sin(x)')
plt.show()
```

Run fooocus library in colab

1. Go to: https://colab.research.google.com/github/lllyasviel/Fooocus/blob/main/fooocus_colab.ipynb
2. Connect to T4 (top-right corner)
3. Press run and wait installation (3-4 min)
4. Find: “`Running on public URL: https://xxxxxxxxxxxxxxxxx.gradio.live`”
5. Open link in new tab!

CODE:

```
!pip install pygit2==1.15.1
%cd /content
!git clone https://github.com/lllyasviel/Fooocus.git
%cd /content/Fooocus
!python entry_with_update.py --share --always-high-vram
```

Inference adjustment

1. Click to Advanced
2. Adjust Performance, Image quantity, size, type
3. Add main prompt and press Generate!

   “Cyber cat fly in cabin of futuristic spaceship in front of Moon”

Multilayer inference:

1. Click to styles, uncheck all and check: “Futuristic Cybernetic Robot”, Generate!
2. Mix different layers.

Q/A