

Face Recognition Kiosk with Voice Interaction

What This Project Is

You're building a smart kiosk that can see who's standing in front of it, ask them their name, and then talk to them naturally using AI.

What It Does, Step by Step

1. A camera watches for faces.
2. When it sees someone, it checks if it already knows that face.
3. If not, it politely asks, "What's your name?"
4. The person says their name out loud.
5. The system transcribes their voice into text using speech-to-text.
6. It saves their name and face in a local database for next time.
7. Then it sends a message to an AI language model (like Google's Gemini or Qwen) to start a natural conversation.
8. The AI sends back a response, like: "Nice to meet you, Jordan. How can I help you today?"
9. That response is spoken aloud using ElevenLabs, a voice generator.
10. The system shows that same response on the screen.

How It's Built

- It's coded in Python.
- It uses a webcam and microphone.
- It runs first on a Windows PC for development.
- Once it's working, it gets moved over to a Raspberry Pi so it can run in a small box on its own.

Tech Stack

Component	Technology / Tool
-----------	-------------------

----- -----	
-------------	--

Programming Lang	Python 3.10+
------------------	--------------

Webcam Access	opencv-python
---------------	---------------

Face Detection	opencv-python (DNN Module, Caffe Model)
----------------	---

Face Recognition	Custom Feature Comparison using numpy
------------------	---------------------------------------

Speech-to-Text	faster-whisper
----------------	----------------

Text-to-Speech	ElevenLabs API
----------------	----------------

Face Recognition Kiosk with Voice Interaction

LLM (primary) | Gemini via Vertex AI
LLM (fallback) | Qwen 3 via Ollama
Storage | SQLite
Display | tkinter with opencv-python video feed
Dev OS | Windows 10/11
Deploy Target | Raspberry Pi 4+ (64-bit OS)

Project Workflow

Phase 1: Single-Camera Proof of Concept (PoC)

- Detect a face using a webcam.
- Recognize the face or ask for a name.
- Transcribe name using Faster-Whisper.
- Save face + name in SQLite database.
- Send interaction to Gemini or Qwen 3.
- Receive a reply.
- Convert it to audio using ElevenLabs.
- Speak it aloud.
- Show everything on screen.

Phase 2: Multi-Camera Upgrade (Post-PoC)

- Add multiple webcams (frontal, angled).
- Capture and process frames from all simultaneously.
- Choose best-quality image or combine angles for encoding.
- Improve recognition accuracy and reliability.
- Extend UI and session logic to handle multi-cam input.

Deliverables

- main.py
- /modules/ folder
- requirements.txt
- .env.example
- README.md
- SQLite DB with test users
- Raspberry Pi deployment instructions