# 🛰️ OMNICOM\_NEBULA

Nebula by OMNICOM — a retro-synthwave inspired network interface blending 1980s computer aesthetics with modern web technology. This project is composed of two main components:  
  
- nebula\_backend — a lightweight FastAPI-based authentication & API service  
- nebula\_frontend — a Flutter desktop/web interface featuring the animated “Nebula Login Terminal”

## ✨ Features

🖥️ Synthwave / cyber-retro visual design  
🔐 JWT-based login and registration system  
🧠 Animated “C64-style” login terminal (laser-to-moon intro)  
🎧 Integrated sound effects for boot, laser, and access feedback  
🌗 Full light/dark mode toggle  
⚙️ Modular backend API, easy to extend

## 📁 Project Structure

OMNICOM\_NEBULA/  
│  
├── nebula\_backend/ # FastAPI backend (Python)  
│ ├── main.py # Entry point  
│ ├── requirements.txt # Python dependencies  
│ └── api/ # Auth routes & logic  
│  
├── nebula\_frontend/ # Flutter UI client  
│ ├── lib/  
│ │ ├── main.dart  
│ │ ├── screens/  
│ │ └── widgets/  
│ ├── assets/ # Images and SFX  
│ └── pubspec.yaml  
│  
└── README.md

## ⚙️ Installation

1️⃣ Clone the repository:

* git clone git@github.com:lionsarmor/OMNICOM\_NEBULA.git  
  cd OMNICOM\_NEBULA

## 🧠 Backend Setup (nebula\_backend)

Requirements:  
- Python 3.10 or higher  
- pip or uv package manager

Install dependencies:  
cd nebula\_backend  
pip install -r requirements.txt

Run the backend:  
python main.py  
Backend available at: http://127.0.0.1:4000

## 🎨 Frontend Setup (nebula\_frontend)

Requirements:  
- Flutter SDK 3.19+  
Install Flutter: https://docs.flutter.dev/get-started/install

Install dependencies:  
cd nebula\_frontend  
flutter pub get

Run frontend (Linux):  
flutter run -d linux

Build for web:  
flutter build web  
flutter serve

Frontend connects to backend at: http://localhost:4000/api/

## 🔊 Sound Effects (Generate via SoX)

Install SoX:  
sudo apt install sox libsox-fmt-all -y

Generate sounds:  
cd nebula\_frontend && mkdir -p assets/sfx && \  
sox -n -r 44100 -c 1 assets/sfx/boot.wav synth 0.3 sine 440 fade q 0.01 0.25 0.05 gain -10 && \  
sox -n -r 44100 -c 1 assets/sfx/laser\_charge.wav synth 0.25 square 200-1200 fade q 0.02 0.20 0.04 gain -5 && \  
sox -n -r 44100 -c 1 assets/sfx/laser\_fire.wav synth 0.25 square 800-2000 fade q 0.01 0.20 0.04 gain -5 && \  
sox -n -r 44100 -c 1 assets/sfx/impact.wav synth 0.2 noise vol 0.2 && \  
sox -n -r 44100 -c 1 assets/sfx/access\_denied.wav synth 0.4 sawtooth 800-100 noise 0.3 mix vol 0.4

## 🚀 Quickstart Summary

git clone git@github.com:lionsarmor/OMNICOM\_NEBULA.git  
cd OMNICOM\_NEBULA  
  
# Start backend  
cd nebula\_backend  
pip install -r requirements.txt  
python main.py  
  
# Start frontend  
cd ../nebula\_frontend  
flutter pub get  
flutter run -d linux

## 🧩 Troubleshooting

Backend not reachable? Ensure main.py is running and port 4000 is free.  
Missing sounds or assets? Regenerate using SoX command.  
Flutter build errors on Linux? Run:  
sudo apt install clang cmake pkg-config libgtk-3-dev libblkid-dev  
flutter clean && flutter pub get

## 🛰️ Notes

Ensure backend and frontend are running locally on the same host.  
For deployment, update API base URL inside:  
nebula\_frontend/lib/services/api.dart  
Example:  
static const String baseUrl = 'https://nebula.omnicom.network/api';

## 🧠 Concept & Credits

“Synchronizing with Nebula Core… Access Granted.”  
  
OMNICOM\_NEBULA reimagines the late-80s cyberterminal aesthetic — merging pixel-perfect retro visuals with modern backend tech.  
  
Project Lead & Concept: James M. Weeks (lionsarmor)  
Frameworks: Flutter, FastAPI  
Theme: 1980s OMNICOM Nebula System