

Railway Deploy:

Step 01:

Make file: => name => Procfile => (o sky ander) => `web: chainlit run filename.py --host 0.0.0.0 --port $PORT`

Railway ko nhi pta hmra code kaise run huta hy os ko btay ky liye ye line likhi hy

Step 02:

Upload code on [github](#)

Step 03:

Go google railway.com => [sign in on github](#) => Dashboard => new right side => github => filename which you want to upload => select github project => .env => c/p all line => and go railway (variables) [new variables](#) then paste => deploye left side

Step 04:

Go railway deployments (active) / requirments.txt agr error aye tu likh dina

Step 05:

Link ky liye => setting => generate Domain => custom => link

OpenRouter:

Openrouter 1 boht bari library hy jahan hmy boht sary 1 jgha llm ai model mil jaty hein free or paid

Q use kry: agr ham chah rhy hein 1 hi jgha hm boht sary llm use kr lian 1 hi jgha tu is ka istmal kr skty hein. Wo API kis ki ho gi gemini ki hra model use kr skty hein

Open router hmy har model ki api use krny deta hy 1

OpenRouter ek **gateway (bridge)** hai jo tumhe **multiple AI models (OpenAI, Anthropic, Google Gemini, DeepSeek, Meta, etc.)** ko **ek hi API** se use karne deta hai.

🔑 Key Points (short & important):

- **Ek API key** → multiple models ka access.
- **Free aur paid models** dono support karta hai.
- **Rate limits handle** karta hai.
- **Compatible with OpenAI API format** → code change kam lagta hai.

Samjho ek **single remote control** jahan se tum alag-alag AI TVs (models) chala sakti ho.

★ OpenRouter kya hai?

Socho tumhare paas **ek bari library** hai jahan **bohot saare AI models** (Gemini, GPT, Claude, DeepSeek, Meta waqera) **ek hi jaga** available hain.

Tumhe alag-alag jagah jaake account banane ki zaroorat nahi, sab models **OpenRouter ke zariye** mil jate hain.

★ Kyun use karein?

1. **Ek hi API key** → tum multiple LLMs chala sakti ho.
 2. **Free + Paid models** dono available.
 3. **Code easy hota hai** kyunki OpenAI ke format jaisa hi hai.
 4. Tum decide karti ho kaunsa model use karna hai (Gemini, GPT, Claude, etc.).
-

★ API ka scene kya hai?

- Agar tum **Gemini** use karti ho, to request Gemini ko hi jati hai.
- Agar tum **GPT** use karti ho, to wo OpenAI ko jati hai.
- Lekin **tamaam requests OpenRouter ke zariye jaati hain** — yani tumhe har model ke liye alag setup karne ki zaroorat nahi.

So, OpenRouter ek traffic signal ki tarah hai:

- Tum request bhejti ho
 - Ye decide karta hai kis model ke paas bhejni hai
 - Aur jawab tumhe wapas deta hai
-

⚡ Short me:

OpenRouter = ek single gate → jahan se tum jitne bhi LLM models chaho, free aur paid, access kar sakti ho.

★ Real Life Example – OpenRouter

Socho tumhari **school library** hai:

- Library ke andar **bohot saari kitaabein** hain (Gemini, GPT, Claude, DeepSeek...).
- Tumhe alag-alag dukano (Google, OpenAI, Anthropic) jaane ki zaroorat nahi.
- Bas **ek library card (OpenRouter API key)** banwa lo.
- Phir jo bhi kitaab (model) chahiye, bas librarian ko bolo:
 - “Mujhe **Gemini** wali kitaab do.”
 - Ya “Mujhe **GPT-4** wali kitaab do.”

Librarian (OpenRouter) tumhara card check karega aur turant sahi kitaab la dega 📖.

↪ Short me:

OpenRouter ek school library hai → ek hi card se bohot si kitaabein (LLM models) mil jati hain.

OpenRouter:

Api_key: Account => create api key right side => **.env => open router api key**

Baseurl: `base_url="https://openrouter.ai/api/v1",`

★ LiteLLM Kya Hai?

Socho tumhare paas **bohot saare AI models** hain (Gemini, GPT, Claude, DeepSeek...)

Ab in sab ke **alag-alag APIs** samajhna mushkil hota hai.

LiteLLM ek translator / helper library hai jo in sab models ko **ek jaisi zubaan** me samjhata hai.

★ Real Life Example

Socho tum ek **teacher** ho aur class me students hain:

- Koi sirf Urdu samajhta hai,
- Koi English,
- Koi Chinese.

Ab tumhe alag-alag language bolni pare to mushkil ho jata.

Tum ek **translator (LiteLLM)** rakh leti ho jo sab students ko **ek hi style** me samjhata hai.

Tum sirf **ek hi zubaan** bolti ho aur translator har student ko samjha deta hai.

★ Short Main

LiteLLM = ek bridge jo har AI model (GPT, Gemini, Claude, DeepSeek) ko same format me chala deta hai.

Tumhe alag-alag APIs ya formats yaad nahi karne padte.

★ Kya dono ek hi kaam karte hain?

☞ **Han**, dono kaam yehi karte hain ke **tum multiple AI models (GPT, Gemini, Claude, etc.) ek hi jaga se use kar sako.**

Lekin... **farq unke tareeqe ka hai.**

★ Real Life Example

◆ OpenRouter = Ek Library Building

- Socho ek **bohot badi library building** hai.
- Isme alag-alag **publishers ki kitaabein** rakhi hui hain (Gemini, GPT, Claude).
- Tum **ek hi card (API key)** banao, aur librarian se jo kitaab chahiye wo manga lo.
- Sab kitaabein ussi building me hain.

→ Matlab: **OpenRouter apna khud ka ek bada system hai jahan models rakhe hain.**

◆ LiteLLM = Ek Translator Guide

- Tum library building ke bahar ho, aur tumhe alag-alag publishers ke offices me jana hai.
- Har office ka form aur zubaan alag hai.
- Tum ek **translator (LiteLLM)** le leti ho jo tumhari baat har office ki language me translate kar deta hai.
- Ab tumhe har publisher ka format seekhne ki zaroorat nahi.

→ Matlab: **LiteLLM tumhe alag-alag APIs (OpenAI, Gemini, Anthropic) ek hi format me use karne deta hai.**

✦ Simple Farq

- **OpenRouter** → Ek **platform** hai jo tumhe models **apne server ke zariye** deta hai.
 - **LiteLLM** → Ek **library/tool** hai jo tumhari request ko alag-alag models ke APIs ke mutabiq translate karta hai.
-

⚡ Short Line:

- **OpenRouter = Library (models already andar available)**
- **LiteLLM = Translator (jo tumhe kisi bhi library / model se connect karwa deta hai)**

👉 Short main:

LiteLLM ka faida ye hai ke **tumhe lamba-chora code likhne ki zarurat nahi.**

- Ye **khud hi model ki base URL aur format set kar leta hai**,
- Tum sirf model ka naam do (jaise gemini/gpt-4)
- Aur LiteLLM sab kuch handle kar leta hai. ✓

⚡ Matlab: **LiteLLM = shortcut jahan tum code simple rakh sakti ho.**

.....