Code With Mohsin

🔥 What is Python UV ?

UV ek **fast Python package manager** hai. Yeh:

- Dependencies manage karta hai
- Virtual environment create karta hai
- Projects initialize karta hai
- CLI tools ke zariye simple aur tezi se kaam karta hai

- Bahut **fast** hai (Rust mein likha gaya hai)
- CLI se kaam asaan ho jata hai
- Separate tools (pip, venv, poetry) ki jagah ek hi tool sab kuch karta hai
- **Modern projects** ke liye best practice

🎯 Basic Concepts with Examples

1. ✅ New Project Initialize Karna

Agar aap ek **new Python project** banana chahte hain, to yeh command likhein:

```
uv init
```

Isse ek **basic project structure** ban jata hai.

---

2. 📦 Package Project Banana

Agar aapka project ek **library** ya **CLI app** banane ke liye hai (jo PyPI pe publish ho), to aap yeh use karein:

```
uv init --package example-pkg
```

Ye command:

- Ek virtual environment banayegi
- Ek proper Python package structure banayegi (`example_pkg/`, `pyproject.toml`, etc.)

---

3. ➕ Dependency Add Karna

Agar aapko koi library chahiye project mein (for example `crewai`), to aap ye likhein:

```
uv add crewai
```

Ye:

- `crewai` ko install karega
- `pyproject.toml` aur `uv.lock` files update karega
- Auto-virtual environment ke andar install karega

🚀 Scripts Run Karna

Aap `uv` ke zariye Python scripts ya CLI tools bhi efficiently chala sakte hain, aur testing structure bana sakte hain.

📚 Official Docs, CLI Help

Agar aapko kisi command ka syntax bhool jaye, use:

```
uv --help
```

Ya koi specific command:

```
uv init --help
```

📌 Remember:

> **"When working in Python always use UV as package manager."**

Yani aap jab bhi Python mein kaam kar rahe ho — projects, packages, ya dependencies — **`uv` use karo, pip nahi**.

🧪 4. Create a Virtual Environment

Inside the project folder, create a virtual environment using:

```
uv venv
```

🔍 `uv init --package explore-uv`

Ye command Python mein ek **isolated environment** banane ke liye `uv` package manager ka use karti hai, jise **Rust** mein likha gaya hai (aur kaafi fast hai). Is command ka purpose hai:

✅ Purpose of `uv init --package explore-uv`

1. **`uv init`**:

- Ye command ek **naya Python environment** banata hai (virtualenv jaisa).
- Ye automatic `pyproject.toml` file generate karta hai (standard Python project configuration file).

2. **`--package explore-uv`**:
   - Ye kehta hai ke initialize karte waqt `explore-uv` package ko install bhi kar do.
   - Iska matlab hai ke aapka virtual environment start hote hi yeh package available ho jaye.

Agar aap `OpenAI Agents SDK` ya `OpenRouter SDK` ka test project bana rahe hain, to aap ye command use karke:

- Ek clean environment bana sakte hain,
- Required packages install kar sakte hain,
- Aur dependency tracking ko `pyproject.toml` ke through manage kar sakte hain.

💡 Extra Info

- `uv` ek alternative hai tools jaise `pip`, `venv`, `pipenv`, aur `poetry` ka.
- Iska focus hai **speed** aur **simplicity**.
- Ye caching, environment creation, aur dependency resolution bahut hi fast karta hai.

🔍 OpenRouter Kya Hai?

OpenRouter aik platform hai jo aapko **50+ Large Language Models (LLMs)** ka access deta hai ek hi jagah se. Jaise ke:

- OpenAI ke GPT models
- Google ke Gemini
- Anthropic ke Claude
- Mistral, LLaMA (Open Source models)

Aapko alag alag API keys aur settings ki zarurat nahi padti — bas ek **OpenRouter API** use karo aur kisi bhi model pe switch kar lo.

🧠 OpenRouter Kaam Kaise Karta Hai?

- Ye models ko **khud host nahi** karta, balkay **proxy** ke taur pe kaam karta hai.

- Aapka request le kar wo **best provider** (like OpenAI, Google, Meta etc.) tak bhejta hai.

- Aapko sirf ek API endpoint use karna hota hai:
  `https://openrouter.ai/api/v1`

🔧 OpenAI SDK / Agents SDK Support:

- OpenRouter aur Gemini dono **OpenAI ke SDK** ya code structure ke sath kaam karte hain.

- Sirf API key aur URL change karo, aur same code use karo.

⚙️ Function Calling (Tool Calling) Kya Hai?

Yeh feature AI ko **external tools** (jaise weather API, calculator, etc.) call karne ki ability deta hai.

- Aap define karte ho tools ka function aur schema

- AI suggest karta hai ke konsa tool call hona chahiye

- Aapke code mein us suggestion ko execute kiya jata hai

🌐 User Interface (UI) + Playground:

- Aap website pe ja ke `https://openrouter.ai/chat` pe direct chat kar sakte ho multiple models ke sath.
- Aapka token usage aur history bhi wahan show hoti hai.
- Developers ke liye testing tools bhi available hain.

🆓 Free Models:

- Filhal tak OpenRouter ke paas **50 free models** hain.
- In mein se kuch 1 million token context tak support karte hain.
- Lekin yaad rahe: **Free models per din sirf 200 requests ka limit** hai — serious development ke liye thoda kam hai.

✅ Conclusion (Nateeja):

- **OpenRouter** aik **multi-model access platform** hai jo aik hi API se kai models ka access deta hai.

- Lekin free usage ka limit kam hai — to **development/test ke liye Google Gemini zyada useful hai.**

- Dono platforms **OpenAI style API** support karte hain — isliye integration asaan hai.

- OpenRouter ko aap **backup** ya **multi-model testing** ke liye rakh sakte hain.