🧠 *Purpose of the Code:*

Yeh code ek "Translation Agent" banata hai jo English se multiple languages mein translate karta hai *using Gemini API (by Google).*
Iska output AI-generated translated paragraph hota hai.

✅ *1. Import Required Modules*

```
from agents import Agent, Runner, AsyncOpenAI,
OpenAIChatCompletionsModel, RunConfig

from dotenv import load_dotenv

import os
```

- ***agents.py*** *se AI-related tools import kiye gaye hain:*
    - *Agent: Jo AI agent banata hai (jaise translator).*
    - *Runner: Jo agent ko chalata hai (run karta hai).*
    - *AsyncOpenAI: Gemini API ko OpenAI-style client banata hai.*
    - *OpenAIChatCompletionsModel: AI ka model define karta hai.*

    - *RunConfig: Configuration settings.*

- *dotenv is used to **load environment variables** (like secret API keys) from a .env file.*

- *os module se environment variables read karte hain.*

✅ 2. Load API Key from .env

```
load_dotenv()
gemini_api_key = os.getenv("GEMINI_API_KEY")
```

- *.env file se **Gemini API Key** load hoti hai.*

- *Yeh key zaroori hai AI service use karne ke liye.*

✅ 3. Error Handling

```
if not gemini_api_key:
    raise ValueError("GEMINI_API_KEY is not set...")
```

- *Agar key nahi mili toh program ruk jaata hai — yeh **security check** hai.*

✅ 4. Connect with Gemini API (via OpenAI-compatible format)

```
external_client = AsyncOpenAI(
    api_key=gemini_api_key,
    base_url="https://generativelanguage.googleapis.com/v1beta/openai/"
)
```

- Gemini ko **OpenAI jaise interface** se access karne ke liye custom client banaya gaya hai.
- base_url woh special Gemini endpoint hai jo OpenAI-style requests accept karta hai.

✅ 5. Define the Model

```
model = OpenAIChatCompletionsModel(
    model="gemini-2.0-flash",
    openai_client=external_client
)
```

- Hum **Gemini 2.0 Flash** model use kar rahe hain (fast and smart).
- Is model ka kaam translation karna hoga.

✅ 6. Configure the Run

```
config = RunConfig(
    model=model,
    model_provider=external_client,
```

```
    tracing_disabled=True
)
```

- RunConfig se agent ke liye settings set ki jaati hain.
- `tracing_disabled=True` means: performance monitoring ko disable kiya gaya hai.

✅ 7. Create the Agent

```
writer = Agent(
    name='Translator Agent',
    instructions="""You are a translation agent. Translate the given English text into Urdu, French, ..."""
)
```

- Yahaan ek **AI translator agent** banaya gaya hai.
- Instructions mein bataya gaya hai ke agent ko **English text ko multiple languages** mein translate karna hai.

✅ 8. Run the Agent with Input

```
response = Runner.run_sync(
    writer,
    input = 'Write a 1 paragraph essay on Generative AI..',
    run_config = config
```

)

- Agent ko ek **input diya gaya hai**: "Write a 1 paragraph essay on Generative AI."
- `Runner.run_sync` function us input ko Gemini model ke through run karta hai.

✅ 9. Print the Output

```
print(response.final_output)
```

- Final translated output console par show kiya jaata hai.

🧪 Output Example (Expected):

The agent might output something like:

- Urdu: "جنریٹو اے آئی کے بارے میں ایک پیراگراف..."
- French: "L'IA générative est un domaine..."
- Chinese: "生成式人工智能是一个..."

📌 Students ke liye Tips:

1. *.env file mein apni API key zaroor daalen.*

2. *Multiple languages ko explore karen — yeh **AI + Linguistics** ka zabardast fusion hai!*

3. *Agent concept ka matlab samjhein: Ek AI "worker" jo koi specific kaam karta hai (jaise translate).*