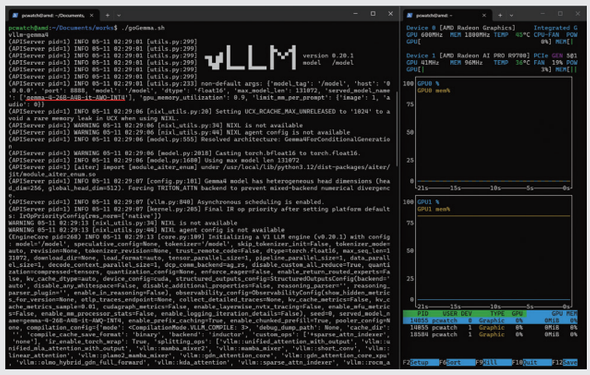
```
INFO:     Uvicorn running on http://0.0.0.0:8888
```



vLLMでgemma-4-26B-A4B-it-AWQ-4bitが起動

ステップ6｜モデル確認

```
curl http://localhost:8888/v1/models
```

ステップ7｜テキスト推論

```
curl http://localhost:8888/v1/chat/completions \
  -H "Content-Type: application/json" \
  -d '{
    "model": "gemma-4-26B-A4B-it-AWQ-INT4",
    "messages": [{"role": "user", "content": "日本語で自己紹介してください"}],
    "max_tokens": 200
  }'
```

ステップ8｜KVキャッシュ・コンテキスト長の確認

```
docker logs vllm-gemma4 2>&1 | grep -E "KV cache|kv_cache|cache blocks|model len|max_model"
```

今回の構成では以下のようなスペックとなっている。

```
Available KV cache memory: 8.78 GiB
GPU KV cache size: 372,445 tokens
Maximum concurrency for 131,072 tokens per request: 2.84x
```

Open WebUIでLAN上のPCからアクセス可能にする

ここまでの手順だけでもローカルLLMは動作するが、このままでは社内の各PCからChatGPTのようにアクセスする手段がない。そこで使うのが「Open WebUI」だ（**表6**）。

Open WebUIは、自己ホスト型（self-hosted）のオープンソースAIプラットフォームで、ChatGPTに近いUIを提供しつつ、OllamaやOpenAI互換APIなど各社LLMプロバイダに対応している。

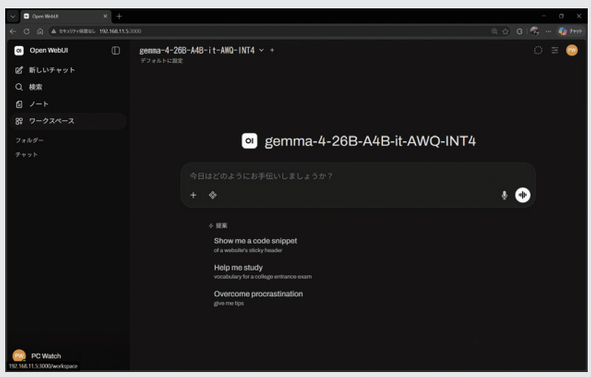
表6 Open WebUIの特徴

機能	説明
マルチモデル対応	Ollama/OpenAI/Anthropicなどの複数モデルを1つのUIで使用可能
オフライン動作	完全ローカルでの運用が可能（DockerまたはPipでインストール）
チャット機能	ファイル添付、検索、コード実行、音声入出力、マルチモデル比較
Knowledge（RAG）	ドキュメントをAIが検索して回答。9種類以上のベクトルDB対応
エージェント	カスタム設定でモデル＋ツール＋ナレッジを組み合わせ
拡張性	PythonツールやMCP、OpenAPIなどによるプラグイン拡張
チーム向け	RBAC、SSO/OIDC/LDAP、チャンネル、リアルタイム協働

起動方法（Docker使用）

```
docker run \
 -p 3000:8080 \
 -v open-webui:/app/backend/data \
 --add-host=host.docker.internal:host-gateway \
 --name open-webui \
 --restart always ghcr.io/open-webui/open-webui:main
```

起動後は「http://サーバー IPアドレス:3000」でアクセスできる。



Open WebUIの画面。チャット欄から質問すれば、ローカルLLMが回答を行なう