Playground

URL: https://main.taila597c2.ts.net/playground

Что это

Интерактивная страница, где реально пробуешь модель flavor-ml. Вбиваешь ингредиенты, жмёшь кнопку — смотришь, что система считает к ним подходящим. Сделано на Next.js (client component), вся интерактивность в браузере; страница вызывает backend через rewrite `/api/*`, который Next.js проксирует на FastAPI-сервис.

Как пользоваться

1. В поле **ingredients** введи названия через запятую (например `tomato, basil, mozzarella`).
2. Нажми **Pairing** — за ~200 мс получишь ранжированный список ингредиентов, которые модель считает дополняющими к введённым, с оценкой уверенности (0–1, чем выше тем ближе в обученном пространстве эмбеддингов).
3. Нажми **Generate** — сейчас возвращает заглушку (см. Подводные камни). Когда подключим LLM-генератор, будет полноценный рецепт: Title / Ingredients / Steps.

Кнопки независимы — запуск одной не очищает результат другой.

Что реально делает «Pairing»

Обученная модель — **Item2Vec** (алгоритм в стиле word2vec, адаптированный под со-встречаемость ингредиентов). Для каждого названия из ввода модель извлекает top-K ближайших соседей в пространстве эмбеддингов, объединяет их, выкидывает то что ты уже ввёл, и ранжирует по максимальной similarity score среди твоих входов. То есть если ты ввёл `tomato, basil`, ингредиент близкий *хотя бы к одному* поднимется наверх; ингредиент близкий *к обоим* может не подняться — это ограничение текущего scoring (max, а не joint).

Типичные значения score: - 0.95+ → очень близкие семантические соседи (например «basil leaves» рядом с «basil») - 0.85–0.95 → сильная кулинарная связь (моцарелла, пармезан с tomato + basil) - < 0.7 → слабые соседи; воспринимай как «возможно интересно», не как факт

Что реально делает «Generate» (сегодня)

Сегодня возвращает заглушку:

```
Title: Demo recipe
Steps:
1. Combine: <твои ингредиенты>.
2. Cook until done.
```

Реальный LLM-генератор **не подключён**. API-контракт уже на месте, чтобы можно было протестировать UI end-to-end до того как модель подключим. Подключить — это заменить stub в `flavor_ml/serving/api.py::generate()` на реальный вызов (vLLM-сервер, OpenAI API, локальная Llama — выбор откладывается).

Что под капотом

- Исходник страницы: `webui/app/playground/page.tsx`
- Вызовы API идут на `/api/pairing` и `/api/generate` (относительные URL)
- Next.js делает rewrite `/api/:path*` → `http://localhost:8080/:path*` на стороне сервера (`webui/next.config.js`)
- FastAPI-обработчики в `flavor_ml/serving/api.py` — собственно эндпоинты
- Модель Item2Vec подгружается лениво из `./artifacts/ingredient-embeddings.bin` при первом запросе `/pairing` (первый запрос ~500 мс дольше последующих)

Подводные камни

- **Словарь модели ограничен.** Если ввести ингредиент, которого не было в обучающем корпусе, `/pairing` вернёт ошибку из lookup модели. UI покажет её красной строкой «error».
- **Орфография имеет значение.** «tomatoes» ≠ «tomato»; «olive oil» — может быть один токен или два, в зависимости от того как корпус токенизирован (см. `flavor_ml/features/recipe_corpus.py`).
- **Никакой истории.** Каждый клик независим; результаты не сохраняются. Перезагрузишь страницу — пусто.
- **Generate — заглушка.** Не делай выводов из «Demo recipe» — это плейсхолдер до подключения LLM.
- **Нет rate limiting.** Кто угодно с URL может молотить API. Для эксперимента ок; если опубликуешь URL — нужен throttling.