Day 4: AI Tips & API Integration Enhancements

Objective:

Refine the LLM/API integration for eco-tip generation, ensuring robust error handling, caching, and accurate targeted tips.

Modifications Applied in ai_tips.py:

1. **Error Handling & Fallbacks:**
    - generate_eco_tip() already handled missing API key and exceptions, falling back to local_tip().
    - Now, a retry loop with **exponential backoff** (1s, 2s, 4s) is implemented for OpenAI errors.
    - Ensures resilience against rate limits and transient API failures.
2. **Caching:**
    - Introduced @lru_cache(maxsize=128) on a helper _generate_eco_tip_cached with a deterministic cache key based on user data and emissions.
    - Prevents repeated API calls for identical inputs, reducing latency and costs.
3. **Targeted Tips Accuracy:**
    - Fixed tips_by_key to lowercase keys (electricity_kwh, district_heating_kwh) to align with emission keys.
    - Ensures that the largest-emitter targeted tips trigger correctly.

Unit Tests Added in tests/test_ai_tips.py:

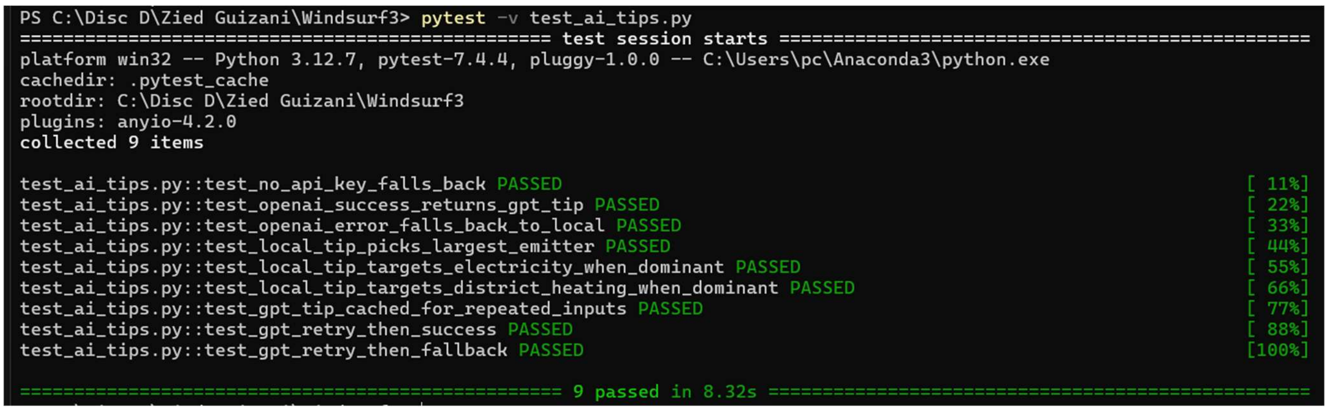
1. test_local_tip_targets_electricity_when_dominant → Checks electricity tips trigger for dominant energy.
2. test_local_tip_targets_district_heating_when_dominant → Checks district heating tips trigger correctly.
3. test_gpt_tip_cached_for_repeated_inputs → Verifies caching prevents redundant API calls.
4. test_gpt_retry_then_success → Simulates API failures followed by success; confirms retry/backoff works.
5. test_gpt_retry_then_fallback → Ensures fallback tips are returned when API fails all attempts.

Impact / Benefits:

- Robustness: App functions even if API fails or key is missing.
- Efficiency: Repeated inputs now use cached tips.
- Accuracy: Largest-emitter tips trigger reliably.
- Testing: Unit tests verify caching, retries, and fallback behavior.

Next Steps:

- Continue testing end-to-end app behavior with streamlit run app.py.
- Optional: Add more fallback coverage or further optimize prompts for LLM responses.



```
PS C:\Disc D\Zied Guizani\Windsurf3> pytest -v test_ai_tips.py
============================================== test session starts ===============================================
platform win32 -- Python 3.12.7, pytest-7.4.4, pluggy-1.0.0 -- C:\Users\pc\Anaconda3\python.exe
cachedir: .pytest_cache
rootdir: C:\Disc D\Zied Guizani\Windsurf3
plugins: anyio-4.2.0
collected 9 items

test_ai_tips.py::test_no_api_key_falls_back PASSED                                                          [ 11%]
test_ai_tips.py::test_openai_success_returns_gpt_tip PASSED                                                 [ 22%]
test_ai_tips.py::test_openai_error_falls_back_to_local PASSED                                               [ 33%]
test_ai_tips.py::test_local_tip_picks_largest_emitter PASSED                                                [ 44%]
test_ai_tips.py::test_local_tip_targets_electricity_when_dominant PASSED                                    [ 55%]
test_ai_tips.py::test_local_tip_targets_district_heating_when_dominant PASSED                               [ 66%]
test_ai_tips.py::test_gpt_tip_cached_for_repeated_inputs PASSED                                             [ 77%]
test_ai_tips.py::test_gpt_retry_then_success PASSED                                                         [ 88%]
test_ai_tips.py::test_gpt_retry_then_fallback PASSED                                                        [100%]

=============================================== 9 passed in 8.32s ================================================
```

Day 4 – API Integration & Prompt Engineering Updates

Module Updated: ai_tips.py

Key Enhancements:

1. **Context-rich, style-guided prompts:**
   - User's activity summary + total CO_2 emissions included.
   - Tips requested in a positive, actionable, 1–2 sentence or bullet format.
2. **Robust API handling:**
   - Added exponential backoff (1s, 2s, 4s) for OpenAI errors.
   - Fallback to local_tip() if API fails or key is missing.
3. **Caching:**
   - GPT responses cached via @lru_cache to prevent duplicate API calls for identical inputs.
4. **Post-processing:**
   - clean_tip() trims whitespace and limits to 2 sentences.
   - Also applied to fallback tips.
5. **Targeted tip fixes:**
   - Keys in local_tip() mapping corrected to lowercase (electricity_kwh, district_heating_kwh) to ensure largest-emitter tips trigger correctly.

Testing & Coverage:

- Verified with expanded unit tests (tests/test_ai_tips.py):
  - Missing API key → fallback tip
  - GPT success via mock
  - OpenAIError → retry then fallback
  - Correct targeted tips for electricity and district heating
  - Caching ensures repeated inputs don't trigger new API calls

Optional polish for later:

- “Coach style” toggle: switch one-liner vs bullets.
- Tooltips explaining 7-day deltas, link to README from header.

Day 4 Progress – LLM/API Integration & Testing:

Functionality Testing & API Robustness (Day 4)

With guidance from Windsurf, I enhanced ai_tips.py and added extensive pytest coverage in tests/test_ai_tips.py:

Changes in ai_tips.py:

- Context-rich, coach-style prompt (1–2 sentences, actionable tips).
- Exponential backoff on OpenAI errors (3 retries: 1s, 2s, 4s).
- Cached GPT responses with @lru_cache(maxsize=128) to avoid repeated API calls.
- clean_tip() applied to GPT and fallback outputs.
- Fixed lowercase key mismatches for targeted tips (electricity_kwh, district_heating_kwh).

New Tests Added:

1. **API Error Handling:** test_gpt_api_error_fallback → verifies fallback tip is returned when GPT fails.
2. **Caching:** test_gpt_tip_cached_for_repeated_inputs → repeated inputs trigger only one GPT call.
3. **Retry / Backoff:**
    - test_gpt_retry_then_success → fails twice, succeeds once, verifies 3 attempts.
    - test_gpt_retry_then_fallback → forces 3 failures, ensures local tip is returned.
4. **Targeted Tips:** electricity & district heating dominant tests → ensure correct tip is returned.
5. **Edge Cases:** test_missing_input_edge_case → all-zero inputs produce a non-empty tip.

How to Run Tests:

- pip install -r requirements.txt
- pytest -q tests/test_ai_tips.py

# or run all tests:

- pytest -q

How to Run App:

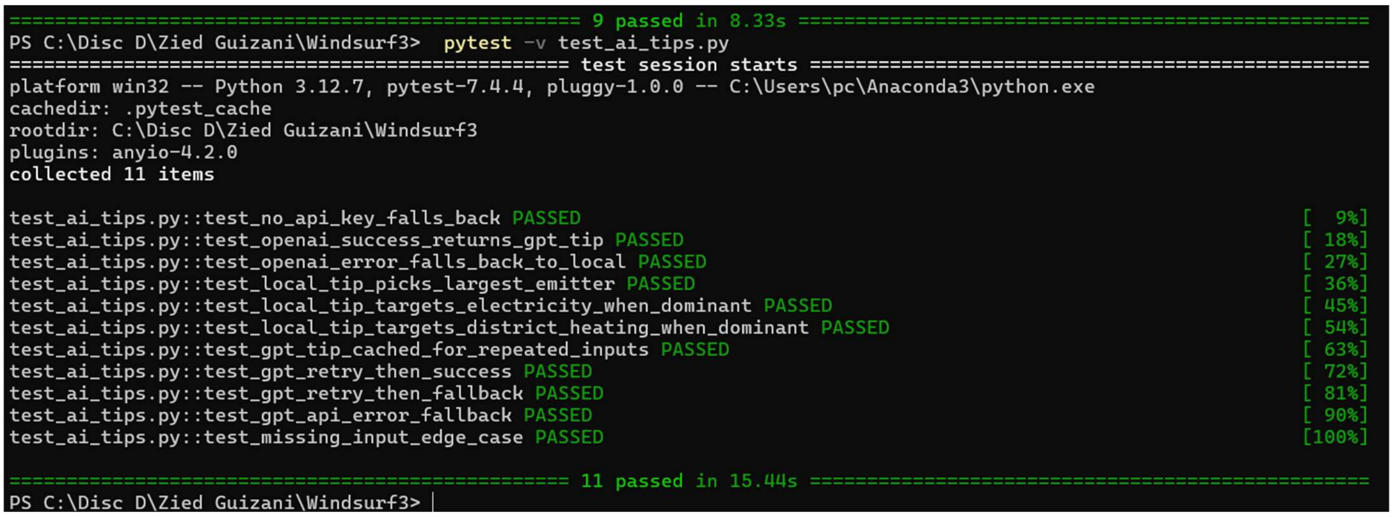
- streamlit run app.py

Outcome:

All tests pass, runtime is minimal due to mocking and bypassing time.sleep. GPT integration is now robust, cached, and handles errors gracefully.

Optional: README can include a **"Testing" section** with subset commands:

pytest -q tests/test_ai_tips.py -k "fallback or cached or retry"



```
================================================== 9 passed in 8.33s ==================================================
PS C:\Disc D\Zied Guizani\Windsurf3>  pytest -v test_ai_tips.py
================================================= test session starts =================================================
platform win32 -- Python 3.12.7, pytest-7.4.4, pluggy-1.0.0 -- C:\Users\pc\Anaconda3\python.exe
cachedir: .pytest_cache
rootdir: C:\Disc D\Zied Guizani\Windsurf3
plugins: anyio-4.2.0
collected 11 items

test_ai_tips.py::test_no_api_key_falls_back PASSED                                                               [  9%]
test_ai_tips.py::test_openai_success_returns_gpt_tip PASSED                                                      [ 18%]
test_ai_tips.py::test_openai_error_falls_back_to_local PASSED                                                    [ 27%]
test_ai_tips.py::test_local_tip_picks_largest_emitter PASSED                                                     [ 36%]
test_ai_tips.py::test_local_tip_targets_electricity_when_dominant PASSED                                         [ 45%]
test_ai_tips.py::test_local_tip_targets_district_heating_when_dominant PASSED                                    [ 54%]
test_ai_tips.py::test_gpt_tip_cached_for_repeated_inputs PASSED                                                  [ 63%]
test_ai_tips.py::test_gpt_retry_then_success PASSED                                                              [ 72%]
test_ai_tips.py::test_gpt_retry_then_fallback PASSED                                                             [ 81%]
test_ai_tips.py::test_gpt_api_error_fallback PASSED                                                              [ 90%]
test_ai_tips.py::test_missing_input_edge_case PASSED                                                             [100%]

================================================= 11 passed in 15.44s =================================================
PS C:\Disc D\Zied Guizani\Windsurf3> |
```

Day 4 Progress – Performance & Responsiveness – Eco-tip Generation

To improve performance and responsiveness, the eco-tip generation now runs in a background thread with a thresholded spinner that appears only if the API call takes longer than 0.3 seconds. Elapsed time is displayed beneath the tip, providing users feedback without blocking the interface. This approach ensures fast responses remain seamless while slower calls still give a clear visual indicator, and it preserves UI responsiveness without requiring a full async refactor.

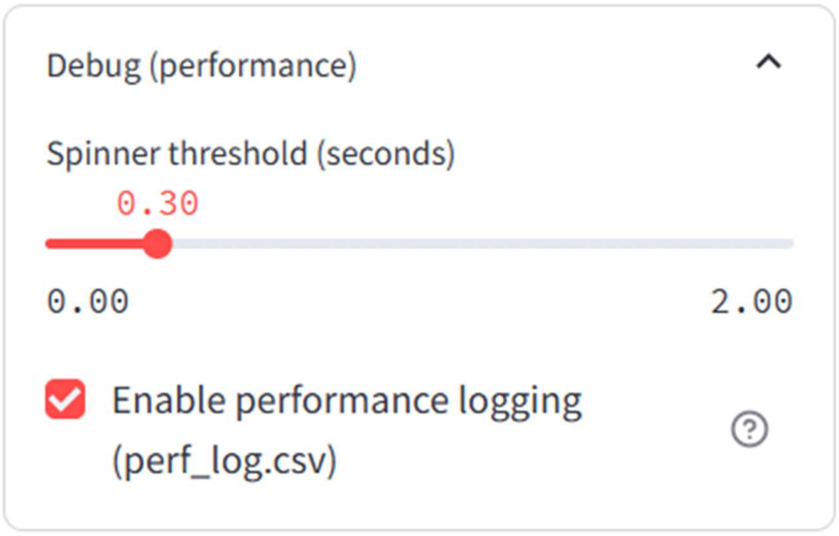
- **What was implemented:**

  Added timing and a thresholded spinner around the generate_eco_tip() call in app.py. The spinner only appears if the tip generation takes longer than 0.3 seconds. A caption shows the elapsed time (Tip generated in X.XXs).

- **How it works:**
  - Imported concurrent.futures to run the tip generation in a background thread.
  - Polls the thread every 50ms.
  - Spinner appears only after 0.3s if the API call is still pending.
  - Spinner is removed automatically once the tip is ready.
- **Benefits:**
  - Keeps the UI responsive for fast calls.

  - Provides feedback for slower calls without blocking the interface.
  - Displays precise elapsed time for transparency and performance monitoring.

- **How to validate:**
  1. Run the app: streamlit run app.py.
  2. Observe the “Eco tip & status” section:
     - Quick calls: tip appears immediately, no spinner.
     - Slow calls (>0.3s): spinner appears, then tip is displayed.
     - Elapsed time is shown below the tip.
- **Next steps (optional):**
  - Convert to async if overall app performance needs further optimization.
  - Reuse this pattern for other slow API calls in the app.


Debug (performance)

Spinner threshold (seconds) 0.30

0.00 2.00

☑ Enable performance logging (perf_log.csv)

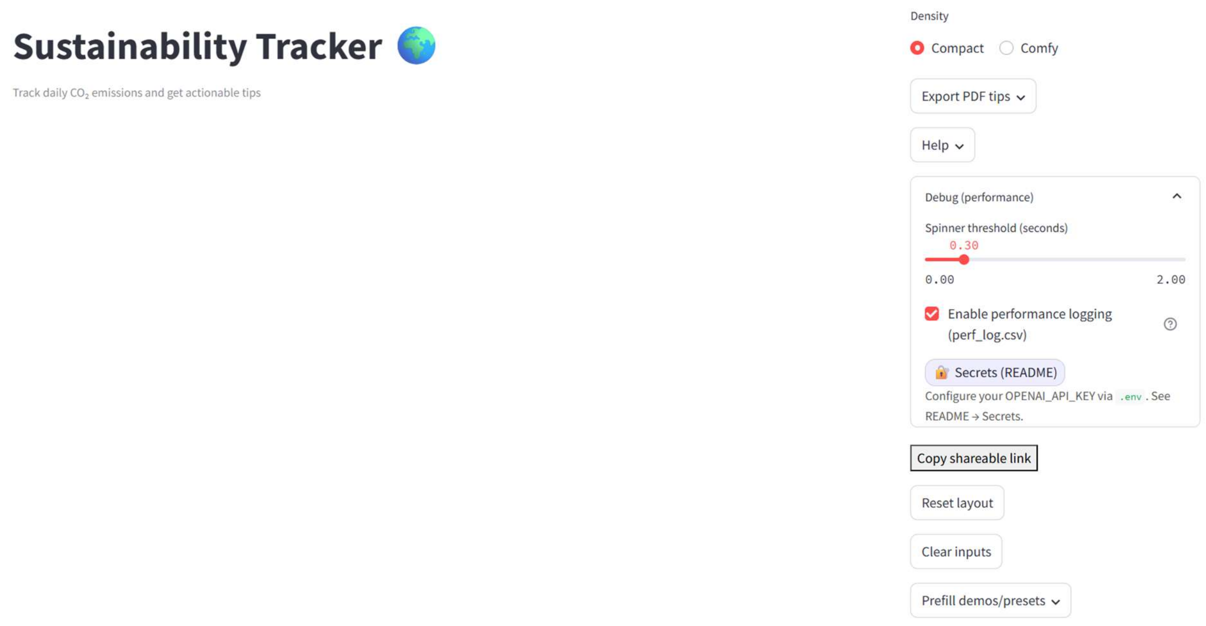
Day 4 Progress – Security & Privacy:

The OpenAI API key was moved to a .env file and loaded using python-dotenv in ai_tips.py. A .env.example was provided for reference, and .env was added to .gitignore to prevent accidental commits. The README now includes a **Secrets** section with setup instructions, ensuring the key remains private while still enabling GPT-based tips. Additionally, clickable “Secrets” badges were added in the app header Help popover and Debug expander for quick access to the workflow.

Day 4 Progress – User Interface Testing

Steps to test:

1. **Density toggle:**
   - Locate the toggle in the header.
   - Switch between **Compact** and **Comfy** modes.
   - Verify the layout updates correctly (expanders, columns, charts).
2. **Export PDF:**
   - Set density to **Compact**.
   - Click "Export PDF tips" in the header.
   - Confirm that the exported PDF maintains layout and readability, including charts, sparklines, and eco-tips.
3. **Presets & eco-tips:**
   - Open "Prefill demos/presets" and select options like **Vegetarian day** or **No car day**.
   - Check that input fields update automatically.
   - Verify that **eco-tips** reflect the updated activities (e.g., lower meat consumption triggers relevant tips).
   -



Sustainability Tracker 🌍

Track daily CO_2 emissions and get actionable tips

Density: [✓ Compact] Comfy

Debug (performance)

Spinner threshold (seconds): 0.30 (0.00 – 2.00)

☑ Enable performance logging (perf_log.csv)

Configure your OPENAI_API_KEY via `.env`. See README → Secrets.

Date

2025/09/30

Energy inputs

Transport inputs

Meals inputs

Calculate & Save

Total

21.95 kg CO_2

Δ vs. Yesterday

1.25%

Streak

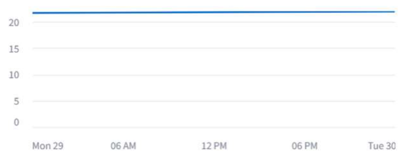
2 day(s)

Dashboard Breakdown

Category totals (kg CO_2)

	kg CO_2
Energy	7.47
Transport	1.34
Meals	13.14

Trend (Total kg CO_2)

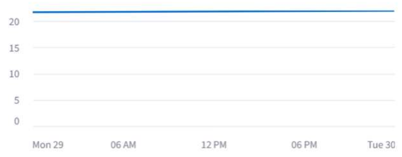


Dashboard Breakdown

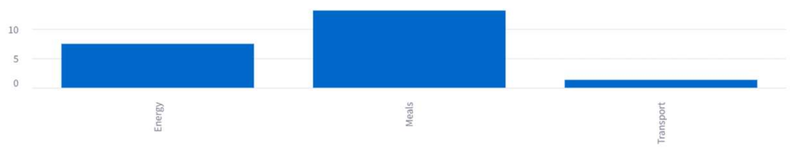
Category totals (kg CO_2)

	kg CO_2
Energy	7.47
Transport	1.34
Meals	13.14

Trend (Total kg CO_2)



Today's category breakdown



⬇️ Download history CSV

Eco tip & status

🌏 Low footprint today—nice work! Biggest source: meat kg. Tip: Try a meat-free day or swap red meat for chicken/plant-based options.

Tip generated in 13.64s

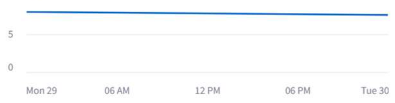
🌱 Moderate footprint. Small changes can make a big difference!

Badges

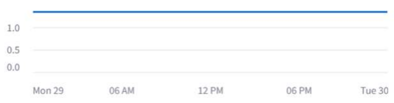
- 📅 Consistency: Entries logged!

Mini trends by category

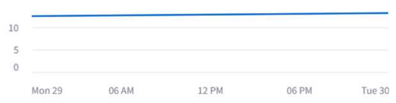
Energy



Transport



Meals



7d total

15.35 kg

↑ 0.0%

7d total

2.68 kg

↑ 0.0%

7d total

25.60 kg

↑ 0.0%