Building a Python Itinerary Generator with AI and Location Scraping

Create a dynamic travel planner using Python, AI, and web scraping.

Based on Building A Python Itinerary Generator With Ai And Location Scraping Cursor Gemini 2 5 Flash by Casey Builds It

Step 1: Define Core Features

Start by defining the essential features for the itinerary generator. This includes date selection, specifying the number of adults, kids, and dogs, setting budget preferences, and allowing for custom activity requests. Also, plan for PDF export functionality.

- Use reference material from existing itinerary features as a starting point.
- Leverage AI tools like Cursor and Gemini 2.5 Flash for conversational feature definition.

↓

Step 2: Set Up Development Environment

Prepare your development environment. This involves setting up Cursor and integrating with an AI model like Gemini 2.5 Flash to facilitate the build process.

- Consider the cost implications of different AI models for scaling.

↓

Step 3: Implement Data Scraping Strategy

Develop a strategy for dynamically pulling location-based data. Utilize free sources such as OpenStreetMap, Wikipedia, and DuckDuckGo to find real businesses, restaurants, and activities.

- Compare the effectiveness and cost of free scraping methods against paid APIs like Google Places API.

↓

Step 4: Build Core Itinerary Logic

Construct the foundational Python code for the itinerary generator. This involves integrating the defined features and the data scraping mechanism.

↓

Step 5: Test with Real Locations

Test the itinerary generator with various real-world locations to identify and address potential issues. Include diverse locations to check handling of ambiguous or duplicate place names.

- Use locations like Colorado Springs and Portland, Maine for initial testing.
- Test against multiple locations early to expose weaknesses.

↓

Step 6: Debug and Refine

Address any bugs or errors encountered during testing, particularly those related to location scraping. Refine the tool's logic and data retrieval processes.

- Document and analyze debugging sessions to understand limitations.

Step 7: Implement PDF Export

Integrate the functionality to export the generated itineraries into PDF format, making them ready for client handover or personal use.

Step 8: Explore Future Enhancements

Consider potential upgrades for the itinerary generator. This could include integrating weather APIs, GPS-based distance filtering, event APIs, and database caching for scalability.

- Caching can set the stage for turning the script into a full web application.

Resources & Further Reading

Related guides for this process:

Build a Python Itinerary Generator (Action Sheet)

https://storage.googleapis.com/casey-builds-it-checklists-20260730-ebab2c/casey-builds-it/Action%20Sheet/building-a-python-itinerary-generator-with-ai-and-location-scraping-cursor-gemin-action-sheet.pdf

Questions to Ask Before Building Your Own AI Itinerary Generator

https://storage.googleapis.com/casey-builds-it-checklists-20260727-d45d2a/casey-builds-it/Questions%20to%20Ask/building-a-python-itinerary-generator-with-ai-and-location-scraping-cursor-gemin-questions-to-ask.pdf

Casey Builds It

https://caseybuildsit.com/

Casey Builds It

https://caseybuildsit.com/the-files/

Downloads

https://caseybuildsit.com/downloads/

Casey Builds It

https://caseybuildsit.com/make-contact/

Casey Builds It | -
- | caseybuildsit.com
Combine AI, web scraping, and Python for dynamic itinerary generation.