

Building a Python Itinerary Generator with AI and Location Scraping

Create a dynamic trip 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, children, and dogs, setting budget preferences, and allowing for custom activity requests. Also, plan for PDF export functionality.

- Feed reference material from existing itinerary features into an AI tool like Cursor as an example.
- Define features conversationally with an LLM.



Step 2: Set Up the Development Environment

Prepare your development environment. This build utilizes Cursor with Google Gemini 2.5 Flash for AI-assisted coding and debugging.



Step 3: Implement Location Data Scraping

Develop the functionality to dynamically pull location-based ideas on the fly. Utilize free sources such as OpenStreetMap, Wikipedia, and DuckDuckGo to find real businesses, restaurants, and activities.

- Consider the tradeoffs between free scraping methods and paid APIs like Google Places API.



Step 4: Build the Itinerary Generation Logic

Integrate the scraped location data with user inputs (dates, party size, budget, activities) to generate a coherent trip plan.



Step 5: Test with Real Locations

Test the itinerary generator with various real-world locations to identify and address potential issues. Examples include Colorado Springs and Portland, Maine.

- Test against multiple, even ambiguous, locations early to expose weaknesses.



Step 6: Debug and Refine

Address any errors or bugs encountered during testing, particularly those related to location scraping or data accuracy. Compare different approaches, such as free scraping versus paid APIs.

- Conduct live debugging sessions to resolve issues as they arise.



Step 7: Implement PDF Export

Add the feature to export the generated itineraries into a PDF format for easy sharing and use.



Step 8: Explore Future Enhancements

Consider potential improvements such as 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-20260729-0c6048/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 planning.