

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, 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 an LLM like Gemini 2.5 Flash within Cursor for conversational feature definition.



Step 2: Set Up Development Environment

Prepare your development environment. This build utilizes Cursor IDE with an integrated LLM (Gemini 2.5 Flash) for a streamlined coding and debugging process.



Step 3: Implement Dynamic Location Data Scraping

Instead of using pre-scraped data, configure the tool to 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 trade-offs between free scraping methods and paid APIs like Google Places API.



Step 4: Develop Itinerary Generation Logic

Build the core logic that takes user inputs (dates, party size, budget, activities) and the scraped location data to generate a coherent trip itinerary.



Step 5: Test with Diverse Locations

Conduct live testing of the itinerary generator with various locations. This helps identify how the tool handles different data types, potential ambiguities in place names, and overall functionality.

- Test with locations that might have ambiguous or duplicate place names to expose weaknesses.



Step 6: Debug and Refine Scraping

Address any issues encountered during testing, particularly with the location scraping process. Debug broken scrapes and compare the effectiveness and cost of different scraping approaches.

- Compare API pricing for LLMs (e.g., Gemini Flash vs. Claude) for cost-effectiveness as the tool scales.



Step 7: Implement PDF Export

Integrate functionality to export the generated itineraries into a PDF format, making them easy to share or 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, or database caching for scalability.

- Caching previously scraped areas can set the stage for a full web application.
- Weather API integration can enable smarter day-by-day planning.

Resources & Further Reading

Related guides for this process:

■ Build a Python Itinerary Generator (Action Sheet)

<https://storage.googleapis.com/casey-builds-it-checklists-20260727-d45d2a/casey-builds-it/Action%20Sheet/building-a-python-itinerary-generator-with-ai-and-location-scraping-cursor-gemin-action-sheet.pdf>

■ Python Itinerary Generator Checklist

<https://storage.googleapis.com/casey-builds-it-checklists-20260727-d45d2a/casey-builds-it/Checklist/python-itinerary-generator-checklist.pdf>

■ Casey Builds It

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

■ Casey Builds It

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

■ Casey Builds It

<https://caseybuildsit.com/>

■ Downloads

<https://caseybuildsit.com/downloads/>

Casey Builds It | -
- | caseybuildsit.com

Scrappy, free-data approaches can yield usable AI-powered tools with iterative development.