

Python Itinerary Generator Checklist

Steps to build a dynamic itinerary planner with 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

1. Project Setup and Core Features

Define the foundational elements of your itinerary generator.

- Set up a Python environment for the project.
- Define core feature set: date selection.
- Define core feature set: party composition (adults, kids, dogs).
- Define core feature set: budget level.
- Define core feature set: custom activity requests.
- Implement PDF export functionality.
- Feed reference material from an existing itinerary feature into Cursor as an example.

2. Location Data Strategy

Plan how your tool will gather location-based information.

- Decide on a strategy for dynamic location data pulling.
- Identify free data sources: OpenStreetMap.
- Identify free data sources: Wikipedia.
- Identify free data sources: DuckDuckGo.
- Compare free scraping methods against paid options (e.g., Google Places API).
- Consider the trade-offs between free and paid API approaches.
- Plan to find real businesses, restaurants, and activities.

3. AI Integration and Tooling

Leverage AI and specific tools for development.

- Utilize Cursor for development.
- Integrate a Large Language Model (LLM) like Google Gemini 2.5 Flash.
- Define features conversationally with the LLM.
- Test features live during development.
- Debug issues in real-time.
- Compare LLM costs (e.g., Gemini Flash vs. Claude) for workload.

4. Testing and Debugging

Validate the tool's functionality with real-world scenarios.

- Test the tool with Colorado Springs as a location.

- Test the tool with Portland, Maine.
- Address issues with ambiguous or duplicate place names.
- Debug broken location scrapes.
- Evaluate the effectiveness of chosen scraping methods.
- Test against multiple, even ambiguous, locations early.

5. Future Enhancements - Core

Plan for advanced features to improve itinerary generation.

- Integrate a Weather API for smarter planning.
- Implement GPS-based distance filtering for realistic suggestions.
- Integrate Event APIs to surface local happenings.
- Consider database caching for previously scraped areas.

6. Scaling and Deployment

Consider how to evolve the tool beyond a standalone script.

- Outline steps for turning the script into a full web application.
- Plan for database caching to support web app functionality.
- Evaluate the scalability of the chosen data sources and AI models.
- Consider user input for dates, number of people, and budget.
- Ensure custom activity requests are incorporated.

Resources & Further Reading

Use these pages while working through the checklist:

■ 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
Build dynamic travel plans using AI, web scraping, and Python.