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 the core feature set for itinerary generation.
- ■ Implement date selection functionality.
- ■ Incorporate party composition input (adults, kids, dogs).
- ■ Add budget level preference input.
- ■ Include custom activity request input.
- ■ Implement PDF export functionality for finished itineraries.

2. Data Sourcing Strategy

Plan how your tool will gather location-based information.

- ■ Decide on using dynamic data pulling instead of pre-scraped content.
- ■ Identify free data sources like OpenStreetMap.
- ■ Identify free data sources like Wikipedia.
- ■ Identify free data sources like DuckDuckGo.
- ■ Plan to find real businesses, restaurants, and activities.
- ■ Consider the trade-offs between free scraping and paid APIs.

3. AI Integration and Tooling

Leverage AI and development tools for efficient building.

- ■ Utilize an LLM (e.g., Google Gemini 2.5 Flash) for feature definition.
- ■ Use Cursor for conversational feature definition and live testing.
- ■ Define features conversationally with the LLM.
- ■ Test features live during the development process.
- ■ Debug issues in real-time.

4. Initial Testing and Location Data

Test the generator with real-world locations.

- ■ Pull reference material from an existing itinerary feature as an example.
- ■ Test the tool against Colorado Springs.
- ■ Test the tool against Portland, Maine.

- Analyze how the tool handles ambiguous place names.
- Analyze how the tool handles duplicate place names.

5. Debugging and API Comparisons

Address issues and evaluate different API options.

- Debug a broken location scrape.
- Compare free scraping methods against paid options (e.g., Google Places API).
- Compare API costs (e.g., Gemini Flash vs. Claude).
- Evaluate which APIs make sense for long-term scalability.

6. Future Enhancements and Scalability

Explore options for improving and expanding the tool.

- Integrate a Weather API for smarter planning.
- Implement GPS-based distance filtering for realistic suggestions.
- Integrate Event APIs to surface local happenings.
- Implement database caching for previously scraped areas.
- Plan for turning the script into a full web application.

7. Key Takeaways and Best Practices

Summarize important lessons learned during the build.

- Recognize that free scraping sources can yield usable results with tradeoffs.
- Understand that conversational AI speeds up prototyping.
- Appreciate the value of early testing against diverse locations.
- Identify the clear path from a standalone script to a web app.
- Document decisions and debugging processes.

Resources & Further Reading

Use these pages while working through the checklist:

- 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
Build dynamic itineraries by combining AI, web scraping, and user input.