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 input for party composition (adults, kids, dogs).
- ■ Add functionality for budget level preferences.
- ■ Include a field for custom activity requests.
- ■ Implement PDF export for finished itineraries.

2. Data Sourcing Strategy

Plan how to dynamically pull location-based information.

- ■ Decide against using a fixed database of pre-scraped locations.
- ■ Plan to pull location ideas on the fly.
- ■ Identify free data sources like OpenStreetMap.
- ■ Identify free data sources like Wikipedia.
- ■ Identify free data sources like DuckDuckGo.
- ■ Configure the tool to find real businesses and restaurants.
- ■ Configure the tool to find real activities.

3. AI Integration and Tooling

Leverage AI and specific development tools for the build process.

- ■ Choose an LLM for AI assistance (e.g., Google Gemini 2.5 Flash).
- ■ Utilize an IDE with AI capabilities (e.g., Cursor).
- ■ Feed reference material from existing itinerary features into the AI.
- ■ Define features conversationally with the LLM.
- ■ Test features live during the development process.
- ■ Debug code in real time with AI assistance.

4. Location Testing and Debugging

Test the itinerary generator with diverse real-world locations.

- ■ Select a test location (e.g., Colorado Springs).

- Select a second test location (e.g., Portland, Maine).
- Test how the tool handles ambiguous place names.
- Test how the tool handles duplicate place names.
- Address issues with broken location scrapes.
- Compare free scraping methods against paid API options.

5. Cost and Performance Considerations

Evaluate the cost-effectiveness and performance of different AI models.

- Compare the cost of different LLMs (e.g., Gemini Flash vs. Claude).
- Consider API pricing for scaling beyond a personal project.
- Analyze the trade-offs between free and paid data sources.
- Assess the usability of results from free scraping methods.
- Determine if a scrappy, free-data approach is sufficient.

6. Future Enhancements and Scalability

Plan for advanced features and potential web application development.

- Integrate a Weather API for smarter planning.
- Implement GPS-based distance filtering for realistic suggestions.
- Incorporate Event APIs to surface local happenings.
- Set up database caching for previously scraped areas.
- Plan the architecture for a full web application.
- Consider user authentication for a web app.
- Develop a user interface for the web application.

7. Key Takeaways from the Build

Summarize the critical lessons learned during the development process.

- Recognize that free scraping sources can yield usable results.
- Understand the real tradeoffs when using free vs. paid APIs.
- Appreciate how conversational feature definition speeds up prototyping.
- Value the benefit of early testing against multiple locations.
- Identify weaknesses exposed by testing ambiguous locations.
- See the clear path from a standalone script to a full web app.

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 Python, AI, and live web data.