

Here's a **clear, structured, end-to-end implementation plan**, broken into practical development phases that will work both for your internship AND as a research-grade project.

This plan assumes:

- you build MVP first
- then gradually add APIs, RAG, and explainability
- each phase produces a demonstrable milestone



Phase-Wise Implementation Roadmap



Phase 1 — MVP Conversational Travel Bot (No APIs Yet)

Goal:

Get a working chatbot that:

- asks for destination + days + month
- generates a basic itinerary
- produces structured responses

This gives you something to demo early.

♦ Implementation Tasks

1 Build a simple backend (FastAPI / Flask)

- chatbot endpoint

- conversation state handling

② Design prompt for itinerary generation

- friendly tone
- structured output format

③ Implement conversation logic

- ask missing details step-by-step
- avoid robotic Q&A style

Output style (consistent structure)

- destination overview
- best travel season
- weather summary (generic)
- day-wise itinerary
- things to do
- tips & suggestions

Deliverable for Phase 1

- ✓ Working chatbot MVP
 - ✓ Can plan trips for any destination
 - ✓ Good enough to present to mentor
-
-

✓ Phase 2 — Add API-Based Real-World Data (Data Building Phase)

🎯 Goal:

Move from generic itineraries → **grounded factual content**

You will:

- call APIs
- collect raw data
- transform & store into your own dataset

This is major research value.

♦ APIs to integrate

- ✓ Open-Meteo → seasonal climate profile
 - ✓ OpenTripMap → attractions list
 - ✓ Wikivoyage / Wikipedia → descriptive text
-

♦ Implementation Tasks

① Create **data collection scripts**

For each destination:

- fetch attractions
- fetch climate summaries
- fetch descriptions

② Normalize and clean outputs

- remove duplicates

- group by attraction category
- compute basic popularity score

3 Save to local dataset

Store in:

- JSON / CSV tables
 - later into vector DB

Dataset tables produced

- destinations
- season_weather_profile
- attractions
- text_corpus (for embeddings)

Deliverable for Phase 2

- ✓ Travel knowledge dataset created
 - ✓ Reusable + cached
 - ✓ Supports research reproducibility
-

Phase 3 — RAG Knowledge Base + Grounded Responses

Goal:

Reduce hallucination + improve reliability

Move from:

- ✗ pure LLM imagination
to
✓ retrieval-guided factual reasoning
-

♦ Implementation Tasks

- 1 Chunk descriptive text (destinations + attractions)
 - 2 Generate embeddings
 - 3 Store in vector DB (FAISS / Chroma)
 - 4 Implement RAG retrieval pipeline
 - retrieve relevant text
 - pass into LLM context
 - generate itinerary using grounded info
-

🧠 Improvements achieved

- accurate attraction names
 - realistic cultural context
 - richer trip descriptions
-

🎯 Deliverable for Phase 3

- ✓ RAG-powered chatbot
 - ✓ Uses your dataset
 - ✓ Research-ready system
-
-

Phase 4 — Intelligent Itinerary Planning Logic

Goal:

Make itinerary feel **human-planned**, not random.

Add reasoning constraints:

- cluster nearby attractions per day
 - mix indoor + outdoor activities
 - adapt schedule to weather season
 - avoid redundant travel movement
-

◆ Implementation Tasks

① Compute features

- attraction category
- region cluster (distance-based)
- relative popularity

② Add heuristic planning rules

Examples:

- Day 1 → city core highlights
- Avoid 3+ museums same day
- Outdoor spots → kept for good weather
- Long trip → include rest day buffer

③ LLM receives:

- retrieved RAG context
- attraction metadata
- planning constraints

Then generates:

- optimized day-wise plan

Deliverable for Phase 4

- ✓ Smart itinerary engine
- ✓ Context-aware planning
- ✓ Experience feels curated

Phase 5 — Explainability Layer (Research Contribution)

Goal:

Make the system interpretable.

Bot should answer:

“Why did you recommend this itinerary?”

Add explanation outputs

For each day / attraction:

- distance grouping reason
- popularity justification

- weather suitability
- category balancing

Example response:

“Louvre and Orsay are grouped on Day-2 because they are nearby and both are indoor-friendly for spring weather conditions.”

Deliverable for Phase 5

- ✓ Human-readable reasoning
 - ✓ Aligns with Explainable AI research
 - ✓ Strong for academic report
-
-

Phase 6 — Agentic Interaction & Personalization

Goal:

Make chatbot adaptive & iterative.

Bot should support:

- modify trip duration
 - change preferences
 - swap attractions
 - add off-beat places
 - generate alternate budget plan
-

◆ Example follow-up flows

User:

Change trip to 3 days

Bot:

Re-optimizes itinerary efficiently

—

User:

Prefer nature over museums

Bot:

Adjusts attraction priority

—

User:

Add a nearby day trip

Bot:

Selects within distance constraints

Deliverable for Phase 6

- ✓ Adaptive agent behavior
- ✓ Feels conversational & human

Evaluation & Validation Plan

Evaluate across:

Dimension

Measure

Relevance	itinerary fits intent
Factual accuracy	grounded via RAG
Coherency	travel flow realistic
Personalization	adapts to preferences
User satisfaction	qualitative feedback

Optional research metrics:

- retrieval hit-rate
- itinerary diversity score
- explanation clarity ratings



Proposed Timeline (Safe & Practical)

Phase	Duration
MVP bot	1 week
Data + APIs	1–2 weeks
RAG KB	1 week
Smart itinerary logic	1 week
Explainability layer	3–5 days
Agentic refinement	3–5 days

Total ≈ **5–6 weeks** (very reasonable for internship)

End Result

You will have:

- ✓ functional industry-ready chatbot
- ✓ strong AI engineering pipeline
- ✓ research-grade methodology
- ✓ extendable system for future work
- ✓ portfolio + publication potential 💪