



AI Agent Developer Intern Technical Assignment

Assignment: Build Bangalore's Smart City Concierge Agent

The Scenario

The Bangalore Municipal Corporation has hired you to build an AI concierge that helps both **tourists** and **new residents** navigate the city. You have their official city guide document, but it's outdated (from 2023). Your agent needs to be smart enough to use this base knowledge while fetching current information when needed.

The Story

Bangalore is transforming into a smart city. Your AI agent will be deployed at airports, train stations, and government offices. It needs to understand who's asking - a weekend tourist or someone moving here for work - and respond appropriately.

Core Challenge

Build an AI agent using Agno that:

1. Uses the provided Bangalore City Guide (PDF) as primary knowledge
2. Identifies user persona from conversation
3. Searches the web when current info is needed
4. Adapts its communication style based on who's asking

The Two Personas

Tourist (Weekend Visitor)

- Wants: Top attractions, must-eat places, Instagram spots
- Tone: Enthusiastic, concise, highlight-focused
- Example: "Skip the traffic! Take the Purple Line metro to Cubbon Park (₹40, 20 mins). Perfect for morning walks. After, try MTR for authentic dosas - tourist favourite since 1924!"

New Resident (Moving for Work)

- Wants: Practical info, neighbourhoods, daily life tips
 - Tone: Detailed, helpful, insider knowledge
 - Example: "For IT professionals, consider Koramangala or Indiranagar - good connectivity to tech parks. Koramangala has younger crowd, more pubs. Indiranagar is slightly pricier but quieter. Both have reliable internet (ACT/Airtel fiber)."
-

Technical Implementations Expected

1) Knowledge Base Setup

- Load the knowledge base onto the agent with the appropriate setup

2) Persona Detection & Adaptation

Your agent must:

- Detect persona from conversation cues
- Adjust response style accordingly
- Remember persona throughout conversation

3) Smart Information Retrieval

Implement logic to:

- First check the PDF knowledge base
- Identify when info might be outdated (prices, events, new places)
- Search web for current information
- Blend both sources naturally

4) Prompt Engineering

Create prompts that:

- Detect user persona in a conversational style
- Generate persona-appropriate responses
- Know when to search vs. use stored knowledge
- Handle Bangalore-specific context (traffic, weather, tech culture)
- Hint - your prompts might be your key to success

Example Interactions

Tourist Conversation:

User: "What should I see in 2 days?"

Agent: [Detects tourist persona]

"Here's your perfect Bangalore weekend!

Day 1: Start with South Indian breakfast at Vidyarthi Bhavan (their masala dosa is legendary!). Head to Lalbagh Botanical Garden for stunning photos. Evening at Commercial Street for shopping and street food.

Day 2: Morning at Bangalore Palace (₹250 entry), then explore the trendy Indiranagar cafes. Don't miss the craft beer scene - Toit is a must-visit!"

[Agent notes: Used PDF for classic spots, searched current entry fees]

New Resident Conversation:

User: "Moving here next month for work at Electronic City"

Agent: [Detects new resident persona]

[Searches for current housing/commute info]

"Welcome to Bangalore! For Electronic City commute, consider:

Housing: BTM Layout or JP Nagar - 45min commute, more affordable (2BHK: ₹25-35K). HSR Layout is pricier but has better cafes/nightlife.

Commute: Company buses are lifesavers. Otherwise, book daily Uber/Ola (₹300-400) or get a two-wheeler. Metro connectivity coming in 2025.

Pro tip: Get a local SIM immediately, sign up for Dunzo/Swiggy for groceries, and join apartment WhatsApp groups - that's where the real city info flows!"

Provided Materials

1. [The Ultimate Bangalore Guide.pdf](#) - Official city guide
2. Reference Agno for agent setup: <http://docs.agno.com/>
3. Test conversations for both personas

Deliverables

1. Link to GitHub repo containing your implementation that includes
 - python files of the agent
 - [conversations/](#) - Directory with:
 - [tourist_session.txt](#) - Complete tourist interaction
 - [resident_session.txt](#) - Complete resident interaction
 - [persona_switch.txt](#) - Show persona detection/switching
 - [prompts.md](#) - Document all prompts:
 - Persona detection prompt
 - Tourist response generation
 - Resident response generation

- Web search decision logic
- **README.md** - Explain:
 - How persona detection works
 - RAG vs. web search decision tree
 - Creative features you added
 - How you handle conflicting information

Submission Guidelines

Format:

Email the Github repository link to: aditya@banza.xyz with the email subject title "AI Agents Developer Intern Technical Assignment Submission"

Tips for Success

You ARE encouraged to use AI tools (ChatGPT, Claude, Copilot, etc.) for this assignment. We want to see how effectively you can leverage modern development tools. Keep in mind, regardless of the AI tools utilised, you will be evaluated on the code quality, so it's your responsibility to ensure good quality of code. It is also your responsibility to ensure you have an understanding of the code submitted as you will be questioned upon the same in the follow-up interview call. Spend sufficient time reading and understanding the Agno documentation.

Get Started Right Way

1. **Test the PDF loading first** - Make sure you can successfully load and query the Bangalore guide before building anything else
2. **Start with obvious persona detection** - Keywords like "moving here", "weekend trip", "just visiting" before getting fancy
3. **Build for one persona first** - Get tourist working perfectly, then add resident logic

Some Pitfalls to Avoid

- Don't over-engineer persona detection - 80% accuracy is fine
 - Don't search for everything - the PDF has good stable info
 - Don't forget to blend sources naturally
 - Don't make responses too long - tourists want quick info
-

Evaluation Criteria

- Persona Adaptation
 - Accurate detection from subtle cues
 - Distinct and appropriate response styles
 - Smooth persona switching
- Information Management
 - Effective use of PDF knowledge
 - Smart web search triggers
 - Natural blending of sources
- Prompt Engineering
 - Clear, effective prompts
 - Bangalore-specific context
 - Error handling
- Creativity & Usefulness
 - Helpful responses
 - Local insights
 - Engaging personality
- Overall code quality and output results

Bonus Creativity Points

- Add Kannada phrases for common situations
- Include auto-rickshaw negotiation tips

- Create a "Bangalore survival kit" summary
 - Add weather-based recommendations
 - Include tech community meetup suggestions
-

FAQs

Q: What if I can't complete everything in 72 hours?

A: Prioritise: PDF integration → Persona detection → Basic responses → Search integration → Polish. Submit what you have with a note explaining what you would do with more time.

Q: What if the PDF doesn't load properly in my Knowledge Base?

A: Try converting to plain text first, or chunk it into smaller sections. Document any issues in your README.

Q: How accurate should persona detection be?

A: Focus on catching obvious cases first. It's better to have reliable detection for clear scenarios than complex logic that fails often.

Q: How do I blend PDF and web search content naturally?

A: Use phrases like "According to the city guide..." for PDF content and "Current information shows..." for web searches. Don't just concatenate responses.

Q: Can I use Bangalore-specific slang or Hindi/Kannada phrases?

A: Yes! Bonus points for authenticity. Just ensure main content is in English and phrases are explained.

Q: What if web search returns outdated info too?

A: Implement basic date checking if possible. Otherwise, document this limitation. Real-world agents face this challenge too.

Q: How many test conversations should I include?

A: Minimum 3 as specified (tourist, resident, another of your choice). Adding 1-2 edge cases shows thoroughness.

Q: How do I handle conflicting information between PDF and web?

A: Acknowledge it! "The city guide mentions X, but recent updates show Y." This shows good judgment.