**Title: Hotel Booking Query System - Implementation & Challenges**

**Step 1: Title Slide**

* **Title:** Hotel Booking Query System
* **Subtitle:** Implementation Choices & Challenges

**Step 2: Introduction**

* Overview of the project
* Objective: Develop an AI-powered query system for hotel booking analytics
* Key features:
  + Dataset analytics
  + Query-based responses
  + Historical chat tracking
  + UI for user interaction

**Step 3: Architecture Overview**

* **Components:**
  + Data Processing (Pandas, NumPy)
  + Analytics Computation
  + Embedding Search (ChromaDB, FAISS)
  + LLM Integration (Llama-3 API)
  + Web Interface (Gradio)
* **Flow:** User Query -> Query Processing -> Dataset Search -> LLM Response

**Step 4: Dataset Preprocessing**

* Handling missing values:
  + Filled ‘agent’ & ‘company’ with 0
  + Used mode for missing ‘country’ values
  + Converted date columns to datetime format
* Dropped incomplete records for better insights

**Step 5: Analytics Computation**

* **Key Metrics Computed:**
  + Cancellation Rate
  + Revenue Trends (Aggregated Monthly)
  + Lead Time Distribution
  + Geographical Booking Distribution
* Stored as precomputed insights for fast retrieval

**Step 6: ChromaDB Integration**

* **Why ChromaDB?**
  + Efficient vector search
  + Persistent storage of embeddings
* **Process:**
  + Convert booking data into sentence embeddings
  + Store embeddings in ChromaDB
  + Retrieve relevant records using similarity search

**Step 7: LLM Integration**

* **Why Llama-3 API?**
  + Natural language understanding
  + Handles complex queries beyond dataset analytics
* **Process:**
  + If query matches dataset analytics, return precomputed insights
  + If not, retrieve relevant bookings and send context to Llama-3
  + Display response in UI

**Step 8: Gradio UI Implementation**

* **Features:**
  + User input for questions
  + Dropdown for Analytics Reports
  + Chat history button
  + Text output for responses
* **Why Gradio?**
  + Quick deployment
  + Interactive UI without frontend complexity

**Step 9: Implementation Challenges**

* **Data Quality Issues:**
  + Missing & inconsistent data needed preprocessing
* **LLM Response Accuracy:**
  + Some responses lacked context from the dataset
* **Efficient Query Matching:**
  + ChromaDB helped, but query structure impacts relevance
* **Deployment Complexity:**
  + API key security & server hosting considerations

**Step 10: Conclusion & Future Improvements**

* **Achievements:**
  + Successfully built a working AI-powered analytics system
  + Integrated LLM & database-driven search
  + Interactive UI for better user experience
* **Future Work:**
  + Improve query matching algorithms
  + Enhance UI with filters & visualizations
  + Experiment with fine-tuned LLMs for better responses

***Thank You..***