Short Report: Implementation Choices & Challenges

Implementation Choices

1. **Data Preprocessing & Analytics:**
   - **Dataset Selection & Cleaning:**
     We used a publicly available hotel bookings dataset (cleaned_hotel_bookings.csv) to ensure the project had rich, relevant data. Preprocessing steps were applied to format dates, compute total revenue, and handle inconsistencies, ensuring the data was in a usable format for analytics.
   - **Visualization Tools:**
     The analytics dashboard leverages Python libraries like Pandas, Matplotlib, and Seaborn. These libraries were chosen for their robust data handling and visualization capabilities, enabling us to generate detailed insights such as revenue trends, cancellation rates, and user demographics.
2. **Retrieval-Augmented Question Answering (RAG):**
   - **Vector Database:**
     Pinecone was used to store pre-computed vector embeddings of the booking data. This choice was driven by its simplicity and performance for vector search, despite alternatives like FAISS or Weaviate also being viable.
   - **LLM Integration:**
     For answering user queries, we integrated an LLM (via the Gemini-pro API). This allowed us to implement a RAG-based approach where relevant data chunks are retrieved and then passed to the generative model to form coherent answers.
   - **Embedding Model:**
     The SentenceTransformer model (all-mpnet-base-v2) was selected for its strong performance on semantic similarity tasks, ensuring that the query-to-document matching was accurate.
3. **User Interface & Deployment:**
   - **Streamlit Application:**
     A Streamlit app was developed as the primary interface. It features a sidebar navigation to switch between the Analytics Dashboard and the QA Bot. This choice provided a rapid development cycle and an intuitive, interactive user interface.
   - **Modular Design:**
     The project was structured into two main components (analytics and QA), allowing clear separation of concerns and easier maintenance. This modular approach helps in future enhancements such as API development or integration of additional features.

Challenges Encountered

1. **Data Quality & Preprocessing:**
    - Handling inconsistent date formats and missing values in the dataset required careful preprocessing. Ensuring the data was clean enough for both visualization and reliable analytics was a key challenge.
2. **Integration of Heterogeneous Components:**
    - Combining diverse components such as a vector database (Pinecone), an LLM API (Gemini-pro), and a visualization dashboard (Streamlit) meant that ensuring smooth data flow and consistent response times was challenging. Coordination between embedding generation, vector search, and LLM response generation required careful tuning.
3. **Performance Optimization:**
    - While the project focuses on interactive analytics and QA, balancing response time (especially for the QA component) with the retrieval-augmented generation was a challenge. Optimizing the query embedding and retrieval process was critical to achieving acceptable latency.

Conclusion

The project successfully integrates data analytics with a retrieval-augmented QA system, demonstrating effective use of modern data science and machine learning tools. While challenges such as data preprocessing and component integration were significant, the modular design and iterative development approach enabled us to build a robust solution. Future work can extend the system with REST API endpoints, real-time data updates, and additional performance optimizations.