**Implementation Choices & Challenges**

**Implementation Choices**

1. **Framework Selection:**
   * We used **Flask** to build the API because it is lightweight and easy to integrate with machine learning models.
   * **FastAPI** was initially considered, but Flask was chosen for its simplicity and better compatibility with existing tools.
2. **Q&A System:**
   * The groq\_qa.py module handles question-answering using an external AI model.
   * We designed an endpoint (/ask) that receives a user query and returns a response.
3. **Data Analysis & Visualization:**
   * The analysis.py module generates analytics reports using **Pandas, Matplotlib, and Seaborn**.
   * Reports include **revenue trends, cancellation rates, booking country distribution, and lead time analysis**.
   * Flask routes return base64-encoded images, but later, we modified them to directly show the plots.
4. **Performance Evaluation:**
   * The performance.py module measures **API response time** and **answer accuracy** to optimize the system.
   * This helps identify slow responses and improve retrieval speed.

**Challenges & Solutions**

1. **Matplotlib GUI Error:**
   * Issue: Running plt.show() inside the Flask app caused errors in non-main threads.
   * Solution: We modified the code to return plots as images instead of opening a GUI.
2. **CUDA Execution Failure (Q&A System):**
   * Issue: When calling the Q&A model via API, it failed due to CUDA errors, even though it worked in the terminal.
   * Solution: We ensured that GPU processing was correctly allocated and used CPU fallback if necessary.
3. **Handling Large Data Files:**
   * Issue: Processing large datasets in analysis.py caused slow API responses.
   * Solution: Optimized data loading using **Pandas caching** and grouped queries to improve efficiency.
4. **Returning Images via API:**
   * Issue: Initially, the API returned base64-encoded images, which were difficult to view.
   * Solution: We updated the API to **directly render plots** in a web browser instead.

This project successfully combines **Flask, Machine Learning, and Data Analysis** to provide valuable insights. Despite some technical challenges, we implemented effective solutions to enhance performance and usability. 🚀