**Questions Answered from the group project instructions:**

Data Collection & Exploration

1. Explain whether the data needs differencing or smoothing.
   1. We used differencing to make the time series stationary, which is required for models like ARIMA. Unlike smoothing, which highlights trends, differencing removes them to stabilize the data for accurate forecasting.
2. Discuss if external factors (e.g., holidays, economic changes) might impact your dataset.
   1. **Summer Peaks:** Clear spikes in bookings between April and September, especially May-August each year. This is consistent with vacation travel during summer months.
   2. ﻿﻿**Holiday Drop-offs:** Notably lower values in January and February, aligning with post-holiday and off-season trends.
   3. ﻿﻿**Consistent Year-over-Year Pattern:** Indicates a seasonal cycle, likely driven by holidays, school breaks, and weather-related travel demand.

Time Series Decomposition

1. Compare Additive vs. Multiplicative decomposition.
   1. For our dataset, **multiplicative decomposition** **is more appropriate** because of the seasonal patterns scale with the overall trend. As total bookings increase over time, the seasonal peaks and dips also grow proportionally. This behavior aligns better with real-world hotel demand, where high-season surges are more pronounced in high-traffic years.
   2. Unlike the additive model, which assumes constant seasonal effects, the multiplicative model captures this dynamic more realistically, making it a better choice for analysis and forecasting.
2. What insights can businesses gain from these components? (E.g., does seasonality indicate predictable demand cycles?)
   1. By analyzing the trend, seasonality, and residual components of the time series, businesses can gain valuable operational insights.
      1. The **trend** shows a steady increase in bookings, indicating long-term growth and rising demand for hotel services.
      2. The **seasonal** **component** reveals predictable demand cycles, with consistent peaks during summer months (April to August), helping hotels plan for staffing, pricing, and marketing.
      3. The **residuals** highlight unexpected variations, such as sudden drops or spikes, which may signal external disruptions or operational inefficiencies.

Together, these components **support better forecasting**, resource allocation, and strategic decision-making.

Forecasting Models & Performance Evaluation

1. Which model works best for short-term vs. long-term forecasting?
   1. **For short-term forecasting**, ETS and LSTM are the most suitable models.
      * ETS works well when demand patterns follow consistent, seasonal cycles, making it ideal for month-to-month planning or staffing decisions.
      * LSTM also performs well in the short term, especially when modeling complex, nonlinear relationships.

For **long-term forecasting**, Prophet is the best option.

* + - It automatically captures trend changes, seasonal effects, and special calendar events.

In this project, **Prophet** achieved the lowest forecast error, making it the most reliable for strategic, high-level planning.

1. How would your choice of model change for different industries (e.g., finance vs. healthcare)?
   1. In hospitality and tourism, models like Prophet and ETS are preferred.
      * These industries often exhibit strong seasonality, and both models handle it well.
      * Prophet is better for long-term demand trends, while ETS is effective for short-term operations.

In finance, LSTM is often more appropriate due to the market's nonlinear and volatile nature.

* + - ARIMA can still be used in more stable, trend-based financial forecasting.

For healthcare, both ETS and Prophet work well.

* + - Healthcare data often follows recurring patterns (e.g., seasonal illnesses), and these models can adapt to that with minimal tuning.

In retail or e-commerce, a hybrid approach works best.

* + - Use Prophet to capture predictable demand spikes during holidays and promotions, and LSTM for real-time or event-driven forecasting.

Deploy the app using Streamlit Cloud:

[https://python-eyvn5awzbjn3zmbdcbmzac.streamlit.app](https://python-eyvn5awzbjn3zmbdcbmzac.streamlit.app/)