**Communication Assistant Bot – Architecture & Approach**

**1**. **Architecture**

**Frontend**:

* Built with Streamlit for a simple web-based user interface.
* Provides text input for users to draft or upload communication samples (emails, messages, responses).
* Displays structured suggestions and final output in real-time.

**Backend:**

* Python engine processes user input.
* Implements parsing and formatting logic to handle structured responses (tone adjustments, corrections, or rewrites).
* Modularized functions for input preprocessing, template matching, and output generation.

**Data Handling:**

* Lightweight in-memory storage (lists/dicts) for session state.
* No external database – ensures faster prototyping and low complexity.

**2. Approach**

**Input Handling:**

* User provides a raw draft or selects from sample data.
* Input is cleaned and normalized.

**Processing Layer:**

* Applies rule-based formatting (grammar fixes, polite structuring).
* Context modules handle different communication modes (email, LinkedIn, formal message, casual reply).

**Output Delivery:**

* Suggestions are displayed side-by-side with the original draft.
* Users can directly copy refined text.
* Designed for quick iteration and practical usability.

**3. Key Design Choices**

* Streamlit chosen for rapid prototyping and interactive UI.
* Modular Python functions for clarity and reusability.
* Lightweight, extensible design – can be integrated with external APIs later if needed.