```
import os
import google.generativeai as genai
from loguru import logger


class GeminiModel:
    """
    Represents a GeminiModel instance for generating text based on user input.
    """

    def __init__(
        self,
        temperature: float,
        top_p: float,
        top_k: float,
    ):
        """
         Initializes the GeminiModel by setting up the API key, generation configuration, and starting a
chat session.
        Raises a KeyError if the GEMINI_API_KEY environment variable is not found.
        """
        try:
            api_key = os.environ["GEMINI_API_KEY"]
            genai.configure(api_key=api_key)
            self.generation_config = {
                "temperature": 1,
            "top_p": 0.95,
            "top_k": 40,
            "max_output_tokens": 8192,
            "response_mime_type": "text/plain",
        }
        self.model = genai.GenerativeModel(
            model_name="gemini-1.5-pro",
            generation_config=self.generation_config,
        )
        self.chat_session = self.model.start_chat(history=[])
    except KeyError as e:
        logger.error(f"Environment variable not found: {e}")
        raise

def run(self, task: str) -> str:
    """
    Sends a message to the chat session and returns the response text.
    Raises an Exception if there's an error running the GeminiModel.

    Args:
        task (str): The input task or message to send to the chat session.

    Returns:
        str: The response text from the chat session.
    """
    try:
            response = self.chat_session.send_message(task)

            return response.text

        except Exception as e:

            logger.error(f"Error running GeminiModel: {e}")

            raise



# Example usage

if __name__ == "__main__":

    gemini_model = GeminiModel()

    output = gemini_model.run("INSERT_INPUT_HERE")

    print(output)
```