```python
import pandas as pd
import json
from loguru import logger


def dict_to_dataframe(data: dict) -> pd.DataFrame:
    """
    Converts a dictionary into a Pandas DataFrame with formatted values.
    Handles non-serializable values gracefully by skipping them.

    Args:
        data (dict): The dictionary to convert.

    Returns:
        pd.DataFrame: A DataFrame representation of the dictionary.
    """
    formatted_data = {}

    for key, value in data.items():
        try:
            # Attempt to serialize the value
            if isinstance(value, list):
                # Format list as comma-separated string
                formatted_value = ", ".join(
                    str(item) for item in value
                )
            elif isinstance(value, dict):
                # Format dict as key-value pairs
                formatted_value = ", ".join(
                    f"{k}: {v}" for k, v in value.items()
                )
            else:
                # Convert other serializable types to string
                formatted_value = json.dumps(
                    value
                )  # Serialize value to string

            formatted_data[key] = formatted_value
        except (TypeError, ValueError) as e:
            # Log and skip non-serializable items
            logger.warning(
                f"Skipping non-serializable key '{key}': {e}"
            )
            continue

    # Convert the formatted dictionary into a DataFrame
    return pd.DataFrame(
        list(formatted_data.items()), columns=["Key", "Value"]
    )


example = dict_to_dataframe(data={"chicken": "noodle_soup"})
```

```
# formatter.print_panel(example)

print(example)
```