DAY – 84

2 December 2025

Tusi call kar rahe ho:

```
get_response(user_message, user_lang, history, last_topic)
```

Par chatbot_logic.py vich function signature hai:

```
def get_response(msg, user_lang="english", history=None, last_topic_in=None):
```

Matched nahi → Python error → front-end “Error fetching response”.

100% WORKING FIX (FINAL)

Step 1: Update only ONE line in app.py

Replace this

```
answer_text, updated_topic = get_response(user_message, user_lang, history, last_topic)
```

With this

```
answer_text, updated_topic = get_response(
    msg=user_message,
    user_lang=user_lang,
    history=history,
    last_topic_in=last_topic
)
```

Arguments keyword-based hone zaroori ne (last_topic_in=).

Step 2: Restart server (very important)

```
CTRL + C
python app.py
```

If you don’t restart, old code hi run hunda rahega → same error.

Step 3: Test locally

Ask inside chatbot:

what is chakbandi

Now response directly return hovega — **error message nahi**.

Optional safety (recommended for demo)

Temporarily disable LLM fallback → error kabhi nahi aayega:

In .env:

```
GROQ_API_KEY=
```

Then restart:

```
python app.py
```

Why error still aa reha even after fixes?

Because **Python backend crash ho reha hai** but **front-end only shows a generic error**.

If you look at terminal when user sends message, you will see something like:

```
TypeError: get_response() got an unexpected keyword argument 'last_topic'
```

ya

TypeError: too many positional arguments

Yahi backend crash → front-end error.

Jadon tusi last_topic_in= likh doge, crash band hojuga → response sahi.

Guarantee

If you apply just this one line change, then:

Before	After
"Error fetching response "	Bot reply properly
Question context break	Context memory works
Chakbandi vs Proforma wrong	Correct topic switching
Project stuck	Fully working & stable