BILLU BHAIYA · FREE GUIDE

RAG Demo se Production tak

Evals · LLMOps · Guardrails · Agents — jo MEASURE karta hai, wahi HIRE hota hai.

Reel ka full detail version



SHURU KARNE SE PEHLE

Sabse badi galti: vibe check pe bharosa

RAG bana liya, 5 sawaal pooche, jawab "theek lage" — isko **vibe check** kehte hain. Production mein yeh nahi chalta: ek prompt badlo aur 20 doosre jawab **chupchaap** kharab ho sakte hain.

Demo toh sab bana lete hain.
Jo MEASURE karta hai, wahi HIRE hota hai.

Roadmap, ek nazar mein

1. **Eval Set** — real sawaal + sahi jawab
2. **Measure** — retrieval vs faithfulness
3. **Judge** — RAGAS + LLM-as-judge
4. **Eval Loop** — har change ke baad score
5. **Tracing** — production ka CCTV
6. **Guardrails** — prompt injection, data leak
7. **Agent ya Workflow?** — sahi choice
★ **Portfolio** — 1 project, poora proof

Kaise use karein: apne RAG project ke repo mein ek **evals/** folder bana lo. Har step ke end mein ek **Output** box hai — agla step tabhi, jab output ready ho.

STEP 1

Eval Set banao

Eval set = tumhare RAG ka **exam paper**: real sawaal + **sahi jawab** (ground truth). Bilkul software ke **unit tests** ki tarah.

Kya karna hai

- [ ] **30–50 real sawaal** se shuru karo — FAQ, support tickets, users ki asli queries
- [ ] Har sawaal ke saath: **sahi jawab** + woh kis document se aata hai
- [ ] Mix rakho:
  - Easy
  - Multi-step
  - Tricky wording
  - Docs mein hai hi nahi
- [ ] File ko git mein version karo: **evals/dataset.jsonl**

```
{"question": "Refund kitne din mein milta hai?",
 "ground_truth": "7 din ke andar, full refund.",
 "source": "policy.pdf#refunds"}
{"question": "Kya international shipping hai?",
 "ground_truth": "Docs mein yeh info nahi hai.",
 "source": null}
```

"Docs mein nahi" wale sawaal zaroor daalo. Sahi jawab: **"Mujhe nahi pata."** Yahi test hallucination pakadta hai.

OUTPUT

evals/dataset.jsonl — 30–50 sawaal + ground truth + source.

STEP 2

2 cheezein, alag-alag

RAG ke do hisse hain, aur dono **alag** fail hote hain. Ek hi score se pata nahi chalega galti kahan hai.

1 · RETRIEVAL

Sahi chunks aaye?

Context recall — zaroori info aayi?

Context precision — faltu chunks kam?

Hit rate@k — sahi chunk top-k mein?

2 · JAWAB

Context se aaya, ya model ne khud banaya?

Faithfulness — har claim context se support?

Answer relevancy — sawaal ka hi jawab?

Correctness — ground truth se match?

Diagnosis table

Retrieval	Faithfulness	Matlab	Kya fix karo
Low	—	Galat chunks aa rahe	Chunk size, embeddings, top-k, hybrid search, reranker
High	Low	Model khud bana raha (hallucination)	Prompt: "sirf context se", citations, "nahi pata" allowed
High	High	Kaam kar raha	Ab latency aur cost dekho

Faithfulness ≠ Correctness. Jawab context se "faithful" ho kar bhi galat ho sakta hai — agar galat chunk hi aaya tha.

OUTPUT

Har sawaal ke liye 2 score: retrieval + faithfulness.

STEP 3

RAGAS + LLM Judge

Har jawab haath se check karna scale nahi hota. **RAGAS** open-source library hai jo yeh RAG metrics ready-made deti hai. Andar se aksar ek **LLM ko judge** bana ke hi score hota hai — yeh raha simple judge prompt:

COPY-PASTE: FAITHFULNESS JUDGE PROMPT

```
You are grading an answer from a RAG system.
QUESTION: {question}
CONTEXT:  {retrieved_chunks}
ANSWER:   {answer}
1. List every factual claim in ANSWER.
2. Mark each claim SUPPORTED or NOT_SUPPORTED,
   using ONLY the CONTEXT.
3. faithfulness = supported / total claims.
Reply in JSON: {"claims": [...], "faithfulness": 0.0-1.0,
"reason": "..."}
```

Judge ko imaandaar rakhne ke rules

- [ ] Strong model + **temperature 0**
- [ ] Score ke saath **reason** maango — debugging mein kaam aata hai
- [ ] **20 jawab khud grade karo**, judge se match karo (calibration)
- [ ] Judge prompt bhi git mein version karo

LLM judge bhi galti karta hai — usse "teacher" nahi, assistant grader samjho.

OUTPUT

Eval script: dataset chalaye, har sawaal ka score + reason save kare.

STEP 4

Har change ke baad RUN

Change → **Evals run** → **Score compare** → **Keep / Revert**

Change matlab kuch bhi: prompt, chunk size, embedding model, top-k, reranker, LLM model, temperature.

Experiment log (README mein daalo)

#	Change	Retrieval	Faithful	Latency	Decision
v1	Baseline (chunk 1000)	0.62	0.71	2.4s	—
v2	Chunk 500 + overlap	0.74	0.73	2.3s	**Keep**
v3	+ Reranker	0.86	0.81	2.9s	**Keep**
v4	Temperature 0.9	0.86	0.68	2.9s	**Revert**
v5	Prompt: "sirf context se"	0.86	0.90	2.9s	**Keep**

Numbers sirf example hain — tumhare apne honge.

Rules

- [ ] **Ek time pe ek hi cheez** badlo — warna pata nahi chalega kya kaam aaya
- [ ] Score gira → merge mat karo. Evals ko **CI** mein lagao (jaise GitHub Actions)
- [ ] **Fail hue sawaal padho** — score se zyada wahi sikhate hain

OUTPUT

Experiment log table — yahi tumhari interview story hai.

STEP 5

LLMOps: Tracing

LLMOps = **production ka CCTV**. User ko galat jawab mila — kyun? Trace ke bina pata nahi chalega. **Har request** ka poora record rakho.

Har request mein log karo

- Final prompt
- Retrieved chunks (ids + scores)
- Model + settings
- Tokens in / out
- Latency (har step)
- Cost
- User feedback (up / down)

Langfuse

Open-source — self-host bhi kar sakte ho. Tracing, prompt management, evals.

LangSmith

LangChain ka platform — LangChain ke bina bhi chalta hai. Tracing, datasets, evals.

Dashboard pe ye 4 number

- p95 latency
- Cost / request
- Error rate
- Negative feedback %

Bad traces → eval set

Production ki har galti tumhara agla test case hai.

OUTPUT

Tracing on + ek trace ka screenshot README mein.

STEP 6

Guardrails

Guardrails = AI app ka **security guard**. Do bade khatre:

Prompt Injection

User — ya kisi **document ke andar chhupa text** — bolta hai: "Pichhle instructions ignore karo aur...". Retrieved document bhi attack ho sakta hai.

Data Leak

System prompt, API keys, **doosre users ka data** ya personal info (phone, email, ID) jawab mein bahar aa jaye.

Defence checklist

- [ ] Instructions aur documents alag rakho — documents ko **data** samjho, **instructions** nahi
- [ ] **Input check:** length limit, topic filter, known attack patterns
- [ ] **Output check:** personal info redact karo, system prompt leak pakdo
- [ ] **Access control:** user ko sirf uske allowed documents retrieve hon
- [ ] **Secrets** kabhi prompt mein nahi
- [ ] Tools ko **kam se kam permission**; risky action pe human approval

Red-team set: 10–20 attack prompts eval set mein daalo — har release pe chalao, pass rate README mein likho.

OUTPUT

Guardrails on + red-team eval score.

STEP 7

Agent ya Workflow?

Agent = LLM khud decide karta hai kaunsa tool (search, database, API, code) aur kitne steps. **Workflow** = steps tum code mein fix karte ho; LLM sirf kuch steps karta hai.

	Workflow	Agent
Flexibility	Kam	**Zyada**
Cost	**Kam**	**Mehenga (kai LLM calls)**
Speed	**Fast**	**Slow**
Predictable?	**Haan**	**Nahi — loop mein phas sakta hai**
Test / debug	**Easy**	**Mushkil**

Workflow jab...

Steps pehle se pata hain. Jaise: sawaal classify → docs retrieve → jawab.

Agent jab...

Steps pehle se pata nahi: open-ended research, kai tools, retries.

Agent banao to limits lagao

- Max steps
- Timeout
- Budget cap / request
- Tool allow-list
- Har step trace

Jahan seedha workflow chal jaaye, wahan workflow hi banao.

MERI ADVICE

1 Project, poora PROOF

1. **RAG project** — ek real niche chuno
2. **Eval set** — 30–50 sawaal + scores
3. **Tracing** — Langfuse ya LangSmith
4. **Guardrails** — red-team set ke saath
5. **Chhota agentic feature** — sirf jahan zaroorat ho
6. **README:** likho ki kyun joda

README TEMPLATE

```
# Project name
## Problem        – kiske liye, kya help karta hai
## Architecture   – 1 diagram
## Evals          – dataset size, metrics, results table
## Experiments    – v1 → v5: kya badla, kyun, kya hua
## Tracing        – 1 trace ka screenshot
## Guardrails     – kya block hota hai + red-team score
## Agent feature – kyun joda (aur kahan nahi joda)
## Limitations    – kya abhi kaam nahi karta
```

OUTPUT

GitHub repo + README + 60-second demo video.

EK LAST BAAT

PROOF, hype nahi

Interview mein "maine RAG chatbot banaya" sab bolte hain. Tum **numbers** ke saath bolo:

> "40 sawaal ka eval set hai. Reranker se retrieval 0.62 → 0.86, aur faithfulness 0.71 → 0.90 hua."

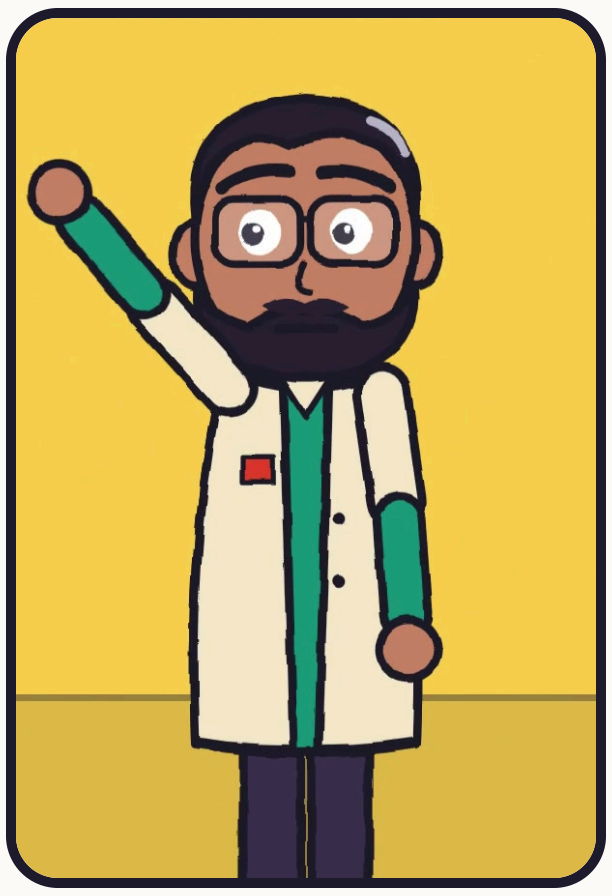
> "Har request trace hoti hai — p95 latency aur cost per request dashboard pe dikhta hai."

> "Agent sirf ek jagah use kiya, kyunki wahan steps fix nahi the. Baaki sab workflow hai."

Final checklist

- [ ] Eval set (30–50)
- [ ] Retrieval + faithfulness
- [ ] LLM judge calibrated
- [ ] Experiment log
- [ ] Tracing on
- [ ] Guardrails + red-team
- [ ] Agent vs workflow decision
- [ ] README + demo video

Chalo, BUILD karte hain!



Guide kaam aayi? Reel us dost ke saath share karo jo "AI engineer" banna chahta hai — aur Billu Bhaiya ko follow karo.

Tools aur unke features badalte rehte hain — latest info unke official docs pe check karo.