Import faiss

Import numpy as np

From sentence\_transformers import SentenceTransformer

Import json

Import datetime

Import spacy # NLP model for named entity recognition

From collections import defaultdict

# Load NLP model

Nlp = spacy.load(“en\_core\_web\_sm”)

# Load embedding model

Embed\_model = SentenceTransformer(‘all-MiniLM-L6-v2’)

# Knowledge base (can be dynamically updated)

Knowledge\_base = {entry[“question”]: entry[“answer”] for entry in [

{“id”: 1, “question”: “What is the return policy?”, “answer”: “Products can be

Returned within 30 days of delivery.”},

{“id”: 2, “question”: “Do you offer free shipping?”, “answer”: “We provide free

Shipping on all orders above $50.”},

{“id”: 3, “question”: “How to track my order?”, “answer”: “Log into your account

And click on ‘My Orders’ to track.”},

{“id”: 4, “question”: “What payment methods do you accept?”, “answer”: “We

Accept all major credit cards, UPI, and PayPal.”},

{“id”: 5, “question”: “Can I cancel my order?”, “answer”: “Yes, you can cancel

Within 12 hours of placing it.”},

{“id”: 6, “question”: “Is there a warranty on electronics?”, “answer”: “Most

Electronics come with a 1-year manufacturer warranty.”},

{“id”: 7, “question”: “Do you ship internationally?”, “answer”: “Yes, we ship to over

50 countries worldwide.”},

{“id”: 8, “question”: “Are gift cards available?”, “answer”: “Yes, you can purchase

Digital gift cards from our store.”},

{“id”: 9, “question”: “What are your customer service hours?”, “answer”: “Our

Support team is available 24/7 via chat and email.”},

{“id”: 10, “question”: “How do I apply a discount code?”, “answer”: “Enter your

Code at checkout under ‘Promo Code’.”},

{“id”: 11, “question”: “Can I change my delivery address?”, “answer”: “Yes, you can

Update your address before the order is shipped.”},

{“id”: 12, “question”: “What if I receive a damaged product?”, “answer”: “Please

Contact support within 48 hours with pictures for a replacement or refund.”},

{“id”: 13, “question”: “Can I schedule delivery?”, “answer”: “Yes, choose ‘Schedule

Delivery’ at checkout if available in your area.”},

{“id”: 14, “question”: “How long does delivery take?”, “answer”: “Standard delivery

Takes 3-5 business days. Express options are available.”},

{“id”: 15, “question”: “Do you provide installation services?”, “answer”: “Yes, for

Selected electronics and appliances, installation is available.”},

{“id”: 16, “question”: “Are there any hidden charges?”, “answer”: “No, the price at

Checkout is final. Taxes and shipping are clearly mentioned.”},

{“id”: 17, “question”: “How do I update my profile information?”, “answer”: “Log

In and go to ‘My Account’ > ‘Edit Profile’.”},

{“id”: 18, “question”: “Can I reorder previous items?”, “answer”: “Yes, go to ‘Order

History’ and click ‘Reorder’.”},

{“id”: 19, “question”: “Do you support Cash on Delivery (COD)?”, “answer”: “Yes,

COD is available on orders below $200.”},

{“id”: 20, “question”: “What is the loyalty rewards program?”, “answer”: “Earn

Points on every purchase. Redeem them for discounts and offers.”},

{“id”: 21, “question”: “How do I unsubscribe from promotional emails?”, “answer”:

“Click the ‘Unsubscribe’ link at the bottom of any promotional email.”},

{“id”: 22, “question”: “Can I gift wrap my order?”, “answer”: “Yes, gift wrap is

Available at checkout for an additional fee.”},

{“id”: 23, “question”: “Do you have a mobile app?”, “answer”: “Yes, download our

App on iOS or Android for exclusive deals.”},

{“id”: 24, “question”: “Can I contact a human agent?”, “answer”: “Yes, you can

Request a call back or chat live with our agents 24/7.”},

{“id”: 25, “question”: “How do I write a product review?”, “answer”: “After delivery,

Go to the product page and click ‘Write a Review’.”},

{“id”: 26, “question”: “Do you offer student discounts?”, “answer”: “Yes, verify your

Student ID during checkout to receive special pricing.”},

{“id”: 27, “question”: “Are there any referral benefits?”, “answer”: “Yes, refer a

Friend and earn up to $10 in credit per referral.”},

{“id”: 28, “question”: “What if my promo code isn’t working?”, “answer”: “Ensure

The code hasn’t expired and meets the minimum purchase requirement.”},

{“id”: 29, “question”: “Do you offer bulk order discounts?”, “answer”: “Yes, for bulk

Or B2B orders, contact [sales@ourstore.com](mailto:sales@ourstore.com).”},

{“id”: 30, “question”: “What is your privacy policy?”, “answer”: “We value your

Privacy. Your data is encrypted and never sold to third parties.”}

]}

# Build FAISS index

Questions = list(knowledge\_base.keys())

Question\_embeddings = embed\_model.encode(questions, convert\_to\_numpy=True)

Dimension = question\_embeddings.shape[1]

Index = faiss.IndexFlatL2(dimension)

Index.add(question\_embeddings)

# Store unknown queries for learning

Unknown\_queries = defaultdict(int)

Def retrieve\_faq(query, top\_k=1):

“””Finds the most relevant answer or marks query for learning.”””

Query\_emb = embed\_model.encode([query], convert\_to\_numpy=True)

Distances, indices = index.search(query\_emb, top\_k)

If indices[0][0] < len(questions):

Return knowledge\_base[questions[indices[0][0]]]

Else:

Unknown\_queries[query] += 1 # Track unanswered questions

Return “I’m sorry, I don’t have an answer for that. I’ll learn from this!”

Def extract\_entities(user\_input):

“””Extracts entities like product names, locations, or requests using NLP.”””

Doc = nlp(user\_input)

Entities = [ent.text for ent in doc.ents]

Return entities if entities else None

Def generate\_response(user\_input):

“””Generate chatbot response while learning from user interactions.”””

Entities = extract\_entities(user\_input)

If entities:

Print(f”Recognized entities: {entities}”) # Can be used for recommendations

Return retrieve\_faq(user\_input)

def log\_conversation(user\_input, bot\_response):

"""Log interactions for continuous learning."""

log\_entry = {

"timestamp": datetime.datetime.now().isoformat(),

"user\_input": user\_input,

"bot\_response": bot\_response

}

with open("chat\_logs.json", "a") as f:

json.dump(log\_entry, f, indent=2)

def chat():

print("Smart E-commerce Chatbot (type 'quit' to stop)")

while True:

user\_input = input("You: ").strip()

if user\_input.lower() == "quit":

break

response = generate\_response(user\_input)

print("Bot:", response)

log\_conversation(user\_input, response)

if \_name\_ == "\_main\_":

chat()