```python
import re
import time
import random
import sys
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from fuzzywuzzy import fuzz

# ———————— CONFIGURATIONS ———————— #

synonyms = {
    "mobile number": "phone_number",
    "phone number": "phone_number",
    "contact number": "phone_number",
    "number": "phone_number",
    "refund": "refund_process",
    "order": "order_details",
    "delivery address": "change_address",
    "address": "change_address",
    "phone no": "change_phone_number",
    "manage": "manage_orders"
}

smalltalk_responses = {
    "how are you": "I'm doing great! Thanks for asking. How can I assist you today? 😊",
    "thank you": "You're very welcome! Always happy to help. 🙏",
    "thanks": "My pleasure! If you need anything else, just ask! 🙌",
    "who are you": "I am your E-Shop Support Assistant 🤖, available 24/7 to assist you!",
    "what's your name": "You can call me E-Shop Bot 🤖!",
    "what can you do": "I can help with your orders, refunds, updates, and connect you with our support team!",
    "hello": "Hello there! How can I help you today?"
}

positive_words = ["thank", "thanks", "great", "awesome", "good", "amazing"]
negative_words = ["bad", "angry", "upset", "worst", "disappointed", "hate"]

memory = {
    "last_intent": None
}

# ———————— FUNCTIONS ———————— #

def normalize_text(text):
    for word, replacement in synonyms.items():
        text = text.replace(word, replacement)
    return text.lower()

def preprocess_text(text):
    text = normalize_text(text)
```

```
    text = re.sub(r'\s+', ' ', text)
    text = re.sub(r'\W', ' ', text)
    return text.strip()

def detect_sentiment(text):
    text = text.lower()
    if any(word in text for word in negative_words):
        return "negative"
    elif any(word in text for word in positive_words):
        return "positive"
    return "neutral"

def simulate_typing():
    duration = random.uniform(1.5, 2.5)
    start = time.time()
    dot_count = 0
    sys.stdout.write("\n🤖 Typing")
    sys.stdout.flush()
    while time.time() - start < duration:
        dot_count = (dot_count + 1) % 4
        sys.stdout.write('\r' + "🤖 Typing" + '.' * dot_count + ' ' * (3 - dot_count))
        sys.stdout.flush()
        time.sleep(0.5)
    print("\r", end="")

def predict_intent(user_text):
    preprocessed = preprocess_text(user_text)
    if preprocessed in smalltalk_responses:
        return 'smalltalk', smalltalk_responses[preprocessed]

    text_vector = vectorizer.transform([preprocessed])
    intent_encoded = classifier.predict(text_vector)
    predicted_intent = label_encoder.inverse_transform(intent_encoded)[0]

    highest_score = 0
    for example in examples:
        score = fuzz.ratio(preprocessed, preprocess_text(example))
        if score > highest_score:
            highest_score = score

    if highest_score > 70:
        return predicted_intent, None
    else:
        return 'unknown', None

def show_menu():
    options = {
        "1": "Order Details 📦",
        "2": "Refund Process 💸",
        "3": "Manage Orders 🛒",
        "4": "Change Delivery Address 🏡",
        "5": "Update Phone Number 📱",
        "6": "Talk to Support Agent 👩‍💻",
        "7": "Exit Chat 🚪"
```

```
    }
    print("\n🔹 Please select an option:")
    for key, value in options.items():
        print(f"{key}. {value}")
    return options

def dynamic_suggestions(intent):
    suggestions = {
        "order_details": ["Track your order", "Change delivery address"],
        "refund_process": ["Check refund status", "Contact refund team"],
        "manage_orders": ["Cancel an order", "Edit your order"],
        "change_address": ["Track updated address order"],
        "change_phone_number": ["Verify new number"]
    }
    extra = suggestions.get(intent, [])
    if extra:
        print("\n🔔 You might also want to:")
        for sug in extra:
            print(f" - {sug}")

# ———————— DATA SETUP ———————— #

examples = [
    "I want your phone number",
    "Can I get your mobile number?",
    "Hi, how are you?",
    "I need help",
    "Exit chat",
    "I want a refund",
    "Where is my order?",
    "Change my delivery address",
    "Update my phone no",
    "Manage my orders",
    "bad service",
    "thank you",
    "who are you",
    "what can you do",
    "hello"
]

intents = [
    'ask_for_phone',
    'ask_for_phone',
    'greeting',
    'help',
    'exit',
    'refund_process',
    'order_details',
    'change_address',
    'change_phone_number',
    'manage_orders',
    'complaint',
    'thanks',
    'identity',
    'identity',
```

```
    'greeting'
]

vectorizer = CountVectorizer()
X = vectorizer.fit_transform(examples)
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(intents)

classifier = MultinomialNB()
classifier.fit(X, y)

# ———————— CHATBOT MAIN ———————— #

def chatbot():
    print("="*60)
    print("🤖 Welcome to E-Shop AI Support System (v2.0)")
    print("     Your Virtual Shopping Assistant is Online ✅")
    print("="*60)

    first_name = input("👤 Please enter your first name: ").strip()
    while not all(c.isalpha() or c.isspace() for c in first_name) or not first_name.strip():
        print("⚠️ Invalid input. Please use letters and spaces only.")
        first_name = input("👤 Please enter your first name: ").strip()

    first_name = ' '.join(word.capitalize() for word in first_name.split())

    simulate_typing()
    print(f"\n👋 Hello {first_name}! How can I help you today?")
    options = show_menu()

    while True:
        try:
            user_input = input("\n🔹 Your input: ").strip()
        except Exception as e:
            print(f"⚠️ Error: {e}")
            continue

        if not user_input:
            print("⚠️ Empty input. Please try again.")
            continue

        sentiment = detect_sentiment(user_input)
        if sentiment == "negative":
            simulate_typing()
            print("😢 I'm sorry to hear that. We'll do our best to assist you!")

        intent = None
        custom_response = None

        if user_input in options:
            intent_map = {
                "1": "order_details",
                "2": "refund_process",
```

```
                "3": "manage_orders",
                "4": "change_address",
                "5": "change_phone_number",
                "6": "help",
                "7": "exit"
            }
            intent = intent_map.get(user_input)
        else:
            intent, custom_response = predict_intent(user_input)

        simulate_typing()

        if intent == 'ask_for_phone':
            print("📞 You can reach our helpline at 1800-123-4567. 📞")
        elif intent == 'greeting':
            print(f"👋 Hello again, {first_name}!")
            show_menu()
        elif intent == 'help':
            print("💬 Connecting you to a support agent... Please wait...")
            break
        elif intent == 'refund_process':
            print("💸 Your refund request is being processed. You will hear from us soon!")
            dynamic_suggestions('refund_process')
        elif intent == 'order_details':
            print("📦 Your order is on its way! Expected delivery in 2-3 business days.")
            dynamic_suggestions('order_details')
        elif intent == 'change_address':
            print("🏡 You can update your address from 'My Profile > Address Book'.")
            dynamic_suggestions('change_address')
        elif intent == 'change_phone_number':
            print("📱 Visit 'My Profile > Edit Phone Number' to update your contact.")
            dynamic_suggestions('change_phone_number')
        elif intent == 'manage_orders':
            print("🛒 Go to 'My Orders' section to manage or cancel orders.")
            dynamic_suggestions('manage_orders')
        elif intent == 'smalltalk':
            print(custom_response)
        elif intent == 'thanks':
            print("🙏 Always happy to help! Have a wonderful day! 🌟")
        elif intent == 'identity':
            print("🤖 I am your E-Shop Virtual Assistant, available 24/7 to support you.")
        elif intent == 'exit':
            print(f"👋 Thank you for chatting, {first_name}! See you again soon. 🌟")
            break
        else:
            print("🤔 Sorry, I didn't quite get that. Please choose from the menu:")
            show_menu()

```

```
261             memory['last_intent'] = intent
262 
263 if __name__ == "__main__":
264     chatbot()
```