



## Analyst Intern – Assignment

By: Deep Dey

### Task 1

Inuit is a luxury online footwear shop. After some success selling premium shoes on their website, they have discovered that a lot of their website visitors are leaving the site without exploring the variety of shoes they offer. To solve this problem Inuit owners decided to build a chatbot on their website and they're relying on your expertise for success.

**Build a chatbot for them which does the following:**

1. Welcome the user and tell them about the products Inuit brand sells
2. Ask the user about some specifics of their choice of shoe
3. Show three videos about how Inuit shoes are made
4. Ask the user to place an order for home delivery

**Things we will be looking at:**

1. Personality and tone of the bot
2. Creative use of different chat elements like "Quick Replies", "Carousels", "Emojis", "Buttons", etc.
3. How do you help the user navigate through the chat conversation flow? (Hint: You can use buttons as the user can always visit his history.)
4. How do you onboard the user for the service and how do you conclude the service.
5. How does the bot handle a message from the user that it doesn't understand?

**Things to keep in mind:**

1. You can make the main chat flow on any tool of your choice or pick one from this page:  
<https://cui.tools/>
2. You can explore the chatbot on the Haptik.ai website for some inspiration
3. Try to make the conversation flow short, simple & personal
4. How easily can someone review your conversation flow

**Answer:**

Use this link to access the chatbot:

[Inuit Chatbot - Streamlit](#)

To create a comprehensive chatbot conversation flow for Inuit. I attempted to create this as an interactive visual flowchart that illustrates the entire user journey.

Inuit Luxury Footwear Chatbot Flow which is Interactive

Chatbot Design Rationale **Personality**

**& Tone**

- Luxury yet Approachable: Uses sophisticated language (timeless elegance, Italian craftsmanship, and many more) while maintaining warmth with emojis and friendly phrasing
- Conversational: Short, personal messages that feel like talking to a knowledgeable friend, not a sales robot
- Confident & Helpful: Assumes positive intent, guides smoothly without being pushy and desperate.

### Creative Chat Elements

- Quick Replies: For simple yes/no or common responses (e.g., "Tell me more," size ranges)
- Buttons: For important decisions (shoe types, occasions, order actions)
- Carousel: Product recommendations with images, prices, and descriptions
- Videos: Behind-the-scenes content to build trust and emotional connection
- Emojis: Strategic use to add personality without overdoing it

### Navigation Design

- Persistent Menu: Main menu always accessible via button
- Back Button: Can return to previous steps
- Progress Indicator: Visual flow shows where users are in the journey
- Fallback Options: Every dead-end offers "Main Menu," "Chat with Expert," or "Browse More"

### Onboarding & Conclusion

- Onboarding: Warm welcome + brand value proposition + immediate engagement question
- Conclusion: Confirmation + next steps + continued engagement options (tracking, browsing more)
- No Dead Ends: Always provides 3 paths forward

### Error Handling

- Fallback Message: "I didn't quite catch that! Would you like to explore our collections, speak with an expert, or return to the main menu?"
- Smart Routing: Unclear messages trigger the human handoff option
- Context Preservation: Saves user choices if they need to backtrack

I have used Python and streamlit for a chatbot interface for Inuit luxury footwear:

....

Inuit Luxury Footwear Chatbot

Built with Python & Streamlit

**Installation:**

**pip install streamlit**

**Run:**

**streamlit run chatbot.py**

....

```
import streamlit as st

from datetime import datetime

import time

# Page configuration

st.set_page_config(
    page_title="Inuit Chatbot",
    page_icon="💻",
    layout="wide"
)

# Custom CSS for styling

st.markdown("""
<style>

.main {background: linear-gradient(135deg, #1e293b 0%, #334155 50%, #1e293b 100%);}

.stButton>button {
    width: 100%;

    background: linear-gradient(90deg, #d97706 0%, #b45309 100%);

    color: white;

    border: none;

    padding: 12px;

    border-radius: 8px;

    font-weight: 600;

    transition: all 0.3s;
}

.stButton>button:hover {
    background: linear-gradient(90deg, #b45309 0%, #92400e 100%);

    transform: translateY(-2px);
}

.chat-message {
    padding: 1rem;

    border-radius: 1rem;

    margin-bottom: 1rem;
}
```

```
display: flex;
gap: 0.75rem;
}

.bot-message {
background-color: white;
border: 1px solid #e2e8f0;
}

.user-message {
background: linear-gradient(90deg, #d97706 0%, #b45309 100%);
color: white;
flex-direction: row-reverse;
}

.avatar {
width: 40px;
height: 40px;
border-radius: 50%;
display: flex;
align-items: center;
justify-content: center;
font-size: 20px;
flex-shrink: 0;
}

.bot-avatar {background-color: #fef3c7;}
.user-avatar {background-color: #334155; }

.quick-reply {
display: inline-block;
padding: 8px 16px;
margin: 4px;
background-color: #fef3c7;
color: #b45309;
border: 1px solid #fbbbf24;
border-radius: 20px;
font-size: 14px;
cursor: pointer;
}
```

```
.product-card {  
background: linear-gradient(135deg, #f8fafc 0%, #f1f5f9 100%);  
padding: 1rem;  
border-radius: 10px;  
border: 1px solid #e2e8f0;  
margin: 8px 0;  
}  
  
.video-item {  
background-color: #f1f5f9;  
padding: 12px;  
border-radius: 8px;  
margin: 6px 0;  
cursor: pointer;  
}  
  
.video-item:hover {background-color: #e2e8f0;}  
</style>  
"""", unsafe_allow_html=True)
```

```
# Initialize session state  
  
if 'chat_history' not in st.session_state:  
    st.session_state.chat_history = []  
  
if 'current_step' not in st.session_state:  
    st.session_state.current_step = 0  
  
if 'user_choices' not in st.session_state:  
    st.session_state.user_choices = {  
        'shoe_type': '',  
        'occasion': '',  
        'size': ''  
    }  
  
if 'initialized' not in st.session_state:  
    st.session_state.initialized = False
```

```
# Conversation steps
```

```
STEPS = [  
    {
```

```
'id': 'welcome',

'message': "Welcome to Inuit! 🎉 We craft luxury footwear that blends timeless elegance with uncompromising comfort. From handcrafted leather boots to sophisticated sneakers, each pair tells a story of Italian craftsmanship.",

'type': 'quick_replies',

'options': ['Tell me more', 'Show me shoes']

},

{

'id': 'intro',

'message': "Perfect! Let's find your ideal pair. What type of shoe are you looking for today?",

'type': 'buttons',

'options': [

('👞 Formal Shoes', 'formal'),

('👟 Sneakers', 'sneakers'),

('👢 Boots', 'boots'),

('👞 Loafers', 'loafers')

]

},

{

'id': 'occasion',

'message': "Excellent choice! What occasion are you shopping for?",

'type': 'buttons',

'options': [

('💼 Work/Business', 'work'),

('🎉 Special Events', 'events'),

('👤 Everyday Wear', 'casual'),

('🎁 Gift', 'gift')

]

},

{

'id': 'size',

'message': "Great! What's your shoe size? (US sizing)",

'type': 'quick_replies',

'options': ['7-8', '9-10', '11-12', "I'm not sure"]

},
```

```
'id': 'recommendations',
'message': "Based on your preferences, here are our top recommendations:",
'type': 'carousel',
'products': [
    {'name': 'Milano Executive', 'price': '$450', 'emoji': ' italiane', 'desc': 'Italian leather, hand-stitched'},
    {'name': 'Urban Elite', 'price': '$380', 'emoji': ' modernas', 'desc': 'Premium comfort, modern design'},
    {'name': 'Heritage Classic', 'price': '$520', 'emoji': ' clasicas', 'desc': 'Timeless craftsmanship'}
]
},
{
'id': 'videos',
'message': "Want to see how we craft perfection? Here's a behind-the-scenes look at our workshop:",
'type': 'videos',
'videos': [
    {'title': 'Leather Selection', 'duration': '2:15'},
    {'title': 'Hand Stitching Process', 'duration': '3:40'},
    {'title': 'Quality Inspection', 'duration': '1:55'}
]
},
{
'id': 'order',
'message': "Ready to experience Inuit luxury? We offer free home delivery worldwide with premium packaging! 📦",
'type': 'buttons',
'options': [
    ('💻 Place Order', 'order'),
    ('💬 Chat with Expert', 'expert'),
    ('✉️ Email Details', 'email')
]
},
{
'id': 'conclusion',
'message': "Thank you for choosing Inuit! Your order will arrive in 5-7 business days. We'll send tracking details to your email. ✉️",
'type': 'quick_replies',
'options': ['Track Order', 'Browse More', 'Main Menu']
}
}
```

]

```
def add_message(sender, message, **kwargs):
    """Add a message to chat history"""
    st.session_state.chat_history.append({
        'sender': sender,
        'message': message,
        'timestamp': datetime.now(),
        **kwargs
    })

def display_message(msg):
    """Display a chat message"""
    if msg['sender'] == 'bot':
        st.markdown(f"""
            <div class="chat-message bot-message">
                <div class="avatar bot-avatar">🤖</div>
                <div>
                    <div style="color: #334155; font-size: 14px;">{msg['message']}</div>
                    <div style="color: #94a3b8; font-size: 11px; margin-top: 4px;">
                        {msg['timestamp'].strftime('%l:%M %p')}
                    </div>
                </div>
            </div>
        """, unsafe_allow_html=True)
    else:
        st.markdown(f"""
            <div class="chat-message user-message">
                <div class="avatar user-avatar">👤</div>
                <div>
                    <div style="font-size: 14px;">{msg['message']}</div>
                    <div style="color: rgba(255,255,255,0.8); font-size: 11px; margin-top: 4px;">
                        {msg['timestamp'].strftime('%l:%M %p')}
                    </div>
                </div>
            </div>
        """, unsafe_allow_html=True)
```

```

</div>

"""", unsafe_allow_html=True)

def handle_choice(choice, display_text=None):
    """Handle user selection"""

    text = display_text if display_text else choice

    add_message('user', text)

# Save user choices

step = st.session_state.current_step

if step == 1:
    st.session_state.user_choices['shoe_type'] = choice

elif step == 2:
    st.session_state.user_choices['occasion'] = choice

elif step == 3:
    st.session_state.user_choices['size'] = choice

# Move to next step

if st.session_state.current_step < len(STEPS) - 1:
    time.sleep(0.5) # Simulate thinking

    st.session_state.current_step += 1

    current_step_data = STEPS[st.session_state.current_step]

    add_message('bot', current_step_data['message'], step_data=current_step_data)

st.rerun()

def reset_chat():
    """Reset the entire chat"""

    st.session_state.chat_history = []

    st.session_state.current_step = 0

    st.session_state.user_choices = {'shoe_type': '', 'occasion': '', 'size': ''}

    st.session_state.initialized = False

    st.rerun()

# Initialize chat with welcome message

```

```

if not st.session_state.initialized:

    add_message('bot', STEPS[0]['message'], step_data=STEPS[0])

    st.session_state.initialized = True


# Header

st.markdown("<h1 style='text-align: center; color: white;'>Inuit Chatbot Experience</h1>", unsafe_allow_html=True)

st.markdown("<p style='text-align: center; color: #cbd5e1; margin-bottom: 2rem;'>Luxury Footwear Shopping Assistant</p>",
unsafe_allow_html=True)


# Layout

col1, col2 = st.columns([2, 1])

with col1:

    # Chat container

    st.markdown("#### ⚡ Chat")

    # Display chat history

    chat_container = st.container()

    with chat_container:

        for msg in st.session_state.chat_history:

            display_message(msg)

    # Display interactive elements for the last bot message

    if msg['sender'] == 'bot' and msg == st.session_state.chat_history[-1]:

        step_data = msg.get('step_data', {})

# Quick replies

    if step_data.get('type') == 'quick_replies':

        cols = st.columns(len(step_data['options']))

        for idx, option in enumerate(step_data['options']):

            with cols[idx]:

                if st.button(option, key=f"quick_{idx}"):

                    handle_choice(option)

# Buttons

    elif step_data.get('type') == 'buttons':

```

```

for idx, (label, value) in enumerate(step_data['options']):
    if st.button(label, key=f"btn_{idx}"):
        handle_choice(value, label)

# Product carousel

elif step_data.get('type') == 'carousel':
    for idx, product in enumerate(step_data['products']):
        st.markdown(f"""


{product['emoji']}

{product['name']}



{product['desc']}



{product['price']}


""", unsafe_allow_html=True)
        if st.button("View Details", key=f"prod_{idx}"):
            handle_choice(f"view_{product['name']}", f"View {product['name']}")

# Videos

elif step_data.get('type') == 'videos':
    for idx, video in enumerate(step_data['videos']):
        st.markdown(f"""


{video['title']}

{video['duration']}


""", unsafe_allow_html=True)

```

```

</div>

<span style="color: #94a3b8;">▶</span>

</div>

</div>

"""", unsafe_allow_html=True)

if st.button("Watch", key=f"vid_{idx}"):
    handle_choice(f"watch_{video['title']}", f"Watch: {video['title']}")

# Input area

st.markdown("---")

col_input, col_send = st.columns([5, 1])

with col_input:

    user_input = st.text_input("Type a message...", key="user_input", label_visibility="collapsed")

    with col_send:

        if st.button("Send ▶", use_container_width=True):

            if user_input.strip():

                add_message('user', user_input)

                time.sleep(0.3)

                add_message('bot', "I didn't quite catch that! Would you like to explore our collections, speak with an expert, or return to the main menu?",

                           step_data={'type': 'quick_replies', 'options': ['⬅ Main Menu', '👤 Human Agent', '💻 Collections']})

                st.rerun()

with col2:

    # Progress tracker

    st.markdown("### 📈 Progress")

for idx, step in enumerate(STEPS):

    if idx < st.session_state.current_step:

        icon = "✅"

        color = "#10b981"

        bg = "#d1fae5"

    elif idx == st.session_state.current_step:

        icon = "🟡"

        color = "#f59e0b"

        bg = "#fef3c7"

```

```

else:
    icon = "●"
    color = "#94a3b8"
    bg = "#f1f5f9"

    st.markdown(f"""
<div style="background-color: {bg}; padding: 12px; border-radius: 8px;
margin-bottom: 8px; border: 2px solid {color};">
<div style="font-weight: 600; color: #1e293b; font-size: 12px;">
{icon} Step {idx + 1}: {step['id']}
</div>
</div>
""", unsafe_allow_html=True)

st.markdown("---")

# Key Features
st.markdown("### 🌟 Features")
st.markdown("""
- 🤝 **Warm Personality**: Luxury tone with friendly engagement
- 🎨 **Rich Elements**: Buttons, carousels, videos & more
- 📈 **Clear Journey**: Welcome → Discover → Convert
""")

st.markdown("---")

# User choices
st.markdown("### 📋 Your Selections")
st.markdown(f"**Shoe Type:** {st.session_state.user_choices['shoe_type']} or 'Not selected'")
st.markdown(f"**Occasion:** {st.session_state.user_choices['occasion']} or 'Not selected'")
st.markdown(f"**Size:** {st.session_state.user_choices['size']} or 'Not selected'")

st.markdown("---")

# Reset button

```

```

if st.button("⟲ Reset Chat", use_container_width=True):
    reset_chat()

# Fallback info
st.markdown("---")
st.markdown("### ⚠️ Fallback Handling")
st.info("Unclear messages trigger helpful navigation options to guide users back on track.", icon="💡")

```

Use this link to access the chatbot:

[Inuit Chatbot · Streamlit](#)

The screenshot shows the Streamlit interface for the Inuit Chatbot Experience. The top navigation bar includes 'Share' and other icons. The main title is 'Inuit Chatbot Experience' with a subtitle 'Luxury Footwear Shopping Assistant'. On the left, there's a 'Chat' section with a yellow profile picture, a welcome message from the bot, and two orange buttons: 'Tell me more' and 'Show me shoes'. A long input field and a 'Send ▶' button are below. On the right, there's a 'Progress' sidebar listing eight steps: Step 1: welcome (blue dot), Step 2: intro (orange dot), Step 3: occasion (orange dot), Step 4: size (orange dot), Step 5: recommendations (orange dot), Step 6: videos (orange dot), Step 7: order (orange dot), and Step 8: conclusion (orange dot). Below the progress bar, there's a 'Features' section with a star icon, a bulleted list of three features (warm personality, rich elements, clear journey), and a 'Your Selections' section showing 'Shoe Type: Not selected', 'Occasion: Not selected', and 'Size: Not selected'. At the bottom is an orange 'Reset Chat' button.

## **Task 2**

Write a paragraph about yourself in less than 300 words.

### **Answer:**

Hi, I'm Deep. I studied Computer Engineering at the Fr. C. Rodrigues Institute of Technology College, but what truly pulled me in was understanding why things work the way they do, especially when it comes to data and people. I enjoy digging into data, finding patterns, and turning them into insights that actually help someone make a better decision. Tools like SQL, Python, Power BI, and Tableau are part of my everyday work, but what matters more to me is clarity, making complex ideas feel simple and useful. Before getting into analytics, I spent time teaching, and that experience shaped how I think and communicate. Explaining concepts to students taught me patience, empathy, and the importance of storytelling skills, which I now bring into my analytical work. I'm naturally curious, always learning, and I love challenging myself, whether that's working on real-world projects or participating in competitions like the Imarticus Data Science Blogathon '25, where I placed in the top 10.

At my core, I'm someone who enjoys learning, collaborating, and growing through meaningful work. I want to be known as a data analyst who doesn't just deliver numbers, but helps people see possibilities, someone who adds value by listening, asking thoughtful questions, and turning information into action.