



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:

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 React for a chatbot interface for Inuit luxury footwear:

```
import React, { useState, useEffect, useRef } from 'react';
import { MessageCircle, ChevronRight, Package, Video, ShoppingBag, CheckCircle, Send, RotateCcw, User, Bot } from 'lucide-react';
```

```
const InuitChatbot = () => {  
  const [currentStep, setCurrentStep] = useState(0);  
  
  const [chatHistory, setChatHistory] = useState([]);  
  
  const [isTyping, setIsTyping] = useState(false);  
  
  const [userInput, setUserInput] = useState("");  
  
  const chatEndRef = useRef(null);  
  
  const [userChoices, setUserChoices] = useState({  
  
    shoeType: "",  
  
    occasion: "",  
  
    size: ""  
  })  
};
```

```
});

const steps = [
  {
    id: 'welcome',
    type: 'bot',
    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.",
    quickReplies: ['Tell me more', 'Show me shoes']
  },
  {
    id: 'intro',
    type: 'bot',
    message: "Perfect! Let's find your ideal pair. What type of shoe are you looking for today?",
    buttons: [
      { label: '👞 Formal Shoes', value: 'formal' },
      { label: '👟 Sneakers', value: 'sneakers' },
      { label: '👢 Boots', value: 'boots' },
      { label: '👞 Loafers', value: 'loafers' }
    ]
  },
  {
    id: 'occasion',
    type: 'bot',
    message: "Excellent choice! What occasion are you shopping for?",
    buttons: [
      { label: '✉️ Work/Business', value: 'work' },
      { label: '🎉 Special Events', value: 'events' },
      { label: '🏃 Everyday Wear', value: 'casual' },
      { label: '🎁 Gift', value: 'gift' }
    ]
  },
  {
    id: 'size',
    type: 'bot'
  }
];
```

```
message: "Great! What's your shoe size? (US sizing)",  
quickReplies: ['7-8', '9-10', '11-12', "I'm not sure"]  
,  
{  
  id: 'recommendations',  
  type: 'bot',  
  message: "Based on your preferences, here are our top recommendations:",  
  carousel: [  
    { name: 'Milano Executive', price: '$450', img: 'img1', desc: 'Italian leather, hand-stitched' },  
    { name: 'Urban Elite', price: '$380', img: 'img2', desc: 'Premium comfort, modern design' },  
    { name: 'Heritage Classic', price: '$520', img: 'img3', desc: 'Timeless craftsmanship' }  
  ]  
,  
{  
  id: 'videos',  
  type: 'bot',  
  message: "Want to see how we craft perfection? Here's a behind-the-scenes look at our workshop:",  
  videos: [  
    { title: 'Leather Selection', duration: '2:15' },  
    { title: 'Hand Stitching Process', duration: '3:40' },  
    { title: 'Quality Inspection', duration: '1:55' }  
  ]  
,  
{  
  id: 'order',  
  type: 'bot',  
  message: "Ready to experience Inuit luxury? We offer free home delivery worldwide with premium packaging! 📦",  
  buttons: [  
    { label: '🛍 Place Order', value: 'order' },  
    { label: '💬 Chat with Expert', value: 'expert' },  
    { label: '✉️ Email Details', value: 'email' }  
  ]  
,  
{  
  id: 'conclusion',
```

```
        type: 'bot',  
  
        message: "Thank you for choosing Inuit! Your order will arrive in 5-7 business days. We'll send tracking details to your email. ♦",  
        quickReplies: ['Track Order', 'Browse More', 'Main Menu']  
    }  
};  
  
useEffect(() => {  
  
    if (chatHistory.length === 0) {  
  
        addBotMessage(steps[0]);  
    }  
}, []);  
  
useEffect(() => {  
  
    chatEndRef.current?.scrollIntoView({ behavior: 'smooth' });  
}, [chatHistory, isTyping]);  
  
const addBotMessage = (stepData) => {  
  
    setIsTyping(true);  
  
    setTimeout(() => {  
  
        setChatHistory(prev => [...prev, { ...stepData, sender: 'bot', timestamp: new Date() }]);  
  
        setIsTyping(false);  
    }, 800);  
};  
  
const addUserMessage = (message) => {  
  
    setChatHistory(prev => [...prev, {  
  
        message,  
        sender: 'user',  
        timestamp: new Date()  
    }]);  
};  
  
const handleChoice = (value, displayText) => {  
  
    addUserMessage(displayText || value);  
}
```

```

if (currentStep === 1) {

  setUserChoices({...userChoices, shoeType: value});

} else if (currentStep === 2) {

  setUserChoices({...userChoices, occasion: value});

} else if (currentStep === 3) {

  setUserChoices({...userChoices, size: value});

}

if (currentStep < steps.length - 1) {

  const nextStep = currentStep + 1;

  setCurrentStep(nextStep);

  setTimeout(() => addBotMessage(steps[nextStep]), 1000);

}

};

const handleSendMessage = () => {

  if (userInput.trim()) {

    addUserMessage(userInput);

    setUserInput("");

    setTimeout(() => {

      addBotMessage({

        type: 'bot',

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

        quickReplies: ['⬅ Main Menu', '➡ Human Agent', '➡ Collections']

      });

    }, 800);

  }

};

const resetChat = () => {

  setCurrentStep(0);

  setUserChoices({ shoeType: "", occasion: "", size: "" });

  setChatHistory([]);

  addBotMessage(steps[0]);

}

```

```

};

return (
  <div className="min-h-screen bg-gradient-to-br from-slate-900 via-slate-800 to-slate-900 p-4 md:p-8">
    <div className="max-w-7xl mx-auto">
      <div className="text-center mb-6">
        <h1 className="text-3xl md:text-4xl font-bold text-white mb-2">Inuit Chatbot Experience</h1>
        <p className="text-slate-300 text-sm md:text-base">Luxury Footwear Shopping Assistant</p>
      </div>

      <div className="grid lg:grid-cols-3 gap-6">
        /* Chat Interface */
        <div className="lg:col-span-2 bg-white rounded-2xl shadow-2xl overflow-hidden flex flex-col" style={{ height: '600px' }}>
          /* Header */
          <div className="bg-gradient-to-r from-amber-600 to-amber-700 p-4 flex items-center justify-between">
            <div className="flex items-center">
              <div className="bg-white rounded-full p-2 mr-3">
                <Bot className="text-amber-600" size={20} />
              </div>
            </div>
            <div>
              <h3 className="text-white font-semibold">Inuit Assistant</h3>
              <p className="text-amber-100 text-xs">Online • Typically replies instantly</p>
            </div>
          </div>
          <button
            onClick={resetChat}
            className="p-2 bg-white/20 rounded-lg hover:bg-white/30 transition"
            title="Reset Chat"
          >
            <RotateCcw className="text-white" size={18} />
          </button>
        </div>
        /* Chat Messages */
        <div className="flex-1 overflow-y-auto p-4 bg-slate-50">

```

```

{chatHistory.map((msg, idx) => (
  <div key={idx} className={`mb-4 flex ${msg.sender === 'user' ? 'justify-end' : 'justify-start'}`}>
    {msg.sender === 'bot' && (
      <div className="flex-shrink-0 mr-2">
        <div className="w-8 h-8 rounded-full bg-amber-100 flex items-center justify-center">
          <Bot size={16} className="text-amber-600" />
        </div>
      </div>
    )}

    <div className={`max-w-md ${msg.sender === 'user' ? 'order-1' : ''}`}>
      <div className={`rounded-2xl p-3 shadow-sm ${msg.sender === 'user' ? 'bg-gradient-to-r from-amber-600 to-amber-700 text-white' : 'bg-white border border-slate-200'}`}>
        <p className={`text-sm ${msg.sender === 'user' ? 'text-white' : 'text-slate-700'}`}>
          {msg.message}
        </p>
      </div>
    </div>

    {msg.quickReplies && msg.sender === 'bot' && (
      <div className="flex flex-wrap gap-2 mt-3">
        {msg.quickReplies.map((reply, i) => (
          <button
            key={i}
            onClick={() => handleChoice(reply, reply)}
            className="px-3 py-1.5 bg-amber-50 text-amber-700 rounded-full text-xs font-medium hover:bg-amber-100 transition border border-amber-200">
            >
              {reply}
            </button>
        ))}
      </div>
    )}
  )
}

{msg.buttons && msg.sender === 'bot' && (

```

```

<div className="space-y-2 mt-3">

  {msg.buttons.map((btn, i) => (
    <button
      key={i}
      onClick={() => handleChoice(btn.value, btn.label)}
      className="w-full px-4 py-2.5 bg-gradient-to-r from-amber-600 to-amber-700 text-white rounded-lg text-sm font-medium
      hover:from-amber-700 hover:to-amber-800 transition shadow-sm"
    >
      {btn.label}
    </button>
  )))
}

</div>
)}

{msg.carousel && msg.sender === 'bot' && (
  <div className="mt-4 space-y-3">
    {msg.carousel.map((item, i) => (
      <div key={i} className="bg-gradient-to-r from-slate-50 to-slate-100 p-3 rounded-lg border border-slate-200">
        <div className="flex items-center justify-between">
          <div className="flex items-center gap-3">
            <span className="text-3xl">{item.img}</span>
            <div>
              <h4 className="font-semibold text-slate-800 text-sm">{item.name}</h4>
              <p className="text-xs text-slate-600">{item.desc}</p>
            </div>
          </div>
          <span className="text-amber-700 font-bold text-sm">{item.price}</span>
        </div>
        <button className="w-full mt-2 px-3 py-1.5 bg-white text-amber-700 rounded text-xs font-medium
        hover:bg-amber-50 transition border border-amber-200">
          View Details
        </button>
      </div>
    )))
}

</div>
)}

```

```
{msg.videos && msg.sender === 'bot' && (
  <div className="mt-4 space-y-2">
    {msg.videos.map((video, i) => (
      <div key={i} className="flex items-center gap-3 bg-slate-100 p-3 rounded-lg hover:bg-slate-200 transition cursor-pointer">
        <Video className="text-amber-600" size={20} />
        <div className="flex-1">
          <p className="text-sm font-medium text-slate-800">{video.title}</p>
          <p className="text-xs text-slate-500">{video.duration}</p>
        </div>
        <ChevronRight className="text-slate-400" size={16} />
      </div>
    )))
  </div>
)}
</div>
<p className="text-xs text-slate-400 mt-1 px-2">
  {msg.timestamp?.toLocaleTimeString([], { hour: '2-digit', minute: '2-digit' })}
</p>
</div>

{msg.sender === 'user' && (
  <div className="flex-shrink-0 ml-2">
    <div className="w-8 h-8 rounded-full bg-slate-700 flex items-center justify-center">
      <User size={16} className="text-white" />
    </div>
  </div>
)}
</div>
))}

{isTyping && (
  <div className="flex items-center gap-2 mb-4">
    <div className="w-8 h-8 rounded-full bg-amber-100 flex items-center justify-center">
      <Bot size={16} className="text-amber-600" />
    </div>
  </div>
)}
```

```

        </div>

        <div className="bg-white rounded-2xl p-3 shadow-sm border border-slate-200">
          <div className="flex gap-1">
            <div className="w-2 h-2 bg-slate-400 rounded-full animate-bounce" style={{ animationDelay: '0ms' }}></div>
            <div className="w-2 h-2 bg-slate-400 rounded-full animate-bounce" style={{ animationDelay: '150ms' }}></div>
            <div className="w-2 h-2 bg-slate-400 rounded-full animate-bounce" style={{ animationDelay: '300ms' }}></div>
          </div>
        </div>
      </div>
    <}>

    <div ref={chatEndRef} />
  </div>

  /* Input Area */
  <div className="p-4 bg-white border-t border-slate-200">
    <div className="flex gap-2">
      <input
        type="text"
        value={userInput}
        onChange={(e) => setUserInput(e.target.value)}
        onKeyPress={(e) => e.key === 'Enter' && handleSendMessage()}
        placeholder="Type a message...">
      <span
        className="flex-1 px-4 py-2.5 border border-slate-300 rounded-lg text-sm focus:outline-none focus:ring-2 focus:ring-amber-500
        focus:border-transparent">
        </span>
      <button
        onClick={handleSendMessage}
        className="px-4 py-2.5 bg-amber-600 text-white rounded-lg hover:bg-amber-700 transition flex items-center gap-2">
        >
        <Send size={18} />
      </button>
    </div>
  </div>
</div>

```

```

/* Flow Visualization & Info */

<div className="space-y-6">

  /* Progress Flow */

  <div className="bg-white rounded-2xl shadow-2xl p-6">
    <h3 className="text-lg font-bold text-slate-800 mb-4 flex items-center gap-2">
      <CheckCircle size={20} className="text-amber-600" />
      Conversation Progress
    </h3>

    <div className="space-y-2">
      {steps.map((step, idx) => (
        <div
          key={step.id}
          className={`p-3 rounded-lg border-2 transition ${idx === currentStep ? 'border-amber-500 bg-amber-50' : idx < currentStep ? 'border-green-300 bg-green-50' : 'border-slate-200 bg-slate-50'}`}
        >
          <div className="flex items-center gap-2">
            {idx < currentStep ? (
              <CheckCircle className="text-green-600 flex-shrink-0" size={18} />
            ) : idx === currentStep ? (
              <div className="w-4 h-4 rounded-full bg-amber-500 animate-pulse flex-shrink-0" />
            ) : (
              <div className="w-4 h-4 rounded-full border-2 border-slate-300 flex-shrink-0" />
            )}
            <span className="font-semibold text-xs text-slate-700">
              {idx + 1}. {step.id}
            </span>
          </div>
        </div>
      )))
    </div>
  </div>
</div>

```

```
</div>

</div>

/* Key Features */

<div className="bg-white rounded-2xl shadow-2xl p-6">
  <h3 className="text-lg font-bold text-slate-800 mb-4">Key Features</h3>

  <div className="space-y-3">
    <div className="flex gap-3">
      <MessageCircle className="text-amber-600 flex-shrink-0" size={20} />
      <div>
        <h4 className="font-semibold text-slate-800 text-sm">Warm Personality</h4>
        <p className="text-xs text-slate-600">Luxury tone with friendly engagement</p>
      </div>
    </div>
  </div>

  <div className="flex gap-3">
    <Package className="text-amber-600 flex-shrink-0" size={20} />
    <div>
      <h4 className="font-semibold text-slate-800 text-sm">Rich Elements</h4>
      <p className="text-xs text-slate-600">Buttons, carousels, videos & more</p>
    </div>
  </div>

  <div className="flex gap-3">
    <ShoppingBag className="text-amber-600 flex-shrink-0" size={20} />
    <div>
      <h4 className="font-semibold text-slate-800 text-sm">Clear Journey</h4>
      <p className="text-xs text-slate-600">Welcome → Discover → Convert</p>
    </div>
  </div>
</div>

/* Error Handling */
```

```
<div className="bg-red-50 border border-red-200 rounded-2xl p-4">  
  <h4 className="text-sm font-semibold text-red-800 mb-2 flex items-center gap-2">  
    <span>⚠</span> Fallback Handling  
  </h4>  
  
  <p className="text-xs text-red-700">  
    Unclear messages trigger helpful navigation options to guide users back on track.  
  </p>  
  </div>  
  </div>  
  </div>  
  </div>  
  </div>  
);  
};
```

```
export default InuitChatbot;
```

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.