1. Introduction

Purpose of the Project

Our chatbot tries to make for a better car buying experience through instant, accurate, and engaging communication between the dealership and potential customers. The personalization of customer service can be a great game changer in the competitive automotive industry, with a relationship directly proportional to the sales and customer satisfaction. In this manner, our chatbot will answer all these queries through automation and manage the inquiries efficiently so that humans can take on higher-order customer service work. In addition, this automation enhances the speed of the process and maintains the service availability to customers for offering information and help at all times, hence reducing the number of people who wait in line.

Overview of the Chatbot

Our chatbot is designed in a pretty peculiar way, it has multiple personalities, but not just simple, strict or jolly personalities, but the ones of famous characters. Each essence gives a different kind of interaction to enhance the customer experience and engagement.

The personalities being:

- **Ali G:** Informal, humorous and British
- **Yoda**: The famous Jedi master
- **Mr Bean:** Doesn’t talk much, but still an enjoyable and funny experience
- **Willy Wonka:** Eccentric and creative
- **Stephen Hawking:** One of the smartest scientists in the world, and even speaks
- **French Guy:** French flair and very charming
- **German Guy:** Similar to the French fellow, just German instead.

These are more than just personalities, in fact, they are crafted in such a manner to look like different ways in which customers would like to receive information and interact with a salesperson. This therefore ensures that each of the customer interactions will not just be informative, but uniquely memorable and potentially can impact customer loyalty and satisfaction.

2. Technical Overview

Architecture

The chatbot project at hand is very robust and scalable, most of it based on a Python backend with main.py and a frontend using HTML, CSS, and JavaScript. In this described architecture, the backend is the brain of the chatbot, processing the input provided by the user, generating an appropriate response based on the active personality, and managing the API request. As such in the file *main.py*:

- **API communication:** They execute all external data source requests to make the chatbot capable of fetching updated information at OpenAI.

- **Personality Module Integration:** Dynamic switching of multiple personality modules based on user interaction or preset conditions. Each module represents an encapsulation of traits and given styles of response.
- **Input Processing:** Uses natural language processing tools to interpret user queries, and keeps sending them, enabling the conversation with the chatbot.

This backend is linked to the frontend through the lightweight WSGI web application framework called Flask. This library takes any incoming request from the users and routes it to the respective Python functions, after which the user may interact in real time with the chatbot.

Frontend Details

The most important parts of the front end of the chatbot are index.html and chatbot.html, which strictly form a crucial part of the interaction the user is having.

- **index.html:** This is the welcome page of our start-up. The user is greeted with the website of the car dealership, he is able to browse a small selection of cars, discover who the founders are, and more. This page is an introduction to the company and it also links to the chatbot.
- **chatbot.html:** This is the file where the core interaction with the chatbot takes place. The designed chat interface is very friendlily in nature, allowing a user to interact with different personalities. The design is very clean without any clutter. Within the chat window, a user can see whether there is history present of the chats; all his or her responses have direct visual feedback, such as typing indicators for the user when the chatbot is preparing a message.

CSS in both cases styles the two HTML files into appealing, cool interfaces, while JavaScript makes both interactive to users, reacting to any kind of input from the user, button clicks, and other on-screen interactions. JavaScript code also manipulates dynamic content of the chat interface, displaying several personalities and fetching responses from the back end.

3. Development Process

Challenges and Solutions

Making Sure the API Connection Works:
One of the first technical challenges was to stabilize the connection with external APIs, from which real-time data was to be fetched to provide fresh information about available car models, features, and prices. We had troubles there making that connection for quite a while, and weren't sure if it was in the code, the API key or OpenAI servers.
Solution
We managed to solve it pretty quickly once we identified the error, our code was seamless it was in OpenAI that we had problems. Not enough tokens to make the connection. So once fixed the code ran beautifully.

Multi personality
The multi-personas integration in the chatbot required a dynamic system that could switch contexts seamlessly but still keep the uniqueness of each personality in a single interaction. That was a bit of a challenge at first. Was that too many personalities? Should we stick to one? A 300-word prompt for each character was enormous.
Solution
We managed to make it one prompt, fixed with a variable for the name of the personality, each time the character was mentioned in the prompt, we had the variable. We had to make sure it still ran perfectly, so we added a few lines amplifying the humour and way of talking of the chosen character.

Tools and Technologies Used

- Python: This language was used for the backend of the project. It took care of all API connections and the handling of chat logic.
- Flask: This Python library linked the backend with the front end. It also managed routing, requests and server-side session control.
- HTML/CSS/JavaScript: These are the languages used for the front end, they provide a seamless and interactive user experience.
- Git: A cloud-based server, where the work could be uploaded for the whole team to see and edit.

The development process, though challenging, was very interesting, from the search for such libraries and technologies, to their actual implementation was very satisfying.

4. Chatbot Personalities

Implementation of Personalities

Any personality in each of our chatbots is a separate module designed to encapsulate some distinct style of behavior in communication. If the modular approach will be followed in our implementations, there will be a clear division of functionalities, hence updates or new additions of personalities will easily be done.

- **Ali G:** Predisposed to slang and street-wise diction, this persona is informal and often humorous in the manner in which it addresses users. His vocabulary includes british slang, known expressions he uses and so much more.
- **Yoda:** Being the extremely famous Star Wars character, this persona talks using speech inversion, whereby responses are formed as typical Yoda would say them.
- **Mr. Bean:** Comic minimal, verbal responses, and eccentric attitude. This Character was added as more of a joke, because like most peolpe know, Mr.Bean doesn’t actually talk, he mumbles, does gestures, saying out loud a word or two one in a while.
- **Willy Wonka:** Imaginative and creative responses with a flair for being overly descriptive with exuberant but fanciful descriptions. Willy Wonka is extravagant in the way he talks using big words.

- **Stephen Hawking:** The persona was simple, he was one of the smartest scientists in the world, so he acted that way. Responded in a technical manner, explaining in great detail. We also added a voice for Stephen Hawking because it was so iconic.
- **French Guy:** Responds in a very sweet mix of English and French, answering questions in a romantic and tasteful manner. If the user writes in French, he will follow in and speak only French.
- **German Guy:** Similarly to the French guy, this one is straight, to the point, and effective. He responds using a mix of german-english linguo. Only offers German cars and brands.

In addition, the use of dynamic content is also facilitated according to the context of the conversation, having each of the personalities carry their own response database and templated phrases. In point of fact, the use of a decision-making algorithm within the chatbot itself actually helps to determine the appropriate personality the user needs based on preferences, past interactions, and contextual cues.

Interaction Examples

Here are some interaction examples:

- **Ali G:**
  User: "What's the best car for a road trip?"
  Ali G: "Yo, for a wicked road trip, you gotta roll in the SUV, innit? Spacious and smooth, bro. Faster than you can say ‘Booyakasha’ "
- **Yoda:**
  User: "I need a car that's good for the environment."
  Yoda: "Good for the environment, hybrid cars are. Save the planet, you will."
- **French guy:**
  User: "I want a car that feels luxurious."
  French guy: "Ah, the luxury you seek, mon ami, is found in our sedans. Smooth like a fine Bordeaux."

These interactions are designed to not only provide information but also to enhance the engagement by adapting to the user's personality preferences, making the chatbot interaction more personal and enjoyable. Simply said, they offer an extremely funny and enjoyable experience.

5. Testing and Results

Testing Methods

The testing was pretty plain and simple, we just kept debugging and running:

- **Automated Testing:** We made sure to check and test each part of the code. We had different test for different areas, like the prompt, the backend-frontend connection, the APIs, or the website which also failed a nice amount at first.
- **User Acceptance Testing (UAT):** Then, once we ran perfectly, we tested the chatbot with real users to see what they thought about it. This especially help us in

the prompt engineering section. From their feedback we kept building up our characters

We faced many challenges during testing, but eventually, we managed to fix them and improve the chatbot’s performance.

Feedback and Iterations

Feedback from users was overwhelmingly positive:

- **Initial Feedback:** At first users like our roommates and friends, loved the idea. They enjoyed the possibility of changing personalities, and being able to choose the one for themselves.
- **First Iteration:** Encouraged by that feedback, we enhanced and amplified the different personalities, focusing on the chatbot's ability to have ongoing conversations and remember what the user said before.
- **Subsequent Feedback:** At this point, we were practically finished. Users continued to praise our project, which gave us confidence for the presentation.

This feedback helped us realise that we were doing something correctly. It led to a more engaging, smoother chatbot, that met to the high expectations of our users.

6. Conclusion

Our project has developed a multi-personality chatbot that successfully enhances the buying experience of any car. Key accomplishments include solid architecture that can handle multiple API connections, engaging user interaction, great personality dynamics, and a back-end system that is able to scale. Going forward, we will be looking to embed advanced understanding of natural language and emotion recognition even as we broaden the scope of the chatbot's capabilities to include more personalities and deeper integration with business systems. This will further improve the user experience and make chatbots much more applicable in broad customer service settings.