**CREATE A CHATBOT IN PYTHON**

**PHASE 2: INNOVATION**

**INTRODUCTION**

In this phase, we will further enhance our chatbot project by exploring advanced techniques and technologies. Here is an overview of the key steps and considerations for this phase:

**PROBLEM DEFINITION:**

The challenge is to create a chatbot in python that provides Exceptional customer service,answering user queries on a website or application. The objective is to deliver high quality support to users,ensuring a positive user experience and customer satisfaction.

**SCOPE OF THE PROJECT:**

Our project’s scope remains the same: to create a chatbot that delivers high-quality customer service for a hotel’s web application. The chatbot will handle user queries, offer assistance and provide information about services.

**DATASET SELECTION:**

While we initially planned to use custom data and the provided dataset (<https://www.kaggle.com/datasets/grafstor/simple-dialogs-for-chatbot>), we will now explore the possibility of using additional datasets specifically curated for training and fine-tuning pre-trained language models.

**ARCHITECTURE/FRAMEWORK:**

Below is a simplified architecture for our enhanced chatbot project:

User Interface:

● Web application

● Chatbot integrated into web app

Web Application:

● Python for web development and chatbot development

● Flask framework used for python web development

● Web server

● Chatbot which uses a ML model

Python:

● NLP library (e.g., spaCy, NLTK)

● Machine learning library (e.g., TensorFlow, PyTorch)

● Machine Learning Model

**ALGORITHM TO CREATE A CHATBOT IN PYTHON:**

Step 1: Install NLTK using pip:

**Pip install nltk**

Step 2: Import the necessary libraries:

**Import nltk**

**From nltk.chat.util import Chat**

Step 3: Create a list of patterns and responses. Each pattern-response pair should be a tuple.

**Pairs = [ [ r”hi|hello|hey”,**

**[“Hello!”, “how are you”, “How can I help you ?”] ],**

**[[“I’m good, thanks. How about you?”, “I’m a chatbot, so I’m always fine.”]],]**

Step 4: Initialize the chatbot using the Chat class from NLTK:

**chatbot = Chat (pairs, reflections)**

Step 5: Create a function to start the chat and interact with the user:

**Def chat\_with\_user(): print(“Hello! I’m your chatbot. Type ‘exit’ to end the conversation.”)**

Step 6: Call the chat\_with\_bot() function to start the bot