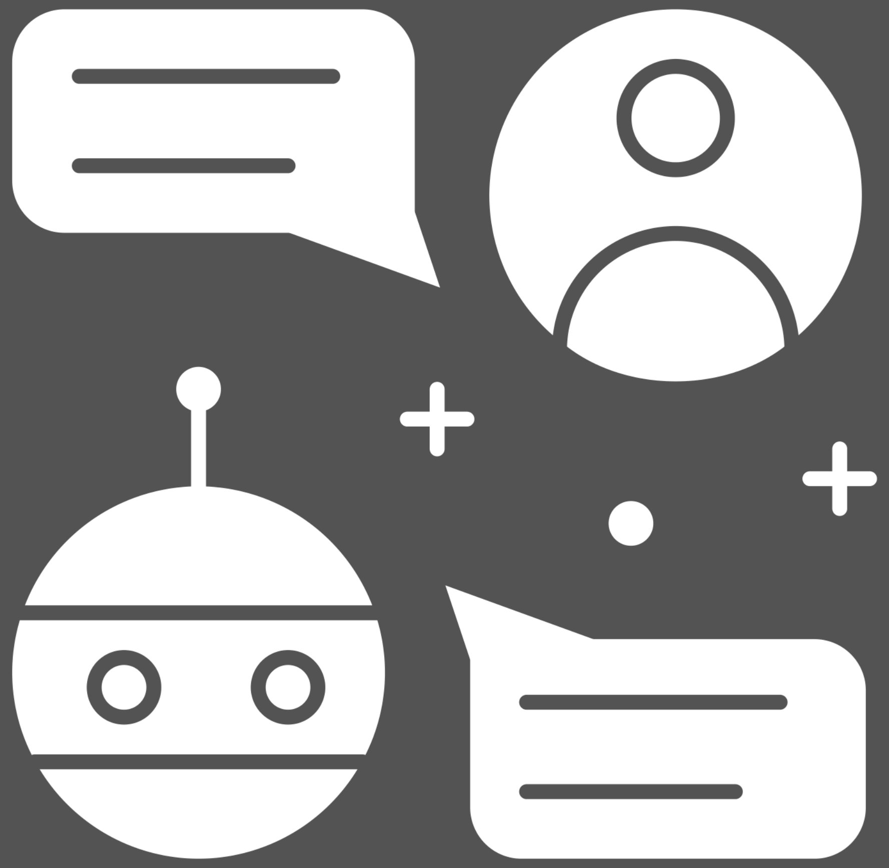
# Create a Chatbot using Python

Artificial Intelligence - Phase 2

Manoj Kumar - au711021104025



## AGENDA

- Environment Setup
- Data Collection and Preprocessing
- Natural Language Processing (NLP) Implementation
- Chatbot Architecture
- Integration
- Testing and Quality Assurance
- Deployment
- Continuous Improvement
- Security and Privacy
- Documentation and Maintenance
- Scaling
- Analytics and Insights

## ENVIRONMENT SETUP

**Choose Programming Language:** Start by selecting the programming language for chatbot development. Python is a popular choice due to its extensive libraries for NLP and machine learning.

**Install Dependencies:** Use package managers like pip to install the necessary libraries and dependencies. Key libraries include NLTK, spaCy, and TensorFlow for NLP tasks.

**Set Up IDE:** Select an Integrated Development Environment (IDE) like PyCharm or Jupyter Notebook for coding, debugging, and testing.

## DATA COLLECTION AND PREPROCESSING

**Data Acquisition:** Gather the dataset needed for training and testing. In this case, you can use the "Simple Dialogs for Chatbot" dataset from Kaggle. Ensure the data aligns with the chatbot's intended purpose.

**Data Cleaning:** Preprocess the data by removing duplicates, handling missing values, and correcting any inaccuracies. This ensures the dataset is clean and ready

**Text Preprocessing:** Tokenize the text data into words or phrases, remove stop words (common words like "the" or "and"), and convert text to lowercase for uniformity.ining.

# NATURAL LANGUAGE PROCESSING (NLP) IMPLEMENTATION

**Text Tokenization:** Implement text tokenization to break down user inputs and responses into manageable tokens. This enables the chatbot to understand sentence structure.

**Entity Recognition:** Incorporate named entity recognition (NER) to identify and extract important entities such as dates, locations, and product names from user inputs.

**Sentiment Analysis:** Integrate sentiment analysis to determine the emotional tone of user queries, aiding in providing context-appropriate responses.

**Intent Detection:** Train machine learning models to classify user intents. This helps the chatbot understand the purpose behind user queries, whether they are asking for information, seeking help, or making a purchase.

## CHATBOT ARCHITECTURE

**Dialog Management:** Create a dialog management system that tracks the context of the conversation. This ensures that the chatbot engages in coherent and context-aware interactions.

**Response Generation:** Develop a response generation module that selects or generates appropriate responses. This can involve rule-based systems or more advanced natural language generation techniques.

## INTEGRATION

**Platform Integration:** Choose the platform where the chatbot will be deployed (e.g., a mobile app). Implement integration using suitable methods such as webhooks, APIs, or software development kits (SDKs**).**

**User Interface:** Design and implement a user-friendly interface that allows users to interact with the chatbot seamlessly. Ensure it is responsive and easy to use.

## TESTING AND QUALITY ASSURANCE

**Functional Testing:** Conduct testing to verify that the chatbot accurately responds to various user queries and intents. Test edge cases and uncommon inputs to uncover potential issues.

**User Testing:** Involve real users in testing to collect feedback on usability and user satisfaction. Use this feedback for iterative improvements.

## DEPLOYMENT

**Production Deployment:** Deploy the chatbot to a production environment, making it accessible to users. Ensure the infrastructure can handle concurrent user interactions without performance issues.

# THANK YOU !