🧾 🎯 **Project Title: CUSTOM AI-POWERED CHATBOT – FULL STACK**  
📅 **Project Timeline:** October 2023 – August 2024  
🎥 YouTube Demo: [https://youtu.be/DtClZqPPPRQ](https://youtu.be/DtClZqPPPRQ?utm_source=chatgpt.com)  
📦 GitHub Source Code: <https://github.com/IvanSicaja/2023.07.31_GitHub_Custom_AI---Powered-Chatbot---Full-Stack>   
----------------------------------------------------------------------------------------------------------------

🏷️ My Personal Profiles: ⬇︎  
🎥 Video Portfolio: To be added  
📦 GitHub Profile: <https://github.com/IvanSicaja>  
🔗 LinkedIn: <https://www.linkedin.com/in/ivan-si%C4%8Daja-832682222>  
🎥 YouTube: <https://www.youtube.com/@ivan_sicaja>  
 ----------------------------------------------------------------------------------------------------------------

### 📚🔍 Project description: ⬇︎⬇︎⬇︎

### 💡 App Purpose

The **AI-powered chatbot** employs a **Feedforward Neural Network**, implemented using **TensorFlow** and **Python**, to generate responses based on **user text inputs**. It utilizes advanced **AI technology** to understand user input text, providing responses in the form of **video, images, or text**.

The entire **web page** is **responsive**, ensuring a seamless **user experience**. Relevant data is saved in a **SQL database** dynamically. The application is **containerized with Docker**, orchestrated using **Kubernetes**, and can be hosted on **AWS EKS** or any other platform that supports Kubernetes.

The **backend** is developed with **Flask**, incorporating **Jinja2** for templating. The **frontend** is designed using **Bootstrap** and enhanced with **JavaScript** for interactive features.

This integrated **technology stack** enables an intelligent and dynamic **chatbot experience**.

### 🧠 How It Works

PROJECT HIGHLIGHTS:

* **Data collection and preprocessing:**  
  Custom **queries** and an **interrogative sentences database** are created for every relevant topic. From the entire **dataset**, a **bag of words** is created with the usage of **scikit-learn** and **Python** module, and every relevant word is **tokenized** and **lemmatized**. Nonrelevant words (e.g. „the“ etc..) and stop characters are excluded from the **bag of words**. As the final dataset, a **.csv** and **.pkl** file are created. Every row of the **.csv** file is represented by the **vector** of the **word tokens** for the corresponding query sentence and the corresponding **label** which represents the token of the relevant topics.
* **Build, train, and evaluate the neural network model:**  
  **TensorFlow** was used to create a **feed-forwarded neural network** from scratch, and train and save the trained **neural network model**. A **validation accuracy score** was used for the **model validation**.
* **Building the web page frontend UI:**  
  Because the idea of this project is to build a **full stack, deployed app**, it was needed to develop the entire **web page**. **HTML, CSS, and Bootstrap framework** were used for the **frontend development**.
* **Building the web page backend:**  
  As a backend **Python framework**, **Flask** is used. It allows us to get the **interrogative sentences** from the user. The **model** is trained with **TensorFlow** and **Keras**. The **user input sentence text** is **tokenized, lemmatized**, and presented like a **vector** that has the corresponding **token values**. That **vector** is sent to the trained **forward-fed neural network** for analysis. The result from the **neural network** is **encoded** to the corresponding **class**, together with one of the random **text responses** for the relevant topic, and sent back to the **frontend** like the **chatbot text** with/without an **image or video**, with the usage of **Jinja2**.
* **Deployment:**  
  The entire **app** is **containerized** with the usage of **Docker Desktop** and the app **image** is uploaded to the **Docker Hub** platform. For the **deployment**, the **Google Kubernetes** container **orchestration tool** is used. The **app** is deployed locally with **Minikube** and tested for **performance**. The **domain** is bought on the **Namecheap** platform and the **app** can be deployed on any platform which supports **Kubernetes** such as: **Amazon Elastic Kubernetes Service (AWS->EKS), Microsoft Azure Kubernetes Service (AKS), Google Kubernetes Engine (GKE)**, etc.

### ⚠️ Note

This concept is for local use only, and significantly improved **training data** is essential for **real-world applications**.

### 🔧 Tech Stack

**Python, Feedforward Neural Network, Natural Language Processing (NLP), SQL, Pandas, TensorFlow, Keras, Scikit-learn, Git, GitHub, Docker Desktop, Docker Hub, Kubernetes, Minikube, Namecheap, HTML, CSS, Bootstrap, JavaScript, Flask, Jinja2, Linux, Similarity search, AI Image Generation (Adobe Firefly, Playground.com)**

### 📣 Hashtags Section

**# #chatbot #tensorflow #keras #docker #kubernetes #flask #bootstrap #javascript #nlp #sql #python #deeplearning #ai #machinelearning #feedforwardneuralnetwork #pandas #scikitlearn #git #github #dockerdesktop #dockerhub #minikube #namecheap #html #css #jinja2 #linux #aigeneration #adobefirefly #playgroundai #similaritysearch**