4222-SURYA GROUP OF INSTITUTION

VIKIRAVANDI-605 652.

NAAN MUDHALVAN PROJECT

CREATE CHATBOT IN PYTHON

PHASE 4: Development Part 2

Presented by:

S.ABUTHA KIR

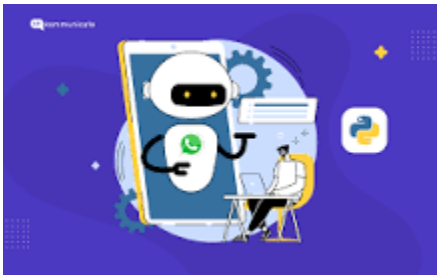
REG NO: 422221106301

ECE DEPARTMENT

AI_Phase 4: Development part 2

What is flask chatbot

Chatbots are software tools created to interact with humans through chat. The first chatbots could create simple conversations based on a complex system of rules. You can build intelligent chatbots for WhatsApp using the Python Framework Flask and the Kompose Bot builder.



The Flask is a Python micro-framework used to create small web applications and websites using Python. Flask works on a popular templating engine called Jinja2, a web templating system combined with data sources to the dynamic web pages.

How to deploy a chatbot on Flask

First, we will install the Flask library in our system using the below command:

```
pip install flask
```

And for Google Colab use the below command, mostly Flask comes pre-install on Google Colab.

```
!pip install flask
```

But first, let’s understand what the Flask framework in Python is.

How to create chatbot using Flask Python

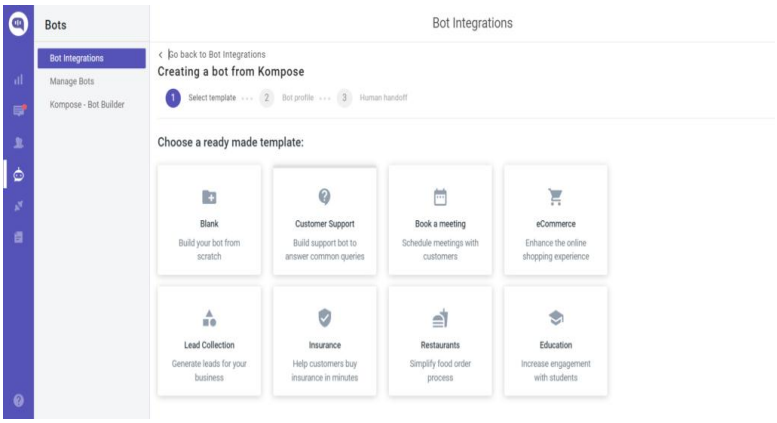
- Step 1: Import necessary methods of Flask and ChatterBot.
- Step 2: Then, we will initialize the Flask app by adding the below code. Flask(name) is used to create the Flask class object so that Python code can initialize the Flask server.
- Step 3: Now, we will give the name to our chatbot.

Developing a chatbot using flask

We will make a Flask chatbot. Flask is a microframework used for web development. We will follow the process given below:

1. Make a web app using the flask.
2. Make a directory for the templates.
3. Train the bot.
4. Make conversation with the bot.

Flask is a Python web framework that is widely used for building web applications. It is a lightweight framework that is easy to learn and has a simple syntax. In this tutorial, you will learn how to use Flask to create a back-end that communicates with a third-party API and renders the response in dynamic HTML using Jinja.



- **Step 1**: Installing Python 3.7 with pip3.7 along with Python3.10

This step is optional if you already have Python Version *<=3.8 or >=3.4*. Please check your python using this command – python -V or python3 -V

```
# Install Python 3.7 Version

sudo apt update

sudo apt install build-essential zlib1g-dev libncurses5-dev libgdbm-dev libnss3-dev libssl-dev libsqlite3-dev libreadline-dev libffi-dev wget libbz2-dev

sudo add-apt-repository ppa:deadsnakes/ppa

sudo apt install python3.7

python3.7 --version

# Install Pip

python3.7 -m pip install pip

# if you are getting distutils error
```

```
sudo apt-get install python3.7-distutilsCopy Code
```

- **Step 2:** Create and Install Virtualenv for Python Version <=3.8 or >=3.4

As we will be using virtual environment for this project. Please read our blog on Using Virtual Environments for Python Projects, if you are not familar with virtualenv in Python.

Let's install virtualenv using Python3.7

```
pip install virtualenvCopy Code
```

Create virtualenv named venv and activate it.

```
python3.7 -m virtualenv venv

source venv/bin/activateCopy Code
```

Please Note: When you have virtualenv activated, you will see python version 3.7. Outside virtualenv it will be different.

- **Step 3**: Installing Required Libraries

```
pip install Flask

pip install ChatterBot==1.0.8

pip install chatterbot-corpus==1.2.0

pip install spacy==2.1.9
```

```
pip install nltk==3.8.1Copy Code
```

- **Step 4**: Initializing Flask App

Create a new Python file (app.py) and add the following code:

```
from chatbot import chatbot

from flask import Flask, render_template, request

app = Flask(__name__)

app.static_folder = 'static'



@app.route("/")

def home():

    return render_template("index.html")

@app.route("/get")

def get_bot_response():

    userText = request.args.get('msg')

    return str(chatbot.get_response(userText))

if __name__ == "__main__":
```

```
app.run()Copy Code
```

- **Step 5**: Interact with Chatterbot Machine Learning Model

Create a new Python file (chatbot.py) and and add the following code:

```
from chatterbot import ChatBot

import spacy

spacy.cli.download("en_core_web_sm")

spacy.cli.download("en")



nlp = spacy.load('en_core_web_sm')

chatbot = ChatBot(

    'CoronaBot',

    storage_adapter='chatterbot.storage.SQLStorageAdapter',

    logic_adapters=[

        'chatterbot.logic.MathematicalEvaluation',

        'chatterbot.logic.TimeLogicAdapter',

        'chatterbot.logic.BestMatch',
```

```
        {
            'import_path': 'chatterbot.logic.BestMatch',
            'default_response': 'I am sorry, but I do not understand. I am still learning.',
            'maximum_similarity_threshold': 0.90
        }
    ],
    database_uri='sqlite:///database.sqlite3'
)
# Training With Own Questions
from chatterbot.trainers import ListTrainer
trainer = ListTrainer(chatbot)
training_data_quesans = open('training_data/ques_ans.txt').read().splitlines()
training_data_personal = open('training_data/personal_ques.txt').read().splitlines()
training_data = training_data_quesans + training_data_personal
trainer.train(training_data)
# Training With Corpus
from chatterbot.trainers import ChatterBotCorpusTrainer
```

```
trainer_corpus = ChatterBotCorpusTrainer(chatbot)

trainer_corpus.train(

    'chatterbot.corpus.english'

)Copy Code
```

In above example, we are loading data from training_data/ques_ans.txt and training_data/personal_ques.txt. You can download the files from here.

- **Step 6:** Creating Chatbot UI

    - HTML Template (index.html) – Create a new HTML file index.html inside the “templates” folder. Please copy the html content from here.
    - CSS File (style.css) – Create a new CSS file inside the “static” folder to style the chat interface. Please copy the css content from here.

Bellow is the folder structure your project should look like:

```
chatbot_project/

├── app.py

├── chatbot.py

├── venv

├── database.sqlite3.py

├── templates/
```

```
│   └── index.html
├── training_data/
│   ├── ques_ans.txt
│   └──personal_ques.txt
└── static/
    └── style.cssCopy Code
```

- **Step 7**: Run the Flask App

Open your terminal, navigate to the chatbot_project directory, and execute the following command:

```
python3.7 app.pyCopy Code
```

Step 8: Interact with the Chatbot

Open your web browser and go to http://localhost:5000 to access the chat interface. Now, you can type messages in the input field, press "Send," and watch your chatbot respond!

Note – If you are not getting right answer for questions, delete the .sqlite3 database file from your folder and re-run the flask app.

Conclusion

A chatbot is one of the simple ways to transport data from a computer without having to think for proper keywords to look up in a search or browse

several web pages to collect information; users can easily type their query in natural language and retrieve information.