**FINAL REPORT - SMART REPLY SYSTEM**

**Submitted to:** **Sagar Pandey Sir**

**Submitted By :**

|  |  |  |
| --- | --- | --- |
| **S. No.** | **NAME** | **REG. NO.** |
| 1. | Krishanpal Singh Rajput | 11805056 |



**ABSTRACT**

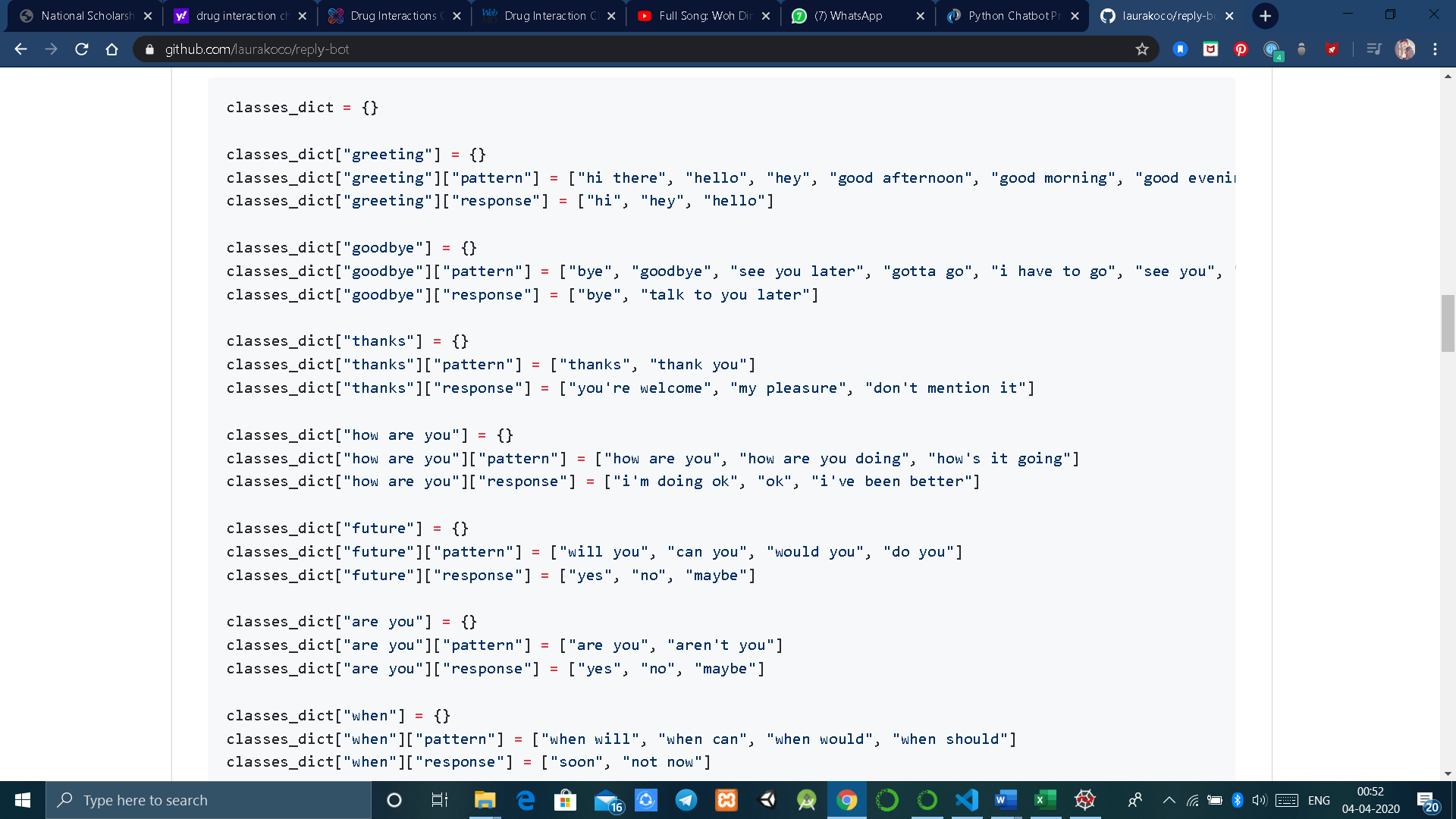
Through continuous efforts and study, this report presents the behaviour and working of THE SMART REPLY system. Smart Reply is a predictive text feature, which currently works in apps like Allo, Gmail, Inbox and from now on, Android Messages too. Functionality gives you the ability to answer messages by selecting pre-written responses based on machine learning and AI. The Smart reply system is based upon the responses being fed in intents.json file. It is a dataset used in the project. This system is focused to answer the medical and drug related queries.

**RELATED WORK (EXISTING WORK ANALYSIS)**

A retrieval-based chatbot uses predefined input patterns and responses. It then uses some type of heuristic approach to select the appropriate response.

<https://github.com/gabrielfarah/QA_Bot> has keras implementation of bot but the system lacks few things such as – the accuracy. Their project also doesn’t have a graphical user interface that can serve the users better.

<https://github.com/laurakoco/reply-bot> : they have used nltk, tensorflow. But the difference lies in the dataset. The dataset they have presented in the python dictionary. What I used is json intent dataset.



**IMPLEMENTATION**

The smart reply system will be trained on the dataset which contains categories (intents), pattern and responses.

We use a special recurrent neural network (LSTM) to classify which category the user’s message belongs to and then we will give a random response from the list of responses.

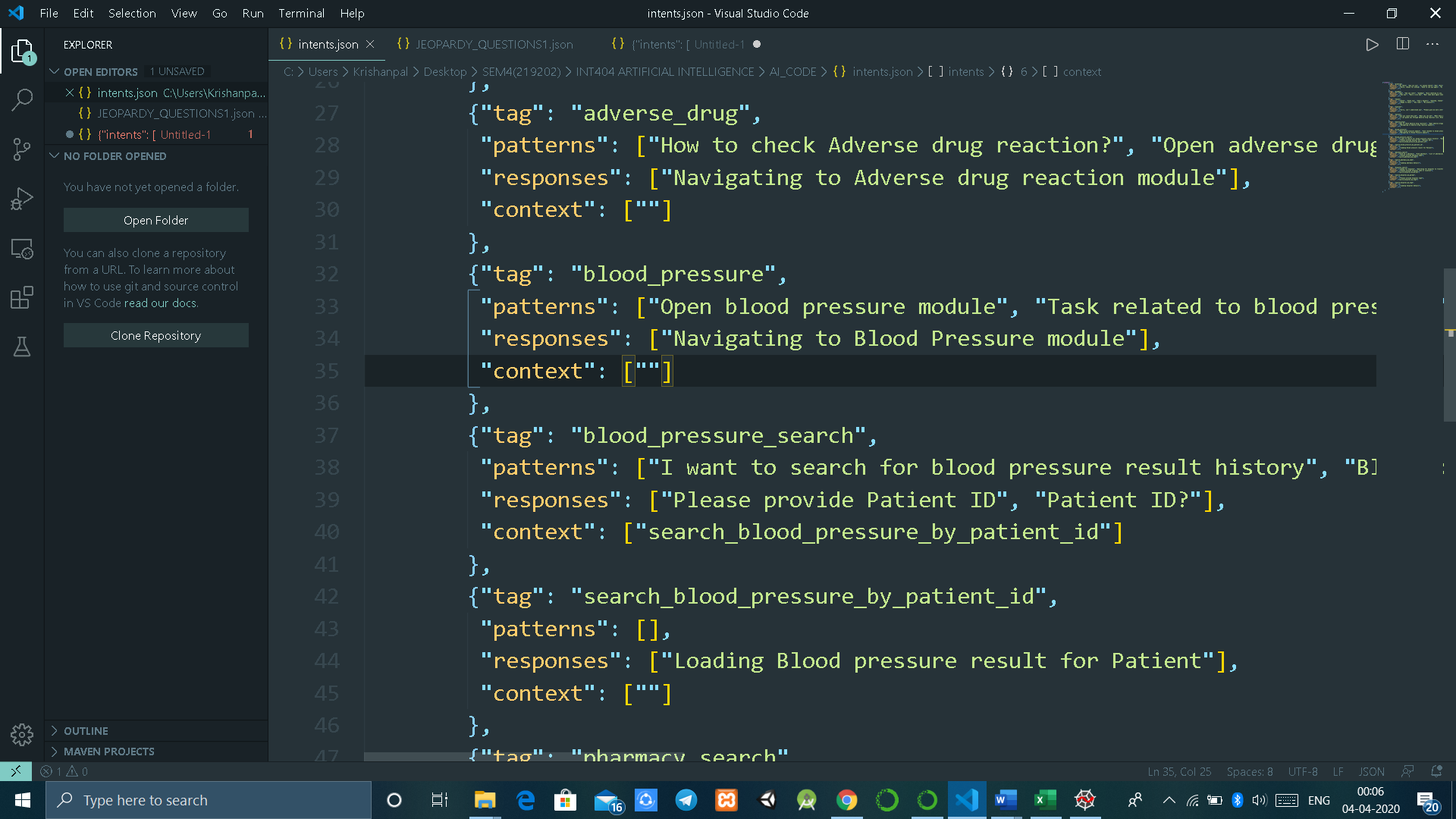
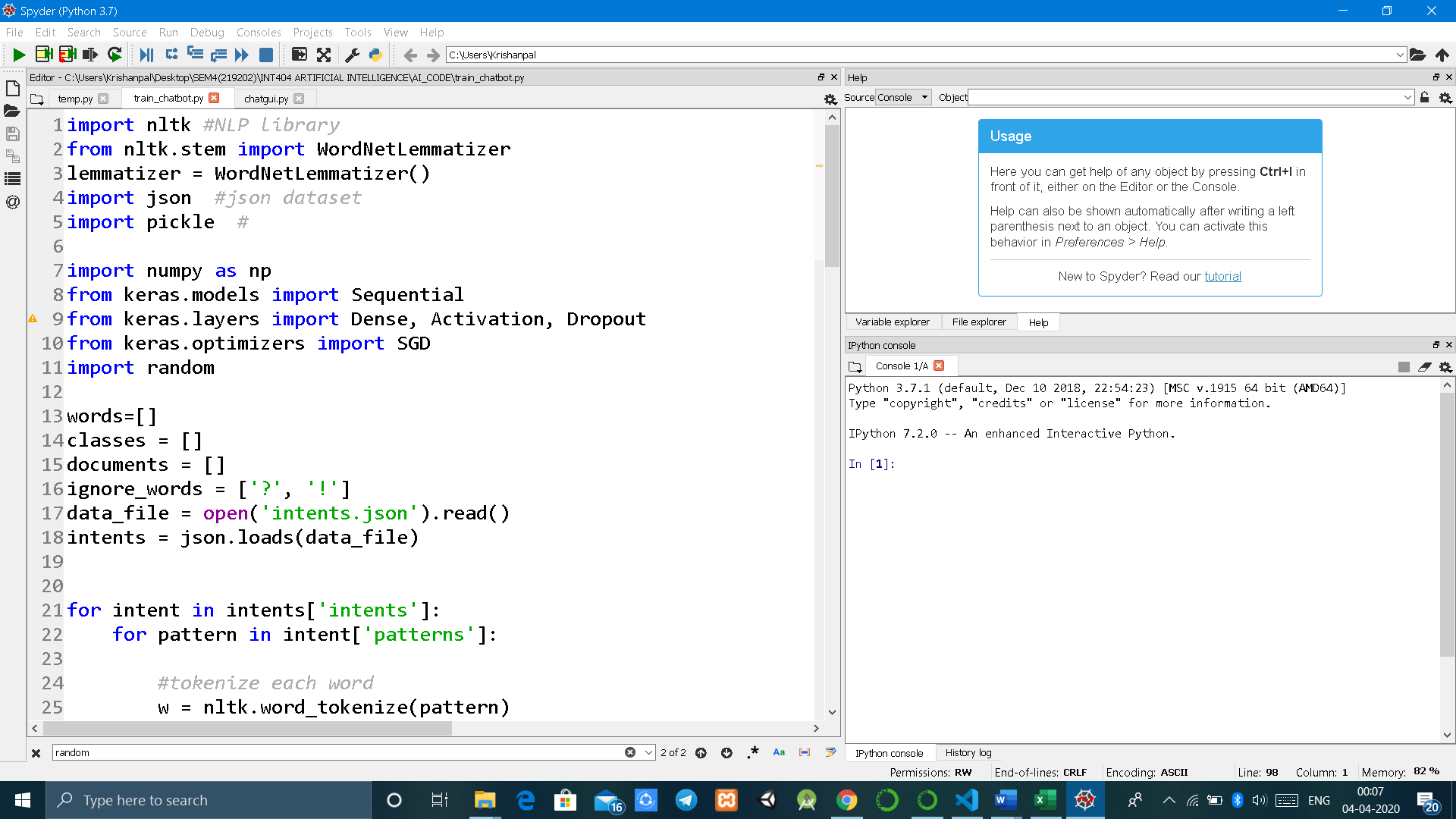
The dataset we will be using is ‘intents.json’. This is a JSON file that contains the patterns we need to find and the responses we want to return to the user.

* **Intents.json –** The data file which has predefined patterns and responses.
* **train\_chatbot.py –** In this Python file, we wrote a script to build the model and train our system.
* **Words.pkl –** This is a pickle file in which we store the words Python object that contains a list of our vocabulary.
* **Classes.pkl –** The classes pickle file contains the list of categories.
* **Chatbot\_model.h5 –** This is the trained model that contains information about the model and has weights of the neurons.
* **Chatgui.py –** This is the Python script in which we implemented GUI for our system. Users can easily interact with the system.

Here are the 5 steps to create this system :

1. Import and load the data file
2. Preprocess data
3. Create training and testing data
4. Build the model
5. Predict the response

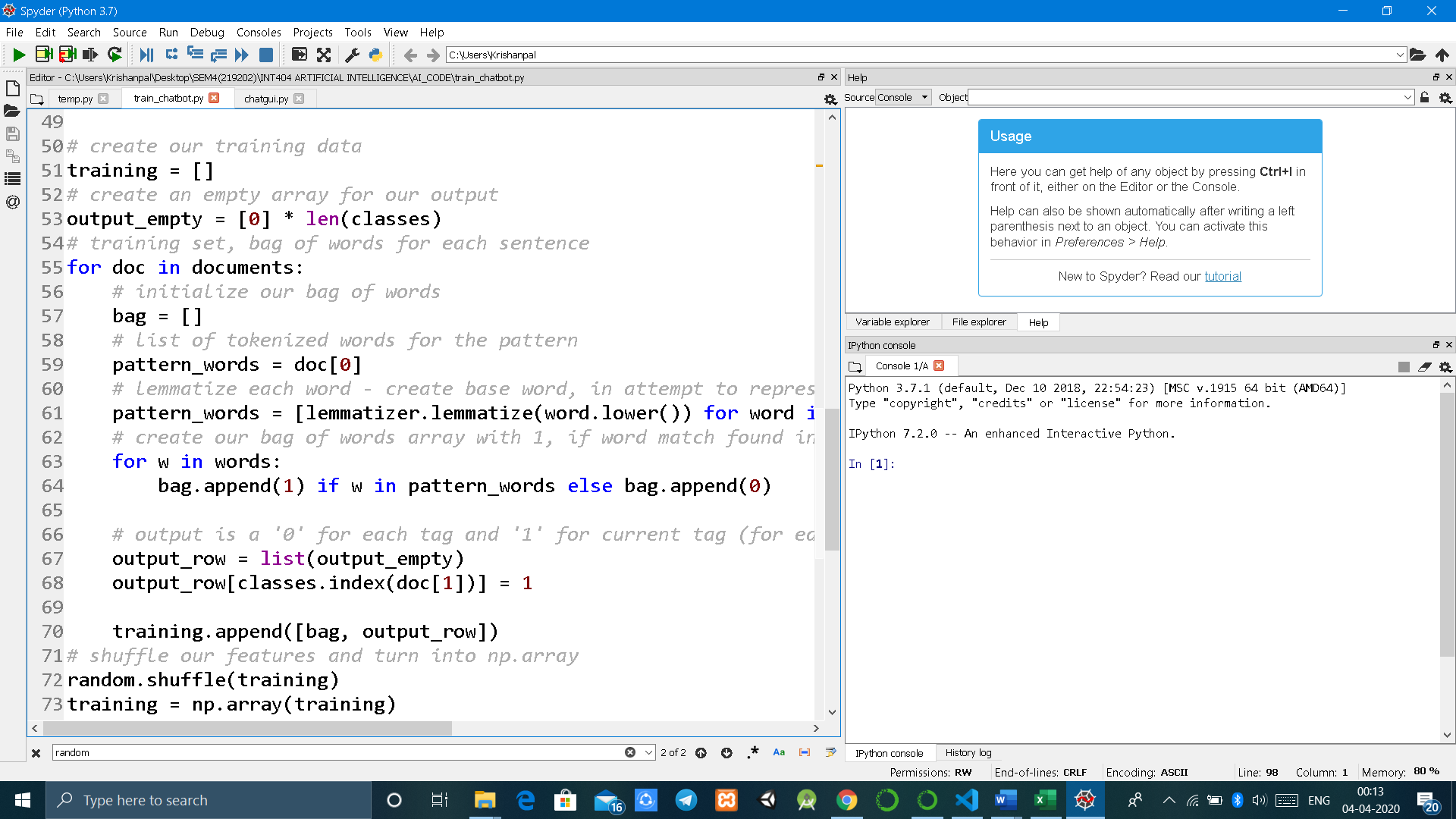
**1. Import and load the data file**



**2. Preprocess data**

Tokenizing is the process of breaking the whole text into small parts like words. Now we iterate through the patterns and tokenize the sentence using nltk.word\_tokenize() function and append each word in the words list.

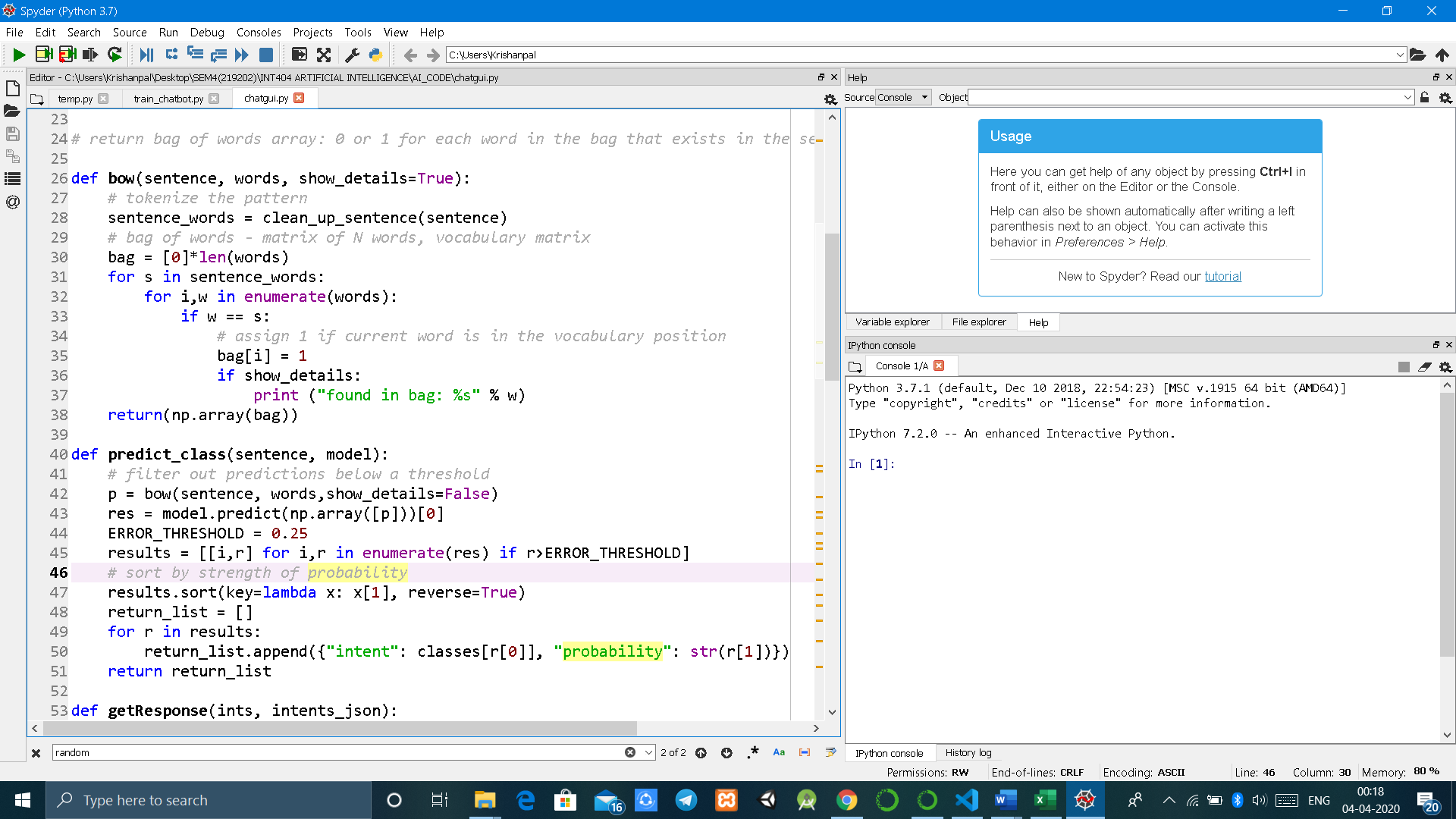
**3. Create training and testing data**



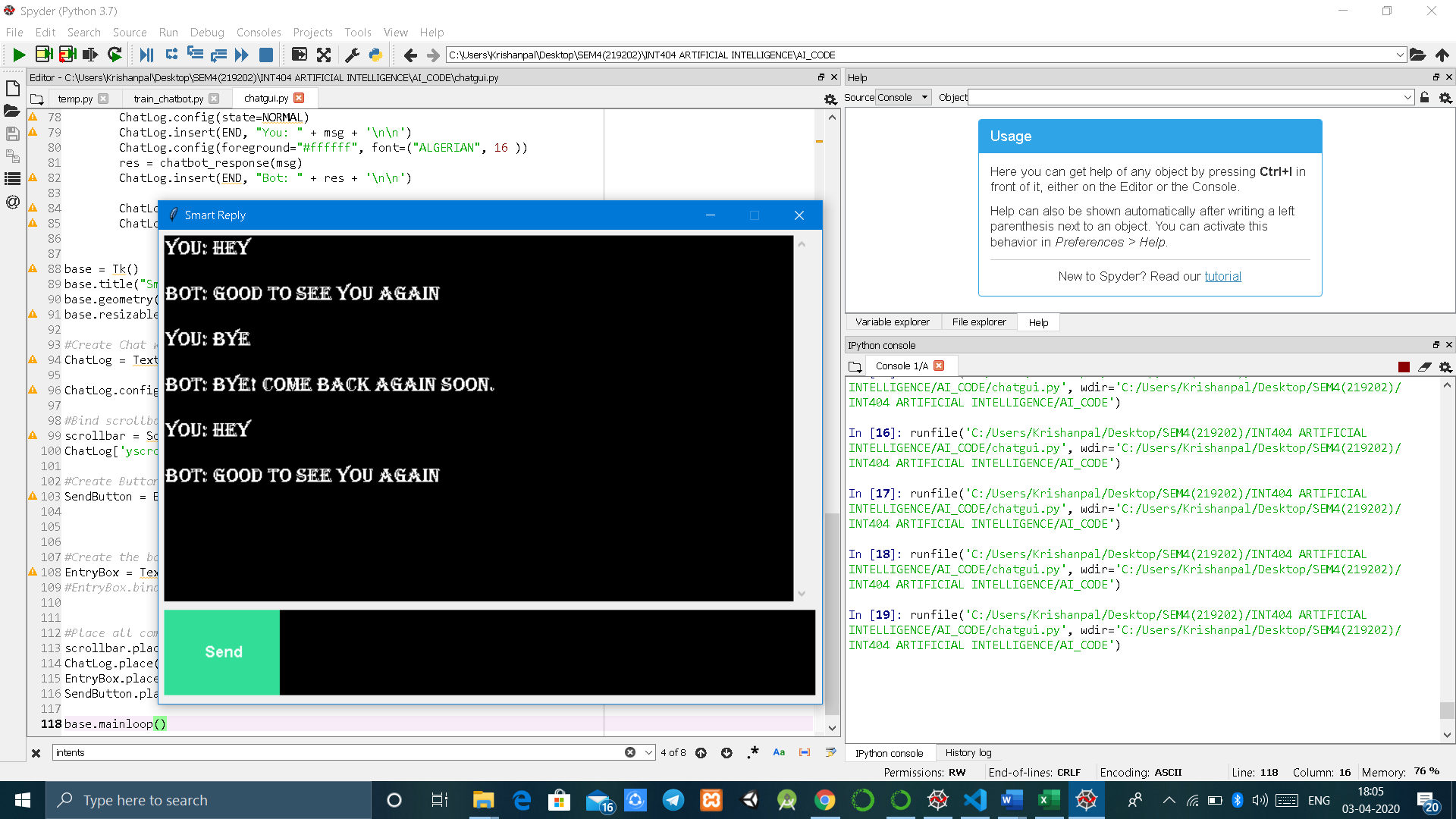
**4. Build the model**

Now data training is done , now we will build a deep neural network that has 3 layers. We use the Keras sequential API for this. After training the model for 200 epochs, we achieved 100% accuracy on our model. Let us save the model in a file.

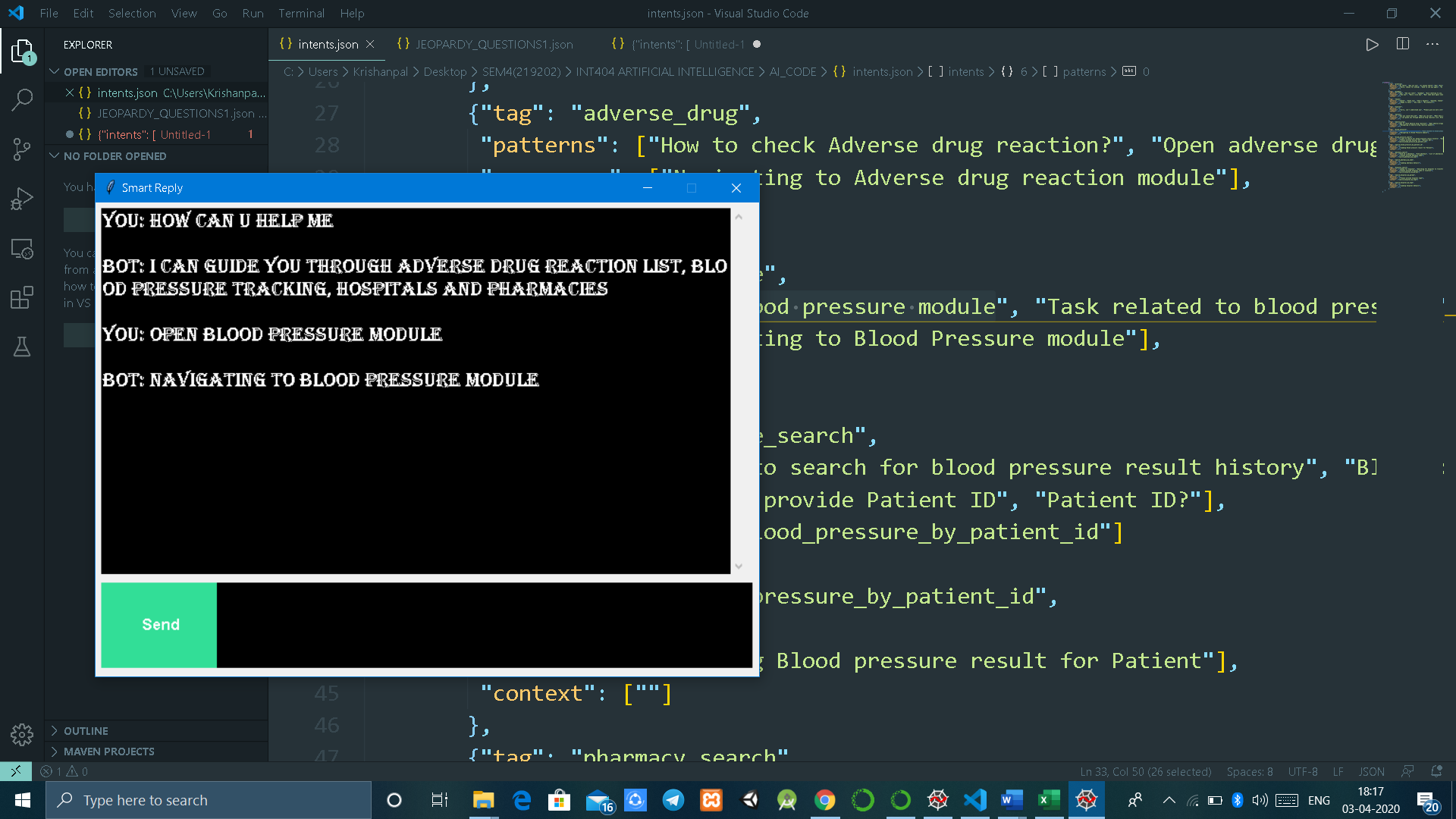
**5.Predict the response**



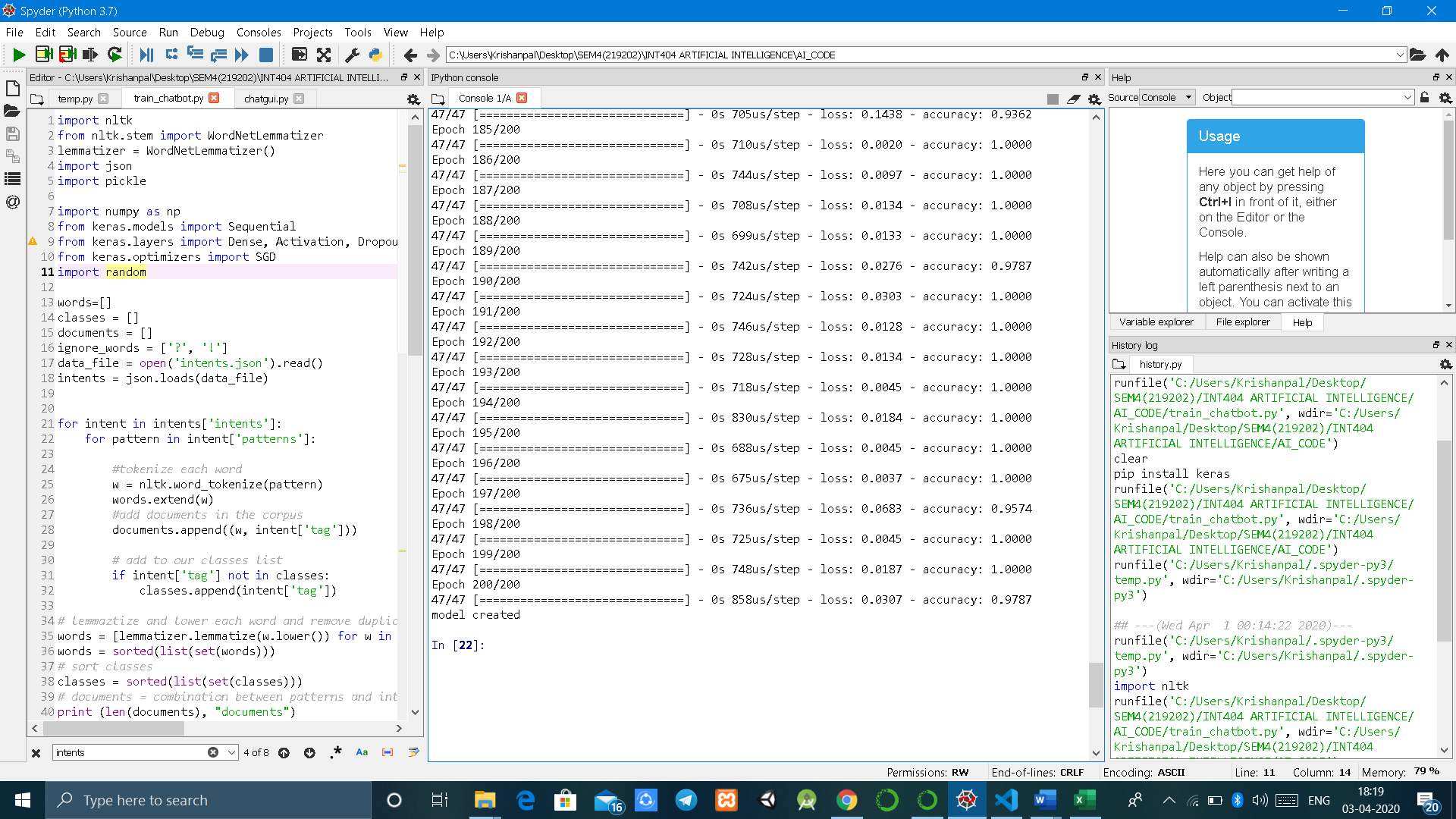
**RESULT (OUTPUT/ SCREENSHOTS)**



PIC: GETTING INITIAL RESPONSES (GREETINGS,GOODBYES)



PIC: GETTING THE SPECIFIC REPLIES (CERTAIN QUERIES)



PIC: MODEL CREATION

**IMPORTANT LIBRARIES USED**

**1.NLTK** : It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning. Tokenization is a very common task in NLP it is basically a task of chopping a character into pieces, called as token, and throwing away the certain characters at the same time, like punctuation.

The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. For instance:

am, are, is $\Rightarrow$ be  
car, cars, car's, cars' $\Rightarrow$ car

The result of this mapping of text will be something like:

the boy's cars are different colors $\Rightarrow$ the boy car be differ color.

**Stemming** usually refers to a crude heuristic process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. 

**Lemmatization** usually refers to doing things properly with the use of a vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma.

**2. Pickle**: Pickling is a way to convert a python object (list, dictionary, etc.) into a character stream. The idea is that this character stream contains all the information necessary to reconstruct the object in another python script.

import pickle

list=[‘a’,’b’,’c’,’d’]

with open(‘data.txt’,’wb’) as fh:

pickle.dump(list,fh)

In the above code, list – “list” contains four elements (‘a’, ‘b’, ‘c’, ‘d’). We open the file in “wb” mode instead of “w” as all the operations are done using bytes in the current working directory. A new file named “data.txt” is created, which converts the list data in the byte stream.

import pickle

pickle\_off=open(‘data.txt’,’rb’)

emp=pickle.load(pickle\_off)

print(emp)

**3.Numpy :** NumPy is the fundamental package for scientific computing with Python. It contains among other things:

* a powerful N-dimensional array object
* sophisticated (broadcasting) functions
* tools for integrating C/C++ and Fortran code
* useful linear algebra, Fourier transform, and random number capabilities.

**4. Random :** The random.choice() method returns a randomly selected element from the specified sequence. The sequence can be a string, a range, a list, a tuple or any other kind of sequence.

**5. Keras :** Keras is a minimalist Python library for deep learning that can run on top of Theano or TensorFlow.

**6. JSON :** Python comes with a built-in package called json for encoding and decoding JSON data. Import json is the code putting right at the top of the file to bring functionality of json into python.

**7.Tkinter** : Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

**TEAM RESPONSIBILITIES**

Single member for the project.

**SCOPE OF THE PROJECT(ADDITIONAL)**

The dataset can be increased to increase the feature of the system and provide better usability to the customer.

**COMMANDS TO RUN THE PROJECT**

1.Run the tain\_chatbot.py file to create the model.

2. After successful running of the above file, run the gui file that will display the system’ gui and you can ask questions given in JSON file. Accordingly, the system will respond.

**REFERENCES**

**1.** <https://www.tensorflow.org/lite/models/smart_reply/overview>

**2.** <https://becominghuman.ai/natural-language-processing-in-python-3-using-nltk-fd0ff4a0da9b>

3. GOOGLE FIREBASE API

4. <https://github.com/gabrielfarah/QA_Bot>