## Project 3: Create a chatbot in Python

Selecting a machine learning algorithm :

* Naïve Bayes Algorithm.
* Support vector Machine.
* Natural language processing (NLP)
* Recurrent neural networks (RNN)
* Long short-term memory (LSTM)
* Markov models for text generation.
* Grammar and Parsing Algorithms.

Hardware Requirements:

Processor : Pentium IV(minimum)

Hard Disk : 40GB

RAM : 256MB (minimum)

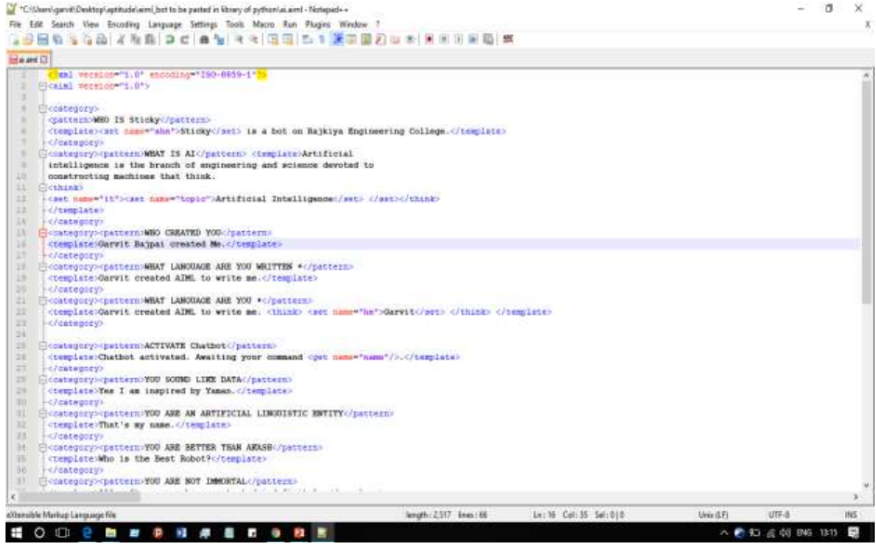
Software Requirements:

Operating System : Windows or Linux

Technology : PYTHON, AIML

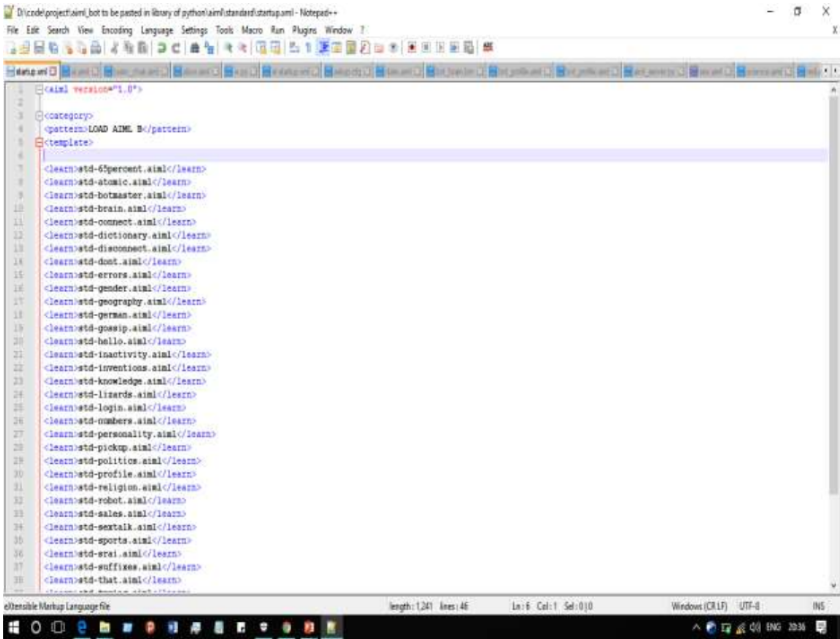
AIML Scripting :

* We created the AIML file that only handles one pattern, load aiml b. When we enter that command to the bot, it will try to load basic\_chat.aiml.
* It won't work unless we actually create it. Here is what you can put inside basic\_chat.aiml. We will match two basic patterns and respond



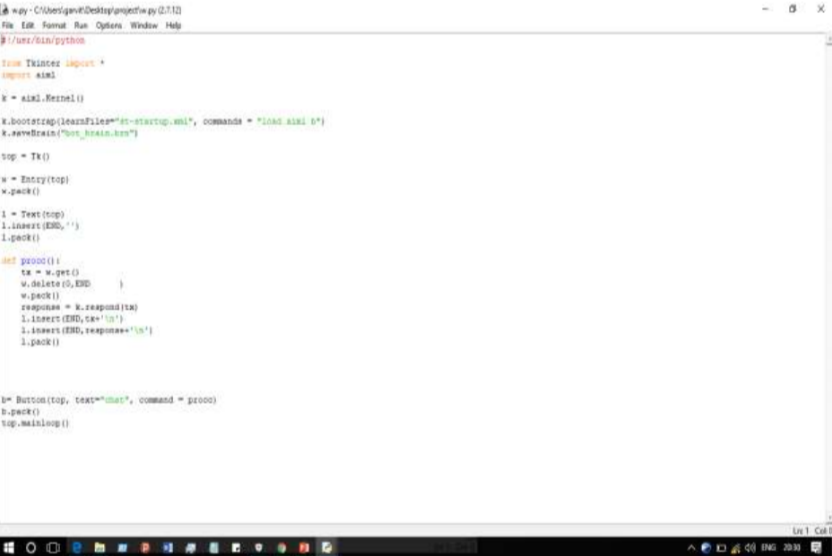
Creating a Startup File :

* It is standard to create a startup file called std-startup.xml as the main entry point for loading AIML files.
* In this case we will create a basic file that matches one pattern and takes one action. We want to match the pattern load aiml b, and have it load our aiml brain in response. We will create the basic\_chat.aiml file in a minute.



Creating Interface :

* The interfaces are the front end chat box for user to talk to the bot, which can be the Bot Portal, Skype, Facebook, etc. The connectorworks as a common gateway for all the interfaces.
* The outbound side calls different APIs to different front end, but the inbound APIs kept the same for our bot to connect.
* Fortunately, this connector has already been implemented by the bot framework SDK, we only need to rightly configure them.



Speeding up Brain Load :

When you start to have a lot of AIML files, it can take a long time to learn. This is where

brain files come in. After the bot learns all the AIML files it can save its brain directly to a

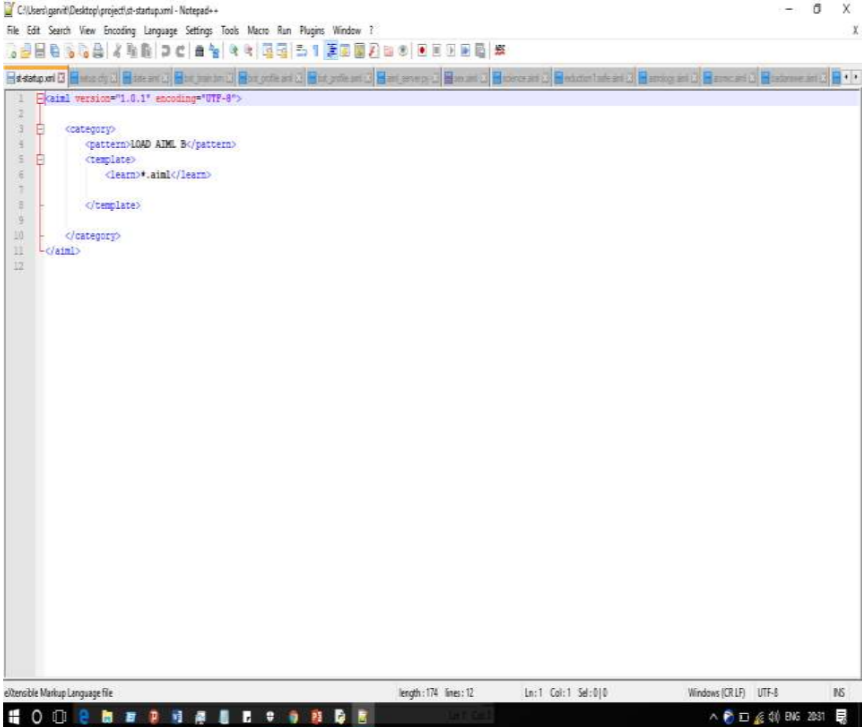
file which will drastically speed up load times on subsequent runs.

When you start to have a lot of AIML files, it can take a long time to learn. This is where

brain files come in. After the bot learns all the AIML files it can save its brain directly to a

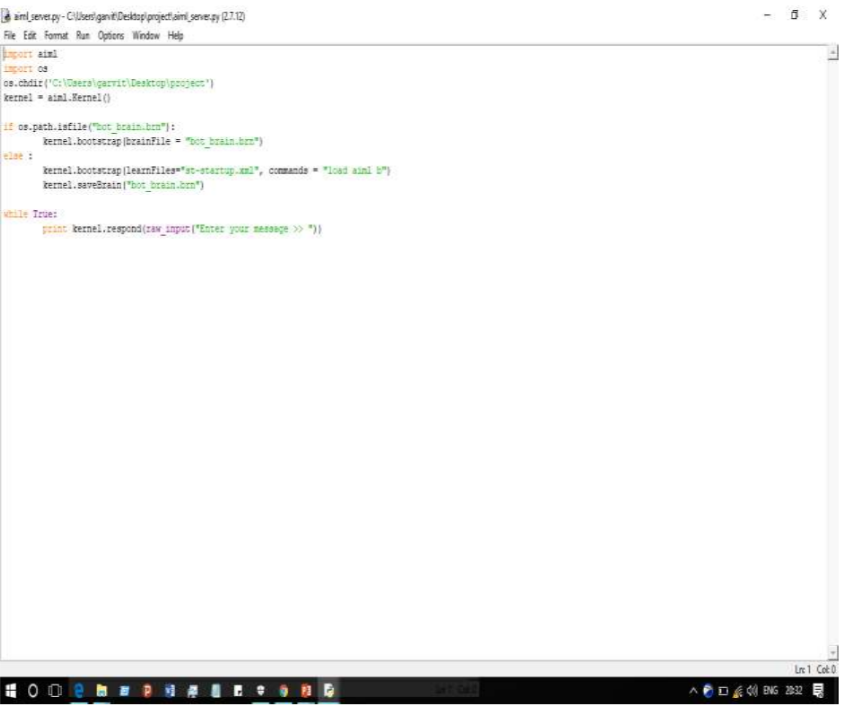
file which will drastically speed up load times on subsequent runs.

When you start to have a lot of AIML files, it can take a long time to learn. This is where brain files come in. After the bot learns all the AIML files it can save its brain directly to a file which will drastically speed up load times on subsequent runs.

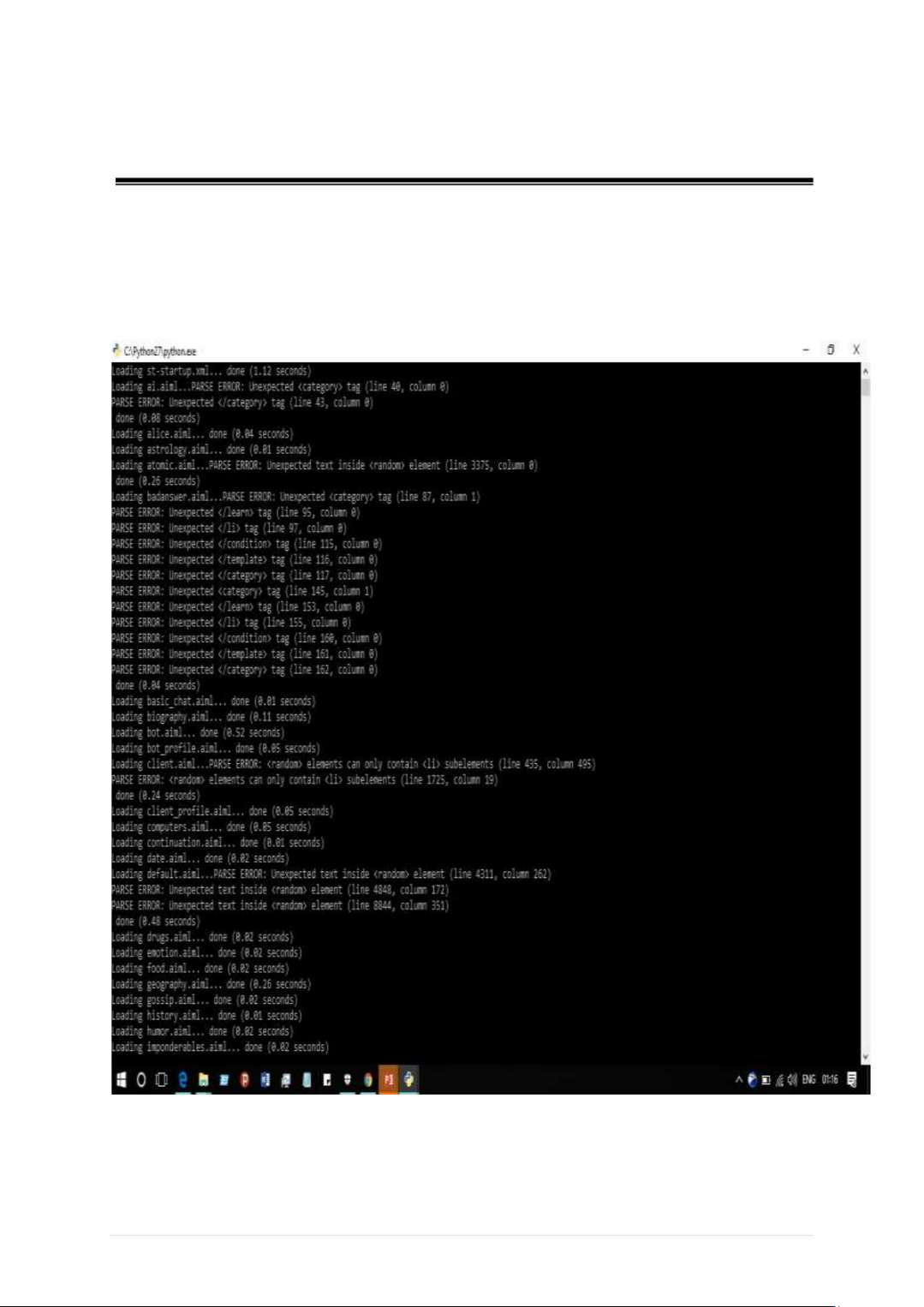


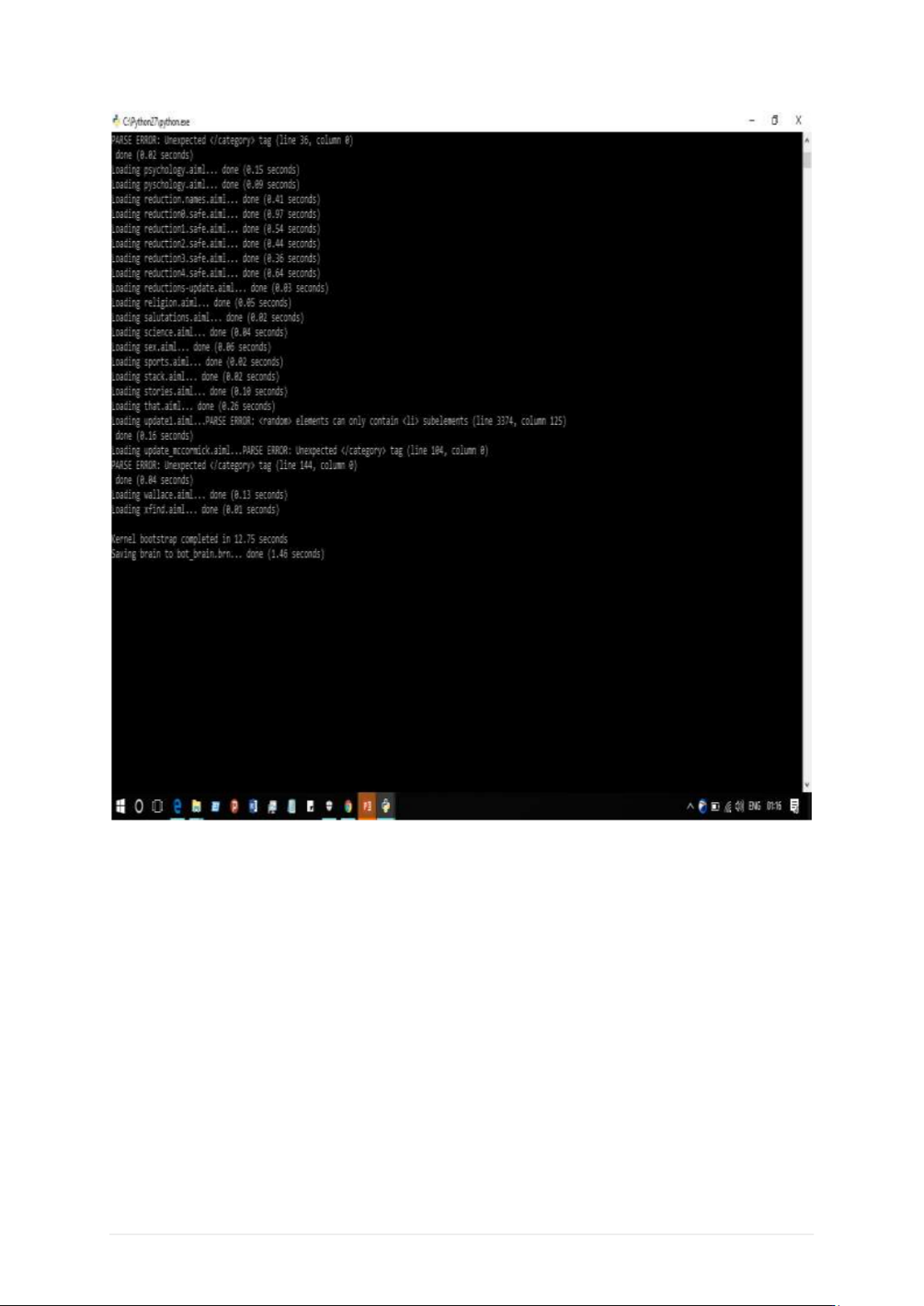
Loading Brain :

The patterns recognized depend on what AIML files you loaded. We create the startup file as a separate entity so that we can add more aiml files to the bot later without having to modify any of the programs source code. We can just add more files to learn in the startup xml file.

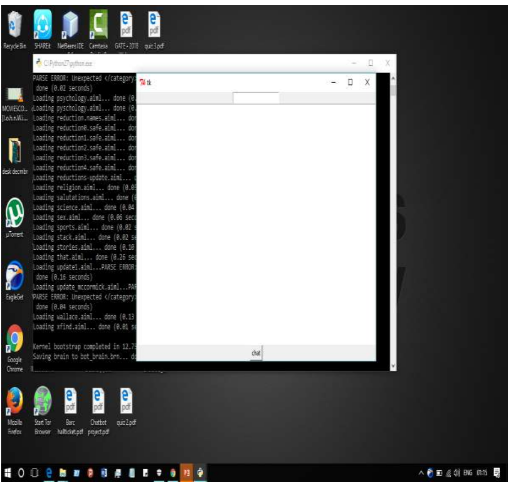


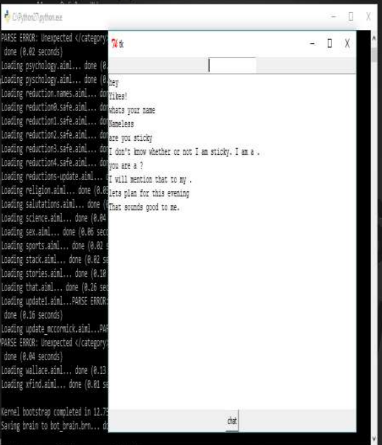
Implementation :





Chatbot Interface :





Advantages :

* Versatility
* simplicity
* abundant libraries for natural language processing

Disadvantges :

* If a consumer does not provide the correct command
* They will not give the correct response

Applications :

* The ChatterBot library combines language corpora
* Text processing
* Machine learning algorithms
* Data storage and retrieval to allow you to build flexible chatbots.