**Project Documentation**

**Title**: Rule Based Chatbot that plays Language Games with the user

**Keywords**: Translation, String manipulation, Regular Expressions, Modules, Chatbot

**Introduction**

The main goal of this project is to create a rule-based chatbot that interacts with the user by doing the following:

1. Greetings
2. Personalized messages
3. Menu driven options
4. Flexible selection and detailed descriptions

The bot – known as **Jabberwocky** is a chatbot that interacts with the user and enables the user to play language games or create translations to coded languages (similar to Pig latin from across the world). Jabberwocky also has an option to morphological stem words from stories based on user input in different languages. The language currently available for this option is “Tamil”

**Language Games**

A language game commonly known as argots or ludlings are games that are used primarily by groups who want to encrypt their conversations. These are usually done for entertainment purposes like kid’s games. Some argots have also evolved historically because of marginalized groups who created these secret languages to converse covertly.

Language games show systematic rules which involve mutations applied to syllables, vowels, consonants and in come cases even tones or meanings. Common languages include pig latin and versions of pig latin across the world.

This project covers three different types of Language games

**UbbiDubbi**: UbbiDubbi is a popular ludling popularized by TV programmes and mainly used by English speakers. Different versions of UbbiDubbi are used by English speakers. The version used in this project is explained with rules and examples in the following section

**King Tut:** King Tut is a secret language developed by African American slaves, to encrypt their conversations from their oppressors, because literacy among slaves was forbidden. It works using a substitution cipher wherein the letters of the alphabet are substituted with a specific set of sounds. The version used in this project is explained with rules and examples in the following section

**The Name Game:** The name game is a popular song by Shirley Ellis from 1964. The song has now become a popular parlour game where a rhyming song is created from nonsense verses applied in a systematic way to proper nouns. The version used in this project is explained with rules and examples in the following section

**Chatbot coding in Python**

Chatbots or conversational agents are coded in languages like Python using two main methods. These are rule based or learning based. Chatbots developed using learning-based techniques are trained on large datasets and use pattern matching and keyword similarity to respond to user queries. For the purpose of this project, I have created a simple rule based chat bot (re: Jabberwocky) which generates appropriate responses to user input. The chatbot offers some flexibility by choosing from a randomized set of responses to different situations – mainly greetings, exit, menu driven options etc

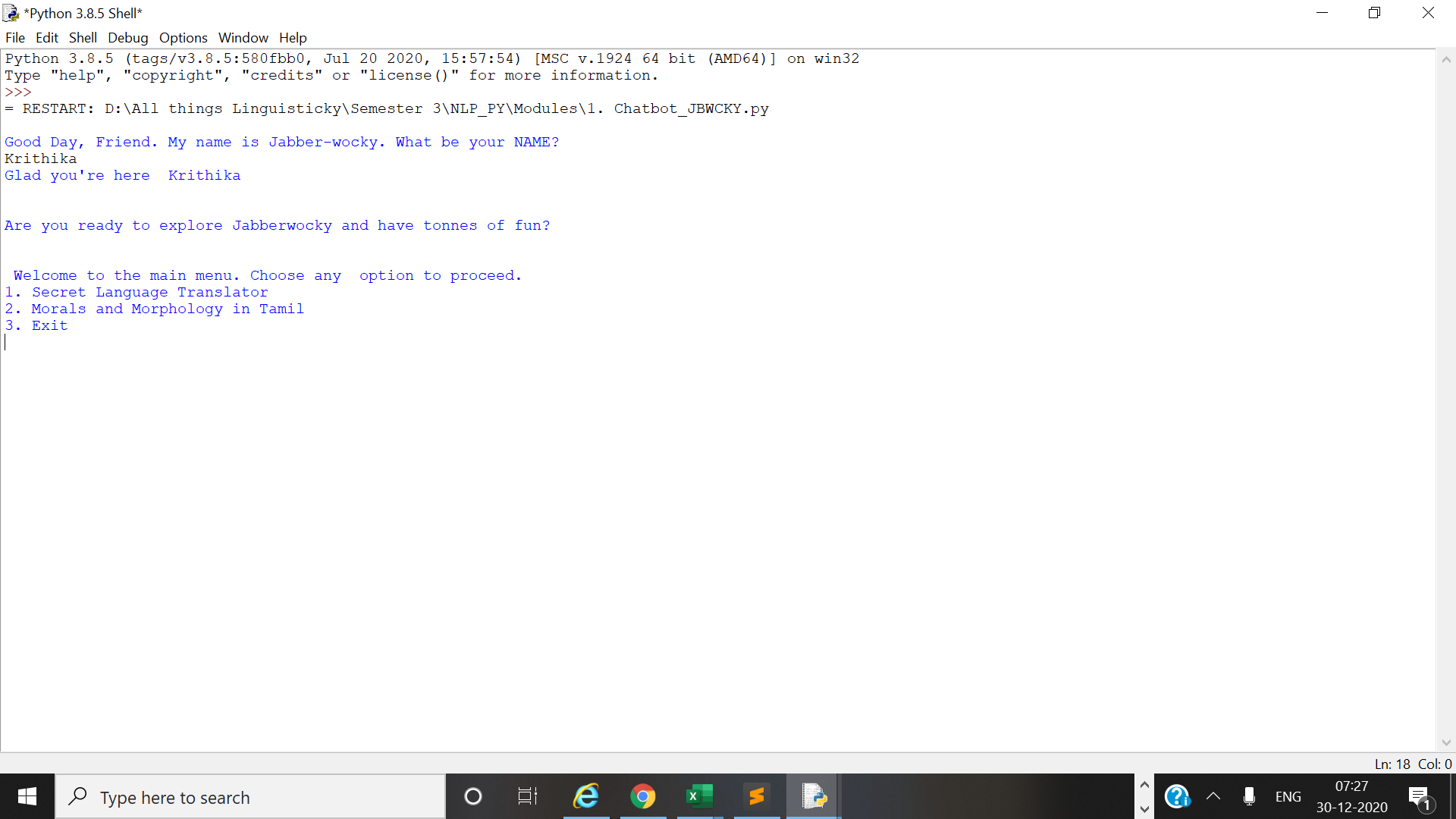
**Problem Statement:**

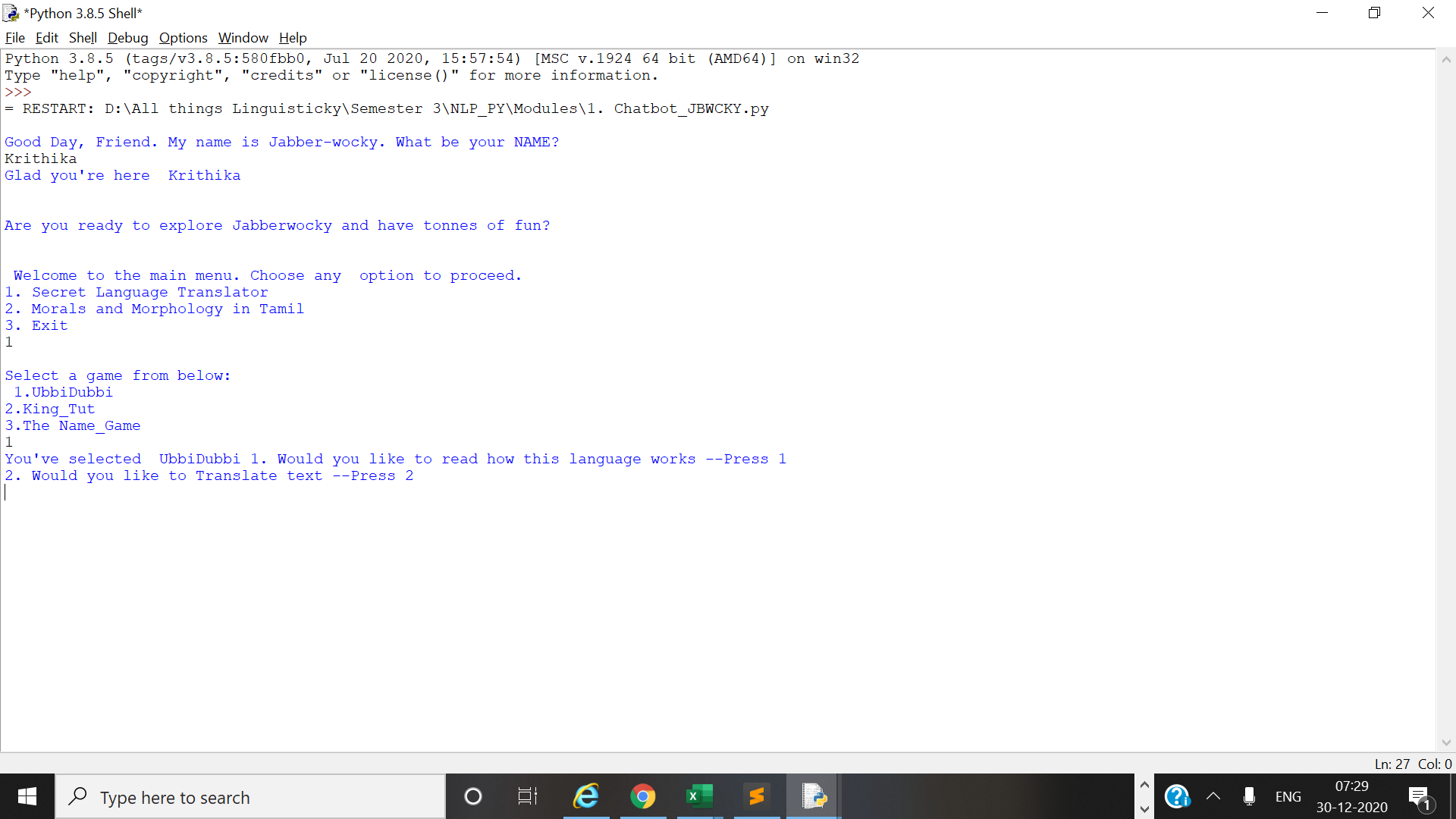
Create a simple chatbot algorithm to greet the user, present the user with multiple menu-driven options to play language games of their choice, provide information on the options chosen, and ultimately provide translation of user input text to the desired secret language. The bot also provides an option to morphologically analyse some text files in a given language (In this case – moral science stories in Tamil. The text is transliterated in English)

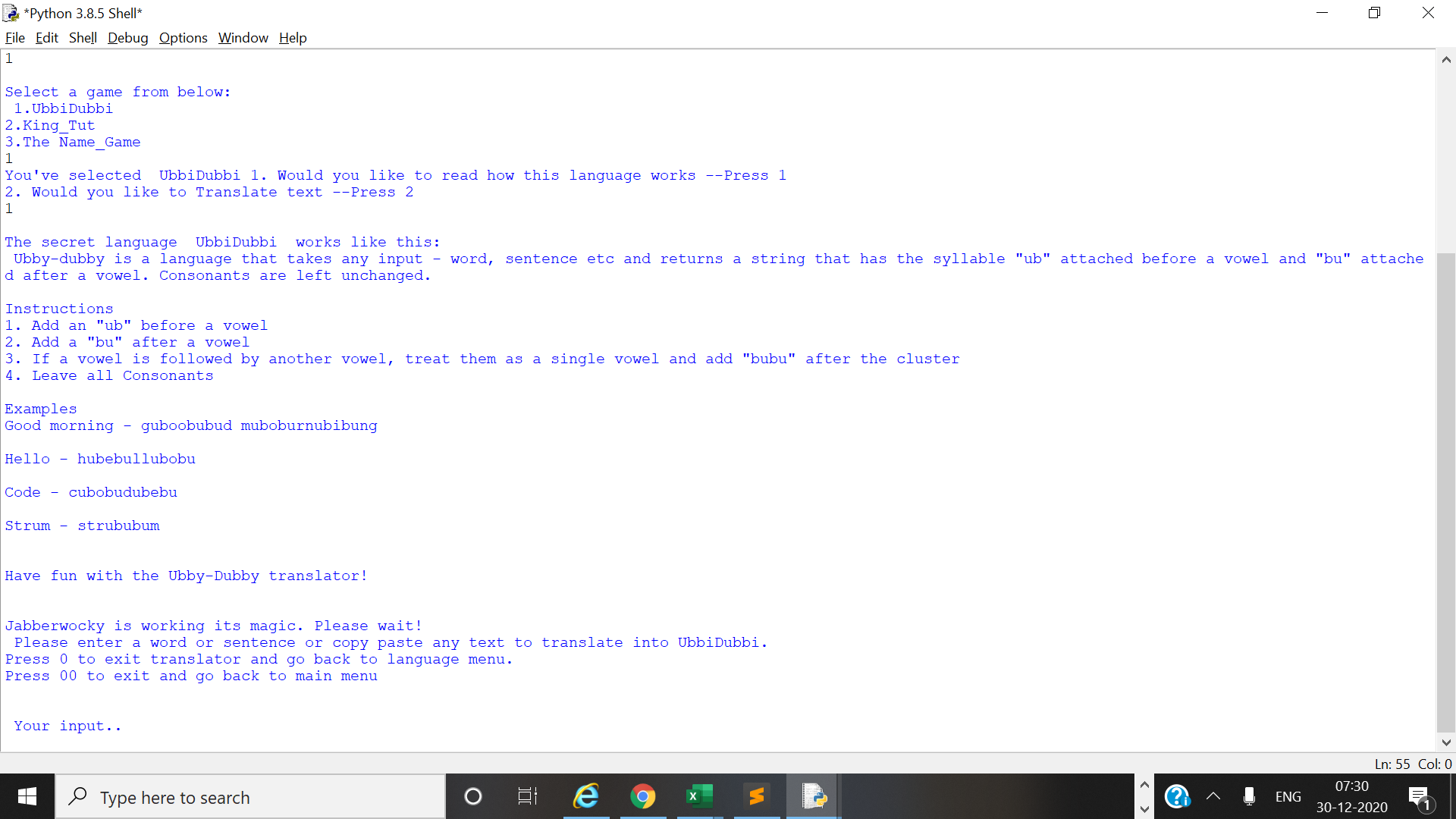
**How to use the program**

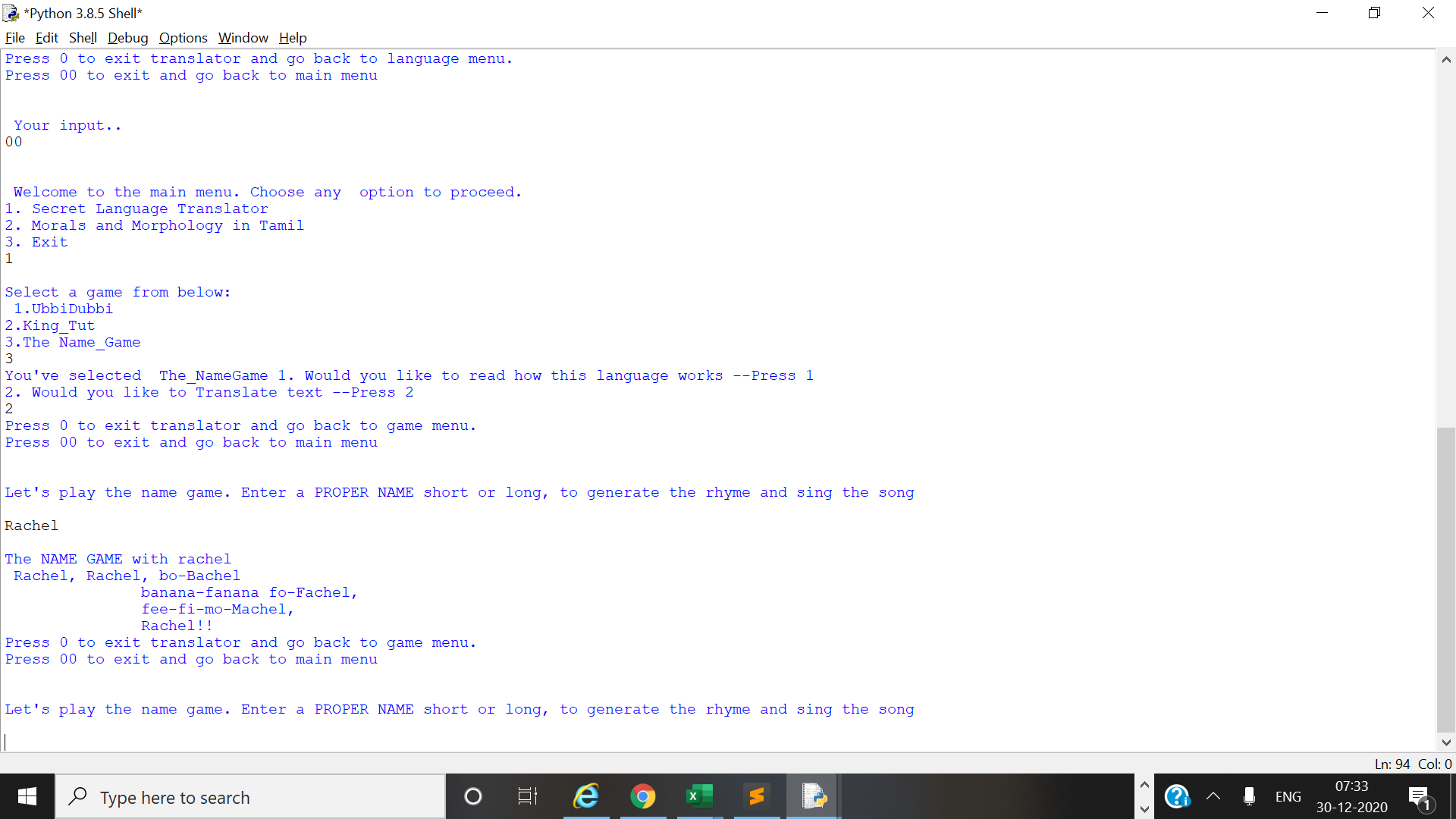
The user only needs to interact with 1 python file which contains the chatbot specifications. This is called 1. Chatbot\_JBWCKY.py. Please execute this file once, while the other files (“.py and .txt files in the folder) don’t need to be open.

The welcome screen is like below:















**Notes:**

I have not used any kind of inbuilt libraries or databases for this project. NLTK and other direct output tools are not used in this code. Standard python functions, knowledge gained during the course, user defined functions and regular expressions mainly form the crux of this code. I have tried to incorporate all the learnings from this semester – Control structures, File I/O operations, User defined functions, Language games, Syllabification, stemming, modules, regular expressions, string functions, formats and encodings etc to complete this project.

**Limitations**

Since the data used for this project is Transliterated data from Tamil, the stemming algorithm is not the most efficient. I have tried to separate out the different markers for affix stripping, and hope to improve the code in the future by using more data and possibly better translation tools.

My chatbot algorithm is rule based and is not trained on datasets. Hence the limitations are that it can work with very limited functionality. Within the framework of rule based coding, I have tried to incorporate different interactive options without doing too much manual coding.

The scope of the project can be widened by doing the following

1. Use Lexical corpora for Tamil-English for morphological analysis
2. Train the chatbot to learn from datasets and question answer systems
3. Incorporate GUI based code snippets to improve the user interface
4. Use better annotated and POS tagged datasets for effective stemming

**References**

1. **Schwartz, Alvin - The Cat's Elbow and Other Secret Languages (2010)**
2. **Hammond, Michael – Python for Linguistis**
3. [**https://en.wikipedia.org/wiki/The\_Name\_Game**](https://en.wikipedia.org/wiki/The_Name_Game)