**SYNOPSIS**

Word Corrector is an interface that enables the user to correct the spelling of the word. Word Corrector is helps the user and is real time technology. This process saves the time of user, because the user can find the lists of similar words which is being typed in the textbox. The working prototype was developed and some functionalities are highlighted. The proposed system helps reducing and minimizing human error. The proposed system decreases the complexity of the users. In this way the user is motivated to correct the spelling as the Word Corrector helps the automated corrected words.

**ACKNOWLEDGEMENT**

At First , We would like to thank Director Dr.Sr. Lalitha for giving the opportunity to do this project for endless support and providing us with the environment and faculty to complete this project successfully.

We express our heartfelt gratitude to our project guide Ms.Irene Getzi , Head of MCA Department for all the valuable guidance and support given for us , helping us to complete this project successfully and well within the stipulated time.

We would also like to thank entire faculty and staff of MCA Department and my friends who devoted their valuable time in completion of this project .

We also thank God Almighty for helping us to complete this project.

**1. INTRODUCTION**

Everyone knows the importance of spell checking in their word documents , etc.  But digital content often gets overlooked and web designers are notoriously bad at spelling! you may want to demonstrate first how the word corrector program we will be working . the program detects a misspelled word.

Google provides a more powerful word corrector for validating the keywords we type into the input text box. It not only checks against a dictionary, but, if it doesn’t find the keyword in the dictionary it suggests a most likely word. To do this it associates with every word in the dictionary, that word is expected to appear in a large document.

There is one major difference. This word corrector will only validate a single word rather than each word in a list of words. For this program we need a dictionary similar to Google’s. The dictionary is generated using a large text file. The text file contains a large number of unsorted words and non-words. A word is defined as a sequence of 1 or more alphabetic characters. Every time your program runs it will create the dictionary from this text file. The dictionary will contain every word in the text file.

If the input string (independent of case) is found in the dictionary, the program will indicate that it is spelled correctly by returning the input string (in lowercase). If the case independent version of the input string is not found, your program will return the most “similar” word (converted to lowercase).

It is usually provided as a dropdown menu of choices below the textbox in which the user is typing. The goal is to anticipate the text , the user intends to type so that instead of typing in the rest of the words, one of the dropdown choices can be selected. Some advantages of autosuggest are queries need not be typed in their entirety.

**1.1 Overview**

This project have chosen since these features help a user by reducing typing, by catching any spelling errors, and by making it easier to repeat searches.

This project aims to implement these computationally intensive functionalities in this constrained environment. This is achieved by performing any needed processing on the client-side without sending queries to the server.

**1.2 Background and Motivation**

It helps the user to easily find out the related words to which user approaching towards.

Since this projects provides the words in the dictionary similar to those when the user types, this helps the user to save the time.

The goal is to anticipate the text the user intends to type so that instead of typing in the rest of the words, one of the dropdown choices can be selected.

Therefore this process is quick as it shows the words very fast to the user when the word is typed. Even if there is a spelling mistake, there is no need to correct the word, because along with the similar words the correct words are also displayed simultaneously.

**1.3 Objectives**

The objective of developing such a computerization system is to reduce the paper work and save the time of users, thereby increasing the efficiency and decreasing the work load.

**ENVIRONMENT SPECIFICATION**

**2.1 Hardware specification.**

* **Hard disk : 65 or more**
* **RAM : 128MB or more**
* **Processor : Intel Pentium**
* **Clock : 133MHz**
* **Monitor : 15” Color**
* **Key board: Serial Keyboard**
* **Mouse : Optical Mouse**

**2.2 Software specification.**

**Operating System: Windows 7**

* **Front End : Python**
* **Back End : MySQL**

**3. REQUIREMENTS SPECIFICATION**

**3.2 Functional Requirements**

Functional requirement are the functions or features that must be included in any system to satisfy the business needs and be acceptable to the users.

Based on this, the functional requirements that the system must require are as follows:

* System should be able to analyze data and classify each words correctly.
* The system should count the distance of Words what the user’s Types.
* The system have to perform sentiment analysis on each Words and determine whether the Word is Correct or Incomplete.
* The system should display the Output correctly whatever the User Types.

**3.3 Non-Functional Requirements**

* Accessibility: To access the data listed on Tkinter, we have to download the Tkinter data from Pycharm terminal.
* Documentation: Proper comments are there within each file for explanation. For better understanding of the project, by extracting the feature words From Dictionary and thereby giving the correct suggestions.
* Maintainability: Codes does not need to be maintained if not altered.
* Response Time : Long reviews can take more time to pre-process it .

**4. SYSTEM DESIGN**

It describes the desired features and operations in detail, including screen layouts, business rule, process diagrams, pseudo code and other documentation. The most creative and challenges phase of the software development life cycle is software design. The term design describes final software and the process by which it is developed. The purpose of the design phase is to plan a solution of the problem specified by the requirements document. It also includes the construction of programs and program testing. Design takes us towards how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phase, particularly testing and maintenance .the output of this phase is the design document. The first step is to determine how the output is to be produced and in what format. Samples of the output and input are to present second, input data and master files (database) have to design to meet the requirements of the purposed output. The operational (processing) phases are handled through program construction and testing, including the list of the programs needed to meet the software objectives and complete documentation.

The design activity is often divided into two phases-system design and detailed design. System design, which is sometime also called top-level design, all the major data structures, file formats, output formats, and the major modules in the system and their specification are decided.

During detailed design, the internal logic of each of the modules specified in system design is decided. During this phase further details of the data structures and algorithmic design of each of the modules is specified.

In system design focus is on identifying the modules, whereas during detailed design focus is on designing the logic for each of the modules .in other words, in system design the attention is on what components are needed , while in detailed design how the component can be implemented in software is the issue.

The design of an information system produces the details that state how a system will meet the requirements identified during system analysis. Often systems specialists refer to this stage as local design, in contrast to developing program software, which is referred to as physical design.

As soon as the user accepts the system proposal, work can start on preparing the system specification. This phase takes the requirements as agreed and the work, which has lead up to producing the proposal and develops the system to the level of the details necessary to prepare the way for programming.at this point the analysts is concerned with the detail of input and output, the processing required, and the way in which the system will operate on a day to day bases. depending on the level of complexity of the system and the amount and the quality of work done at the earlier stages, this phase can take many months of hardwork.it is concerned with the computer-oriented design of the system—the detail of input transactions, the details of a printed reports screen, other outputs, the files or database structure, the contents of records, the processing required and the efficiency of the system from a computer processing point of view.

**4.1 Interface Design**

User interface is the front-end application view to which user interacts in order to use the

Software. User can manipulate and control the software as well as hardware by means of user

Interface.

User interface is part of software and is designed such a way that it is expected to provide

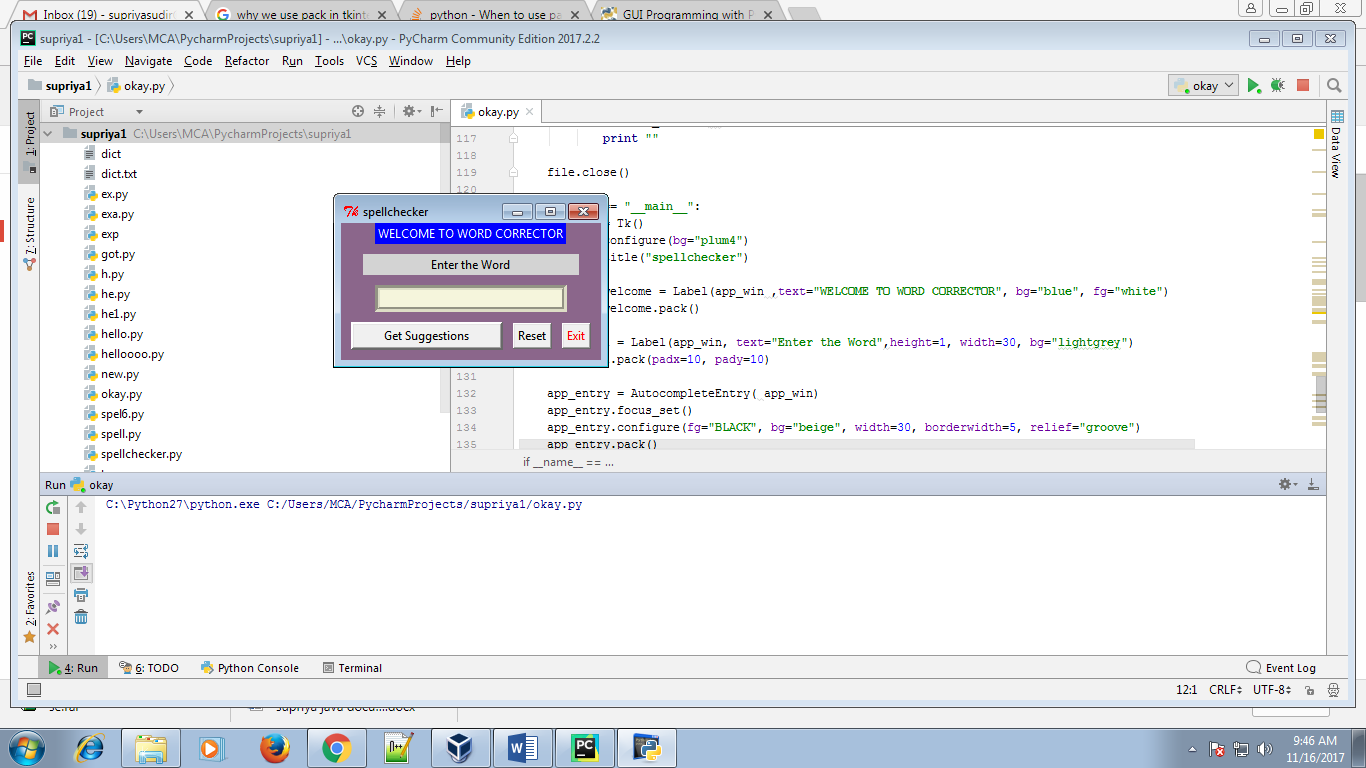
the user insight of the software. UI provides fundamental platform for human-computer

Interaction.

UI can be graphical, text-based depending upon the underlying hardware and software combination. UI can be hardware or software or a combination of both.

The software becomes more popular if its user interface is:

* Attractive
* Simple to use
* Responsive in short time
* Clear to understand
* Consistent on all interfacing screens.

****

tk() is the name given to tkinter package.

Configure is the method, used to access objects attributes after initialization.

Spellchecker is the name given to the form.

A label is created whose background color is blue and foreground color (text color) is white.

Text written is “WELCOME TO WORD CORRECTOR”.

Pack is the easiest to use of the three geometry managers of Tk and Tkinter . Instead of having to declare precisely where a widget should appear on the display screen, user can declare the positions of widgets with the pack command relative to each other. The pack command takes care of the details. Though the pack command is easier to use, this layout managers is limited in its possibilities compared to the grid and place managers. For simple applications it is definitely the manager of choice. For example simple applications like placing a number of widgets side by side, or on top of each other.

Another label is created, in which background color is Lightgrey and foreground color is black i.e. default. Text written is “ENTER THE WORD”.

Padx and Pady are variables used to initialize values. This variables are used for label padding i.e. the space between word typed and its cell.

When a word is entered it suggests the similar word which the user has entered in it . It will display all the similar words in which user can choose the appropriate and correct word.

The words which occur similar to the context entered in the text box are displayed simultaneously as the user starts typing the letters.

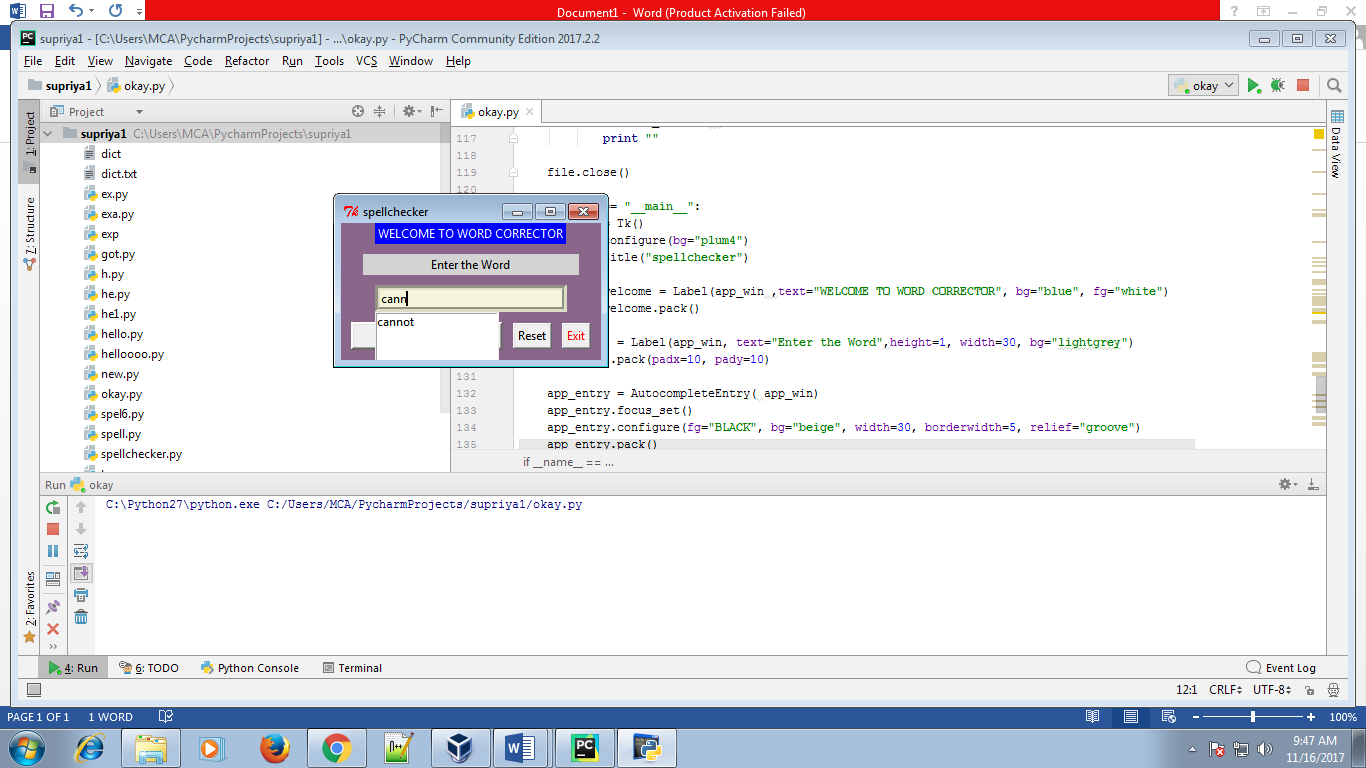
Focus\_set() method is used to blink the cursor. The user can use this cursor on the text bar to type words.

The relief=”goove” is the type of border used for textbar.

Finally the buttons are created , get suggestion is the name of the button.

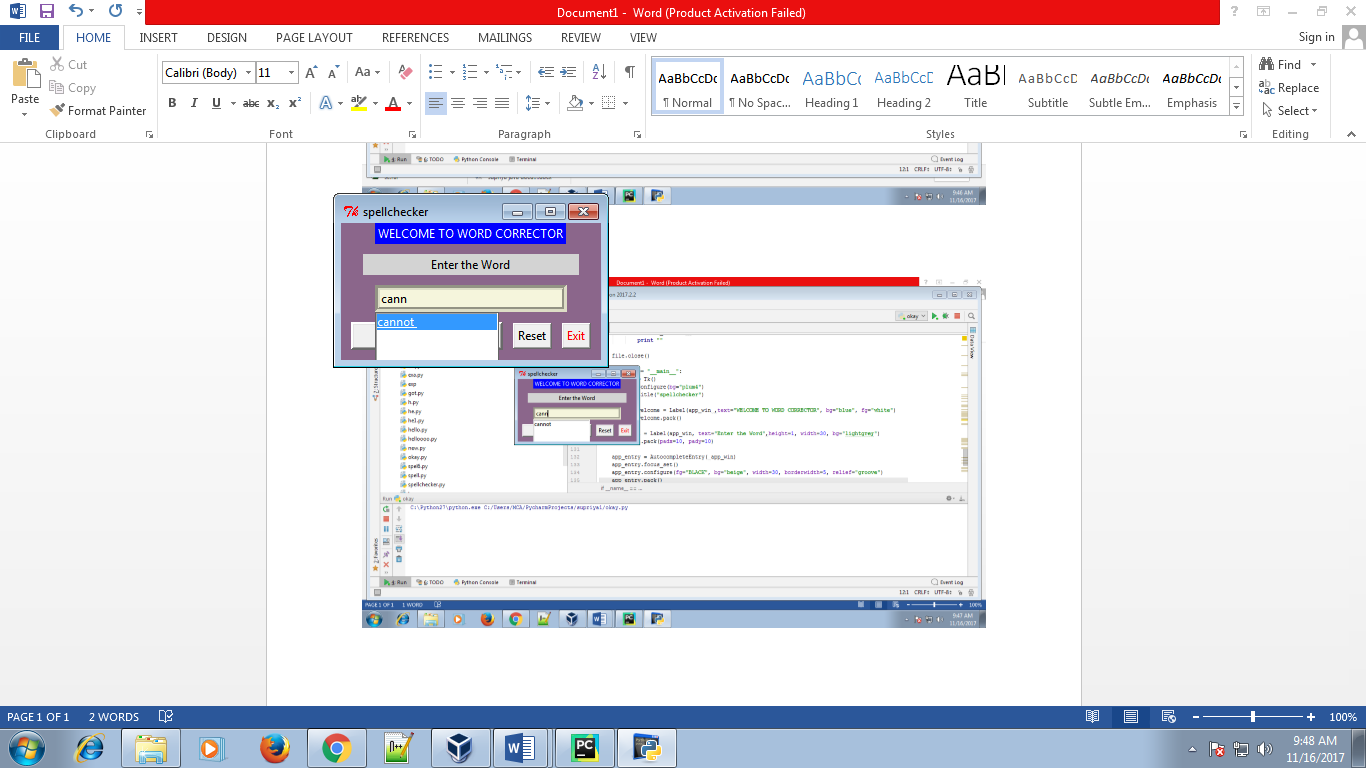
Exit button is created using a command i.e. app\_win.destroy. This command is available in Tkinter itself.

Reset button is created using command i.e. lambda. The thing to remember here is that a tk.button expects a function object as an argument to the command parameter. That function object will be the function that the button calls when it (the button) is clicked. Basically, that function specifies what the GUI will do when the button is clicked. So we must pass a function object in to a button via the commandparameter.



**Autocomplete entry**

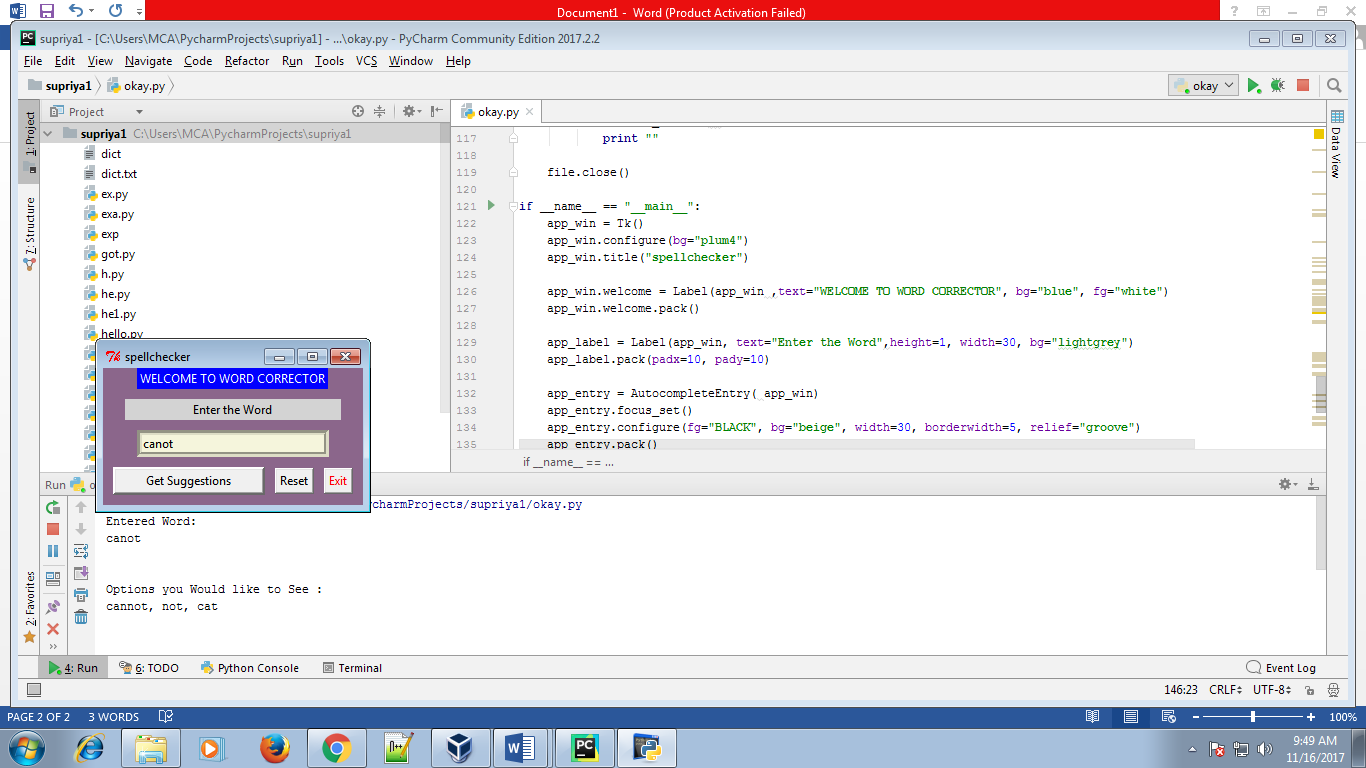
In the text box we are calling the auto entry function that function lists down the words what the user have typed in the textbox. If the word when user types is incomplete, it will list out and suggests the words available. After finding the wanted word the user must double click on that word so that it will automatically set in the label.



**Get Suggestion**

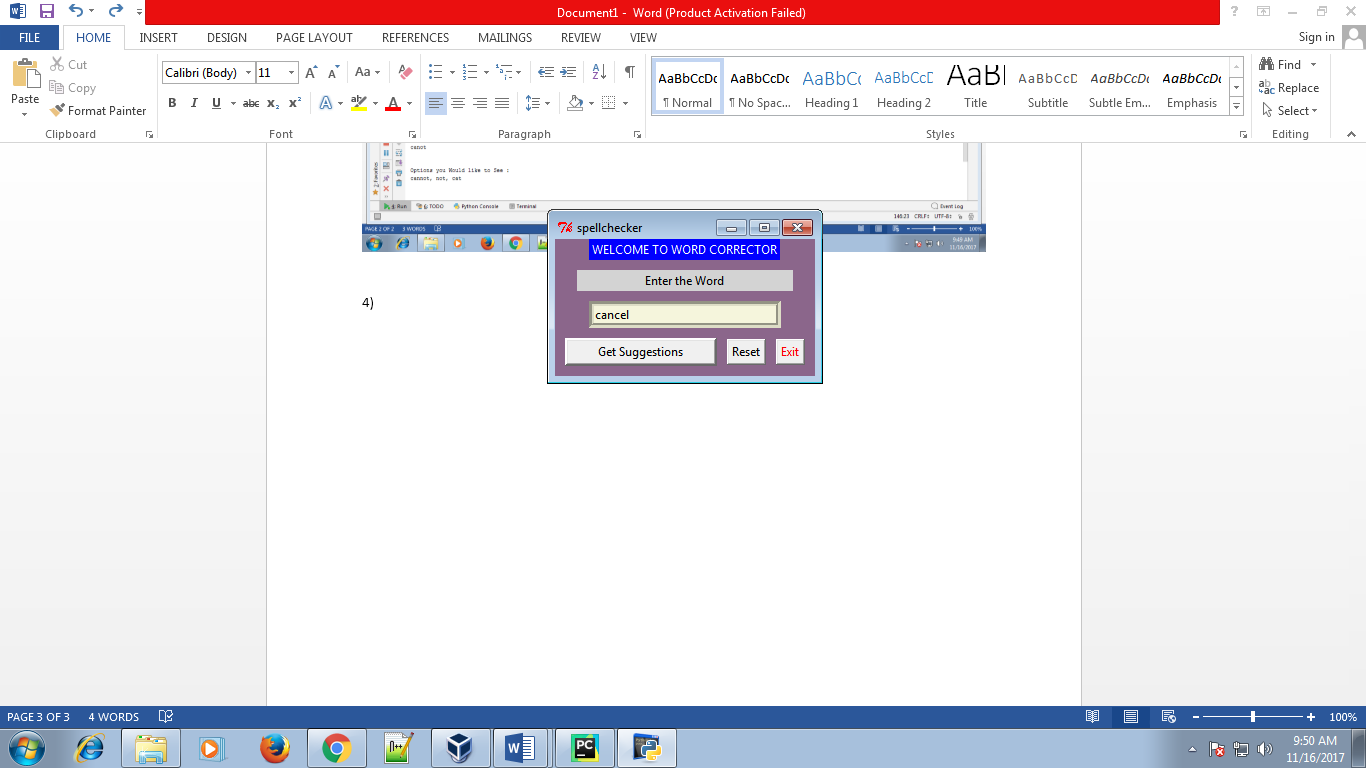
If the user is typing incorrect word, but still user wants to know the correct words, user can click on get suggestion button, hence the user is easily able to see the lists of similar words.

This helps the user to choose the correct words.



**Reset Button**

Reset button is used to clear all the entered context in the textbox.



**Exit Button**

Exit button is used to terminate the process.

**4.2 Data Flow Diagrams**

Get

Suggestion

WORD NOT PRESENT

CORRECT WORD

ENTER THE WORD

Word

Misspelled?

INCOMPLETE WORD

Yes

No

No

No

Yes

WORD PRESENT

**5. SYSTEM IMPLEMENTATION**

**5.1 Tools and Technologies Used**

1. PyCharm provides a dedicated tool for installing, uninstalling, and upgrading Python packages. So doing, if a packaging tool is missing, PyCharm suggests to install it.

PyCharm smartly tracks the status of packages and recognizes outdated versions by showing the number of the currently installed package version (column Version), and the latest available version (column Latest). When a newer version of a package is detected, PyCharm marks it with the arrow sign/help/img/idea/2017.2/arrow_right.png.

**Introduction :**

Tkinter is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) [binding](https://en.wikipedia.org/wiki/Language_binding) to the [Tk](https://en.wikipedia.org/wiki/Tk_(software)" \o "Tk (software)) [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's [defacto standard](https://en.wikipedia.org/wiki/De_facto_standard) GUI. Tkinter is included with the standard [Microsoft Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) and Mac OS X install of Python.

The name Tkinter comes from Tk interface. Tkinter was written by Fredrik Lundh.

As with most other modern Tk bindings, Tkinter is implemented as a Python wrapper around a complete [Tcl](https://en.wikipedia.org/wiki/Tcl" \o "Tcl) interpreter embedded in the Python interpreter. Tkinter calls are translated into Tcl commands which are fed to this embedded interpreter, thus making it possible to mix Python and Tcl in a single application.

Python 2.7 and Python 3.1 incorporate the "themed Tk" ("ttk") functionality of Tk 8.5. This allows Tk widgets to be easily themed to look like the native desktop environment in which the application is running, thereby addressing a long-standing criticism of Tk (and hence of Tkinter).

There are several popular GUI library alternatives available, such as [wxPython](https://en.wikipedia.org/wiki/WxPython" \o "WxPython), [PyQt](https://en.wikipedia.org/wiki/PyQt" \o "PyQt) ([PySide](https://en.wikipedia.org/wiki/PySide" \o "PySide)), [Pygame](https://en.wikipedia.org/wiki/Pygame" \o "Pygame), [Pyglet](https://en.wikipedia.org/wiki/Pyglet" \o "Pyglet), and [PyGTK](https://en.wikipedia.org/wiki/PyGTK" \o "PyGTK).

Tkinter is [free software](https://en.wikipedia.org/wiki/Free_software) released under a [Python license](https://en.wikipedia.org/wiki/Python_license)

**From Tkinter import \*:**

The first line from Tkinter import \* means, to import tkinter module and want to use everything (the \*) defined inside it. The line app\_win = Tk() means to create a top level window. Most people use root instead of parent. Note it is just a name and just choose an identifier . When calling Tk() without giving any parameter means to create a top level window for the application.

**5.2 Installation and deployment**

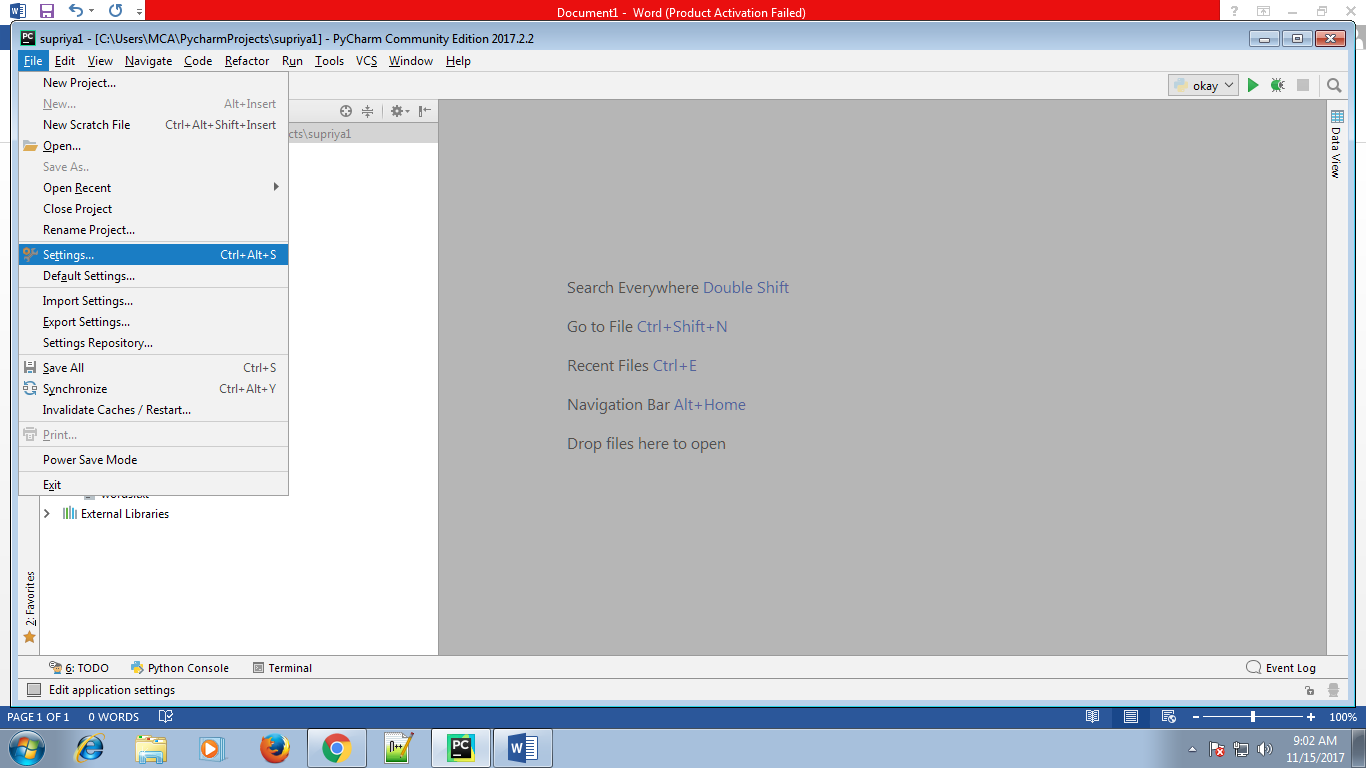
1. PyCharm provides a dedicated tool for installing, uninstalling, and upgrading Python packages. So doing, if a packaging tool is missing, PyCharm suggests to install it.

PyCharm smartly tracks the status of packages and recognizes outdated versions by showing the number of the currently installed package version (column **Version**), and the latest available version (column **Latest**). When a newer version of a package is detected, PyCharm marks it with the arrow sign/help/img/idea/2017.2/arrow_right.png.

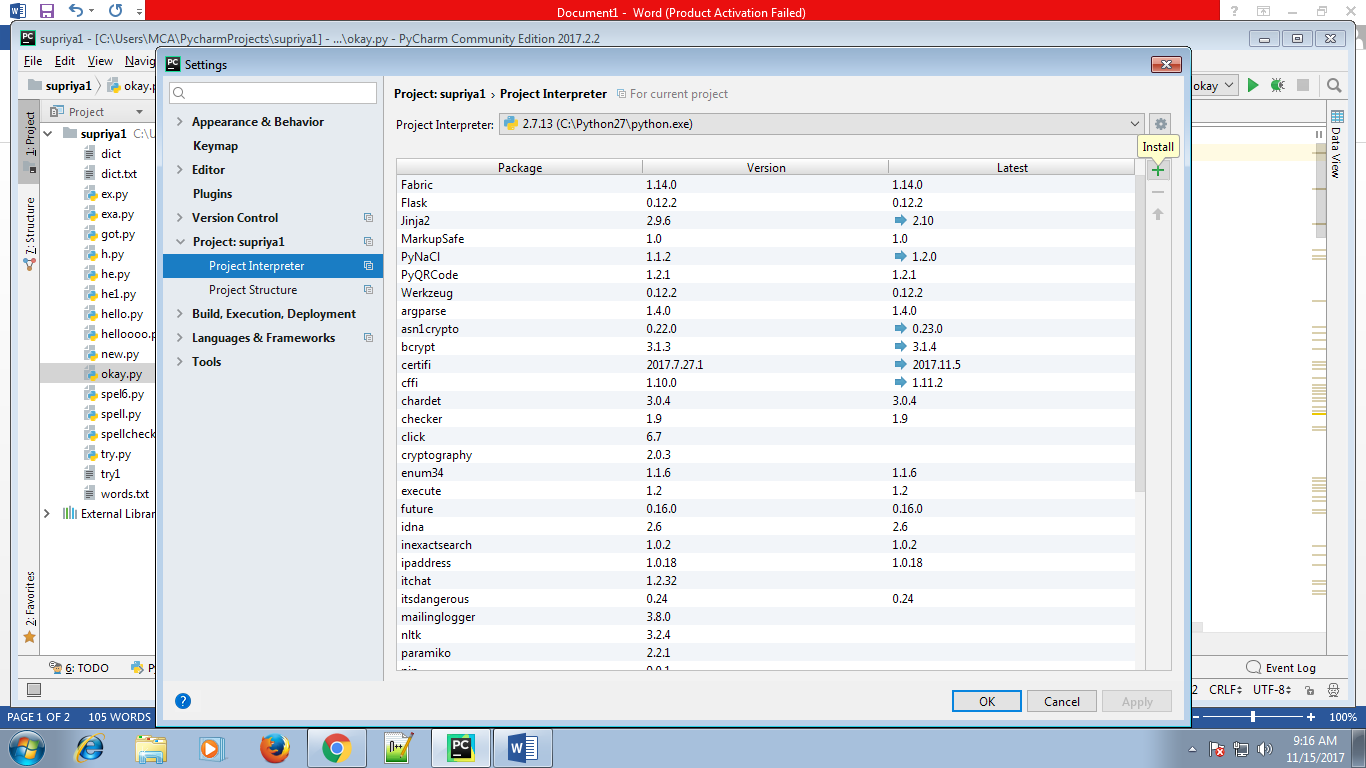
Installing Packages

To install a package Tkinter

1.Go to File and click on Settings



2. In the Project Interpreter page of the project settings, select the desired Python interpreter or virtual environment.



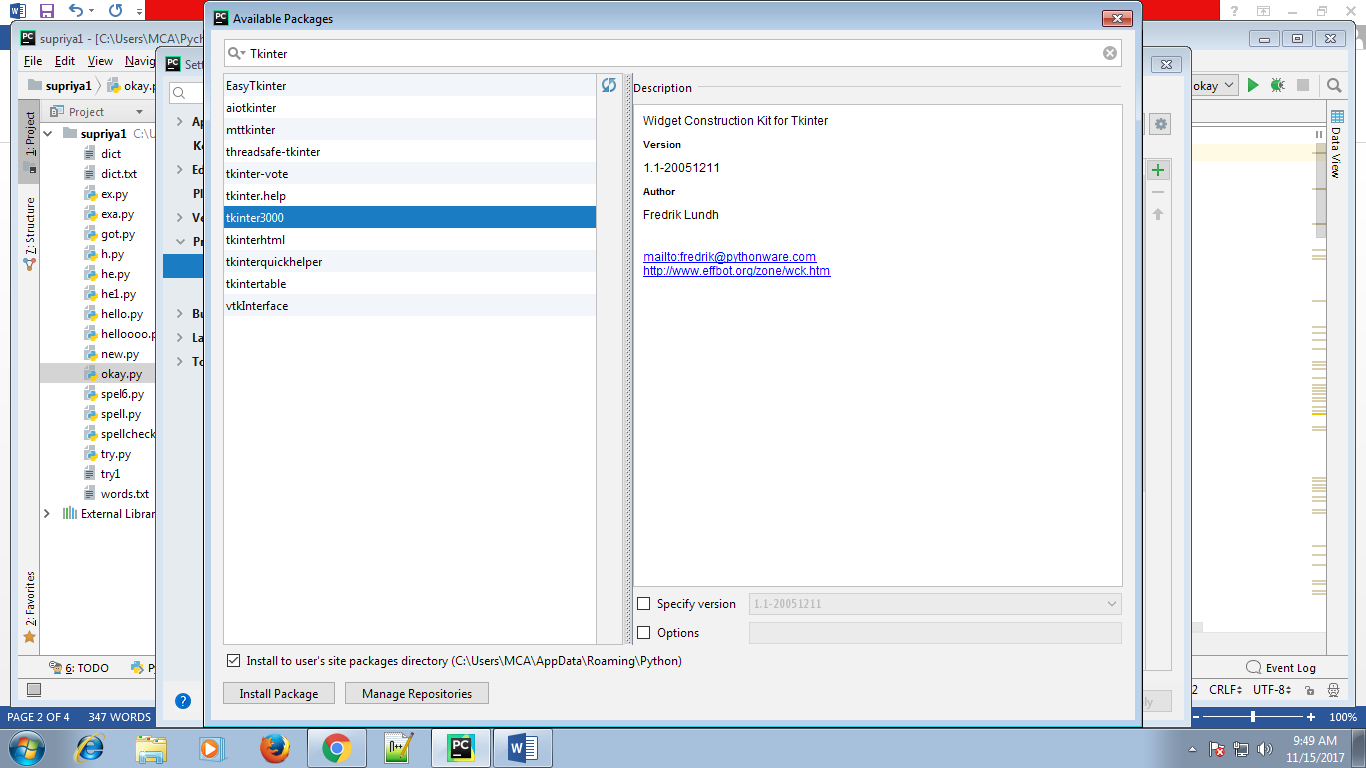
3. Click/help/img/idea/2017.2/add.png.

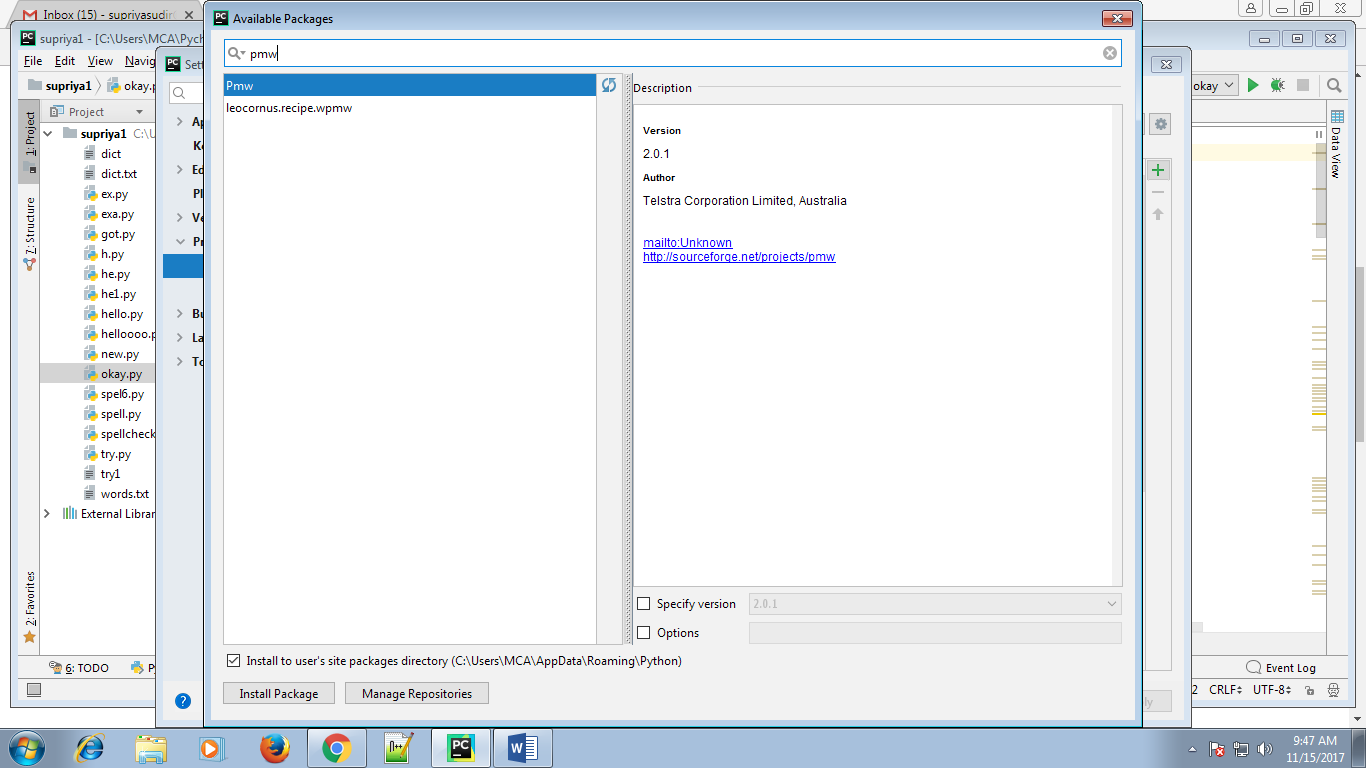
4. In the Available Packages dialog box that opens, select the desired package from the list.

If necessary, use the Search field, Wherever the string can be entered . So doing, the list of packages shrinks to show the matching packages only.

## Tkinter Extensions

* [Pmw](https://wiki.python.org/moin/Pmw) ([http://pmw.sourceforge.net](http://pmw.sourceforge.net/))
* [Tix](https://wiki.python.org/moin/Tix) (<http://www.python.org/doc/current/lib/module-Tix.html>)
* [TkZinc](https://wiki.python.org/moin/TkZinc) ([http://www.tkzinc.org](http://www.tkzinc.org/))
* [Tkinter3000](https://wiki.python.org/moin/Tkinter3000) ([http://tkinter.effbot.org](http://tkinter.effbot.org/))





5. If required, select the following check boxes:

--Specify version: if this check box is selected, select the desired version from the drop-down list of available versions. By default, the latest version is taken.

--Options: If this check box is selected, type the options in the text field.

6. Click Install Package.

Use the various packaging tools, including [devpi](http://doc.devpi.net/latest/) or [PyPi](https://pypi.python.org/pypi).

To specify a custom repository, follow these steps

1. In the [Project Interpreter](https://www.jetbrains.com/help/pycharm/project-interpreter.html) page of the project settings, click/help/img/idea/2017.2/add.png, and then, in the [Available Packages](https://www.jetbrains.com/help/pycharm/available-packages.html) dialog box, click Manage Repositories.
2. In the Manage Repositories dialog box that opens, click /help/img/idea/2017.2/add.png to add a URL of a local repository, for example, something like http://somehost/alice/dev.
3. In the Manage Repositories dialog box, click OK.
4. Back in the [Available Packages](https://www.jetbrains.com/help/pycharm/available-packages.html) dialog box, click /help/img/idea/2017.2/refresh.png to reload the list of packages. As a result, the packages that exist on the local server appear.

**Uninstalling packages**

To uninstall a package

1. On the [Project Interpreter](https://www.jetbrains.com/help/pycharm/project-interpreter.html) page, in the list of packages, select the ones to be deleted.

2. Click/help/img/idea/2017.2/delete.png. The selected packages are removed from disk.

**5.3 Implementation Workflow**

Algorithm Implementation/Strings/Levenshtein distance

In [computational linguistics](https://en.wikipedia.org/wiki/Computational_linguistics) and [computer science](https://en.wikipedia.org/wiki/Computer_science), edit distance is a way of quantifying how dissimilar two [strings](https://en.wikipedia.org/wiki/String_(computing)) (e.g., words) are to one another by counting the minimum number of operations required to transform one string into the other.

Edit distances find applications in [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), where automatic [spelling correction](https://en.wikipedia.org/wiki/Spell_checker) can determine candidate corrections for a misspelled word by selecting words from a dictionary that have a low distance to the word in question.

The Levenshtein distance is a [string metric](https://en.wikipedia.org/wiki/String_metric) for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.

It is named after [Vladimir Levenshtein](https://en.wikipedia.org/wiki/Vladimir_Levenshtein), who considered this distance in 1965.

Levenshtein distance may also be referred to as edit distance, although that term may also denote a larger [family of distance metrics](https://en.wikipedia.org/wiki/Edit_distance).

 It is closely related to [pairwise string alignments](https://en.wikipedia.org/wiki/Sequence_alignment#Pairwise_alignment).

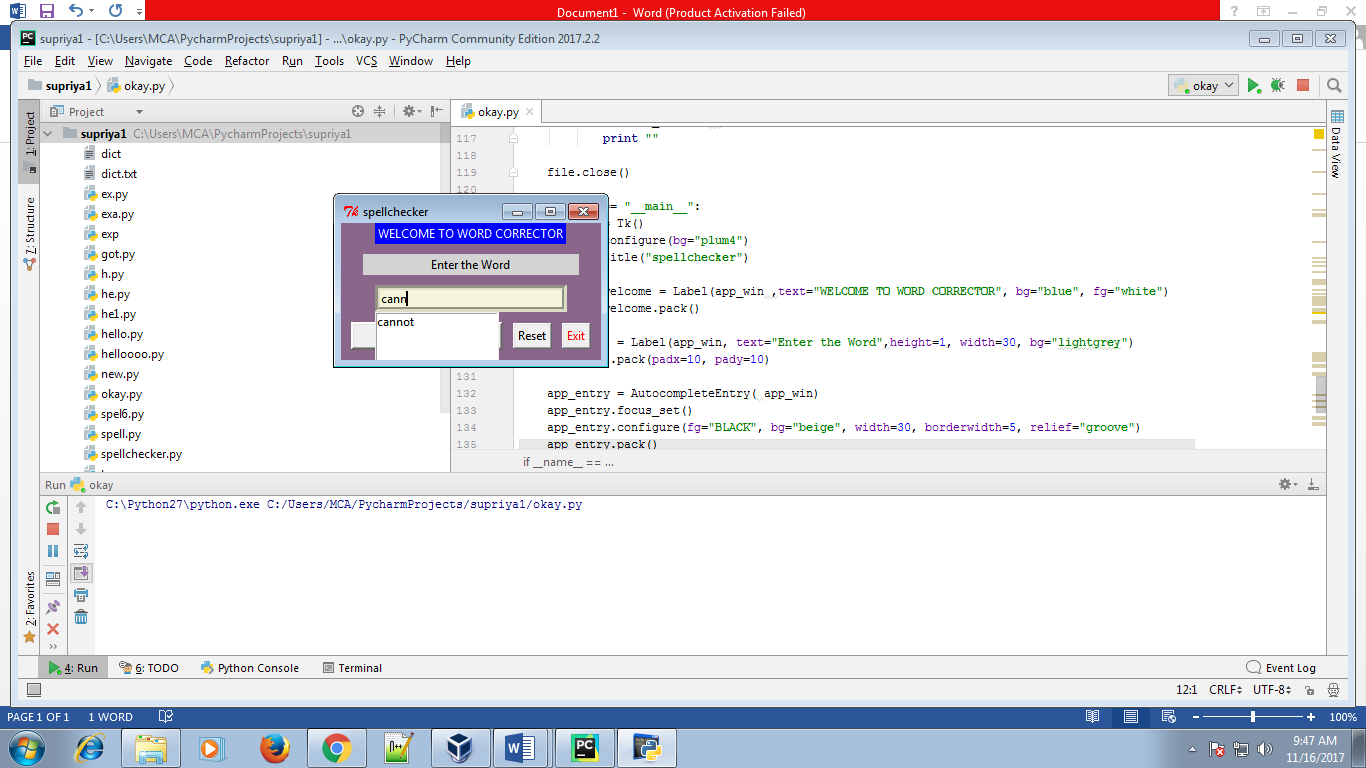
The [Levenshtein distance](https://en.wikipedia.org/wiki/Levenshtein_distance" \o "Levenshtein distance) between "kitten" and "sitting" is 3. A minimal edit script that transforms the former into the latter is:

1. **k**itten → **s**itten (substitution of "s" for "k")
2. sitt**e**n → sitt**i**n (substitution of "i" for "e")
3. sittin → sittin**g** (insertion of "g" at the end).

LCS distance (insertions and deletions only) gives a different distance and minimal edit script:

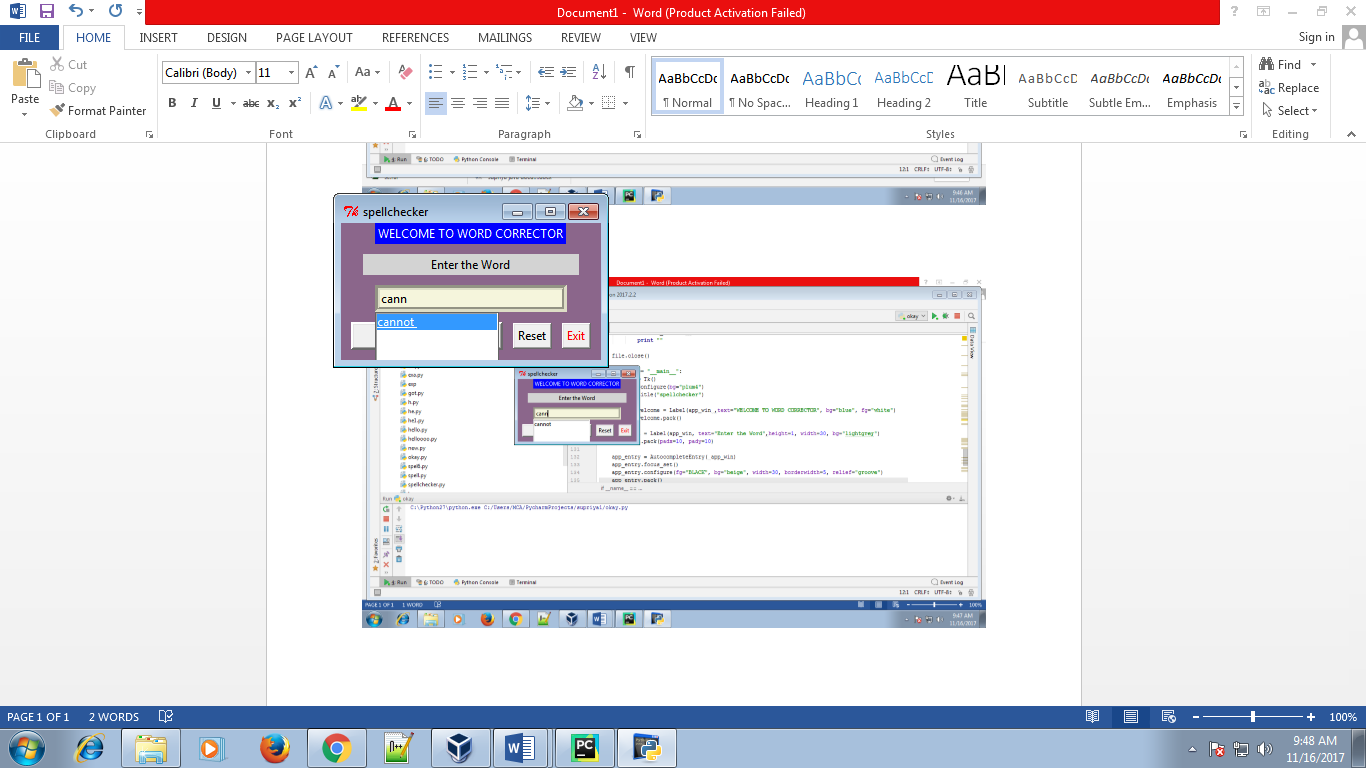
1. delete **k** at 0
2. insert **s** at 0
3. delete **e** at 4
4. insert **i** at 4
5. insert **g** at 6

Step 1:



**Autocomplete entry**

In the text box the Autocomplete entry function Is Called that function lists downs the words what User have typed in the textbox. If the word typed is incomplete, it will list out and suggests the words available based on Levenshtein distance. After finding the wanted word user must double click on that word so that it will automatically set in the label.

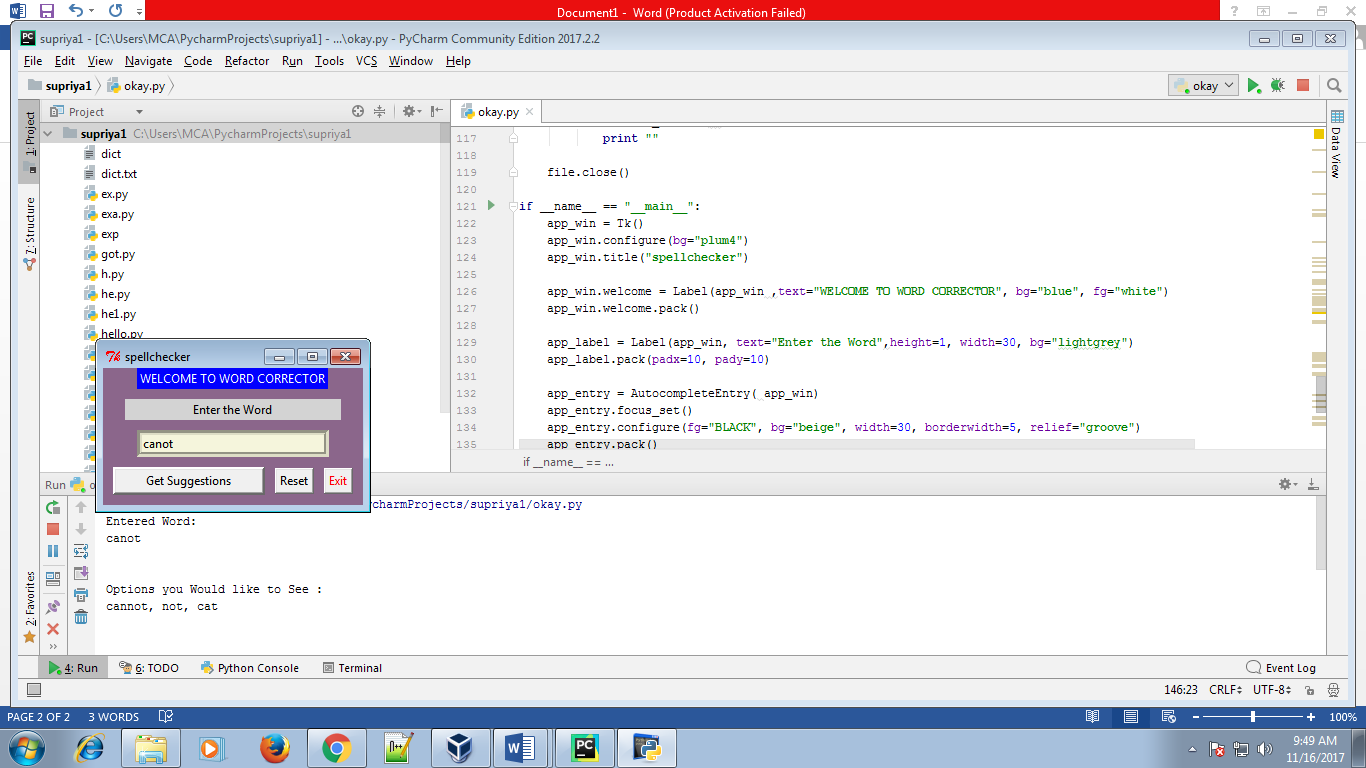


**Step 2:**

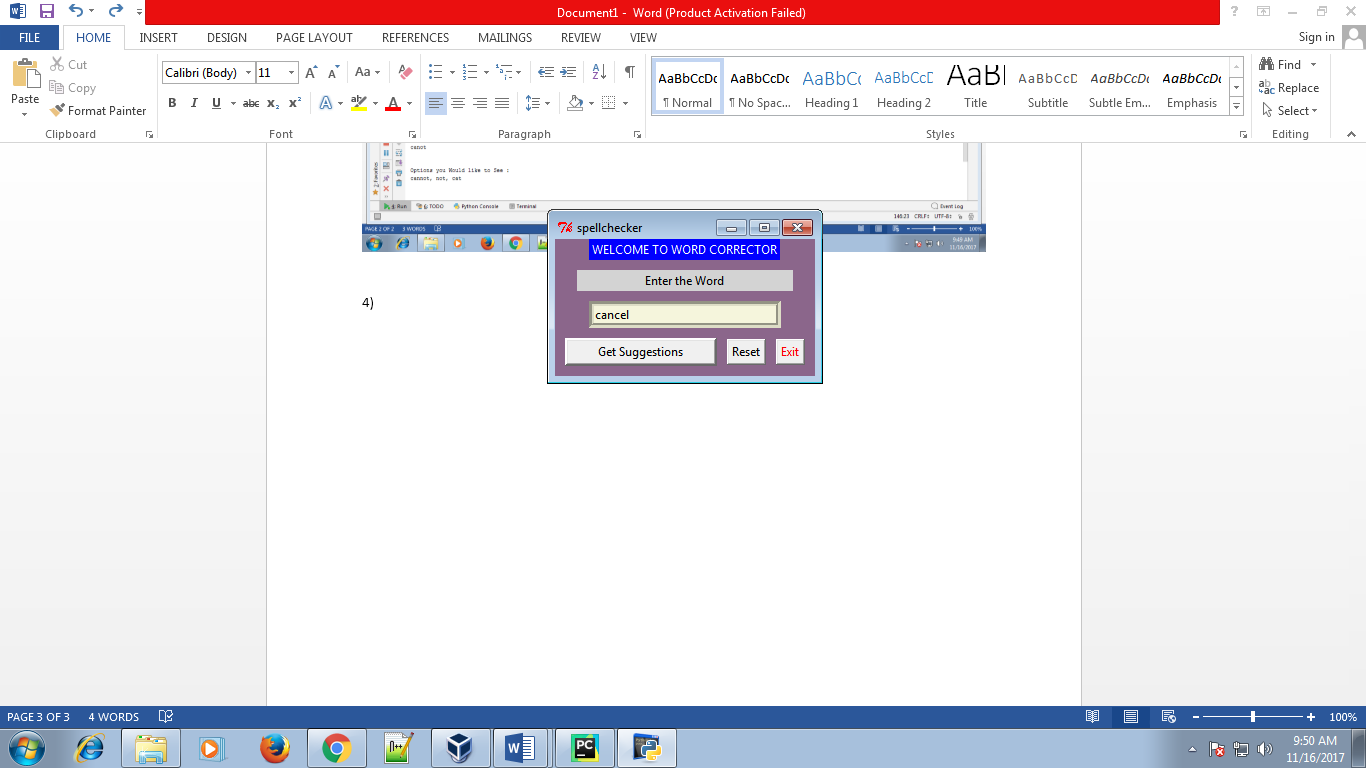
**Get Suggestion**

If user is typing incorrect word, but still wants to know the correct words, can click on get suggestion button, is able to see the lists of similar words.

This helps the user to choose the correct words.



**Reset Button**

Reset button is used to clear all the entered context in the textbox.

**Exit Button**

Exit button is used to terminate the process.

**6. CONCLUSION AND FUTURE SCOPE**

In this project, the features, autosuggest and, word corrector have been implemented for word corrector, keeping in mind its constrained environment. Implemented in python, word corrector includes features like single-words, multi-words, suggest to enhance the performance. Word corrector was implemented for English language, assuming that the frequency of one letter errors is more than multiple-letter errors .These features proved to reduce the typing work and avoid spelling errors in word corrector. In the future, autosuggest and Word corrector features in word corrector can be extended and enhanced in following directions:

.Introducing suggestions for transliterated queries pertaining to languages other than French, Hindi,Marathi etc.

**APPENDIX**

**Source Code**

import Tkinter  
from Tkinter import \*  
class AutocompleteEntry(Entry):  
def \_\_init\_\_(self, \*args, \*\*kwargs):  
 path = "C:\Users\MCA\PycharmProjects\supriya1\dict.txt"  
 file2 = open(path, 'r')  
 line = file2.readline()  
 file1 = []  
 while line:  
 file1.append(line)  
 line = file2.readline()  
 file2.close()  
 Entry.\_\_init\_\_(self, \*args, \*\*kwargs)  
 self.file1 =file1  
 self.var = self["textvariable"]  
 if self.var == '':  
 self.var = self["textvariable"] = StringVar()  
 self.var.trace('w', self.changed)  
 self.bind("<Down>", self.down)  
 self.lb\_up = False  
 def changed(self, name, index, mode):  
 if self.var.get() == '':  
 self.lb.destroy()  
 self.lb\_up = False  
 else:  
 words = self.comparison()  
 if words:  
 if not self.lb\_up:  
 self.lb = Listbox()  
 self.lb.bind("<Double-Button-1>", self.selection)  
 elf.lb.place(x=self.winfo\_x() , y=self.winfo\_y() + self.winfo\_height())  
 self.lb.place= app\_entry  
 self.lb\_up = True  
 self.lb.delete(0, END)  
 for w in words:  
 elf.lb.insert(END, w)  
 else:  
 if self.lb\_up:  
 self.lb.destroy()  
 self.lb\_up = False  
 def selection(self, event):  
 if self.lb\_up:  
 elf.var.set(self.lb.get(ACTIVE))  
 self.lb.destroy()  
 self.lb\_up = False  
 self.icursor(END)  
 def down(self, event):  
 if self.lb.curselection() == ():  
 index = '0'  
 else:  
 index = self.lb.curselection()[0]  
 def comparison(self):  
 pattern = re.compile('.\*' + self.var.get() + '.\*')  
 return [w for w in self.file1 if re.match(pattern, w)]  
 def min\_edit\_dist(word1,word2):  
 len\_1=len(word1)  
 len\_2=len(word2)  
  
 x = [[0]\*(len\_2+1) for \_ in range(len\_1+1)]  
 for i in range(0,len\_1+1):  
 x[i][0]=i  
 for j in range(0,len\_2+1):  
 x[0][j]=j  
 for i in range (1,len\_1+1):  
 for j in range(1,len\_2+1):  
 if word1[i-1]==word2[j-1]:  
 x[i][j] = x[i-1][j-1]  
 else :  
 x[i][j]= min(x[i][j-1],x[i-1][j],x[i-1][j-1])+1  
 return x[i][j]  
 def retrieve\_text():  
 global word1  
 word1=(app\_entry.get())  
 print "Entered Word:"  
 print word1  
 print "\t"  
 from difflib import get\_close\_matches  
 with open('C:\Users\MCA\PycharmProjects\supriya1\dict.txt') as fin:  
 words = set(line.strip().lower() for line in fin)  
 matches = get\_close\_matches(word1, words)  
 if word1 == matches[0]:  
 print 'Above Word you Given is Correct'  
 else:  
 print 'Options you Would like to See :\n', ', '.join(matches)  
 print "\t"  
 path="C:\Users\MCA\PycharmProjects\supriya1\dict.txt"  
 file=open(path,'r')  
 lines=file.readlines()  
 distance\_list=[]  
 *#print "Correct Word is Shown Here :"* for i in range(0,1837):  
 dist=min\_edit\_dist(word1,lines[i])  
 distance\_list.append(dist)  
 for j in range(0,1837):  
 if distance\_list[j]<=2:  
 print ""  
 file.close()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 app\_win = Tk()  
 app\_win.configure(bg="plum4")  
 app\_win.title("spellchecker")  
 app\_win.welcome = Label(app\_win ,text="WELCOME TO WORD CORRECTOR", bg="blue", fg="white")  
 app\_win.welcome.pack()  
 app\_label = Label(app\_win, text="Enter the Word",height=1, width=30, bg="lightgrey")  
 app\_label.pack(padx=10, pady=10)  
 app\_entry = AutocompleteEntry( app\_win)  
 app\_entry.focus\_set()  
 app\_entry.configure(fg="BLACK", bg="beige", width=30, borderwidth=5, relief="groove")  
 app\_entry.pack()  
 app\_button = Button(app\_win, text="Get Suggestions", command=retrieve\_text ,height=1, width=20,relief="raised",borderwidth=3)  
 app\_button.pack(padx=10, pady=10 ,side=LEFT)  
 button = Tkinter.Button(app\_win, text="Exit",fg="red", command=app\_win.destroy)  
 button.pack(side=RIGHT,padx=10, pady=10 )  
 btn = Button(app\_win, text='Reset', command=lambda: app\_entry.delete(0, END))  
 btn.pack(side=RIGHT)  
 app\_win.mainloop()

**Bibliography**

<https://stackoverflow.com/questions/43312657/tkinter-how-to-catch-this-particular-key-event>

<https://www.python-course.eu/python_tkinter.php>

<http://www.tkdocs.com/tutorial/>

<http://www.openbookproject.net/py4fun/gui/tkPhone.html>

https://www.packtpub.com/mapt/book/application\_development/9781849513845/10/ch10lvl1sec102/drop-down-menus-to-select-from-a-list

<https://stackoverflow.com/questions/13928155/spell-checker-for-python>

https://www.oxfordlearnersdictionaries.com/wordlist/english/oxford3000/