

# SAVITRIBAI PHULE UNIVERSITY

## PCCOE,NIGDI



**Department of Computer Engineering**

**Academic Year:2018-2019**

### **Project Report**

**On**

### **RAINFALL ANALYSIS**

**Presented By**

<b>S No.</b>	<b>Roll No.</b>	<b>Name</b>
1.	TECOC335	ISHAN KOAKDWAR
2.	TECOC306	TANAYA BAGADE
3.	TECOC336	ANURAG KULKARNI
4.	TECOC341	KULDEEP MEHTA

**Under The Guidance Of**

**Prof.Ganesh Deshmukh**

**DEPARTMENT OF COMPUTER ENGINEERING,  
PIMPRI CHINCHWAD COLLEGE OF ENGINEERING SECTOR 26, NIGDI,  
PRADHIKARAN**



**PIMPRI CHINCHWAD COLLEGE OF ENGINEERING  
DEPARTMENT OF COMPUTER ENGINEERING**

**CERTIFICATE**

This is to certify that, the project entitled "**Rainfall Analysis**"  
is successfully carried out as a University Project successfully by above  
mentioned students of Pimpri Chinchwad college of Engineering

Akurdi, Pune-44

**Under the guidance of Prof. Ganesh Deshmukh.**

In the partial fulfillment of the requirements for the T.E. (Computer  
Engineering)

**Signature of Project Guide**

## **ABSTRACT**

Rainfall Analysing has now become a major concern for the rapidly changing climatology. Therefore its related parameters which covers the present, past and the future predictions of the rains are well illustrated throughout project. India covers a region of 3.287 million km<sup>2</sup> comprising of 28 states and 7 Union Territories that have a great variance in the pattern, amount, etc are concerned while Rainfall Analysis when represented graphically.

The project "Rainfall Analysis" will give us an idea of what majors should be worked upon like precautions for avoiding drought prone areas by building dams, proper harvesting of rain water, proper management of floods, etc.

Hence the project is developed proficiently to help the Climatology and Meteorology services to predict and taken actions for the upcoming Rainfalls in India.

This system entirely helps and accurately analyses the rainfall over the years and helps in development of new governmental projects etc

RAINFALL ANALYSIS is developed completely in Python Language i.e currently in demand in the market with the GUI Tkinter Programming tools to work accurately and integrate the systems more efficiently.

## INDEX

Sr.No.	TITLE	PAGE NO
<b>1</b>	<b><u>CHAPTER 1-INTRODUCTION</u></b>  <b><u>PYTHON</u></b>  1.1) Python Language Introduction 1.2) History of Python 1.3) Python Features 1.4) Python graphical user interfaces (GUIs)  <b><u>PYTHON TKINTER GUI</u></b>  1.5) Tkinter Widgets 1.6) Geometry Management	1 - 6
<b>2</b>	<b><u>CHAPTER-2 IMPLEMENTATION</u></b>  2.1)Technologies used 2.2)Language used 2.3) <u>MODULES</u>  <b><u>CODE OF PROJECT</u></b>	7-11
<b>3</b>	<b><u>CHAPTER-3 RESULTS</u></b>	12-13
<b>4</b>	<b><u>CHAPTER-4 CONCLUSION &amp; REFERENCES</u></b>	14

## CHAPTER 1-

### INTRODUCTION

#### **1.1) Python Language Introduction**

[Python](#) is a widely used general-purpose, high level programming language. It was initially designed by [Guido van Rossum in 1991](#) and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

**Python is Interpreted** — Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive** — You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** — Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** — Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

#### **1.2) History of Python**

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

### 1.3) Python Features

Python's features include —

- **Easy-to-learn** — Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** — Python is easily understandable and readable.
- **Easy-to-maintain** — Python's source code is fairly easy-to-maintain.
- **A broad standard library** — Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** — Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
- **Portable** — Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** — You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** — Python provides interfaces to all major commercial databases.
- **GUI Programming** — Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** — Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below —

- It supports functional and structured programming methods as well as OOP.

- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

#### 1.4) Python graphical user interfaces (GUIs)

- **Tkinter** – Tkinter is the Python interface to the Tk GUI toolkit shipped with Python. We would look this option in this chapter.
- **wxPython** – This is an open-source Python interface for wxWindows <http://wxpython.org>.
- **JPython** – JPython is a Python port for Java which gives Python scripts seamless access to Java class libraries on the local machine <http://www.jython.org>.

There are many other interfaces available, which you can find them on the net.

#### 1.5) PYTHON TKINTER GUI

Tkinter Programming



Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps –

- Import the *Tkinter* module.
- Create the GUI application main window.
- Add one or more of the above-mentioned widgets to the GUI application.
- Enter the main event loop to take action against each event triggered by the user.

Example

```
#!/usr/bin/python
```

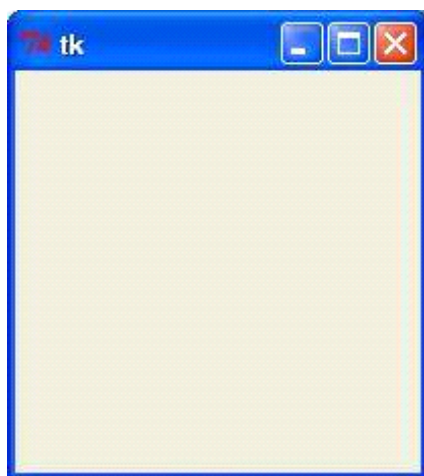
```
import tkinter
```

```
top = tkinter.Tk()
```

```
# Code to add widgets will go here...
```

```
top.mainloop()
```

This would create a following window –



## 1.6) Tkinter Widgets

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

There are currently 15 types of widgets in Tkinter. We present these widgets as well as a brief description in the following table –

Sr.No.	Operator & Description
--------	------------------------



1	<p><b><u>Button</u></b></p> <p>The Button widget is used to display buttons in your application.</p>
2	<p><b><u>Canvas</u></b></p> <p>The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application.</p>
3	<p><b><u>Checkbutton</u></b></p> <p>The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time.</p>
4	<p><b><u>Entry</u></b></p> <p>The Entry widget is used to display a single-line text field for accepting values from a user.</p>
5	<p><b><u>Frame</u></b></p> <p>The Frame widget is used as a container widget to organize other widgets.</p>
6	<p><b><u>Label</u></b></p> <p>The Label widget is used to provide a single-line caption for other widgets. It can also contain images.</p>
7	<p><b><u>Listbox</u></b></p> <p>The Listbox widget is used to provide a list of options to a user.</p>
8	<p><b><u>Menubutton</u></b></p> <p>The Menubutton widget is used to display menus in your application.</p>
9	<p><b><u>Menu</u></b></p> <p>The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton.</p>
10	<p><b><u>Message</u></b></p> <p>The Message widget is used to display multiline text fields for accepting values from a user.</p>
11	<p><b><u>Radiobutton</u></b></p> <p>The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time.</p>

12	<b><u>Scale</u></b> The Scale widget is used to provide a slider widget.
13	<b><u>Scrollbar</u></b> The Scrollbar widget is used to add scrolling capability to various widgets, such as list boxes.
14	<b><u>Text</u></b> The Text widget is used to display text in multiple lines.
15	<b><u>Toplevel</u></b> The Toplevel widget is used to provide a separate window container.
16	<b><u>Spinbox</u></b> The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values.
17	<b><u>PanedWindow</u></b> A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically.
18	<b><u>LabelFrame</u></b> A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts.
19	<b><u>tkMessageBox</u></b> This module is used to display message boxes in your applications.

## Geometry Management

All Tkinter widgets have access to specific geometry management methods, which have the purpose of organizing widgets throughout the parent widget area. Tkinter exposes the following geometry manager classes: pack, grid, and place.

This geometry manager organizes widgets by placing them in a specific position in the parent widget.

## CHAPTER-2

### IMPLEMENTATION

**2.1) Technologies used:-** Python 3.6  
Python Thnikter GUI

**2.2) Language used:-** Python

**Modules:**

**1)Yearwise Analysis:**

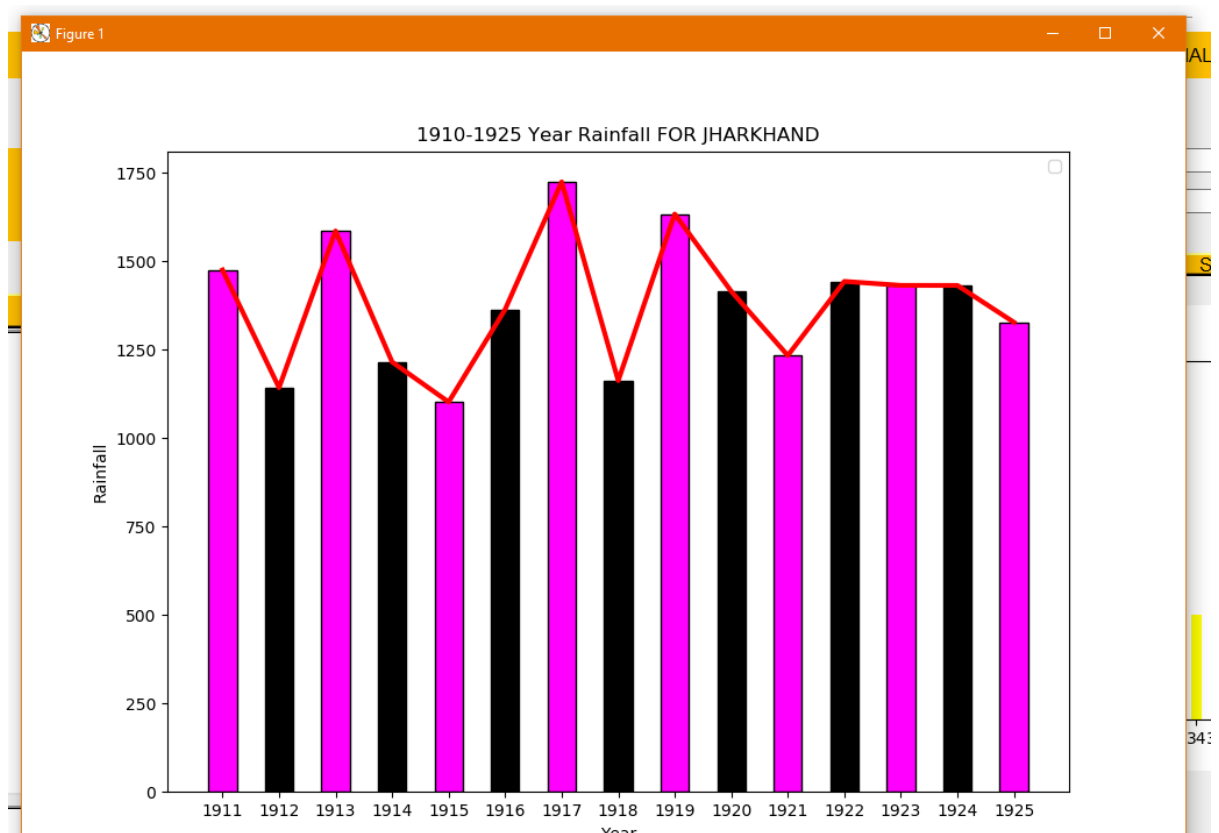
**Input:**

Starting year , End year and Subdivision should be given as the input to this module.

**Processing:**

It processes all the years from starting to end and generates all the rainfall graphs of respective years given as input

**Output:** It generates graphs of all years with details of year wise rainfall as follows



## 2)Monthwise Analysis:

### Input:

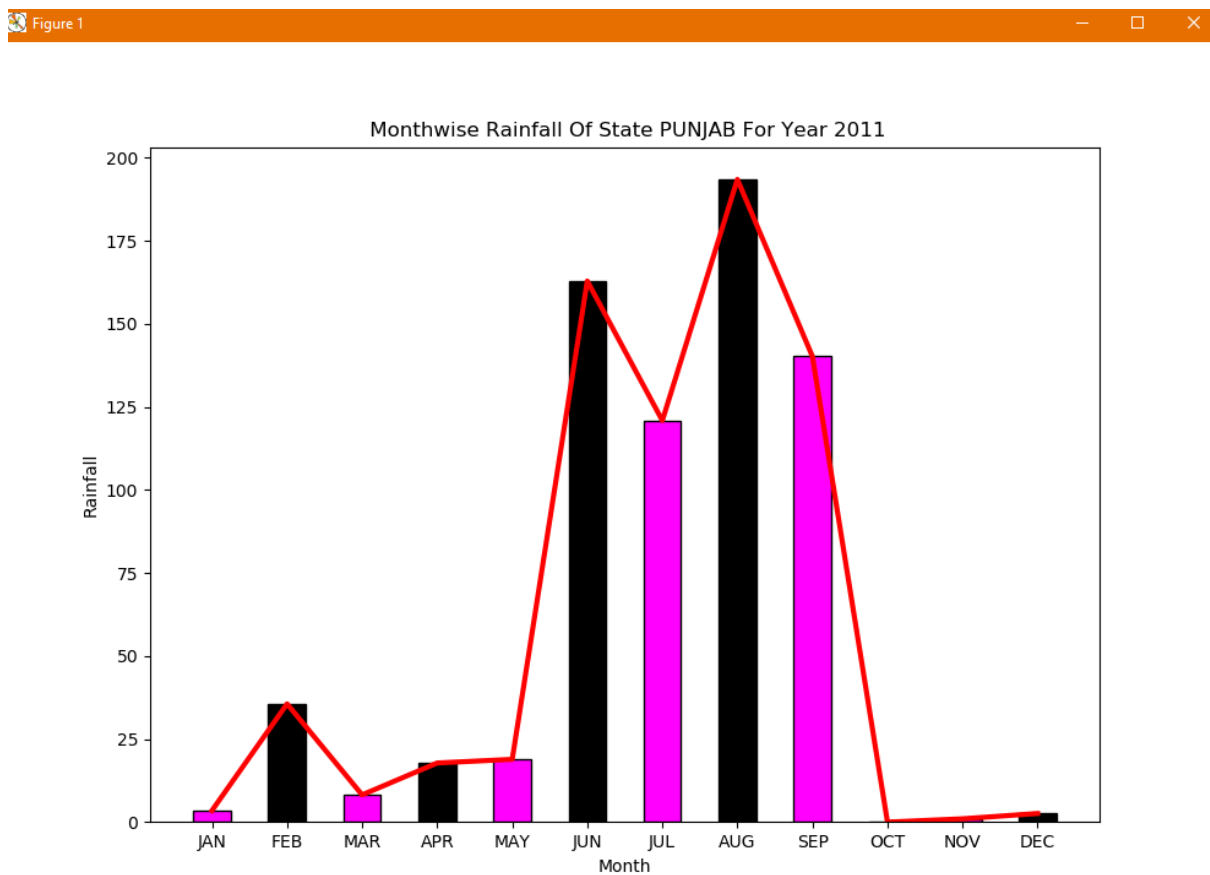
Year and Subdivision should be given as the input to this module.

### Processing:

It processes all Months for given year and Subdivision's rainfall and generates the graphs

### Output:

It generates following graph



### 3)Future Analysis:

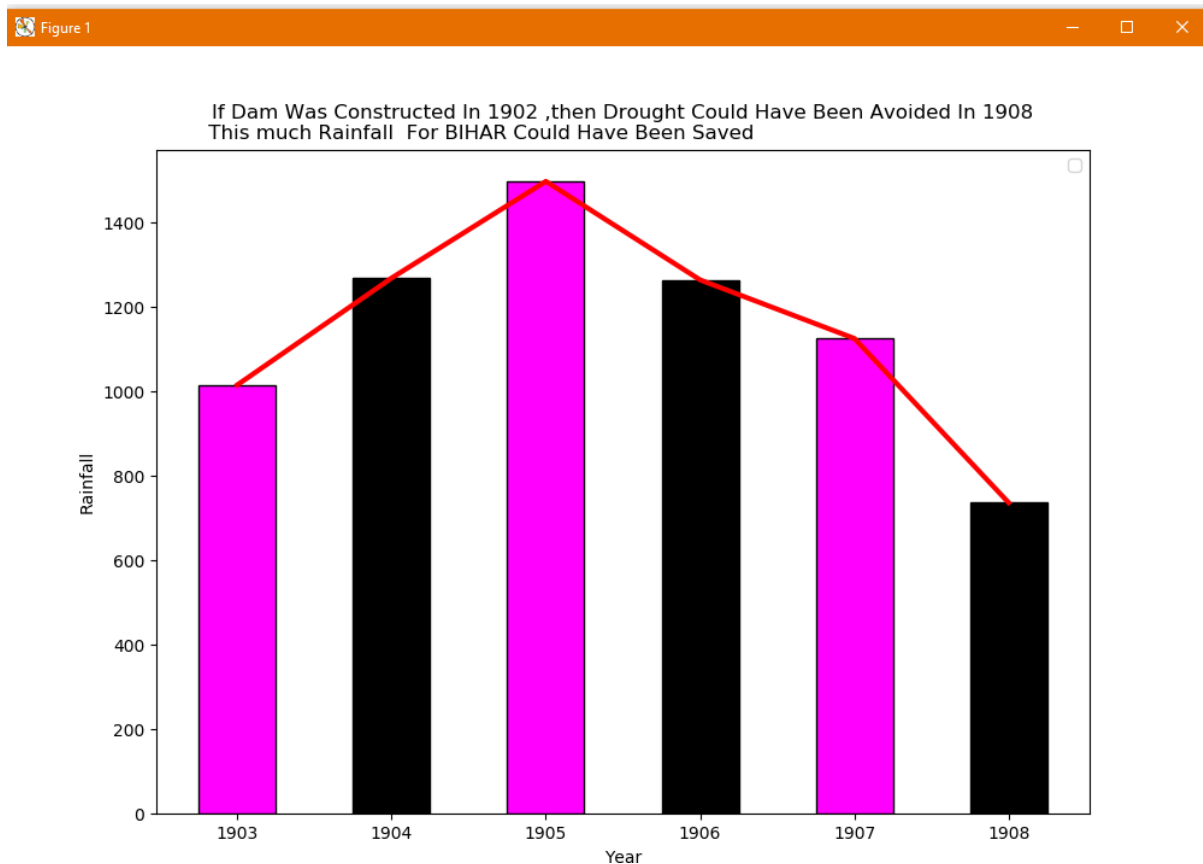
#### Input:

Subdivision in given as input to this module.

#### Processing:

It predicts the years in which there should be dam built so that the Drought could be avoided.

**Output:** It generates graphs of predicted years.



### CODE OF PROJECT:-

```
import pandas as pd
import csv
import numpy as z
import matplotlib.pyplot as plt
#data=pd.read_csv("rainfall1901to2015.csv")
```

```

f=open("rainfall1901to2015.csv",'r')
rowset=csv.DictReader(f)
avg=0.0
rainfall=[]
subdiv=[]
annual=[]
avg1=[]
c=0
for q in rowset:
    if q['SUBDIVISION'] not in subdiv:
        subdiv.append(q['SUBDIVISION'])

f=open("rainfall1901to2015.csv",'r')
rowset=csv.DictReader(f)
for i in subdiv:
    avg=0.0
    f=open("rainfall1901to2015.csv",'r')
    rowset=csv.DictReader(f)
    for w in rowset:

        #print(subdiv[i])
        #print(i)
        if w['SUBDIVISION']==i:
            if w['ANNUAL']=='NA':
                w['ANNUAL']=0
            avg=avg+float(w['ANNUAL'])
            avg1.append(round((avg/115),2))

#plt.plot(YEAR,rainfall,'-')
#plt.show()
x=z.arange(len(avg1))
#plt.subplots()

```

```

c=z.array([(1,0,1),(0,0,0)])
d=z.array([(0,0,0),(0,0,0)])
plt.figure(figsize=(10,10))
plt.bar(x,avg1,align='center',color=c,edgecolor=d,width=0.5)
plt.plot(subdiv,avg1, color='Red', linewidth=3)
plt.xticks(x,tuple(subdiv))
plt.xlabel("Month")
plt.ylabel("Rainfall")
#plt.title(str(year)+" Monthwise Rainfall ")
#legend
##ax.set(title='An Example Axes',
##ylabel='Y-Axis', xlabel='X-Axis')
##ax.legend(loc='best')
#plt.set_position([.5, 1.05])

plt.show()
#print( "\n" ,YEAR, rainfall )

```

## CHAPTER-3

### RESULTS

GUI – Main display window for viewing Rainfall Analysis over the years 1901 to 2015. Here any of the states of India or Union Territories can be given as an input through the drop down boxes provision where yearwise, monthwise and future analysis of the overall rainfall can be viewed.



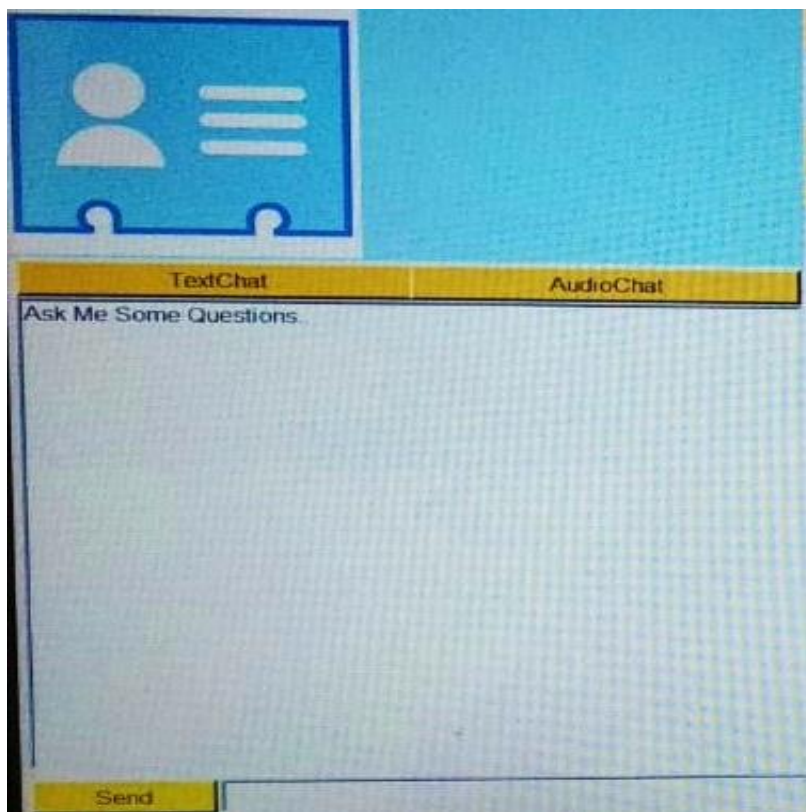
Drop-Down Box:- By DropDown Box we can select various states and Union Territories of India by giving the right state and year as the input. Drop-Down box gives a more clearer and a more organised look in the project which to a very good extent and eases the work of selecting the required parameters.



Comparison				YEAR WISE ANALYSIS		MONTH WISE ANALYSIS	
Start Year	1906	Start Year	1910	Year	2005		
END Year	1908	END Year	1959	STATE	GUJARAT REGION		
STATE	UTTARAKHAN	STATE	JHARKHAND				
STATE 2	MADHYA MAH						
SHOW GRAPH		SHOW GRAPH		SHOW GRAPH			

### User Friendly ChatBot Box:-

This is a completely user friendly based chatterbox in which the question i.e the input can be typed and given and in return the ChatBox replies in the audio form.



## CHAPTER-4

### CONCLUSION & REFERENCES

#### Conclusion:-

The project that we as the group had undertaken "Rainfall Analysis" is successfully implemented by using concepts of Python. While working on this project we have explored and also tried to bring it out in the project the new applications and features of Python language through our skills of programming. Hence an accurate graphical representation with all possible details of the Indian States & Union Territories is covered as a whole through this project.

#### References:-

[www.tutorialspoint.com/python/tk\\_pack.htm](http://www.tutorialspoint.com/python/tk_pack.htm)

[www.datacamp.com](http://www.datacamp.com)

[www.geeksforgeeks.com](http://www.geeksforgeeks.com)

[www.kaggle.com](http://www.kaggle.com)

[www.courseera.com](http://www.courseera.com)