### A Summer Internship Project Report

### on

### “APPLICATION DEVELOPMENT BY PYTHON”

**by**

**TANYA YADAV**

**Under the Guidance of**

**(Mr.Ravi Verma)**

****

****

**ACKNOWLEDGEMENT**

The internship opportunity I had with **IBM** was a great chance for learning and professional development. Therefore, I consider myself as a very lucky individual as I was provided with an opportunity to be a part of it. I am also grateful for having a chance to meet so many wonderful people and professionals who led me though this internship period.

I express my deepest thanks to **Mr. RAVI VERMA** for taking part in useful decision & giving necessary advices and guidance and arranged all facilities to make life easier. I choose this moment to acknowledge his contribution gratefully.

I perceive as this opportunity as a big milestone in my career development. I will strive to use gained skills and knowledge in the best possible way, and I will continue to work on their improvement, in order to attain desired career objectives. Hope to continue cooperation with all of you in the future.

Sincerely,

Tanya Yadav

**SYNOPSIS**

The basic objective of this training is to know the basics of python . In the initial days of our training programme we worked on the basic and fundamental python then we studied about different python libraries.

Then we studied Tkinter and GUI. **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. ... Add one or more of the above-mentioned widgets to the **GUI** application.

INTRODUCTION

**Python** is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language" \o "Interpreted language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [general-purpose](https://en.wikipedia.org/wiki/General-purpose_programming_language) [programming language](https://en.wikipedia.org/wiki/Programming_language). Created by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) and first released in 1991, Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant whitespace](https://en.wikipedia.org/wiki/Off-side_rule). Its language constructs and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help programmers write clear, logical code for small and large-scale projects.

Python's large [standard library](https://en.wikipedia.org/wiki/Standard_library), commonly cited as one of its greatest strengths,provides tools suited to many tasks. It includes modules for creating [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface), connecting to [relational databases](https://en.wikipedia.org/wiki/Relational_database), [generating pseudorandom numbers](https://en.wikipedia.org/wiki/Pseudorandom_number_generator), arithmetic with arbitrary precision decimals,manipulating [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), and [unit testing](https://en.wikipedia.org/wiki/Unit_testing). As of March 2018, the [Python Package Index](https://en.wikipedia.org/wiki/Python_Package_Index) (PyPI), the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

* Graphical user interfaces
* Web frameworks
* Multimedia
* Databases
* Networking
* Test frameworks
* Automation
* Web scraping[[101]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-101)
* Documentation
* System administration
* Scientific computing
* Text processing
* Image processing

**SOFTWARE USED**

**1)PYTHON 3.X:-**

Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

**2)XAMPP:-**

**XAMPP** is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl programming languages.

**LIBRARIES USED**

1. **Tkinter**:- **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. ... Add one or more of the above-mentioned widgets to the **GUI** application.

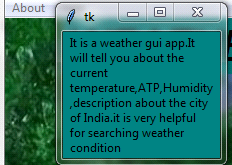
1. **Request:-** The requests module allows to send HTTP requests using Python.
2. **JSON:-** Python has a built-in package called json, which can be used to work with JSON data.
3. **PIL:- Python** Imaging **Library** (abbreviated as **PIL**) (in newer versions known as Pillow) is a free and open-source additional **library** for the **Python** programming language that adds support for opening, manipulating, and saving many different image file formats.
4. **DATETIME:-** date and time are not a data type of its own, but a module named **datetime** can be imported to work with the date as well as time. **Datetime module** comes built into Python, so there is no need to install it externally.

**INTRODUCTION**

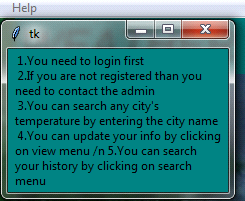
**LOGIN WINDOW**- The user who is already registered can login this window and if not he/she should contact the admin.



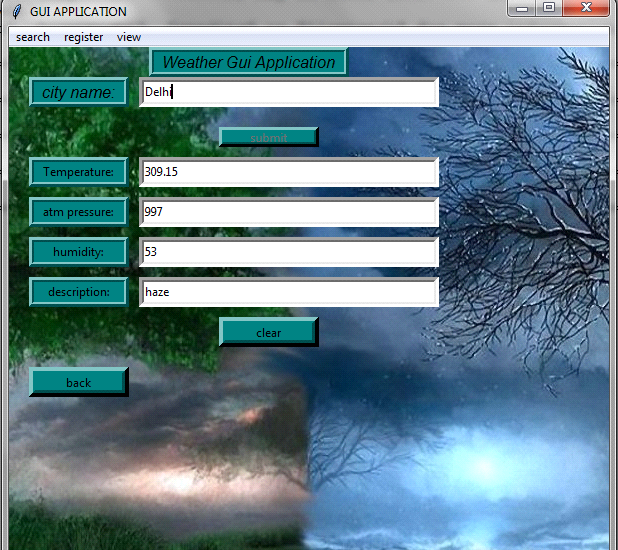
**ABOUT**



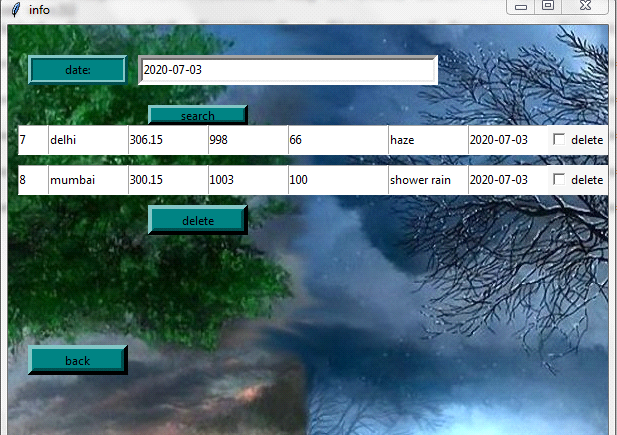
**HELP**



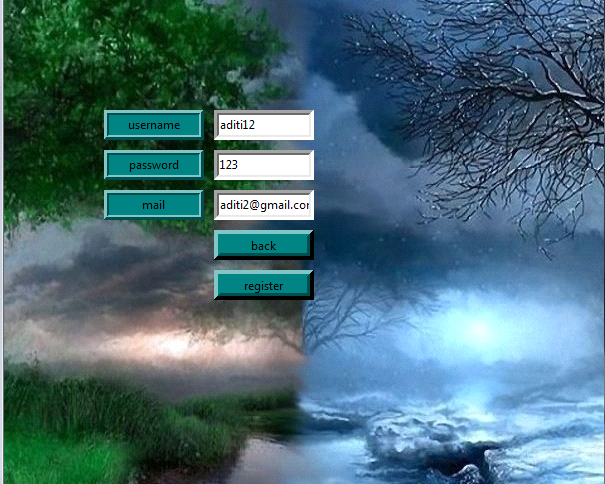
**GUI WEATHER APPLICATION-** This window gives the information of Temperature, atm pressure, humidity, description by entering city name. The clear button clears all the information on the screen so the user can search for other cities also.



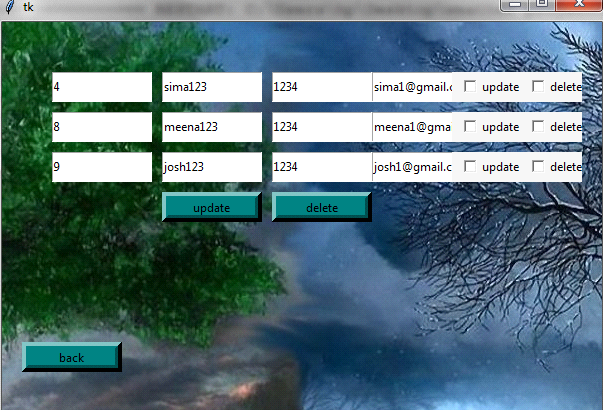
**SEARCH**- User can search the history date wise or can delete history



**REGISTER**- Registration of user is only done by the admin which requires username, password and Email id.

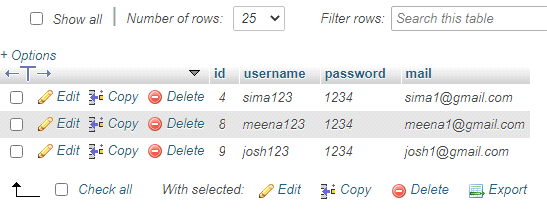


**VIEW**- In this window the admin can view, update or delete users details. Also the user can update their details

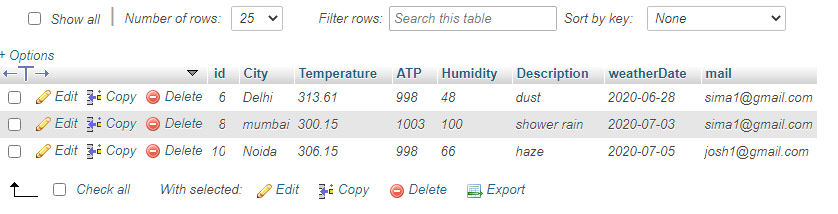


**DATABASE**

**Admin table**



**Info table**



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

#weather GUI application code

def subb():

global tempn

global atpn

global humn

global desn

global cityn

global sub

global cityf

global mycursor

global mydb

global txt8var

apikey="455e95078178ca854b277f7ba5b9ed9b"

base="http://api.openweathermap.org/data/2.5/weather?"

cityname=cityn.get()

sub.config(state=tkinter.DISABLED)

comple=base+"q="+cityname+"&APPID="+apikey

response=requests.get(comple)

x=response.json()

if x["cod"]!="404":

y=x["main"]

t1=y["temp"]

tempn.set(t1)

t2=y["pressure"]

atpn.set(t2)

t3=y["humidity"]

humn.set(t3)

z=x["weather"]

t4=z[0]["description"]

desn.set(t4)

date=datetime.date(datetime.now())

date=str(date)

t1=str(t1)

t2=str(t2)

t3=str(t3)

t4=str(t4)

mycursor.execute("insert into info (City,Temperature,ATP,Humidity,Description,weatherDate,mail) values('"+cityname+"','"+t1+"','"+t2+"','"+t3+"','"+t4+"','"+date+"','"+txt8var.get()+"')")

mydb.commit()

else:

messagebox.showerror("error","city not found")

cityf.delete(0,tkinter.END)

cityf.focus\_set()