

Ajay Kumar Garg Engineering College

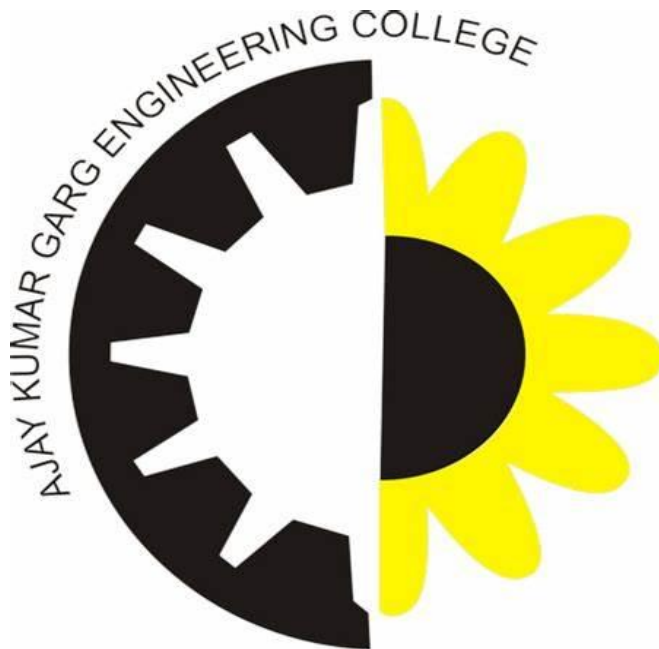
27th KM STONE DELHI-HAPUR BYPASS ROAD, P.O. ADHYATMIK
NAGAR

GHAZIABAD-201009

An Internship Training Report

on

CODING WITH PYTHON



Submitted by: **Shivam Kumar Rajora**

Year: **2nd**

Semester: **3rd**

Section: **CS 2**

Branch: **Computer Science**

Roll No.: **2000270120101**

CERTIFICATE OF TRAINING



Date: 22/10/2021

Ref. No.: SMS12W2021-246

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr. Shivam Kumar Rajora, student of B.Tech CSE, Ajay Kumar Garg Engineering College, Ghaziabad has successfully completed his 6 Weeks Internship started from 01-09-2021 to 17-10-2021 on Covid api Project in Python from ShapeMySkills, Noida. We wish him all the best for his future endeavours.

Regards

prachi 

Prachi Trivedi
Hr Manager
ShapeMySkills, Noida

OUR PARTNERS



Ajay Kumar Garg Engineering College

27th KM STONE DELHI-HAPUR BYPASS ROAD, P.O. ADHYATMIK NAGAR
GHAZIABAD-201009

INTERNSHIP CERTIFICATE

This is to certify that **SHIVAM KUMAR RAJORA** student of **AJAY KUMAR GARG ENGINEERING COLLEGE** B. Tech Second Year **Computer Science** branch, has undergone 6 weeks Internship Training on Covid API project in Python from 1 September 2021 to 17 October 2021.

Ms. Shiva Tyagi
Faculty Internship Assessment Lab

Dr. Sunita Yadav
HoD, CSE

ABSTRACT

Python program to create a GetVaccine Application using Python Programming concepts and Tkinter GUI to develop application.

This is a project report on “GET VACCINE” using Co-Win API. It is a service that provides availabilities of vaccine data, including hospitals name, address, and fee type of vaccine data to the developer of web services and mobile application. It provides an API with JSON, XML and HTML formats.

To use this current availability of vaccine data API, one must need the API key, we use two modules in the program requests and Json.

Application Programming Interface (API):

It is a set of protocols, routines, functions and/or commands that programmers use to develop software or fascilitate interaction between distinct systems.APIs are available for both desktop and mobile use, and are typically useful for programming GUI (graphic user interface) components, as well as allowing a software program to request and accommodate services from another program.

Python GUI Tkinter:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python Interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

TABLE OF CONTENTS

Introduction to organization.....	8-10
Scope of project.....	11
Python.....	12-16
REST Architecture.....	17-18
Status code.....	19
API Endpoints.....	20-25
Source Code.....	26-29
Output.....	30-32
Requirements And Specifications.....	33
Conclusion.....	34
References.....	35

INTRODUCTION TO ORGANIZATION

ShapeMySkills Pvt. Ltd. is one of the Best Training Institute in Noida. In the country, Our Company is the top rates online learning platform and Job-oriented IT Training institute in Delhi NCR. We offer online and offline courses and skill development programs to UG engineering students, working professionals, and college pass-outs. The courses provided by us will help you empower your skills and help various students in career growth.

We have a team of IT who work in several industries and gain enough world exposure for offering courses to students and working professionals as per the latest industry trends. ShapeMySkills Pvt Ltd, the best IT Training institute in Delhi NCR work on 3 Domains:

- Training
- Development
- Placement

In Numerous leading technologies such as Python, Machine Learning, Deep Learning, Digital Marketing, PLC/SCADA, VLSI, PHP & many more, the company offers specialized training programs. **ShapeMySkills Pvt Ltd** has a committed team that contains professional trainers who continuously guide and monitor the students. Our trained students get exclusive personalized attention that helps them for developing solid industry knowledge.

Purpose

Our main purpose is to take you to your “DREAM JOB”. The objective of Training is to develop useful knowledge, techniques, and skills. We prepare the students to carry out the tasks in well-defined job circumstances. Training is the activity that enhances your performance in the present or upcoming job. In our company, you may learn new skills, boost your learning, and get a lot of confidence.

How ShapeMySkills is Helpful?

ShapeMySkills Pvt Ltd is the leader in the specialized training of India and it is the certified training institute in Noida. Our company is known for its best quality of training and we are the largest training service provider in many engineering domains for all Engg students and working professionals.

Our company will prove very helpful to you as we provide all facilities to the students and also having good infrastructure. We focus on the company's working environment and make sure that students do not face any issues related to their course. Our company offers online and offline classes to the students as per their suitability. ShapeMySkills Pvt Ltd is committed to providing quality education and knowledge to the students. We focus on both practical and theoretical knowledge to the students and also give both basic and advanced level knowledge. Our syllabus related to the relevant course is always updated as per the industry standard. So, don't miss the chance to join ShapeMySkills Pvt Ltd and attend the demo session first, and then make your decision. We will make sure that you will never get disappointed.

ShapeMySkills Pvt Ltd Focus On

- Achieving plenty of exposure
- Transform the learning and development standard
- Addressing on-the-job challenges
- Offering tailored training solutions
- E-Learning concept
- Face-to-face learning experiences

SCOPE OF THE PROJECT

Every day a huge volume of data is produced that it is complex or nearly impossible to process that data using traditional methods. This data is handled and processed using emerging technologies such as Big Data Analysis, Machine Learning, AI and NLP. Python's high speed for data processing makes it optimal for usage and python libraries are beneficial in making the implementation and processing of these technologies easy. The reusability of these libraries saves a lot of time when a particular information is needed from a pile of data.

This proves the scope of the project, as it makes easier for computer to take decisions and requires lesser fetching data time from a large amount of data available using python libraries and APIs.

PYTHON

Python is an interpreted high-level general-purpose programming language. Its design philosophy emphasizes code readability with its use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Guido van Rossum began working on Python in the late 1980s, as a successor to the ABC programming language, and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as list comprehensions and a cycle-detecting garbage collection system (in addition to reference counting). Python 3.0 was released in 2008 and was a major revision of the language that is not completely backward-compatible. Python 2 was discontinued with version 2.7.18 in 2020.

Python consistently ranks as one of the most popular programming languages. Python uses dynamic typing and a combination of reference counting and a cycle-detecting garbage collector for memory management.[63] It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has filter, map and reduce functions; list comprehensions,

dictionaries, sets, and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:[66]

- Beautiful is better than ugly.
- Explicit is better than implicit.
- Simple is better than complex.
- Complex is better than complicated.
- Readability counts.

Syntax and Semantics

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons after statements are allowed but are rarely, if ever, used. It has fewer syntactic exceptions and special cases than C or Pascal.

Indentation

Python uses whitespace indentation, rather than curly brackets or keywords, to delimit blocks. An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure. This feature is sometimes termed the off-side rule, which some

other languages share, but in most languages, indentation does not have any semantic meaning. The recommended indent size is four spaces.

Libraries in Python

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary-precision decimals, manipulating regular expressions, and unit testing.

As of September 2021, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 329,000[118] packages with a wide range of functionality, including:

- Automation
- Data analytics
- Databases
- Documentation
- Graphical user interfaces
- Image processing
- Machine learning
- Mobile apps
- Multimedia
- Computer networking
- Scientific computing
- System administration
- Test frameworks

- Text processing
- Web frameworks
- Web scraping

Development Environments

Most Python implementations (including CPython) include a read–eval–print loop (REPL), permitting them to function as a command line interpreter for which the user enters statements sequentially and receives results immediately.

Python comes with an integrated development environment (IDE) called IDLE, which is more beginner based.

Other shells, including IDLE and IPython, add further abilities such as improved auto-completion, session state retention and syntax highlighting.

As well as standard desktop integrated development environments, there are Web browser-based IDEs; SageMath (intended for developing science and math-related Python programs); PythonAnywhere, a browser-based IDE and hosting environment; and Canopy IDE, a commercial Python IDE emphasizing scientific computing.

There's an amazing amount of data available on the Web. Many web services, like YouTube and GitHub, make their data accessible to third-party applications through an application programming interface (API). One of the most popular ways to build APIs is the REST architecture style. Python provides some great tools not only to get data from REST APIs but also to build your own Python REST APIs.

In this tutorial, you'll learn:

- What REST architecture is
- How REST APIs provide access to web data
- How to consume data from REST APIs using the requests library.
- What steps to take to build a REST API
- What some popular Python tools are for building REST APIs

By using Python and REST APIs, you can retrieve, parse, update, and manipulate the data

provided by any web service you're interested in.

REST ARCHITECTURE

REST stands for representational state transfer and is a software architecture style that defines a pattern for client and server communications over a network. REST provides a set of constraints for software architecture to promote performance, scalability, simplicity, and reliability in the system.

REST defines the following architectural constraints:

Stateless: The server won't maintain any state between requests from the client. • Client-server: The client and server must be decoupled from each other, allowing

each to develop independently. • Cacheable: The data retrieved from the server should be cacheable either by the client.

or by the server. • Uniform interface: The server will provide a uniform interface for accessing resources without defining their representation.

• Layered system: The client may access the resources on the server indirectly through other layers such as a proxy or load balancer. • Code on demand (optional): The server may transfer code to the client that it can run,

such as JavaScript for a single-page application. Note, REST is not a specification but a set of guidelines on how to architect a networks connected software system.

HTTP Methods

REST APIs listen for HTTP methods like GET, POST, and DELETE to know which operations to perform on the web service's resources. A resource is any data available in the web service that can be accessed and manipulated with HTTP requests to the REST API. The HTTP method tells the API which action to perform on the resource.

While there are many HTTP methods, the five methods listed below are the most commonly used with REST APIs:

HTTP method	Description
• GET	Retrieve an existing resource.
• POST	Create a new resource.
• PUT	Update an existing resource.
• PATCH	Partially update an existing resource.
• DELETE	Delete a resource.

A REST API client application can use these five HTTP methods to manage the state of resources in the web service.

STATUS CODES

Once a REST API receives and processes an HTTP request, it will return an HTTP response. Included in this response is an HTTP status code. This code provides information about the results of the request. An application sending requests to the API can check the status code and perform actions based on the result. These actions could include handling errors or displaying a success message to a user.

Below is a list of the most common status codes returned by REST APIs:

Code	Meaning	Description
• 200	OK	The requested action was successful.
• 201	Created	A new resource was created.
• 202	Accepted	The request was received, but no modification has been made yet.
• 204	No Content	The request was successful, but the response has no content.
• 400	Bad Request	The request was malformed.
• 401	Unauthorized	The client is not authorized to perform the requested action.
• 404	Not Found	The requested resource was not found.
• 415	Unsupported Media Type	The request data format is not supported by the server.
• 422	Unprocessable Entity	The request data was properly formatted but contained invalid or missing data
• 500	Internal Server Error	The server threw an error when processing the request.

These ten status codes represent only a small subset of the available HTTP status codes.

API ENDPOINTS

A REST API exposes a set of public URLs that client applications use to access the resources of a web service. These URLs, in the context of an API, are called endpoints.

To help clarify this, take a look at the table below. In this table, you'll see API endpoints for a hypothetical CRM system. These endpoints are for a customer resource that represents potential customers in the system:

HTTP method	API endpoint	Description
GET	/customers	Get a list of customers.
GET	/customers/<customer_id>	Get a single customer.
POST	/customers	Create a new customer.
PUT	/customers/<customer_id>	Update a customer.
PATCH	/customers/<customer_id>	Partially update a customer.
DELETE	/customers/<customer_id>	Delete a customer.

Each of the endpoints above performs a different action based on the HTTP method.

REST and Python: Consuming APIs

To write code that interacts with REST APIs, most Python developers turn to requests to send HTTP requests. This library abstracts away the complexities of making HTTP requests. It's one of the few projects worth treating as if it's part of the standard library.

To start using requests, you need to install it first. You can use pip to install it:

```
$ python -m pip install requests
```

Now that you've got requests installed, you can start sending HTTP requests.

GET

GET is one of the most common HTTP methods you'll use when working with REST APIs. This method allows you to retrieve resources from a given API. GET is a read-only operation, so you shouldn't use it to modify an existing resource.

To test out GET and the other methods in this section, you'll use a service called JSONPlaceholder. This free service provides fake API endpoints that send back responses that requests can process.

To try this out, start up the Python REPL and run the following commands to send a GET request to a JSONPlaceholder endpoint:

```
>>> import requests  
>>> api_url = "https://jsonplaceholder.typicode.com/todos/1"  
>>> response = requests.get(api_url)  
>>> response.json()  
{'userId': 1, 'id': 1, 'title': 'delectus aut autem', 'completed': False}
```

This code calls `requests.get()` to send a GET request to `/todos/1`, which responds with the todo item with the ID 1. Then you can call `.json()` on the response object to view the data that came back from the API.

The response data is formatted as JSON, a key-value store similar to a Python dictionary. It's a very popular data format and the de facto interchange format for most REST APIs.

POST

Now, take a look at how you use requests to POST data to a REST API to create a new resource. You'll use JSONPlaceholder again, but this time you'll include JSON data in the request.

This JSON contains information for a new todo item. Back in the Python REPL, run the following code to create the new todo:

```
>>> import requests
>>> api_url = "https://jsonplaceholder.typicode.com/todos"
>>> todo = {"userId": 1, "title": "Buy milk", "completed": False}
>>> response = requests.post(api_url, json=todo)
>>> response.json()
{'userId': 1, 'title': 'Buy milk', 'completed': False, 'id': 201}
>>> response.status_code
201
```

Here, you call `requests.post()` to create a new todo in the system.

PUT

Beyond GET and POST, requests provides support for all the other HTTP methods you would use with a REST API. The following code sends a PUT request to update an existing todo with new data. Any data sent with a PUT request will completely replace the existing values of the todo.

You'll use the same JSONPlaceholder endpoint you used with GET and POST, but this time you'll append 10 to the end of the URL. This tells the REST API which todo you'd like to update:

```
>>> import requests
>>> api_url = "https://jsonplaceholder.typicode.com/todos/10"
>>> response = requests.get(api_url)
>>> response.json()
{'userId': 1, 'id': 10, 'title': 'illo est ... aut', 'completed': True}
>>> todo = {"userId": 1, "title": "Wash car", "completed": True}
>>> response = requests.put(api_url, json=todo)
>>> response.json()
{'userId': 1, 'title': 'Wash car', 'completed': True, 'id': 10}
>>> response.status_code
200
```

Here, you first call `requests.get()` to view the contents of the existing todo. Next, you call `requests.put()` with new JSON data to replace the existing todo's values. You can see the new values when you call `response.json()`. Successful PUT requests will always return 200 instead of 201 because you aren't creating a new resource but just updating an existing one.

PATCH

Next up, you'll use `requests.patch()` to modify the value of a specific field on an existing todo. PATCH differs from PUT in that it doesn't completely replace the existing resource. It only modifies the values set in the JSON sent with the request.

You'll use the same todo from the last example to try out `requests.patch()`. Here are the current values:

```
{'userId': 1, 'title': 'Wash car', 'completed': True, 'id': 10}
```

Now you can update the title with a new value:

```
>>> import requests  
>>> api_url = "https://jsonplaceholder.typicode.com/todos/10"  
>>> todo = {"title": "Mow lawn"}  
>>> response = requests.patch(api_url, json=todo)  
>>> response.json()  
{'userId': 1, 'id': 10, 'title': 'Mow lawn', 'completed': True}  
>>> response.status_code  
200
```

When you call `response.json()`, you can see that title was updated to Mow lawn.

DELETE

Last but not least, if you want to completely remove a resource, then you use DELETE. Here's the code to remove a todo:


```
>>> import requests
>>> api_url = "https://jsonplaceholder.typicode.com/todos/10"
>>> response = requests.delete(api_url)
>>> response.json()

{}
>>> response.status_code
200
```

You call `requests.delete()` with an API URL that contains the ID for the todo you would like to remove. This sends a DELETE request to the REST API, which then removes the matching resource. After deleting the resource, the API sends back an empty JSON object indicating that the resource has been deleted.

Main.py

```
import requests
import json
import push
import otp

url = "https://cdn-api.co-vin.in/api/v2/admin/location/states"

res = requests.get(url = url)
email= input('Enter your email-id:')

get_otp = otp.generateOTP()
push.push_notification(email=email,title="OTP",msg=str(get_otp))
put_otp = input('Enter the otp:')

if put_otp == get_otp:
    data = res.json() #returns dict
    print(data)
    state_name = input('Enter the state name:')
    for i in data["states"]:
        if state_name == i["state_name"]:
            val = i["state_id"]

    state_id = str(val)
```

```
url2 = "https://cdn-api.co-vin.in/api/v2/admin/location/districts/" +  
state_id
```

```
res2 = requests.get(url = url2)
```

```
data2 = res2.json() #returns dict  
print(data2)
```

```
district_name = input("Enter the district name:")
```

```
for i in data2['districts']:
```

```
    if district_name == i['district_name']:
```

```
        val2 = i['district_id']
```

```
district_id = str(val2)
```

```
d = input("Enter the date:")
```

```
date = "&date="+str(d)
```

```
url3 = "https://cdn-api.co-vin.in/" \  
    "api/v2/appointment/sessions/public/findByDistrict?" \  
    "district_id=" + district_id + date
```

```
res3 = requests.get(url = url3)
```

```
data3 = res3.json()
```

```

for i in centers :
    if i["available_capacity"] > 0 and i["vaccine"] ==
"COVISHIELD":

push.push_notification(email=email,title="slots",msg=i["name"]+
i["address"] + str(i["available_capacity"]))

    print(i)

else:

    print("OTP is wrong.")

```

Otp.py

```

# import library

import math, random


# function to generate OTP

def generateOTP():

    # Declare a digits variable
    # which stores all digits

    digits = "0123456789"

    OTP = ""

    # length of password can be changed
    # by changing value in range

    for i in range(4):

```

```
OTP += digits[math.floor(random.random() * 10)]
```

```
return OTP
```

```
# Driver code
```

```
if __name__ == "__main__":
```

```
    print('OTP of 4 digits:', generateOTP())
```

Push.py

```
server = "https://api.pushbullet.com/v2/pushes"
```

```
import requests
```

```
import json
```

```
def push_notification(title,msg,email):
```

```
    headers = {"Content-Type" : "application/json",
```

```
               "Access-Token" :
```

```
"o.YXx1TOEID780wLclEcLs2ktAAmfZqCaO"}
    data = {"type" : "note",
```

```
           "title" : title,
```

```
           "body" : msg,
```

```
           "email" : email}
```

```
    json_data = json.dumps(data)
```

```
    res = requests.post(url=server, data=json_data, headers=headers)
```

```
    print(res.content)
```

OUTPUT:

The image shows a PyCharm IDE interface. At the top, there are tabs for 'otp.py' and 'push.py'. The 'push.py' tab is active, showing a Python script. The script starts with a function call 'push_notification()' and then prompts the user for an email ID, an OTP, a state name, and a date. It then sends notifications to various locations in Uttar Pradesh, including Indrapuri, Davlatpura, and Khairati Nagar. The Run console at the bottom shows the execution of the script, with the output of the notifications being displayed. The Run console also shows the file structure of the project, including 'Structure' and 'Favorites' tabs. The Run console has a 'Run' button and a 'Terminal' button. The Run console also shows the 'Python Packages' and 'Python Console' tabs.

```
main.py
otp.py
push.py
push_notification()

Run: main x

{
  "center_id": 702520,
  "name": "UPHC Jawaharnagar..",
  "address": "Saharanpur - Delhi Rd Indra Enclave Gulab Vatika Jawahar Nagar Ghaziabad",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Sadhna Enclave",
  "active": true,
  "id": "ujAvng5ZJlcsjAYceqSDU",
  "created": "1635258182.227605",
  "modified": "1635258182.2327409",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjAYceqSDU"
},
{
  "center_id": 697436,
  "name": "CHC Dasna...",
  "address": "Community Health Centre Dasna Near Devi Mandir Road Ghaziabad",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Sadhna Enclave",
  "active": true,
  "id": "ujAvng5ZJlcsjZElgPufca",
  "created": "1635258183.2663674",
  "modified": "1635258183.2815588",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjZElgPufca"
},
{
  "center_id": 661018,
  "name": "UPHC Sadhna Enclave",
  "address": "Sadhna Enclave Khora Gaon",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Sadhna Enclave",
  "active": true,
  "id": "ujAvng5ZJlcsjXlW6V2Ie",
  "created": "1635258184.0862935",
  "modified": "1635258184.0925612",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjXlW6V2Ie"
},
{
  "center_id": 884059,
  "name": "Jeevan Asha Hospital..",
  "address": "Village DIDOLI GANG CANAL ROAD MURADNAGAR GHAZIABAD",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Sadhna Enclave",
  "active": true,
  "id": "ujAvng5ZJlcsjW0H3at8UW",
  "created": "1635258185.8577752",
  "modified": "1635258185.8664872",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjW0H3at8UW"
},
{
  "center_id": 567998,
  "name": "Max Hospital Vaishali-1",
  "address": "W-3 Sector-1 Vaishali Ghaziabad",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Sadhna Enclave",
  "active": true,
  "id": "ujAvng5ZJlcsjDgl55gLCK",
  "created": "1635258187.3719237",
  "modified": "1635258187.3777344",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjDgl55gLCK"
},
{
  "center_id": 915439,
  "name": "UPHC Rahul Garden Loni",
  "address": "Near Bansal Lakkar Wale Rahul Garden Mandoli",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Sadhna Enclave",
  "active": true,
  "id": "ujAvng5ZJlcsjCIH8mh0hw",
  "created": "1635258188.1408632",
  "modified": "1635258188.1544995",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjCIH8mh0hw"
},
{
  "center_id": 922980,
  "name": "UPHC Hansaw.",
  "address": "13-22 C Block Rd Shatabdi Puram Block I Block E Govindpuram Ghaziabad Uttar Pradesh 201015",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Sadhna Enclave",
  "active": true,
  "id": "ujAvng5ZJlcsjAauMUX9g",
  "created": "1635258189.678662",
  "modified": "1635258189.6832912",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjAauMUX9g"
},
{
  "center_id": 567998,
  "name": "Max Hospital Vaishali-1",
  "address": "W-3 Sector-1 Vaishali Ghaziabad",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Sadhna Enclave",
  "active": true,
  "id": "ujAvng5ZJlcsjBUfC0c0TM",
  "created": "1635258190.5544634",
  "modified": "1635258190.5595245",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjBUfC0c0TM"
},
{
  "center_id": 812201,
  "name": "Chhabil Dass Public School",
  "address": "CH CHHABIL DASS B - Block Patel Mang Sector 7 Pocket B Patel Nagar 2nd Patel Nagar Ghaziabad",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Sadhna Enclave",
  "active": true,
  "id": "ujAvng5ZJlcsjB3sf05x3Q",
  "created": "1635258194.4628298",
  "modified": "1635258194.4670177",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjB3sf05x3Q"
},
{
  "center_id": 915451,
  "name": "UPHC Tilakram Colony",
  "address": "Tilakram Colony Near Satty Sabhasad Indrapuri Colony Lal Bagh Colony 201102",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Sadhna Enclave",
  "active": true,
  "id": "ujAvng5ZJlcsjArFnFcYmq",
  "created": "1635258195.6065102",
  "modified": "1635258195.6113915",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjArFnFcYmq"
},
{
  "center_id": 689171,
  "name": "LYF Hospital",
  "address": "NH-4 GYANKHAND-1 INDIRAPURAM GHAZIABAD",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Makana",
  "active": true,
  "id": "ujAvng5ZJlcsjEWXfV06GW",
  "created": "1635258196.7618024",
  "modified": "1635258196.7680163",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjEWXfV06GW"
},
{
  "center_id": 728422,
  "name": "UPHC Pappu Colony",
  "address": "Composite School Pappu Colony Near Bharat Petrol Pump Bhopura Ghaziabad",
  "state_name": "Uttar Pradesh",
  "district_name": "Ghaziabad",
  "block_name": "Sadhna Enclave",
  "active": true,
  "id": "ujAvng5ZJlcsjwiSQsIe6u",
  "created": "1635258198.1984051",
  "modified": "1635258198.1964746",
  "type": "note",
  "dismissed": false,
  "direction": "self",
  "sender_id": "ujAvng5ZJlcsjwiSQsIe6u"
}
```


FRIENDS

ME

FOLLOWING

Oct 26, 8:57 PM

OTP

5860

slots

UPHC Indrapuri 18 To 44A-10 Block A Opp.Rana Pratap Park Indrapuri Colony Loni Ghaziabad430

slots

UPHC Daulatpura,Uphe Daulatpura FCI Road Near Gunga Halwai Ghaziabad673

slots

Chandra Laxmi HealthCareNH-3 Gyan Khand-1 Indirapuram Ghaziabad182

slots

UPHC Khairati NagarNear Tigri Gol Chakkar Khairati Nagar Ghaziabad201

slots

UPHC Saraswati Colony 18 To 44Saraswati Colony Sahibabad Ghaziabad320

All Devices

Send a message

Pushing

Mirroring

SMS

Channels

Account

FRIENDS

ME

FOLLOWING

slots

UPHC Tilakram ColonyTilakram Colony Near Satty Sabhasad Indrapuri Colony Lal Bagh Colony 201102398

slots

LYF HospitalNH-4 GYANKHAND-1 INDIRAPURAM GHAZIABAD160

slots

UPHC Pappu ColonyComposite School Pappu Colony Near Bharat Petrol Pump Bhopura Ghaziabad399

slots

APHC BhovapurRaza Pur BHOWAPUR GHAZIABAD96

slots

LYF HospitalNH-4 GYANKHAND-1 INDIRAPURAM GHAZIABAD197

slots

UPHC Makanpur...UPHC Makanpur 1st Gyankhand 4IndrapuramGhaziabad314

All Devices

Send a message

Pushing

Mirroring

SMS

Channels

Account

FRIENDS

ME

FOLLOWING

slots

UPHC Vijay Nagar Ist...Central Samudayik Kendra Chadakya Chowk Sec 9 Ramleela Groun D304

slots

UPHC Sadik NagarE 847 Sadik Nagar Nandgram Ghaziabad332

slots

UPHC Jawaharnagar..Saharanpur - Delhi Rd Indra Enclave Gulab Vatika Jawahar Nagar Ghaziabad486

slots

CHC Dasna...Community Health Centre Dasna Near Devi Mandir Road Ghaziabad151

slots

UPHC Sadhna EnclaveSadhna Enclave Khora Gaon336

slots

Jeevan Asha Hospital.Village DIDOLI GANG CANAL ROAD MURADNAGAR GHAZIABAD390

All Devices

Send a message

Pushing

Mirroring

SMS

Channels

Account

FRIENDS

ME

FOLLOWING

slots

Divoc HealthF-22232425 Durga Plaza District Centre Sanjay Nagar Ghaziabad189

slots

APHC TalhetaModinagar Rd Bhojpur400

slots

DCH Sanjay Nagar...Sec 23 Rajnagar Ghaziabad305

slots

Columbia Asia HospitalNH-24 Near Land Craft Golf Link Ghaziabad6

slots

UPHC BLS Inds. AreaBls Inds Area Yadav Ngr Near Rto Office Ghaziabad300

Apex Vaccination Centre158 Rajnagar Extension Rajnagar Ghaziabad678

Oct 26

All Devices

Send a message

Pushing

Mirroring

SMS

Channels

Account

32 | Page

REQUIREMENTS AND SPECIFICATIONS

Hardware Requirements

Processor	: Pentium IV(minimum)
Hard Disk	: 40 GB
RAM	:256 MB(minimum)

Software Requirements

Operating System	: Windows
Technology	: PYTHON
IDE Used	: Pycharm Community2019.1.3x64

CONCLUSION

The Project has been successfully completed by having established the user friendly interface with the help of python tkinter library. At the same time there is some scope for improvements in the future. It can be possible to make it more Attractive by adding more variety of functions. This application may helpful for the humans as it show the weather details of a city.

I believe the trial has shown conclusively that it is both possible and desirable to use Python as the principal teaching language:

- It is Free (as in both cost and source code).
- It is trivial to Install on a Windows PC allowing students to take their Interest further. For many the hurdle of installing a Pascal or C compiler on a Windows machine is either too expensive or too complicated.

Future Scope:

- Enhancement of Security.
- Availability across 10km.
- Notifications and reminders.
- Map view of available slots.

The training program having three destination was a lot more useful than staying at one place throughout the whole 6 weeks. In my opinion, I have gained lots of knowledge and experience needed to be successful in great engineering challenge as in my opinion, Engineering is after all a Challenge and not a job.

REFERENCES

- Video lectures of the internship training programme
- <https://apisetu.gov.in/api/cowin/cowin>
- [https://en.wikipedia.org/wiki/Python_\(programming_language\)](https://en.wikipedia.org/wiki/Python_(programming_language))
- www.w3schools.com/python