**INDUSTRIAL TRAINING REPORT**

**DOCTOR WEB SCRAPPER**

Submitted in partial fulfillment of the

Requirements for the award of

**Degree of Bachelor of Technology in Computer Science Engineering**



Submitted By

Name: DANISH KHAN

University Roll No. 1814310071

**SUBMITTED TO:**

**Department of Computer Science & Engineering**

IMS Engineering college

GHAZIABAD (UP).

**DECLARATION**

I hereby declare that the Industrial Training Report entitled (" Doctor Web Scrapping ") is an authentic record of my own work as requirements of Industrial Training during the period from 02.08.2021 to 02.09.2021 for the award of degree of B.Tech. (Computer Science & Engineering), IMS ENGINEERING COLLEGE, GZB , under the guidance of Mr. Ravi Sir

**(Signature of student)**

**(ROLL NO - 1814310071)**

**Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Head of Department**

**(Signature and Seal)**

**ACKNOWLEDGEMENT**

An endeavor is not complete and successful till the people who make it possible are given due credit for making it possible. I take this opportunity to thank all those who have made the endeavor of mine successful for me and for all.

I take the opportunity to express my gratitude to all of them who in some or the other way helped me to accomplish this project. The study cannot be completed without their guidance, assistance ,inspiration and co-operation.

I would like to express my deep sense of gratitude to my respected project guide Mr. Hakim Singh for his support & cooperation.

Secondly I would like to thank my close friends &well wishers who were there for me whenever required.

Last but not the least WWW which provided with all the required data & information.

**About Company/Industry/Institute**

At SkillUp Online, we focus on creating enriching, engaging learning journeys that helps our learners their build career in desired technology skills in easy, fun & meaningful way. With the aim of deriving maximum learning outcomes, our courses are developed by industry leaders like Microsoft, leverage innovative learning strategies, hands on labs & interactive activities, personalized mentoring & coaching services. And all this with complete flexibility of learning anytime, anywhere! What’s more is that for each course you complete successfully, you earn a verified course completion certificate by industry leaders like Microsoft.

#### Our Misson :

Our Mission is to deliver top of the line technology learning experiences that are affordable, flexible and available to people across the globe. With SkillUp Online, anybody who is focused towards building their career in new age technology skills should be easily able to so at their convenience.

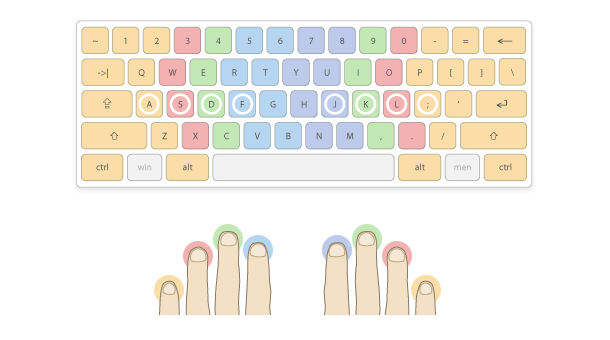
**CERTIFICATES**

**TABLE OF CONTENTS**

* **PROBLEMS**
* **INTRODUCTION**
* **WORKING DIAGRAM**
* **LANGUAGE USED**
* **FEATURES**
* **WORKING**
* **CODE**

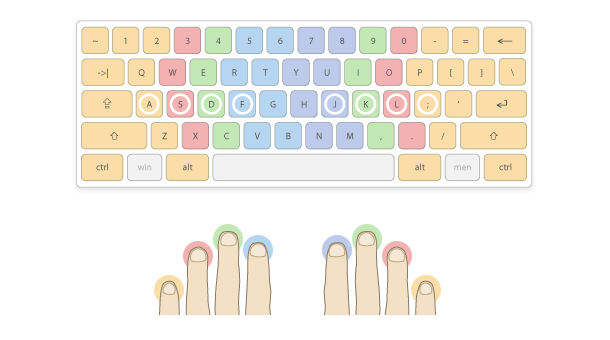
**PROBLEMS IN DAILY LIFE**

Data is very important nowadays so people are collecting data just by copying data from web pages and pasting it to the file which is very hectic kind of work because what happens if the data is too big than in this case we will waste our lot of time to copy data from the web pages and then paste it into the file, so there is need of an agent ( Program ) which will do our work in a fraction of second and thus here WEB SCRAPPING program comes into the picture.



**INTRODUCTION**

The idea behind the project is to make scrapping of web page easy and fast so that people who waste a lot of the time in scrapping the data line by line word by word would be able to do it in a fraction of the second just by typing two to three words as input to the program and rest of the thing will be done by the program.



**LANGUAGE USED**

**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.

**MODULE USED**

* **BEAUTIFULSOUP**
* **REQUESTS**
* **MATH**
* **FLASK**
* **XLSXWRITER**

**Beautiful Soup**

[Beautiful Soup](http://www.crummy.com/software/BeautifulSoup/) is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

These instructions illustrate all major features of Beautiful Soup 4, with examples. I show you what the library is good for, how it works, how to use it, how to make it do what you want, and what to do when it violates your expectations.

This document covers Beautiful Soup version 4.9.3. The examples in this documentation should work the same way in Python 2.7 and Python 3.8.

You might be looking for the documentation for [Beautiful Soup.](http://www.crummy.com/software/BeautifulSoup/bs3/documentation.html) If so, you should know that Beautiful Soup 3 is no longer being developed and that support for it will be dropped on or after December 31, 2020. If you want to learn about the differences between Beautiful Soup 3 and Beautiful Soup 4, see [Porting code to BS4](https://www.crummy.com/software/BeautifulSoup/bs4/doc/#porting-code-to-bs4).

**FLASK**

Flask is a micro [web framework](https://en.wikipedia.org/wiki/Web_framework) written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)). It is classified as a [micro framework](https://en.wikipedia.org/wiki/Microframework) because it does not require particular tools or libraries. It has no [database](https://en.wikipedia.org/wiki/Database) abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

Applications that use the Flask framework include [Pinterest](https://en.wikipedia.org/wiki/Pinterest" \o "Pinterest) and [LinkedIn](https://en.wikipedia.org/wiki/LinkedIn)

**REQUESTS**

Requests is a [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) [library](https://en.wikipedia.org/wiki/Software_library) for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) programming language. The goal of the project is to make HTTP requests simpler and more human-friendly. The current version is 2.26.0. Requests is released under the [Apache License 2.0](https://en.wikipedia.org/wiki/Apache_License).

Requests is one of the most popular Python libraries that is not included with Python. It has been proposed that Requests be distributed with Python by default.

**MATH**

The Python Math Library **provides us access to some common math functions and constants** in Python, which we can use throughout our code for more complex mathematical computations. The library is a built-in Python module, therefore you don't have to do any installation to use it

In [computer science](https://en.wikipedia.org/wiki/Computer_science), a math library (or maths library) is a component of a [programming language](https://en.wikipedia.org/wiki/Programming_language)'s [standard library](https://en.wikipedia.org/wiki/Standard_library) containing [functions](https://en.wikipedia.org/wiki/Subroutine) (or [subroutines](https://en.wikipedia.org/wiki/Subroutine)) for the most common [mathematical functions](https://en.wikipedia.org/wiki/Mathematical_function), such as [trigonometry](https://en.wikipedia.org/wiki/Trigonometry) and [exponentiation](https://en.wikipedia.org/wiki/Exponentiation).

Examples include:

* the C standard library math functions,
* Java maths library
* 'Prelude.Math' in haskell.

In some languages (such as haskell) parts of the standard library (including maths) are imported by default.

More advanced functionality such as [linear algebra](https://en.wikipedia.org/wiki/Linear_algebra) is usually provided in 3rd party libraries, such as a [linear algebra library](https://en.wikipedia.org/wiki/Linear_algebra_library) or [vector maths library](https://en.wikipedia.org/wiki/Vector_maths_library).

**XLSXWRITER**

XlsxWriter is a Python module for writing files in the XLSX file format. It can be used to write text, numbers, and formulas to multiple worksheets. Also, it supports features such as formatting, images, charts, page setup, auto filters, conditional formatting and many others.

XlsxWriter is a Python module that can be used to write text, numbers, formulas and hyperlinks to multiple worksheets in an Excel 2007+ XLSX file. It supports features such as formatting and many more, including:

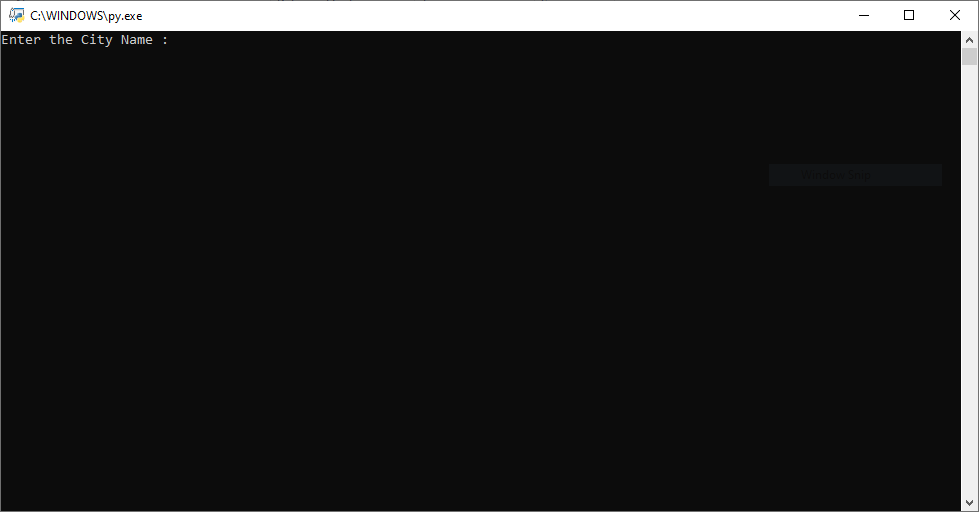
* 100% compatible Excel XLSX files.
* Full formatting.
* Merged cells.
* Defined names.
* Charts.
* Autofilters.
* Data validation and drop down lists.
* Conditional formatting.
* Worksheet PNG/JPEG/GIF/BMP/WMF/EMF images.
* Rich multi-format strings.
* Cell comments.
* Textboxes.
* Integration with Pandas.
* Memory optimization mode for writing large files.

**Features :-**

* It will collect all the names of doctors who have a particular speciality district wise.
* It will display information in the web page and will also create a Excel file and store all the information in the Excel file.
* It will take just two input to perform the task.
* It will take just 2 minute to fetch data from more than 120 pages.

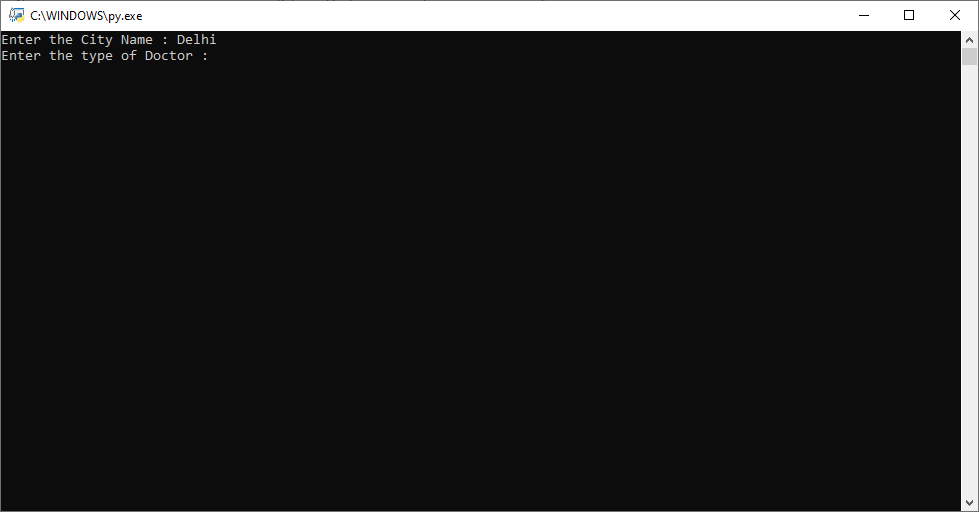
**WORKING**

**Step 1 :- Execute web scrapping Program and Enter name of the city**



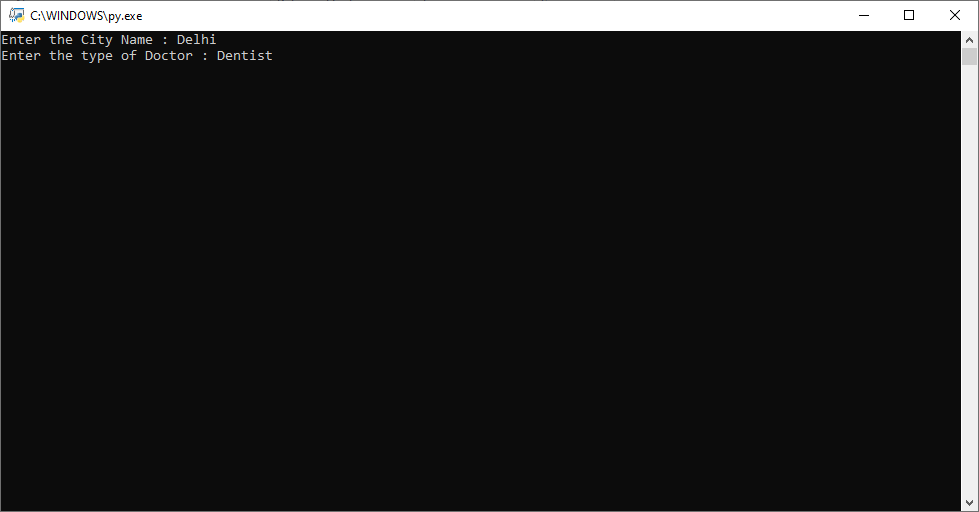
**WORKING**

**Step 2 :- Enter the Type of Doctor ( Specialty )**



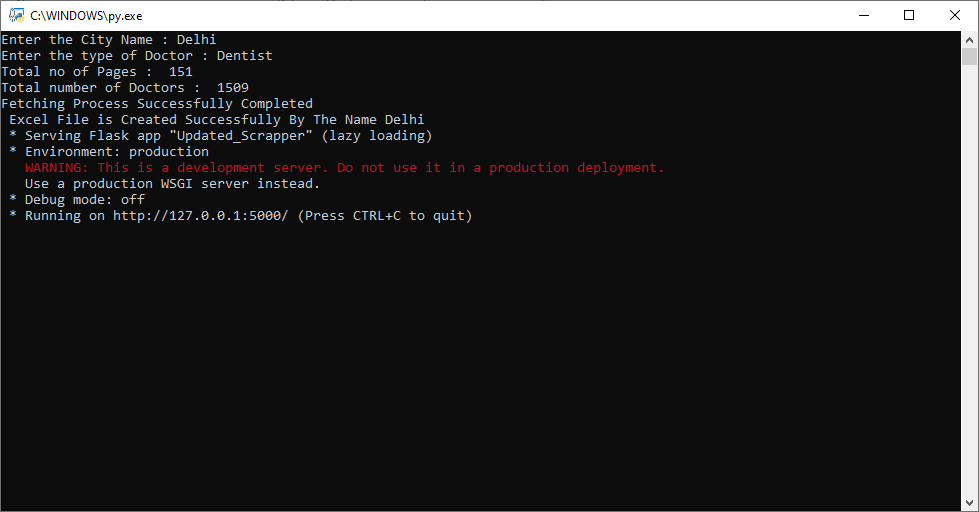
**WORKING**

**Step 3 :- Press enter button and wait 1 to 2 minute**



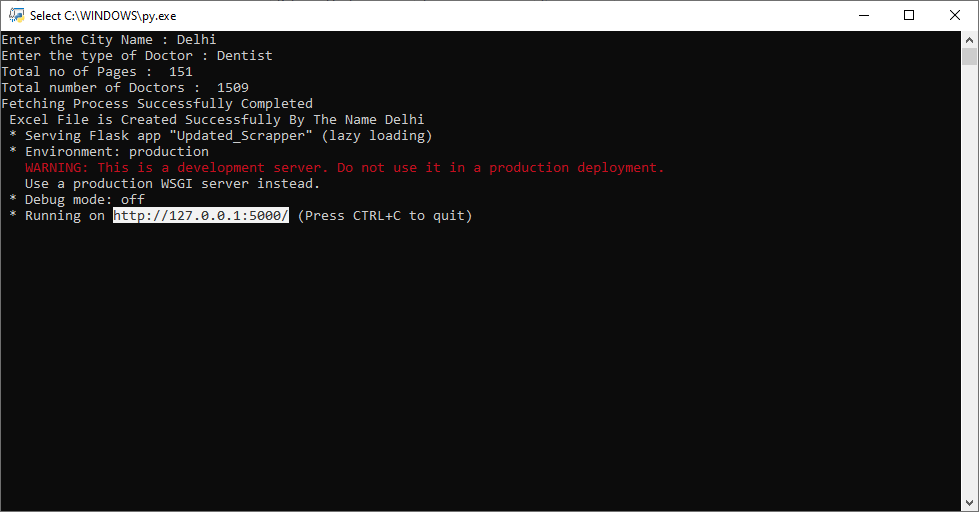
**WORKING**

**Step 4 :- Data Fetching Process successfully completed**



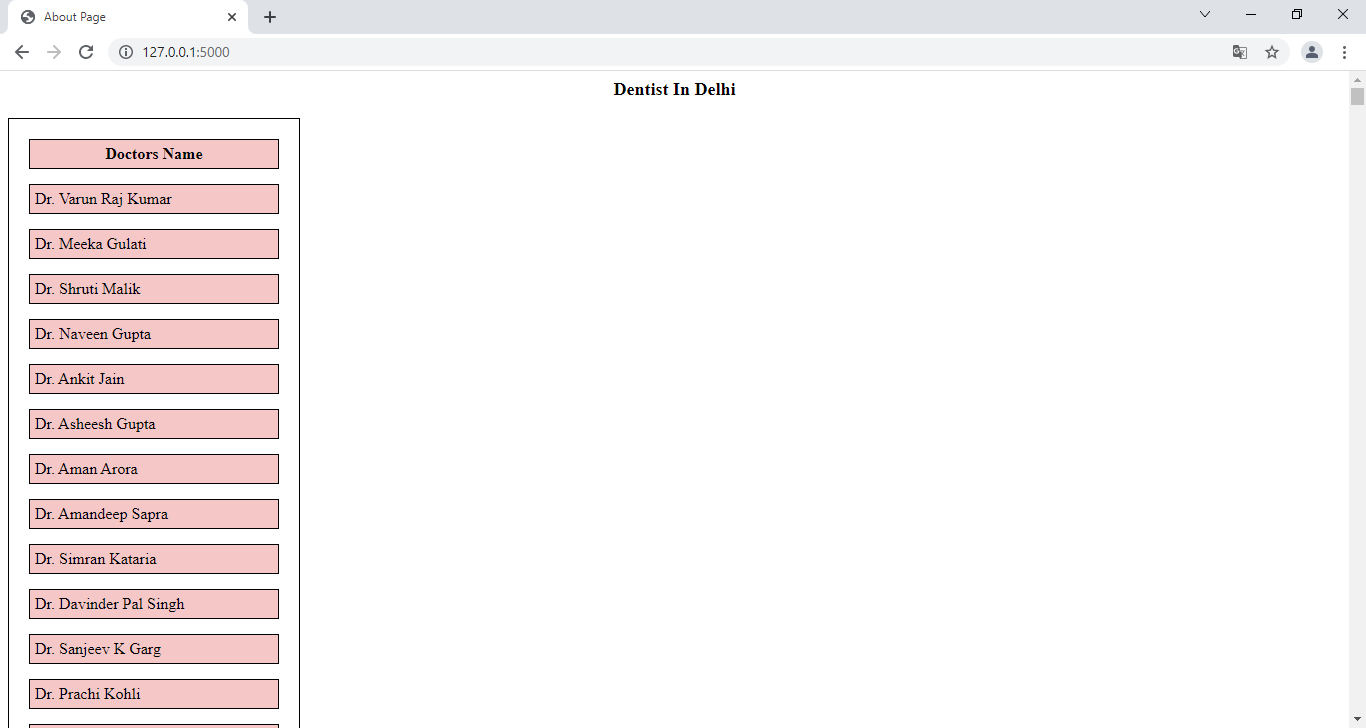
**WORKING**

**Step 5 :- Copy the HTTP Link and Paste it into the Browser**



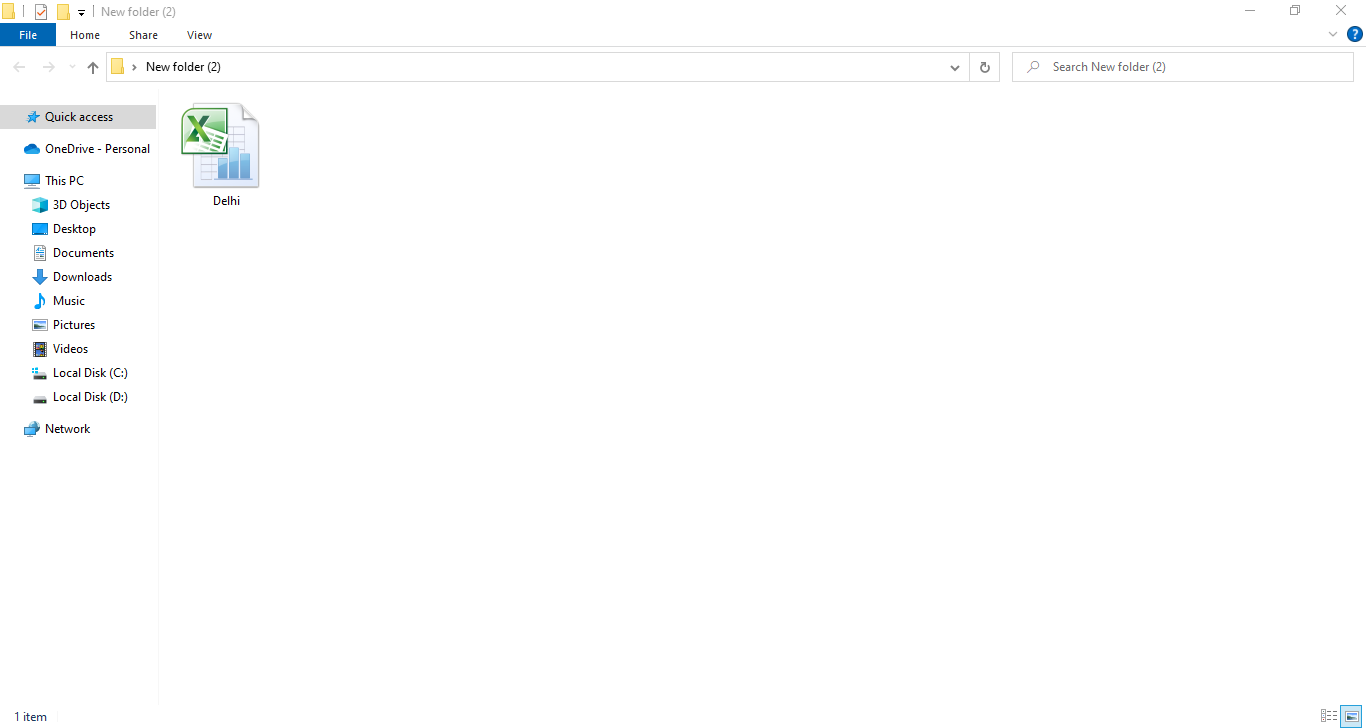
**WORKING**

**Step 6 :- All the Fetch data is visible in the browser**



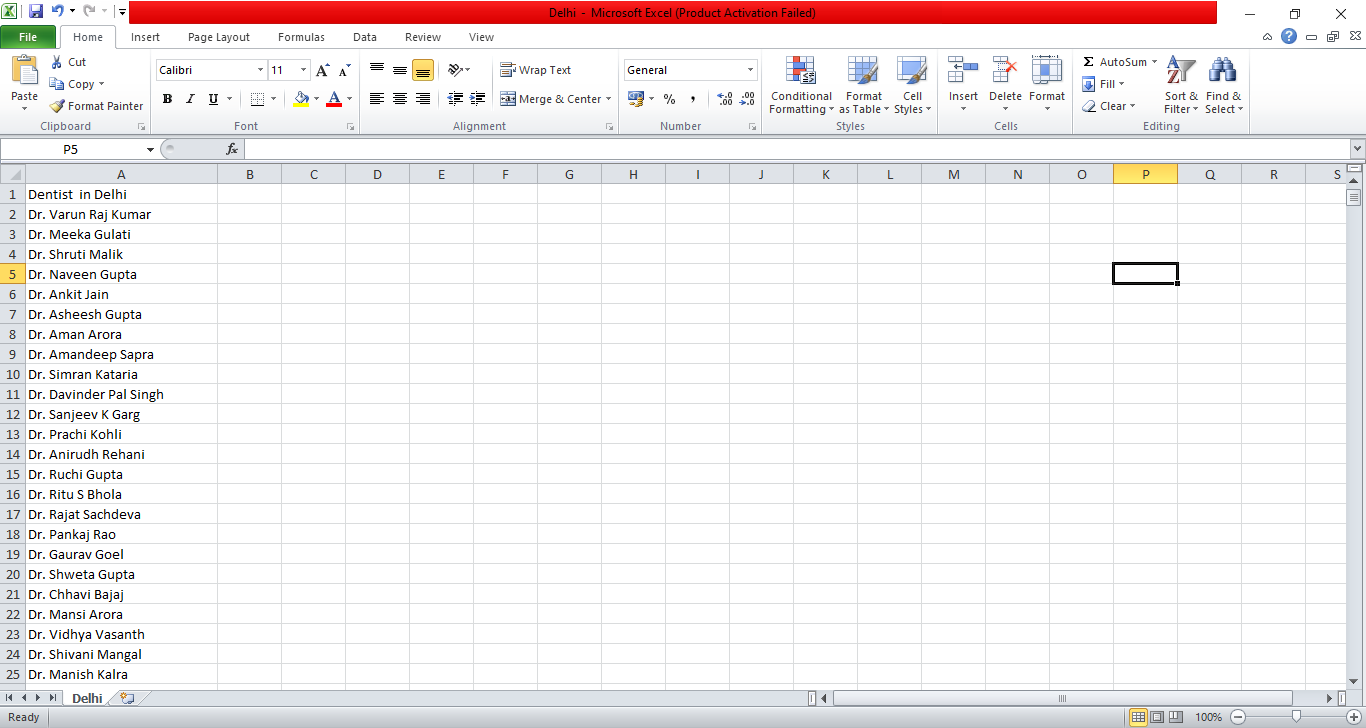
**WORKING**

**Step 7 :- Excel File will get create by the name of city**



**WORKING**

**Step 8 :- Excel File contains all the list of Doctors**



**CODE ( WEB SCRAPPER )**

from bs4 import BeautifulSoup

import requests

import math

from flask import Flask, render\_template

import xlsxwriter as xl

app = Flask (\_\_name\_\_)

def excel\_data():

    address = "C:\\Users\\Danish\\Desktop\\" + city + ".xlsx"

    Excel\_file = xl.Workbook(address)

    Excel\_sheet = Excel\_file.add\_worksheet(city)

    Excel\_sheet\_heading = speciality +  ' ' + 'in' + ' ' + city

    Excel\_sheet.write(0,0,Excel\_sheet\_heading)

    for i in range(1,len(l)+1):

        Excel\_sheet.write(i,0,l[i-1])

    Excel\_file.close()

splitting = []

def fetch(url):

    Not\_found=0

    Count = 0

    data = requests.get(url)

    plane\_text = BeautifulSoup(data.text,'html.parser')

    Doctor\_name = plane\_text.find\_all("h2",class\_= "doctor-name")

    if(Doctor\_name == []):

        Not\_found = 1

    if Not\_found == 0:

        for i in Doctor\_name:

            l.append(i.text)

            # print(i.text)

    if(Not\_found == 1):

        print("NOT!!! FOUND")

def url\_generator(url,pages):

    old\_url = url

    count\_pages = 1

    for i in range(1,pages+1):

        new\_url = old\_url+"&page={}".format(i)

        count\_pages = count\_pages + 1

        if count\_pages == pages:

            print("Fetching Process Successfully Completed")

            excel\_data()

            print(" Excel File is Created Successfully By The Name" + ' '+ city)

        # print(new\_url)

        fetch(new\_url)

def total\_page(url):

    datas = requests.get(url)

    datas = BeautifulSoup(datas.text,'html.parser')

    #datas = datas.find\_all("h1",class\_="u-xx-large-font u-bold u-t-grey5") <---- i was using this previously

    datas = datas.find\_all("p",class\_="u-xx-large-font u-bold")

    global l

    global splitting

    for i in datas:

        #print(i.text)

        splitting = (i.text).split()

    #print("Total number of Doctors :: ",splitting[0])

    splitted = (int(splitting[0]))/10

    return math.ceil(splitted)

city = input("Enter the City Name : ")

doctor\_type = list(map(str,input("Enter the type of Doctor : ").split()))

speciality = ''

for i in doctor\_type:

    speciality = speciality + i + ' '

len\_list = len(doctor\_type) - 1

str1 = 'https://www.practo.com/search/doctors?results\_type=doctor&q=%5B%7B%22word%22%3A%22'

for i in doctor\_type:

    str1 = str1 + i

    if (len\_list) :

        str1 = str1 + '%20'

        len\_list = len\_list - 1

url = str1 + '%22%2C%22autocompleted%22%3Atrue%2C%22category%22%3A%22subspeciality%22%7D%5D&city=' + city

l = []

# url = "https://www.practo.com/search/doctors?results\_type=doctor&q=%5B%7B%22word%22%3A%22{}%22%2C%22autocompleted%22%3Atrue%2C%22category%22%3A%22subspeciality%22%7D%5D&city={}".format(doctor\_type,city)

print("Total no of Pages : ",total\_page(url))

print("Total number of Doctors : ",splitting[0])

url\_generator(url,total\_page(url))

@app.route('/')

# def index():

#     return render\_template("index.html")

# @app.route('/about')

def about():

    return render\_template("about.html", city = city, Doctors = speciality ,items = l)

if \_\_name\_\_=="\_\_main\_\_":

    app.run()

**CODE ( TEMPLATES )**

<html>

    <head>

        <title>

            About Page

        </title>

        <style>

         p{

             text-align: center;

             font-size: large;

             font-weight: bold;

             text-transform: capitalize;

             color :black

         }

            table, th, td{

  border: 1px solid black;

  padding: 5px;

}

th, td{

    background-color:rgb(245, 199, 199) ;

}

table {

  border-spacing: 15px;

}

        </style>

    </head>

    <body>

        <p>{{ Doctors }} in {{ city }}</p>

        <table >

            <tr>

                <th> Doctors Name </th>

            </tr>

            {% for i in items %}

            <tr>

                <td>{{ i }}</td>

            </tr>

            {% endfor %}

        </table>

    </body>

</html>