

219017

**Appendix 1**  
**Health Monitoring System**

**A PROJECT REPORT**

*Submitted by*

**Shubham Thaker**

**180130111108**

*In partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

*In*  
**Electronics and Communication**  
**Gec, Gandhinagar**



**Gujarat Technological University, Ahmedabad**

[April 2022]

## Appendix 2



## InfoLabz IT Services PVT.LTD

**Vraj Avenue, 405, above SAM'S Pizza,**  
**Nr.Commerce Six Road,**  
**Navrangpura, Ahmedabad,**  
**Gujarat 380009**

## CERTIFICATE

This is to certify that the project report submitted along with the project entitled **Health Monitoring System** has been carried out by **Shubham Thaker** under my guidance in partial fulfillment for the degree of Bachelor of Engineering in **Electronics and communication**, 8<sup>th</sup> Semester of Gujarat Technological University, Ahmadabad during the academic year 2021-22.

Neetirajsingh Chhasatia

K.G. Maradia

Internal Guide

Head of the Department

219017



INfolabz IT SERVICES  
WEB DEVELOPMENT | APP DEVELOPMENT | IOT

Date: 20 / 04 / 2022

**TO WHOM IT MAY CONCERN**

This is to certify that Thaker Shubham, a student of Government Engineering College, Gandhinagar has successfully completed his internship in the field of Web Development from 3 January 2022 to 16 April 2022 (Total number of Weeks: 15) under the guidance of Mr. Chintan Nagrecha.

His internship activities include Web Development in Django framework, API development, IoT component handling and circuit designing.

During the period of his internship program with us, he had been exposed to different processes and was found diligent, hardworking and inquisitive.

We wish him every success in his life and career.



Ms. Zarna Shah,  
Human Resources Department,  
INfolabz IT SERVICES PVT. LTD.



+91 8866662662  
+91 9725790521



contact@infolabz.in  
www.infolabz.in



405 Vraj Avenue, Above SAM'S Pizza  
Nr, Commerce Six Rd, Navrangpura,  
Ahmedabad, Gujarat 380009

## Appendix 4



## Government Engineering College Sector 28, Gandhinagar

### DECLARATION

We hereby declare that the Internship report submitted along with the Project entitled **Health Monitoring System** submitted in partial fulfillment for the degree of Bachelor of Engineering in Electronics and communication to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me / us at Infolabz IT Services PVT. LTD under the supervision of Chintan Nagrecha and that no part of this report has been directly copied from any students' reports or taken from any other source, without providing due reference.

Name of the Student

Sign of Student

1      Shubham Thaker



## Government Engineering College

### Sector 28, Gandhinagar

### ACKNOWLEDGEMENT

I wish to express our sincere gratitude to our External guide **Mr. Chintan Nagrecha** for continuously guiding me at the company and answering all my doubts with patience.

I also thank our parents, friends and all the members of the family for their precious support and encouragement, which they had provided in completion of our work. In addition to that, we would also like to mention the company personals who gave us the permission to use and experience the valuable resources required for the internship.

Thus, In conclusion to the above said, we once again thank the staff members of **InfoLabz IT services Pvt. Ltd.** for their valuable support in completion of the project.

Thank You,  
Shubham Thaker

## **ABSTRACT**

This internship was carried out at InfoLabz IT Services Pvt. Ltd. InfoLabz is a IT services company that is committed to providing quality resources delivering a very personable experience to our clients and it offers internship opportunities to students in need of education in web development, App development. Internship is an opportunity to relate what has been covered in class and what is applicable in the field in an operational environment. The purpose of this internship is to fulfil the core equipment for the award of Bachelor of Degree in Electronics and Communication to get a practical aspect of theoretical work studied at the university and to understand the operations in the IT sector and to enable students to get experience in different tasks.

During my internship period a number of approaches and exposure methods were used which included: hands on, through relevant material, and also questions and answer approaches.

I worked as a Python Programmer who was given a responsibility of developing a website at the end of the internship I gained knowledge on how to design and develop a Website using the Django.

In conclusion, this was an opportunity to develop and enhance skills and competencies in my career field which I actually achieved.

## List of figures

<b>Fig 1.1 Scope of work.....</b>	<b>13</b>
<b>Fig 3.1 – 3.4 List Methods .....</b>	<b>17-18</b>
<b>Fig 3.6 – 3.10 Tuple Methods and Functions.....</b>	<b>19-20</b>
<b>Fig 3.12 – 3.13 String Methods .....</b>	<b>21</b>
<b>Fig 3.15 – 3.18 Dictionary Methods .....</b>	<b>22-23</b>
<b>Fig 3.19 – 3.21 If And Else.....</b>	<b>24-26</b>
<b>Fig 3.23 – 3.40 API And Its Application .....</b>	<b>26-35</b>
<b>Fig 4.1 – 4.24 OOP Examples.....</b>	<b>36-47</b>
<b>Fig 5.1 – 5.6 Random Examples.....</b>	<b>48-50</b>
<b>Fig 5.8 – 5.33 Xlrd Examples .....</b>	<b>51-59</b>
<b>Fig 5.34 – 3.43 Datetime Examples.....</b>	<b>61-63</b>
<b>Fig 5.44 – 5.54 Numpy Examples.....</b>	<b>63-68</b>
<b>Fig 5.55 – 5.66 Pandas Examples.....</b>	<b>68-72</b>
<b>Fig 5.68 – 5.89 Matplotlib Examples .....</b>	<b>73-86</b>
<b>Fig 6.1 Arduino Board.....</b>	<b>88</b>
<b>Fig 6.2 Esp8266 Board.....</b>	<b>89</b>
<b>Fig 6.3 Esp32 Board.....</b>	<b>90</b>
<b>Fig 6.4 Arduino IDE .....</b>	<b>90</b>
<b>Fig 7.1 Types of Sensors .....</b>	<b>91</b>
<b>Fig 7.2 Water level Sensor.....</b>	<b>92</b>
<b>Fig 7.3 Water level Sensor Circuit.....</b>	<b>92</b>
<b>Fig 7.4 Output Water level Sensation .....</b>	<b>93</b>
<b>Fig7.5 Temperature and Humidity Sensor (DHT11).....</b>	<b>93</b>
<b>Fig 7.6 Circuit for DHT11 .....</b>	<b>94</b>
<b>Fig 7.7 Output for DHT11.....</b>	<b>94</b>
<b>Fig 7.8 Rain drop Sensor .....</b>	<b>95</b>
<b>Fig 7.9 Circuit for RDS.....</b>	<b>95</b>
<b>Fig 7.10 Output of RDS .....</b>	<b>96</b>
<b>Fig 7.11 Soil Moisture Sensor .....</b>	<b>96</b>
<b>Fig 7.12 Circuit SMS.....</b>	<b>97</b>
<b>Fig 7.13 Output of SMS .....</b>	<b>97</b>

219017

<b>Fig 8.1 Django Launcher Page.....</b>	<b>100</b>
<b>Fig 8.2 Login Page.....</b>	<b>101</b>
<b>Fig 8.3 Django Administration Page .....</b>	<b>102</b>
<b>Fig 8.4-8.5 Permissions .....</b>	<b>102-103</b>
<b>Fig 8.6-8.7Models and Admin.....</b>	<b>103-104</b>
<b>Fig 8.8 Output After runserver.....</b>	<b>104</b>
<b>Fig 8.9 – 8.10 Models, Admin.....</b>	<b>105-106</b>
<b>Fig 8.11-8.12 Output after runserver .....</b>	<b>106</b>
<b>Fig 8.13 Output of After URL handling .....</b>	<b>109</b>
<b>Fig 8.14 Urls,Views .....</b>	<b>110</b>
<b>Fig 8.15 Urls of Project.....</b>	<b>111</b>
<b>Fig 8.16 Settings file of project.....</b>	<b>111</b>
<b>Fig 8.17 Urls file of Project.....</b>	<b>112</b>
<b>Fig 8.18 Output after adding media .....</b>	<b>112</b>
<b>Fig 9.1 Circuit for ECG and Temperature.....</b>	<b>115</b>
<b>Fig 9.2 Circuit for Pulse and Oxygen .....</b>	<b>115</b>
<b>Fig 9.3 Sensors Used .....</b>	<b>115</b>
<b>Fig 9.4-9.7 Website.....</b>	<b>116-117</b>
<b>Fig 10.1 Code for SMS.....</b>	<b>118</b>
<b>Fig 10.2 Code for gmail .....</b>	<b>118</b>

## Table of Contents

Acknowledgement .....	i
Abstract.....	ii
List of Figures.....	iii
List of Tables .....	iv
Table of Contents.....	v
<b>Chapter 1 Overview of the Company.....</b>	<b>13</b>
1.1 History.....	13
1.2 Different Scope / Work .....	13
<b>Chapter 2 Introduction to Internship and Project .....</b>	<b>14</b>
2.1 Summary .....	14
2.2 Purpose.....	14
2.3 Objective .....	15
2.4 Scope.....	15
2.5 Technology and Literature Review.....	15
2.6 Internship Planning .....	15
2.6.1 Internship Development Approach and Justification.....	15
2.6.2 Internship Efforts and Time, Cost Estimation.....	16
2.6.3 Roles and Responsibilities .....	16
2.6.4 Group Dependencies .....	16
<b>Chapter 3 Basics of Python .....</b>	<b>17</b>
3.1 List and its methods .....	17
3.2 Tuple .....	18
3.3 string and its methods .....	20
3.4 dictionary .....	21
3.5 Condition (If else statement).....	23
3.6 API .....	26
<b>Chapter 4 Object Oriented Programming (OOP) .....</b>	<b>36</b>
4.1Explanation .....	36
<b>Chapter 5 Important Library .....</b>	<b>48</b>
5.1 Random.....	48
5.2 Xlrd .....	50
5.3 DateTime.....	59

219017	
5.4 Numpy .....	63
5.5 Pandas .....	68
5.6 Matplotlib.....	73
<b>Chapter 6 Basics of Internet of things.....</b>	<b>87</b>
6.1 Arduino .....	87
6.2 EspSeries (NodeMcu) .....	88
6.3 Arduino IDE.....	89
<b>Chapter 7 Sensors .....</b>	<b>91</b>
7.1 Explanation .....	91
7.2 Water Level Sensor.....	92
7.3 Temperature Sensor .....	93
7.4 Rain Drop Sensor.....	95
7.5 Soil Moisture Level Sensor.....	96
<b>Chapter 8 Django Framework.....</b>	<b>98</b>
8.1 Introduction to Django.....	98
8.2 Procedure .....	99
8.3 Permission.....	102
8.4 URL Handling.....	103
8.5 Primary and Foreign key.....	106
8.6 URL Handling.....	108
8.7 Media .....	111
<b>Chapter 9 Project work .....</b>	<b>113</b>
9.1 Introduction.....	113
9.2 Problem Definition.....	113
9.3 Working .....	113
9.4 Flow Chart of the project .....	114
9.5 Circuit .....	115
9.6 Sensors used.....	115
9.7 Website .....	116
<b>Chapter 10 Notification .....</b>	<b>118</b>
10.1 Messages .....	118
10.2 Email .....	118

219017

<b>Chapter 11 Conclusion .....</b>	<b>119</b>
11.1 Overall analysis.....	119
11.2 Problem Encounters and Possible Solutions.....	119
11.3 Summary of Internship .....	119
11.4 Limitation and Future Enhancement.....	120
<b>References .....</b>	<b>121</b>

# 1. OVERVIEW OF THE COMPANY

## 1.1 History

Incorporated in 2011, along with leading parent IT Subsidiary Company, we have our own image and reputation in IT industry. We are having qualified and experienced team who are willing and ready to take any challenges. We know the upgradation in technologies, so instead of working as an individual, we work as a team and ready to take all challenges that come across.

We have divided our work in main two segments 1) Web Development & 2) App Development. In Web Development process, we are having dedicated team who are having utmost experience of all major technologies. We have dedicated team for UI/UX and Graphics designs as well. We have hundreds of clients who are divided across the globe and enjoying our services. In the other segment of APP Development, We are having dedicated team for API/Web Service handling and Material Designs. Our extreme effort to give something new in every app helps us to learn something new at every step.

InfoLabz offers Industry oriented Live Projects Training for CE/IT (BE/B.TECH & DIPLOMA ENGINEERING), BCA/MCA, BScIT/MScIT students. Students are provided with hands-on experience. This industry Oriented Projects training enables the student to understand the engineering concepts in practical way. The student can apply his knowledge in implementing practical application software / LIVE websites.

## 1.2 Organization chart:-

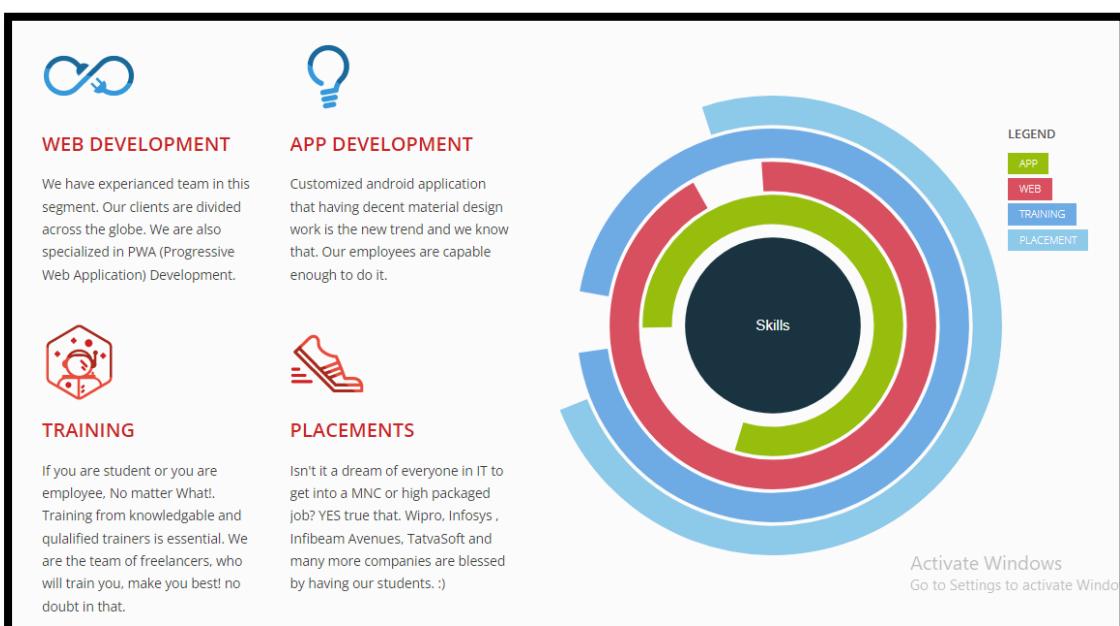


Fig. 1.1

## 2. INTRODUCTION TO INTERNSHIP AND PROJECT

### 2.1 Summary

The project, which was assigned to me, is related health department. The project was health-monitoring system in which I am using 3 different types of the sensors which helps to monitor patients temperature, oxygen, pulse, ECG. So at the times of covid-19 this device which was been created by us is very impactful in monitoring patients health.

In this project we have made a device which is been connected with server and the value fetched by the sensors is updated in the database and if any exceeds the normal value than there will be an email pass to the doctor regarding the excitation in the value.

By using the database, we have created a website in which there is a dashboard, which contains different parameters values, and it has features like nearby hospitals in addition, pharmacy.

In that we have assigned different tables like oxygen, temperature etc.

Addition to this I am in the web development team as a python developer so I have successfully gone through web development in Django framework, API development.

### 2.2 Purpose

The main purpose behind this whole project is prevent the death of covid-19 as in this project it has a function of measuring the oxygen value in the patient's body it will prevent the death of patients at certain limit as it will warn the doctors and other family members about the condition of the patient. Other than this, the product is also useful to the normal people as they can maintain their health using the different parameters given by this product.

## **2.3 Objective**

The objective of the project is to be as contactless as possible with patients as we have seen the situation at the time of the pandemic the infected patients infects the doctors too as doctors has to go closer to them. Therefore, in such case this product is useful as it has connection with Wi-Fi so doctors can monitor the health of the patients by watching their past readings from different tables present in the dashboard.

## **2.4 Scope**

Scope of this project includes the ability to measure the temperature, oxygen, pulse, ECG to maintain their health and helpful to the Government Hospitals as there are too many patients at a time so by using this device it will be easier for the doctors to monitor their respective patients and cure them as fast as possible.

## **2.5 Technology and Literature Review**

Technology used in my project is mainly Arduino in which coding is being done regarding to the circuit operation and other than that there is also UI designing in which we have used bootstrap HTML, CSS, JAVASCRIPT mainly Django.

## **2.6 Internship Planning**

### A. Internship Development Approach and Justification:

In the beginning of the internship I have started working on the API than I got good amount knowledge of the different datatypes than I have proceed to numpy, pandas, matplotlib, than finally I have to Django and at last created a website using template given and by the Django. Addition to that I have also get to work on different sensors like rain, moisture level sensor, temperature sensor, ECG sensor, pulse oximeter sensor, water level sensor.

### B. Internship Effort and Time, Cost Estimation:

As this is a very basic project that we are building a system which will be beneficial to different categories of people the main aim is to know what we have gained Throughout the internship and we have made the team effort to make this project close to perfection as much as possible and our industrial guide has designed the series of the task in such way that in the terms of efforts, one can learn to make this project in about a span of 3 months easily.

### C. Roles and Responsibilities:

I have assigned the role of python developer in the team of web development my roles is to first learn the python language as accurate as possible and then apply that knowledge in the given task which is been given by the industrial guide.

Addition to that I have assigned the project of health monitoring system so in that I have a part in creation of the website and circuit.

### D. Group Dependencies:

As I am from the EC department, I have already the knowledge regarding the circuit making and in the span of the 3 months internship, I have enough knowledge regarding the sensors and their operators so I am more into circuit and less in UI designing and other teammates are in remaining part.

### 3. Basics of Python

#### 3.1 List and its method:-

- The list is created by placing elements inside square brackets “[ ]” separated by commas.



- The above shown are some of the list methods which are used frequently ,the following are the use of list methods:-
  - .append is used add any element in the list.
  - .sort is used for sorting in descending and ascending manner.
  - .clear is used to clear the whole list by deleting all list elements.
  - .count is used to the number of elements in the list.
- Some examples:-

```

my_list = [1, 2, 3, 4, 5]
my_list.append(6)
print(my_list)
[1, 2, 3, 4, 5, 6]
  
```

Fig. 3.1

A screenshot of the PyCharm IDE interface. On the left is the project tree with a folder named 'http' containing files 'venv', '2.py', and 'main(3).py'. The 'External Libraries' section is also visible. In the center is the code editor with the following Python code:

```

1 my_list = [1, 2, 3, 4, 5]
2 my_list.pop(1)
3 print(my_list)

```

The line 'my\_list.pop(1)' is highlighted in blue. Below the code editor is the run configuration panel showing 'Run: 2' and the command 'D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py'. The output window shows the result: '[1, 3, 4, 5]'.

Fig. 3.2

A screenshot of the PyCharm IDE interface. The project tree and run configuration are identical to Fig. 3.2. The code editor contains the following Python code:

```

1 my_list = [2, 1, 4, 5, 3]
2 my_list.sort()
3 print(my_list)

```

The line 'my\_list.sort()' is highlighted in blue. The output window shows the result: '[1, 2, 3, 4, 5]'.

Fig. 3.3

A screenshot of the PyCharm IDE interface. The project tree and run configuration are identical to Fig. 3.2. The code editor contains the following Python code:

```

1 my_list = [1, 2, 3, 4, 5]
2 print(my_list.count(1))
3

```

The line 'print(my\_list.count(1))' is highlighted in blue. The output window shows the result: '1'.

Fig. 3.4

### 3.2 Tuple:-

- Tuples are written with round brackets “( )” separated by commas.
- A tuple is a collection, which is ordered and unchangeable.
- The difference between the list and the tuple is that tuples are immutable and the list are mutable.
- Following are the tuple functions used frequently:-

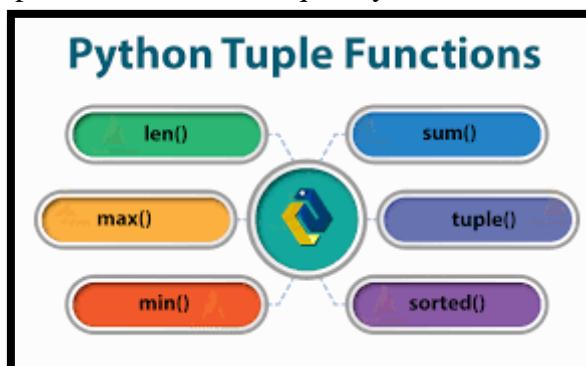
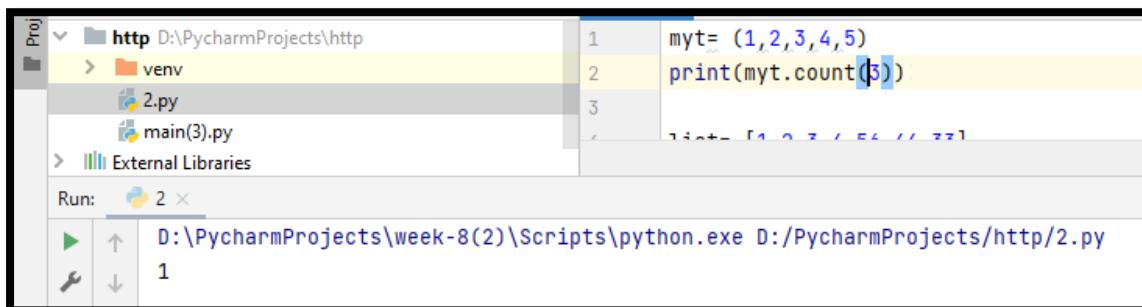


Fig. 3.5

- Len () function is used to find the number of objects in the tuple
- Max () is used to find the object which has the maximum value in the tuple
- Min () is used to find the object which has the minimum value in the tuple.
- Tuple () is used to make a tuple
- Sorted () to sort the objects in ascending order.
- Sum () to find the sum of the objects in the tuple.
- Addition to that the tuple has also the methods namely count and index, in which count gives the number of the occurrence of that specific object and index gives the index of that object on the tuple



```

Proj http D:\PycharmProjects\http
  venv
  2.py
  main(3).py
External Libraries

Run: 2 ×
  D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
  1

```

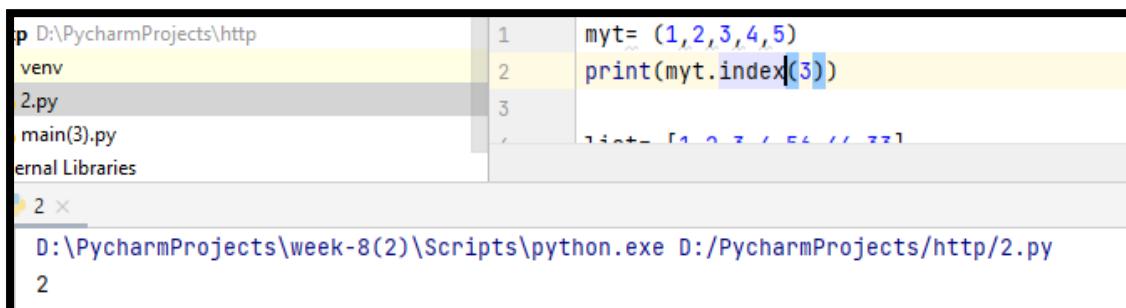
The screenshot shows the PyCharm IDE interface. The project navigation bar on the left lists 'http' (selected), 'venv', '2.py', 'main(3).py', and 'External Libraries'. Below the project bar, there's a 'Run' section with a dropdown set to '2 ×' and a run button. The main code editor window contains three lines of Python code:

```

1 myt= (1,2,3,4,5)
2 print(myt.count(3))
3

```

Fig. 3.6



```

Proj http D:\PycharmProjects\http
  venv
  2.py
  main(3).py
  External Libraries

Run: 2 ×
  D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
  2

```

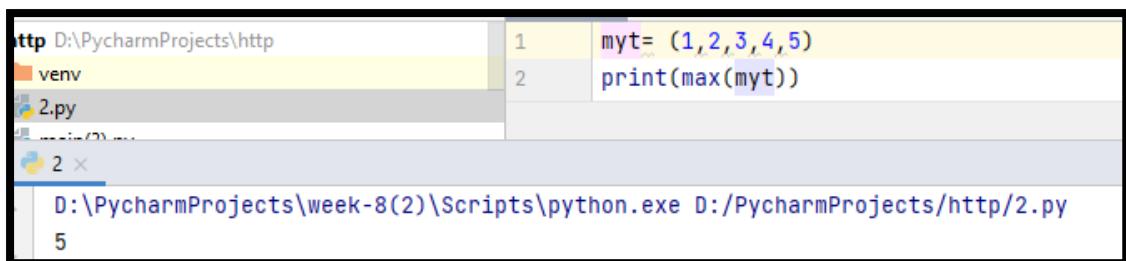
The screenshot shows the PyCharm IDE interface. The project navigation bar on the left lists 'http' (selected), 'venv', '2.py', 'main(3).py', and 'External Libraries'. Below the project bar, there's a 'Run' section with a dropdown set to '2 ×' and a run button. The main code editor window contains three lines of Python code:

```

1 myt= (1,2,3,4,5)
2 print(myt.index(3))
3

```

Fig. 3.7



```

Proj http D:\PycharmProjects\http
  venv
  2.py
  main(3).py
  External Libraries

Run: 2 ×
  D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
  5

```

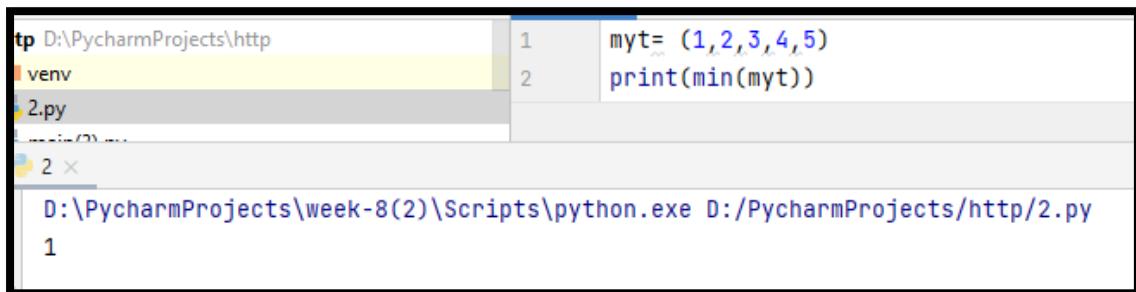
The screenshot shows the PyCharm IDE interface. The project navigation bar on the left lists 'http' (selected), 'venv', '2.py', 'main(3).py', and 'External Libraries'. Below the project bar, there's a 'Run' section with a dropdown set to '2 ×' and a run button. The main code editor window contains three lines of Python code:

```

1 myt= (1,2,3,4,5)
2 print(max(myt))
3

```

Fig. 3.8



```

tp D:\PycharmProjects\http
venv
2.py
D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
1

```

The screenshot shows a PyCharm interface. The project navigation bar at the top has 'tp' and 'venv'. Below it, the file tree shows '2.py'. The code editor contains two lines of Python code:

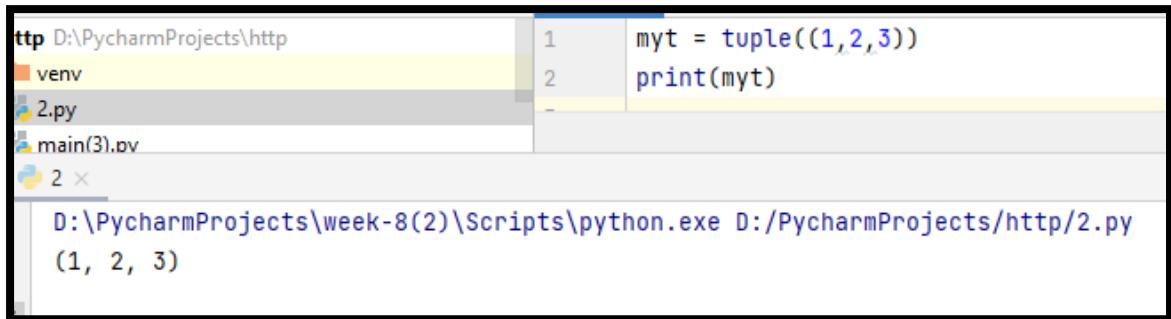
```

1 myt= (1,2,3,4,5)
2 print(min(myt))

```

The output window at the bottom shows the command run: 'D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py' followed by the result '1'.

Fig. 3.9



```

tp D:\PycharmProjects\http
venv
2.py
main(3).py
D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
(1, 2, 3)

```

The screenshot shows a PyCharm interface. The project navigation bar at the top has 'tp' and 'venv'. Below it, the file tree shows '2.py' and 'main(3).py'. The code editor contains two lines of Python code:

```

1 myt = tuple((1,2,3))
2 print(myt)

```

The output window at the bottom shows the command run: 'D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py' followed by the result '(1, 2, 3)'.

Fig. 3.10

### 3.3 String:-

- In Python, **Strings** are arrays of bytes representing Unicode characters. However, Python does not have a character data type, a single character is simply a string with a length of one. Square brackets can be used to access elements of the string.

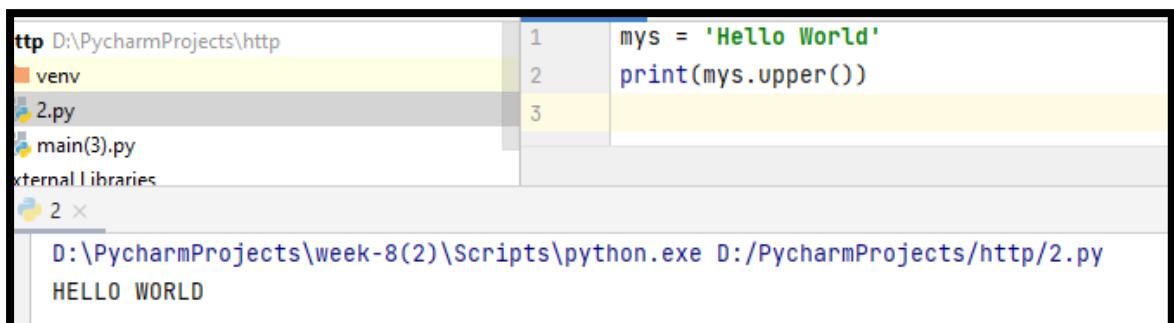
### Common String Methods

---

- `upper()`
- `lower()`
- `capitalize()`
- `startswith()`
- `endswith()`
- `strip()`
- `find()`
- `split()`
- `join()`

Fig. 3.11

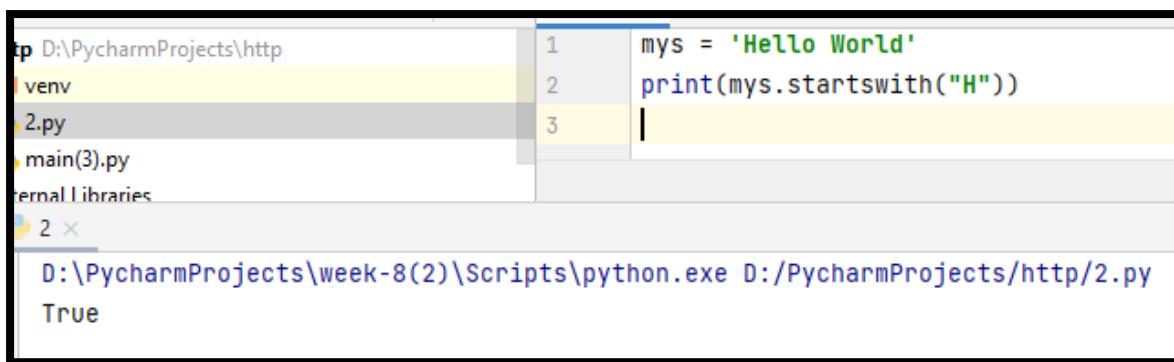
- Following are some of the applied example:-



```
tp D:\PycharmProjects\http
venv
2.py
main(3).py
External Libraries
2 ×

D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
HELLO WORLD
```

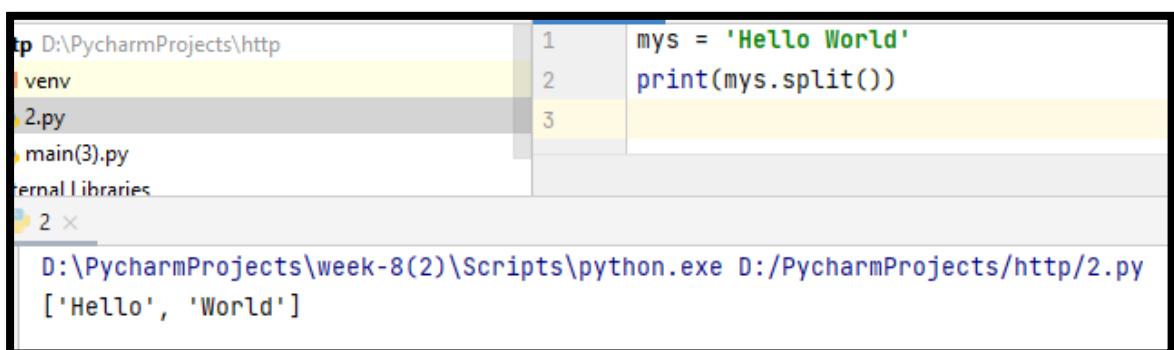
Fig. 3.12



```
tp D:\PycharmProjects\http
venv
2.py
main(3).py
External Libraries
2 ×

D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
True
```

Fig. 3.12(1)



```
tp D:\PycharmProjects\http
venv
2.py
main(3).py
External Libraries
2 ×

D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
['Hello', 'World']
```

Fig. 3.13

### 3.4 Dictionary:-

- Dictionary** in Python is an unordered collection of data values, used to store data values like a map, which, unlike other Data Types that hold only a single value as an element, Dictionary holds key: value pair. Key-value is provided in the dictionary to make it more optimized.
- Dictionary methods:-

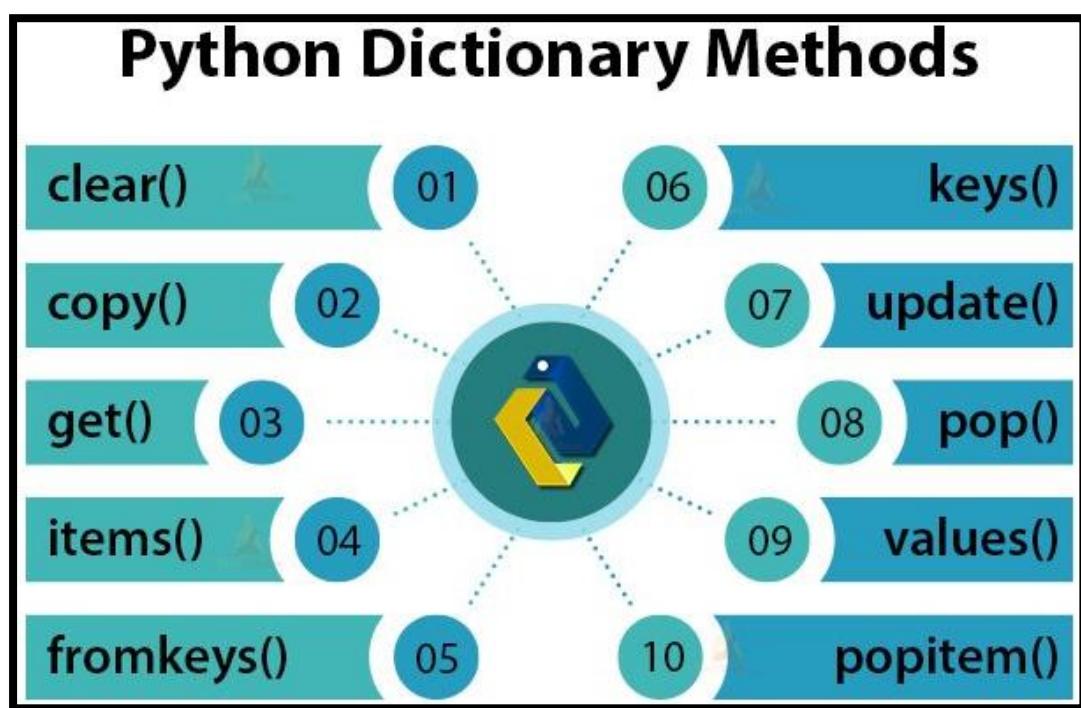


Fig. 3.14

- Following are the applied Examples:-

```
tp D:\PycharmProjects\http
venv
2.py
main(3).py
Internal Libraries
2
D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
Ahmedabad
```

Fig. 3.15

```
tp D:\PycharmProjects\http
venv
2.py
main(3).py
Internal Libraries
2
D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
{2: 'Rajkot', 3: 'Jamnagar'}
```

Fig. 3.16

```

http D:\PycharmProjects\http
venv
2.py
main(3).py
External Libraries
2 x
D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
{1: 'Ahmedabad', 2: 'Rajkot'}

```

Fig. 3.17

```

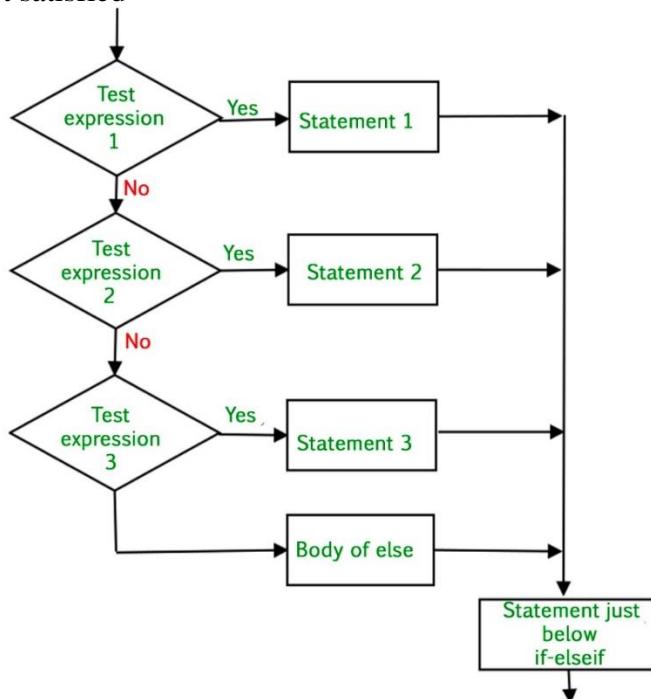
http D:\PycharmProjects\http
venv
2.py
main(3).py
External Libraries
2 x
D:\PycharmProjects\week-8(2)\Scripts\python.exe D:/PycharmProjects/http/2.py
dict_values(['Ahmedabad', 'Rajkot', 'Jamnagar'])

```

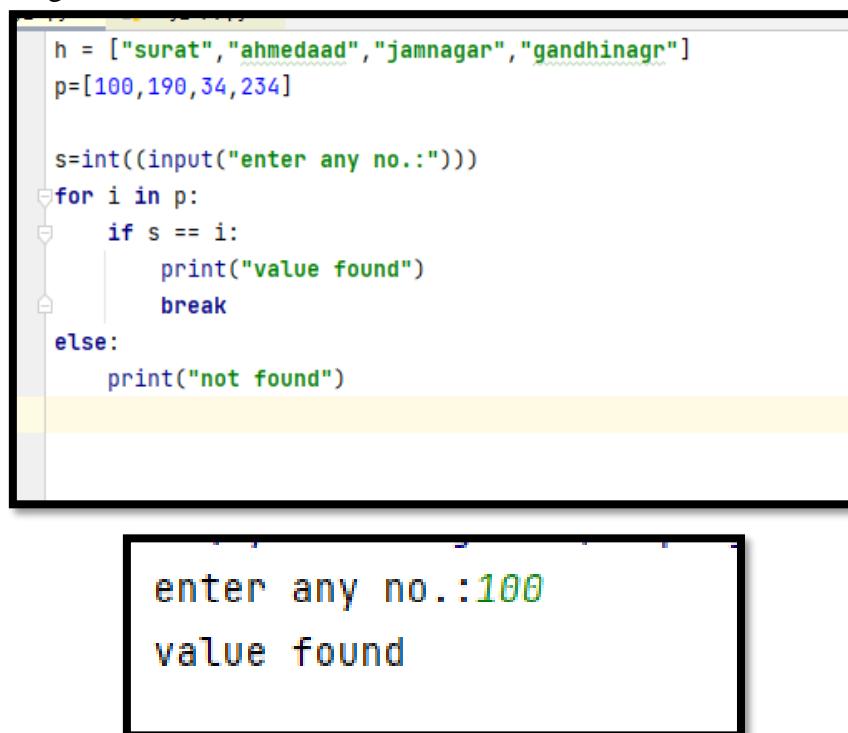
Fig. 3.18

### 3.5 Conditional Statements (If else):-

- There are different types of condition like “ if ” after if there will condition written in which there will some value and ended with a colon the next should start with an indent in the “ else ” part there is also a condition written which came in action when if condition is not satisfied



- Some of the examples:-
- As shown in the bellow code there was value which user will input and the program will search that specific value from list if the value is in the list the “value found” message will be given other than that “value not found” will be reflected on the screen
- It is searching from the list.



```

h = ["surat", "ahmedaad", "jamnagar", "gandhinagr"]
p=[100,190,34,234]

s=int((input("enter any no.:")))
for i in p:
    if s == i:
        print("value found")
        break
else:
    print("not found")

```

enter any no.:100  
value found

Fig. 3.19

- Here to access the list we can use the nested if condition to check for the multiple things and to access the list
- As shown in the below I worked to verification of the email id and password if they are entered correctly than it will reflect login successful other than that it will reflect login failure.
- There multiple condition checked here as shown like:
  1. If the email id is blank and password is written than it will reflect “email id cannot be blank”
  2. If email is written perfectly but password is not written it will give the message “password cannot be blank”
  3. If both are given input but they are not in the list then it will give the message “user not found”
  4. If both password and email id is not written by the user than it will show the messages “email id and password cannot be blank”
- This way we can check for multiple condition

```

1 up=["a@b.com","j@k.com","c@s.com"]
2 p=['abc123','bcd234',' sdf123']
3 upname=input("enter an email: ")
4 pname=input("enter password: ")
5 for i in range (0,len(up)):
6     if upname=='' and pname!="":
7         print("email can not be blank")
8         break
9     elif upname!='' and pname=="":
10        print("password can not be blank")
11        break
12     elif upname=='' and pname=='':
13        print ("both can not be blank")
14        break
15     else:
16         if upname == up[i] and pname == p[i]:
17             print("login succesful")
18             break
19         else:
20             print("yes")
21     else:
22         print("not succesful")

```

enter an email: *a@b.com*  
 enter password: *abc123*  
*login succesful*

Fig. 3.20

- One Interesting example on which I have worked is as shown as below which contains the concepts of nested if else and use of for loop.
- EMAIL LIST
- PASSWORD LIST
- SUBSCRIPTION LIST = [0,1,1,0]
- STATUS LIST = [1,1,1,1]
- Login check:
  1. Fields can not be blank, if any field is blank, display specific message: - email can not be blank/ password can not be blank or both fields can not be blank.
  2. If email entered by user is not found then display message: user is not registered with us.
  3. If subscription status is 0 then not allowed to login, display message your subscription is over, please renew to continue.
  4. If status field is 0 then display you are blocked. If not blocked then and only then check his subscription status.
  5. If user is registered and password is not correct : incorrect password

Code and output regarding to this:-

```

1 up=["a@b.com","j@k.com","c@s.com"]
2 p=['abc123','bcd234',' sdf123']
3 slist =[1,0,1]
4 stlist=[1,1,0]
5 upname=input("enter an email: ")
6 pname=input("enter password: ")
7 for i in range(0, len(up)):
8     if upname == '' and pname != '':
9         print("email can not be blank")
10        break
11    elif upname != '' and pname == '':
12        print("password can not be blank")
13        break
14    elif upname == '' and pname == '':
15        print("All Fields are required")
16        break
17    else:
18        for i in up:
19            if upname == i:
20                print("hello user")
21                for i in range(0, len(up)):
22                    d1 = {up[i]: slist[i]}
23                    d2 = {up[i]: stlist[i]}
24                    if upname == up[i]:
25                        if pname == p[i]:
26                            for j in d2:
27                                if upname == j:
28                                    if (d2[j]) == 1:
29                                        for k in d1:
30                                            if upname == k:
31                                                if (d1[k]) == 1:
32                                                    print("Welcome")
33                                                    break
34                                                else:
35                                                    print("your subscription is over, please renew to continue.")
36                                                    break
37                                            else:
38                                                print("you are blocked.")
39                                                break
40                                        else:
41                                            print("incorrect password")
42                                            break
43                                break
44                            break

```

Fig. 3.21

### 3.6 API:-

- The full form of the API id Application program interface which have the data of database shown in json format.

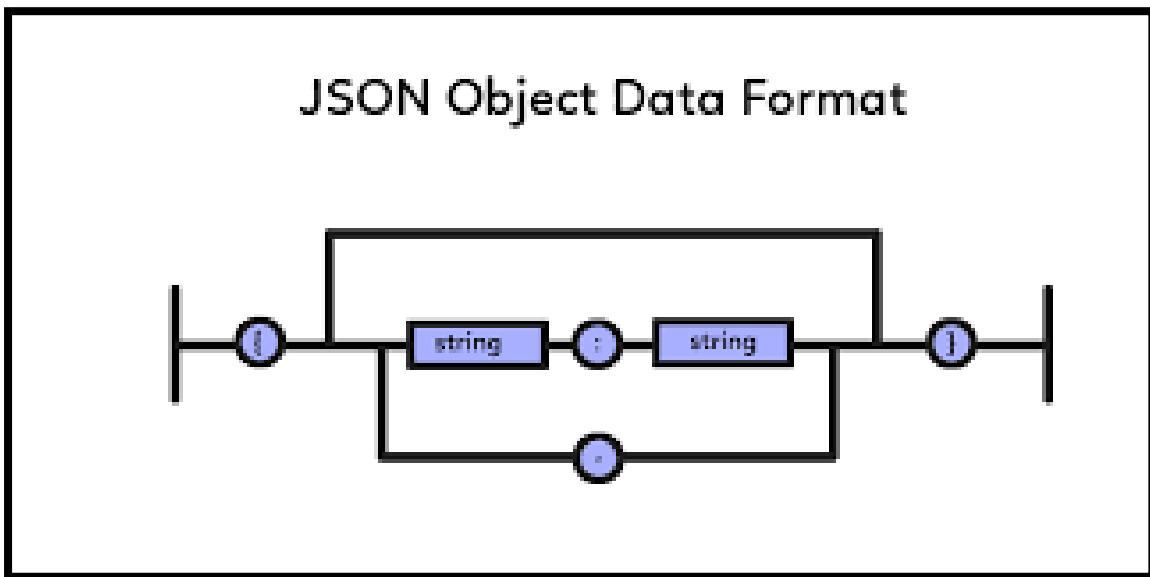


Fig. 3.22

- Json formatting is nothing but dictionary datatype in which we have pair of keys and values
- So we can access the API which will be available in the internet I will show the example of an API and then I show how to access that
- There is package called “requests” which has to be installed in our python library what this package do is it will requests the link which we will give inside the url in this “requests.get(url)”
- Using this it will requests the URL and then we need to convert that fetched data in the json format by using “.json”
- First, I will show a normal example of json format and how it works than I will shift the URL part.
- Made a program user can insert the key that he/she want to delete from the dictionary and at the output that key and its value is removed from the dictionary.
- Code is shown as below:

```
dict = {  
    "a": "one",  
    "b": "two",  
    "c": "three"  
}  
dict["d"] = "four"  
print(dict)  
u = input("enter the key you want to delete: ")  
for i in dict:  
    if i == u:  
        dict.pop(i)  
        print(dict)  
        break  
    else:  
        print("not found")
```

Fig. 3.23

- The output:

```
{'a': 'one', 'b': 'two', 'c': 'three', 'd': 'four'}  
enter the key you want to delete: a  
{'b': 'two', 'c': 'three', 'd': 'four'}
```

Fig. 3.24

- The link for API - <https://data.covid19india.org/data.json>

```
{  
    "cases_time_series": [  
        {  
            "dailyconfirmed": "1",  
            "dailydeceased": "0",  
            "dailyrecovered": "0",  
            "date": "30 January 2020",  
            "dateymd": "2020-01-30",  
            "totalconfirmed": "1",  
            "totaldeceased": "0",  
            "totalrecovered": "0"  
        },  
        {  
            "dailyconfirmed": "0",  
            "dailydeceased": "0",  
            "dailyrecovered": "0",  
            "date": "31 January 2020",  
            "dateymd": "2020-01-31",  
            "totalconfirmed": "1",  
            "totaldeceased": "0",  
            "totalrecovered": "0"  
        }  
    ]  
}
```

Fig. 3.25

- The format above shown is the API of the Covid 19 data now using this API I have done certain tasks by going through them we can understand the JSON formatting and API handling in a great extent following are the code and output regarding to them.
- Addition to that there two types of api:
  1. Structured api
  2. Unstructured api
- There are some pros and cons of both of them.
- As simple if we wanted to get the number of the keys of the api we can run a for inside the api and we can get that
- The following is an example of that
- Other than there are some major things given to us as to find different values inside an api so the code regarding to them is shown below
- The API is nothing but displaying the data of the database in json format, this thing is done by company because from that no employee can change the data inside the database they can just access it.
- Therefore, API is useful tool for the security purpose of the company.

```

non_work D:\PycharmProjects\python_work
letsdoi library root
main.py
External Libraries
Screencasts and Consoles

1 import requests
2
3 covid = requests.get("https://data.covid19india.org/data.json")
4 coviddata = covid.json()
5
6 for i in coviddata :
7     print(i)
8
9
10
11

```

```

main(3) ×
D:\PycharmProjects\python_work\letsdoi\Scripts\python.exe "C:/Users/Shubham/OneDrive/Desktop/all about int
cases_time_series
statewise
tested

```

Fig. 3.26

- The above code is for the normal key fetching from an api
- We can also print our output according to our need the following is an example of that

```

import requests

alldata= requests.get("https://data.covid19india.org/data.json")
ldata= alldata.json()

print ("data available of",len(ldata["cases_time_series"]),"days")

for i in range(0,len(ldata["cases_time_series"])):
    print(ldata["cases_time_series"][i]["date"],
          " New cases:",ldata["cases_time_series"][i]["dailyconfirmed"],
          " total cases:",ldata["cases_time_series"][i]["totalconfirmed"],
          " New deaths:",ldata["cases_time_series"][i]["dailydeceased"],
          " Total deaths:",ldata["cases_time_series"][i]["totaldeceased"],
          " Today recoverd:",ldata["cases_time_series"][i]["dailyrecovered"],
          " Total recoverd:",ldata["cases_time_series"][i]["totalrecovered"])

```

Fig. 3.27

```

live x
16 July 2021 New cases: 38117 total cases: 31063776 New deaths: 560 Total deaths: 412532 Today recovered: 43878 Total recovered: 30220021
17 July 2021 New cases: 41283 total cases: 31105059 New deaths: 517 Total deaths: 413049 Today recovered: 42051 Total recovered: 30262072
18 July 2021 New cases: 38330 total cases: 31143389 New deaths: 501 Total deaths: 413550 Today recovered: 38545 Total recovered: 30300617
19 July 2021 New cases: 29420 total cases: 31172809 New deaths: 372 Total deaths: 413922 Today recovered: 45356 Total recovered: 30345973
20 July 2021 New cases: 42128 total cases: 31214937 New deaths: 3998 Total deaths: 417920 Today recovered: 36876 Total recovered: 30382849
21 July 2021 New cases: 41687 total cases: 31256624 New deaths: 510 Total deaths: 418430 Today recovered: 38891 Total recovered: 30421740
22 July 2021 New cases: 34863 total cases: 31291487 New deaths: 481 Total deaths: 418911 Today recovered: 38403 Total recovered: 30460143
23 July 2021 New cases: 39501 total cases: 31330988 New deaths: 542 Total deaths: 419453 Today recovered: 35144 Total recovered: 30495287
24 July 2021 New cases: 40286 total cases: 31371274 New deaths: 541 Total deaths: 419994 Today recovered: 40038 Total recovered: 30535325
25 July 2021 New cases: 38179 total cases: 31409453 New deaths: 411 Total deaths: 420405 Today recovered: 35945 Total recovered: 30571270
26 July 2021 New cases: 30820 total cases: 31440273 New deaths: 418 Total deaths: 420823 Today recovered: 42503 Total recovered: 30613773
27 July 2021 New cases: 42971 total cases: 31483244 New deaths: 641 Total deaths: 421464 Today recovered: 41653 Total recovered: 30655426
28 July 2021 New cases: 7115 total cases: 31504100 New deaths: 110 Total deaths: 422011 Today recovered: 70577 Total recovered: 30697077

```

Fig. 3.28

Other api for practice purpose are:

1. <https://api.coindesk.com/v1/bpi/currentprice.json>
2. <https://www.boredapi.com/api/activity?participants=1>
3. <https://datausa.io/api/data?drilldowns=Nation&measures=Population>

- There also another concepts of dynamic api
- In which user I have written certain codes in which user will input certain data and it will merge with API and it will show the needed data to the user.
- Code:

```

1 import requests
2 #inshorts
3 uin=input("enter the catogary")
4 api="https://inshortsapi.vercel.app/news?category="+uin
5 our=requests.get(api)
6 data=our.json()
7 if (data["success"]) == True:
8     print(data)
9 else:
10     print("invalid input")

```

Fig. 3.29

- Output:

```

D:\PycharmProjects\newproject(2)\letsdoi\Scripts\python.exe "C:/Users/Shubham/OneDrive/Desktop/all about internship/day17(dynamicapi).py"
enter the catogarysports
{'category': 'sports', 'data': [{"author': 'Anmol Sharma', 'content': "India defeated West Indies by 44 runs in the second ODI in Ahmedabad to take an unassailable 2-0 lead in the three-match series. With this, India have now won 11 consecutive ODI series against West Indies. India's ODI series winning streak against West Indies had started in January 2007. The third ODI will take place on Friday.", 'date': '09 Feb 2022,Wednesday', 'imageUrl': 'https://static.inshorts.com/inshorts/images/v1/variants/jpg/m/2022/02_feb/9_wed/img_1644419502620_740.jpg?', 'readMoreUrl': 'https://www.icc-cricket.com/match/3376?utm_campaign=fullarticle&utm_medium=referral&utm_source=inshorts', 'time': '09:29 pm', 'title': 'India clinch their 11th consecutive ODI series vs WI, extend winning run to 15 years', 'url': 'https://www.inshorts.com/en/news/india-clinch-their-11th-consecutive-odi-series-vs-wi-extend-winning-run-to-15-years-1644422382858'}, {"author": "Anmol Sharma", "content": "Team India limited-overs captain Rohit Sharma, who scored 60(51) in the first ODI against West Indies, is now 21 rating points behind second-placed Virat Kohli in the latest rankings for ODI batters. Kohli has 828 rating points, while third-placed Rohit has 807. Pakistan's Fakhar Zaman and England's Joe Root jumped one spot each to be ranked 4th and 10th respectively.", "date": "10 Feb 2022,Wednesday", "imageUrl": "https://static.inshorts.com/inshorts/images/v1/variants/jpg/m/2022/02_feb/10_wed/img_1644422382858.jpg?", "readMoreUrl": "https://www.icc-cricket.com/match/3377?utm_campaign=fullarticle&utm_medium=referral&utm_source=inshorts", "time": "09:29 pm", "title": "Rohit Sharma jumps to 2nd in ODI rankings", "url": "https://www.inshorts.com/en/news/rohit-sharma-jumps-to-2nd-in-odi-rankings-1644422382858"}]

```

Fig. 3.30

- Code :

```

11 import requests
12 uiin=input("enter the pincode")
13 apii="https://api.zippopotam.us/IN/"+uiin
14 ourr=requests.get(apii)
15 datta=ourr.json()
16 a={}
17 if datta==a:
18     print("enter valid input for pin")
19 else:
20     for i in datta["places"]:
21         print(i["place name"])
22

```

Fig. 3.31

- Output

```

enter the pincode360001
Rajkot H 0
Ramnathpara
Race Course Road
Rajputpara
Kalavad Road
Jairaj Plot
Parsi Agiyari Chwok
Mandvi Chowk

```

Fig. 3.32

- As shown as above I have worked on 2 different API
- In which 1<sup>st</sup> is inshorts API and other is postal code API
- In the 1<sup>st</sup> case I have asked user to input the category of the news he/she want to see.
- And by merging it with API I have generated the needed output
- In the 2<sup>nd</sup> case I have made a program in such a way that I am asking user the code of the place if which he/she want to see the post office or which area comes under that specific ZIP code.
- Their respective are as shown as above.
- Task given are as follows:

In my internship there were task assigned to me so following are the task performed by me their respective code and the output are as follows.

#### COUNTRY INFO LOOKUP:

- API: <https://restcountries.com/v3.1/name/india?fullText=true>
  - Take country name from user
  - Display name of Country, Capital Of Country, Currency of Country and Symbol of County
- Code:

```

1 import requests
2 uin=input("enter the country name")
3 newurl="https://restcountries.com/v3.1/name/"+uin+"?fullText=true"
4 aurl=requests.get(newurl)
5 adata=aurl.json()
6 a=[]
7 c=[]
8 d=[]
9 for i in adata:
10     for j in i["currencies"]:
11         c=j
12     for k in i["capital"]:
13         d=k
14     for l in i["altSpellings"]:
15         a.append(l)
16 print("Country Name is:",uin,",Country Currencies is:",c,",Country Capital is:",d,",Country symbol is:",a[0])
17

```

Fig. 3.33

- Output:

```

enter the country nameindia
Country Name is: india ,Country Currencies is: INR ,Country Capital is: New Delhi ,Country symbol is: IN

```

Fig. 3.34

#### 2) DEBIT / CREDIT CARD LOOKUP: -

ALLOW USER TO ENTER FIRST 8 DIGITS OF DEBIT / CREDIT CARD.

<https://lookup.binlist.net/xxxxxxxx>

- Display type of Card: Debit / Credit
- Display Name of company: MasterCard / Visa etc.
- Display country of card: India



Fig. 3.36

- Code:

```

1 import requests
2
3 vin=input("enter first 8 digits of your respective words")
4 newurl="https://lookup.binlist.net/"+vin
5 aurl=requests.get(newurl)
6 adata=aurl.json()
7
8 print("This card belongs to:",adata["country"]["name"])
9 print("Its a:",adata["type"],"Card")
10 print("It Belongs To:",adata["scheme"],"Category")
11

```

Fig. 3.37

- Output:

```

enter first 8 digits of your respective words45441500
This card belongs to: Singapore
Its a: credit Card
It Belongs To: visa Category

```

Fig. 3.38

### 3) DICTIONARY LOOKUP: -

Allow user to insert a word, suppose user have input word: what

- Display all meanings of that word.

API -- <https://api.dictionaryapi.dev/api/v2/entries/en/what>

- Code:

```
1 import requests
2 q=0
3 uin=input("insert a word")
4 newurl="https://api.dictionaryapi.dev/api/v2/entries/en/"+uin
5 arl=requests.get(newurl)
6 adata=arl.json()
7 for i in adata:
8     for j in (i["meanings"]):
9         for k in j["definitions"]:
10            q=q+1
11            print(q,":",k["definition"])
12
```

Fig. 3.39

- Output

```
insert a wordhow
1 : in what way or manner; by what means.
2 : used to ask about the condition or quality of something.
3 : used to ask about the extent or degree of something.
4 : the way in which; that.
5 : a greeting attributed to North American Indians (used in humorous imitation).
```

Fig. 3.40

## 4. Object Oriented Programming (OOP)

### 4.1 Explanation:-

- Basically OOP concept involves :
  1. Class
  2. Object
  3. Inheritance
- Class are different than function although it also definition but the function are callable whereas the class are not callable
- To call the class we need to create an object through which we can call the class and use its properties
- Class has certain variables, which are called class properties.
- Class properties can be changed by using some coding which will be shown below code.
- Each and every class has some methods which are defined in class itself
- We can access the method by using dot(.) after class object name.
- In every method there is “self” written inside its space variables
- Class methods must have an extra first parameter in the method definition. We do not give a value for this parameter when we call the method, Python provides it
- If we have a method that takes no arguments, then we still have to have one argument.
- We can also access the properties if the but first we need to create object of the class than “objectname.properties” by this we can access the parameters
- The class takes no argument so that we can class n number of times and also we should not write the print statement inside the class as we want to access that n number times
- Code regarding to all this explanation is given below
- Code:

```

1 i=40
2 j=50
3 class myclass():
4     a=10
5     b=20
6     def mymethod(self,a,b):
7         print(self.a+self.b)
8 myobject=myclass
9 newobject=myclass
10 print(newobject.a)      #1
11 print(myobject.a)       #2
12 print(newobject.b)      #3
13 print(myobject.b)       #4
14 newobject.mymethod(10,20)

```

Fig. 4.1

- Output:

```

10
10
20
20

Traceback (most recent call last):
  File "C:\Users\Shubham\OneDrive\Desktop\all about internship\day14(1).py", line 14, in <module>
    newobject.mymethod(10,20)
TypeError: myclass.mymethod() missing 1 required positional argument: 'b'

```

Fig. 4.2

- As it is shown in the and as I have explained we can't write print function inside the definition of method that is inside the class because we are writing the class writing the class with arguments that's reason it is having all this kind of problems
- If we not write the class a without argument than there will be no issue like this all the class we will see further are without argument only. So at there will be no issue like this.
- As shown in the i and j are global variables and a and b are class arguments which are also called as class properties.
- Mymethod is the method defined inside the class
- As we can see there self is also written with other two arguments
- As said class is not taking any arguments
- Myobject and newobject are the two object formed to access the class
- As said, earlier by the use of object here is printing the class properties using (.).
- Now following is the study of importance of self
- By using the self, we can access the properties inside the definition of method.
- Code:

```

1  class myclass:
2      a=10
3      b=20
4      def mymethod(self):
5          print("print this")
6      def addnos(self):
7          print(self.a+self.b)
8      def newadd(self,i,j):
9          print(i+j)
10     newobject=myclass()
11     newobject.mymethod()
12     newobject.addnos()
13     newobject.newadd(100,100)

```

Fig. 4.3

- Output:

```
print this
30
200
```

Fig. 4.4

- Moreover worked more OOP in which I specially worked on the constructor of the class, which is used to change the properties of the class.
- The constructor and the class has always the same name.
- Code:

```
1  class one:
2      a=10
3      b=20
4      def __init__(self,a,b):
5          self.a=a
6          self.b=b
7      def addn(self):
8          print(self.a+self.b)
9      objone=one(30,20)
10     objone.addn()
11     objone=one(23,13)
12     objone.addn()
```

Fig. 4.5

- Output:

```
50
36
```

Fig. 4.6

- As shown here “`__init__()`” is the constructor of class
- Actually it is used for the initialization of the
- At the place of the print we can also use return in the definition but by using we need write print when we are accessing the properties to print them.
- Code:

```
1 class one:
2     a=10
3     b=20
4     def __init__(self,a,b):
5         self.a=a
6         self.b=b
7     def addn(self,i,j):
8         return i+j
9     def subn(self):
10        return self.a-self.b
11 objone=one(100,40)
12 c=objone.addn(20,30)+objone.subn()
13 print(c)
14 print(objone.a+objone.b)
```

Fig. 4.7

- Output:

```
110
140
```

Fig. 4.8

- As shown by using the constructor the class properties are changed.
- Furthermore I worked on the inheritance of the two class
- So basically what is inheritance,
- So there are two classes namely parent and child so here we inheriting the parent properties inside the child
- Therefore, the child can also access the parent class properties it will more clearly after seeing the code regarding to it.
- Code:

```

15     class parent:
16         a=10
17         b=20
18         def __init__(self,a,b):
19             self.a=a
20             self.b=b
21         def parentmethod(self):
22             print("parent method")
23     class child(parent):
24         c=30
25         d=50
26         def __init__(self, a, b):
27             self.c = a
28             self.d = b
29         def childmethod(self):
30             print("child method")
31 parentobj=parent(12,43)
32 childobject=child(34,45)
33 print(childobject.a)
34 print(childobject.b)
35 print(childobject.c)
36 print(childobject.d)
37 childobject.childmethod()
38 childobject.parentmethod()

```

Fig. 4.9

- Output:

```

10
20
34
45
child method
parent method

```

Fig. 4.10

- As we can see the inheritance by child through which it can access the properties of the parent class also
- But we can't use in both the class as shown it will create the problem and it will overwrite the value and the value of the child is only changed not the parent as shown in the output(at line 26).
- Therefore, by using this, we have to create only one object for child and by using it; we can access all the properties and method of the parent too.

Code on class:

```
1  class one:
2      a=10
3      b=20
4      c=30
5      def __init__(self,a,b,c):
6          self.a=a
7          self.b=b
8          self.c=c
9      def sub(self,i,j):
10         print(i+j)
11     def mul(self,i,j):
12         print(i*j)
13
14 obj1=one(30,20,40)
15 obj1.sub(20,10)
16 obj1.mul(30,20)
```

Fig. 4.11

Output:

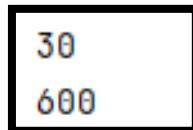


Fig. 4.12

- Now I have written some code on creating the inheritance without writing in this manner “child(parent)” basically crating the inheritance without using actual inheritance.
- So regarding to that the code is as shown as below:

```

1 class newclass:
2     def __init__(self):
3         self.msg="hello"
4     def msgdesplay(self):
5         print(self.msg)
6     newobj=newclass()
7     newobj.msgdesplay()
8 class parent:
9     a=10
10    b=20
11    def parentmethod(self):
12        print("parent method")
13 class child:
14    c=30
15    d=50
16    def childmethod(self):
17        print("child method")
18    parentobject=parent()
19    childobject=child()
20    print(childobject.parentobject.a)
21    print(childobject.parentobject.b)
22    print(childobject.c)
23    print(childobject.d)
24    childobject.childmethod()
25    childobject.parentobject.parentmethod()

```

Fig. 4.13

- Output:

```

hello
10
20
30
50
child method
parent method

```

Fig. 4.14

- After than there are two things which can create issue in class
  1. Method overriding
  2. Method overloading

- 1) Code showing the method overriding:

```

1  class parent:
2      def show(self):
3          print(" hello parent ")
4  class child(parent):
5      def show(self):
6          print(" hello i am child")
7  childobj=child()
8  childobj.show()

1  class parent:
2      def show(self):
3          print(" hello parent ")
4  class child(parent):
5      Overrides method in parent
6          print(" hello i am child")
7  childobj=child()
8  childobj.show()
9

```

The image shows two snippets of Python code. The top snippet defines a parent class with a show() method that prints "hello parent". It then defines a child class that inherits from parent and overrides the show() method to print "hello i am child". A variable childobj is created and its show() method is called. The bottom snippet is similar but includes a tooltip for the overridden method in the child class, stating "Overrides method in parent".

Fig. 4.15

- Output:

hello i am child

Fig. 4.16

- As shown in the above code when same name method is defined in two different class than it fall into method overriding as it also shown in the python code also
- And can be verify in the output also as it is printing according to the recent definition only
- 2)Method overloading:

```

1  class parent:
2      def show(self):
3          print(" hello parent ")
4      def show(self):
5          print(" hello i am child")
6  class child(parent):
7      def mymethod(self):
8          print("hello world")
9  childobj=child()
10 childobj.show()
11

```

The image shows a Python code editor with a class parent containing two methods named show(). The class child inherits from parent and defines its own method mymethod(). The code then creates a child object and calls its show() method. The tooltip for the child's show() method indicates it overrides the parent's method.

Fig. 4.17

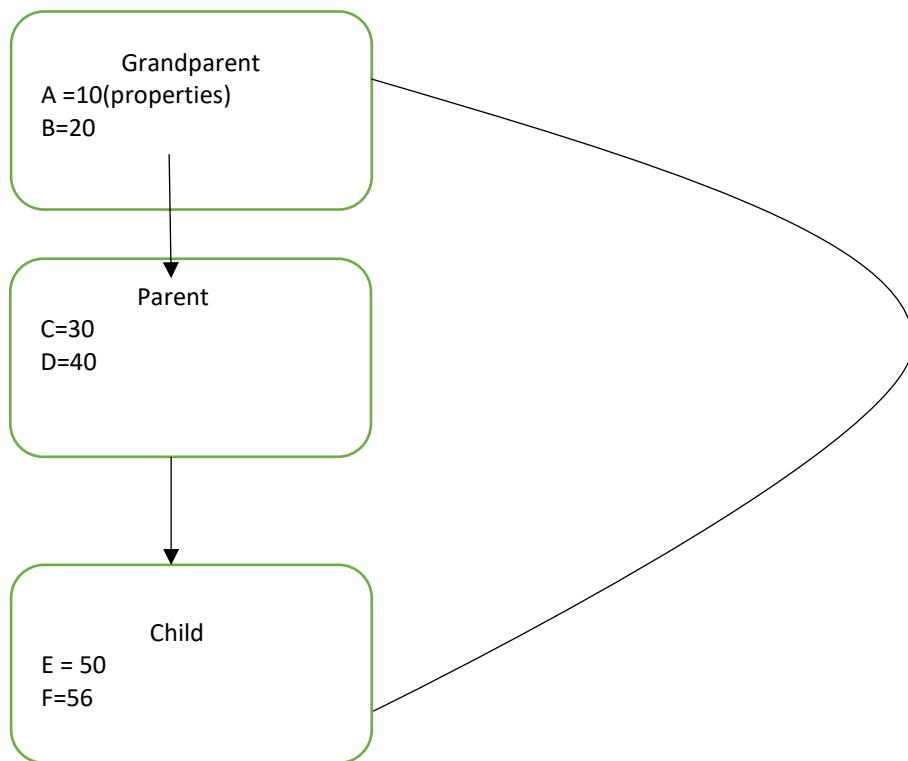
- Output:

hello i am child

Fig. 4.18

- As shown we cannot have same name method in the same class it will led to method overloading and it will print according to recent definition only.

- In other languages this problem doesn't arise as the y can differentiate the methods in three categories which are as follows:
  - 1) Number of arguments are different
  - 2) Data types are different
  - 3) Sequence of data type are different
- As the inheritance is already discussed now its turn of types of Inheritance.
- There are basically two types of inheritances:
  1. Multi-level inheritance
  2. Multiple inheritance
- Multi-level inheritance
- 



- So as shown in the figure is called as multilevel inheritance in which child can access both parent and grandparent methods as well as their respective properties.
- Code:

```
44 class grand:  
45     o=555  
46     p=666  
47     def gm(self):  
48         print("i am grandparent")  
49 class pa(grand):  
50     y=888  
51     x=999  
52     def pam(self):  
53         print("i am parent")  
54 class chi(pa):  
55     l=123  
56     k=234  
57     def chim(self):  
58         print("i am child")  
59 childddobj=chi()  
60 print(childddobj.o)  
61 print(childddobj.p)  
62 print(childddobj.y)  
63 print(childddobj.x)  
64 print(childddobj.l)  
65 print(childddobj.k)  
66 childddobj.gm()  
67 childddobj.pam()  
68 childddobj.chim()
```

Fig. 4.19

- Output:

```
i am child  
555  
666  
888  
999  
123  
234  
i am grandparent  
i am parent  
i am child
```

Fig. 4.20

- Now multiple inheritance:

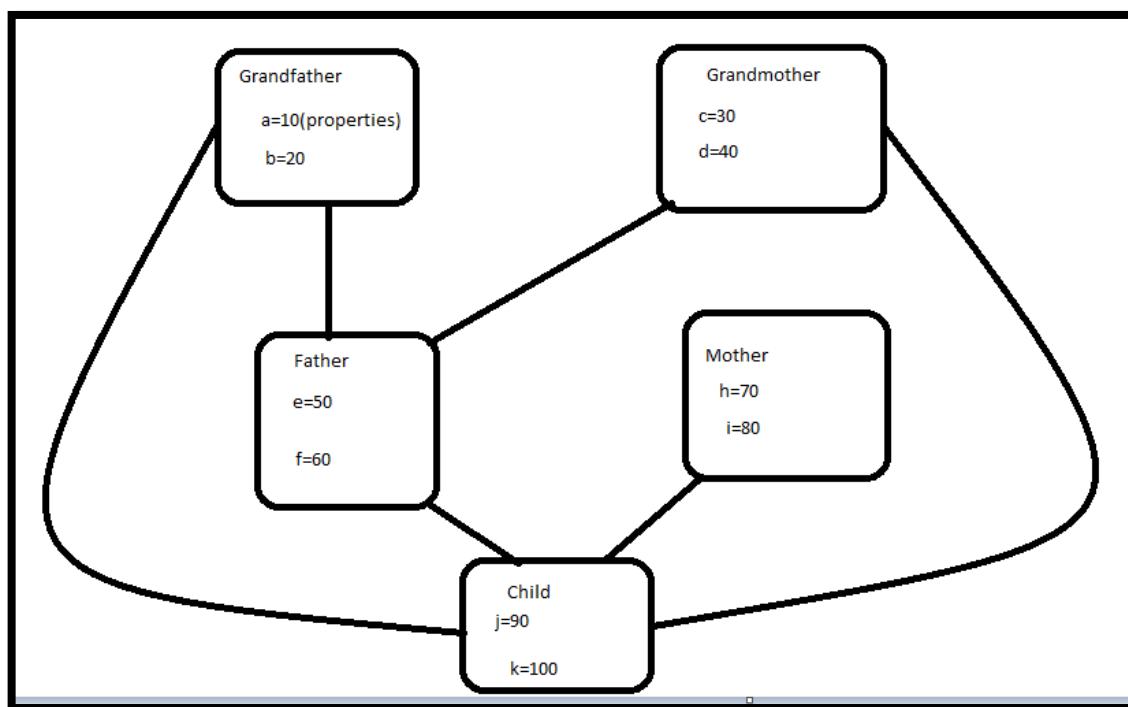


Fig. 4.21

Code:

```

2   class grandf:
3       a=10
4       b=20
5       def grandmethod(self):
6           print("hello i am grand father")
7   class grandm:
8       c=5
9       d=15
10      def grandmothermethod(self):
11          print("hello i am grand mother")
12  class father(grandf,grandm):
13      e=45
14      f=55
15      def fathermethod(self):
16          print("hello i am father")
17  class mother:
18      g=12
19      h=13
20      def mothermethod(self):
21          print("hello i am mother")
22  class child(father,mother):
  
```

Fig. 4.22

```
23     i=56
24     j=78
25     def childmethod(self):
26         print("i am child")
27     childobj=child()
28     print(childobj.a)
29     print(childobj.b)
30     print(childobj.c)
31     print(childobj.d)
32     print(childobj.e)
33     print(childobj.f)
34     print(childobj.g)
35     print(childobj.h)
36     print(childobj.i)
37     print(childobj.j)
38     childobj.grandmothermethod()
39     childobj.grandmethod()
40     childobj.fathermethod()
41     childobj.mothermethod()
42     childobj.childmethod()
```

Fig. 4.23

- Output:

```
20
5
15
45
55
12
13
56
78
hello i am grand mother
hello i am grand father
hello i am father
hello i am mother
i am child
```

Fig. 4.24

- In the multiple inheritance child is properties of both mother and father, grandmother and grandfather.
- And so on as shown in the program.

## 5. Important library

### 5.1 Random

- The random2 library is being used to generate a random things in a random manner
- For an example if we want to generate an OTP for mobile number verification of any email verification we can use this library as there is one methods By using “random2.randint(start, end)”this will generate the OTP which will not once shown we are able to give the for that too.
- There some code and the regarding to the use of this library they are as below:-

```

1 import random2
2 x=random2.random()
3 print(x)
4
5 otp = random2.randint(100000, 999999)
6 print(otp)

```

Fig. 5.1

```

0.5409936849116647
593233

```

Fig. 5.2

- We have assigned a task to enter six numbers from user and compare that to the randomly generated numbers using random2 package and the show the message that how many numbers matched.
- And also show the actual series number generated by the package.
- Code:

```
1 import random2
2 a=[]
3 for i in range(0,5):
4     ln=random2.randint(1,9)
5     a.append(ln)
6 d=[]
7 for j in range(0,5):
8     c=int(input("enter any no."))
9     d.append(c)
10 q=0
11 for g in range(0,5):
12     if d==a:
13         q=q+1
14 print(q,"match found")
15 print("the series was:",a)
```

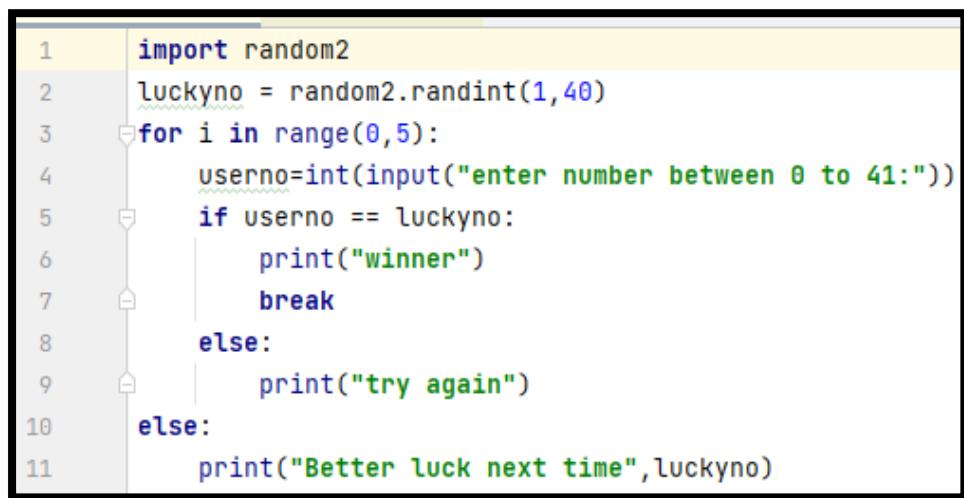
Fig. 5.3

➤ Output:

```
enter any no.1
enter any no.4
enter any no.2
enter any no.6
0 match found
the series was: [1, 4, 8, 5, 4]
```

Fig. 5.4

- One more task also assigned, in which we have generate a number from 0 to 40 and give user five tries to guess the number  
➤ Code:



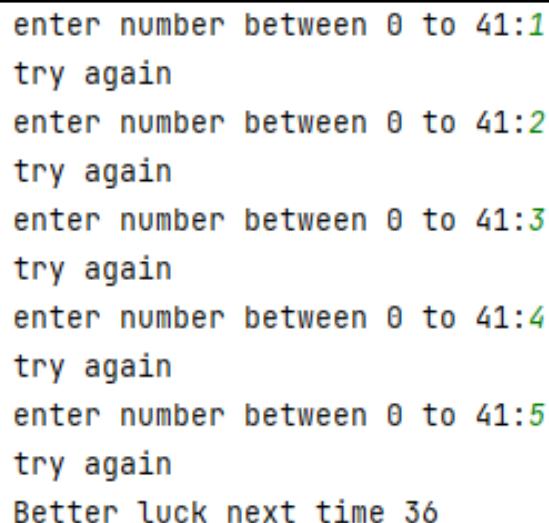
```

1 import random2
2 luckyno = random2.randint(1,40)
3 for i in range(0,5):
4     userno=int(input("enter number between 0 to 41:"))
5     if userno == luckyno:
6         print("winner")
7         break
8     else:
9         print("try again")
10    else:
11        print("Better luck next time",luckyno)

```

Fig. 5.5

➤ Output:



```

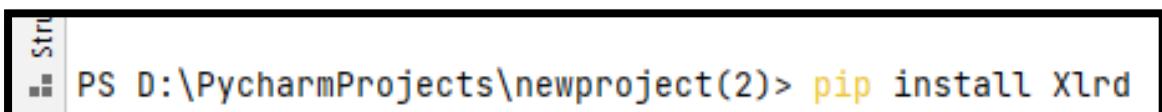
enter number between 0 to 41:1
try again
enter number between 0 to 41:2
try again
enter number between 0 to 41:3
try again
enter number between 0 to 41:4
try again
enter number between 0 to 41:5
try again
Better luck next time 36

```

Fig. 5.6

## 5.2 Xlrd:-

- So basically first we need to install a library called “Xlrd”
- By the writing following code in the terminal panel :



```

PS D:\PycharmProjects\newproject(2)> pip install Xlrd

```

Fig. 5.7

- As shown as below we can access the excel using that code the main thing here is to save the excel in the same folder of the project from this specific code is running
- If not done in this way than this code will show up an error.

- So first we need to give the location of the file and as shown in the below code by that we can access the excel sheet.
- The excel sheet:

	Cricket	Rohit	Kohli	Dhawan
1	1	17	70	10
2	2	81	12	7
3	3	25	55	23
4	4	34	36	37
5	5	45	49	101
6				

Fig. 5.8

- Code:

```

1 import xlrd
2
3 loc="mydata.xlsx"
4 wb=xlrd.open_workbook(loc)
5 sheet=wb.sheet_by_index(0)
6 print(sheet.cell_value(0,2))
7 print("no of rows are:",sheet.nrows)
8 print("no of column are:",sheet.ncols)
9

```

Fig. 5.9

- Output:

```

KOHLI
no of rows are: 6
no of column are: 4

```

Fig. 5.10

- Now if we want all rows only or we need some specific rows only than it is possible by the following code.
- Code:

```

10 for i in range(0,4):
11     print(sheet.cell_value(0,i))
12

```

Fig. 5.11

- Output:

CRICKET  
ROHIT  
KOHLI  
DHAWAN

Fig. 5.12

- In addition, we are able to access limited number of rows only than it can be done as follows.
- Code:

```
13     for j in range(1,4):
14         print(sheet.cell_value(0,j))
15
```

Fig. 5.13

- Output:

ROHIT  
KOHLI  
DHAWAN

Fig. 5.14

- Now as we have done with the rows we also can do with column and can access them specifically or for specific number or can access all column also.
- Code:

```
16     for g in range(0,5):
17         print(sheet.cell_value(g,1))
```

Fig. 5.15

- Output:

ROHIT  
17.0  
81.0  
25.0  
34.0

Fig. 5.16

- Code:

```

20     for i in range(1,sheet.nrows):
21         print(sheet.cell_value(i,2))
22     print(sheet.row_values(0))
23     print(sheet.col_values(0))
24

```

Fig. 5.17

- Output:

CRICKET  
ROHIT  
KOHLI  
DHAWAN

Fig. 5.18

- As the need of further program if we want this rows or column in the form of list than it can is possible by the following code.
- Code:

```

22     print(sheet.row_values(0))
23     print(sheet.col_values(0))
24

```

Fig. 5.19

- Output:

```

['CRICKET', 'ROHIT', 'KOHLI', 'DHAWAN']
['CRICKET', 1.0, 2.0, 3.0, 4.0, 5.0]

```

Fig. 5.20

- Task of the day:
- Here it is said to print the player score according to the match.
- The code is shown as follows:
- Code:

```

for i in range(1,sheet.nrows):
    print("MATCH:",i)
    for j in range(1,sheet.ncols):
        print("score of",sheet.cell_value(0,j),"is",sheet.cell_value(i,j))

```

Fig. 5.21

- Output:

```

MATCH: 1
score of ROHIT is 17.0
score of KOHLI is 70.0
score of DHAWAN is 10.0
MATCH: 2
score of ROHIT is 81.0
score of KOHLI is 12.0
score of DHAWAN is 7.0
MATCH: 3
score of ROHIT is 25.0
score of KOHLI is 55.0
score of DHAWAN is 23.0
MATCH: 4
score of ROHIT is 34.0
score of KOHLI is 36.0
score of DHAWAN is 37.0
MATCH: 5
score of ROHIT is 45.0
score of KOHLI is 49.0
score of DHAWAN is 101.0

```

Fig. 5.23

- Another Task of the day:
- To get the name from the user and show in the excel sheet is that player is in the sheet or if it is there than print “player found” other than that print “player not found”.
- Code:

```

1 import xlrd
2 vin=input("enter the player name:")
3 o=vin.lower()
4 loc="mydata.xlsx"
5 wb=xlrd.open_workbook(loc)
6 sheet=wb.sheet_by_index(0)
7 for i in range(1,sheet.ncols):
8     k=(sheet.cell_value(0,i))
9     if k==o:
10         print("Player found")
11         break
12     else:
13         print("Player not found")

```

Fig. 5.24

Output:

```
enter the player name:dhawan
Player found
enter the player name:shubham
Player not found
```

Fig. 5.25

- Excel for the task:

CRICKET	ROHIT	KOHLI	DHAWAN
1	17	70	10
2	81	12	7
3	25	55	23
4	34	36	37
5	45	49	101

- Task day:
- First Task:
  1. Print all scores of DHAWAN
  2. Print score of all players of third match.
- Code:

```
1 import xlrd
2 loc="mydata.xlsx"
3 wb=xlrd.open_workbook(loc)
4 sheet=wb.sheet_by_index(0)
5 print("DHAWAN scores are:")
6 for i in range(1,sheet.nrows):
7     print(sheet.cell_value(i,3))
8 print("Scores of 3rd match of all players:")
9 for i in range(1, sheet.ncols):
10    print(sheet.cell_value(3,i))
```

Fig. 5.26

Output:

```

DHAWAN scores are:
10.0
7.0
23.0
37.0
101.0
Scores of 3rd match of all players:
25.0
55.0
23.0

```

Fig. 5.27

- Second task:
  1. Allow user to enter match number, Print score of all players in that match
- Code:

```

import xlrd
loc="mydata.xlsx"
wb=xlrd.open_workbook(loc)
sheet=wb.sheet_by_index(0)
uin=int(input("Enter a Match Number:"))
for i in range(1, sheet.ncols):
    print(sheet.cell_value(uin,i))

```

Fig. 5.28

- Output:

```

Enter a Match Number:2
81.0
12.0
7.0

```

Fig. 5.29

- Third task:
- Print score of all players in following format. (Recent match score first)

Match 5:

Score of ROHIT 45

Score of KOHLI 49

Score of DHAWAN 101

Match 4:

....  
....  
....  
....

- Code:

```
import xlrd
loc="mydata.xlsx"
wb=xlrd.open_workbook(loc)
sheet=wb.sheet_by_index(0)
c=[]
for k in range(1,sheet.nrows):
    c.append(k)
d=c[::-1]
for i in d:
    print("Match",i)
    for j in range(1,sheet.ncols):
        print("score of:",sheet.cell_value(0,j),"is",sheet.cell_value(i,j))
```

Fig. 5.30

- Output:

```
Match 5
score of: ROHIT is 45.0
score of: KOHLI is 49.0
score of: DHAWAN is 101.0
Match 4
score of: ROHIT is 34.0
score of: KOHLI is 36.0
score of: DHAWAN is 37.0
Match 3
score of: ROHIT is 25.0
score of: KOHLI is 55.0
score of: DHAWAN is 23.0
Match 2
score of: ROHIT is 81.0
score of: KOHLI is 12.0
score of: DHAWAN is 7.0
Match 1
score of: ROHIT is 17.0
score of: KOHLI is 70.0
score of: DHAWAN is 10.0
```

Fig. 5.31

Fourth task:

- Print total score of ROHIT
- Print Highest score of KOHLI
- Print Lowest score of DHAWAN
- Print Total score of all players in five Match.

Total of ROHIT:

Total of KOHLI:

Total of DHAWAN:

- Code:

```
1 import xlrd
2 loc="mydata.xlsx"
3 wb=xlrd.open_workbook(loc)
4 sheet=wb.sheet_by_index(0)
5 a=0
6 c=[]
7 b=[]
8 for i in range(1,sheet.nrows):
9     k=sheet.cell_value(i,1)
10    a=a+k
11    l=sheet.cell_value(i,2)
12    c.append(l)
13    j=sheet.cell_value(i,3)
14    b.append(j)
15    print("Total score of rohit is:",a)
16    print("Highest score of KOHLI is:",max(c))
17    print("Lowest score of DHAWAN is",min(b))
18    print("Total score of rohit is:",a)
19    print("total score of kohli is:",sum(c))
20    print("total score of dhawan is:",sum(b))
```

Fig. 5.32

- Output:

```
Total score of rohit is: 202.0
Highest score of KOHLI is: 70.0
Lowest score of DHAWAN is 7.0
Total score of rohit is: 202.0
total score of kohli is: 222.0
total score of dhawan is: 178.0
```

Fig. 5.33

### 5.3 datetime:-

- The datetime needs install first from the terminal and than we can access that and get the need full format of the time week date any thing
- The description of all types of format is inside the following table:-

Directive	Meaning	Example	Notes
%a	Weekday as locale's abbreviated name.	Sun, Mon, ..., Sat	(1)
%A	Weekday as locale's full name.	Sonntag, Sunday, Monday, ..., Saturday	(1)
%w	Weekday as a decimal number, where 0 is Sunday and 6 is Saturday.	0, 1, ..., 6	
%d	Day of the month as a zero-padded decimal number.	01, 02, ..., 31	(9)
%b	Month as locale's abbreviated name.	Jan, Feb, ..., Dec	(1)
%B	Month as locale's full name.	January, February, ..., December	(1)
%m	Month as a zero-padded decimal number.	01, 02, ..., 12	(9)
%Y	Year without century as a zero-padded decimal number.	00, 01, ..., 99	(9)
%Y	Year with century as a decimal number.	0001, 0002, ..., 2013, 2014, ..., 9998, 9999	(2)
%H	Hour (24-hour clock) as a zero-padded decimal number.	00, 01, ..., 23	(9)

Directive	Meaning	Example	Notes
%I	Hour (12-hour clock) as a zero-padded decimal number.	01, 02, ..., 12	(9)
%p	Locale's equivalent of either AM or PM.	AM, PM	(1), (3)
%M	Minute as a zero-padded decimal number.	00, 01, ..., 59	(9)
%S	Second as a zero-padded decimal number.	00, 01, ..., 59	(4), (9)
%f	Microsecond as a decimal number, zero-padded to 6 digits.	000000, 000001, ..., 999999	(5)
%z	UTC offset in the form <code>±HHMM[SS[.ffffff]]</code> (empty string if the object is naive).	(empty), +0000, -0400, +1030, +063415, -030712.345216	(6)
%Z	Time zone name (empty string if the object is naive).	(empty), UTC, GMT	(6)
%j	Day of the year as a zero-padded decimal number.	001, 002, ..., 366	(9)
%U	Week number of the year (Sunday as the first day of the week) as a zero-padded decimal number. All days in a new year preceding the first Sunday are considered to be in week 0.	00, 01, ..., 53	(7), (9)
%W	Week number of the year (Monday as the first day of the week) as a zero-padded decimal number. All days in a new year preceding the first Monday are considered to be in week 0.	00, 01, ..., 53	(7), (9)
%C	Locale's appropriate date and time representation.	Tue Aug 16 21:30:00 1988	(1)
%x	Locale's appropriate date representation.	08/16/88 (None); 08/16/1988	(1)
%X	Locale's appropriate time representation.	21:30:00	(1)

- The table above shown are the variable thought which we can have date, time, number of week, current number of week and many more things.
- Code:

```
import datetime

dt=datetime.datetime.now()
print(dt)
```

Fig. 5.34

- Output:

```
2022-02-10 09:28:01.465015
```

Fig. 5.35

- Code:

7	print(dt.minute)
8	print(dt.hour)
9	print(dt.second)
10	print(dt.year)
11	print(dt.month)
12	print(dt.day)

Fig. 5.36

- Output:

28
9
1
2022
2
10

Fig. 5.37

- Code:

```
o1=dt.strftime("%a") #Short Day
print(o1)
```

Fig. 5.38

- Output:

Thu
-----

Fig. 5.39

- Code:

17	o2=dt.strftime("%A") # full name
18	print(o2)

Thursday
----------

Fig. 5.40  
Government Engineering College, Gandhinagar

- Code:

```
19      o3=dt.strftime("%b")
20      o4=dt.strftime("%B")
21      o5=dt.strftime("%Y")
22      o6=dt.strftime("%y")
23      print(o3)
24      print(o4)
25      .
26      print(o5)
27      print(o6)
28
```

Feb  
February  
2022  
22

Fig. 5.41

- Task of the day: (duration:2hr)

```
# every day 2 to 4 PM show msg : happy hours
# every day 10 to 11 AM show msg : 50% OFF
# Every Wednesday and Friday : 50 % off
# Every Thursday evening 6 to 8 : FLASH SALE
# Count down to 12 AM. It should be display like below format
# 6 hours 34 minutes and 25 seconds |
```

Fig. 5.42

- Code:

```

import datetime
dt = datetime.datetime.now()
a = int(dt.strftime("%H"))
b = int(dt.strftime("%w"))
c = int(dt.strftime("%M"))
d = int(dt.strftime("%S"))
f=23
g=59
h=59
if a == 14 or a == 15 or a == 16:
    print("happy hour")
if a == 10 or a == 11:
    print("50% off")
if b == 2:
    print("50% off")
if b == 4:
    if a == 18 or a == 19 or a == 20:
        print("flash sale")
print(f-int(a),"Hours",g-int(c),"minutes",h-int(d),"Seconds Remaining")

```

Fig. 5.43

#### 5.4 Numpy:-

- The numpy is very much useful in doing mathematical operation, statistical operation, data analysis and data cleaning.
- Code regarding to some of its application and theory of it.
- Code and Output:

1	<code>import numpy as np</code>
2	<code>a=np.array([1,2,3])</code>
3	<code>print(a)</code>
4	<code>print(a.shape)</code>
	<code>[1 2 3]</code>
	<code>(3, )</code>

Fig. 5.44

- This way we can form the array regarding to the numpy in which it has the application.
- Shape gave its row and column matrix.
- Code and Output:

```
b=np.array([[1, 2, 3], [4, 5, 6]])
print(b)
print(b.shape)
```

```
[[1 2 3]
 [4 5 6]]
(2, 3)
```

Fig. 5.45

- Therefore, as we can see here it will give 2 rows and 3 column.
- If we want to print specific row or column we can do that by the following way
- Code and Output:

```
x=b.reshape(3,2)
print(x)
print(x[0,0])
```

```
[[1 2]
 [3 4]
 [5 6]]
1
```

Fig. 5.46

- The same way using numpy it is able to do the average of the given set of the values, it can find the maximum value, minimum value and other application of numpy is applicable. Code regarding to that is as follows.
- Code:

```
19 o=np.array([[1,2],[3,4]])
20 h=np.array([[1,2],[3,4]])
21 k=o+h
22 print(k)
23 l=o-h
24 print(l)
25 n=np.dot(o,h)
26 print(n)
27 print(np.min(o))
28 print(np.max(o))
29 print(np.average(o))
```

Fig. 5.47

- Output:

```
[[2 4]
 [6 8]]
 [[0 0]
 [0 0]]
 [[ 7 10]
 [15 22]]
 1
 4
 2.5
```

Fig. 5.48

- Task:

Code: In this specific task, it is asked to get the player name from the user and print its total run scored, maximum run scored, minimum run scored, average run scored and number of matches played.

The code regarding to this is shown as below:

- Code:

```
1 import xlrd
2 import numpy as np
3 loc="mydata.xlsx"
4 wb=xlrd.open_workbook(loc)
5 sheet=wb.sheet_by_index(0)
6 uin=input("Enter the player name:")
7 o=uin.upper()
8 c=[]
9 for i in range(1,sheet.ncols):
10     k=(sheet.cell_value(0,i))
11     if k==0:
12         print("Highest run scored:",np.sum(sheet.col_values(i,1)))
13         print("Maximum run scored:",np.max(sheet.col_values(i,1)))
14         print("Minimum run scored:",np.min(sheet.col_values(i,1)))
15         print("Average run:",np.average(sheet.col_values(i,1)))
16         print("No. of matches played:",len(sheet.col_values(i,1)))
```

```
Enter the player name:rohit
Highest run scored: 202.0
Maximum run scored: 81.0
Minimum run scored: 17.0
Average run: 40.4
No. of matches played: 5
```

Fig. 5.49

- Here it asked to analyses the layers performance from a given match till the given match only
- So the code regarding to that is as shown as below
- Code and Output:

```

import xlrd
import numpy as np
loc="mydata.xlsx"
wb=xlrd.open_workbook(loc)
sheet=wb.sheet_by_index(0)
u=input("Enter from which match you want to start analysing:")
m=input("until which match you want ro analyse:")
v=input("Enter the player name:")
o=vin.upper()
for i in range(0,sheet.ncols):
    k = (sheet.cell_value(0, i))
    if k == o:
        print(np.sum(sheet.col_values(i,int(u),(int(m)+1))))
        break
    else:
        print(np.sum(sheet.col_values(i,int(u),(int(m)+1))))
```

Enter from which match you want to start analysing:2  
until which match you want ro analyse:5  
Enter the player name:rohit  
185.0

Fig. 5.50

- As already discussed numpy is powerful package and here its asked to print the maximum form rows given
- Thus, here I have used the axis parameter in which if we want to access the rows we can write axis=1 and if we want access to the column than we can write axis=0
- The sample code regarding to them are as shown
- Here the process will execute from right to left
- Code and Output:

```

print(np.sum(nparray1[0]))
print(np.max(nparray1, axis=1))
print(np.sum(nparray1, axis=1))
```

40  
[25 23 45 85]  
[ 40 35 79 160]

Fig. 5.51

- Here I have compared that numpy package is very power full as shown below there is a code which commented if the code for the generation of the random 3 $\times$ 3 matrix

- In addition, the non-commented code gives the same by writing little bit of code.
- Code and Output:

```

1 import numpy as np
2
3 a = np.random.randint(0,9,[3,3])
4 print(a)
5
6 #import random2
7 #c=[]
8 #for i in range (0,9):
9 #    a=random2.randint(0,9)
10 #    c.append(a)
11 #d=np.array(c)
12 #b=d.reshape(3,3)
13 #print(b)

```

```

[[6 8 6]
 [5 4 1]
 [0 8 1]]

```

Fig. 5.52

- Task of the day:
- Here it is asked to create a game in which we have generated a 20 × 20 matrix randomly having the figures from 0-9 and let the user input the row value and column value and if that position have 0 or 8 than the user is out other than that the value is added to the score of the user and finally print that.
- Code: (Duration:4hr)

```

1 import numpy as np
2 s=0
3 i=np.random.randint(0,9,[20,20])
4 for j in i:
5     ruin = int(input("enter the rows no:"))
6     cuin = int(input("enter the coloumn no:"))
7     if i[(ruin-1),(cuin-1)] == 0 :
8         print("game over")
9         break
10    elif i[(ruin-1),(cuin-1)] == 8:
11        print("game over")
12        break
13    else:
14        s=s+i[(ruin-1),(cuin-1)]
15        print("Enter again You are winning,Your current score is:",s)
16 print("Your total score is:",s)

```

Fig. 5.53

- There are also some of statistical application, of numpy

- On this I lean about the saving of the file directly from the python code by writing the following code we can save the file and if we access that first we have to load the file to a certain variable and after we are able to do that.
- Here as I had discussed already here we need to load that specific file and after that, only we can see what is there inside it
- Therefore, if we want to directly see what inside that file we can also save the file in the txt form also.
- Code and Output:

```
a=np.array([1,2,3,4])
b=np.array([4,5,6,7])
d=np.savetxt("data.txt",a)
print(np.loadtxt("data.txt"))
print(type(np.loadtxt("data.txt")))

[1. 2. 3. 4.]
<class 'numpy.ndarray'>
```

Fig. 5.54

## 5.5 Pandas

- Pandas is mainly used for the labelling of the data given so once the data was labelled than we can access the data regarding to their respective labels only
- Pandas has three data structures:
  1. Series (1D)
  2. Dataframe (2D)
  3. Panel (3D)
- Here we will discuss more about the series and the Dataframe data structure
- As the name suggests the series data structure will label the data in series manner that is row form it will label each data with row
- Here are some the application of the series data structure with data types like with constant, dictionary, list, tuple.
- Code:

```
import pandas as pd
import numpy as np

mydata = 100
a=pd.Series(mydata)      0    100
print(a)                 dtype: int64
print(type(a))           <class 'pandas.core.series.Series'>
```

Fig. 5.55

```

1 newdata=[10,15,20,25,30,35,40,45]
2 pddata=pd.Series(newdata,index=['even','odd','even','odd','even','odd','even','odd'])
3 print(pddata)
4
5 even    10
6 odd     15
7 even    20
8 odd     25
9 even    30
10 odd    35
11 even   40
12 odd    45
13 dtype: int64

```

Fig. 5.56

- As I have discussed after labelling the data we can easily access them:-
- Code and output:

```

1 25 print(pddata['even'])
2
3 even    10
4 even    20
5 even    30
6 even    40
7 dtype: int64

```

Fig. 5.57

- In addition discussion regarding the Dataframe data structure
- Here the labelling is done in 2-dimensional way that is in the row and the column in both the format
- Code and output:

```

1 1 import pandas as pd
2
3 2 lst = ['hello', 'how', 'are', 'you', 'my', 'frind']
4
5 3 a = pd.DataFrame(lst)
6
7 4 print(a)
8
9 5

```

	0
0	hello
1	how
2	are
3	you
4	my
5	frind

Fig. 5.58

- On this day, I have worked more on the Dataframe data structure of the pandas
- The main thing here is to take that when we want to access the any value in the matrix first we have to write column and that row in Dataframe data-structure mechanism.
- Here are some the application of the Dataframe data structure with data types like with constant, dictionary, and list.
- Code and output:

```

1 import pandas as pd
2 import numpy as np
3
4 a = [15]
5 df1 = pd.DataFrame(a)
6 print(df1)
7
8
9      0
10     0  15

```

Fig. 5.59

- Here if give same key and different value it will overwrite all the other and print the last one entered in the dictionary
- Thus overwriting problem can be seen in the following code.
- Code and Output:

```

1 import pandas as pd
2
3 k={"a":[10,20,30],"b":[40,50,60],"c":[70,80,90],"b":[110,120,130]}
4 df6=pd.DataFrame(k,index=['even','odd','even'])
5 print(df6)
6
7
8      a    b    c
9 even  10  110   70
10 odd   20  120   80
11 even  30  130   90

```

Fig. 5.60

- Furthermore as I have discussed of saving the normal file and txt file
- We can also make directly the excel file from Dataframe data structure by using the `to_excel` same way we can create `to_csv` and other files too.
- Following is the code
- Code:

```

20
21 e={"a":[10,20,30],"b":[40,50,60],"c":[70,80,90]}
22 df5=pd.DataFrame(e)
23 print(df5)
24 print(df5['c'][1])
25 df5.to_excel(r'openfile.xlsx',sheet_name='dataframe.py',index=False)

```

Fig. 5.61

- Here are the more example of the creation of the excel sheets.

- Code and output:

```

f={"SR NO":[1,2,3,4,5],"BRANCH":['CE','CE','IT','IT','CE'], 'NAME':['RAHESH','SURESH','MAHESH','NARESH','JAYESH'],
    'TOTAL':[210,150,225,180,90], 'PERCENTA':[70,50,75,60,30], 'PASSFAIL':[1,1,1,1,0]}
df7 = pd.DataFrame(f)
df7.to_excel(r'RESULT1.xlsx',sheet_name='dataframe1.py',index=False)

L=[{"SR NO":1, "BRANCH":'EC', "NAME":'RATAN', "TOTAL":150, "PERCENTA":50, "PASSFAIL":1},
   {"SR NO":2, "BRANCH":'CE', "NAME":'JATAN', "TOTAL":270, "PERCENTA":90, "PASSFAIL":1},
   {"SR NO":3, "BRANCH":'IT', "NAME":'KATHAN', "TOTAL":285, "PERCENTA":95, "PASSFAIL":1},
   {"SR NO":4, "BRANCH":'EC', "NAME":'NAYAN', "TOTAL":195, "PERCENTA":65, "PASSFAIL":1},
   {"SR NO":5, "BRANCH":'IT', "NAME":'RAMAN', "TOTAL":165, "PERCENTA":55, "PASSFAIL":1}]
df8 = pd.DataFrame(L)
df8.to_excel(r'RESULT2.xlsx',sheet_name='dataframe2.py',index=False)

```

	SR NO	BRANCH	NAME	TOTAL	PERCENTA	PASSFAIL
1	1	CE	RAHESH	210	70	1
2	2	CE	SURESH	150	50	1
3	3	IT	MAHESH	225	75	1
4	4	IT	NARESH	180	60	1
5	5	CE	JAYESH	90	30	0

	SR NO	BRANCH	NAME	TOTAL	PERCENTA	PASSFAIL
1	1	EC	RATAN	150	50	1
2	2	CE	JATAN	270	90	1
3	3	IT	KATHAN	285	95	1
4	4	EC	NAYAN	195	65	1
5	5	IT	RAMAN	165	55	1

Fig. 5.62

- Thus we can create the excel sheets can load the excel sheets to any variable and can read the excel sheet using the read command and by that we can have full access on the excel sheet.
- Code:

```

import pandas as pd
import numpy as np

file1 = ('RESULT1.xlsx')
file2 = ('RESULT2.xlsx')

file1data = pd.read_excel(file1)
print(file1data)
print(type(file1data))
print(np.max(file1data['TOTAL']))
file2data = pd.read_excel(file2)
print(np.max(file2data['TOTAL']))

```

Fig. 5.63

- Output:

SR NO	BRANCH	NAME	TOTAL	PERCENTA	PASSFAIL
0	1	CE RAHESH	210	70	1
1	2	CE SURESH	150	50	1
2	3	IT MAHESH	225	75	1
3	4	IT NARESH	180	60	1
4	5	CE JAYESH	90	30	0

<class 'pandas.core.frame.DataFrame'>

225

285

Fig. 5.64

- By the concat command we can join the two files and directly analyze them from one file only
- Code and output:

```

13
14     alldata = pd.concat([file1data,file2data])
15
16     print(alldata)
17
18     print(alldata['TOTAL'])

```

SR NO	BRANCH	NAME	TOTAL	PERCENTA	PASSFAIL
0	1	CE RAHESH	210	70	1
1	2	CE SURESH	150	50	1
2	3	IT MAHESH	225	75	1
3	4	IT NARESH	180	60	1
4	5	CE JAYESH	90	30	0
0	1	EC RATAN	150	50	1
1	2	CE JATAN	270	90	1
2	3	IT KATHAN	285	95	1
3	4	EC NAYAN	195	65	1
4	5	IT RAMAN	165	55	1

0 210  
1 150  
2 225  
3 180  
4 90  
0 150  
1 270  
2 285  
3 195  
4 165

Name: TOTAL, dtype: int64

Fig. 5.66

## 5.6 Matplotlib:-

- This package is mostly used for the data visualization
- As we know the visualized data is more suitable than tabular data

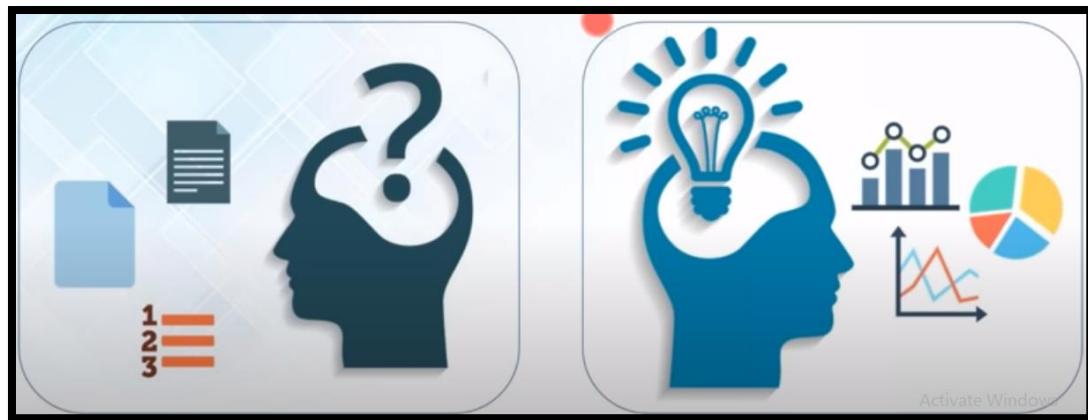


Fig. 5.67

- So for that purpose in python we have the ‘matplotlib’ package which is mainly used for the data visualization through different form of graph like:-
  1. Bar graph
  2. Histogram
  3. Pie chart
  4. Line graph
- The basic code regarding showing the data is as shown as below
- Here its regarding the creation of pie chart
- Now for that purpose we need to import the packages first
- Command for that is ‘from matplotlib import pyplot as plt’
- Here for my convenience I have consider as plt
- Now to get pie chart and other properties of the pie chart as showing the percentage, showing the shadow at percentage part and many more are as shown below.
- Code and output:-

```
from matplotlib import pyplot as plt

branches = ['CE', 'IT', 'CS', 'CIVIL', 'ME', 'EE', 'EC', 'AUTO']
strength = [200, 180, 145, 100, 80, 40, 35, 30]

fig = plt.figure(figsize=(15,10))
plt.pie(strength, labels=branches, autopct='%1.0f%%', shadow=True)
plt.show()
```

Fig. 5.68

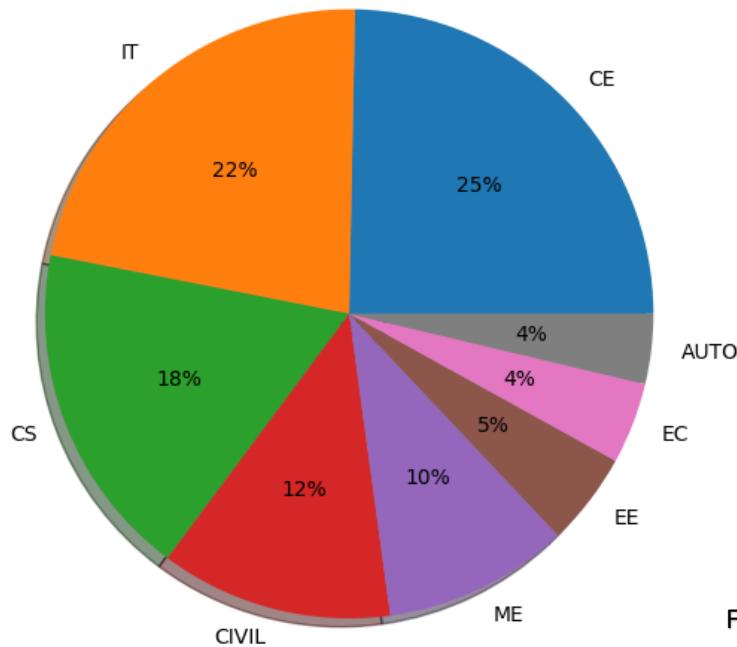


Fig. 5.69

- As shown below I have used its new property of separating the part we want to examine by the command ‘explode’
- The code regarding to that is as shown as below
- Code and output:-

```

1  from matplotlib import pyplot as plt
2
3  branches = ['CE','IT','CS','CIVIL','ME','EE','EC','AUTO']
4  strength = [200,180,145,100,80,40,35,30]
5
6  fig = plt.figure(figsize=(15,10))
7  plt.pie(strength,labels=branches,autopct='%1.0f%%',explode=[0,0,0,0,0,0.2,0,0])
8  plt.show()
9
10
11
12

```

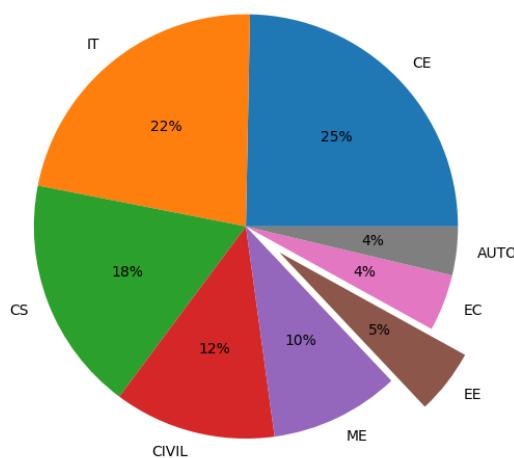


Fig. 5.70

- As discussed for the pie chart I have also worked for the bar graph and its properties
- The below code shown is regarding to the bar graph
- In the bar graph we need to give the:
  - Title.
  - Label to the x-axis that is xlabel.
  - Label to the y-axis that is ylabel.
- Thus, the graph and the code regarding to this is as shown as below.
- Code and Output:-

```

1  from matplotlib import pyplot as plt
2
3  branches = ['CE', 'IT', 'CS', 'CIVIL', 'ME', 'EE', 'EC', 'AUTO']
4  strength = [200, 180, 145, 100, 80, 40, 35, 30]
5
6
7  fig = plt.figure(figsize=(10, 5))
8  plt.bar(branches, strength)
9  plt.xlabel('BRANCHES')
10 plt.ylabel('NUMBER OF STUDENTS')
11 plt.title('BRANCHWISE COLLEGE STRENGTH ANALYSIS')
12 plt.show()

```

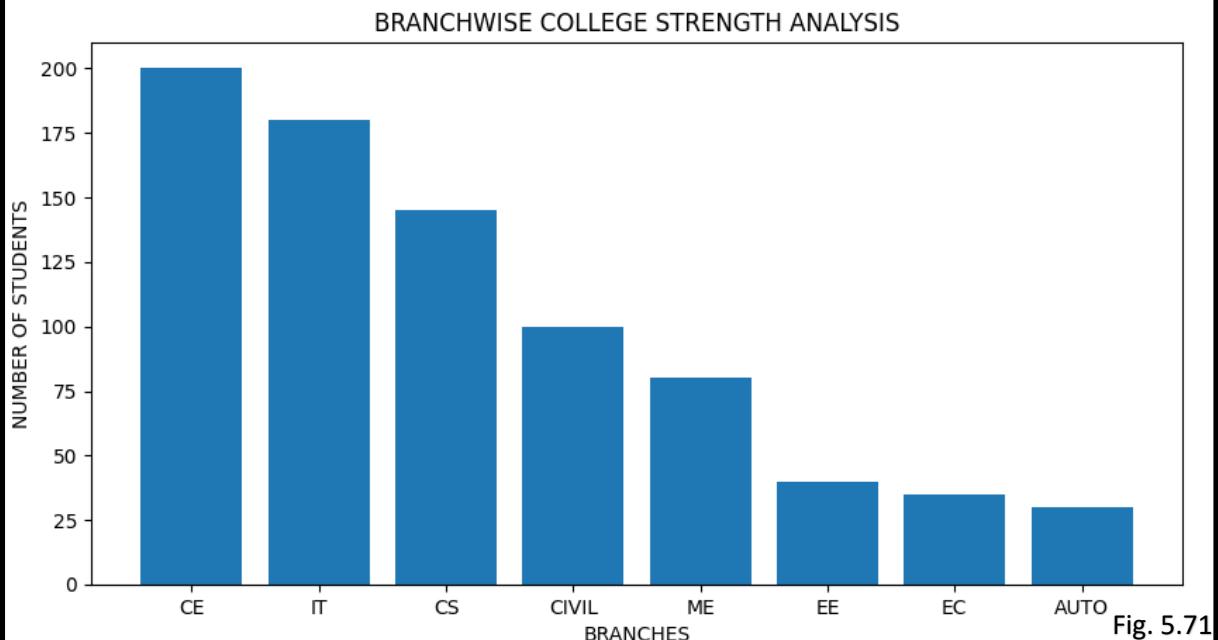


Fig. 5.71

- Moreover there are different properties of the bar graph too
- We can also represent two representation in one graph too
- And for that main we have to add the "legend()" command so that it is use to identify which color graph is for whom.
- And other properties are used as shown in the below code.

- In this I have given certain width and as I am representing 2 things at a time so I have decided to divide that width
- Moreover, to show the scores on the top of the bar the command is given and to fit all this in the same command is given.
- Code:-

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 labels = ['G1', 'G2', 'G3', 'G4', 'G5']
5 men_means = [20, 34, 30, 35, 27]
6 women_means = [25, 32, 34, 20, 25]
7
8 x = np.arange(len(labels)) # the label locations
9 width = 0.35 # the width of the bars
10
11 fig, ax = plt.subplots()
12 rects1 = ax.bar(x - width/2, men_means, width, label='Men')
13 rects2 = ax.bar(x + width/2, women_means, width, label='Women')
14
15 # Add some text for labels, title and custom x-axis tick labels, etc.
16 ax.set_ylabel('Scores')
17 ax.set_title('Scores by group and gender')
18 ax.set_xticks(x, labels)
19 ax.legend()
20
21 ax.bar_label(rects1, padding=3)
22 ax.bar_label(rects2, padding=3)
23
24 fig.tight_layout()
25 plt.show()
```

Fig. 5.72

- Output:-

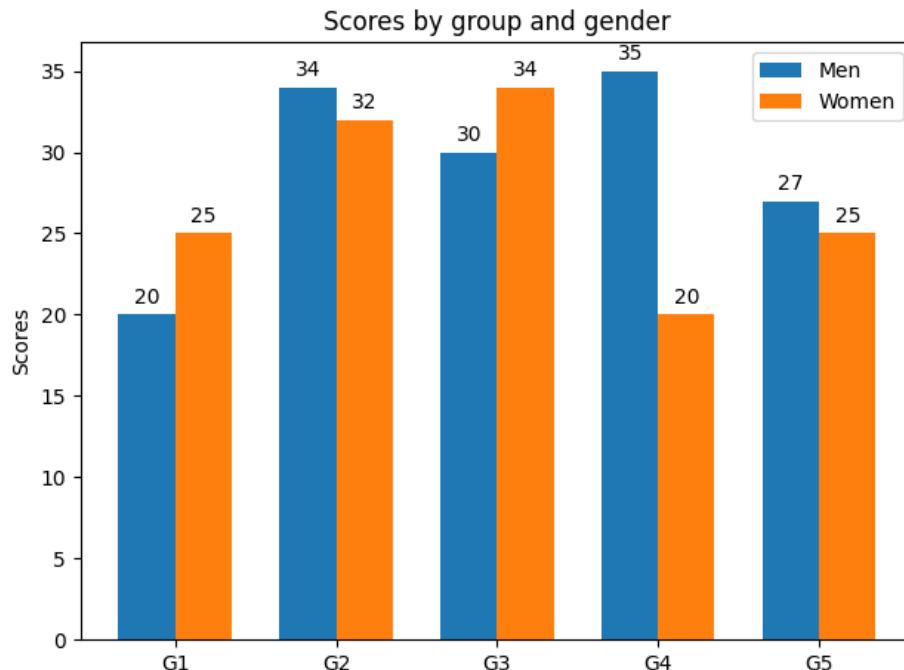


Fig. 5.73

- As I have made the pie chart previously using the data which is given inside the list
- I have made the use of the data which given inside the dictionary type data set too
- In this I have used the keys and the values in the following way for the plotting purpose
- The code and the output regarding to that is shown as below
- Code:- (duration:1hr)

```
from matplotlib import pyplot as plt

mydata = {'CE':200, 'IT':180, 'CS':145, 'CIVIL':100, 'ME':80, 'EE':40, 'EC':35, 'AUTO':30}

fig = plt.figure(figsize=(15,10))
plt.pie(mydata.values(), labels=mydata.keys(), autopct='%1.0f%%', explode=[0,0,0,0.1,0,0,0,0])
plt.show()

fig = plt.figure(figsize=(10,5))
plt.bar(mydata.keys(), mydata.values())
plt.xlabel('BRANCHES')
plt.ylabel('NUMBER OF STUDENTS')
plt.title('BRANCHWISE COLLEGE STRENGTH ANALYSIS')
plt.show()
```

Fig. 5.74

- As I have made the pie chart, it is also possible to make the bar graph by using the dictionary keys and the values.
- The code and the regarding to that are as shown as below.

- Code and Output:-

```
1 from matplotlib import pyplot as plt  
2  
3 mydata ={'CE':200,'IT':180,'CS':145,'CIVIL':100,'ME':80,'EE':40,'EC':35,'AUTO':30}  
4  
5 fig = plt.figure(figsize=(10,5))  
6 plt.bar(mydata.keys(),mydata.values())  
7 plt.xlabel('BRANCHES')  
8 plt.ylabel('NUMBER OF STUDENTS')  
9 plt.title('BRANCHWISE COLLEGE STRENGTH ANALYSIS')  
10 plt.show()
```

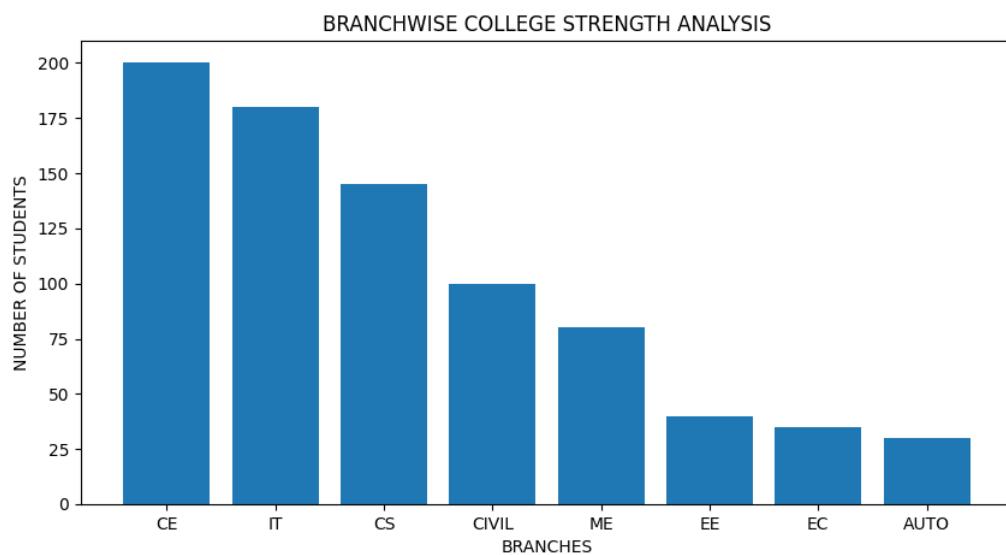


Fig. 5.75

- The line is a the most commonly used in the day to day life and we have seen this type of representations in many places like stock market, cricket, physics and at many more places.
  - Code:-

```
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 xpoints = np.array([0, 6])
5 ypoints = np.array([0, 250])
6
7 plt.plot(xpoints, ypoints)
8 plt.show()
```

Fig. 5.76

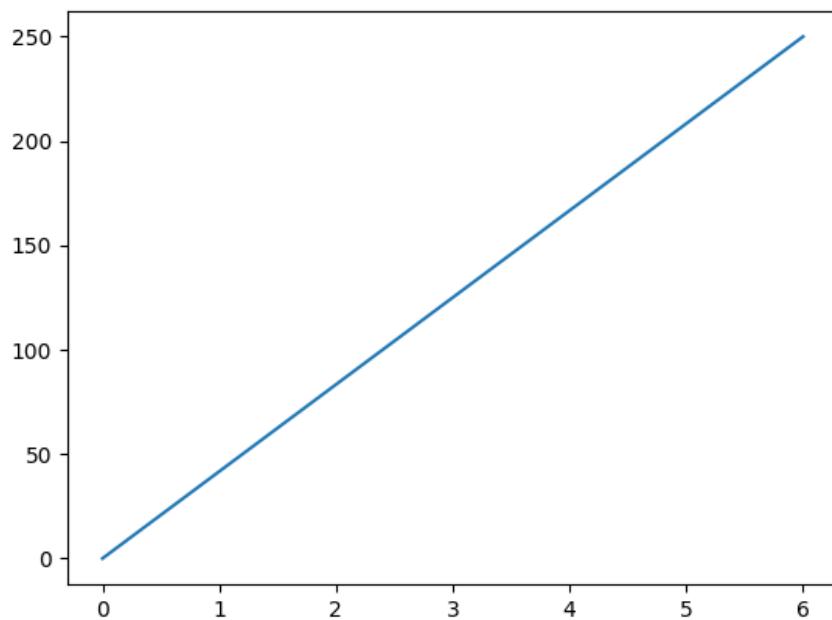


Fig. 5.77

- On this day I worked accessing the excel file by the use of pandas and sorting them according to the need using numpy and finally for the representation I have used the matplotlib package.
- As shown in the below code I have worked on the excel sheet the excel sheet is given below.

CRICKET	ROHIT	KOHLI	DHAWAN
1	17	70	10
2	81	12	7
3	25	55	23
4	34	36	37
5	45	49	101

- As shown in the below code I have worked on accessing the excel by using the reading command of the pandas and arranging them according to the need using the numpy.
- Code:-

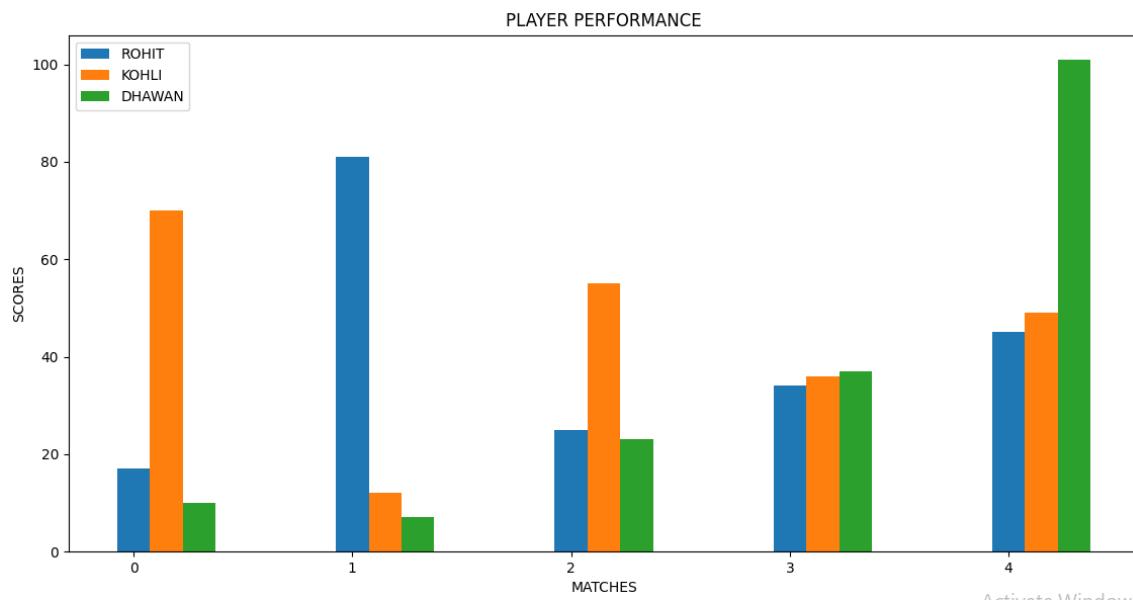
```

1 import pandas as pd
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 file1 = ("mydata.xlsx")
6 filedata = pd.read_excel(file1)
7 matches = list(filedata['CRICKET'])
8 ROHIT = list(filedata['ROHIT'])
9 KOHLI = list(filedata['KOHLI'])
10 DHAWAN = list(filedata['DHAWAN'])
11
12 barwidth = 0.15
13 fig = plt.figure(figsize=(15,10))
14 plt.bar(np.arange(len(matches)), ROHIT, width=barwidth, label="ROHIT")
15 plt.bar(np.arange(len(matches))+0.15, KOHLI, width=barwidth, label="KOHLI")
16 plt.bar(np.arange(len(matches))+0.30, DHAWAN, width=barwidth, label="DHAWAN")
17
18 plt.xlabel("MATCHES")
19 plt.ylabel("SCORES")
20 plt.title("PLAYER PERFORMANCE")
21 plt.legend()
22 plt.show()

```

Fig. 5.78

- Output:-



- As discussed earlier the numpy was used for the arrangement of the data.
- The representation of the data is also possible of some part of the data given that is from excel As shown below I have only represented the specific column only that 'NAME' and 'TOTAL'

- Code:-

```

1 import pandas as pd
2 import numpy as np
3 from matplotlib import pyplot as plt
4
5 f1 = pd.read_excel("RESULT1.xlsx")
6 f2 = pd.read_excel("RESULT2.xlsx")
7
8 alldata = pd.concat([f1,f2])
9
10 NAME = list(alldata['NAME'])
11 TOTAL = list(alldata['TOTAL'])
12
13 barwidth = 0.15
14 fig = plt.figure(figsize=(15,10))
15
16 plt.bar(np.arange(len(NAME)),NAME,width=barwidth,label="NAME")
17 plt.show()

```

- Output:-

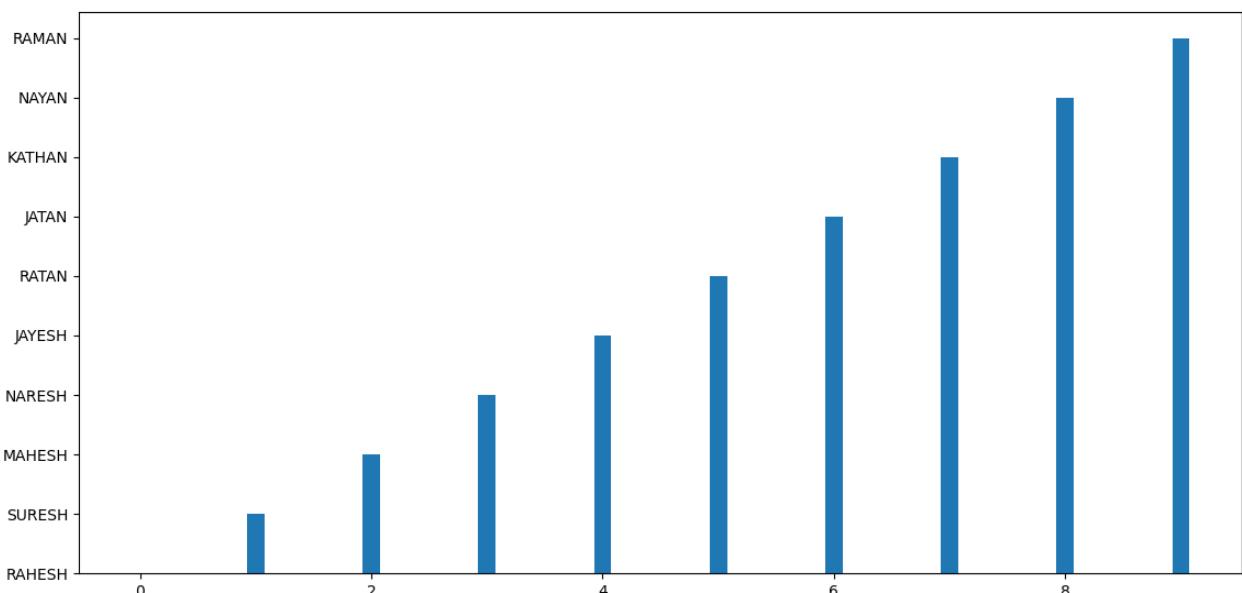


Fig. 5.79

- The following code and output is respected to the total column which specifically represented from the given excel sheet
- So this way we can represent the data specifically also by the use pandas, numpy and matplotlib.
- On this day, I worked on the line graph and its different properties of styling and many more.

- The below code shows the representation of the same excel sheet which is shown as above, the below is regarding to the line graph and its properties are also shown like legend to separate which color is for whom.
- Code:-

```
1  from matplotlib import pyplot as plt  
2  
3  matches = [1,2,3,4,5]  
4  rohit = [25,38,47,55,87]  
5  virat = [10,20,20,25,40]  
6  dhawan = [45,50,55,60,80]  
7  
8  plt.plot(matches,rohit,label = "ROHIT")  
9  plt.plot(matches,virat,label = "VIRAT")  
10 plt.plot(matches,dhawan,label = "DHAWAN")  
11  
12 plt.legend()  
13 plt.show()
```

Fig. 5.80

- Output:-

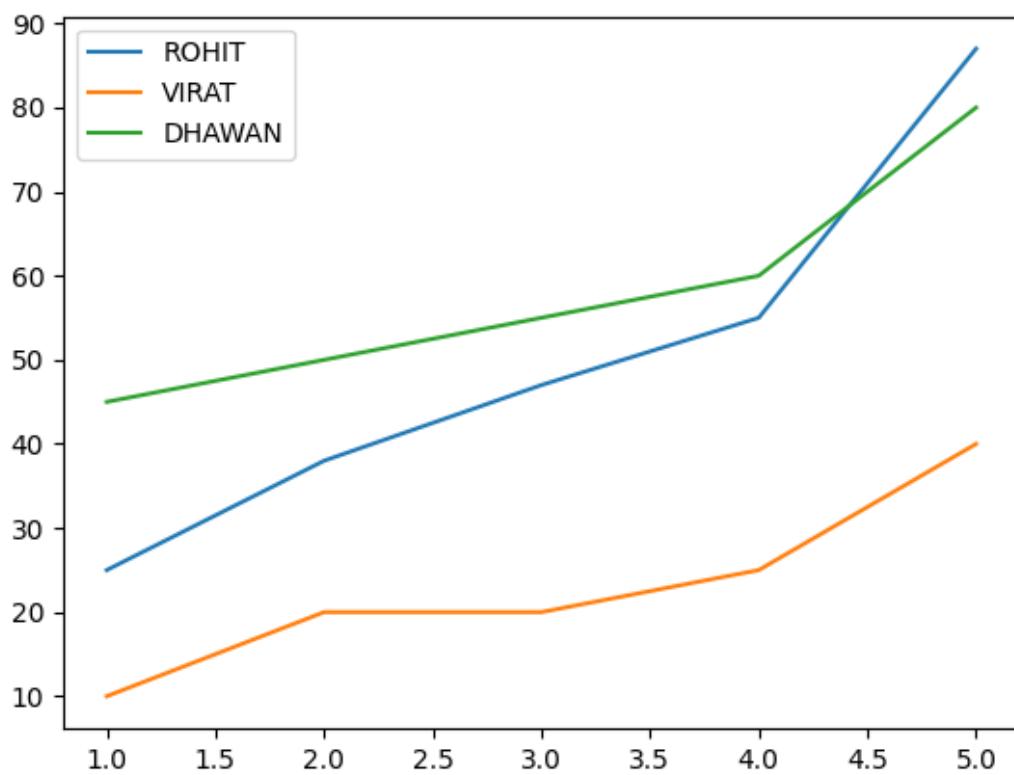


Fig. 5.81

- The below is the application of style in which I can show more than one data representation and by the different color and by using the legend I have noted them which one is for whom
- Code:-

```

1  from matplotlib import pyplot as plt
2  from matplotlib import style as st
3  st.use('ggplot')
4  x=[1,5,7]
5  y=[2,4,6]
6  x2=[12,14,16]
7  y2=[11,13,15]
8  fig = plt.figure(figsize=(10,5))
9  plt.plot(x,y,"g",label='FirstLine',linewidth=5)
10 plt.plot(x2,y2,'b',label='SecondLine',linewidth=5)
11 plt.title('New property')
12 plt.xlabel('X-axis')
13 plt.ylabel('Y-axis')
14 plt.legend()
15 plt.grid(True,color='k')
16 plt.show()

```

- Output:-

Fig. 5.82

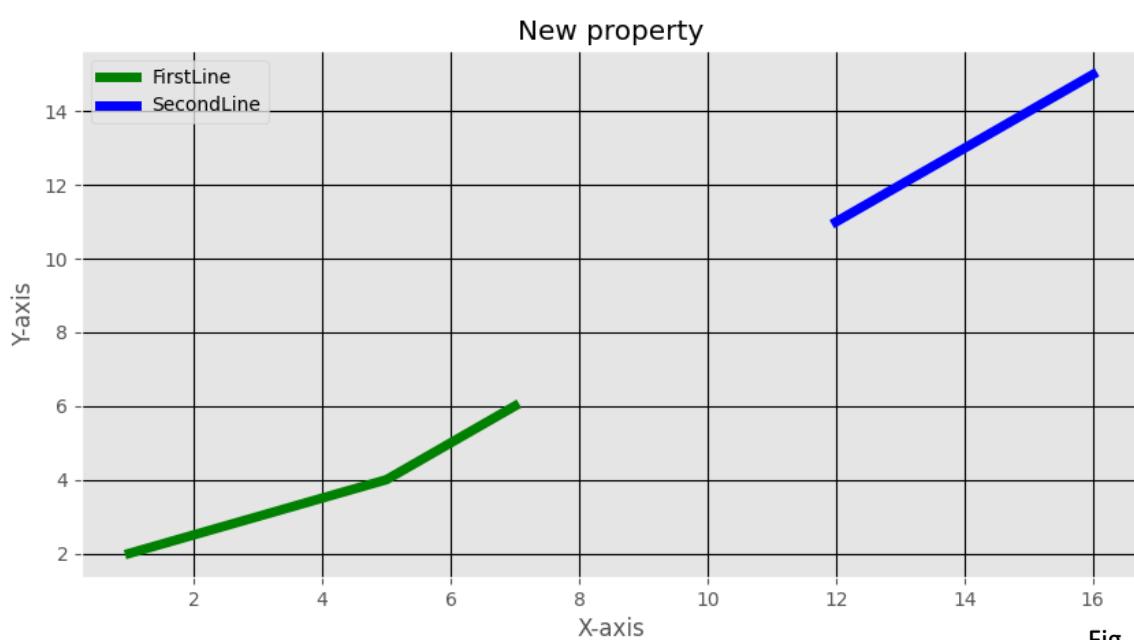


Fig. 5.83

- On this I worked on the accessing the dictionary and try to apply them on the presentation from it

- Thus the following code is regarding to the representation of the data which is stored in the dictionary format
- In addition, do this all stuff by accessing the dictionary keys and values.
- Code:-

```

1  from matplotlib import pyplot as plt
2
3  citydata = {'AHMEDABAD':50000,'SURAT':43000,'RAJKOT':55000,'VADODARA':22000}
4  cities = list(citydata.keys())
5  cases = list(citydata.values())
6
7  fig = plt.figure(figsize=(10,5))
8
9  plt.bar(cities,cases,width=0.65)
10 plt.xlabel("cities")
11 plt.ylabel("cases")
12 plt.title("city wise total cases")
13 plt.show()

```

Fig. 5.84

- Output:-

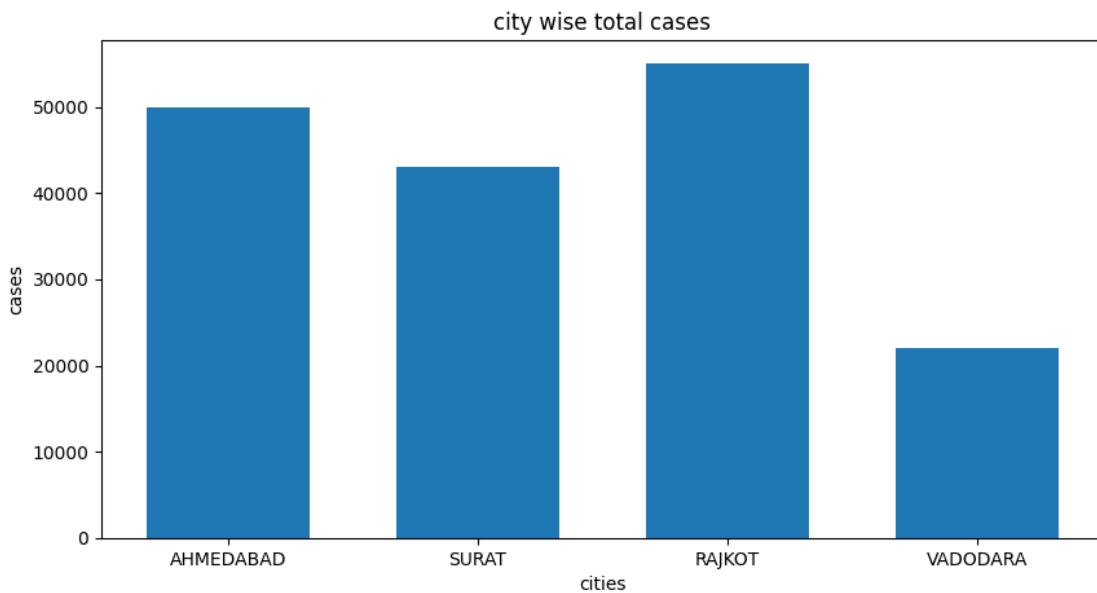


Fig. 5.85

- The API can also be represented using the matplotlib package
- For the API access, we need to have the requests package also for requesting the API for the data.

- Following is the code regarding to the representation of the USA population data by using the USA population API
- So first, we need to requests the data and get the data and by accessing the data, we can have the required representation.
- Following code is regarding to that.
- Code:-

```

1 import requests
2 from matplotlib import pyplot as plt
3
4 url = requests.get("https://datausa.io/api/data?drilldowns=Nation&measures=Population")
5 data = url.json()
6
7 year = []
8 population = []
9
10 for i in range(0,len(data["data"])):
11     year.append(data["data"][i]["Year"])
12     population.append(data["data"][i]["Population"])
13
14 fig = plt.figure(figsize=(10,5))
15 plt.bar(year,population,width=0.50)
16 plt.xlabel("year")
17 plt.ylabel("population")
18 plt.title("USA population data")
19 plt.show()

```

Fig. 5.86

- Output:-

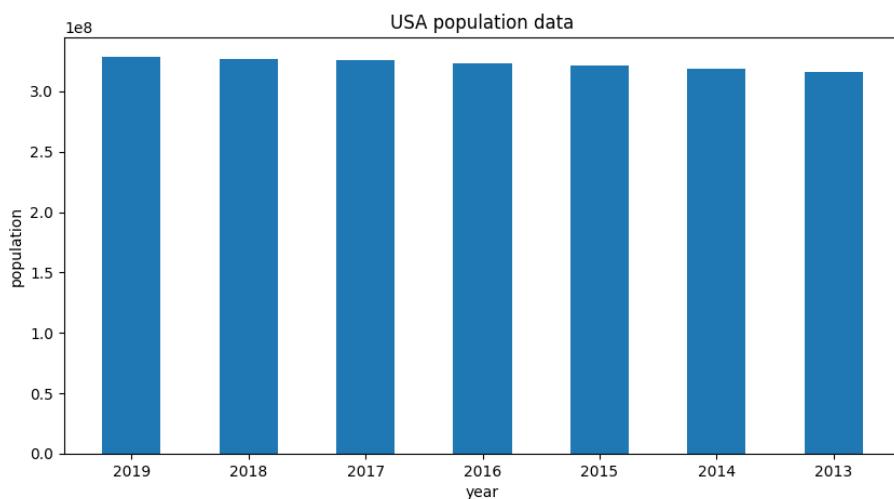


Fig. 5.87

- Same I have done to show the covid cases state wise in a bar-graph format by using the COVID-19 API
- The code regarding to them is shown as below

- Code:-

```

1 import requests
2 from matplotlib import pyplot as plt
3
4 url = requests.get("https://api.rootnet.in/covid19-in/stats/latest")
5 apidata = url.json()
6
7 location = []
8 totalcases = []
9
10 for i in range(0,len(apidata["data"]["regional"])):
11     location.append(apidata["data"]["regional"][i]["loc"])
12     totalcases.append(apidata["data"]["regional"][i]["totalConfirmed"])
13
14 fig = plt.figure(figsize=(20,20))
15 plt.barh(location,totalcases)
16 plt.xlabel("location")
17 plt.ylabel("totalcases")
18 plt.title("Total Covid19 cases")
19 plt.show()

```

Fig. 5.88

- Output:-

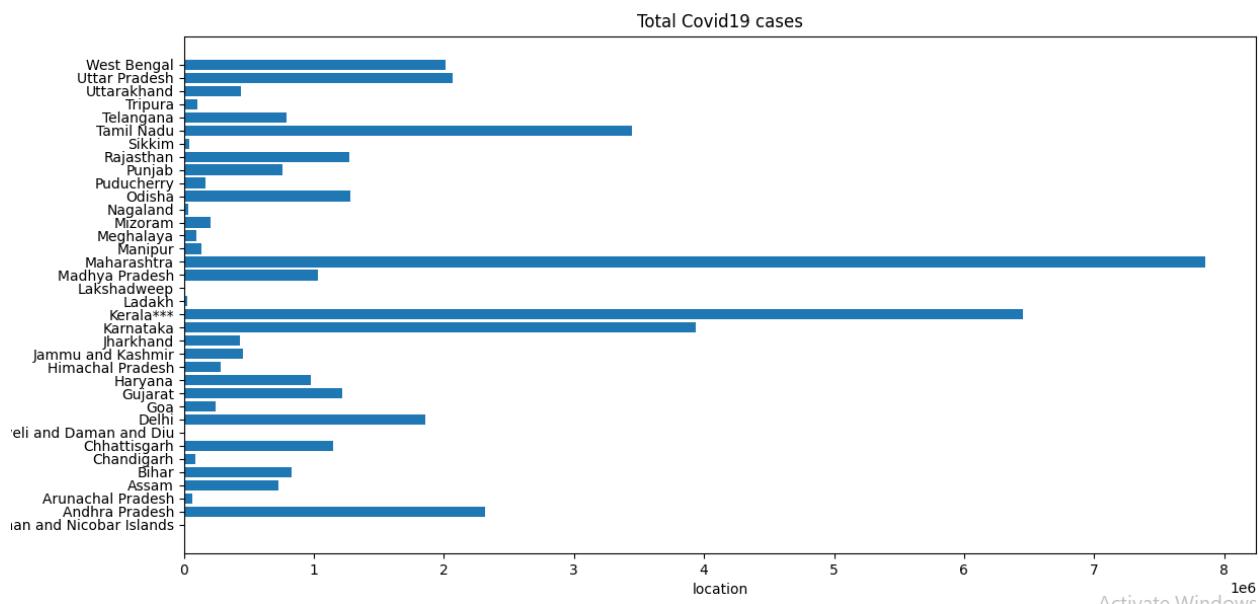
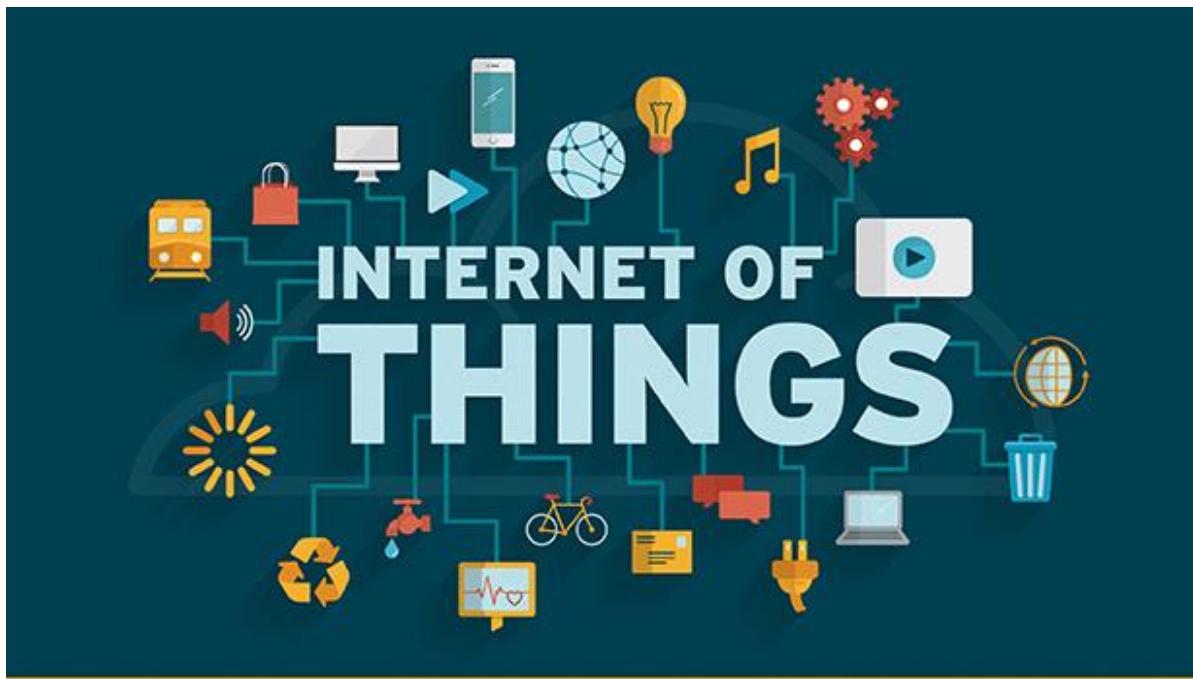


Fig. 5.89

## 6. Basics of Internet of things (IOT)

### 6.1 Explanation:-



- Internet of things is connecting everything to the internet so by using them we can control things remotely and also make the machines do their work automatically
- For the basic use of the IOT there many microcontrollers available in the market some them are:-

ATMEL AVRs:

- ATMEGA32 series
- ATMEGA16 series
- AVR128 series

Texas Instruments:

- SimpleLink series (ARM Cortex M4)
- MSP430 series (MSP432)

Espressif:

- ESP8266
- ESP32

Others:

- 8051 Intel

- Addition to that there are some widely used IOT used which has this microcontrollers in built and others pins are also available so we can make use of them very easily.
- Following are boards which are widely used:-

## 6.2 Arduino:-

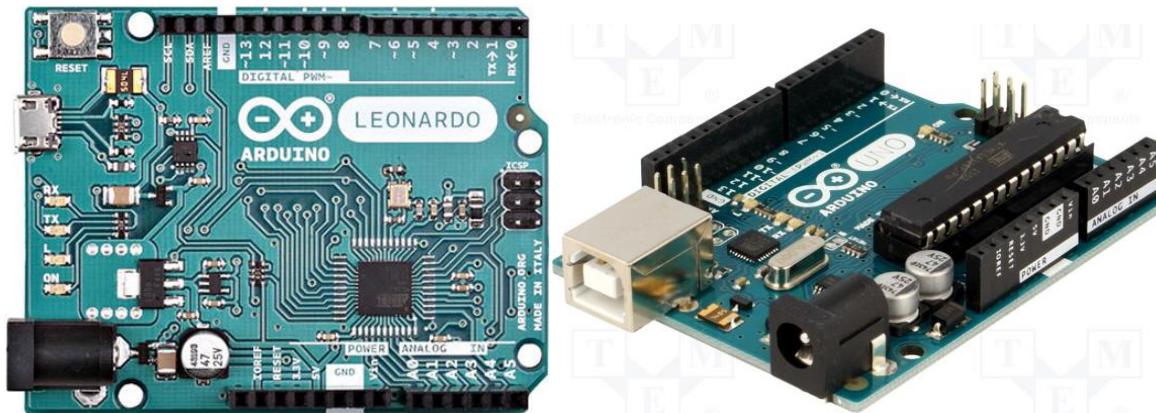


Fig. 6.1

- The above shown board is Arduino there are many variations available but this is most popular one

## 6.3 espseries:-

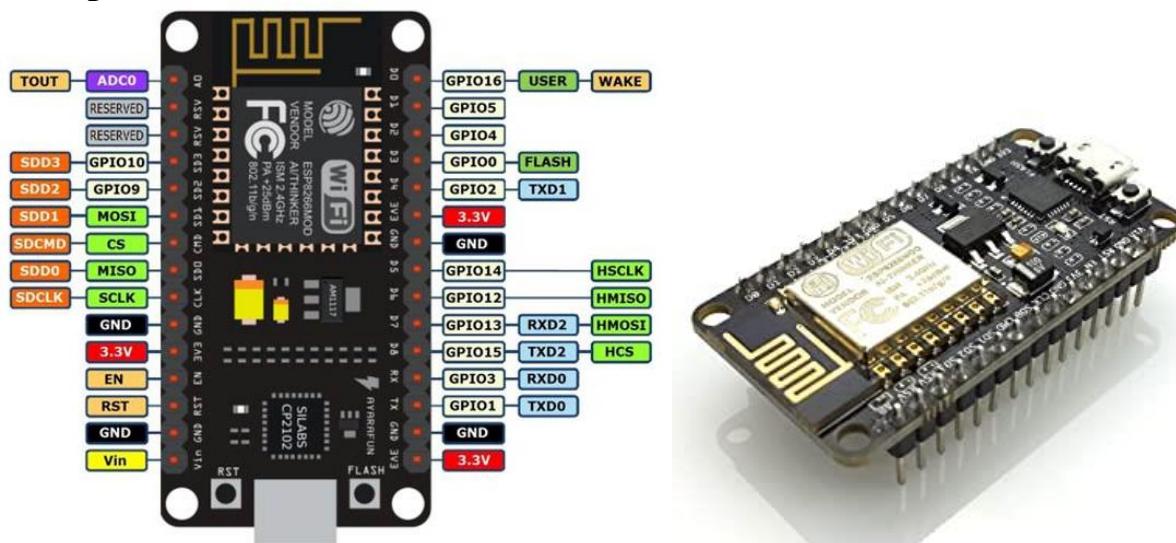


Fig. 6.2

- The board shown above is from NodeMcu ESP8266, which is board, which has Wi-Fi inbuilt so, it can connect to the server directly and we send the data to the server by using this board.
- There also an another board which is the updated version of this that board is known as NodeMcu ESP32 which has Bluetooth and Wi-Fi inbuilt and have range of ip address greater than Esp8266.
- The board is as shown below:-

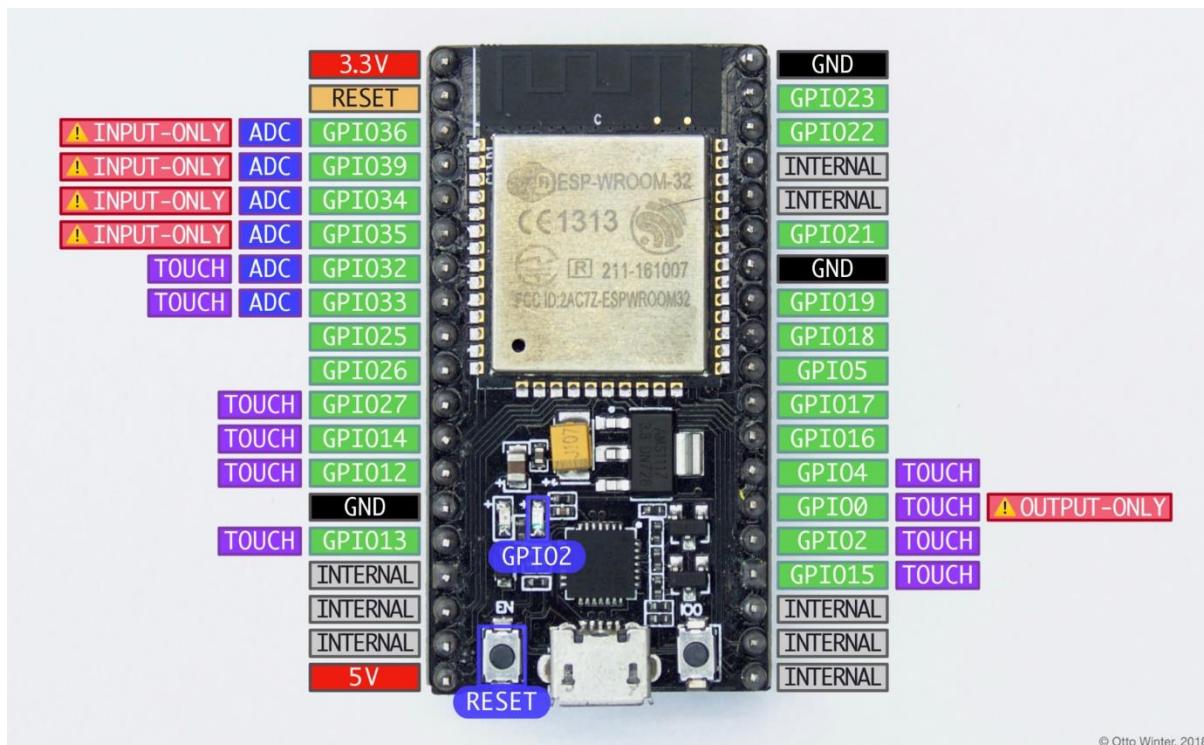


Fig. 6.3

### 6.3 ARDUINO IDE:-

- To use all these boards we need an integrated development environment so this there is an IDE which is known as Arduino IDE
- In which we can add this board from the tools, add the library related to them, can code in that , finally we can upload the code and able to see the output



- This link for the software:-<https://www.arduino.cc/en/software>
- As we can see in the interface we have tools inside that we boards manager in which we add some of the boards directly and also add zip from or pc
- After writing the code we need to verify the code first in which it will compile and give completion if any after that we can upload the code
- There is also serial monitor present in this software which we can see inside the tools menu which is show the output in software.
- The interface will look like:-

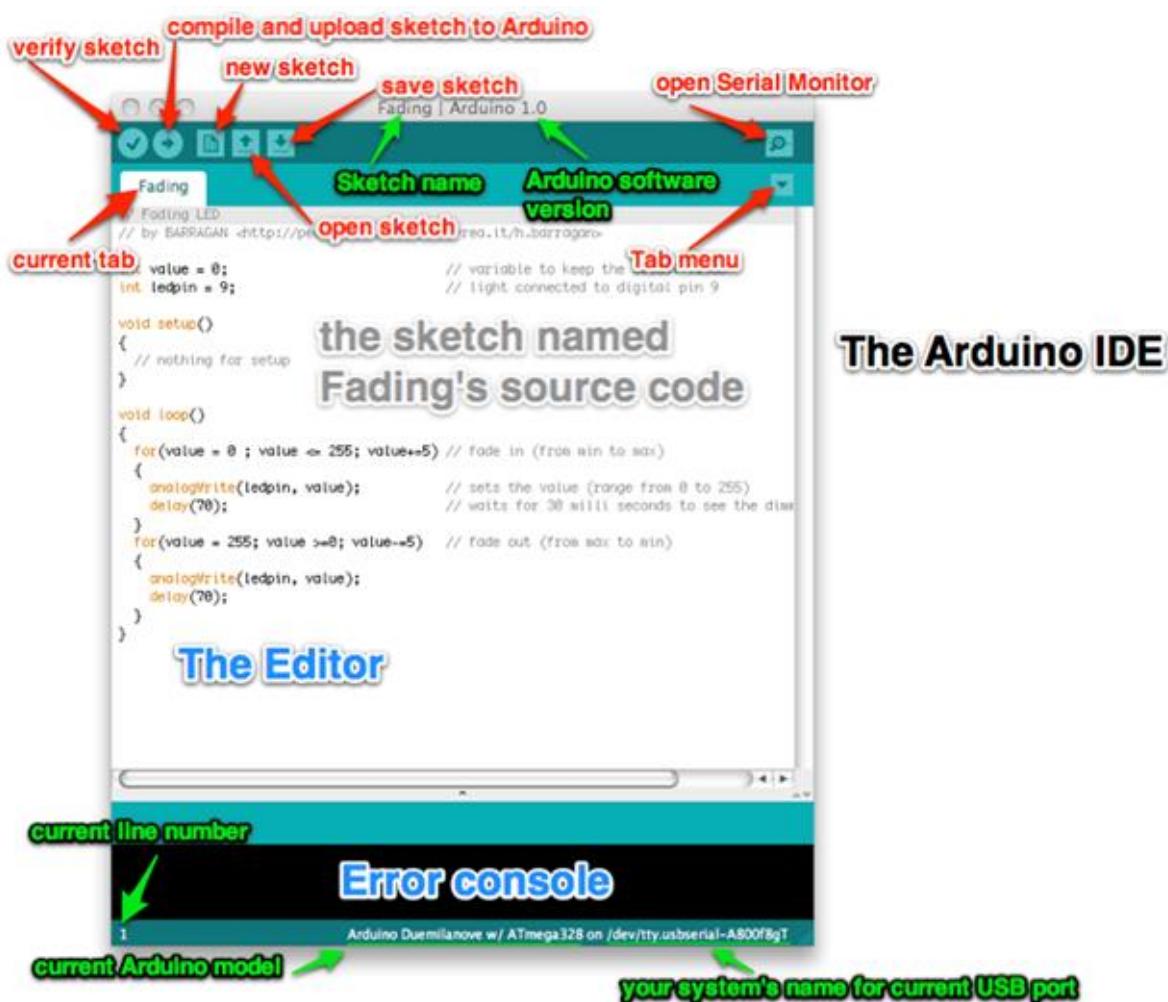


Fig. 6.4

## 7. Sensors

### 7.1 Explanation:-

- A sensor is a device that detects and responds to some type of input from the physical environment. The specific input could be light, heat, motion, moisture, pressure, or any one of a great number of other environmental phenomena. The output is generally a signal that is converted to human-readable display at the sensor location or transmitted electronically over a network for reading or further processing.
- The sensors are the integral part of IOT as it uses the sensors in many ways
- There are lots of sensors available in the market every sensor has a different use
- For an example:
  1. IR sensor is used to detect the motion
  2. Temperature sensor is used to sense the temperature of the surroundings

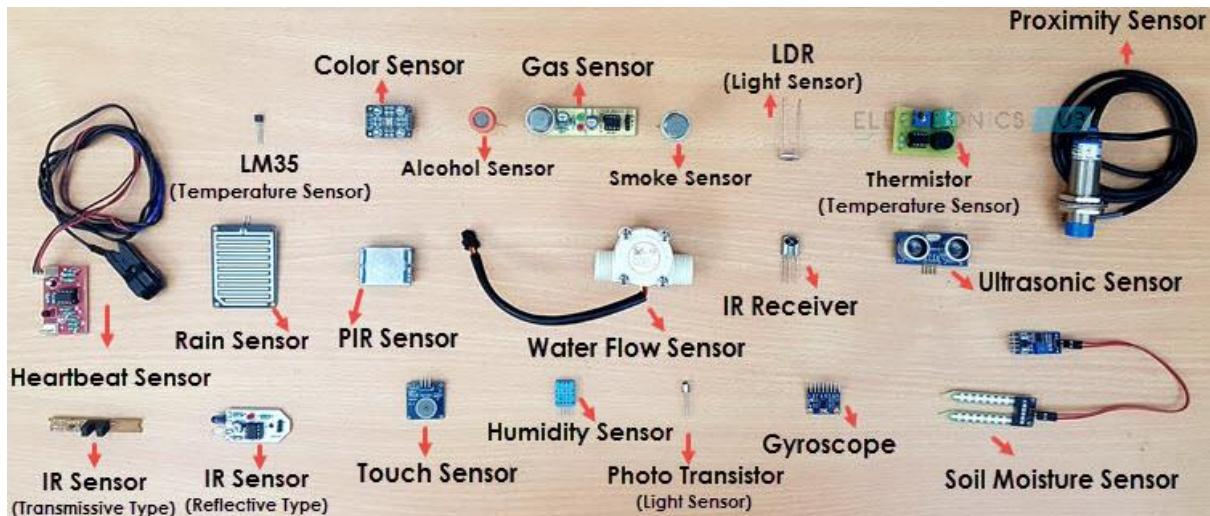


Fig. 7.1

- The basic what sensors does it sees the changes in the current and the voltage when they feel the change they get excited and gives the respective output according to its function.

## 7.2 Water level sensor:-

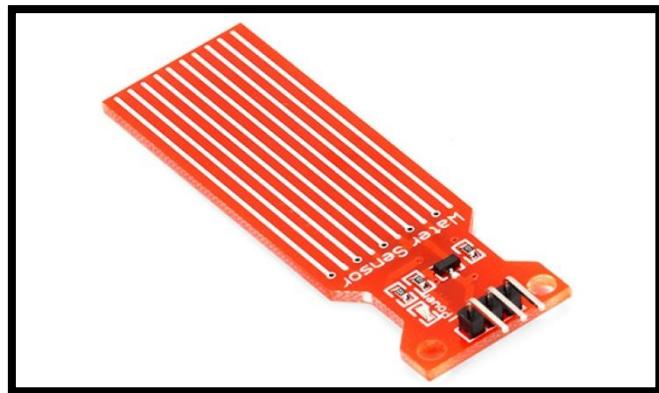


Fig. 7.2

- The above shown is the water level sensor used to check the level of the water inside any kind of container
- This type of sensors are being used in big water tanks, boilers.
- In the span of 3 Months of the internship, I have engaged with many sensors, which I will be listing down and water level sensors is one of them.

Circuit:-

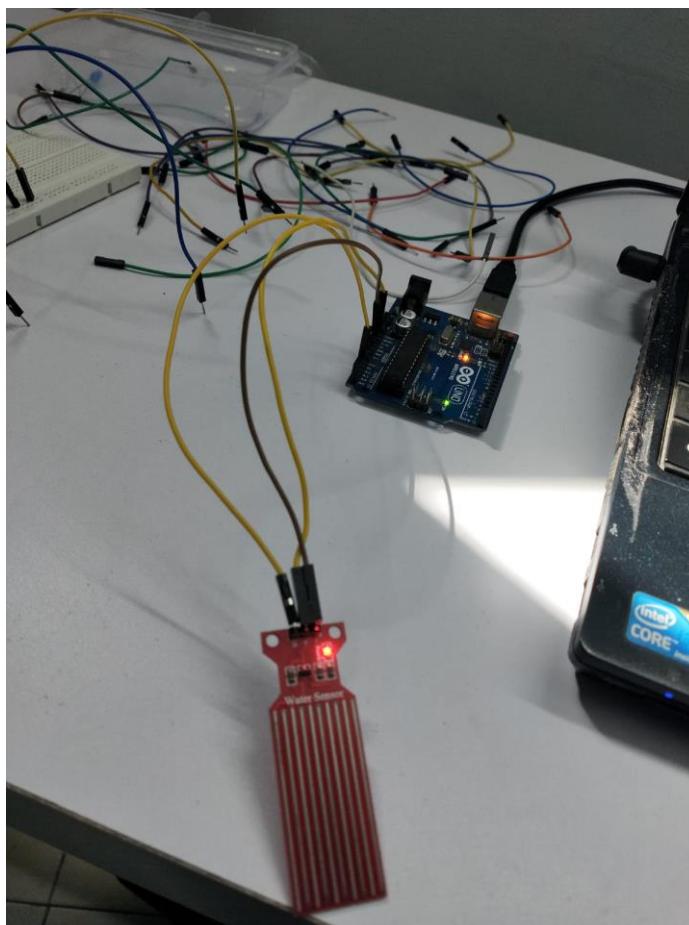
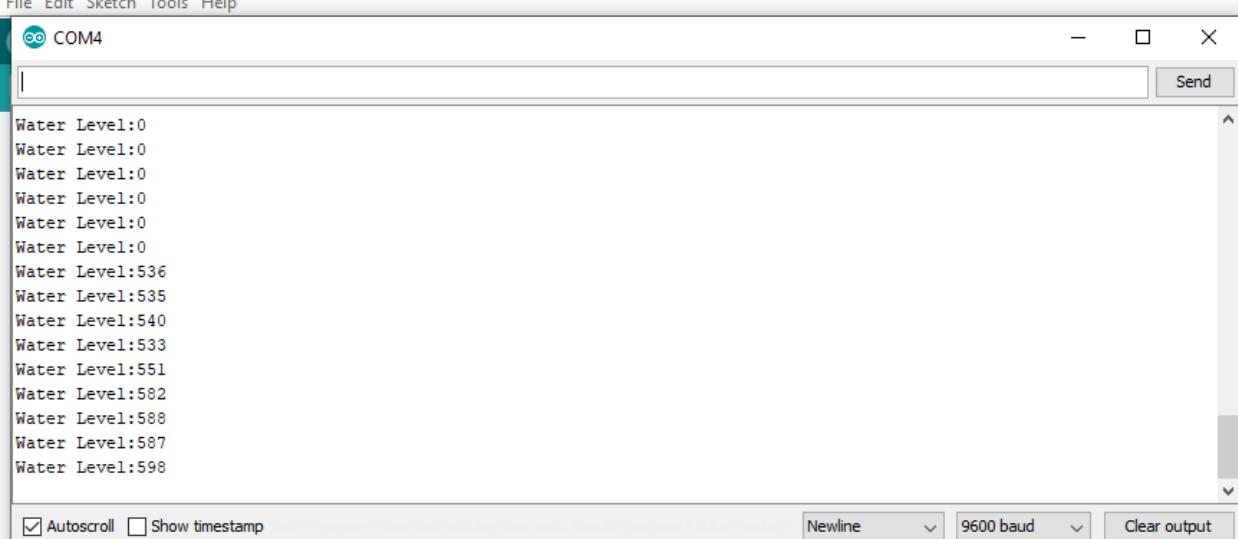


Fig. 7.3

- In this, I have plugged in with the Arduino and coded so I can have the water of any container.

**Output:-**



The screenshot shows the Arduino Serial Monitor window titled "sketch\_mar05c | Arduino 1.8.10". The port is set to "COM4". The text area displays a series of "Water Level" readings starting from 0 and increasing in increments of 1 up to 598. The bottom of the window includes checkboxes for "Autoscroll" and "Show timestamp", and buttons for "Send", "Newline", "9600 baud", and "Clear output".

```
Water Level:0
Water Level:536
Water Level:535
Water Level:540
Water Level:533
Water Level:551
Water Level:582
Water Level:588
Water Level:587
Water Level:598
```

Fig. 7.4

### 7.3 Temperature and Humidity Sensor

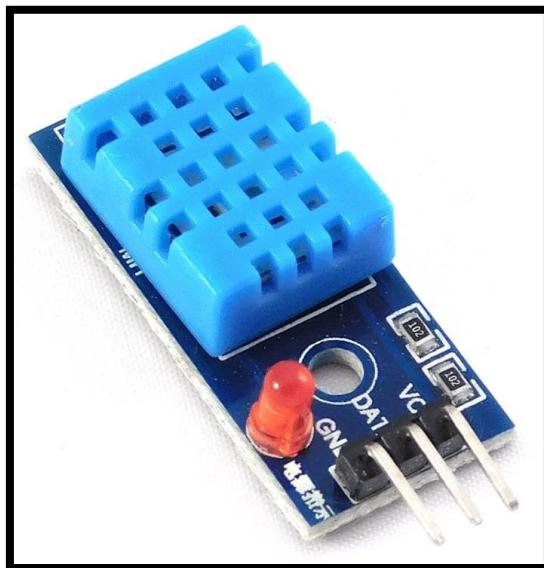


Fig. 7.5

- The above shown is the temperature sensor (DHT11)
  - Used to sense the temperature and the Humidity of the surroundings.
  - After connecting it with Arduino it will give the output in the serial monitor
- Circuit:-

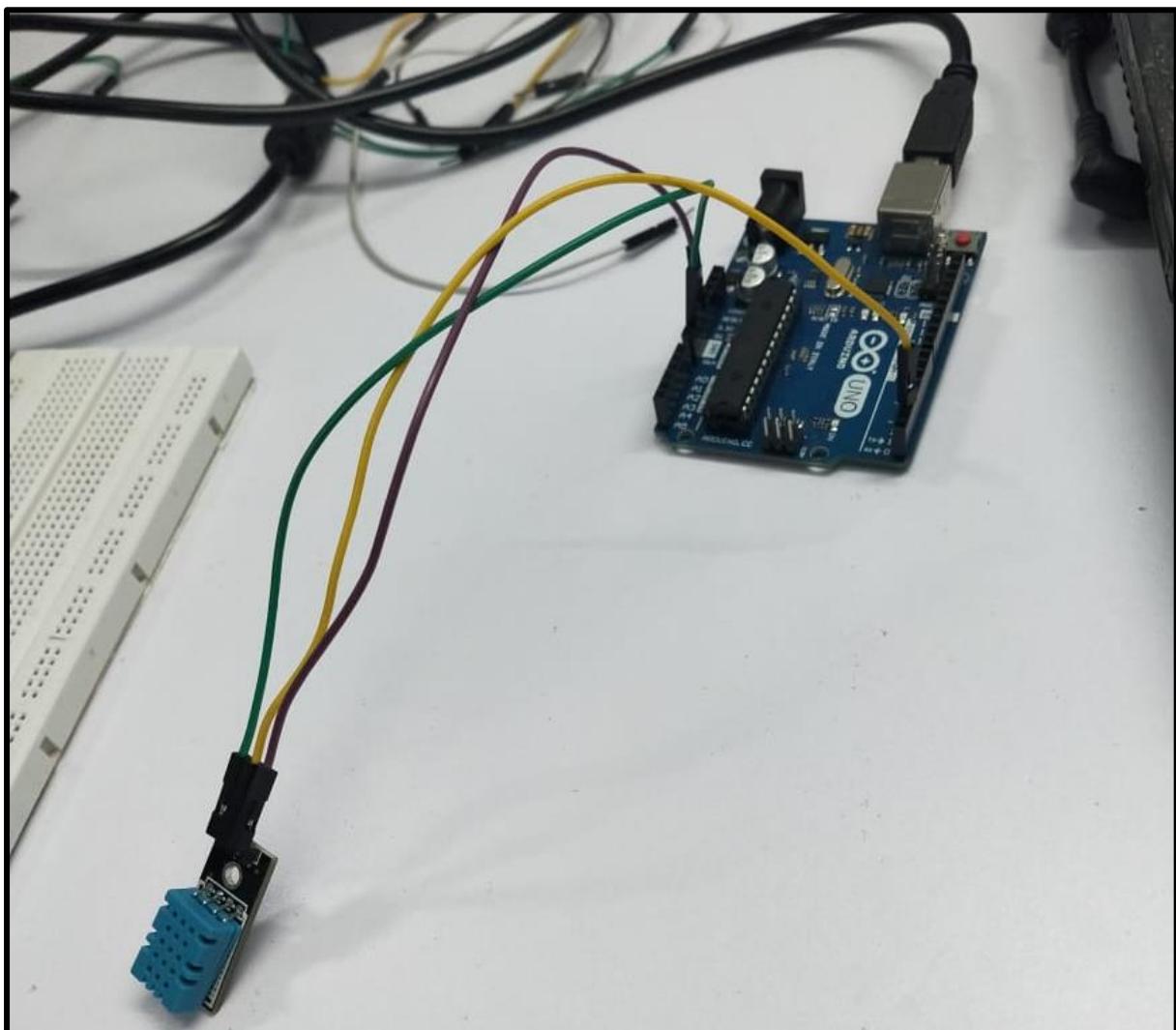


Fig. 7.6

Output:-

∞ COM3

```
38.00% Temperature: 28.00 Humidity: 38.00% Temperature: 28.00 Humidity: 38.00% Tempera
```

Fig. 7.7

## 7.4 Rain Drop Sensor

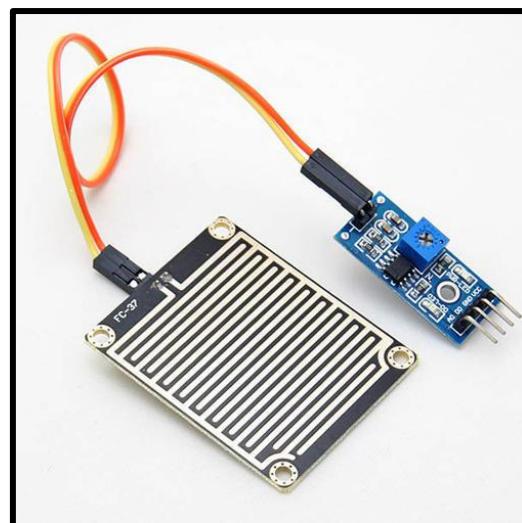


Fig. 7.8

- The above shown in the rain sensor it senses the intensity of the rain falling on it.
- After connecting it with Arduino it will give the output in the serial monitor
- The following circuit is made in such way that if in case the Rain value goes High to a certain limit it will light up the led and Buzzer will start

Circuit:-

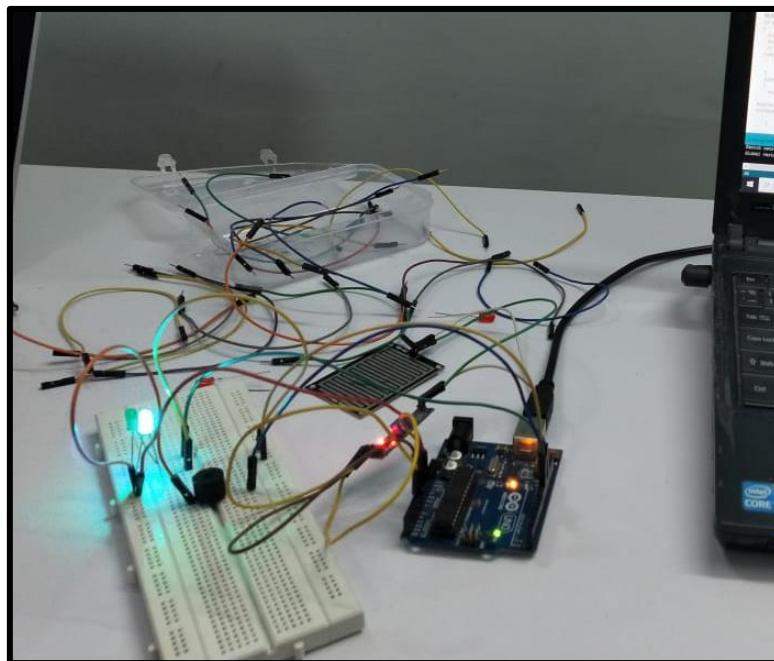
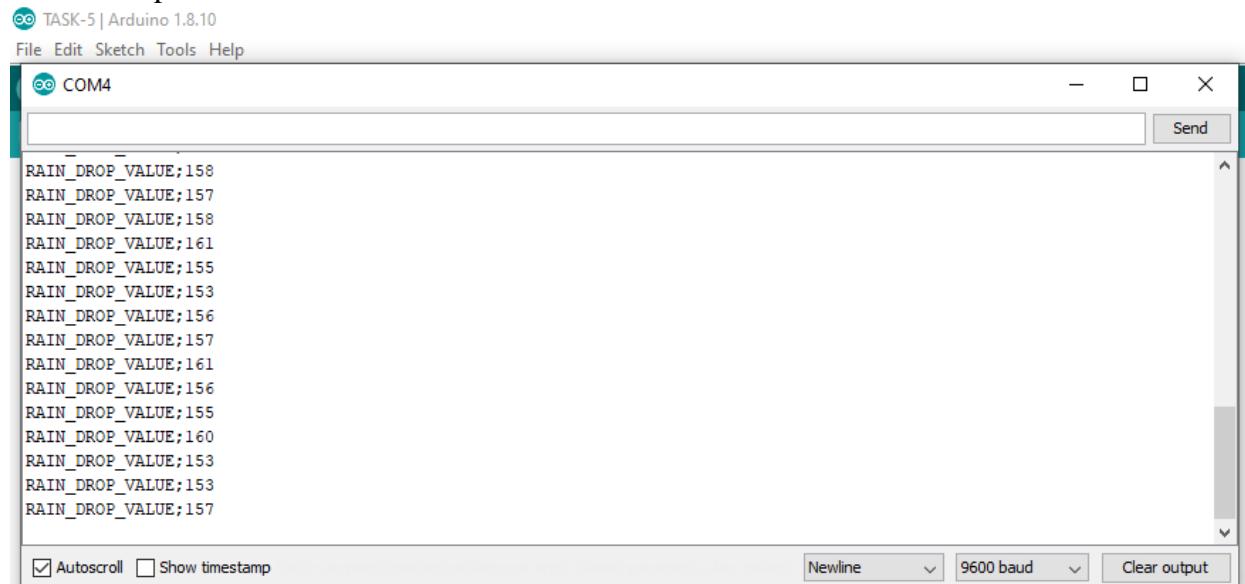


Fig. 7.9

**Output:-**

The screenshot shows the Arduino Serial Monitor window titled "TASK-5 | Arduino 1.8.10". The window has a menu bar with File, Edit, Sketch, Tools, and Help. The main area displays a series of text lines: "RAIN\_DROP\_VALUE;158", "RAIN\_DROP\_VALUE;157", "RAIN\_DROP\_VALUE;158", "RAIN\_DROP\_VALUE;161", "RAIN\_DROP\_VALUE;155", "RAIN\_DROP\_VALUE;153", "RAIN\_DROP\_VALUE;156", "RAIN\_DROP\_VALUE;157", "RAIN\_DROP\_VALUE;161", "RAIN\_DROP\_VALUE;156", "RAIN\_DROP\_VALUE;155", "RAIN\_DROP\_VALUE;160", "RAIN\_DROP\_VALUE;153", "RAIN\_DROP\_VALUE;153", "RAIN\_DROP\_VALUE;157". At the bottom of the monitor, there are checkboxes for "Autoscroll" and "Show timestamp", and buttons for "Newline", "9600 baud", and "Clear output".

Fig. 7.10

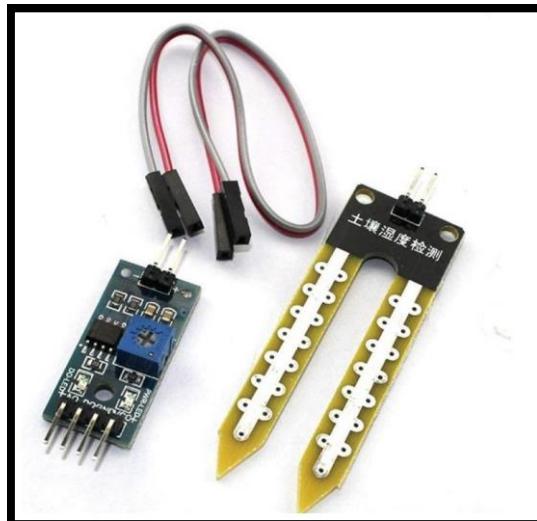
**7.5 Soil Moisture Level Sensor**

Fig. 7.11

- The above shown is the soil moisture level sensor which senses the moisture present inside the soil.
- After connecting it with Arduino it will give the output in the serial monitor
- The following circuit is made in such way that if in case the moisture value goes low to a certain limit it will light up the led and Buzzer will start

Circuit:-

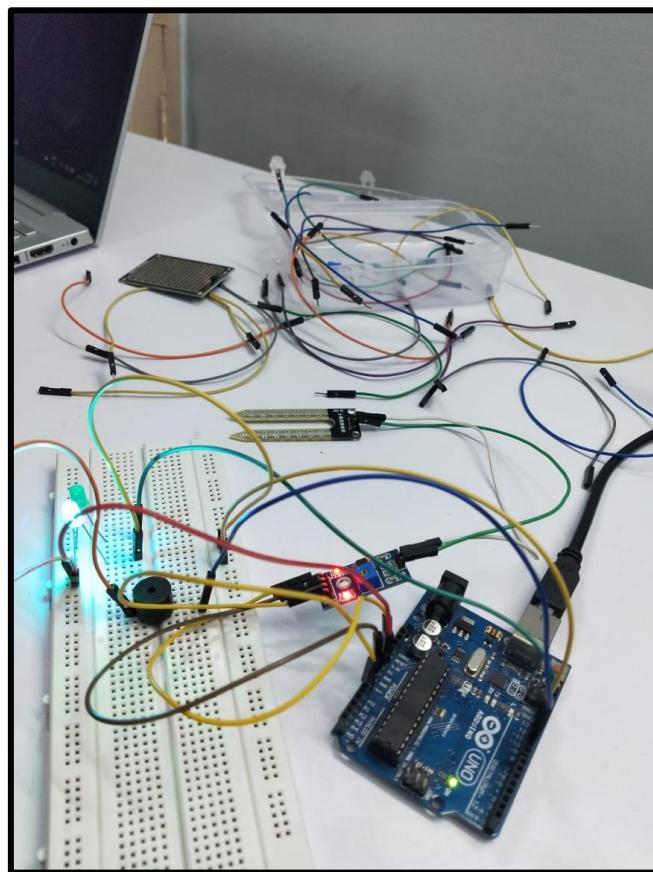


Fig. 7.12

Output:-

```
∞ TASK-5 | Arduino 1.8.10
File Edit Sketch Tools Help
COM4
Send
MOISTURE_VALUE;214
MOISTURE_VALUE;209
MOISTURE_VALUE;212
MOISTURE_VALUE;213
MOISTURE_VALUE;214
MOISTURE_VALUE;212
MOISTURE_VALUE;212
MOISTURE_VALUE;212
MOISTURE_VALUE;213
MOISTURE_VALUE;210
MOISTURE_VALUE;211
MOISTURE_VALUE;209
MOISTURE_VALUE;213
MOISTURE_VALUE;208
MOISTURE_VALUE;211
Autoscroll  Show timestamp
Newline 9600 baud Clear output
```

Fig. 7.13

- I have gone through all this sensors during the internship

## 8. Django framework

### 8.1 Explanation:-

- Django is an MVT web framework that is used to build web applications. The huge Django web-framework comes with so many “batteries included” that developers often get amazed as to how everything manages to work together. The principle behind adding so many batteries is to have common web functionalities in the framework itself instead of adding latter as a separate library.
- One of the main reasons behind the popularity of Django framework is the huge Django community. The community is so huge that a separate website was devoted to it where developers from all corners developed third-party packages including authentication, authorization, full-fledged Django powered CMS systems, e-commerce add-ons and so on. There is a high probability that what you are trying to develop is already developed by somebody and you just need to pull that into your project.

Why should you use Django?

- Django is designed in such a way that encourages developers to develop websites fast, clean and with practical design. Django’s practical approach to getting things done is where it stands out from the crowd.
- If you’re planning to build a highly customizable app, such as social media website, Django is one of the best frameworks to consider. Django strength lies in its interaction between users or its ability to share different types of media. One of the great advantage of django is its ability to utilize large community-based support which gives you highly customizable third-party ready to use plugins in your applications.
- Below are the top ten reasons to choose Django for web development –

#### Python

- Python is arguably one of the easiest programming languages to learn because of its simple language constructs, flow structure and easy syntax. It is versatile and runs websites, desktop applications and mobile applications embedded in many devices and is used in other applications as a popular scripting language.

#### Batteries Included

- Django comes with common libraries, which are essential to build common functionalities like URL routing, authentication, an object-relational mapper (ORM), a templating system and db-schema migrations.

#### Built-in admin

- Django has an in-built administration interface which lets you handle your models, user/ group permissions and to manage users. With model interface in place, there is no need for a separate database administration program for all but advanced database functions.

### Doesn't get in your way

- Creating a Django application adds no boilerplate and no unnecessary functions. There's no mandatory imports, third-party libraries and no XML configuration files.

### Scalable

- Django is based on MVC design pattern. It means that all the entities like db (database), back-end and front-end code are individual entity. Django allows us to separate code from the static media including pictures, files, CSS and JavaScript that make up your site.
- Django supports a full list of third-party libraries for web servers, caching, performance management, clustering and balancing. One of the advantages Django provides is the support for major email and messaging applications and services like ReST and OAuth.

### Battle tested

- Django was first open-sourced in 2005. After 12 years of growth, Django now not only runs news publishing websites but also runs all or part of major global enterprise like Pinterest, Instagram, Disqus, Bitbucket, EventBrite and Zapier. This makes it a robust and reliable web framework to work with.

### Huge package support

- Because of its large community support and huge developers network, there is a high possibility that whatever you intend to do might have been done before. Large international community of developers contribute to the community by releasing their projects as open-source packages.
- One such repository of these projects is Django Package site. Currently, Django packages list over 3400 plus reusable Django apps, sites and tools to use in our Django projects.

### Actively developed

- One of the biggest risks associated with open source project is its sustainability. We cannot be sure if it lasts long.
- There is no such risk with Django as it is 12 years old. Its consistent releases, newer/better versions and active community is growing every-day with a large core team of voluntary contributors who maintains and improve the code base every-day.

## **8.2Procedure:-**

- So I followed the below point-
  1. Install Django by the command ‘pip install Django’
  2. Than check is the Django is installed properly or not
  3. Now if we want to start any project than give the following command and the name of the project ‘django-admin startproject name\_of\_project.’

4. than give the command ‘python manage.py runserver’ to see the server and it will show the following window in the chrome and this link will be provided in the terminal after writing the command

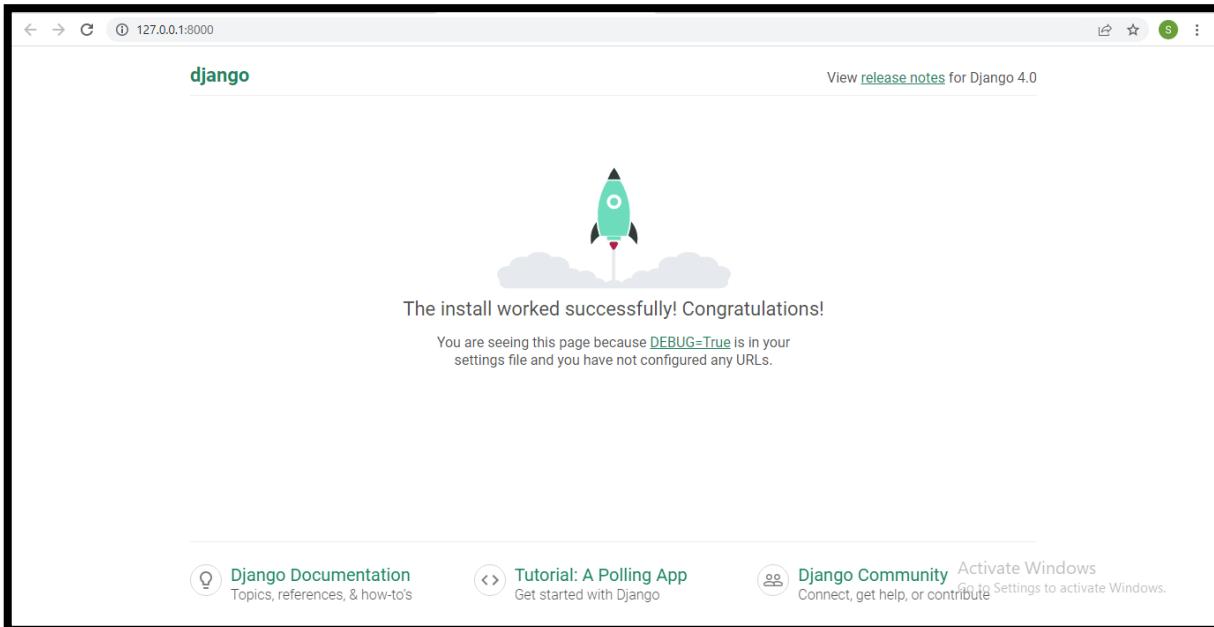


Fig. 8.1

5. if we have changed anything in the given frame than to detect that changes and save them command is given ‘python manage.py makemigrations’
  6. now to send the change on the server we need to run the command ‘python manage.py migrate’
  7. After starting the project we can create a user which will be helpful to login inside the project site. It is possible by the command ‘python manage.py createsuperuser’. After running this we need add the username, email, password, confirm password.
- So basically I have created the ‘infolabz’ project and the username and password are:- thaker45 and week1234
  - Link for the server:-- <http://127.0.0.1:8000/>
  - After writing admin at the back of the link I can have the following window open and by giving the details that is username and password I can login.

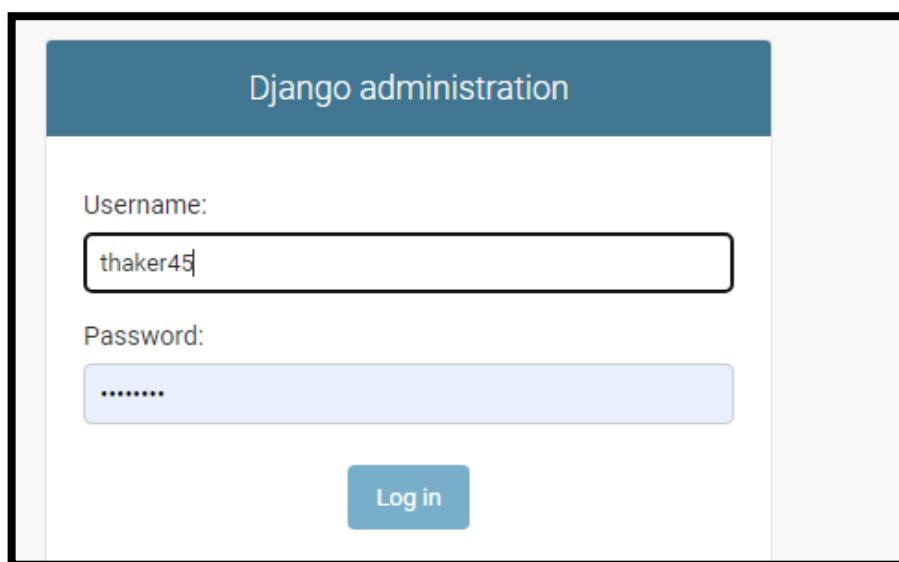


Fig. 8.2

- So this way I have created the ‘infolabz’ named project and created the user credentials
- On this day I have gone ahead in the learning and the application of the Django framework
- So basically I created the application in the Django framework
- Following are steps to be followed:-
- I have written the command ‘django-admin startapp app\_name’
- As an example I have created the app called ‘new’
- After creation of the application we need to register that inside the setting of the respective created project
- Following is the changes I made in the settings
- This setting file comes under the project whichever is created and here I have under the infolabz
- It is mandatory to write in the above format only
- Now as discussed above we need create user id and password to login and after we have the following page:

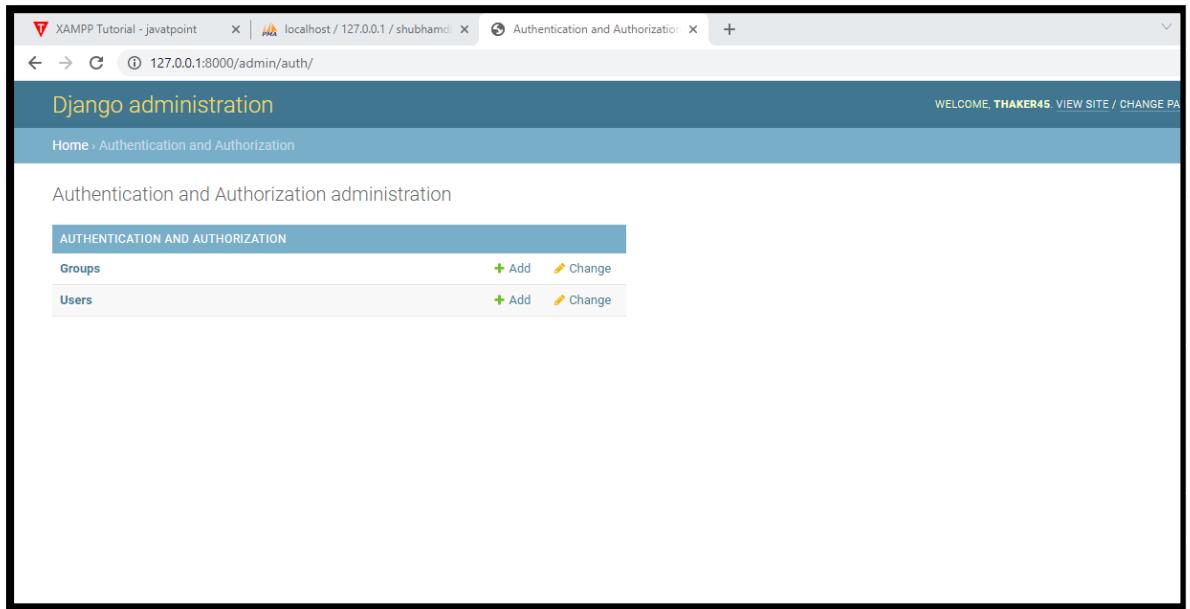


Fig. 8.3

### 8.3 Permissions:-

- Under the user sections I am able to give permissions to different users following are the permissions

```
admin | log entry | Can add log entry
admin | log entry | Can change log entry
admin | log entry | Can delete log entry
admin | log entry | Can view log entry
auth | group | Can add group
auth | group | Can change group
auth | group | Can delete group
auth | group | Can view group
auth | permission | Can add permission
auth | permission | Can change permission
auth | permission | Can delete permission
auth | permission | Can view permission
```

Fig. 8.4

- As shown as above we can change the permissions like:--
  - Log entry
  - Group
  - Permission

```

auth | user | Can add user
auth | user | Can change user
auth | user | Can delete user
auth | user | Can view user
contenttypes | content type | Can add content type
contenttypes | content type | Can change content type
contenttypes | content type | Can delete content type
contenttypes | content type | Can view content type
sessions | session | Can add session
sessions | session | Can change session
sessions | session | Can delete session
sessions | session | Can view session

```

Fig. 8.5

1. User
  2. Content type
  3. Session
- This all are the permission can be added to the respective user to change, add, delete, view.

#### **8.4 Models,Admin:-**

- Than after I build up different models and after creation of the model I registered them in the admin file of the respective application as in here I have the ‘new’ application
- Code for building the model and the admin registration file
- Here I am making the site for the name and its respective price and it will be shown in the product named table.
- Code for the model.py

```

1   from django.db import models
2
3   # Create your models here.
4
5   class product(models.Model):
6       name = models.CharField(max_length=30)
7       Price = models.IntegerField()
8

```

Fig. 8.6

- Code for the admin.py

```

1  from django.contrib import admin
2  from .models import product
3
4  # Register your models here.
5  class productdisplay(admin.ModelAdmin):
6      list_display = ('name','Price')
7
8  admin.site.register(product,productdisplay)

```

Fig. 8.7

- Output:

The figure consists of three vertically stacked screenshots of the Django administration interface.

- Screenshot 1:** Shows the 'New administration' page with a 'Products' list. A yellow box highlights the 'list\_display' line in the code above, which defines the fields 'name' and 'Price' for the 'productdisplay' class.
- Screenshot 2:** Shows the 'Add product' page. A product named 'fridge' with a price of '20000' is being added. The 'Name' field contains 'fridge' and the 'Price' field contains '20000'. Buttons for 'Save and add another', 'Save and continue editing', and 'SAVE' are visible.
- Screenshot 3:** Shows the 'Select product to change' page after the product was saved. It displays a table with one row for 'fridge' at a price of '20000'. A green success message at the top states: 'The product "product object (1)" was added successfully.'

Fig. 8.8

- I have made this table using the CharField in which I have given value max\_length of 30 it means I can name the product in max 30 characters
- I have used here integer field it takes the integer values in the price as shown as above
- Also on this day I have used how to make the drop down menu the code regarding to that is as shown as below
- Code for model.py

```

1  from django.db import models
2
3  # Create your models here.
4  class userdata(models.Model):
5      firstname = models.CharField(max_length=30)
6      lastname = models.CharField(max_length=30)
7      emailid = models.CharField(max_length=50)
8
9      list=((('electrical','ELECTRICAL'),('clothing','CLOTHING'),('food related','FOOD RELATED')))
10
11 class productdata1(models.Model):
12     productname = models.CharField(max_length=30)
13     productcategory = models.CharField(max_length=40,choices=list)
14     productprice = models.IntegerField()
15     productqunty = models.IntegerField()

```

Fig. 8.9

- Code for admin.py

```

1  from django.contrib import admin
2  from .models import userdata
3  from .models import productdata1
4  # Register your models here.
5
6  class userdatadisplay(admin.ModelAdmin):
7      list_display = ('firstname','lastname','emailid')
8
9  class prductdata1display(admin.ModelAdmin):
10     list_display = ('productname','productcategory','productprice','productqunty')
11
12 admin.site.register(userdata,userdatadisplay)
13 admin.site.register(productdata1,prductdata1display)

```

Fig. 8.10

- Output:-

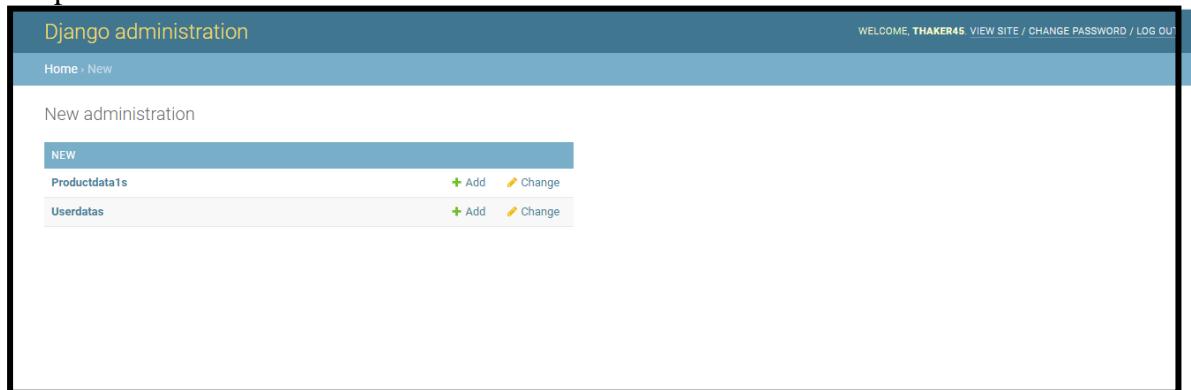


Fig. 8.11

The image contains two screenshots of the Django administration interface.

**Screenshot 1: Adding Userdata**

- The title bar says "Django administration" and "WELCOME, THAKER45 VIEW SITE / CHANGE PASSWORD / LOG OUT".
- The left sidebar shows "Home", "New", "Userdatas", and "Add userdata".
- The main area is titled "Add userdata" and contains fields for "Firstname", "Lastname", and "Emailid".
- Buttons at the bottom include "Save and add another", "Save and continue editing", and a large blue "SAVE" button.

**Screenshot 2: Adding Productdata1**

- The title bar says "Django administration" and "WELCOME, THAKER45 VIEW SITE / CHANGE PASSWORD / LOG OUT".
- The left sidebar shows "Home", "New", "Productdatas", and "Add productdata1".
- The main area is titled "Add productdata1" and contains fields for "Productname", "Productcategory" (with a dropdown menu showing "ELECTRICAL", "CLOTHING", and "FOOD RELATED"), "Productprice", and "Productquantity".
- A dropdown menu for "Productcategory" is open, showing the three options listed.
- Buttons at the bottom include "Save and add another", "Save and continue editing", and a large blue "SAVE" button.

Fig. 8.12

- This way as shown as above I created the dropdown type field
- And output are as shown as above

## 8.5 Primary and Foreign key:-

- A primary key is used to ensure data in the specific column is unique. It is a column cannot have NULL values. It is either an existing table column or a column that is specifically generated by the database according to a defined sequence.
- **Example:** Refer the figure — STUD\_NO, as well as STUD\_PHONE both, are candidate keys for relation STUDENT but STUD\_NO can be chosen as the primary key (only one out of many candidate keys).
- **Foreign Key:** A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables. It is a column (or columns) that references a column (most often the primary key) of another table.
- **Example:** Refer the figure — STUD\_NO in STUDENT\_COURSE is a foreign key to STUD\_NO in STUDENT relation.
- **Figure:**

**STUDENT**

<b>STUD_NO</b>	<b>STUD_NAME</b>	<b>STUD_PHONE</b>	<b>STUD_STATE</b>	<b>STUD_COUNTRY</b>	<b>STUD_AGE</b>
1	RAM	9716271721	Haryana	India	20
2	RAM	9898291281	Punjab	India	19
3	SUJIT	7898291981	Rajasthan	India	18
4	SURESH		Punjab	India	21

**Table 1****STUDENT\_COURSE**

<b>STUD_NO</b>	<b>COURSE_NO</b>	<b>COURSE_NAME</b>
1	C1	DBMS
2	C2	Computer Networks
1	C2	Computer Networks

**Table 2**

- Let's see the difference between Primary Key and Foreign Key:

**S.NO. PRIMARY KEY**

1 A primary key is used to ensure data in the specific column is unique.

2 It uniquely identifies a record in the relational database table.

3 Only one primary key is allowed in a table.

4 It is a combination of UNIQUE and Not Null constraints.

5 It does not allow NULL values.

6 Its value cannot be deleted from the parent table.

**FOREIGN KEY**

A foreign key is a column or group of columns in a relational database table that provides a link between data in two tables.

It refers to the field in a table, which is the primary key of another table.

Whereas more than one foreign key are allowed in a table.

It can contain duplicate values and a table in a relational database.

It can also contain NULL values.

Its value can be deleted from the child table.

S.NO.	PRIMARY KEY	FOREIGN KEY
7	It constraint can be implicitly defined on the temporary tables.	It constraint cannot be defined on the local or global temporary tables.

## 8.6 URL Handling:-

- In the URL Handling part we need handle the template according to the need of the client
- So first, after downloading the template and unzipping it we will get folder assets in which we css, js, images, fonts.
- Addition to that there will be html pages like 404, index, login, signup, etc.
- In some of the cases, we will not get the assets folder; we will directly get the css, js, images, fonts.
- Now for the editing part there will divisions inside every html page we can remove the division which is not needed to us and finally we have the needful html pages
- Now we create the project in the Django than we create an application after registration of the application we need to create a new python file named it as “ urls.py ” in we need to create the urls regarding to the views of the html pages
- Now in the views we need define each and every page
- After the creation, we need to copy every page in templates directory and assets in static file, which are created in the project.
- After all this we need to have the modified files so for that we need to install djangify
- Djangify is way to convert html templates in to Django compatible templates.
- After installing that file we need to write “djangify –d appname/templates” after running we will get modified files name folder created in the template folder where we modified in which we have files are change from
- Before djangify

```
<link href=" assets/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet"/>
```

- After djangify

```
<link href=" {% static 'assets/vendor/bootstrap/css/bootstrap.min.css' %} " rel="stylesheet"/>
```

- After this we need write “ { % load static % } ” at the top every html this is because we need to load static files in html for that load static is necessarily written at the top the html files
- Then we need to static in to url as we have made different url for our every html views

- After all this finished we run
- Python manage.py makemigration
- Python manage.py migrate
- Python manage.py runserver
- We will get our output.
- Following is the code regarding to that.

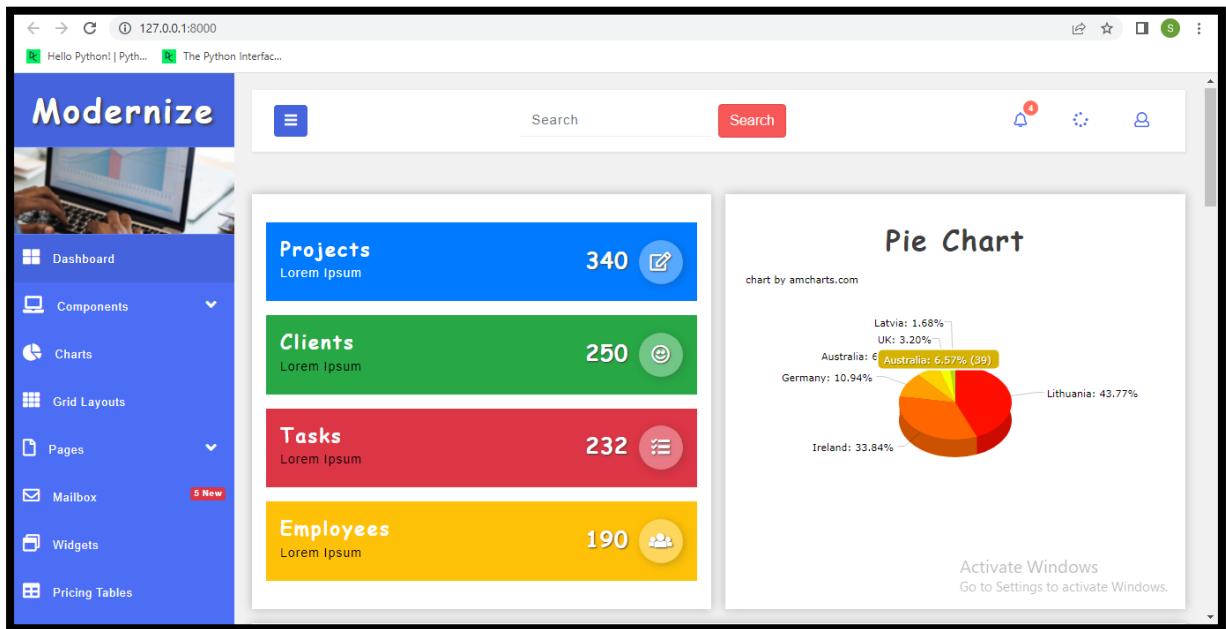


Fig. 8.13

This is the website template, which I have edited following are the code of views and urls

```

from django.shortcuts import render

# Create your views here.

def index(request):
    return render(request,'index.html')

def error(request):
    return render(request,'404.html')

def noerror(request):
    return render(request,'500.html')

def blank(request):
    return render(request,'blank.html')

def cards(request):
    return render(request,'cards.html')

def carousels(request):
    return render(request,'carousels.html')

def charts(request):
    return render(request,'charts.html')

def forgot(request):
    return render(request,'forget.html')

```

```

from django.urls import path
from . import views
urlpatterns = [
    path('',views.index,name='index.html'),
    path('404', views.error, name='404.html'),
    path('500', views.noerror, name='500.html'),
    path('blank', views.blank, name='blank.html'),
    path('cards', views.cards, name='cards.html'),
    path('carousels', views.carousels, name='carousels.html'),
    path('charts', views.charts, name='charts.html'),
    path('forget', views.forgot, name='forgot.html'),
    path('forms', views.forms, name='forms.html'),
    path('grids', views.grid, name='grids.html'),
    path('login', views.login, name='login.html'),
    path('mailbox', views.mailbox, name='mailbox.html'),
    path('maps', views.maps, name='maps.html'),
    path('modals', views.modals, name='modals.html'),
    path('pricing', views.pricing, name='pricing.html'),
    path('register', views.register, name='register.html'),
    path('tables', views.table, name='tables.html'),
    path('widgets', views.widgets, name='widgets.html')
]

```

Fig. 8.14

- Addition to this we also need to make changes in the urls file of the project also the code is as shown as below.

```

2
3     The `urlpatterns` list routes URLs to views.
4         https://docs.djangoproject.com/en/4.0/topics/http/urls/
5     Examples:
6         Function views
7             1. Add an import: from my_app import views
8                 2. Add a URL to urlpatterns: path('', views.home, name='home')
9
10    Class-based views
11        1. Add an import: from other_app.views import CBVName
12        2. Add a URL to urlpatterns: path('', CBVName.as_view(), name='name')
13
14    Including another URLconf
15        1. Import the include() function: from django.urls import include
16        2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
17
18
19 from django.contrib import admin
20 from django.urls import path, include
21
22 urlpatterns = [
23     path('admin/', admin.site.urls),
24     path('new', include('new.urls'))
25 ]

```

Fig. 8.15

## 8.7 Media:-

- We can also make the login and sign up page and that pages we can add a field in which we make user to upload his/her dp in a file field and that photo will be shown in the dashboard of the user.
- For the media upload we need to have changes in the settings of the project as we need to give the path in that I will look like if my directory name is media and in that I have a folder photos so the settings will be:-

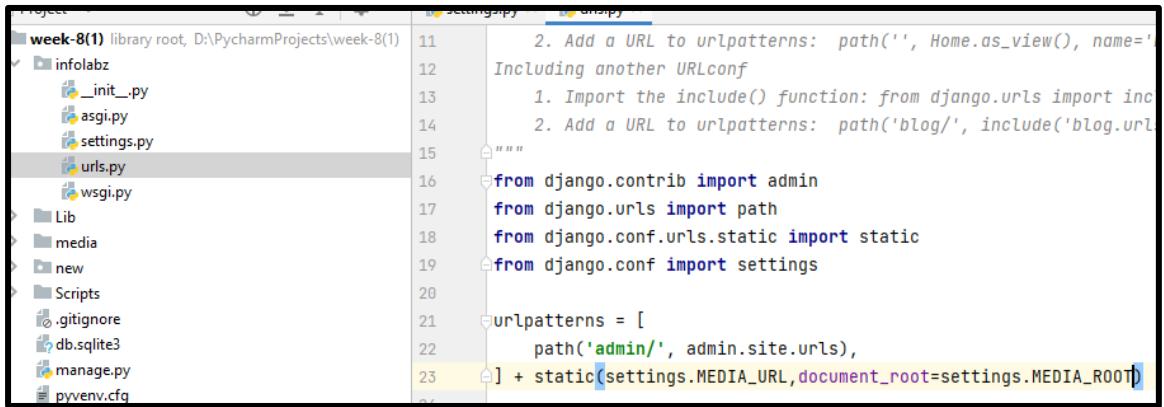
```

114
115
116     # Static files (CSS, JavaScript, Images)
117     # https://docs.djangoproject.com/en/4.0/howto/static-files/
118     import os
119     STATIC_URL = 'static/'
120     STATICFILES_DIRS = [os.path.join(BASE_DIR, 'infolabz/static')]
121     STATIC_ROOT = os.path.join(BASE_DIR, 'static')
122     MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
123     MEDIA_URL = '/media/'
124
125     # Default primary key field type
126     # https://docs.djangoproject.com/en/4.0/ref/settings/#default-auto-field
127
128     DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'

```

Fig. 8.16

- At the file we need to add this part and in the urls.py file we need add a static path to that will look like.



The screenshot shows the PyCharm interface with the project 'week-8(1)' open. The 'urls.py' file is selected in the left sidebar. The code editor displays the following content:

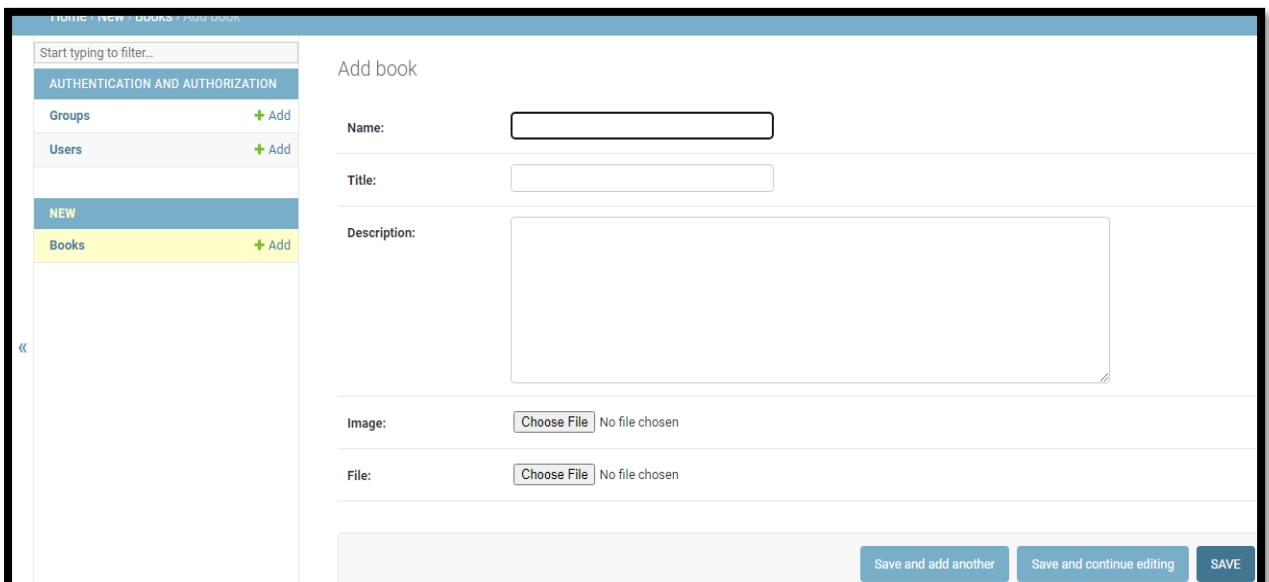
```

11     2. Add a URL to urlpatterns: path('', Home.as_view(), name='
Including another URLconf
12         1. Import the include() function: from django.urls import inc
13             2. Add a URL to urlpatterns: path('blog/', include('blog.url
14                 ''')
15
16             from django.contrib import admin
17             from django.urls import path
18             from django.conf.urls.static import static
19             from django.conf import settings
20
21             urlpatterns = [
22                 path('admin/', admin.site.urls),
23             ] + static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

```

Fig. 8.17

- After this we can make our website with an available option to upload files and this files will be saved in our database the website will look like:-



The screenshot shows the Django admin interface with the title 'Add book'. On the left, there is a sidebar with 'AUTHENTICATION AND AUTHORIZATION' and a 'NEW' section containing 'Books'. The main area has fields for 'Name', 'Title', 'Description', 'Image' (with a 'Choose File' button), and 'File' (with a 'Choose File' button). At the bottom, there are buttons for 'Save and add another', 'Save and continue editing', and 'SAVE'.

Fig. 8.18

## 9. Project work

### 9.1 Introduction

- In this internship, I am in the web Development team as python developer with an assigned project.
- During the COVID-19 pandemic, Internet of Things based health monitoring systems are potentially immensely beneficial.
- This study presents of this system that is a real-time health monitoring system utilizing the measured values of body temperature, pulse rate, oxygen saturation of the patients, which are the most important measurements required for critical care.
- Furthermore there is a lot of parameters which should be taken care of the patients but the main oxygen level and pulse is being measured through this
- In this project we are showing four parameters on our website so the health of the patients can be maintained , in this we are also creating the database through which the doctors can prescribe the medication, In the case of the emergency our page nearby medical shops are being tagged and also nearby hospitals to visit
- Keywords:Max30100,MLx90614,Ad8232,ESP8266,IOT

### 9.2 Problem Definition

- The IoT Based Health Care System is the affordable healthcare device for patients and doctors. It provides a solution for the measurement of body parameters like ECG, Temperature, Pulse rate, and Oxygen level. To get this I am using different sensors.

### 9.3 Working

- The ECG and the Temperature of the patients is being measured and send to the server using an API there is also one another API which is used to show the data from the database and from that API I am fetching the data and showing it on the website.
- The Pulse and Oxygen is being measured through one sensor only namely MAX30100.
- This sensor gives the data in a loop so we can not directly send that in the database Thus As shown in the circuit diagram there a master slave combination as there is one another Nodemcu used to send the data to the database.

## 9.4 Flow Chart of the Project



## 9.5 Circuit

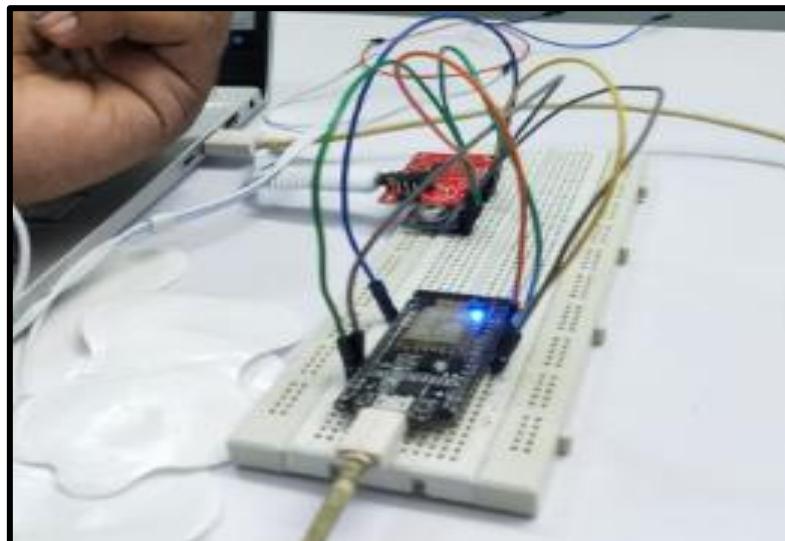


Fig. 9.1

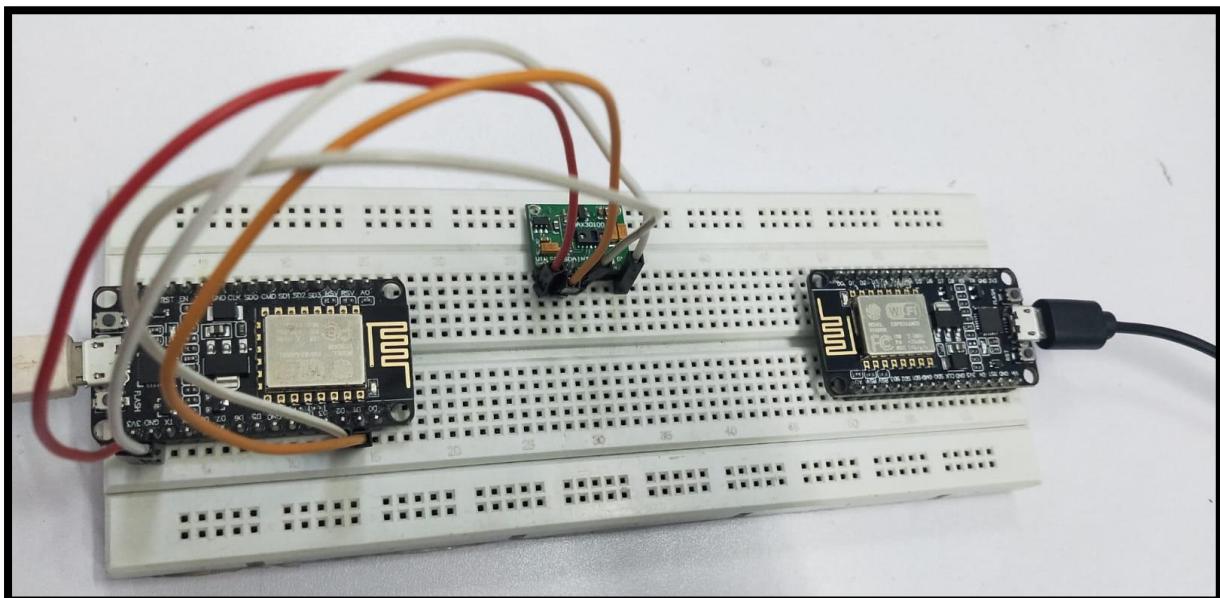


Fig. 9.2

## 9.6 Sensors Used-

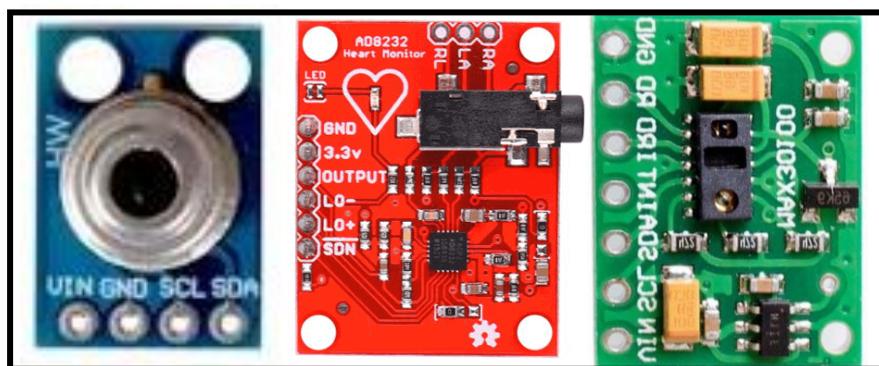


Fig. 9.3

## 9.6 Website

- Following is the dashboard of the project and other tables which were present in the dashboard side menu.

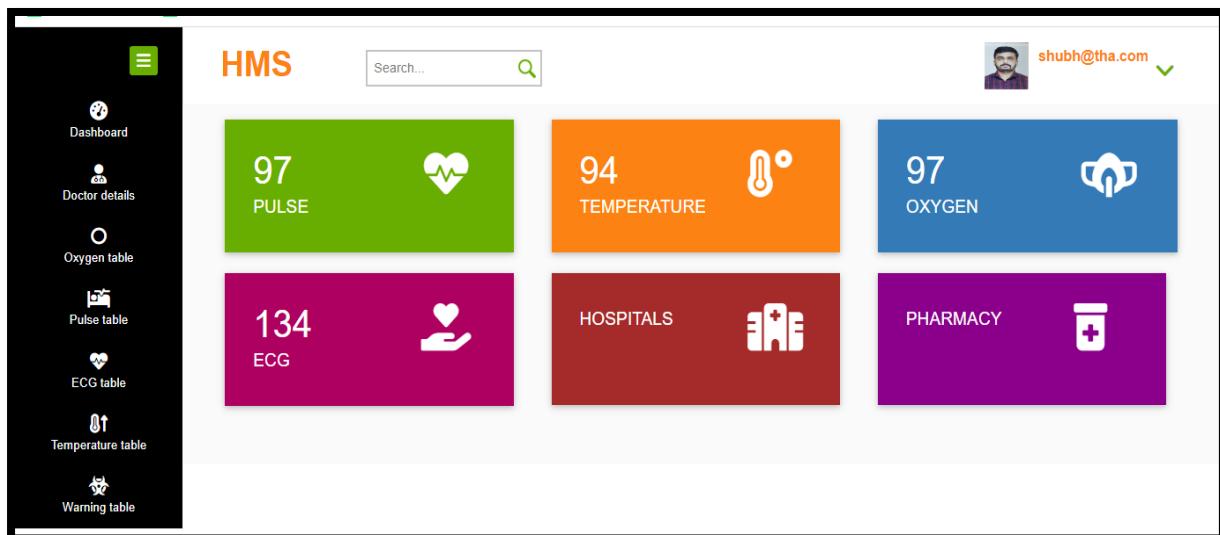
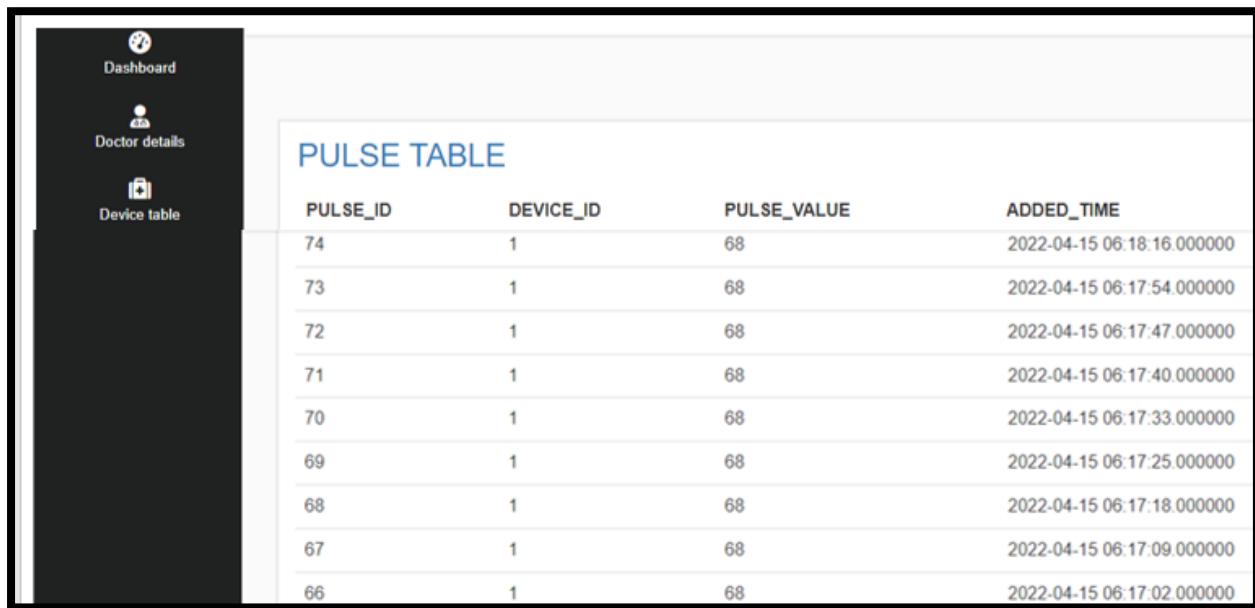


Fig. 9.4

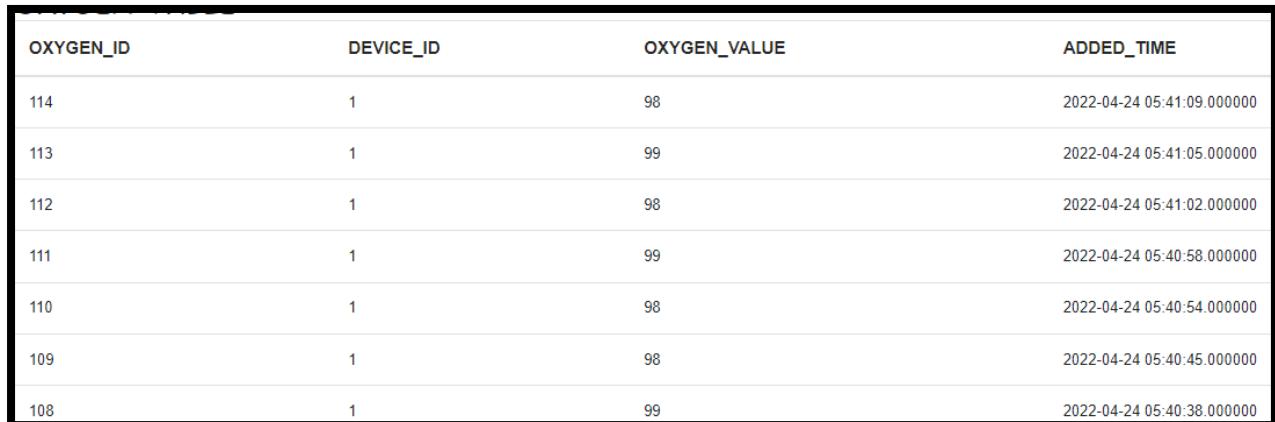
TEMP_ID	DEVICE_ID	TEMP_VALUE	ADDED_TIME
147	1	89	2022-04-14 06:26:43.000000
146	1	89	2022-04-14 06:26:38.000000
145	1	89	2022-04-14 06:26:33.000000
144	1	88	2022-04-14 06:26:28.000000
143	1	89	2022-04-14 06:26:23.000000
142	1	89	2022-04-14 06:26:18.000000
141	1	89	2022-04-14 06:26:13.000000
140	1	89	2022-04-14 06:26:09.000000
139	1	89	2022-04-14 06:26:04.000000
138	1	89	2022-04-14 06:25:59.000000
137	1	88	2022-04-14 06:25:54.000000

Fig. 9.5



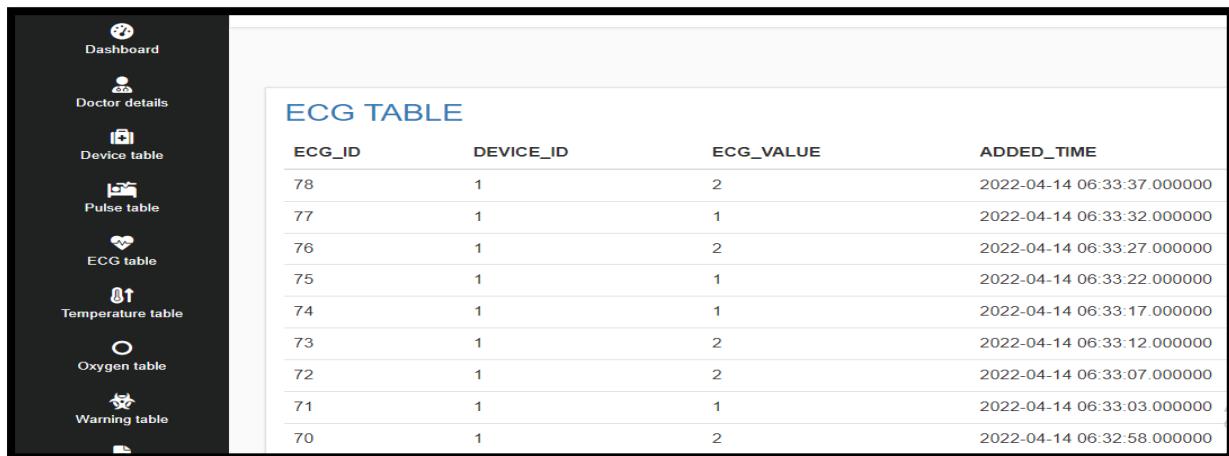
PULSE TABLE			
PULSE_ID	DEVICE_ID	PULSE_VALUE	ADDED_TIME
74	1	68	2022-04-15 06:18:16.000000
73	1	68	2022-04-15 06:17:54.000000
72	1	68	2022-04-15 06:17:47.000000
71	1	68	2022-04-15 06:17:40.000000
70	1	68	2022-04-15 06:17:33.000000
69	1	68	2022-04-15 06:17:25.000000
68	1	68	2022-04-15 06:17:18.000000
67	1	68	2022-04-15 06:17:09.000000
66	1	68	2022-04-15 06:17:02.000000

Fig. 9.6



OXYGEN_ID	DEVICE_ID	OXYGEN_VALUE	ADDED_TIME
114	1	98	2022-04-24 05:41:09.000000
113	1	99	2022-04-24 05:41:05.000000
112	1	98	2022-04-24 05:41:02.000000
111	1	99	2022-04-24 05:40:58.000000
110	1	98	2022-04-24 05:40:54.000000
109	1	98	2022-04-24 05:40:45.000000
108	1	99	2022-04-24 05:40:38.000000

Fig. 9.7



The image shows a mobile application interface. On the left is a dark sidebar with white icons and text labels: Dashboard, Doctor details, Device table, Pulse table, ECG table, Temperature table, Oxygen table, and Warning table. The main content area has a white background and a black border. It contains a table titled "ECG TABLE" with four columns: ECG\_ID, DEVICE\_ID, ECG\_VALUE, and ADDED\_TIME. The data in the table is as follows:

ECG_ID	DEVICE_ID	ECG_VALUE	ADDED_TIME
78	1	2	2022-04-14 06:33:37.000000
77	1	1	2022-04-14 06:33:32.000000
76	1	2	2022-04-14 06:33:27.000000
75	1	1	2022-04-14 06:33:22.000000
74	1	1	2022-04-14 06:33:17.000000
73	1	2	2022-04-14 06:33:12.000000
72	1	2	2022-04-14 06:33:07.000000
71	1	1	2022-04-14 06:33:03.000000
70	1	2	2022-04-14 06:32:58.000000

## 10. Notification

### 10.1 Messaging

- Directly from a python script we can pass messages to several users for this we need a service provider that provides its server and we can pass messages from them to several users
- In that we need to write the API key and other needed things and then we can send the messages

Code:-

```
import requests

url = "https://www.fast2sms.com/dev/bulkV2"

mobile = input('Enter Mobile Number: ')
message = input('Enter Message: ')

querystring = {
    "authorization": "YOUR_API_KEY",
    "sender_id": "TXTIND",
    "message": message,
    "route": "v3",
    "numbers": mobile}

headers = {
    'cache-control': "no-cache"
}

response = requests.request("GET", url, headers=headers, params=querystring)

print(response.text)
```

Fig. 10.1

### 10.2 Email

- For email we need to update the security settings and get the app activation key after that we can use the python scripts to email to many people at a time using gmail as a service provider.

Code:-

```
import smtplib

gmail_user = ""
gmail_password = ""

useremail = input("enter email")
mail = input("enter mail content")

sent_from = gmail_user
to = [useremail]
body = mail

email_text = mail

try:
    smtp_server = smtplib.SMTP_SSL('smtp.gmail.com', 465)
    smtp_server.ehlo()
    smtp_server.login(gmail_user, gmail_password)
    smtp_server.sendmail(sent_from, to, email_text)
    smtp_server.close()
    print ("Email sent successfully!")
except Exception as ex:
    print ("Something went wrong....",ex)
```

Fig. 10.2

## 11. Conclusion and Discussion

### 11.1 Overall analysis

Overall, the project was a good opportunity for the intern to learn how industry standards works and which way is better than which. There are many technology out there doing same things. Choosing the one which best fits for you is also the main task that is being taught to the student. Apart from the learning experience interns are also able to learn about the importance of communication, that it is the main key in development of any project, and is more important than code itself. You may be able to create a masterpiece, but if you are not able to communicate that thing to others, then it is not sufficient.

### 11.2 Problem Encounters and Possible Solutions

There are small areas where some of difficulties arise. One such instance was when we are using pulse-oximeter sensor we made the circuit but we are unable to code that as we have to use two ESP8266 and we have made two codes regarding to master and slave circuit.

Than my external guide helped us in that and we learned the coding of the master slave circuit in the Arduino IDE.

Addition to that there are some major problems also that at this time if we make this product and if there will be a big order from any giant company than how will we manage the production as there will be shortage of chips happening all around the world so the company decided to be in touch with multiple suppliers of the chips and this way the problem is being solved

### 11.3 Summary of Internship

The internship was of 3 months and student can extend it to his will. These 3 months are full of knowledge and were a great experience at InfoLabz IT Services.

I got a chance for converting my acquired theoretical knowledge into practical industry knowledge in these 3 months. The making of this project gave me a much-needed confidence to be able to compete in this up surging market of IT Industries.

#### **11.4 Limitation and Future Enhancement**

As years passes by we have to face different types of diseases as we are already fighting with the Covid 19 there are many variations coming out from that so in all that situation this project will be helpful and needs to be updated according to the future needs as we have much importance of oxygen level in the case covid 19 we have implemented in near future any disease have another parameter importance so we can add that feature in our project too.

## REFERENCES

- 1 <https://pubmed.ncbi.nlm.nih.gov/35079705/#:~:text=A%20smart%20health%20monitoring%20system,and%20temperature%20of%20a%20person.>
- 2 <https://pubmed.ncbi.nlm.nih.gov/33063046/>
- 3 <https://www.hindawi.com/journals/cmmm/2021/8591036/>
- 4 <https://www.sciencedirect.com/science/article/pii/S2214785321045545>
- 5 <https://www.programiz.com/python-programming>
- 6 <https://www.python.org/about/gettingstarted/>
- 7 <https://www.geeksforgeeks.org/python-programming-language/>
- 8 <https://www.tutorialspoint.com/python/index.htm>
- 9 <https://data-flair.training/blogs/best-books-for-numpy-scipy/>
- 10 <https://www.amazon.in/Python-Data-Analysis-Wrangling-Ipython/dp/9352136411>
- 11 <https://www.djangoproject.com/>
- 12 [https://www.tutorialspoint.com/python\\_web\\_development\\_libraries/python\\_web\\_development\\_libraries\\_django\\_framework.htm](https://www.tutorialspoint.com/python_web_development_libraries/python_web_development_libraries_django_framework.htm)
- 13 <https://stackoverflow.com/>