

**L D College of Engineering, Ahmedabad**

**Department of Information Technology**

**Academic Year :- 2021-22**



**GTU B.E. 7th Semester**

**Name:Romil Zalavadiya**

**Subject / Topic : Python(Django)**

# **Python(Django) Internship Report**

## **Personal Details**

Name : Romil Zalavadiya  
College Name : L. D. College Of Engineering  
Degree : BE  
Semester : 7  
Github URL :  
[https://github.com/RomilZalavadiya/Akash\\_Internship.git](https://github.com/RomilZalavadiya/Akash_Internship.git)

## **Company Details**

Company Name : Akash Technolabs  
External Guide : Akash Padhiyar  
Training Duration : 25-05-2021 to 7-06-2021

## Index

<u>Index</u>	<u>Task Detail</u>	<u>Page Name</u>
1	Introduction to Organization	4
2	Simple Registration Form(HTML)	6
3	Python CRUD Operation using MySQL	16
4	Some basic tasks related to IF....ELSE (Conditional)	19
5	Do All tasks when have done in session(Operators)	25
6	Tasks Based on OOPS Concepts	30
7	No---Task(Introduction to Django)	36
8	Hello World(Django)	39
9	Page Navigation(Django)	41
10	Print Data entered in html form in another html page(Django)	44
11	CRUD Operation using database(MYSQL/SQLite)	49

# **INTRODUCTION TO ORGANIZATION**



## Akash Technolabs

K-6, Shree Krishna Center Above Crossword Library Mithakhali Six Road, Navrangpura Ahmedabad, Gujarat, India – 380009  
<https://akashtechnolabs.com/>

### Introduction

- It is a website development company ,with 11 years of experience ,which holds a reputed image among contemporaries. They always try to provide best services to their clients.
- Website development of their firm is completely unique and adorable as well as original. They care for the trust that their clients have in them and so they assure clients to keep their quality up to the mark. They work on various website development projects on an international level too.
- They also hire new developers and give them golden opportunity to show their work and talent.
- They also give SEO services and Social Media Marketing to their clients. At Akash Technolabs they have highly professional and experienced team of developers.

### What they do ? :

- Mobile Development
- Dynamic Website Development
- PHP Development
- Laravel Development
- Node JS Development
- Angular JS Development
- React Native Development

### On Which Technologies they work most:

- Python
- Angular
- Node Js
- React Native
- Android
- IOS and Flutter
- Php, MongoDB

## Day 1 :-

### What we have learned ?:

- We learned basic Introduction of python programming language. In this we got basic idea about python like it is high-level, interpreted, object-oriented as well as scripting language.
- We also learned the most important and amazing features of python which may be the reason behind of its success at current technology driven.
  1. Easy to learn
  2. Easy to read
  3. Easy to maintain
  4. Portable
  5. Free and open source and object oriented
  6. Extensible and Embeddable
- We also came to know about in which field and which big-tech companies are using python to make their service effective.
- Trainer show how to install python setup and add to environment variable and run first program of hello world using notepad++ and command prompt.
- Then we learned about IDEs and Pycharm which is most usable IDE for python development. We also learned to install it and run our program of hello world in IDE (Pycharm).

**Task:** Make Simple Registration form in html using div/table tags.

- To complete this task, using div tags

**Code:-**

```
<html>
<head>
<title>Student Registration Form</title>
</head>
<body>
<h3>STUDENT REGISTRATION FORM</h3>
```

```
<table align="center" cellpadding = "10">
<!---- First Name ----->
<tr>
<td>FIRST NAME</td>
<td><input type="text" name="First_Name" maxlength="30"/>
(max 30 characters a-z and A-Z)
</td>
</tr>
<!---- Last Name ----->
<tr>
<td>LAST NAME</td>
<td><input type="text" name="Last_Name" maxlength="30"/>
(max 30 characters a-z and A-Z)
</td>
</tr>Romil Zalavadiya
<!---- Date Of Birth ----->
<tr>
<td>DATE OF BIRTH</td>
<td>
<select name="Birthday_day" id="Birthday_Day">
<option value="-1">Day:</option>
<option value="1">1</option>
<option value="2">2</option>
<option value="3">3</option>
<option value="4">4</option>
<option value="5">5</option>
<option value="6">6</option>
<option value="7">7</option>
<option value="8">8</option>
<option value="9">9</option>
<option value="10">10</option>
<option value="11">11</option>
<option value="12">12</option>
<option value="13">13</option>
```

```
<option value="14">14</option>
<option value="15">15</option>
<option value="16">16</option>
<option value="17">17</option>
<option value="18">18</option>
<option value="19">19</option>
<option value="20">20</option>
<option value="21">21</option>
<option value="22">22</option>
<option value="23">23</option>
<option value="24">24</option>Romil Zalavadiya
<option value="25">25</option>
<option value="26">26</option>
<option value="27">27</option>
<option value="28">28</option>
<option value="29">29</option>
<option value="30">30</option>
<option value="31">31</option>
</select>
<select id="Birthday_Month" name="Birthday_Month">
<option value="-1">Month:</option>
<option value="January">Jan</option>
<option value="February">Feb</option>
<option value="March">Mar</option>
<option value="April">Apr</option>
<option value="May">May</option>
<option value="June">Jun</option>
<option value="July">Jul</option>
<option value="August">Aug</option>
<option value="September">Sep</option>
<option value="October">Oct</option>
<option value="November">Nov</option>
<option value="December">Dec</option>
</select>
```

```
<select name="Birthday_Year" id="Birthday_Year">
<option value="-1">Year:</option>
<option value="2012">2012</option>
<option value="2011">2011</option>
<option value="2010">2010</option>
<option value="2009">2009</option>
<option value="2008">2008</option>
<option value="2007">2007</option>Romil Zalavadiya
<option value="2006">2006</option>
<option value="2005">2005</option>
<option value="2004">2004</option>
<option value="2003">2003</option>
<option value="2002">2002</option>
<option value="2001">2001</option>
<option value="2000">2000</option>
<option value="1999">1999</option>
<option value="1998">1998</option>
<option value="1997">1997</option>
<option value="1996">1996</option>
<option value="1995">1995</option>
<option value="1994">1994</option>
<option value="1993">1993</option>
<option value="1992">1992</option>
<option value="1991">1991</option>
<option value="1990">1990</option>
<option value="1989">1989</option>
<option value="1988">1988</option>
<option value="1987">1987</option>
<option value="1986">1986</option>
<option value="1985">1985</option>
<option value="1984">1984</option>
<option value="1983">1983</option>
<option value="1982">1982</option>
<option value="1981">1981</option>
```

```
<option value="1980">1980</option>
</select>
</td>
</tr>
<!---- Email Id ----->
<tr>Romil Zalavadiya
<td>EMAIL ID</td>
<td><input type="text" name="Email_Id" maxlength="100"
/></td>
</tr>
<!---- Mobile Number -----
--->
<tr>
<td>MOBILE NUMBER</td>
<td>
<input type="text" name="Mobile_Number" maxlength="10"
/>
(10 digit number)
</td>
</tr>
<!---- Gender ----->
<tr>
<td>GENDER</td>
<td>
Male <input type="radio" name="Gender" value="Male" />
Female <input type="radio" name="Gender" value="Female" />
</td>
</tr>
<!---- Address ----->
<tr>
<td>ADDRESS <br /><br /><br /></td>
<td><textarea name="Address" rows="4"
cols="30"></textarea></td>
</tr>
```

```
<!---- City ----->
<tr>
<td>CITY</td>Romil Zalavadiya
<td><input type="text" name="City" maxlength="30" />
(max 30 characters a-z and A-Z)
</td>
</tr>
<!---- Pin Code ----->
<tr>
<td>PIN CODE</td>
<td><input type="text" name="Pin_Code" maxlength="6" />
(6 digit number)
</td>
</tr>
<!---- State ----->
<tr>
<td>STATE</td>
<td><input type="text" name="State" maxlength="30" />
(max 30 characters a-z and A-Z)
</td>
</tr>
<!---- Country ----->
<tr>
<td>COUNTRY</td>
<td><input type="text" name="Country" value="India"
readonly="readonly" /></td>
</tr>
<!---- Hobbies ----->
<tr>
<td>HOBBIES <br /><br /><br /></td>
<td>
DrawingRomil Zalavadiya
<input type="checkbox" name="Hobby_Drawing"
value="Drawing" />
```

Singing

```
<input type="checkbox" name="Hobby_Singing"  
value="Singing" />
```

Dancing

```
<input type="checkbox" name="Hobby_Dancing"  
value="Dancing" />
```

Sketching

```
<input type="checkbox" name="Hobby_Cooking"  
value="Cooking" />
```

```
<br />
```

Others

```
<input type="checkbox" name="Hobby_Other" value="Other">  
<input type="text" name="Other_Hobby" maxlength="30" />  
</td>  
</tr>
```

----- Qualification-----

```
>
```

```
<tr>
```

```
<td>QUALIFICATION <br /><br /><br /><br /><br /><br /><br /><br /><br />
```

```
</td>
```

```
<td>
```

```
<table>
```

```
<tr>
```

```
<td align="center"><b>Sl.No.</b></td>
```

```
<td align="center"><b>Examination</b></td>
```

```
<td align="center"><b>Board</b></td>
```

```
<td align="center"><b>Percentage</b></td>
```

```
<td align="center"><b>Year of Passing</b></td>
```

```
</tr>Romil Zalavadiya
```

```
<tr>
```

```
<td>1</td>
```

```
<td>Class X</td>
```

```
<td><input type="text" name="ClassX_Board" maxlength="30" /></td>
```

```
<td><input type="text" name="ClassX_Percentage"
maxlength="30" /></td>
<td><input type="text" name="ClassX_YrOfPassing"
maxlength="30" /></td>
</tr>
<tr>
<td>2</td>
<td>Class XII</td>
<td><input type="text" name="ClassXII_Board"
maxlength="30" /></td>
<td><input type="text" name="ClassXII_Percentage"
maxlength="30" /></td>
<td><input type="text" name="ClassXII_YrOfPassing"
maxlength="30" /></td>
</tr>
<tr>
<td>3</td>
<td>Graduation</td>
<td><input type="text" name="Graduation_Board"
maxlength="30" /></td>
<td><input type="text" name="Graduation_Percentage"
maxlength="30" /></td>
<td><input type="text" name="Graduation_YrOfPassing"
maxlength="30" /></td>
</tr>
<tr>
<td>4</td>Romil Zalavadiya
<td>Masters</td>
<td><input type="text" name="Masters_Board"
maxlength="30" /></td>
<td><input type="text" name="Masters_Percentage"
maxlength="30" /></td>
<td><input type="text" name="Masters_YrOfPassing"
maxlength="30" /></td>
```

```
</tr>
<tr>
<td></td>
<td></td>
<td align="center">(10 char max)</td>
<td align="center">(upto 2 decimal)</td>
</tr>
</table>
</td>
</tr>
<!---- Course ----->
<tr>
<td>COURSES<br />APPLIED FOR</td>
<td>
BCA
<input type="radio" name="Course_BCA" value="BCA">
B.Com
<input type="radio" name="Course_BCom" value="B.Com">
B.Sc
<input type="radio" name="Course_BSc" value="B.Sc">
B.A
<input type="radio" name="Course_BA" value="B.A">
</td>
</tr>Romil Zalavadiya
<!---- Submit and Reset ----->
<tr>
<td colspan="2" align="center">
<input type="submit" value="Submit">
<input type="reset" value="Reset">
</td>
</tr>
</table>
</form>
</body> </html>
```

**STUDENT REGISTRATION FORM**

FIRST NAME	<input type="text"/> (max 30 characters a-z and A-Z)
LAST NAME	<input type="text"/> (max 30 characters a-z and A-Z)
DATE OF BIRTH	Day: <input type="text"/> Month: <input type="text"/> Year: <input type="text"/>
EMAIL ID	<input type="text"/>
MOBILE NUMBER	<input type="text"/> (10 digit number)
GENDER	Male <input type="radio"/> Female <input type="radio"/>
ADDRESS	<input type="text"/>
CITY	<input type="text"/> (max 30 characters a-z and A-Z)
PIN CODE	<input type="text"/> (6 digit number)
STATE	<input type="text"/> (max 30 characters a-z and A-Z)
COUNTRY	<input type="text"/> India

**STUDENT REGISTRATION FORM**

CITY	<input type="text"/> (max 30 characters a-z and A-Z)																									
PIN CODE	<input type="text"/> (6 digit number)																									
STATE	<input type="text"/> (max 30 characters a-z and A-Z)																									
COUNTRY	<input type="text"/> India																									
HOBBIES	Drawing <input type="checkbox"/> Singing <input type="checkbox"/> Dancing <input type="checkbox"/> Sketching <input type="checkbox"/> Others <input type="checkbox"/> <input type="text"/>																									
QUALIFICATION	<table border="1"> <thead> <tr> <th>SL.No.</th> <th>Examination</th> <th>Board</th> <th>Percentage</th> <th>Year of Passing</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Class X</td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td>2</td> <td>Class XII</td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td>3</td> <td>Graduation</td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td>4</td> <td>Masters</td> <td><input type="text"/></td> <td><input type="text"/></td> <td><input type="text"/></td> </tr> </tbody> </table> (10 char max)      (upto 2 decimal)	SL.No.	Examination	Board	Percentage	Year of Passing	1	Class X	<input type="text"/>	<input type="text"/>	<input type="text"/>	2	Class XII	<input type="text"/>	<input type="text"/>	<input type="text"/>	3	Graduation	<input type="text"/>	<input type="text"/>	<input type="text"/>	4	Masters	<input type="text"/>	<input type="text"/>	<input type="text"/>
SL.No.	Examination	Board	Percentage	Year of Passing																						
1	Class X	<input type="text"/>	<input type="text"/>	<input type="text"/>																						
2	Class XII	<input type="text"/>	<input type="text"/>	<input type="text"/>																						
3	Graduation	<input type="text"/>	<input type="text"/>	<input type="text"/>																						
4	Masters	<input type="text"/>	<input type="text"/>	<input type="text"/>																						
COURSES APPLIED FOR	BCA <input type="radio"/> B.Com <input type="radio"/> B.Sc <input type="radio"/> B.A <input type="radio"/>																									

Submit  Reset

## Day 2 :-

### What we have learned ? :

- In today's session we get introduced to python variables, comments, and various data types and its methods.
- We learned mainly three ways to write down comments in python.
  1. Using (#)
  2. Using (")
  3. Using ("")
- Then we perform a task related to comments to make concept more clear practically.
- Then we learned how name and declared variable that is used to store data in memory.
- Then we learned how to assign multiple variable and single value to multiple variable in single and perform task related to this.
- Then we get to know about standard data types of python programming language.
- We get small introduction and simple programs of all these datatypes.
- Some of these data types are very important in day to day life of any python developer like List, Tuple, Dictionary.
- Then we learned how to take a input from user with for loop and list method (lst.append(element))
- Then we learned various methods of tuple data type of python like wise we perform practicals of dictionary and its methods

Task:

**Code:**

Code:-

```
print("Hello, World!")
x = 5
y = "Romil"
print(x)
print(y)
print(type(x))
print(type(y))
thislist = ["apple", "banana", "cherry"]
print(thislist)
print(thislist[1])
thislist[1] = "blackcurrant"
print(thislist)
thislist.append("orange")
print(thislist)
thislist.remove("apple")
print(thislist)
for x in thislist:
    print(x)
thistuple = ("apple", "banana", "cherry")
print(thistuple)
thisset = {"apple", "banana", "cherry"}print(thisset)
thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year": 1964
}
print(thisdict)
a = 33
b = 200
if b > a:
    print("b is greater than a")
i = 1
while i < 6:
    print(i)
```

```
i += 1
fruits = ["apple", "banana", "cherry"]
for x in fruits:
    print(x)
```

**Output:-**

```
Hello, World!
5
Romil
<class 'int'>
<class 'str'>
['apple', 'banana', 'cherry']
banana
['apple', 'blackcurrant', 'cherry']
['apple', 'blackcurrant', 'cherry', 'orange']
['blackcurrant', 'cherry', 'orange']
blackcurrant
cherry
orange
('apple', 'banana', 'cherry')
{'banana', 'cherry', 'apple'}
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
b is greater than a
1
2
3
4
5
apple
banana
cherry
```

## Day 3 :-

### What we have learned ?:

- In this session we are introduced to Input/Output functions input() and print() and their syntax.
- Then we learned Conditional Statements.
  1. If statement.
  2. If....else statement.
  3. If...elif....else statement.
  4. Nestef If statement.
- After that we practice some programs of all these conditional statements like check number is even or odd / +ve or -ve.
- After conditional statements we got introduced by for loop and perform task of print 1 to 100 using loop.
- We also learned how to use for loop for range() and lists.
- Before that we learned syntax of range() and for loop.
- Then we got to know about statements used in for loop like continue and break as well as pass.
- While learning them we learned terminologies of these statements for when and why we use these.
- Onward we also learned how to use loop with else. when condition of loop get to false else statement execute.

## Task:-

1. Calculate average of 5 numbers.
2. Check whether number is even or odd.
3. Take a year and check whether it is leap year or not
4. Take a number and check whether it is zero, positive or negative.
5. Take 2 numbers and display greatest number. (Also check equal number condition)
6. Take a number and find factorial of that number.
7. Write a program to swap 2 numbers using third variable.
8. Take 2 numbers and find smallest number.
9. Take a number check if a number is less than 100 or not. If it is less than 100 then check if it is odd or even.
10. Take a number to print the square of a number if it is less than 10.
11. Take a number and check whether it is zero, positive or negative using nested IF...ELSE statement .
12. Take 3 numbers and find greatest number using nested IF....ELSE statement.
13. Take 3 numbers and find smallest number using logical operator.
14. Write a program to swap 2 numbers without taking third variable.
15. Take starting number and ending number from the user and print following series.

30 27 24 21 18 15 12 9 6 3 0

```
In [2]: #1.Calculate average of 5 numbers.
```

```
total = 0
for i in range(5):
    a = int(input("Enter the number"))
    total = total + a
print("Total",total)
```

```
Enter the number4
Enter the number5
Enter the number6
Enter the number7
Enter the number8
Total 30
```

```
In [3]: #2.Check whether number is even or odd.
```

```
num = int(input("Enter the number"))

if num%2==0:
    print("Number is even")
else:
    print("Number is odd")
```

```
Enter the number3
Number is odd
```

```
In [6]: #3.Take a year and check whether it is leap year or not
```

```
year = int(input("Enter the year"))

if(year%4==0 and year%100!= 0 or year%400==0):
    print("The year is a leap year")
else:
    print("The year isn't a leap year")
```

```
Enter the year2000
The year is a leap year
```

```
In [8]: #4.Take a number and check whether it is zero, positive or negative.
```

```
a = int(input("Enter the number"))

if a<0:
    print("This number is negative")
elif a>0:
    print("This number is positive")
else:
    print("Given number is zero")
```

```
Enter the number0
Given number is zero
```

```
In [9]: #5.Take 2 numbers and display greatest number. (Also check equal number condition)
```

```
a = int(input("Enter 1st number"))
b = int(input("Enter 2nd number"))

if a>b:
    print("Greater number :",a)
elif b>a:
    print("Greater number :",b)
else:
    print("Both numbers are equal")
```

```
Enter 1st number23
Enter 2nd number45
Greater number : 45
```

```
In [25]: #6.Take a number and find factorial of that number.

num = int(input("Enter the number"))

if num < 0:
    print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
    print("The factorial of 0 is 1")
else:
    factorial=1
    for i in range(1,num + 1):
        factorial = factorial*i
    print("The factorial of",num,"is",factorial)

Enter the number5
The factorial of 5 is 120
```

```
In [26]: #7.Write a program to swap 2 numbers using third variable.

a = int(input("Enter the number"))
b = int(input("Enter the number"))

print("Given numbers a and b are:",a,b)

c=a
a=b
b=c

print("After sawapping numbers a and b are",a,b)

Enter the number4
Enter the number5
Given numbers a and b are: 4 5
After sawapping numbers a and b are 5 4
```

```
In [27]: #8.Take 2 numbers and find smallest number.

a = int(input("Enter 1st number"))
b = int(input("Enter 2nd number"))

if a>b:
    print("smaller number :",b)
elif b>a:
    print("smaller number :",a)
else:
    print("Both numbers are equal")

Enter 1st number5
Enter 2nd number7
smaller number : 5
```

```
In [28]: #9.Take a number check if a number is less than 100 or not. If it is less than 100 then check if it is odd or even.

num = int(input("Enter the number"))

if num>100:
    print("Given number is greter than 100")

else:
    if num%2==0:
        print("Given number is less than 100 and even")
    else:
        print("Given number is less than zero and odd")

Enter the number5
Given number is less than zero and odd
```

```
In [29]: #10. Take a number to print the square of a number if it is less than 10.

num = int(input("Enter the number"))

if num<100:
    print("Number is less than 100 and square of number is :",num*num)
else:
    print("Number is greater than 100")

Enter the number11
Number is less than 100 and square of number is : 121
```

```
In [30]: #11. Take a number and check whether it is zero, positive or negative using nested IF...ELSE statement .

num = float(input("Enter a number: "))
if num >= 0:
    if num == 0:
        print("Zero")
    else:
        print("Positive number")
else:
    print("Negative number")

Enter a number: 0
Zero
```

---

```
In [31]: #12. Take 3 numbers and find greatest number using nested IF...ELSE statement.

# input three integer numbers
a=int(input("Enter A: "))
b=int(input("Enter B: "))
c=int(input("Enter C: "))

# conditions to find largest
if a>b:
    if a>c:
        g=a
    else:
        g=c
else:
    if b>c:
        g=b
    else:
        g=c

# print the largest number
print("Greater = ",g)

Enter A: 4
Enter B: 5
Enter C: 6
Greater = 6
```

In [33]: #13. Take 3 numbers and find smallest number using logical operator.

```
def smallest(x, y, z):
    c = 0

    while ( x and y and z ):
        x = x-1
        y = y-1
        z = z-1
        c = c + 1

    return c

x=int(input("Enter A: "))
y=int(input("Enter B: "))
z=int(input("Enter C: "))

print("Minimum of 3 numbers is",
      smallest(x, y, z))
```

```
Enter A: 5
Enter B: 4
Enter C: 6
Minimum of 3 numbers is 4
```

In [34]: #14. Write a program to swap 2 numbers without taking third variable.

```
x=int(input("Enter x: "))
y=int(input("Enter y: "))

x = x + y
y = x - y
x = x - y
print("After Swapping: x =", x, " y =", y)
```

```
Enter x: 3
Enter y: 4
After Swapping: x = 4  y = 3
```

In [35]: #15. Take starting number and ending number from the user and print following series.

```
num = int(input("Enter stating number"))

while num>=0:
    print(num)
    print(" ")
    num=num-3
```

```
Enter stating number30
30
```

```
27
```

```
24
```

```
21
```

```
18
```

```
15
```

```
12
```

```
9
```

```
6
```

```
3
```

```
0
```

## Day 4 :-

### What we have learned ? :

- On this day we learned the most important feature of any programming language which is also extended in Python too which is “Functions”
- In today’s session we get to know about moto behind usage of function which is to perform specific task of any program.
- Then we learned syntax of defining function in python programming language.
- Any function is divided in small parts
  1. “def” keyword
  2. Function name
  - 3.(arguments)
  4. “.” symbol
  5. Function “body”.
  6. Return statement (optional)
- We also learned about three different types of arguments which can be provided in python function
  1. Default arguments
  2. Keyword arguments
  3. Variable-Length arguments
- We perform all types of arguments program to understand well and this helps us in remembering for long time.
- We also learned about non keyword arguments as well as keyword arguments where in non keyword we provide arguments without using argument name where as in keyword arguments we provide arguments name which is defined in calling function.
- Then after we get the information about Scope of variable which is mainly two typed 1
  1. Global Variable
  2. Local variable
- Variables that are defined inside a function body have a local scope, and those defined outside have a global scope.
- We also get the information about ‘indentation error: expected an indented block’
- After that we get to know about module functionality of python programming.
- That are used to break down big programs in small ones as well as make program or project organized.
- We can define our most used functions in a module and import it, instead of copying their definitions into different programs. We also perform example program.

- Then we dived into operator world and get information about how many type of various operators exists in python programming language.
  1. Arithmetic
  2. Comparison
  3. Logical
  4. Assignment
  5. Membership
  6. Identity
- Arithmetic operators are used to perform arithmetic computing(add, subtract, multiply, Division etc...)
- Logical operators(and,or,not) are used to accomplish logical tasks.
- Like wise we have done all program of these operators.

**Task:-** Perform all program performed in session

**Code:-**

```
def fact(n):  
    fac = 1  
    for i in range(n,1,-1):  
        fac = fac*i  
    return fac  
print("-----Operators-----")  
x=10  
y=6  
z=20  
lst = [10,20,30,40,50,60,'hello' , 'Guys']  
print("x:",x)  
print("y:",y)  
print('z:',z)  
print("lst:",lst)  
print("-----")  
print("<----Arithmetic operators---->")  
print("x+y=",x+y)  
print("x-y=",x-y)  
print("x*y=",x*y)  
print("x/y=",x/y)  
print("x//y=",x//y)  
print("x%y=",x%y)  
print("<----Comparision Operators---->")  
print(" x>y =",x>y)  
print(" x<y =",x<y)  
print(" x==y =",x==y)  
print(" x>=y =",x>=y)  
print(" x<=y =",x<=y)  
print(" x!=y =",x!=y)  
print("<----Logical Operators---->")  
print("----and----")
```

```
if x>y and x>z:  
print("x is the largest")  
if y>x and y>z:  
print("y is the largest")  
if(z>x and z>y):  
print("z is the largest")  
print("----or----")  
ch=input("enter char:")  
if(ch=='A' or ch=='a' or ch=='E' or ch=='e' or ch=='I' or ch=='i' or ch=='O' or  
ch=='o' or ch=='U' or ch=='u'):  
print(ch," is Vowel")  
else:  
print(ch," is consonant")  
print("<---Membership Operator--->")  
print("x in lst:",x in lst)  
print("y in lst:",y in lst)  
print("y not in lst:",y not in lst)  
print("<---Identity Operator--->")  
print("x is y:",x is y)  
print("x is not y:",x is not y)  
def function():  
print("hello")  
def functionarg(arg):  
print(arg)  
def returnfun():  
a=10  
b=20  
return "a+b={}".format(a+b)  
def mreturn():  
name = "LDCE"  
depart = "IT(7th sem)"  
return name,depart  
def default(a=10,b=20):  
return a+b  
def keyargs(a,b):  
return a-b  
def varlength(*num):  
obj=[]  
for i in num:  
obj.append(i)  
return obj  
def varlengthk(**keyargs):  
dis = {}  
for key,val in keyargs.items():  
dis[key] = val  
return dis  
def scope():  
x=10  
print("Value inside function : ",x)  
print("----simple function----")  
function()  
print("----function with arguments----")  
functionarg("Hello World")  
print("----function with return----")
```

```

print(returnfun())
print("---function with multiple return---")
name,depart = mreturn()
print("College = {}".format(name))
print("Department = {}".format(depart))
print("---Default arguments---")
print("default() :",default())
print("default(4,5) :",default(4,5))
print("---Keyword arguments---")
print("keyargs(a=10,b=20)",keyargs(a=10,b=20))
print("keyargs(b=10,a=20)",keyargs(b=10,a=20))
print("---Var-length(non-keyword) arguments---")
print("varlength(10,20) : ",varlength(10,20))
print("varlength(10,20,30) : ",varlength(10,20,30))
print("varlength(10,20,30,40) : ",varlength(10,20,30,40))
print("---Var-length(keyword) arguments---")
print('varlengthk(car="BMW",price=2500000) :::',varlengthk(car="BMW",price=2500000))
print('varlengthk(car="BMW",price=2500000,country="india") :::',varlengthk(car="BMW",price=2500000,country="india"))
print('---Scope of Variable---')
x=20
scope()
print("value outside function:",x)
print('---Module Function---')
print(fact(5))
print()
print()
print()

```

**Output:-**

```

-----Operators-----
x: 10
y: 6
z: 20
lst: [10, 20, 30, 40, 50, 60, 'hello', 'Guys']
-----Arithmetic operators-----
x+y= 16
x-y= 4
x*y= 60
x/y= 1.6666666666666667
x//y= 1
x%y= 4
-----Comparision Operators-----
x>y = True
x<y = False
x==y = False
x>=y = True
x<=y = False
x!=y = True
-----Logical Operators-----
----and----
z is the largest
----or----
enter char:r
r is consonant
-----Membership Operator-----
x in lst: True
y in lst: False
y not in lst: True
-----Identity Operator-----
x is y: False
x is not y: True

```

```
----or----
enter char:r
r is consonant
<----Membership Operator---->
x in lst: True
y in lst: False
y not in lst: True
<----Identity Opearator---->
x is y: False
x is not y: True
----simple function----
hello
----function with arguments----
Hello World
----function with return----
a+b=30
----function with multiple return---
College = LDCE
Department = IT(7th sem)
---Default arguments---
default() : 30
default(4,5) : 9
---Keyword arguments---
keyargs(a=10,b=20) -10
keyargs(b=10,a=20) 10
---Var-length(non-keyword) arguments---
varlength(10,20) : [10, 20]
varlength(10,20,30) : [10, 20, 30]
varlength(10,20,30,40) : [10, 20, 30, 40]
---Var-length(keyword) arguments---
varlengthk(car="BMW",price=2500000) ::: {'car': 'BMW', 'price': 2500000}
varlengthk(car="BMW",price=2500000,country="india") ::: {'car': 'BMW',
'price': 2500000, 'country': 'india'}
----Scope of Variable----
Value inside function : 10
value outside function: 20
----Module Function----
120
```

## Day 5:-

### What We have learned ? :

- In today's session of python for Django we learned about class concept in object oriented programming language.
- Firstly we learned 'what is class and why it is so important for any OOP languages?'. Which help us to make ourselves comfortable before diving more deeply in it.
- Then we learned how to define any class in python and basic syntax of defining class in python programming language.

#### **Syntax: "class MyClass:"**

- Classes are mostly used to contain data field to store the data and defining various useful methods'.
- Then we learned how to access class field like variables and it defined methods to perform any according tasks. This requirement is fulfilled by Object of that class which is also known as instance of class which provide access for any element or method of that related class.

#### **Syntax: "object = MyClass()"**

- Then we perform our first program of this session related to class to understand well practically.
- Then we differentiate method and function and understand what are various difference between methods and functions.
- Then we get to know about 'self' argument which are mostly used in method of class call initializer this method is also known as '\_init\_' method its work is to initialize the variable of class.
- This\_init\_method is also called constructor of class. There are mainly two type of constructor in python.

##### **1. Default Constructor**

##### **2. Parameterized Constructor**

- Then we learned how to use and when to use these above mentioned constructors by taking one example.
- Then we got introduced to the most important and enrich concept of OOP known as 'INHERITANCE' and its various types. It allows user to make general class and then extend that class in more specialized class (parent-child class concept).

#### **Syntax: class Subclass(Superclass):**

**#body**

- Types:
  1. Single-Level Inheritance
  2. Multi-level Inheritance
  3. Multiple Inheritance
  4. Hierarchical Inheritance
  5. Hybrid Inheritance
- The we learned these types of inheritance deeply with example of each type which help use to make understand very well and conceptual way.
- Then we learned 2<sup>nd</sup> most important topic of OOP called '**Polymorphism**'. Which is ability to use common interfaces for multiple form
  1. **Overriding Methods**
  2. **Overloading Methods**
- We performed some example related to both type of polymorphism. And dive into base of OOP.

**Task :We are given several task based on class and inheritance here are their output**

```
In [19]: #1.Create a class cal1 that will calculate sum of three numbers. Create
#setdata() method which has three parameters that contain numbers.
#create display() method that will calculate sum and display sum.

class cal1:
    def setdata(self,a,b,c):
        self.a=a
        self.b=b
        self.c=c

    def display(self):
        sum=self.a+self.b+self.c
        print(sum)

a=cal1()
a.setdata(5,3,6)
a.display()
```

14

```
In [20]: #2. Create a class cal2 that will calculate area of a circle. Create set
data()
#method that should take radius from the user. Create area() method
#that will calculate area . Create display() method that will display ar
ea .

class cal2:
    def setdata(self,r):
        self.r=float(input("Enter the radias"))

    def area(self):
        self.area=3.14*3.14*self.r

    def display(self):
        print("Area of the circle is: ",self.area)

a=cal2()
a.setdata(4)
a.area()
a.display()
```

```
Enter the radias5
Area of the circle is:  49.298
```

```
In [23]: #3. Create a class cal3 that will calculate simple interest. Create
#constructor method which has three parameters .Create calInterest()
#method that will calculate Interest . Create display() method that will
#display Interest.

class cal3:
    def __init__(self,p,r,t):
        self.p=p
        self.r=r
        self.t=t

    def calInterest(self):
        self.Interest=self.p*self.r*self.t

    def display(self):
        print(self.Interest)

a = cal3(1000,3,1)
a.calInterest()
a.display()

3000
```

```
In [28]: #4. Create a class cal4 that will calculate square of a number. Create
#setData() method which has one parameters that contain number.
#Create display() method that will calculate sum. (Function should
#return value)

class cal4:
    def setData(self,a):
        self.a=a

    def display(self):
        self.area=self.a*self.a
        return self.area

a=cal4()
a.setData(4)
a.display()
```

Out[28]: 16

```
In [41]: #5. Consider an employee class, which contains fields such as name and
#designation. And a subclass, which contains a field salary. Write a
#program for inheriting this relation.

class employee:
    def __init__(self,name,designation):
        self.name=name
        self.designation=designation

    def print(self):
        print(self.name,self.designation)

class salary(employee):
    def __init__(self,name,designation,salary):
        super().__init__(name,designation)
        self.salary=salary

a=salary("Romil","HeadProgrammer",200000)

a.print()
a.salary
```

Romil HeadProgrammer

Out[41]: 200000

```
In [44]: #6. Create a class cal5 that will calculate area of a rectangle. Create
#constructor method which has two parameters .Create calArea()
#method that will calculate area of a rectangle. Create display() method
#that will display area of a rectangle.

class cal5:
    def __init__(self,length,width):
        self.length=length
        self.width=width

    def calArea(self):
        self.area=self.length*self.width

    def display(self):
        print("Area=",self.area)

a=cal5(5,4)
a.calArea()
a.display()
```

Area= 20

```
In [1]: #7. Create a class cal6 that will calculate area of a square. Create set  
#data()  
#method that should take length from the user. Create area() method  
#that will calculate area . Create display() method that will display ar  
ea .  
  
class cal6:  
    def setdata(self):  
        self.l = float(input("Enter the lenght"))  
  
    def area(self):  
        self.area=self.l*self.l  
  
    def display(self):  
        print(self.area)  
  
a=cal6()  
a.setdata()  
a.area()  
a.display()  
  
Enter the lenght4  
16.0
```

```
In [3]: #8. Write a program with use of inheritance: Define a class publisher th  
at  
#stores the name of the title. Derive two classes book and tape, which  
#inherit publisher. Book class contains member data called page no and  
#tape class contain time for playing. Define functions in the appropriat  
e  
#classes to get and print the details.  
  
class publisher:  
    def title(self):  
        self.title = input("Enter the name of the book")  
  
    def print_title(self):  
        print(self.title)  
  
class book(publisher):  
    def pages(self):  
        self.pages = int(input("Enter the no of pages"))  
  
    def print_pages(self):  
        print(self.pages)  
  
class time(book):  
    def time(self):  
        self.time = float(input("Enter the time required for playing"))  
  
    def print_time(self):  
        print(self.time)  
  
a = time()  
a.title()  
a.print_title()  
a.pages()  
a.print_pages()  
a.time()  
a.print_time()  
  
Enter the name of the bookAtomic Habits  
Atomic Habits  
Enter the no of pages233  
233  
Enter the time required for playing122  
122.0
```

```
In [6]: #9. Create a class called scheme with scheme_id, scheme_name, outgoing_rate, and message_charge. Derive customer class from scheme and include cust_id, name and mobile_no. Define necessary functions to read and display data.

class scheme:
    def scheme_info(self):
        self.scheme_id=int(input("Enter scheme id"))
        self.scheme_name=input("Enter scheme name")
        self.outgoing_rate=float(input("Enter outgoing rate"))
        self.message_charge=int(input("Enter message charges"))

class customer(scheme):
    def cust_info(self):
        self.cust_id=int(input("Enter customer id"))
        self.name=input("Enter customer name")
        self.mobile_no=int(input("Enter mobile number"))

    def display(self):
        print("scheme id:",self.scheme_id)
        print("scheme name:",self.scheme_name)
        print("outgoing rate:",self.outgoing_rate)
        print("Message charges:",self.message_charge)
        print("Customer id:",self.cust_id)
        print("Customer name:",self.name)
        print("Mobile number:",self.mobile_no)

a=customer()
a.scheme_info()
a.cust_info()
a.display()

Enter scheme id2
Enter scheme namexyz
Enter outgoing rate1233
Enter message charges12
Enter customer id3
Enter customer nameromil
Enter mobile number8767656564
scheme id: 2
scheme name: xyz
outgoing rate: 1233.0
Message charges: 12
Customer id: 3
Customer name: romil
Mobile number: 8767656564
```

```
In [18]: #10.Create a arith class. The class should have a parameterized constructor and methods to add, subtract and multiply two numbers and to return the answers

class arith:
    def __init__(self,num1,num2):
        self.num1=int(num1)
        self.num2=int(num2)

    def add(self):
        self.sum=self.num1+self.num2
        print(self.sum)

    def subtract(self):
        self.sub=self.num1-self.num2
        print(self.sub)

    def multiply(self):
        self.mul=self.num1*self.num2
        print(self.mul)

a=arith(5,4)
a.add()
a.subtract()
a.multiply()

9
1
20
```

## Day 6:-

### What we have learnt?

- From Today's session we started our main agenda of this amazing internship program which is **Django**.
- Today we didn't begin programming. This was just theoretical session of python framework Django.
- We begin with little introduction of all web based python frameworks some of them are following
  - 1. [Django](#)
  - 2. [Web2py](#)
  - 3. [Flask](#)
  - 4. [Tornado](#)
  - 5. [Cherrypy](#) and many more ...
- Then we dived in to our main topic Django deeply and understand
  - 1. what it is?
  - 2. Why it is so popular in market?
  - 3. Which are the amazing features those Django provide?
- We learn moto of Django framework or we can say we learn principle behind Django which is DRY (Don't Repeat Yourself)
- We also get to know about Django has an inbuilt supportive library for multiple databases,
  - 1. MySQL
  - 2. PostgreSQL
  - 3. SQLite3
  - 4. Oracle
- Then we are got aware about what we are going to learn about Django in this internship like Internship Highlights
  - 1. Authentication support
  - 2. Database schema migrations
  - 3. Object-relational mapper (ORM)
  - 4. Support for web servers
  - 5. Template engine
  - 6. URL routing
- Then we suddenly jumped into history of Django like who made it and how they

named this framework “Django” and we also gathers info about version time line of Django which shows version and its release dates.

- Features:
  1. It's fast and simple
  2. Open Source
  3. It's secure
  4. It suits any web application project
  5. It's well-established
- After that we get an information about which companies are using Django in this contemporary world to grow their business and marketing
- Then we learned MVT structure which are used by Django to develop any Django project in organized way.

### **Model**

- Defines the data structure.
- Takes care for querying the database.

### **View**

- Defines what data should be presented
- Returns HTTP response

### **Template**

- Renders the data in suitable format – HTML/XML/etc...
- After that we know what are prerequisites of learning Django which include basic of python and its functionalities.
- Then we choose our code editor(VS Code) for programming in Django and installed it.
- Then using ‘pip’ we install Django into our system but before that we set our python path to environment variable.
- Then we learn about ‘pip’ command also called ‘Pip Installs Packages’.
- Then we learned some basic command like how to download, upgrade, and uninstall any python library using “pip” command.
- We also learned how to find version of any installed python libraries using “pip”. As well as we learned how to see all installed python libraries in our system.
- Then we visited [Django official website](#) where all documentation of Django is already there.
- Then we learned how to start Django project using Django-admin startproject command.

- This command will generate some base file of Django in your working directory.  
Then we run our first program of Django in this internship using  
**python manage.py runserver.**
- Then we understood all file provided inbuilt by Django one by one .And learned  
Django project life cycle.
- We run program of Django in VS Code as well as Pycharm.

## Day 7:-

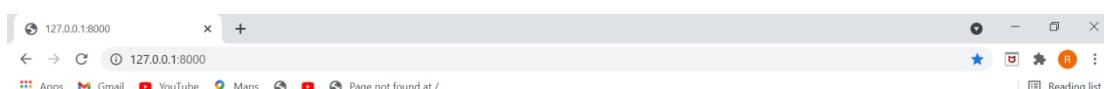
### What we have learnt?

- Our today's was completely practical based on Django framework. We first created new Django project using **Django-admin startproject** command and again revised file structure and standard flow of Django project.
- Then we learned about makemigrations in Django which is mainly used to migrate created model and tables to connected database.
- For today's session we used SQLite database and to see database we used SQLite viewer Software.
- Then we created our first app in our Django project using "**Python manage.py startapp myapp**".
- Then we configure all requirements for start our project on local host using "**python manage.py runserver**".
- Then we created our first function in views.py to redirect project to HttpResponse and run project to chrome browser and print hello world.

The screenshot shows the Visual Studio Code interface with a Django project named 'hello\_world'. The Explorer sidebar on the left lists files and folders: 'OPEN EDITORS 1 UNSAVED' (urls.py), 'HELLO\_WORLD' (admin.py, apps.py, migrations, settings.py, urls.py, views.py), and 'hi\_world' (\_\_init\_\_.py, migrations, \_\_pycache\_\_, \_\_init\_\_.py). The 'views.py' tab in the top bar is active, displaying the following Python code:

```
from django.shortcuts import render,HttpResponse
# Create your views here.
def world(request):
    return HttpResponse("<h1>Hello World.</h1><h2>Welcome to AkashTechnolabs</h2>")
```

The bottom status bar shows 'Python 3.8.3 64-bit' and other system details like 'Ln 6, Col 42' and '23:11 13-06-2021'.



Hello World.  
Welcome to AkashTechnolabs



## Day 8:-

## What we have learnt?

- In today's session we learned how to navigate through multiple pages using various urls and its associated functions of views.py.
  - We also learned how to setup path for templates and static file which are most useful features of Django we import images for static folder using different syntax of jinja2 than regular html syntax for importing CSS,JS, Images files into html page.

File Edit Selection View Go Run Terminal Help base.html - mysite - Visual Studio Code

EXPLORER

OPEN EDITORS

- index.html templates
- base.html templates

MYSITE

- migrations
- \_init\_.py
- admin.py
- apps.py
- models.py
- tests.py
- urls.py
- views.py

mysite

- \_pycache\_
- \_init\_.py
- asgi.py
- settings.py

urls.py

wsgi.py

templates

- about.html
- base.html
- contact.html
- index.html

db.sqlite3

manage.py

OUTLINE

index.html X base.html X

templates > base.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <meta http-equiv="X-UA-Compatible" content="IE=edge">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <title>Document</title>
8     <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" in
9       <style>
10         img{
11           height: 92vh;
12         }
13         .d-none h1{
14           font-size: 100px;
15         }
16       </style>
17     </head>
18   <body>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

(c) Microsoft Corporation. All rights reserved.

C:\Users\romil\Desktop\Aakash\mysite>python manage.py runserver  
Watching for file changes with StatReloader  
Performing system checks...

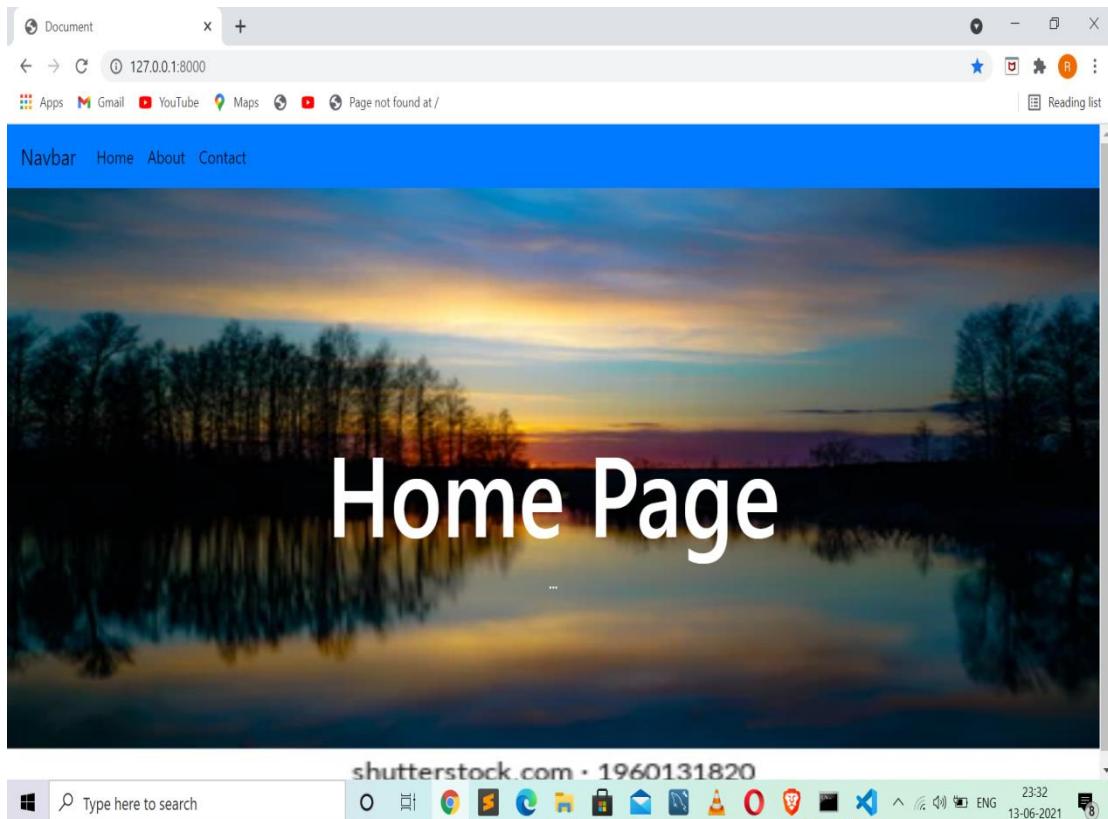
System check identified no issues (0 silenced).  
June 13, 2021 - 23:18:57  
Django version 3.2.4, using settings 'mysite.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.

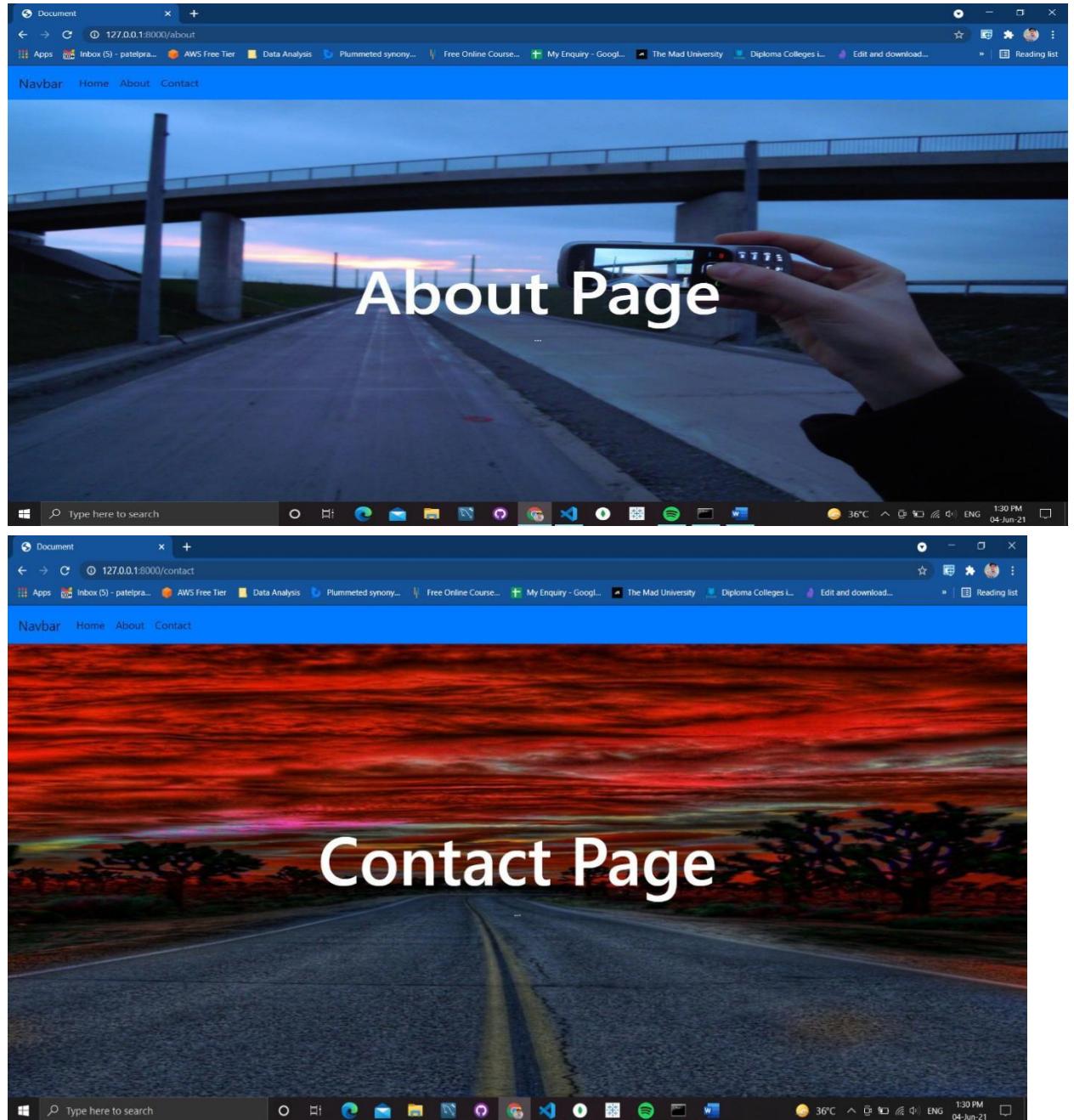
Python 3.8.3 64-bit 0 0 ▲ 0

Ln 1, Col 1 Spaces: 2 UTF-8 LF HTML R

23:33 13-06-2021

Type here to search

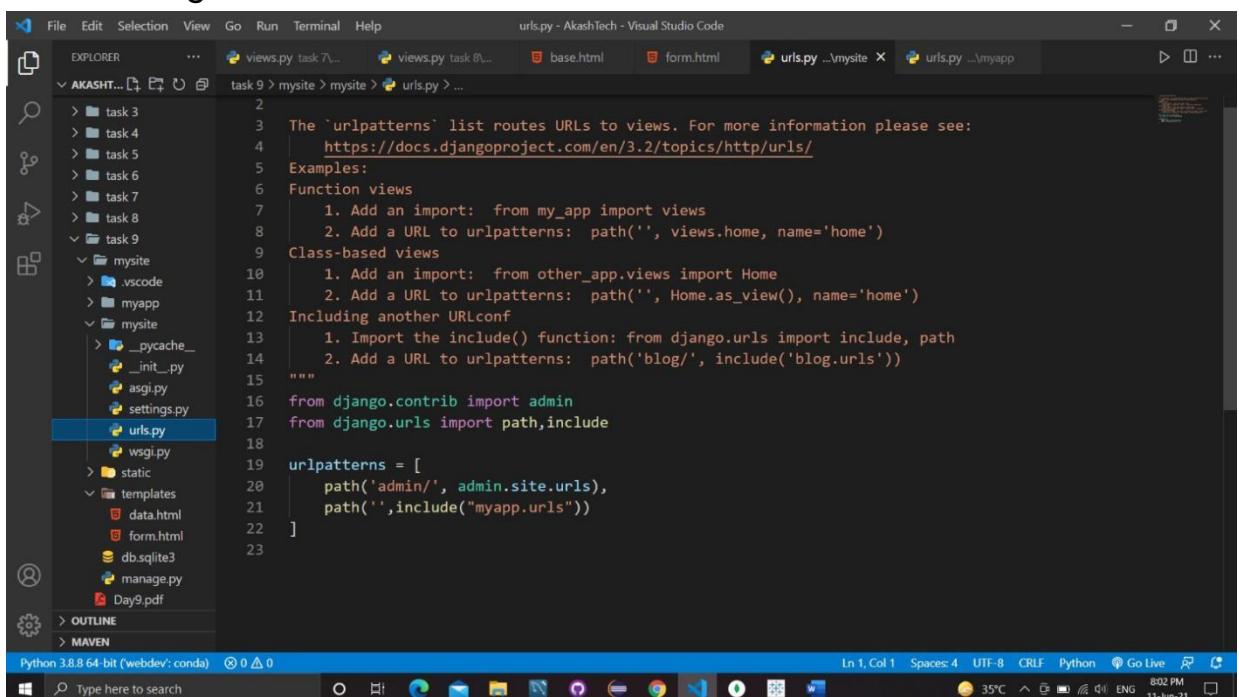




## Day 9:-

### What we have learnt?

- This session was also completely based on practical. In this session we learned how to get data from html form and print them in console as well as transfer them to another html page using “context” variable which is passed through render() which used to render template.
- Then we learned a bit about { % csrf\_token % }. The **CSRF token** only ensures that only forms that have originated from trusted domains can be used to POST data back.
- Then lecturer made basic form with some fields and showed us how's that working in real life.



The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "AKASHTECH...". The "mysite" directory contains "task 3", "task 4", "task 5", "task 6", "task 7", "task 8", "task 9", and "mysite" (which further contains ".vscode", "myapp", and "mysite" sub-directories). Inside "mysite", there are files: ".pycache\_.\_\_init\_\_.py", "asgi.py", "settings.py", "urls.py" (selected), and "wsgi.py". Other files like "static", "templates", "data.html", "form.html", "db.sqlite3", and "manage.py" are also listed.
- Code Editor:** Displays the content of "urls.py". The code defines the URL patterns for the application, including imports for views, URLs, and admin, and includes the "myapp.urls" for the blog section.
- Bottom Status Bar:** Shows system information: Python 3.8.8 64-bit ('webdev': conda), 0 errors, 0 warnings, 0 info, 0 notices, 0 deprecations, 0 criticals, 35°C, 802 PM, 11-Jun-21, and various system icons.

```

2
3 The `urlpatterns` list routes URLs to views. For more information please see:
4     https://docs.djangoproject.com/en/3.2/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 Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 Including another URLconf
13    1. Import the include() function: from django.urls import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path,include
18
19 urlpatterns = [
20     path('admin/', admin.site.urls),
21     path('',include("myapp.urls"))
22 ]

```

urls.py - AkashTech - Visual Studio Code

```

1 from django.urls import path
2 from . import views
3 urlpatterns = [
4     path("",views.form,name="index"),
5 ]

```

views.py - AkashTech - Visual Studio Code

```

1 From django.shortcuts import render
2 # Create your views here.
3
4 def form(request):
5
6     if request.method == 'POST':
7         fn = request.POST.get("fn")
8         ln = request.POST.get("ln")
9         email = request.POST.get("email")
10        gender = request.POST.get("gender")
11        contact= request.POST.get("contact")
12        address= request.POST.get("address")
13        zipcode = request.POST.get("zipcode")
14        college = request.POST.get("college")
15        dept = request.POST.get("dept")
16
17        context={
18            'fn':fn,
19            'ln':ln,
20            'email':email,
21            'gender':gender,
22            'contact':contact,
23            'address' :address,
24            'zipcode':zipcode,
25            'college':college,
26        }

```

The screenshot displays two instances of Visual Studio Code side-by-side, both running on Python 3.8.8 64-bit (webdev: conda).

**Top Instance:**

- File Explorer:** Shows the project structure under "AKASHTech".
- Code Editor:** Displays the file `views.py` with the following content:

```
task 9 > mysite > myapp > views.py > ...
12     contact= request.POST.get("contact")
13     address= request.POST.get("address")
14     zipcode = request.POST.get("zipcode")
15     college = request.POST.get("college")
16     dept = request.POST.get("dept")
17
18     context={
19         'fn':fn,
20         'ln':ln,
21         'email':email,
22         'gender':gender,
23         'contact':contact,
24         'address' :address,
25         'zipcode':zipcode,
26         'college':college,
27         'dept':dept
28     }
29
30     return render(request,'data.html',context)
31
32     return render(request,"form.html")
```
- Status Bar:** Shows "Ln 1, Col 1" and "Python" along with other standard status indicators.

**Bottom Instance:**

- File Explorer:** Shows the project structure under "AKASHTech".
- Code Editor:** Displays the file `form.html` with the following content:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-gg0yRiXcLmPjP1iLBeBzq7a7iDqXrCv2QqCz8Xeev8ABUxTz5yXqo7aFZj/"/>
        <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@5.15.3/css/fontawesome.min.css" integrity="undefined" crossorigin="anonymous">
        <title>Document</title>
    </head>
    <body>
        <div class="container mt-2" style="width: 60%;>
            <form class="needs-validation p-3" action="" method="POST" style="border: 2px solid blue; border-radius: 10px;">
                {% csrf_token %}
                <div align="center">
                    <h3>Registration Form</h3>
                </div>
                <div class="form-row">
                    <div class="col-md-3 mb-3">
                        <label for="validationTooltip01">First name:</label>
                        <input type="text" class="form-control" id="validationTooltip01" name="fn" placeholder="First name" required>
                        <div class="valid-tooltip">
                            Looks good!
                        </div>
                    </div>
                    <div class="col-md-3 mb-3">
                        <label for="validationTooltip02">Last name:</label>
                        <input type="text" class="form-control" id="validationTooltip02" name="ln" placeholder="Last name" value="Otto" required>
                        <div class="valid-tooltip">
                            Looks good!
                        </div>
                    </div>
                </div>
            </form>
        </div>
```
- Status Bar:** Shows "Ln 1, Col 1" and "HTML" along with other standard status indicators.

File Edit Selection View Go Run Terminal Help form.html - AkashTech - Visual Studio Code

EXPLORER AKASHTECH

task 9 > mysite > templates > form.html > ...

```
46     </div>
47     <input type="email" class="form-control" id="validationTooltipUsername" placeholder="Enter Email" aria-describedby="validationTooltipUsernamePrepend" name="email" required>
48     | Please choose a unique and valid email
49   </div>
50   </div>
51   <div class="col-md-3 mb-3">
52     <label for="inputState">Gender</label>
53     <select id="inputState" class="form-control" name="gender">
54       <option selected="Choose..."></option>
55       <option value="male">Male</option>
56       <option value="female">Female</option>
57     </select>
58   </div>
59   </div>
60   <div class="form-row">
61     <div class="col-md-3 mb-3">
62       <label for="validationTooltip03">Contact</label>
63       <input type="text" class="form-control" id="validationTooltip03" placeholder="Phone Number" name="contact" required>
64       | Please provide a valid Contact Number
65     </div>
66     <div id="date-picker-example" class="col-md-6 mb-3 md-form md-outline input-with-post-icon datepicker">
67       <label for="inputAddress">Address</label>
68       <input type="text" class="form-control" id="inputAddress" placeholder="1234 Main St" name="address">
69     </div>
70     <div class="col-md-3 mb-3">
71       <label for="validationTooltip05">Zip</label>
72       <input type="text" class="form-control" id="validationTooltip05" placeholder="Zip" name="zipcode" required>
73       | Please provide a valid zip.
74     </div>
75   </div>
76   </div>
```

Python 3.8.8 64-bit ('webdev': conda) 0 0 0 Type here to search 11-Jun-21 801 PM 35°C ENG

File Edit Selection View Go Run Terminal Help form.html - AkashTech - Visual Studio Code

EXPLORER AKASHTECH

task 9 > mysite > templates > form.html > ...

```
73     <div id="date-picker-example" class="col-md-6 mb-3 md-form md-outline input-with-post-icon datepicker">
74       <label for="inputAddress">Address</label>
75       <input type="text" class="form-control" id="inputAddress" placeholder="1234 Main St" name="address">
76     </div>
77     <div class="col-md-3 mb-3">
78       <label for="validationTooltip05">Zip</label>
79       <input type="text" class="form-control" id="validationTooltip05" placeholder="Zip" name="zipcode" required>
80       | Please provide a valid zip.
81     </div>
82   </div>
83   </div>
84   <div class="form-row">
85     <div class="col-md-6 mb-3">
86       <label for="validationTooltip03">College Name</label>
87       <input type="text" class="form-control" id="validationTooltip03" placeholder="College Name" name="college" required>
88       | Please provide a valid Contact Number
89     </div>
90     <div class="col-md-6 mb-3">
91       <label for="inputState">Department</label>
92       <select id="inputState" class="form-control" name="dept">
93         <option selected="Choose..."></option>
94         <option value="Information Technology">IT</option>
95         <option value="Computer">Computer</option>
96         <option value="Mechanical">Mechanical</option>
97         <option value="Electrical">Electrical</option>
98         <option value="Environmental">Environmental</option>
99         <option value="Civil">Civil</option>
100      </select>
101    </div>
102    <div class="col-12">
103      <button class="btn btn-primary" type="submit">Submit form</button>
104    </div>
105  </div>
106  </div>
107  </div>
108  </div>
109  </div>
110  </div>
```

Python 3.8.8 64-bit ('webdev': conda) 0 0 0 Type here to search 11-Jun-21 804 PM 35°C ENG

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure under "AKASHT...". The "mysite/templates" folder contains "data.html", "form.html", "db.sqlite3", "manage.py", and "Day9.pdf".
- Code Editor:** The "data.html" file is open, displaying the following content:

```
1 Name : {{fn}} {{ln}} <br>
2 Email : {{email}}<br>
3 Gender : {{gender}}<br>
4 Contact : {{contact}}<br>
5 Address : {{address}},zip={{zipcode}}<br>
6 College : {{dept}},{{college}} <br>
```

- Bottom Status Bar:** Shows "Python 3.8.8 64-bit (webdev:conda)" and other system information like "35°C" and "8:02 PM".

## Day 10:-

- This was our last days of this amazing internship. In this session we learned about databases and its applications.
- Django has a inbuilt support of mainly five databases.
  1. PostgreSQL
  2. MariaDB
  3. MySQL
  4. SQLite
  5. Oracle
- Then we learned how to connect MySQL and SQLite databases to our Django project and perform basic operations on databases with use of our Django application.
- Then we learned how make models and how to migrate into our connected databases.
- Then we learned how to perform CRUD operation separately. And saw how those work.
- Then we discussed some problems and errors which were coming to some student and solve them.
- Then we also discussed which are various platforms where we can deploy our Django project very easily.
- After that we jumped into Latest Technologies and discussed about them that which technologies are ruling market at present and also will rule in coming years.

The screenshot shows a Windows desktop environment. At the top is a taskbar with various icons. Below it is a command prompt window titled 'C:\Windows\System32\cmd.exe - python manage.py runserver'. The window displays Python code for creating a superuser and logs of a Django development server running on port 8000. Below the command prompt is a Visual Studio Code editor window. The title bar says 'urls.py - Django\_DB\_CRUD - Visual Studio Code'. The editor shows the 'urls.py' file for a Django application named 'Django\_DB\_CRUD'. The file contains URL patterns for the admin site and other views. The bottom of the screen shows another taskbar with the same icons as the top one.

```
(webdev) C:\Users\patel\Pictures\Internship-Akash-Technolabs-main\Day-10_Django_DB_CRUD\ Django_DB_CRUD>python manage.py createsuperuser
Username (leave blank to use 'patel'): Prakhar
Email address: patel.prakhar09@gmail.com
Password:
Password (again):
The password is too similar to the username.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.

(webdev) C:\Users\patel\Pictures\Internship-Akash-Technolabs-main\Day-10_Django_DB_CRUD\ Django_DB_CRUD>python manage.py runserver
watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 11, 2021 - 20:16:32
Django version 3.2.3, using settings 'Django_DB_CRUD.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[11/Jun/2021 20:16:38] "POST /admin/login/?next=/admin/ HTTP/1.1" 302 0
[11/Jun/2021 20:16:38] "GET /admin/ HTTP/1.1" 200 3981
[11/Jun/2021 20:16:38] "GET /static/admin/css/dashboard.css HTTP/1.1" 200 380
[11/Jun/2021 20:16:38] "GET /static/admin/img/icon-addlink.svg HTTP/1.1" 200 331
[11/Jun/2021 20:16:38] "GET /static/admin/img/icon-changelink.svg HTTP/1.1" 200 380
[11/Jun/2021 20:16:43] "GET /admin/DB_app/student/add/ HTTP/1.1" 200 7152
[11/Jun/2021 20:16:43] "GET /admin/jsi18n/ HTTP/1.1" 200 3195
[11/Jun/2021 20:16:43] "GET /static/admin/js/jquery.init.js HTTP/1.1" 200 347
[11/Jun/2021 20:16:43] "GET /static/admin/css/forms.css HTTP/1.1" 200 8804
[11/Jun/2021 20:16:43] "GET /static/admin/js/urify.js HTTP/1.1" 200 7902
[11/Jun/2021 20:16:43] "GET /static/admin/js/actions.js HTTP/1.1" 200 7664
[11/Jun/2021 20:16:43] "GET /static/admin/js/prepopulate.js HTTP/1.1" 200 1531
[11/Jun/2021 20:16:43] "GET /static/admin/js/core.js HTTP/1.1" 200 5698
[11/Jun/2021 20:16:43] "GET /static/admin/js/RelatedObjectLookups.js HTTP/1.1" 200 5984
[11/Jun/2021 20:16:43] "GET /static/admin/js/vendor/jquery/jquery.js HTTP/1.1" 200 287630
[11/Jun/2021 20:16:43] "GET /static/admin/js/prepopulate_init.js HTTP/1.1" 200 492
[11/Jun/2021 20:16:43] "GET /static/admin/css/widgets.css HTTP/1.1" 200 11097
[11/Jun/2021 20:16:43] "GET /static/admin/js/change_form.js HTTP/1.1" 200 606
[11/Jun/2021 20:16:43] "GET /static/admin/js/vendor/xregexp/xregexp.js HTTP/1.1" 200 232381
Starting development server at http://127.0.0.1:8000/
  File Edit Selection View Go Run Terminal Help urls.py - Django_DB_CRUD - Visual Studio Code
  Explorer ... settings.py urls.py
  DJANG... .vscode ...
  > DB.app
  > Django_DB_CRUD
  > _pycache_
  > __init__.py
  > asgi.py
  > settings.py
  > urls.py
  > wsgi.py
  db.sqlite3
  manage.py
  Python 3.8.8 64-bit ('webdev': conda)
  Type here to search
  34°C ⚡ ENG 8:20 PM 11-Jun-21
```



Django administration

WELCOME, ROMIL ZALAVADIYA

Home > Db\_App > Students

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

DB\_APP

Students + Add

Select student to change

Action: — Go 0 of 2 selected

STUDENT

Bono

Bibin

2 students

This screenshot shows the Django Admin interface for the 'Students' model. A success message at the top right indicates that 'The student "Bono" was added successfully.' The main area displays a list of two students: 'Bono' and 'Bibin'. A 'Select student to change' dropdown is open, showing the 'STUDENT' option and the names of the selected students.

Django administration

WELCOME, BIBIN VIEW SITE / CHANGE PWD

Home > Db\_App > Students

AUTHENTICATION AND AUTHORIZATION

Groups + Add

Users + Add

DB\_APP

Students + Add

Select student to change

Action: — Go 0 of 2 selected

STUDENT

Bono

Bibin

2 students

This screenshot shows the same Django Admin interface as the previous one, but with a different success message: 'The student "Bono" was changed successfully.' The list of students remains the same, and the 'Select student to change' dropdown shows that 'Bono' is now selected.

The screenshot shows the Django administration interface for the 'Students' model. The top navigation bar includes links for 'Home', 'Db\_App', and 'Students'. A green success message at the top right states: 'The student "Bono" was deleted successfully.' Below this, the main content area has a heading 'Select student to change'. A sidebar on the left lists 'AUTHENTICATION AND AUTHORIZATION' (Groups, Users) and 'DB\_APP' (Students). At the bottom of the sidebar, there is a link labeled 'ADD STUDENT +'. The main content area contains a table header row with columns for 'Action', 'STUDENT', and 'Bibin'. A note below the table says '1 student'.