**Python with Django Internship Report**

**Personal Details**

Name : Jhanvi Parmar

College Name : LJ Institute of Engineering and Technology

Degree : BE

Semester : 7

Github URL : https://github.com/Jhanviparmar

**Company Details**

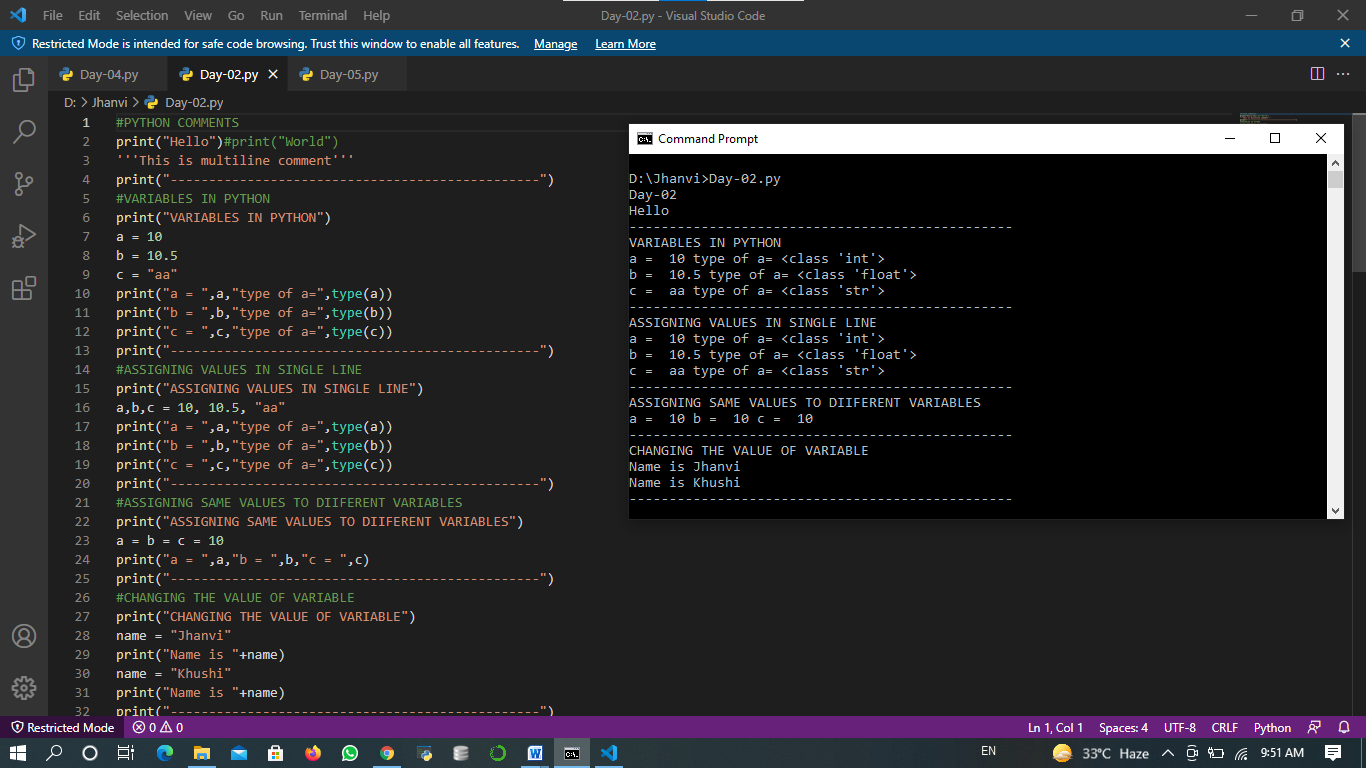
Company Name : Akash Technolabs

External Guide : Akash Padhiyar

Training Duration : 26-05-2021 to 21-06-2021

INDEX

|  |  |  |
| --- | --- | --- |
| **Index** | **Task Detail** | **Page Name** |
| 1. | Python Datatypes | 3 |
| 2. | Conditional Statements and loops | 5 |
| 3. | Python Functions and Operators | 7 |
| 4. | Python oop’sconcepts  (self, class,\_\_init()\_\_,contructors) | 11 |
| 5. | Django framework | 16 |
| 6. | Static website using django framework | 17 |
| 7. | My project(Registation page using django framework) | 26 |

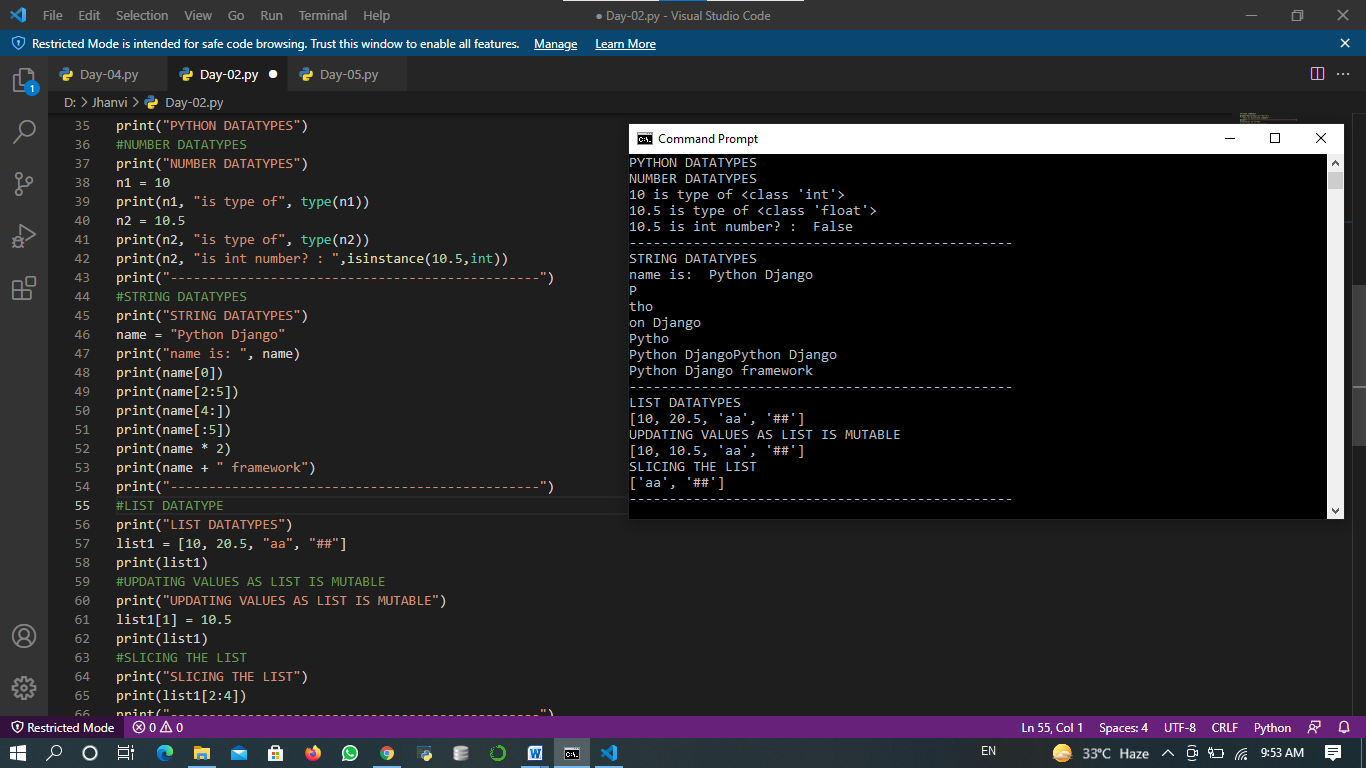
* **PYTHON BASICS:**
* ****

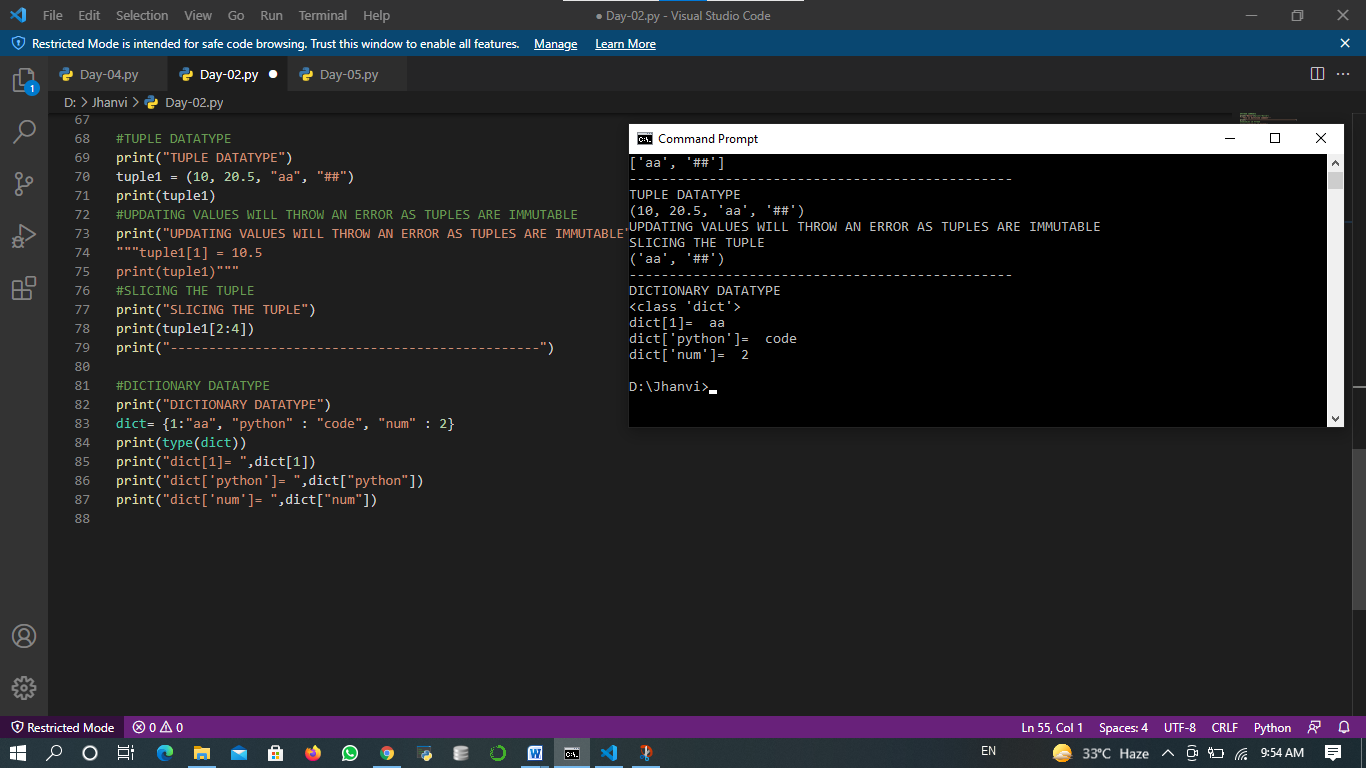
Here, The working on print() function has been shown. How to assign values to variables and also the use of Type() function , by which we can get the types of our variable , are shown.

* PYTHON DATATYPES:

There are five different data types in Python:

1. Number Data type:
2. String Data type:
3. List Data type:
4. Tuple Data type:
5. Dictionary Data type:





All kind of Data types are displayed in above Images.

1. In number Data type we can assign different values to variable and with help of type() and isinstance() function , we can find the type of variable.
2. In string Data type we can concat two strings and also can splitting of string with split function.
3. List is mutable and very much similar to array object. We can access list objects with the help of index numbers. Also we can drop and add objects in list.
4. Tuple is immutable Data type and can accessed by index numbers. If we try to update any existing value then it throws error.
5. Dictionary use keys and values to store objects.

* INPUT FUNCTION:

If we want to take input from user, then we use input() function.

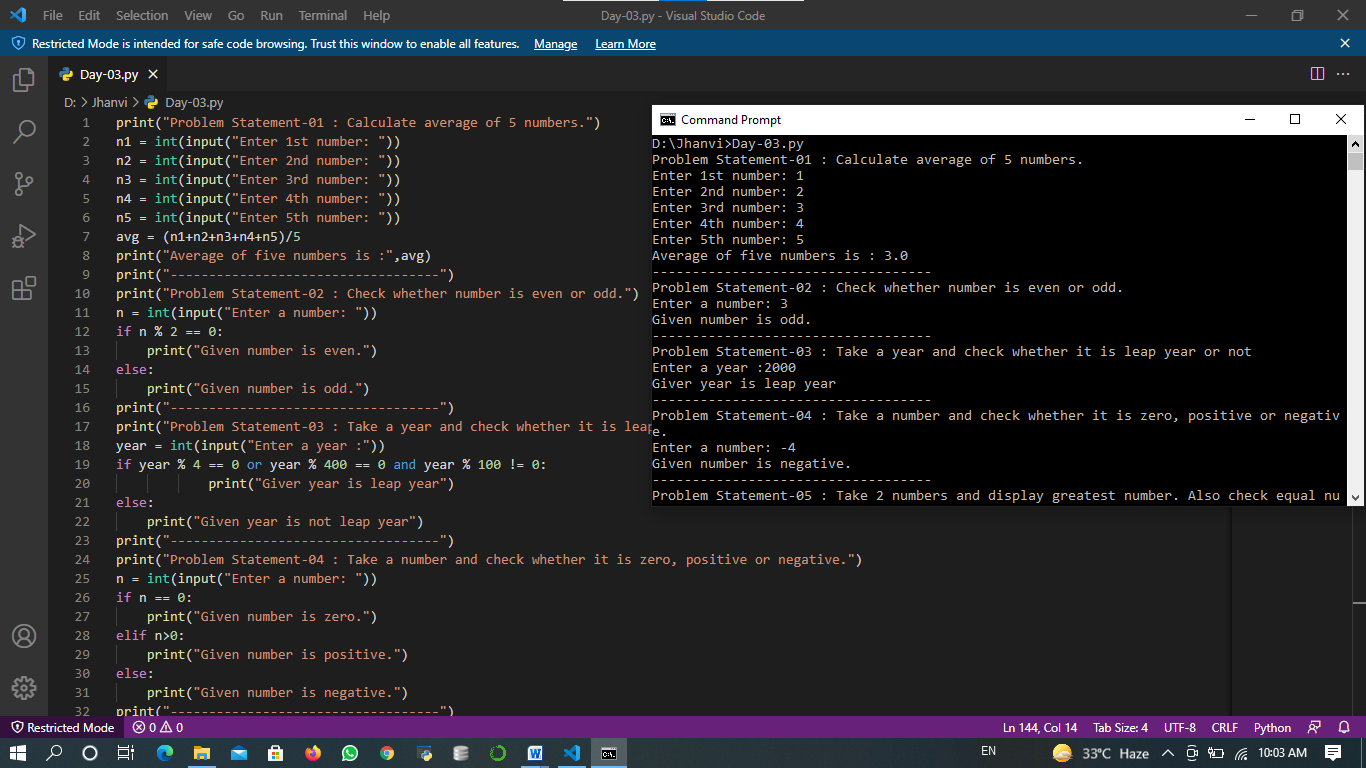
* CONDITIONAL STATEMENTS:

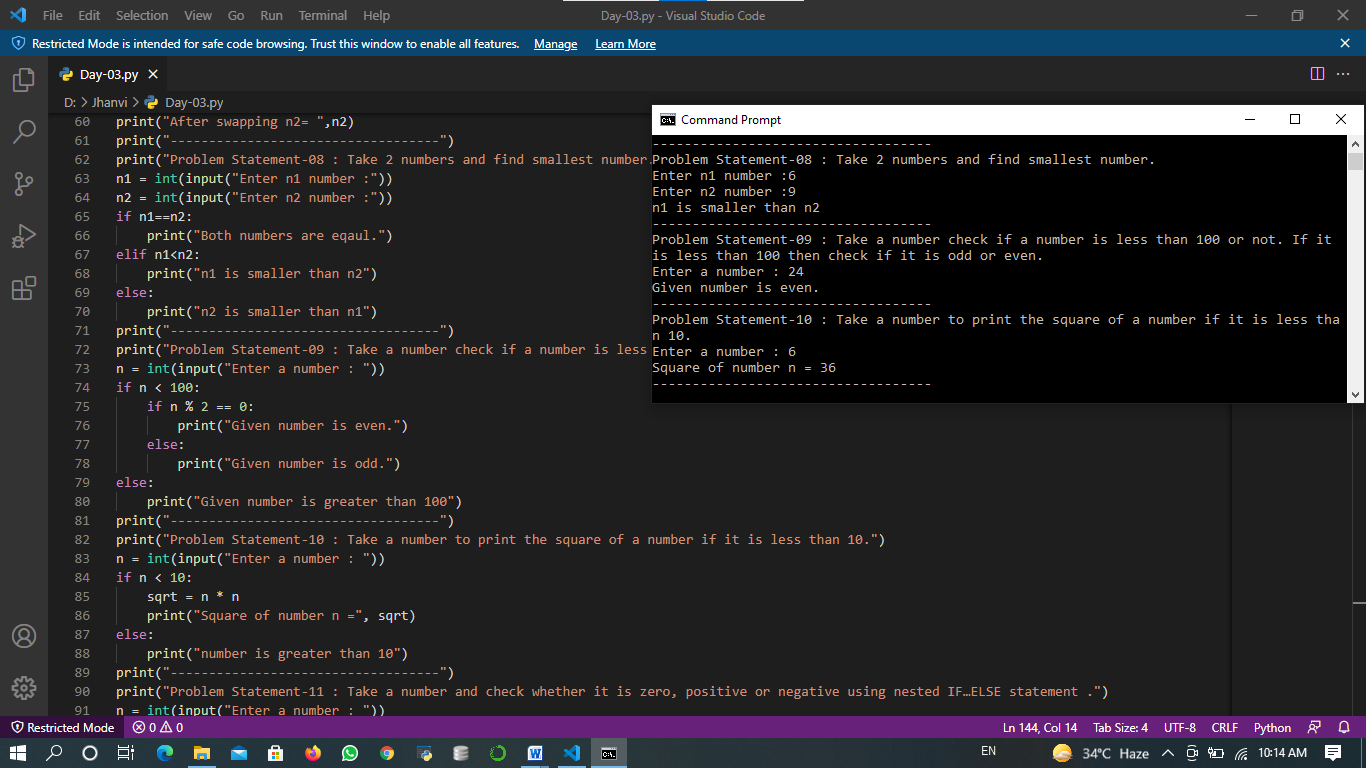
There are three types of conditional statements in python listed below:

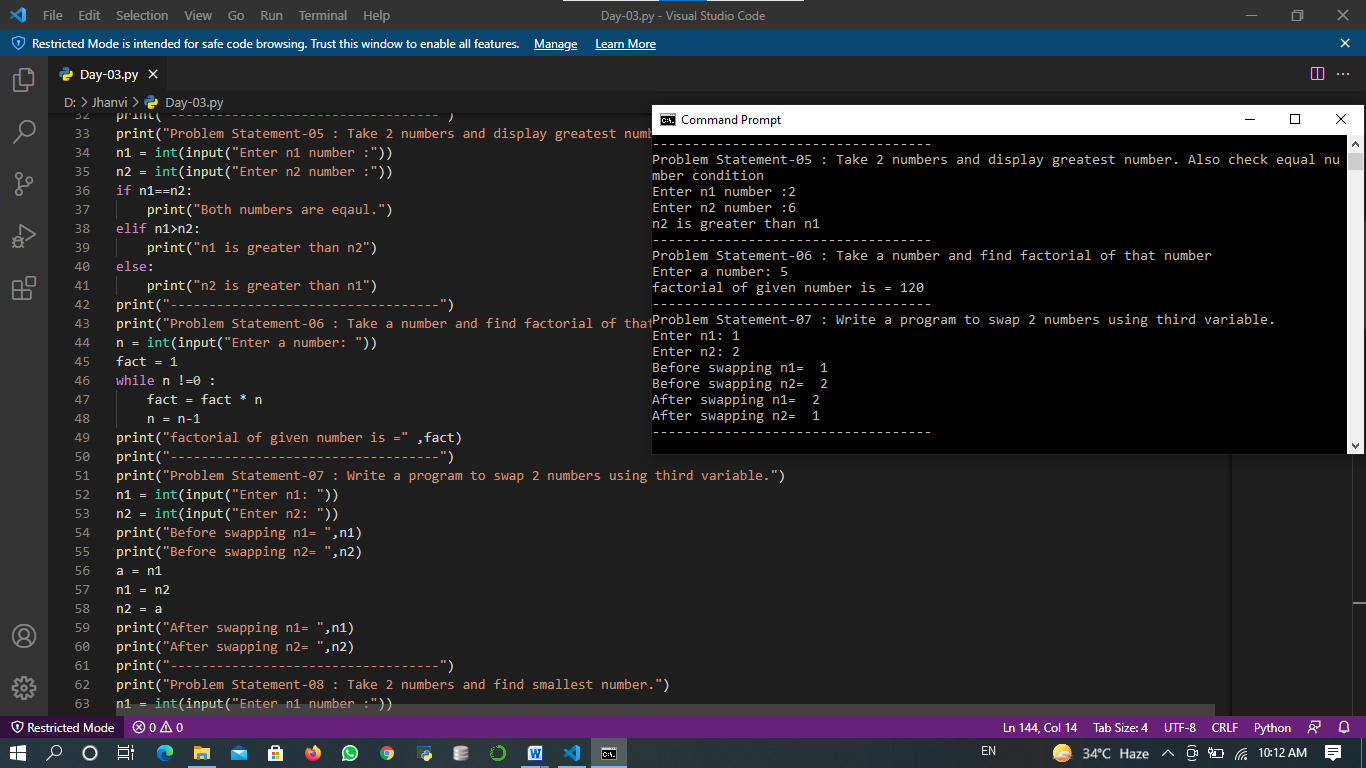
1. If
2. Else
3. Nested if…else

* LOOPS :

1. For loop
2. While loop







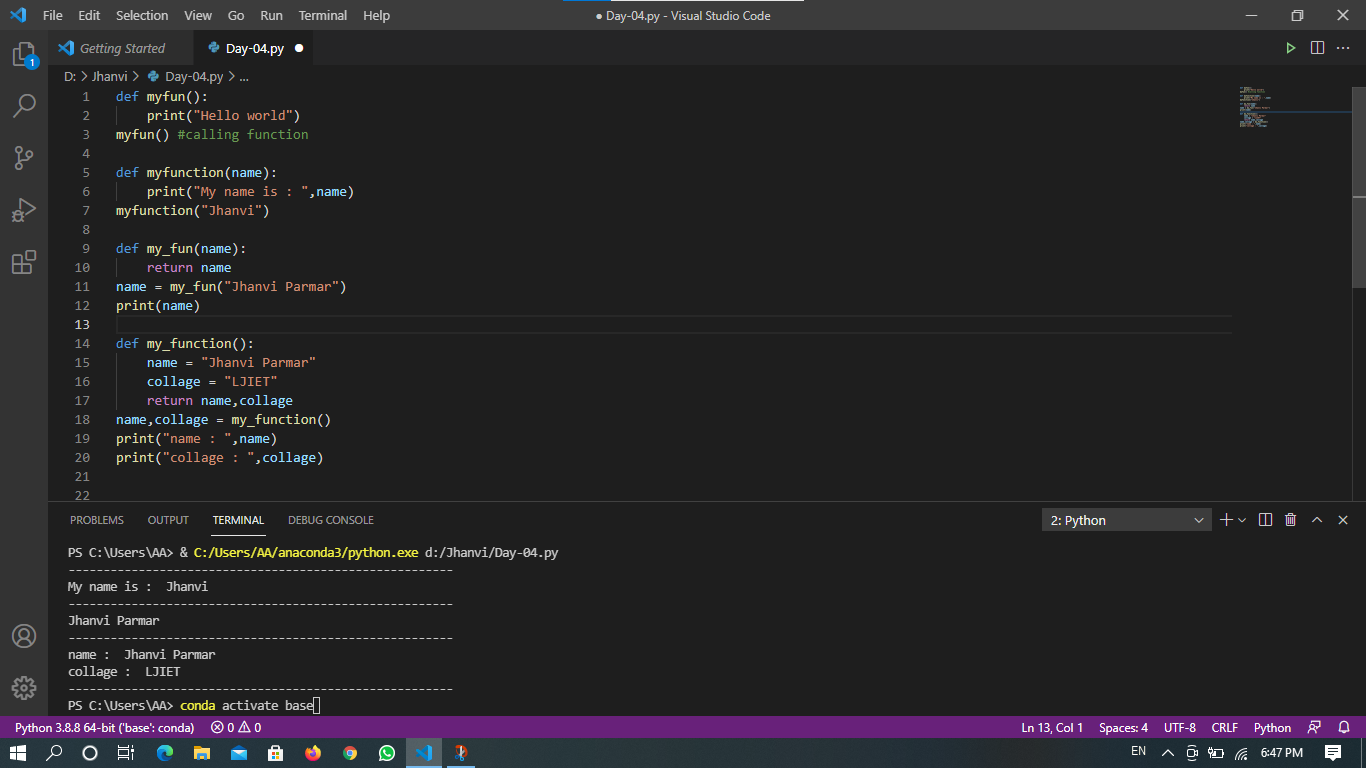
Here in above all examples, inputs are taken from user and many problems are solved like finding maximum – minimum numbers, average of numbers, swapping the numbers and more.

Here, input() function is used, loops and conditional statements and also logical operators like “and”, “or” and “not” are used.

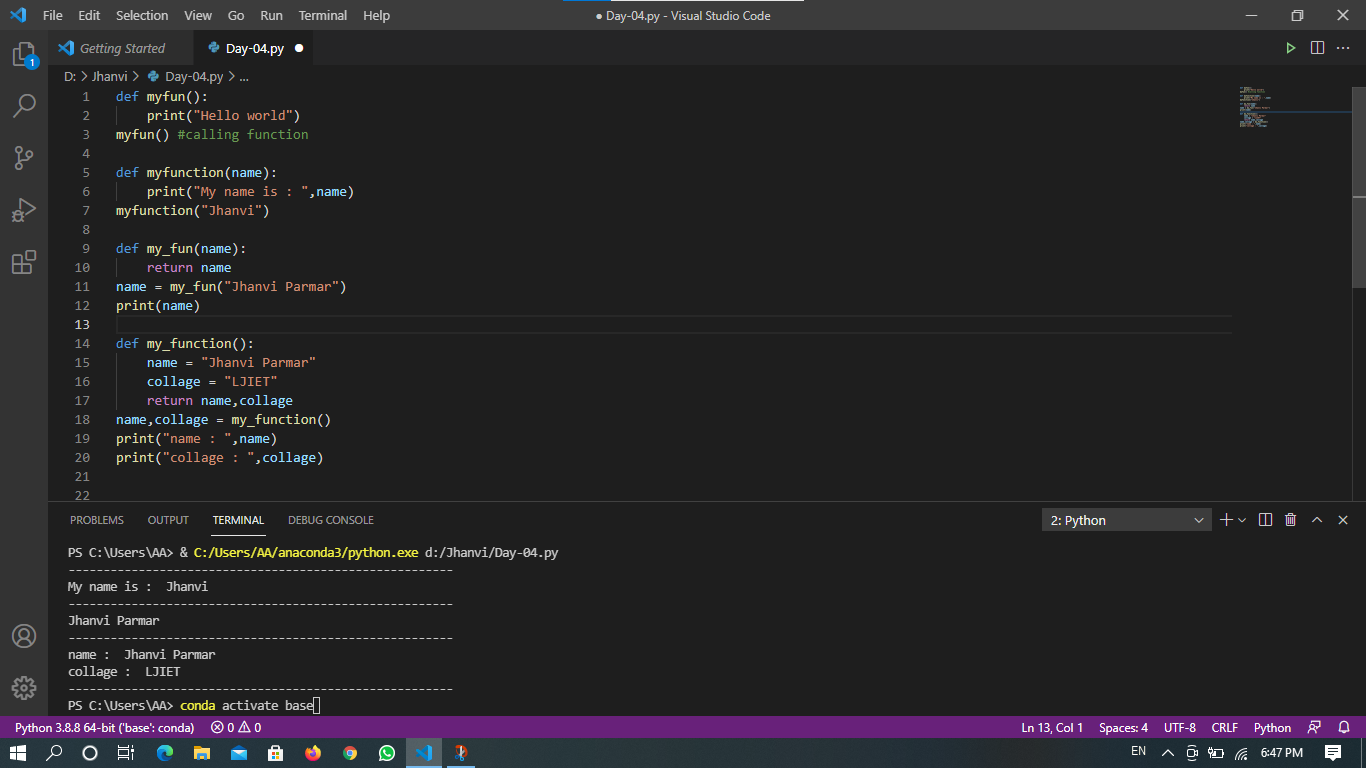
* FUNCTIONS IN PYTHON:

Function is a group of related statements that perfume a specific task.

* Def keyword marks the start of header.
* A colon(:) to mark the end of functions header.
* return statement to return value from function.



In above image, function with and without arguments and function with return statement is displayed.



* PYTHON FUNCTION ARGUMENTS:

There are three types of python function arguments using which we can call a function.

1. Default Arguments:
2. Keyword Arguments:
3. Variable-length Arguments:

* OPERATORS IN PYTHON:

1. Arithmetic Operator :

(+, -, \*, /, %, //, \*\*)

1. Comparison Operator:

(>, <, ==, !=, >=, <=)

1. Logical Operator:

(and, or, not)

1. Assignment Operator:

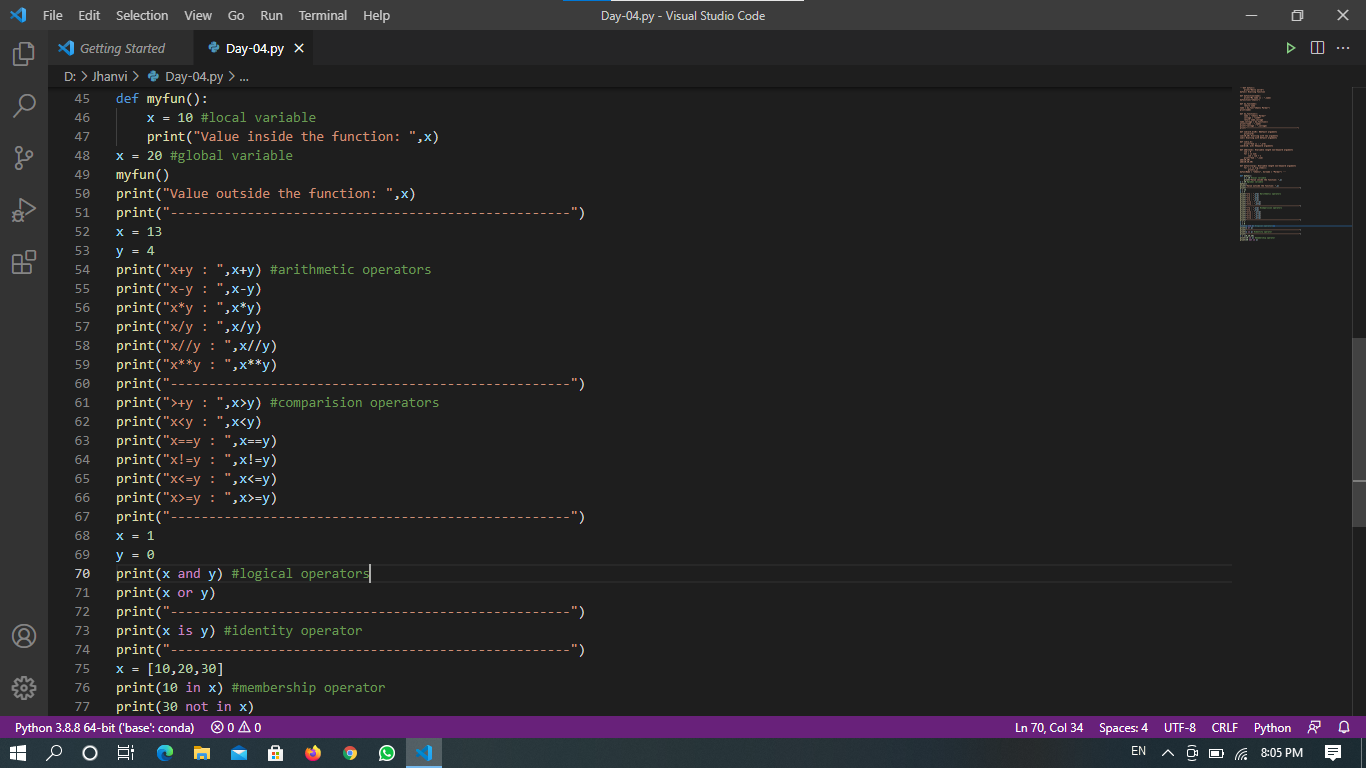
(=, +=, -=, \*=, /=, %=, //=, \*\*=)

1. Membership Operator:

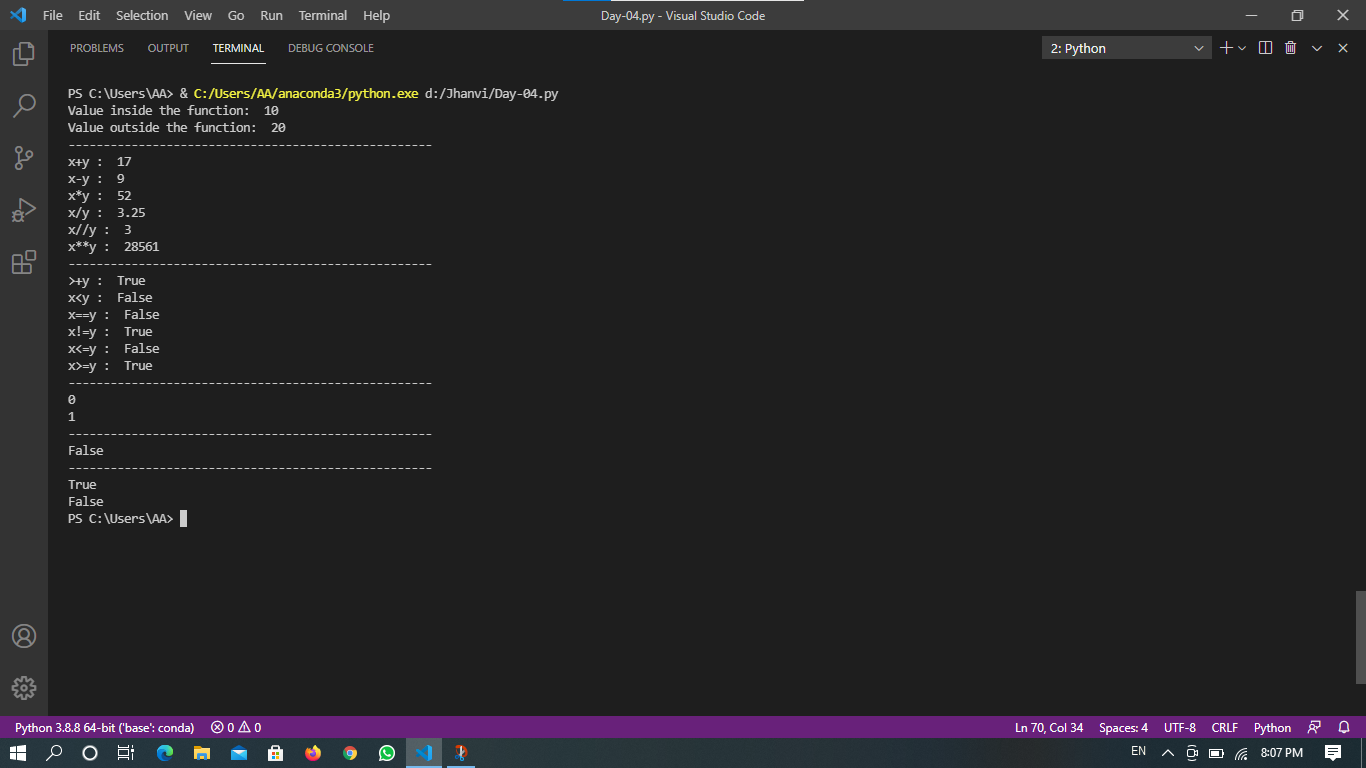
(in, not in)

1. Identity Operator:

(is, is not)



* Output:



* PYTHON oop’s Concepts:
* CLASS:

A class is blue print for the object. Classes are defined by the “Class” keyword.

* SELF:

This parameter refers to the object which invokes the method.

When you create new object the self parameter in the \_\_init\_\_ method is automatically set to reference the object you have just created.

* Python Constructors:

A constructor is a class function that instantiates an object to predefined values.

--There are two types of Constuctor:

1. Default Constructor

2. Parameterized Constructor

* INHERITENCE:

Inheritance allows programmer to create a general class first then later extend it to more specialized class.

--There are five types of inheritance:

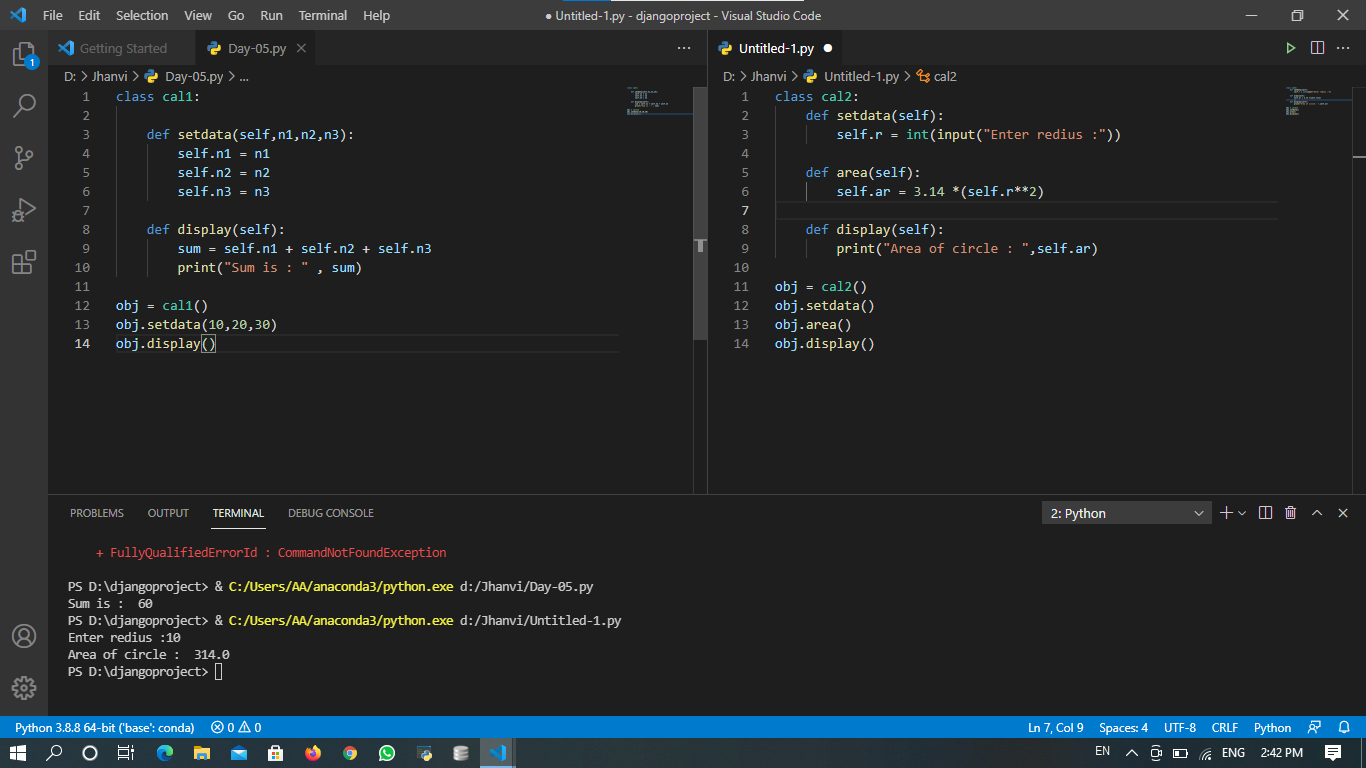
1. Single-level Inheritance:

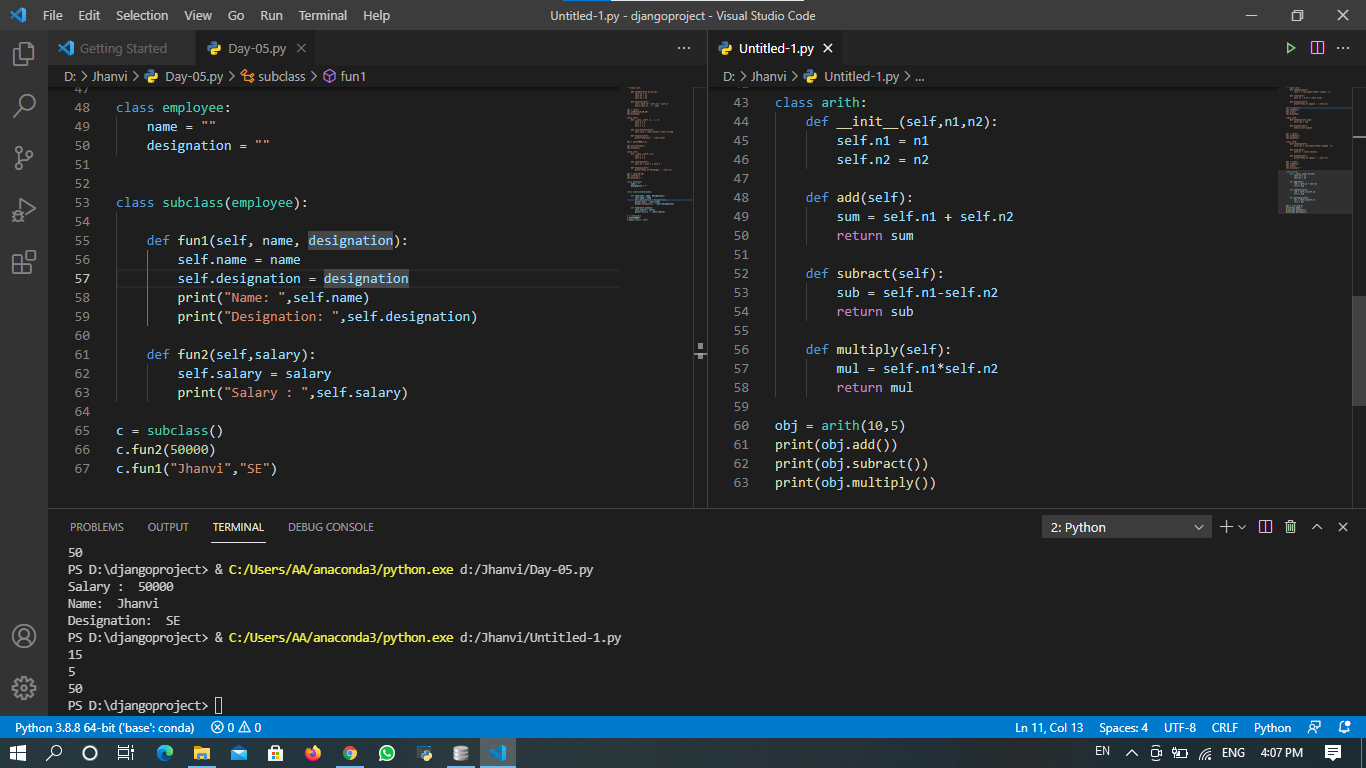
2. Multi-level Inheritance:

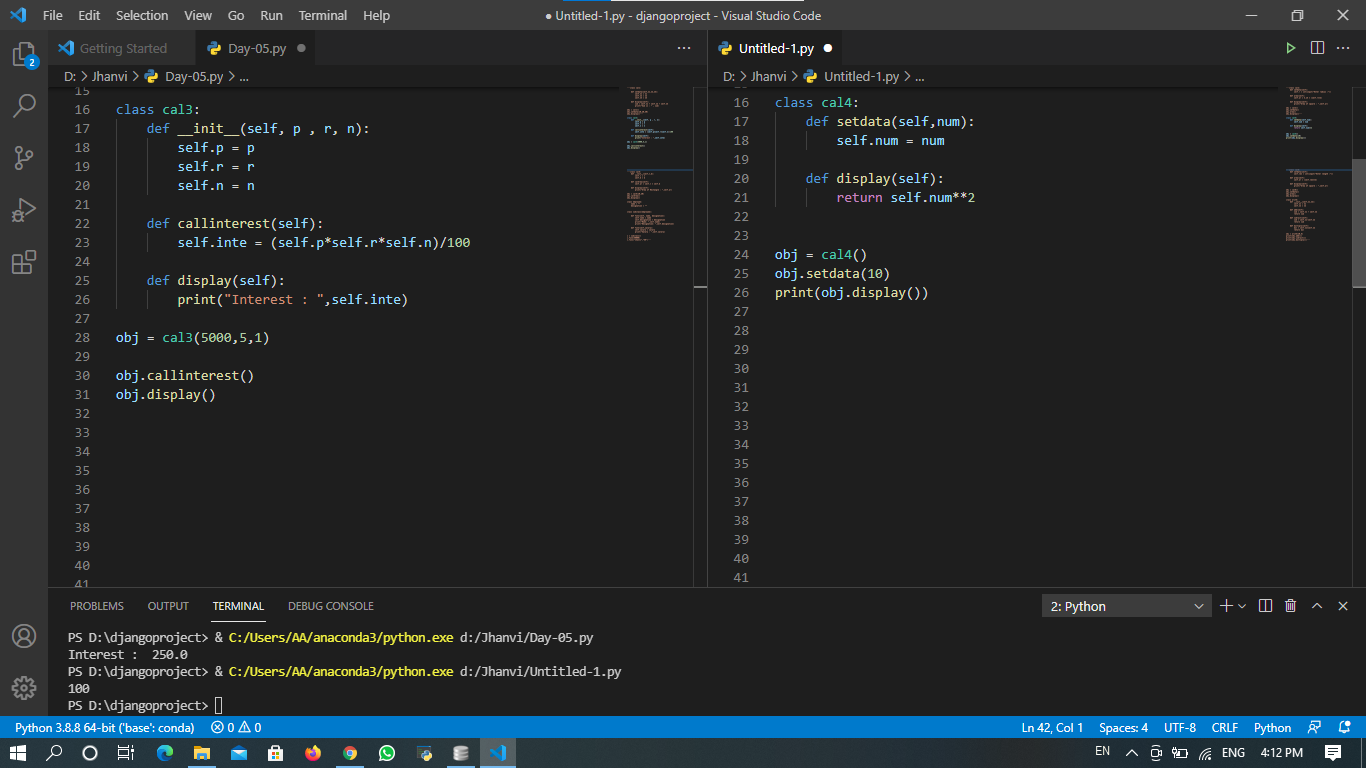
3. Multiple Inheritance:

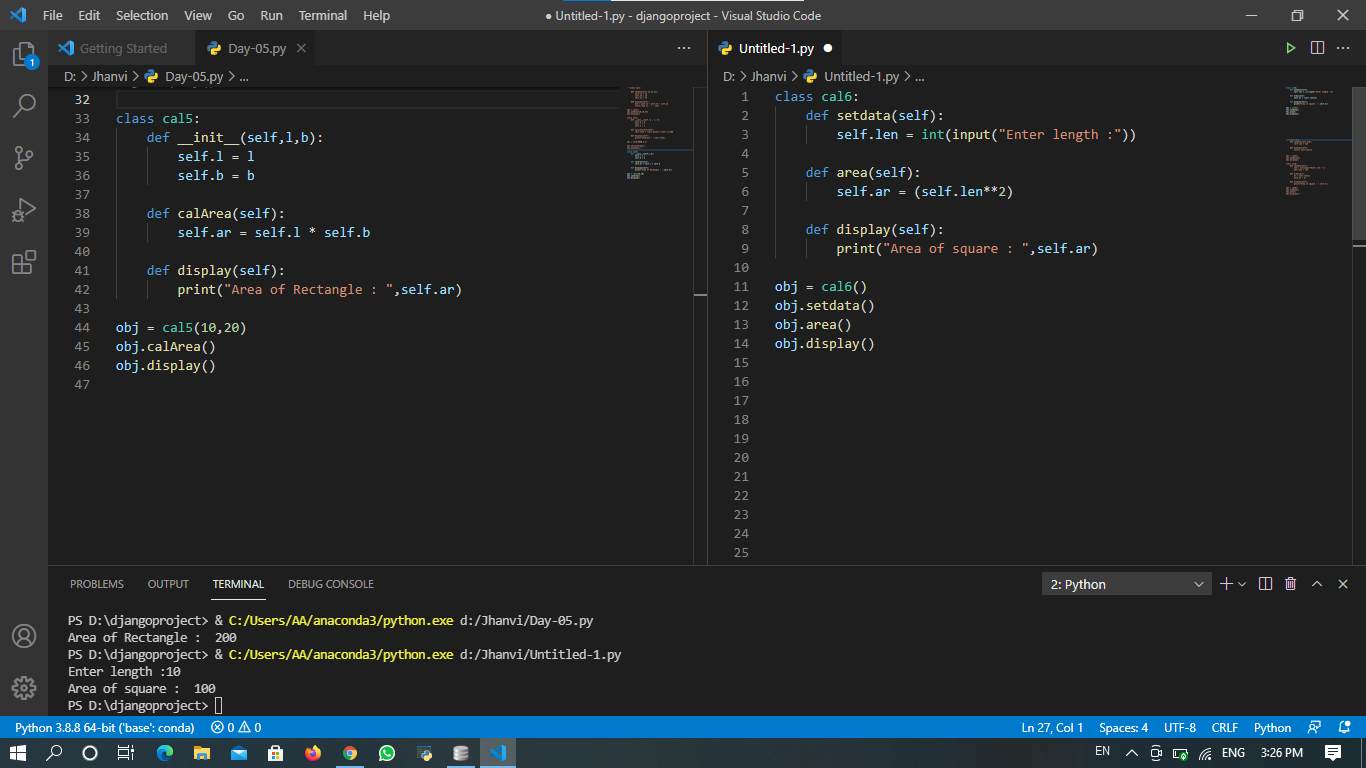
4. Hierarchical Inheritance:

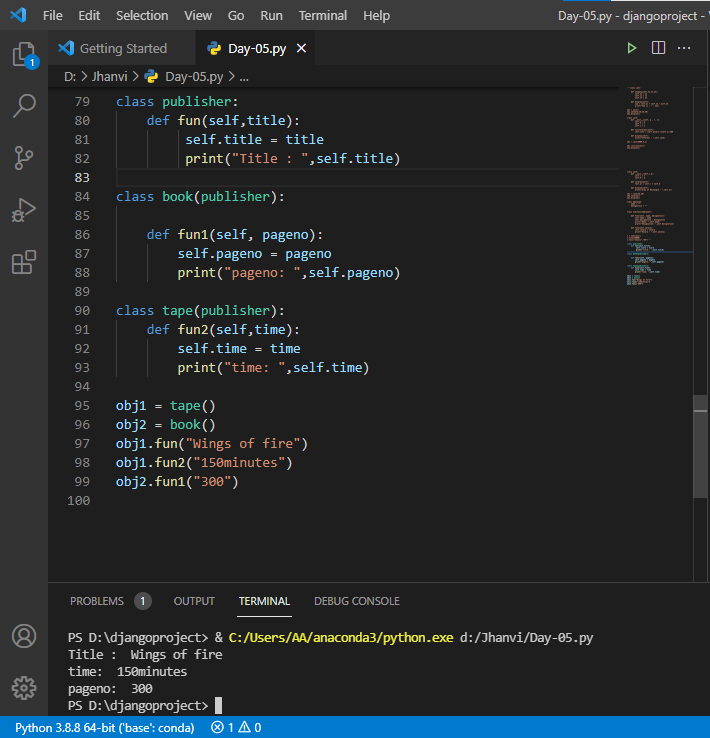
5. Hybrid Inheritance:

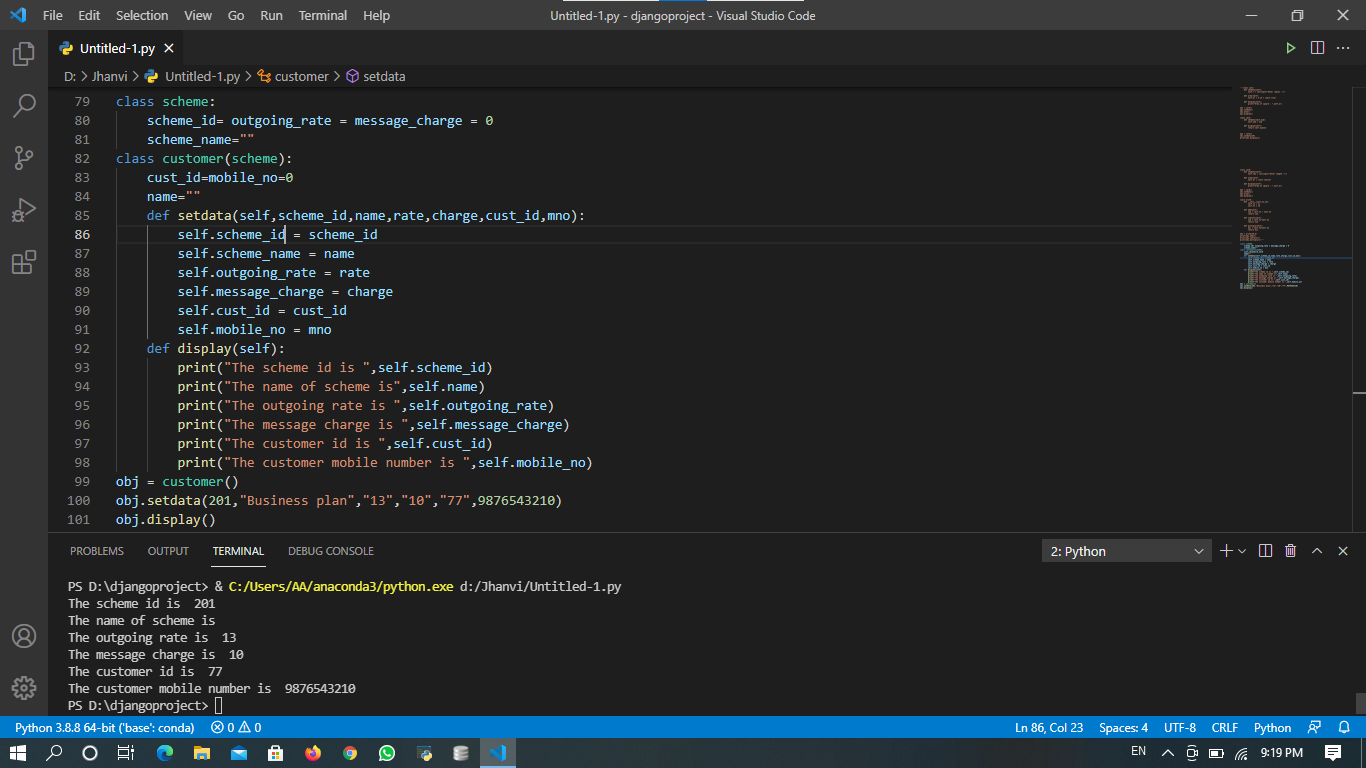












All these examples displayed above , has used oop’s concepts.

* Django Framework:

-Django is a free and open source web application framework, written in Python.

-A web framework is a set of components that helps you to develop websites faster and easier.

-It provides support for multiple databases such as MySQL, SQLite, and Oracle.

-Django was developed by Adrian Holovaty and Jacob Kalpan-moss at World Online News for efficient development.

* Django follows MVT:
* 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..

* PROJECT USING DJANGO:

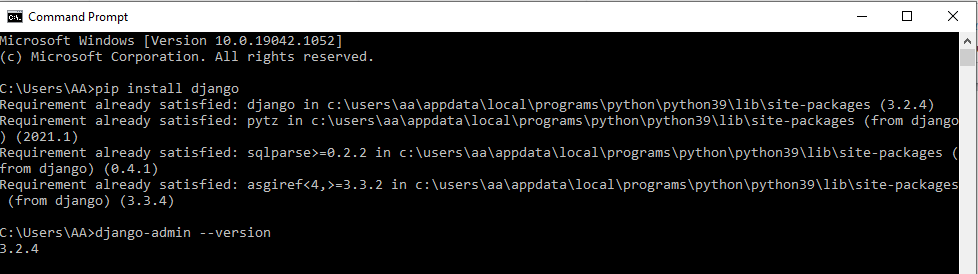
In this project, I have designed one small static website with navigation like homepage, about page, contact page.

Also I have designed a form and have returned all the data which is given as input in form.

I have also created admin panel and database.

Below shown all the screenshots display different operations in different files of djangoproject.

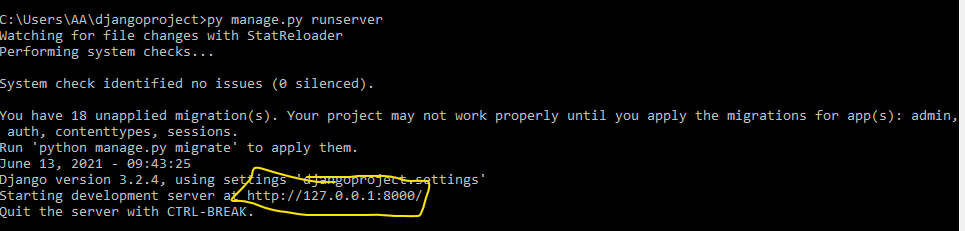
>Django installation and version check:

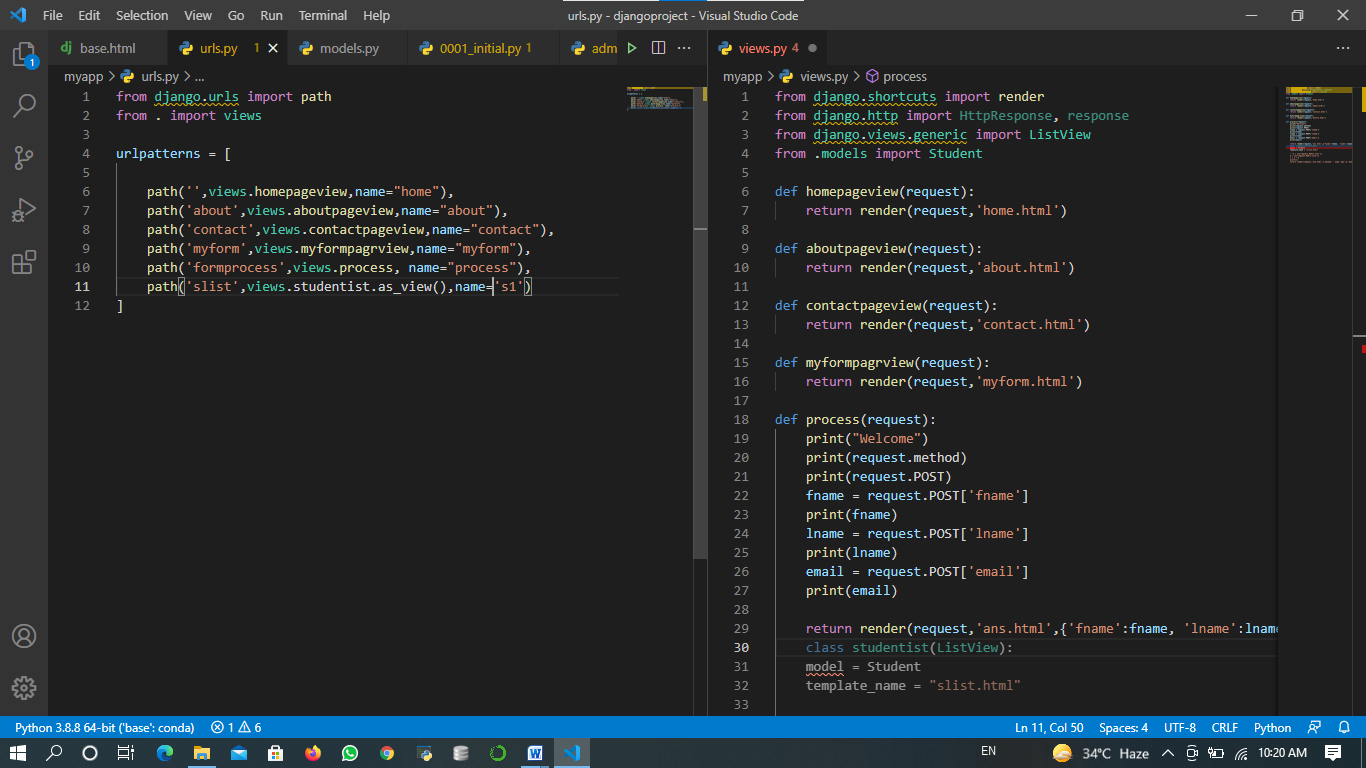


>Starting project and creating app:

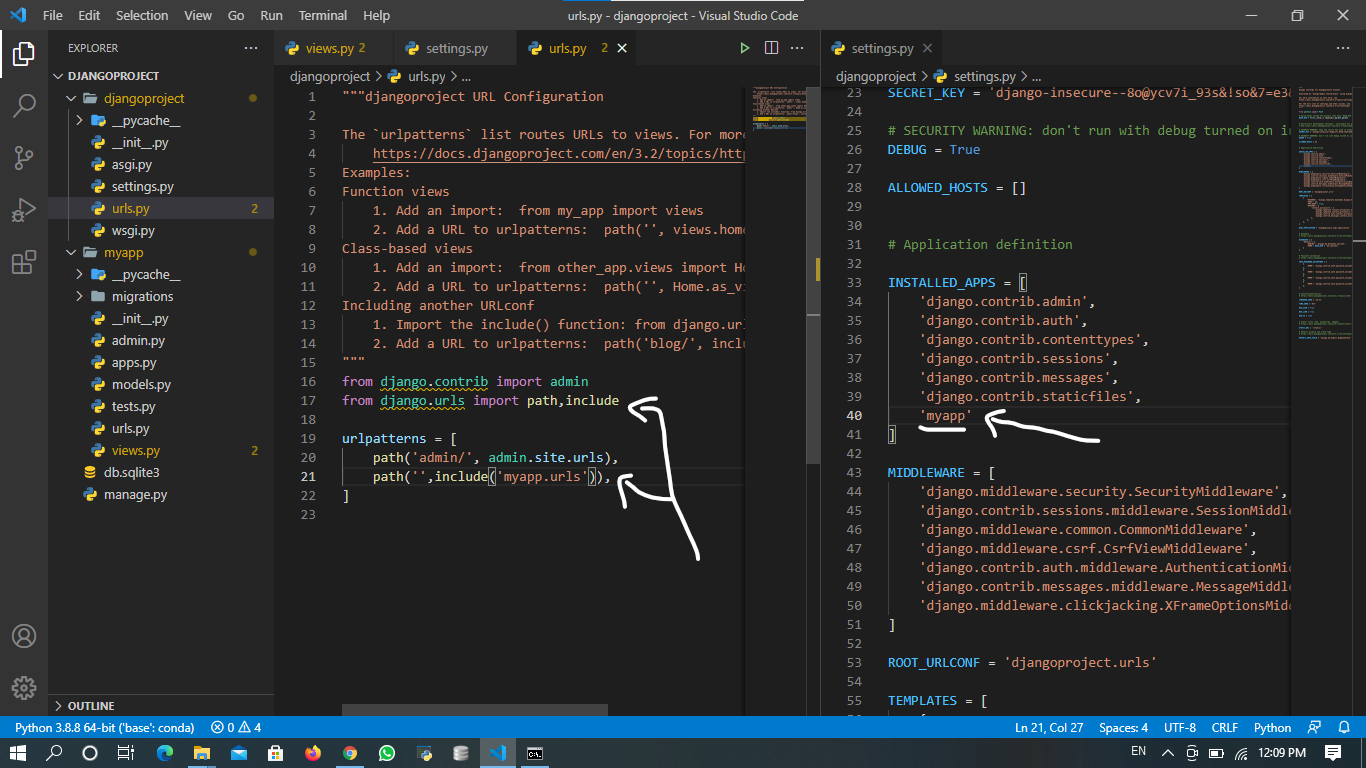


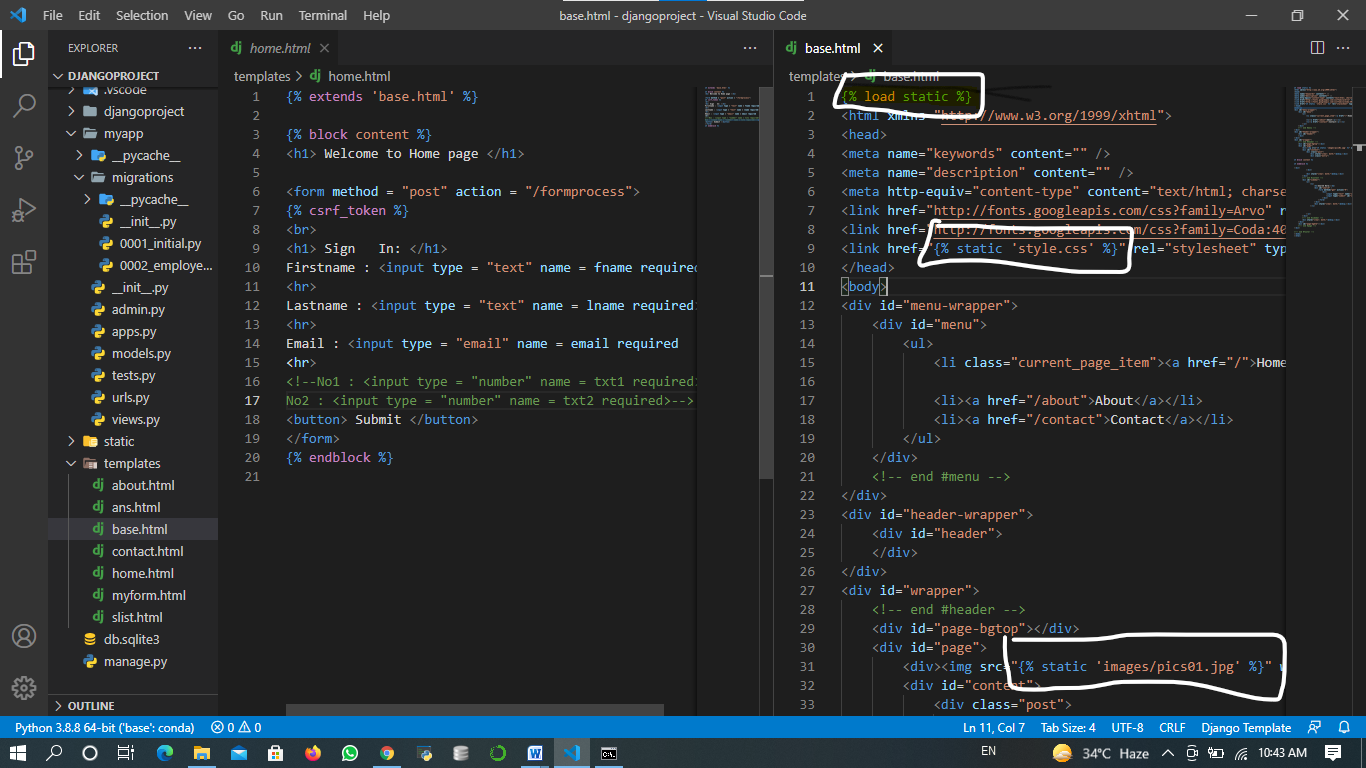
>Runserver:

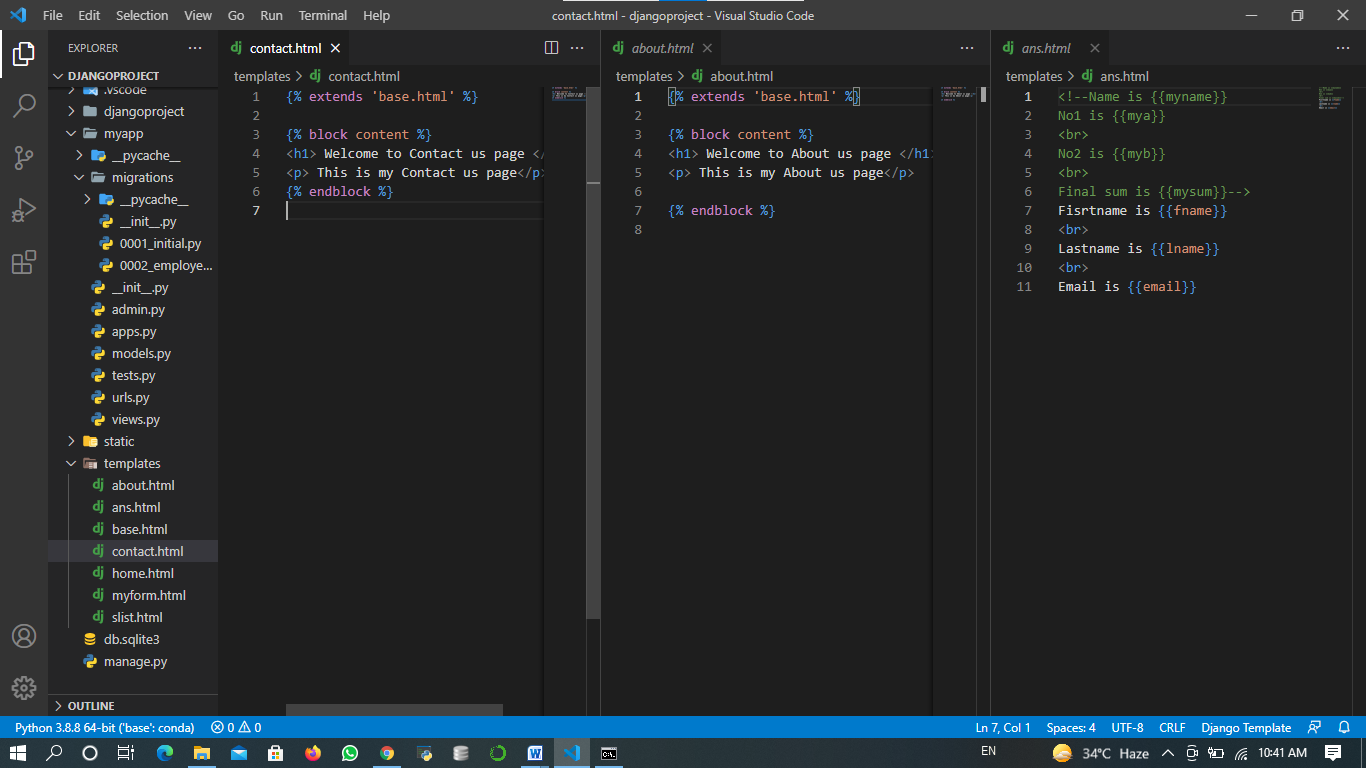


> views.py and urls.py:

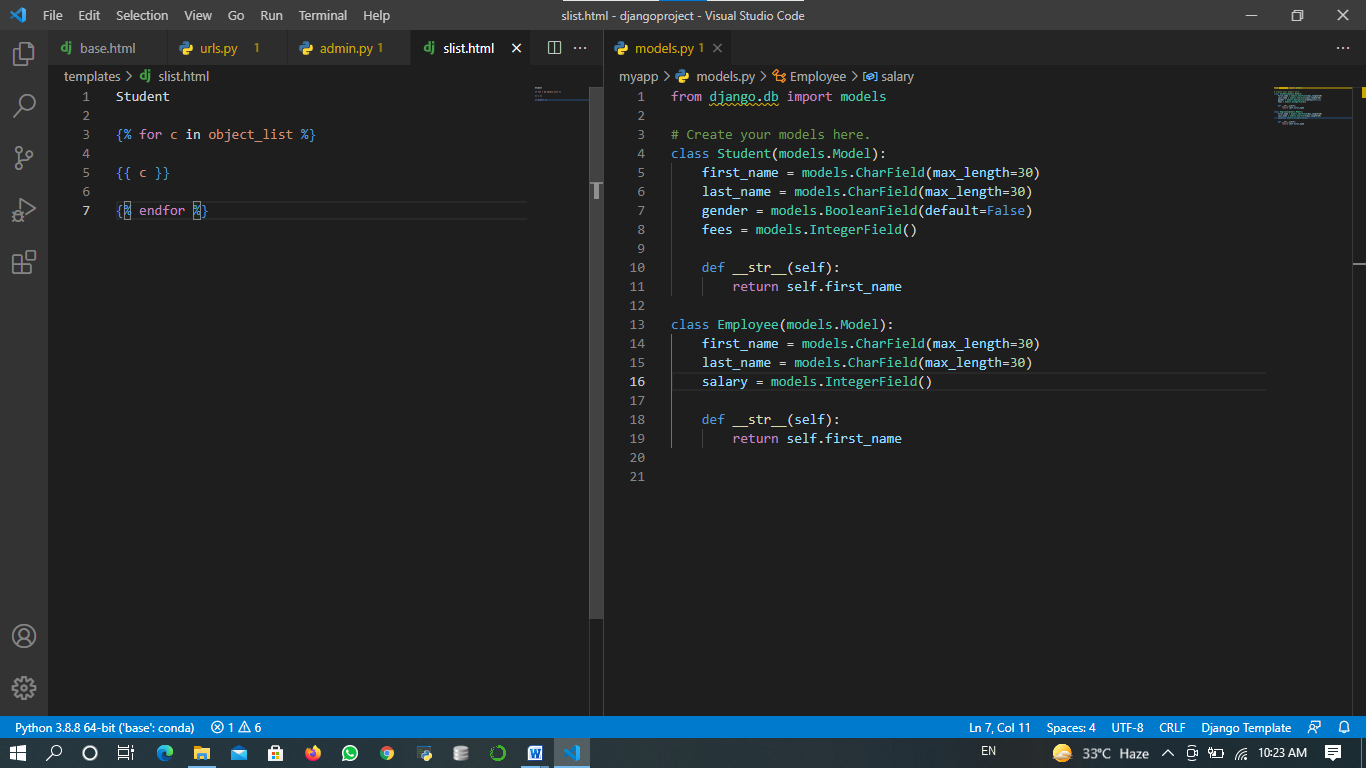
* settings.py and urls.py:



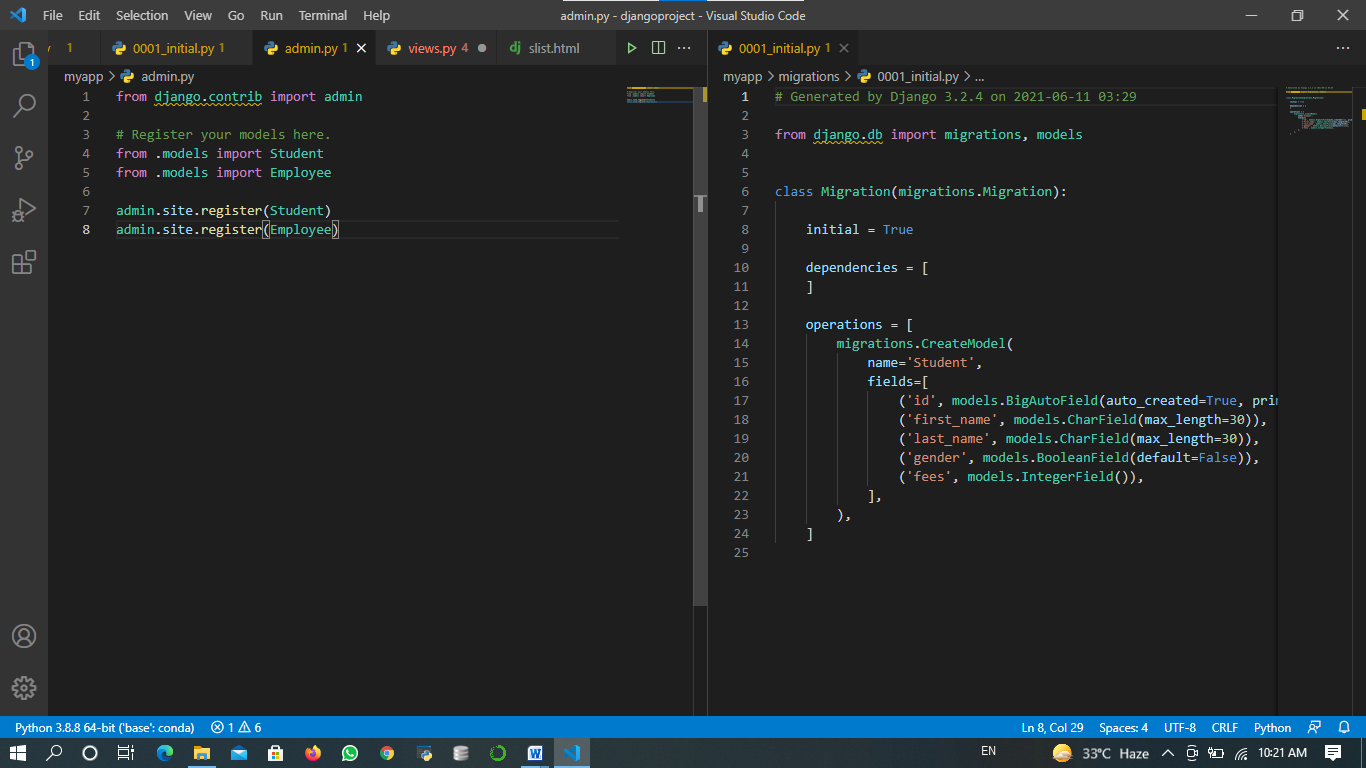
* home.html and base.html:
* about.html & contact.html & ans.html:



* models.py and slist.html:



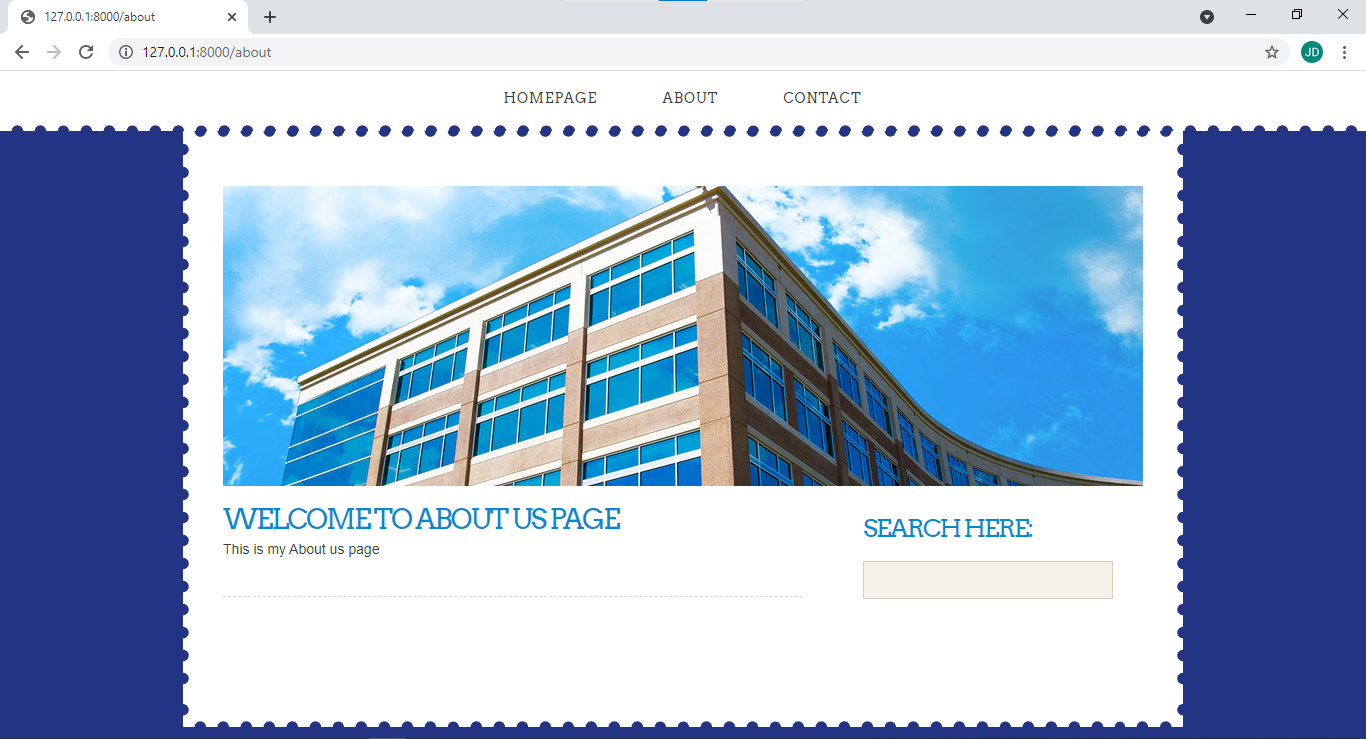
* admin.py & initial.py:



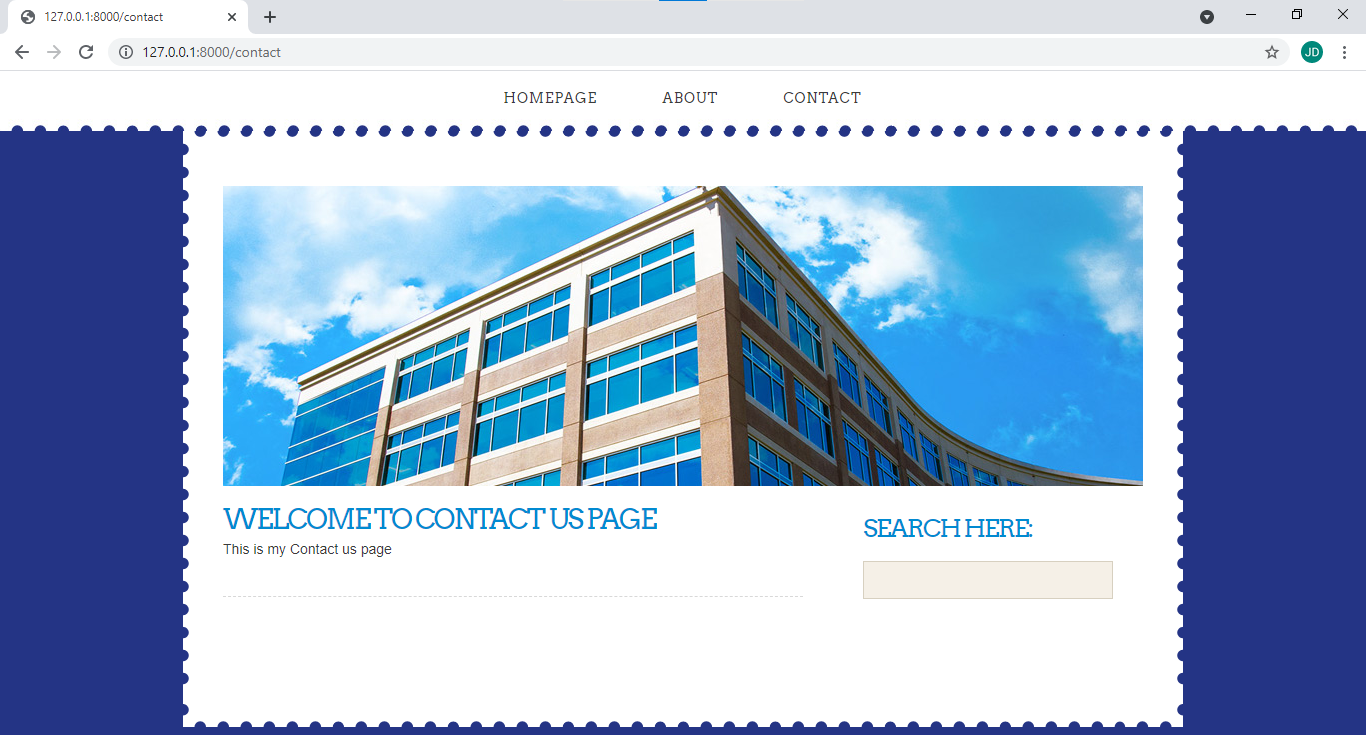
* OUTPUT:
* Homepage



* Aboutpage :



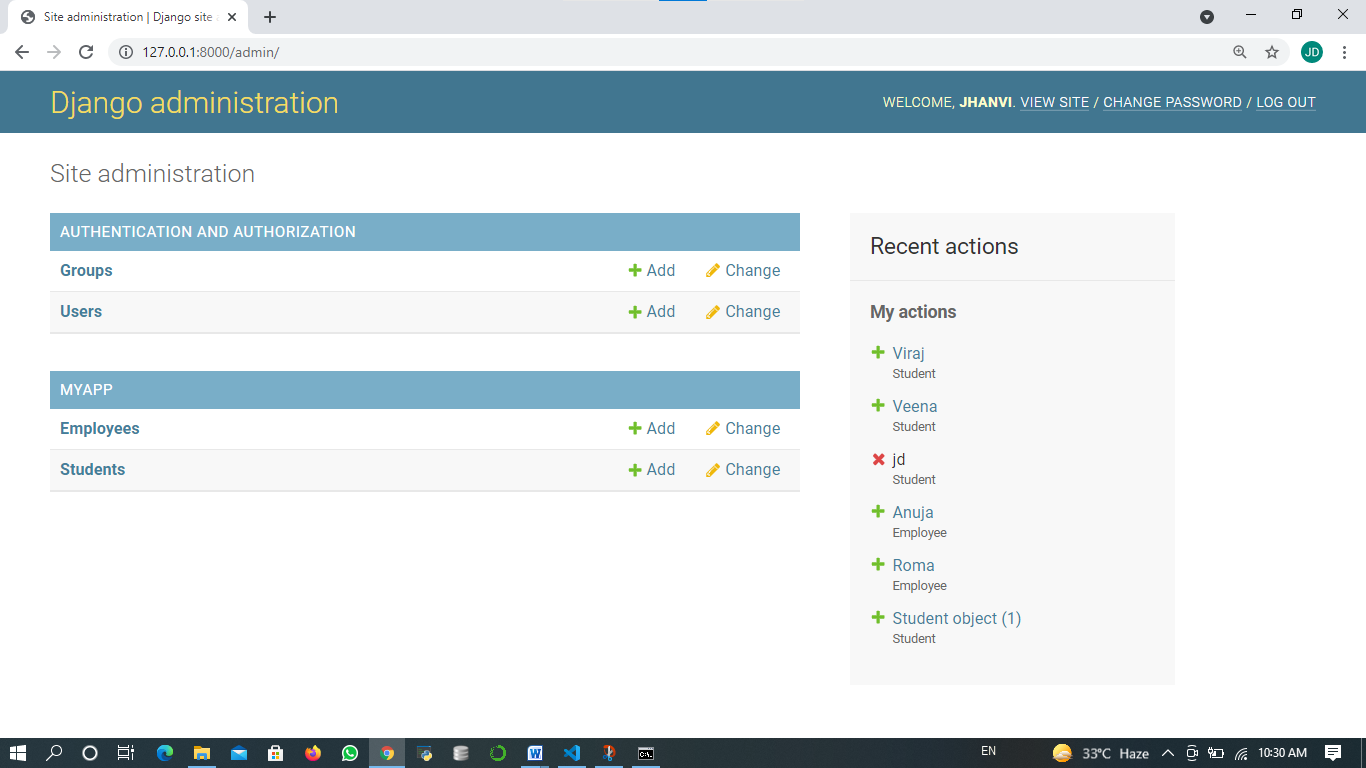
* Contactpage:



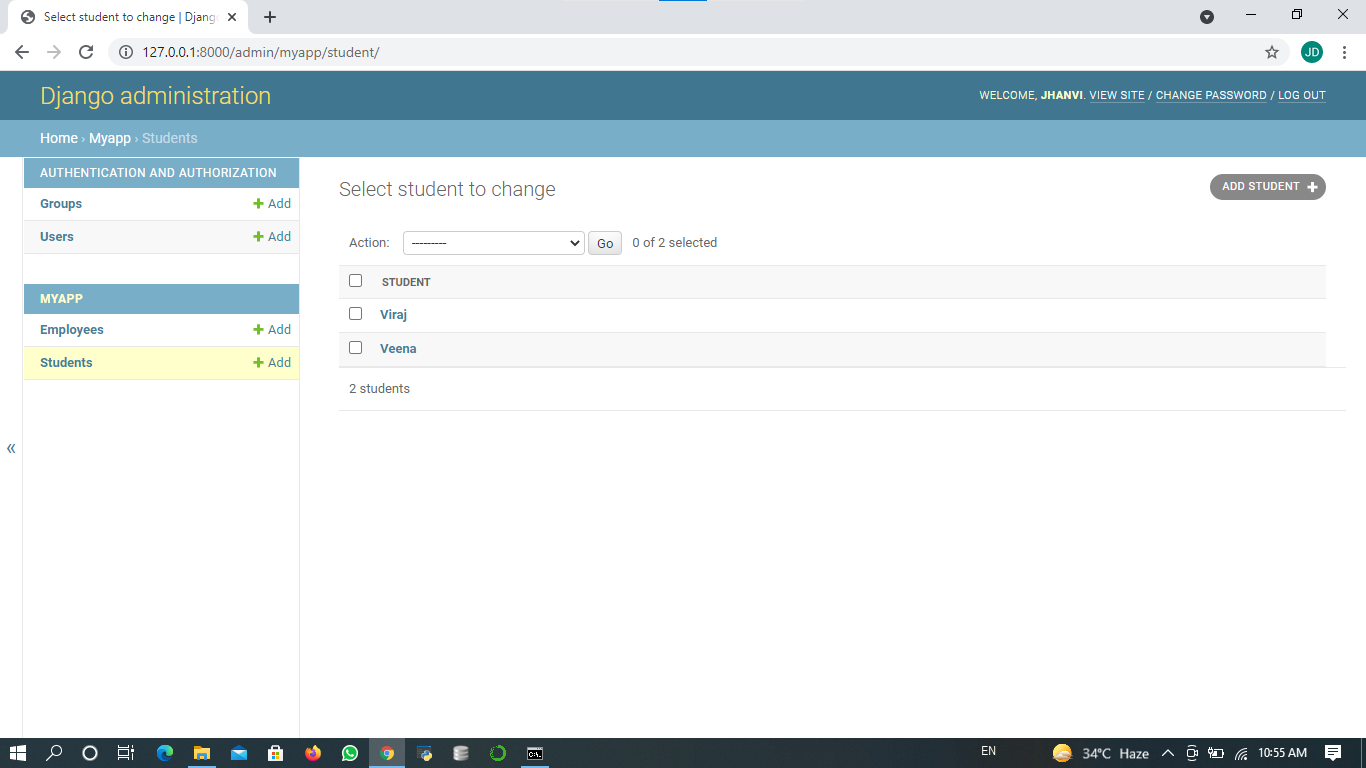
* Form values:

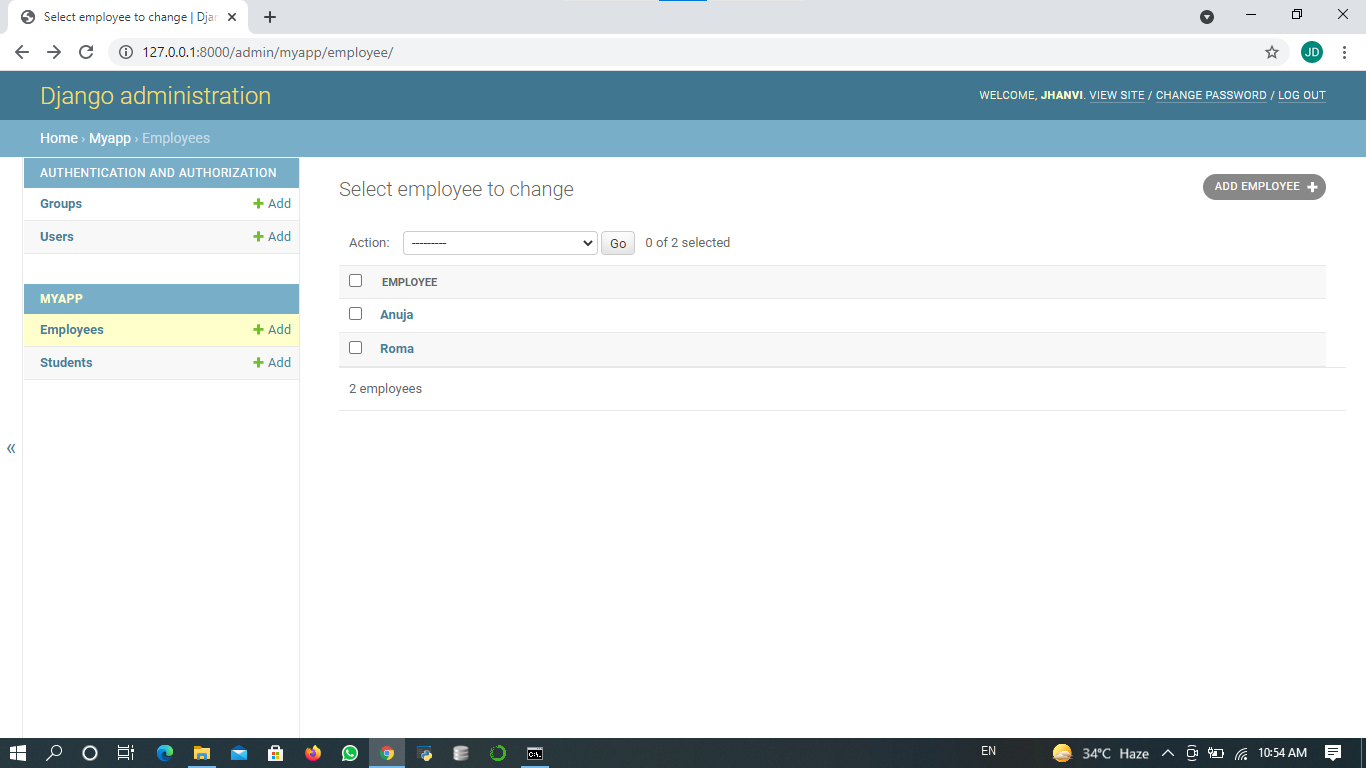


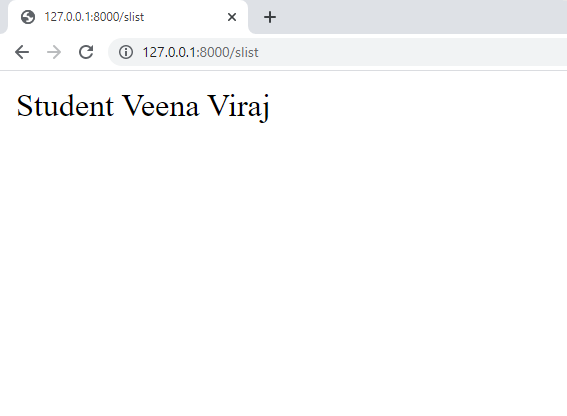
* Admin page with database:



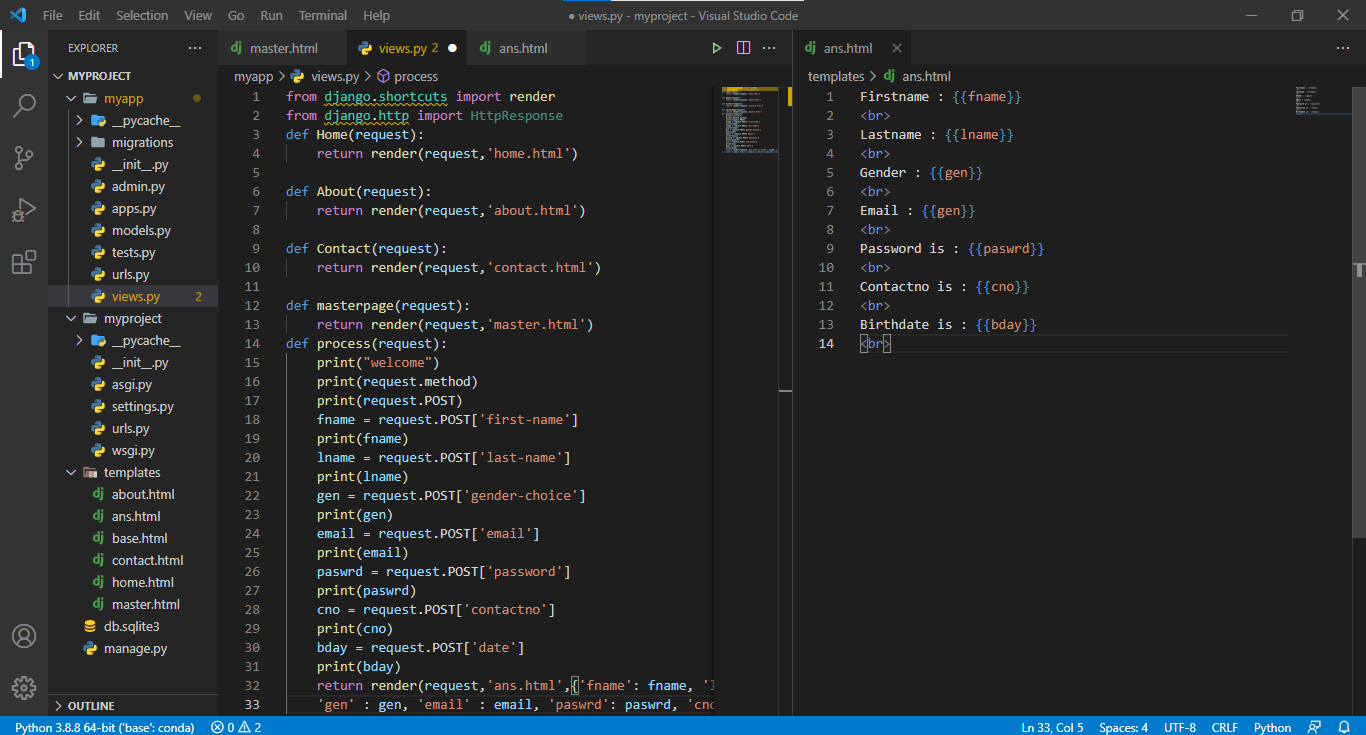
* Student data:

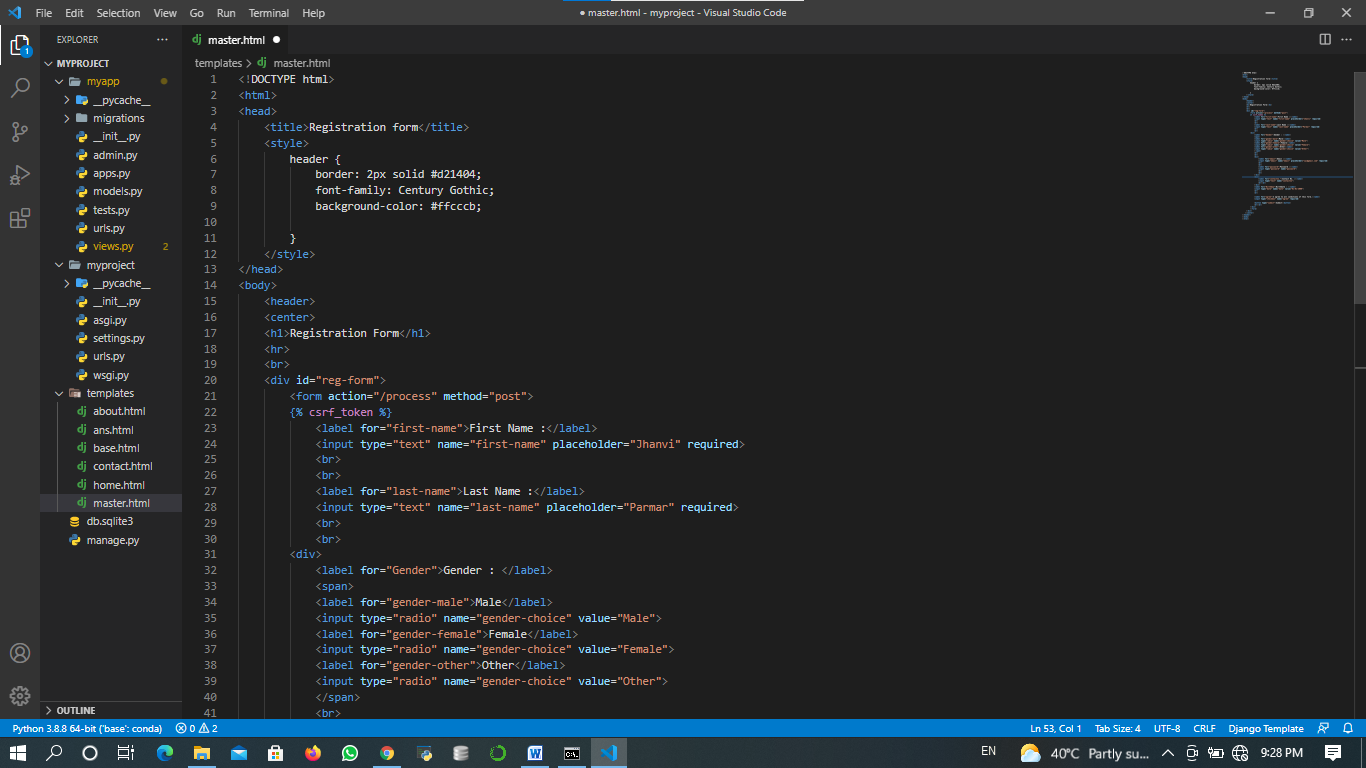


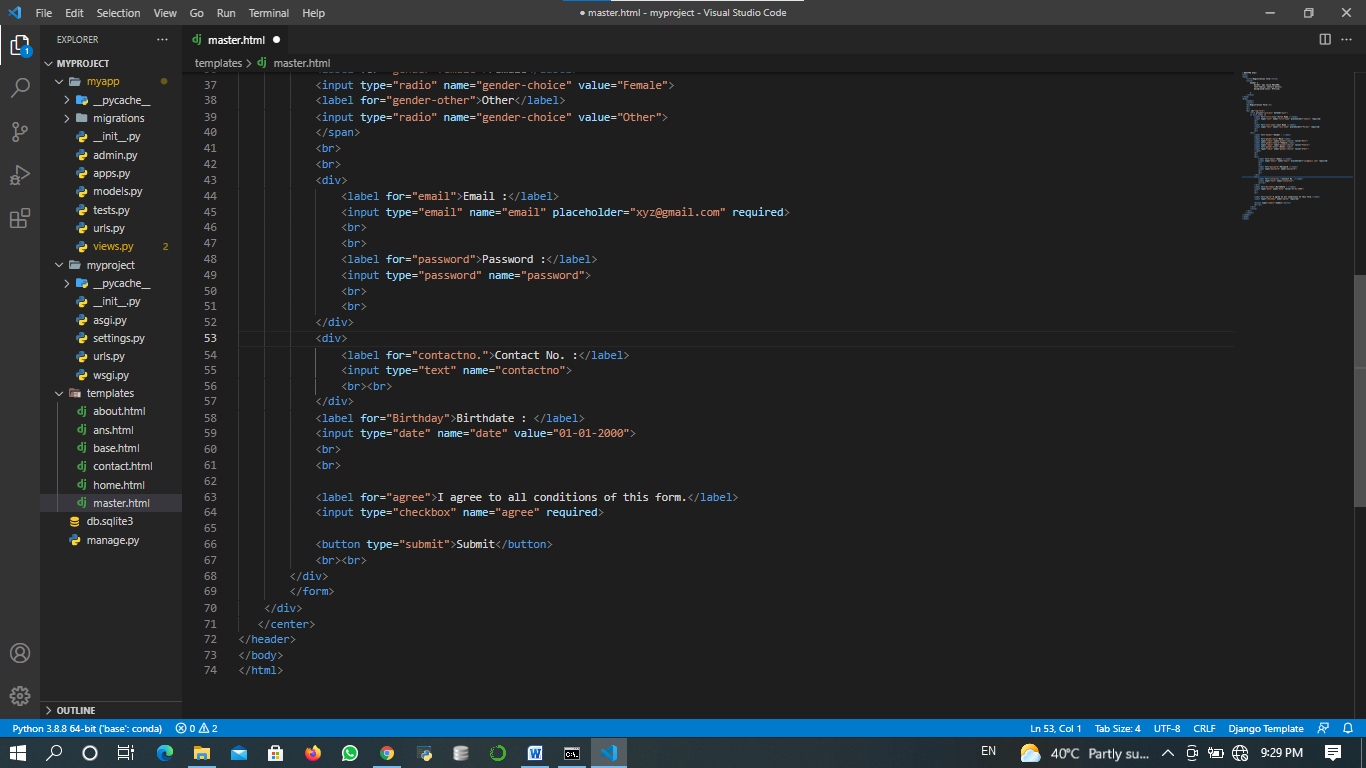
* Employee data:
* slist.html:



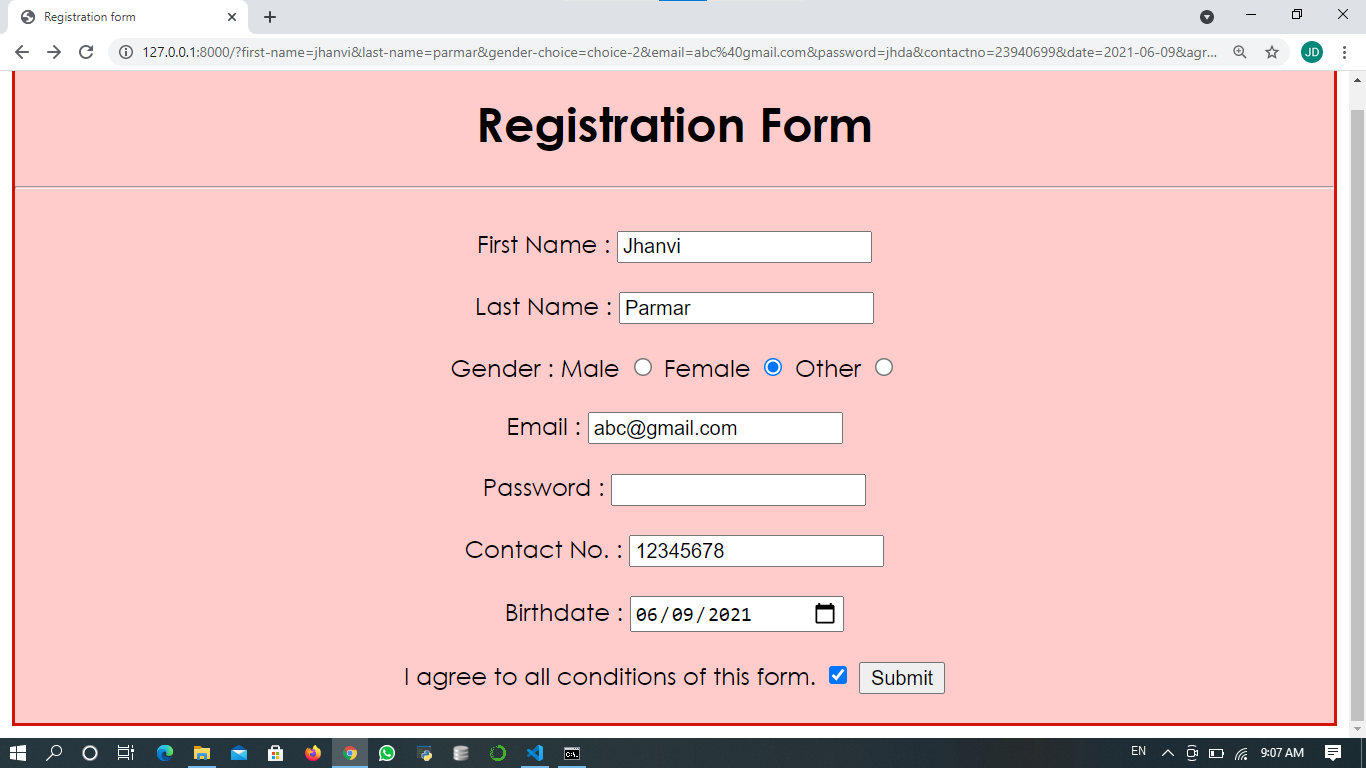
MYPROJECT : Registation Form

* I have designed one registration form using HTML, CSS, Python and django framework displayed below
* views.py and ans.html:

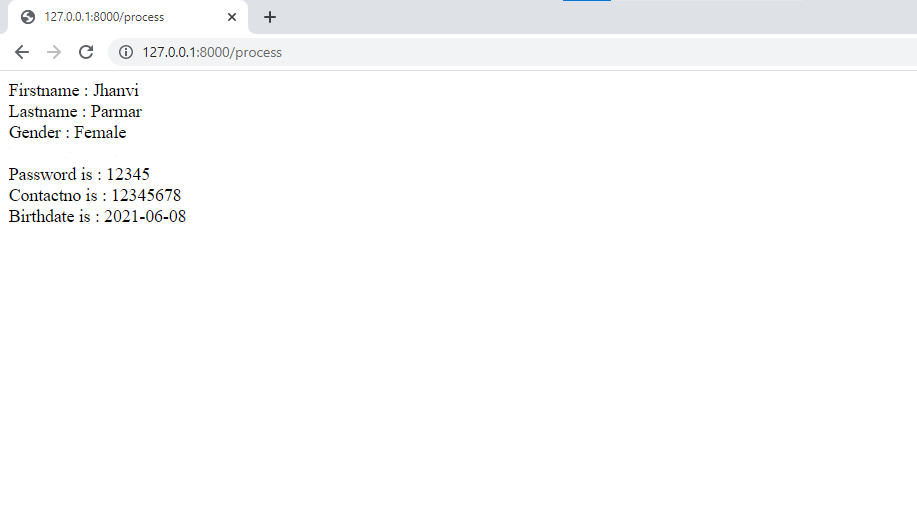
master.html:

. 

* Registration form:



* Output:



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_End\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_