Practical No. 1

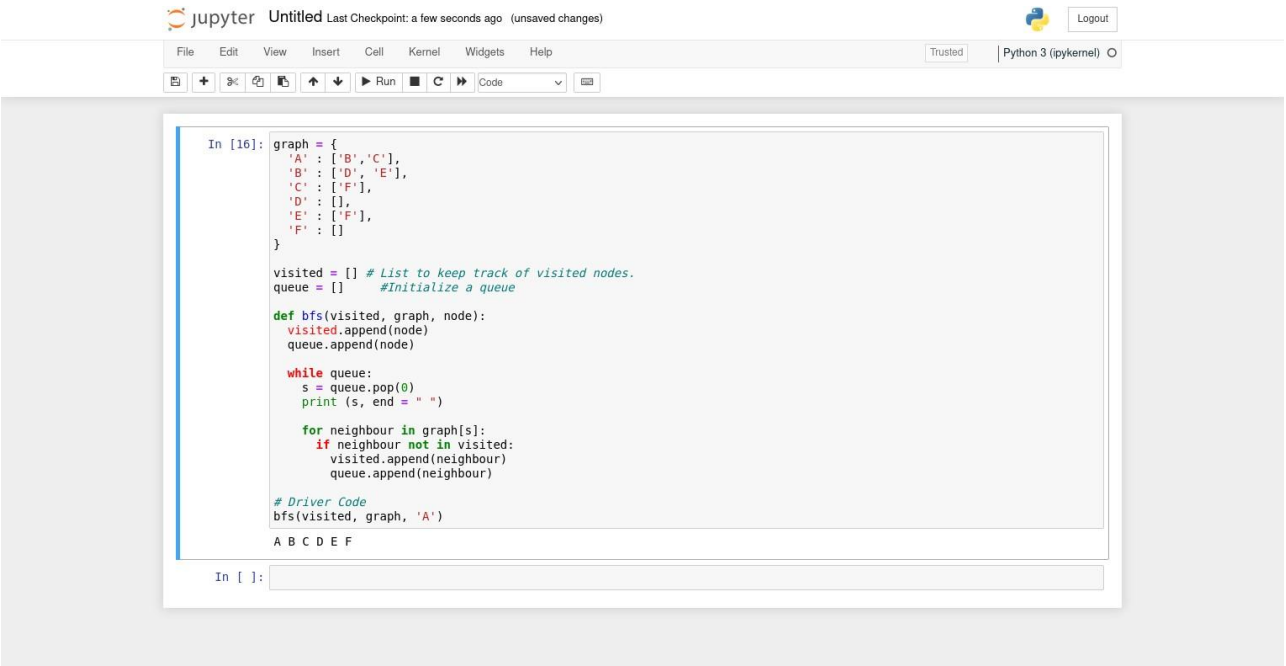
Name: Sakshi Mahendra Jagtap
Div: A
Roll No:49

Title:- Implement DFS and BFS Algorithm. Use and Undirected Graph and develop a Recursive
Algorithm for searching all the vertices of the graph or tree data structure.

Program :-
Breadth First Search(BFS):-

```
graph = {
  'A' : ['B','C'],
  'B' : ['D', 'E'],
  'C' : ['F'],
  'D' : [],
  'E' : ['F'],
  'F' : []
}

visited = [] # List to keep track of visited nodes.
queue = []     #Initialize a queue

def bfs(visited, graph, node):
  visited.append(node)
  queue.append(node)

  while queue:
    s = queue.pop(0)
    print (s, end = " ")

    for neighbour in graph[s]:
      if neighbour not in visited:
        visited.append(neighbour)
        queue.append(neighbour)

# Driver Code
```

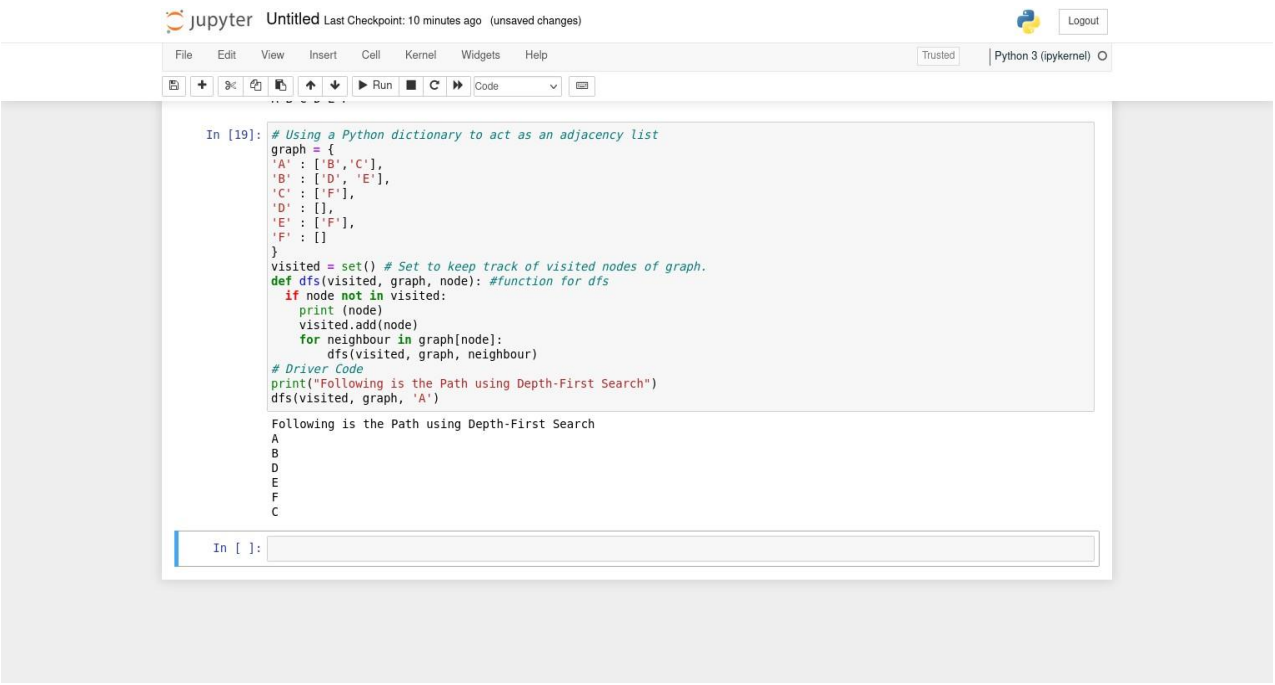
bfs(visited, graph, 'A')



2. Depth-first Search:

```
# Using a Python dictionary to act as an adjacency list
graph = {
'A' : ['B','C'],
'B' : ['D', 'E'],
'C' : ['F'],
'D' : [],
'E' : ['F'],
'F' : []
}
visited = set() # Set to keep track of visited nodes of graph.
def dfs(visited, graph, node): #function for dfs
if node not in visited:
print (node)
visited.add(node)
for neighbour in graph[node]:
dfs(visited, graph, neighbour)
```

# Driver Code
print("Following is the Path using Depth-First Search")
dfs(visited, graph, 'A')



Practical No. 2

Title:- Implement A star Algorithm for any game search problem

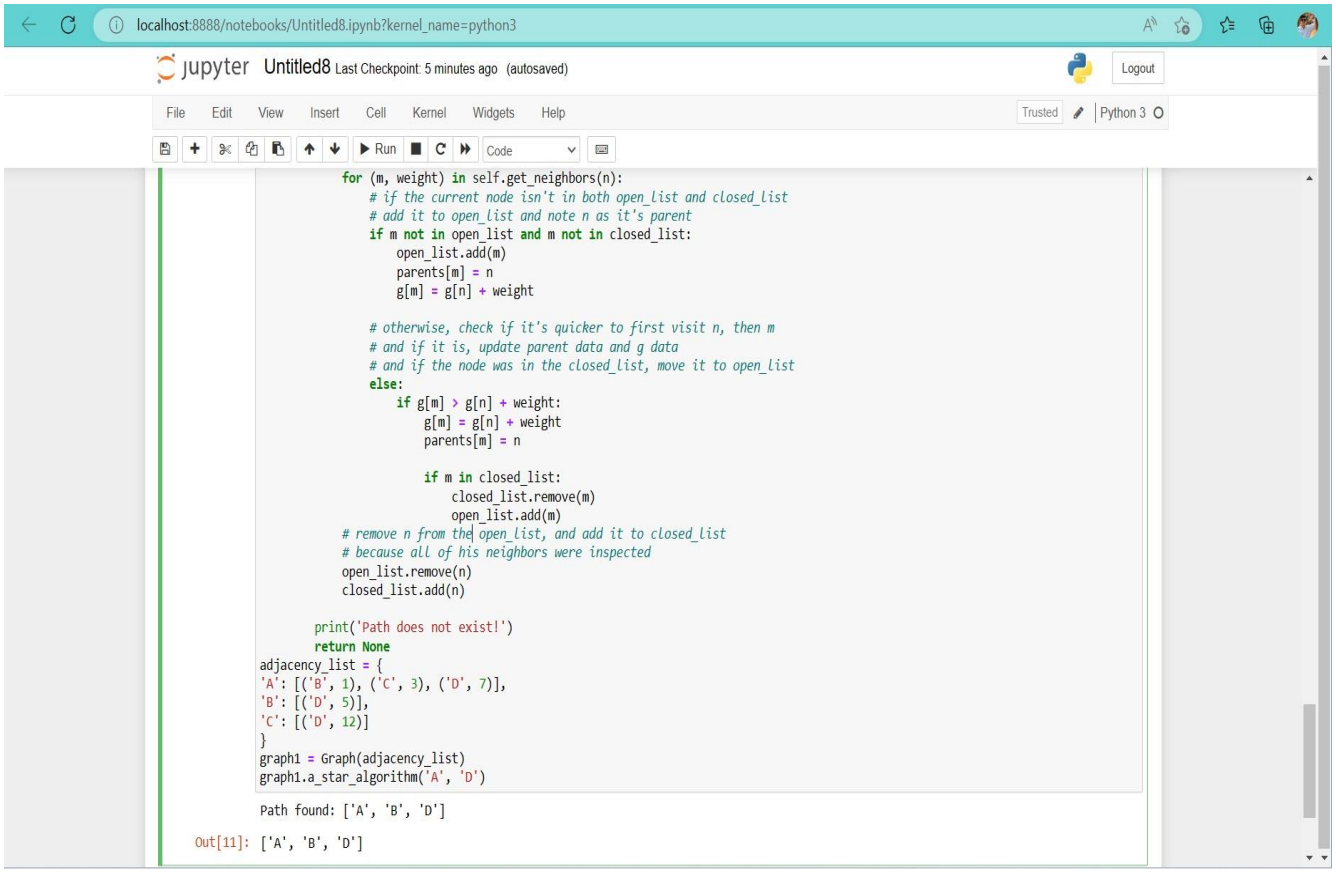
Program :-
A* Algorithm

```
from collections import deque

class Graph:
    # example of adjacency list (or rather map)
    # adjacency_list = {
    # 'A': [('B', 1), ('C', 3), ('D', 7)],
    # 'B': [('D', 5)],
    # 'C': [('D', 12)]
    # }

    def__init__(self, adjacency_list):
        self.adjacency_list = adjacency_list

    def get_neighbors(self, v):
        return self.adjacency_list[v]

    # heuristic function with equal values for all nodes
    def h(self, n):
        H = {
            'A': 1,
            'B': 1,
            'C': 1,
            'D': 1
        }

        return H[n]

    def a_star_algorithm(self, start_node, stop_node):
```

```
        # open_list is a list of nodes which have been visited, but who's
neighbors
        # haven't all been inspected, starts off with the start node
        # closed_list is a list of nodes which have been visited
        # and who's neighbors have been inspected
        open_list = set([start_node])
        closed_list = set([])

        # g contains current distances from start_node to all other nodes
        # the default value (if it's not found in the map) is +infinity
        g = {}

        g[start_node] = 0

        # parents contains an adjacency map of all nodes
        parents = {}
        parents[start_node] = start_node

        while len(open_list) > 0:
            n = None

            # find a node with the lowest value of f() - evaluation function
            for v in open_list:
                if n == None or g[v] + self.h(v) < g[n] + self.h(n):
                    n = v;

            if n == None:
                print('Path does not exist!')
                return None

            # if the current node is the stop_node
            # then we begin reconstructin the path from it to the start_node
            if n == stop_node:
                reconst_path = []

                while parents[n] != n:
                    reconst_path.append(n)
                    n = parents[n]
```

```
                reconst_path.append(start_node)

                reconst_path.reverse()

                print('Path found: {}'.format(reconst_path))
                return reconst_path

            # for all neighbors of the current node do
            for (m, weight) in self.get_neighbors(n):
                # if the current node isn't in both open_list and closed_list
                # add it to open_list and note n as it's parent
                if m not in open_list and m not in closed_list:
                    open_list.add(m)
                    parents[m] = n
                    g[m] = g[n] + weight

                # otherwise, check if it's quicker to first visit n, then m
                # and if it is, update parent data and g data
                # and if the node was in the closed_list, move it to open_list
                else:
                    if g[m] > g[n] + weight:
                        g[m] = g[n] + weight
                        parents[m] = n

                        if m in closed_list:
                            closed_list.remove(m)
                            open_list.add(m)

            # remove n from the open_list, and add it to closed_list
            # because all of his neighbors were inspected
            open_list.remove(n)
            closed_list.add(n)

        print('Path does not exist!')
        return None
adjacency_list = {
'A': [('B', 1), ('C', 3), ('D', 7)],
'B': [('D', 5)],
'C': [('D', 12)]
```

}
graph1 = Graph(adjacency_list)
graph1.a_star_algorithm('A', 'D')

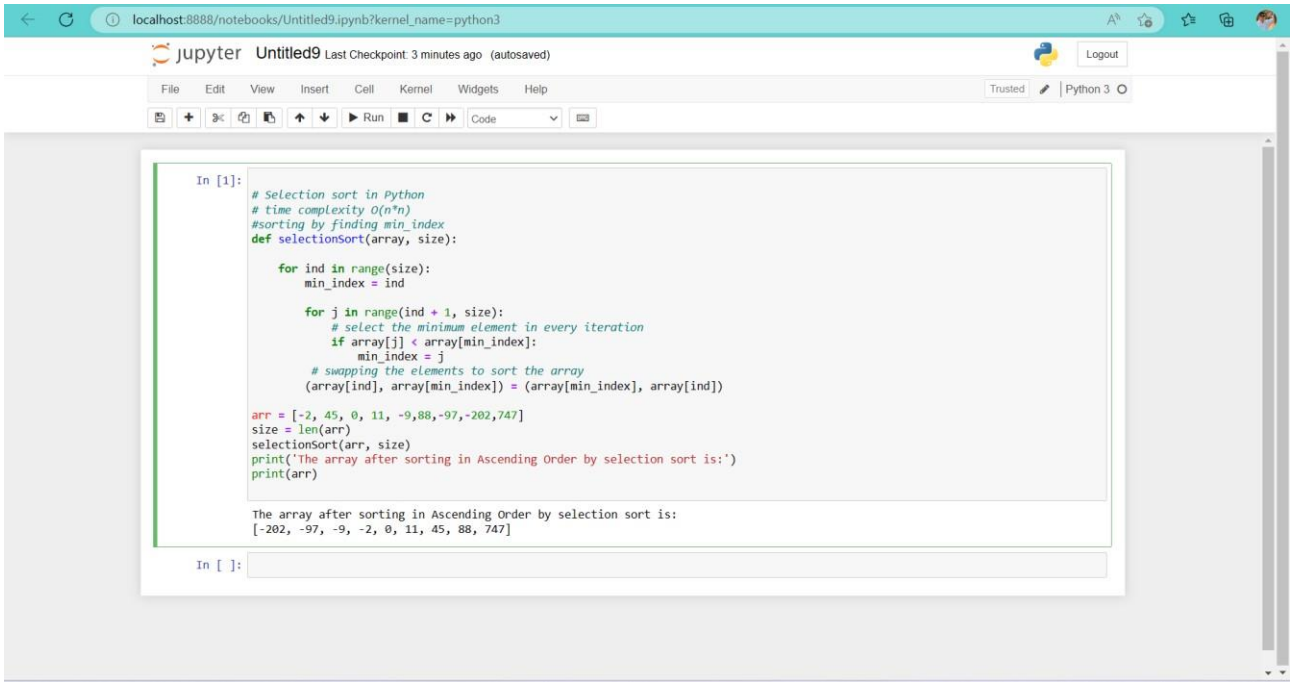


Practical No. 3

Title:- Implement Greedy search algorithm for selection sort

Program :-
Greedy search Algorithm for selection sort program

```
# Selection sort in Python
# time complexity O(n*n)
#sorting by finding min_index
def selectionSort(array, size):

    for ind in range(size):
        min_index = ind

        for j in range(ind + 1, size):
            # select the minimum element in every iteration
            if array[j] < array[min_index]:
                min_index = j
         # swapping the elements to sort the array
        (array[ind], array[min_index]) = (array[min_index], array[ind])

arr = [-2, 45, 0, 11, -9,88,-97,-202,747]
size = len(arr)
selectionSort(arr, size)
print('The array after sorting in Ascending Order by selection sort is:')


print(arr)
```


localhost:8888/notebooks/Untitled9.ipynb?kernel_name=python3
jupyter Untitled9 Last Checkpoint: 3 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
Run Code
In [1]:
Selection sort in Python
time complexity O(n*n)
#sorting by finding min_index
def selectionSort(array, size):
for ind in range(size):
min_index = ind
for j in range(ind + 1, size):
# select the minimum element in every iteration
if array[j] < array[min_index]:
min_index = j
# swapping the elements to sort the array
(array[ind], array[min_index]) = (array[min_index], array[ind])
arr = [-2, 45, 0, 11, -9,88,-97,-202,747]
size = len(arr)
selectionSort(arr, size)
print('The array after sorting in Ascending Order by selection sort is:')
print(arr)
The array after sorting in Ascending Order by selection sort is:
[-202, -97, -9, -2, 0, 11, 45, 88, 747]
In [ ]:

Practical No. 4

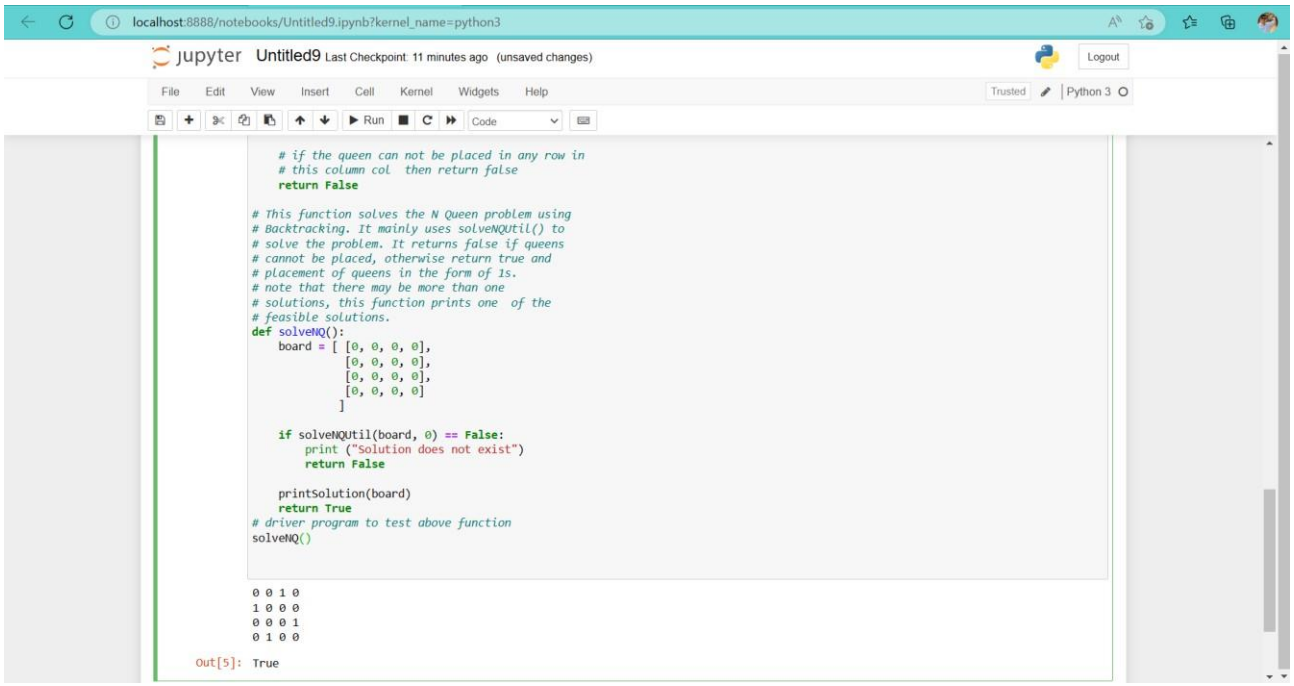
Title:- Implement a solution for a constraint satisfaction problem using branch and bound and bracktracking for n-queens problem or a graph coloring problem

Program :-
n-queens problem

```
# Python program to solve N Queen
# Problem using backtracking

global N
N = 4

def printSolution(board):
    for i in range(N):
        for j in range(N):
            print (board[i][j],end=' ')
        print()


# A utility function to check if a queen can
# be placed on board[row][col]. Note that this
# function is called when "col" queens are
# already placed in columns from 0 to col -1.
# So we need to check only left side for
# attacking queens
def isSafe(board, row, col):

    # Check this row on left side
    for i in range(col):
        if board[row][i] == 1:
            return False

    # Check upper diagonal on left side
```

```
    for i, j in zip(range(row, -1, -1), range(col, -1, -1)):
        if board[i][j] == 1:
            return False

    # Check lower diagonal on left side
    for i, j in zip(range(row, N, 1), range(col, -1, -1)):
        if board[i][j] == 1:
            return False

    return True

def solveNQUtil(board, col):
    # base case: If all queens are placed
    # then return true
    if col >= N:
        return True

    # Consider this column and try placing
    # this queen in all rows one by one
    for i in range(N):

        if isSafe(board, i, col):
            # Place this queen in board[i][col]
            board[i][col] = 1

            # recur to place rest of the queens
            if solveNQUtil(board, col + 1) == True:
                return True

            # If placing queen in board[i][col
            # doesn't lead to a solution, then
            # queen from board[i][col]
            board[i][col] = 0

    # if the queen can not be placed in any row in
    # this column col  then return false
    return False

# This function solves the N Queen problem using
```

```
# Backtracking. It mainly uses solveNQUtil() to
# solve the problem. It returns false if queens
# cannot be placed, otherwise return true and
# placement of queens in the form of 1s.
# note that there may be more than one
# solutions, this function prints one of the
# feasible solutions.
def solveNQ():
    board = [ [0, 0, 0, 0],
              [0, 0, 0, 0],
              [0, 0, 0, 0],
              [0, 0, 0, 0]
             ]

    if solveNQUtil(board, 0) == False:
        print ("Solution does not exist")
        return False

    printSolution(board)
    return True
# driver program to test above function
solveNQ()
```



Practical No. 5

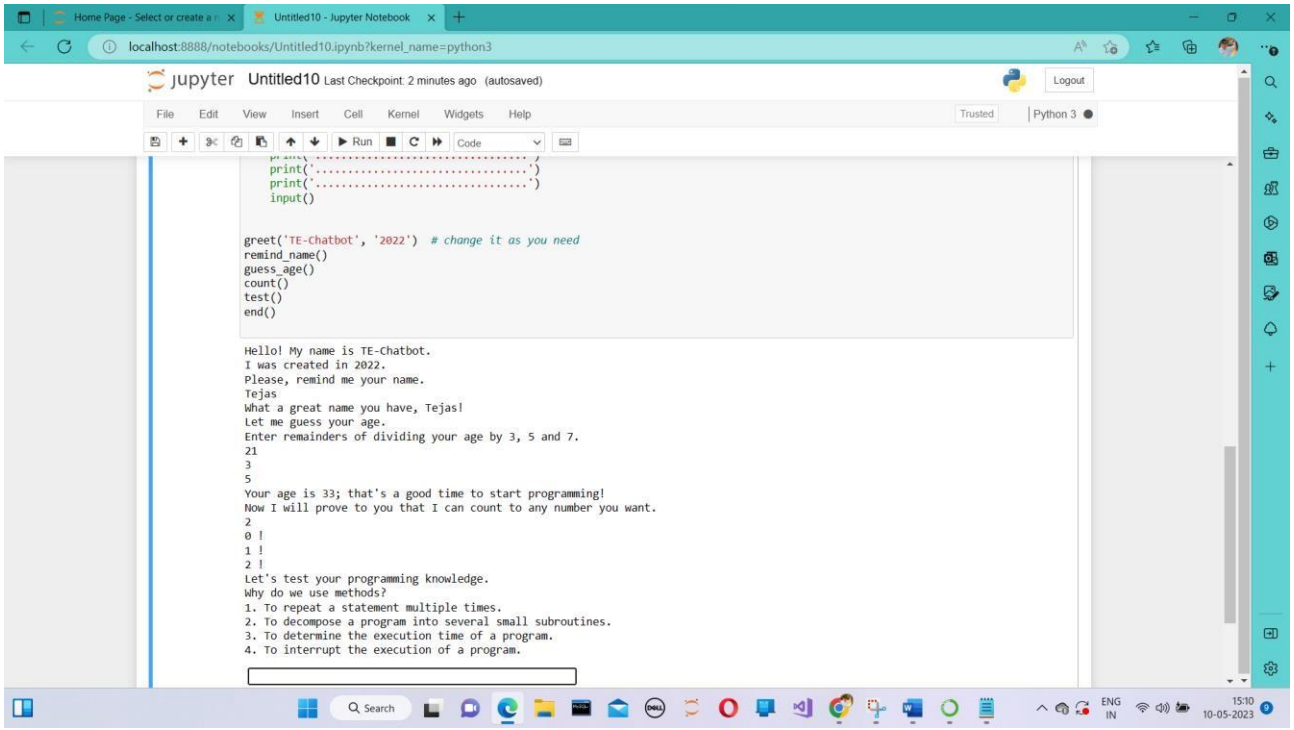
Title:- Develop an elementary chatbot for any suitable customer interaction application

Program :-
Chatbot Program

```
def greet(bot_name, birth_year):
    print("Hello! My name is {0}.".format(bot_name))
    print("I was created in {0}.".format(birth_year))


def remind_name():
    print('Please, remind me your name.')
    name = input()
    print("What a great name you have, {0}!".format(name))


def guess_age():
    print('Let me guess your age.')
    print('Enter remainders of dividing your age by 3, 5 and 7.')

    rem3 = int(input())
    rem5 = int(input())
    rem7 = int(input())
    age = (rem3 * 70 + rem5 * 21 + rem7 * 15) % 105

    print("Your age is {0}; that's a good time to start
programming!".format(age))


def count():
    print('Now I will prove to you that I can count to any number you
want.')
    num = int(input())
```

```
    counter = 0
    while counter <= num:
        print("{0} !".format(counter))
        counter += 1


def test():
    print("Let's test your programming knowledge.")
    print("Why do we use methods?")
    print("1. To repeat a statement multiple times.")
    print("2. To decompose a program into several small subroutines.")
    print("3. To determine the execution time of a program.")
    print("4. To interrupt the execution of a program.")

    answer = 2
    guess = int(input())
    while guess != answer:
        print("Please, try again.")
        guess = int(input())

    print('Completed, have a nice day!')
    print('. ..............................')
    print('. ..............................')
    print('. ..............................')


def end():
    print('Congratulations, have a nice day!')
    print('. ..............................')
    print('. ..............................')
    print('. ..............................')
    input()


greet('TE-Chatbot', '2022') # change it as you need
remind_name()
guess_age()
count()
```

test()
end()


Home Page - Select or create a
Untitled10 - Jupyter Notebook
localhost:8888/notebooks/Untitled10.ipynb?kernel_name=python3
jupyter Untitled10 Last Checkpoint: 2 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3
Run
Code
print('.....')
print('.....')
print('.....')
input()
greet('TE-Chatbot', '2022') # change it as you need
remind_name()
guess_age()
count()
test()
end()
Hello! My name is TE-Chatbot.
I was created in 2022.
Please, remind me your name.
Tejas
What a great name you have, Tejas!
Let me guess your age.
Enter remainders of dividing your age by 3, 5 and 7.
21
3
5
Your age is 33; that's a good time to start programming!
Now I will prove to you that I can count to any number you want.
2
0 !
1 !
2 !
Let's test your programming knowledge.
Why do we use methods?
1. To repeat a statement multiple times.
2. To decompose a program into several small subroutines.
3. To determine the execution time of a program.
4. To interrupt the execution of a program.
Search
ENG
IN
15:10
10-05-2023

Practical No. 6

Title:- Implement of Expert system Help Desk Management System

Program :-
Help Desk Management System

```
# Define a dictionary of common problems and their solutions
problem_dict = {
    "Printer not working": "Check that it's turned on and connected to the network",
    "Can't log in": "Make sure you're using the correct username and password",
    "Software not installing": "Check that your computer meets the system requirements",
    "Internet connection not working": "Restart your modem or router",
    "Email not sending": "Check that you're using the correct email server settings"
}

# Define a function to handle user requests
def handle_request(user_input):
    if user_input.lower() == "exit":
        return "Goodbye!"
    elif user_input in problem_dict:
        return problem_dict[user_input]
    else:
        return "I'm sorry, I don't know how to help with that problem."

# Main loop to prompt user for input
while True:
    user_input = input("What's the problem? Type 'exit' to quit. ")
    response = handle_request(user_input)
    print(response)
```

Output:-

```
What's the problem? Type 'exit' to quit. Printer not working
Check that it's turned on and connected to the network
What's the problem? Type 'exit' to quit. Can't log in
Make sure you're using the correct username and password
What's the problem? Type 'exit' to quit. Software not installing
Check that your computer meets the system requirements
What's the problem? Type 'exit' to quit. Internet connection not working
Restart your modem or router
What's the problem? Type 'exit' to quit. Email not sending
I'm sorry, I don't know how to help with that problem.
What's the problem? Type 'exit' to quit. exit
Goodbye!
What's the problem? Type 'exit' to quit.
```

PRACTICAL NO : 7

Title : Case study on Amazon EC2 and learn about Amazon EC2 web services.

STEPS TO STUDY AMAZON EC2 :

How to Use AWS EC2

Step 1 − Sign-in to AWS account and open IAM console by using the following link **https://console.aws.amazon.com/iam/.**

Step 2 − In the navigation Panel, create/view groups and follow the instructions.

Step 3 − Create IAM user. Choose users in the navigation pane. Then create new users and add users to the groups.

Step 4 − Create a Virtual Private Cloud using the following instructions.

Open the Amazon VPC console by using the following link − https://console.aws.amazon.com/vpc/

Select VPC from the navigation panel. Then select the same region in which we have created key-pair.

Select start VPC wizard on VPC dashboard.

Select VPC configuration page and make sure that VPC with single subnet is selected. The choose Select.

VPC with a single public subnet page will open. Enter the VPC name in the name field and leave other configurations as default.

Select create VPC, then select Ok.

Step 5 − Create WebServerSG security groups and add rules using the following instructions.

On the VPC console, select Security groups in the navigation panel.

Select create security group and fill the required details like group name, name tag, etc.

Select your VPC ID from the menu. Then select yes, create button.

Now a group is created. Select the edit option in the inbound rules tab to create rules.

Step 6 − Launch EC2 instance into VPC using the following instructions.

Open EC2 console by using the following link − https://console.aws.amazon.com/ec2/

Select launch instance option in the dashboard.

A new page will open. Choose Instance Type and provide the configuration. Then select Next: Configure Instance Details.

A new page will open. Select VPC from the network list. Select subnet from the subnet list and leave the other settings as default.

Click Next until the Tag Instances page appears.

Step 7 − On the Tag Instances page, provide a tag with a name to the instances. Select Next: Configure Security Group.

Step 8 − On the Configure Security Group page, choose the Select an existing security group option. Select the WebServerSG group that we created previously, and then choose Review and Launch.

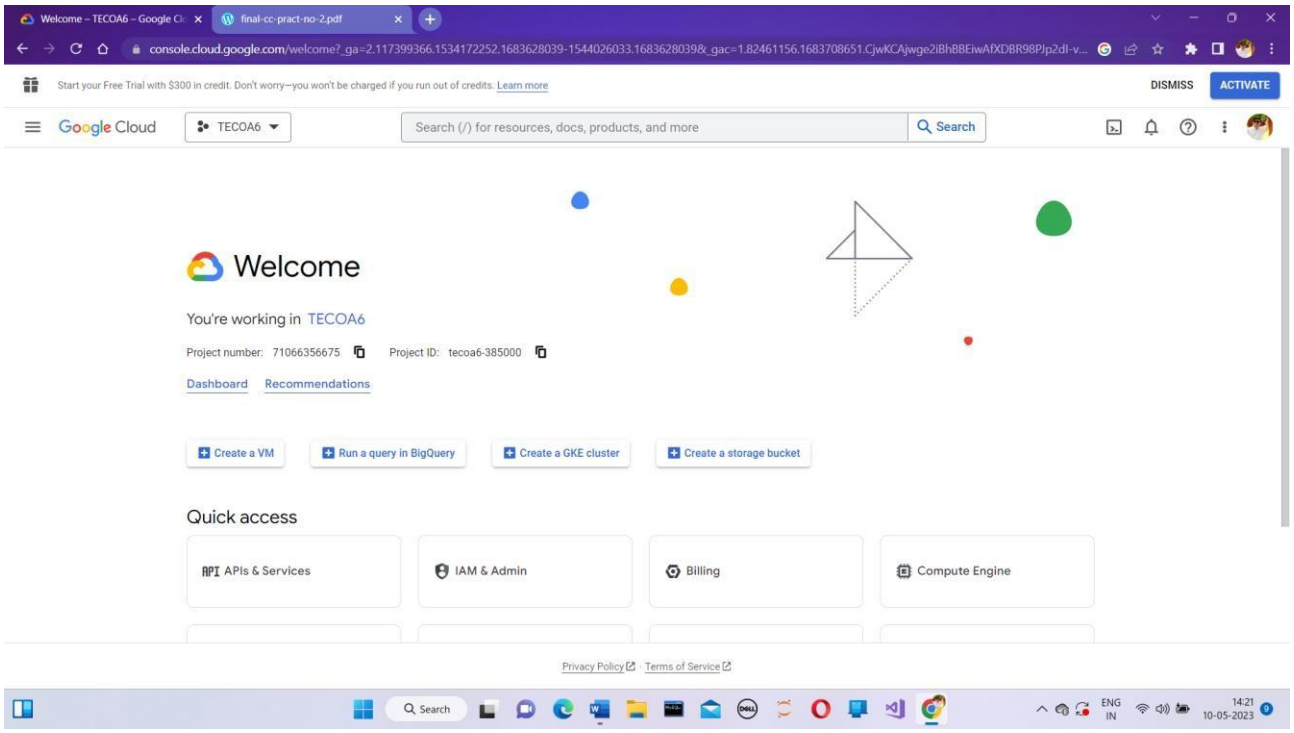
Step 9 − Check Instance details on Review Instance Launch page then click the Launch button.

Step 10 − A pop up dialog box will open. Select an existing key pair or create a new key pair. Then select the acknowledgement check box and click the Launch Instances button.

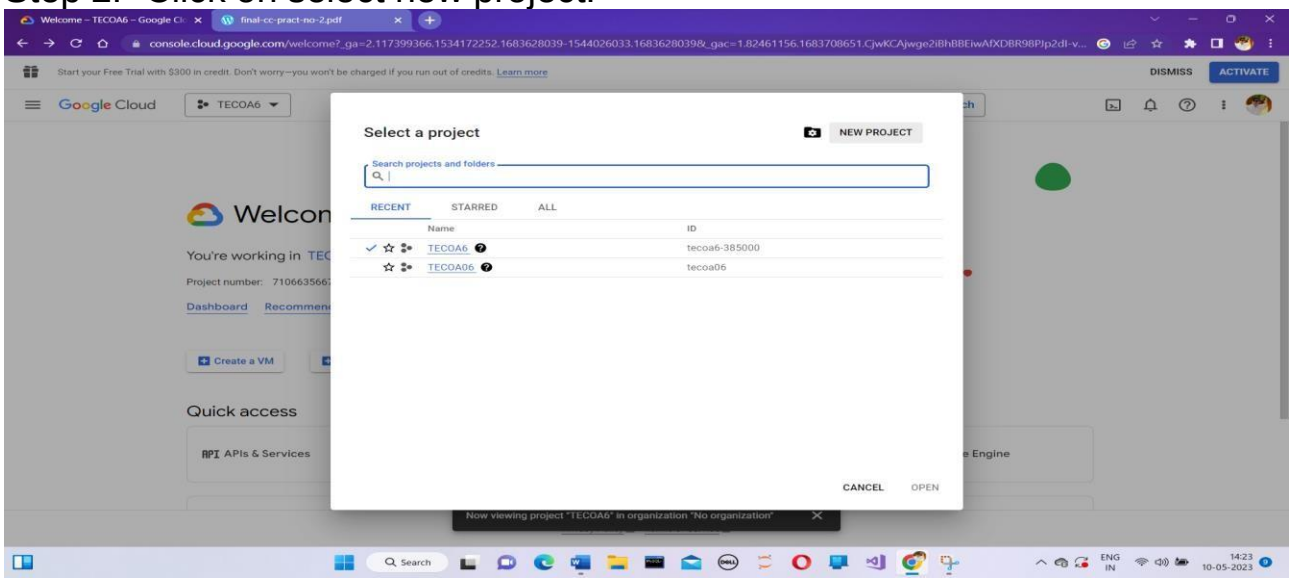
Practical No. 8

Title:- Installation and configure Google app engine

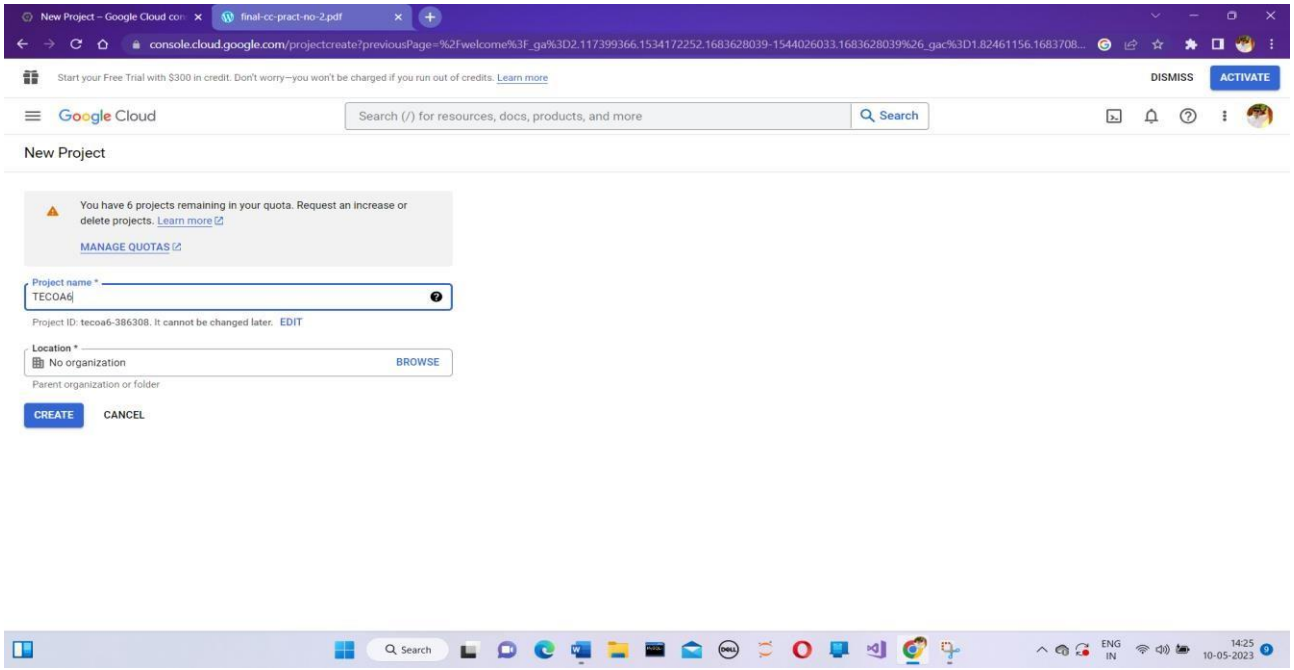
Step 1:- Search Google Cloud Platform in a any search engine& Click on Console.



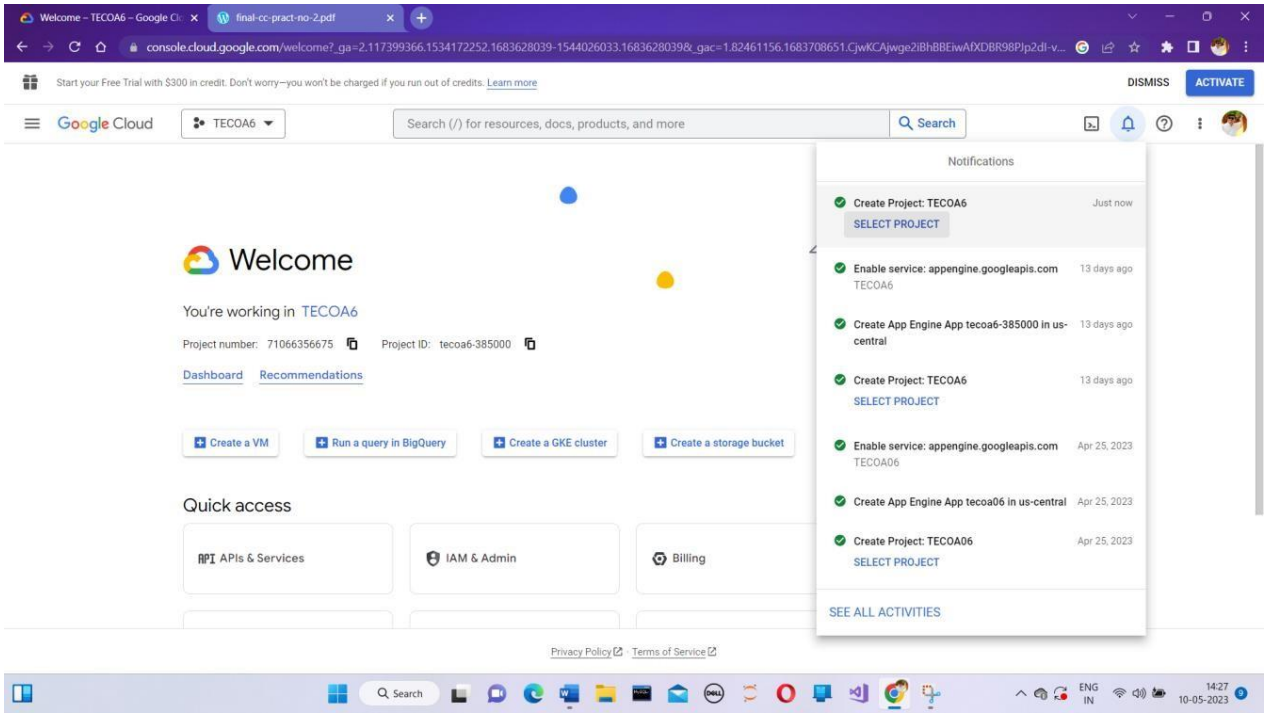
Step 2:- Click on select new project.



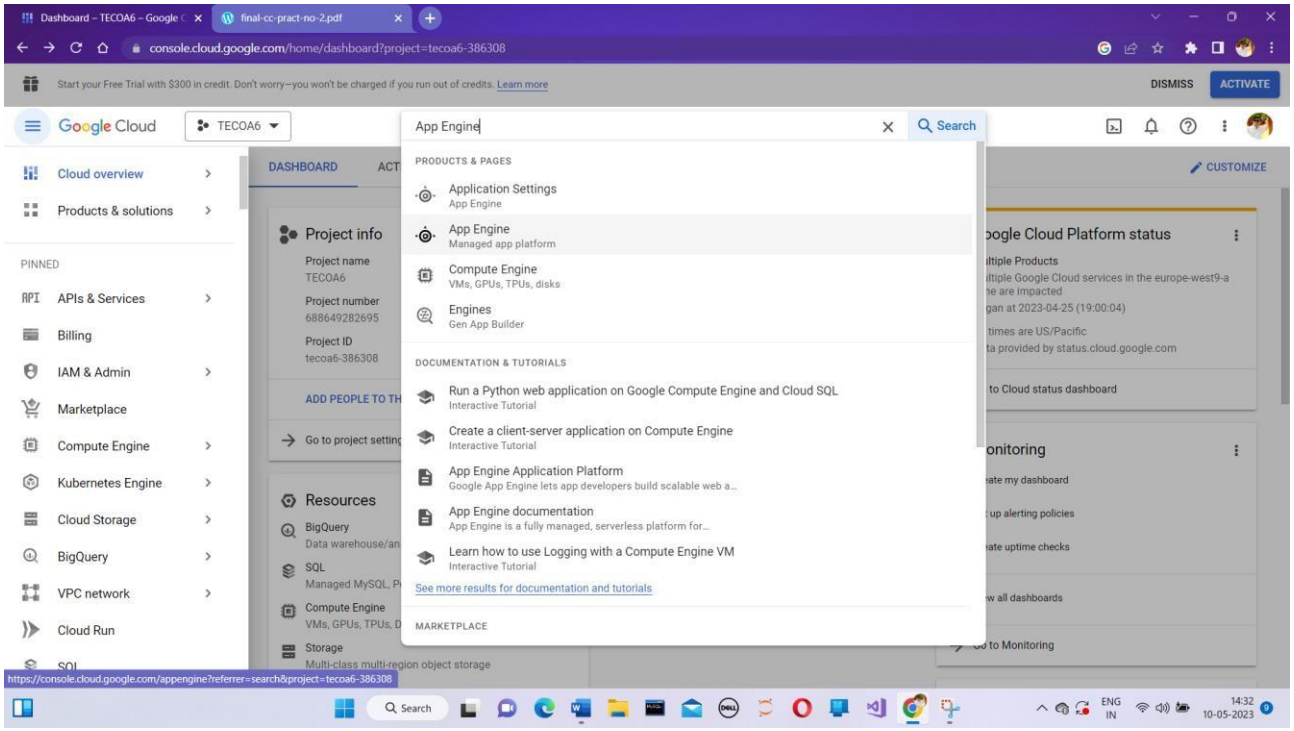
Step3:- Give Project name and click on create.



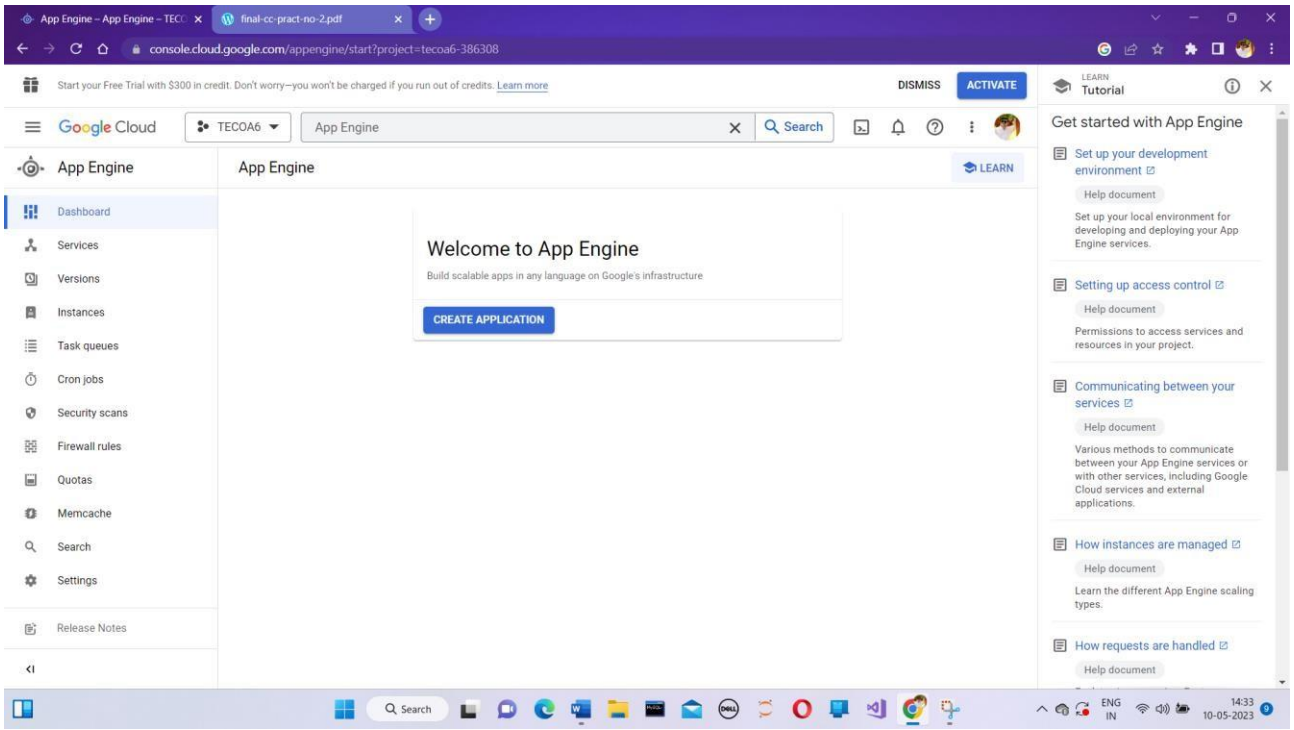
Step4:- Click on select project



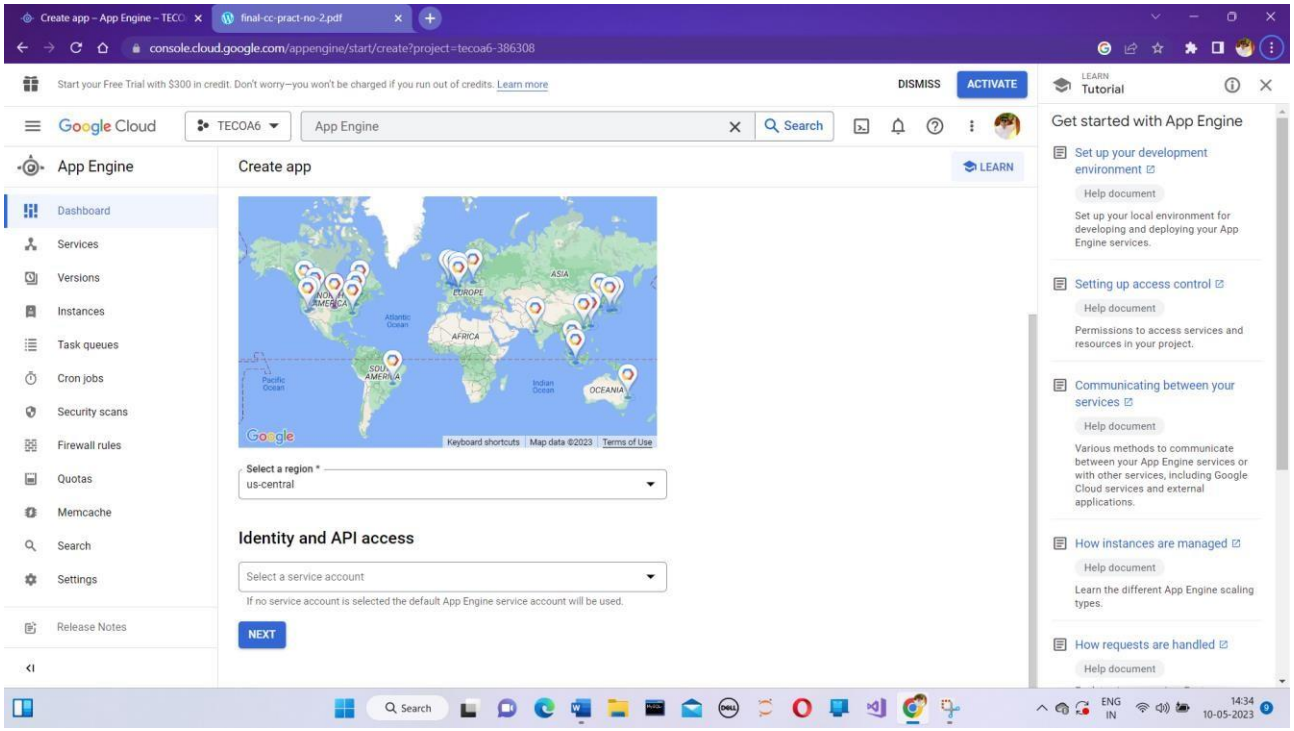
Step5:- In a search bar type search App Engine



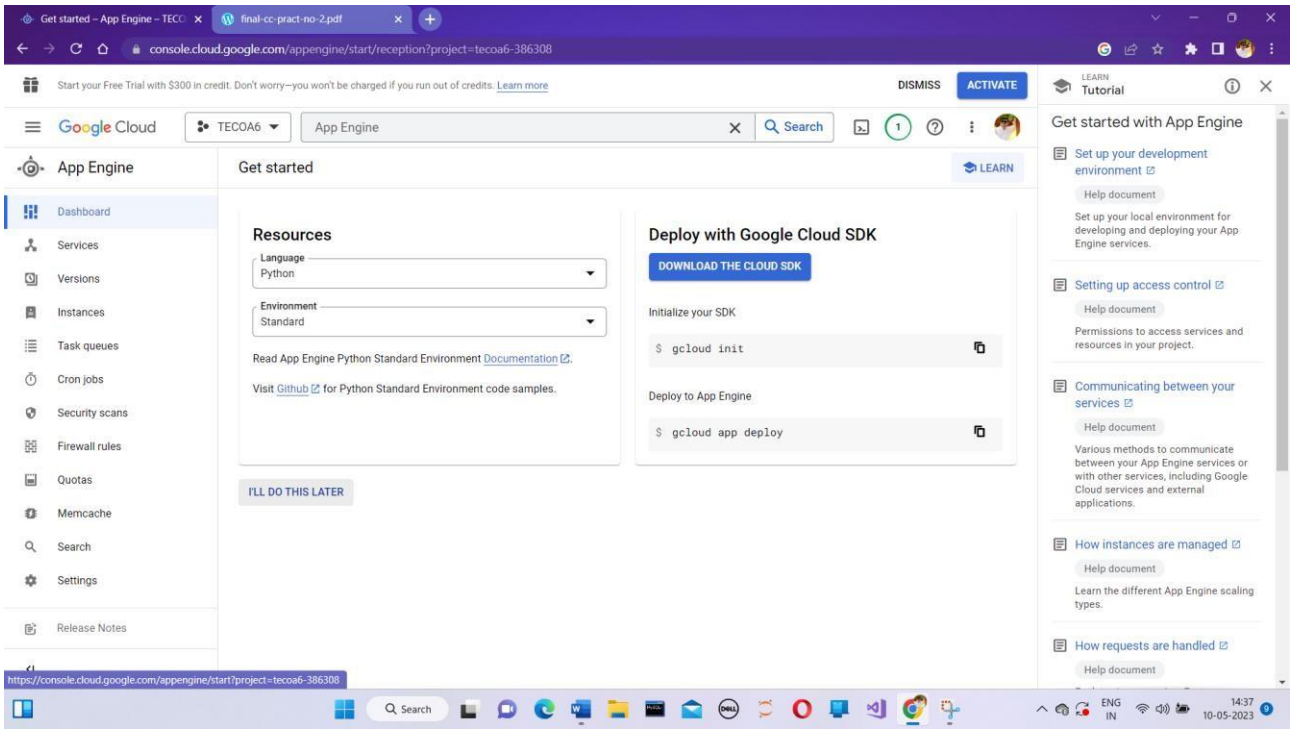
Step 6:-Click on App Engine and Following screen will appear& Click on Create Application.



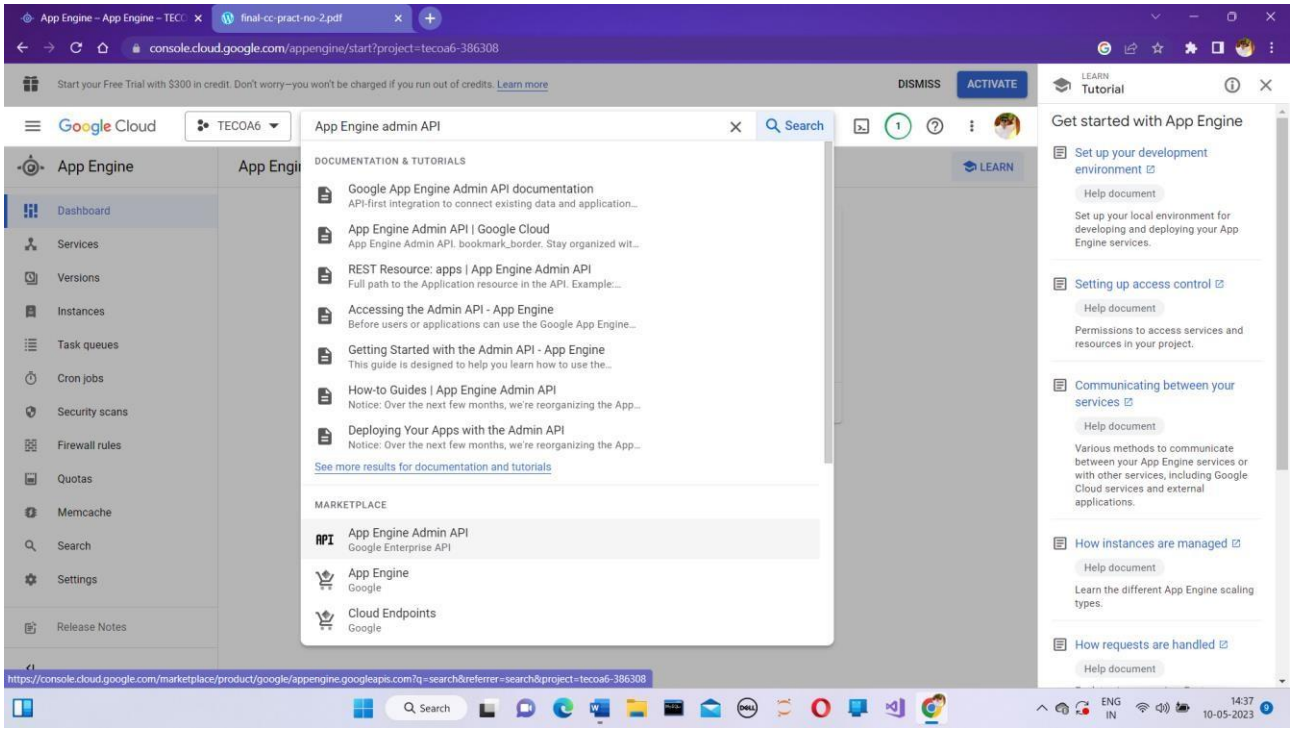
Step 7:- Click on next



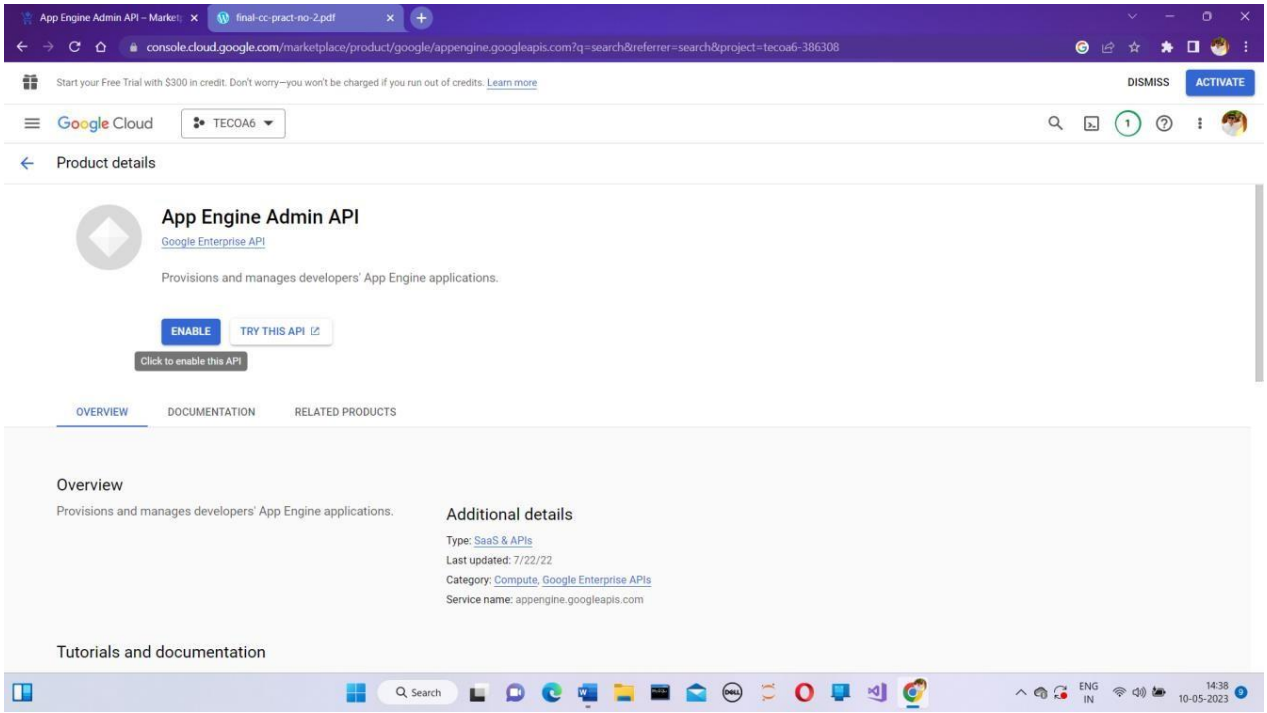
Step 8:- Scroll down and click on l'll do this later.



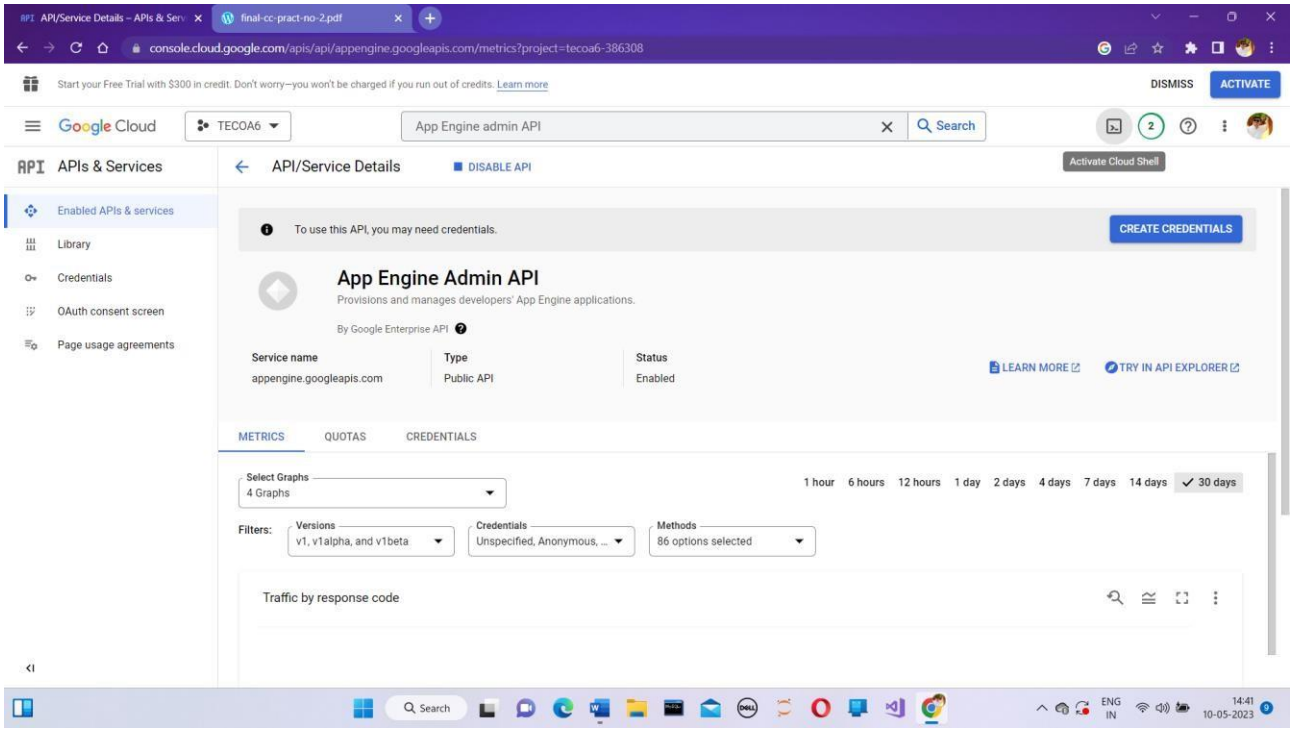
Sterp 9:- In search bar type App Engine Admin API.



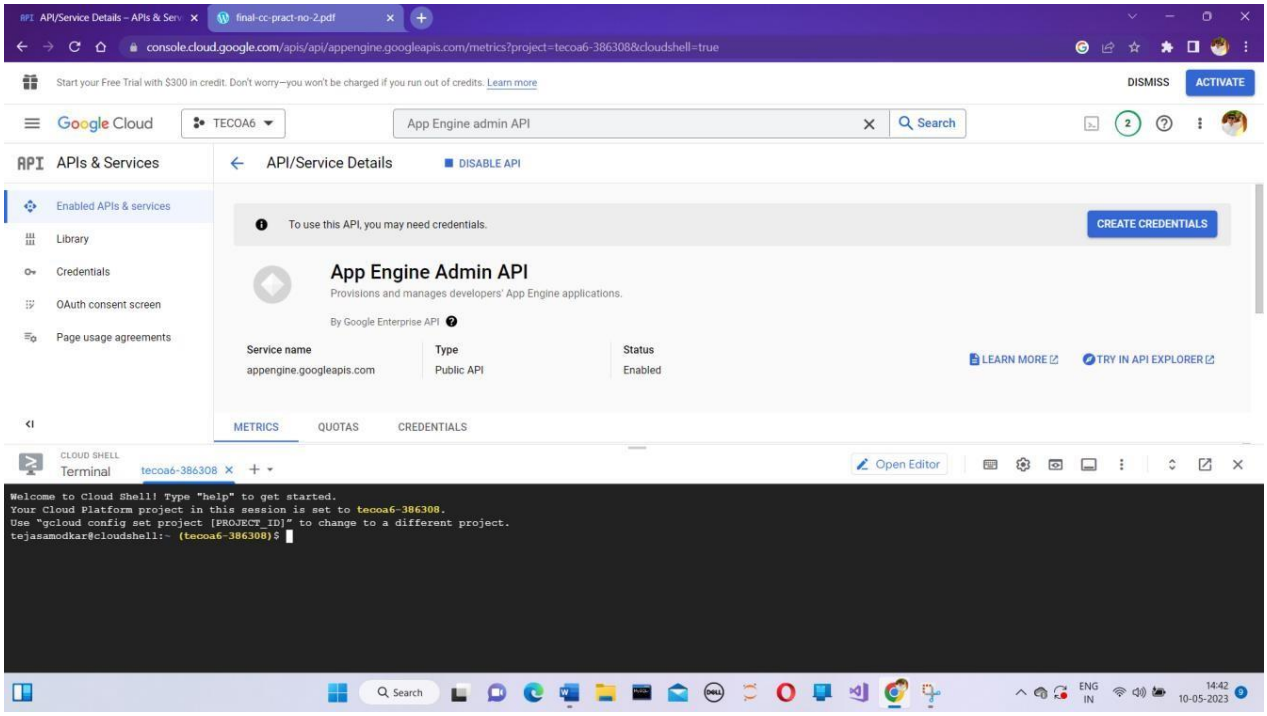
Step 10:- Click Enables



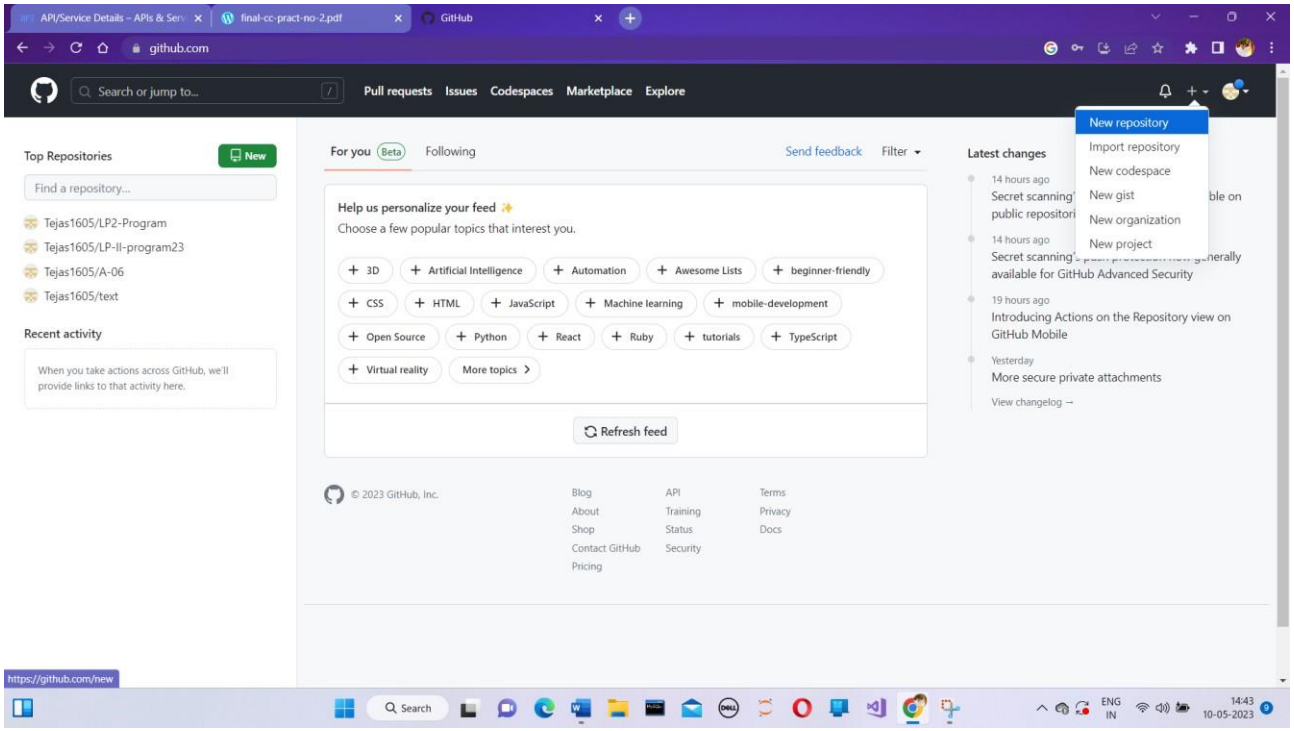
Step 11:- Click Activate Cloud Shell:-



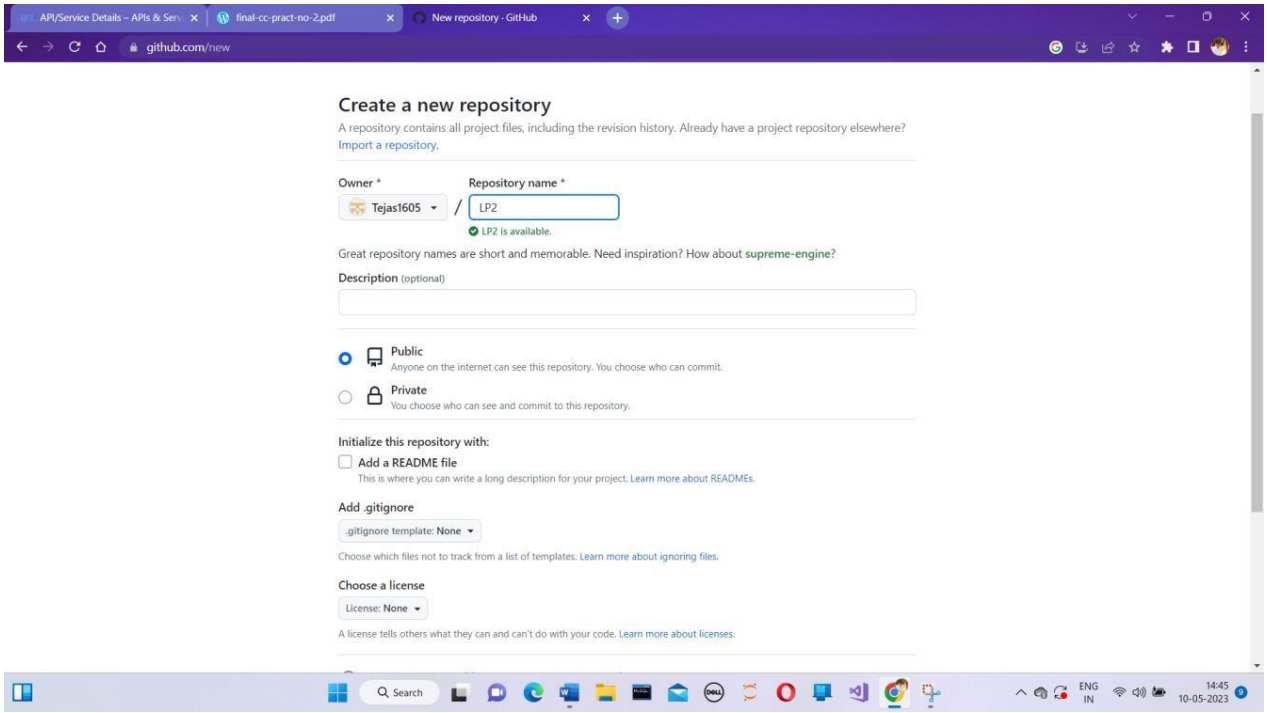
Step 12 :- Following screen will appear:-



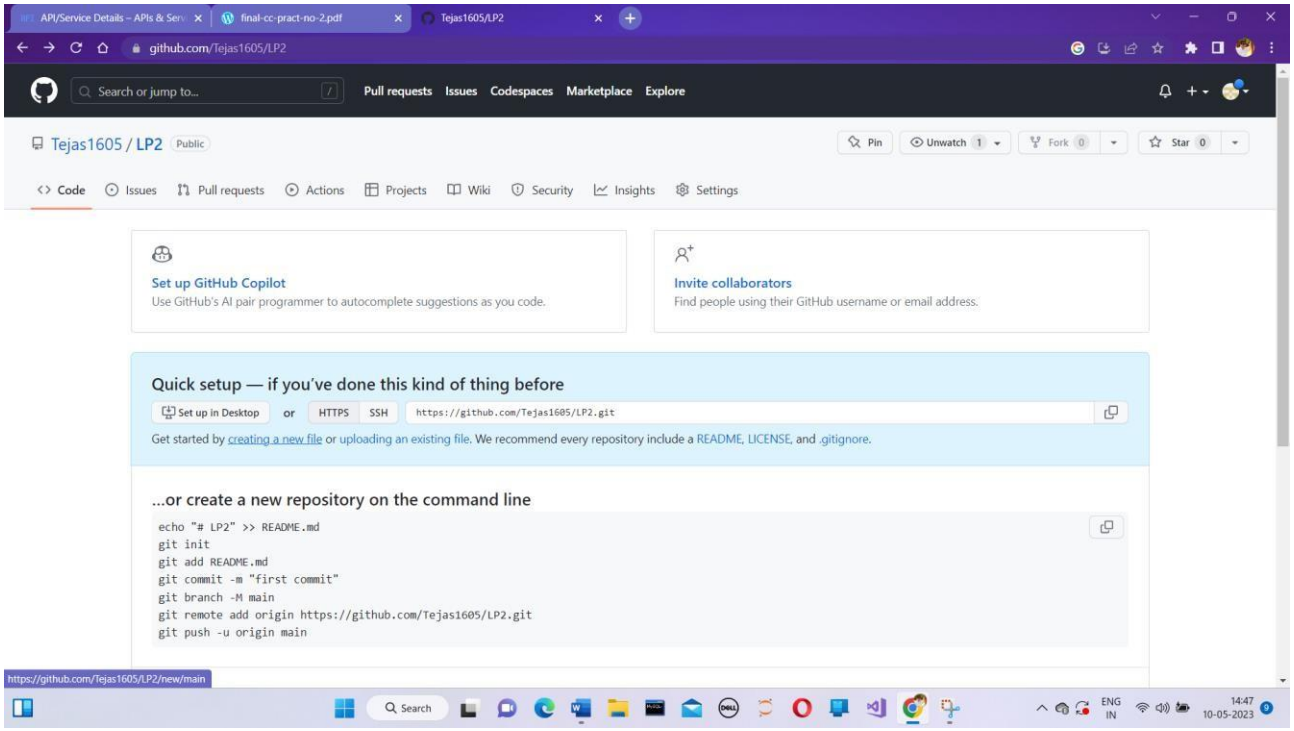
Step 13:- Login into your GitHub account and click on new repository.



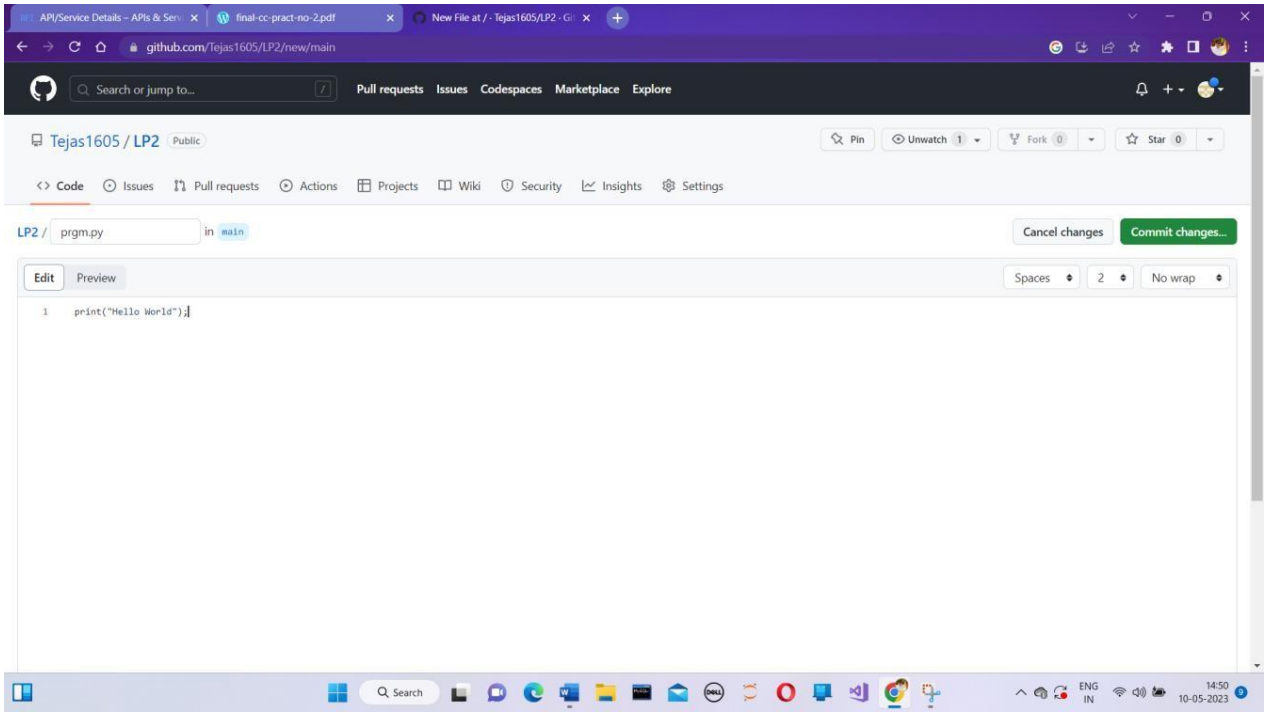
Step 14 :- Give name to your repository and click create.



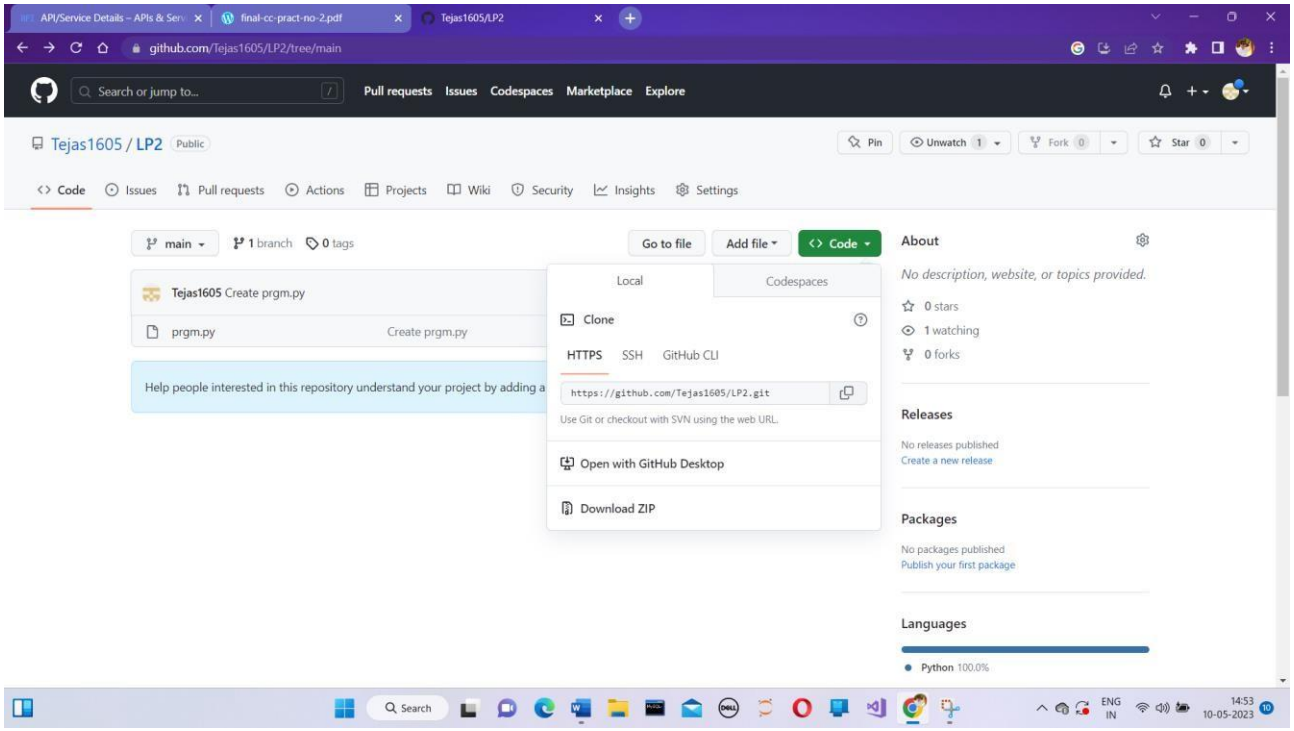
Step 15:- Click on creating new file



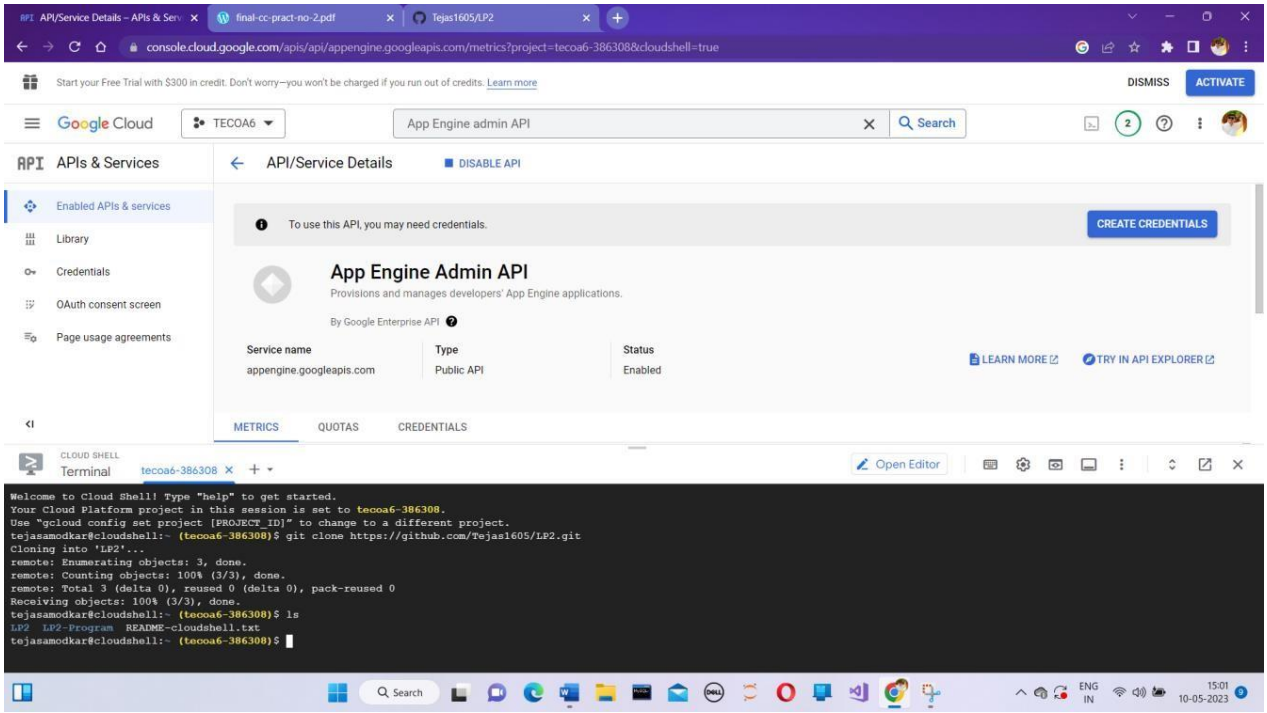
Step 16 :- Give name to the python file & type your code.



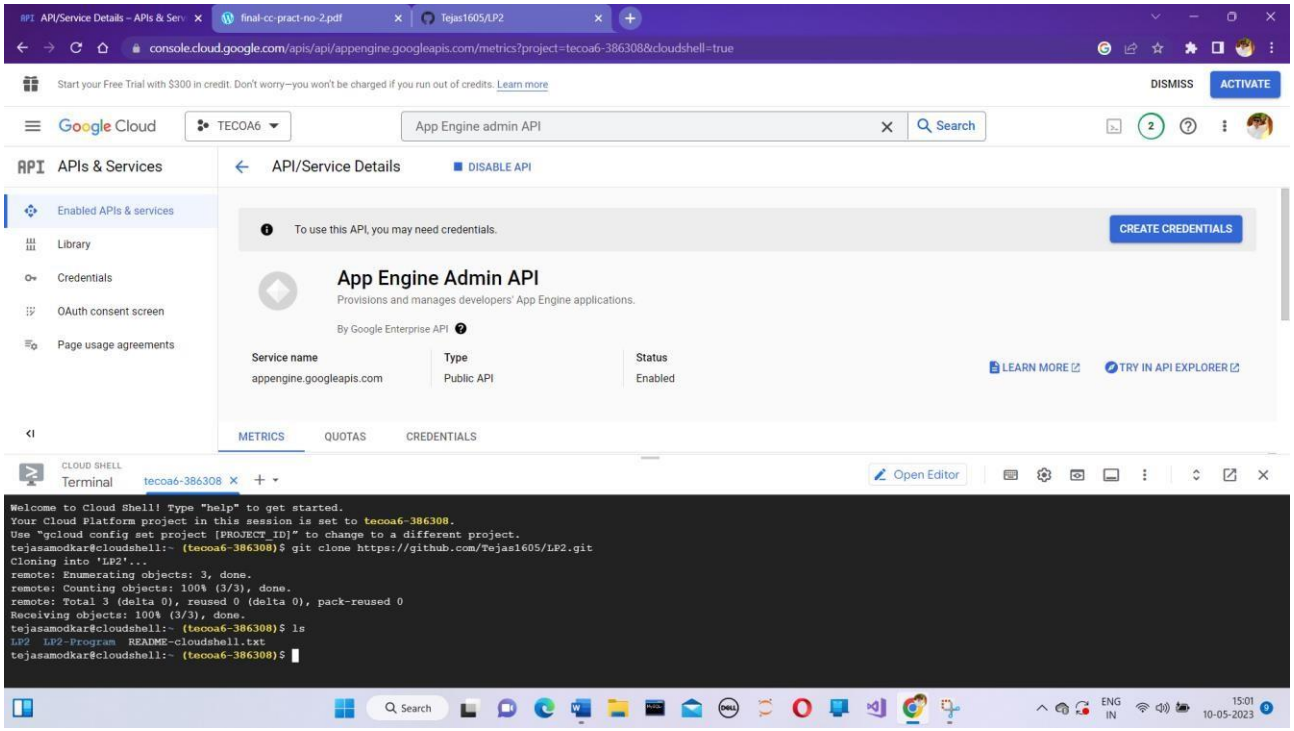
Step 17:- Click on code and copy URL.



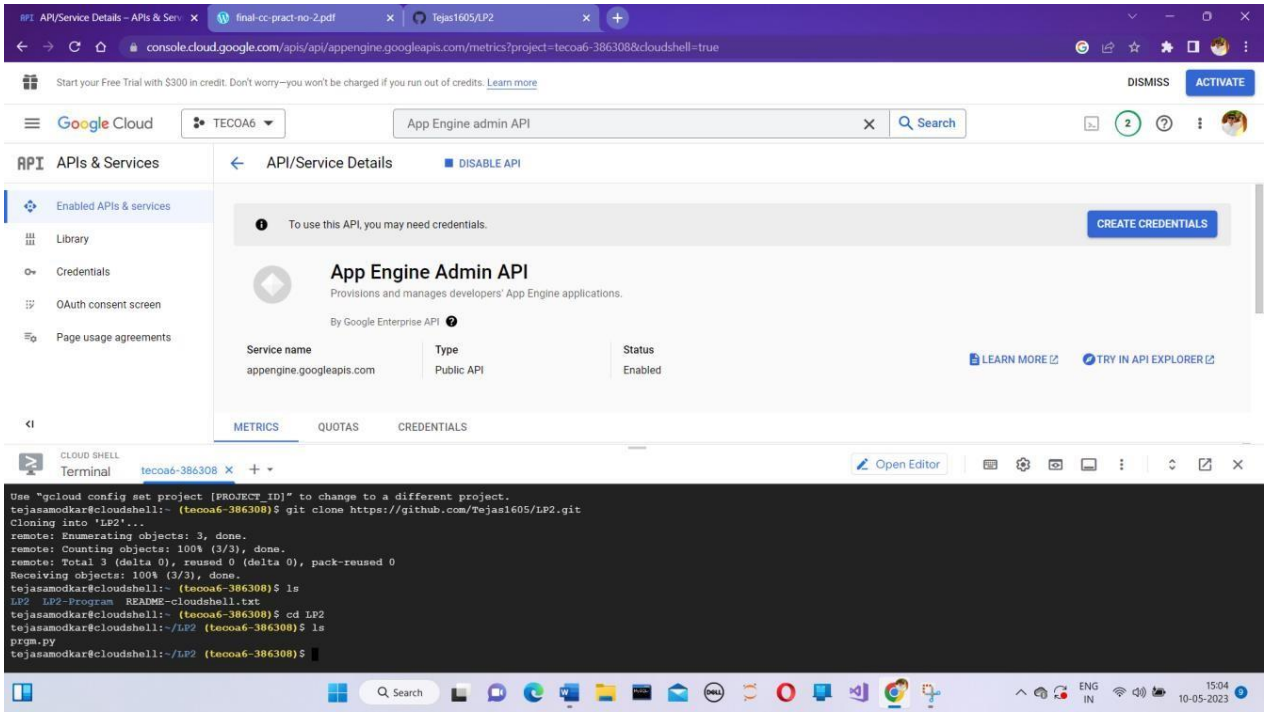
Step 18:- Go to cloud and type – git clone and paste url.



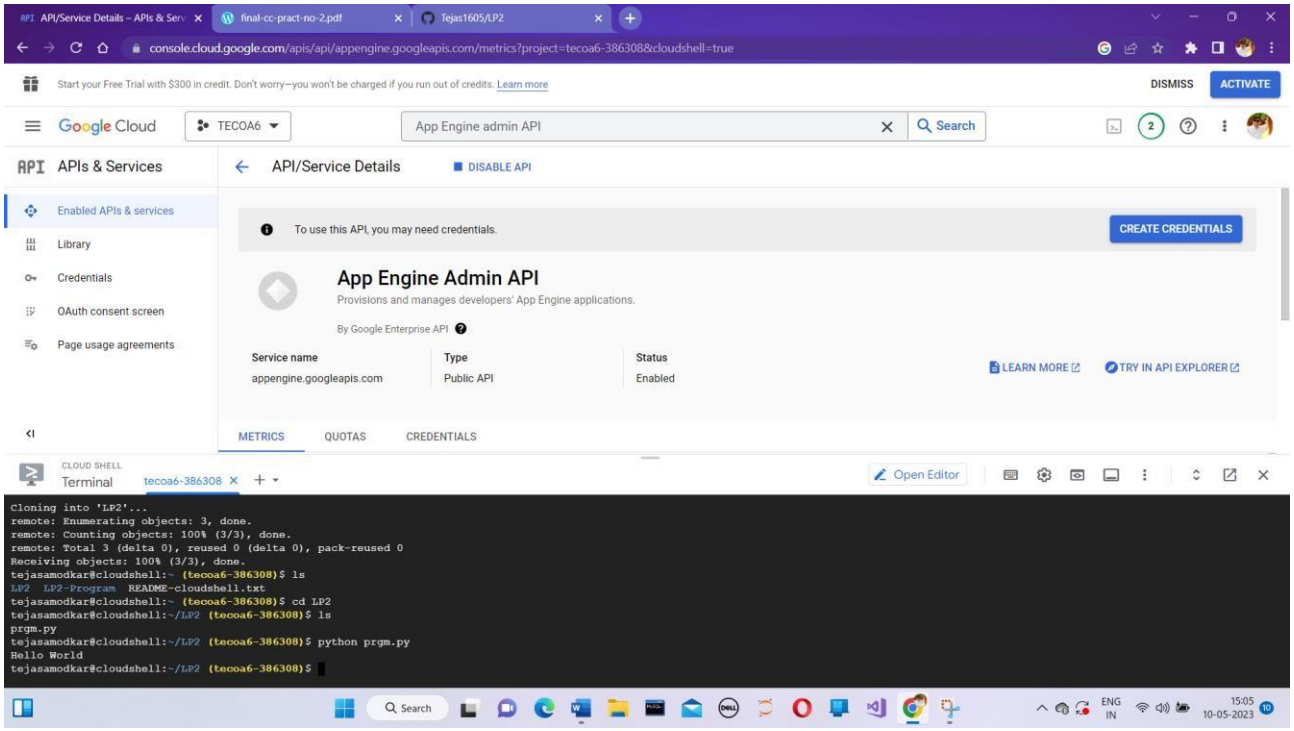
Step 19 :- Type ls



Step 20 :- Enter cd-repository name.



Step 21:- Type ls and to run python code type python-program name.



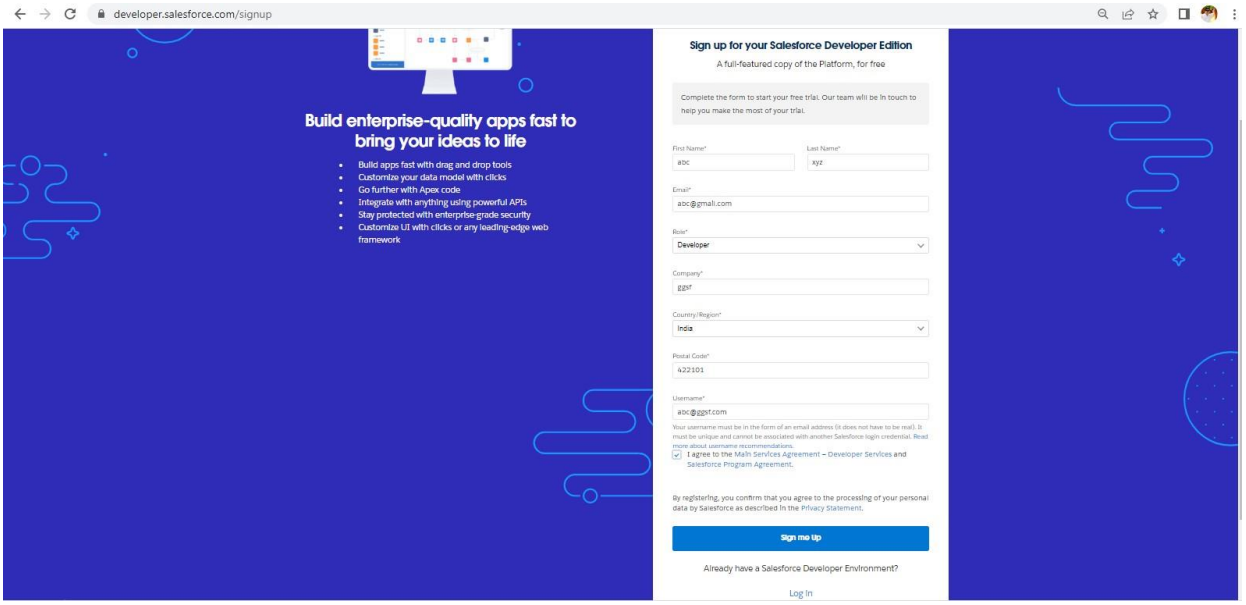
PRACTICAL NO : 9

Title : Creating an Application in SalesForce.com using Apex programming Language

STEPS TO CREATE AN APPLICATION :

Step No 1: Create new org:

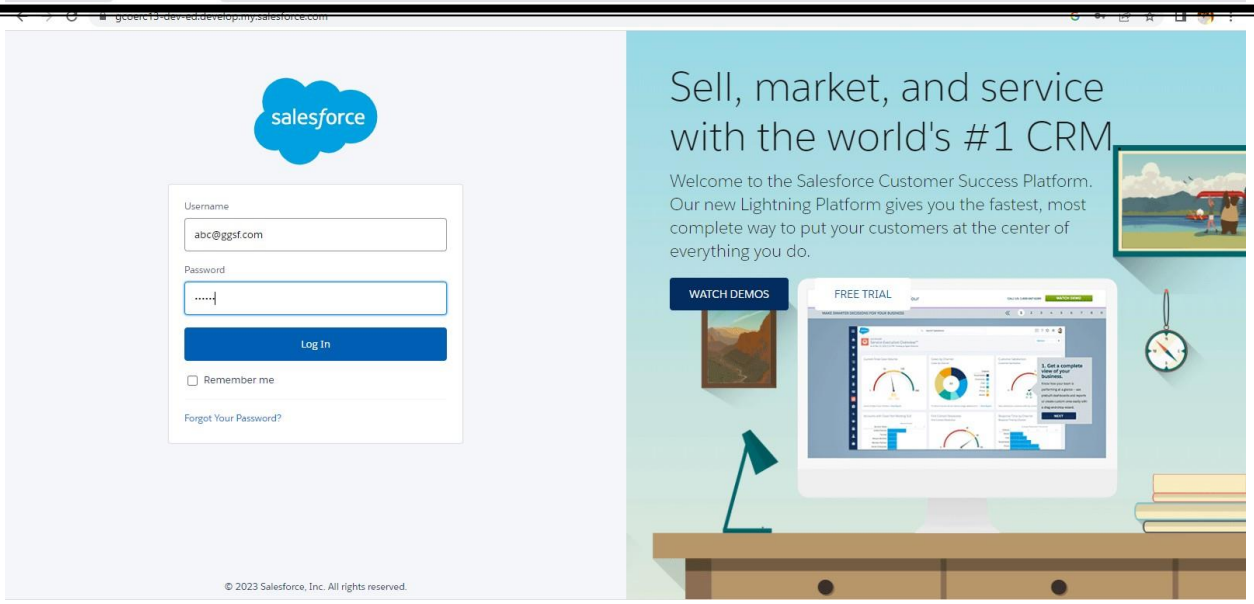
https://developer.salesforce.com/signup



Step No 2: After signup, loging using following URL

https://login.salesforce.com/

Step No 3: Login Page (Enter your credential to login)



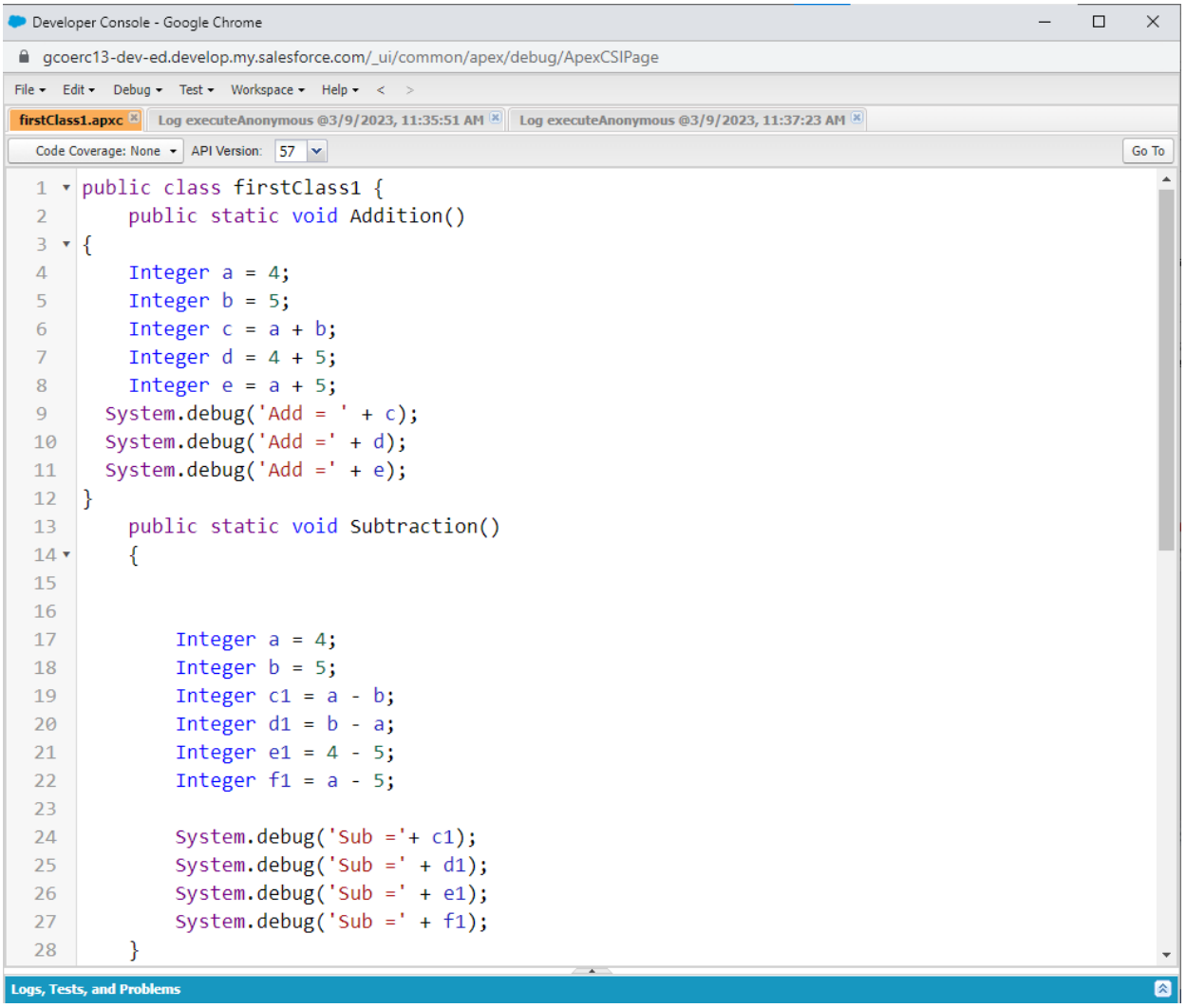
1. Open Developer Console

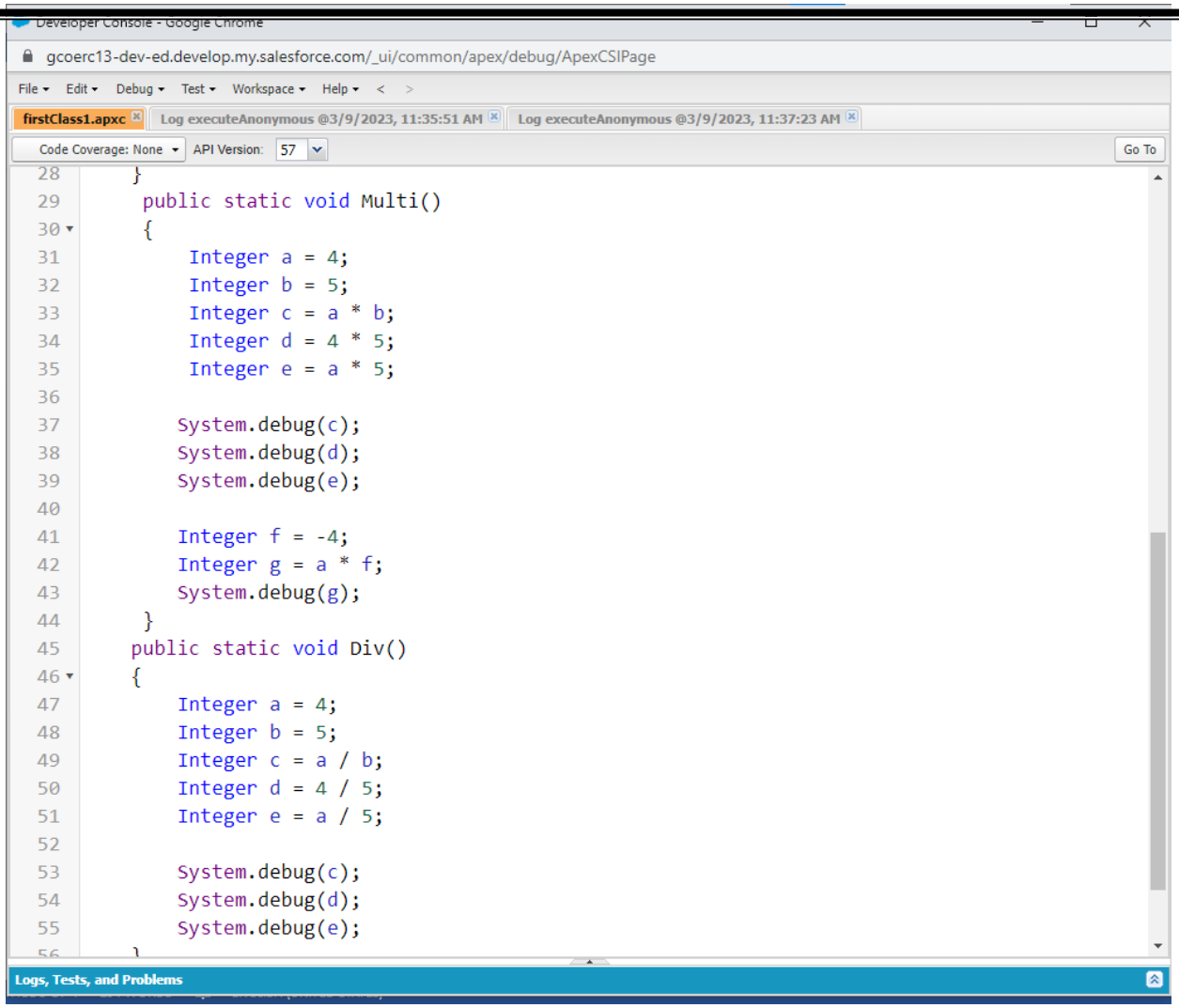
2. File ---> New ---> Select Apex Class

Type below mentioned code

```
public class firstClass1 {
   public static void Addition()
{
   Integer a = 4;
   Integer b = 5;
   Integer c = a + b;
   Integer d = 4 + 5;
   Integer e = a + 5;
  System.debug('Add = ' + c);
  System.debug('Add =' + d);
  System.debug('Add =' + e);
}
   public static void Subtraction()
   {


      Integer a = 4;
      Integer b = 5;
      Integer c1 = a - b;
                  Integer d1 = b - a;
                  Integer e1 = 4 - 5;
                  Integer f1 = a - 5;

                  System.debug('Sub ='+ c1);
                  System.debug('Sub =' + d1);
                  System.debug('Sub =' + e1);
                  System.debug('Sub =' + f1);
   }
    public static void Multi()
    {
       Integer a = 4;
                  Integer b = 5;
```

```
                Integer c = a * b;
                Integer d = 4 * 5;
                Integer e = a * 5;

                System.debug(c);
                System.debug(d);
                System.debug(e);

                Integer f = -4;
                Integer g = a * f;
                System.debug(g);
    }
    public static void Div()
    {
        Integer a = 4;
                Integer b = 5;
                Integer c = a / b;
                Integer d = 4 / 5;
                Integer e = a / 5;

                System.debug(c);
                System.debug(d);
                System.debug(e);
    }
}
```





3. Click on Debug ---> Open Execute Acronymous Window

4. Type below code(Apex Code)

```
firstClass1.Addition();
firstClass1.Subtaction();
firstClass1.Multi();
firstClass1.Div();
```

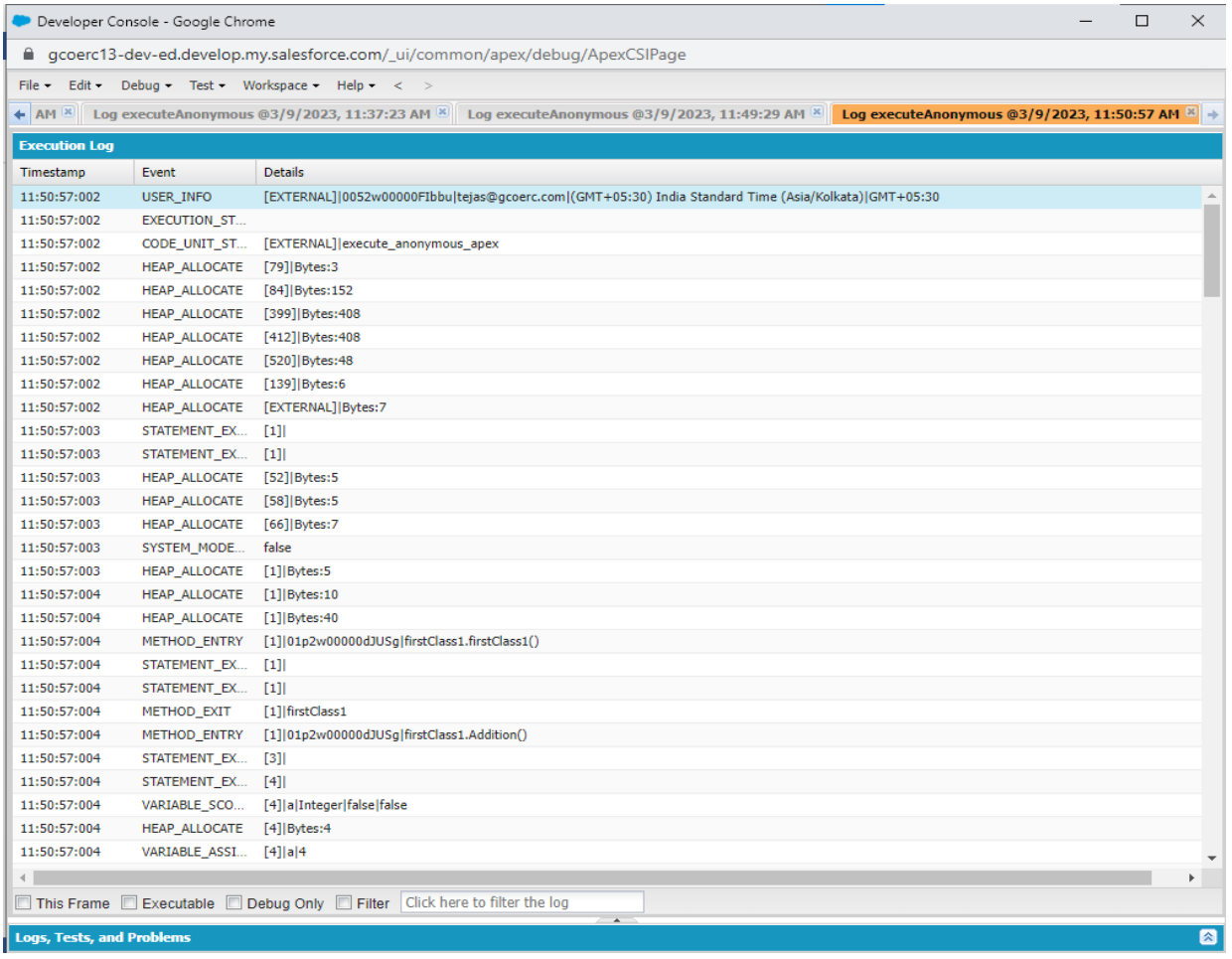
5. Click on Open log then Execute code

Enter Apex Code

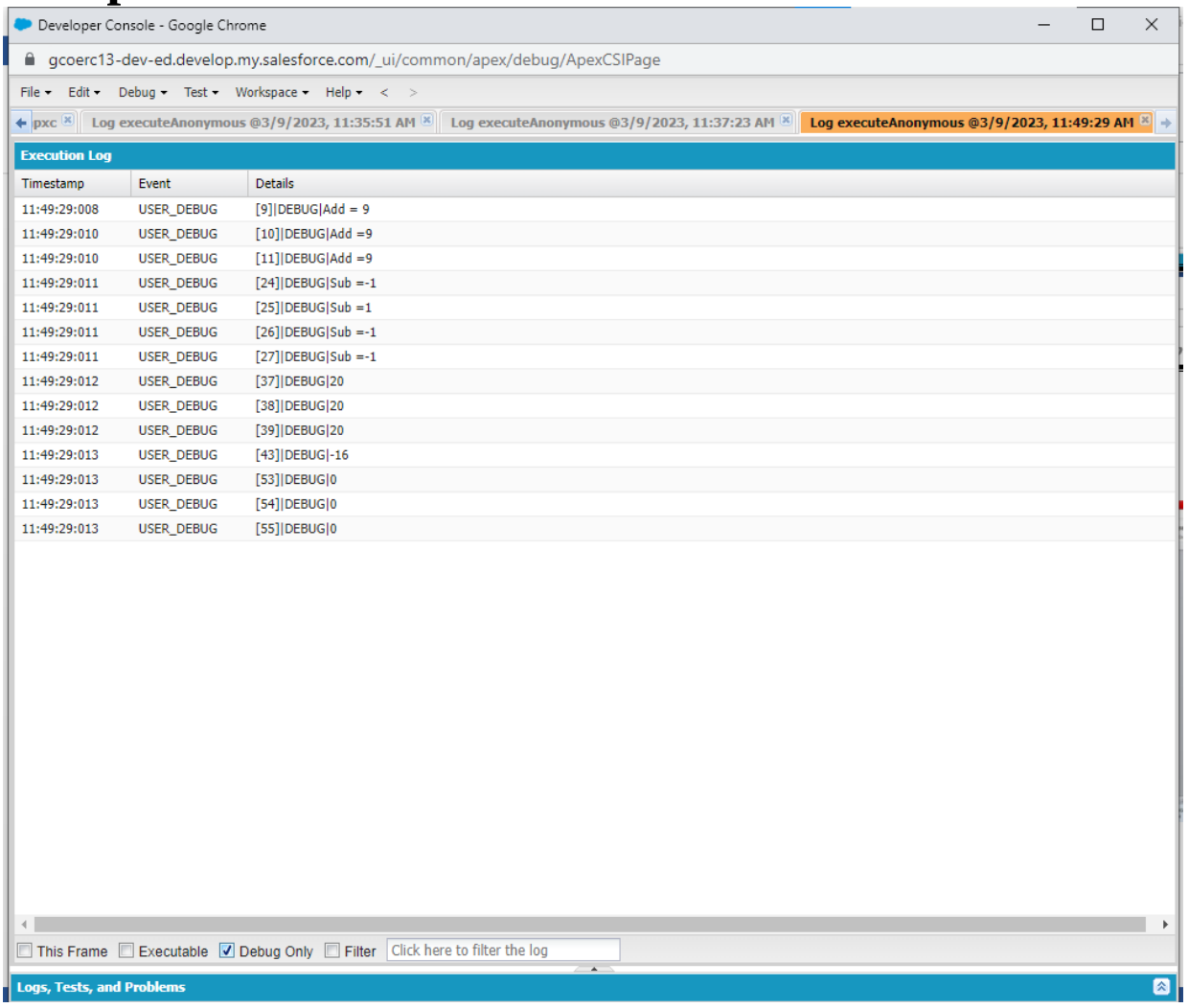
```
1  firstClass1.Addition();
2  firstClass1.Subtraction();
3  firstClass1.Multi();
4  FirstClass1.Div();
```

Open Log | Execute | Execute Highlighted

6. Click on Debug only (You will get output)



Output :

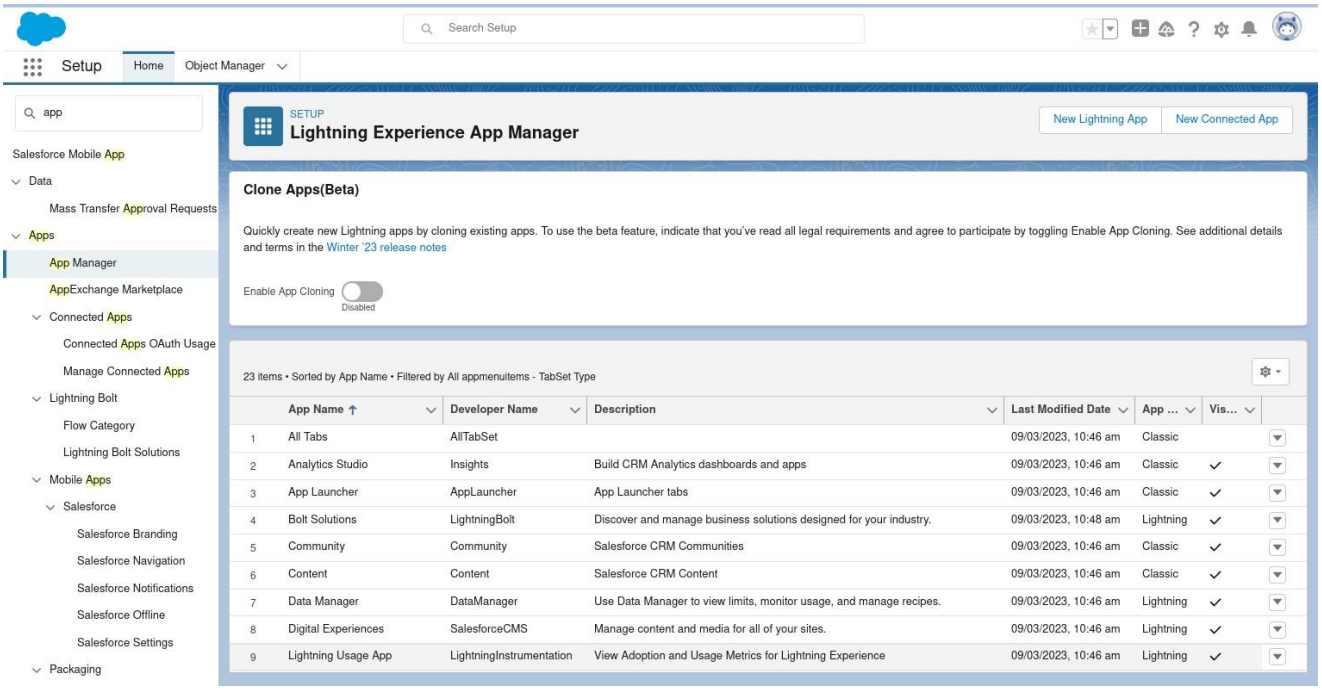

Developer Console - Google Chrome
gcoerc13-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage
File Edit Debug Test Workspace Help
pxc
Log executeAnonymous @3/9/2023, 11:35:51 AM
Log executeAnonymous @3/9/2023, 11:37:23 AM
Log executeAnonymous @3/9/2023, 11:49:29 AM
Execution Log
Timestamp Event Details
11:49:29:008 USER_DEBUG [9]|DEBUG|Add = 9
11:49:29:010 USER_DEBUG [10]|DEBUG|Add =9
11:49:29:010 USER_DEBUG [11]|DEBUG|Add =9
11:49:29:011 USER_DEBUG [24]|DEBUG|Sub =-1
11:49:29:011 USER_DEBUG [25]|DEBUG|Sub =1
11:49:29:011 USER_DEBUG [26]|DEBUG|Sub =-1
11:49:29:011 USER_DEBUG [27]|DEBUG|Sub =-1
11:49:29:012 USER_DEBUG [37]|DEBUG|20
11:49:29:012 USER_DEBUG [38]|DEBUG|20
11:49:29:012 USER_DEBUG [39]|DEBUG|20
11:49:29:013 USER_DEBUG [43]|DEBUG|-16
11:49:29:013 USER_DEBUG [53]|DEBUG|0
11:49:29:013 USER_DEBUG [54]|DEBUG|0
11:49:29:013 USER_DEBUG [55]|DEBUG|0
This Frame Executable Debug Only Filter Click here to filter the log
Logs, Tests, and Problems

PRACTICAL NO : 10

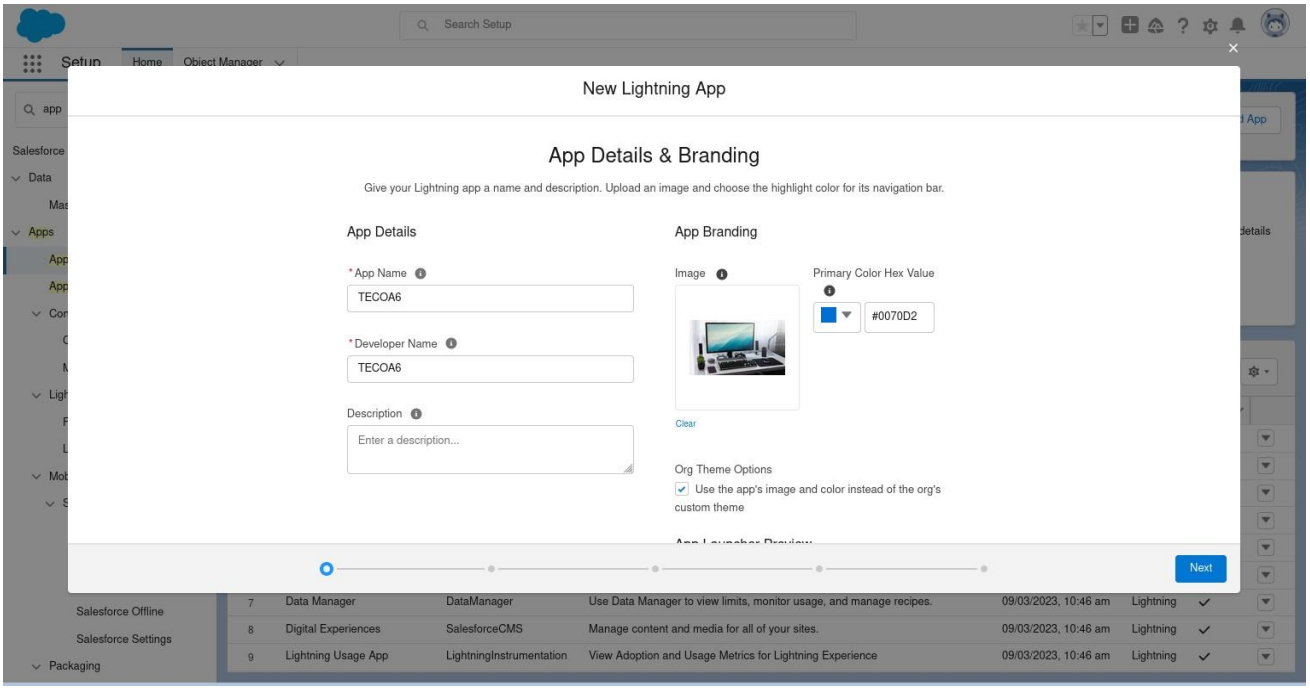
Title : Design and develop custom Application (Mini Project) using Salesforce Cloud.

STEPS TO CREATE AN APPLICATION(mini project) :

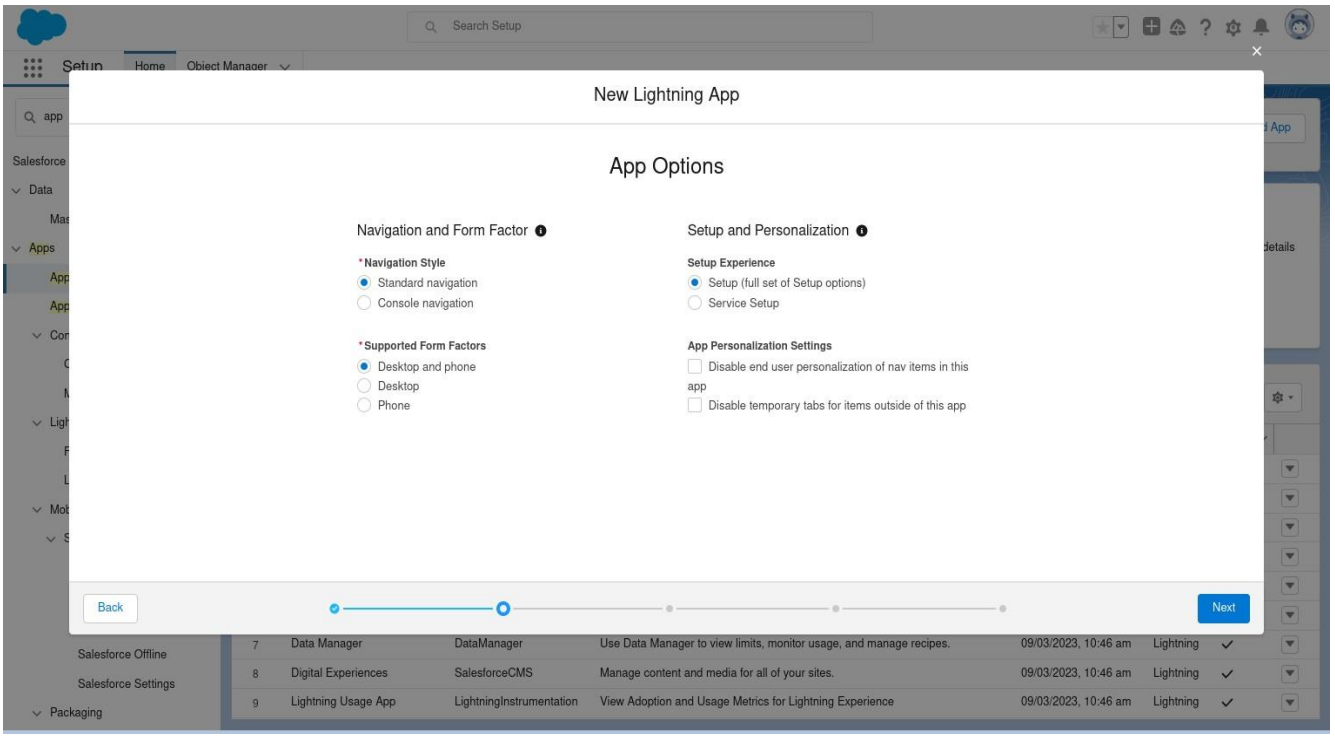
1. Open salesforce in Lightning Experience.
2. Open App Launcher ▪ View all ▪ Quick Search ▪ App Manager ▪ Click on New Lightning App.



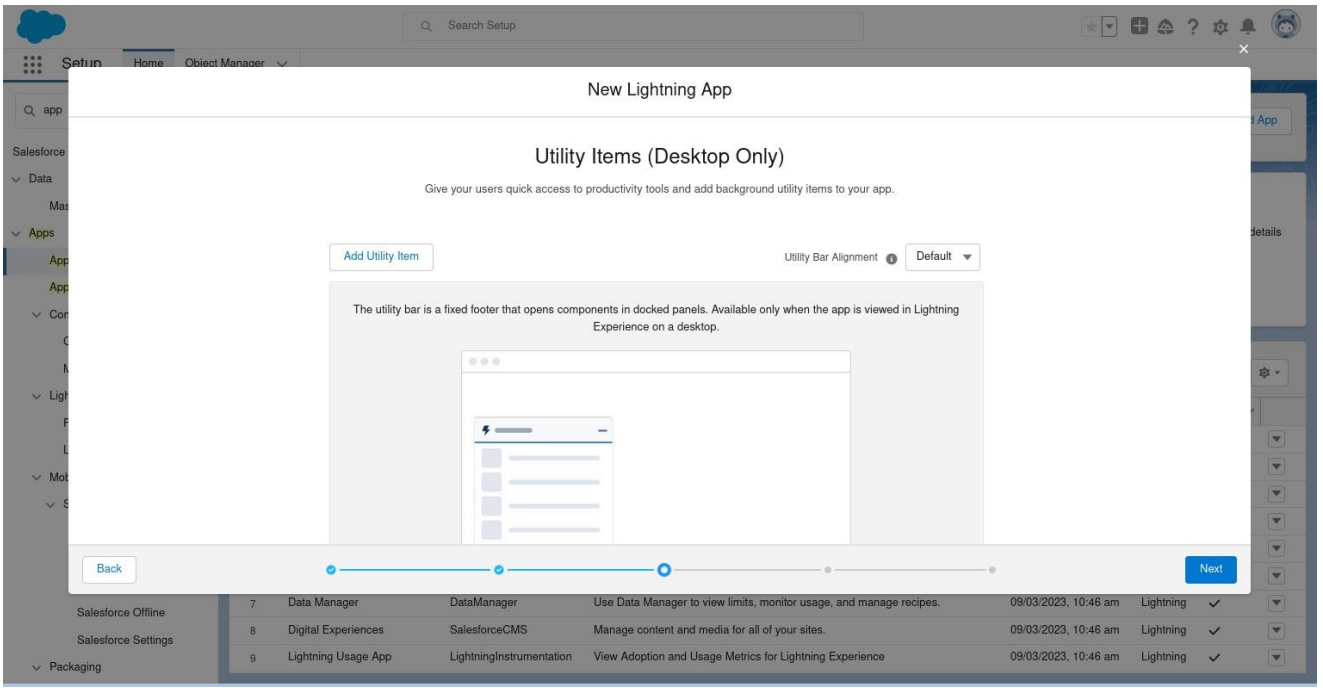
3. fill the mention information.



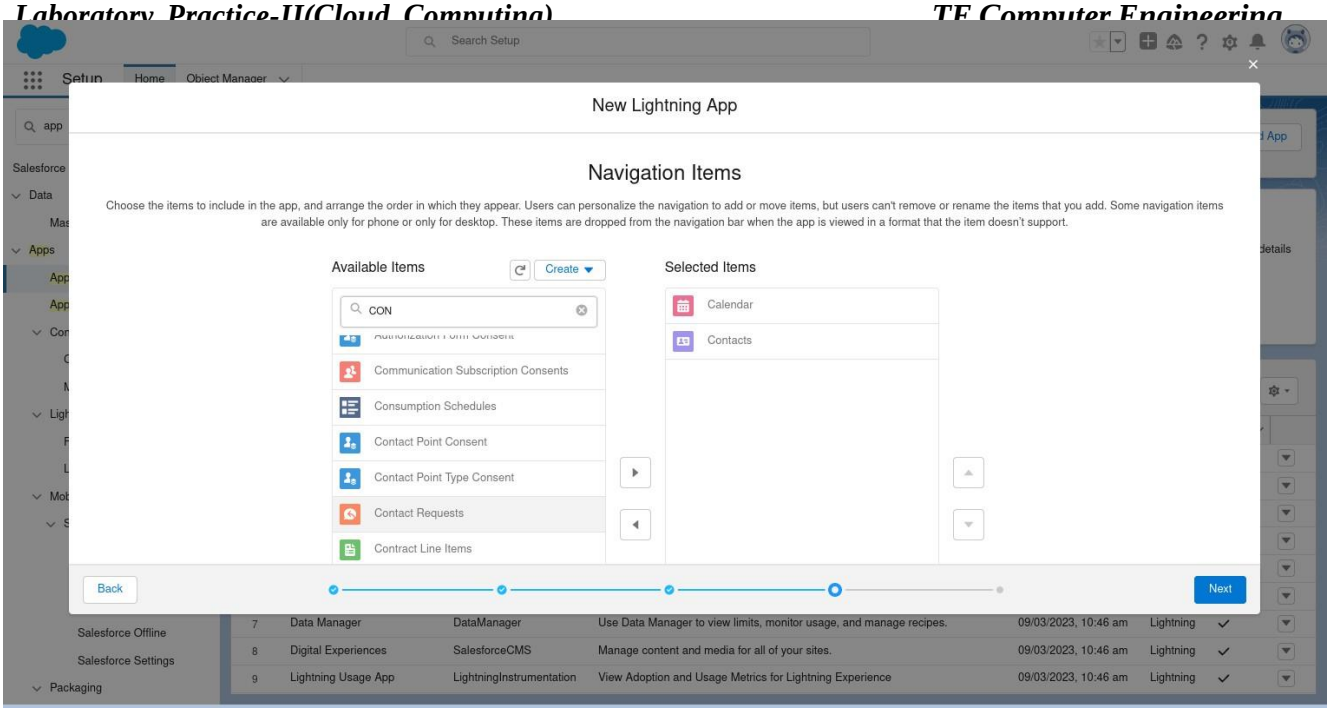
4. Next ▪ App option(optional).



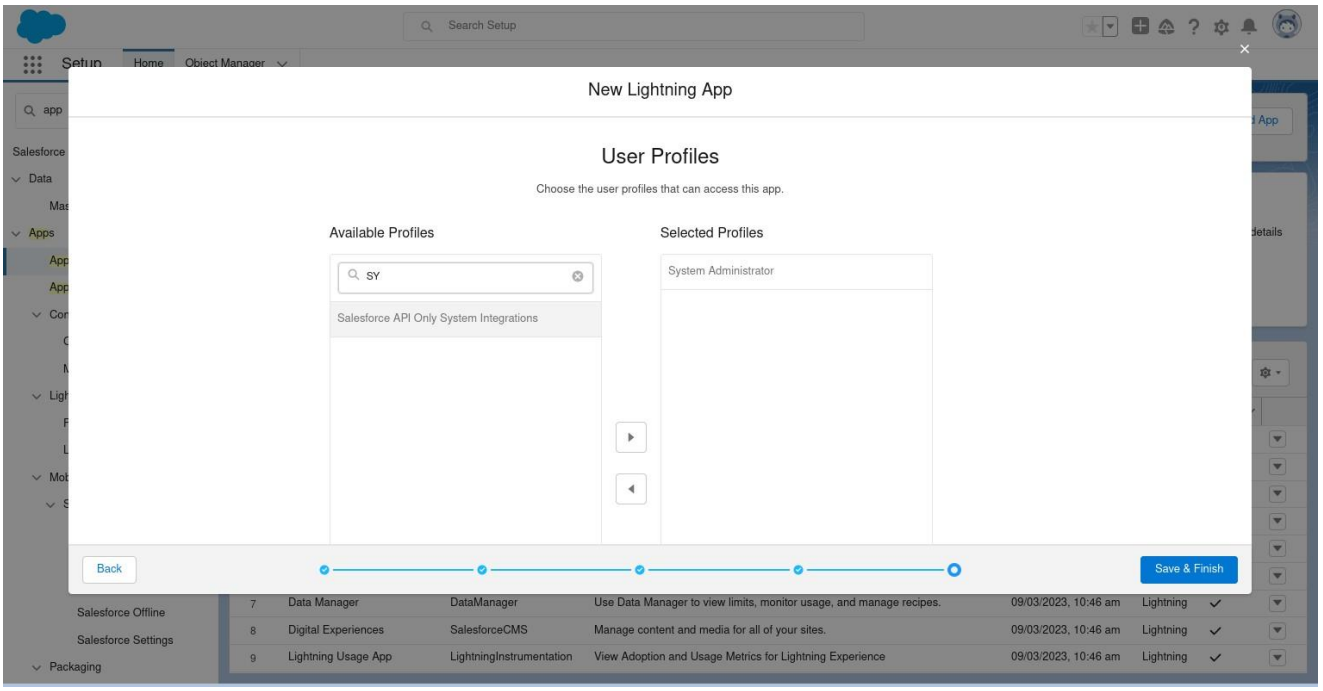
5. Next ▪ Add Utility (optoinal).



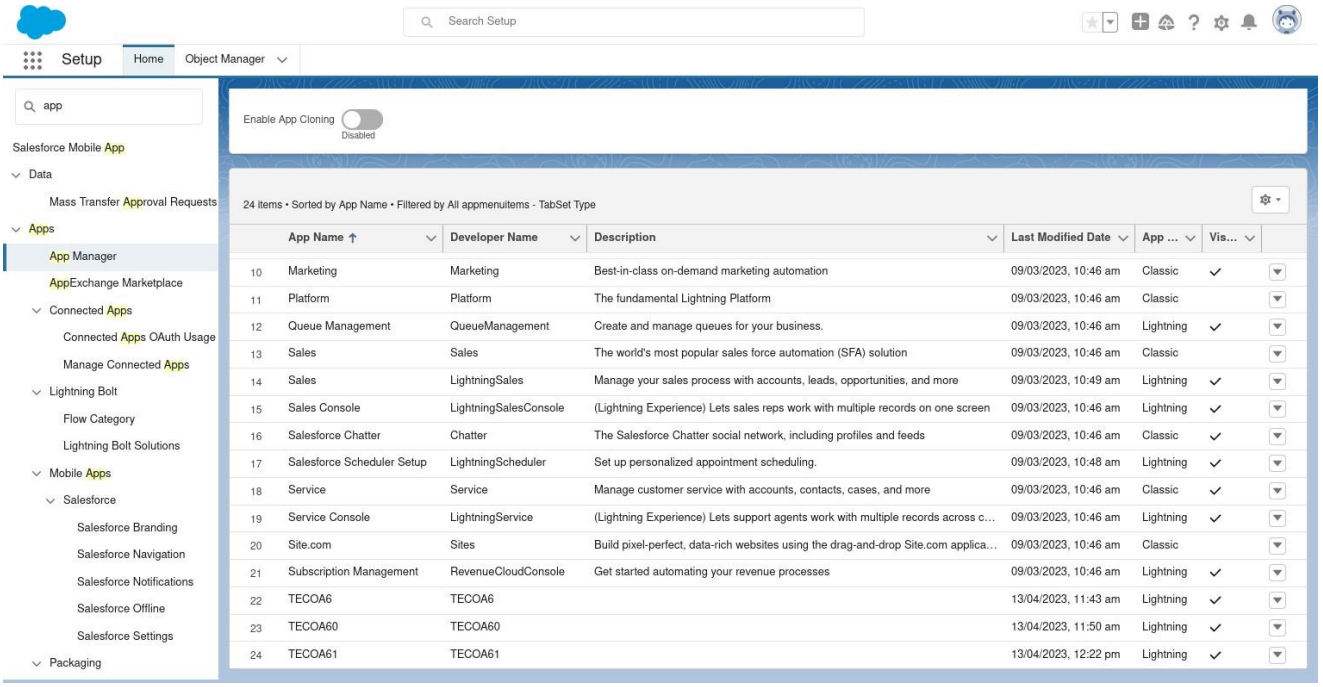
6. Next ▪ Navigation Item(you can add item).



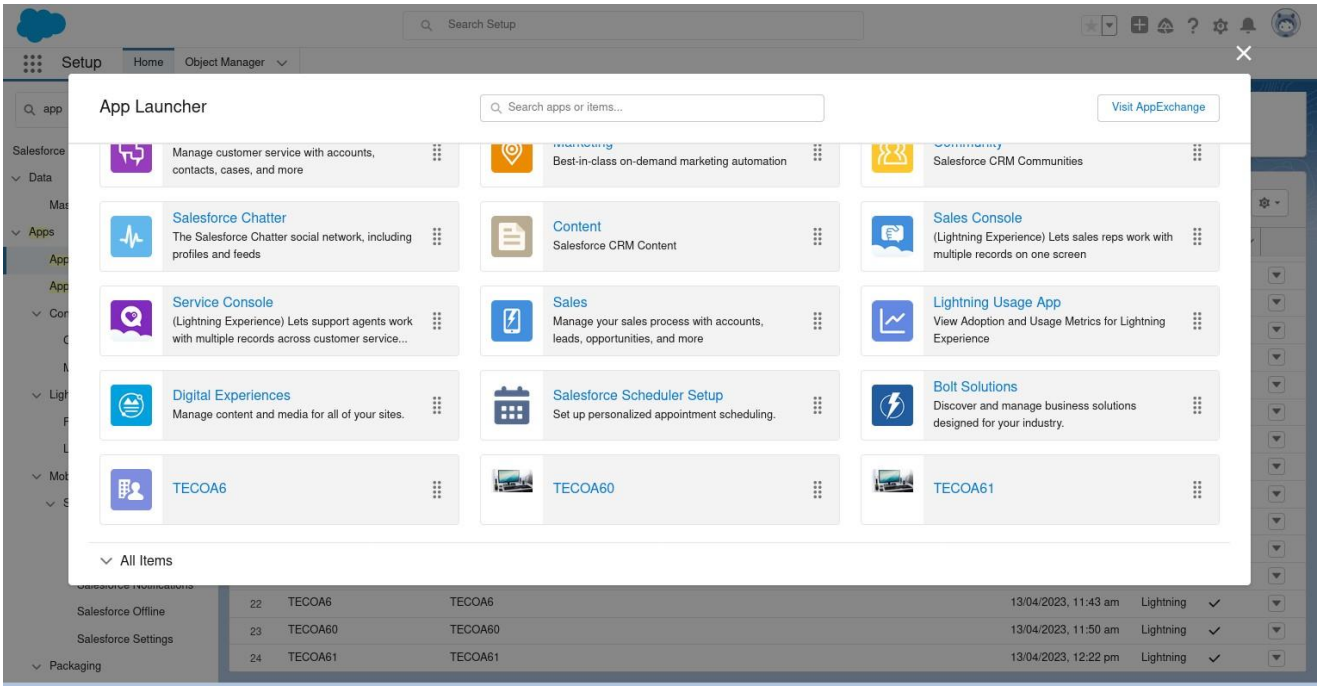
7. Next ▪ User Profile(use the user profilethat can access this app) ▪ Save and Finish.



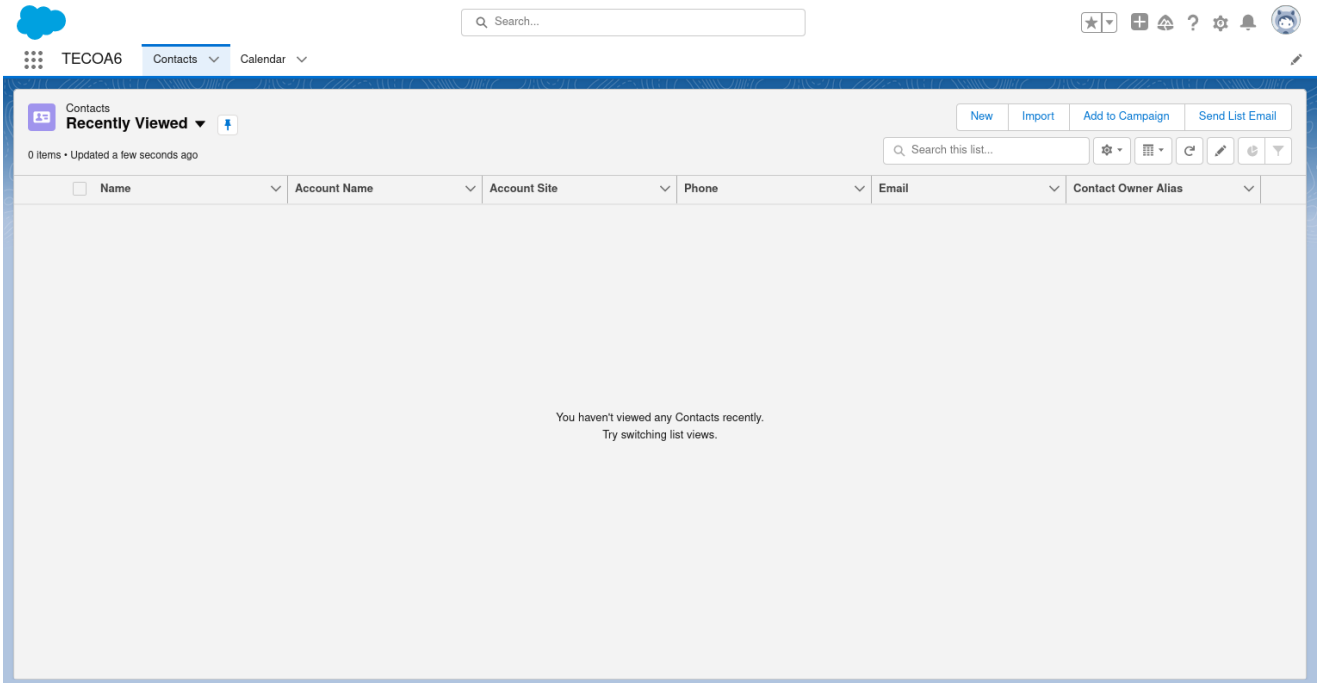
8. You will get Lightning Experience App Manager Window.

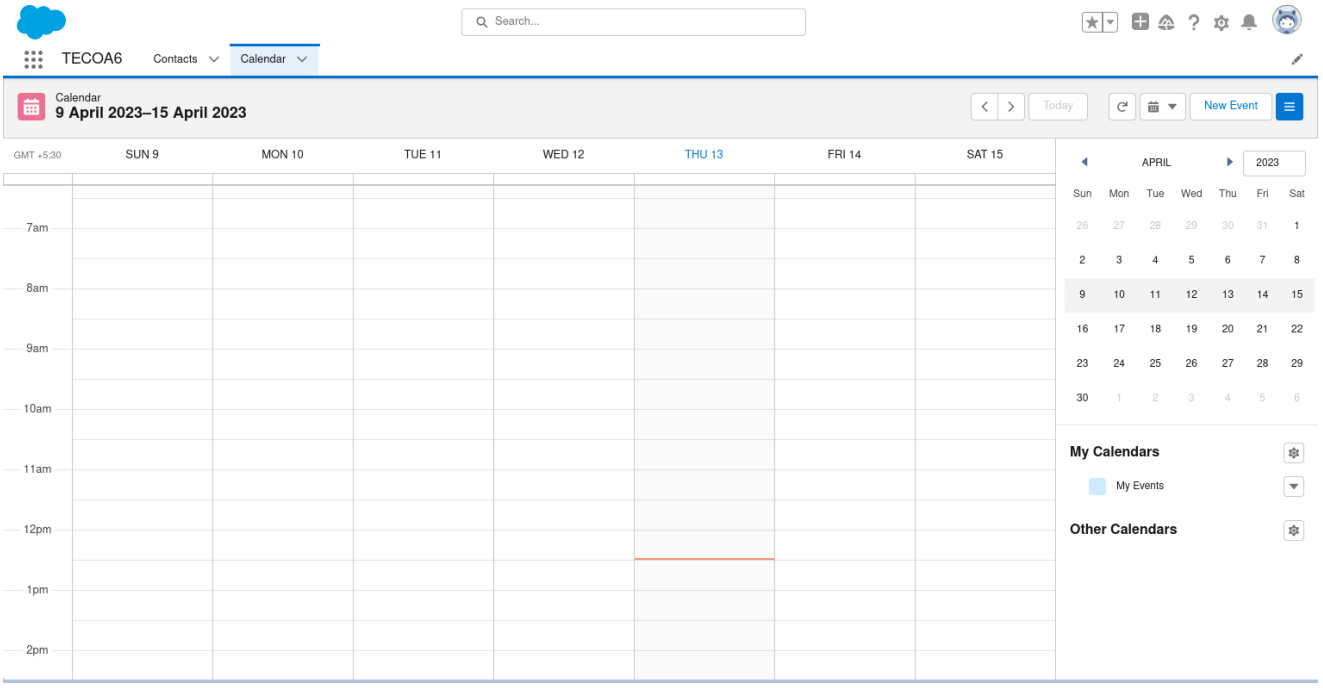


9. Search Your App by its name in App Launcher. just by entering some character, you can see app name with its logo.



10. Click on App ▪ you can see your application appearance in window ▪ Output.


Search...
TECOA6
Contacts
Calendar
Contacts
Recently Viewed
0 items • Updated a few seconds ago
New
Import
Add to Campaign
Send List Email
Search this list...
Name
Account Name
Account Site
Phone
Email
Contact Owner Alias
You haven't viewed any Contacts recently.
Try switching list views.


Search...
TECOA6
Contacts
Calendar
Calendar
9 April 2023–15 April 2023
Today
New Event
GMT +5:30
SUN 9
MON 10
TUE 11
WED 12
THU 13
FRI 14
SAT 15
7am
8am
9am
10am
11am
12pm
1pm
2pm
APRIL
2023
Sun Mon Tue Wed Thu Fri Sat
My Calendars
My Events
Other Calendars

Practical No : 11

Practical Title: Setup your own cloud for Software as a Service (SaaS) over the existing LAN in your laboratory. In this assignment you have to write your own code for cloud controller using open-source technologies to implement with HDFS. Implement the basic operations may be like to divide the file in segments/blocks and upload/ download fileon/from cloud in encrypted form.

Objectives:

- To set your own cloud for SaaS over existing LAN
- To implement the basic operations may be like to divide the file in segments/blocks

Hardware Requirements :

- Pentium IV with latest configuration

Software Requirements :

- Ubuntu 20.04, VMwareESXi cloud

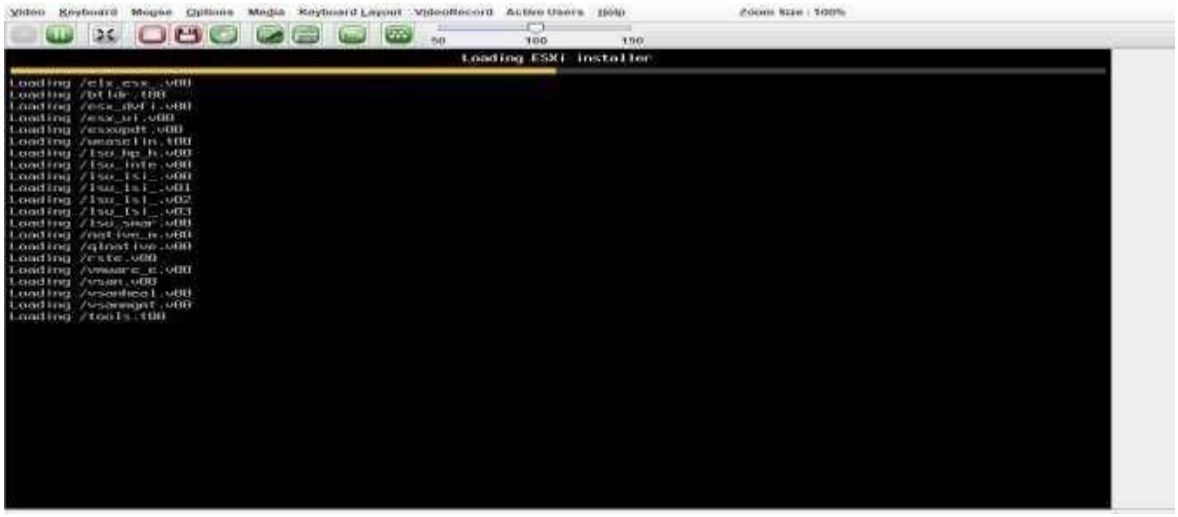
Theory:

Here we are installing VMwareESXi cloud

- Host/NodeESXi installation:-
- **ESXiHardwareRequirements:-**

- ESXi6.7requiresahostmachinewithatleasttwoCPUcores.
- ESXi6.7supports64-bitx86processors
- ESXi6.7requirestheNX/XDbit to be enabled for the CPU in the BIOS.
- ESXi6.7requiresaminimumof4GBofphysicalRAM.Itisrecommended to provide atleast 8 GB of RAM to run virtual machines in typical productionenvironments.
- Tosupport64-bitvirtualmachines,support for hardware virtualization (IntelVT-xor AMDRVI) mustbeenabledonx64CPUs.
- One or more Gigabit or faster Ethernet controllers. For a list of supportednetwork adapter models.
- SCSI disk oralocal,non-network,RAIDLUN with unpartitioned space for the virtualmachines.

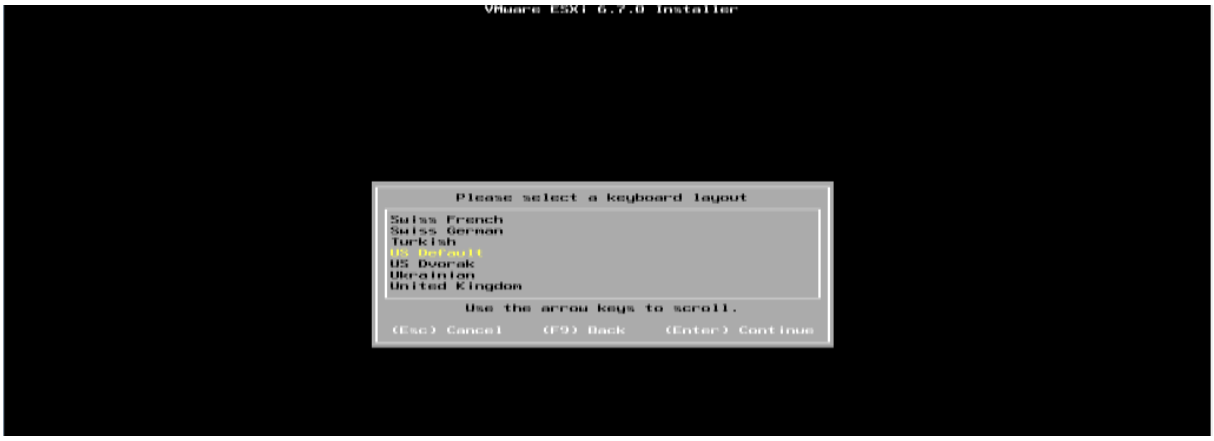
ForSerialATA(SATA), a disk connected through supported SAS controller or supported on board SATA controllers. SATA disks are considered remote not local. These disks are not used as a scratch partition by default be cause they are seen as remote.



ESXiInstaller:

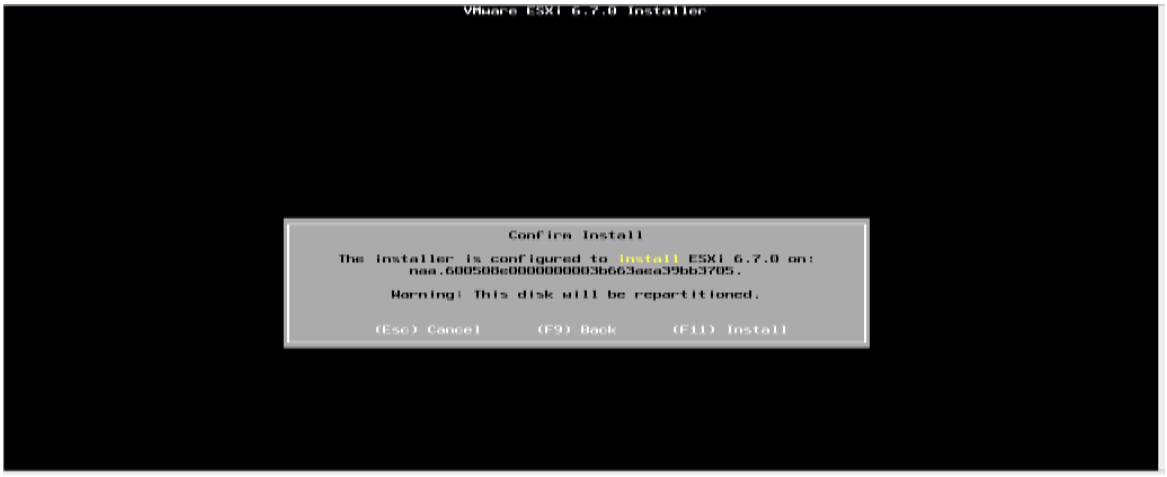
Accept Agreement:

Select storage :



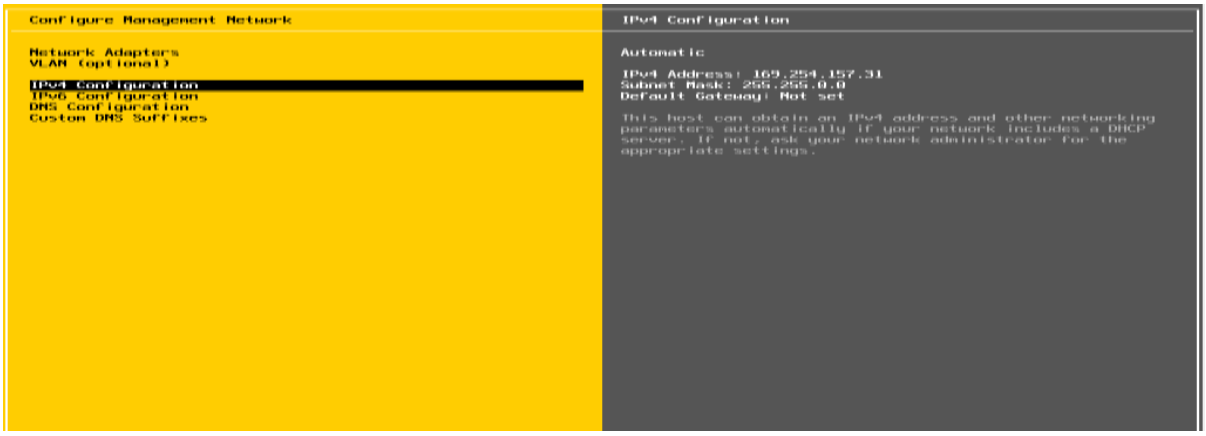
Select Keyboard Layout :

Set NodeESXi Root Password :



Installation complete (Reboot)CLII interface to configuration

CLI Interface to Configuration:

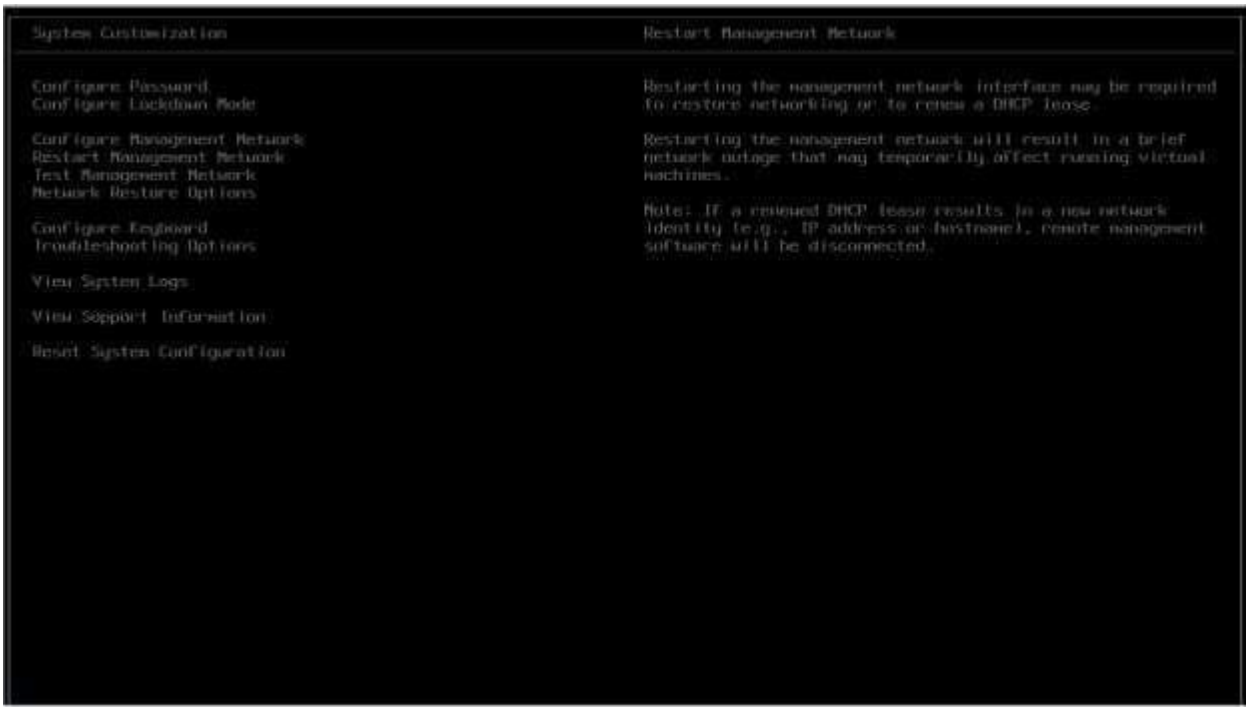


Configure Management Network

Set IPV4

Set DNSeriver :

Restart Management Network

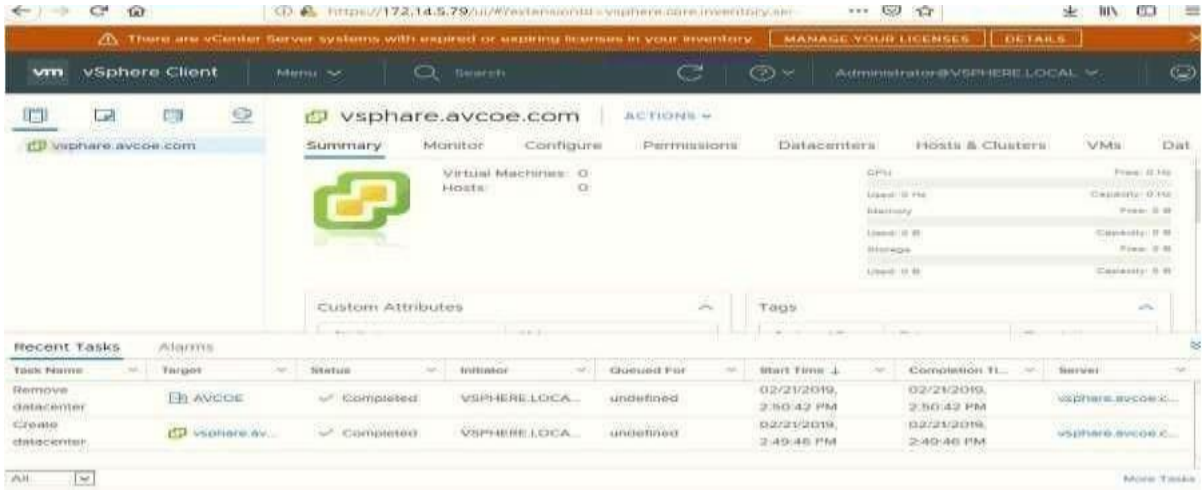


GUIAccess :

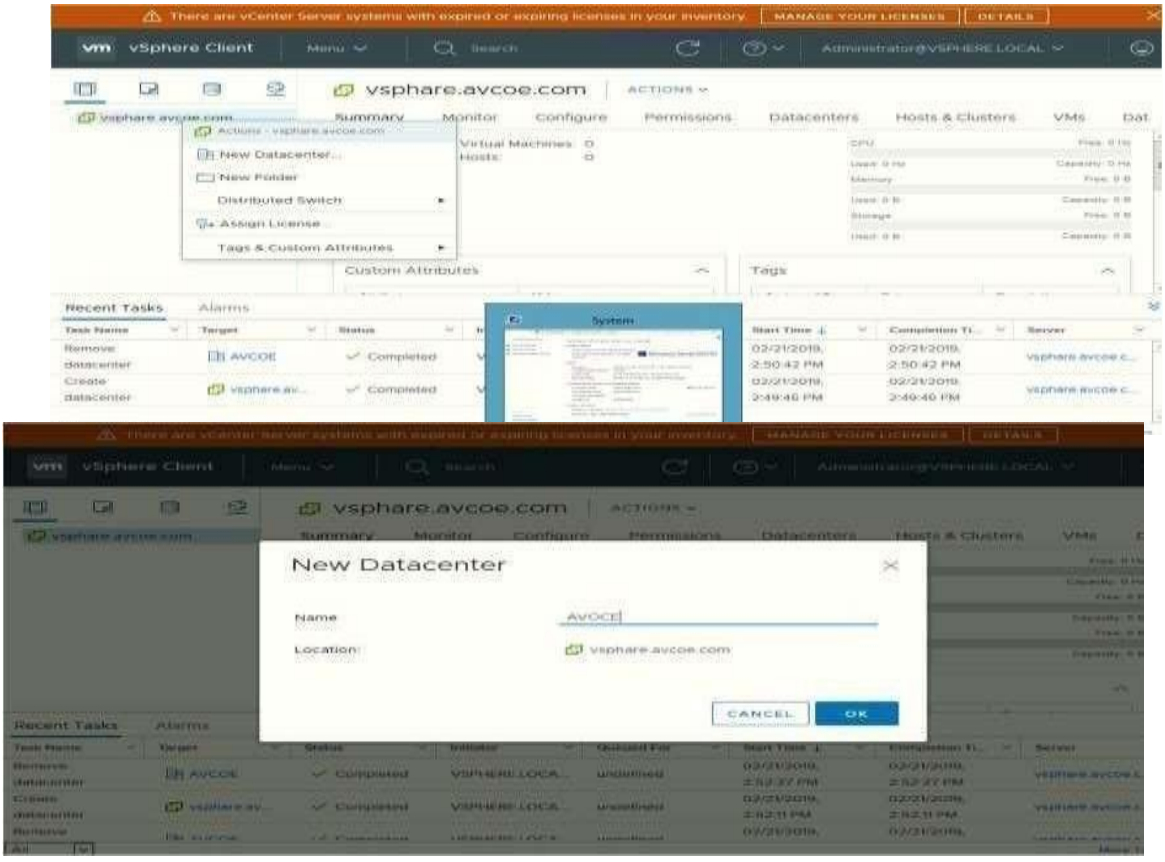
ClusterSetup

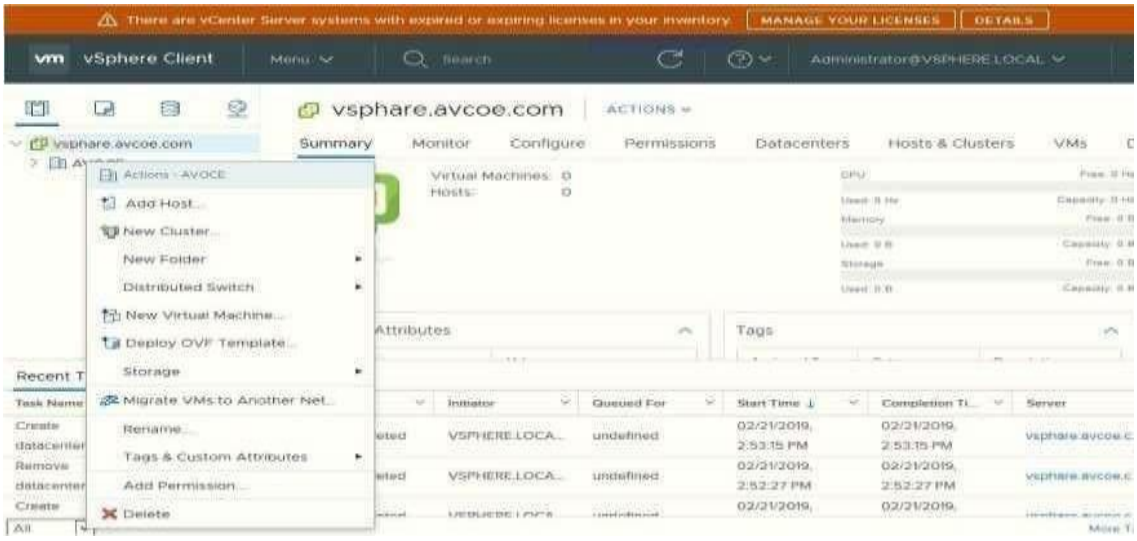
- **CreatingDatacenter**
- **CreatingCluster**
- **Adding Hosts incluster**
- **Resourcesafteraddingcluster.**
- **DRS**
- **Failover**

VCenter Access:



Create Datacenter:

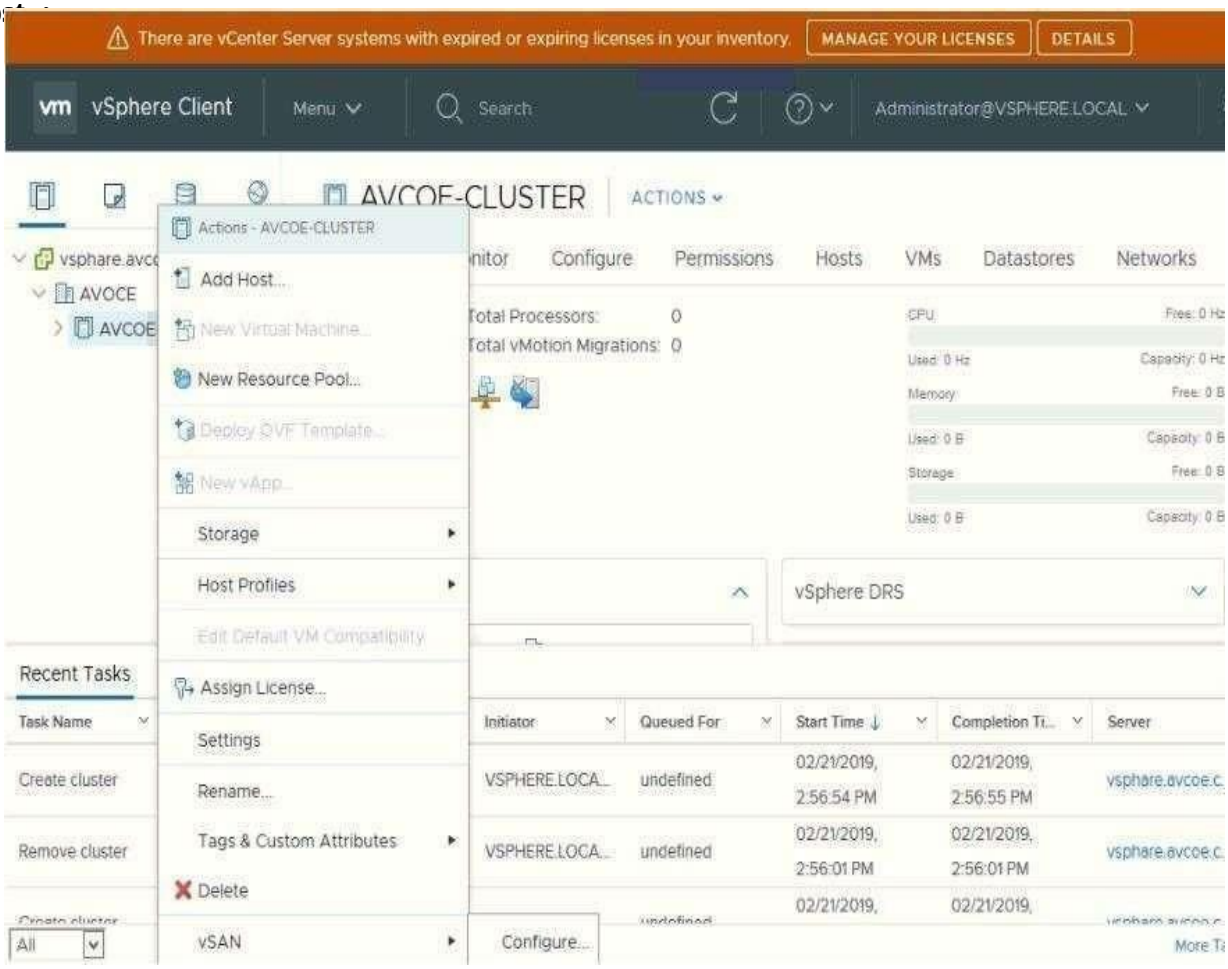




Create cluster :

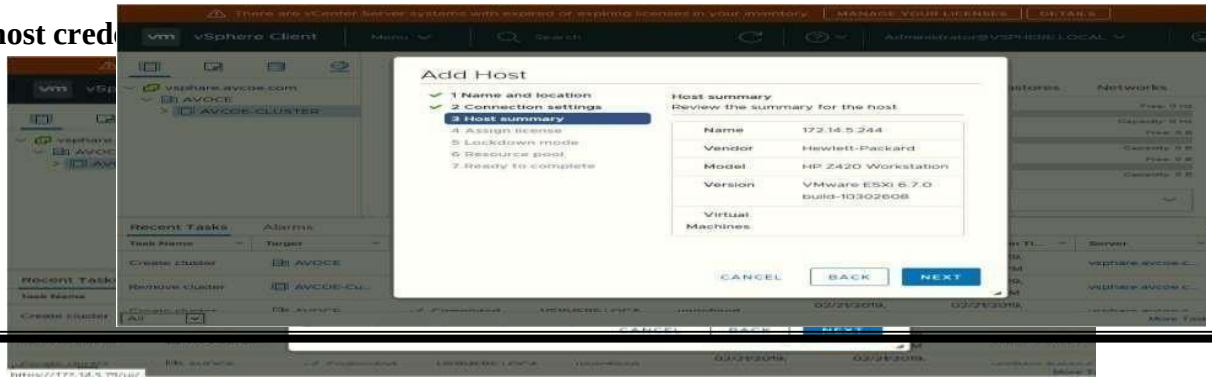
Assign cluster name :

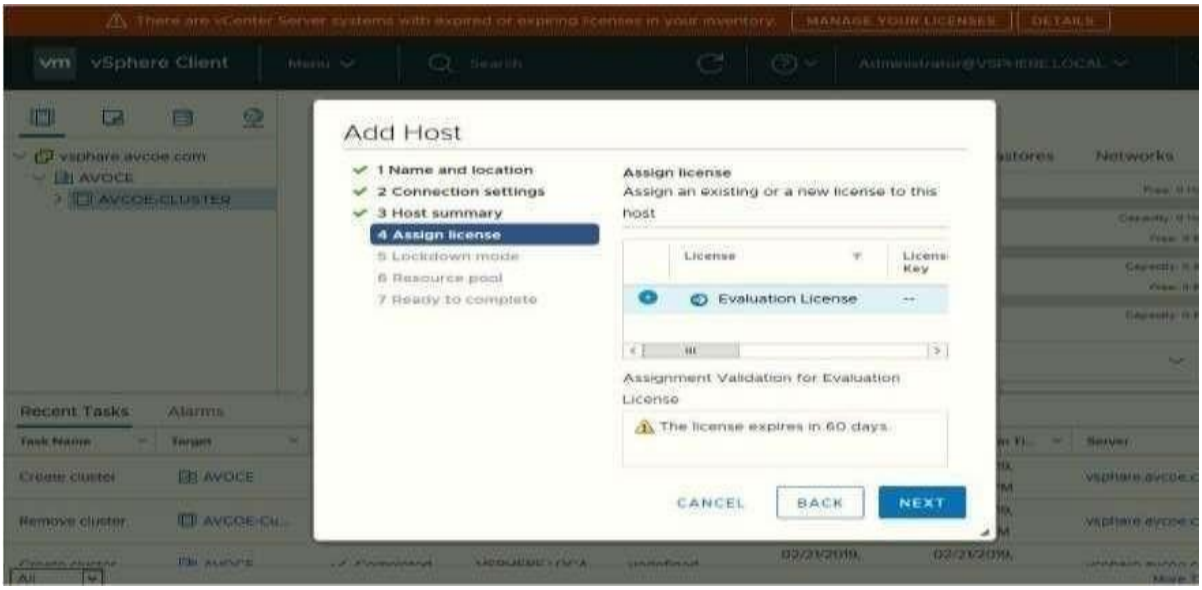
Add ho



Add host IP :

Enter host cred

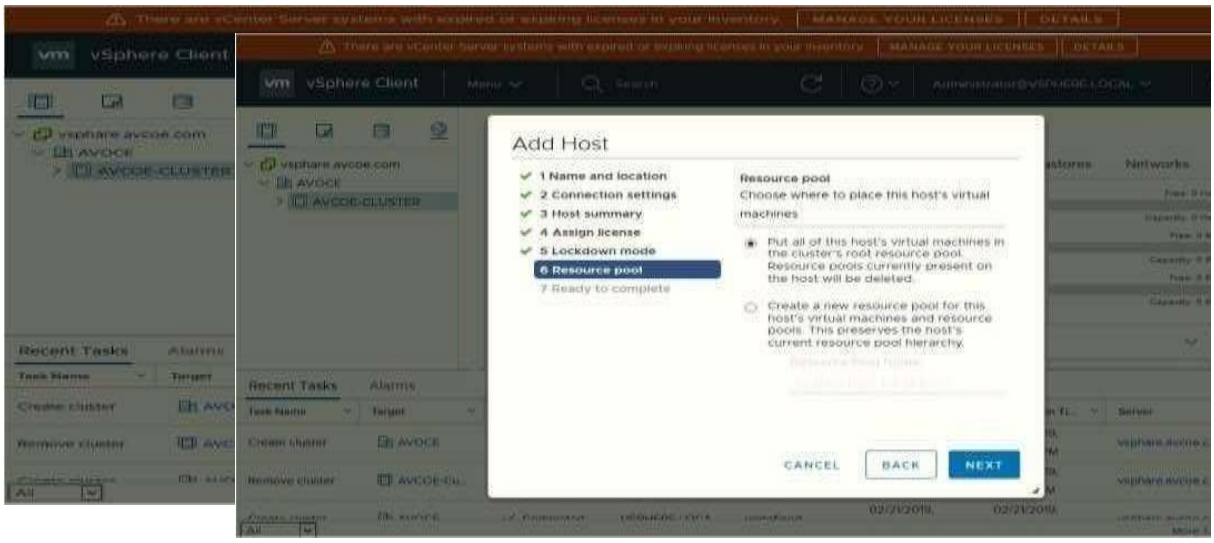




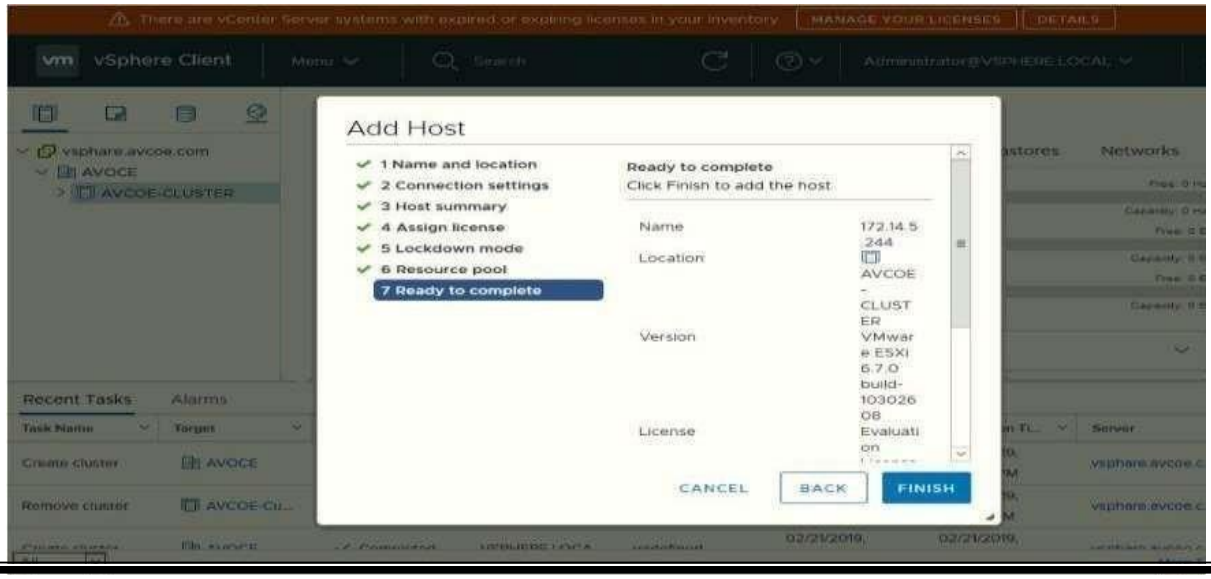
Hot summary :

Lock Down mode:

Add Host In Pool:

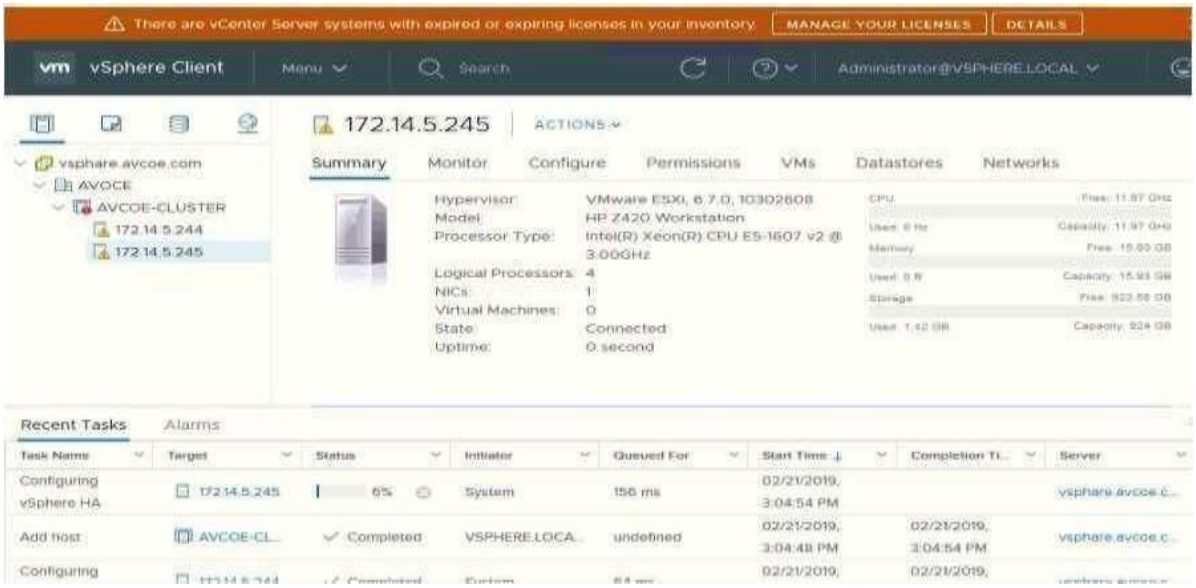


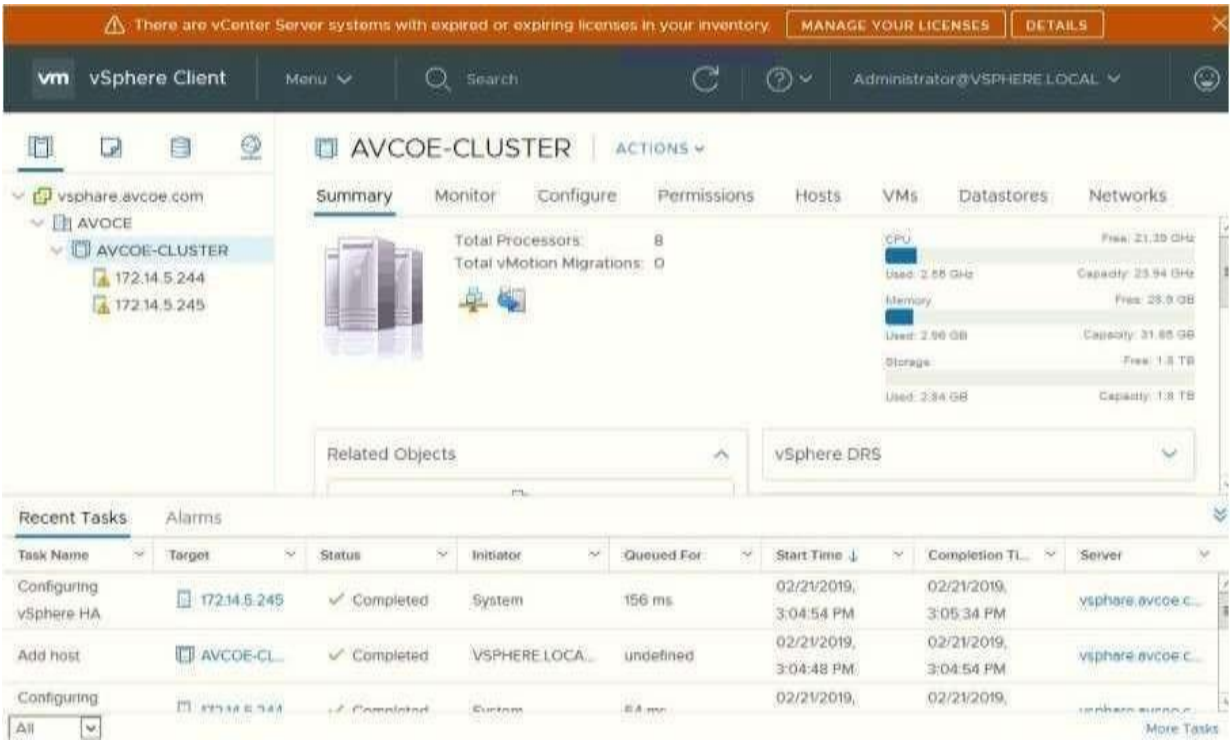
Finish:



Host View and View Config:

Cluster View and Configuration:





Conclusion: Like this we have configure VSphere Private Cloud