//Practical 1

BFS-DFS

//Cell-1

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

//Cell 2

```
def bfs(visited, graph, start_node, goal_node):
   visited.append(start_node)
   queue.append(start_node)
   while queue:
      m = queue.pop(0)
      print(m)
      if m == goal_node:
         print("Node is Found !!! ")
         break
      else:
         for n in graph[m]:
            if n not in visited:
               visited.append(n)
               queue.append(n)
```

```
print("The BFS Traversal is : ")
bfs(visited, graph, 'A', 'D')
```

//Cell 3

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

//Cell 4

```
def dfs(graph, start, goal):
    print("DFS traveral is: ")
    stack.append(start)
    visited.append(start)
    while stack:
        node = stack[-1]
        stack.pop()
        print("Node: ", node)
        if node == goal:
            print("Goal node found!")
            return
        for n in graph[node]:
            if n not in visited:
```

```
            visited.append(n)
            stack.append(n)
dfs(graph, 'A', "D")
```

----------END----------

//Practical 2

Implement A* Algorithm

//Cell 1

```
Import copy
final = [[1,2,3],[4,5,6],[7,8,-1]]
initial = [[1,2,3],[-1,4,6],[7,5,8]]
```

//Cell 2

```
#function to find heuristic cost
def gn(state, finalstate):
        count = 0
        for i in range(3):
                for j in range(3):
                        if(state[i][j]!=-1):
                                if(state[i][j] != finalstate[i][j]):
                                        count+=1
        return count
def findposofblank(state):
        for i in range(3):
                for j in range(3):
                        if(state[i][j] == -1):
                                return [i,j]
def move_left(state, pos):
        if(pos[1]==0):
                return None
```

```
        retarr = copy.deepcopy(state)
        retarr[pos[0]][pos[1]],retarr[pos[0]][pos[1]-1] = retarr[pos[0]][pos[1]-1],retarr[pos[0]][pos[1]]
        return retarr
def move_up(state, pos):
        if(pos[0]==0):
                return None
        retarr = copy.deepcopy(state)
        #for i in state:
                #retarr.append(i)
        retarr[pos[0]][pos[1]],retarr[pos[0]-1][pos[1]] = retarr[pos[0]-1][pos[1]],retarr[pos[0]][pos[1]]
        return retarr
def move_right(state, pos):
        if(pos[1]==2):
                return None
        retarr = copy.deepcopy(state)
        #for i in state:
                #retarr.append(i)
        retarr[pos[0]][pos[1]],retarr[pos[0]][pos[1]+1] = retarr[pos[0]][pos[1]+1],retarr[pos[0]][pos[1]]
        return retarr
def move_down(state, pos):
        if(pos[0]==2):
                return None
        retarr = copy.deepcopy(state)
        retarr[pos[0]][pos[1]],retarr[pos[0]+1][pos[1]] = retarr[pos[0]+1][pos[1]],retarr[pos[0]][pos[1]]
        return retarr
def printMatrix(matricesArray):
        print("")
        counter = 1
```

```
        for matrix in matricesArray:
                print("Step {}".format(counter))
                for row in matrix:
                        print(row)
                counter+=1
                print("")
def eightPuzzle(initialstate, finalstate):
        hn=0
        explored = []
        while(True):
                explored.append(initialstate)
                if(initialstate == finalstate):
                        break
                hn+=1
                left = move_left(initialstate, findposofblank(initialstate))
                right = move_right(initialstate, findposofblank(initialstate))
                up = move_up(initialstate, findposofblank(initialstate))
                down = move_down(initialstate, findposofblank(initialstate))
                fnl=1000
                fnr=1000
                fnu=1000
                fnd=1000
                if(left!=None):
                        fnl = hn + gn(left,finalstate)
                if(right!=None):
                        fnr = hn + gn(right,finalstate)
                if(up!=None):
                        fnu = hn + gn(up,finalstate)
```

```
            if(down!=None):
                fnd = hn + gn(down,finalstate)
            minfn = min(fnl, fnr, fnu, fnd)
            if((fnl == minfn) and (left not in explored)):
                initialstate = left
            elif((fnr == minfn) and (right not in explored)):
                initialstate = right
            elif((fnu == minfn) and (up not in explored)):
                initialstate = up
            elif((fnd == minfn) and (down not in explored)):
                initialstate = down
        printMatrix(explored)
#eightPuzzle(initial, final)
def main():
    while(True):
        ch = int(input("PRESS 1 to continue and 0 to Exit : "))
        if(not ch):
            break
        start = []
        print("START STATE\n")
        for i in range(3):
            arr=[]
            for j in range(3):
                a = int(input("Enter element at  {},{}: ".format(i,j)))
                arr.append(a)
            start.append(arr)
        final = []
        print("FINAL STATE\n")
```

```
        for i in range(3):
            arr=[]
            for j in range(3):
                a = int(input("Enter element at  {},{}: ".format(i,j)))
                arr.append(a)
            final.append(arr)
        eightPuzzle(start, final)
```

//Cell 3

```
main()
```

----------END----------

//Practical 3

Greedy Search for Selection Sort

```
def selectionSort(array, size):
  for step in range(size):
    min_idx = step
    for i in range(step + 1, size):
      if array[i] < array[min_idx]:
        min_idx = i
    (array[step], array[min_idx]) = (array[min_idx], array[step])
data = [-2, 45, 0, 11, -9]
size = len(data)
selectionSort(data, size)
print('Sorted Array in Ascending Order:')
print(data)
```

----------END----------

//Practical 4

Constraint Satisfaction Problem

```
n=4
```

```
a=[[0,0,0,0],[0,0,0,0],[0,0,0,0],[0,0,0,0]]
b={}
def isColumnSafe(r,c):
    while(r>=0):
        if(a[r][c] == 1):
            return 0
        r = r-1
    return 1
def isLeftDiagonalSafe(r,c):
    while(r>=0 and c>=0):
        if(a[r][c] == 1):
            return 0
        r = r-1
        c = c-1
    return 1
def isRightDiagonalSafe(r,c):
    while(r>=0 and c<n):
        if(a[r][c]==1):
            return 0
        r = r-1
        c = c+1
    return 1
def isSafe(r,c):
    if(isColumnSafe(r,c) and isLeftDiagonalSafe(r,c) and isRightDiagonalSafe(r,c)):
        return True
    return False
def chessboard(r,c):
    if(r>=n):
```

```
        return
    p = 0
    while c<n:
        p = isSafe(r,c)
        if p == 1:
            a[r][c] = 1
            b.update({r:c})
            break
        c=c+1
    if p==1:
        chessboard(r+1,0)
    else:
        a[r-1][b.get(r-1)]=0
        chessboard(r-1,int(b.get(r-1))+1)
chessboard(0,0)
print("Matrix is:- ",a)
print("Dictionary is:- ",b)
```

----------END----------

//Practical 5

Chatbot

//Cell 1

```
import chatterbot
```

//Cell 2

```
import random
# Define possible responses
greetings = ['hello', 'hi', 'hey']
farewells = ['goodbye', 'bye', 'see you later']
questions = ['how are you?', 'what is your name?', 'what do you do?']
```

```
responses = {
    'hello': ['Hello!', 'Hi there!', 'Hey!'],
    'goodbye': ['Goodbye!', 'Bye!', 'See you later!'],
    'how are you?': ['I am doing well, thank you!', 'Not too bad, thanks for asking!', 'I am great!'],
    'what is your name?': ['My name is Chatbot!', 'You can call me Chatbot!', 'I am Chatbot!'],
    'what do you do?': ['I am here to help you with your queries!', 'I can assist you with anything you need!', 'I am a customer service chatbot!']
}
# Define function to generate response
def generate_response(message):
    if message.lower() in greetings:
        return random.choice(responses['hello'])
    elif message.lower() in farewells:
        return random.choice(responses['goodbye'])
    elif message.lower() in questions:
        return random.choice(responses[message.lower()])
    else:
        return "I'm sorry, I didn't understand your message."
# Define main function to run chatbot
def main():
    print("Welcome to the chatbot!")
    while True:
        message = input("You: ")
        if message.lower() == 'exit':
            print("Chatbot: Goodbye!")
            break
        else:
            response = generate_response(message)
            print("Chatbot:", response)
```

```
if __name__ == '__main__':

 main()
```

----------END----------

Creating an Application in SalesForce.com using Apex programming Language

Before moving to the practical lets understand some important topics related to Salesforce.

What is Salesforce?

Salesforce provides software and services aimed at creating relevant customer experiences. Businesses can use Salesforce services to better connect with partners, customers, and potential. Companies can track customer activity, market to customers, and perform many other tasks using Salesforce CRM customers.

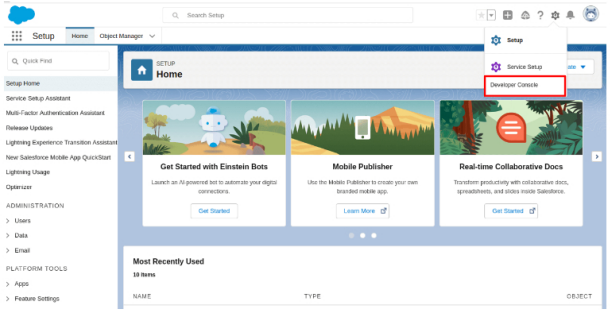
What are the services provided Salesforce?

- SAAS (Software-as-a-Service)
- PAAS (Platform-as-a-Service)
- IAAS (Infrastructure-as-a-Service)

Sign up to the Salesforce using the following link:

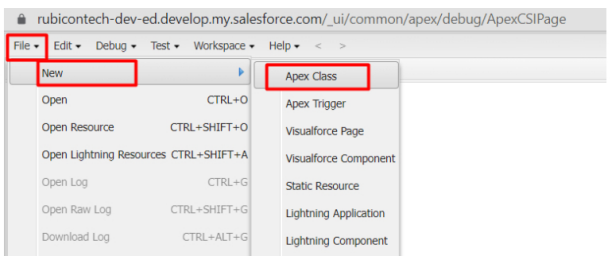
https://developer.salesforce.com/signup

After signing up go to **Setup** and click on **Developer Console**.



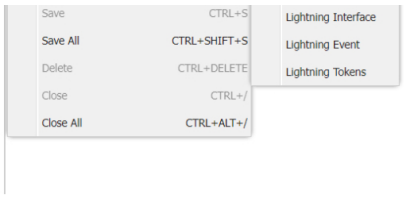
Now, in Developer Console click on **File** -> **New** -> **Apex Class** and Name it as **Demo1**

Note: If your **Apex Class is disabled** then signup through the given link: **https://developer.salesforce.com/signup**



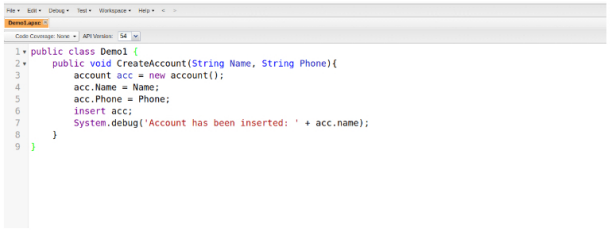
Recent Posts

Categories



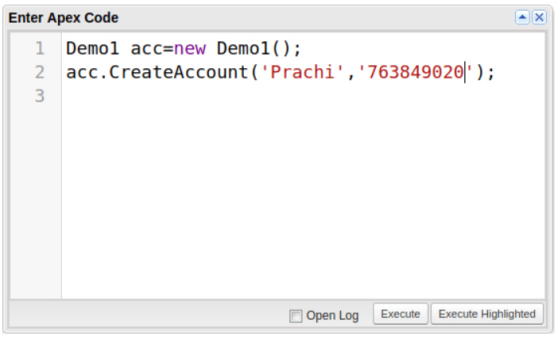
1. Write the code to create an account.

Do not forget to save it.

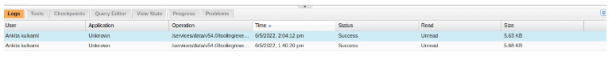


```
public class Demo1 {
    public void CreateAccount(String Name, String Phone){
        account acc = new account();
        acc.Name = Name;
        acc.Phone = Phone;
        insert acc;
        System.debug('Account has been inserted: ' + acc.name);
    }
}
```

Click on **Debug** -> **Open Execute Anonymous Window** and write the code and then **Execute** it.

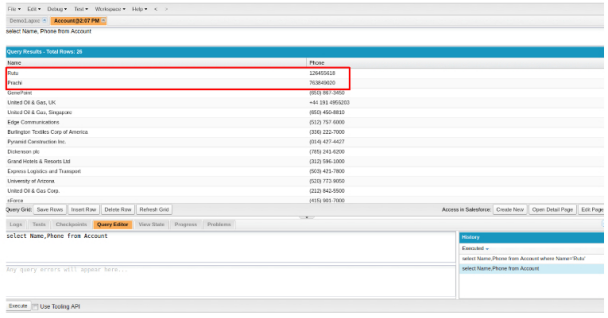


You can see the record has been successfully added.



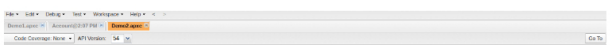
Similarly, you can add number of records. To retrieve the records go to **Query Editor** and type the query language and **Execute** it.

You can see the records added.



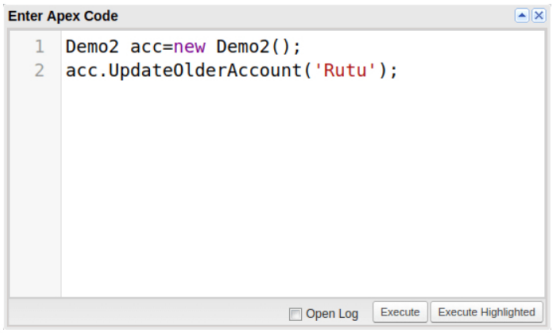
2. Write the code to update the account.

Follow the same steps: **File** -> **New** -> **Apex Class** and name it as **Demo2**.



```
public class Demo2 {
    public void UpdateOlderAccount(String Name){
        List<Account> oldAccounts = new List<Account>();
        for(Account acc: [SELECT Name from Account WHERE Phone='785323674']){
            acc.Name = Name;
            oldAccounts.add(acc);
            System.debug('Account has been updated ' + acc.name);
        }
        update oldAccounts;
    }
}
```

Again click on **Debug** -> **Open Execute Anonymous Window** and write the code.



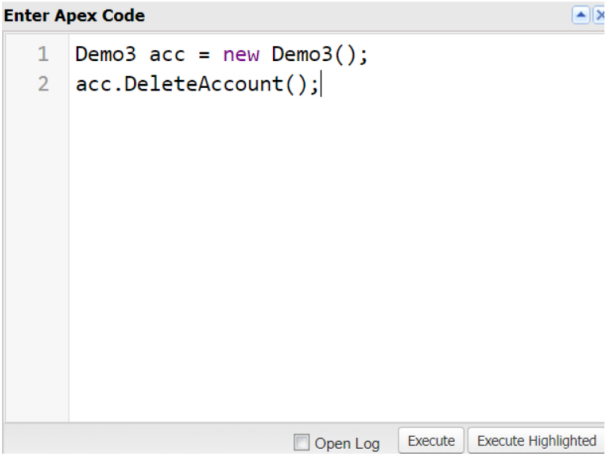
To view the records in Salesforce

3. Write the code to delete the records.

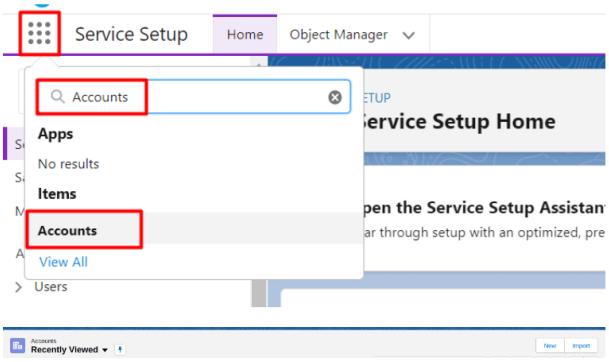
Follow the above steps: **File** -> **New** -> **Apex Class** and name it as **Demo3**.

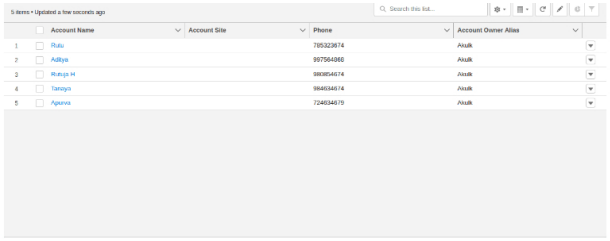
```
public class Demo3 {
    public void DeleteAccount(){
        List<Account> oldAccounts = new List<Account>();
        for(Account acc:[select Name,Phone from Account where Name='Smita']){
            delete acc;
            System.debug('Account deleted: '+acc.name);
        }
        update oldAccounts;
    }
}
```

Again click on **Debug** -> **Open Execute Anonymous Window** and write the code.



To view the records in Salesforce go to **Setup** and search for **Accounts** and click on it.





Spread this useful information with your friends if you liked.

← Previous Post

Next Post →

Leave a Comment

Your email address will not be published. Required fields are marked *

Type here..

Name*

Email*

Website

Save my name, email, and website in this browser for the next time I comment.

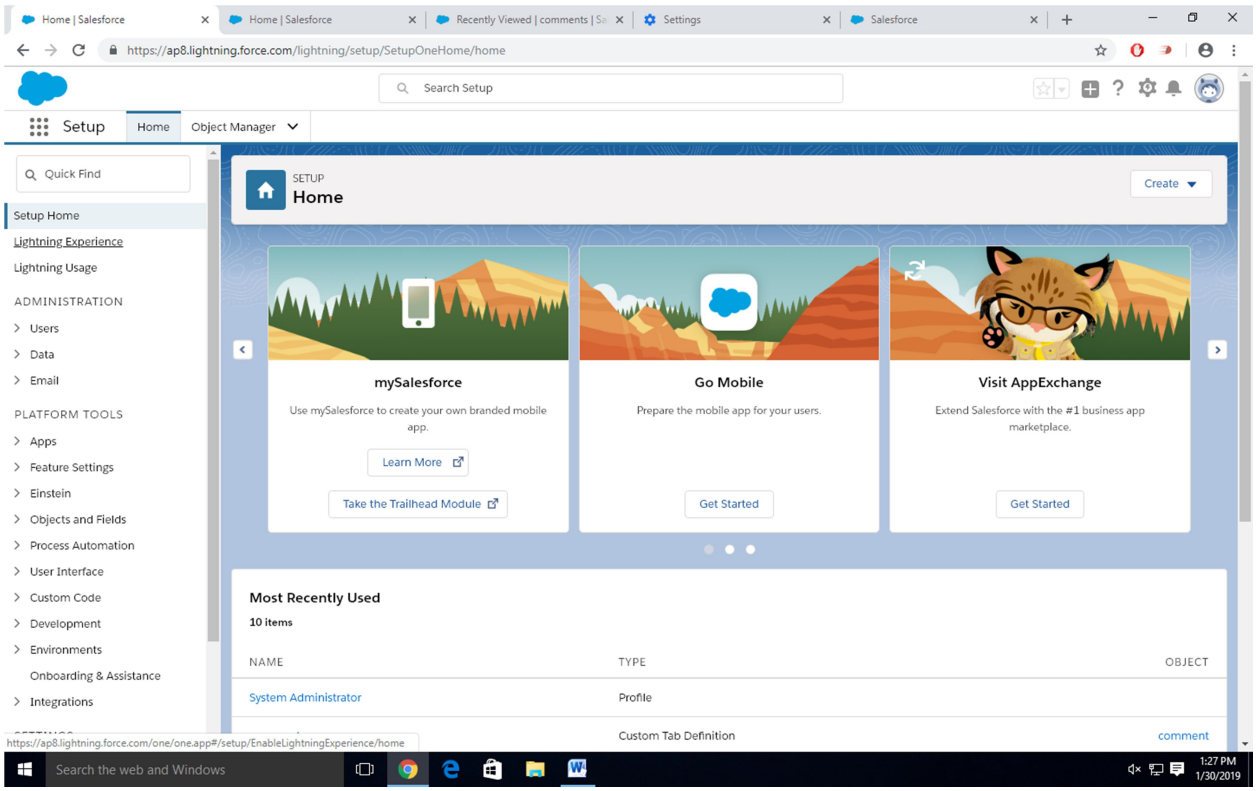
POST COMMENT »

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

Step-1: Click on Lightning Experience

Step-2: Click on Setup and select Setup for current App.

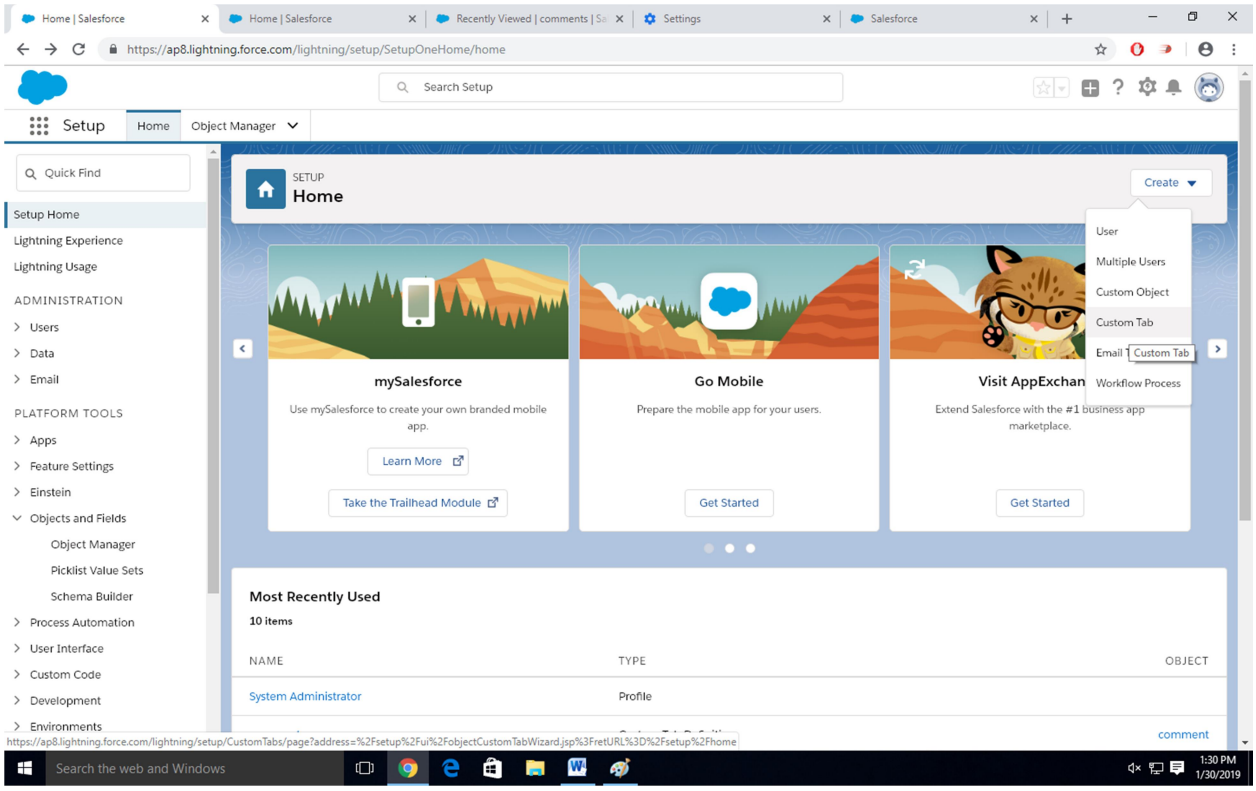
Step-3:



Click on Create an Object

So Click on Object Manager Tab next to Home Tab

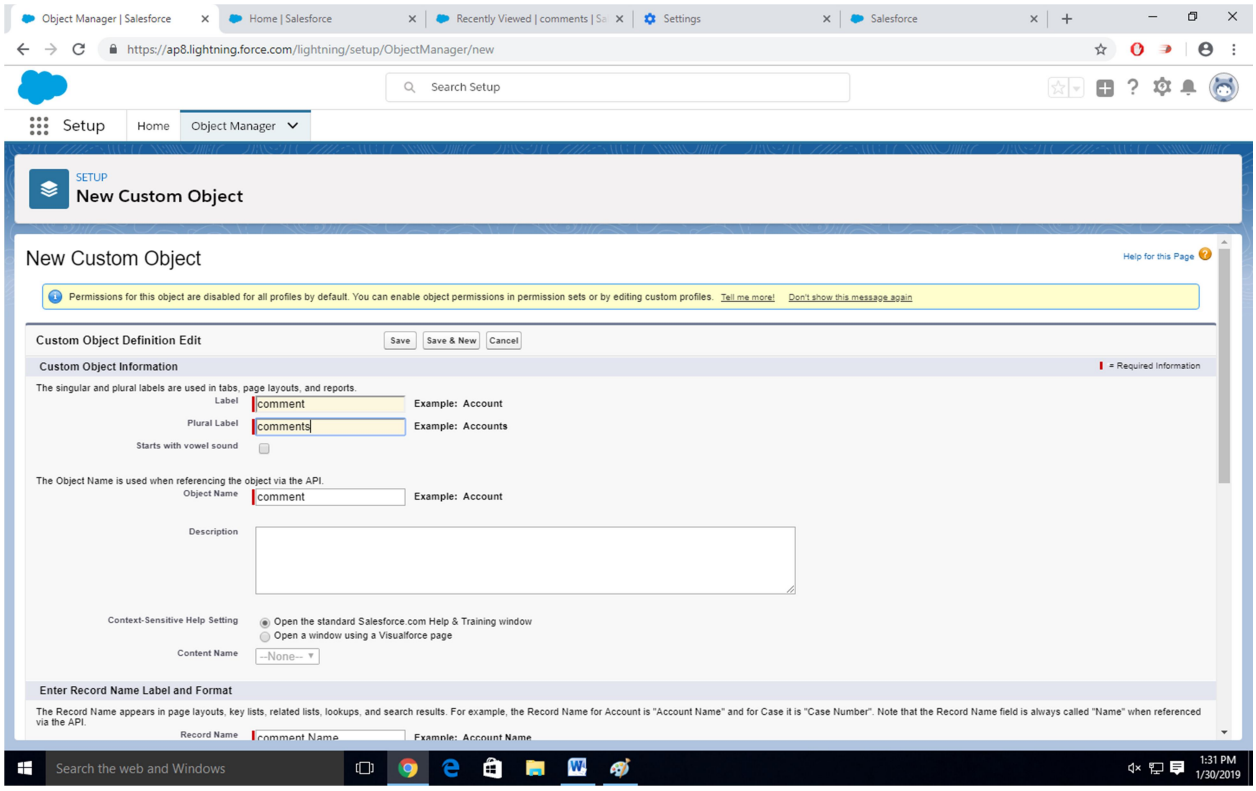
Click on Create –Custom Object



Step-4 New custom object page Open

Label as a-Comment

Plural label- comments

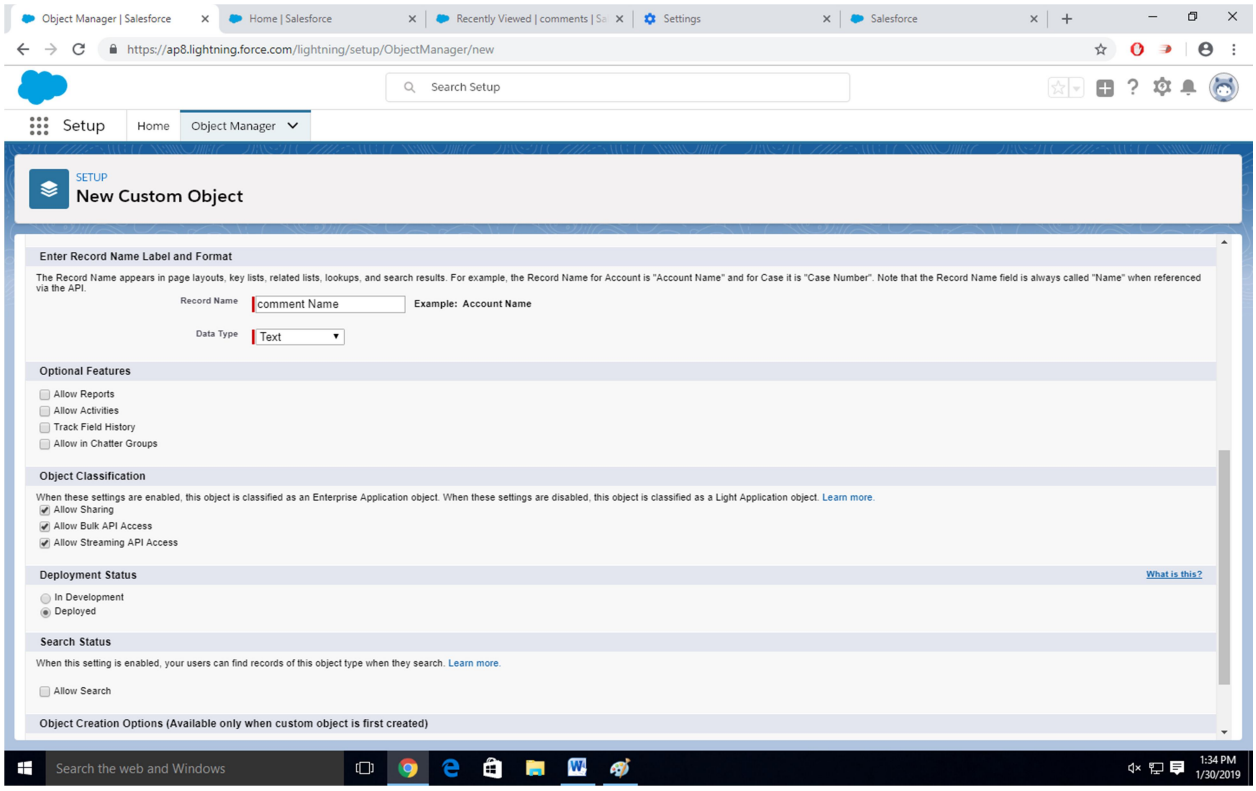


Give Record Name as –comment name

Data type- text

Select Allow Reports Check Box

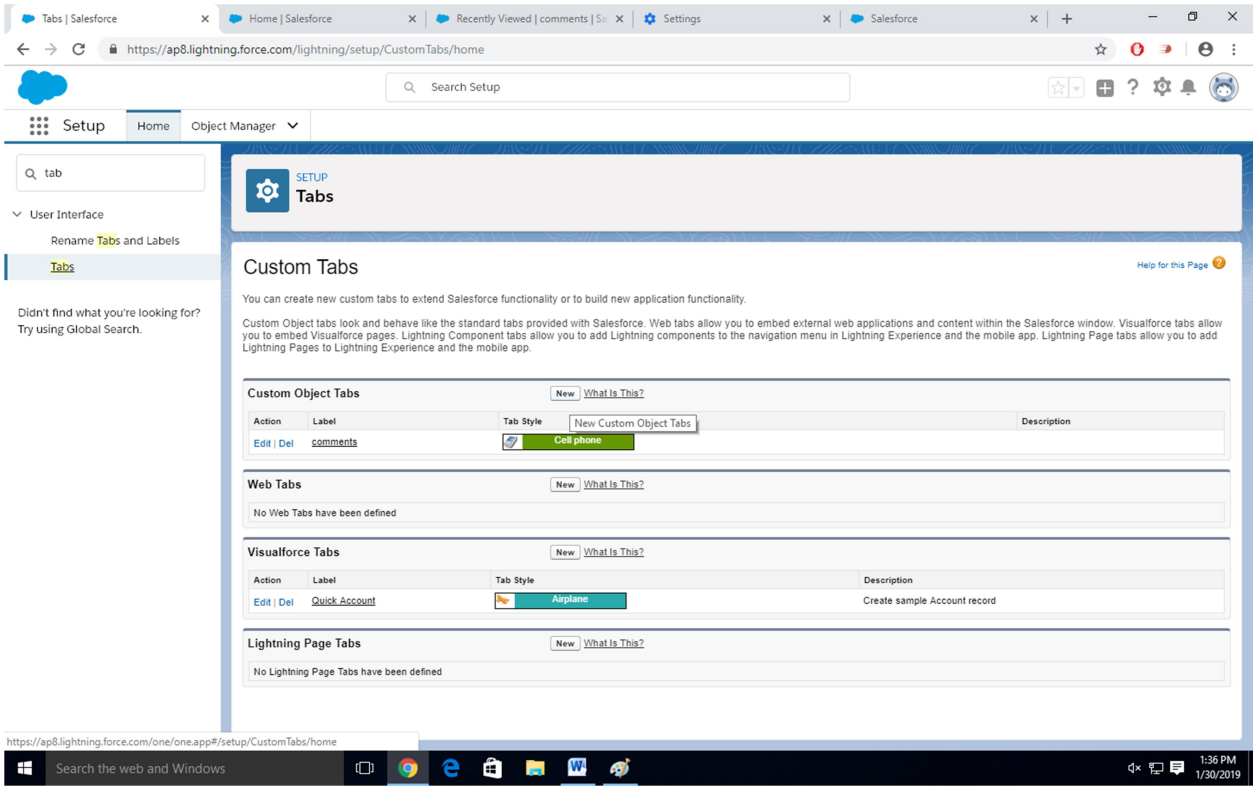
Click on Save



Step-5

Click on Home-Search Tabs in Quick search

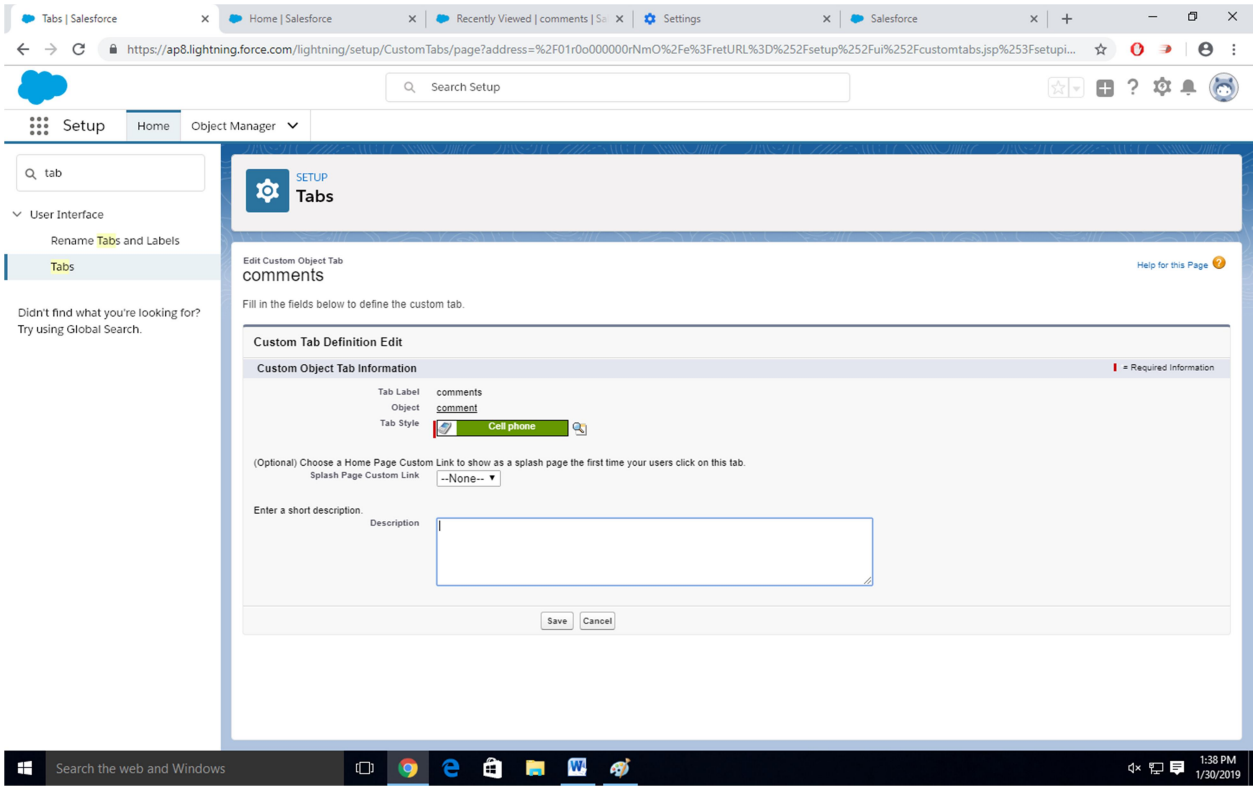
Select Custom Object-Click on New



Step-6

For Object Select Comment

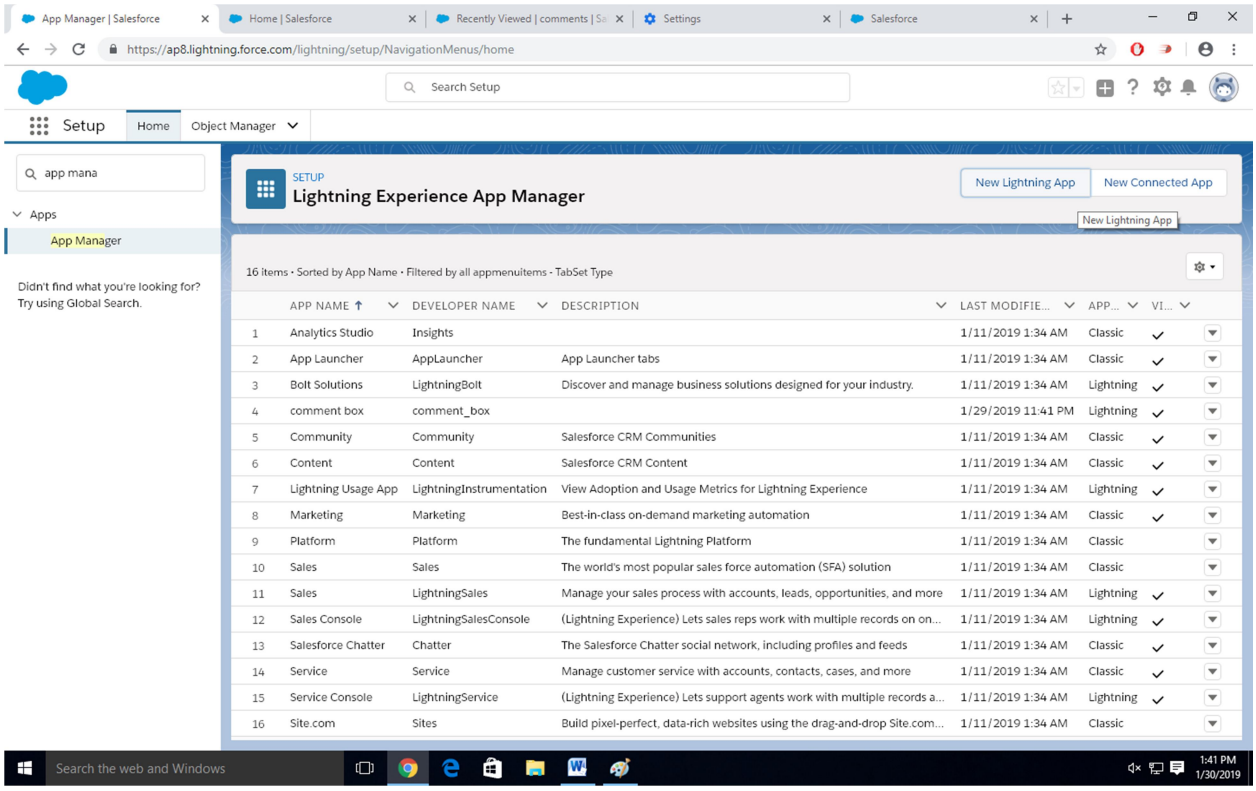
For Tab Style Select Any Icon



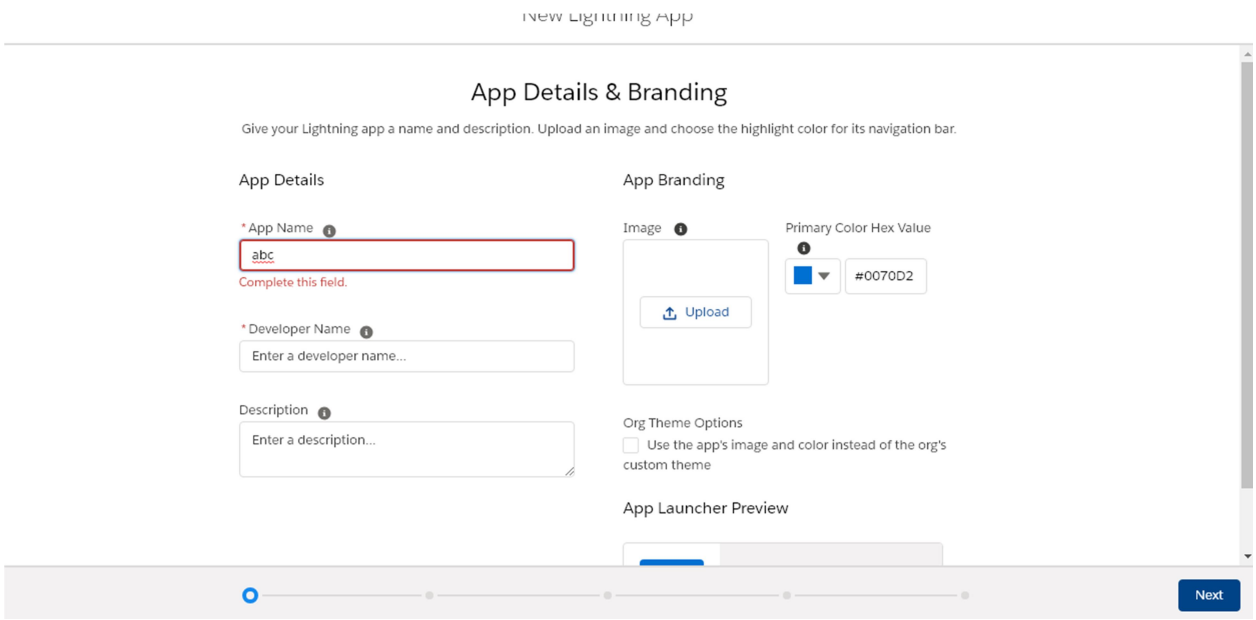
Click-Next-Next-Save

Step-7

Search App Manager in Quick Search and select app manager



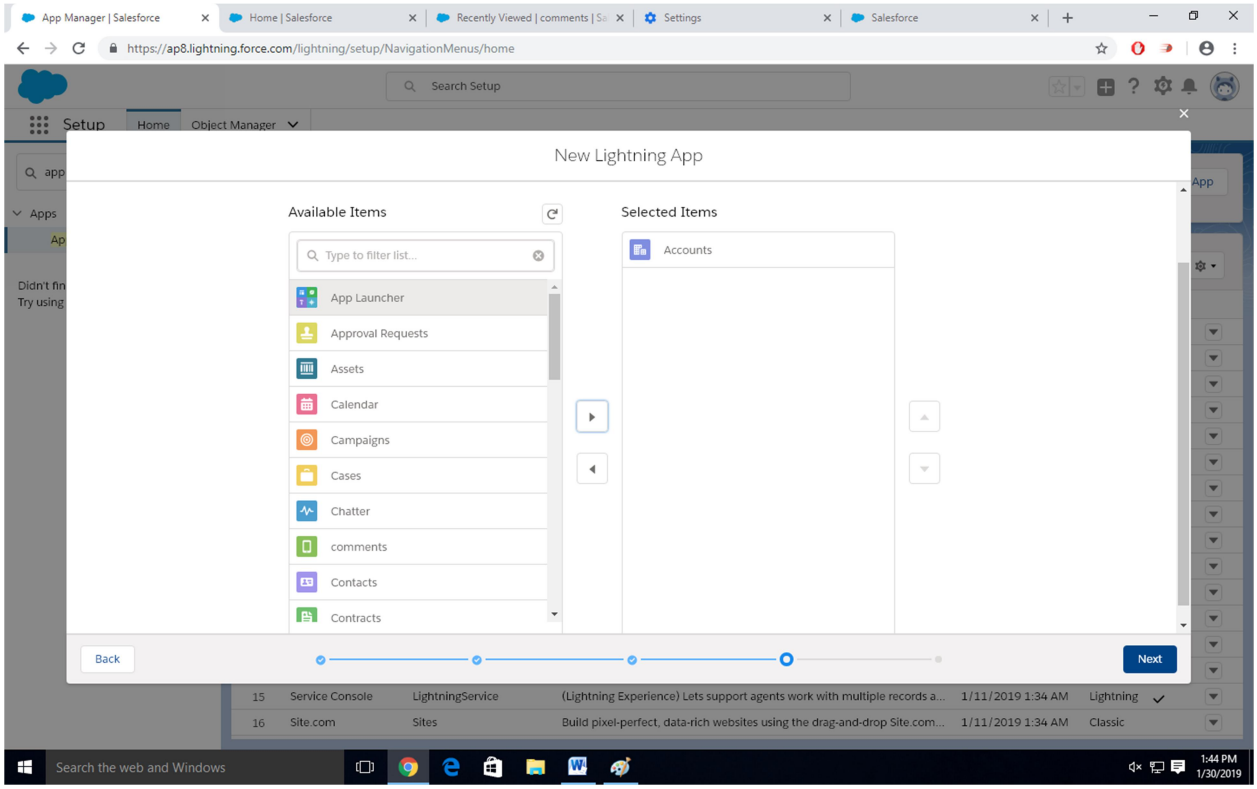
Enter name to app name



Click on Next-Next-Next.

Select Items (Contacts,Comment)

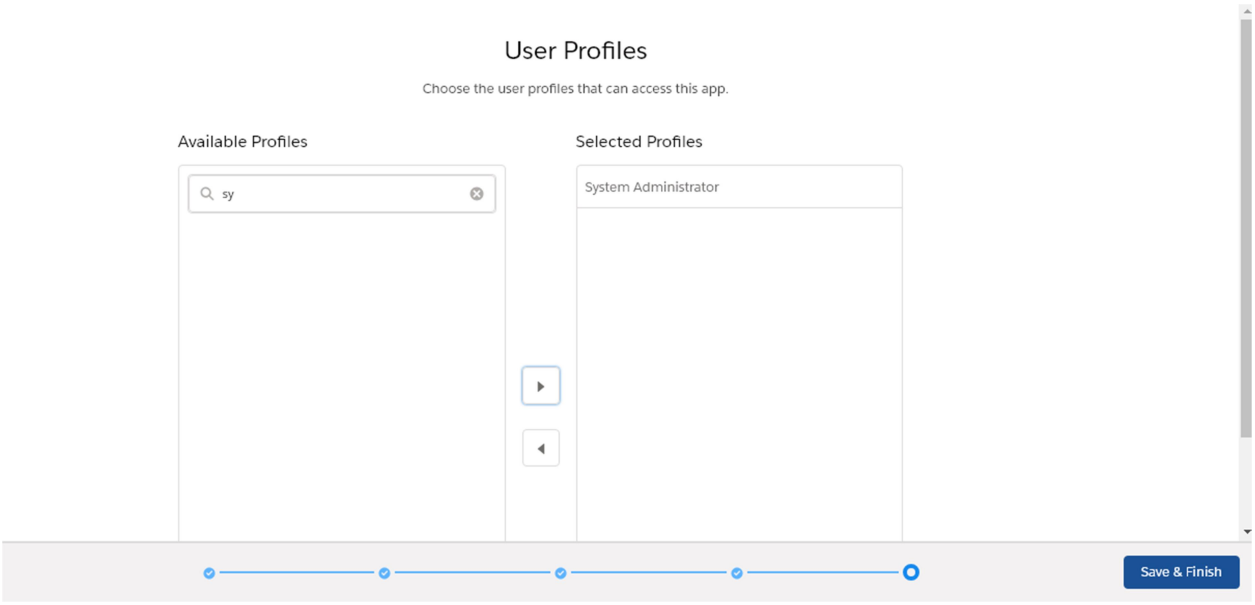
Click on Next



Step-8

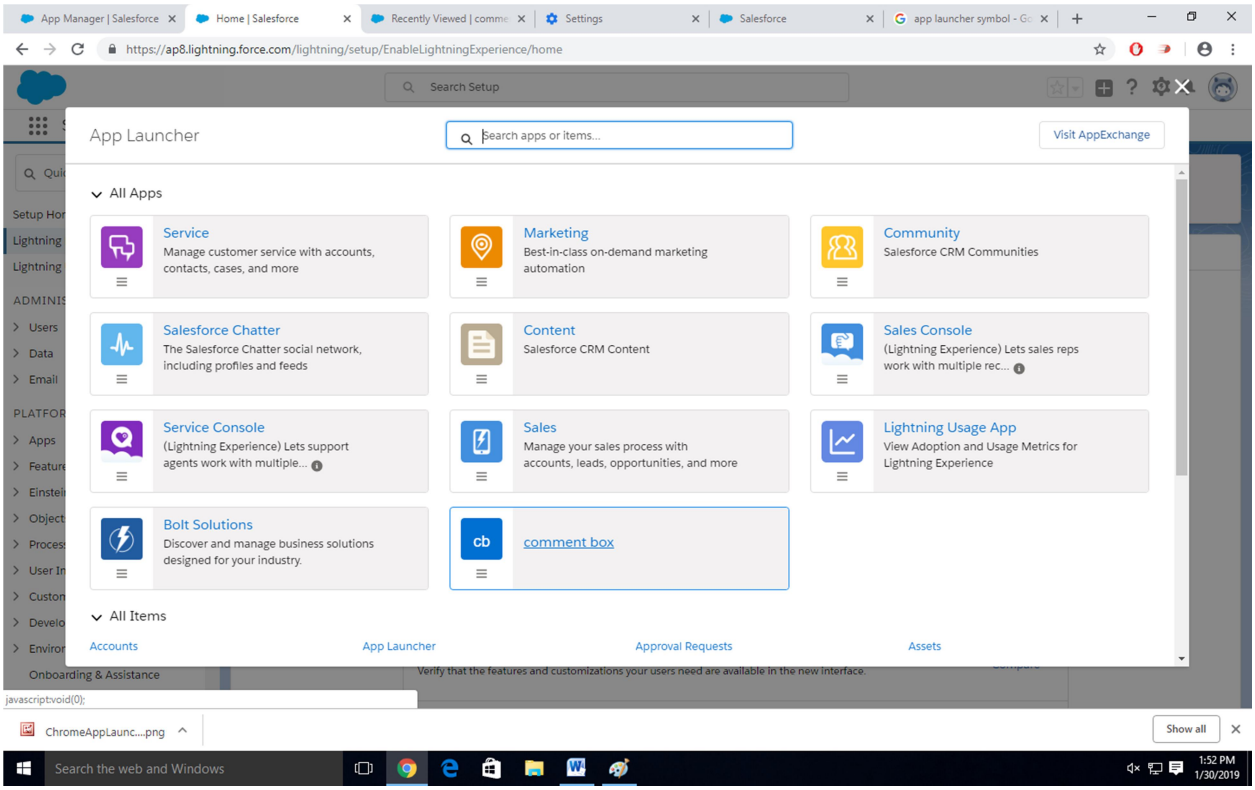
Select Profiles (System Administrator) and move to selected profile.

Click on Save and Finish.



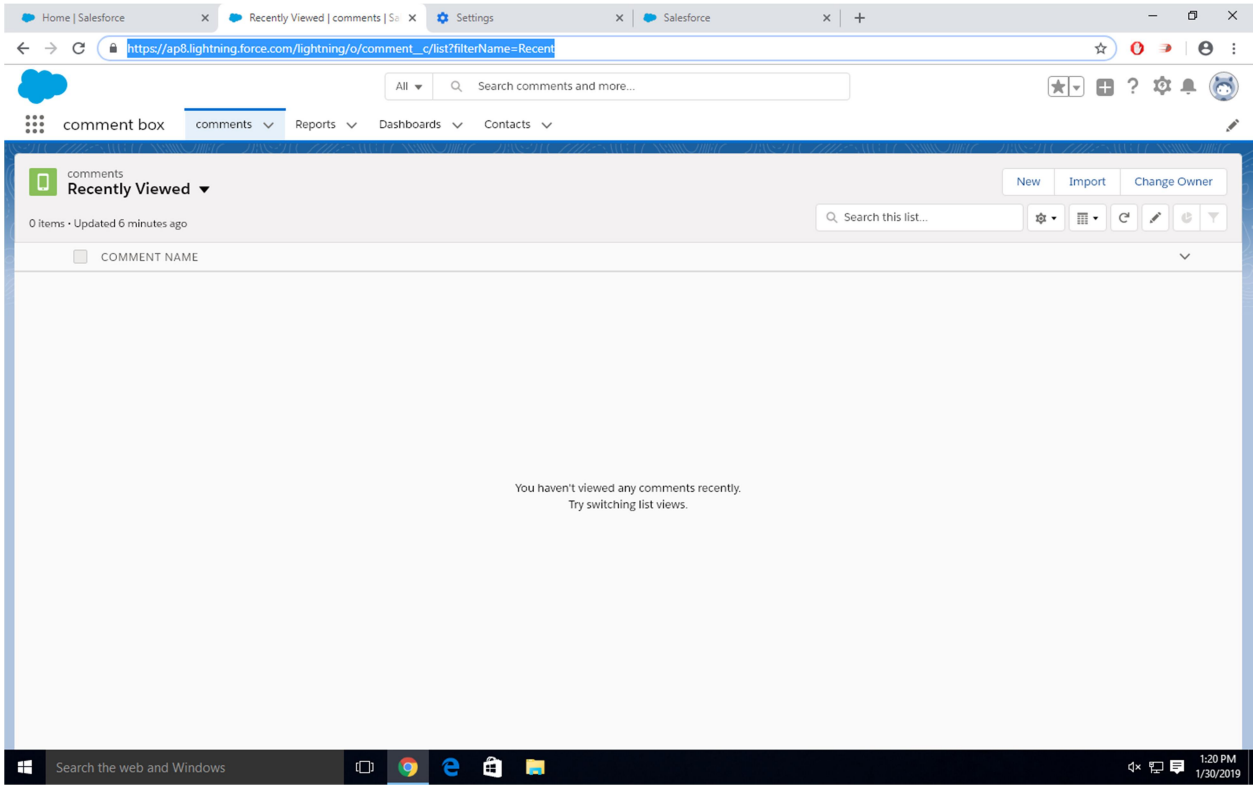
Step-9

Click on App Launcher  Symbol and Select Comment Box App



Step-11

Tour the app



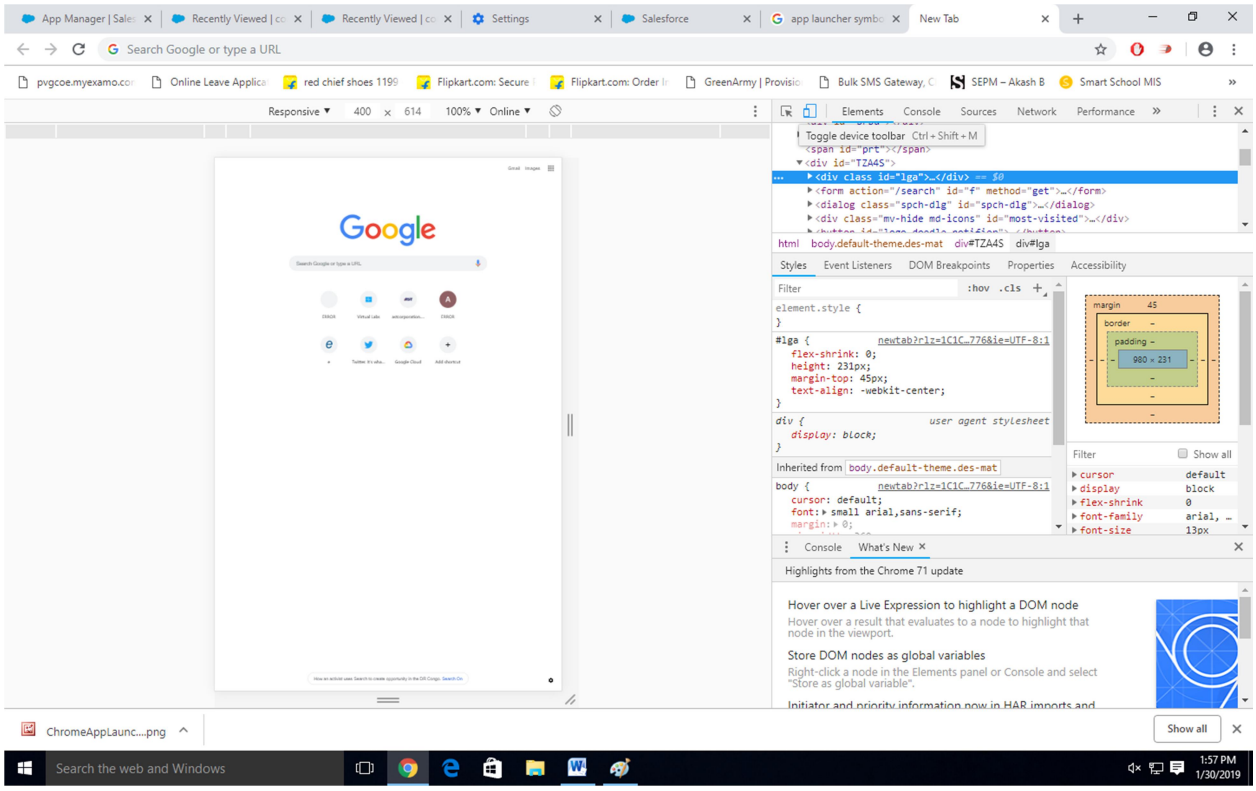
Step-12

Try out mobile app

-Select Chrome developer tools

-Open Chrome-Right Click on Chrome page- Select Inspect

-Click Toggle Device Mode Button to simulate your browser as a mobile device



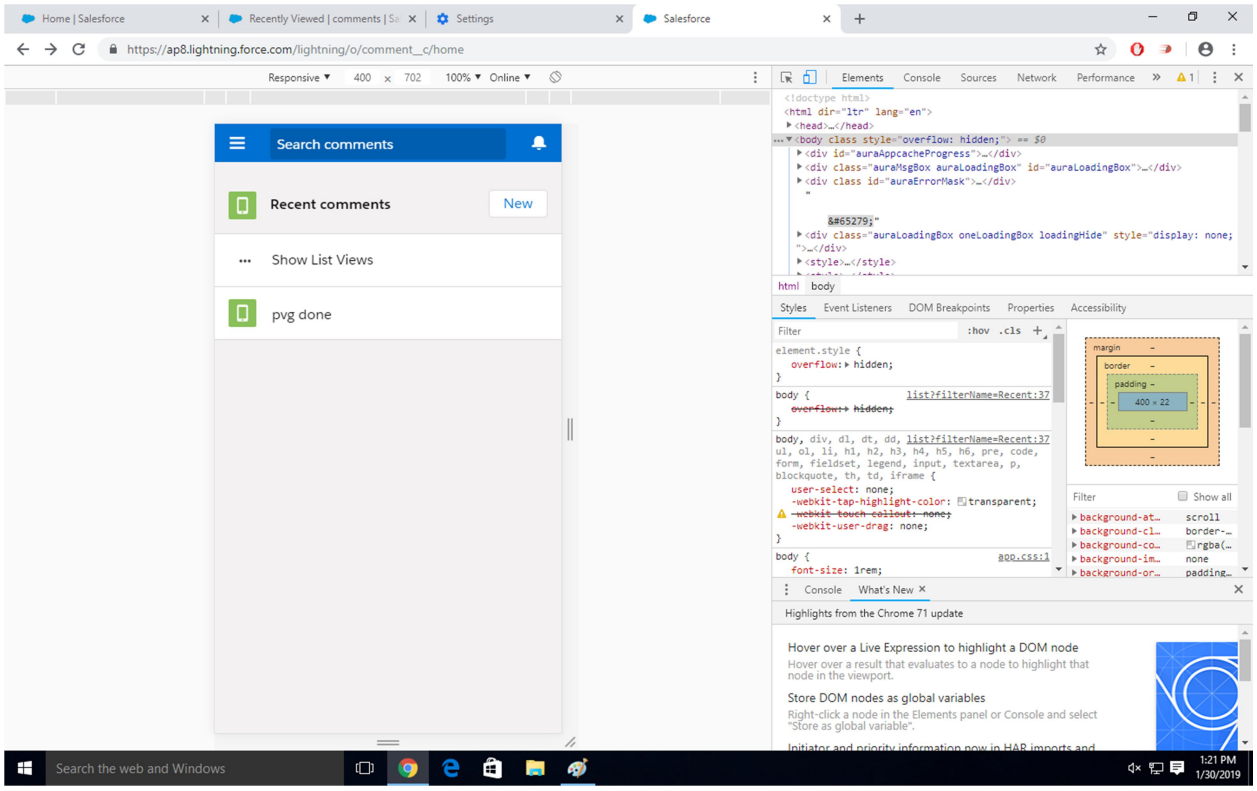
Step-13

To simulate the sales force mobile app in your browser, copy and paste in url from previous tab.Delete the part of the url immediately.

-Click on Left navigation bar

-Find comment object under recent and click on it

-Click new to create a comment


Home | Salesforce
Recently Viewed | comments | Sa
Settings
Salesforce
https://ap8.lightning.force.com/lightning/o/comment__c/home
Search comments
Recent comments
New
Show List Views
pvg done
Elements
Console
Sources
Network
Performance
Styles
Event Listeners
DOM Breakpoints
Properties
Accessibility
Console
What's New
Highlights from the Chrome 71 update
Hover over a Live Expression to highlight a DOM node
Store DOM nodes as global variables
Search the web and Windows
1:21 PM
1/30/2019