

//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 traversal 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?']

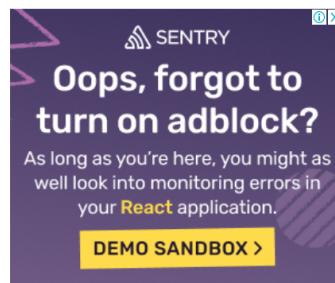
```
responses = {  
    'hello': ['Hello!', 'Hi there!', 'Hey!'],  
    'goodbye': ['Goodbye!', 'Bye!', 'See you later!'],  
    'how are you?:': ['I am doing well, thank you!', 'Not too bad, thanks for asking!', 'I am great!'],  
    'what is your name?:': ['My name is Chatbot!', 'You can call me Chatbot!', '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

Spread this useful information with your friends if you liked.





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.

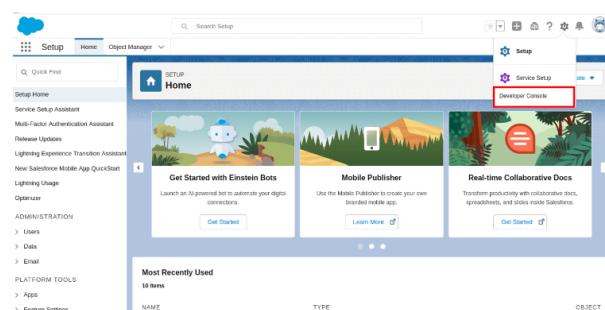
What are the services provided Salesforce?

- SaaS (Software-as-a-Service)
- PaaS (Platform-as-a-Service)
- IaaS (Infrastructure-as-a-Service)

Sign up to the Salesforce using the following link:

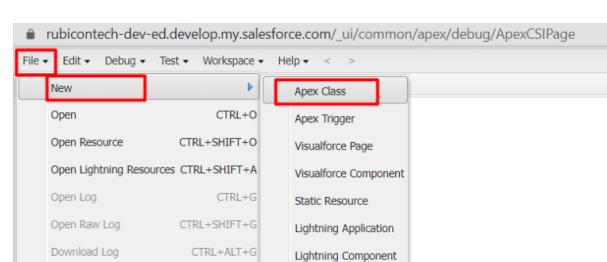
<https://developer.salesforce.com/signup>

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

[What is Deep Learning? Explained in 3 figures](#)

[SVM Algorithm with its types and terminologies with notes and Video for explanation](#)

[KNN Algorithm with notes and video for explanation with examples.](#)

[Implement a depth-first search algorithm and breadth-first search algorithm, use an undirected graph, and develop a recursive algorithm for searching for all the vertices of a graph or a tree data structure.](#)

[Sources of Big Data](#)

Categories

[App Development](#)

[JAVA](#)

[JavaScript](#)

[Python](#)

[Uncategorized](#)

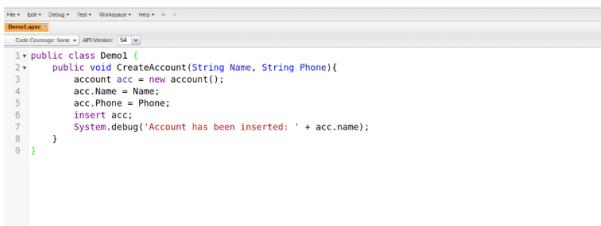
[Web Development](#)



Save	CTRL+S	Lightning Interface
Save All	CTRL+SHIFT+S	Lightning Event
Delete	CTRL+DELETE	Lightning Tokens
Close	CTRL+/	
Close All	CTRL+ALT+/	

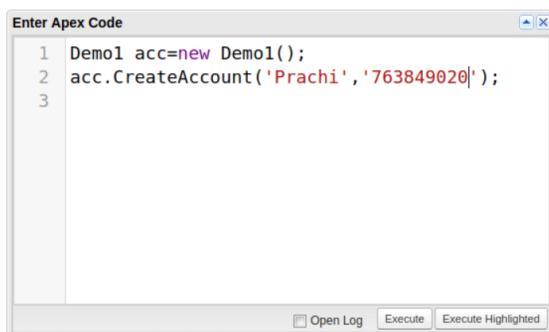
1. Write the code to create an account.

Do not forget to save it.



```
1• public class Demo1 {
2•     public void CreateAccount(String Name, String Phone){
3•         account acc = new account();
4•         acc.name = Name;
5•         acc.Phone = Phone;
6•         insert acc;
7•         System.debug('Account has been inserted: ' + acc.name);
8•     }
9• }
```

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



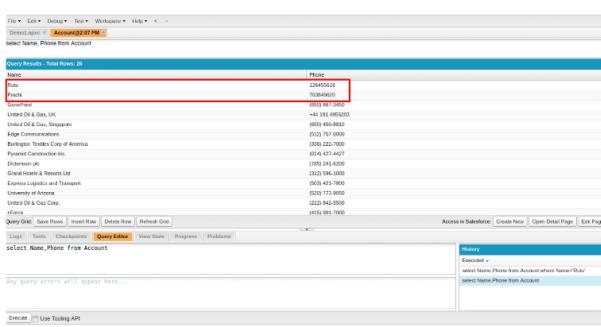
You can see the record has been successfully added.

User	Application	Operation	Time	Status	Retain	Size
Aditi Acharya	Unknown	Anonymous/54.180.104.194:443	6/5/2022, 2:04:12 pm	Success	Unread	5.66 KB
Aditi Acharya	Unknown	Anonymous/54.180.104.194:443	6/5/2022, 1:40:39 pm	Success	Unread	5.66 KB

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.



Oops, forgot to turn on adblock?

As long as you're here, you might as well look into monitoring errors in your **React** application.

DEMO SANDBOX >

SENTRY

2. Write the code to update the account.

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

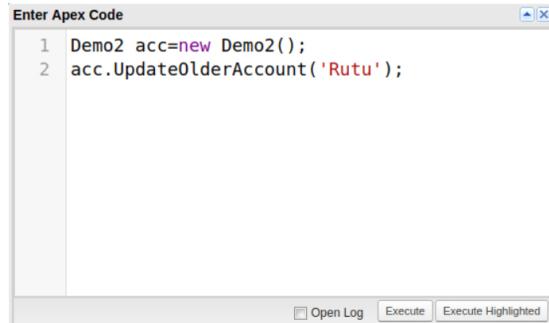


```

1 * public class Demo2 {
2     public void UpdateOlderAccount(String Name){
3         List<Account> oldAccounts = new List<Account>();
4         for(Account acc:[SELECT Name from Account WHERE Phone='785323674']){
5             acc.Name = Name;
6             oldAccounts.add(acc);
7             System.debug('Account has been updated ' + acc.name);
8         }
9         update oldAccounts;
10    }
11 }
12

```

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.

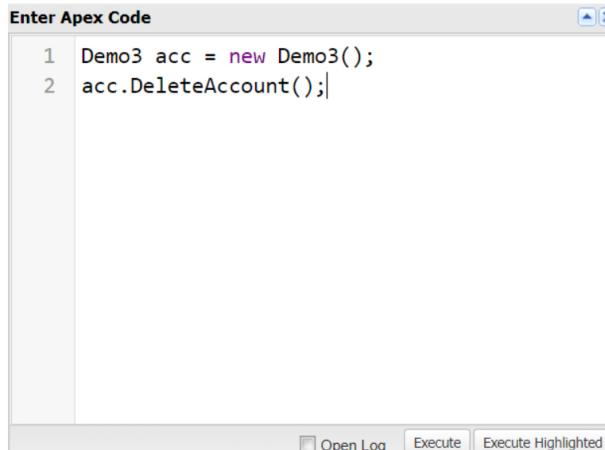
The screenshot shows the Salesforce IDE with the 'Demo3.apxc' file open. The code for Demo3 is:

```

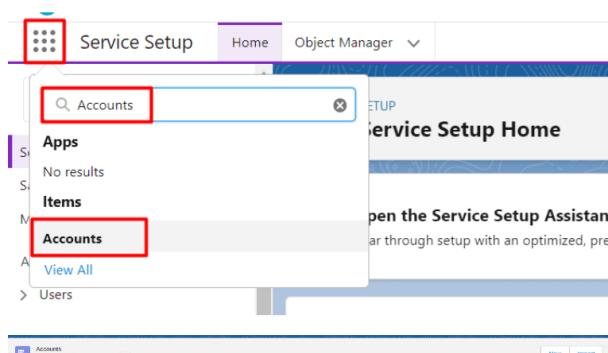
1 * public class Demo3 {
2     public void DeleteAccount(){
3         List<Account> oldAccounts = new List<Account>();
4         for(Account acc:[select Name,Phone from Account where Name='Smita']){
5             delete acc;
6             System.debug('Account deleted: '+acc.name);
7         }
8         update oldAccounts;
9     }
10 }

```

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.



5 items - Updated in few seconds ago			Search this list...	...
	Account Name	Account Site	Phone	Account Owner Alias
1	<input type="checkbox"/> Rite		780328114	Audit
2	<input type="checkbox"/> Adsys		979746966	Audit
3	<input type="checkbox"/> Rite H		980046114	Audit
4	<input type="checkbox"/> Tata		984634614	Audit
5	<input type="checkbox"/> Agenzia		724034719	Audit

 Summer Sport Products & Gear
Decathlon Wagholi

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 *

Name*	Email*	Website
-------	--------	---------

POST COMMENT

Email*

Website

Powered By Rubicon's Tech



Learn

[App Development](#)
[Web Development](#)
[Programming Languages](#)
[Visit Articles](#)

Our Services

WordPress Development

Web App Development

Contact Information

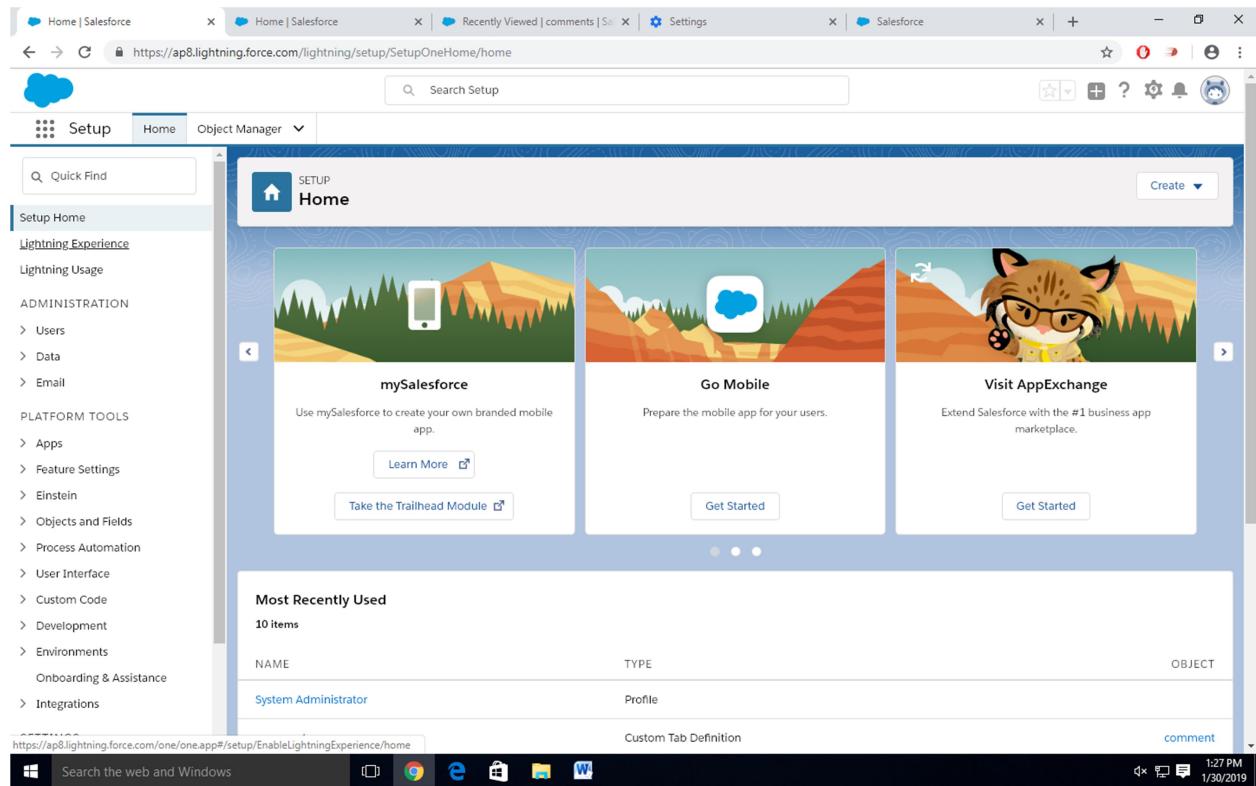
Instagram
LinkedIn
Twitter
netrofly@gmail.com
studyherofficial@gmail.com

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.

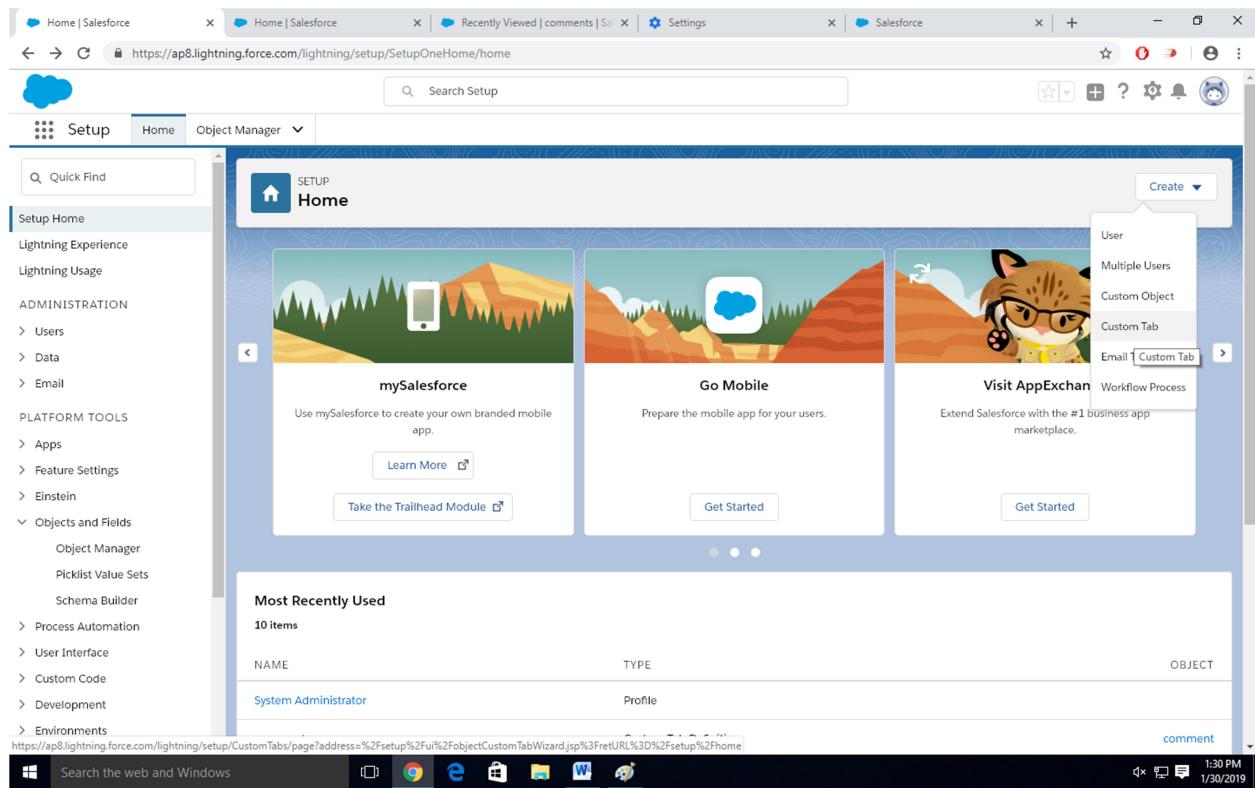
Step-3:



Click on Create an Object

So Click on Object Manager Tab next to Home Tab

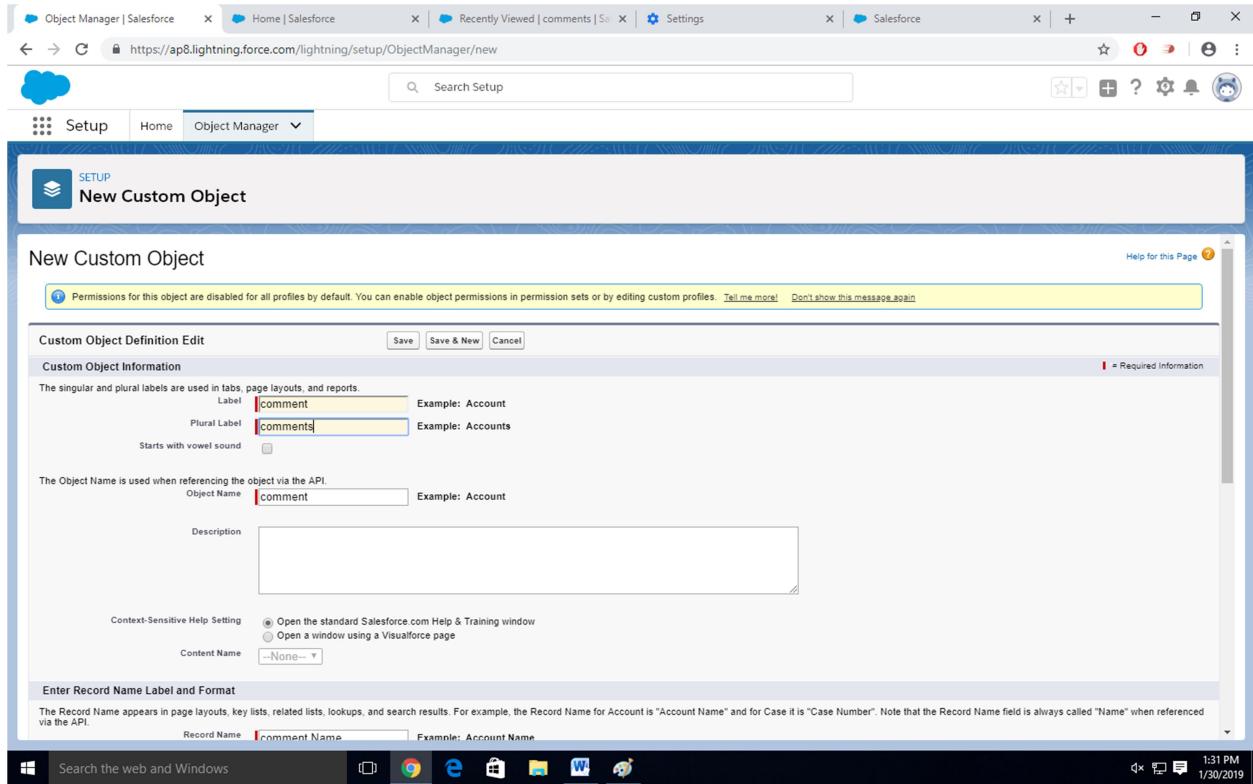
Click on Create –Custom Object



Step-4 New custom object page Open

Label as a-Comment

Plural label- comments

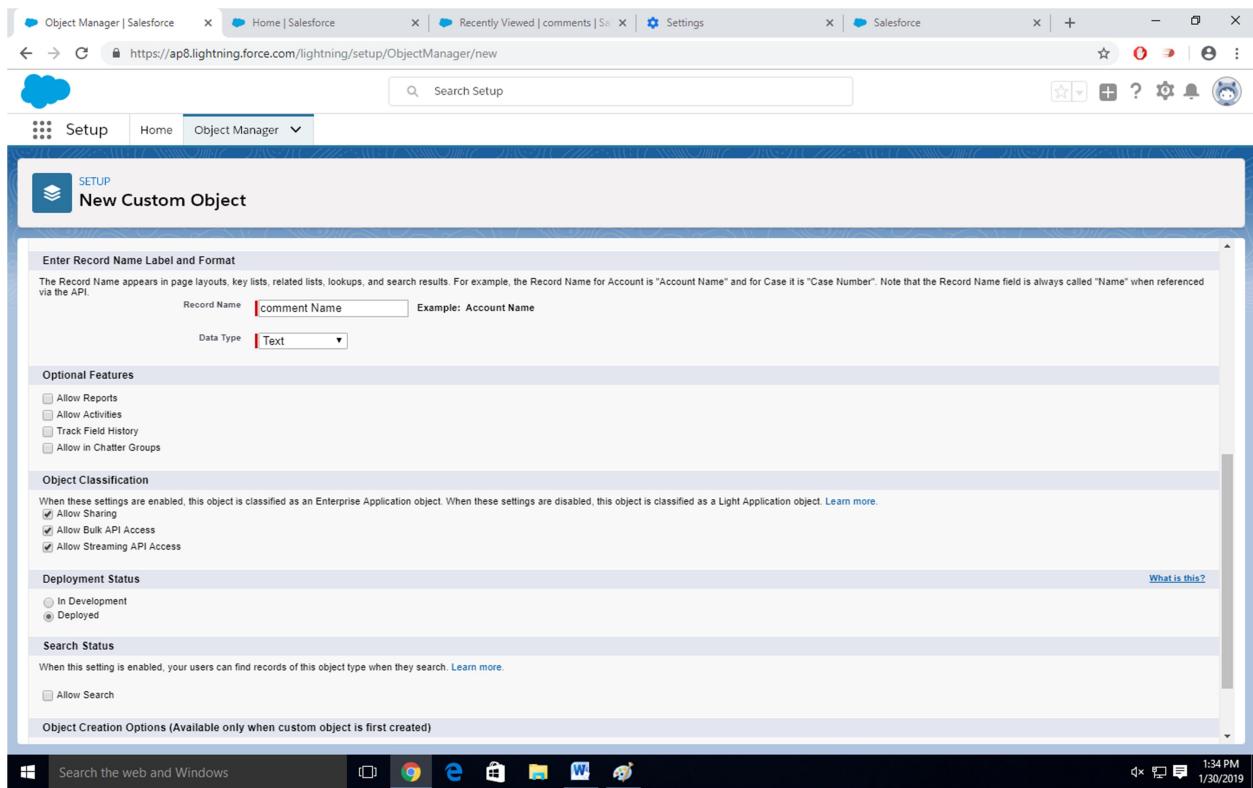


Give Record Name as –comment name

Data type- text

Select Allow Reports Check Box

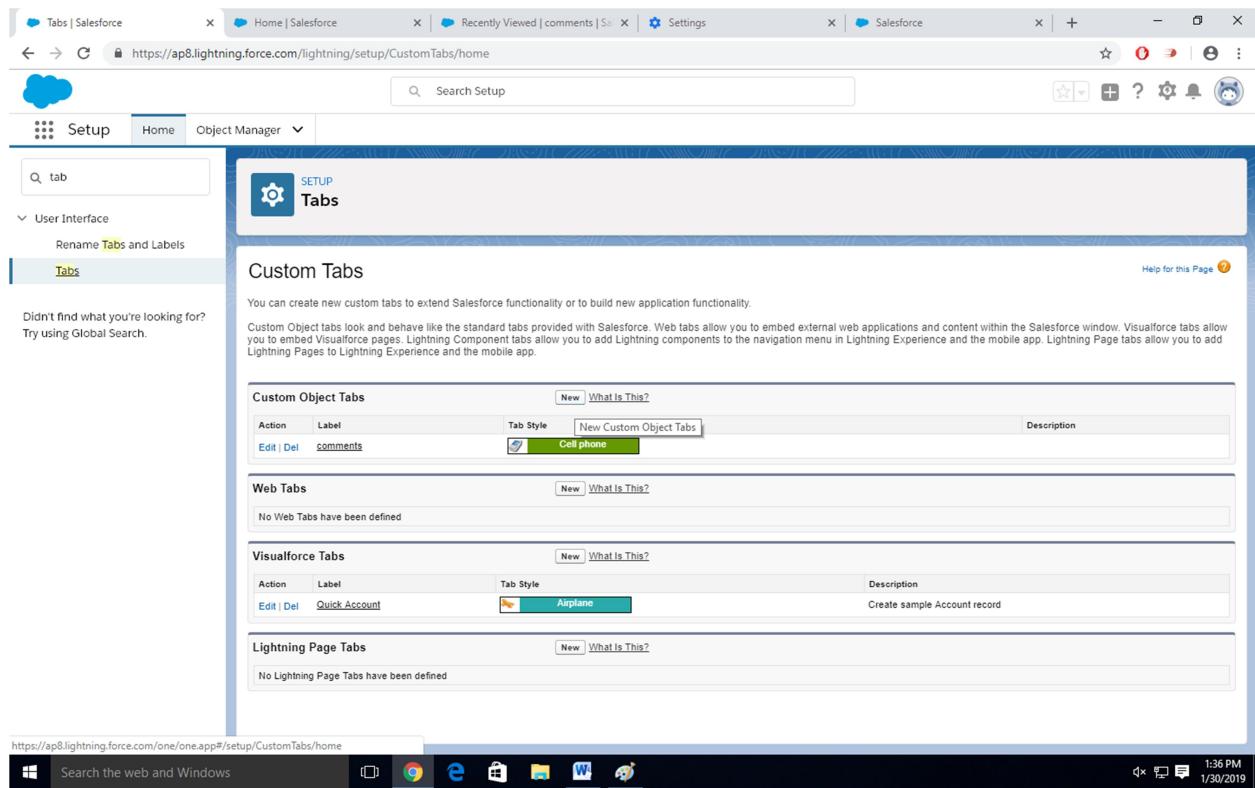
Click on Save



Step-5

Click on Home-Search Tabs in Quick search

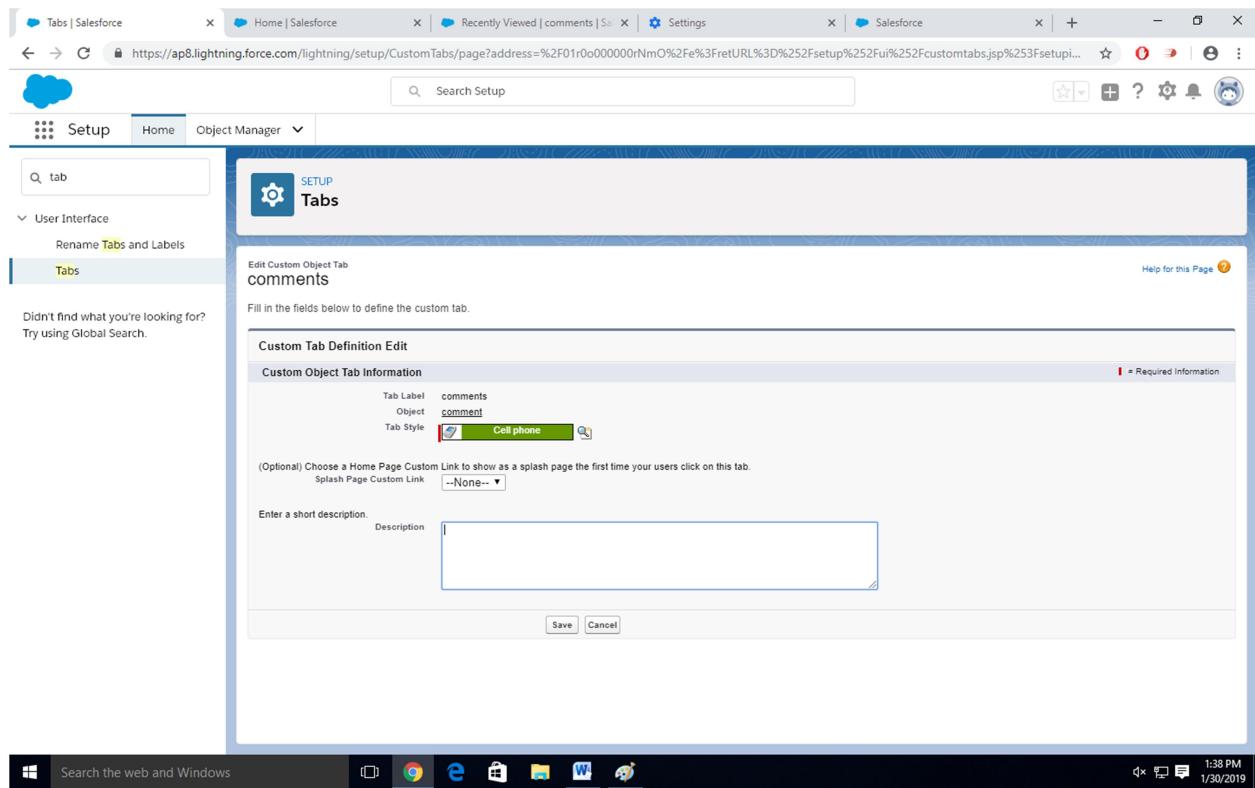
Select Custom Object-Click on New



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

Didnt find what you're looking for?
Try using Global Search.

APP NAME	DEVELOPER NAME	DESCRIPTION	LAST MODIFIED	APP TYPE	VISIBILITY
1 Analytics Studio	Insights		1/11/2019 1:34 AM	Classic	✓
2 App Launcher	AppLauncher	App Launcher tabs	1/11/2019 1:34 AM	Classic	✓
3 Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your industry.	1/11/2019 1:34 AM	Lightning	✓
4 comment box	comment_box		1/29/2019 11:41 PM	Lightning	✓
5 Community	Community	Salesforce CRM Communities	1/11/2019 1:34 AM	Classic	✓
6 Content	Content	Salesforce CRM Content	1/11/2019 1:34 AM	Classic	✓
7 Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	1/11/2019 1:34 AM	Lightning	✓
8 Marketing	Marketing	Best-in-class on-demand marketing automation	1/11/2019 1:34 AM	Classic	✓
9 Platform	Platform	The fundamental Lightning Platform	1/11/2019 1:34 AM	Classic	✓
10 Sales	Sales	The world's most popular sales force automation (SFA) solution	1/11/2019 1:34 AM	Classic	✓
11 Sales	LightningSales	Manage your sales process with accounts, leads, opportunities, and more	1/11/2019 1:34 AM	Lightning	✓
12 Sales Console	LightningSalesConsole	(Lightning Experience) Lets sales reps work with multiple records on...	1/11/2019 1:34 AM	Lightning	✓
13 Salesforce Chatter	Chatter	The Salesforce Chatter social network, including profiles and feeds	1/11/2019 1:34 AM	Classic	✓
14 Service	Service	Manage customer service with accounts, contacts, cases, and more	1/11/2019 1:34 AM	Classic	✓
15 Service Console	LightningService	(Lightning Experience) Lets support agents work with multiple records a...	1/11/2019 1:34 AM	Lightning	✓
16 Site.com	Sites	Build pixel-perfect, data-rich websites using the drag-and-drop Site.com...	1/11/2019 1:34 AM	Classic	✓

Enter name to app name

Give your Lightning app a name and description. Upload an image and choose the highlight color for its navigation bar.

App Details	App Branding
* App Name <input type="text" value="abc"/> Complete this field.	Image <input type="button" value="Upload"/>
* Developer Name <input type="text" value="Enter a developer name..."/>	Primary Color Hex Value <input type="color" value="#0070D2"/> #0070D2
Description <input type="text" value="Enter a description..."/>	Org Theme Options <input type="checkbox"/> Use the app's image and color instead of the org's custom theme

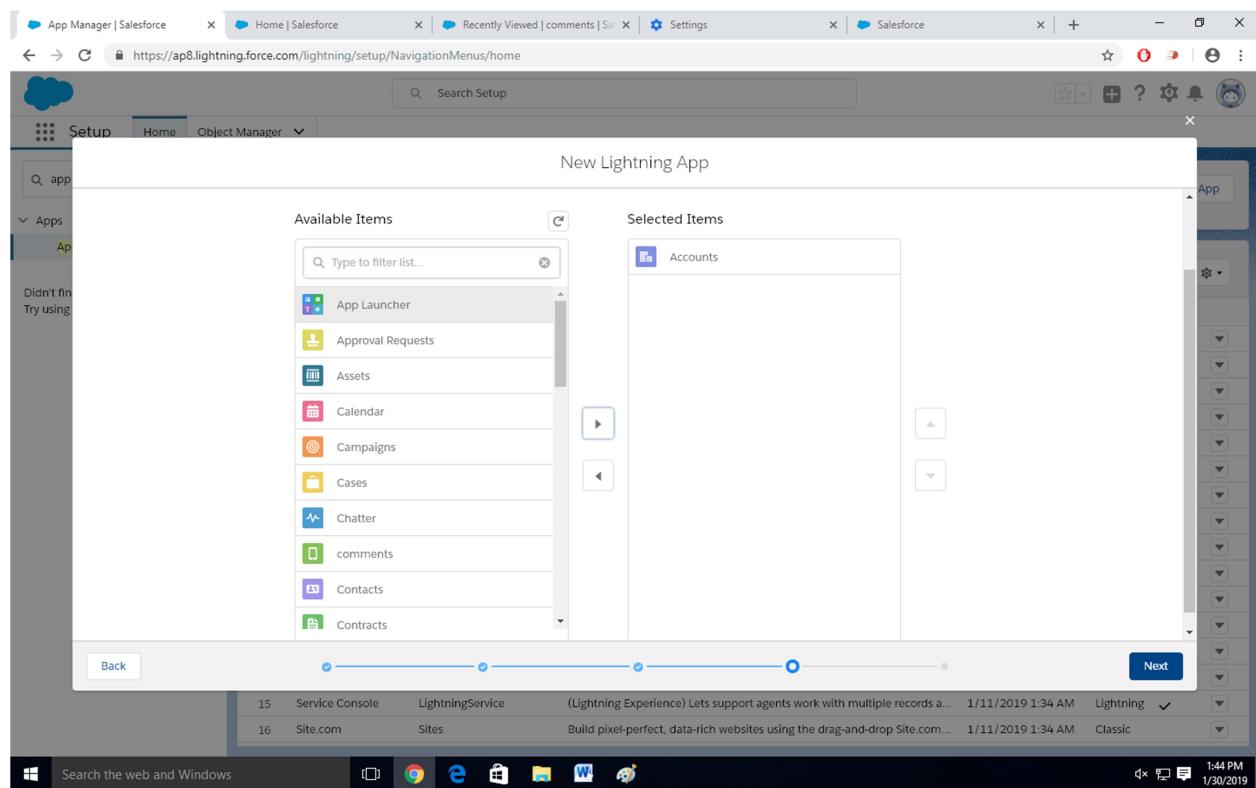
App Launcher Preview

Next

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.

User Profiles

Choose the user profiles that can access this app.

Available Profiles

X

Selected Profiles

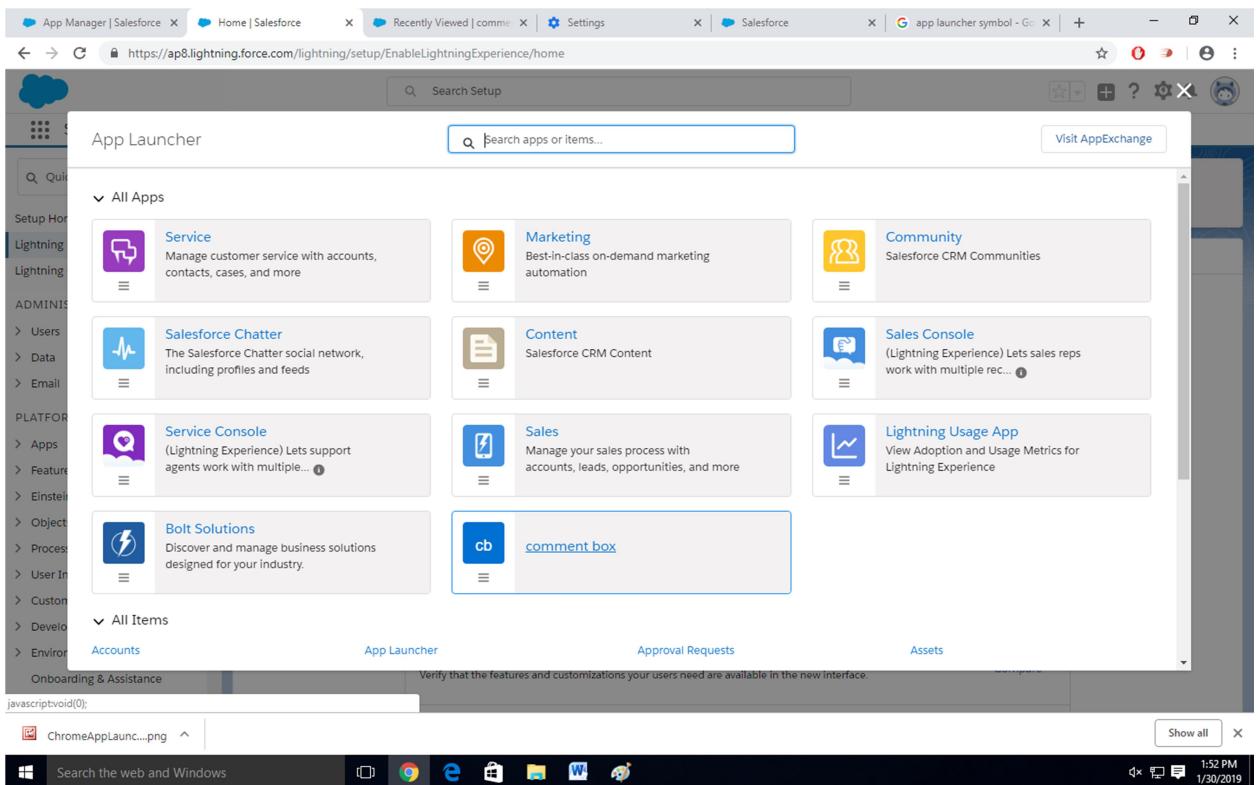
System Administrator

X

Save & Finish

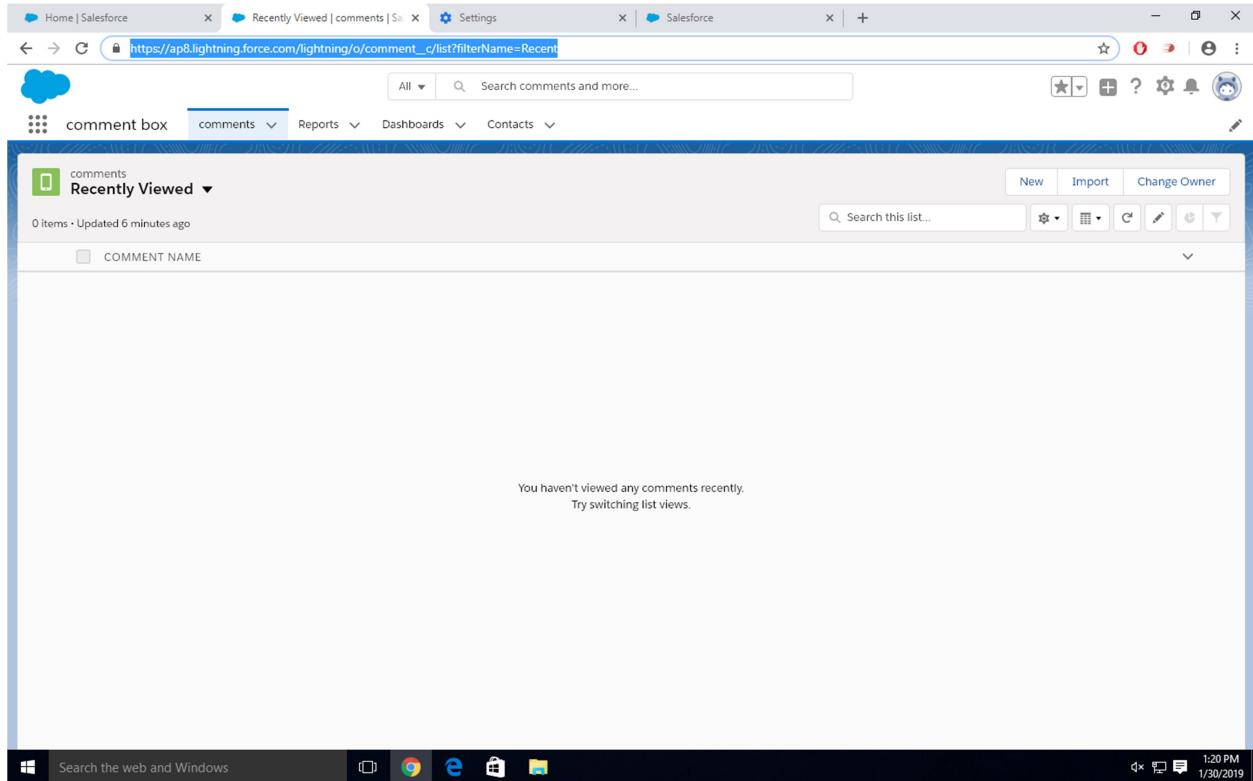
Step-9

Click on App Launcher  Symbol and Select Comment Box App



Step-11

Tour the app



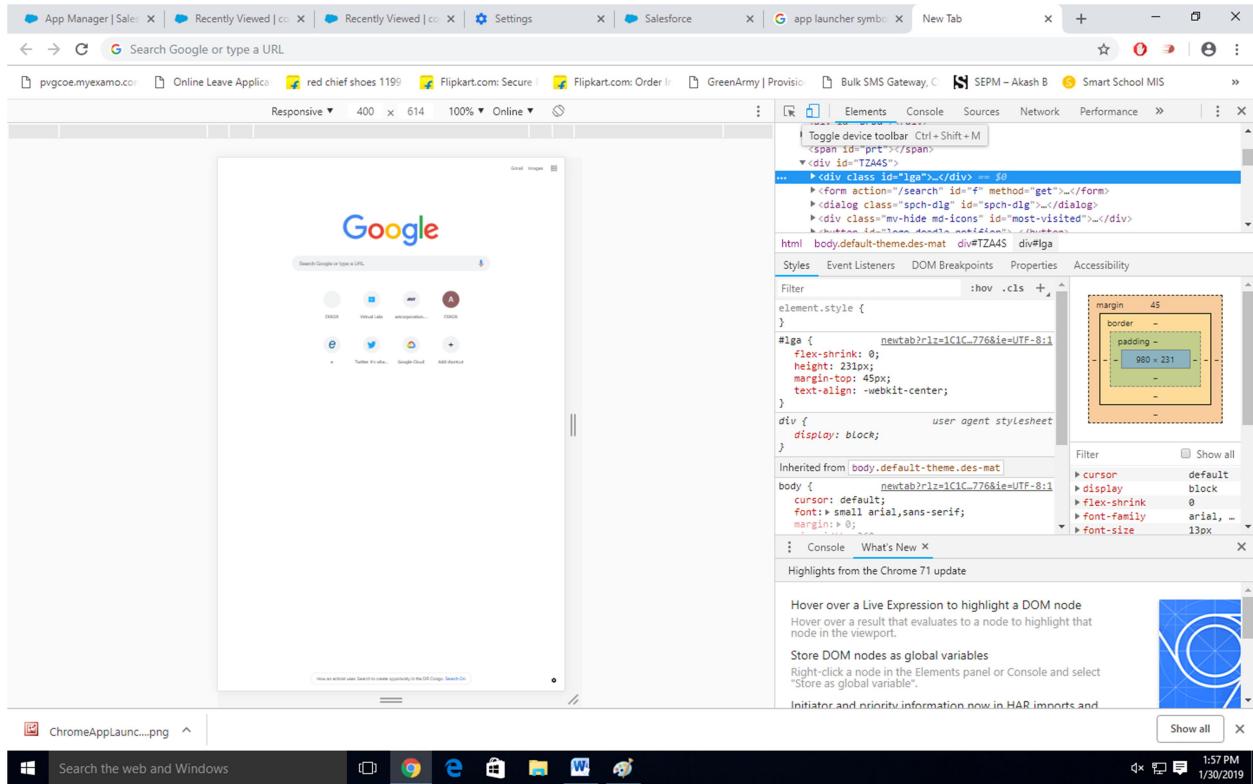
Step-12

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

