

**Dr. D. Y. Patil Pratishthan's**

**DR. D. Y. PATIL INSTITUTE OF ENGINEERING, MANAGEMENT &  
RESEARCH**

**Approved by A.I.C.T.E, New Delhi , Maharashtra State Government, Affiliated to Savitribai Phule Pune University  
Sector No. 29, PCNTDA , Nigidi Pradhikaran, Akurdi, Pune 411044. Phone: 020-27654470, Fax: 020-27656566  
Website :[www.dypiemr.ac.in](http://www.dypiemr.ac.in) Email : [principal.dypiemr@gmail.com](mailto:principal.dypiemr@gmail.com)**

## **Department of Computer Engineering**

# **LAB MANUAL Laboratory Practice II (TE 2019 pattern) Semester II**

**Prepared by:  
Mrs. Ketaki Bhoyar  
Mrs. Swati Rajput  
Mr. Prateek Meshram**



# Laboratory Practice II

Course Code	Course Name	Teaching Scheme (Hrs./ Week)	Credits
310258	Laboratory Practice II	4	2

## Course Objectives:

- To learn and apply various search strategies for AI
- To Formalize and implement constraints in search problems
- To understand the concepts of Information Security / Augmented and Virtual Reality/Cloud Computing/Software Modeling and Architectures

## Course Outcomes:

On completion of the course, learner will be able to—

- CO1: Design system using different informed search / uninformed search or heuristic approaches
- CO2: Apply basic principles of AI in solutions that require problem solving, inference, perception, knowledge representation, and learning
- CO3: Design and develop an expert system
- CO4: Use tools and techniques in the area of Cloud Computing OR Use tools and techniques in the area Software Modeling and Architectures
- CO5: Use the knowledge of Cloud Computing for problem solving OR Use the knowledge of Software Modeling and Architectures for problem solving
- CO6: Apply the concepts Cloud Computing to design and develop applications OR Apply the concepts Software Modeling and Architectures to design and develop applications

**Operating System recommended:** 64-bit Windows OS and Linux

## Programming tools recommended:

**Cloud Computing:** -

**Software Modeling and Architectures:** Front end:HTML5, Bootstrap, jQuery, JS etc. Backend: MySQL/MongoDB/NodeJS

# Table of Contents

Sr. No	Title of Experiment	CO Mapping	Page No
<b>Part I : Artificial Intelligence</b>			
<b>Group A</b>			
<b>All assignments are compulsory</b>			
1	Implement depth first search algorithm and Breadth First Search algorithm, Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.	CO1	
2	Implement A star Algorithm for any game search problem.	CO1	
3	Implement Greedy search algorithm for any of the following application: I. Selection Sort II. Minimum Spanning Tree III. Single-Source Shortest Path Problem IV. Job Scheduling Problem V. Prim's Minimal Spanning Tree Algorithm VI. Kruskal's Minimal Spanning Tree Algorithm VII. Dijkstra's Minimal Spanning Tree Algorithm logical operators etc.).	CO1, CO2	
<b>Group B</b>			
4	Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem.	CO2	
5	Develop an elementary chatbot for any suitable customer interaction application.	CO2, CO3	
<b>Group C</b>			
6	Implement any one of the following Expert System I. Information management II. Hospitals and medical facilities III. Help desks management IV. Employee performance evaluation V. Stock market trading VI. Airline scheduling and cargo schedules	CO2, CO3	
<b>Part II</b>			
<b>Elective II – Cloud Computing</b>			
<b>(All assignments are compulsory)</b>			
1	Case study on Microsoft azure to learn about Microsoft Azure is a cloud computing platform and infrastructure, created by Microsoft, for building, deploying and managing applications and services through a global network of Microsoft-managed data centers. <b>OR</b> Case study on Amazon EC2 and learn about Amazon EC2 web services develop a recursive algorithm for searching all the vertices of a graph or tree data structure.	CO4	
2	Installation and configure Google App Engine. OR	CO5	

3	Creating an Application in SalesForce.com using Apex programming Language.	CO5, CO6	
4	Design and develop custom Application (Mini Project) using Salesforce Cloud.	CO6	
5	<p style="text-align: center;"><b>Mini-Project</b></p> <p>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 file on/from cloud in encrypted form.</p>	CO6	

### Elective II – Software Modelling and Design

**(Problem statement 1, 2 , 3 or 4, Problem statement 5 and 6 are mandatory )**

1	Consider a library, where a member can perform two operations: issue book and return it. A book is issued to a member only after verifying his credentials. Develop a use case diagram for the given library system by identifying the actors and use cases and associate the use cases with the actors by drawing a use case diagram. Use UML tool.	CO4	
2	Consider online shopping system. Perform the following tasks and draw the class diagram using UML tool. Represent the individual classes, and objects Add methods Represent relationships and other classifiers like interfaces.	CO4,CO5	
3	Consider the online shopping system in the assignment 2. Draw the sequence diagram using UML tool to show message exchanges.	CO5	
4	<p>Consider your neighboring travel agent from whom you can purchase flight tickets. To book a ticket you need to provide details about your journey i.e., on which date and at what time you would like to travel. You also need to provide your address. The agency has recently been modernized. So, you can pay either by cash or by card. You can also cancel a booked ticket later if you decide to change your plan. In that case you need to book a new ticket again. Your agent also allows you to book a hotel along with flight ticket. While cancelling a flight ticket you can also cancel hotel booking. Appropriate refund as per policy is made in case of cancellation.</p> <p>Perform the following tasks and draw the use case diagram using UML tool.</p> <ul style="list-style-type: none"> <li>a. Identify the use cases from a given non-trivial problem statement.</li> <li>b. Identify the primary and secondary actors for a system.</li> <li>c. Use to generalization of use cases and «include» stereotypes to prevent redundancy in the coding phase</li> </ul>	CO4,CO5	
5	<p style="text-align: center;"><b>Mini-Project</b></p> <p>Select a moderately complex system and narrate concise requirement Specification for the same. Design the system indicating system elements organizations using applicable architectural styles and design patterns with the help of a detailed Class diagram depicting logical architecture. Specify and document the architecture and design pattern with the help of templates. Implement the system features and judge the benefits of the design patterns accommodated.</p>	CO6	

# Part I

# Artificial Intelligence

# Group A

<b>Lab Assignment No.</b>	A1
<b>Title</b>	Implement depth first search algorithm and Breadth First Search algorithm. Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.
<b>Roll No.</b>	
<b>Class</b>	TE
<b>Date of Completion</b>	
<b>Subject</b>	Laboratory Practice II
<b>Assessment Marks</b>	
<b>Assessor's Sign</b>	

## ASSIGNMENT No: A1

**Title:** Implement depth first search algorithm and Breadth First Search algorithm.

**Problem Statement:** Implement depth first search algorithm and Breadth First Search algorithm. Use an undirected graph and develop a recursive algorithm for searching all the vertices of a graph or tree data structure.

### Objective:

- Understand and implement BFS/DFS algorithm.
- Analyze time complexity of the search algorithm.

### Outcomes:

- Ability to choose an appropriate problem-solving method and knowledge representation technique.

### Software Required:

- Python

### Theory:

- Depth First Search:

Depth first Search or Depth first traversal is a recursive algorithm for searching all the vertices of a graph or tree data structure. Traversal means visiting all the nodes of a graph.

➤ Depth First Search Algorithm:

A standard DFS implementation puts each vertex of the graph into one of two categories:

- Visited
- Not Visited

The purpose of the algorithm is to mark each vertex as visited while avoiding cycles.

The DFS algorithm works as follows:

1. Start by putting any one of the graph's vertices on top of a stack.
2. Take the top item of the stack and add it to the visited list.
3. Create a list of that vertex's adjacent nodes. Add the ones which aren't in the visited list to the top of the stack.
4. Keep repeating steps 2 and 3 until the stack is empty.

➤ Depth First Search Example:

Let's see how the Depth First Search algorithm works with an example. We use an undirected graph with 5 vertices.

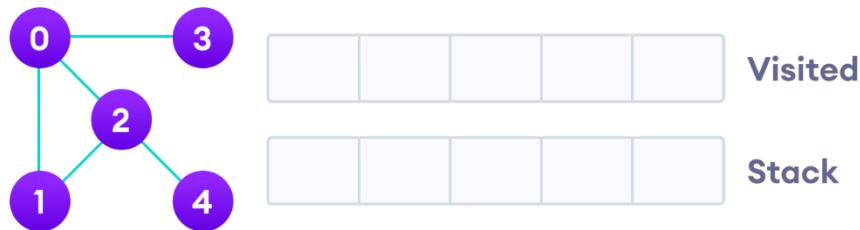


Figure 1: Undirected graph with 5 vertices

We start from vertex 0, the DFS algorithm starts by putting it in the Visited list and putting all its adjacent vertices in the stack.

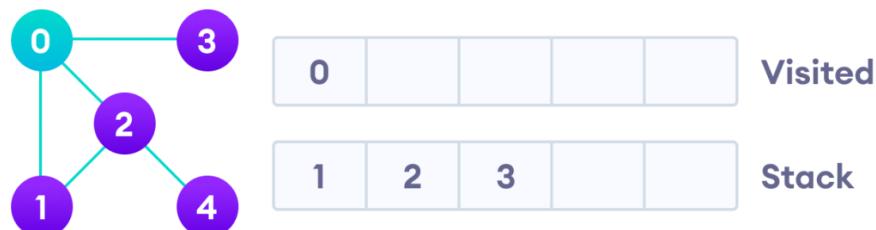


Figure 2: Visit the element and put it in the visited list

Next, we visit the element at the top of stack i.e. 1 and go to its adjacent nodes. Since 0 has already been visited, we visit 2 instead.

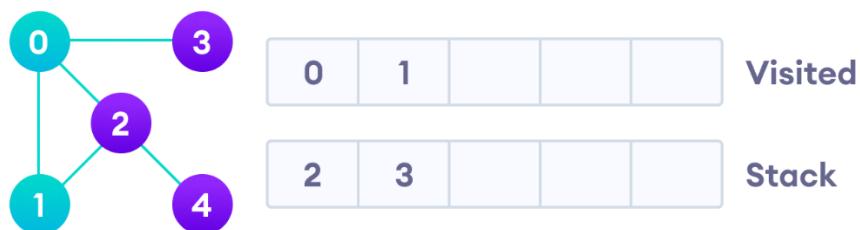


Figure 3: Visit the element at the top of stack

Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the top of the stack and visit it.

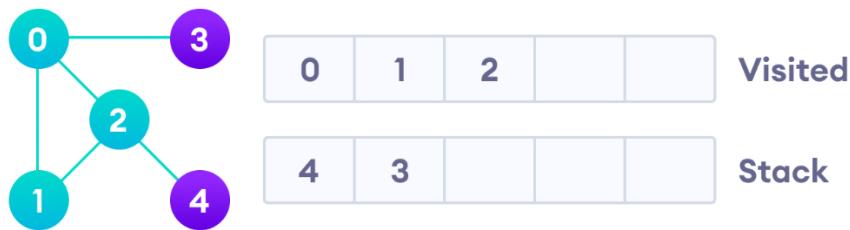


Figure 4: Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the top of the stack and visit it.

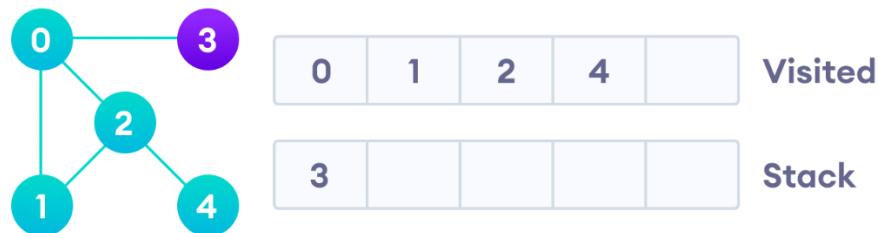


Figure 5: Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the top of the stack and visit it.

After we visit the last element 3, it doesn't have any unvisited adjacent nodes, so we have completed the Depth First Traversal of the graph.

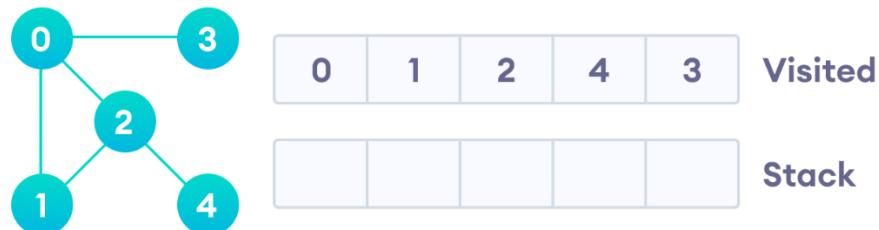


Figure 6: After we visit the last element 3, it doesn't have any unvisited adjacent nodes, so we have completed the Depth First Traversal of the graph.

➤ DFS Pseudocode (recursive implementation):

The pseudocode for DFS is shown below. In the init() function, notice that we run the DFS function on every node. This is because the graph might have two different disconnected parts so to make sure that we cover every vertex, we can also run the DFS algorithm on every node.

DFS(G, u)

```

u.visited = true
for each v ∈ G.Adj[u]
    if v.visited == false
        DFS(G,v)
    
```

```

init() {
    For each u ∈ G
        u.visited = false
    For each u ∈ G
        DFS(G, u)
}
    
```

➤ Complexity of Depth First Search:

- Time complexity of the DFS algorithm:  $O(V + E)$ , where V is the number of nodes and E is the number of edges.
- Space complexity of the algorithm:  $O(V)$ .

➤ Application of DFS Algorithm:

- For finding the path
- To test if the graph is bipartite
- For finding the strongly connected components of a graph
- For detecting cycles in a graph

● Breadth First Search:

Traversal means visiting all the nodes of a graph. Breadth First Traversal or Breadth First Search is a recursive algorithm for searching all the vertices of a graph or tree data structure.

➤ Breadth First Search Algorithm:

A standard BFS implementation puts each vertex of the graph into one of two categories:

- Visited
- Not Visited

The purpose of the algorithm is to mark each vertex as visited while avoiding cycles.

The BFS algorithm works as follows:

1. Start by putting any one of the graph's vertices at the back of a queue.
2. Take the front item of the queue and add it to the visited list.
3. Create a list of that vertex's adjacent nodes. Add the ones which aren't in the visited list to the back of the queue.
4. Keep repeating steps 2 and 3 until the queue is empty.

➤ Breadth First Search Example:

Let's see how the Breadth First Search algorithm works with an example. We use an undirected graph with 5 vertices.



Figure 1: Undirected graph with 5 vertices

We start from vertex 0, the BFS algorithm starts by putting it in the Visited list and putting all its adjacent vertices in the queue.



Figure 2: Visit start vertex and add its adjacent vertices to queue

Next, we visit the element at the front of queue i.e. 1 and go to its adjacent nodes. Since 0 has already been visited, we visit 2 instead.

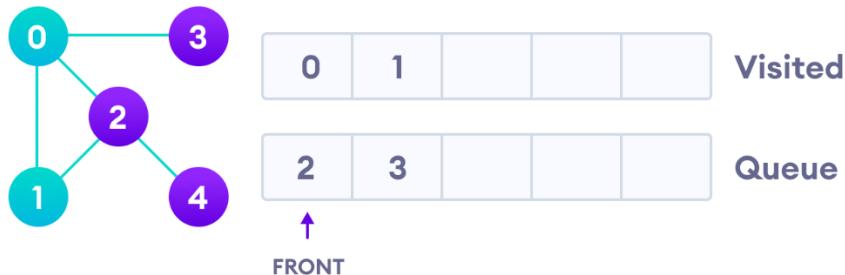


Figure 3: Visit the first neighbour of start node 0, which is 1

Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the back of the queue and visit 3, which is at the front of the queue.

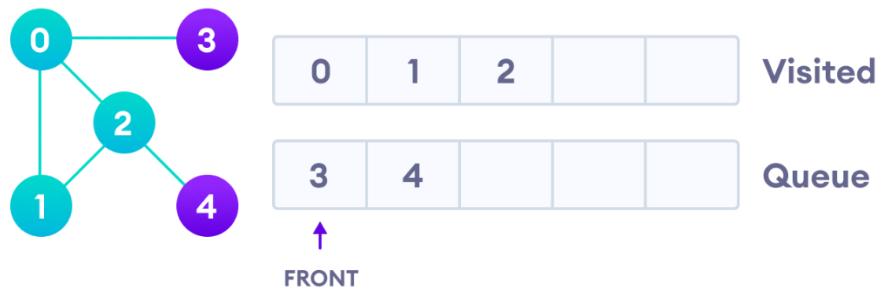


Figure 4: Visit 2 which was added to queue earlier to add its neighbors

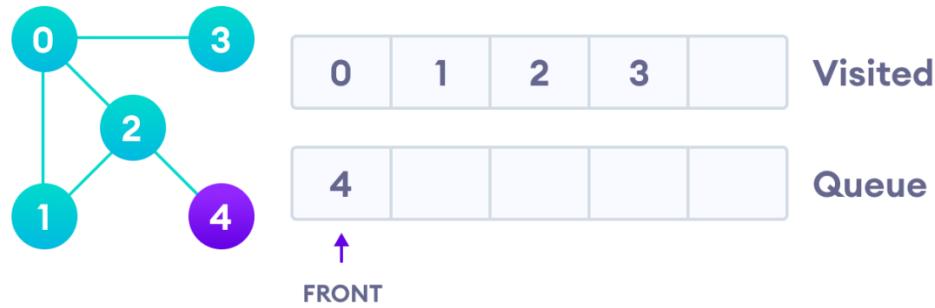


Figure 5: 4 remains in the queue

Only 4 remains in the queue since the only adjacent node of 3 i.e. 0 is already visited. We visit it.

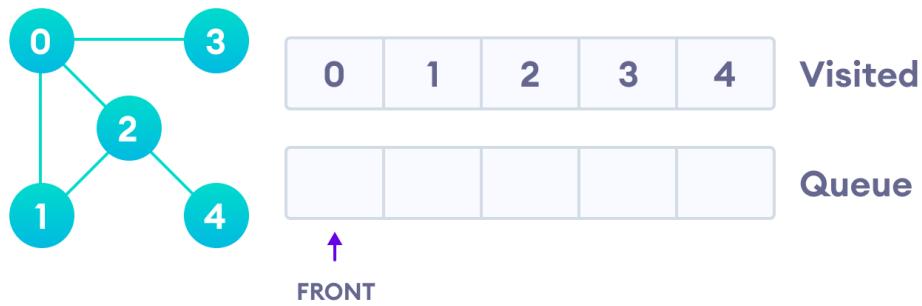


Figure 6: Visit last remaining item in the queue to check if it has unvisited neighbors

Since the queue is empty, we have completed the Breadth First Traversal of the graph.

➤ BFS Pseudocode:

```
create a queue Q  
mark v as visited and put v into Q  
while Q is non-empty  
    remove the head u of Q  
    mark and enqueue all (unvisited) neighbours of u
```

➤ Complexity of Breadth First Search:

- Time complexity of the BFS algorithm:  $O(V + E)$ ,  
where V is the number of nodes and E is the number of edges.
- Space complexity of the BFS algorithm:  $O(V)$ .

➤ Application of DFS Algorithm:

- To build index by search index
- For GPS navigation
- Path finding algorithms
- In Ford-Fulkerson algorithm to find maximum flow in a network
- Cycle detection in an undirected graph
- In minimum spanning tree

**Conclusion:** Implemented depth first search algorithm and Breadth First Search algorithm for an undirected graph using recursive algorithm for searching all the vertices of a graph or tree data structure.

<b>Lab Assignment No.</b>	A2
<b>Title</b>	Implement A star (A*) Algorithm for any game search problem.
<b>Roll No.</b>	
<b>Class</b>	TE
<b>Date of Completion</b>	
<b>Subject</b>	Laboratory Practice II
<b>Assessment Marks</b>	
<b>Assessor's Sign</b>	

## ASSIGNMENT No: A2

**Title:** Implement A star (A\*) Algorithm for any game search problem.

**Problem Statement:** Implement A star (A\*) Algorithm for any game search problem.

### **Objective:**

- Understand the working of informed search algorithm
- Implement A\* search algorithm

### **Outcome:**

- Ability to choose an appropriate problem solving method and knowledge representation technique

### **Software Required:**

- Python

### **Theory:**

An informed search strategy-one that uses problem-specific knowledge-can find solutions more efficiently.A key component of these algorithms is a heuristic function  $h(n)$

$h(n)$  = estimated cost of the cheapest path from node n to a goal node.

Admissible /heuristic never over estimated i.e.  $h(n) \leq$  Actual cost. For example, Distance between two nodes(cities)=> straight line distance and for 8-puzzel problem- Admissible heuristic can be number of misplaced tiles  $h(n)=8$ .

- **A\* Search technique:**

It is informed search technique. It uses additional information beyond problem formulation and tree. Search is based on Evaluation function  $f(n)$ . Evaluation function is based on both heuristic function  $h(n)$  and  $g(n)$ .

$$f(n)=g(n) + h(n)$$

It uses two queues for its implementation: open, close Queue. Open queue is a priority queue which is arranged in ascending order of  $f(n)$

- **Algorithm:**

1. Create a single member queue comprising of Root node
2. If FIRST member of queue is goal then goto step 5
3. If first member of queue is not goal then remove it from queue and add to close queue.
4. Consider its children if any, and add them to queue in ascending order of evaluation function  $f(n)$ .
5. If queue is not empty then goto step 2.

6. If queue is empty then goto step 6
7. Print ‘success’ and stop
8. Print ‘failure’ and stop.

- Performance Comparison:

- Completeness: yes
- Optimality: yes

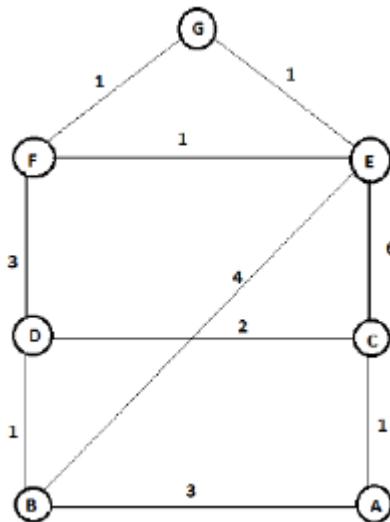
- Limitation:

- • It generate same node again and again
- • Large Memory is required

- Observation:

Although A\* generate many nodes it never uses those nodes for which  $f(n) > c^*$  where  $c^*$  is optimum cost.

- Consider an example as below



OPEN/FRINGE	CLOSE
[A]	[ ]
[C,B]	[A]
[D,B,E,A]	[A,C]
[F,E,B,C,A]	[A,C,D]
[G,E,B,C,A,D]	[A,C,D,F]

SUCCESS  
Node A:

$$f(B) = g(B) + h(B) = 3 + 5 = 8$$

$$f(C) = g(C) + h(C) = 1 + 6 = 7$$

Node C:

$$\begin{aligned}f(A) &= g(A) + h(A) = 2+7=10 \\f(D) &= g(D) + h(D) = 3+4=7 \\f(E) &= g(E) + h(E) = 7+1=8\end{aligned}$$

Node D:

$$\begin{aligned}f(F) &= g(F) + h(F) = 6+1=7 \\f(C) &= g(C) + h(C) = 5+6=11 \\f(B) &= g(B) + h(B) = 4+5=9\end{aligned}$$

Node F:

$$\begin{aligned}f(E) &= g(E) + h(E) = 7+1=8 \\f(D) &= g(D) + h(D) = 9+4=13 \\f(G) &= g(G) + h(G) = 7+0=7\end{aligned}$$

Final path: A → C → D → F → G  
Total cost = 7

**Conclusion:** A good example of heuristic search is A\* search algorithm. The performance of such heuristic search algorithm depends upon the quality of heuristic function. A\* algorithm is executed and the path with minimum cost from source node to destination node is calculated.

<b>Lab Assignment No.</b>	A3
<b>Title</b>	Implement Greedy search algorithm for any of the following application: <ul style="list-style-type: none"><li>• Selection Sort</li><li>• Minimum Spanning Tree</li><li>• Single-Source Shortest Path Problem</li><li>• Job Scheduling Problem</li><li>• Prim's Minimal Spanning Tree Algorithm</li><li>• Kruskal's Minimal Spanning Tree Algorithm</li><li>• Dijkstra's Minimal Spanning Tree Algorithm</li></ul>
<b>Roll No.</b>	
<b>Class</b>	TE
<b>Date of Completion</b>	
<b>Subject</b>	Laboratory Practice II
<b>Assessment Marks</b>	
<b>Assessor's Sign</b>	

## ASSIGNMENT No: A3

**Title:** Implement Greedy search algorithm for any of the following application:

- Selection Sort
- Minimum Spanning Tree
- Single-Source Shortest Path Problem
- Job Scheduling Problem
- Prim's Minimal Spanning Tree Algorithm
- Kruskal's Minimal Spanning Tree Algorithm
- Dijkstra's Minimal Spanning Tree Algorithm

**Problem Statement:** Implement Greedy search algorithm for any of the following application:

- Prim's Minimal Spanning Tree Algorithm
- Kruskal's Minimal Spanning Tree Algorithm

### Objective:

- Prim's Minimal Spanning Tree Algorithm
- Kruskal's Minimal Spanning Tree Algorithm

### Outcome:

- Ability to choose an appropriate problem solving method and knowledge representation technique

### Software Required:

- Python

### Theory:

- Kruskal's Minimum Spanning Tree Algorithm:
  - What is a Spanning Tree?

A Spanning tree is a subset to a connected graph G, where all the edges are connected, i.e, we can traverse to any edge from a particular edge with or without intermediates.

Also, a spanning tree must not have any cycle in it. Thus we can say that if there are n vertices in a connected graph then the no. of edges that a spanning tree may have is n-1.

- What is Minimum Spanning Tree?

Given a connected and undirected graph, a spanning tree of that graph is a subgraph that is a tree and connects all the vertices together. A single graph can have many different spanning trees. A minimum spanning tree (MST) or minimum weight spanning tree for a weighted, connected, undirected graph is a spanning tree with a weight less than or equal to the weight of every other spanning tree. The weight of a spanning tree is the sum of weights given to each edge of the spanning tree.

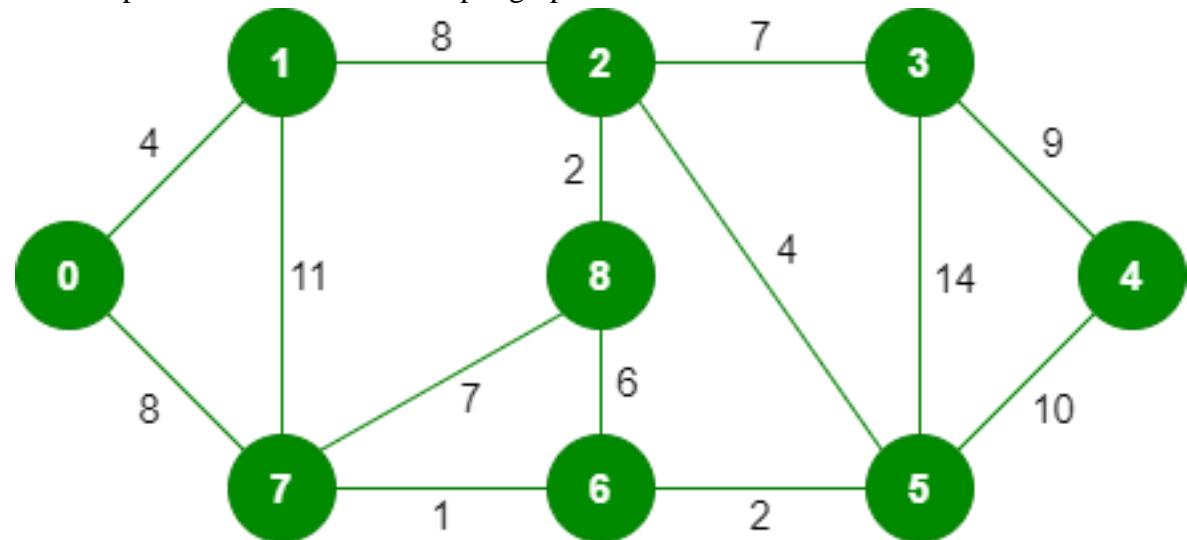
How many edges does a minimum spanning tree has?

A minimum spanning tree has  $(V - 1)$  edges where  $V$  is the number of vertices in the given graph.

What are the applications of the Minimum Spanning Tree?

See this for applications of MST.

- Steps for finding MST using Kruskal's algorithm:
  1. Sort all the edges in non-decreasing order of their weight.
  2. Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include this edge. Else, discard it.
  3. Repeat step#2 until there are  $(V-1)$  edges in the spanning tree.
- The algorithm is a Greedy Algorithm. The Greedy Choice is to pick the smallest weight edge that does not cause a cycle in the MST constructed so far. Let us understand it with an example: Consider the below input graph.



The graph contains 9 vertices and 14 edges. So, the minimum spanning tree formed will be having  $(9 - 1) = 8$  edges.

After sorting:

Weight	Src	Dest
1	7	6
2	8	2

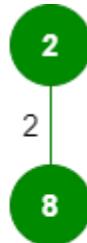
2	6	5
4	0	1
4	2	5
6	8	6
7	2	3
7	7	8
8	0	7
8	1	2
9	3	4
10	5	4
11	1	7
14	3	5

Now pick all edges one by one from the sorted list of edges

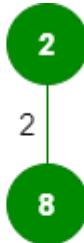
1. Pick edge 7-6: No cycle is formed, include it.



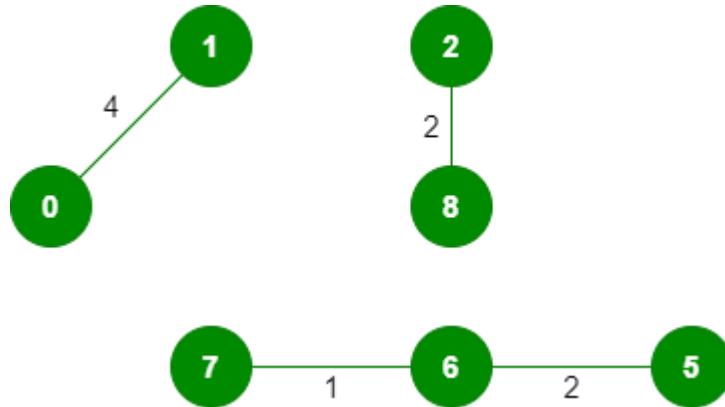
2. Pick edge 8-2: No cycle is formed, include it.



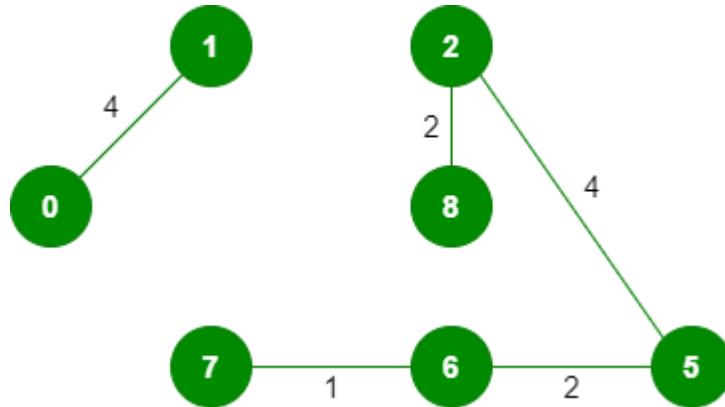
3. Pick edge 6-5: No cycle is formed, include it.



4. Pick edge 0-1: No cycle is formed, include it.

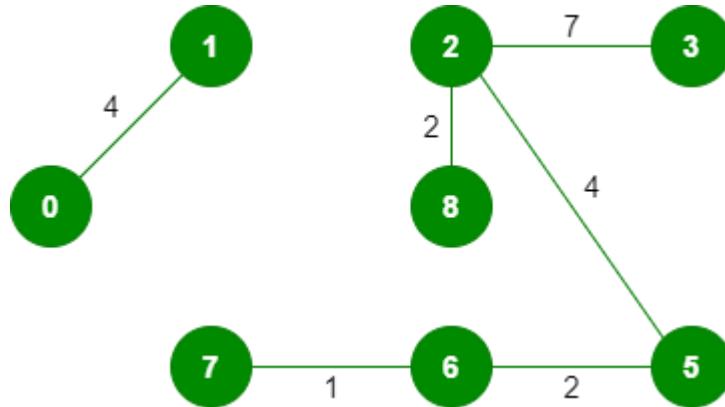


5. Pick edge 2-5: No cycle is formed, include it.



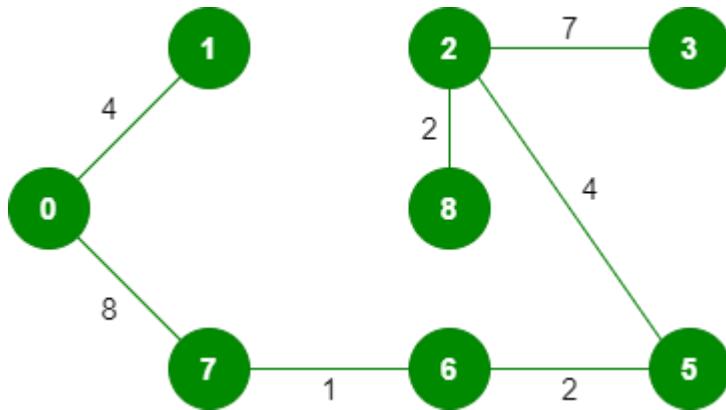
6. Pick edge 8-6: Since including this edge results in the cycle, discard it.

7. Pick edge 2-3: No cycle is formed, include it.



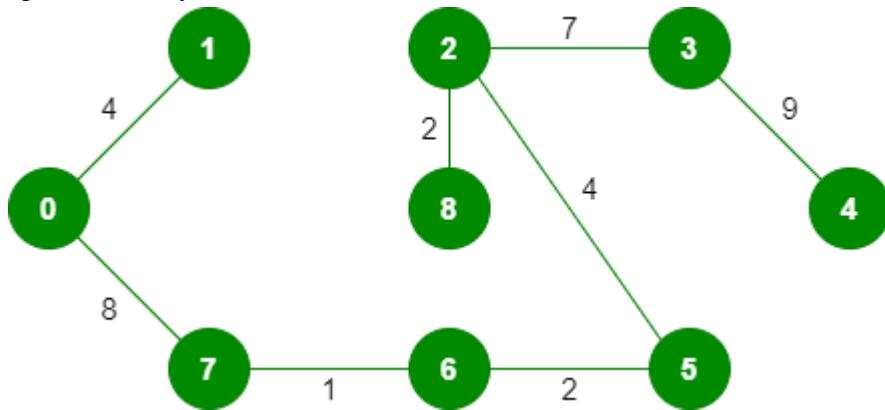
8. Pick edge 7-8: Since including this edge results in the cycle, discard it.

9. Pick edge 0-7: No cycle is formed, include it.



10. Pick edge 1-2: Since including this edge results in the cycle, discard it.

11. Pick edge 3-4: No cycle is formed, include it.



Since the number of edges included equals  $(V - 1)$ , the algorithm stops here.

- Prim's Minimum Spanning Tree Algorithm:

We have discussed Kruskal's algorithm for Minimum Spanning Tree. Like Kruskal's algorithm, Prim's algorithm is also a Greedy algorithm. It starts with an empty spanning tree. The idea is to maintain two sets of vertices. The first set contains the vertices already included in the MST, the other set contains the vertices not yet included. At every step, it considers all the edges that connect the two sets and picks the minimum weight edge from these edges. After picking the edge, it moves the other endpoint of the edge to the set containing MST.

A group of edges that connects two sets of vertices in a graph is called cut in graph theory. So, at every step of Prim's algorithm, we find a cut (of two sets, one contains the vertices already included in MST and the other contains the rest of the vertices), pick the minimum weight edge from the cut, and include this vertex to MST Set (the set that contains already included vertices).

➤ How does Prim's Algorithm Work?

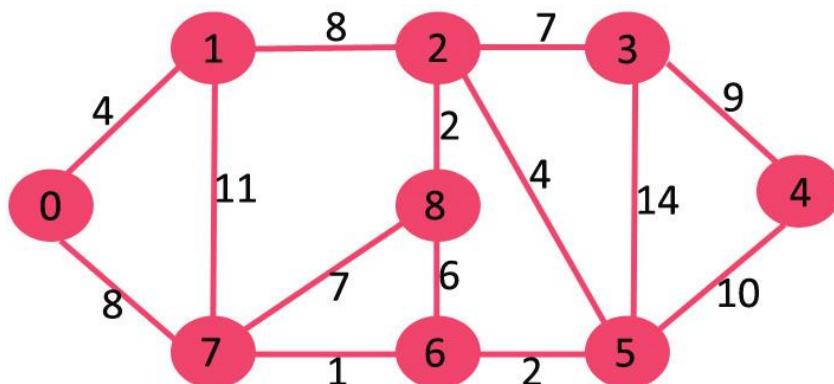
The idea behind Prim's algorithm is simple, a spanning tree means all vertices must be connected. So the two disjoint subsets (discussed above) of vertices must be connected to make a Spanning Tree. And they must be connected with the minimum weight edge to make it a Minimum Spanning Tree.

➤ Algorithm:

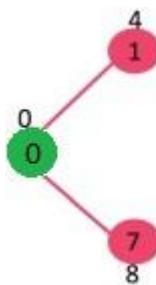
1. Create a set mstSet that keeps track of vertices already included in MST.
2. Assign a key value to all vertices in the input graph. Initialize all key values as INFINITE. Assign the key value as 0 for the first vertex so that it is picked first.
3. While mstSet doesn't include all vertices
  - a) Pick a vertex u which is not there in mstSet and has a minimum key value.
  - b) Include u to mstSet.
  - c) Update key value of all adjacent vertices of u. To update the key values, iterate through all adjacent vertices. For every adjacent vertex v, if the weight of edge u-v is less than the previous key value of v, update the key value as the weight of u-v

The idea of using key values is to pick the minimum weight edge from cut. The key values are used only for vertices that are not yet included in MST, the key value for these vertices indicates the minimum weight edges connecting them to the set of vertices included in MST.

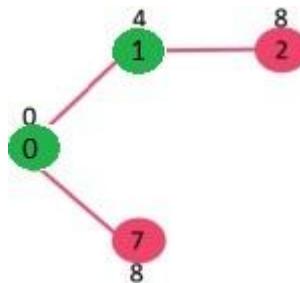
➤ Let us understand with the following example:



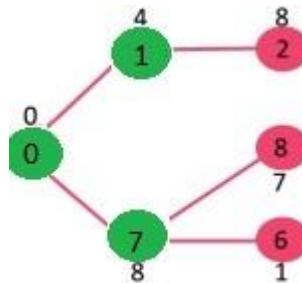
The set mstSet is initially empty and keys assigned to vertices are {0, INF, INF, INF, INF, INF, INF, INF} where INF indicates infinite. Now pick the vertex with the minimum key value. The vertex 0 is picked, include it in mstSet. So mstSet becomes {0}. After including to mstSet, update key values of adjacent vertices. Adjacent vertices of 0 are 1 and 7. The key values of 1 and 7 are updated as 4 and 8. Following subgraph shows vertices and their key values, only the vertices with finite key values are shown. The vertices included in MST are shown in green color.



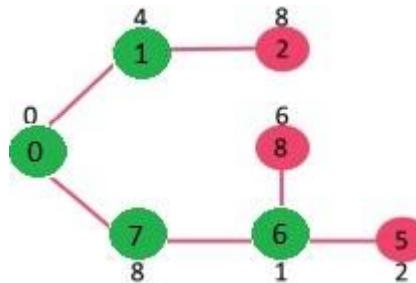
Pick the vertex with minimum key value and not already included in MST (not in mstSET). The vertex 1 is picked and added to mstSet. So mstSet now becomes {0, 1}. Update the key values of adjacent vertices of 1. The key value of vertex 2 becomes 8



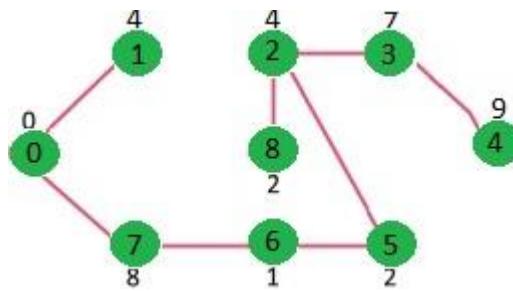
Pick the vertex with minimum key value and not already included in MST (not in mstSET). We can either pick vertex 7 or vertex 2, let vertex 7 is picked. So mstSet now becomes {0, 1, 7}. Update the key values of adjacent vertices of 7. The key value of vertex 6 and 8 becomes finite (1 and 7 respectively).



Pick the vertex with minimum key value and not already included in MST (not in mstSET). Vertex 6 is picked. So mstSet now becomes {0, 1, 7, 6}. Update the key values of adjacent vertices of 6. The key value of vertex 5 and 8 are updated.



We repeat the above steps until mstSet includes all vertices of given graph. Finally, we get the following graph.



**Conclusion:** Thus we have implemented Greedy search algorithm Prim's Minimal Spanning Tree Algorithm and Kruskal's Minimal Spanning Tree Algorithm.

# Group B

<b>Lab Assignment No.</b>	B4
<b>Title</b>	Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem.
<b>Roll No.</b>	
<b>Class</b>	TE
<b>Date of Completion</b>	
<b>Subject</b>	Laboratory Practice II
<b>Assessment Marks</b>	
<b>Assessor's Sign</b>	

## ASSIGNMENT No: B4

**Title:** Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem or a graph coloring problem.

**Problem Statement:** Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem.

### **Objective:**

- Understand and implement Constraint Satisfaction Problem using Branch and Bound for n-queens problem
- Understand and implement Constraint Satisfaction Problem using Backtracking for n-queens problem

### **Outcome:**

- Ability to choose an appropriate problem solving method and knowledge representation technique

### **Software Required:**

- Python

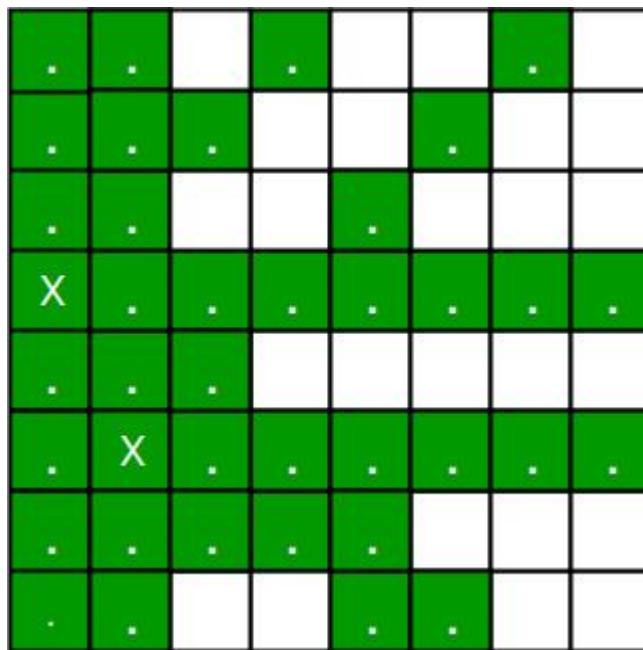
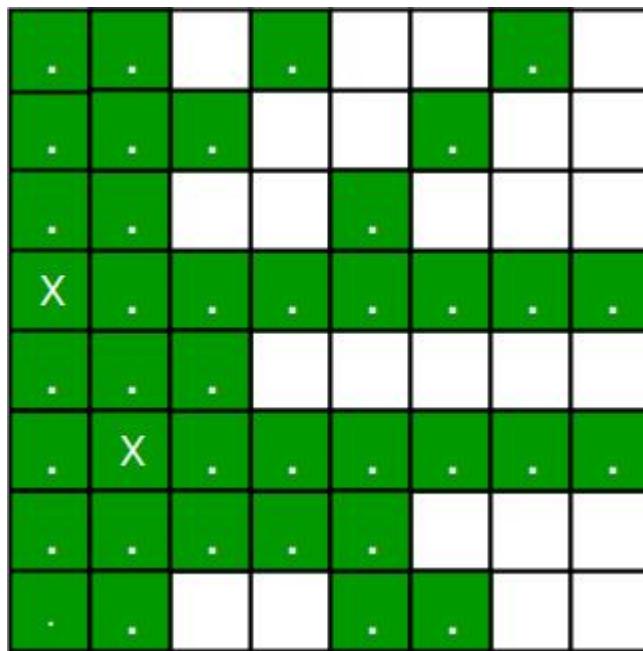
### **Theory:**

- N Queen Problem using Branch and Bound:

The N queens puzzle is the problem of placing N chess queens on an  $N \times N$  chessboard so that no two queens threaten each other. Thus, a solution requires that no two queens share the same row, column, or diagonal.

Backtracking Algorithm for N-Queen is already discussed here. In backtracking solution we backtrack when we hit a dead end. In Branch and Bound solution, after building a partial solution, we figure out that there is no point going any deeper as we are going to hit a dead end.

Let's begin by describing backtracking solution. "The idea is to place queens one by one in different columns, starting from the leftmost column. When we place a queen in a column, we check for clashes with already placed queens. In the current column, if we find a row for which there is no clash, we mark this row and column as part of the solution. If we do not find such a row due to clashes, then we backtrack and return false."



1. For the 1st Queen, there are total 8 possibilities as we can place 1st Queen in any row of first column. Let's place Queen 1 on row 3.
2. After placing 1st Queen, there are 7 possibilities left for the 2nd Queen. But wait, we don't really have 7 possibilities. We cannot place Queen 2 on rows 2, 3 or 4 as those cells are under attack from Queen 1. So, Queen 2 has only  $8 - 3 = 5$  valid positions left.
3. After picking a position for Queen 2, Queen 3 has even fewer options as most of the cells in its column are under attack from the first 2 Queens.

We need to figure out an efficient way of keeping track of which cells are under attack. In previous solution we kept an 8--by--8 Boolean matrix and update it each time we placed a queen, but that required linear time to update as we need to check for safe cells.

Basically, we have to ensure 4 things:

1. No two queens share a column.
2. No two queens share a row.
3. No two queens share a top-right to left-bottom diagonal.
4. No two queens share a top-left to bottom-right diagonal.

Number 1 is automatic because of the way we store the solution. For number 2, 3 and 4, we can perform updates in O(1) time. The idea is to keep three Boolean arrays that tell us which rows and which diagonals are occupied.

Lets do some pre-processing first. Let's create two N x N matrix one for / diagonal and other one for \ diagonal. Let's call them slashCode and backslashCode respectively. The trick is to fill them in such a way that two queens sharing a same /-diagonal will have the same value in matrix slashCode, and if they share same \ -diagonal, they will have the same value in backslashCode matrix.

For an N x N matrix, fill slashCode and backslashCode matrix using below formula –

$$\text{slashCode[row][col]} = \text{row} + \text{col}$$

$$\text{backslashCode[row][col]} = \text{row} - \text{col} + (\text{N}-1)$$

Using above formula will result in below matrices

7	6	5	4	3	2	1	0
8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2
10	9	8	7	6	5	4	3
11	10	9	8	7	6	5	4
12	11	10	9	8	7	6	5
13	12	11	10	9	8	7	6
14	13	12	11	10	9	8	7

 $r - c + 7$ 

0	1	2	3	4	5	6	7
1	2	3	4	5	6	7	8
2	3	4	5	5	7	8	9
3	4	5	6	7	8	9	10
4	5	6	7	8	9	10	11
5	5	7	8	9	10	11	12
6	7	3	9	10	11	12	13
7	8	9	10	11	12	13	14

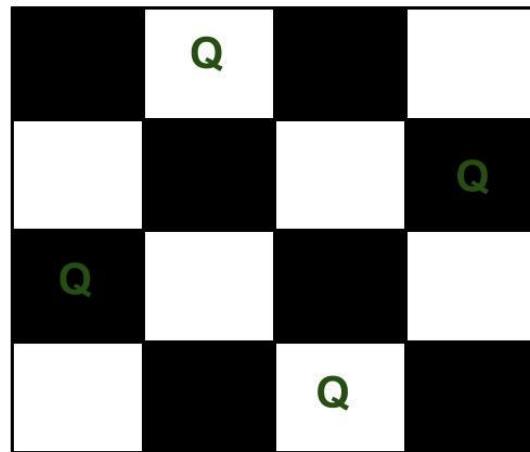
 $r + c$

The ‘N – 1’ in the backslash code is there to ensure that the codes are never negative because we will be using the codes as indices in an array.

Now before we place queen i on row j, we first check whether row j is used (use an array to store row info). Then we check whether slash code (  $j + i$  ) or backslash code (  $j - i + 7$  ) are used (keep two arrays that will tell us which diagonals are occupied). If yes, then we have to try a different location for queen i. If not, then we mark the row and the two diagonals as used and recurse on queen i + 1. After the recursive call returns and before we try another position for queen i, we need to reset the row, slash code and backslash code as unused again, like in the code from the previous notes.

- N Queen Problem using Backtracking:

The N Queen is the problem of placing N chess queens on an  $N \times N$  chessboard so that no two queens attack each other. For example, the following is a solution for the 4 Queen problem.



The expected output is a binary matrix that has 1s for the blocks where queens are placed. For example, the following is the output matrix for the above 4 queen solution.

```
{ 0, 1, 0, 0}
{ 0, 0, 0, 1}
{ 1, 0, 0, 0}
{ 0, 0, 1, 0}
```

➤ Backtracking Algorithm:

The idea is to place queens one by one in different columns, starting from the leftmost column. When we place a queen in a column, we check for clashes with already placed queens. In the current column, if we find a row for which there is no clash, we mark this row and column as part of the solution. If we do not find such a row due to clashes, then we backtrack and return false.

1. Start in the leftmost column
2. If all queens are placed  
return true
3. Try all rows in the current column.  
Do following for every tried row.
  - a) If the queen can be placed safely in this row then mark this [row, column] as part of the solution and recursively check if placing queen here leads to a solution.
  - b) If placing the queen in [row, column] leads to a solution then return true.
  - c) If placing queen doesn't lead to a solution then unmark this [row, column] (Backtrack) and go to step (a) to try other rows.
4. If all rows have been tried and nothing worked, return false to trigger backtracking.

**Conclusion:** Thus we have Implement a solution for a Constraint Satisfaction Problem using Branch and Bound and Backtracking for n-queens problem.

<b>Lab Assignment No.</b>	B5
<b>Title</b>	Develop an elementary chatbot for any suitable customer interaction application.
<b>Roll No.</b>	
<b>Class</b>	TE
<b>Date of Completion</b>	
<b>Subject</b>	Laboratory Practice II
<b>Assessment Marks</b>	
<b>Assessor's Sign</b>	

## ASSIGNMENT No: B5

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

**Problem Statement:** Develop an elementary chatbot for any suitable customer interaction application.

**Objective:**

- Understand and Implement chatbot for any suitable customer interaction application

**Outcome:**

- Ability to choose an appropriate problem solving method and knowledge representation technique

**Software Required:**

- Python

**Theory:**

- Understanding the Chatbot:

A Chatbot is an Artificial Intelligence-based software developed to interact with humans in their natural languages. These chatbots are generally converse through auditory or textual methods, and they can effortlessly mimic human languages to communicate with human beings in a human-like way. A chatbot is considered one of the best applications of natural languages processing.

- Two category the Chatbots:

➤ Rule-based Chatbots:

The Rule-based approach trains a chatbot to answer questions based on a list of pre-determined rules on which it was primarily trained. These set rules can either be pretty simple or quite complex, and we can use these rule-based chatbots to handle simple queries but not process more complicated requests or queries.

➤ Self-learning Chatbots:

Self-learning chatbots are chatbots that can learn on their own. These leverage advanced technologies such as Artificial Intelligence (AI) and Machine Learning (ML) to train themselves from behaviours and instances. Generally, these chatbots are quite smarter

than rule-based bots. We can classify the Self-learning chatbots furtherly into two categories - Retrieval-based Chatbots and Generative Chatbots.

- Retrieval-based Chatbots:

A retrieval-based chatbot works on pre-defined input patterns and sets responses. Once the question or pattern is inserted, the chatbot utilizes a heuristic approach to deliver the relevant response. The model based on retrieval is extensively utilized to design and develop goal-oriented chatbots using customized features such as the flow and tone of the bot in order to enhance the experience of the customer.

- Generative Chatbots:

Unlike retrieval-based chatbots, generative chatbots are not based on pre-defined responses - they leverage seq2seq neural networks. This is constructed on the concept of machine translation, where the source code is converted from one language to another language. In the seq2seq approach, the input is changed into an output.

The first chatbot named ELIZA was designed and developed by Joseph Weizenbaum in 1966 that could imitate the language of a psychotherapist in only 200 lines of code. But as the technology gets more advance, we have come a long way from scripted chatbots to chatbots in Python today.

- Chatbot in present Generation:

Today, we have smart Chatbots powered by Artificial Intelligence that utilize natural language processing (NLP) in order to understand the commands from humans (text and voice) and learn from experience. Chatbots have become a staple customer interaction utility for companies and brands that have an active online existence (website and social network platforms).

With the help of Python, Chatbots are considered a nifty utility as they facilitate rapid messaging between the brand and the customer. Let us think about Microsoft's Cortana, Amazon's Alexa, and Apple's Siri. Aren't these chatbots wonderful? It becomes quite interesting to learn how to create a chatbot using the Python programming language.

Fundamentally, the chatbot utilizing Python is designed and programmed to take in the data we provide and then analyze it using the complex algorithms for Artificial Intelligence. It then delivers us either a written response or a verbal one. Since these bots can learn from experiences and behavior, they can respond to a large variety of queries and commands.

Although chatbot in Python has already started to rule the tech scenario at present, chatbots had handled approximately 85% of the customer-brand interactions by 2020 as per the prediction of Gartner.

In light of the increasing popularity and adoption of chatbots in the industry, we can increase the market value by learning how to create a chatbot in Python - among the most extensively utilized programming languages globally.

- Understanding the ChatterBot Library:

ChatterBot is a Python library that is developed to provide automated responses to user inputs. It makes utilization of a combination of Machine Learning algorithms in order to generate multiple types of responses. This feature enables developers to construct chatbots using Python that can communicate with humans and provide relevant and appropriate responses. Moreover, the ML algorithms support the bot to improve its performance with experience.

Another amazing feature of the ChatterBot library is its language independence. The library is developed in such a manner that makes it possible to train the bot in more than one programming language.

- Understanding the working of the ChatterBot library:

When a user inserts a particular input in the chatbot (designed on ChatterBot), the bot saves the input and the response for any future usage. This information (of gathered experiences) allows the chatbot to generate automated responses every time a new input is fed into it.

The program picks the most appropriate response from the nearest statement that matches the input and then delivers a response from the already known choice of statements and responses. Over time, as the chatbot indulges in more communications, the precision of reply progresses.

- Creating a Chatbot using Python:

We will follow a step-by-step approach and break down the procedure of creating a Python chat.

We will begin building a Python chatbot by importing all the required packages and modules necessary for the project. We will also initialize different variables that we want to use in it. Moreover, we will also be dealing with text data, so we have to perform data preprocessing on the dataset before designing an ML model.

This is where tokenizing supports text data - it converts the large text dataset into smaller, readable chunks (such as words). Once this process is complete, we can go for lemmatization to transform a word into its lemma form. Then it generates a pickle file in order to store the objects of Python that are utilized to predict the responses of the bot.

Another major section of the chatbot development procedure is developing the training and testing datasets.

- Why Chatbots are important for a Business or a Website:
  - Quick resolution for a complaint or a problem.
  - Improve business branding thereby achieving great customer satisfaction.
  - Answering questions and answers for customers.
  - Making a reservation at hotel or at restaurant.
  - Save human effort 24×7.
  - Enhance business revenue by providing ideas and inspirations.
  - Finding details about business such as hours of operation, phone number and address.
  - Automate sales and lead generation process.
  - Reduce customer agents waiting time answering phone calls.
- Benefits of using Chatbots:
  - 24×7 availability.
  - Instant answers to queries.
  - Support multi-language to enhance businesses.
  - Simple and Easy to Use UI to engage more customers.
  - Cost effective and user interactive.
  - Avoid communication with call agents thereby reducing the time consuming tasks.
  - Understand the Customer behavior
  - Increase sales of business by offering promo codes or gifts.

**Conclusion:** Thus we have developed an elementary chatbot for any suitable customer interaction application.

# Group C

<b>Lab Assignment No.</b>	C6
<b>Title</b>	Mini Project: Implement any one of the following Expert System <ul style="list-style-type: none"><li>• Information management</li><li>• Hospitals and medical facilities</li><li>• Help desks management</li><li>• Employee performance evaluation</li><li>• Stock market trading</li><li>• Airline scheduling and cargo schedules</li></ul>
<b>Roll No.</b>	
<b>Class</b>	TE
<b>Date of Completion</b>	
<b>Subject</b>	Laboratory Practice II
<b>Assessment Marks</b>	
<b>Assessor's Sign</b>	

## ASSIGNMENT No: C6

**Title:** Implement any one of the following Expert System

- Information management
- Hospitals and medical facilities
- Help desks management
- Employee performance evaluation
- Stock market trading
- Airline scheduling and cargo schedules

**Problem Statement:** Implement any one of the following Expert System

- Information management
- Hospitals and medical facilities
- Help desks management
- Employee performance evaluation
- Stock market trading
- Airline scheduling and cargo schedules

**Objective:**

- Understand and implement Expert System

**Outcome:**

- Ability to choose an appropriate problem solving method and knowledge representation technique

**Software Required:**

- Python

**Theory:**

- Expert Systems:

Artificial Intelligence is a piece of software that simulates the behaviour and judgement of a human or an organization that has experts in a particular domain is known as an expert system. It does this by acquiring relevant knowledge from its knowledge base and interpreting it according to the user's problem. The data in the knowledge base is added by humans that are expert in a particular domain and this software is used by a non-expert user to acquire some information. It is widely used in many areas such as medical diagnosis, accounting, coding, games etc.

An expert system is AI software that uses knowledge stored in a knowledge base to solve problems that would usually require a human expert thus preserving a human expert's knowledge in its knowledge base. They can advise users as well as provide explanations to them about how they reached a particular conclusion or advice. Knowledge Engineering is the term used to define the process of building an Expert System and its practitioners are called Knowledge Engineers. The primary role of a knowledge engineer is to make sure that the computer possesses all the knowledge required to solve a problem. The knowledge engineer must choose one or more forms in which to represent the required knowledge as a symbolic pattern in the memory of the computer.

- Example :

There are many examples of an expert system. Some of them are given below –

➤ MYCIN –

One of the earliest expert systems based on backward chaining. It can identify various bacteria that can cause severe infections and can also recommend drugs based on the person's weight.

➤ DENDRAL –

It was an artificial intelligence-based expert system used for chemical analysis. It used a substance's spectrographic data to predict its molecular structure.

➤ R1/XCON –

It could select specific software to generate a computer system wished by the user.

➤ PXDES –

It could easily determine the type and the degree of lung cancer in a patient based on the data.

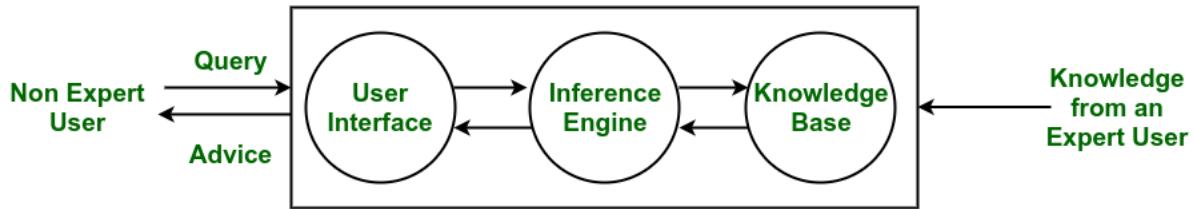
➤ CaDet –

It is a clinical support system that could identify cancer in its early stages in patients.

➤ DXplain –

It was also a clinical support system that could suggest a variety of diseases based on the findings of the doctor.

- Components of an Expert System:



Architecture of an Expert System

➤ Knowledge Base –

The knowledge base represents facts and rules. It consists of knowledge in a particular domain as well as rules to solve a problem, procedures and intrinsic data relevant to the domain.

➤ Inference Engine –

The function of the inference engine is to fetch the relevant knowledge from the knowledge base, interpret it and to find a solution relevant to the user's problem. The inference engine acquires the rules from its knowledge base and applies them to the known facts to infer new facts. Inference engines can also include an explanation and debugging abilities.

➤ Knowledge Acquisition and Learning Module –

The function of this component is to allow the expert system to acquire more and more knowledge from various sources and store it in the knowledge base.

➤ User Interface –

This module makes it possible for a non-expert user to interact with the expert system and find a solution to the problem.

➤ Explanation Module –

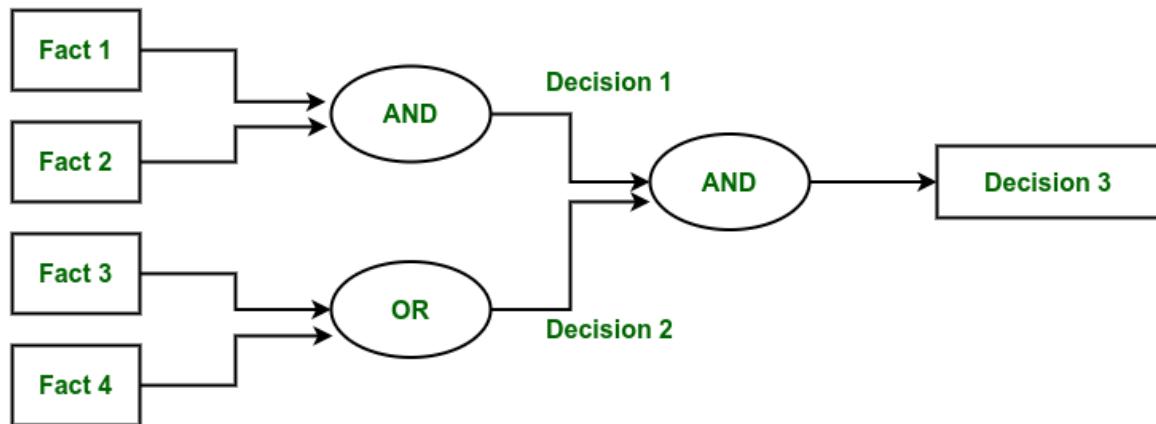
This module helps the expert system to give the user an explanation about how the expert system reached a particular conclusion.

- Two strategies:

The Inference Engine generally uses two strategies for acquiring knowledge from the Knowledge Base, namely –

➤ Forward Chaining:

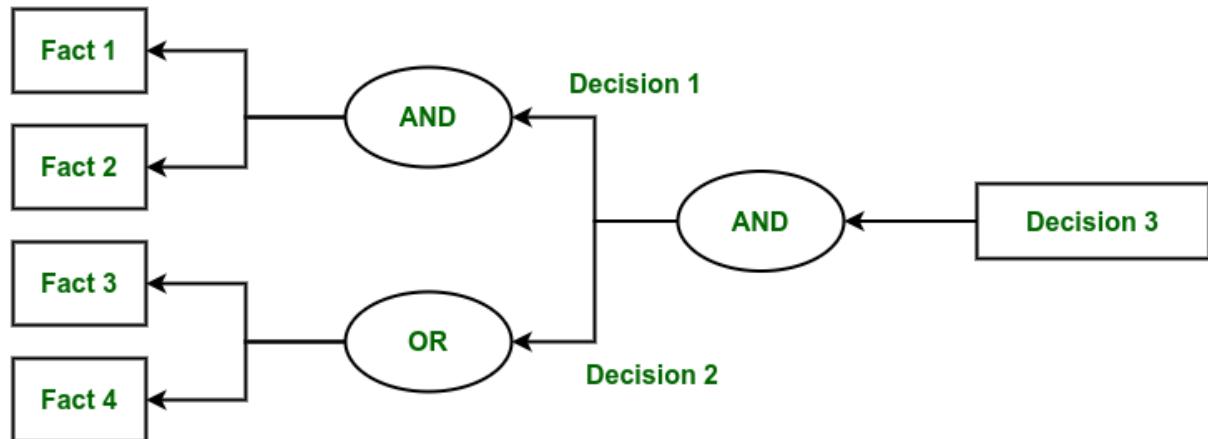
It is a strategic process used by the Expert System to answer the questions – What will happen next. This strategy is mostly used for managing tasks like creating a conclusion, result or effect. Example – prediction or share market movement status.



Forward Chaining

- Backward Chaining

It is a storage used by the Expert System to answer the questions – Why this has happened. This strategy is mostly used to find out the root cause or reason behind it, considering what has already happened. Example – diagnosis of stomach pain, blood cancer or dengue, etc.



Backward Chaining

- Characteristics of an Expert System:
  - Human experts are perishable, but an expert system is permanent.
  - It helps to distribute the expertise of a human.
  - One expert system may contain knowledge from more than one human experts thus making the solutions more efficient.
  - It decreases the cost of consulting an expert for various domains such as medical diagnosis.
  - They use a knowledge base and inference engine.

- Expert systems can solve complex problems by deducing new facts through existing facts of knowledge, represented mostly as if-then rules rather than through conventional procedural code.
  - Expert systems were among the first truly successful forms of artificial intelligence (AI) software.
- Limitations:
    - Do not have human-like decision-making power.
    - Cannot possess human capabilities.
    - Cannot produce correct result from less amount of knowledge.
    - Requires excessive training.
  - Advantages:
    - Low accessibility cost.
    - Fast response.
    - Not affected by emotions, unlike humans.
    - Low error rate.
    - Capable of explaining how they reached a solution.
  - Disadvantages:
    - The expert system has no emotions.
    - Common sense is the main issue of the expert system.
    - It is developed for a specific domain.
    - It needs to be updated manually. It does not learn itself.
    - Not capable to explain the logic behind the decision.
  - Applications:

The application of an expert system can be found in almost all areas of business or government. They include areas such as –

- Different types of medical diagnosis like internal medicine, blood diseases and so on.
- Diagnosis of the complex electronic and electromechanical system.
- Diagnosis of a software development project.
- Planning experiment in biology, chemistry and molecular genetics.
- Forecasting crop damage.
- Diagnosis of the diesel-electric locomotive system.
- Identification of chemical compound structure.
- Scheduling of customer order, computer resources and various manufacturing task.
- Assessment of geologic structure from dip meter logs.
- Assessment of space structure through satellite and robot.

- The design of VLSI system.
- Teaching students specialize task.
- Assessment of log including civil case evaluation, product liability etc.

Expert systems have evolved so much that they have started various debates about the fate of humanity in the face of such intelligence, with authors such as Nick Bostrom (Professor of Philosophy at Oxford University), pondering if computing power has transcended our ability to control it.

**Conclusion:** Thus we have implemented Expert System

## PRACTICAL NO : 01

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 keypair. 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 buttonNow 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 : 02

Title : Assignment to install and configure Google App Engine  
STEPS TO INSTALL & CONFIGURE :

**Created By :Ms. Manisha Vilas Khadse**

Before begin

1. Create a Google Cloud Platform project, if you don't have one already.
2. Make sure that Python 2.7 is installed on your system: python -V Note: As of Cloud SDK version 206.0.0, the gcloud CLI has experimental support for running using a Python 3.4+ interpreter (run gcloud topic startup for exclusions and more information on configuring your Python interpreter). All other Cloud SDK tools still require a Python 2.7 interpreter.
3. Download the archive file best suited to your operating system. Most machines will run the 64-bit package. If you'd like to check, run uname -m to verify if you're running a 64-bit system.  
Platform Package Size SHA256 Checksum Linux For 64-bit google-cloud-sdk25.6  
b1c87fc9451598a76cf66978dd8aa06482bfced639b56cf31559dc2c7f8b7b90 229.0.0-linuxMB  
For 32-bit google-cloud-sdk25.2  
ee8c45f8018d0fee92b07c32cc6d8c891241da0b88bfe289d4e58e6746c3f668 229.0.0-linuxMB  
(x86) x86.tar.gz Alternatively, to download the Linux 64-bit archive file from your command-line, run: curl -O [https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-sdk-229.0.0-linux-x86\\_64.tar.gz](https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-sdk-229.0.0-linux-x86_64.tar.gz)curl -O <https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-sdk-229.0.0-linux-x86.tar.gz>
4. Extract the archive to any location on your file system; preferably, your Home folder. On Linux, you can extract the archive file by running this command: tar zxvf [ARCHIVE\_FILE]  
google-cloud-sdk
5. If you're having trouble getting the gcloud command to work, ensure your \$PATH is defined appropriately. Use the install script to add Cloud SDK tools to your path. You will also be able to opt-in to command-completion for your bash shell and usage statistics collection during the installation process.

Run the script using this command:

`./google-cloud-sdk/install.sh`

Restart your terminal for the changes to take effect. Alternatively, you can call Cloud SDK after extracting the downloaded archive by invoking its executables via the full path.

Initialize the SDK Use the gcloud init command to perform several common SDK setup tasks. These include authorizing the SDK tools to access Google Cloud Platform using your user account credentials and setting up the default SDK configuration.

To initialize the SDK:

1. Run the following at a command prompt: gcloud init Note: To prevent the command from launching a web browser, use gcloud init -- console- only instead.  
To authorize without a web browser and non-interactively, create a service account with the appropriate scopes using the Google Cloud Platform Console and use gcloud auth activate-service-account with the corresponding JSON key file.
2. Accept the option to log in using your Google user account: To continue, you must log in.  
Would you like to log in (Y/n)? Y
3. In your browser, log in to your Google user account when prompted and click Allow to grant permission to access Google Cloud Platform resources.
4. At the command prompt, select a Cloud Platform project from the list of those where you have Owner, Editor or Viewer permissions:

Pick cloud project to use:

- [1] [my-project-1]
- [2] [my-project-2] ..

. Please enter your numeric choice: If you only have one project, gcloud init selects it for you.

5. If you have the Google Compute Engine API enabled, gcloud init allows you to choose a default Compute Engine zone:

Which compute zone would you like to use as project default?

- [1] [asia-east1-a]
- [2] [asia-east1-b] ..

. [14] Do not use default zone Please enter your numeric choice:

gcloud init confirms that you have complete the setup steps successfully:gcloud has now been configured!

You can use [gcloud config] to change more gcloud settings. Your active configuration is:  
[default]

Run core gcloud commands

Run these gcloud commands to view information about your SDK installation:

1. To list accounts whose credentials are stored on the local system:gcloud auth list gcloud displays a list of credentialed accounts: Credential Accounts ACTIVE ACCO UNT \* example-user1@gmail.com [exampleuser-2@gmail.com](mailto:exampleuser-2@gmail.com)

2. To list the properties in your active SDK configuration:

gcloud config list gcloud displays the list of properties: [core] account = example-user1@gmail.com disable\_usage\_reporting = False project = example-project

3. To view information about your Cloud SDK installation and the active SDK configuration: gcloud info gcloud displays a summary of information about your Cloud SDK installation. This includes information about your system, the installed SDK components, the active user account and current project, and the properties in the active SDK configuration.

4. To view information about gcloud commands and other topics from the command line: gcloud help For example, to view the help for gcloud compute instances create: gcloud help compute instances create gcloud displays a help topic that contains a description of the command, a list of commandflags and arguments, and examples of how to use it.

#### How to Run Program:

Now as we have finished installing app engine, now it's time to createand upload an app. In this case we will be taking example of a "HELLO WORLD" app in python.

1. As we already have made sure that we have python installed in our system, It will be easier for us to clone existing code and deploy it rather than creating our own so we will use python docs- sample. Run the command "git clone

<https://github.com/GoogleCloudPlatform/python-docs-samples>".

2. cd python-docs-samples/appengine/standard/hello\_world

3. dev\_appserver.py app.yaml It will run and give you the url of default and admin. If you go to the link of default you see the text hello world like this.

## **Output :**



**OR**

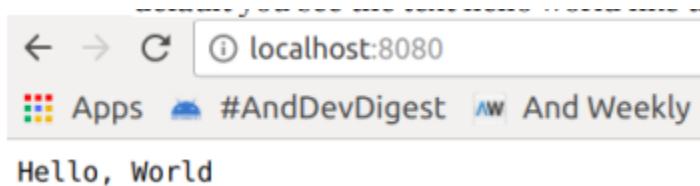
Google App Engine, Installing it the right way in ubuntu.

1. Make sure you have python installed in your ubuntu system. run the command “python -V” and most probably you will get “Python 2.7.6” or above.
2. Crul <https://sdk.cloud.google.com> and use bash to run the commands by typing this command `curl https://sdk.cloud.google.com | bash`
3. Whenever you get to choose directories just hit enter, “YEAH IT WILL BE FINE”.
4. Follow the instructions in the installation process.
5. Then run `gcloud init`
6. Follow the installation instructions as they are very straight forward.
7. Choose the account you want to use for google app engine.
8. Choose the project with numeric choice (don’t use textual, you might make mistake). If you do not already have a google app engine project create a app engine project by following this link.  
<https://console.cloud.google.com/start>
9. Enable google api by pressing Y in the command line prompt. Now as we have finished installing appengine, now it’s time to create and upload an app. In this case we will be taking example of a “HELLO WORLD” app in python.

1. As we already have made sure that we have python installed in our system, It will be easier for us to clone existing code and deploy it rather than creating our own so we will use pythondocs-sample. Run the command “git clone <https://github.com/GoogleCloudPlatform/python-docssamples>”.

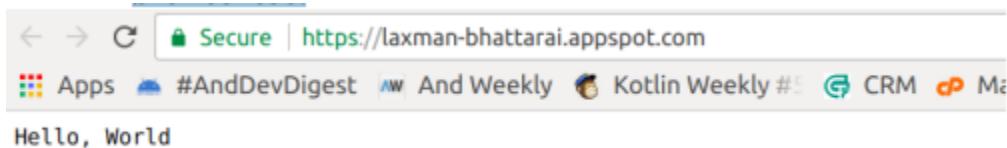
2. cd to hello world sample by typing the command “ cd pythondocs-samples/appengine/standard/hello\_world”.

3. Then run the command “dev\_appserver.py app.yaml”. It will run and give you the url of default and admin. If you go to the link of default you see the text hello world like this.



This is how you run the python app in your local server. But what we have to do is hosting the app in google app engine. To do so Now let's follow the following instructions.

1. Run the command Ctrl + C .
2. Being in the same working directory hello-world runt he command gcloud app deploy
3. Select the project you want to deploy the app , press Y and enter to continue. after that you will get the console output “Deployed service[default] to [Your web url for appengine] ”
4. If you copy and paste the url, you will see the hello world in the browser too.



## PRACTICAL NO : 03

**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(enter name to apex class and save Clt+s)

3.Type below mentioned code

```
public class MySampleApexClass {      //Class definition and body
    public Double myValue = 0; //Class Member variable
    //public String myString = ""; //Class Member variable

    public Double get() {
        // Method definition and body
        // do some calculation
        myValue = myValue+10;
        return myValue;
    }
}
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is [https://dypiemr3-dev-ed.develop.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](https://dypiemr3-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The code editor contains the following Apex class:

```
1 public class MySampleApexClass {           //Class definition and body
2     public Double myValue = 0; //Class Member variable
3     //public String myString = ''; //Class Member variable
4
5     public Double get() {
6         // Method definition and body
7         // do some calculation
8         myValue = myValue+10;
9         return myValue;
10    }
11 }
```

Below the code editor, there are tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The View State tab is selected. A table below shows the current state of variables:

Name	Type	Value	Size (KB)	% Of Parent

The system status bar at the bottom shows: Search, 29°C, ENG, 22:09.

#### 4. Click on Debug ---> Open Execute Acronymous Window

#### 5. Type below code(Apex Code)

```
MySampleApexClass m=new MySampleApexClass();
Double Value=m.get();
system.debug('valued='+Value);
```

The screenshot shows the Salesforce Developer Console in Google Chrome. The URL is [https://dypiemr3-dev-ed.develop.my.salesforce.com/\\_ui/common/apex/debug/ApexCSIPage](https://dypiemr3-dev-ed.develop.my.salesforce.com/_ui/common/apex/debug/ApexCSIPage). The Execution Log pane shows the following log entries:

Timestamp	Event	Details
22:00:39:004	USER_DEBUG	[1]DEBUG hello world
22:00:39:004	USER_DEBUG	[2]DEBUG pune
22:00:39:004	USER_DEBUG	[3]DEBUG hello
22:00:39:007	USER_DEBUG	[6]DEBUG valued=10.0

An 'Enter Apex Code' dialog box is open in the foreground, containing the following Apex code:

```
1 system.debug('hello world');
2 System.debug('pune');
3 system.debug('hello');
4 MySampleApexClass m=new MySampleApexClass();
5 Double Value=m.get();
6 system.debug('valued='+Value);
```

The developer console interface includes tabs for Logs, Tests, Checkpoints, Query Editor, View State, Progress, and Problems. The View State tab is selected. At the bottom, there are buttons for Open Log, Execute, and Execute Highlighted.

#### 6. Click on Open log then Execute code

#### 7. Click on Debug only (You will get output)

Developer Console - Google Chrome

dypiemr3-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File ▾ Edit ▾ Debug ▾ Test ▾ Workspace ▾ Help ▾ < >

↳ teAnonymous @4/8/2024, 11:49:05 AM [ ] Log executeAnonymous @4/8/2024, 9:59:03 PM [ ] Log executeAnonymous @4/8/2024, 10:00:39 PM [ ] Log executeAnonymous @4/8/2024, 10:17:11 PM [ ]

### Execution Log

Timestamp	Event	Details
22:17:11:001	USER_INFO	[EXTERNAL][005GA00000AC0h6]manishakdyp5@gmail.com[(GMT+05:30) India Standard Time (Asia/Kolkata)][GMT+05:30]
22:17:11:001	EXECUTION_ST...	
22:17:11:001	CODE_UNIT_ST...	[EXTERNAL]execute_anonymous_apex
22:17:11:001	VARIABLE_SCO...	[5]mValue[Double]false[false]
22:17:11:002	VARIABLE_SCO...	[4]mMySampleApexClass[true]false
22:17:11:002	HEAP_ALLOCATE	[79]Bytes:3
22:17:11:002	HEAP_ALLOCATE	[84]Bytes:152
22:17:11:002	HEAP_ALLOCATE	[399]Bytes:408
22:17:11:002	HEAP_ALLOCATE	[412]Bytes:408
22:17:11:002	HEAP_ALLOCATE	[520]Bytes:48
22:17:11:002	HEAP_ALLOCATE	[139]Bytes:6
22:17:11:002	HEAP_ALLOCATE	[EXTERNAL]Bytes:11
22:17:11:002	STATEMENT_EX...	[1]
22:17:11:002	STATEMENT_EX...	[1]
22:17:11:002	HEAP_ALLOCATE	[1]Bytes:11
22:17:11:003	HEAP_ALLOCATE	[52]Bytes:5

This Frame Executable Debug Only Filter Click here to filter the log

View State Progress Problems

Name	Type	Value	Size (KB)	% Of Parent
------	------	-------	-----------	-------------

Developer Console - Google Chrome

dypiemr3-dev-ed.develop.my.salesforce.com/\_ui/common/apex/debug/ApexCSIPage

File Edit Debug Test Workspace Help < >

↳ teAnonymous @4/8/2024, 11:49:05 AM [x] MySampleApexClass.apxc [x] Log executeAnonymous @4/8/2024, 9:59:03 PM [x] Log executeAnonymous @4/8/2024, 10:00:39 PM [x] Log executeAnonymous @4/8/2024, 10:17:11 PM [x]

**Execution Log**

Timestamp	Event	Details
22:17:11:003	USER_DEBUG	[1]:DEBUG hello world
22:17:11:003	USER_DEBUG	[2]:DEBUG pune
22:17:11:003	USER_DEBUG	[3]:DEBUG hello
22:17:11:006	USER_DEBUG	[6]:DEBUG valued=10.0

This Frame  Executable  Debug Only  Filter [Click here to filter the log](#)

[Logs](#) [Tests](#) [Checkpoints](#) [Query Editor](#) [View State](#) [Progress](#) [Problems](#)

Name	Type	Value	Size (KB)	% Of Parent
------	------	-------	-----------	-------------

**Thank You!!!!**

## Apex Sample Code

Code Coverage: None API Version: 56

```
1 public class Calculator {
2     public double result;
3
4     public double add(integer x, integer y)
5     {
6         result=x+y;
7         return result;
8     }
9
10    public double sub(integer x, integer y)
11    {
12        result=x-y;
13        return result;
14    }
15 }
```

Logs Tests Checkpoints Query Editor View State Progress Problems

Enter Apex Code

```
1 Calculator c=new Calculator();
2 double sum=c.add(3,5);
3 double difference=c.sub(5, 7);    I
4
5 System.debug('Sum='+sum);
6 System.debug('Difference='+difference);
```

## PRACTICAL NO : 04

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

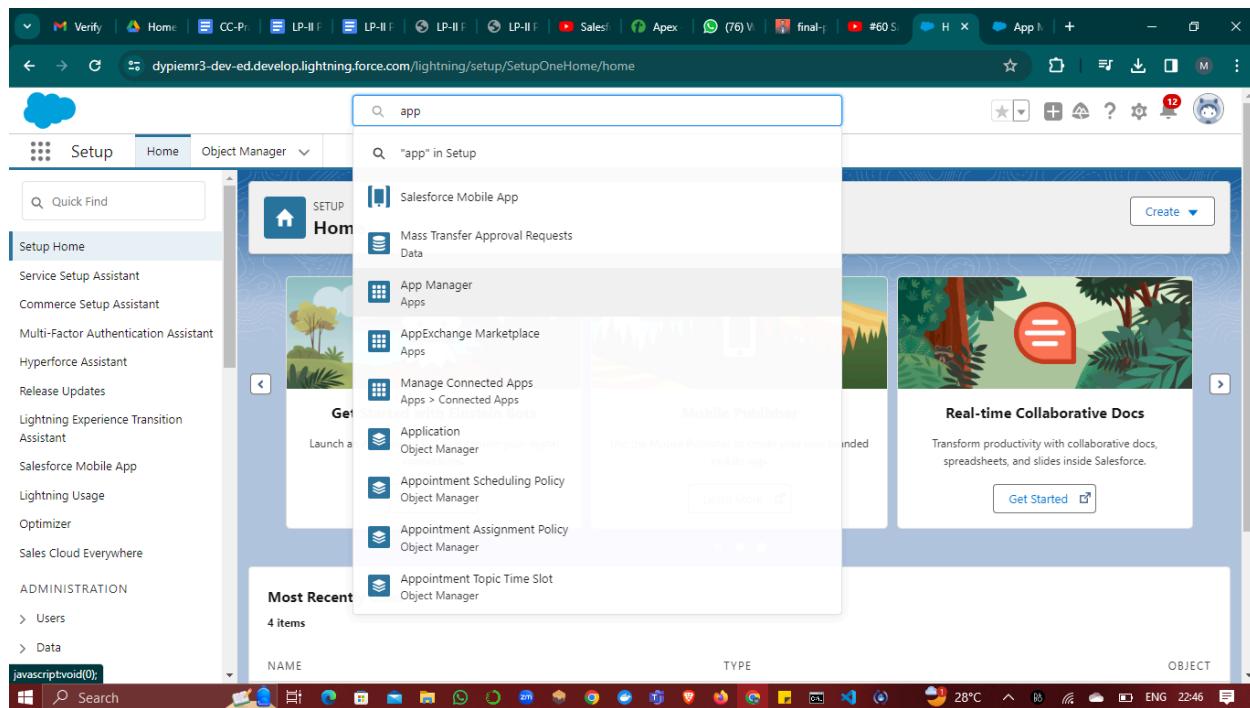
Created By Ms. Manisha Vilas Khadse

### STEPS TO CREATE AN APPLICATION(mini project) :

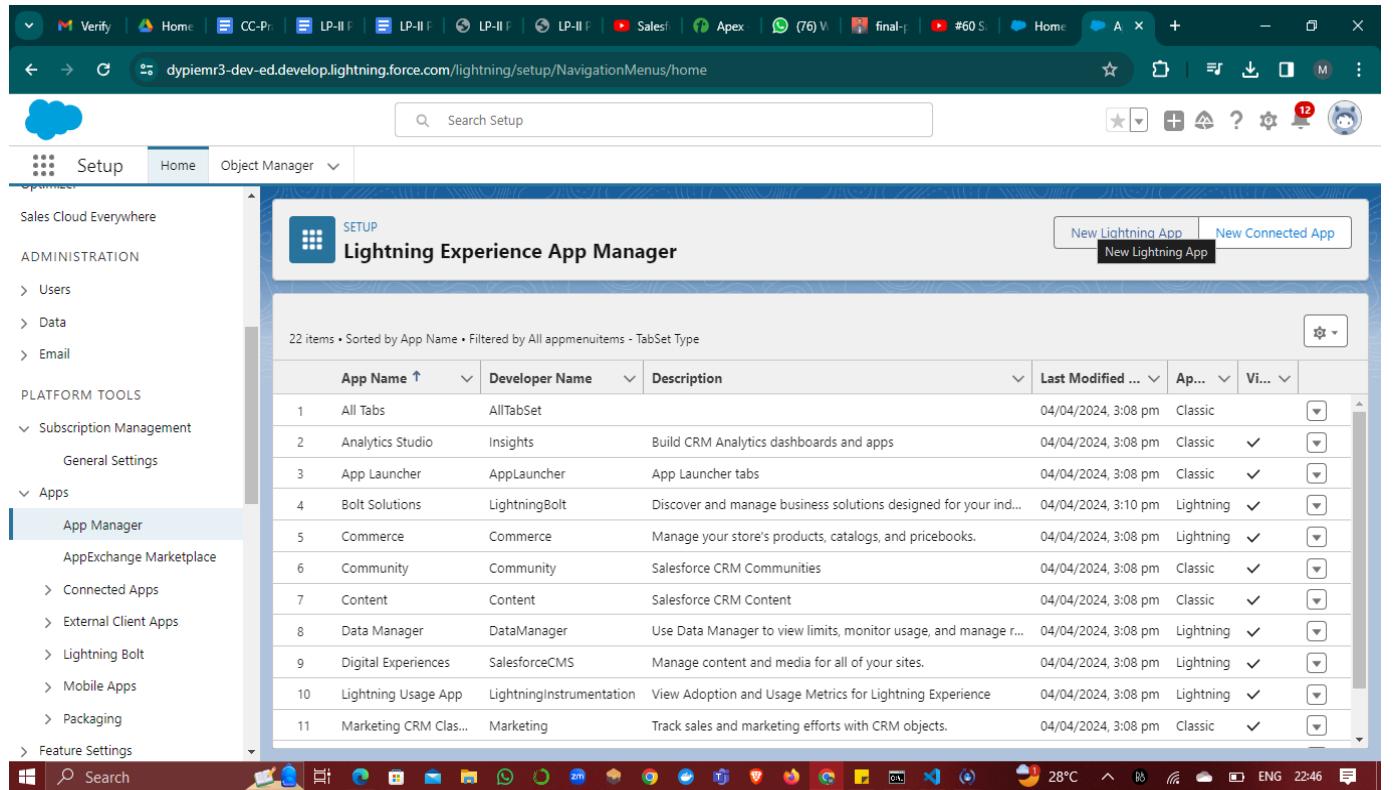
1. Open salesforce in Lightning Experience
2. Open App Launcher View all Quick Search App Manager Click on New Lightning App fill the mention information Next App option(optional ) Next Add Utility (optional) Next Navigation Item(you can add item) Next User Profile(use the user profile that can access this app) Save and Finish. You will get Lightning Experience App Manager Window Search Your App by its name in App Launcher just by entering some character, you can see app name with its logo Click on App you can see your application appearance in window Output

**Note :** steps images are provided here.....

#### 1. . Open salesforce in Lightning Experience



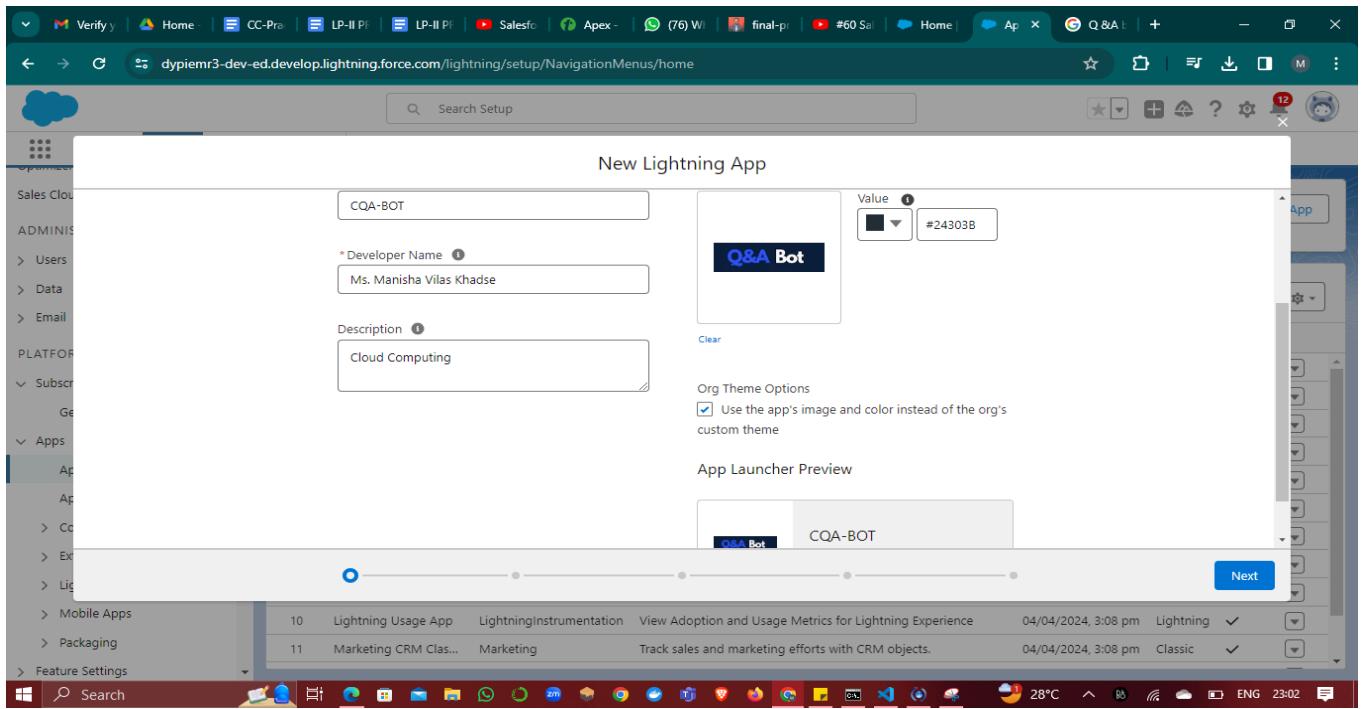
## 2. Open App Launcher View all ( Quick Search App Manager) Click on New Lightning App



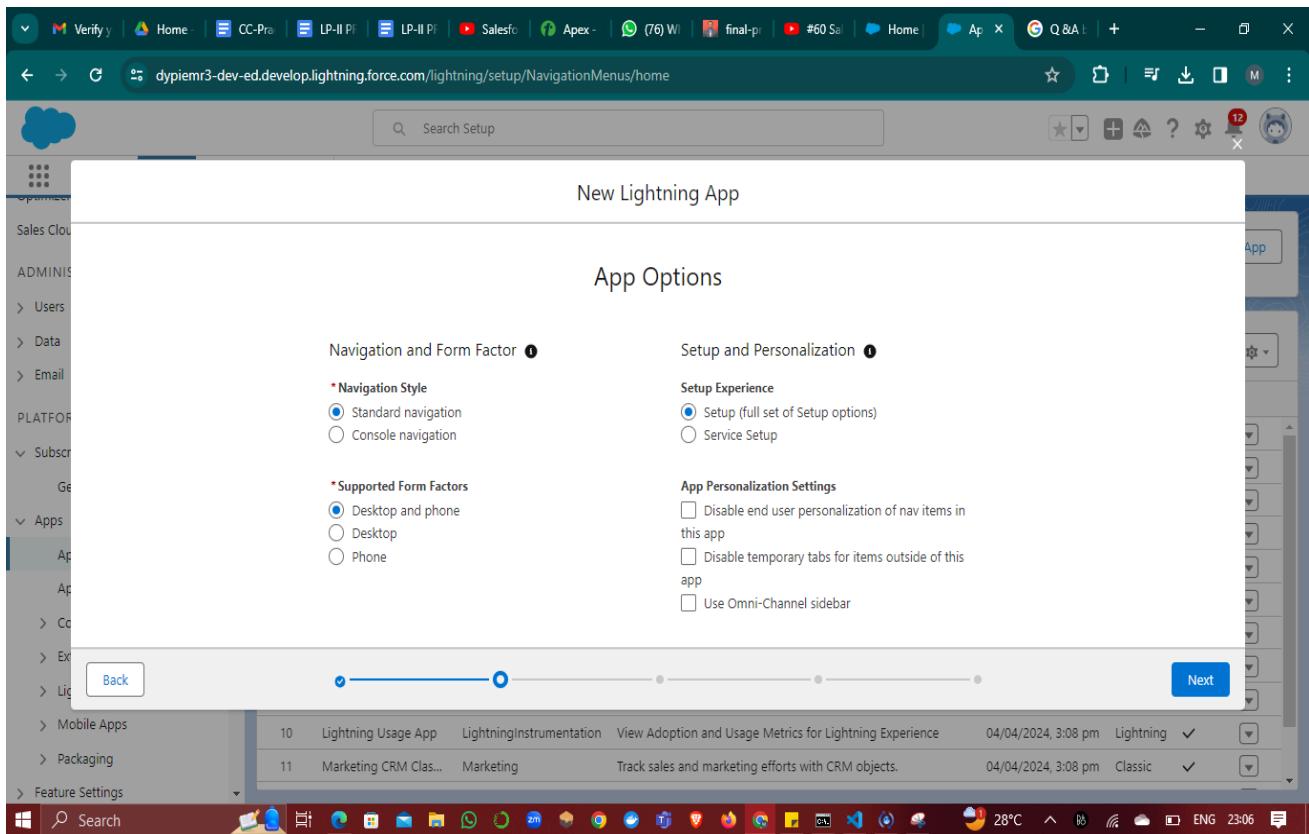
The screenshot shows the Salesforce App Manager interface. The left sidebar is collapsed, and the main area displays a table of apps. The table has columns for App Name, Developer Name, Description, Last Modified, Type, and various checkboxes. The first few rows include All Tabs, Analytics Studio, App Launcher, Bolt Solutions, Commerce, Community, Content, Data Manager, Digital Experiences, Lightning Usage App, and Marketing CRM Clas... . The 'Type' column indicates most are Classic, except for Bolt Solutions and Lightning Usage App which are Lightning.

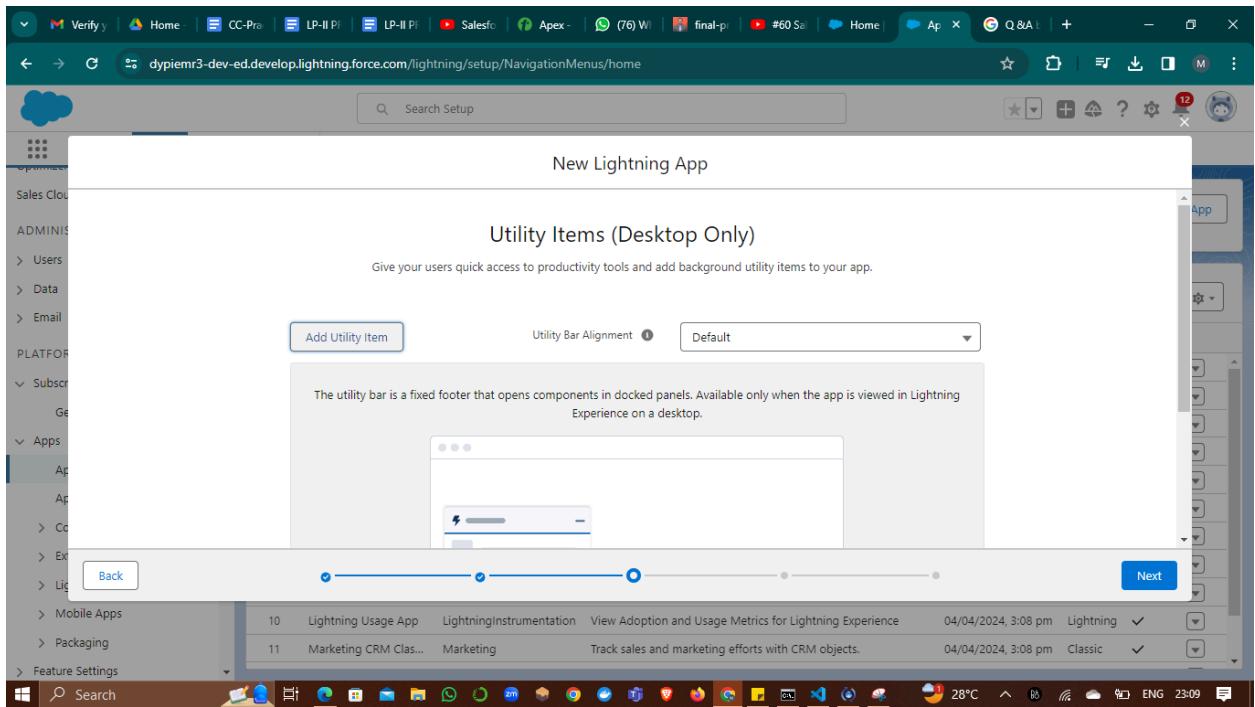
App Name ↑	Developer Name	Description	Last Modified ...	Type
1 All Tabs	AllTabSet		04/04/2024, 3:08 pm	Classic
2 Analytics Studio	Insights	Build CRM Analytics dashboards and apps	04/04/2024, 3:08 pm	Classic
3 App Launcher	AppLauncher	App Launcher tabs	04/04/2024, 3:08 pm	Classic
4 Bolt Solutions	LightningBolt	Discover and manage business solutions designed for your ind...	04/04/2024, 3:10 pm	Lightning
5 Commerce	Commerce	Manage your store's products, catalogs, and pricebooks.	04/04/2024, 3:08 pm	Lightning
6 Community	Community	Salesforce CRM Communities	04/04/2024, 3:08 pm	Classic
7 Content	Content	Salesforce CRM Content	04/04/2024, 3:08 pm	Classic
8 Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage r...	04/04/2024, 3:08 pm	Lightning
9 Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	04/04/2024, 3:08 pm	Lightning
10 Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	04/04/2024, 3:08 pm	Lightning
11 Marketing CRM Clas...	Marketing	Track sales and marketing efforts with CRM objects.	04/04/2024, 3:08 pm	Classic

## 3. fill the mention information in new lightning app : then next

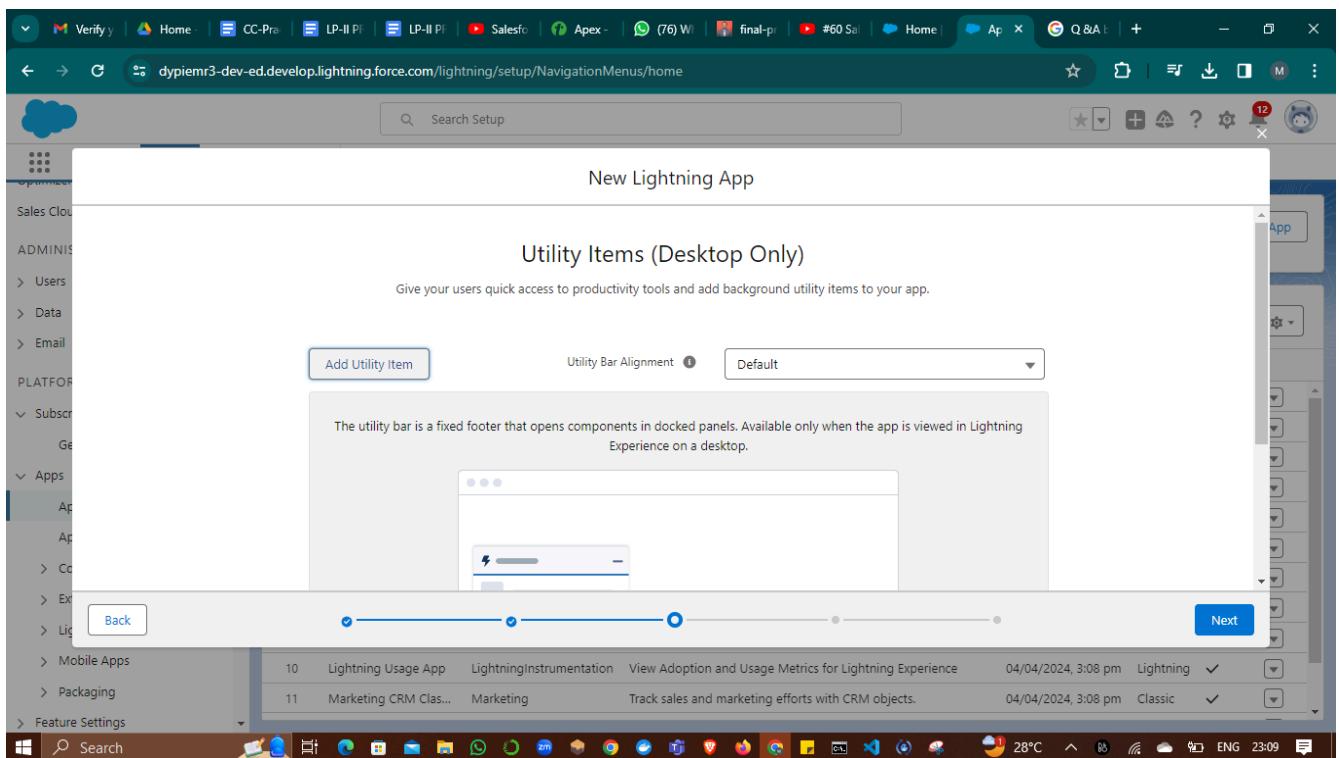


#### 4. App option(optional ) Next

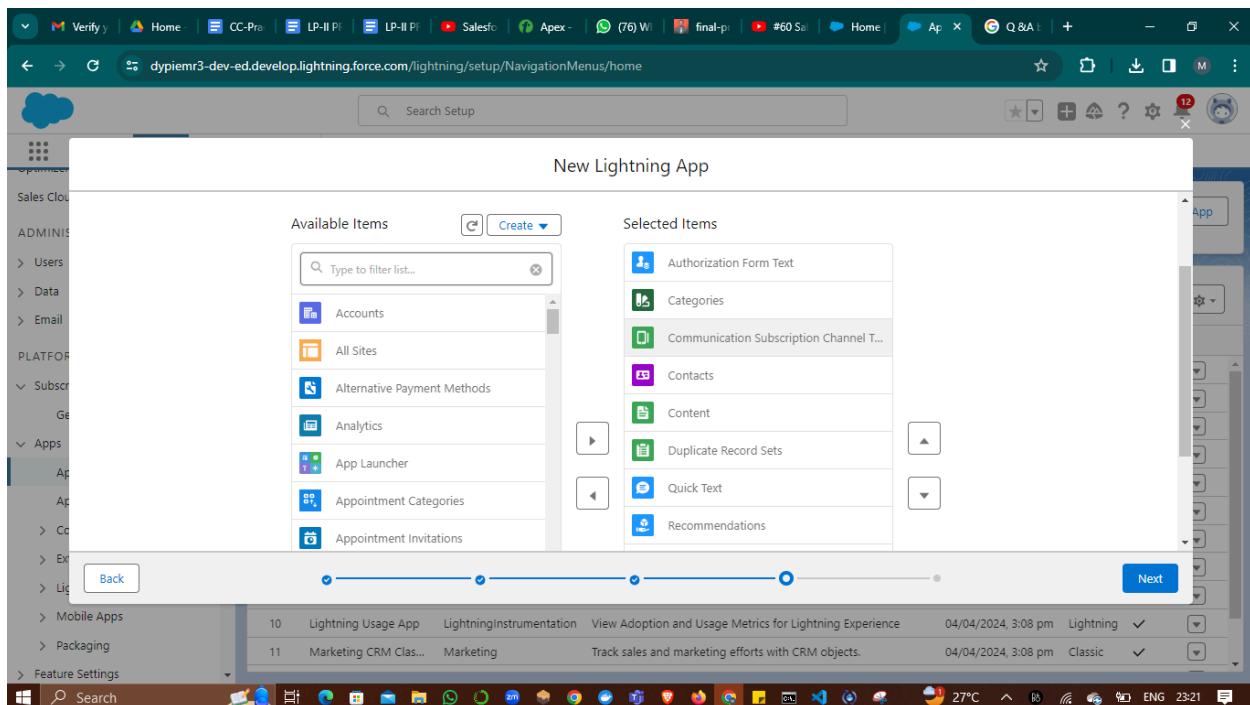
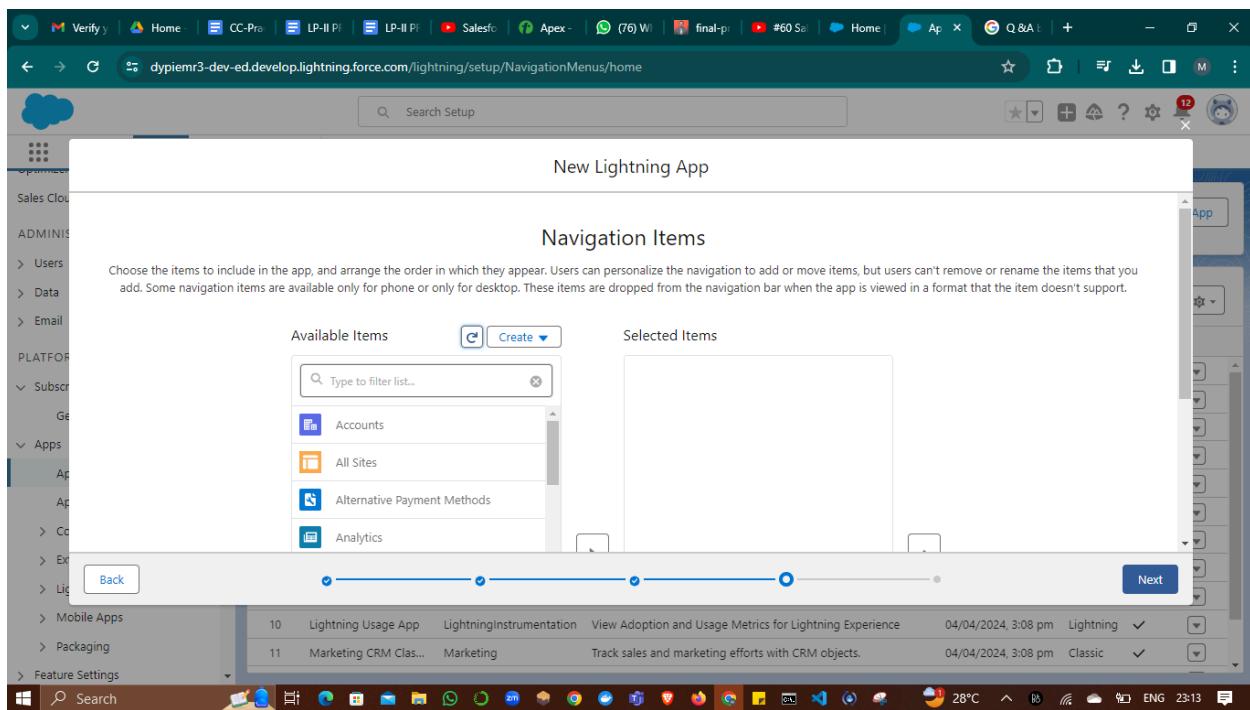




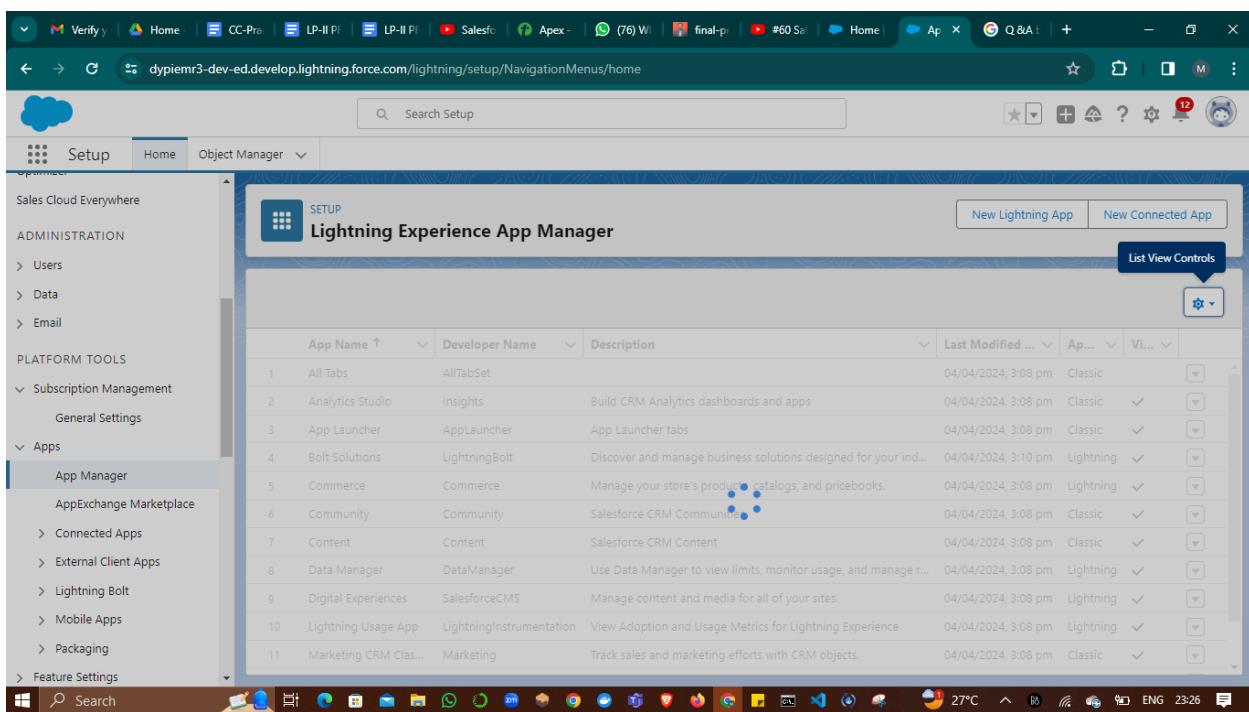
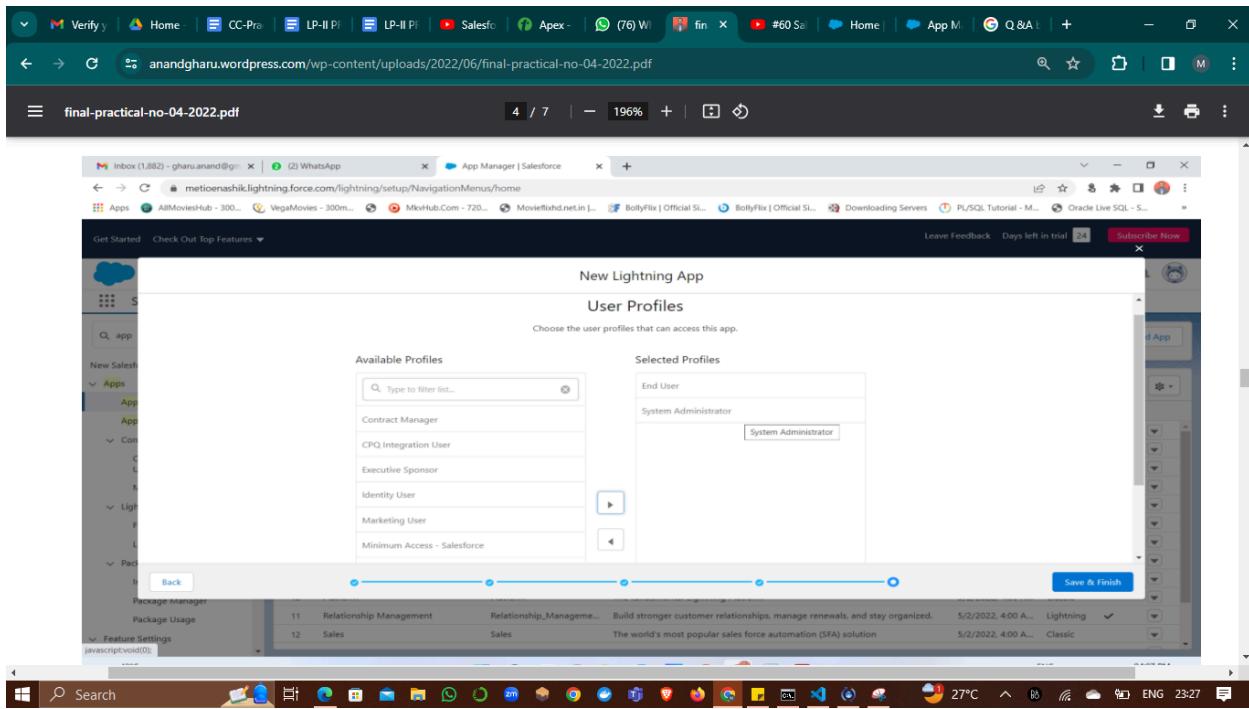
## 5. Add Utility (optional) Next .



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



**7. User Profile(use the user profile that can access this app) Save and Finish**



**8. You will get Lightning Experience App Manager Window(here, you can see your application)**

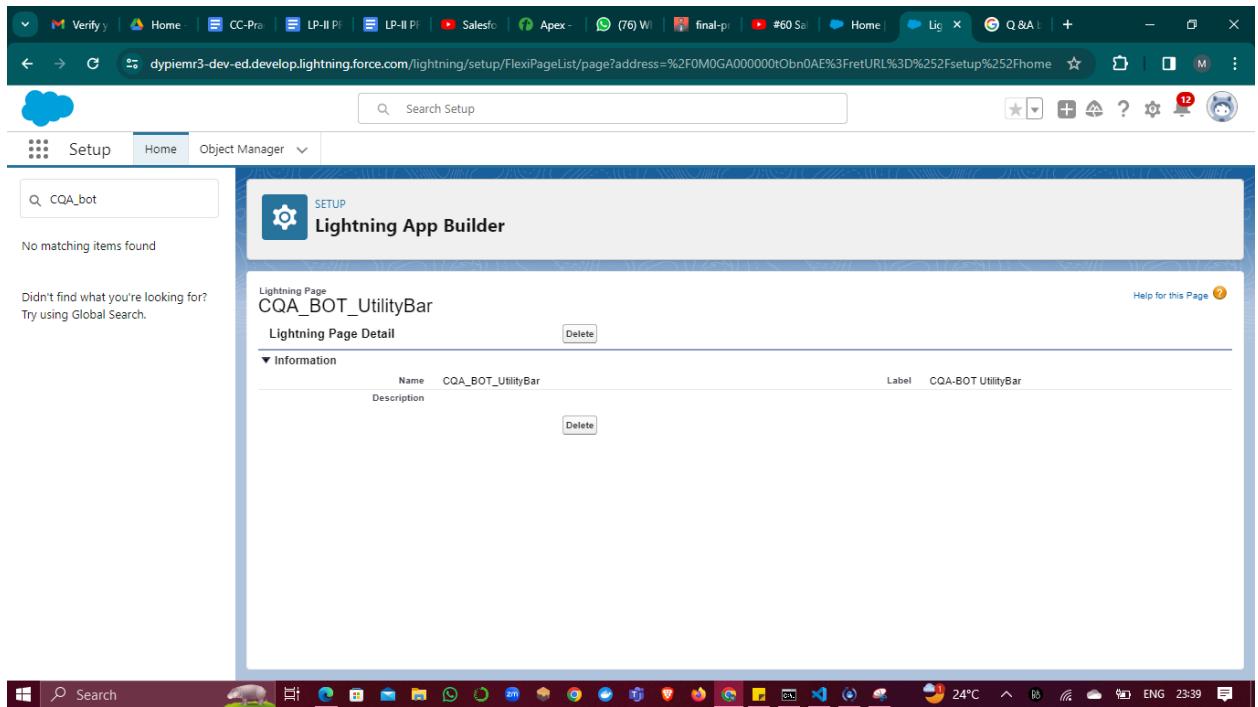
The screenshot shows the 'Lightning Experience App Manager' page. The left sidebar lists various setup categories like Service Setup Assistant, Commerce Setup Assistant, and Administration. The main area displays a table of installed apps:

App Name	Developer Name	Description	Last Modified	Type
All Tabs	AllTabSet	Build CRM Analytics dashboards and apps	04/04/2024, 3:08 pm	Classic
Analytics Studio	Insights	Discover and manage business solutions designed for your in...	04/04/2024, 3:08 pm	Lightning
App Launcher	AppLauncher	App Launcher tabs	04/04/2024, 3:08 pm	Classic
Bolt Solutions	LightningBolt	Manage your store's products, catalogs, and pricebooks.	04/04/2024, 3:10 pm	Lightning
Commerce	Commerce	Salesforce CRM Communities	04/04/2024, 3:08 pm	Lightning
Community	Community	Salesforce CRM Content	04/04/2024, 3:08 pm	Classic
Content	Content	Cloud Computing	08/04/2024, 11:26 pm	Lightning
Data Manager	DataManager	Use Data Manager to view limits, monitor usage, and manage...	04/04/2024, 3:08 pm	Lightning
Digital Experiences	SalesforceCMS	Manage content and media for all of your sites.	04/04/2024, 3:08 pm	Lightning
Lightning Usage App	LightningInstrumentation	View Adoption and Usage Metrics for Lightning Experience	04/04/2024, 3:08 pm	Lightning

**9. Search Your App by its name in App Launcher→ just by entering some character, you can see app name with its logo → Click on App→you can see your application appearance in window→ Output.**

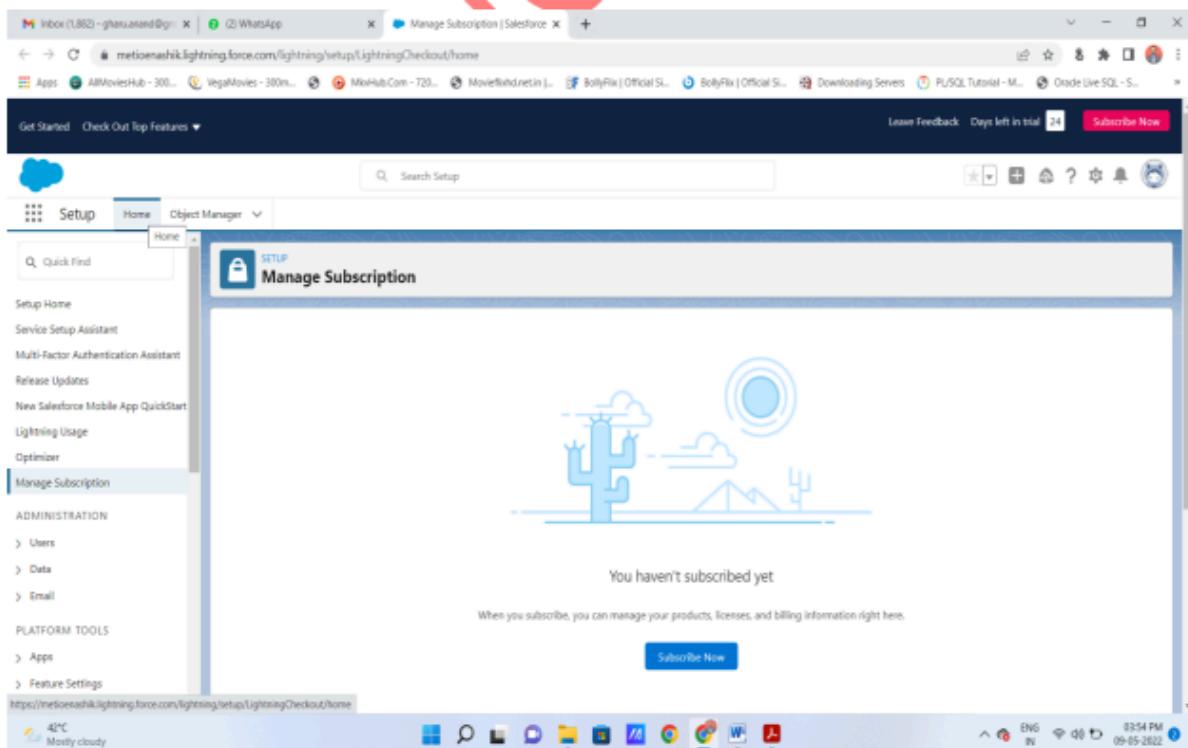
The screenshot shows the 'App Launcher' page. The left sidebar has a search bar with 'apps' typed in. The main area shows a list of recent items:

- CQA-BOT UtilityBar (Lightning Page)
- Calculator (Apex Class)
- MySampleApexClass (Apex Class)
- manisha khadse (User)
- SFDC\_DevConsole (Debug Level)

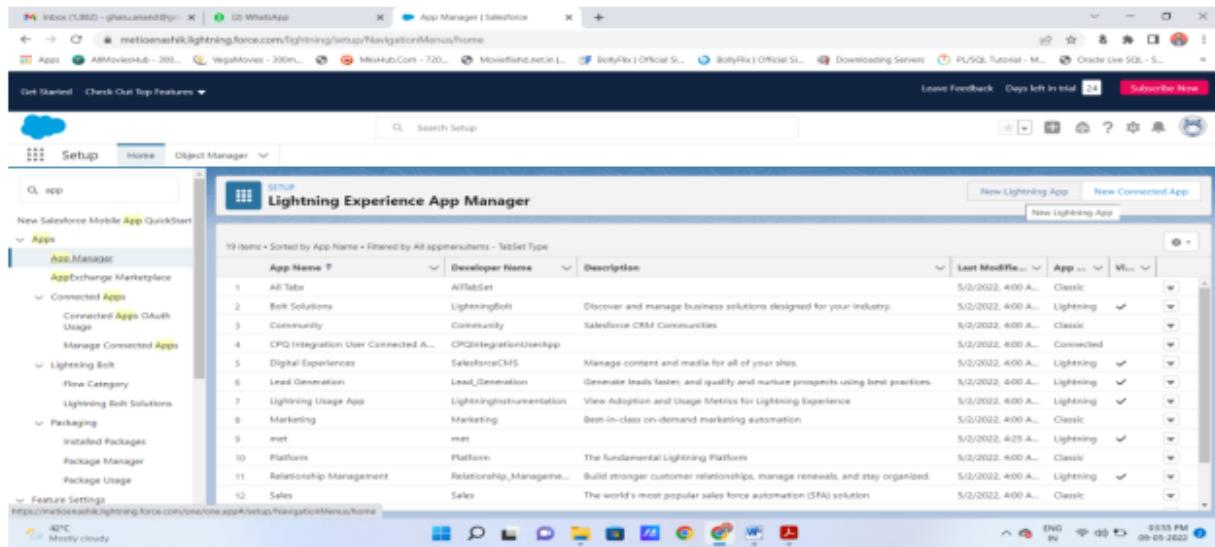


**OR.....another application**

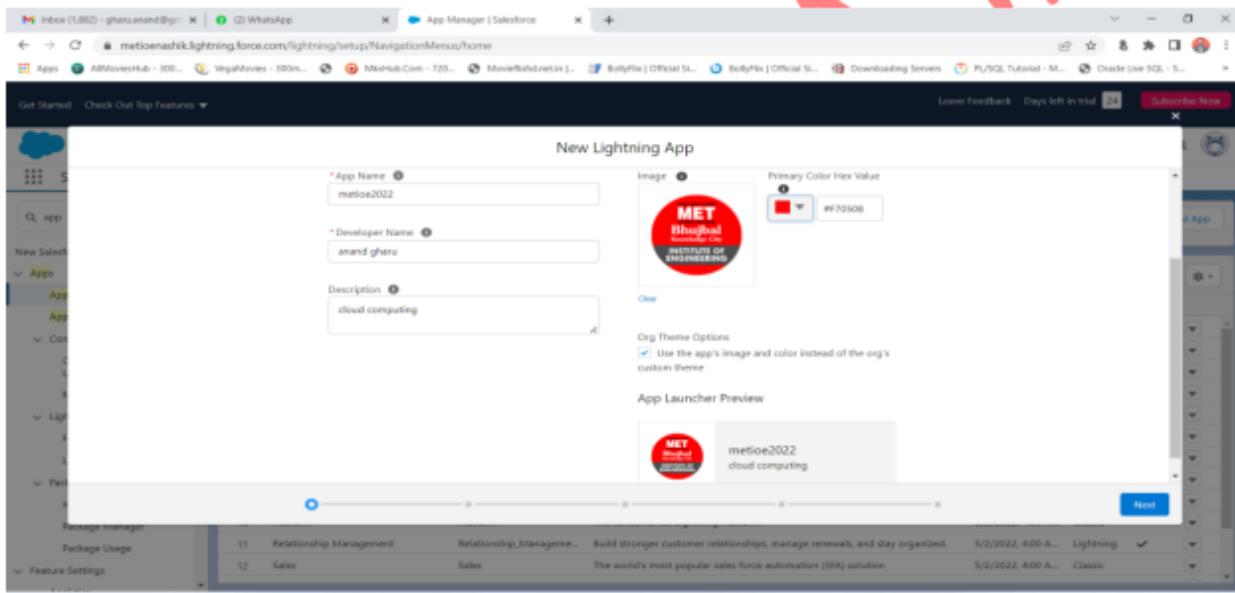
**1. . Open salesforce in Lightning Experience**



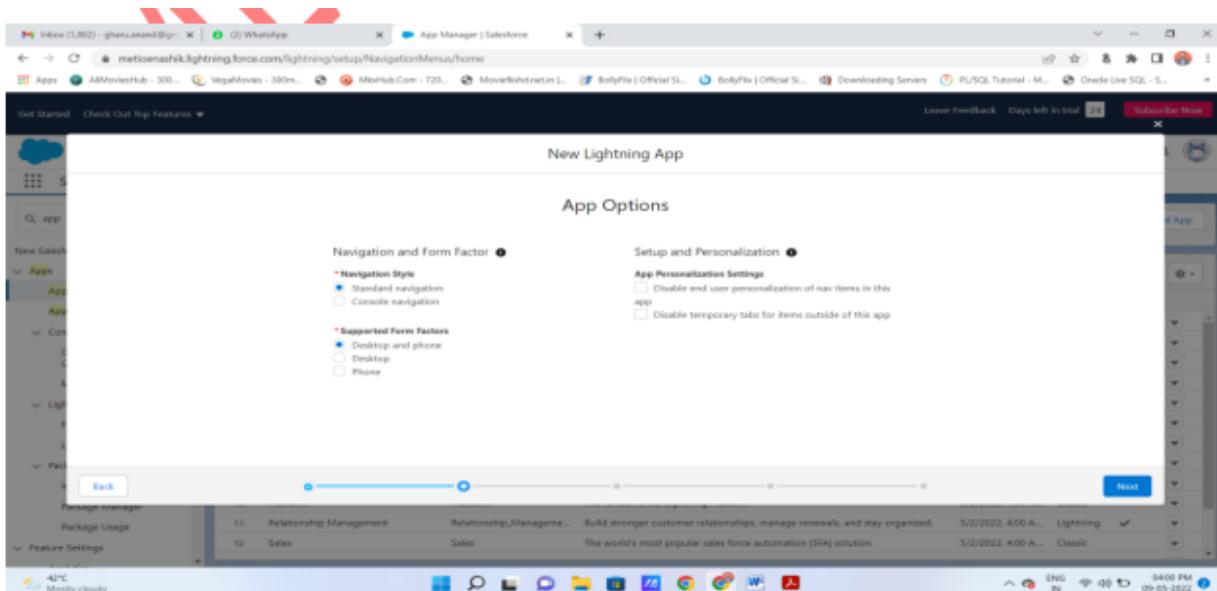
## 2. Open App Launcher View all ( Quick Search App Manager) Click on New Lightning App



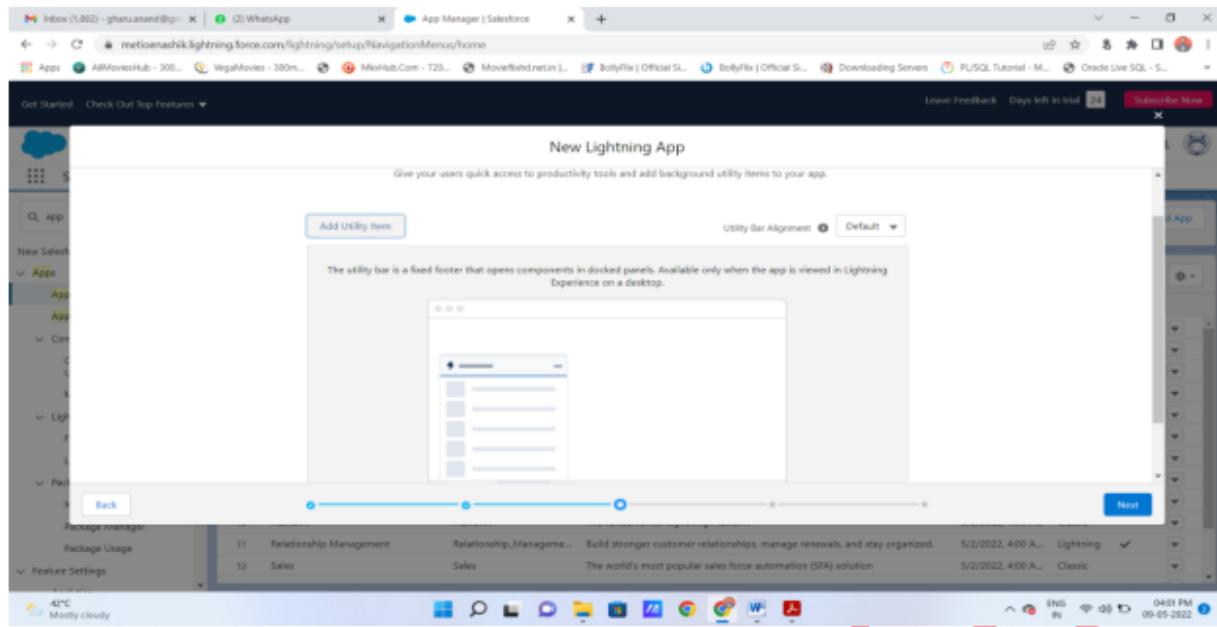
## 3. fill the mention information in new lightning app : then next



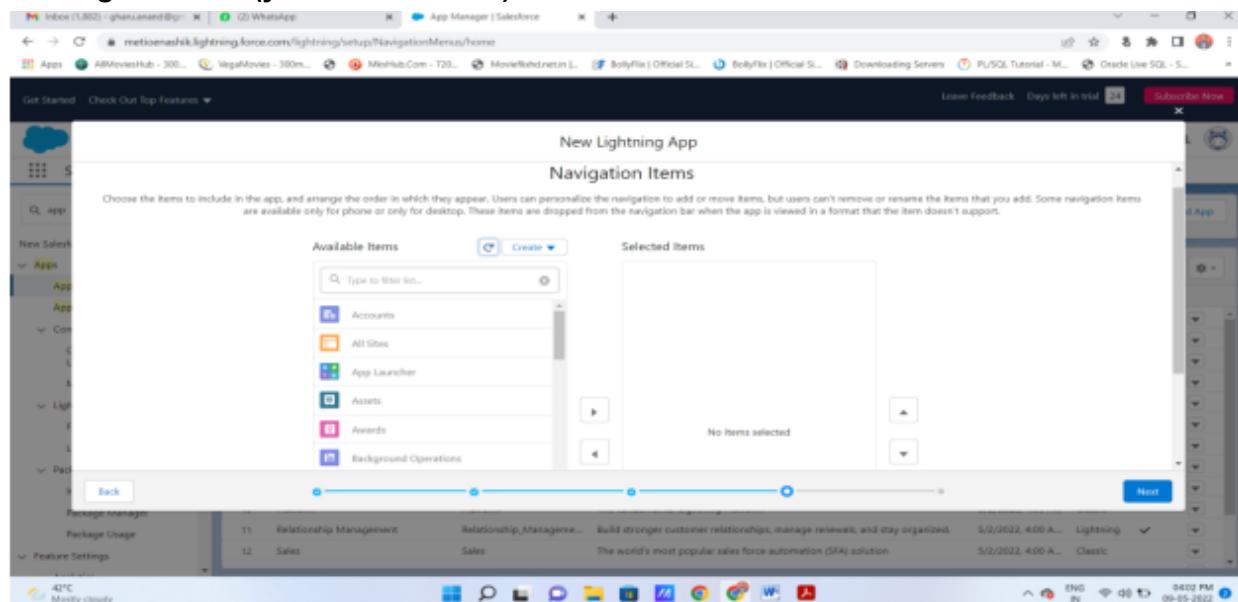
#### 4. App option(optional ) Next

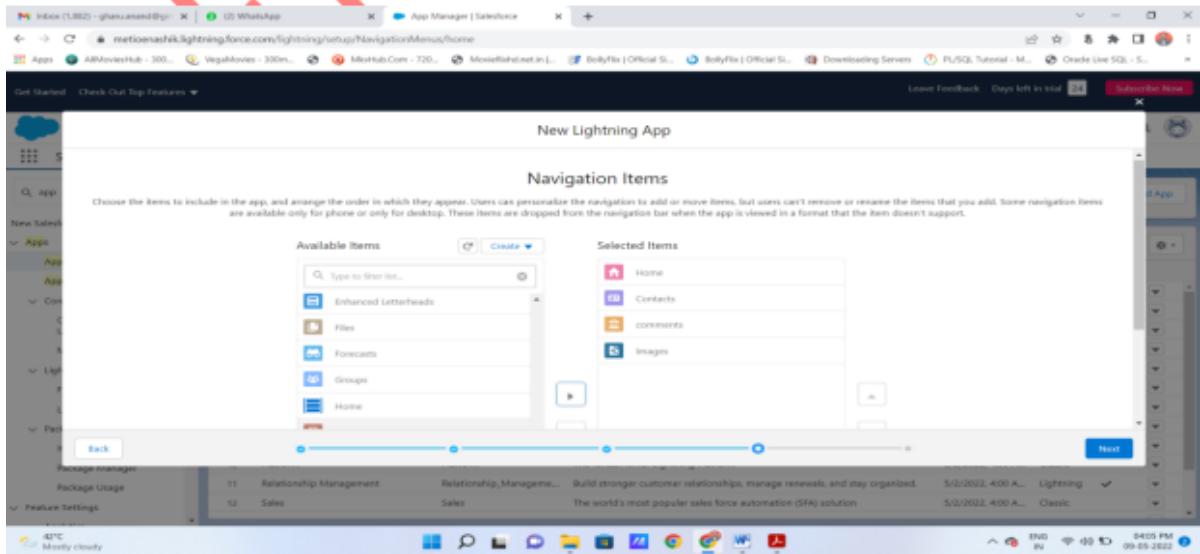


#### 5. Add Utility (optional) Next .

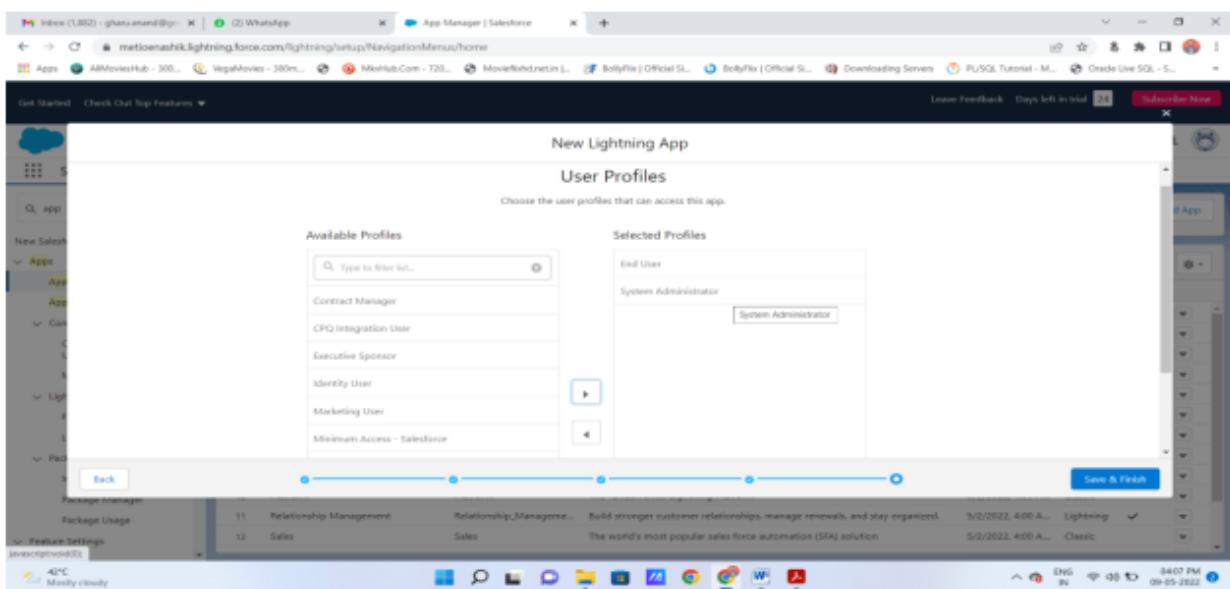


## 6. Navigation Item(you can add item) Next

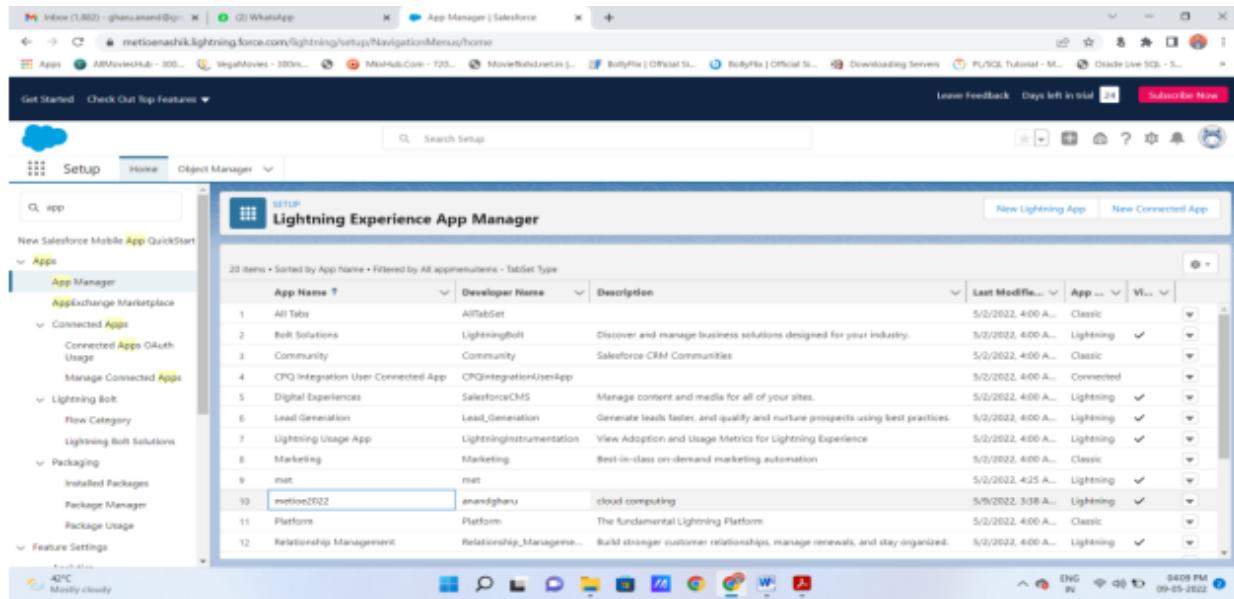




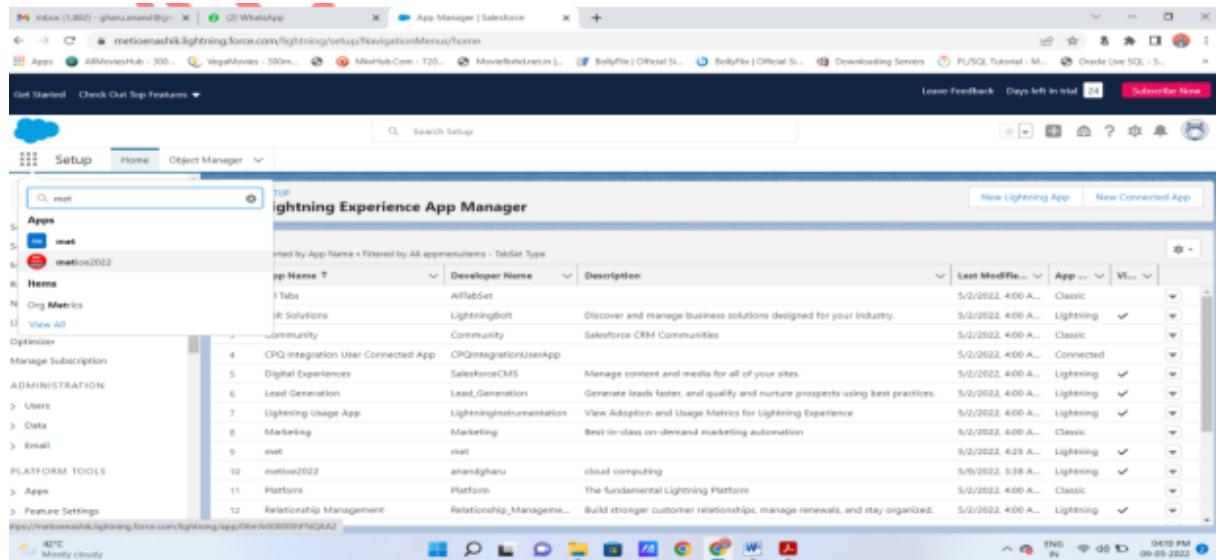
## 7. User Profile(use the user profile that can access this app) Save and Finish



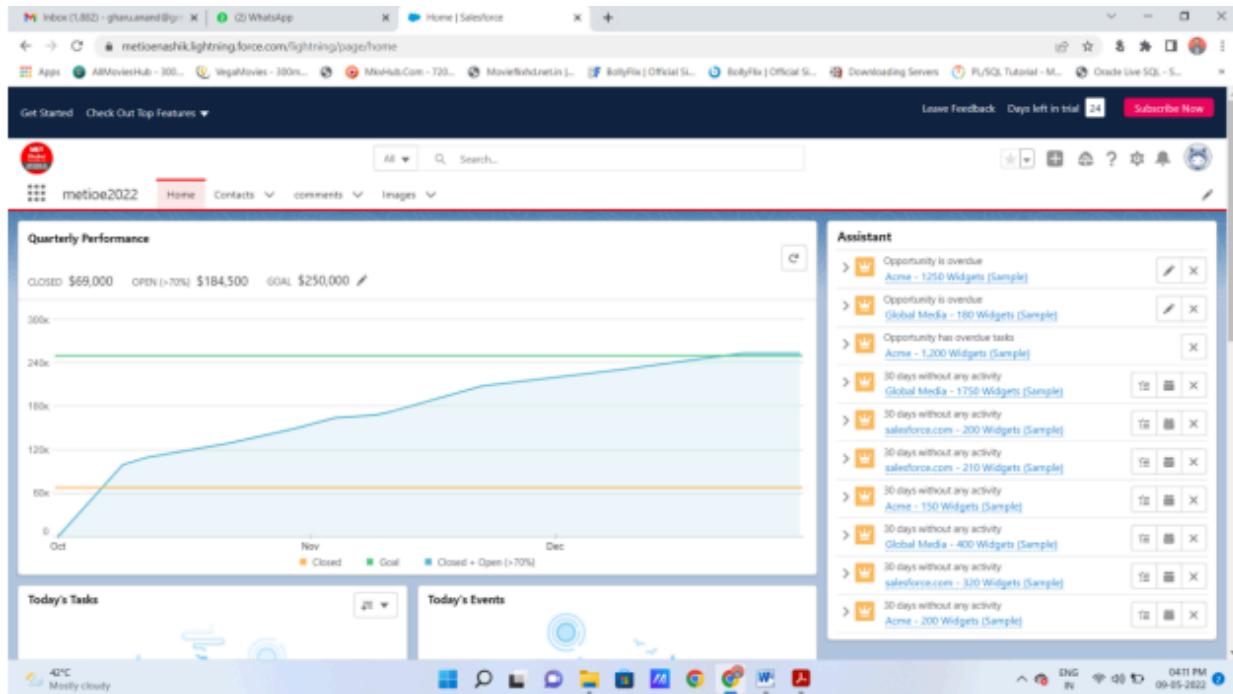
## 8. You will get Lightning Experience App Manager Window(here, you can see your application)



**9. Search Your App by its name in App Launcher just by entering some character, you can see app name with its logo Click on App you can see your application appearance in window Output.**



**Output : You can see your application here....**



## 10. You can add new contact in your application :

The screenshot shows the 'New Contact' form in the Salesforce application. The title bar says 'New Contact | Salesforce'. The form is titled 'Contact Information' and contains the following fields:

Field	Value
Name	Adi ANAND GHANUAMAND
Middle Name	HANVILAL
Last Name	GHANU
Title	Cloud Computing
Account Name	metioe
Contact Owner	ANAND GHANU
Reports To	(Search Contacts...)
Department	Computer Engineering

At the bottom of the form are three buttons: 'Cancel', 'Save & New', and 'Save'. The background shows the Salesforce interface with a 'Recently Viewed' section on the left.

Get Started Check Out Top Features

metiose2022 Home Contacts comments Images

Contact Mr. ANAND NANDLAL GHARU ANG

Title Cloud Computing Account Name metiose Phone (02) Email ghanu.anand@gmail.com Contact Owner ANAND GHARU

Related Details News

We found no potential duplicates of this Contact.

Related List Quick Links: Opportunities (0) Related Accounts (1) Files (0) Notes (0) Cases (0)

Opportunities (0)

Related Accounts (1)

metiose Direct Refer:

View All

Activity Chatter

Log a Call New Task New Event Email

Recap your call... Add

Filters: All time • All activities • All types Refresh • Expand All • View All

Upcoming & Overview

No next steps. To get things moving, add a task or set up a meeting.

No past activity. Past meetings and tasks marked as done show up here.

42°C Mostly cloudy

## 11. You can add images also :

Get Started Check Out Top Features

metiose2022 Home Contacts comments Images

Images Recently Viewed

0 items • Updated a few seconds ago

Name Last Modified By Last Modified Date

You haven't viewed any Images recently. Try switching list views.

https://metashik.in/images/logo/metCircularLogo.png

42°C Mostly cloudy

Index (1,000) - ghanu@metiose2022 10 WhatsApp New Image | Salesforce

metiose2022 Home Contacts

Images Recently Viewed

0 items • Updated 2 minutes ago

Name

Information

Name MET'S ITC INSTITUTE OF ENGINEERING NASHIK

Image Orientation UP-2

Title CLOUD COMPUTING

Accessibility Text met

URL https://metashik.in/images/logo/metCircularLogo.png

Active

Category Logos

Image Type training

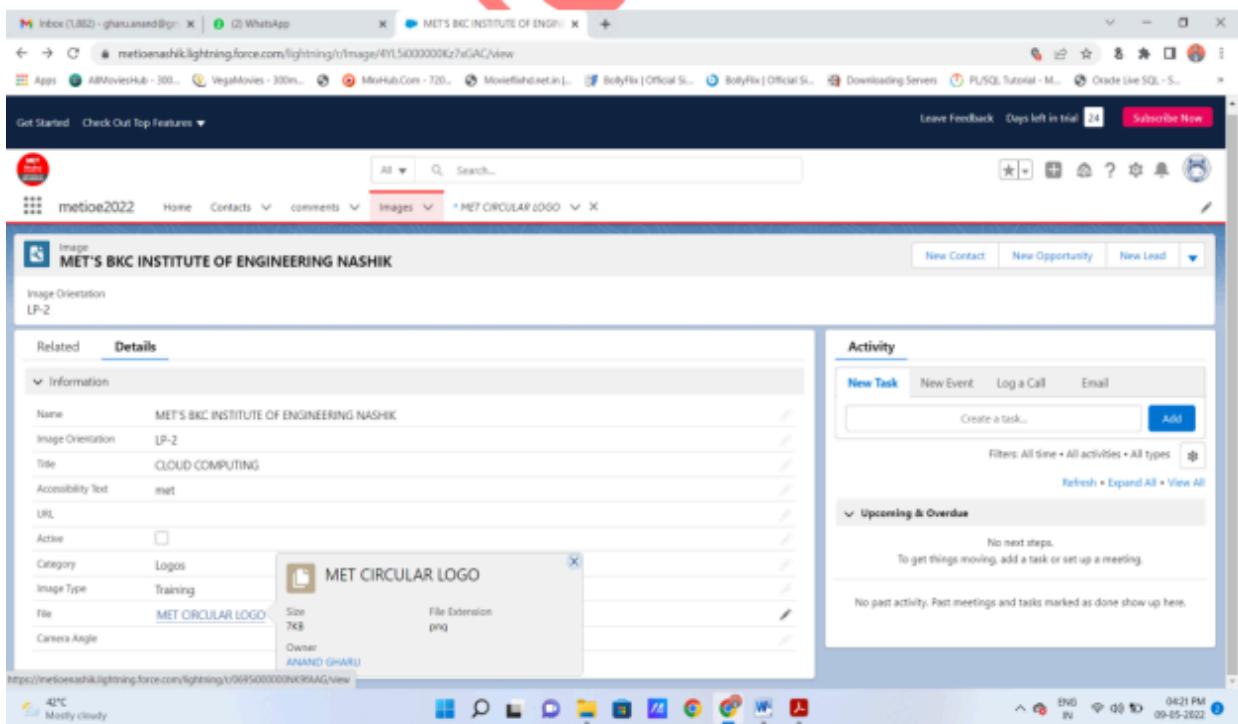
File MET CIRCULAR LOGO

Camera Angle

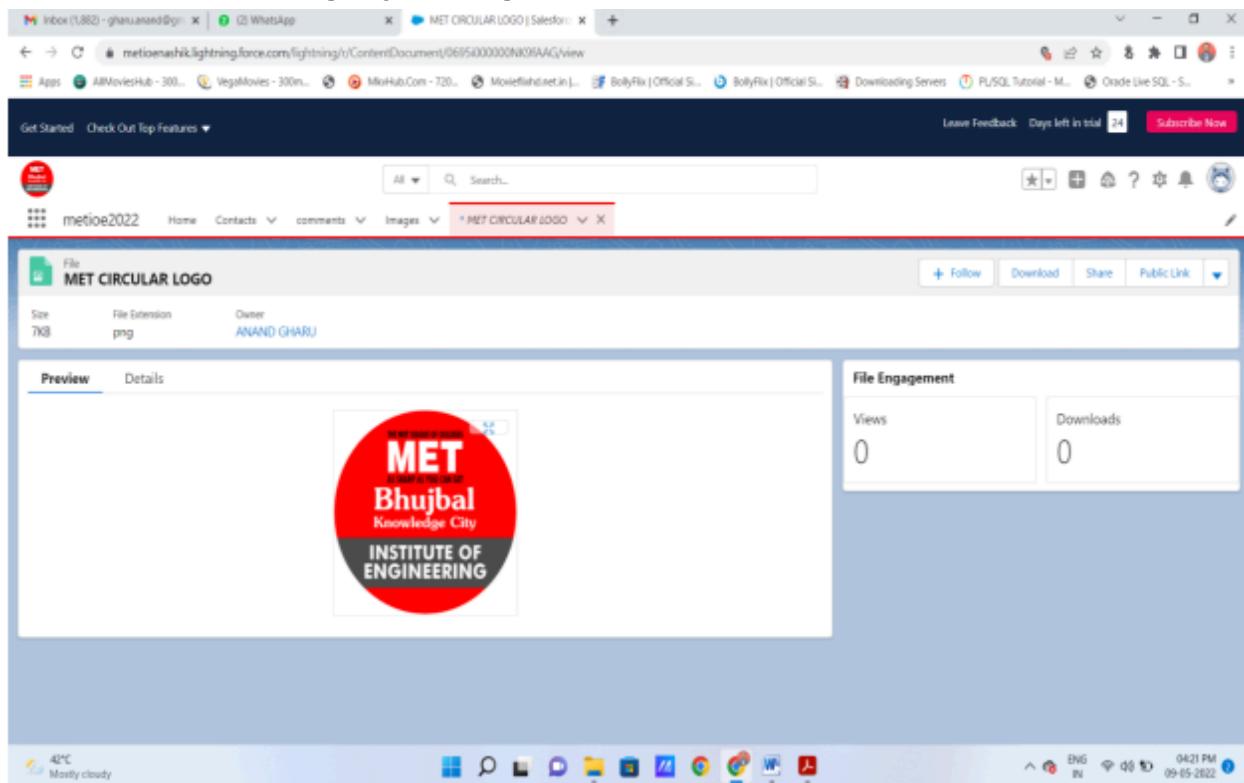
Cancel Save & New Done

42°C Mostly cloudy

You can see here that image is added as MET Circular logo(its image file name) :



You can see actual image by clicking on link :



You mini project application get created

**Note : you can add navigations, utilities , profiles as per your requirements for your mini project application**

# **Practical No : 05**

## **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 file on/from cloud in encrypted form.

## **Created By :Ms. Manisha Vilas Khadse**

**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

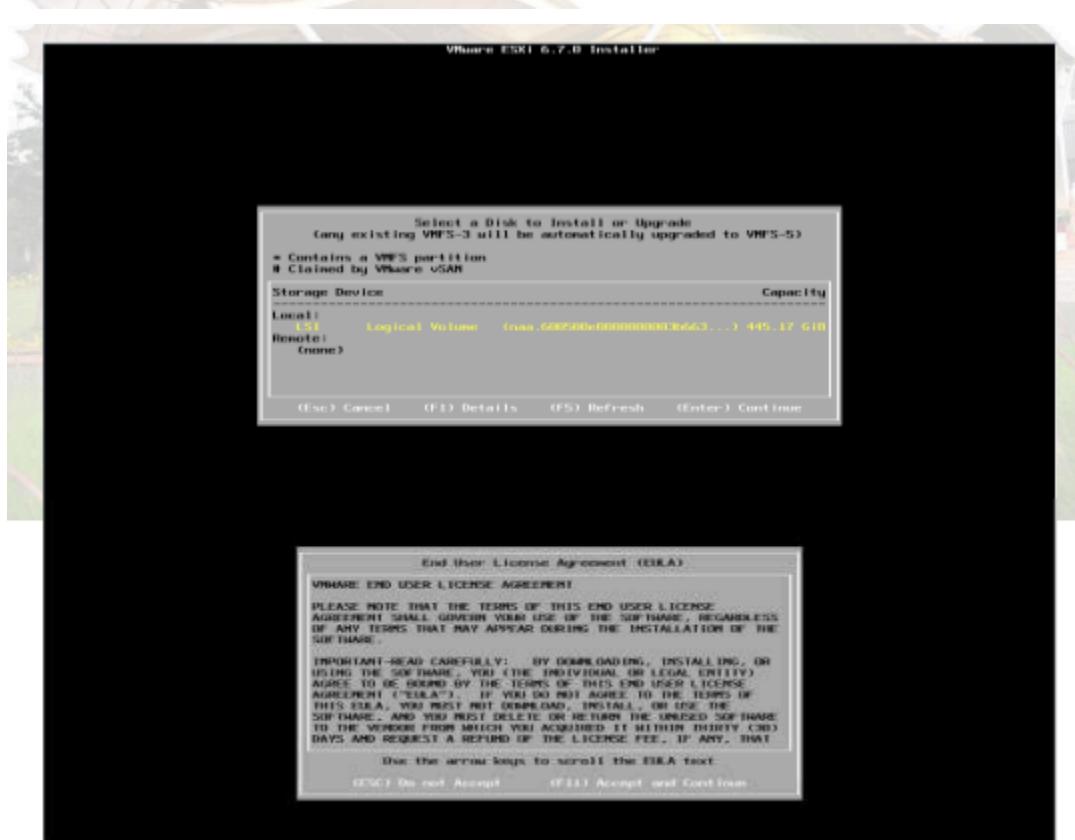
Host/NodeESXi installation:- ESXiHardwareRequirements:-

- ESXi6.7 requires a host machine with at least two CPU cores.
- ESXi6.7 supports 64-bit x86 processors
- ESXi6.7 requires the NX/XD bit to be enabled for the CPU in the BIOS.
- ESXi6.7 requires a minimum of 4 GB of physical RAM. It is recommended to provide at least 8 GB of RAM to run virtual machines in typical production environments.
- To support 64-bit virtual machines, support for hardware virtualization (Intel VT-x or AMDRVI) must be enabled on x64 CPUs.
- One or more Gigabit or faster Ethernet controllers. For a list of supported network adapter models.
- SCSI disk or local, non-network, RAID LUN with unpartitioned space for the virtual machines. For Serial ATA (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 because they are seen as remote.



## **ESXiInstaller:**

**Accept Agreement:**



**Select storage :**



**Select Keyboard Layout :**

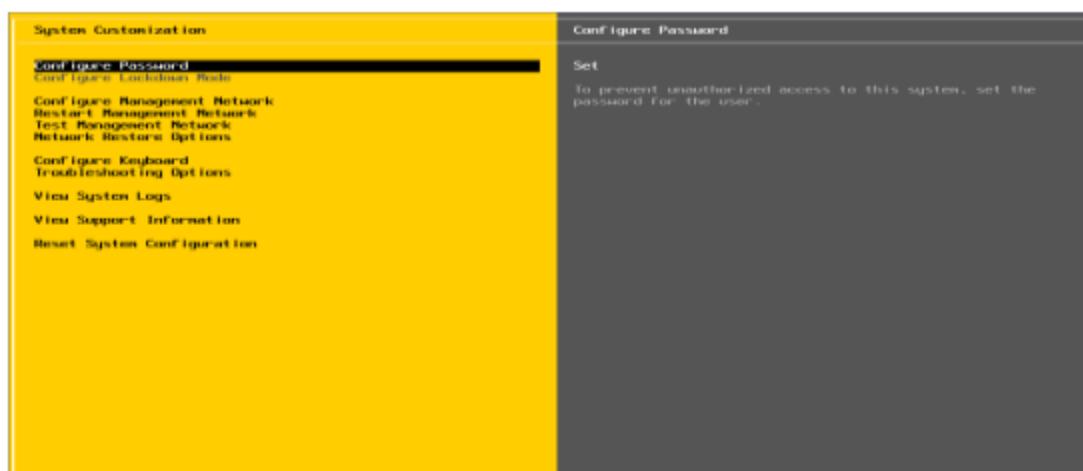


**Set NodeESXi Root Password :**

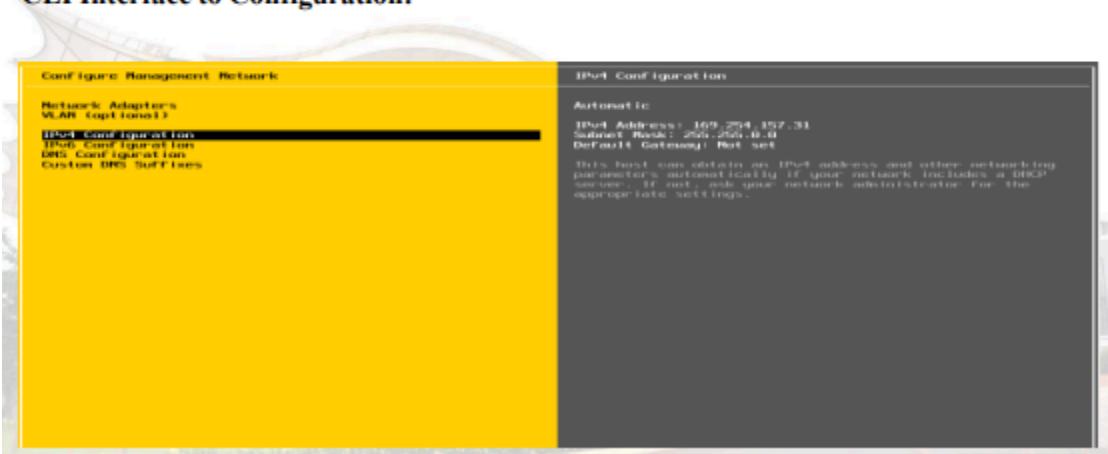
---



## Installation complete (Reboot)CLII interface to configuration



## CLI Interface to Configuration:



## Configure Management Network



## Set IPV4



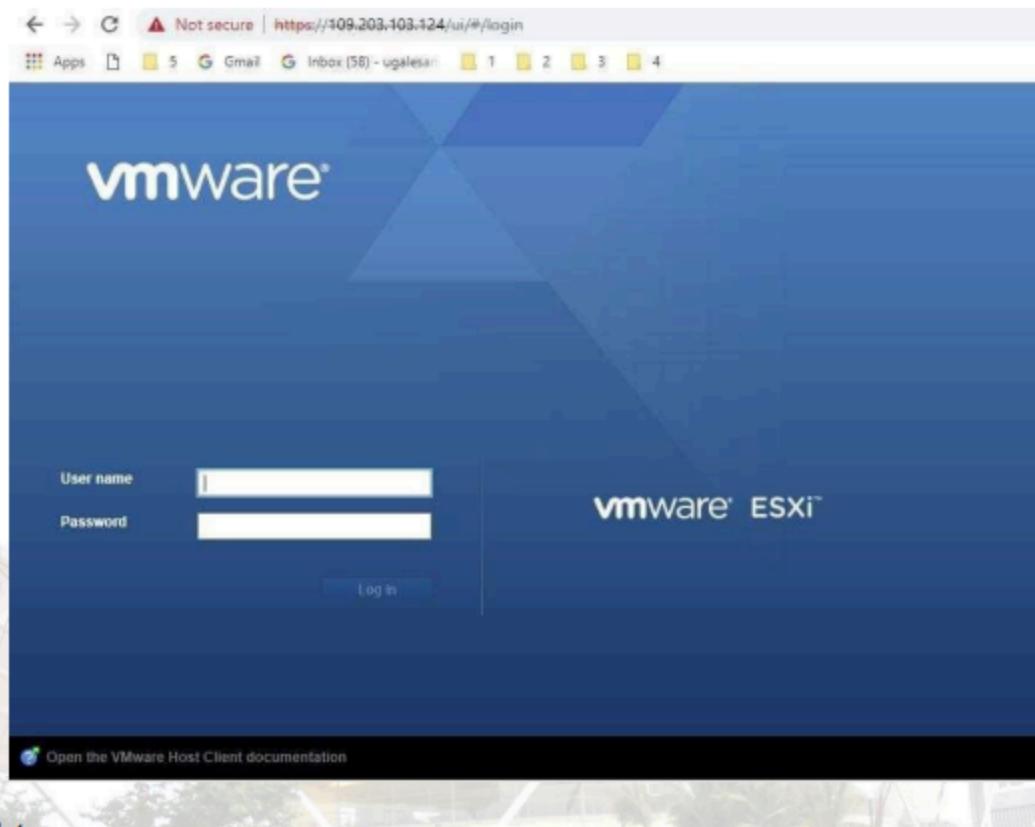
## Set DNServer :

**Restart Management Network**

## Restart Management Network

System Customization	Restart Management Network
<a href="#">Configure Password</a>	
<a href="#">Configure Lifecycle Mode</a>	
<a href="#">Configure Management Network</a>	
<a href="#">Restart Management Network</a>	<p>Restarting the management network interface may be required to restore networking or to renew a DHCP lease.</p>
<a href="#">Test Management Network</a>	<p>Restarting the management network will result in a brief network outage that may temporarily affect running virtual machines.</p>
<a href="#">Network Restore Options</a>	
<a href="#">Configure Keyboard</a>	
<a href="#">Troubleshooting Options</a>	<p>Note: If a renewed DHCP lease results in a new network identity (e.g., IP address or hostname), remote management software will be disconnected.</p>
<a href="#">View System Logs</a>	
<a href="#">View Support Information</a>	
<a href="#">Reset System Configuration</a>	

## GUIAccess :



## ClusterSetup

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

## VCenter Access:

A screenshot of the vSphere Client interface. The title bar says 'vSphere Client' and the URL is https://172.14.5.79/ui/#/clusters/vsphere.core.inventory.ws. The main pane shows a cluster summary with 0 virtual machines and 0 hosts. Below this, there are sections for 'Custom Attributes' and 'Recent Tasks'. The 'Recent Tasks' table lists two entries: 'Remove datacenter' and 'Create datacenter', both completed successfully. The 'Custom Attributes' section shows CPU, Memory, Storage, and Disk usage statistics.

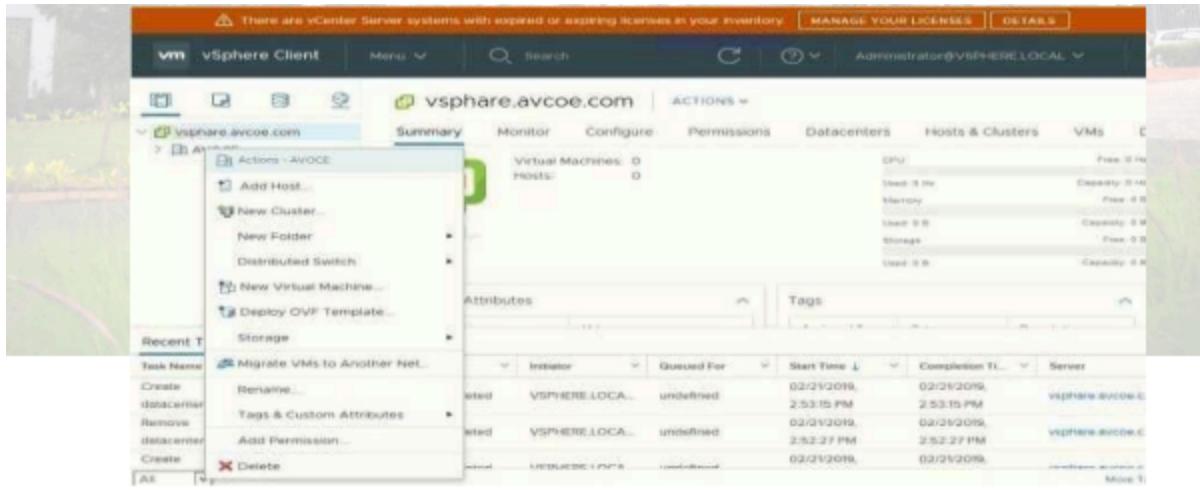
## Create Datacenter:

The screenshot shows the vSphere Client interface on a web browser. The URL is [vsphere.avcoe.com](https://vsphere.avcoe.com). The main menu bar includes 'VM', 'vSphere Client', 'Menu', 'Search', 'Administrator@VSPHERE LOCAL'. The top status bar indicates 'There are vCenter Server systems with expired or missing licenses in your inventory.' with links to 'MANAGE YOUR LICENSES' and 'DETAILS'.

The main content area has tabs: 'Summary', 'Monitor', 'Configure', 'Permissions', 'Datacenters', 'Hosts & Clusters', 'VMs', 'Data'. A context menu is open over the 'Datacenters' tab, showing options: 'New Datacenter...', 'New Folder', 'Distributed Switch', 'Assign License', and 'Tags & Custom Attributes'. Below this, there's a 'Recent Tasks' table and an 'Alarms' section.

A modal dialog box titled 'New Datacenter' is open in the center. It has fields for 'Name' (set to 'AVCOE') and 'Location' (set to 'vsphere.avcoe.com'). At the bottom right of the dialog are 'CANCEL' and 'OK' buttons.

The background shows a summary view with resource usage statistics: CPU (Used: 0.0%, Capacity: 0.0%, Free: 100%), Memory (Used: 0.0%, Capacity: 0.0%, Free: 100%), Storage (Used: 0.0%, Capacity: 0.0%, Free: 100%), and Network (Used: 0.0%, Capacity: 0.0%, Free: 100%).



**Create cluster :**

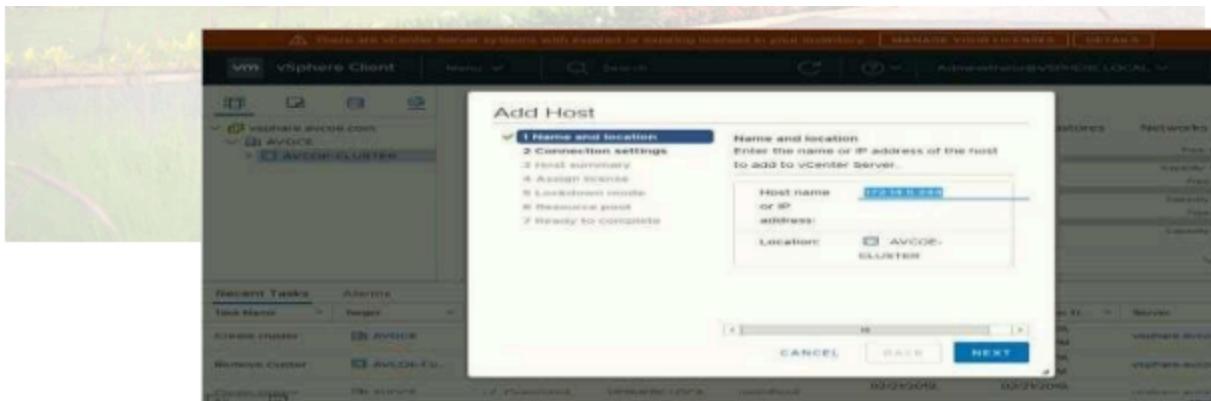
**Assign cluster name :**



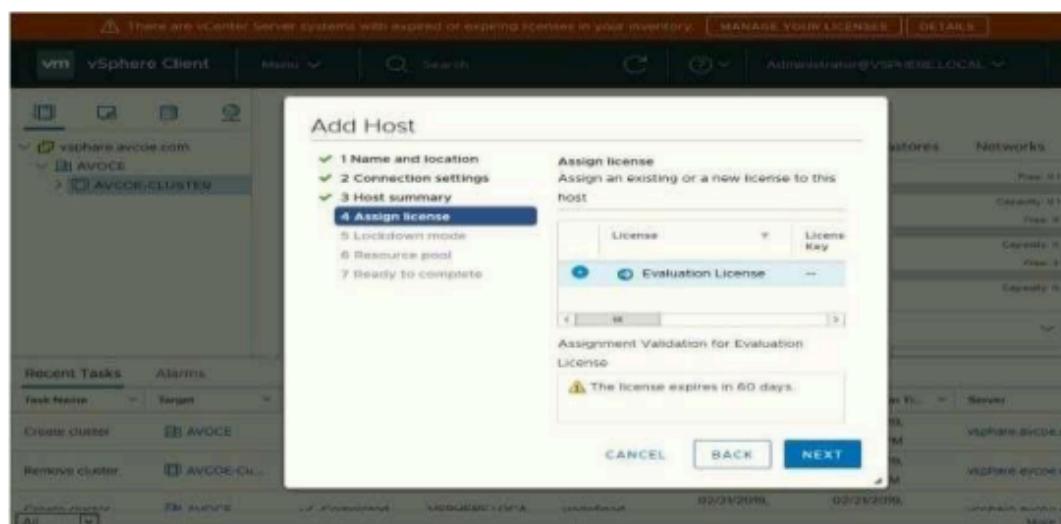
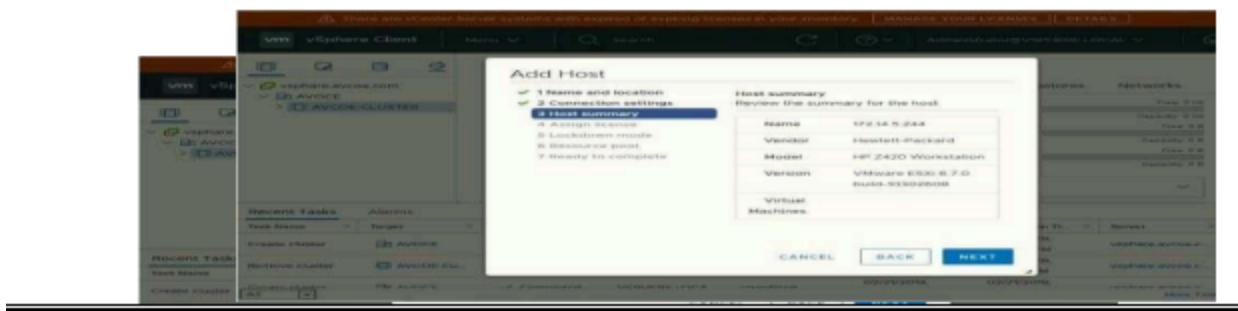
### Add host ::

The screenshot shows the vSphere Client interface. At the top, there is a banner with a warning icon stating "There are vCenter Server systems with expired or expiring licenses in your inventory." with buttons for "MANAGE YOUR LICENSES" and "DETAILS". The title bar says "vSphere Client" and "Administrator@VSPHERE LOCAL". The main pane shows a cluster named "AVCOE-CLUSTER". On the left, a sidebar lists "Recent Tasks" with entries for "Create cluster", "Remove cluster", and "vSAN". The main content area has a "Actions" dropdown menu open, showing options like "Add Host...", "New Virtual Machine...", "New Resource Pool...", "Deploy OVF Template...", "New vApp...", "Storage", "Host Profiles", "Edit Default VM Compatibility", "Assign License...", "Settings", "Rename...", "Tags & Custom Attributes", "Delete", and "vSAN". Below the actions menu, there are tabs for "Monitor", "Configure", "Permissions", "Hosts", "VMs", "Datastores", and "Networks". A "vSphere DRS" section is also visible. The bottom of the screen shows a decorative image of a building.

### Add host IP :



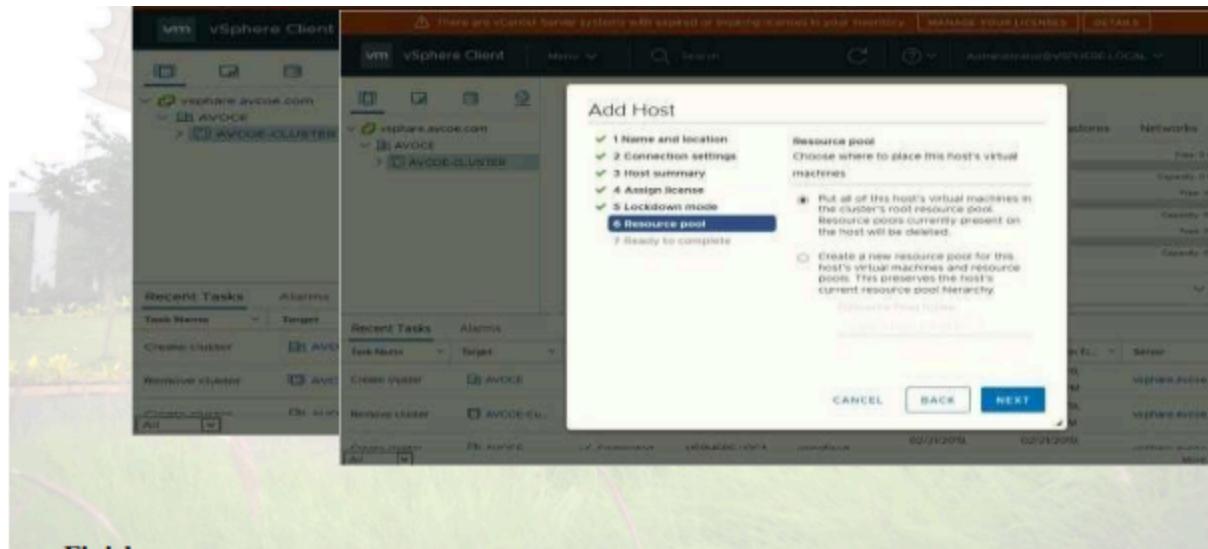
### Enter host credential :



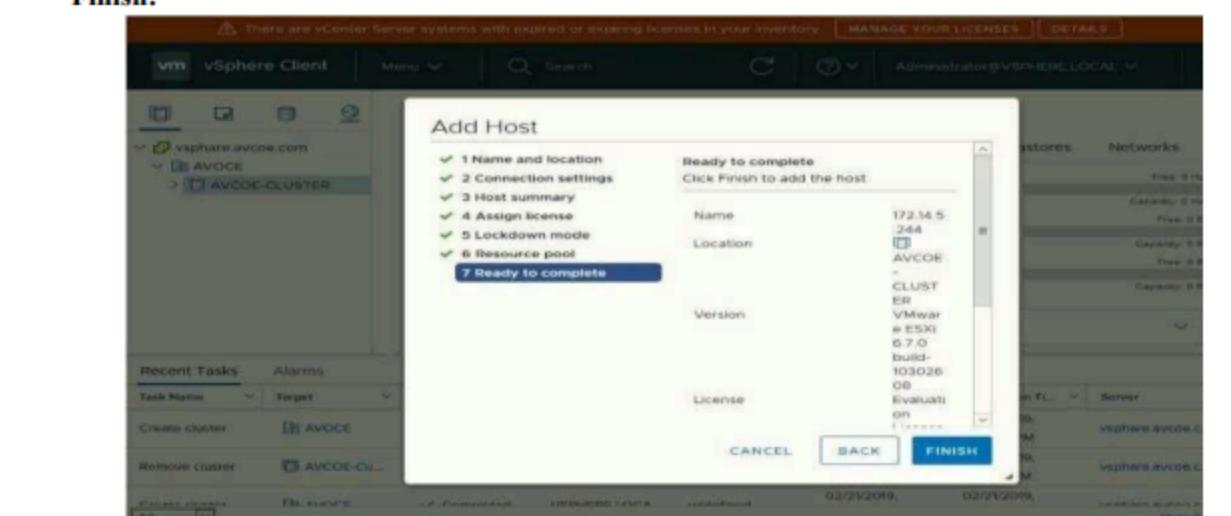
### Hot summary :

### Lock Down mode:

### Add Host In Pool:



**Finish:**



## Host View and View Config:

### Cluster View and Configuration:

**Host View (Top Screenshot):**

Category	Value
Hypervisor	VMware ESXi 6.7.0, 10302608
Model	HP Z420 Workstation
Processor Type	Intel(R) Xeon(R) CPU ES-1607 v2 @ 3.00GHz
Logical Processors	4
NICs	1
Virtual Machines	0
State	Connected
Uptime	0 second

**Cluster View (Bottom Screenshot):**

Category	Value
Total Processors	8
Total vMotion Migrations	0

The screenshot shows the vSphere Client interface with the following details:

- Top Bar:** Shows the title "vSphere Client", a search bar, and a message: "⚠ There are vCenter Server systems with expired or expiring licenses in your inventory." with buttons "MANAGE YOUR LICENSES" and "DETAILS".
- Left Sidebar:** Displays the navigation tree with "vsphere.avcoe.com" expanded, showing "AVCOE" and "AVCOE-CLUSTER" which contains hosts "172.14.5.244" and "172.14.5.245".
- Main Content Area:**
  - Summary Tab:** Shows cluster statistics: Total Processors: 8, Total vMotion Migrations: 0. It also displays resource usage for CPU, Memory, and Storage.
  - Related Objects:** A list of objects related to the cluster.
  - vSphere DRS:** A section for managing Distributed Resource Scheduler.
- Bottom Section:** A table titled "Recent Tasks" listing completed tasks:

Task Name	Target	Status	Initiator	Queued For	Start Time	Completion T...	Server
Configuring vSphere HA	172.14.5.245	✓ Completed	System	156 ms	02/21/2019, 3:04:54 PM	02/21/2019, 3:05:34 PM	vsphere.avcoe.c...
Add host	AVCOE-CL...	✓ Completed	VSPHERE.LOCA...	undefined	02/21/2019, 3:04:48 PM	02/21/2019, 3:04:54 PM	vsphere.avcoe.c...
Configuring	172.14.5.244	✓ Pending	Overhead	8.8 ms	02/21/2019,	02/21/2019,	overhead.vsphere...

**Conclusion:** Like this we have configure VSphere Private Cloud