AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

Department of Computer Engineering

Vision

"Contributing to the welfare of society through technical and quality education."

Mission

- To produce Best Quality Computer Science Professionals by imparting quality training, hands on experience and value education.

- To Strengthen links with Industry through partnerships and collaborative developmental works.

- To attain self-sustainability and overall development through Research, Consultancy and Development Activities.

- To extend technical expertise to other technical Institutions of the region and play a lead role in imparting technical education."

Programme Outcomes (POs)

PO1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. Problem analysis: Identify, formulate, research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

PO6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Specific Outcomes (PSOs)

Computer Engineering graduate will be able to,

PSO1: Project Development: Successfully complete hardware and/or software related system or application projects, using the phases of project development life cycle to meet the requirements of service and product industries; government projects; and automate other engineering stream projects.

PSO2: Domain Expertise: Demonstrate effective application of knowledge gained from different computer domains like, data structures, data bases, operating systems, computer networks, security, parallel programming, in project development, research and higher education.

PSO3: Career Development: Achieve successful Career and Entrepreneurship- The ability to employ modern computer languages, environments, and platforms in creating innovative career paths to be an entrepreneur, and a zest for higher studies.

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय
Accredited by NAAC with "A+" Grade

Lab Manual

Laboratory Practice II

(310258)

Artificial Intelligence
(310253)

Cloud Computing
310254(C)

TE Computer

YEAR: 2022-2023

SEM-II

Prepared by,

Prof. Neha Rai

Prof. S S Kolte

Marking Scheme

Term work: 50 Marks
Practical: 25 Marks

Companion Course: Laboratory Practice II (310258)

Course Objectives:

- ● To understand the concept of Artificial Intelligence (AI) in the form of various Intellectual tasks
- ● To understand Problem Solving using various peculiar search strategies for AI
- ● To understand multi-agent environment in competitive environment
- ● To acquaint with the fundamentals of knowledge and reasoning
- ● To devise plan of action to achieve goals as a critical part of AI
- ● To develop a mind to solve real world problems unconventionally with optimality

Course Outcomes:

After completion of the course, students should be able to

CO1: Identify and apply suitable Intelligent agents for various AI applications

CO2: Build smart system using different informed search / uninformed search or heuristic approaches

CO3: Identify knowledge associated and represent it by ontological engineering to plan a strategy to solve given problem

CO4: Apply the suitable algorithms to solve AI problems

CO5: Implement ideas underlying modern logical inference systems

CO6: Represent complex problems with expressive yet carefully constrained language of representation

Artificial Intelligence
INDEX

Group C				
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			

Artificial Intelligence
Group A

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.

BFS is one of the traversing algorithm used in graphs. This algorithm is implemented using a queue data structure. In this algorithm, the main focus is on the vertices of the graph. Select a starting node or vertex at first, mark the starting node or vertex as visited and store it in a queue. Then visit the vertices or nodes which are adjacent to the starting node, mark them as visited and store these vertices or nodes in a queue. Repeat this process until all the nodes or vertices are completely visited.

Advantages of BFS

1. It can be useful in order to find whether the graph has connected components or not.
2. It always finds or returns the shortest path if there is more than one path between two vertices.

Disadvantages of BFS

1. The execution time of this algorithm is very slow because the time complexity of this algorithm is exponential.
2. This algorithm is not useful when large graphs are used.

Explanation:

1. Create a graph.
2. Initialize a starting node.
3. Send the graph and initial node as parameters to the bfs function.

4. Mark the initial node as visited and push it into the queue.
5. Explore the initial node and add its neighbours to the queue and remove the initial node from the queue.
6. Check if the neighbours node of a neighbouring node is already visited.
7. If not, visit the neighbouring node neighbours and mark them as visited.
8. Repeat this process until all the nodes in a graph are visited and the queue becomes empty.

Output:

```
['A', 'B', 'C', 'E', 'D', 'F', 'G']
```

Implementation of BFS in Python (Breadth First Search)

Source Code I: BFS in Python

```
graph = {'A': ['B', 'C', 'E'],
'B': ['A','D', 'E'],
'C': ['A', 'F', 'G'],
'D': ['B'],
'E': ['A', 'B','D'],
'F': ['C'],
'G': ['C']}
def bfs(graph, initial):
visited = []
queue = [initial]
while queue:
node  =  queue.pop(0)
if node not in visited:
visited.append(node)
neighbours = graph[node]
```

```
for neighbour in neighbours:
queue.append(neighbour)
return visited
print(bfs(graph,'A'))
```

Source Code II # BFS algorithm in Python

```
import collections


# BFS  algorithm
def bfs(graph, root):

    visited, queue = set(), collections.deque([root])
    visited.add(root)

    while queue:

        # Dequeue a vertex from queue
        vertex = queue.popleft()
        print(str(vertex) + " ", end="")

        # If not visited, mark it as visited, and
        # enqueue it
        for neighbour in graph[vertex]:
            if neighbour not in visited:
                visited.add(neighbour)
                queue.append(neighbour)



if__name__== '_main_':
    graph = {0: [1, 2], 1: [2], 2: [3], 3: [1, 2]}
```

print

Depth First Search (DFS)

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:

1. Visited
2. Not Visited

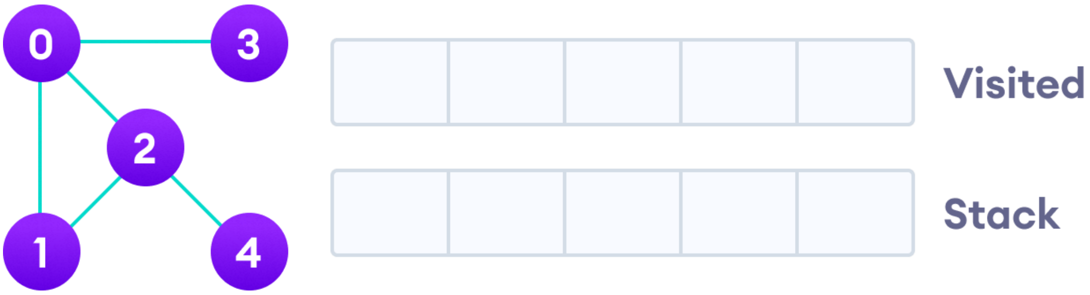
The purpose of the algorithm is to mark each vertex as visited while avoiding cycles.

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.

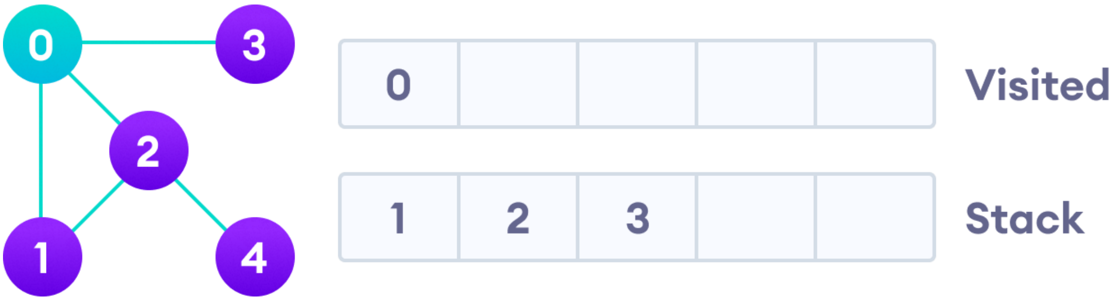
Depth First Search Example

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



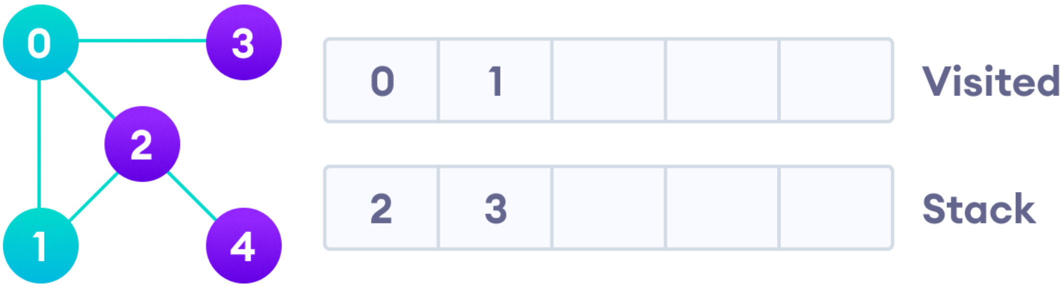
Undirected graph with 5 vertices

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.



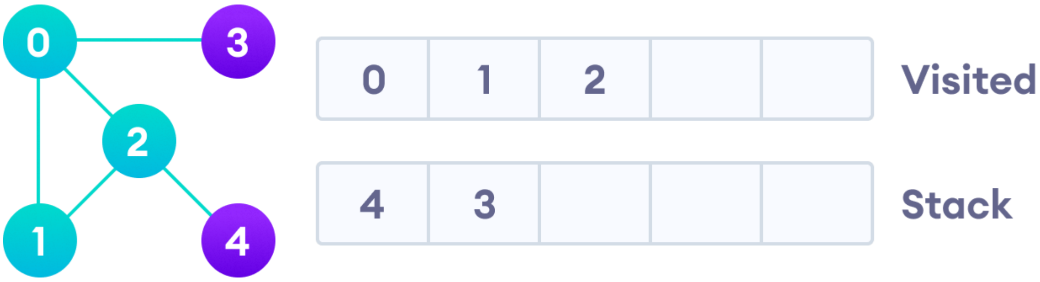
Visit the element and put it in the visited list

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.

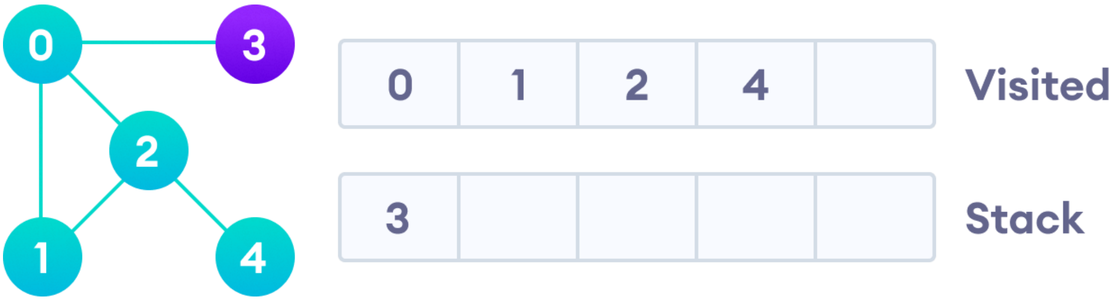


Visit the element at the top of stack

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

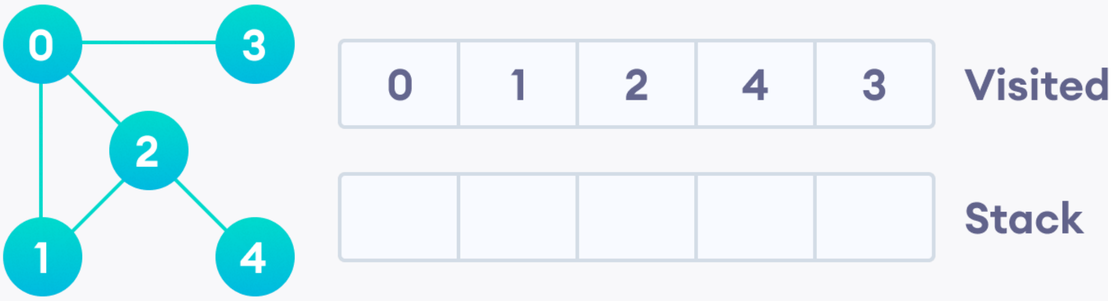


Vertex 2 has an unvisited adjacent vertex in 4, so we add that to the top of the stack and visit it.



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.



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.

Complexity of Depth First Search

The time complexity of the DFS algorithm is represented in the form of O(V + E), where V is the number of nodes and E is the number of edges.

The space complexity of the algorithm is O(V).

Application of DFS Algorithm

1. For finding the path
2. To test if the graph is bipartite
3. For finding the strongly connected components of a graph
4. For detecting cycles in a graph

// DFS algorithm in Java

```
import java.util.*;


class Graph {
  private LinkedList<Integer> adjLists[];
  private boolean visited[];


  // Graph creation
  Graph(int vertices) {
    adjLists = new LinkedList[vertices];
    visited = new boolean[vertices];


    for (int i = 0; i < vertices; i++)
    adjLists[i] = new LinkedList<Integer>();
  }

  // Add edges
  void addEdge(int src, int dest) {
    adjLists[src].add(dest);
  }

  // DFS algorithm
  void DFS(int vertex) {
    visited[vertex] = true;
    System.out.print(vertex + " ");

    Iterator<Integer> ite = adjLists[vertex].listIterator();
    while (ite.hasNext()) {
      int adj = ite.next();
      if (!visited[adj])
      DFS(adj);
    }
  }

  public static void main(String args[]) {
    Graph g = new Graph(4);

    g.addEdge(0, 1);
    g.addEdge(0, 2);
    g.addEdge(1, 2);
    g.addEdge(2, 3);

    System.out.println("Following is Depth First Traversal");

    g.DFS(2);
  }
}
```

# DFS algorithm in Python

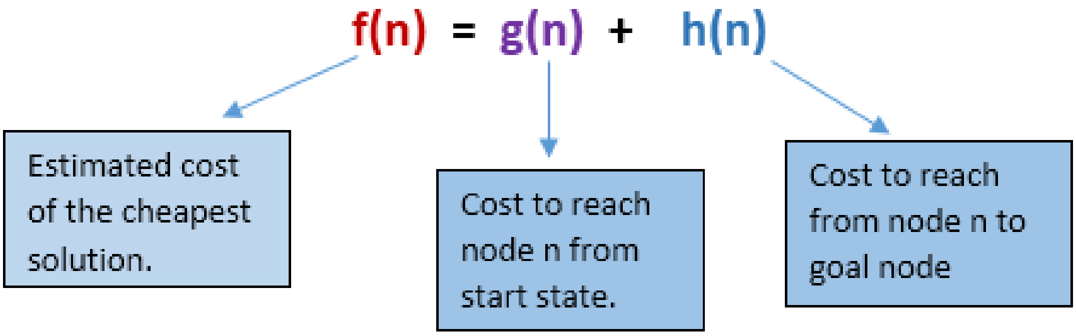
```
# DFS algorithm
def dfs(graph, start, visited=None):
    if visited is None:
        visited = set()
    visited.add(start)

    print(start)

    for next in graph[start] - visited:
        dfs(graph, next, visited)
    return visited



graph = {'0': set(['1', '2']),
         '1': set(['0', '3', '4']),
         '2': set(['0']),
         '3': set(['1']),
         '4': set(['2', '3'])}

dfs(graph, '0')
```

2.Implement A star Algorithm for any game search problem.

A* Search

A* search is the most commonly known form of best-first search. It uses heuristic function h(n), and cost to reach the node n from the start state g(n). It has combined features of UCS and greedy best-first search, by which it solve the problem efficiently. A* search algorithm finds the shortest path through the search space using the heuristic function. This search algorithm expands less search tree and provides optimal result faster. A* algorithm is similar to UCS except that it uses g(n)+h(n) instead of g(n).

In A* search algorithm, we use search heuristic as well as the cost to reach the node. Hence we can combine both costs as following, and this sum is called as a **fitness number**.


f(n) = g(n) + h(n)
Estimated cost of the cheapest solution.
Cost to reach node n from start state.
Cost to reach from node n to goal node

Algorithm of A* search:

Step1: Place the starting node in the OPEN list.

Step 2: Check if the OPEN list is empty or not, if the list is empty then return failure and stops.

Step 3: Select the node from the OPEN list which has the smallest value of evaluation function (g+h), if node n is goal node then return success and stop, otherwise

Step 4: Expand node n and generate all of its successors, and put n into the closed list. For each successor n', check whether n' is already in the OPEN or CLOSED list, if not then compute evaluation function for n' and place into Open list.

Step 5: Else if node n' is already in OPEN and CLOSED, then it should be attached to the back pointer which reflects the lowest g(n') value.

Step 6: Return to **Step 2**.

Advantages:

1. A* search algorithm is the best algorithm than other search algorithms.
2. A* search algorithm is optimal and complete.
3. This algorithm can solve very complex problems.

Disadvantages:

1. It does not always produce the shortest path as it mostly based on heuristics and approximation.
2. A* search algorithm has some complexity issues.
3. The main drawback of A* is memory requirement as it keeps all generated nodes in the memory, so it is not practical for various large-scale problems.

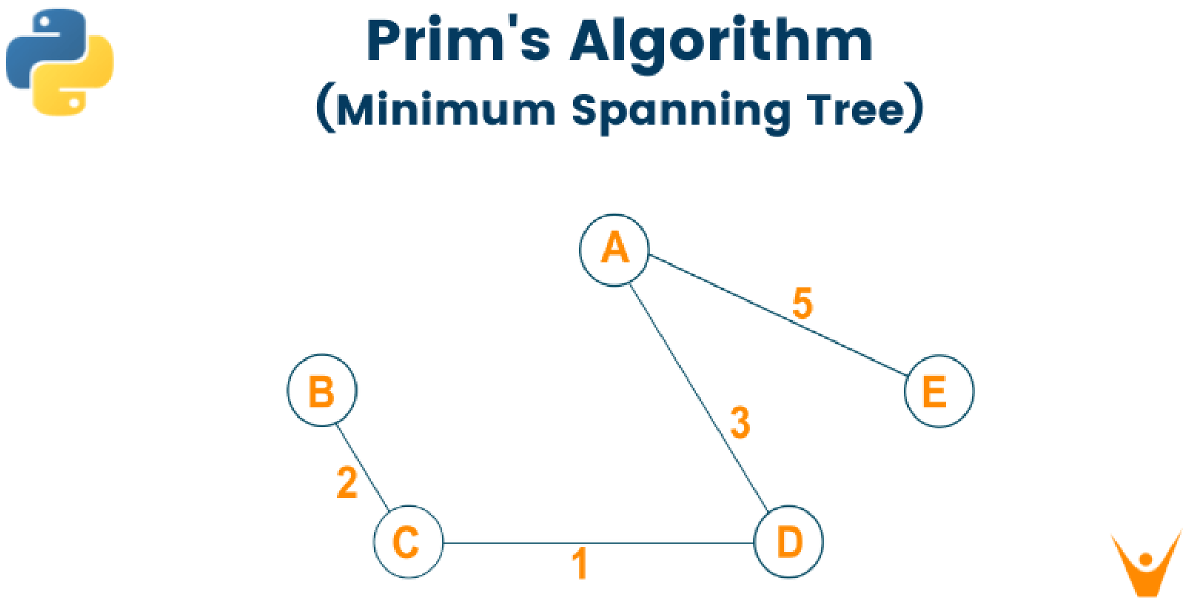
#AIM: Implement A* search.

```
dict_hn={'Arad':336,'Bucharest':0,'Craiova':160,'Drobeta':242,'Eforie':161,
      'Fagaras':176,'Giurgiu':77,'Hirsova':151,'Iasi':226,'Lugoj':244,
      'Mehadia':241,'Neamt':234,'Oradea':380,'Pitesti':100,'Rimnicu':193,
      'Sibiu':253,'Timisoara':329,'Urziceni':80,'Vaslui':199,'Zerind':374}


dict_gn=dict(
Arad=dict(Zerind=75,Timisoara=118,Sibiu=140),
Bucharest=dict(Urziceni=85,Giurgiu=90,Pitesti=101,Fagaras=211),
Craiova=dict(Drobeta=120,Pitesti=138,Rimnicu=146),
Drobeta=dict(Mehadia=75,Craiova=120),
Eforie=dict(Hirsova=86),
Fagaras=dict(Sibiu=99,Bucharest=211),
Giurgiu=dict(Bucharest=90),
Hirsova=dict(Eforie=86,Urziceni=98),
Iasi=dict(Neamt=87,Vaslui=92),
Lugoj=dict(Mehadia=70,Timisoara=111),
Mehadia=dict(Lugoj=70,Drobeta=75),
Neamt=dict(Iasi=87),
Oradea=dict(Zerind=71,Sibiu=151),
Pitesti=dict(Rimnicu=97,Bucharest=101,Craiova=138),
Rimnicu=dict(Sibiu=80,Pitesti=97,Craiova=146),
Sibiu=dict(Rimnicu=80,Fagaras=99,Arad=140,Oradea=151),
Timisoara=dict(Lugoj=111,Arad=118),
Urziceni=dict(Bucharest=85,Hirsova=98,Vaslui=142),
Vaslui=dict(Iasi=92,Urziceni=142),
Zerind=dict(Oradea=71,Arad=75)
)
import queue as Q
#from RMP import dict_gn
#from RMP import dict_hn

start='Arad'
goal='Bucharest'
result="

def get_fn(citystr):
    cities=citystr.split(" , ")
    hn=gn=0
    for ctr in range(0, len(cities)-1):
        gn=gn+dict_gn[cities[ctr]][cities[ctr+1]]
    hn=dict_hn[cities[len(cities)-1]]
    return(hn+gn)

def expand(cityq):
    global result
    tot, citystr, thiscity=cityq.get()
    if thiscity==goal:
        result=citystr+" : : "+str(tot)
        return
    for cty in dict_gn[thiscity]:
        cityq.put((get_fn(citystr+" , "+cty), citystr+" , "+cty, cty))
    expand(cityq)

def main():
    cityq=Q.PriorityQueue()
    thiscity=start
    cityq.put((get_fn(start),start,thiscity))
    expand(cityq)
    print("The A* path with the total is: ")
    print(result)

main()

"""
OUTPUT:
The A* path with the total is:
Arad , Sibiu , Rimnicu , Pitesti , Bucharest : : 418
"""
```

3. Implement Greedy search algorithm for any of the following application:

Prim's Minimal Spanning Tree Algorithm

Prim's Algorithm | Minimum Spanning Tree (Python Code)



We will study what is the minimum spanning tree and how to convert a graph into a minimum spanning tree using Prim's Algorithm. We will learn the algorithm and python code for prim's algorithm and an example for better understanding. Lastly, we will study the running time complexity and applications of prim's algorithm in real life.

What is a Minimum Spanning Tree?

As we all know, the graph which does not have edges pointing to any direction in a graph is called an undirected graph and the graph always has a path from a vertex to any other vertex. A spanning tree is a subgraph of the undirected connected graph where it includes all the nodes of the graph with the minimum possible number of edges. Remember, the subgraph should contain each and every node of the original graph. If any node is missed out then it is not a spanning tree and also, the spanning tree doesn't contain cycles. If the graph has n number of nodes, then the total number of spanning trees created from a complete graph is equal to n^(n-2). In a spanning tree, the edges may or may not have weights associated with them. Therefore, the spanning tree in which the sum of edges is minimum as possible then that spanning tree is called the minimum spanning tree. One graph can have multiple spanning-tree but it can have only one unique minimum spanning tree. There are two different ways to find out the minimum spanning tree from the complete graph i.e Kruskal's algorithm and Prim's algorithm. Let us study prim's algorithm in detail below:

What is Prim's Algorithm?

Prim's algorithm is a minimum spanning tree algorithm which helps to find out the edges of the graph to form the tree including every node with the minimum sum of weights to form the minimum spanning tree. Prim's algorithm starts with the single source node and later explore all the adjacent nodes of the source node with all the connecting edges. While we are exploring the graphs, we will choose the edges with the minimum weight and those which cannot cause the cycles in the graph.

Prim's Algorithm for Minimum Spanning Tree

Prim's algorithm basically follows the greedy algorithm approach to find the optimal solution. To find the minimum spanning tree using prim's algorithm, we will choose a source node and keep adding the edges with the lowest weight.

The algorithm is as given below:

- Initialize the algorithm by choosing the source vertex
- Find the minimum weight edge connected to the source node and another node and add it to the tree
- Keep repeating this process until we find the minimum spanning tree

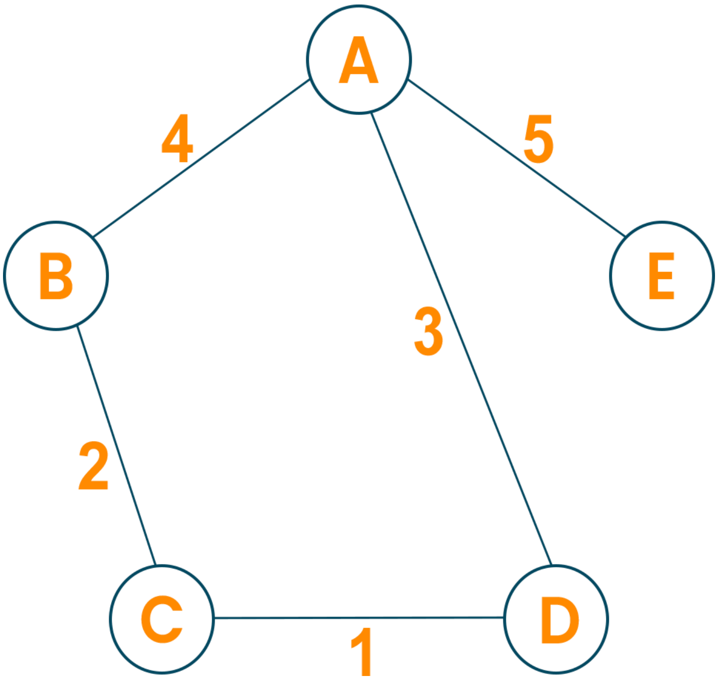
Pseudocode

```
T = $\emptyset$;
M = { 1 };
while (M $\neq$ N)
    let (m, n) be the lowest cost edge such that m $\in$ M and n $\in$ N - M;
    T = T $\cup$ {(m, n)}
    M = M $\cup$ {n}
```

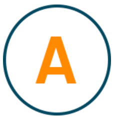
Here we create two sets of nodes i.e M and M-N. M set contains the list of nodes that have been visited and the M-N set contains the nodes that haven’t been visited. Later, we will move each node from M to M-N after each step by connecting the least weight edge.

Example

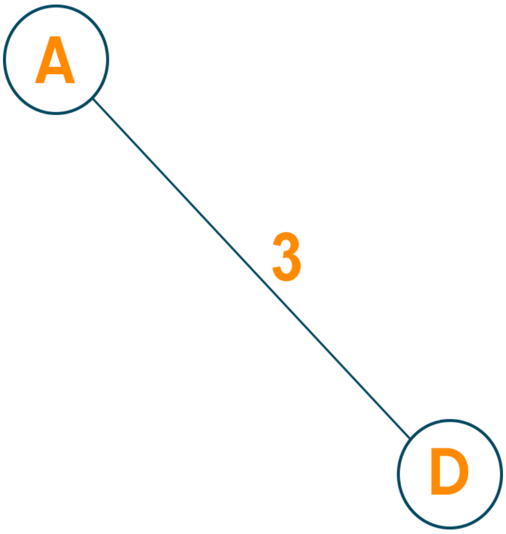
Let us consider the below-weighted graph



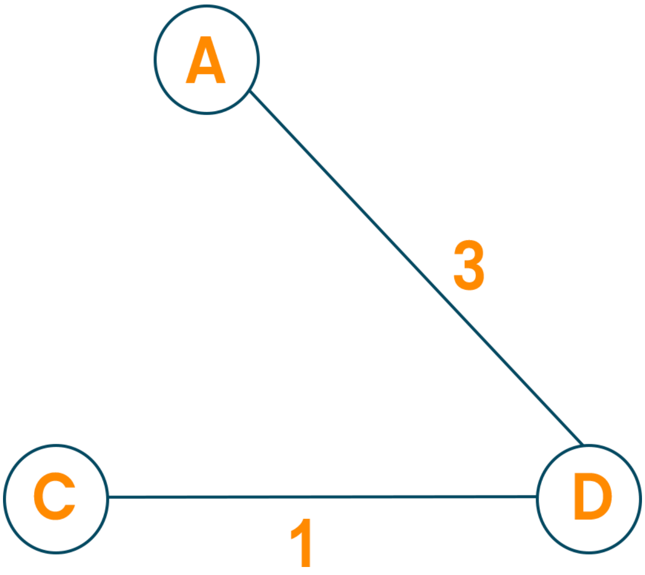
Later we will consider the source vertex to initialize the algorithm



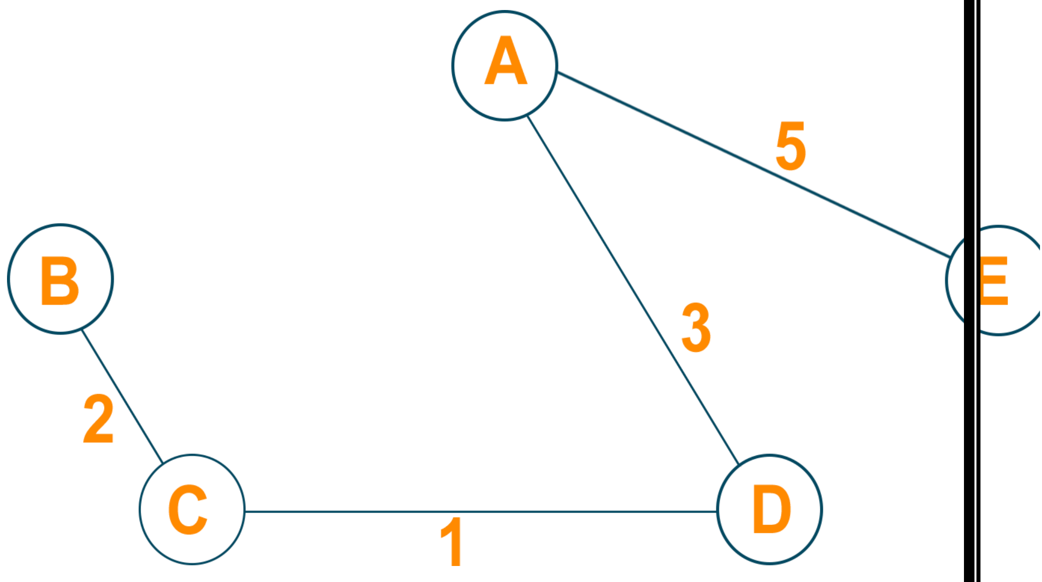
Now, we will choose the shortest weight edge from the source vertex and add it to finding the spanning tree.



Then, choose the next nearest node connected with the minimum edge and add it to the solution. If there are multiple choices then choose anyone.



Continue the steps until all nodes are included and we find the minimum spanning tree.



```
# Prim's Algorithm in Python

INF = 9999999
# number of vertices in graph
N = 5
#creating graph by adjacency matrix method
G = [[0, 19, 5, 0, 0],
     [19, 0, 5, 9, 2],
     [5, 5, 0, 1, 6],
     [0, 9, 1, 0, 1],
     [0, 2, 6, 1, 0]]

selected_node = [0, 0, 0, 0, 0]

no_edge = 0

selected_node[0] = True

# printing for edge and weight
print("Edge : Weight\n")
while (no_edge < N - 1):

    minimum = INF
    a = 0
    b = 0
    for m in range(N):
        if selected_node[m]:
            for n in range(N):
                if ((not selected_node[n]) and G[m][n]):
                    # not in selected and there is an edge
                    if minimum > G[m][n]:
                        minimum = G[m][n]
                a = m
                b = n
print(str(a) + "-" + str(b) + ":" + str(G[a][b]))
selected_node[b] = True
no_edge += 1
```

Time Complexity:

The running time for prim’s algorithm is O(VlogV + ElogV) which is equal to O(ElogV) because every insertion of a node in the solution takes logarithmic time. Here, E is the number of edges and V is the number of vertices/nodes. However, we can improve the running time complexity to O(E + logV) of prim’s algorithm using Fibonacci Heaps.

Applications

- Prim’s algorithm is used in network design
- It is used in network cycles and rail tracks connecting all the cities
- Prim’s algorithm is used in laying cables of electrical wiring
- Prim’s algorithm is used in irrigation channels and placing microwave towers
- It is used in cluster analysis
- Prim’s algorithm is used in gaming development and cognitive science
- Pathfinding algorithms in artificial intelligence and traveling salesman problems make use of prim’s algorithm.

Conclusion

As we studied, the minimum spanning tree has its own importance in the real world, it is important to learn the prim’s algorithm which leads us to find the solution to many problems. When it comes to finding the minimum spanning tree for the dense graphs, prim’s algorithm is the first choice.

Group-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

8 Queens Problem using Branch and Bound

The N-Queens problem is a puzzle of placing exactly N queens on an NxN chessboard, such that no two queens can attack each other in that configuration. Thus, no two queens can lie in the same row,column or diagnol.

The branch and bound solution is somehow different, it generates a partial solution until it figures that there's no point going deeper as we would ultimately lead to a dead end.

In the backtracking approach, we maintain an 8x8 binary matrix for keeping track of safe cells (by eliminating the unsafe cells, those that are likely to be attacked) and update it each time we place a new queen. However, it required O(n^2) time to check safe cell and update the queen.

In the 8 queens problem, we ensure the following:

1. no two queens share a row
2. no two queens share a column
3. no two queens share the same left diagnol
4. no two queens share the same right diagnol

we already ensure that the queens do not share the same column by the way we fill out our auxiliary matrix (column by column). Hence, only the left out 3 conditions are left out to be satisfied.

Applying the branch and bound approach :

The branch and bound approach suggets that we create a partial solution and use it to ascertain whether we need to continue in a particular direction or not. For this problem, we create 3 arrays to check for conditions 1,3 and 4. The boolean arrays tell which rows and diagnols are already occupied. To achieve this, we need a numbering system to specify which queen is placed.

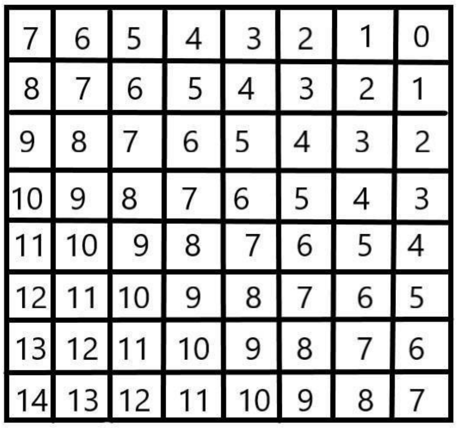
The indexes on these arrays would help us know which queen we are analysing.

Preprocessing - create two NxN matrices, one for top-left to bottom-right diagnol, and other for top-right to bottom-left diagnol. We need to fill these in such a way that two queens sharing same top-left_bottom-right diagnol will have same value in slashDiagonal and two queens sharing same top-right_bottom-left diagnol will have same value in backSlashDiagnol.

slashDiagnol(row)(col) = row + col

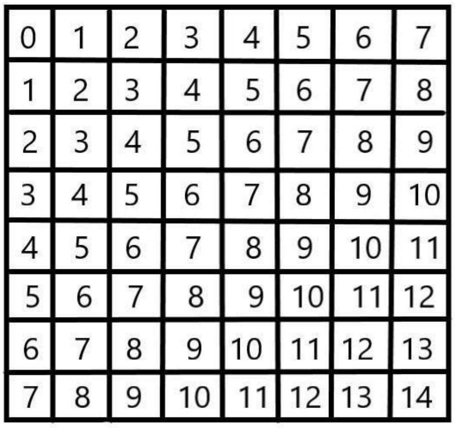
backSlashDiagnol(row)(col) = row - col + (N-1) { N = 8 }

{ we added (N-1) as we do not need negative values in backSlashDiagnol }



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

slash diagnol[row][col] = row + col



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

backslash diagnol[row][col] = row-col+(N-1)

For placing a queen *i* on row *j*, check the following :

1. whether row 'j' is used or not
2. whether slashDiagnol 'i+j' is used or not
3. whether backSlashDiagnol 'i-j+7' is used or not

If the answer to any one of the following is true, we try another location for queen **i** on row **j**, mark the row and diagnols; and recur for queen **i+1**.

```
#include<bits/stdc++.h>

using namespace std; int

board[8][8]

int n

// function to print solution

void printSolution(int board[n][n])

{

   for (int i = 0; i < n; i++)

   {

      for (int j = 0; j < n; j++)
```

```
            { cout<<board[i][j]<<" ";
            }
          cout<<endl;
      }
  }

  //function to check if queen can
  // be placed on board[row][col]

  bool isPossible(int row, int col, int slashDiagnol[n][n],
              int backSlashDiagnol[n][n], bool rowLook[n],
              bool slashDiagnolLook[], bool backSlashDiagnolLook[])
  {
      if (slashDiagnolLook(slashDiagnol[row][col] || backSlashDiagnol[row][col]
          || rowLook[row] )
      return false;

      return true;
  }

  //A recursive utility function to solve N Queen problem
  bool solveNQueensUtil(int board[n][n], int col,
      int slashDiagnol[n][n],int backSlashDiagnol[n][n],
      bool rowLook[n], bool slashDiagnolLook[],
      bool backSlashDiagnolLook[] )
  {
      //base case: If all queens are placedif
      (col >= N)
          return true;
```

```
//Consider this column and try placing
// queen in all rows one by one
for (int i = 0; i < n; i++)
{

    if ( isPossible(i, col, slashDiagnol, backSlashDiagnol,
         rowLook, slashDiagnolLook,
         backSlashDiagnolLook) )
    {
        board[i][col] = 1;
        rowLookup[i] = true;
        slashDiagnolLook[slashDiagnol[i][col]] = true;
        backSlashDiagnolLook[backSlashDiagnol[i][col] = true;

        //recur to place rest of the queens
        if ( solveNQueensUtil(board, col + 1, slashCode, backslashCode,
                rowLookup, slashCodeLookup, backslashCodeLookup) )
            return true;

        // placing queen in board[i][col]
        // dosen't yield a solution, backtrack

        board[i][col] = 0; rowLook[i]
        = false;
        slashDiagolLook[slashDiagnol[i][col]] = false;
        backSlashDiagnolLook[backSlashDiagnol[i][col]] = false;
    }
}

//If queen can not be place in any row in
```

```
   //this colum col then return false
    return false;
}

/* This function solves the N Queen problem using Branch and Bound. It mainly uses
solveNQueensUtil() to solve the problem. It returns false if queens cannot be placed,
otherwise return true and prints placement of queens in the form of 1s. Please note that there
may be more than one solutions, this function prints one of the feasible solutions.*/
bool solveNQueens(n)
{

    memset(board, 0, sizeof(board));

    // helper matrices
    int slashDiagnol[n][n];
    int backSlashDiagnol[n][n];

    // arrays to tell us which rows are occupied
    bool rowLook[n] = {false};

    //keep two arrays to tell us which diagonals are occupied
    bool slashDiagnolLook[2*n-1] = {false};
    bool backSlashDiagnolLook[2*n-1] = {false};

    // initialize helper matrices
    for (int r = 0; r < n; r++)
        for (int c = 0; c < n; c++)
        {
            slashDiagnol[r][c] = r+c;
            backSlashDiagnol[r][c] = (r+c-7);
         }
    if (solveNQueensUtil(board, 0, slashDiagnol, backSlashDiagnol,
     rowLook, slashDiagnolLook, backSlashDiagnolLook) == false)
    {
       cout<<"No solution"<<endl;
       return false;
    }

    // solution found
    printSolution(board);
    return true;
}

// main function
int main()
{
    cin>>n; // can take any size from 0 to 8
    solveNQueens(n);
    return 0;
}
```

Output- for
(n = 8)

```
1  0  0  0  0  0  0  0
0  0  0  0  0  0  1  0
0  0  0  0  1  0  0  0
0  0  0  0  0  0  0  1
0  1  0  0  0  0  0  0
0  0  0  1  0  0  0  0
0  0  0  0  0  1  0  0
0  0  1  0  0  0  0  0
```

Graph coloring problem’s solution using backtracking algorithm

Graph coloring

The **graph coloring problem** is to discover whether the nodes of the graph G can be covered in such a way, that no two adjacent nodes have the same color yet only m colors are used. This graph coloring problem is also known as M-colorability decision problem.

The M – colorability optimization problem deals with the smallest integer m for which the graph G can be colored. The integer is known as a chromatic number of the graph.

Here, it can also be noticed that if d is the degree of the given graph, then it can be colored with d+ 1 color.

A graph is also known to be planar if and only if it can be drawn in a planar in such a way that no two edges cross each other. A special case is the 4 - colors problem for planar graphs. The problem is to color the region in a map in such a way that no two adjacent regions have the same color. Yet only four colors are needed. This is a problem for which graphs are very useful because a map can be easily transformed into a graph. Each region of the map becomes the node, and if two regions are adjacent, they are joined by an edge.

Graph coloring problem can also be solved using a state space tree, whereby applying a backtracking method required results are obtained.

For solving the **graph coloring problem**, we suppose that the graph is represented by its adjacency matrix G[1:n, 1:n], where, G[i, j]= 1 if (i, j) is an edge of G, and G[i, j] = 0 otherwise.

The colors are represented by the integers 1, 2, ..., m and the solutions are given by the n-tuple (x1, x2, x3, ..., xn), where x1 is the color of node i.

Algorithm for finding the m - colorings of a graph

```
Algorithm mcoloring ( k )
// this algorithm is formed using the recursive backtracking
// schema. The graph is represented by its Boolean adjacency
// matrix G [1: n, 1: n]. All assignments of 1, 2, …, m to the
// vertices of the graph such that adjacent vertices are
// assigned distinct are printed. K is the index
// of the next vertex to color.
```

```
{
Repeat
{
// generate all legal assignments for x[k],
Next value (k); // assign to x[k] a legal color.
 If ( x[k] = 0 ) then return; // no new color possible
 If (k = n) then // at most m colors have been used to color the n vertices.
  Write (x[1 : n ]);
 Else mcoloring (k + 1);
}
Until (false);
}
```

This algorithm uses the recursive backtracking schema. In this algorithm colors to be assigned are to determine from the range (0, m), i.e., m colors are available.

The total time required by the above algorithm is **O (nm^n)**.

Implementation of Backtracking solution

C/C++

```
/* C/C++ program to solve N Queen Problem using
   backtracking */
#define N 4
#include <stdbool.h>
#include <stdio.h>

/* A utility function to print solution */
void printSolution(int board[N][N])
{
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++)
            printf(" %d ", board[i][j]);
        printf("\n");
    }
}
```

```
/* A utility function to check if a queen canbe placed on board[row][col]. Note
that this function is called when "col" queens arealready placed in columns from
0 to col -1. So we need to check only left side for attacking queens */

bool isSafe(int board[N][N], int row, int col)
{
    int i, j;

    /* Check this row on left side */
    for (i = 0; i < col; i++)
        if (board[row][i])
            return false;

    /* Check upper diagonal on left side */
    for (i = row, j = col; i >= 0 && j >= 0; i--, j--)
        if (board[i][j])
            return false;

    /* Check lower diagonal on left side */
    for (i = row, j = col; j >= 0 && i < N; i++, j--)
        if (board[i][j])
            return false;

    return true;
}

/* A recursive utility function to solve N
   Queen problem */
bool solveNQUtil(int board[N][N], int col)
{
    /* base case: If all queens are placed
      then return true */
    if (col >= N)
        return true;

    /* Consider this column and try placing
       this queen in all rows one by one */
    for (int i = 0; i < N; i++) {
        /* Check if the queen can be placed on
          board[i][col] */
        if (isSafe(board, i, col)) {
            /* Place this queen in board[i][col] */
            board[i][col] = 1;
            /* recur to place rest of the queens */
            if (solveNQUtil(board, col + 1))
                return true;

            /* If placing queen in board[i][col]
               doesn't lead to a solution, then
               remove queen from board[i][col] */
            board[i][col] = 0; // BACKTRACK
        }
    }

    /* If the queen cannot be placed in any row in
        this column col then return false */
    return false;
}

/* This function solves the N Queen problem using   Backtracking. It mainly
uses solveNQUtil() to solve the problem. It returns false if queens cannot be
placed, otherwise, return true and prints placement of queens in the form of 1s.
Please note that there may be more than one solutions, this  function prints one
of the feasible solutions.*/
bool solveNQ()
{
    int board[N][N] = { { 0, 0, 0, 0 },
                        { 0, 0, 0, 0 },
                        { 0, 0, 0, 0 },
                        { 0, 0, 0, 0 } };

    if (solveNQUtil(board, 0) == false) {
        printf("Solution does not exist");
        return false;
    }

    printSolution(board);
    return true;
}

// driver program to test above function
int main()
{
    solveNQ();
    return 0;
```

```
}
```

Output:The 1 values indicate placements of queens

```
0 0 1 0
1 0 0 0
0 0 0 1
0 1 0 0
```

5. Develop an elementary chatbot for any suitable customer interaction application.

What is a chatbot?

A chatbot is a computer program designed to have a conversation with human beings over the internet. It’s also known as conversational agents, which communicate and collaborate with human users, through text messaging, in order to accomplish a specific task. Basically, there are two types of chatbots. The one that uses **Artificial Intelligence**, and another one is based on multiple choice scripts.

Both types of chatbots aim to create a more personalized content experience for the users, whether that's while watching a video, reading articles or buying new shoes.

These Chatbots hold the promise of being the next generation of technology that people use to interact online with business enterprises. These Chatbots offer a lot of advantages, one of which is that, because Chatbots communicate using a natural language, users don't need to learn yet another new website interface, to get comfortable with the unavoidable quirks.

Chatbots are capable to interpret human speech, and decide which information is being sought. Artificial intelligence is getting smarter each day, and brands that are integrating Chatbots with the artificial intelligence, can deliver one-to-one individualized experiences to consumers.

Why chatbot?

Chatbots can be useful in many aspects of the customer experience, including providing customer service, presenting product recommendations and engaging customers through targeted marketing campaigns. If a customer has an issue with a product, she can connect with a chatbot to explain the situation and the chatbot can input that information to provide a recommendation of how to fix the product. On the recommendation side, chatbots can be used to share popular products with customers that they might find useful and can act as a sort of personal shopper or concierge service to find the perfect gift, meal or night out for a customer with just a few basic questions. Brands are also using chatbots to connect their customers with thought leaders and add personality to their products. In all cases, brands seem to be having great success and experiencing increased engagement and revenue.

Chatbots are easy to use and many customers prefer them over calling a representative on the phone because it tends to be faster and less invasive. They can also save money for companies and are easy to set up.

Chatbots are relatively new and most companies haven't implemented them yet, it's only natural that users are interested in them. Hence, people want to discover what chatbots can and cannot do.

The number of businesses using chatbots has grown exponentially. Chatbots have increased from 30,000 in 2016 to over 100,000 today. Every major company has announced their own chatbot and 60% of the youth population uses them daily.

These statistics prove that chatbots are the new-gen tech. No more waiting for the right time to incorporate them into your business. The time is now. By the year 2020, nearly 80% of businesses will have their own chatbot.

Billions of people are already using chatbots, so it's time your business did too.

Benefits of chatbot?

Chatbots are being made to ease the pain that the industries are facing today. The purpose of chat bots is to support and scale business teams in their relations with customers.

Chatbots may sound like a futuristic notion, but according to Global Web Index statistics, it is said that 75% of internet users are adopting one or more messenger platforms. Although research shows us that each user makes use of an average of 24 apps a month, wherein 80% of the time would be in just 5 apps. This means you can hardly shoot ahead with an app, but you still have high chances to integrate your chatbot with one of these platforms.

Now lets go through some of the benefits that chatbots provide:

1. Available 24*7:

I'm sure most of you have experienced listening to the boring music playing while you're kept on hold by a customer care agent. On an average people spend 7 minutes until they are assigned to an agent. Gone are the days of waiting for the next available operative. Bots are replacing live chat and other forms of contact such as emails and phone calls.

Since chat bots are basically virtual robots they never get tired and continue to obey your command. They will continue to operate every day throughout the year without requiring to take a break. This improves your customer satisfaction and helps you rank highly in your sector.

2. Handling Customers:

We humans are restricted to the number of things we can do at the same time. A study suggests that humans can only concentrate on 3–4 things at the same time. If it goes beyond that you are bound to meet errors.

Chatbots on the other hand can simultaneously have conversations with thousands of people. No matter what time of the day it is or how many people are contacting you, every single one of them will be answered instantly. Companies like Taco Bell and Domino’s are already using chatbots to arrange delivery of parcels.

3. Helps you Save Money:

If you are a business owner you are bound have a lot of employees who need to be paid for the work they do. And these expenses just keep adding up as business grows. Chatbots are a one time investment which helps businesses reduce down on staff required.

You could integrate a customer support chatbot in your business to cater to simple queries of customers and pass on only the complex queries to customer support agents.

4. Provides 100% satisfaction to customers:

Humans react to others based on their mood and emotions. If a agent is having a good attitude or is in good mood he will most probably talk to customers in a good way. In contrary to this the customer will not be satisfied.

Whereas chatbots are bound by some rules and obey them as long as they’re programmed to. They always treat a customer in the most polite and perfect way no matter how rough the person is. Also, in the travel and hospitality industry where travelers do not speak the same language, a bot can be trained to communicate in the language of the traveler.

5. Automation of repetitive work:

Lets be honest, no one likes doing the same work again and again over brief period of time. In the case of humans, such tasks are prone to errors. Chatbots now help automate tasks which are to be done frequently and at the right time.

Also, now there are numerous slack bots which automate repetitive tasks. This helps people save time and increase productivity. For example, there are new items bought from your eCommerce site or there is a bug reported then it sends a short summary to a slack channel.

6. Personal Assistant:

People could use Bots as a fashion advisor for clothing recommendations, or ask trading tips from a finance bot, suggest places to visit from a travel bot and so forth. This would help the users get a more personal touch from the chatbot. Also, the chatbot will remember all your choices and provide you with relevant choices the next time you visit it.

How chatbot can drive revenue for you?

Below we have compiled reasons why chatbots are important for your business and how can they help in increasing revenues:

a. Higher user customer engagement

Most businesses these days have a web presence. But with being on the internet, boundaries of day and night, availability and unavailability have changed, so have user expectations. This is probably the biggest reason to use them. Bots give the user an interactive experience. It makes customers feel they are working with someone to help resolve their issue. If done right, bots can help customers find what they are looking for and make them more likely to return.

Customer Engagement

- Clearance Sale : Notify users about on-going clearance sale of products relevant to the users at their nearest outlets.
- Product Finder : Enable consultative selling without the need of a call center
- It offer Notification : Notify users about offers, product launches on products/ services they've shown interest in, and products that's back in stock

b. Mobile-ready and immediate availability

Along with a web presence, it has also become increasingly important for brands to have a mobile presence - mobile apps, mobile-optimized websites. Considering how chat has been around on the mobile for ages, most chatbot implementations don’t need you to work on tweaking their UI, they are ready to implement and so available to your customers immediately

You might argue that you have an app for that. Having an app for your brand is great, but having users discover that app, download it and use it to stay engaged is not an easy deal. Instead, implementing a chatbot - which works on the mobile browser or a messaging-app which the user regularly uses - makes it all the more reason for a customer to be engaged with the brand

c. It can drive sales

Chatbots can be intelligent. Depending on a user’s preferences or purchases, it can send products to customers which are more likely to convert into sales. Or it can send coupons to users for in-store purchases/discounts. Bots can also be used to link the user to your mCommerce site/app so they can buy the product directly from the convenience of their phones

Sell Intelligently

- Product Recommendations : Push proactive recommendations to users based on their preferences and search and order history.
- Enable order booking over chat.

d. Minimal cost - Maximum return

The best part about bots is they are cheap. Chatbot provide the necessary infrastructure and APIs for creating these bots. They require minimal maintenance and since it is automated, there is no labor-intensive work that goes in there.

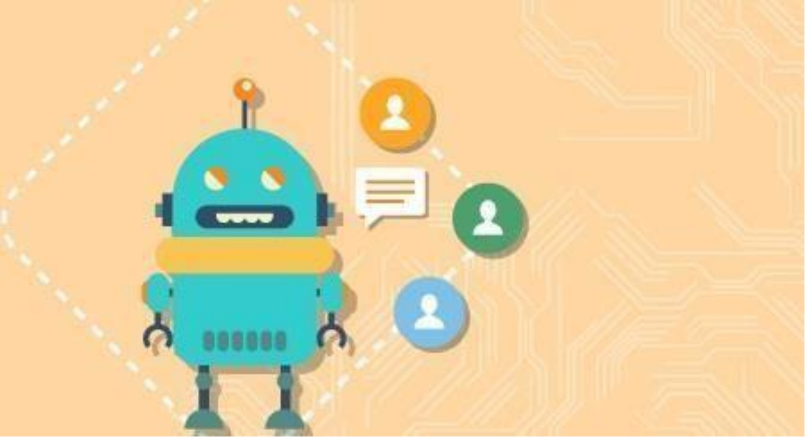
e. Customer Service

- Track Order : Keep users up to date with order status. Schedule or reschedule delivery to a provided address or request to pick it up at any other Best Buy outlet.
- Stock outs : Notify users when desired product is available and place order over a chat.
- Returns and Replacements : No waiting time to reach customer care. Customers can instantly place request to replace or return an order.
- Seek Reviews : Reach out to users to seek reviews on the products recently bought

Gift Recommendations

- Recommend relevant gifting options to users, accessing calendar events and understanding the likes and style of beneficiary.
- Opportunity to upsell gift cards for the users for every occasion.

Application across Industries



According to a new survey, 80% of businesses want to integrate chatbots in their business model by 2020. So which industries can reap the greatest benefits by implementing consumer-facing chatbots? According to a chatbot, these major areas of direct-to-consumer engagement are prime:

Chatbots in Restaurant and Retail Industries

Famous restaurant chains like Burger King and Taco bell has introduced their Chatbots to stand out of competitors of the Industry as well as treat their customers quickly. Customers of these restaurants are greeted by the resident Chatbots, and are offered the menu options- like a counter order, the Buyer chooses their pickup location, pays, and gets told when they can head over to grab their food. Chatbots also works to accept table reservations, take special requests and go take the extra step to make the evening special for your guests.

Chatbots are not only good for the restaurant staff in reducing work and pain but can provide a better user experience for the customers.

Chatbots in Hospitality and Travel

For hoteliers, automation has been held up as a solution for all difficulties related to productivity issues, labour costs, a way to ensure consistently, streamlined production processes across the system. Accurate and immediate delivery of information to customers is a major factor in running a successful online Business, especially in the price sensitive and competitive Travel and Hospitality industry. Chatbots particularly have gotten a lot of attention from the hospitality industry in recent months.

Chatbots can help hotels in a number of areas, including time management, guest services and cost reduction. They can assist guests with elementary questions and requests. Thus, freeing up hotel staff to devote more of their time and attention to time-sensitive, critical, and complicated tasks. They are often more cost effective and faster than their human counterparts. They can be programmed to speak to guests in different languages, making it easier for the guests to speak in their local language to communicate.

Chatbots in Health Industry

Chatbots are a much better fit for patient engagement than Standalone apps. Through these Health-Bots, users can ask health related questions and receive immediate responses. These responses are either original or based on responses to similar questions in the database. The impersonal nature of a bot could act as a benefit in certain situations, where an actual Doctor is not needed.

Chatbots ease the access to healthcare and industry has favourable chances to serve their customers with personalised health tips. It can be a good example of the success of Chatbots and Service Industry combo.

Chatbots in E-Commerce

Mobile messengers- connected with Chatbots and the E-commerce business can open a new channel for selling the products online. E-commerce Shopping destination “Spring” was the early adopter. E-commerce future is where brands have their own Chatbots which can interact with their customers through their apps.

Chatbots in Fashion Industry

Chatbots, AI and Machine Learning pave a new domain of possibilities in the Fashion industry, from Data Analytics to Personal Chatbot Stylists. Fashion is such an industry where luxury goods can only be bought in a few physical boutiques and one to one customer service is essential. The Internet changed this dramatically, by giving the customers a seamless but a very impersonal experience of shopping. This particular problem can be solved by Chatbots. Customers can be treated personally with bots, which can exchange messages, give required suggestions and information. Famous fashion brands like Burberry, Tommy Hilfiger have recently launched Chatbots for the London and New York Fashion Week respectively. Sephora a famous cosmetics brand and H&M– a fashion clothing brand have also launched their Chatbots.

Chatbots in Finance

Chatbots have already stepped in Finance Industry. Chatbots can be programmed to assists the customers as Financial Advisor, Expense Saving Bot, Banking Bots, Tax bots, etc. Banks and Fintech have ample opportunities in developing bots for reducing their costs as well as human errors. Chatbots can work for customer's convenience, managing multiple accounts, directly checking their bank balance and expenses on particular things. Further about Finance and Chatbots have been discussed in our earlier blog: Chatbots as your Personal Finance Assistant.

Chatbots in Fitness Industry

Chat based health and fitness companies using Chatbot, to help their customers get personalised health and fitness tips. Tech based fitness companies can have a huge opportunity by developing their own Chatbots offering huge customer base with personalised services. Engage with your fans like never before with news, highlights, game-day info, roster and more.

Chatbots and Service Industry together have a wide range of opportunities and small to big all size of companies using chatbots to reduce their work and help their customers better.

Chatbots in Media

Big publisher or small agency, our suite of tools can help your audience chatbot experience rich and frictionless. Famous News and Media companies like The Wall Street Journal, CNN, Fox news, etc have launched their bots to help you receive the latest news on the go.

Chatbot in Celebrity:

With a chatbot you can now have one-on-one conversation with millions of fans.

Chatbot in Marketing

SMS Marketing

- Why promote just a coupon code that the customer does not know how to use?
- Improve conversions from your existing SMS campaigns.
- Talk to your customers when they want to using "Talk to an Agent" feature.

Email Marketing

- So your eMail has made a solid elevator pitch about your product.
- As a next step, is making customers fill an online form the most exciting way to engage with your customers?
- It’s time to rethink the landing page.
- Instantly engage in a conversation with your customers.
- Address their concerns and queries

Social Media Triage

- How effectively are you addressing the negative sentiment around your brand on social media?
- Addressing queries instantly and effectively can convert even an angry customer into a loyal fan.
- Leverage a chatbot as your first response strategy and comfort that customer.

Process

Stage #1: Chatty Bot welcomes you
Teach your assistant to introduce itself in the console.
Stage #2: Print your name
Introduce yourself to the bot.
Stage #3: Guess the age
Use your knowledge of strings and numbers to make the assistant guess your age.
Stage #4: Learning numbers
Your assistant is old enough to learn how to count. And you are experienced enough to apply a for loop at this stage!
Stage #5: Multiple Choice
At this point, the assistant will be able to check your knowledge and ask multiple-choice questions. Add some functions to your code and make the stage even better.

How To Run The Project?

To run this project, you must have installed Python on your PC. After downloading the project, follow the steps below:
Step1: Extract/Unzip the file
Step2: Go inside the project folder, open cmd then type bot.py and enter to start the system.
OR
Step2: Simply, double-click the bot.py file and you are ready to go.

Group-C

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

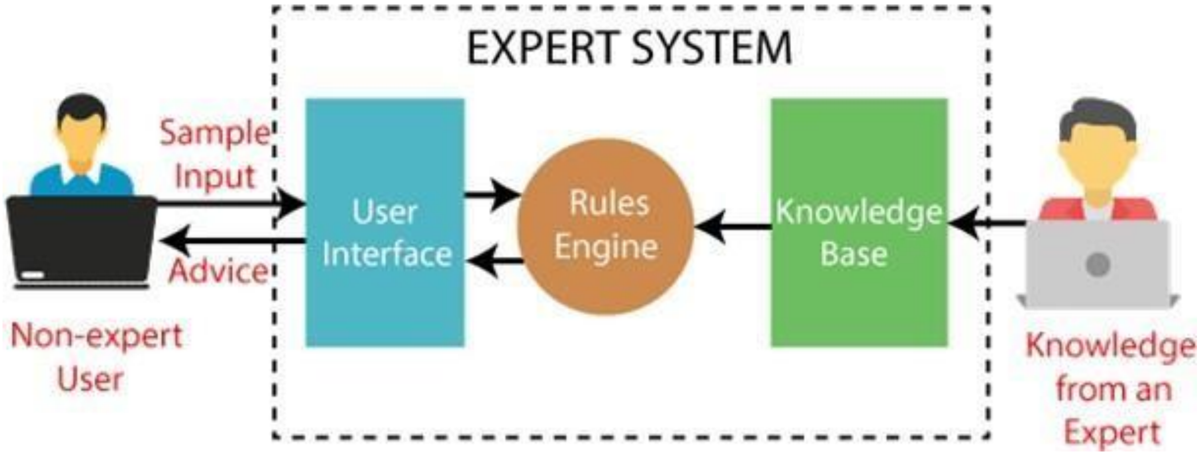
What is an Expert System?

An expert system is a computer program that is designed to solve complex problems and to provide decision-making ability like a human expert. It performs this by extracting knowledge from its knowledge base using the reasoning and inference rules according to the user queries.

The expert system is a part of AI, and the first ES was developed in the year 1970, which was the first successful approach of artificial intelligence. It solves the most complex issue as an expert by extracting the knowledge stored in its knowledge base. The system helps in decision making for compsex problems using **both facts and heuristics like a human expert**. It is called so because it contains the expert knowledge of a specific domain and can solve any complex problem of that particular domain. These systems are designed for a specific domain, such as **medicine, science,** etc.

The performance of an expert system is based on the expert's knowledge stored in its knowledge base. The more knowledge stored in the KB, the more that system improves its performance. One of the common examples of an ES is a suggestion of spelling errors while typing in the Google search box.

Below is the block diagram that represents the working of an expert system:



Below are some popular examples of the Expert System:

- **DENDRAL:** It was an artificial intelligence project that was made as a chemical analysis expert system. It was used in organic chemistry to detect unknown organic molecules with the help of their mass spectra and knowledge base of chemistry.
- **MYCIN:** It was one of the earliest backward chaining expert systems that was designed to find the bacteria causing infections like bacteraemia and meningitis. It was also used for the recommendation of antibiotics and the diagnosis of blood clotting diseases.
- **PXDES:** It is an expert system that is used to determine the type and level of lung cancer. To determine the disease, it takes a picture from the upper body, which looks like the shadow. This shadow identifies the type and degree of harm.
- **CaDeT:** The CaDet expert system is a diagnostic support system that can detect cancer at early stages.

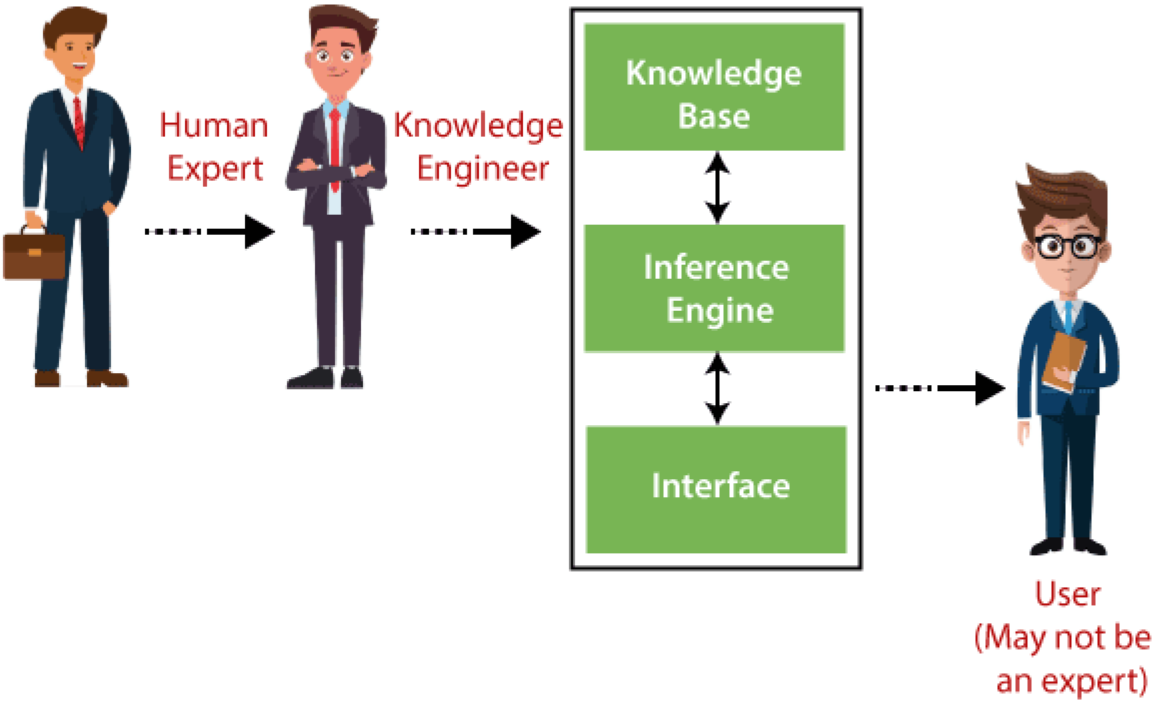
Characteristics of Expert System

- **High Performance:** The expert system provides high performance for solving any type of complex problem of a specific domain with high efficiency and accuracy.
- **Understandable:** It responds in a way that can be easily understandable by the user. It can take input in human language and provides the output in the same way.
- **Reliable:** It is much reliable for generating an efficient and accurate output.
- **Highly responsive:** ES provides the result for any complex query within a very short period of time.

Components of Expert System

An expert system mainly consists of three components:

- **User Interface**
- **Inference Engine**
- **Knowledge Base**



Participants in the development of Expert System

There are three primary participants in the building of Expert System:

1. **Expert:** The success of an ES much depends on the knowledge provided by human experts. These experts are those persons who are specialized in that specific domain.
2. **Knowledge Engineer:** Knowledge engineer is the person who gathers the knowledge from the domain experts and then codifies that knowledge to the system according to the formalism.
3. **End-User:** This is a particular person or a group of people who may not be experts, and working on the expert system needs the solution or advice for his queries, which are complex.

Advantages of Expert System

- These systems are highly reproducible.
- They can be used for risky places where the human presence is not safe.

- Error possibilities are less if the KB contains correct knowledge.
- The performance of these systems remains steady as it is not affected by emotions, tension, or fatigue.
- They provide a very high speed to respond to a particular query.

Limitations of Expert System

- The response of the expert system may get wrong if the knowledge base contains the wrong information.
- Like a human being, it cannot produce a creative output for different scenarios.
- Its maintenance and development costs are very high.
- Knowledge acquisition for designing is much difficult.
- For each domain, we require a specific ES, which is one of the big limitations.
- It cannot learn from itself and hence requires manual updates.

Applications of Expert System

- **In designing and manufacturing domain**
  It can be broadly used for designing and manufacturing physical devices such as camera lenses and automobiles.
- **In the knowledge domain**
  These systems are primarily used for publishing the relevant knowledge to the users. The two popular ES used for this domain is an advisor and a tax advisor.
- **In the finance domain**
  In the finance industries, it is used to detect any type of possible fraud, suspicious activity, and advise bankers that if they should provide loans for business or not.
- **In the diagnosis and troubleshooting of devices**
  In medical diagnosis, the ES system is used, and it was the first area where these systems were used.
- **Planning and Scheduling**
  The expert systems can also be used for planning and scheduling some particular tasks for achieving the goal of that task.

Companion Course: Laboratory Practice II (310258)

Course Objectives:

- To study fundamental concepts of cloud computing
- To learn various data storage methods on cloud
- To understand the implementation of Virtualization in Cloud Computing
- To learn the application and security on cloud computing
- To study risk management in cloud computing
- To understand the advanced technologies in cloud computing

Course Outcomes:

On completion of the course, learners should be able to

CO1: Understand the different Cloud Computing environment

CO2: Use appropriate data storage technique on Cloud, based on Cloud application

CO3: Analyze virtualization technology and install virtualization software

CO4: Develop and deploy applications on Cloud

CO5: Apply security in cloud applications

CO6: Use advance techniques in Cloud Computing

Sr. No.	**Name of Assignment**	**Page No.**	**Date**	**Remark**
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 centres. OR Case study on Amazon EC2 and learn about Amazon EC2 web services.			
2	Installation and configure Google App Engine. OR Installation and Configuration of virtualization using KVM.			
3	Creating an Application in SalesForce.com using Apex programming Language.			
4	Design and develop custom Application (Mini Project) using Sales force Cloud.			
5	**Mini-Project** 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.			

Assignment No: 1

Aim:

Case study on Amazon EC2 to learn about Amazon EC2,Amazon Elastic Compute Cloud is a central part of Amazon.com's cloud computing platform, Amazon Web Services. How EC2 allows users torrent virtual computers on which to run their own computer applications.

Objectives:

1. To learn Amazon Web Services.
2. To case study the Amazon EC2.

Software Requirements:

Ubuntu 18.04
PHP
MySQL

Hardware Requirements:

Pentium IV system with latest configuration

Theory:

Amazon Elastic Compute Cloud (EC2)
Elastic IP addresses allow you to allocate a static IP address and programmatically assign it to an instance. You can enable monitoring on an Amazon EC2 instance using Amazon CloudWatch2 in order to gain visibility into resource utilization, operational performance and overall demand patterns (including metrics such as CPU utilization, disk reads and writes, and network traffic). You can create Auto feature3 to automatically scale your capacity on certain conditions based on m Amazon CloudWatch collects. You can also distribute incoming traffic by creating an elastic load balancer using the Elastic Load Balancing4 service. Amazon Elastic Block Storage (EBS)5 volumes provide network Point-in-time consistent snapshots of EBS volumes can be created and stored on Amazon Simple Storage Service (Amazon S3)6. Amazon S3 is highly durable and distributed data store. With a simple web services interface, you can store and retr at any time, from anywhere on the web using standard HTTP verbs. Copies of objects can be distributed and cached at 14 edge locations around the world by creating a distribution using Amazon CloudFront7 service content). Amazon SimpleDB8 is a web service that provides the core functionality of a database- real-time lookup and simple querying of structured data complexity. You can organize the dataset into domains and can run queries across all of the Auto-scaling Group using the Auto network-attached persistent storage to Amazon EC2 instances. retrieve large amounts of data as objects in buckets (containers) udFront7 – a web service for content delivery (static or streaming – without the operational plexity. performance,Auto-scaling metric that n even data stored in a particular domain. Domains are collections of items that are described by attribute-value pairs.

Amazon Relational Database Service9 (Amazon RDS) provides an easy way to setup, operate and scale a relational database in the cloud. You can launch a DB Instance and get access to a full-featured MySQL database and not worry about common database administration tasks like backups, patch management etc.

Amazon Simple Queue Service (Amazon SQS)10 is a reliable, highly scalable, hosted distributed queue for storing messages as they travel between computers and application components.

Amazon Simple Notifications Service (Amazon SNS) provides a simple way to notify applications or people from the cloud by creating Topics and using a publish-subscribe protocol.

Amazon Elastic MapReduce provides a hosted Hadoop framework running on the webscale infrastructure of Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3) and allows you to create customized JobFlows. JobFlow is a sequence of MapReduce steps.

Amazon Virtual Private Cloud (Amazon VPC) allows you to extend your corporate network into a private cloud contained within AWS. Amazon VPC uses IPSec tunnel mode that enables you to create a secure connection between a gateway in your data center and a gateway in AWS.

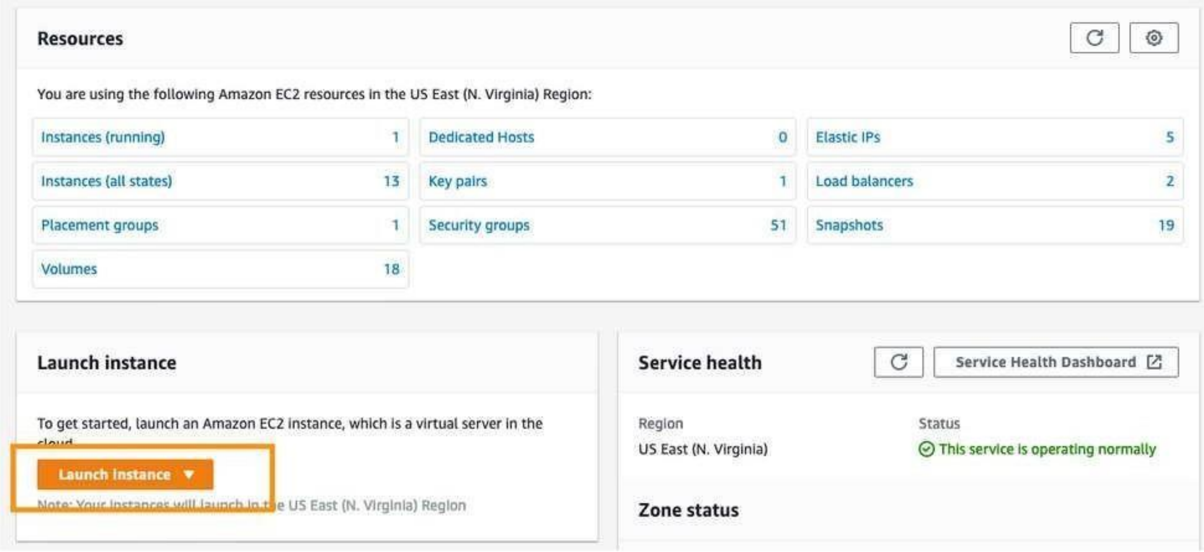
Amazon Route53 is a highly scalable DNS service that allows you manage your DNS records by creating a HostedZone for every domain you would like to manage.

AWS Identity and Access Management (IAM) enable you to create multiple Users with unique security credentials and manage the permissions for each of these Users within your AWS Account. IAM is natively integrated into AWS Services. No service APIs have changed to support IAM, and exiting applications and tools built on top of the AWS service APIs will continue to work when using IAM.
AWS also offers various payment and billing services that leverages Amazon's payment infrastructure. All AWS infrastructure services offer utility-style pricing that require no longterm commitments or contracts. For example, you pay by the hour for Amazon EC2 instance usage and pay by the gigabyte for storage and data transfer in the case of Amazon S3. More information about each of these services and their pay-as-you-go pricing is available on the You are free to use the programming model, language, or operating system (Windows, OpenSolaris or any flavor of Linux) of your choice.

You are free to pick and choose the AWS products that best satisfy your requirements—you can use any of the services individually or in any combination.
Because AWS provides resizable (storage, bandwidth and computing) resources, you are free to consume as much or as little and only pay for what you consume.
You are free to use the system management tools you've used in the past and extend your datacenter into the cloud.

Launching an EC2 instance

1. Sign in to the preview version of the AWS Management Console
2. Open the Amazon EC2 console by choosing EC2 under Compute.
3. From the EC2 Console, click Launch Instance.



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) Region:

Instances (running)	1	Dedicated Hosts	0	Elastic IPs	5
Instances (all states)	13	Key pairs	1	Load balancers	2
Placement groups	1	Security groups	51	Snapshots	19
Volumes	18				

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

Note: Your instances will launch in the US East (N. Virginia) Region

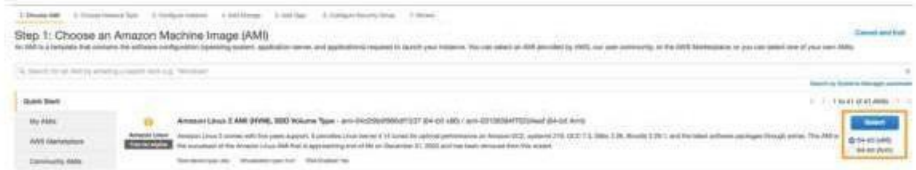
Service health

Region: US East (N. Virginia)

Status: This service is operating normally

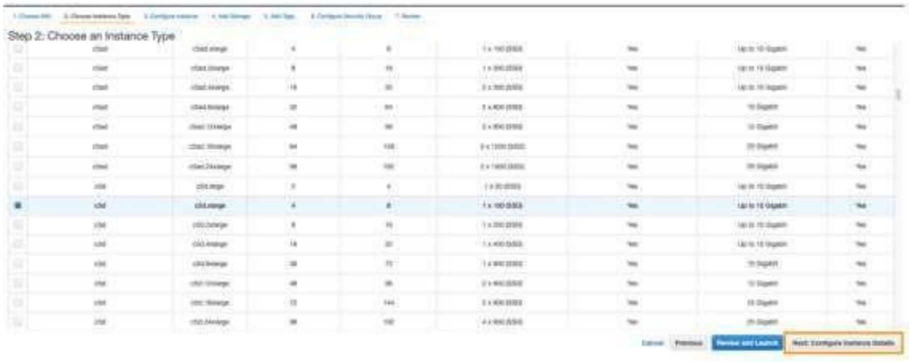
Zone status

The Choose an Amazon Machine Image (AMI) page displays a list of basic configurations called Amazon Machine Images (AMIs) that serve as templates for your instance. Select the HVM edition of the Amazon Linux 2 AMI.



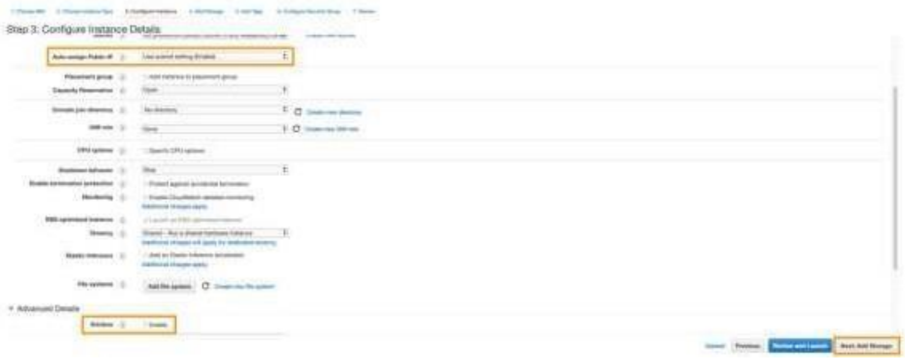
Step 1: Choose an Amazon Machine Image (AMI)

On the Choose an Instance Type page, choose c5d.xlarge as the hardware configuration of your instance and Review and Launch.



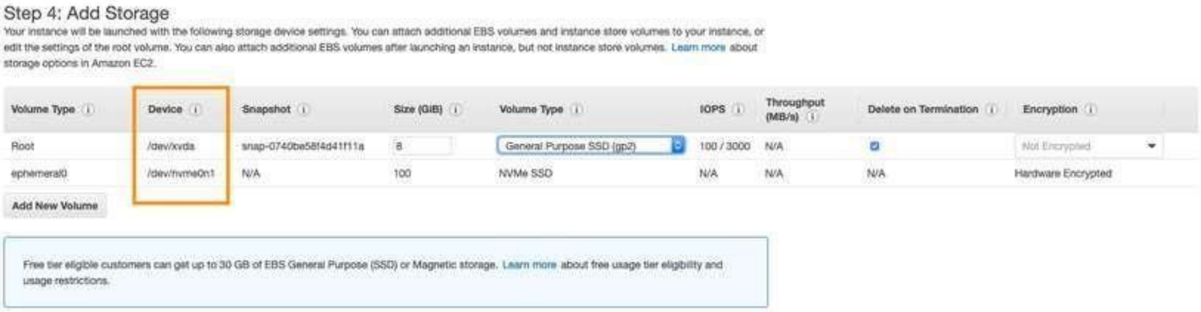
Step 2: Choose an Instance Type

On Instances details, make sure the Auto-assign Public IP is Enable and you selected Enclave as Enable. Click on Next: Add Storage



Step 3: Configure Instance Details

Review the configurations and click next: Add Tages The ephemeral0 is the Instance Storage based on NVMe SSD.



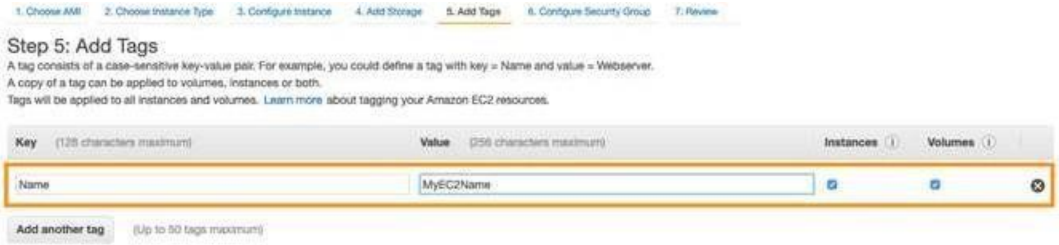
Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. Learn more about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-0740be58f4d41f11a	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted
ephemeral0	/dev/nvme0n1	N/A	100	NVMe SSD	N/A	N/A	N/A	Hardware Encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. Learn more about free usage tier eligibility and usage restrictions.

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. Add a tag and click Next: Configure Security Group

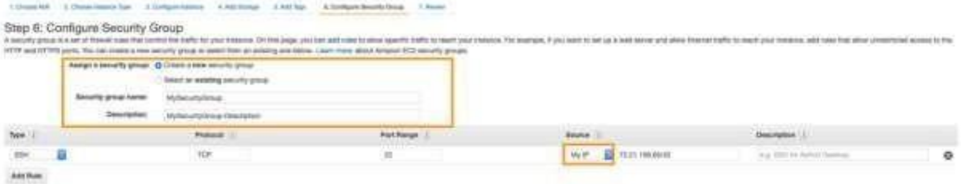


Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. Learn more about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	MyEC2Name	☑	☑

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. Give the Security group a name and Description. Select source as My IP to avoid expose SSH port 22 to the world. Click Review and Launch.

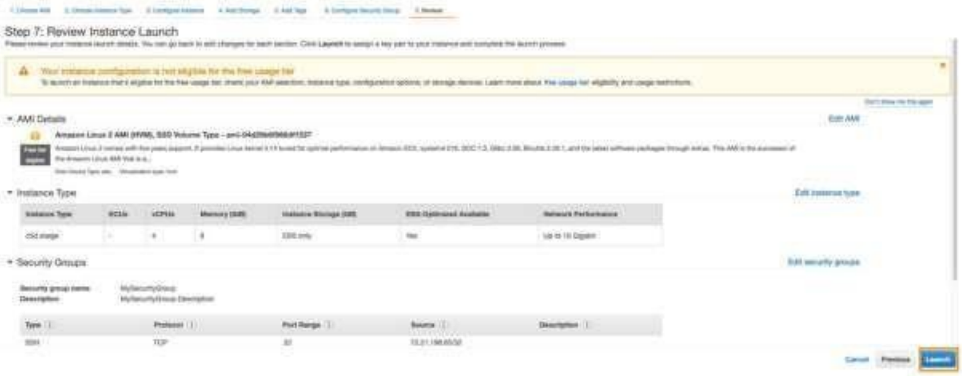


Step 6: Configure Security Group

Assign a security group: ● Create a new security group
○ Select an existing security group
Security group name: MySecurityGroup
Description: MySecurityGroup Description

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	My IP	

Review Instance Launch and click Launch



Step 7: Review Instance Launch

[illegible]

Your instance configuration is not eligible for the free usage tier

[illegible]

AMI Details

Amazon Linux 2 AMI (HVM), SSD Volume Type – [illegible]

[illegible]

Instance Type

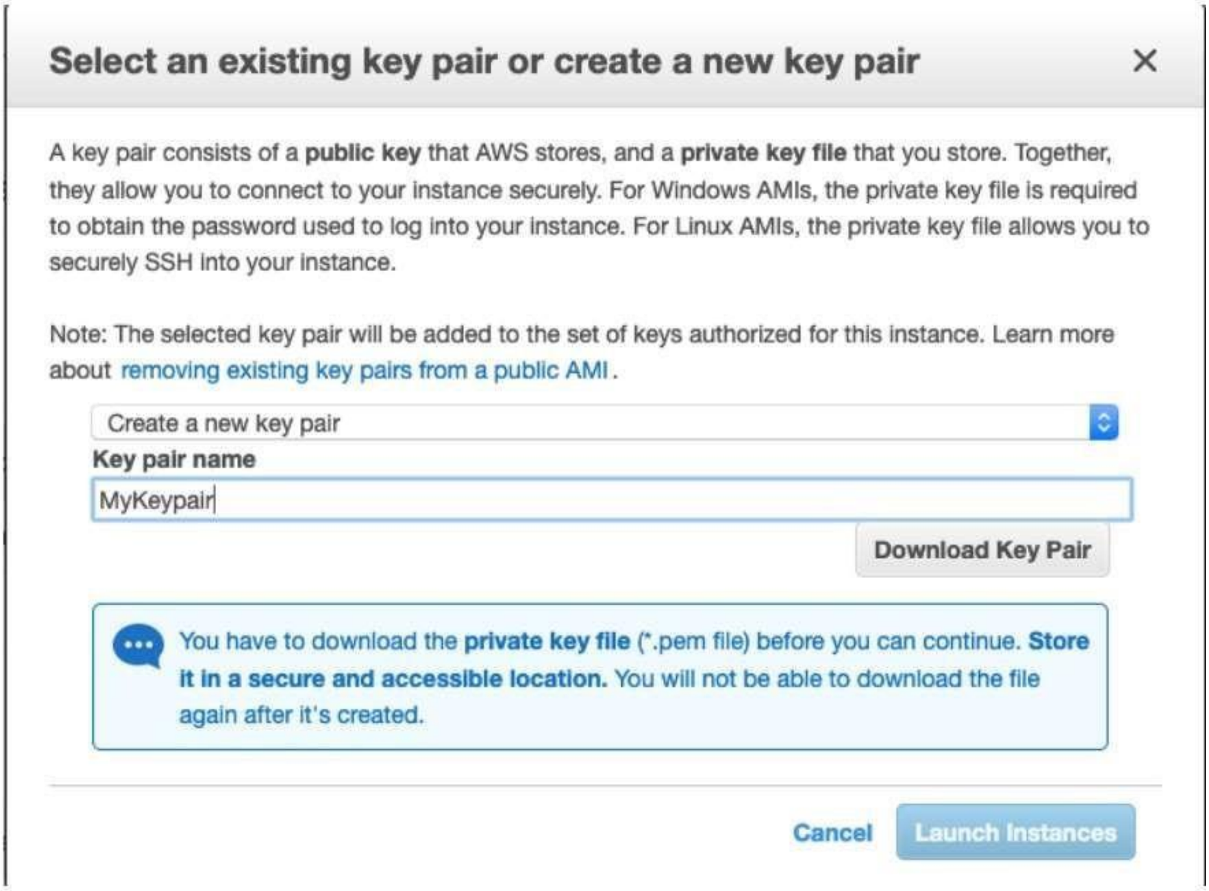
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
[illegible]	-	4	8	[illegible]	Yes	Up to 10 Gigabit

Security Groups

Security group name MySecurityGroup
Description MySecurityGroup Description

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	[illegible]	

In the Select an existing key pair or create a new key pair dialog box, choose Create a new key pair, enter a name for the key pair, and then choose Download Key Pair. This is the only chance for you to save the private key file, so be sure to download it. Save the private key file in a safe place. You can use C:\user\yourusername.ssh\myfirstkey.pem if you are on a Windows machine, and ~/.ssh/myfirstkey.pem if you are on a Mac or Linux machine. You need to provide the name of your key pair when you launch an instance, and the corresponding private key each time you connect to the instance.



Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about removing existing key pairs from a public AMI .

Create a new key pair

Key pair name

MyKeypair

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

A confirmation page lets you know that your instance is launching. Choose View Instances to close the confirmation page and return to the console. On the Instances page, you can view the status of your instance. It takes a short time for an instance to launch. When you launch an instance, its initial state is pending. After the instance starts, its state changes to running, and it receives a public DNS name.

Conclusion:

Performed case study of Amazon web services: Amazon EC2.

Assignment No: 2

Aim:

Case study on Microsoft azure to learn about Microsoft Azure is a cloud computing platform and infrastructure, created by Microsoft, forbuilding, deploying and managing applications and services through a global network of Microsoft-managed datacenters. How it work, different services provided by it.

Objectives:

1. To learn Microsoft Azure Cloud computing platform.
2. To case study the Microsoft Azure cloud services.

Software Requirements:

Ubuntu 18.04
PHP
MySQL

Hardware Requirements:

Pentium IV system with latest configuration

Theory:

Execution Environment

The Windows Azure execution environment consists of a platform for applications and services hosted within one or more roles. The types of roles you can implement in Windows Azure are:

• **Azure Compute (Web and Worker Roles)**. A Windows Azure application consists of one or more hosted roles running within the Azure data centers. Typically there will be at least one Web role that is exposed for access by users of the application. The application may contain additional roles, including Worker roles that are typically used to perform background processing and support tasks for Web roles. For more detailed information see ―Overview of Creating a Hosted Service for Windows Azure‖ athttp://technet.microsoft.com/en-au/library/gg432976.aspx and ―Building an Application that Runs in a Hosted Service‖ at http://technet.microsoft.com/enau/ library/hh180152.aspx.

• **Virtual Machine (VM role)**. This role allows you to host your own custom instance of the Windows Server 2008 R2 Enterprise or Windows Server 2008 R2 Standard operating system within a Windows Azure data center. For more detailed information see ―Creating Applications by Using a VM Role in Windows Azure‖ athttp://technet.microsoft.com/enau/ library/gg465398.aspx.

Data Management

Windows Azure, SQL Azure, and the associated services provide opportunities for storing and managing data in a range of ways. The following data management services and features are available:

- **Azure Storage:** This provides four core services for persistent and durable data storage in the cloud. The services support a REST interface that can be accessed from within Azurehosted or on-premises (remote) applications. For information about the REST API, see ―Windows Azure Storage Services REST API Reference‖ athttp://msdn.microsoft.com/enus/ library/dd179355.aspx.

o **The Azure Table Service** provides a table-structured storage mechanism based on the familiar rows and columns format, and supports queries for managing the data. It is primarily aimed at scenarios where large volumes of data must be stored, while being easy to access and update. For more detailed information see ―Table Service Concepts‖ at http://msdn.microsoft.com/en-us/library/dd179463.aspx and ―Table Service API‖ at http://msdn.microsoft.com/en-us/library/dd179423.aspx.

o **The Binary Large Object (BLOB) Service** provides a series of containers aimed at storing text or binary data. It provides both Block BLOB containers for streaming data, and Page BLOB containers for random read/write operations. For more detailed information see ―Understanding Block Blobs and Page Blobs‖ athttp://msdn.microsoft.com/en-us/library/ee691964.aspx and ―Blob Service API‖ at http://msdn.microsoft.com/en-us/library/dd135733.aspx.

o **The Queue Service** provides a mechanism for reliable, persistent messaging between role instances, such as between a Web role and a Worker role. For more detailed information see ―Queue Service Concepts‖ at http://msdn.microsoft.com/enus/ library/dd179353.aspx and ―Queue Service API‖ athttp://msdn.microsoft.com/enus/ library/dd179363.aspx.

o **Windows Azure Drives** provide a mechanism for applications to mount a single volume NTFS VHD as a Page BLOB, and upload and download VHDs via the BLOB. For more detailed information see ―Windows Azure Drive‖ (PDF) at http://go.microsoft.com/?linkid=9710117.

- **SQL Azure Database:** This is a highly available and scalable cloud database service built on SQL Server technologies, and supports the familiar T-SQL based relational database model. It can be used with applications hosted in Windows Azure, and with other applications running on-premises or hosted elsewhere. For more detailed information see ―SQL Azure Database‖ at http://msdn.microsoft.com/en-us/library/ee336279.aspx.

- **Data Synchronization:** SQL Azure Data Sync is a cloud-based data synchronization service built on Microsoft Sync Framework technologies. It provides bi-directional data synchronization and data management capabilities allowing data to be easily shared between multiple SQL Azure databases and between on-premises and SQL Azure databases. For more detailed information see ―Microsoft Sync Framework Developer Center‖ at http://msdn.microsoft.com/en-us/sync.

- **Caching:** This service provides a distributed, in-memory, low latency and high throughput application cache service that requires no installation or management, and dynamically increases and decreases the cache size automatically as required. It can be used to cache application data, ASP.NET session state information, and for ASP.NET page output caching. For more detailed information see ―Caching Service (Windows Azure AppFabric)‖ at http://msdn.microsoft.com/en-us/library/gg278356.aspx.

Networking Services

Windows Azure provides several networking services that you can take advantage of to Maximize performance, implement authentication, and improve manageability of your hosted applications.

These services include the following:

- **Content Delivery Network (CDN).** The CDN allows you to cache publicly available static data for applications at strategic locations that are closer (in network delivery terms) to end users. The CDN uses a number of data centers at many locations around the world, which store the data in BLOB storage that has anonymous access. These do not need to be locations where the application is actually running. For more detailed information see ―Delivering High-Bandwidth Content with the Windows Azure CDN‖ athttp://msdn.microsoft.com/enus/ library/ee795176.aspx.

- **Virtual Network Connect.** This service allows you to configure roles of an application running in Windows Azure and computers on your on-premises network so that they appear to be on the same network. It uses a software agent running on the on-premises computer to establish an IPsec-protected connection to the Windows Azure roles in the cloud, and provides the capability to administer, manage, monitor, and debug the roles directly. For more detailed information see ―Connecting Local Computers to Windows Azure Roles‖ at http://msdn.microsoft.com/en-us/library/gg433122.aspx.

- **Virtual Network Traffic Manager.** This is a service that allows you to set up request redirection and load balancing based on three different methods. Typically you will use Traffic Manager to maximize performance by redirecting requests from users to the instance in the closest data center using the Performance method. Alternative load balancing methods available are Failover and Round Robin. For more detailed information see ―Windows Azure Traffic Manager‖ at http://msdn.microsoft.com/enus/ WAZPlatformTrainingCourse_WindowsAzureTrafficManager.

- **Access Control.** This is a standards-based service for identity and access control that makes use of a range of identity providers (IdPs) that can authenticate users. ACS acts as a Security Token Service (STS), or token issuer, and makes it easier to take advantage of federation\ authentication techniques where user identity is validated in a realm or domain other than that in which the application resides. An example is controlling user access based on an identity verified by an identity provider such as Windows Live ID or Google. For more detailed information see ―Access Control Service 2.0‖ at http://msdn.microsoft.com/en us/library/gg429786.aspx and ―Claims Based Identity & Access Control Guide‖ at http://claimsid.codeplex.com/.

- **Service Bus.** This provides a secure messaging and data flow capability for distributed and hybrid applications, such as communication between Windows Azure hosted applications and on-premises applications and services, without requiring complex firewall and security infrastructures. It can use a range of communication and messaging protocols and patterns to provide delivery assurance, reliable messaging; can scale to accommodate varying loads; and can be integrated with on-premises BizTalk Server artifacts. For more detailed information see ―AppFabric Service Bus‖ at http://msdn.microsoft.com/en-us/library/ee732537.aspx

Conclusion

Performed case study of Microsoft Azure Cloud computing platform and services.

Assignment No: 3

Aim:

Assignment to install and configure Google App Engine.

Objectives:

1. To learn basics of Google App Engine.
2. To install and configure Google App Engine.

Software Requirements:

Ubuntu 18.04
Python
MySQL

Hardware Requirements:

Pentium IV system with latest configuration

Theory:

Google App Engine is Google's platform as a service offering that allows developers and businesses to build and run applications using Google's advanced infrastructure. These applications are required to be written in one of a few supported languages, namely: Java, Python, PHP and Go. It also requires the use of Google query language and that the database used is Google Big Table. Applications must abide by these standards, so applications either must be developed with GAE in mind or else modified to meet the requirements. GAE is a platform, so it provides all of the required elements to run and host Web applications, be it on mobile or Web. Without this all-in feature, developers would have to source their own servers, database software and the APIs that would make all of them work properly together, not to mention the entire configuration that must be done. GAE takes this burden off the developers so they can concentrate on the app front end and functionality, driving better user experience.

Advantages of GAE include:

- Readily available servers with no configuration requirement
- Power scaling function all the way down to "free" when resource usage is minimal
- Automated cloud computing tools

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 python-docs-sample. Run the command ―git clone https://github.com/GoogleCloudPlatform/python-docs-samples‖.
2. cd to hello world sample by typing the command ― cd pythondocssamples/ appengine/standard/hello_world‖.
3. Then run the command ―dev_appserver.py app.yml‖. 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.

Web output Now you have successfully uploaded your web app into app engine.

Conclusion

Hence we learnt to install and configure Google App Engine.

Assignment No: 4

Aim:

Creating an Application in SalesForce.com using Apex programming Language

Objectives:

3. To learn Sales Force Cloud
4. To study Apex programming Language

Software Requirements:

Ubuntu 18.04
PHP
MySQL
Apex

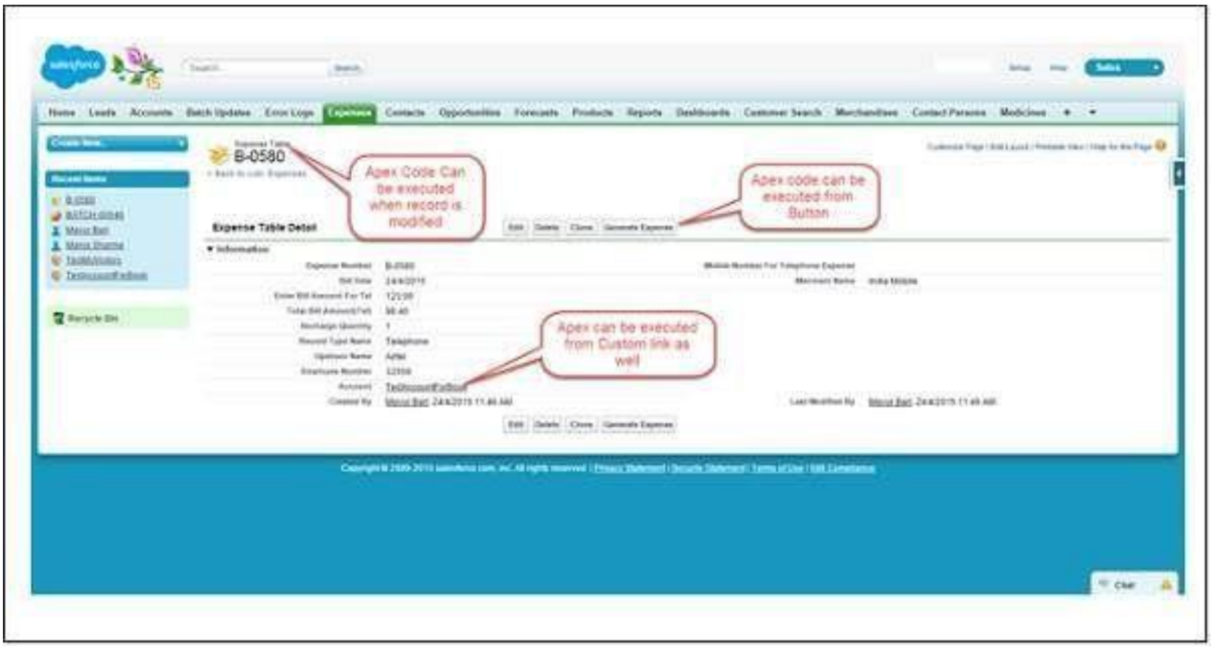
Hardware Requirements:

Pentium IV system with latest configuration

Theory:

What is Apex?

Apex is a proprietary language developed by the Salesforce.com. As per the official definition, Apex is a strongly typed, object-oriented programming language that allows developers to execute the flow and transaction control statements on the Force.com platform server in conjunction with calls to the Force.com API. It has a Java-like syntax and acts like database stored procedures. It enables the developers to add business logic to most system events, including button clicks, related record updates, and Visual force


Home Leads Accounts Batch Updates Error Logs Expenses Contacts Opportunities Forecasts Products Reports Dashboards Customer Search Merchandises Contact Persons Medicines
B-0580
Expense Table Detail
Edit Delete Clone Generate Expenses
Apex Code Can be executed when record is modified
Apex code can be executed from Button
Apex can be executed from Custom link as well

pages. Apex code can be initiated by Web service requests and from triggers on objects. Apex is
included in Performance Edition, Unlimited Edition, Enterprise Edition, and Developer Edition.

Features of Apexasa Language

Let us now discuss the features of Apex as a Language −

- Integrated

Apex has built in support for DML operations like INSERT, UPDATE, DELETE and also DML Exception handling. It has support for inline SOQL and SOSL query handling which returns the set of sObject records. We will study the sObject, SOQL, SOSL in detail in future chapters.

- Java like syntax and easy to use

Apex is easy to use as it uses the syntax like Java. For example, variable declaration, loop syntax and conditional statements.

- Strongly Integrated With Data

Apex is data focused and designed to execute multiple queries and DML statements together.
It issues multiple transaction statements on Database.

- Strongly Typed

Apex is a strongly typed language. It uses direct reference to schema objects like sObject and
any invalid reference quickly fails if it is deleted or if is of wrong data type.

- Multitenant Environment

Apex runs in a multitenant environment. Consequently, the Apex runtime engine is designed to guard closely against runaway code, preventing it from monopolizing sharedresources. Any
code that violates limits fails with easy-to-understand error messages.

- Upgrades Automatically

Apex is upgraded as part of Salesforce releases. We don't have to upgrade it manually.

- Easy Testing

Apex provides built-in support for unit test creation and execution, including test results that
indicate how much code is covered, and which parts of your code can be more efficient.

When Should Developer Choose Apex?

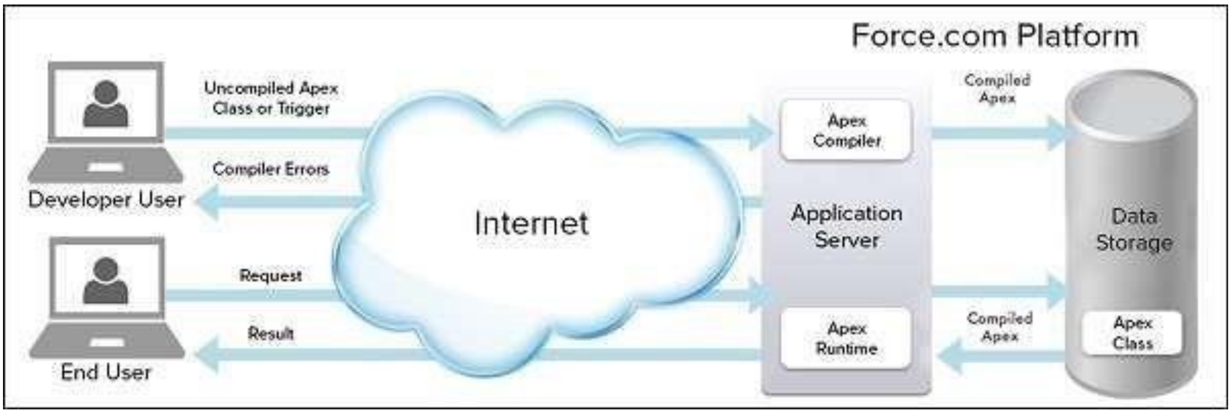
Apex should be used when we are not able to implement the complex business functionality using the pre-built and existing out of the box functionalities. Below are the cases where we need to use apex over Salesforce configuration.

Apex Applications

We can use Apex when we want to − Create Web services with integrating other systems. Create email services for email blast or email setup. Perform complex validation over multiple objects at the same time and also custom validation implementation.
Create complex business processes that are not supported by existing work flow functionality or flows.
Create custom transactional logic (logic that occurs over the entire transaction, not justwith a single record or object) like using the Database methods for updating the records.
Perform some logic when a record is modified or modify the related object's record when there is some event which has caused the trigger to fire.

Working Structure of Apex

As shown in the diagram below (Reference: Salesforce Developer Documentation), Apex run sentirely on demand Force.com Platform



Flowof Actions

There are two sequence of actions when the developer saves the code and when an end user performs some action which invokes the Apex code as shown below –

Developer Action

When a developer writes and saves Apex code to the platform, the platform application server first compiles the code into a set of instructions that can be understood by the Apex runtime interpreter, and then saves those instructions as metadata.

End User Action

When an end-user triggers the execution of Apex, by clicking a button or accessing a Visualforce page, the platform application server retrieves the compiled instructions from the metadata and sends them through the runtime interpreter before returning the result. The end-user observes no differences in execution time as compared to the standard application platform request. Since Apex is the proprietary language of Salesforce.com, it does not support some features which a general programming language does. Following are a few features which Apex does not support − It cannot show the elements in User Interface. You cannot change the standard SFDC provided functionality and also it is not possible to prevent the standard functionality execution. You cannot change the standard SFDC provided functionality and also it is not possible to prevent the standard functionality execution.

Creating multiple threads is also not possible as we can do it in other languages.

Understanding the Apex Syntax

Apex code typically contains many things that we might be familiar with from other programming languages.

Variable Declaration

As strongly typed language, you must declare every variable with data type in Apex. As seen in the code below (screenshot below), lst Acc is declared with data type as List of Accounts.

SOQL Query

This will be used to fetch the data from Salesforce database. The query shown in screenshot below is fetching data from Account object.

Loop Statement

This loop statement is used for iterating over a list or iterating over a piece of code for a specified number of times. In the code shown in the screenshot below, iteration will be same as the number of records we have.

Flow Control Statement

The If statement is used for flow control in this code. Based on certain condition, it is decided whether to go for execution or to stop the execution of the particular piece of code. For example, in the code shown below, it is checking whether the list is empty or it contain srecords.

DML Statement

Performs the records insert, update, upsert, delete operation on the records in

Apex Code Development Tools

In all the editions, we can use any of the following three tools to develop the code – Force.com Developer Console Force.com IDE Code Editor in the Salesforce User Interface

Conclusion:

Hence we learnt to create an Application in SalesForce.com using Apex programming Language

Assignment No: 5

Aim:

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

Objectives:

1. To design application custom Application.
2. To develop application using sales force cloud

Software Requirements:

Ubuntu 18.04
PHP/python
MySQL
Apex

Hardware Requirements:

Pentium IV system with latest configuration

Theory:

Introduction

Salesforce.com Inc. is an American cloud-based software company headquartered in San Francisco, California. Though the bulk of its revenue comes from a customer relationship management (CRM) product, Salesforce also sells a complementary suite of enterprise applications focused on customer service, marketing automation, analytics and application development. Salesforce is the primary enterprise offering within the Salesforce platform. It provides companies with an interface for case management and task management, and a system for automatically routing and escalating important events. The Salesforce customer portal provides customers the ability to track their own cases, includes a social networking plug-in that enables the user to join the conversation about their company on social networking websites, provides analytical tools and other services including email alert, Google search, and access to customers' entitlement and contracts.

Lightning Platform

Lightning Platform (also known as Force.com) is a platform as a service (PaaS) that allows developers to create add-on applications that integrate into the main Salesforce.com application. These third party applications are hosted on Salesforce.com's infrastructure. Force.com applications are built using declarative tools, backed by Lightning and Apex (a proprietary Java-like programming language for Force.com) and Lightning and Visual force (a framework that includes an XML syntax typically used to generate HTML). The Force.com platform typically receives three complete releases a year. As the platform is provided as a service to its developers, every single development instance also receives all these updates.

Community Cloud

Community Cloud provides Salesforce customers the ability to create online web properties for external collaboration, customer service, channel sales, and other custom portals in their instance of Salesforce. Tightly integrated to Sales Cloud, Service Cloud, and App Cloud, Community Cloud can be quickly customized to provide a wide variety of web properties

Salesforce Sales Cloud

Salesforce Sales Cloud is a customer relationship management (CRM) platform designed to support sales, marketing and customer support in both business-to-business (B2B) and business-to-customer (B2C) contexts. Sales Cloud is a fully customizable product that brings all the customer information together in an integrated platform that incorporates marketing, lead generation, sales, customer service and business analytics and provides access to thousands of applications through the AppExchange. The platform is provided as Software as a Service (SaaS) for browser-based access; a mobile app is also available. A real-time social feed for collaboration allows users to share information or ask questions of the user community.Salesforce.com offers five versions of Sales Cloud on a per-user, per month basis, from lowest to highest: Group, Professional, Enterprise, Unlimited and Performance. The company offers three levels of support contracts: Standard Success Plan, Premier Success Plan and Premier+ Success Plan.

Create Custom Apps for Salesforce Classic

Create custom apps to give your Salesforce Classic users' access to everything they need all in one place. If you're new to custom apps, we recommend using Lightning Platform quick start to create an app. With this tool, you can generate a basic working app in just one step. If you've already created the objects, tabs, and fields you need for your app, follow these steps. With this option, you create an app label and logo, add items to the app, and assign the app to profiles.

1. From Setup, enter Apps in the Quick Find box, then select **Apps**.
2. Click **New**.
3. If the Salesforce console is available, select whether you want to define a custom app or a Salesforce console.
4. Give the app a name and description.
   An app name can have a maximum of 40 characters, including spaces.
5. Optionally, brand your app by giving it a custom logo.
6. Select which items to include in the app.
7. Optionally, set the default landing tab for your new app using the **Default Landing Tab** dropdown menu below the list of selected tabs. This determines the first tab a user sees when logging into this app.
8. Choose which profiles the app will be visible to.
9. Check the Default box to set the app as that profile's default app, meaning that new users with the profile see this app the first time they log in. Profiles with limits are excluded from this list.
10. Click **Save**

What is the difference between custom application and console application in sales force?

A custom application is a collection of tabs, objects etc that function together to solve a particular problem. A console application uses a specific Salesforce UI - the console. Console applications are intended to enhance productivity by allowing everything to be done from a single, tabbed, screen**.**

Conclusion:

Hence we learnt to design and develop custom Application (Mini Project) using Sales force Cloud

Mini-Project

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.

https://www.tecmint.com/openstack-installation-guide-rhel-centos/
To set static IP address
http://www.mustbegeek.com/configure-static-ip-address-in-centos/
https://www.tecmint.com/create-deploy-and-launch-virtual-machines-in-openstack/
https://www.tecmint.com/openstack-networking-guide/

configure Sahara in Openstack