Report No: 01
Report Name: WAP to find shortest path using Dijkstra's Algorithm

Code:

```
#include<iostream>
#include <limits.h>
using namespace std;

int miniDist(int distance[], bool Tset[])
{
   int minimum=INT_MAX,ind;

   for(int k=0;k<6;k++)
   {
      if(Tset[k]==false && distance[k]<=minimum)
      {
         minimum=distance[k];
         ind=k;
      }
   }
   return ind;
}

void dijkstra(int graph[6][6],int src){
   int distance[6];
   bool Tset[6];


   for(int k = 0; k<6; k++)
   {
      distance[k] = INT_MAX;
      Tset[k] = false;
   }

   distance[src] = 0;

   for(int k = 0; k<6; k++)
   {
      int m=miniDist(distance,Tset);
      Tset[m]=true;
      for(int k = 0; k<6; k++)
      {
```

```
        if(!Tset[k] && graph[m][k] && distance[m]!=INT_MAX &&
distance[m]+graph[m][k]<distance[k])
            distance[k]=distance[m]+graph[m][k];
      }
   }
   cout<<"Vertex\t\tDistance from source vertex"<<endl;
   for(int k = 0; k<6; k++)
   {
      char str=65+k;
      cout<<str<<"\t\t\t"<<distance[k]<<endl;
   }
}

int main()
{
   int graph[6][6]={
      {0, 1, 2, 0, 0, 0},
      {1, 0, 0, 5, 1, 0},
      {2, 0, 0, 2, 3, 0},
      {0, 5, 2, 0, 2, 2},
      {0, 1, 3, 2, 0, 1},
      {0, 0, 0, 2, 1, 0}};
   dijkstra(graph,0);
   return 0;
}
```

Output:

Vertex	Distance from source vertex
A	0
B	1
C	2
D	4
E	2
F	3