**SVKM’s NMIMS**

**Mukesh Patel School of Technology Management & Engineering**

**Computer Engineering Department**

Program: B.Tech. Sem III

**Subject: Data Structures Division: E**

**Mini - Project Report**

|  |  |  |
| --- | --- | --- |
| Name of the Project: | Uber | |
|  | | |
| Details of Project Members |  |  |
| Batch | Roll No. | Name |
| E2 | E025 | DHWANI SHAH |
| E2 | E030 | NISHEE SHELAT |
|  |  |  |
| Date of Submission: 2/10/19 | | |
| Grades/Marks: | | |

A.1 **Aim of the Project:**

Aim of the project was to use Dijkstra’s algorithm to find shortest distance from location of user to the destination by Uber.

A.2 **Application of the Project:**

The project first focuses on where the user wants to go as per the locations displayed by uber. Uber as a form of service, is used worldwide and has a high success review. This might be because not only does it take you from one place to another in a vehicle of your own choice but it also is cheaper compared to travelling in your own vehicle. One of its key advantage is to reduce Greenhouse gases emitted by vehicles, considering it is one of the main reason for global warming.

A.3 **Data Structures used and justification:**

The main data structure used is graph. By the usage of vertices of graph locations, destinations and drivers can be assigned. But the graph is created by using two-dimensional arrays and one-dimensional arrays are used for Dijkstra’s algorithm.

A.4 **Description of project:**

(Describe various modules and function of each module)

**Graph:**

* We have created a 20x20 two dimensional array for graph wherein locations, destinations and drivers are assigned to each vertices and pre-defined. On this matrix by this two dimensional, we use the greedy approach of Dijkstra’s algorithm using three arrays: visited, distance and parent.

**Uber:**

* Uber’s first job is to ask the user to enter any one of the locations mentioned in the list and it’s second job is to ask the user for a destination. It checks whether the location and destination are appropriate or not and displays the same.
* After the assignment of location and destination, uber applies Dijkstra’s on the two dimensional graph matrix, which consists of two for loops inside a while loop, in which for every while loop there will be a new source but starting with the location as the first source and every source once assigned will have been visited , and in every for loop parent will be assigned as the source of while loop and in the same for loop distance is assigned by checking the shortest distance from source to the said vertex if any. And in the second for loop source is assigned to the vertex that has not being visited and has minimum distance.
* Once uber have the shortest distance, uber displays which driver is inside 10 mins radius and what time it will reach the user.
* Uber then gives user the choice to select cab based on type of functionality it has and its fare. Once the cab is selected, user is asked to apply for discount coupons if any and new fare is displayed.
* Confirmation of booking is asked, at the last, if user decides to use the Uber services, details of driver and cab model are printed and mode of payment is asked but if confirmation is denied, we ask for reason of rejection to improve Uber services.

**User:**

* As any application, here, the job of user is as minimal as possible.
* User first has to select location and then has to select destination.
* User will be showed as to how far the driver is, how far the destination is and is asked to select a type of cab after knowing it’s functionalities and fare.
* If the user after knowing selects a cab then user is asked of it wants to apply discount coupons and then is asked to confirm ride.
* If the user decides to confirm ride, uber would like to know their mode of payment and if denied confirmation, uber would like review from the User.

A.5 **Contribution of each project Members:**

|  |  |  |
| --- | --- | --- |
| Roll No: E025 | Name:  DHWANI SHAH | Contribution:  1.Decided number of functions.  2.Hepled in the GUI of the code.  3.Helped in the process of finding path for dijkstra’s algorithm. |
| E030 | NISHEE SHELAT | 1.Helped in writing the code for Dijkstra’s algorithm.  2.Helped in make the graph for uber.  3.Helped in writing the different functionalities of the uber. |

A.6 **Properly commented Software Code:**

**Clearly mention the purpose of each function. No single letter variable names.**

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

using namespace std;

int graph[20][20]={0};

int visited[20]={0};

int parent[20]={0};

void assign\_driver(int,int);

int assign\_destination();

int assign\_location();

void assign\_driver(int i,int j)

{

int c1,c,c2,min=10,minimum,temp,counter=20,time,charge;

string choice1="0",choice="0",payment="0";

string discount="0";

string reason;

int distance[20]={0};

for(c=0;c<20;c++) //assigning the driver which is in 10 i.e.min distance from location of user

{

if(graph[i][c]<min)

{

min=graph[i][c];

}

}

for(c1=0;c1<20;c1++) //assigining maximum distance for dijkstra algorithm

{

distance[c1]=236;

}

c1=i;

distance[c1]=0;

while(counter!=0) //dijkstra algorithm start

{

minimum=236;

visited[c1]=1;

for(c2=0;c2<20;c2++)

{

if(graph[c1][c2]!=236)

{

temp=graph[c1][c2]+distance[c1];

if(temp<distance[c2]&&visited[c2]!=1)

{

distance[c2]=temp;

parent[c2]=c1+1;

}

}

}

for(c2=0;c2<20;c2++)

{

if((distance[c2]<minimum||distance[c2]==minimum)&&(visited[c2]!=1))

{

minimum=distance[c2];

c1=c2;

}

}

counter--;

} //dikjstra algorithm end

time=distance[j];

if(distance[j]<=10) //calculating charge for ride

charge=100;

else if(distance[j]<=20)

charge=200;

else if(distance[j]<=30)

charge=300;

else

charge=500;

cout<<"\t\t\t\t\t Time for driver to reach you: "<<min<<" "<<"minutes"<<endl<<endl;

cout<<"\t\t\t\t\t Time for you to reach your destination: "<<time<<" "<<"minutes"<<endl<<endl;

do

{

cout<<"\t\t\t\t\t Types of cab: "<<endl<<endl; //displaying types of ride available

cout<<"\t\t\t\t\t 1.UberGo "<<endl<<endl;

cout<<"\t\t\t\t\t Affordable,compact rides"<<endl<<endl;

cout<<"\t\t\t\t\t Capacity: 4 People"<<endl<<endl;

cout<<"\t\t\t\t\t Fare: "<<charge<<" "<<"Rupees"<<endl<<endl;

cout<<"\t\t\t\t\t 2.Uber Premier"<<endl<<endl;

cout<<"\t\t\t\t\t Comfortable sedans "<<endl<<endl;

cout<<"\t\t\t\t\t Capacity: 4 People"<<endl<<endl;

cout<<"\t\t\t\t\t Fare: "<<charge + 100<<" "<<"Rupees"<<endl<<endl;

cout<<"\t\t\t\t\t Select cab"<<endl<<endl;

cout<<"\t\t\t\t\t ";

cin>>choice1;

system("cls");

if(choice1!="1"&&choice1!="2")

{

choice1="0";

cout<<"\t\t\t\t\t Invalid Choice"<<endl<<endl;

}

}while(choice1=="0");

do

{

cout<<"\t\t\t\t\t Do you have any discount coupons to apply? "<<endl<<endl; //checking for discount coupons

cout<<"\t\t\t\t\t 1.Yes 2.No"<<endl<<endl;

cout<<"\t\t\t\t\t ";

cin>>discount;

system("cls");

if(discount=="1")

{

cout<<"\t\t\t\t\t New Fare: "<<charge-20<<" "<<"Rupees"<<endl<<endl;

}

if (discount=="2")

{

cout<<"\t\t\t\t\t Fare: "<<charge<<" "<<"Rupees"<<endl<<endl;

}

if(discount!="2"&&discount!="1")

{

cout<<"\t\t\t\t\t Invalid Choice"<<endl<<endl;

discount="0";

}

}while(discount=="0");

do

{

cout<<"\t\t\t\t\t Do you want to confirm booking?"<<endl<<endl; //confirming booking after showing fare for the ride

cout<<"\t\t\t\t\t 1.Yes 2.No"<<endl<<endl;

cout<<"\t\t\t\t\t ";

cin>>choice;

system("cls");

if(choice=="1") //displaying driver details sfter confirmation

{

if(choice1=="1") //driver depends on the type of cab

{

cout<<"\t\t\t\t\t Driver's name: Suchit Singh"<<endl<<endl;

cout<<"\t\t\t\t\t Driver's contact number: 8369066288"<<endl<<endl;

cout<<"\t\t\t\t\t Car license number: MH03AR4961"<<endl<<endl;

cout<<"\t\t\t\t\t Car model: Swift"<<endl<<endl;

}

else

{

cout<<"\t\t\t\t\t Driver's name: Himanshu Pahwa"<<endl<<endl;

cout<<"\t\t\t\t\t Driver's contact number: 7977046779"<<endl<<endl;

cout<<"\t\t\t\t\t Car license number: MH03AM7624"<<endl<<endl;

cout<<"\t\t\t\t\t Car model: Innova"<<endl<<endl;

}

}

else if(choice=="2") //asking for reason of cancellation to improve services

{

cout<<"\t\t\t\t\t Please list your reason for rejection"<<endl<<endl;

cout<<"\t\t\t\t\t ";

cin>>reason;

system("cls");

cout<<"\t\t\t\t\t Thank you for visiting our sevices"<<endl<<endl;

exit(0);

}

else if(choice!="2"&&choice!="1")

{

cout<<"\t\t\t\t\t Invalid Choice"<<endl<<endl;

choice="0";

}

}while(choice=="0");

do

{ //asking for type of payment after confirming

cout<<"\t\t\t\t\t Choose mode of payment: "<<endl<<endl;

cout<<"\t\t\t\t\t 1.Paytm"<<endl<<endl;

cout<<"\t\t\t\t\t 2.Paypal"<<endl<<endl;

cout<<"\t\t\t\t\t 3.Google Wallet"<<endl<<endl;

cout<<"\t\t\t\t\t 4.Net Banking"<<endl<<endl;

cout<<"\t\t\t\t\t 5.Cash"<<endl<<endl;

cout<<"\t\t\t\t\t ";

cin>>payment;

system("cls");

if(payment!="1"&&payment!="2"&&payment!="3"&&payment!="4"&&payment!="5")

{

cout<<"\t\t\t\t\t Invalid Choice"<<endl<<endl;

payment="0";

}

} while(payment=="0");

cout<<"\t\t\t\t\t Hope you have a safe and happy ride!"<<endl<<endl;

}

int assign\_location()

{

string location;

int i;

string choice="1";

do{

cout<<"\t\t\t\t\t\t\tUBER SERVICES"<<endl<<endl;

cout<<"\t\t\t\t\t \*\*\*At your beck and call\*\*\*"<<endl<<endl;

cout<<"\t\t\t\t\t Select your location number"<<endl<<endl; //displaying locations available

cout<<"\t\t\t\t\t 1.Borivali"<<endl<<endl;

cout<<"\t\t\t\t\t 2.Kandivali"<<endl<<endl;

cout<<"\t\t\t\t\t 3.Kurla"<<endl<<endl;

cout<<"\t\t\t\t\t 4.Ghatkopar"<<endl<<endl;

cout<<"\t\t\t\t\t 5.Andheri"<<endl<<endl;

cout<<"\t\t\t\t\t 6.Bandra"<<endl<<endl;

cout<<"\t\t\t\t\t 7.Vile Parle"<<endl<<endl;

cout<<"\t\t\t\t\t ";

cin>>location;

system("CLS");

if(location=="1") //checking whether input is valid or not

{

i=1;

choice="0";

return i;

}

else if(location=="2")

{

i=8;

choice="0";

return i;

}

else if(location=="3")

{

i=3;

choice="0";

return i;

}

else if(location=="4")

{

i=7;

choice="0";

return i;

}

else if(location=="5")

{

i=10;

choice="0";;

return i;

}

else if(location=="6")

{

i=14;

choice="0";

return i;

}

else if(location=="7")

{

i=18;

choice="0";

return i;

}

else

{

while(choice!="2")

{

cout<<"\t\t\t\t\t \*\*Sorry\*\*"<<endl<<endl; //if the input is not valid

cout<<"\t\t\t\t\t \*\*Location not found\*\*"<<endl<<endl;

cout<<"\t\t\t\t\t Do you want to change location? "<<endl<<endl;

cout<<"\t\t\t\t\t 1.Yes 2.No"<<endl<<endl;

cout<<"\t\t\t\t\t ";

cin>>choice;

system("CLS");

if(choice=="1")

break;

}

}

}while(choice=="1");

if(choice=="2") //if the user decides to not use the application

{

cout<<"\t\t\t\t\t Sorry for the inconvenience caused"<<endl<<endl;

cout<<"\t\t\t\t\t Thank you for using our services"<<endl<<endl;

exit(0);

}

}

int assign\_destination(int i)

{

string destination;

int j=0;

string choice="1";

do{

cout<<"\t\t\t\t\t Select your destination number"<<endl<<endl; //displaying destinations available

cout<<"\t\t\t\t\t 1.Borivali"<<endl<<endl;

cout<<"\t\t\t\t\t 2.Kandivali"<<endl<<endl;

cout<<"\t\t\t\t\t 3.Kurla"<<endl<<endl;

cout<<"\t\t\t\t\t 4.Ghatkopar"<<endl<<endl;

cout<<"\t\t\t\t\t 5.Andheri"<<endl<<endl;

cout<<"\t\t\t\t\t 6.Bandra"<<endl<<endl;

cout<<"\t\t\t\t\t 7.Vile Parle"<<endl<<endl;

cout<<"\t\t\t\t\t ";

cin>>destination;

system("CLS");

if(destination=="1") //checking whether input is valid or not

{

j=1;

choice="0";

}

else if(destination=="2")

{

j=8;

choice="0";

}

else if(destination=="3")

{

j=3;

choice="0";

}

else if(destination=="4")

{

j=7;

choice="0";

}

else if(destination=="5")

{

j=10;

choice="0";

}

else if(destination=="6")

{

j=14;

choice="0";

}

else if(destination=="7")

{

j=18;

choice="0";

}

if(i==j) //if destination and location are same

{

cout<<"\t\t\t\t\t Not approriate destination"<<endl<<endl;

cout<<"\t\t\t\t\t Do you want to change Destination? "<<endl<<endl;

cout<<"\t\t\t\t\t 1.Yes 2.No"<<endl<<endl;

cout<<"\t\t\t\t\t ";

cin>>choice;

system("CLS");

}

else if(destination!="7"&&destination!="6"&&destination!="5"&&destination!="4"&&destination!="3"&&destination!="2"&&destination!="1")

{

while(choice!="2")

{

cout<<"\t\t\t\t\t \*\*Sorry\*\*"<<endl<<endl; //if the input is not valid

cout<<"\t\t\t\t\t \*\*Destination not found\*\*"<<endl<<endl;

cout<<"\t\t\t\t\t Do you want to change Destination? "<<endl<<endl;

cout<<"\t\t\t\t\t 1.Yes 2.No"<<endl<<endl;

cout<<"\t\t\t\t\t ";

cin>>choice;

system("cls");

if(choice=="1")

break;

}

}

else

return j;

}while(choice=="1");

if(choice=="2") //if the user decides to not use the application

{

cout<<"\t\t\t\t\t Sorry for the inconvenience caused"<<endl<<endl;

cout<<"\t\t\t\t\t Thank you for using our services"<<endl<<endl;

exit(0);

}

}

int main()

{

int i,j;

system("color 3f");

graph[1][4]=5;

graph[4][1]=5;

graph[1][2]=10;

graph[2][1]=10;

graph[3][2]=5;

graph[2][3]=5;

graph[2][4]=8;

graph[3][6]=15;

graph[3][6]=15;

graph[3][12]=6;

graph[12][3]=6;

graph[4][20]=7;

graph[4][8]=6;

graph[8][4]=6;

graph[4][6]=7;

graph[6][11]=9;

graph[7][5]=10;

graph[8][15]=5;

graph[15][8]=5;

graph[8][9]=16;

graph[9][8]=16;

graph[9][14]=10;

graph[14][9]=10;

graph[9][10]=8;

graph[10][9]=8;

graph[9][16]=20;

graph[10][6]=9;

graph[6][10]=9;

graph[11][12]=5;

graph[11][18]=13;

graph[18][11]=13;

graph[10][13]=5;

graph[13][10]=5;

graph[13][19]=6;

graph[14][17]=8;

graph[17][14]=8;

graph[16][15]=15;

graph[15][7]=9;

graph[7][15]=9;

graph[18][19]=10;

graph[19][18]=10;

graph[19][17]=20;

for(i=0;i<20;i++)

{

for(j=0;j<20;j++)

{

if(graph[i][j]==0)

graph[i][j]=236;

}

}

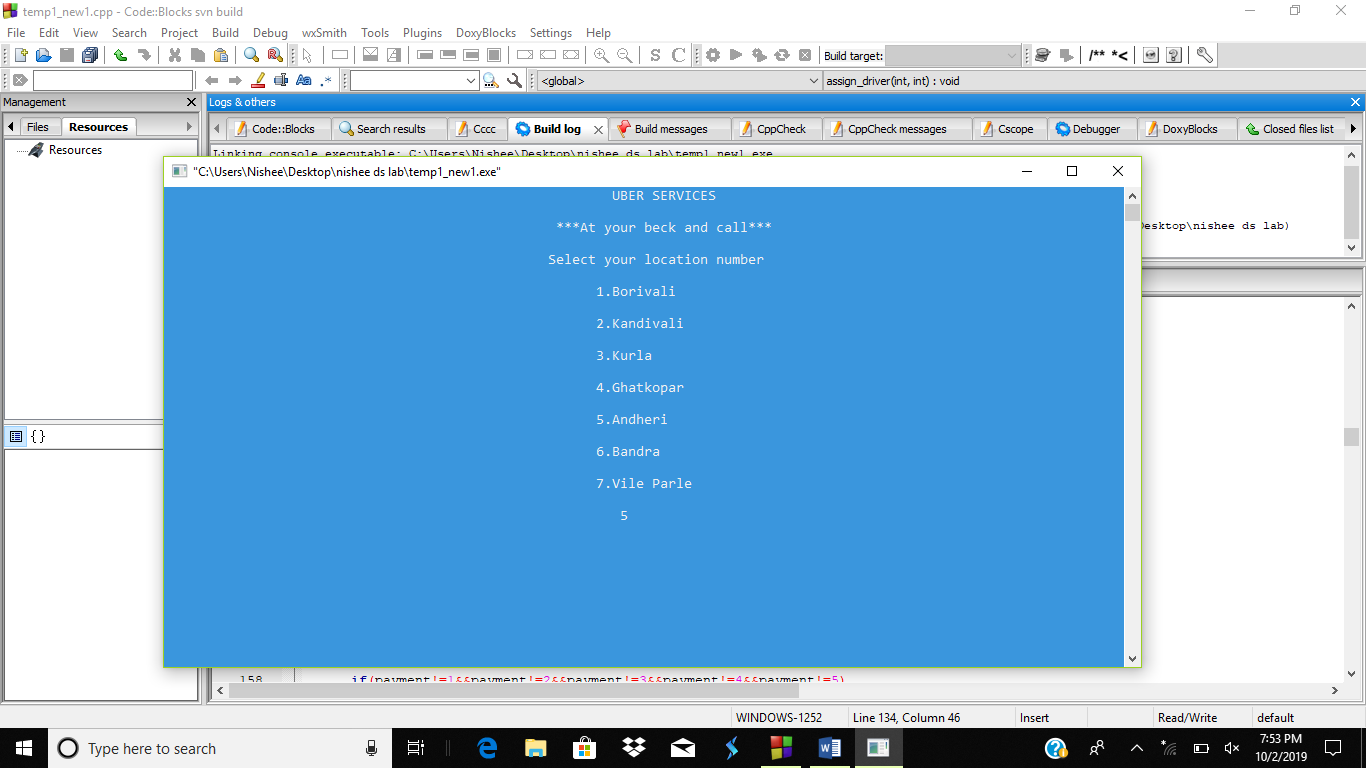
i=assign\_location(); //calling function to take user location

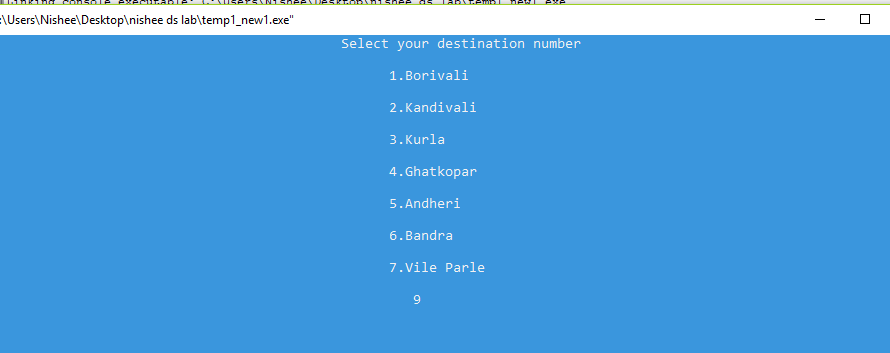
j=assign\_destination(i); //calling function to take user destination

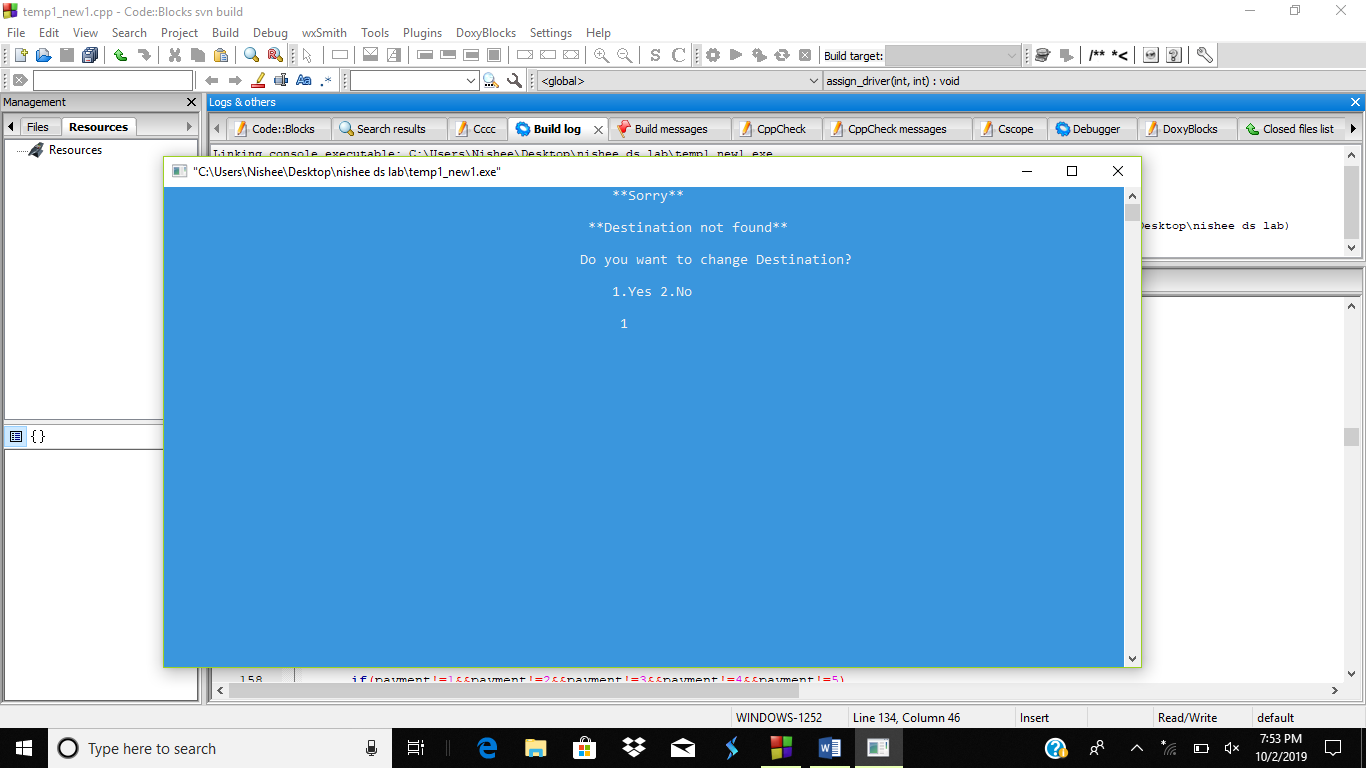
assign\_driver(i,j); //calling function to display necessary details for the ride

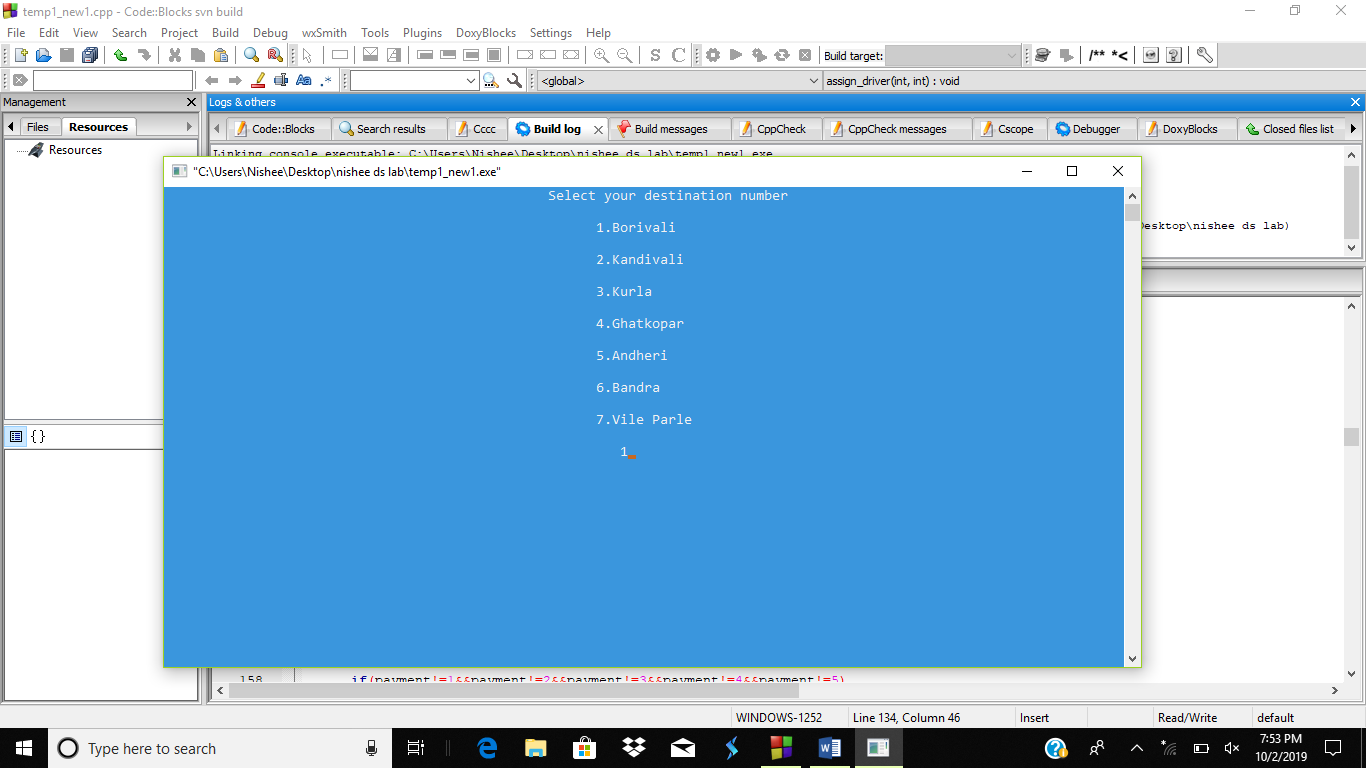
}

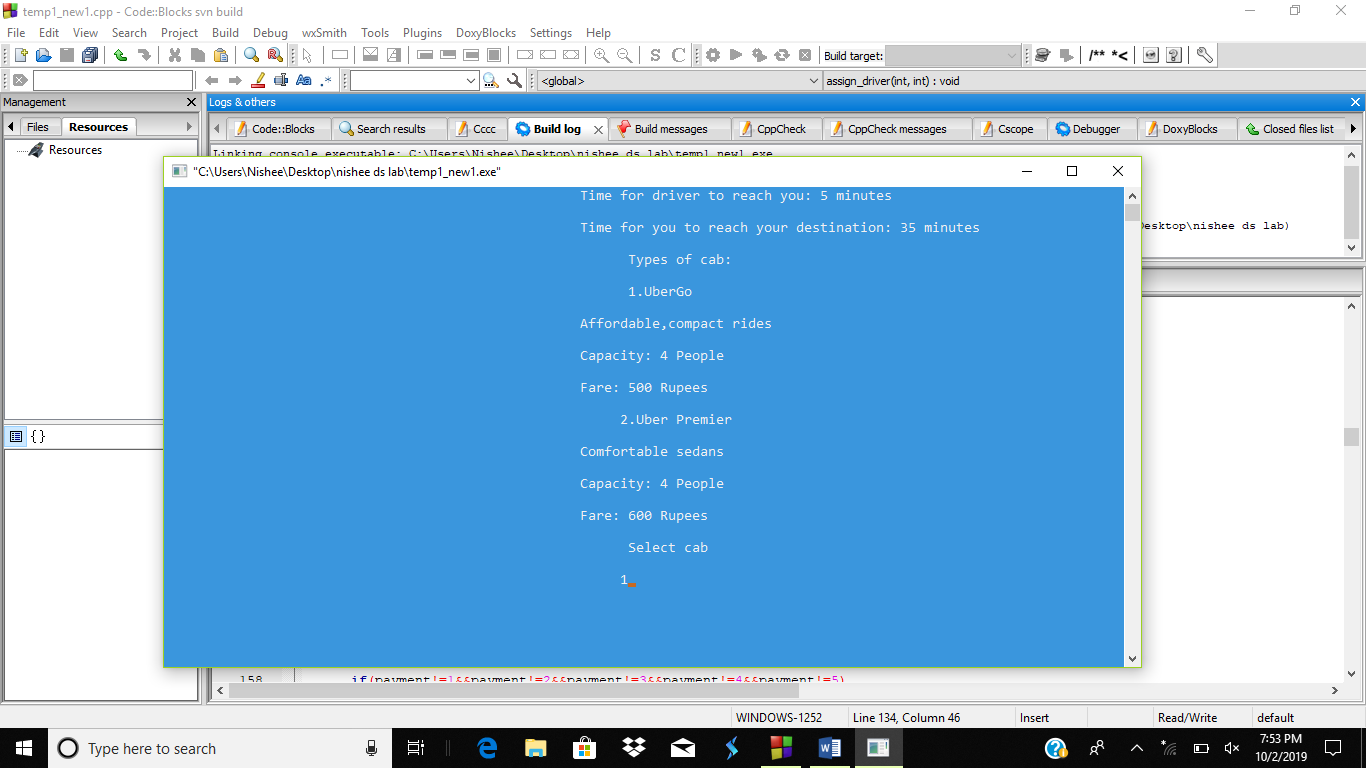
A.7 **Input and Output:**

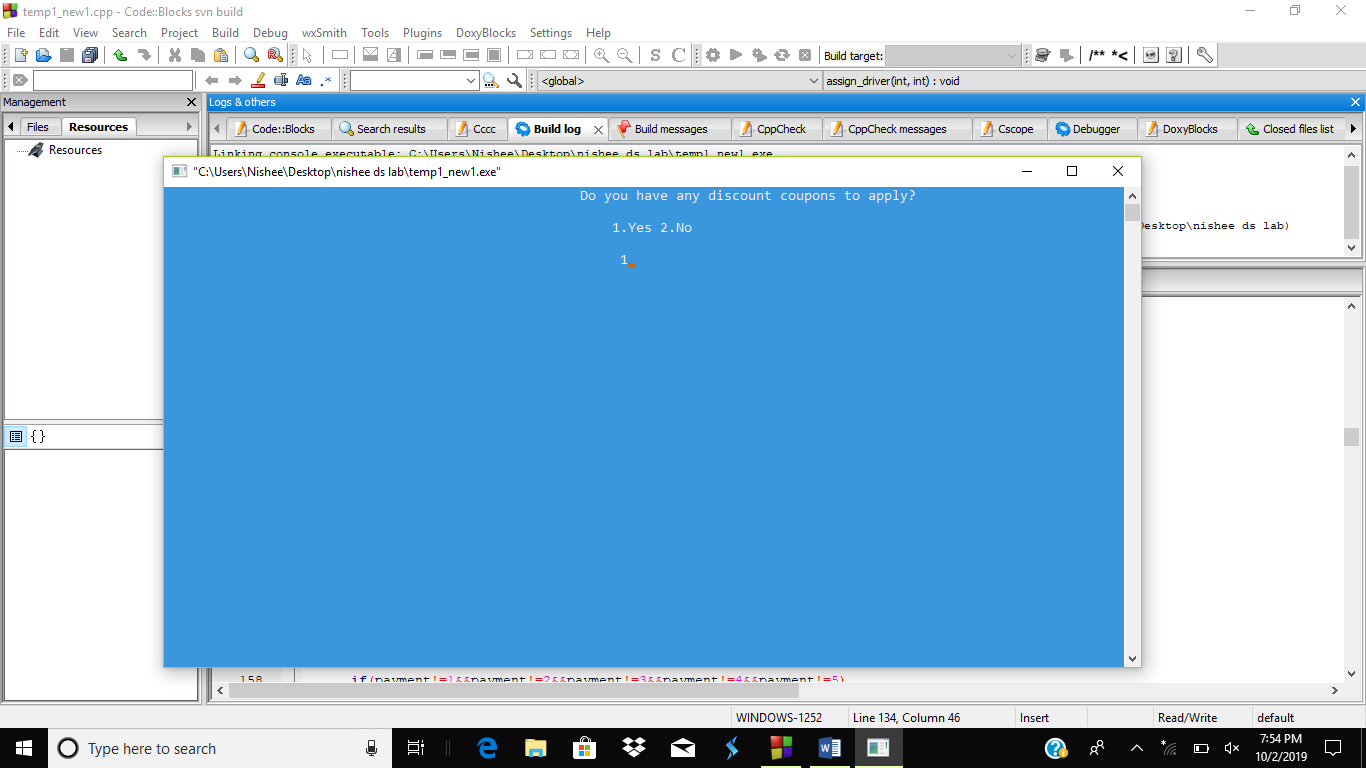


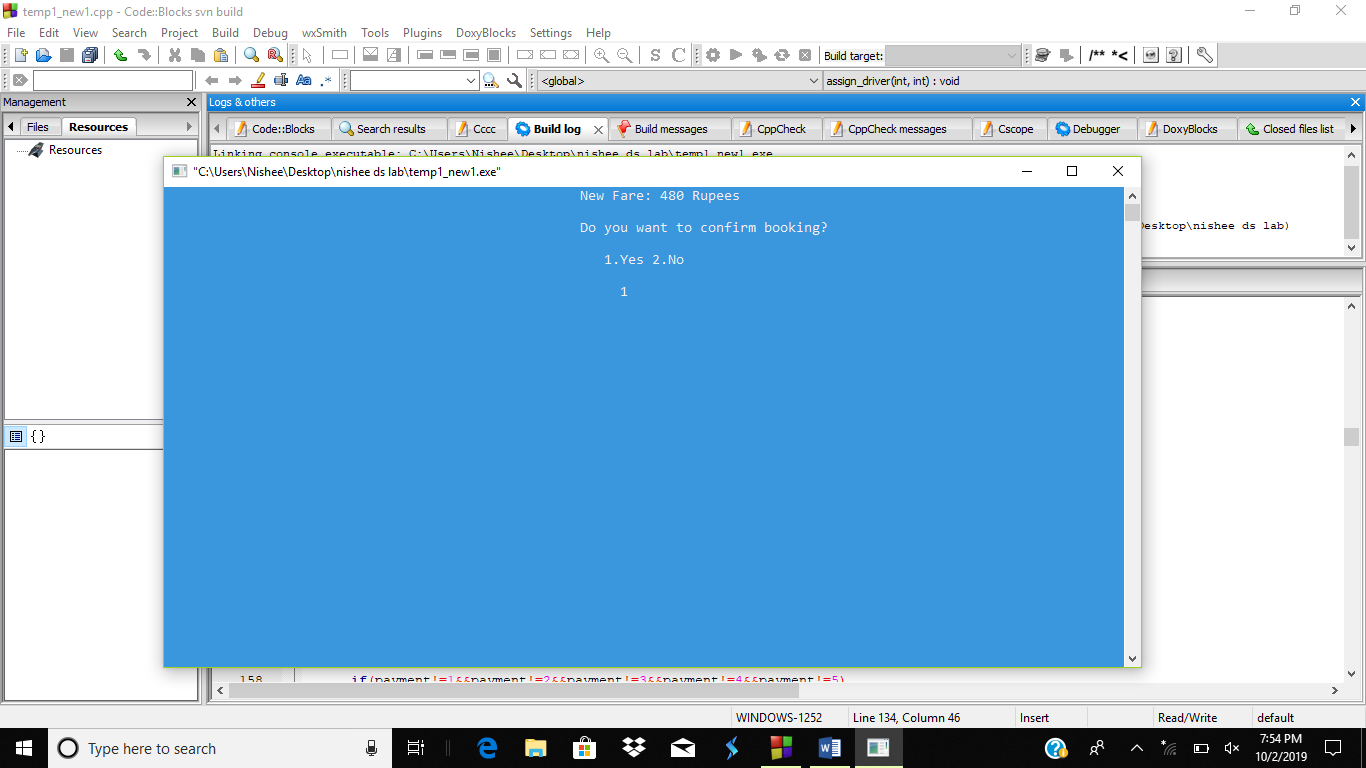


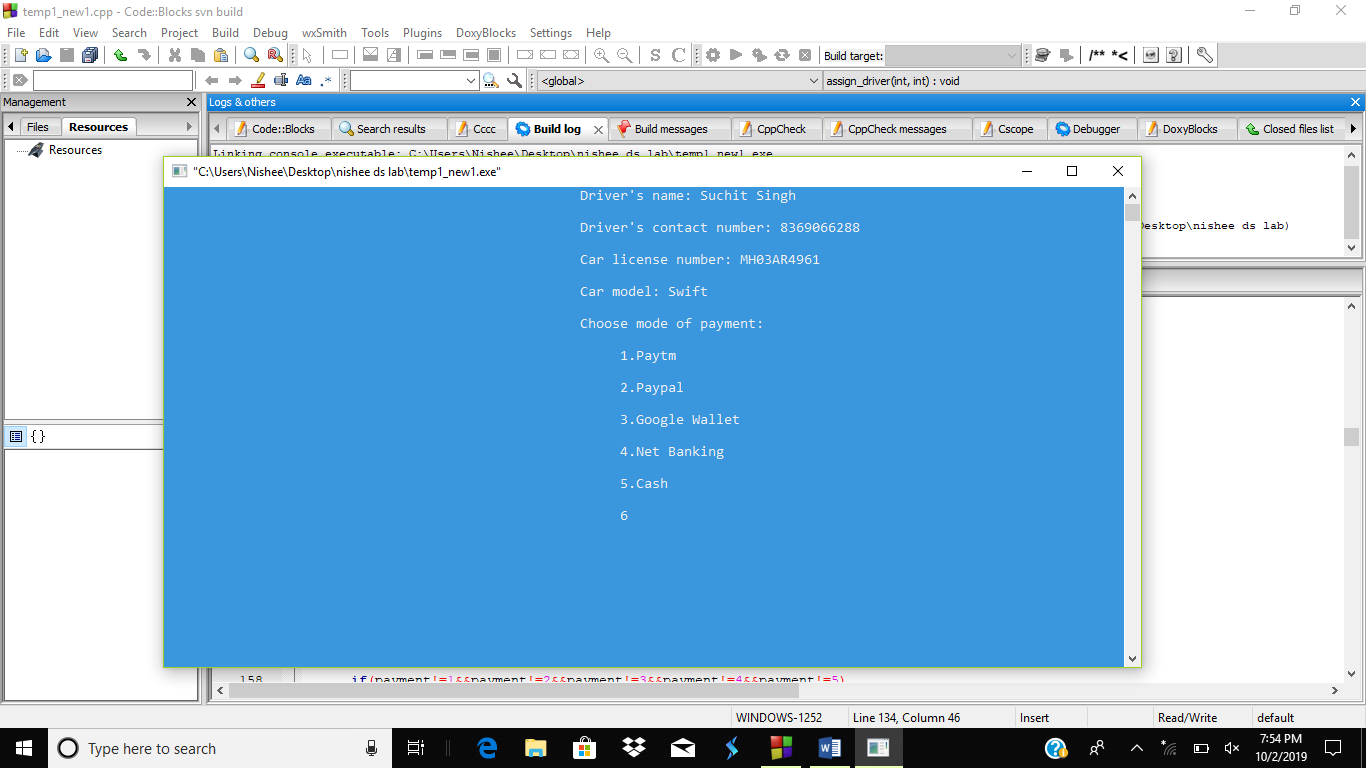


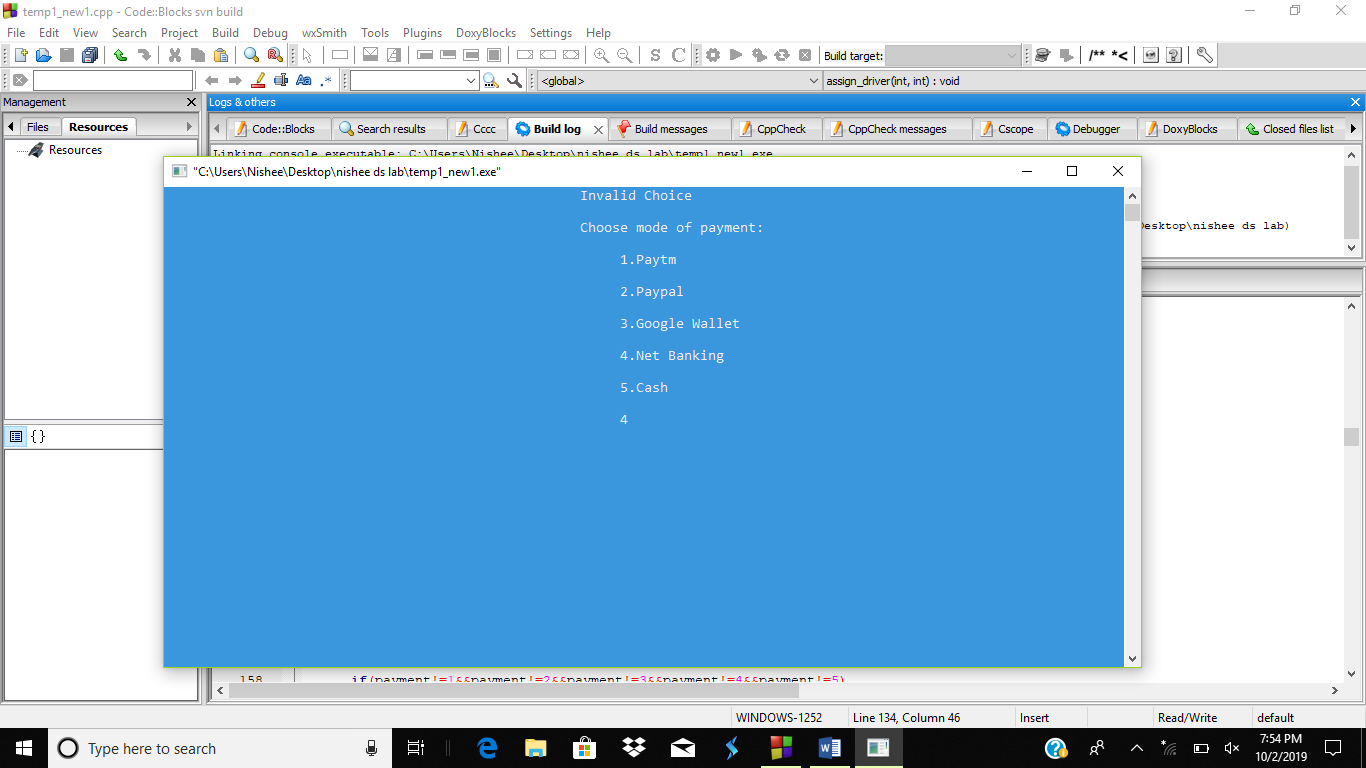


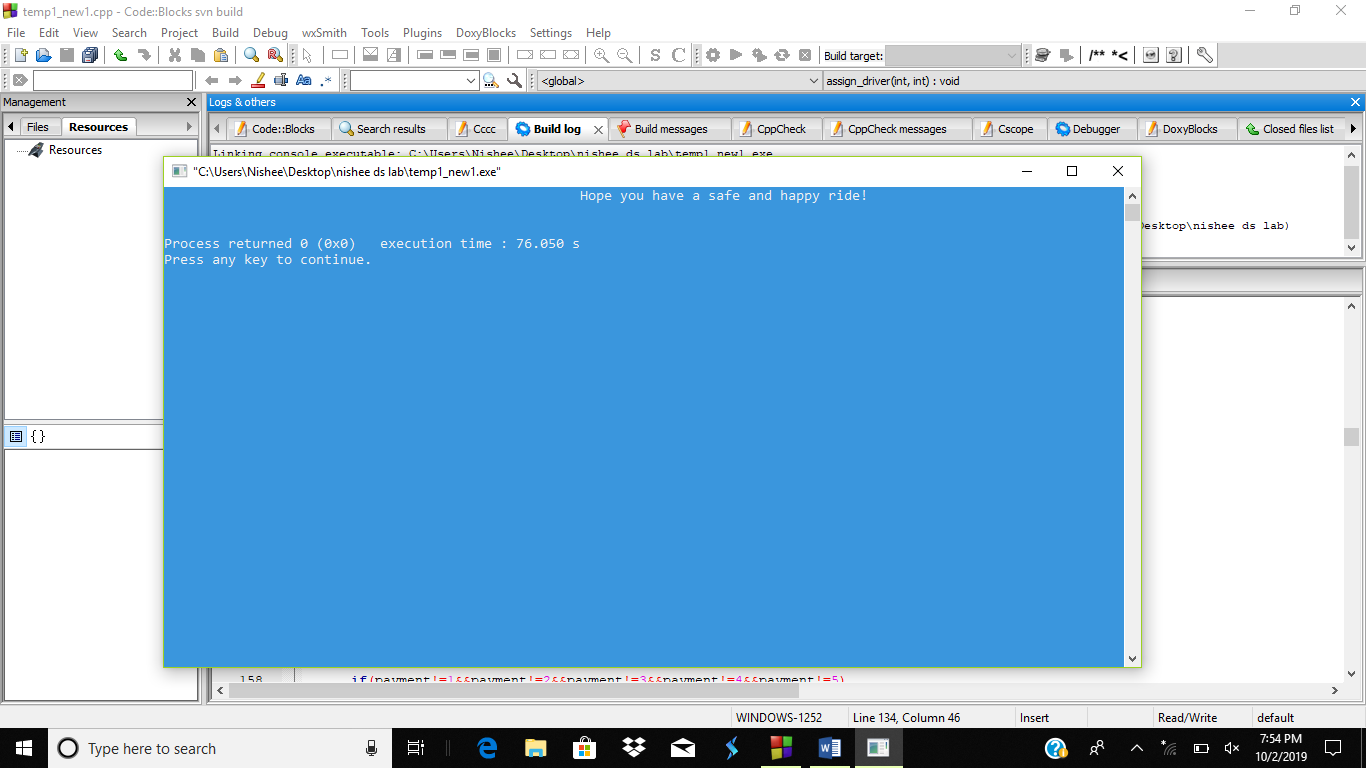












A.8 **Observations and learnings:**

We observed that as challenging as we thought implementing Dijkstra’s might be, along the way we realized that through brainstorming different ideas, we ourselves came up with more than half of solutions for various errors and problems. Through this project , not only did we learn the process of implementing Dijkstra’s algorithm but we learnt the need for team work and got a head start for the upcoming projects.

A.9 **Conclusion:**

We now conclude that impelementation of Dijkstra’s algorithm for finding shortest distance is successfully been implemented in our project.