



NEW HORIZON
COLLEGE OF ENGINEERING

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by **NAAC** with 'A' Grade.

WORLD TOUR

A MINI PROJECT

REPORT

Submitted by

ANIKET KUMAR YADAV

In partial fulfillment for the award of

the degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING



NEW HORIZON
COLLEGE OF ENGINEERING

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade, Accredited by NBA



TOP



ENGINEERING COLLEGE OF INDIA as per
NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Certificate

This is to certify that the mini project work titled

World Tour

Submitted in partial fulfillment of the degree of Bachelor of Engineering

ANIKET KUMAR YADAV

1NH18CS022

DURING

ODD SEMESTER 2019-2020

For

CSE56

Signature of Reviewer

Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner

Signature with date

1. _____

2. _____

CONTENTS

ABSTRACT	I
-----------------	----------

ACKNOWLEDGEMENT	II
------------------------	-----------

LIST OF FIGURES	III
------------------------	------------

1. INTRODUCTION

1.1.PROBLEM DEFINITION	1
1.2. METHODOLOGY TO BE FOLLOWED	1
1.3.METHODOLOGY TO BE FOLLOWED	1
1.4.EXPECTED OUTCOMES	1
1.5. HARDWARE REQUIREMENTS	2
1.6. SOFTWARE REQUIREMENTS	2

2. DATASTRUCTURES

2.1.STACK	3
2.2.QUEUES	4
2.3.LINKED LIST	4-8
2.4.TREES	8
2.5.GRAPHS	9

3. DESIGN

3.1. FLOWCHART	10
3.2. ALGORITHM	11
4. IMPLEMENATION	
4.1. MODULE1 FUNCTIONALITY	12
4.2. MODULE2 FUNCTIONALITY	12-14
4.3. MODULE3 FUNCTIONALITY	14-15
4.4. MODULE4 FUNCTIONALITY	15-18
5. RESULTS	19-21
6. CONCLUSION	22
7.REFERENCES	23

LIST OF FIGURES

<u>Fig. No</u>	<u>Figure Description</u>	<u>Page No</u>
1.1	STACK	3
1.2	QUEUES	4
1.3	LINKED LIST	5
1.4	TREES	7
1.5	GRAPHS	8

ABSTRACT

In previous day people had to visit a lot of places to find a travel agency or agent. The travel agent also sometimes does fraud with the people. The tourist has to wonder to travel the various places around the world. The tourists didn't know about which place they have to visit and what would be the total fare throughout their tour. People had to face a lot of problems in order to book tickets. If they have to visit some places for their entertainment then also, they have to search them. This was not an easy way for people because they had gone to enjoy on their vacation and not to worry about everything and then find out where to go. Outcomes of this project are If a user wants to travel a new place then he would have a knowledge about that place already. The user can book his/her ticket which will take them to different places in the world. This software will also have information about few places in the world where a user can relax and refresh their mood. These few services will be available in this software which will make the work of the user more convenient and easier.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be, but impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned my efforts with success.

I thank the management, **Dr. Mohan Manghnani**, Chairman of NEW HORIZON EDUCATIONAL INSTITUTIONS for providing necessary infrastructure and creating good environment.

I also record here the constant encouragement and facilities extended to me by **Dr. Manjunatha**, Principal, NHCE, **Dr. Prashanth.C.S.R**, Dean Academics, **Dr. B. Rajalakshmi**, Head of the Department of Computer Science and Engineering. I extend my sincere gratitude to them.

I express my gratitude to **Ms. ALPHA VIJAYAN**, Senior Asst. Professor, my project reviewer for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

Finally a note of thanks to all the teaching and non-teaching staff of Computer Science and Engineering Department for their cooperation extended to me and my friends, who helped me directly or indirectly in the course of the project work.

ANIKET KUMAR YADAV

1NH18CS022

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

In previous day people had to visit a lot of places to find a travel agency or agent. The travel agent also sometimes does fraud with the people. The tourist has to wonder to travel the various places around the world. The tourists didn't know about which place they have to visit and what would be the total fare throughout their tour. People had to face a lot of problems in order to book tickets. If they have to visit some places for their entertainment then also, they have to search them. This was not an easy way for people because they had gone to enjoy on their vacation and not to worry about everything and then find out where to go.

1.2 METHODOLOGY TO BE FOLLOWED

The program work on the basic of single linked list to store the details of the users like email and password. If users want to book the ticket or print the ticket then users have to go through the login phase in which they have to enter their username and password. When they are successfully logged in then they can book the ticket for various places of the world.

Structures are also important for the program to store the details. To create the node structures are used in the program and the node are connected using Single linked list. linked list is basic of storing data in "c" because it is easy to access the list and edit the list.

1.3 EXPECTED OUTCOMES

Outcomes of this project are:

- If users want to travel a new place then he would have a knowledge about that place already.
- The users can book his/her ticket which will take them to different places in the world.
- This software will also have information about few places in the world where users can relax and refresh their mood.
- These few services will be available in this software which will make the work of the user more convenient and easier.

1.4 HARDWARE REQUIREMENTS

- Processor : Any processor above 500MHz
- RAM : 8 GB
- Hard Disk : 64 GB
- Input Device : Keyboard and Mouse
- Output Device : High Resolution Monitor

1.5 SOFTWARE REQUIREMENTS

- Operating System : Windows 10
- Developing Environment : Turbo C++ or any another C compile

CHAPTER 2

DATA STRUCTURE

2.1 STACK

Stack is a linear list in which insertions and deletions are allowed only at one end, called top of the stack. We can see examples of stack in our daily life like stack of trays in a cafeteria, stack of books or stack of tennis balls. In all these cases we can see that any object can be removed or added only at the top. The insertion and deletion operations are given special names in the case of stack.

FUNCTIONS OF STACK

- PUSH ()**: The push operation inserts an element in the stack. When the stack is full and popping is not possible, then it is known as Stack-overflow.
- POP ()**: The pop operation is used to remove or deletes or pop out elements from the stack. When the stack is empty and popping is not possible, then it is known as Stack-underflow.
- DISPLAY ()**: This operation is used to display the elements present in the stack.

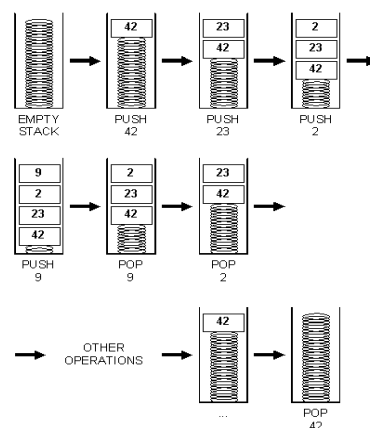


Fig:1.1 STACK

2.2 QUEUES

Queue is a linear list in which elements can be inserted only at one end called rear of the queue and deleted at another end called front of the queue. We can see examples of queue in daily life like queue of people waiting at a counter or a queue of cars etc. In the queue of people and queue of cars, the person or car that enters first in the queue will be out first. It is also known as FIFO due to its behaviour.



Fig:1.2 QUEUE

FUNCTIONS OF QUEUE

- a. **INSERT ()**: This function is used to insert an element in the Queue. When the queue is full & insertion is not possible, then it is called as Queue overflow.
- b. **DELETE ()**: This function is used to delete or remove elements from the queue. When the queue is empty and deletion isn't possible then it is known Queue underflow.
- c. **DISPLAY ()**: This is used to print the elements which are present in the queue list.

2.3 LINKEDLIST

List is a collection of similar types of elements. There are two ways of maintaining a list in a memory. The first way is to store the elements of the list in an array, but arrays have some restrictions and disadvantages. The second way of maintaining a list in memory is

through linked list. Now let us see what linked list is and after that we can overcome the limitation of array. There are three types of linked list:

- Single linked list
- Double linked list
- Circular linked list

Single Linked List: A single linked list is made up of nodes where each node has two parts, the first one is the info part that contains the actual data of the list and the second one is the link part that points to the next node of the list or we can say that it contains the address of next node. From the fig. it is clear that last value is null and head pointer is pointing to first node.

The program works on the basis of single linked list to store the details of the users like email and password. If users want to book the ticket or print the ticket then users have to go through the login phase in which they have to enter their username and password, this all information will be stored in single linked list. When they are successfully logged in then they can book the ticket for various places of the world.

Structures are also important for the program to store the details. To create the node structures are used in the program and the nodes are connected using Single linked list. linked list is basic of storing data in "c" because it is easy to access the list and edit the list. To store the information like email and password structure is used using linked list. Similarly, to store the user information structure is used which is stored in single linked list because it is easy to access the user information using single linked list. Using single linked list, it is easy to create the node and store the information which made the program easier.



Fig:1.3 LINKEDLIST

Double linked list: The linked list that we have studied contained only one link, this is why these lists are called single linked list or one-way list. We could move only in one direction because each node has address of next node only. Suppose we are in the middle of linked list and we want the address of previous node then we have no way of doing this except repeating the traversal from the starting node. To overcome this problem of single linked list we have another data structure called doubly linked list or two-way list, in which each node has two pointers. One of these pointers point to the next node and the other points to the previous node. It contains three lists in the node: one will hold the address of previous node, another will hold the data, and another will hold the address of another node address. Here prev is a pointer that will contain the address of previous node and next will contain the address of next node in the list. So, we can move in both directions at any time. The next pointer of last node and prev pointer of first node are NULL.

The basic logic for all operations is the same as in single linked list, but here we have to do a little extra work because there is one more pointer that has to be updated each time. The main function for the program is written first. Similarly, there are four techniques of insertion.

- Insertion at the beginning of the list
- Insertion in an empty list
- Insertion at the end of the list
- Insertion in between the nodes

As in single linked list here also the node is first logically removed by rearranging the pointers and then it is physically removed by calling the function free (). Let us see the four cases of deletion:

- Deletion of first node.
- Deletion of the only node.
- Deletion in between the nodes.
- Deletion at the end.

In all the cases we will take a pointer variable temp which will point to the node being deleted. It is very easy to delete the node and to traverse the node in the double linked list. Reversing can be also done in doubly linked list. Let us take a doubly linked list and see what changes need to be done for its reversal. so doubly linked list are beneficial only when we frequently need the predecessor of a node.

Circular linked list: In a single linked list for accessing any node of linked list we have to start traversing from the first node. If we are at any node in the middle of the list, then it is not possible to access nodes that precede the given node. This problem can be solved by slightly altering the structure of single linked list. In a single linked list, the link part of last node is null if we utilize this link to point to the first node then we can have some advantage. Each node has a successor and all the nodes form a ring. Now we can access any node of the linked list without going back and starting traversal again from the first node because list is in the form of a circle and we can go from last node to first node. similarly, there are four technique of insertion and deletion:

- Insertion at the beginning of the list
- Insertion in an empty list
- Insertion at the end of the list
- Insertion in between the nodes

Deletion technique:

- Deletion of first node.
- Deletion of the only node.
- Deletion in between the nodes.

- Deletion at the end.

2.3 Trees

A tree structure represents hierarchical relationship among its elements. It is very useful for information retrieval and searching in it is very fast. In a binary tree no node can have more than two children i.e. a node can have 0, 1 or 2 children. Each child is designated as either left child or right child. Similarly, a binary tree is a finite set of nodes that is either empty or consists of a distinguished node called root and remaining nodes called root and remaining nodes are partitioned into two disjoint sets T_1 and T_2 and both of them are binary trees. T_1 is called left subtrees and T_2 is called the right subtree.

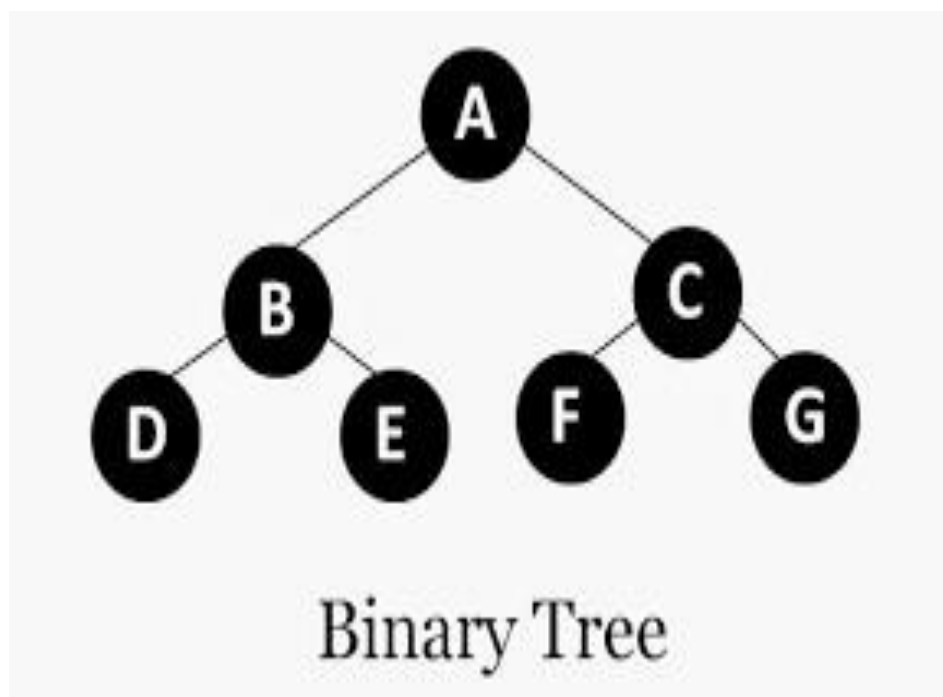


Fig:1.4 TREE

2.4 Graphs

A Graph is a non-linear data structure which consists of nodes and edges. The nodes are referred to as vertices and the edges are lines that connect any two nodes in the graph

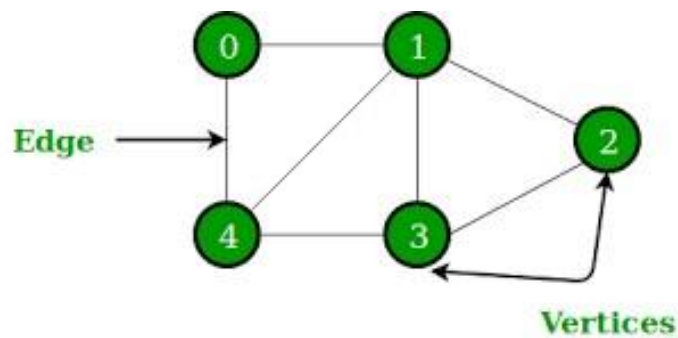


Fig:1.5 GRAPH

CHAPTER 3

DESIGN

3.1 FLOWCHART

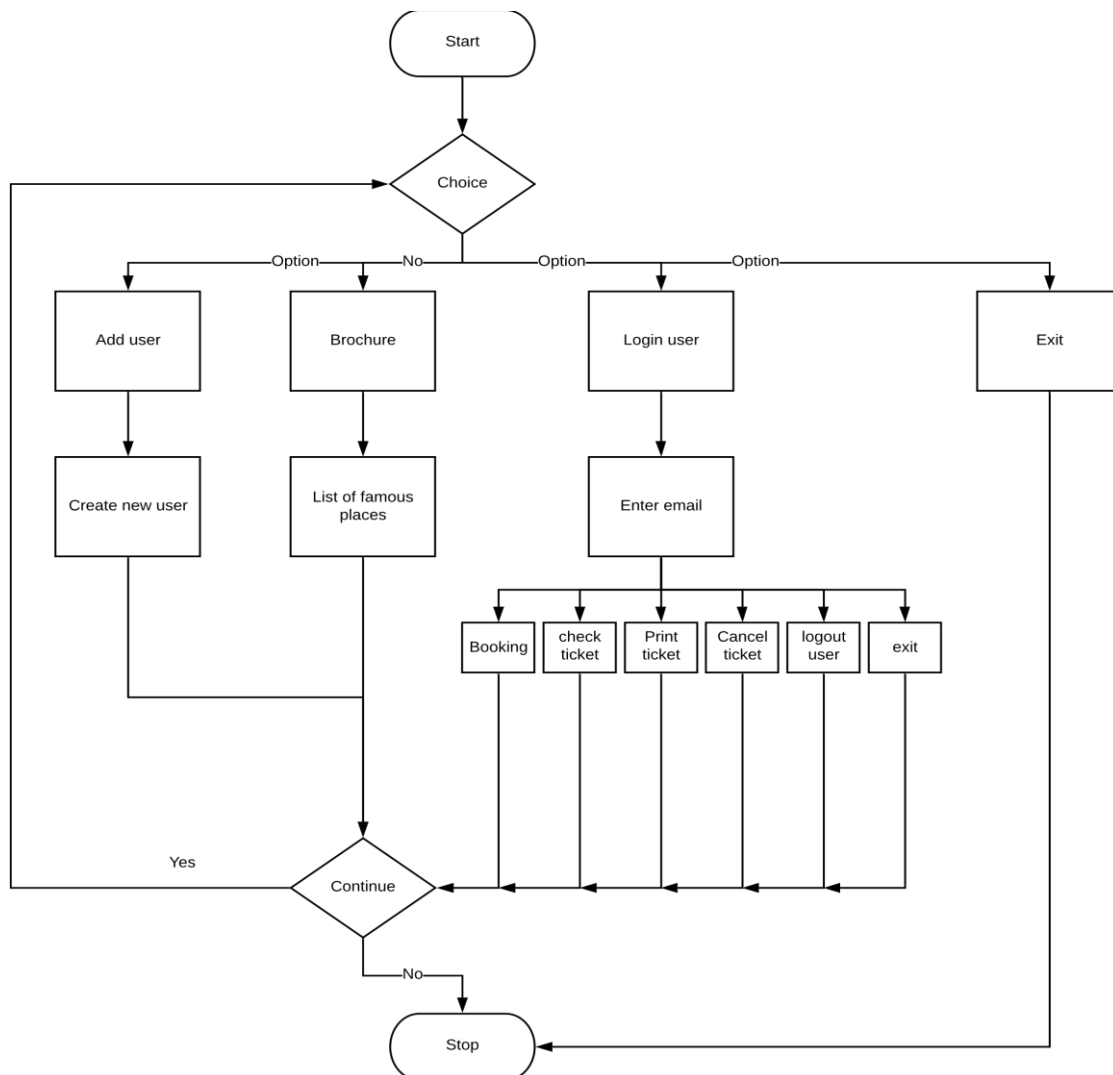


Fig: Flowchart

3.2Algorithm:

This program is about booking the ticket through which user can book the ticket. Visit the different places throughout the world. For this user have to follow the following algorithm:

Step 1: Start

Step 2: Enter your choice.

2.1. Add user

2.2. Login

2.3. Brochure

2.4. Exit

Step 3: After the successful login users will get the following choice:

Case1. Users will get the first option as a book ticket in which user have to provide his detail information as well as the place where he wants to visit.

Case 2. After booking the ticket user can print the ticket. If user hadn't booked the ticket then to print the ticket first user had to book the ticket.

Case 3. After Booking the ticket user can cancel the ticket within 24 hours of booking then only the amount will be refunded to the original account.

Case 4. After booking ticket user can come out of the program.

Step 4: Exit from the program.

Step 5: Stop.

CHAPTER 4

IMPLEMENTATION

4.1MODULE1 FUNCTIONALITY

The main page will appear in which there will be several options. Using those options users can create their account, similarly another user can create their account too. Accordingly, we can see how many users are using the software. With the same Id and password users can login. If user will enter another password, email or without creating their Id then they can't get access in the software. After successful login, users will get several options in which users can book their ticket, check their ticket, either it is booked or not, similarly users can print their ticket as well as they can cancel their ticket. similarly, while booking ticket they can book n numbers of ticket. There is one restriction while booking the ticket i.e. they have to write the place code exactly as it is, because this software is case sensitive for writing the place code.

4.2MODULE2 FUNCTIONALITY

```
struct slist
{
    char email[30];
    char password[30];

    struct slist *ptr;
};

struct node1{
    char email[40];
    char place[40];
    float price;
```

```
int numtick;
struct node1 *link1;
};
```

I have created two structure one for the login page where I am storing the user e-mail id and password and second structure for the users data where I am storing name of the users, the place where he want to visit and the number of tickets he want to book as well as the total fare throughout his journey.

```
void login () {
    start=head;
    printf("Enter the email \n");
    scanf("%s",e);
    printf("Enter password\n");
    scanf("%s",p);
    while(start!=NULL)
    {
        if(strcmp(start->email,e)==0){
            if(strcmp(start->password,p)==0){

                printf("*****\n");
                printf("login sucessfull\n");flag=1;
                printf("*****\n");
                break;
            }
        }
        start=start->ptr;
    }
    if(flag==0) {
        printf("*****\n");
        printf("login failed\n");
        printf("*****\n");
    }
```

After creating his username and password user have to type the same id and password where string compare is being done through the pre-defined function known as string function called string compare. The email Id and Password is being compared in the login function where the flag value is being changed from 0 to 1 which means that user have successfully logged in the software. If the flag value is 0 which means that login is failed due to string comparison function where username and password is not matching. After successful login user will be redirected to second function where I have used three function named as book ticket, print ticket and cancel ticket

4.3 MODULE 3 FUNCTIONALITY

```
void addUser1()
{
if(head==NULL)
{
head=(node*)malloc(sizeof(node));
printf("Create your Email.Id\n");
scanf("%s",head->email);
printf("Create your password\n");
scanf("%s",head->password);
head->ptr=NULL;
}
else
{
printf("head is already created\n");
}
}
```

In this code user can create their user id and password which is being stored in the single linked list with the help of structure created known as slist. I am storing all the information known as head for which memory is assigned using malloc function. If the head is null it means user hadn't entered any data but if head is not equals to null it shows that the account is already created.

```
void printticket(struct node1 *q){

    struct node1 *start1=q;

    if(start1==NULL){

        printf("Not any ticket Booked\n");

        return;

    }

    printf("Name\tplace\tprice\ttno. of ticket\n");

    while(start1!=NULL){

        printf("%s\t%s\t%.2f\t%d\n",start1->email,start1->place,start1->price,start1->numtick);

        start1=start1->link1;

    }
```

After booking the ticket user can print his information like email, place, no. of ticket and total price. If the node start1 is null means user hadn't booked any ticket so there will be message printed to the user that not any ticket booked if the node start1 isn't null it means user has entered all his information like his email place and total no. of tickets.

4.4 MODULE 4 FUNCTIONALITY

```
void bookticket(struct node1 **q){

    char e[40];
```

```
char pl[40];

float p;

int n,choice;

float
pricelist[]={40000.0,60000.0,25000.0,38000.0,120000.0,10000.0,30000.0,32000.0,45000
.0,250000.0};

struct node1 *nn=(struct node1*)malloc(sizeof(struct node1));

printf("Enter your full Name \n");

scanf("%s",e);

printf("\nWould You Like to Confirm Booking?\n[1] - Yes\n[2] - No\n");

scanf("%d",&choice);

if(choice==1)
{
up:
brochure();

printf("Enter the place\n");

scanf("%s",pl);

if(strcmp(pl,"AUS")==0)

p=pricelist[0];

else if(strcmp(pl,"US")==0)

p=pricelist[1];

else if(strcmp(pl,"NP")==0)
```

```
p=pricelist[2];

else if(strcmp(pl,"IND")==0)

p=pricelist[3];

else if(strcmp(pl,"BHU")==0)

p=pricelist[4];

else if(strcmp(pl,"SRI")==0)

p=pricelist[5];

else if(strcmp(pl,"AFG")==0)

p=pricelist[6];

else if(strcmp(pl,"SA")==0)

p=pricelist[7];

else if(strcmp(pl,"VC")==0)

p=pricelist[8];

else if(strcmp(pl,"FR")==0)

p=pricelist[9];


else

{

printf("enter the correct name!!!\n");

goto up;

}

}

if(choice==2)

{
```



```
printf("booking cancelled");  
  
return;
```

This code is written for the purpose of booking of ticket. In this code node is created called struct node 1 in which I will store the address of node1 and I will create one array of float type in which I will store the price of different places. After that username will be taken and stored then I will ask choice 1 and 2. Choice 1 is for that user want to books the ticket, if user will choose 2 it means that user don't want to book the ticket and the message will be displayed as user don't want to book the ticket and also the program will be terminated. If user will choose option 1 then it will go inside the function where I have used string compare function where user will give the place code and it will be compared accordingly. Suppose user will give IND as a place code the it will compare with the place which is stored in pl and it will go to price list which is of array type then it will take the price from the list and total price is calculated by the simple mathematical formula after that a message will be printed as ticket booked , if user will give incorrect code then message will be printed as please write the correct code place. Go to function is used for the reference of user. For e.g. if user will write incorrect code then it will go up and call brochure function in which the all place list is described for the reference of user to write the place code correctly and all the information stored in brochure will be displayed.

CHAPTER 5

RESULTS

The sample outputs of the software are shown below:

Output screen 1: This is the first interface of the project. This is how users will get the output screen when codes are executed. Where the users can create their id and password as well as, owners can see the total numbers of using their software. Using option 4 users can login. Using options 5 they see the places where they want to travel. Option 2 is available is for another user who can also create their id and password. When user will press option 6, they will exit from the program.

Case1: create ID and PASSWORD

```
==== || WORLD TOUR || ====
1.Add First User
2.Add Users
3.All users
4.Log In
5.Brochure
6.Exit
enter your choice
-
```

Fig: 5.1 OUTPUT SCREEN 1

Output screen 2: This is the second interface of the project. This is how users will get the output screen when codes are executed. After successful login user can book the ticket, print the ticket and cancel the ticket. Using the option given below user can come back first interface of the project.

Case 2: LOGIN

```
1.Add First User
2.Add Users
3.All users
4.LogIn
5.Brochure
6.Exit
enter your choice
4
Enter the email
yadav12@gmail.com
Enter password
yadav123
*****
login sucessfull
*****

=====
WORLD TOUR
=====

1.Booking Package
2.print Ticket
3.cancel Ticket
4.Exit
```

Fig:5.2 OUTPUT SCREEN 2

This is the second interface of the project where there will be four information for creating the id and second option will be for another user and there will another option for login similarly there another option for the exit of the program. Some more function is there like brochure where detail information is stored of different places and last option is for the exit of the program. After successful login user can book the ticket print the ticket cancel the ticket and if user will choose option 4 then user will be back to the main interface of the software.

Case:3 BOOK TICKET

```
[2] - No
1
      PRICE LIST
=====
1. AUS- AUSTRALIA Tours - Rs 40000
2. US - UNITED STATE Tours - Rs 60000
3. NP - NEPAL Tours - Rs 25000
4. IND - INDIA TOUR - Rs 38000
5. BHU - BHUTAN - Rs 120000
6. SRI - SRILANKA VACATION TOUR - Rs 10000
7. AFG - AFGHANISTAN EDUCATIONAL TOUR - Rs 30000
8. SA - SOUTH AFRICA TOUR - Rs 32000
9. UC - SMALLEST CITY TOUR - Rs 45000
10. FR - FRANCE TOUR - Rs 250000
Enter the place
AUS
Enter the no.of ticket
3
Ticked Booked

1.Booking Package
2.print Ticket
3.cancel Ticket
4.Exit
```

Fig: 5.3 OUTPUT SCREEN 3

Case:4 PRINT TICKET AND CANCEL TICKET

```
5. BHU - BHUTAN - Rs 120000
6. SRI - SRILANKA VACATION TOUR - Rs 10000
7. AFG - AFGHANISTAN EDUCATIONAL TOUR - Rs 30000
8. SA - SOUTH AFRICA TOUR - Rs 32000
9. UC - SMALLEST CITY TOUR - Rs 45000
10. FR - FRANCE TOUR - Rs 250000
Enter the place
AUS
Enter the no.of ticket
3
Ticked Booked

1.Booking Package
2.print Ticket
3.cancel Ticket
4.Exit

2
Name    place    price    no. of ticket
aniket  AUS      120000.00  3

1.Booking Package
2.print Ticket
3.cancel Ticket
4.Exit
```

Fig: 5.4 OUTPUT SCREEN 4

CHAPTER 6

CONCLUSION

In previous day people had to visit a lot of places to find a travel agency or agent. The travel agent also sometimes does fraud with the people. The tourist has to wonder to travel the various places around the world. The tourists didn't know about which place they have to visit and what would be the total fare throughout their tour. People had to face a lot of problems in order to book tickets. If they have to visit some places for their entertainment then also, they have to search them. This was not an easy way for people because they had gone to enjoy on their vacation and not to worry about everything and then find out where to go. To solve all these problems user can use this software and can visit around anywhere in the world. This software will also have information about few places in the world where a user can relax and refresh their mood. These few services will be available in this software which will make the work of the user more convenient and easier.

This software makes the user-friendly environment for the user to get what they need. The user can simply choose what they want and then after they are done with the process they just need to provide their details. This software deals with mainly four option book ticket print ticket cancel ticket and exit from the program. User can use this software according to their accordance.

The software is generally designed for helping people by letting them book every package without wandering for finding the travel agencies. The main purpose of this project is to make the travelling user friendly, reliable and simple.

CHAPTER 7

REFERENCE

- DATA STRUCTURE THROUGH C IN DEPTH BY S.K. SRIVASTAVA AND DEEPALI SRIVASTAVA
- C IN DEPTH BY S.K. SHRIVASTAVA AND DEEPALI SRIVATAVA