A PROJECT REPORT ON TRAVEL MANAGEMENT FOR SEMESTER ENDING EVALUATION

at

[CRIE, MODEL INSTITUTE OF ENGINEERING AND TECHNOLOGY]

**BACHELOR OF TECHNOLOGY**

(**Computer Engineering**)

**December 2021 - March 2022**



**SUBMITTED BY:**

GROUP: B-13

MEMBERS: 2021A1R110, Manav Gupta  
 : 2021A1R111, Arjumand

: 2021A1R112, Kumud Kesar  
 : 2021A1R113, Vaibhav Jamwal

BRANCH: CSE

SEMESTER: 1st SEMESTER

# ACKNOWLEDGEMENT

*Through this section of our report, we want to present our gratitude towards our dean Prof. Ankur Gupta and the institute MIET itself, that we were given the opportunity to make our very first group project on the topic TRAVEL MANAGEMENT.*

*We would also like to thank our professors Mrs. Bhagyalakshmi Magotra and Ms. Pragati Jamwal for being a guiding force and being so much supportive of us. It was a great experience to work a project as a team.*

*We would also like to thank our HOD, Mr. Anand Kumar and the CSE Department who encouraged and inspired us for learning new skills.*

INDEX

* ACKNOWLEGEMENT
* INTRODUCTION AND PURPOSE
* DRAWBACKS AND CONCLUSION
* SOURCE CODE
* WORKING OF PROGRAM
* OUTPUTS
* REFERENCES

­TARVEL MANGEMENT

* INTRODUCTION: This is program that has been designed to provide a complete automated solution to Travel Agencies. This program includes all basic operations that take place in Travel agencies, which helps to develop Traveling activities. It stores / gives information of customer booking customer account and customer details.
* OBJECTIVE AND SCOPE: The main objective of developing this kind of program to apply the information technology in the traveling agency industry and to provide staff and client a well-maintained information structure. Every industry wants to provide the best service to the client and this one of the main objectives to develop the program.
* DRAWBACKS OF SYSTEM:
* All the business aspects are not covered in the projects
* Does not provide online processing facility as Not a network-based software
* Does not provide printed bills or reports
* Limited for small organizations
* Not a network-based software
* One can encounter a glitch sometimes
* CONCLUSION:
* Travel management system can be applied in all types of travel agencies. The use of this application will result in better customer service and faster processing. The travel management system after some modification can be put on the internet and this way on line booking for the transportation can take place. Moreover, the tourist sitting at home any know about the facilities and services that are being offered by the agency. The program is very useful and easy to use and has been designed keeping in view the complete functioning of the agency. The reports on different requirements can easily be generated in seconds

**Travel Structure**

First, we create a structure to store travel data

struct Travel

{

int password;

int id;

char name[100];

char location[100];

char destination[100];

char date[20];

struct Travel \*next;

}\* head;

**head** is a global pointer of type Student which will point to the first node in the linked list. The advantage of declaring **head** global is we can directly use it in any function without passing it as a parameter.

**Insert Function**

After that, we create a function insert, that will add a new node to the linked list. The insert function accepts travel details as an argument.   
It creates a new node with the travel details passed to the function and then inserts the new node at the beginning of the linked list.

void insert(int password, int id, char\* name, char\* location, char\* destination, char\* date)

{

struct Travel \* travel = (struct Travel \*) malloc(sizeof(struct Travel));

travel->id = id;

travel->password = password;

strcpy(travel->name, name);

strcpy(travel->location, location);

strcpy(travel->destination, destination);

strcpy(travel->date, date);

travel->next = NULL;

if(head==NULL){

// if head is NULL

// set travel as the new head

head = travel;

}

else{

// if list is not empty

// insert travel in beginning of head

travel->next = head;

head = travel;

}

}

# ****Login Function****

# The login function searches the record based on the id number of the traveler. The search function accepts 2 parameters that are the id number and the entered password of the travel account we want to search. The function traverses all the nodes of the linked list to find the required record.

# void login(int id, int password)

# {

# struct Travel \* temp = head;

# while(temp!=NULL){

# if((temp->id==id)&&(temp->password==password))

# {

# printf("ID: %d\n", temp->id);

# printf("Password: %d\n", temp->password);

# printf("Name: %s\n", temp->name);

# printf("Location: %s\n", temp->location);

# printf("Destination: %s\n", temp->destination);

# printf("Date: %s\n", temp->date);

# return;

# }

# temp = temp->next;

# }

# printf("Account with entered password %d associated to entered id number %d is not found !!!\n", password , id);

# }

# Update Function

The update function first searches for the node with the required id number. If the node is found, the program asks the user to first enter password to confirm and then enter new updated values.

void update(int id, int password)

{

struct Travel \* temp = head;

while(temp!=NULL){

if((temp->id==id)&&(temp->password==password))

{

printf("Record with password %d associated with entered id number %d Found !!!\n", password, id);

printf("Enter new location: ");

scanf("%s", temp->location);

printf("Enter new destination: ");

scanf("%s", temp->destination);

printf("Enter new date: ");

scanf("%s", temp->date);

printf("Updation Successful!!!\n");

return;

}

temp = temp->next;

}

printf("Account with entered password %d associated with entered id number %d is not found !!!\n", password, id)

}

**Delete Function**

Delete works similar to login. We search for the record by its id number. If the record is found, we delete it from the linked list.

void Delete(int id, int password)

{

struct Travel \* temp1 = head;

struct Travel \* temp2 = head;

while(temp1!=NULL){

if((temp1->id==id)&&(temp1->password==password))

{

printf("Record with password %d associated with id number %d Found !!!\n", password, id);

if(temp1==temp2){

// this condition will run if

// the record that we need to delete is the first node

// of the linked list

head = head->next;

free(temp1);

}

else{

// temp1 is the node we need to delete

// temp2 is the node previous to temp1

temp2->next = temp1->next;

free(temp1);

}

printf("Your Trip has been cancelled sucessfully!!!\n");

printf("Account Successfully Deleted !!!\n");

return;

}

temp2 = temp1;

temp1 = temp1->next;

}

printf("Account with entered password %d associated with id number %d is not found !!!\n", password, id);

}

**Main Function**

int main()

{

head = NULL;

int choice;

char name[100];

char location[100];

char destination[100];

char date[20];

int password;

int id;

printf("1 to create travel account\n2 to login to your account\n3 to cancel the trip\n4 to modify your scheduled trip\n");

do

{

printf("\nEnter Choice: ");

scanf("%d", &choice);

switch (choice)

{

case 1:

printf("Enter last 4 digits of your mobile number as your travel ID: ");

scanf("%d", &id);

printf("Create 4 digit password: ");

scanf("%d", &password);

printf("Enter name: ");

scanf("%s", name);

printf("Enter location: ");

scanf("%s ", location);

printf("Enter destination: ");

scanf("%s", destination);

printf("Enter date: ");

scanf("%s", date);

insert(id, password, name, location, destination, date);

break;

case 2:

printf("Enter id number to login: ");

scanf("%d", &id);

printf("Enter password for verification: ");

scanf("%d", &password);

login(id, password);

break;

case 3:

printf("Enter id number to login: ");

scanf("%d", &id);

printf("Enter password for verification: ");

scanf("%d", &password);

Delete(id, password);

break;

case 4:

printf("Enter id number to update your travel plan: ");

scanf("%d", &id);

printf("Enter name for verification: ");

scanf("%d", &password);

update(id, password);

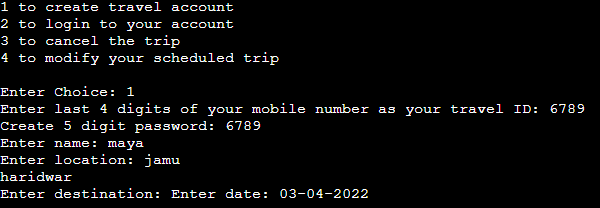
break;

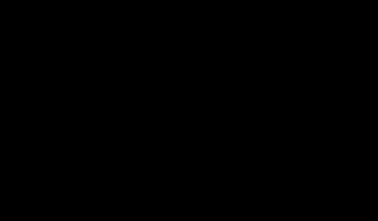
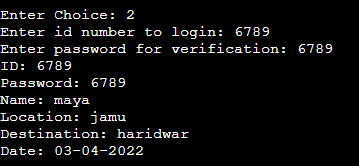
}

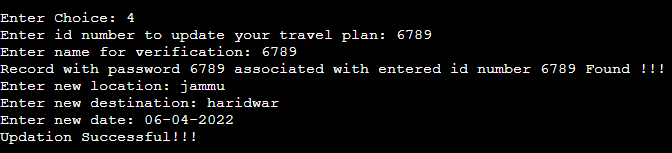
} while (choice != 0);

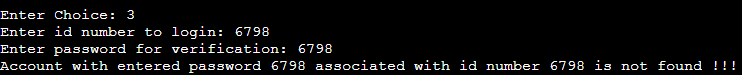
}

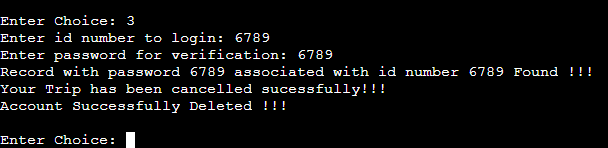
Outputs











References

* Youtube
* Wikipedia
* Support material from C-classes
* Google classroom -> C programming lab