

## Mini Project Report On

**DATA STRUCTURES AND APPLICATIONS (18CS32)**

**DIGITAL TICKETING SYSTEM**

**Submitted By**

**USN NAME**

**1BI20CS112 MUSTHAFFA SHAIK**

## For the academic year 2021-22



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING BANGALORE INSTITUTE OF TECHNOLOGY**

**K.R. Road, V.V. Puram, Bengaluru-560 004**

## BANGALORE INSTITUTE OF TECHNOLOGY

**K.R. Road, V.V. Puram, Be ngaluru-560 004**



## Department of Computer Science & Engineering

***Certificate***

This is to certify that the implementation of **Data Structures and Applications (18CS32) Mini Project** entitled **“DIGITAL TICKETING SYSTEM”** has been successfully completed by SUDHANSHU ARYA **(1BI20CS171),** of III semester B.E. for the partial fulfillment of the requirements for the Bachelor's degree in **Computer Science & Engineering** of the **Visvesvaraya Technological University** during the academic year **2021-2022**.

**In charge:**

**Prof. ARCHANA Dr. J. Girija**

Assistant Professor Professor and Head Department of CS&E Department of CS&E

Bangalore Institute of Technology Bangalore Institute of Technology

**ACKNOWLEDGEMENT**

I wish to express my sincere gratitude to the people who have helped and supported me throughout the project.

I express my sincere gratitude to the HOD of CS&E, Dr.J.Girija for providing us with an environment to complete our project successfully.

I extend my deepest thanks to our DSA professor, Prof. ARCHANA A. for her continuous and unwavering support during the entire course of project making the necessary corrections.

I would like to thank my project partners, family and friends for their constant support throughout the project.

**CONTENTS Page No.**

**CHAPTER 1: INTRODUCTION 5**

**1.1 Overview**

**1.2 Problem Statement**

**1.3 Objectives**

**1,4 Working**

**CHAPTER 2: SYSTEM REQUIREMENT SPECIFICATIONS 7**

**2.1 Software Requirements**

**2.2 Hardware Requirements**

**CHAPTER 3: DESIGN**

**3.1 Algorithm 8**

**3.2 Flowchart 12**

**CHAPTER 4: IMPLEMENTATION 16**

**4.1 Modules Description**

**4.2 Source code**

**CHAPTER 5: RESULTS/SNAPSHOTS 18**

**CHAPTER 6: APPLICATIONS 23**

**CHAPTER 7: CONCLUSION 25**

**CHAPTER 1: INTRODUCTION**

* 1. **Overview**

The railway reservation system facilitates the passengers to enquiry about the trains available on the basis of source and destination, booking and cancellation of

tickets, enquiry about the status of the booked ticket, etc.

The aim is to design and develop a data base maintaining records of different trains, train status and passengers. This project contains introduction to

the railways reservation system.

It is the computerized system of reserving the seats of train seats in advance. It is mainly used for a long route. Online reservation has made the process for the reservation of seats very much easier than ever before.

**1.2 Problem Statement**

**To develop a software that can keep track of all its passengers, thus schedule their journey accordingly and also store and retrieve information about the various transactions related to Rail Travel.**

* 1. **Objectives**

• To provide the customers with an easy and simple way of booking train tickets at a comfort of their homes.  
• To reduce the time, energy, and resources that was being consumed while performing a counter -based reservation.

• To generate a ticket that is valid throughout the journey.  
• To provide a convenient solution of billing pattern.

**WORKING:**

As we run the program, initially it displays the introduction of all the teammates and asks the user to enter the password.  
If you type the correct password, then it moves further to the next part else it will ask the user to try again. The next part consists of three options -:  
1st -new user 2nd -login 3rd -exit.  
It will ask the user to enter 1 ,2 or 3.

If you choose Login, it will ask for username and if no existing user is found, it displays username not found and returns to the beginning. So now you have to create a new user. To create a new user, Username, DOB, Phone Number and Password needs to be entered and it will store in its database and make you logged in with that account.

As soon as you login, it will ask for source and destination stations.  
In our program you have to be careful with your spellings, if you enter wrong spelling, then it will ask you to enter a valid station. After this, it will ask for the date on which you wish to travel. After entering the date, it will display a chart of available trains with their details

Then you have to enter the train number as per your convenience, which is present in the chart. After that, it will show you fare details. To continue booking, press 2 but if you wish to select another train press1. If you select 1, it will show the train chart again but, if you press 2, it will ask you for number of passengers and their details.

**CHAPTER 2: SYSTEM REQUIREMENT SPECIFICATIONS**

**2.1 Software Requirements**

1. C-compiler (cc, gcc, egcs, ...)
2. GNU make is recommended.
3. zlib compression library (already installed on modern systems)
4. lexical analyzer generator (flex)
5. parser generator (yacc, bison)
6. libncurses4.x/5.x (already installed on modern systems)

**2.2 Hardware Requirements**

A) Memory:

i. Minimum-512 MB

ii. Recommended -1GB or more

B) Free Disc Space:

i. Minimum-300 MB

ii. Recommended -1GB or more

C) Processor speed:

i. Minimum-800 GHz

ii. Recommended-1.5GHz or faster

**CHAPTER 3: DESIGN**

**3.1 Algorithm**

ALGORITHM of our code

MAIN function

STEP 1: - Declare the "First" USER\_NODE and assign it a NULL value.

STEP 2: - Call the starting\_display() function, which displays the initial page of the project.

STEP 3: - Assign the return value of user() function to "First".

STEP 4: - Display the options of cities available to the user and prompt him to enter one.

STEP 5: - Compare the option selected by the user and jump to the appropriate function.

STEP 6: - If the user types a city name that is not available in our code prompt him to try again.

STEP 7: - After the function returns from the appropriate source function go back to STEP 3.

STEP 8:- return.

STARTING\_DISPLAY function

STEP 1: - Display in a systematic and in an ordered manner the name of our project.

STEP 2: - Display the names of the team members of our project in certain time intervals which can be done by using a built-in clock() function in the "time.h" header file.

STEP 3: - Ask the user to enter the password and enter a while loop to check whether the password is correct.

If the password is correct break the for loop and returns to the main() function, else continue.

STEP 4: - return.

USER function

STEP 1: - Provide the user three options to choose from: -

1.Sign up 2. Login 3. Exit

STEP 2: - After the user enters the option use a switch case to jump to appropriate function

case 1: create();

case 2:login();

case 3: exit(0); -- if the user chooses option should exit the whole program.

STEP 3: - return.

CREATE function

STEP 1: - Check the value of the first NODE of the linked list.

STEP 2: - If NULL, create space for a new USER\_NODE and assign it to "First".

STEP 3: - Input all the values of the "user" structure.

STEP 4: - Set First->link=NULL.

STEP 5: - return First.

LOGIN function

STEP 1:- If the first NODE of the linked list is not NULL ask the user to enter the username.

STEP 2: - Compare the entered username with all the values of usernames in the "User" singly linked list.

STEP 3: - If found, go to STEP 9, else ask the user to enter a valid username again.

STEP 4: - Now ask the user to enter the password for his/her login account.

STEP 5: - Enter a while loop and check the entered password with the password of the current username.

STEP 6: - If the password is correct, break the while loop, else go to STEP 5.

STEP 7: - return.

The city() function is a basic template for all similar functions.

CITY() function

STEP 1:- Ask user for destination.

STEP 2:- Compare the entered destination with the names of available cities.

STEP 3:- If matched, branch to the appropriate "source\_destination", else ask the user to try again.

STEP 4:- return;

The source\_destination() function is a basic template for all similar functions.

SOURCE\_DESTINATION() function

STEP 1:- Display all the available trains and their information.

STEP 2:- Ask the user for the train number of the train he/she wants to book.

STEP 3:- Jump to the particular train by using a "switch" statement and ask the user to type the number of tickets he/she wants.

STEP 4:- Obtain the user details by using the passdata() and Readdata() functions.

STEP 5:- Calculate the sum of amount of all the tickets by using the calculate() functions and assign the

return value to amount global variable.

STEP 6:- Print the ticket by using the ticket() function.

STEP 7:- return.

PASSDATA() function

STEP 1:- Run a "for" loop till the number of tickets the user wants.

STEP 2:- Call the ReadData() function.

STEP 3:- After the data is read, add the user to the details linked list by using the "front insertion" method of linked list.

STEP 4:- Return the linked list.

READDATA() function

STEP 1:- Read all the info of the user into the globally declared variables.

STEP 2:- return.

INSFRONT() function

STEP 1:- Create space for a new node by using the MALLOC macro

and assign all its variables the values of the globally declared variables read by ReadData() function.

STEP 2:- Use the front insertion method of the singly linked list to insert the new node at the front of linked list.

STEP 3:- return the linked list.

CALC() function

STEP 1:- Calculate the amount of 'n' tickets by using their type and using the switch statement to do the same.

STEP 2:- return the amount of 'n' tickets calculated.

TICKET() function

STEP 1:- Display the basic information of the train, date of travel, amount.

STEP 2:- Call the coach() function.

STEP 3:- Call the seat() function.

STEP 4:- Display the information of the users.

STEP 5:- return.

SEAT() function

STEP 1:- Use the rand() function to generate random seat numbers and use srand(time(t)) to seed the rand() function

so that it generates different random values every time.

STEP 2:- According to the random number obtained, calculate the berth type by dividing the number by 8

(In a train coach the type of berths repeat after an octave).

STEP 3:- Print the seat number.

STEP 4:- return.

PAYMENT() function

STEP 1:- Generate a random 4 digit number designated as 'OTP' by usin a ran() function and using

srand(time(t)) to seed the rand() function so that it generates different random values every time.

STEP 2:- Display the OTP on the right hand side of the screen.

STEP 3:- Ask user to enter the OTP.

STEP 4:- Enter a while loop, if the OTP entered is wrong go to step 2, else break.

STEP 5:- return.

COACH() function

STEP 1:- Use the ran() function with range to give a random number in the range of 1 to 12.

STEP 2:- Print that random number as the coach number.

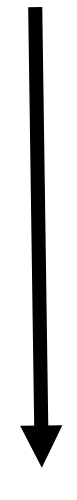
STEP 3:- return.

END

**3.2 Flowchart**

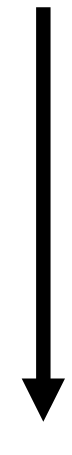
**1)**

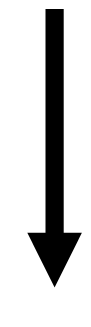
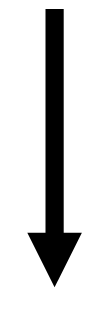
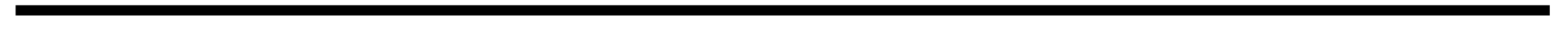
user



Welcome to

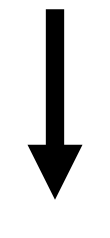
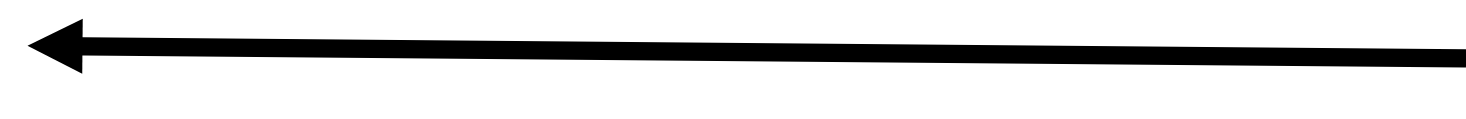
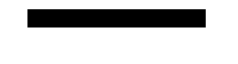
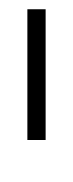
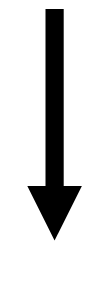
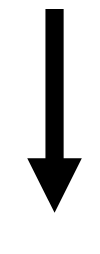
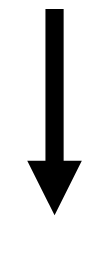
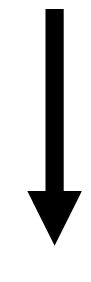
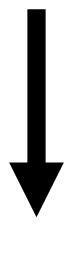
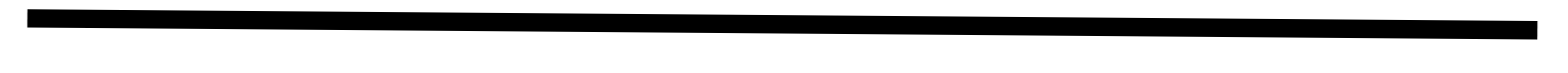
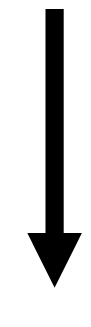
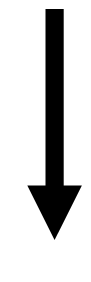
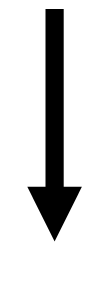
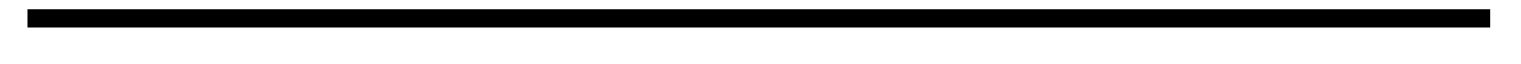
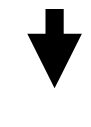
IRCTC





New user

Existing user



Sign in / log in:

Welcome to

IRCTC

New user

Existing user

User name

DOB

Phone number

password

User name

if

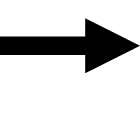
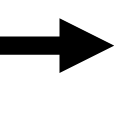
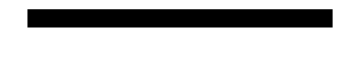
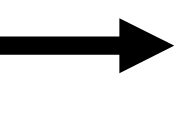
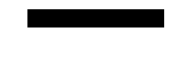
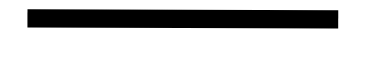
password

if

(wrong)

Login

successfull



Booking system:

source

(Bengaluru, Hyderabad, Mumbai, Delhi, Chennai, Kolkata)

(Bengaluru, Hyderabad, Mumbai, Delhi, Chennai, Kolkata)

destination

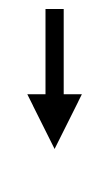
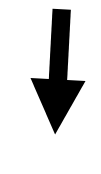
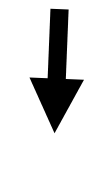
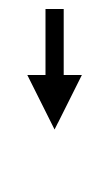
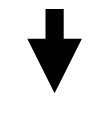
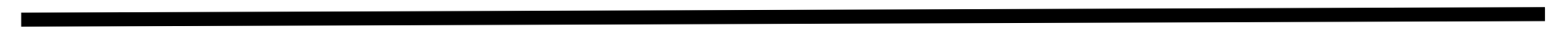
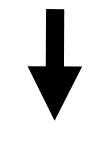
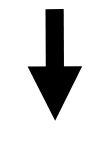
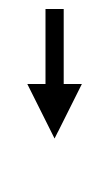
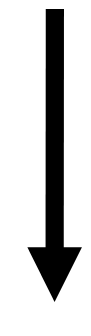
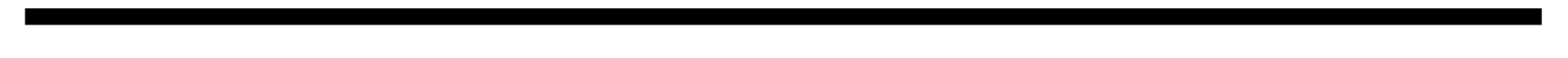
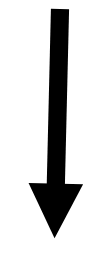
(The date which you want to book ticket)

Date

Train

Shows available trains with the departure, arrival timings and the fare in different classes

Select train



In selected train:

Selected train

passengers

class

Number of passengers

details

sleeper

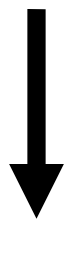
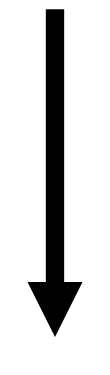
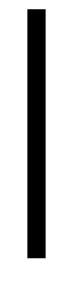
1A

2A

3A

(Name, gender, age, phone number)

Payment and ticket:



Fair of the class \*number of passengers

OTP

User device

(displays)

(To registered mobile number)

Display’s ticket

(After entering the correct OTP)

(With all the details)

**CHAPTER 4: IMPLEMENTATION**

**4.1 Modules Description**

* **STDIO. H is a header file which has the necessary information to include the input/output related functions in our program. Example printf, scanf etc.**
* **STDLIB.H is the header of the general purpose standard library of C programming language which includes functions involving memory allocation, process control, conversions and others. eg. int rand(void) exit.**
* **The STRING. H header defines one variable type, one macro, and various functions for manipulating arrays of characters. eg. strcmp().**
* **This is known as dynamic memory allocation in C programming. To allocate memory dynamically, library functions are malloc() , calloc() , realloc() and free() are used. These functions are defined in the <stdlib. h> header file.**
* **The TIME.H header file contains definitions of functions to get and manipulate date and time information. eg. time\_t**
* **Using the system header file STDBOOL. H allows you to use bool as a Boolean data type. true evaluates to 1 and false evaluates to 0 .**

**4.2 Source code**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<malloc.h>

#include<math.h>

#include<time.h>

#include<stdbool.h>

#define MAX 80

struct details

{

char Name[MAX];

char Gender[MAX];

int Age;

char Phone[MAX];

struct details \*link;

};

typedef struct details \*NODE;

struct user

{

char name[MAX];

char dob[10];

char phno[10];

char pass[MAX];

struct user \*link;

};

typedef struct user\* USER\_NODE;

#define MALLOC(p,s,t)\

p=(t)malloc(s);\

if(p==NULL)\

{\

printf("Insufficient Memory.\n");\

exit;\

}

char name[MAX];

char gen[MAX];

int age;

char ph[MAX];

int SL,Third,Second,First;

char source[MAX],dest[MAX],tname[MAX],tno[MAX],cla[MAX],dur[MAX],date[12];

char berth[5];

int sno[MAX];

int amount,n;

float arr,depart;

char uname[MAX],dob[10],uph[10],upass[MAX],h;

void ban();

void hyd();

void mum();

void del();

void chen();

void kol();

void ban\_hyd(void);

void ban\_mum(void);

void ban\_del(void);

void ban\_chen(void);

void ban\_kol(void);

NODE InsFront(NODE,int);

void ticket(NODE,int);

NODE passdata(int,NODE);

int calc(int,int,int,int,int);

void seat(int);

void coach();

void starting\_display(void);

void payment(void);

USER\_NODE user(USER\_NODE);

USER\_NODE create(USER\_NODE);

bool login(USER\_NODE);

void main()

{

USER\_NODE First=NULL;

char q,g;

starting\_display();

g:First=user(First);

printf("\n\n\n");

getchar();

printf("\n\tThe available citites are\n");

printf("Bengaluru\tHyderabad\tMumbai\t\tDelhi\t\tKolkata\t\tChennai\n");

q:printf("Source:");

gets(source);

if(strcmp(strlwr(source),"bengaluru")==0)

ban();

else if(strcmp(strlwr(source),"hyderabad")==0)

hyd();

else if(strcmp(strlwr(source),"mumbai")==0)

mum();

else if(strcmp(strlwr(source),"delhi")==0)

del();

else if(strcmp(strlwr(source),"chennai")==0)

chen();

else if(strcmp(strlwr(source),"kolkata")==0)

kol();

else

{

printf("Enter Valid Station.\n");

goto q;

}

goto g;

return;

}

void starting\_display(void)

{

printf("\n\n\n\n\n\n\t\t\t\t\t");

int i=0;

while(i++<70)

putchar('-');

printf("\n\t\t\t\t\t|");

i=0;while(i++<68)printf(" ");

putchar('|');

printf("\n\t\t\t\t\t|");

printf("\tWELCOME TO IRCTC RAILWAY RESERVATION SYSTEM SIMULATION");

clock\_t start1=clock();

while(clock()<start1+1000);//this is a clock() function which halts program for 1000ms

i=0;

while(i++<6)printf(" ");printf(" |");

printf("\n\t\t\t\t\t|");

i=0;while(i++<68)printf(" ");

printf("|");

printf("\n\t\t\t\t\t|\t\t----------MADE BY-----------");

i=0;while(i++<25)printf(" ");printf("|");

const char\* array[]={

"MUSTHAFFA","SOHAN JAGTAP","RAJA SAGAR","SATYAM","RUMAN","SUDHANSHU","HARSH","SHASHANK"

};

i=0;

while(i<8)

{

clock\_t start=clock();

while(clock()<start+700);

printf("\n\t\t\t\t\t|");

printf("\t\t\t%s",array[i]);

int a=0;while(a++<(45-strlen(array[i])))printf(" ");putchar('|');

i++;

}

printf("\n\t\t\t\t\t");

i=0;

while(i++<70)

putchar('-');

printf("\n\n\t\t\t\tPlease type the project password:\t");

int k;

while(scanf("%d",&k)==1)

{

if(k==786)

break;

printf("\n\t\t\t\tWrong password!! Type again:\t");

}

system("cls");//clears the display screen

}

USER\_NODE user(USER\_NODE First)

{

char c;

bool find;

c:printf("\nChoose any one of the following:\t");

printf("\n\n1.New user\t2.login\t\t3.exit\n");

int n;scanf("%d",&n);

switch(n)

{

case 1:First=create(First);find=true;break;

case 2:find=login(First);break;

case 3:exit(0);

default:("Please choose the right option\n");goto c;

}

if(find==false)

goto c;

return First;

}

USER\_NODE create(USER\_NODE first)

{

USER\_NODE curr=first;

if(first==NULL)

{

first=(USER\_NODE)malloc(sizeof(struct user));

printf("this is the first user\n");

printf("\nType the username:\t");

scanf("%s",first->name);

printf("\nType the phone number:\t");

scanf("%s",first->phno);

printf("\nType the the date of birth:\t");

scanf("%s",first->dob);

printf("\nType the password:\t");

scanf("%s",first->pass);

first->link=NULL;

printf("\n\n\t\tWElCOME\t%s",first->name);

return first;

}

while((curr->link)!=NULL){

curr=curr->link;}//this searches till the last of the linked list

USER\_NODE temp=NULL;

MALLOC(temp,sizeof(struct user),USER\_NODE);

printf("\nType the username:\t");

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

printf("\nType the phone number:\t");

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

printf("\nType the the date of birth:\t");

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

printf("\nType the password:\t");

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

temp->link=NULL;

curr->link=temp;

printf("\n\n\t\tWelcome!!\t%s \n",temp->name);

getchar();

return first;

}

bool login(USER\_NODE first)

{

if(first==NULL)

{

printf("\n\tThere is no existing user present\n");

return false;//NO user was find;

}

printf("\nType the username:\t");

char username[30];scanf("%s",username);

USER\_NODE curr=first;

while(curr!=NULL&&strcmp(curr->name,username))

curr=curr->link;

if(curr==NULL)

{

printf("\n There is no person with this name in our list\n");

return false;

}

printf("\nType your password");

char password[MAX];scanf("%s",password);

while(strcmp(password,curr->pass))

{

printf("\n\t Try again\n");

scanf("%s",password);

}

return true;

}

void ban()

{

char q;

q:printf("Destination:");

gets(dest);

if(strcmp(strlwr(dest),"hyderabad")==0)

{

printf("Date:");

gets(date);

ban\_hyd();

}

else if(strcmp(strlwr(dest),"mumbai")==0)

{

printf("Date:");

gets(date);

ban\_mum();

}

else if(strcmp(strlwr(dest),"delhi")==0)

{

printf("Date:");

gets(date);

ban\_del();

}

else if(strcmp(strlwr(dest),"chennai")==0)

{

printf("Date:");

gets(date);

ban\_chen();

}

else if(strcmp(strlwr(dest),"kolkata")==0)

{

printf("Date:");

gets(date);

ban\_kol();

}

else

{

printf("Enter Valid Station.\n");

goto q;

}

}

void ban\_hyd(void)

{

int z,y;

char q;

NODE fi=NULL;

q:printf("\nTrain Number\tSource\t\t Destination\t\t Train Name\t Departure\t Arrival\tDuration\n");

printf("17604\t\tYelahanka(YNK)\t Kacheguda(KCG)\t\t YNK KCG EXP\t 16.20\t\t 5.00\t\t12hr40min\n");

printf("12786\t\tBengaluru(SBC)\t Kacheguda(KCG)\t\t Kacheguda EXP\t 18.20\t\t 5.40\t\t11hr20min\n");

printf("22691\t\tBengaluru(SBC))\t Secundrabad Jn(SC)\t RAJDHANI EXP\t 20.00\t\t 7.05\t\t11hr5min\n");

printf("\nEnter train number:");

scanf("%d",&z);//enter train number

switch(z)

{

case 17604:printf("\nFare Chart:\n");

printf("SL:365/-\t3A:990/-\t2A:1410/-\t1A:2360\n");

printf("To select another train, press 1\n");

printf("To continue with booking, press 2\n");

scanf("%d",&y);

if(y==1)

goto q;

printf("\nEnter number of passengers:");

scanf("%d",&n);

strcpy(source,"Yelahanka Jn(YNK)");

strcpy(dest,"Kacheguda(KCG)");

strcpy(tname,"YNK KCG EXP");

strcpy(tno,"17604");

strcpy(dur,"12hr40min");

depart=16.20;

arr=5.00;

fi=passdata(n,fi);

SL=365;

Third=990;

Second=1410;

First=2360;

amount=calc(SL,Third,Second,First,n);

ticket(fi,n);

break;

case 12786:printf("\nFare Chart:\n");

printf("SL:385/-\t3A:1005/-\t2A:1415/-\t1A:2360\n");

printf("To select another train, press 1\n");

printf("To continue with booking, press 2\n");

scanf("%d",&y);

if(y==1)

goto q;

printf("\nEnter number of passengers:");

scanf("%d",&n);

strcpy(source,"Krantivira Sangoli Rayanna(SBC)");

strcpy(dest,"Kacheguda(KCG)");

strcpy(tname,"KACHEGUDA EXP");

strcpy(tno,"12786");

strcpy(dur,"11hr20min");

depart=18.20;

arr=5.40;

fi=passdata(n,fi);

SL=385;

Third=1005;

Second=1415;

First=2360;

amount=calc(SL,Third,Second,First,n);

ticket(fi,n);

break;

case 22691:printf("\nFare Chart:\n");

printf("3A:1790/-\t2A:2100/-\t1A:3190\n");

printf("To select another train, press 1\n");

printf("To continue with booking, press 2\n");

scanf("%d",&y);

if(y==1)

goto q;

printf("\nEnter number of passengers:");

scanf("%d",&n);

strcpy(source,"Krantivira Sangoli Rayanna(SBC)");

strcpy(dest,"Secundrabad Jn(SC)");

strcpy(tname,"RAJDHANI EXP");

strcpy(tno,"22691");

strcpy(dur,"11hr5min");

depart=20.00;

arr=7.05;

fi=passdata(n,fi);

SL=0;//No SL Class Coach

Third=1790;

Second=2100;

First=3190;

amount=calc(SL,Third,Second,First,n);

ticket(fi,n);

break;

default:printf("Enter valid Train Number.\n");

goto q;

break;

}

}

NODE passdata(int m,NODE t)

{

int i;

for(i=1;i<=m;i++)

{

printf("\nEnter Passenger %d details:\n",i);

ReadData();

t=InsFront(t,m);

}

return t;

}

//Reading Passenger Details

void ReadData()

{

printf("Enter Name:\n");

getchar();

gets(name);

printf("Enter Gender:\n");

scanf("%s",gen);

printf("Enter Age:\n");

scanf("%d",&age);

printf("Enter Phone Number:\n");

scanf("%s",ph);

}

NODE InsFront(NODE first,int y)

{

NODE q=NULL;

MALLOC(q,sizeof(struct details),NODE);

strcpy(q->Name,name);

strcpy(q->Gender,gen);

q->Age=age;

strcpy(q->Phone,ph);

q->link=first;

return q;

}

//Train Fare Calculation

int calc(int x1,int x2,int x3,int x4,int y)

{

int d,h;

printf("Select Class:\n");

printf("1.SL\t2.3A\t3.2A\t4.1A\n");

scanf("%d",&d);

switch(d)

{

case 1:strcpy(cla,"SL");

h=x1\*y;

payment();

return h;

case 2:strcpy(cla,"3A");

h=x2\*y;

payment();

return h;

case 3:strcpy(cla,"2A");

h=x3\*y;

payment();

return h;

case 4:strcpy(cla,"1A");

h=x4\*y;

payment();

return h;

default:printf("Select valid Class.\n");

break;

}

}

void seat(int p)

{

time\_t t;

int i,x[MAX];

srand((unsigned) time(&t));

if(p<6)

{

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

sno[i]=(rand()%25)+32;

}

else

{

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

sno[i]=(rand()%72)+1;

}

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

x[i]=sno[i]%8;

printf("\nBerth:\t");

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

{

if(x[i]==1||x[i]==4)

printf("LB\t");//Lower Berth

else if(x[i]==2||x[i]==5)

printf("MB\t");//Middle Berth

else if(x[i]==3||x[i]==6)

printf("UB\t");//Upper Berth

else if(x[i]==7)

printf("SLB\t");//Side lower Berth

else

printf("SUB\t");//Side Upper Berth

}

printf("\n");

printf("Seat No.:");

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

{

printf("%d\t",sno[i]);

}

printf("\n");

return;

}

void payment(void)

{

time\_t t;

srand((unsigned) time(&t));

while(1){

printf("\n\n\t\tPlease type the otp received on your mobile number:(wait 5 seconds)\t");

int k=(rand()%8999)+1000;//rand() function with ranges to generate random otp

clock\_t start=clock();

while(clock()<start+5000);//will receive otp in five seconds

printf("\t\t\t\t\t\t|OTP--%d|\n",k);

int y;

printf("Enter OTP:");

scanf("%d",&y);

if(y==k){

printf("\nYour payment is being processed\n");

start=clock();

while(clock()<start+2000);

printf("\nYour payment was successful.\n");

start=clock();

while(clock()<start+2000);

break;

}

printf("\nType again");

continue;

}

return;

}

void coach()

{

int x;

time\_t t;

srand((unsigned) time(&t));

x=1+(rand()%12);

printf("\nCoach:S%d\n",x);

}

void ticket(NODE q,int p)

{

int i;

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TICKET\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\nSource\t\tDestination\t\tTrain Name\tTrain No.\tClass\n");

printf("%s\t%s\t%s\t%s\t\t%s\n",source,dest,tname,tno,cla);

printf("\nDate:%s\n",date);

printf("\nDeparture\tArrival\t\tDuration\n");

printf("%.2f\t\t%.2f\t\t%s\n",depart,arr,dur);

printf("\nFare:%d\n",amount);

clock\_t start=clock();

while(clock()<start+500);//it takes a little time to print the further ticket

coach();

seat(p);

printf("\n");

printf("\nPassenger Details:\n");

start=clock();

while(clock()<start+300);

while(q)

{

printf("\nName:%s\nGender:%s\nAge:%d\nPhone Number:%s\n",q->Name,q->Gender,q->Age,q->Phone);

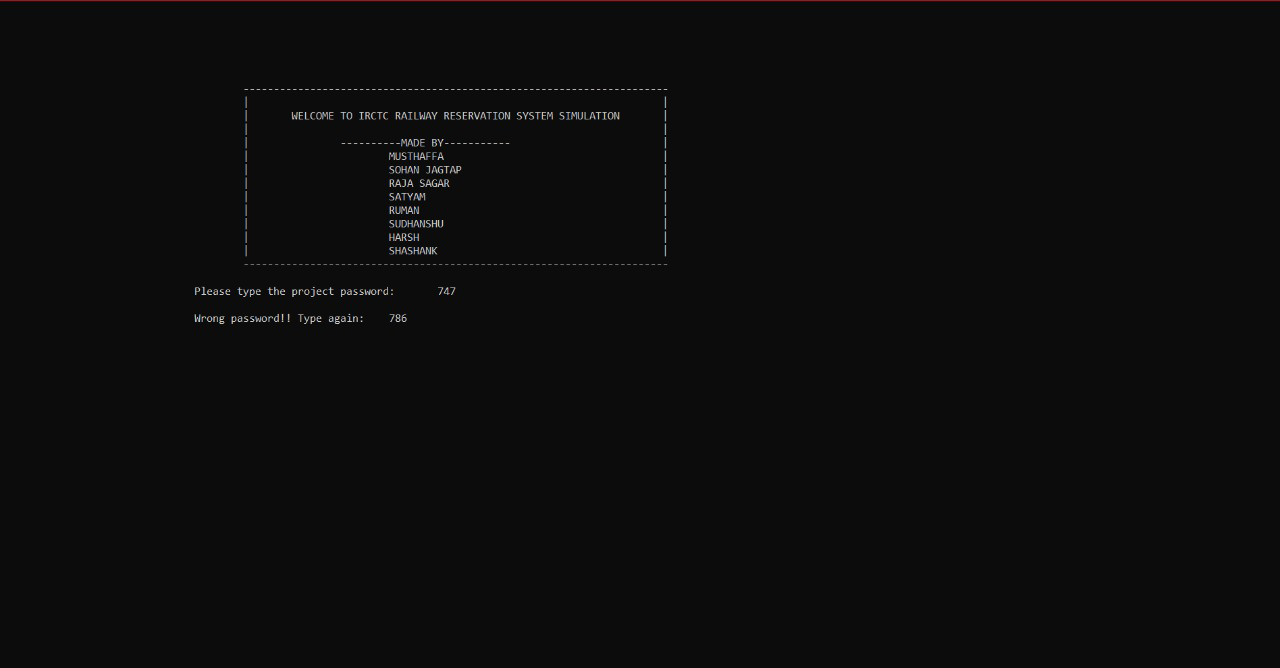
q=q->link;

}

printf("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*HAPPY JOURNEY\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n\n");

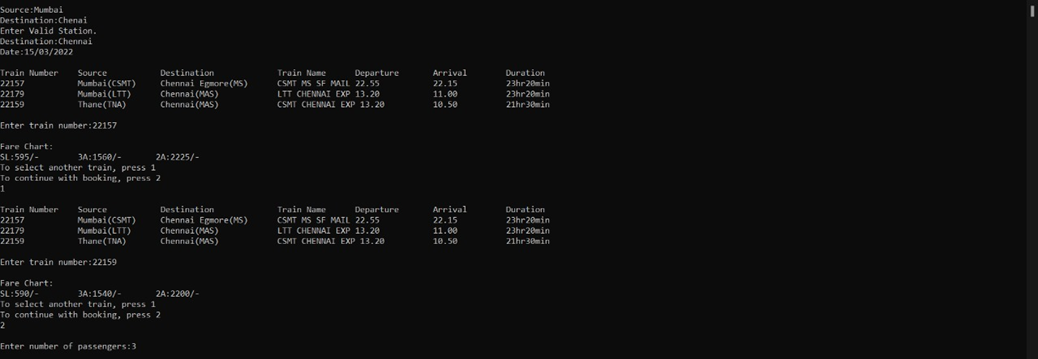
}

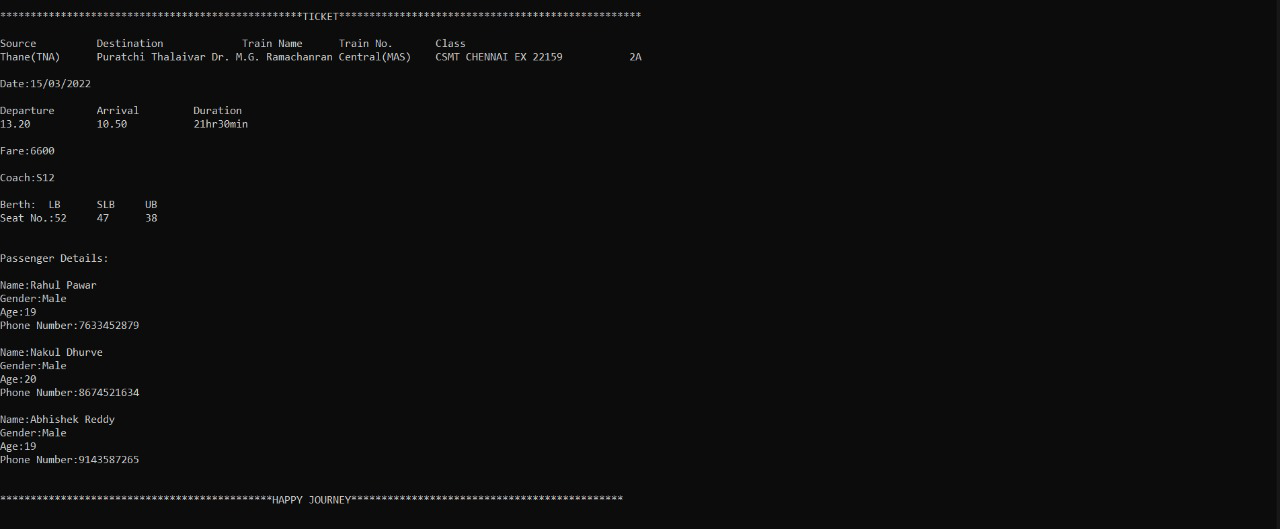
**CHAPTER 5: RESULTS/SNAPSHOTS**











**CHAPTER 6: APPLICATIONS**

* **The ticket is Portable and safe to carry which reduces the chances of theft.**
* **Software can be slightly modified to create airline, bus and even movie ticketing platforms.**
* **A well designed and operated software can reduce processing time by great margins.**
* **Easy to keep record of travel and passengers for business and analytical purposes.**
* **No tension of Long Queues and it provides a Hassle Free user experience.**
* **Ticket generated digitally is Environment Friendly as there is no wastage of Paper.**

**CHAPTER 7: CONCLUSION**

This was our project of Data Structures and Applications about “DIGITAL TICKETING SYSTEM”.

Development of this project did take a lot of efforts from but we did learn and pick up a variety of new skills on the way.

Though every task is never said to be perfect in this development field even more

improvement may be possible in this system.

We learned many things and gained a lot of knowledge and hopefully we will be to build more upon it in the future.