VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

April 2020

RAILWAY MANAGEMENT

A Project Report

Under the Guidance of,

Prof. A. Srivani

19MID0073 [SANJAY KUMAR.K]

19MID0070 [SATHYA PRIYA.R]

DECLARATION BY THE CANDIDATE

We hereby declare that the project report entitled **"Title of the Project"** submitted by us to VIT University, Vellore in partial fulfilment of the requirement for the award of the degree of **Integrated M.Tech CSE with Specialization in Data Science** is a record of J- component of project work carried out by me/us under the guidance of **Prof. A.Srivani**. We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

PLACE : VIT UNIVERSITY , VELLORE

DATE : 07.06.2020

TABLE OF CONTENTS

1. Introduction
1.1 Abstract

2. Overview and Planning
2.1 Proposed Work
2.2 Hardware Requirements
2.3 Software Requirements

3. Literature Survey and Review
3.1 Literature Summary

4. Methodology
4.1 Method Used
4.2 Applications

5. System Implementation
5.1 Code
5.2 Results and discussion

6. Conclusion
6.1 Conclusion

7. References

1.INTRODUCTION

1.1 Abstract

Our project introduce railway reservation system with an objective to make the reservation system more efficient ,easier and faster. this project is dedicated to model existing railways reservation system that aims at development to reservation that facilitate the railways customer to manage their reservations and the railways administrator to modify backup database in user friendly manner The main purpose of this project is to describe the railways reservation system which provides the rail timing details, enquiry, reservation.

2. OVERVIEW AND PLANNING

2.1 Proposed Work

This project aim development of a railway reservation system that facilitates the railway customer to manage their reservation and railway administrator to modify their backend database in user friendly manner .

This program includes the following functions:

1) Create New Database

2) Add New Records

3) Ticket Reservation

4) Display Records

5) Check Availability.

2.2 Hardware Requirements

- Intel Core i5

2.3 Software Requirements

- Code Blocks

3. METHODOLOGY

3.1 Methodology

It enables us to maintain the railway train details like timing ,number of seat available ,reservation tickets. when the program is executed the following process happens..

The program contains one class with some data members like

1. Train No
2. Seat No
3. Arrival Time
4. Depature Time
5. From
6. To

and along with that some member functions for several uses like to

Load The Trains Details :

In install function it ask you to load the train details like train name ,when its starts and when it departure time, and along with that we have load the start and terminus point of the train.

Reservation :

In reserve function the reservation process takes place .it asks you to enter the train number and passenger name and seat number to reserve the ticket .

Availability :

In available function the availability of the seat is calculated it. first it ask you to enter the train number then show the availability.

Empty:

In empty function the data structure of the train compartment was displayed if you reserved any seats it also displayed with the reserved person's name.

Allotments :

In allotment function the train seat structure was generated their and along with that if you allotted any seat it calculates and reserve that seat and make it as no one could reserve it.

3.2 Applications

* In Railway Station.

* In Railway Managements.

4.SYSTEM IMPLEMENTATION

4.1 C++ CODE :

```
#include<iostream>
#include<string.h>
using namespace std;
static int T=0;
class TRAIN
{
   char T_No[5], Arrival[5], Depart[5], From[10], To[10], seat[16][2][10];
   public:
      void Upload();
      void Booking();
      void Train_Availability();
      void Empty();
      void Allotment();
      void position(int i);
}
train[10];

void TRAIN::Upload()
{
   cout<<"\nEnter Train Number: ";
   cin>>train[T].T_No;
   cout<<"Arrival time      : ";
   cin>>train[T].Arrival;
   cout<<"Departure time    : ";
   cin>>train[T].Depart;
   cout<<"From              : ";
   cin>>train[T].From;
```

```
    cout<<"To              : ";
    cin>>train[T].To;
    train[T].Empty();
    T++;
}
void TRAIN::Train_Availability()
{
    for(int n=0;n<T;n++)
    {
        cout<<".....\n";
        cout<<"\nTrain Number: "<<train[n].T_No
        <<"\nArrival time: "<<train[n].Arrival
        <<"\t\tDeparture Time: "<<train[n].Depart
        <<"\nFrom       : "<<train[n].From
        <<"\t\tTo          : "<<train[n].To<<"\n";
        cout<<".....\n";
    }
}
void TRAIN::Booking()
{
    int seat,n,age;
    char num[10];
    Top:
        cout<<"Train Number: ";
        cin>>num;
        for(n=0;n<=T;n++)
        {
            if(strcmp(train[n].T_No, num)==0)
            break;
```

```
}
while(n<=T)
{
   cout<<"Number of the seat: ";
   cin>>seat;
   if(seat>32)
   {
      cout<<"\nThere are only 32 seats Available in this train.\n";
   }
   else
   {
      if (strcmp(train[n].seat[seat/2][(seat%2)-1], "Empty")==0)
      {
         cout<<"Enter passenger's name: ";
         cin>>train[n].seat[seat/2][(seat%2)-1];
         cout<<"Enter the Age :";
         cin>>age;
         break;
      }
      else
      {
         cout<<"The seat Number is already reserved.\n";
      }
   }
}
if(n>T)
{
cout<<"Enter correct Train Number.\n";
goto Top;
```

```
        }
}
void TRAIN::Empty()
{
   for(int i=0; i<16;i++)
   {
      for(int j=0;j<2;j++)
      {
         strcpy(train[T].seat[i][j], "Empty");
      }
   }
}
void TRAIN::Allotment()
{
   int n,i,j;
   char number[5];
   top:
   cout<<"Enter Train Number: ";
   cin>>number;
   for(n=0;n<=T;n++)
   {
      if(strcmp(train[n].T_No, number)==0)
      break;
   }
   while(n<=T)
   {
      cout<<".....\n";
      cout<<"\nTrain Number: "<<train[n].T_No
      <<"\nArrival time: "<<train[n].Arrival
```

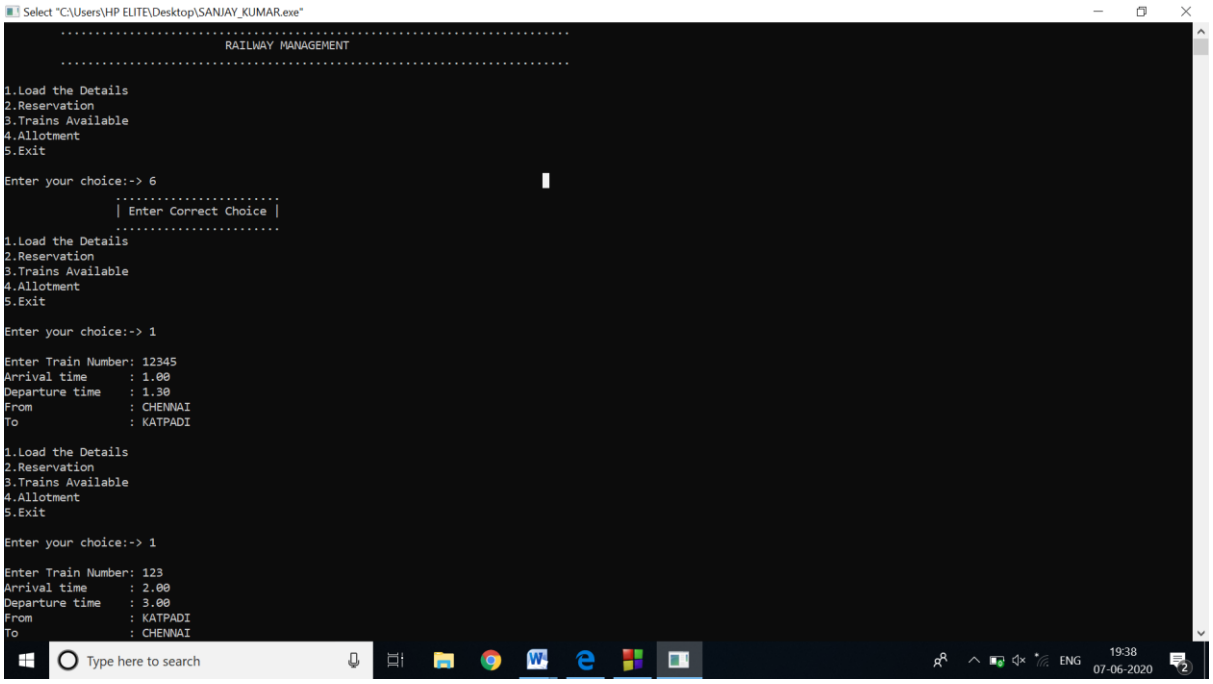
```
        <<"\t\tDeparture Time: "<<train[n].Depart
        <<"\nFrom       : "<<train[n].From
        <<"\t\tTo          : "<<train[n].To<<"\n";
        cout<<".....\n";
        train[T].position(n);
        int a=1;
        for (int i=0; i<16; i++)
        {
           for(int j=0;j<2;j++)
              a++;
           if(strcmp(train[n].seat[i][j],"Empty")!=0)
              cout<<"\nThe seat number "<<(a-1)<<" is reserved for "<<train[n].seat[i][j]<<".";
        }
     }
     if(n>T)
     {
        cout<<"\nEnter correct Train Number: ";
        goto top;
     }

}
void TRAIN::position(int l)
{
     int s=0;T=0;
     for (int i =0; i<16;i++)
     {
        cout<<"\n";
        for (int j = 0;j<2; j++)
        {
```

```
            s++;
            if(strcmp(train[l].seat[i][j], "Empty")==0) {
               cout.width(5);
               cout<<s<<".";
               cout.width(10);
               cout<<train[l].seat[i][j];
               T++;
            }
            else{
               cout.width(5);
               cout<<s<<".";
               cout.width(10);
               cout<<train[l].seat[i][j];
            }
         }
      }
      cout<<"\n\nThere are "<<T<<" seats empty in Train Number: "<<train[l].T_No;
}
int main()
{
   int ch;
   cout<<"\t.....\n";
   cout<<"\t\t\t\tRAILWAY MANAGEMENT\n";
   cout<<"\t.....\n";
   while(1)
   {
      cout<<"\n";
      cout<<"1.Load the Details \n"
      <<"2.Reservation\n"
```

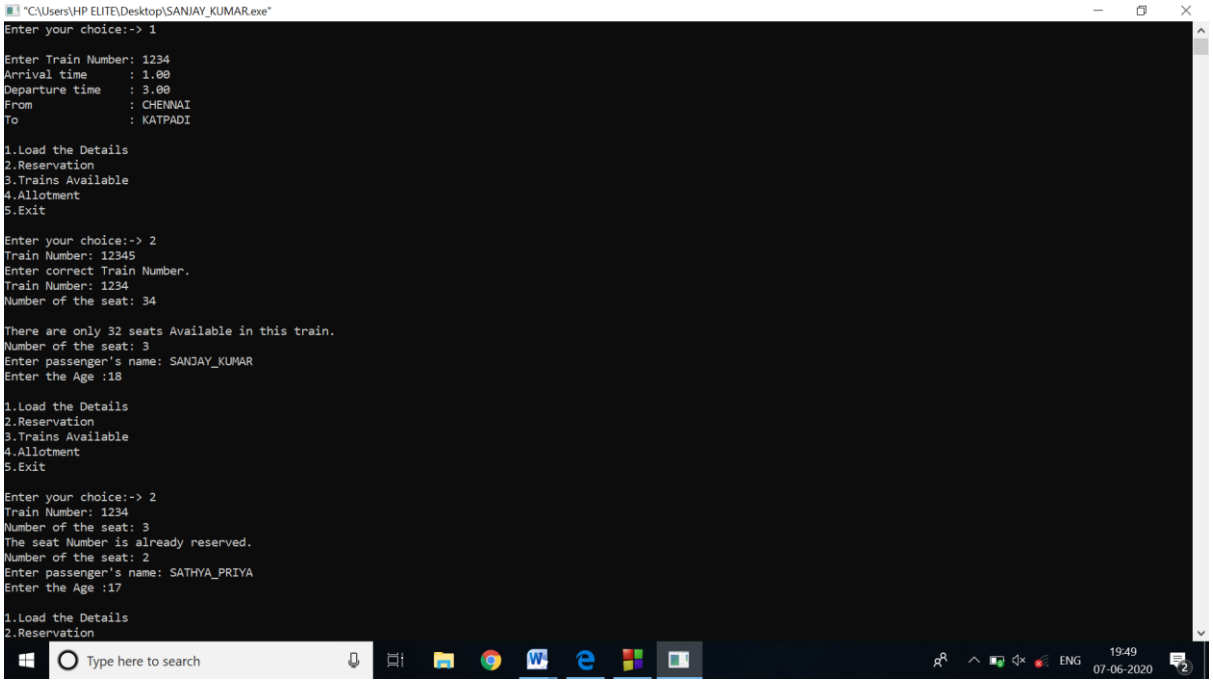
```
        <<"3.Trains Available\n"
        <<"4.Allotment\n"
        <<"5.Exit\n";
        cout<<"\nEnter your choice:-> ";
        cin>>ch;
        switch(ch)
        {
           case 1  :
                 train[T].Upload();
                 break;
           case 2  :
                 train[T].Booking();
                 break;
           case 3  :
                 train[T].Train_Availability();
                 break;
           case 4  :
                 train[T].Allotment();
                 break;
           case 5  :
                 return 0;
           default :
                 cout<<"\t\t.....";
                 cout<<"\n\t\t| Enter Correct Choice |\n";
                 cout<<"\t\t.....";
        }
}
   return 0;
}
```

4.2 Results and Discussions

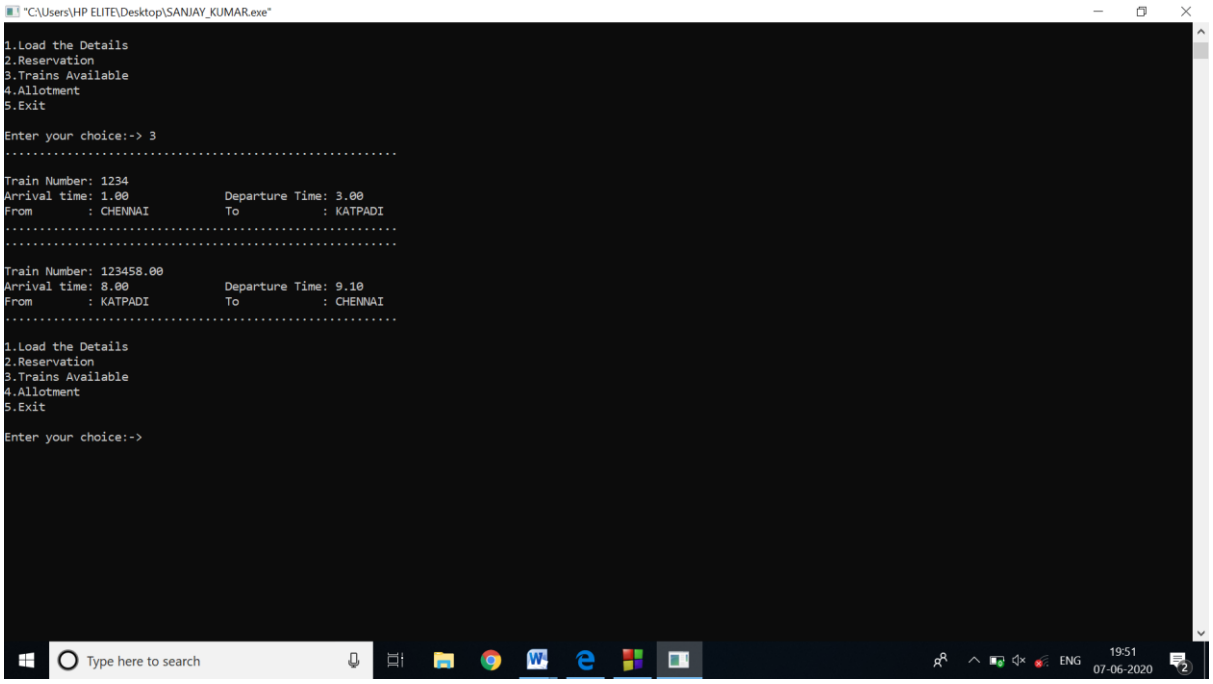
1.Load The Trains Details :



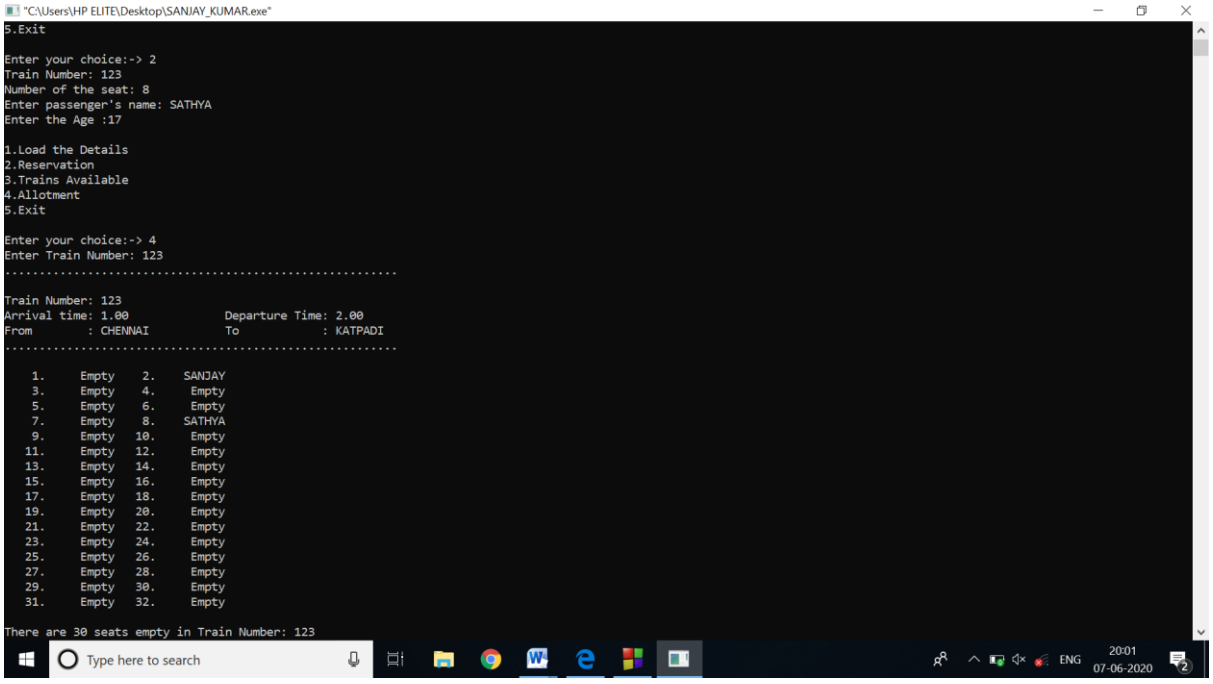
Reservation :



Availability :



Allotments :



6.REFERENCES

- PROGRAMMING C++ & DATA STRUCTURES__BALAGURUSAMY
- https://play.google.com/store/apps/detail?id=cpp.programming