ACKNOWLEDGEMENT

We would like to express our deepest gratitude to Mrs. Apeksha Aggarwal Ma'am, our professor-in-charge for their tremendous support and guidance in completing our project on the topic- Railway Management System. It was a great learning experience for the group to learn the basic applications of Data structures and Algorithms and more importantly the spirit to work in a group.

We would also like to take this opportunity to expand our gratitude to our families, friends, and colleagues. The project would not have been successful without their cooperation and inputs.

Team (B14-IT)

Name	Shubham Rawat	Shikhar Gupta	Shahrukh Khan	Tanmay Singh
Enrolment No.	21104002	21104006	21104021	21104022

CHAPTER 1: Problem Statement & Objectives

1.1 WHY THIS PROJECT

The reason for choosing this project was due to its ability to accomplish different concepts of data structures and algorithms. It has the ability to be used as a real time project for ease of travellers and passengers. We wanted to apply our knowledge of C++ programming language to develop a project related to railway management system.

1.2 OBJECTIVES

- Main objective of this project is to explore or to display the stored information.

  1. Uses vectors to assemble a train from different coaches.
  2. Uses Maps (int to string) to map stations to locations.
  3. Uses Graphs to assemble route for the train.
  4. Uses file handling to store train information in text file.

- It uses different concepts of linked list, vectors, and graphs; moreover, the project comprises the knowledge of different algorithms learned through the course curriculum.

CHAPTER 2: Approach

➢ Project Conception and Initiation

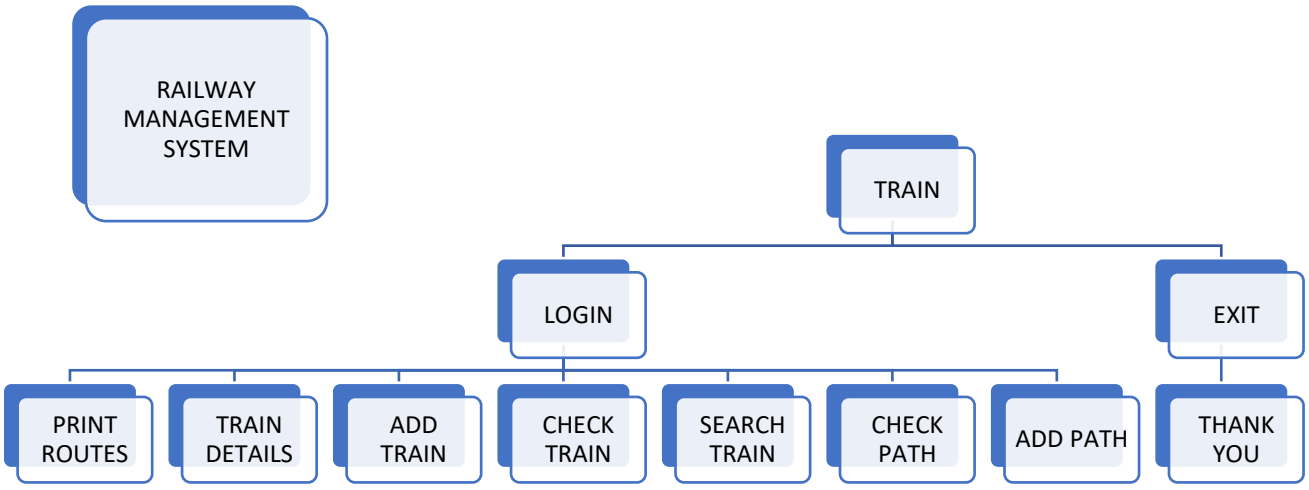
The concept was to use the data structures learned throughout the course curriculum and use it in Railway management as Railways are the lifeline of the country's Transportation, which enables the maintainer the ease of usage.

➢ Project Definition and Planning

Project definition basically includes the use of Vectors, graphs and file handling along with the concepts of graph traversal.

➢ Project Launch and Execution

Project launch is basically dealing with the error debugging of codeblocks and getting comfortable with different file formats.



CHAPTER 3: Code Snippets

```
#include<bits/stdc++.h>
#include<string.h>
#include<map>
using namespace std;
string
location[10]={"NEWDELHI","MUMBAI","BENGALURU","CHENNAI","KOLKATA
","JAIPUR","BHOPAL","LUCKNOW","CHANDIGARG","AHMEDABAD"};
map<int, string> mti;
map<char,string> m;
vector <int> graph[10];
bool vis[10];
int k=0;
void map_out()
{
    string
location[10]={"NEWDELHI","MUMBAI","BENGALURU","CHENNAI","KOLKATA
","JAIPUR","BHOPAL","LUCKNOW","CHANDIGARG","AHMEDABAD"};
    string num="0123456789";
    for(int i=0;i<10;i++)
     {
         char s1=num[i];
         string s=location[i];
         m.insert(pair<char, string>(s1, s));
     }
/*
     for (auto itr = m.begin(); itr != m.end(); ++itr)
    {
        cout << itr->first<< '\t' << itr->second << '\n';
    }
*/
}
string find_s(char s)
{
    for (auto itr = m.begin(); itr != m.end(); ++itr)
    {
        if(itr->first==s)
        {
            return itr->second;
        }
    }
}
char find_c(string s)
{
    for (auto itr = m.begin(); itr != m.end(); ++itr)
    {
        if(itr->second==s)
```

```
        {
            return itr->first;
        }
    }
}
int find_out(string s)
{
    for (int i=0;i<10;i++)
    {
        if(location[i]==s)
        {
            return i;
        }
    }

    return -1;
}
string find_it(int x)
{
      for (auto itr = mti.begin(); itr != mti.end(); ++itr)
    {
        if(itr->first==x)
        {
            return itr->second;
        }
    }
}
void add(vector<int> graph[],int s,int d)
{
    graph[s].push_back(d);
    graph[d].push_back(s);
}
void printgraph(vector<int> graph[])
{
    cout<<"AVAILABLE ROUTES ";
    for (auto itr = mti.begin(); itr != mti.end(); ++itr)
    {
        cout<<"\n ROUTE :"<<find_it(itr->first);
        for(auto x: graph[itr->first])
        {
            cout<<"->"<<find_it(x);
        }
        cout<<endl;
    }
}
void map_it()
{
    // initialized a string array
    string
location[10]={"NEWDELHI","MUMBAI","BENGALURU","CHENNAI","KOLKATA
","JAIPUR","BHOPAL","LUCKNOW","CHANDIGARG","AHMEDABAD"};
```

```
    // mapping of string to integer
    for(int i=0;i<10;i++)
     {
         mti.insert({i,location[i]});
     }
/*
    for (auto itr = mti.begin(); itr != mti.end(); ++itr)
    {
        cout << itr->first<< '\t' << itr->second << '\n';
    }
*/

}
void train_details()
{
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"                  TRAIN DETAILS
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<endl;
    fstream file;
    string word,filename;
    string get;
    filename="serial1.txt";

    file.open(filename.c_str());
    int num1,num2;
     while (file >> word)
    {
        // displaying content
        get=word;

        cout<<"TRAIN ID: "<<get<<endl;
        cout<<"Source Station: "<<find_s(get[0])<<endl;
        cout<<"Destination Station: "<<find_s(get[1])<<endl;
```

```
        cout<<"Departing Time:
"<<get[2]<<get[3]<<":"<<get[4]<<get[5]<<endl;
        cout<<"Arriving Time:
"<<get[6]<<get[7]<<":"<<get[8]<<get[9]<<endl;
        cout<<endl;

    }
}
void search_train()
{
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"                  TRAIN DETAIL
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<endl;
    fstream file;
    string word, t, q, filename;
    string get;
    filename="serial1.txt";

    string srch;
    int flag=0;
    cout<<"ENTER THE UNIQUE TRAIN IDENTIFICATION NUMBER: ";
    cin>>srch;
    file.open(filename.c_str());
    int num1,num2;
     while (file >> word)
    {
        // displaying content
        if(word==srch)
        {
        cout<<"TRAIN FOUND"<<endl;
        get=word;

        cout<<"TRAIN ID: "<<get<<endl;
```

```
        cout<<"Source Station: "<<find_s(get[0])<<endl;
        cout<<"Destination Station: "<<find_s(get[1])<<endl;
        cout<<"Departing Time:
"<<get[2]<<get[3]<<":"<<get[4]<<get[5]<<endl;
        cout<<"Arriving Time:
"<<get[6]<<get[7]<<":"<<get[8]<<get[9]<<endl;
        cout<<endl;
        }
        flag=1;
    }
    if(flag==0)
    {
      cout<<"INVALID TRAIN ID .....TRAIN NOT FOUND"<<endl;
    }
}
void add_train()
{
    ofstream my_train("serial1.txt",ios::app);
    string train_num;

    // Instructions for entering the train identification number
    cout<<endl;
    cout<<"KINDLY READ THE INSTRUCTIONS CAREFULLY TO ALLOT A
TRAIN-ID "<<endl;
    cout<<"TRAIN NUMBER FORMAT A-B-XXXX-YYYY"<<endl;
    cout<<"A-> SOURCE STATION CODE "<<endl;
    cout<<"B-> DESTINATION STATION CODE"<<endl;
    cout<<"XXXX-> DEPARTING TIME"<<endl;
    cout<<"YYYY-> ARRIVAL TIME"<<endl;
    cout<<"KINDLY USE 24HR TIME FORMAT"<<endl;
    cout<<endl;

    for (auto itr = mti.begin(); itr != mti.end(); ++itr)
    {
        cout << itr->first<< '\t' << itr->second << '\n';
    }
    cout<<endl;
    cout<<"ENTER THE UNIQUE TRAIN IDENTIFICATION NUMBER: ";
    cin>>train_num;
    my_train<<train_num<<endl;

    cout<<endl;

    my_train.close();
}
void check_trains()
{
    string a;
    string b;
    cout<<"ENTER THE SOURCE STATION TO START YOUR JOURNEY: ";
    cin>>a;
```

```
    cout<<"ENTER THE DESTINATION STATION TO END YOUR JOURNEY: ";
    cin>>b;
    char c1;
    char c2;
    c1=find_c(a);
    c2=find_c(b);

    int flag=0;

    fstream file;
    string word, t, q, filename;
    string get;
    filename="serial1.txt";

    file.open(filename.c_str());

    while (file >> word)
    {
        // displaying content
        get=word;

        if(get[0]==c1 && get[1]==c2)
        {
        cout<<"TRAIN ID: "<<get<<endl;
        cout<<"Source Station: "<<find_s(get[0])<<endl;
        cout<<"Destination Station: "<<find_s(get[1])<<endl;
        cout<<"Departing Time:
"<<get[2]<<get[3]<<":"<<get[4]<<get[5]<<endl;
        cout<<"Arriving Time:
"<<get[6]<<get[7]<<":"<<get[8]<<get[9]<<endl;
        cout<<endl;

            flag=1;
            break;
        }

    }
    if(flag==0)
    {
        cout<<"TRAIN NOT FOUND !!!!";
    }
}
bool haspath(vector<int> graph[],int src,int dest,bool vis[],int
size1)
{
    if(src==dest)
        return true;

    vis[src]=true;
        for(int i=0;i<size1;i++)        // check for it's
neighbors
```

```
        {
            for(auto x: graph[i])
            {
                if(!vis[x] && haspath(graph,x,dest,vis,size1))
                    return true;
            }
    }
    return false;
}
void check_path()
{
    string a;
    string b;
    cout<<"ENTER THE SOURCE STATION TO START YOUR JOURNEY: ";
    cin>>a;
    cout<<"ENTER THE DESTINATION STATION TO END YOUR JOURNEY: ";
    cin>>b;
    int n1,n2;
    n1=find_out(a);
    n2=find_out(b);
    if(0<=n1 && n1<=9 && 0<=n2 && n2<=9)
    {
    if(haspath(graph,n1,n2,vis,10))
        {
            cout<<"PATH EXIST BETWEEN "<<a<<" AND "<<b<<endl;
        }
    else
        {
            cout<<"PATH DON'T EXIST BETWEEN "<<a<<" AND
"<<b<<endl;
        }
    }
    else
    {
        cout<<"PATH DON'T EXIST BETWEEN "<<a<<" AND "<<b<<endl;
    }
}
void add_path()
{
    string a;
    string b;
    cout<<"ENTER THE SOURCE STATION TO START YOUR JOURNEY: ";
    cin>>a;
    cout<<"ENTER THE DESTINATION STATION TO END YOUR JOURNEY: ";
    cin>>b;

    int n1,n2;
    n1=find_out(a);
    n2=find_out(b);

   // cout<<n1<<" "<<n2;
```

```
    if(0<=n1 && n1<=9 && 0<=n2 && n2<=9)
    {
    add(graph,n1,n2);
    cout<<endl;
    cout<<"PATH ADDED SUCCESSFULLY";
    }
    else
    {
        cout<<"INVALID STATIONS ENTERED ";
    }
}
int main()
{

    map_it();
    map_out();

    add(graph,1,2);
    add(graph,1,3);
    add(graph,1,4);
    add(graph,2,5);
    add(graph,5,8);
    add(graph,3,8);
    add(graph,4,6);
    add(graph,6,7);
    add(graph,7,8);
    add(graph,0,9);
    add(graph,0,1);

    // function call to allot all seats to true

    cout<<endl;
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"       DATA STRUCTURES AND ALGORITHMS PROJECT
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
```

```
    cout<<"
";

    string user_id;
    string password;

    cout<<endl;


   ofstream my_file("username.txt",ios::app);

   cout<<"ENTER YOUR USERID TO START: "<<endl;
   cin>>user_id;
   my_file<<user_id;

   my_file<<endl;
   my_file.close();

    cout<<"ENTER PASSWORD TO LOGIN: "<<endl;
    cin>>password;

    int choice;
    do
    {
    if(password=="abcd123")
    {
    cout<<endl;
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"!!!!!!!!!!!!!!WELCOME TO INDIAN
RAILWAYS!!!!!!!!!!!!!!!";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<"
";
    cout<<endl;

        int info;
        cout<<"MAIN MENU: "<<endl;
```

```
        cout<<"PRESS 1 TO LIST TRAIN DETAILS: "<<endl;
        cout<<"PRESS 2 ADD TRAIN DETAILS IN DATABASE: "<<endl;
        cout<<"PRESS 3 TO CHECK TRAINS BETWEEN STATIONS:
"<<endl;
        cout<<"PRESS 4 TO TO ADD PATH BETWEEN TWO STATIONS:
"<<endl;
        cout<<"PRESS 5 TO CHECK PATH BETWEEN STATIONS: "<<endl;
        cout<<"PRESS 6 TO CHECK ALL AVAILABLE ROUTES: "<<endl;
        cout<<"PRESS 7 TO SEARCH TRAIN: "<<endl;
        cout<<"PRESS 8 TO EXIT: "<<endl;
        cout<<endl;
        cout<<"KINDLY PRESS TO PROCEED FURTHER: ";
        cin>>info;

        switch(info)
            {
                case 1:
                    system("cls");
                    train_details();
                    break;

                case 2:
                    system("cls");
                    add_train();
                    break;

                case 3:
                    system("cls");
                    check_trains();
                    break;

                case 4:
                    system("cls");
                    add_path();
                    break;

                case 5:
                    system("cls");
                    check_path();
                    break;

                case 6:
                    system("cls");
                    printgraph(graph);
                    break;

                case 7:
                    search_train();
                    break;

                case 8:
```

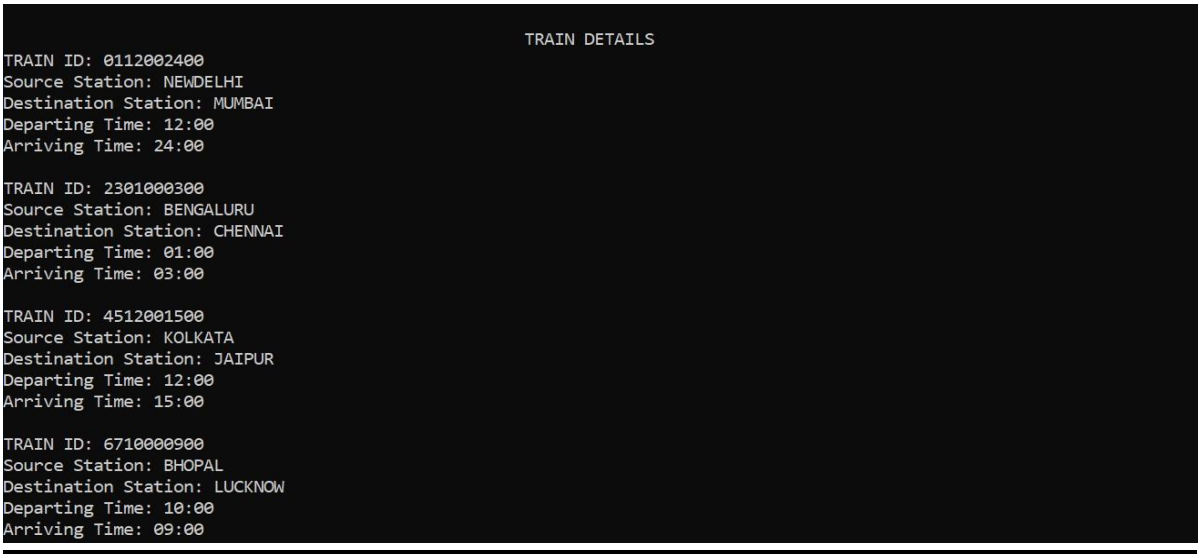
```
                    cout<<endl;
                    cout<<"THANK YOU FOR YOUR VISIT....INDIAN
RAILWAYS HOPES FOR YOU HAPPY JOURNEY";
                    exit(1);

                default:
                    cout<<"THANK YOU FOR YOUR VISIT....INDIAN
RAILWAYS HOPES FOR YOU HAPPY JOURNEY";
                    exit(1);
            }
    }
    else
    {
        cout<<"RE-ENTER THE PASSWORD OR PRESS 0 TO EXIT "<<endl;
        cin>>password;
    }
    }while(password!="0");

     cout<<endl;
     cout<<endl;
     cout<<endl;
     cout<<endl;
     cout<<"THANK YOU "<<endl;

    return 0;
}
```

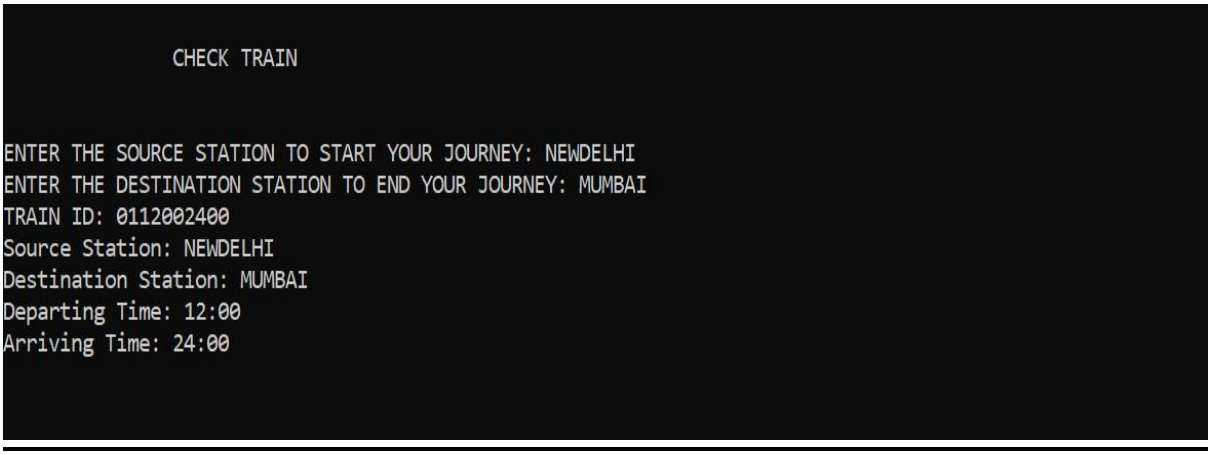
CHAPTER 4: Output Snapshot

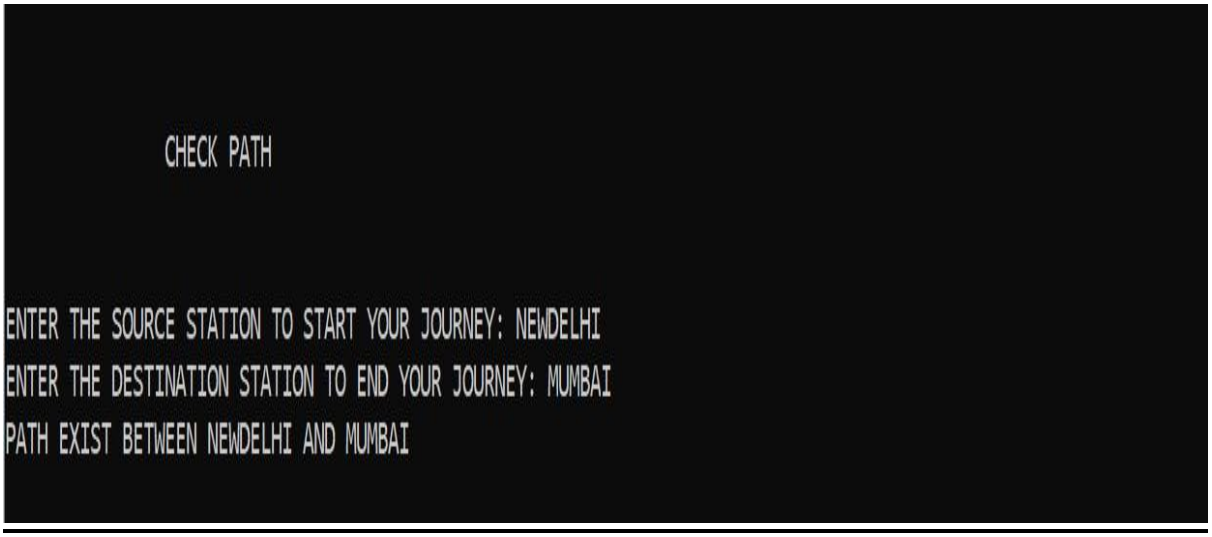
```
                              !!!!!!!!!!!!!!WELCOME TO INDIAN RAILWAYS!!!!!!!!!!!!!!!


MAIN MENU:
PRESS 1 TO LIST TRAIN DETAILS:
PRESS 2 ADD TRAIN DETAILS IN DATABASE:
PRESS 3 TO CHECK TRAINS BETWEEN STATIONS:
PRESS 4 TO TO ADD PATH BETWEEN TWO STATIONS:
PRESS 5 TO CHECK PATH BETWEEN STATIONS:
PRESS 6 TO CHECK ALL AVAILABLE ROUTES:
PRESS 7 TO SEARCH TRAIN:
PRESS 8 TO EXIT:

KINDLY PRESS TO PROCEED FURTHER:
```

```
                                DATA STRUCTURES AND ALGORITHMS PROJECT



ENTER YOUR USERID TO START:
Shikhar_07
ENTER PASSWORD TO LOGIN:
abcd123
```

```
KINDLY READ THE INSTRUCTIONS CAREFULLY TO ALLOT A TRAIN-ID
TRAIN NUMBER FORMAT A-B-XXXX-YYYY
A-> SOURCE STATION CODE
B-> DESTINATION STATION CODE
XXXX-> DEPARTING TIME
YYYY-> ARRIVAL TIME
KINDLY USE 24HR TIME FORMAT

0       NEWDELHI
1       MUMBAI
2       BENGALURU
3       CHENNAI
4       KOLKATA
5       JAIPUR
6       BHOPAL
7       LUCKNOW
8       CHANDIGARG
9       AHMEDABAD

ENTER THE UNIQUE TRAIN IDENTIFICATION NUMBER: 0217001900
```


TRAIN DETAILS
TRAIN ID: 0112002400
Source Station: NEWDELHI
Destination Station: MUMBAI
Departing Time: 12:00
Arriving Time: 24:00
TRAIN ID: 2301000300
Source Station: BENGALURU
Destination Station: CHENNAI
Departing Time: 01:00
Arriving Time: 03:00
TRAIN ID: 4512001500
Source Station: KOLKATA
Destination Station: JAIPUR
Departing Time: 12:00
Arriving Time: 15:00
TRAIN ID: 6710000900
Source Station: BHOPAL
Destination Station: LUCKNOW
Departing Time: 10:00
Arriving Time: 09:00


CHECK TRAIN
ENTER THE SOURCE STATION TO START YOUR JOURNEY: NEWDELHI
ENTER THE DESTINATION STATION TO END YOUR JOURNEY: MUMBAI
TRAIN ID: 0112002400
Source Station: NEWDELHI
Destination Station: MUMBAI
Departing Time: 12:00
Arriving Time: 24:00


CHECK PATH
ENTER THE SOURCE STATION TO START YOUR JOURNEY: NEWDELHI
ENTER THE DESTINATION STATION TO END YOUR JOURNEY: MUMBAI
PATH EXIST BETWEEN NEWDELHI AND MUMBAI

```
AVAILABLE ROUTES

 ROUTE :NEWDELHI

 ROUTE :MUMBAI->BENGALURU->CHENNAI->KOLKATA

 ROUTE :BENGALURU->MUMBAI->JAIPUR

 ROUTE :CHENNAI->MUMBAI->CHANDIGARG

 ROUTE :KOLKATA->MUMBAI->BHOPAL

 ROUTE :JAIPUR->BENGALURU->CHANDIGARG

 ROUTE :BHOPAL->KOLKATA->LUCKNOW

 ROUTE :LUCKNOW->BHOPAL->CHANDIGARG

 ROUTE :CHANDIGARG->JAIPUR->CHENNAI->LUCKNOW

 ROUTE :AHMEDABAD
```

```
                                        TRAIN DETAIL


ENTER THE UNIQUE TRAIN IDENTIFICATION NUMBER: 1234567890
TRAIN FOUND
TRAIN ID: 1234567890
Source Station: MUMBAI
Destination Station: BENGALURU
Departing Time: 34:56
Arriving Time: 78:90

TRAIN FOUND
TRAIN ID: 1234567890
Source Station: MUMBAI
Destination Station: BENGALURU
Departing Time: 34:56
Arriving Time: 78:90
```

```
                        !!!!!!!!!!!!!!WELCOME TO INDIAN RAILWAYS!!!!!!!!!!!!!!!


MAIN MENU:
PRESS 1 TO LIST TRAIN DETAILS:
PRESS 2 ADD TRAIN DETAILS IN DATABASE:
PRESS 3 TO CHECK TRAINS BETWEEN STATIONS:
PRESS 4 TO TO ADD PATH BETWEEN TWO STATIONS:
PRESS 5 TO CHECK PATH BETWEEN STATIONS:
PRESS 6 TO CHECK ALL AVAILABLE ROUTES:
PRESS 7 TO SEARCH TRAIN:
PRESS 8 TO EXIT:

KINDLY PRESS TO PROCEED FURTHER: 8

THANK YOU FOR YOUR VISIT....INDIAN RAILWAYS HOPES FOR YOU HAPPY JOURNEY
Process returned 1 (0x1)   execution time : 32.114 s
Press any key to continue.
```

CHAPTER 5: Conclusion & Future Work

5.1 Conclusion

The project comprises of different functions:

A. To search train using:
   1. Train Id
   2. Source station and destination station

B. Display Train Details

C. Add path between two locations

D. Check for all Available routes

Moreover, the project has the capability to be used in real world scenario to help organization like IRCTC to maintain and track the record for future uses and purposes.

5.2 Future Work

A. The project can be modified into weighted graphs which enables them to:
   1. Find shortest route between stations.
   2. Check for time between stations using average train speeds.

B. Boolean arrays can be stored alongside Train Id which enables users to book birth and check for seat birth availability.

C. N- Array Trees can be used to form a structure of Birth allotment.

D. Doubly Linked List can be used to connect train coaches upon further division.

CHAPTER 6: References

- https://www.geeksforgeeks.org/data-structures/linked-list/
- https://www.programiz.com/cpp-programming/vectors
- Wikipedia