PROJECT:RAILWAY TICKET MANAGEMENT SYSTEM

NAME –ABHINAV MENON
SAP ID – 590021930
SUBJECT-PROGRAMMING IN C

INTRODUCTION

- The *Railway Ticket Management System* is a C programming project designed to simplify the process of booking railway tickets. In traditional methods, passengers have to stand in long queues and manually fill forms to book their tickets. This system helps to automate the booking process by storing passenger details, selecting destinations, choosing trains, calculating fares, and generating tickets through a computer program.

Purpose of this Project:

- The Purpose of this Project is to simplify the ticket booking services.
- The Project aims to reduce human effort , save time , and minimize errors during manually booking.
- It also helps to understand how real world applications such as structures , functions , loops and many more.
- Overall , the system provides an efficient and user-friendly way to manage railway ticket operations.

Constraints:

- Limited Data storage
- No real time train information
- Single user access
- No online payment system
- Fixed train and Route data

Features of the system:

- View train list
- Book a ticket
- Display ticket details
- Calculate fare
- Exit the system

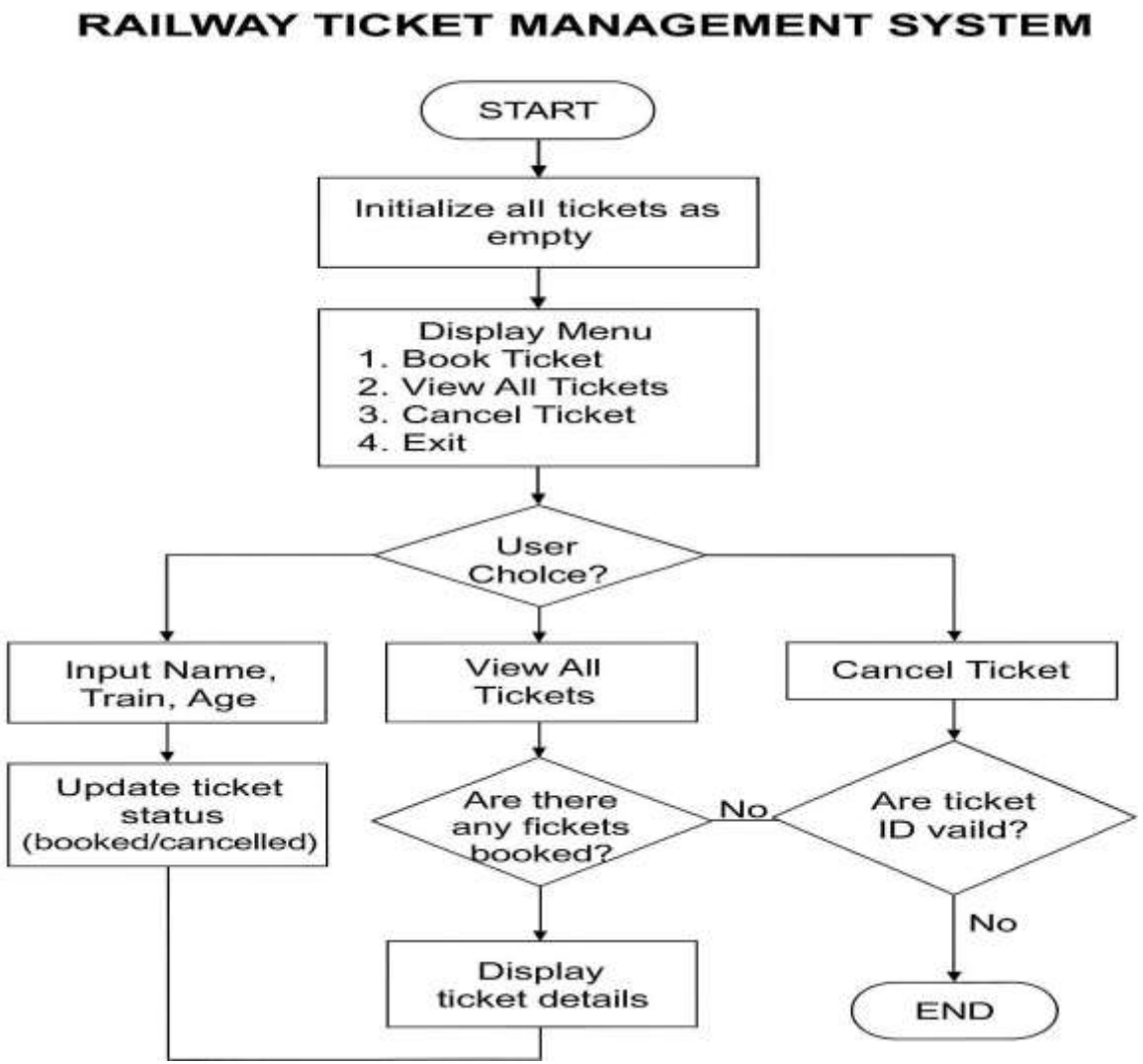
Syntax Concepts Used:

- Variables and Data types
- Input and output statements
- Conditional statements
- Loops
- Functions
- Structures
- Arrays
- Switch case

Keywords Used:

- Struct , for , while , if , else if , else .
- Int , char , #include , #define

Flowchart:


RAILWAY TICKET MANAGEMENT SYSTEM
START
Initialize all tickets as empty
Display Menu
1. Book Ticket
2. View All Tickets
3. Cancel Ticket
4. Exit
User Cholce?
Input Name, Train, Age
Update ticket status (booked/cancelled)
View All Tickets
Are there any fickets booked?
No
Display ticket details
Cancel Ticket
Are ticket ID vaild?
No
END

PROGRAM:

```c
#include "railway.h"

// Fixed trains list
const char trains[TRAIN_COUNT][30] = {
    "Rajdhani Express",
    "Shatabdi Express",
    "Duronto Express"
};

void init(struct Ticket t[]) {
    for (int i = 0; i < MAX; i++) {
        t[i].booked = 0;
    }
}

void showTrains() {
    printf("\nAvailable Trains:\n");
    for (int i = 0; i < TRAIN_COUNT; i++) {
        printf("%d. %s\n", i + 1, trains[i]);
    }
}

void display(struct Ticket t[]) {
    printf("\n--- Ticket List ---\n");
    for (int i = 0; i < MAX; i++) {
        if (t[i].booked == 1) {
            printf("Seat %d: %s | Train: %s | Age: %d\n",
                   i, t[i].name, t[i].train, t[i].age);
        } else {
            printf("Seat %d: EMPTY\n", i);
        }
    }
}

void book(struct Ticket t[]) {
    int seat = -1;
    for (int i = 0; i < MAX; i++) {
        if (t[i].booked == 0) {
            seat = i;
            break;
        }
    }

    if (seat == -1) {
        printf("\nNo seats available!\n");
        return;
    }

    printf("\nEnter Name: ");
    scanf("%s", t[seat].name);

    showTrains();
    int trainChoice;

    printf("Select Train (1-%d): ", TRAIN_COUNT);
    scanf("%d", &trainChoice);
```

```c
    if (trainChoice < 1 || trainChoice > TRAIN_COUNT) {
        printf("Invalid train choice!\n");
        return;
    }

    strcpy(t[seat].train, trains[trainChoice - 1]);

    printf("Enter Age: ");
    scanf("%d", &t[seat].age);

    t[seat].booked = 1;

    printf("\nTicket booked! Seat Number: %d\n", seat);
}

void cancel(struct Ticket t[]) {
    int seat;
    printf("\nEnter seat number to cancel: ");
    scanf("%d", &seat);

    if (seat < 0 || seat >= MAX) {
        printf("Invalid seat number!\n");
        return;
    }

    if (t[seat].booked == 0) {
        printf("Seat already empty!\n");
    } else {
        t[seat].booked = 0;
        printf("Ticket cancelled successfully!\n");
    }
}
```

MAIN.c

```c
#ifndef RAILWAY_H
#define RAILWAY_H

#include <stdio.h>
#include <string.h>

#define MAX 5   // max 5 tickets
#define TRAIN_COUNT 3

struct Ticket {
    char name[30];
    char train[30];
    int age;
    int booked;   // 1 = booked, 0 = empty
};

// Function declarations
void init(struct Ticket t[]);
void display(struct Ticket t[]);
void book(struct Ticket t[]);
void cancel(struct Ticket t[]);
void showTrains();

#endif

```

Header file

```c
#ifndef RAILWAY_H
#define RAILWAY_H

#include <stdio.h>
#include <string.h>

#define MAX 5   // max 5 tickets
#define TRAIN_COUNT 3

struct Ticket {
    char name[30];
    char train[30];
    int age;
    int booked;   // 1 = booked, 0 = empty
};

// Function declarations
void init(struct Ticket t[]);
void display(struct Ticket t[]);
void book(struct Ticket t[]);
void cancel(struct Ticket t[]);
void showTrains();

#endif

```

OUTPUT:

```
--- Railway Ticket Menu ---
1. Display Tickets
2. Book Ticket
3. Cancel Ticket
4. Show Train List
5. Exit
Enter choice: 1

--- Ticket List ---
Seat 0: EMPTY
Seat 1: EMPTY
Seat 2: EMPTY
Seat 3: EMPTY
Seat 4: EMPTY

--- Railway Ticket Menu ---
1. Display Tickets
2. Book Ticket
3. Cancel Ticket
4. Show Train List
5. Exit
Enter choice: 2

Enter Name: Abhijeet

Available Trains:
1. Rajdhani Express
2. Shatabdi Express
3. Duronto Express
Select Train (1-3): 1
Enter Age: 18

Ticket booked! Seat Number: 0

--- Railway Ticket Menu ---
1. Display Tickets
2. Book Ticket
3. Cancel Ticket
4. Show Train List
5. Exit
```

```
Enter choice: 4

Available Trains:
1. Rajdhani Express
2. Shatabdi Express
3. Duronto Express

--- Railway Ticket Menu ---
1. Display Tickets
2. Book Ticket
3. Cancel Ticket
4. Show Train List
5. Exit
Enter choice: 3

Enter seat number to cancel: 0
Ticket cancelled successfully!

--- Railway Ticket Menu ---
1. Display Tickets
2. Book Ticket
3. Cancel Ticket
4. Show Train List
5. Exit
Enter choice: 5
```

Conclusion:

- • The project demonstrates basic C programming concepts.
- • It shows how a real-life system can be simulated through code.
- • Students learn structured programming, modular design, and logical flow.