



Subject: PROGRAMMING IN C

PROJECT: TRAIN TICKET MANAGEMENT SYSTEM

BTech CSE 1st Year

Name: SHREY BISHT & GAURAV LODHIYAL

SAP ID: 590027563 & 590027771

Batch: 68

Semester: 1

Year: 2025

Date of submission: 30-11-2025

FACULTY: Mr. VINOD KUMAR

TRAIN TICKET RESERVATION SYSTEM-PROJECT DOCUMENT

1. INTRODUCTION:

The Train Ticket Reservation System is a C-language based console application designed to simplify booking and cancellation of train tickets. It provides an easy-to-use menu-driven interface where users can view available trains, reserve seats, generate PNR numbers, cancel tickets, and view existing bookings.

This project demonstrates the use of arrays, structures, file handling, functions, and control statements in C programming.

2. OBJECTIVES:

- .To develop a simple reservation system in C.
- .To store and manage passenger booking information.
- .To generate unique PNR numbers for each booking.
- .To demonstrate file handling for saving and loading bookings.
- .To implement a user-friendly menu interface.

3. SYSTEM FEATURES:

✓ A. List Available Trains

Shows train details such as Train ID, Name, Source, Destination, and Available Seats.

✓ B. Book a Ticket

User enters passenger name, age, gender, and train ID.

The system assigns an available seat and generates a unique PNR.

✓ C. Cancel Ticket

User enters PNR, and if found, the booking gets cancelled.

✓ D. View All Bookings

Displays all bookings with:

PNR

Train ID

Passenger name

Age

Gender

Seat number

Status (Booked/Cancelled)

✓ E. File Handling (Bonus)

All bookings are saved in a binary file (bookings.dat) so they remain available after the program closes.

4. TOOLS & TECHNOLOGIES

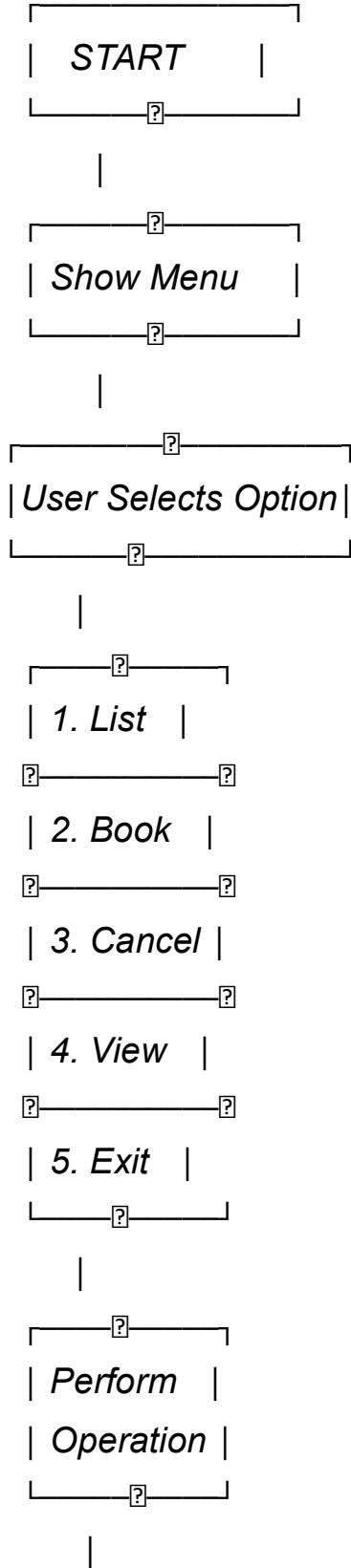
USED: .Programming Language: C

.Concepts Used:

.Structures

.*Arrays*
. *Functions*
. *File Handling*
. *Loops & Conditionals*
. *String handling*
. *Compiler: GCC or CodeBlocks*

5.FLOW CHART:



```

[?] ---| Continue? |---No---> END
[?] ---| Yes
[ ] ---> Back to Menu

```

6.DATA STRUCTURES USED:

Train Structure:-

```

typedef struct {
    int train_id;
    char name[50];
    char source[30];
    char dest[30];
    int total_seats;
    int seats_booked;
} Train;

```

Booking Structure:-

```

typedef struct {
    char pnr[20];
    int train_id;
    char passenger_name[50];
    int age;
    char gender[10];
    int seat_no;
    int active;
} Booking;

```

7.WORKING OF THE SYSTEM:

1. The program loads the trains from predefined data.
2. When a user books a ticket:
 - . The system checks if seats are available.
 - . A seat number is assigned automatically.
 - . A PNR is generated using date and counter.
3. Bookings are stored in a file.
4. For cancellation, the system matches PNR and marks it cancelled.
5. All bookings can be viewed at any time.

8.SAMPLE OUTPUT:

```
===== Train Ticket Reservation System =====
1. List Trains
2. Book Ticket
3. Cancel Ticket
4. View All Bookings
5. Exit
Select option: 1

Available Trains:
ID      Name           From -> To      Seats(available/total)
101    Coastal Express   Mumbai -> Goa  50/50
102    Mountain Flyer    Delhi -> Shimla 40/40
103    Desert Rider      Jaipur -> Jodhpur 60/60
104    Riverine Line     Kolkata -> Puri 55/55
105    Southern Star      Bangalore -> Chennai 45/45

===== Train Ticket Reservation System =====
1. List Trains
2. Book Ticket
3. Cancel Ticket
4. View All Bookings
5. Exit
Select option: 2

Available Trains:
ID      Name           From -> To      Seats(available/total)
101    Coastal Express   Mumbai -> Goa  50/50
102    Mountain Flyer    Delhi -> Shimla 40/40
103    Desert Rider      Jaipur -> Jodhpur 60/60
104    Riverine Line     Kolkata -> Puri 55/55
105    Southern Star      Bangalore -> Chennai 45/45

Enter train ID to book: 105
Enter passenger name: BISHT
Enter age: 33
Enter gender (M/F/Other): M

Ticket booked successfully!
PNR: 20251130-1001
Train: 105 (Southern Star)
Passenger: BISHT
Seat no: 1

===== Train Ticket Reservation System =====
1. List Trains
2. Book Ticket
3. Cancel Ticket
4. View All Bookings
5. Exit
Select option: 4
Select option: 4

All Bookings:
PNR      TrainID Seat  Name       Age  Gender Status
20251130-1000 101    1    shrey      21   m    Cancelled
20251130-1001 105    1    BISHT     33   M    Booked
```

9.CONCLUSION:

This project successfully implements a basic Train Ticket Reservation System using C programming. It demonstrates practical concepts like structures, arrays, file handling, logical decision making, and modular programming.

It provides a functional, user-friendly system that can be further expanded with additional features such as:

- . Multiple passengers per ticket
- . Train schedule search
- . Online payment simulation
- . Seat class system (AC / Sleeper)

10.FUTURE ENHANCEMENTS:

- . Adding user login system
- . Adding date-wise booking
- . Introducing multiple coaches and classes
- . Printing ticket as receipt
- . Graphical User Interface (GUI)

CODE:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#define MAX_TRAINS 5
#define MAX_SEATS 100
#define MAX_BOOKINGS 1000
#define NAME_LEN 50
#define PNR_LEN 20
#define FILENAME "bookings.dat"

typedef struct {
    int train_id;
    char name[50];
    char source[30];
    char dest[30];
    int total_seats;
    int seats_booked;
} Train;

typedef struct {
    char pnr[PNR_LEN];
    int train_id;
    char passenger_name[NAME_LEN];
    int age;
    char gender[10];
    int seat_no;
    int active;
} Booking;

Train trains[MAX_TRAINS];
Booking bookings[MAX_BOOKINGS];
int booking_count = 0;
```

```

void save_bookings_to_file() {
    FILE *fp = fopen(FILENAME, "wb");
    if (!fp) {
        perror("Unable to save bookings");
        return;
    }
    fwrite(&booking_count, sizeof(int), 1, fp);
    fwrite(bookings, sizeof(Booking), booking_count, fp);
    fclose(fp);
}

void load_bookings_from_file() {
    FILE *fp = fopen(FILENAME, "rb");
    if (!fp) {
        return;
    }
    int cnt = 0;
    if (fread(&cnt, sizeof(int), 1, fp) != 1) {
        fclose(fp); return;
    }
    if (cnt > MAX_BOOKINGS) cnt = MAX_BOOKINGS;
    if (fread(bookings, sizeof(Booking), cnt, fp) != (size_t)cnt) {
        /* partial read ok */
    }
    booking_count = cnt;

    for (int i = 0; i < MAX_TRAINS; ++i) trains[i].seats_booked = 0;
    for (int i = 0; i < booking_count; ++i) {
        if (bookings[i].active) {
            Train *t = find_train_by_id(bookings[i].train_id);
            if (t) t->seats_booked++;
        }
    }
    fclose(fp);
}

void clear_stdin() {
    int c;
    while ((c = getchar()) != '\n' && c != EOF) {}
}

void init_trains() {

trains[0].train_id = 101; strcpy(trains[0].name, "Coastal Express"); strcpy(trains[0].source, "Mumbai");
strcpy(trains[0].dest, "Goa"); trains[0].total_seats = 50; trains[0].seats_booked = 0;
trains[1].train_id = 102; strcpy(trains[1].name, "Mountain Flyer");
strcpy(trains[1].source, "Delhi"); strcpy(trains[1].dest, "Shimla"); trains[1].total_seats = 40; trains[1].seats_booked = 0;
trains[2].train_id = 103; strcpy(trains[2].name, "Desert Rider");
strcpy(trains[2].source, "Jaipur"); strcpy(trains[2].dest, "Jodhpur"); trains[2].total_seats = 60; trains[2].seats_booked = 0;
trains[3].train_id = 104; strcpy(trains[3].name, "Riverine Line");
strcpy(trains[3].source, "Kolkata"); strcpy(trains[3].dest, "Puri"); trains[3].total_seats = 55; trains[3].seats_booked = 0;
trains[4].train_id = 105; strcpy(trains[4].name, "Southern Star");
strcpy(trains[4].source, "Bangalore"); strcpy(trains[4].dest, "Chennai"); trains[4].total_seats = 45; trains[4].seats_booked = 0;
}

Train* find_train_by_id(int id) {
    for (int i = 0; i < MAX_TRAINS; ++i) {
        if (trains[i].train_id == id) return &trains[i];
    }
    return NULL;
}

void generate_pnr(char *buf, int bufsize) {
    static int counter = 1000;
    time_t t = time(NULL);
    struct tm *tm = localtime(&t);

    snprintf(buf, bufsize, "%04d%02d%02d-%04d",
    | | | tm->tm_year + 1900, tm->tm_mon + 1, tm->tm_mday, counter++);
}

```

```

void list_trains() {
    printf("\nAvailable Trains:\n");
    printf("ID\tName\tFrom -> To\tSeats(available/total)\n");
    printf("-----\n");
    for (int i = 0; i < MAX_TRAINS; ++i) {
        printf("%d\t%-18s\t%s -> %s\t%d/%d\n",
               trains[i].train_id,
               trains[i].name,
               trains[i].source,
               trains[i].dest,
               trains[i].total_seats - trains[i].seats_booked,
               trains[i].total_seats);
    }
    printf("\n");
}

void view_bookings() {
    if (booking_count == 0) {
        printf("\nNo bookings yet.\n");
        return;
    }
    printf("\nAll Bookings:\n");
    printf("PNR\tTrainID\tSeat\tName\tAge\tGender\tStatus\n");
    printf("-----\n");
    for (int i = 0; i < booking_count; ++i) {
        Booking *b = &bookings[i];
        printf("%s\t%d\t%d\t%-12s\t%d\t%-6s\t%s\n",
               b->pn,
               b->train_id,
               b->seat_no,
               b->passenger_name,
               b->age,
               b->gender,
               b->active ? "Booked" : "Cancelled");
    }
    printf("\n");
}

int find_free_seat_number(Train *t) {

    int used[MAX_SEATS + 1] = {0};
    for (int i = 0; i < booking_count; ++i) {
        if (bookings[i].active && bookings[i].train_id == t->train_id) {
            if (bookings[i].seat_no >= 1 && bookings[i].seat_no <= t->total_seats)
                used[bookings[i].seat_no] = 1;
        }
    }
    for (int s = 1; s <= t->total_seats; ++s) {
        if (!used[s]) return s;
    }
    return -1;
}

void book_ticket() {
    int train_id;
    char name[NAME_LEN];
    int age;
    char gender[10];

    list_trains();
    printf("Enter train ID to book: ");
    if (scanf("%d", &train_id) != 1) { clear_stdin(); printf("Invalid input.\n"); return; }
    clear_stdin();

    Train *t = find_train_by_id(train_id);
    if (!t) { printf("Train with ID %d not found.\n", train_id); return; }

    if (t->seats_booked >= t->total_seats) {
        printf("No seats available on this train.\n");
        return;
    }

    printf("Enter passenger name: ");
    if (!fgets(name, sizeof(name), stdin)) { printf("Input error\n"); return; }
    name[strcspn(name, "\n")] = 0;
}

```

```

if (strlen(name) == 0) {
    printf("Name cannot be empty.\n"); return;
}

printf("Enter age: ");
if (scanf("%d", &age) != 1) { clear_stdin(); printf("Invalid age.\n"); return; }
clear_stdin();

printf("Enter gender (M/F/Other): ");
if (!fgets(gender, sizeof(gender), stdin)) { printf("Input error\n"); return; }
gender[strcspn(gender, "\n")] = 0;

int seat_no = find_free_seat_number(t);
if (seat_no == -1) { printf("No free seat found (unexpected).\n"); return; }

if (booking_count >= MAX_BOOKINGS) {
    printf("Booking storage full.\n"); return;
}

Booking newb;
generate_pnr(newb.pnr, sizeof(newb.pnr));
newb.train_id = train_id;
strncpy(newb.passenger_name, name, NAME_LEN);
newb.age = age;
strncpy(newb.gender, gender, sizeof(newb.gender));
newb.seat_no = seat_no;
newb.active = 1;

bookings[booking_count++] = newb;
t->seats_booked++;

printf("\nTicket booked successfully!\n");
printf("PNR: %s\nTrain: %d (%s)\nPassenger: %s\nSeat no: %d\n\n",
       newb.pnr, t->train_id, t->name, newb.passenger_name, newb.seat_no);

save_bookings_to_file();
}

void cancel_ticket() {
char pnr[PNR_LEN];
printf("Enter PNR to cancel: ");
if (!fgets(pnr, sizeof(pnr), stdin)) { printf("Input error\n"); return; }
pnr[strcspn(pnr, "\n")] = 0;

for (int i = 0; i < booking_count; ++i) {
    if (strcmp(bookings[i].pnr, pnr) == 0) {
        if (!bookings[i].active) {
            printf("Booking already cancelled.\n");
            return;
        }
        bookings[i].active = 0;
        Train *t = find_train_by_id(bookings[i].train_id);
        if (t && t->seats_booked > 0) t->seats_booked--;
        printf("Booking with PNR %s cancelled successfully.\n", pnr);
        save_bookings_to_file();
        return;
    }
}
printf("PNR not found.\n");
}

void show_menu() {
printf("\n===== Train Ticket Reservation System =====\n");
printf("1. List Trains\n");
printf("2. Book Ticket\n");
printf("3. Cancel Ticket\n");
printf("4. View All Bookings\n");
printf("5. Exit\n");
printf("Select option: ");
}

int main() {
init_trains();
load_bookings_from_file();
}

```

```

int choice = 0;
while (1) {
    show_menu();
    if (scanf("%d", &choice) != 1) {
        clear_stdin();
        printf("Invalid choice.\n");
        continue;
    }
    clear_stdin();

    switch (choice) {
        case 1:
            list_trains();
            break;
        case 2:
            book_ticket();
            break;
        case 3:
            cancel_ticket();
            break;
        case 4:
            view_bookings();
            break;
        case 5:
            printf("Goodbye – saving bookings and exiting.\n");
            save_bookings_to_file();
            exit(0);
            break;
        default:
            printf("Invalid option. Try again.\n");
    }
}
return 0;
}

```

OUTPUT:

```

===== Train Ticket Reservation System =====
1. List Trains
2. Book Ticket
3. Cancel Ticket
4. View All Bookings
5. Exit
Select option: 1

Available Trains:
ID      Name                From -> To      Seats(available/total)
-----  

101     Coastal Express     Mumbai -> Goa   48/50
102     Mountain Flyer      Delhi -> Shimla 40/40
103     Desert Rider        Jaipur -> Jodhpur 60/60
104     Riverine Line        Kolkata -> Puri  55/55
105     Southern Star        Bangalore -> Chennai 45/45

```

===== Train Ticket Reservation System =====

1. List Trains
 2. Book Ticket
 3. Cancel Ticket
 4. View All Bookings
 5. Exit
- Select option: 2

Available Trains:

ID	Name	From -> To	Seats(available/total)
101	Coastal Express	Mumbai -> Goa	48/50
102	Mountain Flyer	Delhi -> Shimla	40/40
103	Desert Rider	Jaipur -> Jodhpur	60/60
104	Riverine Line	Kolkata -> Puri	55/55
105	Southern Star	Bangalore -> Chennai	45/45

Enter train ID to book: 101

Enter passenger name: SHREEAANSH SINGH BISHT

Enter age: 25

Enter gender (M/F/Other): M

Ticket booked successfully!

PNR: 20251130-1000

Train: 101 (Coastal Express)

Passenger: SHREEAANSH SINGH BISHT

Seat no: 3

===== Train Ticket Reservation System =====

1. List Trains
 2. Book Ticket
 3. Cancel Ticket
 4. View All Bookings
 5. Exit
- Select option: 4

All Bookings:

PNR	TrainID	Seat	Name	Age	Gender	Status
20251130-1000	101	1	shrey	21	m	Cancelled
20251130-1001	105	1	BISHT	33	M	Cancelled
20251130-1000	101	1	wrw	23	M	Booked
20251130-1001	101	2	SHREY BISHT	21	M	Booked
20251130-1000	101	3	SHREEAANSH SINGH BISHT	25	M	Booked

===== Train Ticket Reservation System =====

1. List Trains
2. Book Ticket
3. Cancel Ticket
4. View All Bookings
5. Exit

Select option: 3

Enter PNR to cancel: 20251130-1000

Booking already cancelled.

===== Train Ticket Reservation System =====

1. List Trains
2. Book Ticket
3. Cancel Ticket
4. View All Bookings
5. Exit

Select option: 5

Goodbye – saving bookings and exiting.

shreybisht@Shreys-MacBook-Air C %

