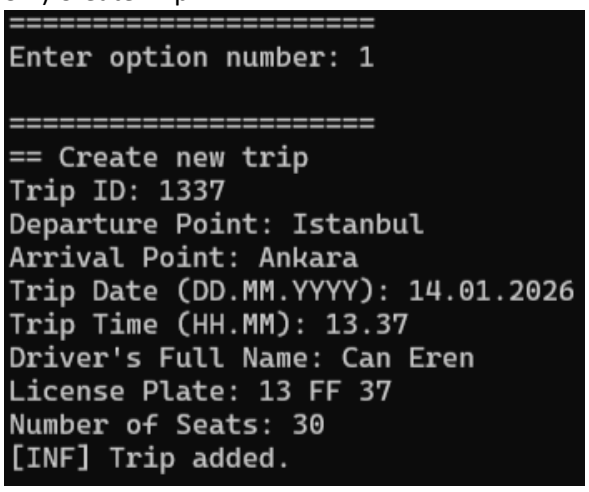
1. General Purpose of the Program
   "The Bus Trip", is a Command Line Interface (CLI) tool developed for small business bus firms. It digitializing and making easy and fast to trip administration and ticket sell. Main purpose's is digitalize manuel processes such as creating trip, selling and cancelling ticket. Target users are terminal users who don't need to use complex interfaces.

2. Program Flow
   Program serves a menu with infinite while loop to user. Depends on the choice the menu redirect to the related function. When a trip created, it saved as "trips.tbt" file. Updating and deleting trip functions performed with creating a "tmp.tbt" temporary file with filteration and then swapping names between trips.tbt and tmp.tbt. A trip can be created, listed, queried, updated and deleted. A ticket can be sold and cancelled.

3. Interface
   The program has a CLI interface and it designed for using in terminal. Used only basic symbols, row and columns used for creating a structred output. All interactions are done by entering number or strings. Every function and the menu has its own interface. The menu and functions has its own interface.

   3.1) The Bus Trip Menu

   3.2) Create Trip



3.3) List Trips

```
====================
Enter option number: 2

====================
== List trips
Trip ID  Departure Point    Arrival Point   Trip Date    Trip Time    Driver Full Name   Licence Plate   Total Seats  Sold Tickets
1337     Istanbul           Ankara          14.01.2026   13.37        Can Eren           13 FF 37        30           0
[INF] Trips listed.
```

3.4) Query Trip

```
====================
Enter option number: 3

====================
== Query trip
Enter ID: 1337
Trip ID  Departure Point    Arrival Point   Trip Date    Trip Time    Driver Full Name   Licence Plate   Total Seats  Sold Tickets
1337     Istanbul           Ankara          14.01.2026   13.37        Can Eren           13 FF 37        30           0
```

3.5) Update Trip

```
======================
Enter option number: 4

======================
== Update trip
Enter ID: 1337
New Departure Point: Bitlis
New Arrival Point: Muğla
New Trip Date (DD.MM.YYYY): 14.01.2026
New Trip Time (HH.MM): 13.30
New Driver's Full Name: Şahin Demir
New Plate Licence: 13 FF 37
New Number of Seats: 30
[INF] Trip updated.
```

3.6) Delete Trip

```
======================
Enter option number: 5

======================
== Delete trip
Enter ID: 1337
[INF] ID 1337 found. It will be deleted.
```

3.7) Sell Ticket

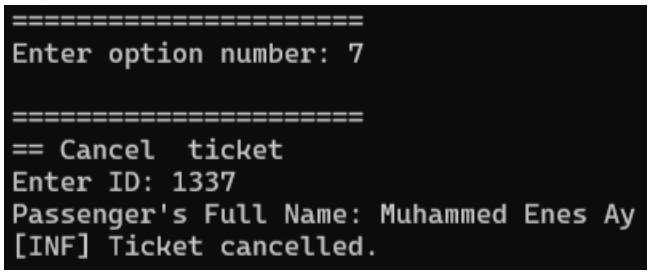
```
======================
Enter option number: 6

======================
== Sell ticket
Enter ID: 1337
Passenger's Full Name: Muhammed Enes Ay
Passenger's Citizen ID: 13371337133
[INF] Ticket sold. Receipt created.
```

```
======================
=====THE BUS TRIP=====
========RECEIPT=======
Trip ID: 1337
Passenger's Full Name: Muhammed Enes Ay
Passenger's Citizen ID: 13371337133
======================
```

receipt_1337_Muhammed Enes Ay.txt 25.12.2025 20:18 Metin Belgesi 1 KB

3.8) Cancel Ticket



Related receipt file removed from directory.

4. Data Structures
   The fundamental data structure is “struct Trip” block. All the properties of the trip including trip ID, departure point, arrival point, trip date, trip time, driver’s full name, licence plate, number of seat and number of sold seat are stored in “Trip” struct.

5. Functions
   create_trip, creates a struct and trips.tbt file with given struct input.
   list_trip, lists all the trips with formatted string from trips.tbt.
   query_trip, queries a trip with given trip ID input.
   update_trip, updates struct and trips.tbt file with given struct input.
   delete_trip, deletes a trip with given trip ID input.
   sell_ticket, sells a ticket, updates number of sold seat in trips.tbt and creates a receipt file.
   cancel_ticket, cancels tickets and updates number of sold seat in trips.tbt
   create_receipt, creates a receipts with given trip ID, passenger name and citizen ID input.
   check_ID, checks if given trip ID exists in trips.tbt.
   print_banner, prints menu and redirects to related function with given number input.

6. File Operations
   Data’s processed in binary mode with wb, ab, rb. Struct’s are directly writed to the trips.tbt and tmp.tbt files. Receipt is in readable text format.

7. Error Handling
   File opening handling performed with NULL check. Handled ID collision with check_ID. Detailed error handling messages throwing with [ERR] tag. The user informed with [INF] tag in some operations.

8. Dependencies
   Written using only standart libraries: stdio.h, for input/output operations, stdlib.h for running OS operations and string.h for string processing. It can be run on any operating system.

9. Security
   Prevented buffer overflow with input limitation in scanf.
   Prevented resource leak by correctly implementing the fclose function.

10. Conclusion
    “The Bus Ticket”, is a functional prototype that performs basic Create, Read, Update, Delete (CRUD) operations.
    Proper for basic bus and ticket register needs.