UNIVERSITY OF PETROLEUM AND ENERGY STUDEIES

PROJECT REPORT

FOR THE

MAJOR PROJECT

(TOPIC : VEHICLE TOLL CONTROL SYSTEM)

USING

PROGRAMMING IN C (COURSE CODE : CSEG1032)

SESSION : 2025-2026 / 1ST SEMESTER

SUBMITTED TO :	SUBMITTED BY :
DR. TANU SINGH	NAME : TARUN MITTAL
	SAP ID : 590024396
	BATCH NO .: B37

ABSTRACT :

The VEHICLE TOLL CONTROL SYSTEM is a console based program written in C programming language to control all the management of the toll booth .

The system helps to record the history of the vehicles passing through the toll booth on the basis of date of the day .

The system also helps to calculate the amount collected at the toll booth through the different modes of the payments i.e. cash , online , fastag , toll pass , for the complete day . On the basis of the date and print the summary of the amount collected in the receipt format .

The main aim of the project is to make the collection easier , faster and more organised compared to doing everything manually .

PROBLEM DEFINATION :

Since manual toll management is slow , error-prone , and difficult to maintain the history record . Common issues include :

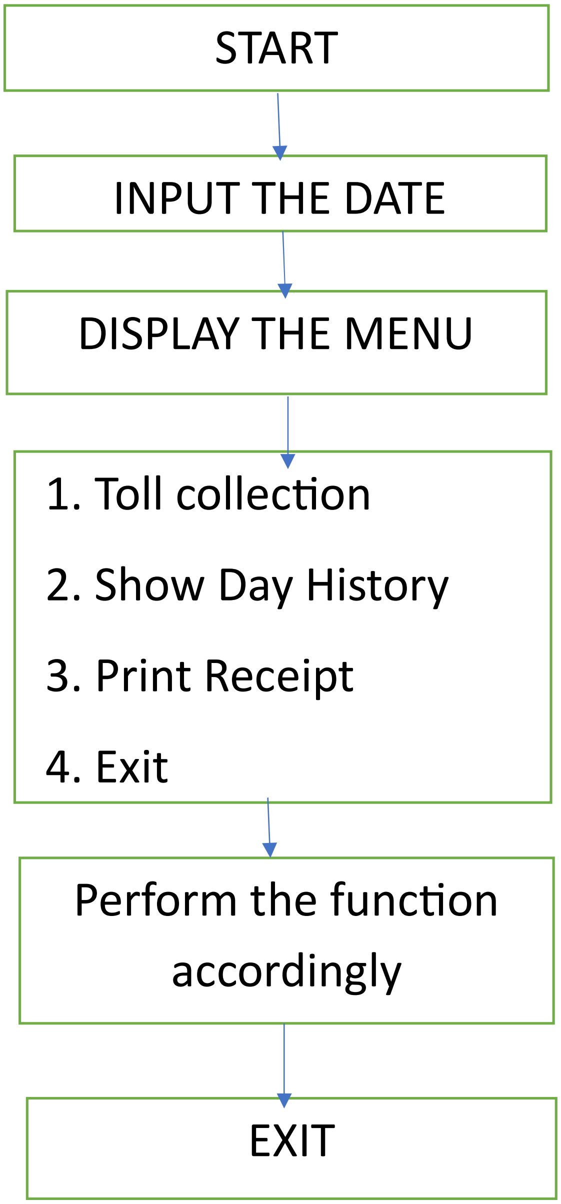
- Difficulty in generating the daily report .
- Manual calculation of the amount can be difficult .
- Lack of automated receipt generation .

Objective:
To develop a C program that automates toll collection with the features:

1. Add new vehicle toll entry and amount collection .
2. Retrieve daily toll history by date .
3. Generate toll receipt providing the summary for any selected date .
4. Store all the details permanently in a file using file handling.

SYSTEM DESIGN :

FLOWCHART :



ALGORITHM :

1 START

2 INPUT THE DATE FROM THE USER

3 DISPLAY THE MAIN MENU ON THE TERMINAL

Toll collection for vehicle , show day history , print receipt for the amount summary according to the date , exit

4 PERFORM THE TASK SELECTED BY THE USER UNTIL THE USER SELECTS **4.EXIT**

5 STOP

Algorithm 1: Toll collection for vehicle

1. Input vehicle number
2. Date to be filled automatically as inputed by the user in the starting
3. Enter the toll amount to be paid
4. Enter the mode of payment
5. Open the file "toll_records.txt" in append mode.
6. Store date | vehicle number | toll amount | mode of payment
7. Close the file

Algorithm 2: Search History by Date

1. Input date from user
2. Open "toll_records.txt" in read mode
3. Read each record until the end of the file
4. Compare stored date with user input date
5. Print all matching records
6. Display the number of vehicles passed that day with the amount paid by them
7. Close file

Algorithm 3: Print Receipt

1. Input date from the user
2. Open "toll_records.txt" in read mode
3. Read each record until the end of the file
4. Compare stored date with user input date
5. Read all matching records
6. Calculate the amount paid by the different modes of payment separately
7. Calculate the total amount collected that day
8. Close the file "toll_records.txt"
9. Display the amount collected by the different modes of payment separately and the total amount collected

IMPLEMENTAQTION DETAILS:

FILE STRUCTURE :

```
/src
    main.c
    tollsystem.c
    utils.c
/include
    tollsystem.h
    utils.h
/docs
    ProjectReport.txt
```

STRUCTURE USED TO STORE TOLL REOCRD

```
struct VehicleRecord {
char date[20];
char vehicleNumber[20];
float amount;
char paymentMode[20];
};
```

FUNCTION TO ENTER DATE

```
void inputDate(char *date)
{
printf("Enter Date in the format (DD-MM-YYYY): ");
scanf("%s", date);
}
```

TOLL COLLECTION FUNCTION

```
void TollCollection(char *date)
{
struct VehicleRecord record;
strcpy(record.date,date);
printf("\n--- Toll Deduction ---\n");
printf("Enter Vehicle Number: ");
scanf("%s",record.vehicleNumber);
printf("Enter Amount: ");
scanf("%f",&record.amount);
int modeChoice;
printf("Select Mode of Payment:\n");
printf("1. Cash\n2. Online\n3. Fastag\n4. Valid Toll Pass\n");
printf("Enter choice: ");
scanf("%d",&modeChoice);
switch(modeChoice)
{
case 1: strcpy(record.paymentMode, "Cash"); break;
case 2: strcpy(record.paymentMode, "Online"); break;
case 3: strcpy(record.paymentMode, "Fastag"); break;
case 4: strcpy(record.paymentMode, "TollPass"); break;
default: strcpy(record.paymentMode, "Cash"); break;
}

FILE *details = fopen("toll_records.txt","a");
if(details == NULL)
{
printf("Error opening file!\n");
return;
}
fprintf(details,"%s | %s | %.2f | %s\n" ,record.date
,record.vehicleNumber,record.amount,record.paymentMode);
fclose(details);
printf("Toll Deduction Recorded Successfully!\n");
}
```

FUNTION TO SHOW DAILY VEHICLE HISTORY

```
void ShowHistory()
{
char searchDate[20];
printf("\nEnter date to view history (DD-MM-YYYY): ");
scanf("%s",searchDate);
FILE *details = fopen("toll_records.txt","r");
if(details == NULL)
{
printf("No records found!\n");
return;
}

char date[20],vehicle[20],mode[20];
float amount;
int found = 0;

printf("\n--- Toll History for %s ---\n",searchDate);
while(fscanf(details, "%s | %s | %f | %s",date,vehicle,&amount,mode)
!= EOF)
{
if(strcmp(date,searchDate) == 0)
{
found = 1;
printf("Vehicle: %s | Amount: %.2f | Mode:
%s\n",vehicle,amount,mode);
}
}

if(!found)
printf("No records found for this date.\n");

fclose(details);
}
```

FUNTION FOR DAILY RECIEPT SUMMARY

```
void dailyReceipt()
{
char searchDate[20];
printf("\nEnter date for receipt (DD-MM-YYYY): ");
scanf("%s",searchDate);

FILE *details = fopen("toll_records.txt","r");
if(details == NULL)
{
```

```
printf("No records found!\n");
return;
}

float cash = 0, online = 0, fastag = 0, pass = 0, amount;
char date[20], vehicle[20], mode[20];
int found = 0;

while(fscanf(details,"%s | %s | %f | %s",date,vehicle,&amount,mode) != EOF)
{
if(strcmp(date,searchDate) == 0)
{
found = 1;
if(strcmp(mode,"Cash") == 0)
cash += amount;
else if(strcmp(mode,"Online") == 0)
online += amount;
else if(strcmp(mode,"Fastag") == 0)
fastag += amount;
else if(strcmp(mode,"TollPass") == 0)
pass += amount;
}
}

fclose(details);

if(!found)
{
printf("No data found for this date.\n");
return;
}

float total = cash + online + fastag + pass;
printf("\n===== Toll Receipt for %s =====\n", searchDate);
printf("Cash Collection      : %.2f\n", cash);
printf("Online Collection    : %.2f\n", online);
printf("Fastag Collection    : %.2f\n", fastag);
printf("Toll Pass Collection : %.2f\n", pass);
printf("----------------------------------\n");
printf("TOTAL COLLECTION     : %.2f\n", total);
printf("==================================\n");
}
```

TESTING AND RESULT :

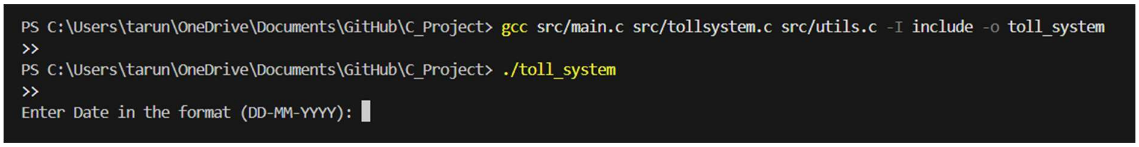
COMMAND TO RUN THE CODE ON THE TERMINAL

STEP 1 . Compilation command

```
gcc src/main.c src/tollsystem.c src/utils.c -I include -o toll_system
```

STEP 2 . Run the Program

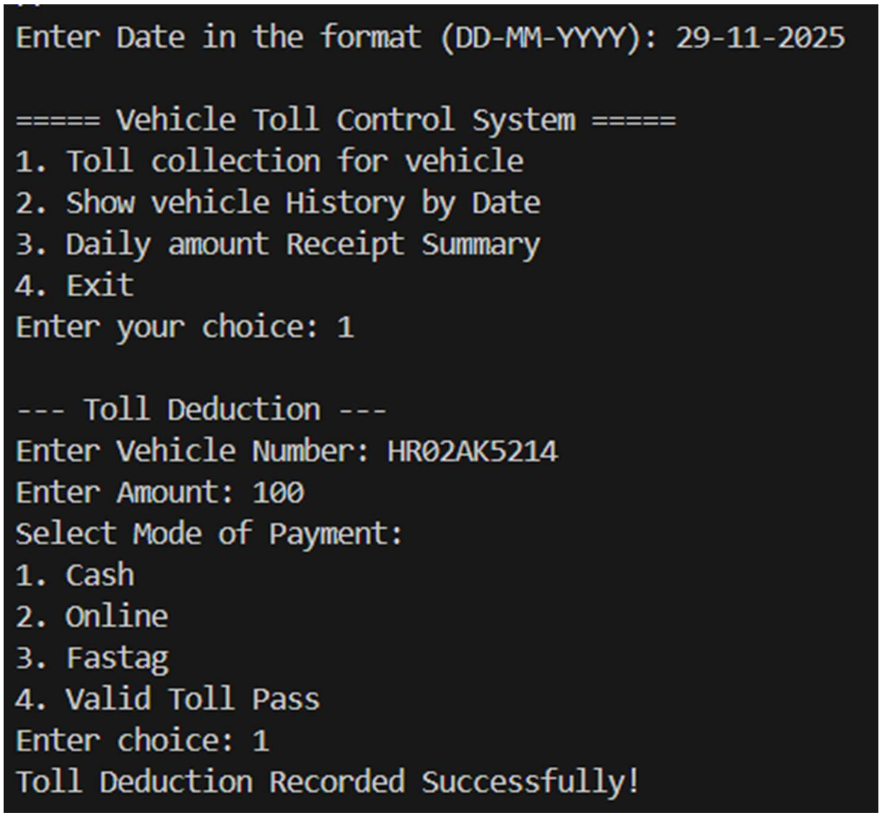
```
./toll_system
```



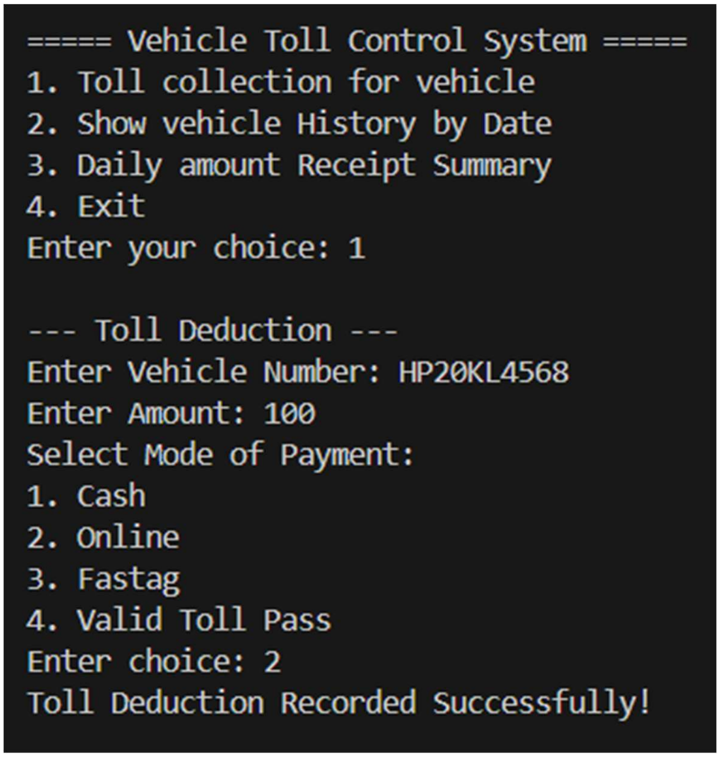
```
PS C:\Users\tarun\OneDrive\Documents\GitHub\C_Project> gcc src/main.c src/tollsystem.c src/utils.c -I include -o toll_system
>>
PS C:\Users\tarun\OneDrive\Documents\GitHub\C_Project> ./toll_system
>>
Enter Date in the format (DD-MM-YYYY):
```

Now check functions one by one

1. Execute the first option and add some vehicle toll collection details

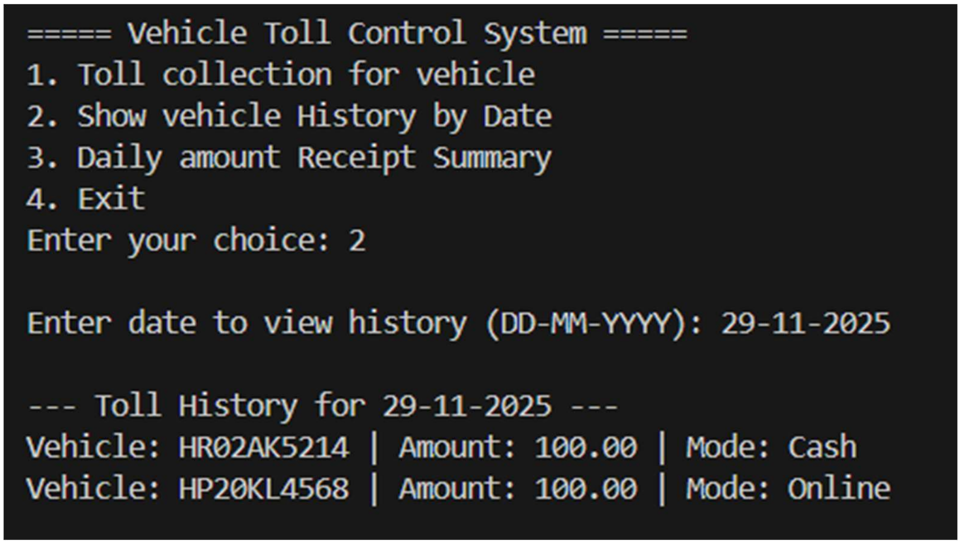


```
Enter Date in the format (DD-MM-YYYY): 29-11-2025

===== Vehicle Toll Control System =====
1. Toll collection for vehicle
2. Show vehicle History by Date
3. Daily amount Receipt Summary
4. Exit
Enter your choice: 1

--- Toll Deduction ---
Enter Vehicle Number: HR02AK5214
Enter Amount: 100
Select Mode of Payment:
1. Cash
2. Online
3. Fastag
4. Valid Toll Pass
Enter choice: 1
Toll Deduction Recorded Successfully!
```



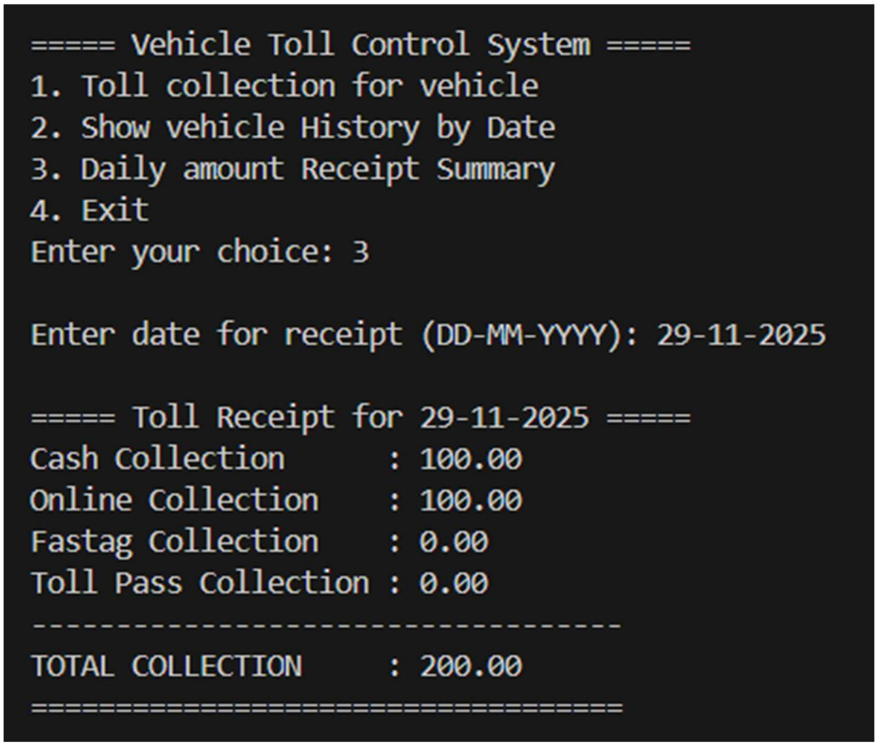
```
===== Vehicle Toll Control System =====
1. Toll collection for vehicle
2. Show vehicle History by Date
3. Daily amount Receipt Summary
4. Exit
Enter your choice: 1

--- Toll Deduction ---
Enter Vehicle Number: HP20KL4568
Enter Amount: 100
Select Mode of Payment:
1. Cash
2. Online
3. Fastag
4. Valid Toll Pass
Enter choice: 2
Toll Deduction Recorded Successfully!
```

2. Execute the second option and enter the date



```
===== Vehicle Toll Control System =====
1. Toll collection for vehicle
2. Show vehicle History by Date
3. Daily amount Receipt Summary
4. Exit
Enter your choice: 2

Enter date to view history (DD-MM-YYYY): 29-11-2025

--- Toll History for 29-11-2025 ---
Vehicle: HR02AK5214 | Amount: 100.00 | Mode: Cash
Vehicle: HP20KL4568 | Amount: 100.00 | Mode: Online
```

3. Execute the third option and enter the date



```
===== Vehicle Toll Control System =====
1. Toll collection for vehicle
2. Show vehicle History by Date
3. Daily amount Receipt Summary
4. Exit
Enter your choice: 3

Enter date for receipt (DD-MM-YYYY): 29-11-2025

===== Toll Receipt for 29-11-2025 =====
Cash Collection      : 100.00
Online Collection    : 100.00
Fastag Collection    : 0.00
Toll Pass Collection : 0.00
-----------------------------------
TOTAL COLLECTION     : 200.00
===================================
```

RESULT :

All features of the Toll Control System worked successfully:

- **Toll entries were stored correctly.**
- **Searching by date showed accurate results.**
- **Daily receipt summary calculated correct totals.**
- **No crashes or wrong outputs were observed during testing.**

The system is fully functional for basic toll management operations.

CONCLUSION :

- The Toll Control System developed in C successfully automates the basic operations of a toll booth. It records vehicle details, stores the data safely in a text file, displays the complete history of a particular day, and generates a receipt showing the total collection.

- The project reduces manual work, avoids calculation mistakes, and ensures that every toll entry is stored in an organized way. Overall, the system meets its main objective of making toll management simple, accurate, and easy to use.

FUTURE WORK :

1. **Implement a graphical user interface (GUI)**
2. **Automatic toll calculation**
3. **Integration with FASTag system**
   The system can be connected with RFID/FASTag scanners for real-time automatic toll detection.
4. **Data analytics and reports**
   Features like hourly traffic count, busiest days, and monthly revenue charts can be added.

REFERENCES :

Let Us C – Yashavant Kanetkar

Class notes by DR. TANU SINGH