



Visvesvaraya Technological University
BELAGAVI, KARNATAKA

ವಿಶ್ವವರಾಹ ತಾಂತ್ರಿಕ ವಿಶ್ವವಿದ್ಯಾಲಯ
ಬೆಳಗಾವಿ, ಕರ್ನಾಟಕ

Report on File Structure Mini Project
**“VEHICLE PARKING MANAGEMENT
SYSTEM USING VARIABLE LENGTH
RECORD”**

for the course

FILE STRUCTURES LABORATORY WITH MINI PROJECT (18ISL67)

Submitted by

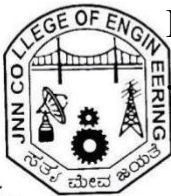
1. MEGHANA BEKAL - 4JN18IS044

2. MEGHANA M - 4JN18IS045

Under the guidance of

Mr. SHARATH KUMAR S R. B.E., M. Tech.

Asst. Professor, Dept. of IS&E JNNCE, Shivamogga.



Department of Information Science & Engineering
Jawaharlal Nehru National College of Engineering
Shivamogga - 577 2042020-21

National Education Society ®



JAWAHARLAL NEHRU NATIONAL COLLEGE OF
ENGINEERING SHIVAMOGGA- 577204.
DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

CERTIFICATE

This is to certify that Project entitled

**“VEHICLE PARKING MANAGEMENT SYSTEM
USING VARIABLE LENGTH RECORD”**

Submitted by

1. MEGHANA BEKAL - 4JN18IS044

2. MEGHANA M - 4JN18IS045

**for the course FILE STRUCTURE LABORATORY WITH MINI PROJECT
(18ISL67)**, students of 6th semester B.E. ISE, in partial fulfillment of the requirement
for the award of degree of Bachelor of Engineering in Information Science and
Engineering of Visvesvaraya Technological University, Belagavi during the year
2020-21..

Signature of Guide

Mr. SHARATH KUMAR SR B.E., M.Tech.

**Assistant Professor,
Dept. of IS&E,
JNNCE, Shivamogga**

Signature of HOD

Dr. R Sanjeev Kunte B.E., M. Tech., Ph. D

**Professor & Head,
Dept. of IS&E,
JNNCE, Shivamogga**

Examiners 1: _____

Examiners 2: _____

ACKNOWLEDGEMENT

On presenting the File structure Mini - Project report on "**Vehicle Parking Management using variable length record**", We feel great to express our humble feelings of thanks to all those who have helped us directly or indirectly in the successful completion of the project work.

We would like to thank our respected guide **Mr.Sharath kumar.S.R** B.E., M.Tech., Dept. of IS& E, who has helped us a lot in completing this task, for his continuous encouragement and guidance throughout the project work.

We would like to thank **Dr. Sanjeev Kunte R** M. Tech., Ph.D., Professor and Head, Dept of IS & E, JNNCE, Shivamogga and **Dr. Manjunatha P**, The Principal, JNNCE, Shivamogga for all their support and encouragement.

We are grateful to **Department of Information Science and Engineering** and our institution **Jawaharlal Nehru National College of Engineering** and for imparting us the knowledge with which we can do our best.

Finally, we would like to thank the whole teaching and non-teaching staff of Information Science and Engineering Dept.

Thanking you all,

MEGHANA BEKAL (4JN18IS44)

MEGHANA M (4JN18IS045)

ABSTRACT

The number of vehicles in cities has increased dramatically due to rapid economic development. However, the infrastructure for accommodating these vehicles has grown relatively slow. Alleviating the pressure on the urban transport system and solving the ‘parking difficulty’ problem have thus become hot topics recently. This project helps to solve parking difficulty.

Online Vehicle parking management system is an application that allows to park vehicles without difficulty.. The parking system is developed using variable length record.

TABLE OF CONTENTS

Contents	Page No
Chapter 1. Introduction	01-03
Introduction to File Structures	
Variable length record	
Chapter 2. Algorithm	04-05
Insert function	
Search function	
Delete function	
Display function	
Chapter 3. Implementation	06
Problem Statement	
Objectives of the project	
About the programming language.	
Chapter 4. Snapshots	07-11
Chapter 5. Conclusion	12
Chapter 6. References	13

INTRODUCION

Introduction to File Structures

File Structures is the Organization of Data in Secondary Storage Device in Such a way that minimizes the access time and the storage space.

File Structure: It is a combination of

- Representation for data in files and
- The operations for accessing the data

It allows applications to read write and modify data search .An improvement in file Structure design make a application hundred times faster .To access the data faster From the storage disk we use file structure .Disks are slow which are used to pack Thousands of megabytes of data.

Records and its types:

Records are collection of fields, possibly of different Data types, typically in fixed number of sequence.

The fields of is also be called members. For example, a date could be sorted as a record containing a numeric year field, a Month field represented as a string, and a numeric day of month field. Records are distinguished from arrays by the fact that the number of fields is typically fixed, each field has a name, and that each field may have a different types.

A record type is a data type that describes such values and variables. The Definition includes specifying the data type of each field and an identifier by which it can be accessed.

Why we need File Structure??

As we know without proper structure of organizing is ordinary system, it will generate some types of problems. So avoiding this kind of problems we go for File Structure.

The File structure means it's tell about how the system will stored and access the record from memory and also its tell how the disk are performed its tasks and also how to speed up the execution of transferring data.

Data processing from a computer science prospective:

- storage of data
- organization of data
- access to data

This will be built on your on your knowledge of Data Structures. Some of the important concepts used

Field Structures:

1. Fixed Length Fields: The method to organize fields is by limiting the maximum size of each field. The advantage in this method is that since the size of each field is fixed, the entire field can be read at once.
2. Length Indicator Fields: The length of each field is specified as a prefix to actual data.
3. Delimited Fields: Any special character which is not a part of actual data can be used as separator.

Record Structures:

1. Fixed Length Record Structure: Each record is stored in fixed size. The size can be determined by adding the maximum space occupied by each field and some space reserved for the header data.
2. Fixed Field Count: The number of the fields in each record is fixed.
3. Index Structure for Records: An index is a collection of key field and reference field.
4. Key Field: Key Field is a member of record which can uniquely identify the record.
5. Reference Field: Reference Field contains the value that points to the address of the corresponding record in the file.

A file can contain:

1. Fixed-length records - all the records are exactly the same length
2. Variable-length records - the length of each record varies.

Using variable-length records might enable you to save disk space. When you use fixed-length records, you need to make the record length equal to the length of the longest record. If your application generates many short records with occasional long ones, using fixed-length records wastes a lot of disk space, so variable-length records would be a better choice.

CHAPTER 3

ALGORITHM

INSERT FUNCTION:

1. Declare variables
2. Read the details with Id
3. Open the text file in input mode using fstream
4. If file does not exist, open the text file in output and append mode, write the details into the file and close the file
5. Else fetch details from the file using temporary variables
6. While not the end of file
 - a. if the inputted id matches with the fetched id
 - b. close the file output a display message that id already exist
 - c. goto the beginning of insert function
7. close the file
8. Open the text file in output and append mode to write the new details
9. close the file

SEARCH FUNCTION:

1. Declare variables
2. Read the id to search
3. Open the text file in input mode using fstream
4. If file does not exist, call artwork function
5. Else fetch the details from the file
6. while not the end of file
 - a. if the inputted id matches with the fetched Id
 - b. display the details of the inputted Id and increment the found variable
7. close the file
8. if found equal to 0
9. display a output message that artwork Id not found

DELETE FUNCTION:

1. Declare variables
2. Read the Id to delete
3. Open the text file in input mode using fstream
4. If file does not exist call artwork function
5. Else fetch the details from the file
6. While not the end of file
 - a. If the inputted Id matches with the fetched Id
 - b. Display the message stating Record deleted successfully and increment the found variable
 - c. Else append the details into another file by opening that file in out and append mode.
7. close 2 files
8. Remove the old file
9. Rename the new file to old file name.
10. If found equals to 0
11. Display a output message stating artwork not found.

DISPLAY FUNCTION:

1. Declare variables
2. Open txt file in input mode using fstream
3. if file does not exist, calls artwork function
4. Else fetch the details from the file
5. while not the end of file
 - a. display the details from the fetched file
 - b. then increment the found variable
6. close the file
7. if found equals to 0
8. display a output message stating records not found

CHAPTER 3

IMPLEMENTATION

Problem statement

The Proposed Vehicle parking management is the completely computerized. In proposed system there is no need to maintain data about vehicle number. To maintains the record of vehicles and their check in and checkout time . The objective of developing such computerized system is to reduce the paper work and safe of time, thereby increasing the efficiency and decreasing the work load.

Objectives of the project

The main objective of the Parking System is **to manage the details of Duration, Vehicles, Parking Slots, Customers, Parking Fees.** ... The purpose of the project is to build an application program to reduce the manual work for managing the Duration, Vehicles, Types, Parking Slots.

About the Programming Language C++

In the program “Time Complexity of Search and Traverse of a Key in primary indexing”, we have used C++ as a medium to compile and run the program because of its versatile uses in the industry.

It was designed with a bias toward system programming and embedded, resource-constrained and large systems, with performance, efficiency and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, servers (e.g. e-commerce, web search or SQL servers), and performance-critical applications. C++ is a compiled language, with implementations of it available on many platforms. Many vendors provide C++ compilers, including the Free Software Foundation, Microsoft, Intel, and IBM.

CHAPTER 4

SNAPSHOTS

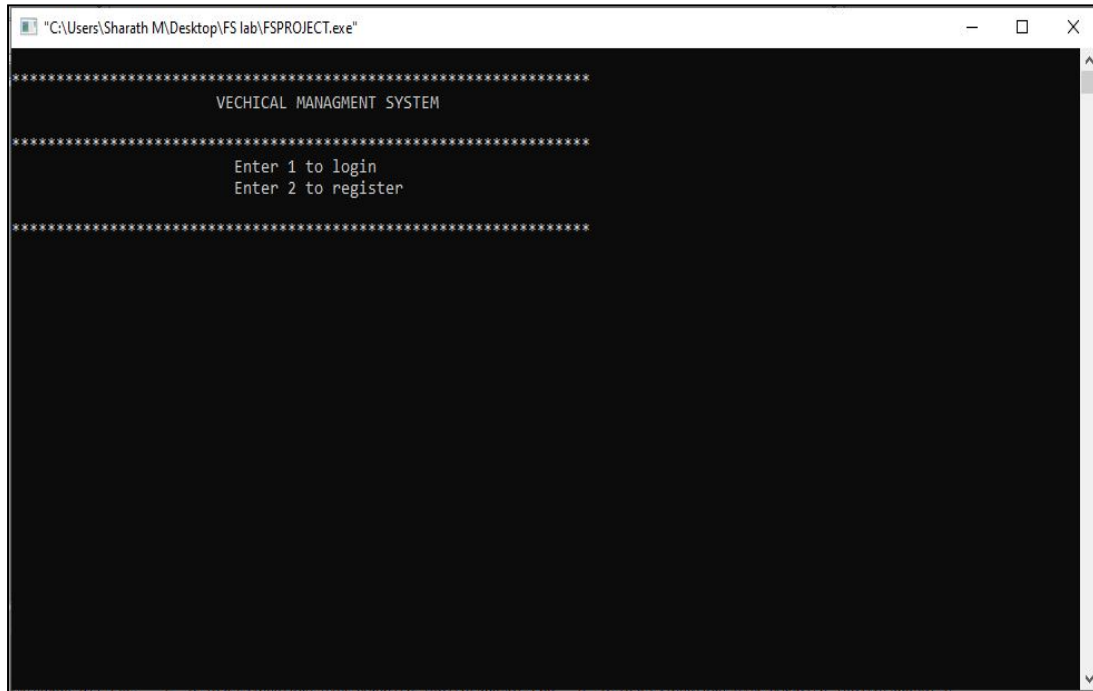


Fig. 4.1 Main panel

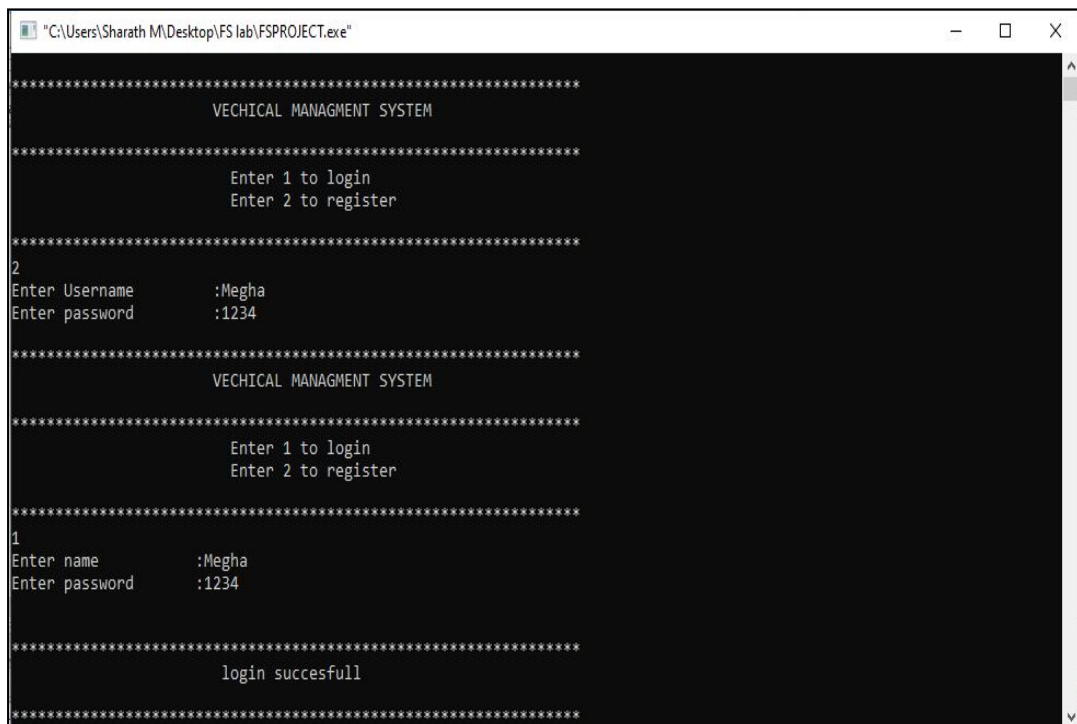


Fig. 4.2 Login and Registration window

```
"C:\Users\Sharath M\Desktop\FSPROJECT.exe"
*****
TOTAL VEHICLES PARKED      :      0
TOTAL VEHICLES CAN BE PARKED:    50
*****

1:CHECK-IN
2:CHECK-OUT
3:VIEW ALL AMOUNT
4:VIEW ALL VEHICAL DEPARTURE TIME
*****
```

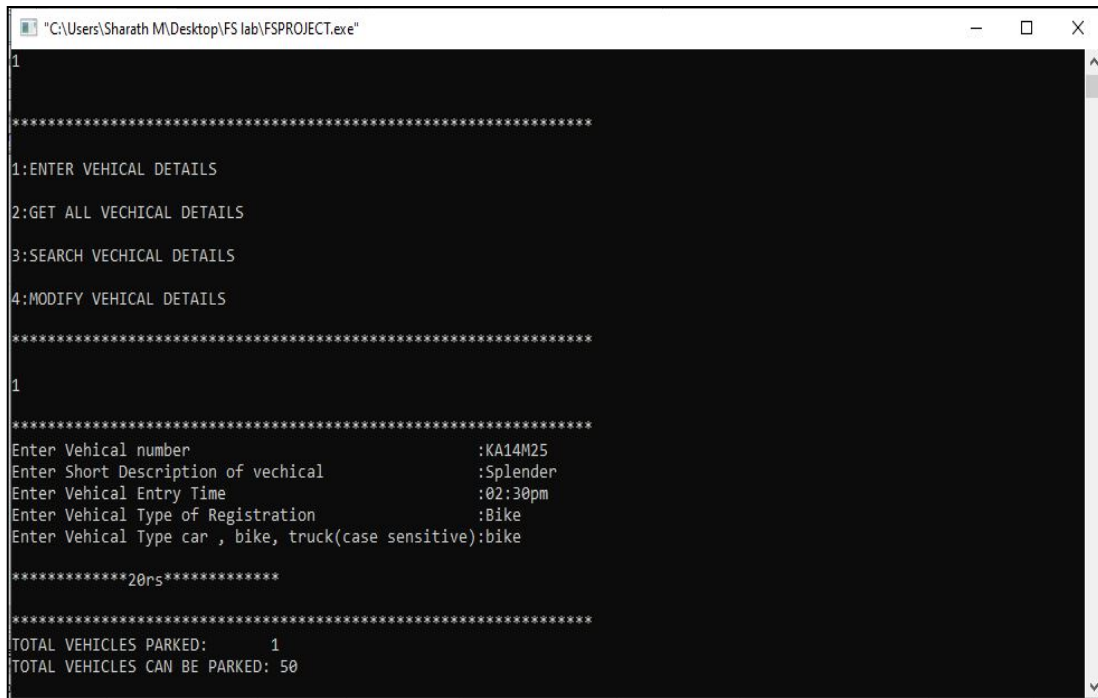
Fig 4.3 Initial status of the parking slot

```
"C:\Users\Sharath M\Desktop\FSPROJECT.exe"
*****

1:CHECK-IN
2:CHECK-OUT
3:VIEW ALL AMOUNT
4:VIEW ALL VEHICAL DEPARTURE TIME
*****
1
*****

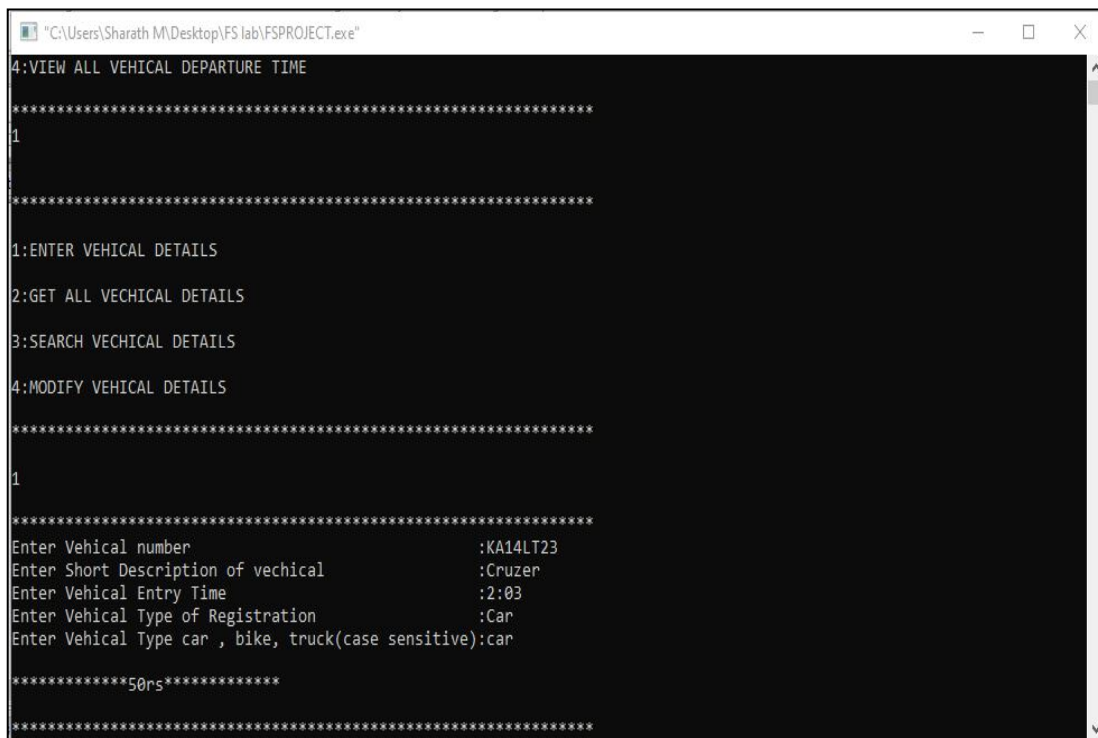
1:ENTER VEHICAL DETAILS
2:GET ALL VEHICAL DETAILS
3:SEARCH VEHICAL DETAILS
4:MODIFY VEHICAL DETAILS
*****
```

Fig 4.4 Attributes of the check in



```
"C:\Users\Sharath M\Desktop\FS lab\FSPROJECT.exe"
1
*****
1:ENTER VEHICAL DETAILS
2:GET ALL VECHICAL DETAILS
3:SEARCH VECHICAL DETAILS
4:MODIFY VEHICAL DETAILS
*****
1
*****
Enter Vehical number           :KA14M25
Enter Short Description of vehical :Splender
Enter Vehical Entry Time       :02:30pm
Enter Vehical Type of Registration :Bike
Enter Vehical Type car , bike, truck(case sensitive):bike
*****20rs*****
*****
TOTAL VEHICLES PARKED:      1
TOTAL VEHICLES CAN BE PARKED: 50
```

Fig 4.5 Check in details (bike)



```
"C:\Users\Sharath M\Desktop\FS lab\FSPROJECT.exe"
4:VIEW ALL VEHICAL DEPARTURE TIME
*****
1
*****
1:ENTER VEHICAL DETAILS
2:GET ALL VECHICAL DETAILS
3:SEARCH VECHICAL DETAILS
4:MODIFY VEHICAL DETAILS
*****
1
*****
Enter Vehical number           :KA14LT23
Enter Short Description of vehical :Cruzer
Enter Vehical Entry Time       :2:03
Enter Vehical Type of Registration :Car
Enter Vehical Type car , bike, truck(case sensitive):car
*****50rs*****
*****
```

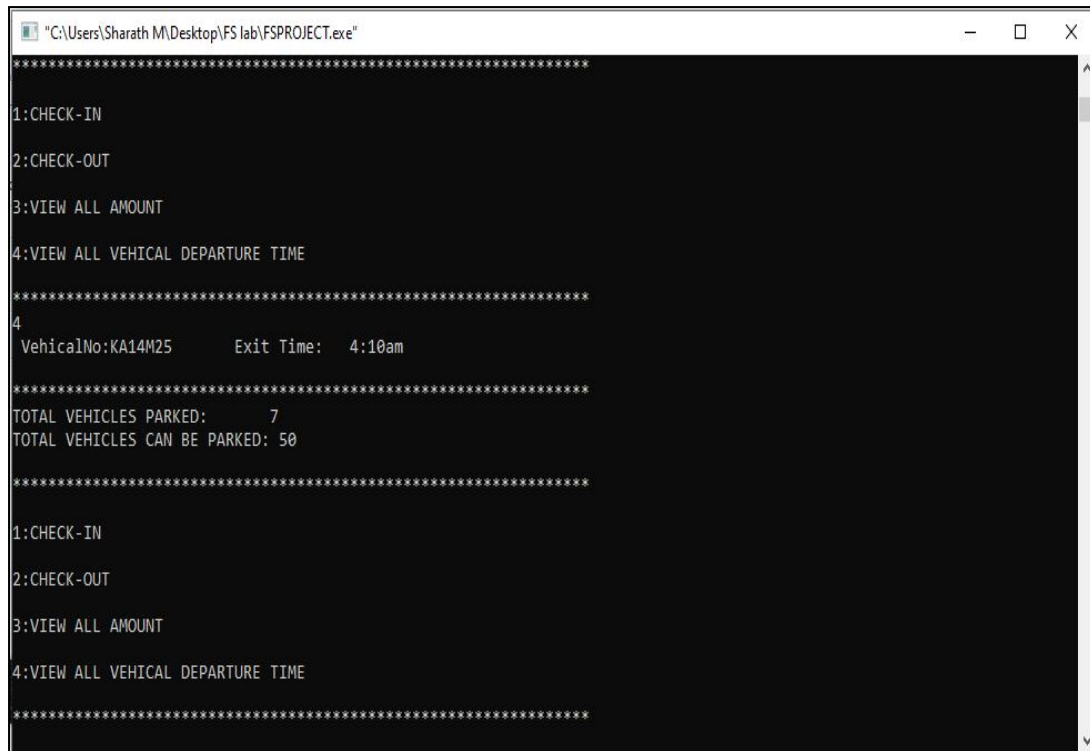
Fig 4.6 Check in details (car)

```
*****
1:CHECK-IN
2:CHECK-OUT
3:VIEW ALL AMOUNT
4:VIEW ALL VEHICAL DEPARTURE TIME
*****
2
1:ENTER VEHICAL NUMBER
KA14M25
Record is KA14M25|Splender|02:30pm|Bike|bike|$
Record Found at Position :6
Enter Vehical Exit Time
4:10am
vechicle checked out
```

Fig 4.7 Checkout details

```
"C:\Users\Sharath M\Desktop\FS lab\FSPROJECT.exe"
*****
1:CHECK-IN
2:CHECK-OUT
3:VIEW ALL AMOUNT
4:VIEW ALL VEHICAL DEPARTURE TIME
*****
3
VehicalNo:KA14M25      Cost:*****20rs*****
VehicalNo:KA15N02      Cost:*****20rs*****
VehicalNo:KA14LT23     Cost:*****50rs*****
*****
TOTAL VEHICLES PARKED      :      3
TOTAL VEHICLES CAN BE PARKED:    50
*****
1:CHECK-IN
2:CHECK-OUT
3:VIEW ALL AMOUNT
4:VIEW ALL VEHICAL DEPARTURE TIME
```

Fig 4.8 Amount details



```
"C:\Users\Sharath M\Desktop\FS lab\FSPROJECT.exe"
*****
1:CHECK-IN
2:CHECK-OUT
3:VIEW ALL AMOUNT
4:VIEW ALL VEHICAL DEPARTURE TIME
*****
4
VehicalNo:KA14M25      Exit Time: 4:10am
*****
TOTAL VEHICLES PARKED:      7
TOTAL VEHICLES CAN BE PARKED: 50
*****
1:CHECK-IN
2:CHECK-OUT
3:VIEW ALL AMOUNT
4:VIEW ALL VEHICAL DEPARTURE TIME
*****
```

Fig 4.9 Checked out vehicle

CHAPTER 5

CONCLUSION

The computation of “Vehicle Parking Management system” has been developed by us through applying knowledge referring to certain vehicles, browsing some sites and through the help of external and internal facilities and using our own knowledge related to the subject itself.

The system has been made to transparent for the users so that client trust and can have faith in reliability of information. Our system is users friendly and users will be satisfied with the system.

CHAPTER 6

REFERENCES

Books:

1. File Structures with C++ by Michael J Folk.
2. File Structures using C++ by K R Venugopal, K G Srinivasa, P M Krishnaraj, Tata McGraw-Hill ,2009, Referred Page No-4,39-40.

Website:

1. https://youtu.be/3tx_VSuUePc
2. <https://www.w3schools.com/CPP/default.asp>