

**A MINI PROJECT REPORT**

*for*

**Mini Project I (21CSE38A)**

*on*

**EASY CATELLING**

*Submitted by*

**N. Mahith Kumar**

**USN: 1NH21CS170, Sem-Sec: 3-C**

*In partial fulfillment for the award of the degree of*

**BACHELOR OF ENGINEERING**

*in*

**COMPUTER SCIENCE AND ENGINEERING**

**Academic Year: 2022-23(ODD SEM)**



**CERTIFICATE**

This is to certify that the mini project work titled

**EASY CATELLING**

submitted in partial fulfillment of the degree of Bachelor of Engineering in Computer Science and Engineering by

**N. Mahith Kumar**

**USN:1NH21CS170**

*DURING*

*ODD SEMESTER 2022-2023*

*for*

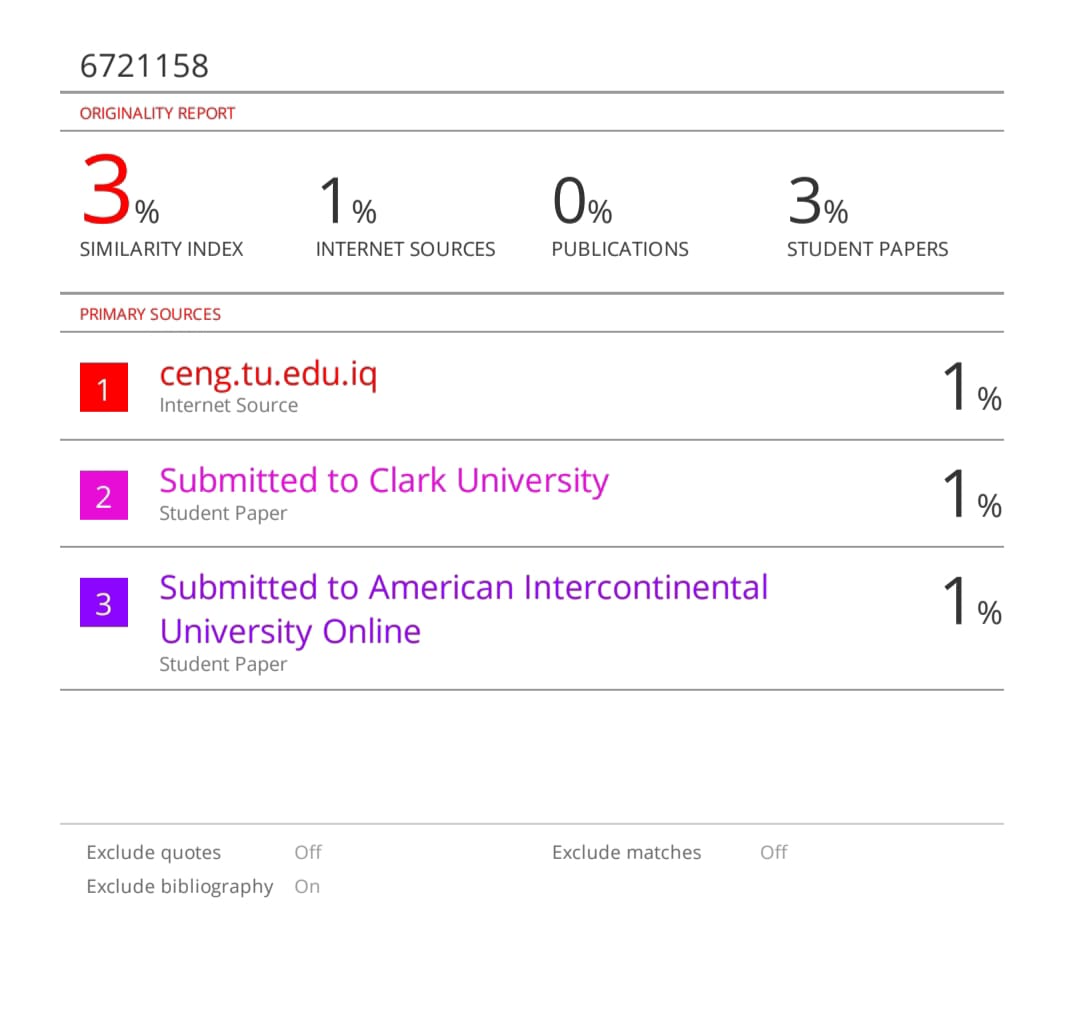
*Course: Mini Project I -21CSE38A*

Signature of Reviewer Signature of HOD

**SEMESTER END EXAMINATION**

*Name of the Examiner Signature with date*

|  |  |
| --- | --- |
| 1. |  |
| 2. |  |

**PLAGIARISM REPORT**

INDEX

|  |  |  |
| --- | --- | --- |
| CHAPTER NO. | TITLE | PAGE NO. |
|  | ABSTRACT | 06 |
|  | ACKNOWLEDGEMENT | 07 |
|  | INTRODUCTION | 08 |
| 1.1 | PROBLEM STATEMENTS | 08 |
| 1.2 | OBJECTIVES | 08 |
| 1.3 | METHODOLOGY TO BE FOLLOWED | 08 |
| 1.4 | EXPECTED OUTCOMES | 09 |
| 1.5 | REQUIREMENTS | 09 |
| 2.1 | DATA STRUCTURE | 10 |
| 2.2 | LOOPING STATEMENTS | 11 |
| 2.3 | PREPROCESSOR DIRECTIVES | 11-12 |
| 2.4 | LIBRARY FUNCTIONS | 13 |
| 2.5 | ARRAY WITH STRUCTURES | 14-15 |
| 3.1 | DESIGN | 16 |
| 3.2 | ALGORITHM/FLOWCHART | 17-18 |
| 4.1 | IMPLEMENTATION (INPUT) | 20-24 |
| 5.1 | IMPLEMENTATION (OUTPUT) | 25-27 |
| 6.1 | CONCLUSION | 28 |
| 7.1 | REFERENCES | 29 |

**ABSTRACT**

This sector continues to grow, the poor and downtrodden sections of our society are being left behind. They have to make do with the crumbling public healthcare system which by all means would mostly fail due to lack of adequate infrastructure and doctors.

The COVID 19 pandemic has also taught the world about the various means of self employment oppurtunities ,among them catelling is also one of its kind, last year India produced close to 146.31 million tonnes of milk, 50% more than the US ,and to maintain the record of such large number of cattle ‘EASY CATTLING’ project will be more helpfull

This mini-project solves the above-mentioned problems through the development of a desktop application. This desktop application would maintain a database of the users and other relevant information and would have a graphical user interface that would be simple and easy to use. Livestock plays an important role in Indian economy. Livestock contributed 16% to the income of small farm households as against an average of 14% for all rural households. Livestock provides livelihood to two-third of rural community. It provides employment to about 8.8 % of the population in India. India has vast livestock resources. Livestock sector contributes 4.11% GDP and 25.6% of total Agriculture GDP.

The entire program has been developed in C language and uses compilers like dev++ , turbo c++ to run the program.

The mini-project is completely based on the C programming to provide a simple and easy to understand platform for the users.

**ACKNOWLEDGEMENT**

**T**he satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman, New Horizon Educational Institutions, for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha,** Principal, New Horizon College of Engineering, for his constant support and encouragement.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and HOD, Department of Computer Science and Engineering, for her constant support.

I also express my gratitude to **Ms.M. Thanga Subha Devi**, Assistant professor, Department of Computer Science and Engineering,my mini project reviewer, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

Student name : N.MAHITH KUMAR

USN:1NH21CS170

**CHAPTER 1**

**INTRODUCTION**

**1.1 PROBLEM STATEMENTS**

EASY CATELLING is very difficult to maintain every record of the pet ID details in a book manually and also difficult to remove the ID details as well as for searching the details of pet ID is also difficult

to perform for customers. To avoid all these difficulties we can use this for project for storing the pet ID details as well.

**1.2 OBJECTIVES**

On behalf of the system there are four details which is manually enter to search name, breed, color , age which helps to differentiate from other cattles easily.Also we can delete a record by searching the particular cattle based on their ID view of all details.

## **1.3 METHODOLOGY OF THE PROJECT**

Structures of files contains are usually with name, breed, colour, age. To input of storing all this data we use to define the files which is very flexible to use and contaminate access data is easy. To do function,

We should use the start application and proceed by choosing the following options given the functions that appears first after opening the program is add, remove, search, view, logout application.

## **1.4 EXPECTED OUTCOMES**

Displaying and existing the code with the ID cattles by adding the information into an ID display. This will be easy and fast to perform and it reduces the time consistency daily. We can develop an app by this stuff to get more popularity and productivity by the sake of consequential data.

**1.5 REQUIREMENT SPECIFICATION**

**Software requirements:**

* C COMPILER or CODING C

**Hardware requirements:**

* RAM Required – 2 GB
* Above WINDOWS-7
* 64 BIT PROCESSOR

**CHAPTER 2**

**DATA STRUCTURE**

**2.1 INTRODUCTION TO DATA STRUCTURE**

* Data is having with the collection of elements and key words together to form a structure.

Data structure is a study of:

* How effectively the data is stored for persuant.
* How effectively the data is organized, retrieved and glory.

Data structures are of different structures are divided as primitive data structures and non-primitive structures.

Data structures which are primitive typed can takes the way in a program of Direct in manner.

**Primitive data structures**:

. int

. float

. char

. Pointer

**Non-Primitive data structures:** The data structures are maintained to non-sequential or indirective synthesis of structure.

Examples:

. Arrays

. Lists

. Files

* Lists are classified into two types are:

Linear lists consist of:

. Stacks

. Queue

. Linked list

Non-linear consists of:

. Trees

. Graph

* 1. **Looping Statements:**

**WHILE LOOP:**

This While is also a loop, it defines the statement to recall until the function gets true other terms will not execute or goes to out of the loop automatically shifting process made by this loop.

It is also called as entry controlled loop like it checks the statement primary then execute under the statements.

This will do a statement as increment or decrement of certain condition. In this all functions, variables are allocated.

This loop is of following variations: ¬ Variable initialization.

¬ Test condition.

¬ Loop-update.

Syntax of while loop Loop\_intialization;

While

(

test\_condition

)

{

Loop\_update;

Statement;

}

**FOR LOOP:**

This is a loop of conditioned with ‘for’ , it defines that the statement until the condition will get false other will goes out of the loop. It runs by automatically to define the conditions.

It involves the arrays, linked lists, strings etc., Mainly used for justify the programming who uses daily.

Loop will dictate from primary of the starting line of the loop, when the condition is approved the it will simply from starting value onwards.

Syntax of For loop

for(loop\_initialization;test\_condition;loop\_update)

{

Statements;

}

* 1. **PREPROCESSOR DIRECTIVES:**
* define: It support to take a substitution of a directive of a header file using main.
* include: A Header file will be insert from other also.
* Un-def: It makes non-defined to the certain statement.
* If-def: Macro is defined when it returns true.
* If-n-def: It may gets return true when certain statement will not be defined.

**2.4 LIBRARY FUNCTIONS:** In C the library functions are mostly made by the preprocessor directives and they are inbuilt functions. These inbuilt are located in certain location.

Generally these library functions are used to define the outputs. It should be stores in a header file called as #include<stdlib.h>.

By these library functions only the variable conditions are executing frequently to one by another otherwise my loss.

The functions with library are major importance to holding or to storing the heavy details in the program.

**HEADER FILES**

* Stdio.h - Defines the standard input and output files in which the functions must be declared.
* Conio.h - Consoles for doing the clrscr() file.
* String.h - All string functions are defined in this header file.

**2.5 ARRAYS WITH STRUCTURES:**

* A Set of collection of a Similar function of data type of variables are allocated at same position with a certain condition said to be an Array.
* A Set of collection of a Different function of data type of variables are allocated at same position with a certain condition said to be an Structure.

Syntax for a structure:

struct structname

{

// Structure definition

Datatype1 membername1;

Datatype2 membername2;

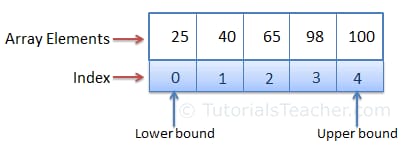
Datatype3 membername3;

};

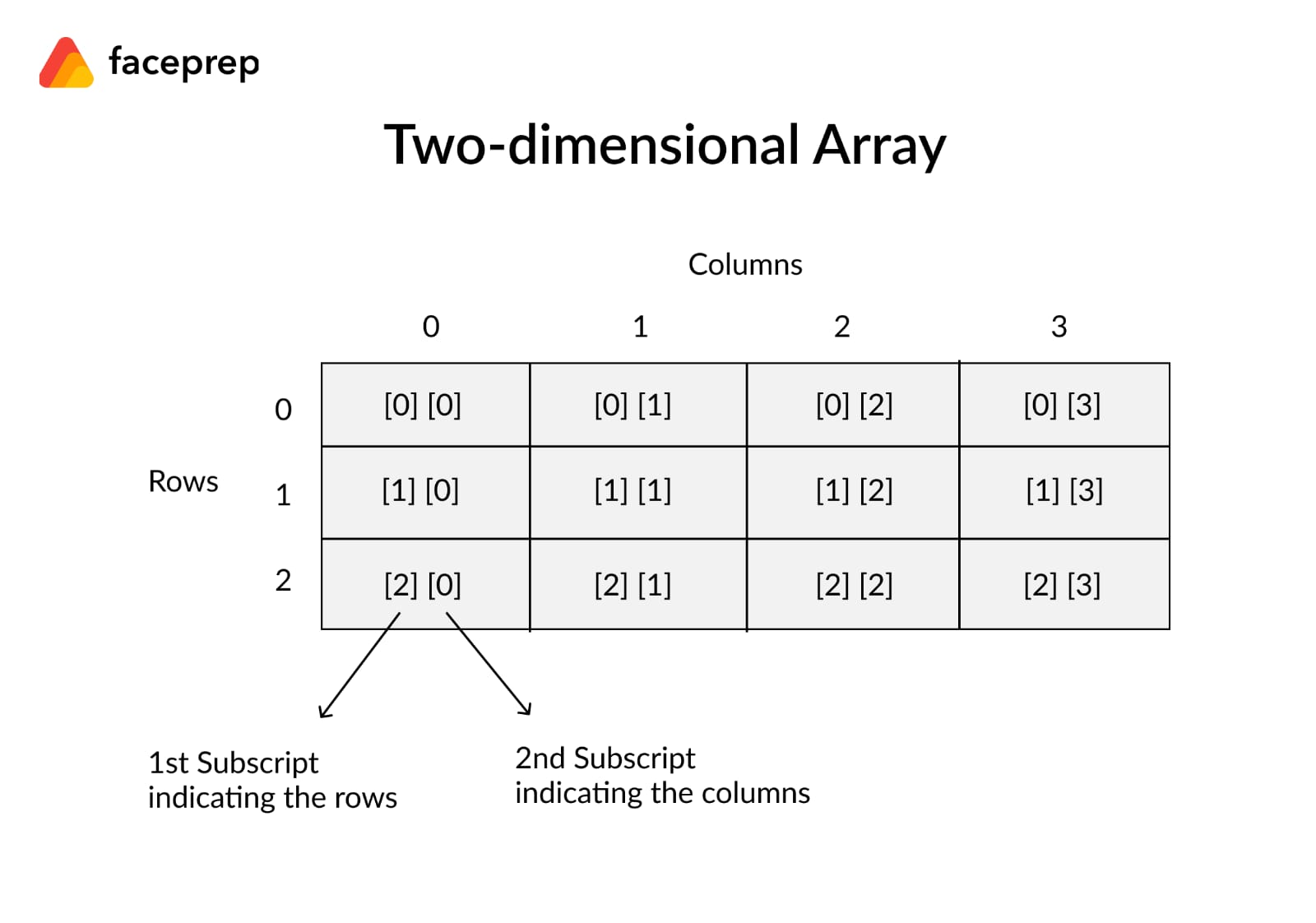
* The **Struct** is defined in C as key word.
* An Array is like {1,2,3,4,5}.
* By declaring an array in a structure to a variable type should be as symbolically first index.
* Array is of two types: 1.one-dimensional array.

2.multi-dimensional array.

1.**One-Dimensional Array**:-Array which the components are in the single list form.



2.**Multi-Dimensional Array**:- In these array the dimensions are listed in the shape of tabular form.



**CHAPTER 3**

**DESIGN**

**3.1 DESIGN GOALS**



* Our design mechanism having two main things:
  + 1. Record the details of cattle.
    2. Reduces complexity with time to add, modify,search and also deleting the data

**3.2 ALGORITHM / FLOWCHART**

START THE PROGRAM;

MAIN IS WEB PAGE;

Login with correct user ID and password;

Read the options for functions:

1.NEW ENTRY:

Takes name,age,breed,colour into an ID.

2.DISPLAY ALL ENTRIES:

It displays the previous all entries.

3.DELETE ENTRIES:

For deleting the particular entry as meanwhile.

1. :EXIT(0);

To exit from the main menu.

login

menu

write into file

input name, breed, age, colour

1.New entry

Print all details

read file

2.display all entries

Print all details in another file2 except record to be deleted

read file

Input id

3.delete

Rename file 2 as file 1

Delete the file 1

4.logout

**CHAPTER 4**

**IMPLEMENTATION**

**Top of FormBottom of Form**

* Program will saves the records of an ID’S for a long time.
* Unable to support by entering the duplicate passwords.
* By adding, displaying, searching and finally exiting from program can easily modified.
* Time is less consistency by using this program.
* **INPUT CODE IMPLEMENTATION**:-

struct Details

{ //Structures are implemented

char name[100];

char bf[100];

int age;

char color[100];

};

{

struct Details info[1000];

int n,m;

for(n=0;n<=10;n++)

{

printf("\*");

}

printf(" WELCOME TO ");

for(m=0;m<=10;m++)

{

printf("\*");

}

printf("\n\*\*\*\*\*\*\*\*\*\* EASY CATTELING \*\*\*\*\*\*\*\*\*\*\n\n\n");

while(1)

{

char user[]="admin";

int PW=123456;

char str[10];

int pw;

printf("ENTER THE USER ID :");

scanf("%s",str);

printf("\n ---------\*\*\*\*\*\*----------");

printf("\n ENTER THE PASSWORD:");

scanf("%d",&pw);

if(PW==pw)

{

printf("YOU HAVE SUCESSFULLY LOGGED IN.....!!!");

printf("PRESS ENTER >>>");

break;

}

else

{

printf("INCORRECT LOGIN DETAILS");

printf("PLEASE TRY AGAIN!!!\n\n");

printf("PRESS ENTER >>>");

}

}

while(1)

{

printf("\n\*\*\*\*\*\*\*\*\* MENU \*\*\*\*\*\*\*\*\*\n");

printf("1.new entry\n");

printf("2.view all entries\n");

printf("3.to view perticular entry\n");

printf("5.logout\n"); //Arrays are implemented

// char name[100];

// char bf[100];

// int age;

// char color[100];

int cou;

while(1){

int id;

printf("enter the id of cattle to be searched : ");

scanf("%d",&id);

if(id==1){

printf("enter the count of details\n");

scanf("%d",&cou);

for(int i=0;i<cou;i++)

{

printf("ENTER NAME:\n");

scanf("%s", &info[i].name);

printf("ENTER BREED:\n");

scanf("%s", &info[i].bf);

printf("ENTER age:\n");

scanf("%d", &info[i].age);

printf("ENTER color:\n");

scanf("%s",&info[i].color);

printf("DETAILS HAVE BREED SAVED SUCCESSFULLY\n");

}

}

else if(id==2){

for(int i = 0; i < cou; i++){

printf("\n NAME: %s\n", info[i].name);

printf("\n BREED: %s\n",info[i].bf);

printf("\n color: %s\n",info[i].color);

printf("\n AGE: %d\n",info[i].age);

}

}

if(id==3){

printf("\nTOTAL NUMBER OF RECORDS:%d\n",cou);

int index;

printf("Enter the index to view a particular entry ");

scanf("%d", &index);

if(index > cou){

printf("Error!");

}

else{

printf("\n NAME: %s\n", info[index].name);

printf("\n BREED: %s\n",info[index].bf);

printf("\n color: %s\n",info[index].color);

printf("\n AGE: %d\n",info[index].age);

}

}

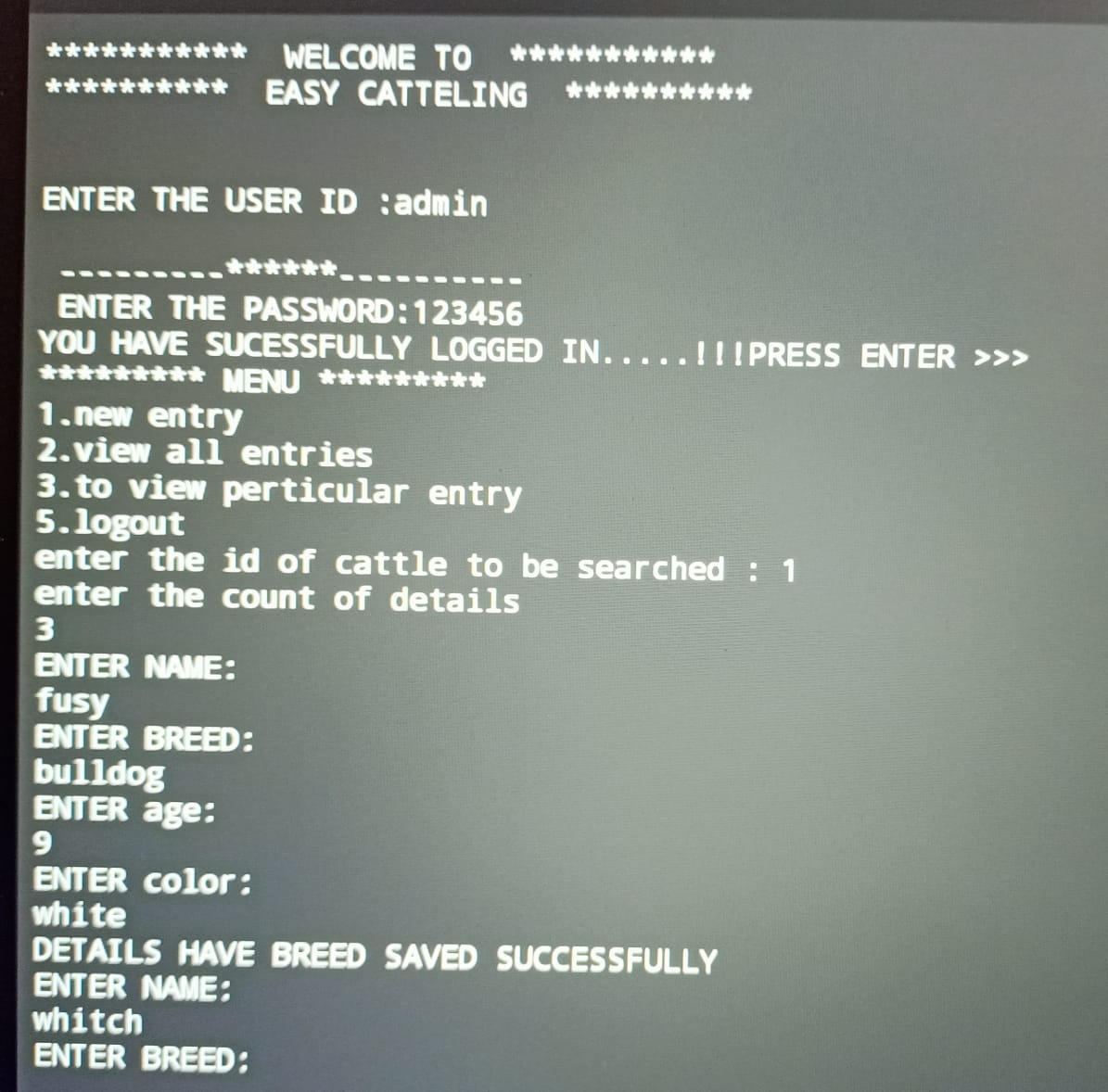
else if(id == 5){

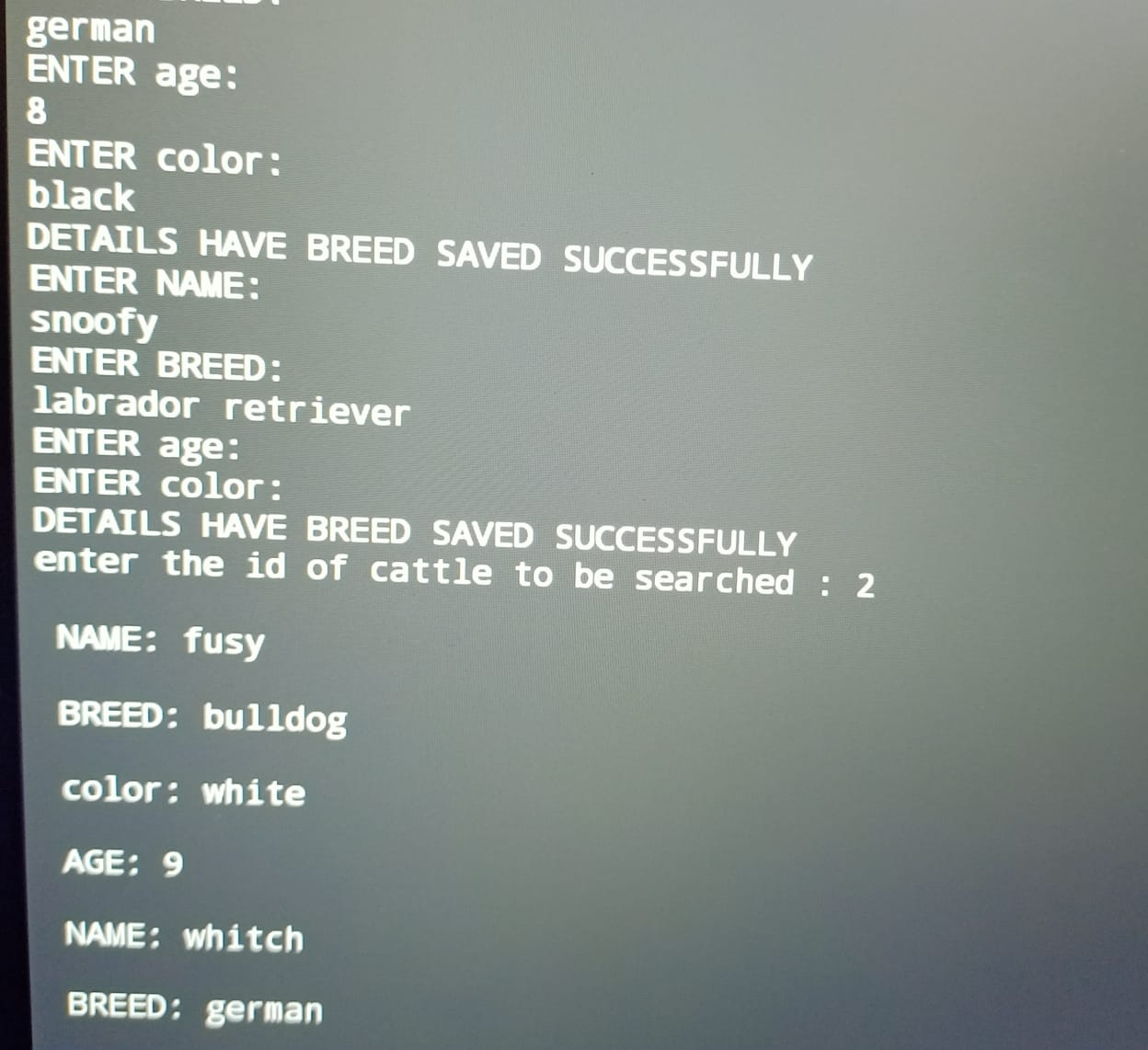
printf("logout \n");

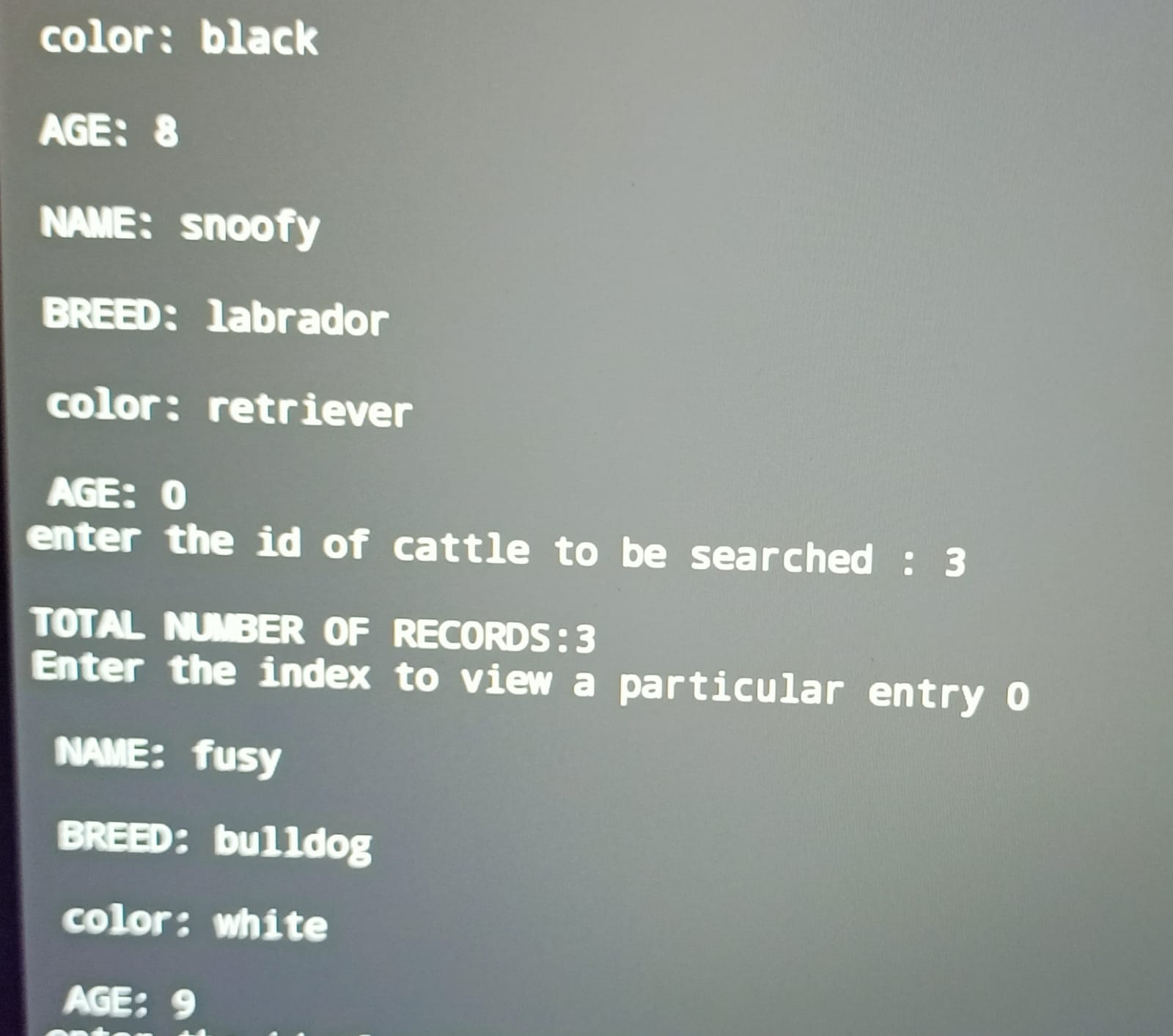
printf("\n THANKS FOR USING EASY CATTELING \n");

**CHAPTER 5**

* **OUTPUT CODE IMPLEMENTATION**:-



****

****

**CHAPTER 6**

**CONCLUSION**

On my observation of this perform only I uploaded the content properly. The application was built on the several steps, consequently to take up the project next level and implementing stage of course as a significant figures of the code is having and without error making and unofficial category functions are not applied here.

Report consisting with proper stuff with consequence code along with algorithm and flow chart.

Generally this program is cattle workers, it consumes to take less time for working and doing is easy process for executing to customers for sale on public for pet lovers. this makes to easy for adding, searching, deleting processes.

**CHAPTER 7**

**REFERENCE**

* + [www.geeksforgeeksorg/data-structures/linked-list/](http://www.geeksforgeeksorg/data-structures/linked-list/)
  + www.udemy.com/course/datastructurescncpp/learn/lecture/