**NITTE MEENAKSHI INSTITUTE OF TECHNOLOGY**

(AN AUTONOMOUS INSTITUTION, AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, APPROVED BY AICTE & GOVT.OF KARNATAKA

****

**OOP PROJECT REPORT**

*Submitted in partial fulfilment of the requirement for the award of Degree of*

*Bachelor of Engineering*

*in*

*Computer Science and Engineering*

Submitted by:

|  |  |
| --- | --- |
| NEELU KUMARI | USN: 1NT19CS124 |
| SHASHANK KUMAR | USN: 1NT19CS175 |
| STUTI PRASAD | USN: 1NT19CS188 |
| RAGHAV SWAMI | USN: 1NT19CS147 |



**Department of Computer Science and Engineering**

2020-21

|  |  |  |
| --- | --- | --- |
|  | **Nitte Meenakshi Institute of Technology**  (AN AUTONOMOUS INSTITUTION AFFILIATED TO VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM)  PB No. 6429, Yelahanka, Bangalore 560-064, Karnataka  Telephone: 080- 22167800, 22167860  Fax: 080 - 22167805 |  |

**Department of Computer Science and Engineering**

**CERTIFICATE**

This is to certify that the Course Project titled “We Ensure , You get Insured” is an authentic work carried out by NEELU KUMARI (1NT19CS124), SHASHANK KUMAR (1NT19CS175), STUTI PRASAD (1NT19CS188) , RAGHAV SWAMI (1NT19CS124) Bonafede students of **Nitte Meenakshi Institute of Technology**, Bangalore in partial fulfilment for the award of the degree of ***Bachelor of Engineering*** in COMPUTER SCIENCE AND ENGINEERING of Visvesvaraya Technological University, Belagavi during the academic year ***2020-2021.***

|  |  |  |
| --- | --- | --- |
| **Name Signature of the Faculty In charge** |  | **Name and Signature of the HOD** |

**DECLARATION**

We hereby declare that

(i) This Presentation does not contain text, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the report and in the References sections.

(ii) All corrections and suggestions indicated during the internal presentation have been incorporated in the report.

(iii) Content of the report has been checked for the plagiarism requirement

Name : SHASHANK KUMAR USN : 1NT19CS175 Signature :

Name : RAGHAV SWAMI USN : 1NT19CS147 Signature :

Name : NEELU KUMARI USN : 1NT19CS124 Signature :

Name : STUTI PRASAD USN : 1NT19CS188 Signature :

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Date: 07/01/2021**

**ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our effort with success. we express our sincere gratitude to our Principal **Dr. H. C. Nagaraj**, Nitte Meenakshi Institute of Technology for providing facilities.

We wish to thank our HoD**, Dr. Thippeswamy M.N** for the excellent environment created to further educational growth in our college. We also thank him for the invaluable guidance provided which has helped in the creation of a better technical report.

Thanks to our Subject Faculty. We also thank all our friends, teaching and non-teaching staff at NMIT, Bangalore, for all the direct and indirect help provided in the completion of the presentation.

|  |  |  |
| --- | --- | --- |
| Name  SHASHANK KUMAR  NEELU KUMARI  STUTI PRASAD  RAGHAV SWAMI | USN  1nt19cs175  1nt19cs124  1nt19cs188  1nt19cs147 | Signature |

Signature

Date: 07/01/2021

**ABSTRACT**

Nowadays animals are not taken care of properly mainly the pets who don’t have much experience in wild. So **Pet NGO** is a solution for this problem. It is an organization with people of like minds towards animals specially for pets.

To solve there need of providing a simple and reliable management system we have come up with a solution in c++ with our database as simple text files. The software will be a terminal program which later can be upgraded into graphical user interface. In terminal as well we have made it user friendly and can be easily used by anyone with no programming background as well. We have used basic concepts of object oriented programming such inheritance privacy and authentication system a decent authentication system. We have implemented date and time for storing everything date and timewise in our file based database.

**CITATION**

1. Stack overflow
2. GeeksForGeeks
3. Muythuraj sir notes

**Table of Contents**

1. Explanation of project partwise
2. Chapters Used in the project from OOP

**Explanation of Project Partwise**

1. Introduction and Flow

2. Customer

3. Donate, Report , Adoption , Ticket booking

4. Admin

5. File management

6. Utility functions

7. Designs

1. **Introduction and flow of work**

We have created basically a customer class for the clients visiting PET NGO and then various other classes is inherited from that Customer class as Donate for donation, Report for reporting a pet, Adoption for adopting a reported pet, and at last we are making a ticket booking class for booking ticket for shows going on in PET NGO. Then we have created an independent class for admin access and then there is a header file for utility functions for various functions that we have used so much. And for designing in terminal we have created a design header file. And at last the whole backend management or database management is done in a header file called storing.h .

So our file structure for the project is like this

**/project/**

**data/**

**classes.h**

**design.h**

**main.cpp**

**storing.h**

**utility.h**

data/ folder contains our database which is fully automated if we delete it will be created automatically when program is ran.

1. **Customer class**

Customer class has various data members such as first name, last name, age, gender, username, password. There are two functions called **signup** and **signin.**  As the name suggest the functions do their work. First the user should be authenticated to do anything at all in the system

**Case 1: Signup**:

User will be asked to enter basic details and then a username the program will check if the username already exists or not if not the username will be created and a folder of name username will created inside data folder and then the user will be asked to enter a secret number and a password for his/her authentication. The password is first hashed using the secret key the user chose and then all the info including password is stored in two text files (separate file for password). And then the client is logged in and asked for which path he want go.

**Case 2: Signin:**

If a user has already created their account with us then we would be prompted to enter his/her credentials as username, secret key and password. First validity of username is checked if the username exists or not and if it exists then password entered by the user is hashed again and compared with already stored hashed password in the database if both matches he/she is authenticated and logged in and prompted to choose path.

1. **Donate, Report, Adoption and TicketBooking**

These are various classes which is inherited from Customer class. Below they are explained one by one

**Case1 : Donate:**

If user chooses Donate after authentication he/she will be asked if he wants to donate some money or see his previous transaction that he did with the PET NGO.

If the user chooses to donate he/she will be asked how much money they want to donate and then it will be saved in our database with transaction Id and date time of the system

**Case2: RetrieveDoantion:**

For retrieval of the ticket of transaction the user will be asked to to enter the transactionId for the transaction he wants to see

**NOTE: Same will be done for Report class as well as done with donation class but while storing report we will also store the report of animal as our stock of pets for adoption purposes and taking care of pets who are left on street**

**Adoption:**

For adoption we will ask user which type of pet he wants( e.g. dog, cat)

Suppose user entered dog then we will search for dog in our database if we have any dog left in the database (we stored pets from the report class in our database).

We will just retrieve and give it to customer.

**Ticket Booking:**

There are various shows that the PET NGO conducts on a daily basis for spreading awareness to people about taking care of the pets. For that shows we have implemented a decent ticket booking system in C++ where a person is stored in a structure of name gender and age and this structures are stored as an array of structure later on all of this is stored in our database for future references or if the user forgets his or her tickets we can provide it.

1. **Admin**

We have implemented a whole new header file with just one class as admin in which we have used same authentication system for authenticating the admin and then there are various functions for admin to see various things.

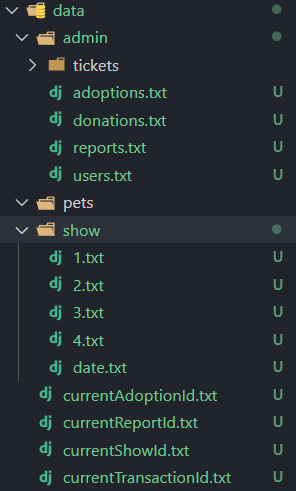
|  |  |
| --- | --- |
| **Function name** | **Use** |
| displayUserByName(string) | Display all users with same name |
| displayDonationByUser(string) | Display all donations made by a user |
| displayReportsByUser(string) | Display all reports made by a user |
| displayPets(string) | Display all the pets in stock |
| displayAllTicketsOfToday() | Display all the tickets booked for today |

1. **File management**

For storing all the data we have implemented a file structure system which stores all the information separably and easy to sort.

There are various functions to do this they all basically store data of different class so that we can retrieve it very quickly

The basic file structure without any user is like this



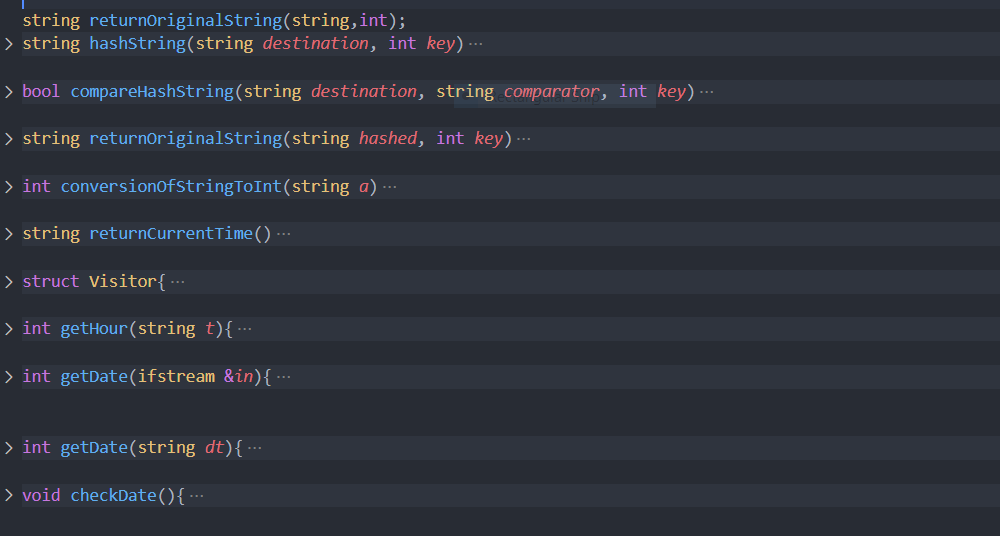
For user it will be created a new folder of his username and his all the transactions and everything will be stored inside its own file structure

The admin folder contains a copy or link of the contents for the things user do in the system. Admin can access anything anytime

1. **Utility functions**

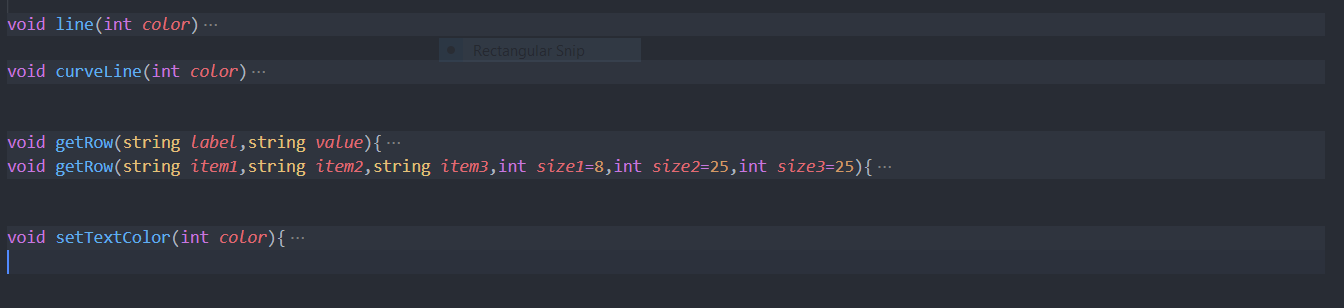
Above all the backend system is implemented for implementing those backend we need some utility functions for conversions grabing time and many other things so me have implemented various functions for that purposes in a file called **utility.h**

The snapshot of the names of the functions are as followed the name will only define the use of the functions

****

1. **Design**

Now who doesn’t want a good looking app we all want that so for that we have implemented some basic functions for creating tables on terminal and changing colors of the text in the terminal and drawing a line and curve lines on the terminal below are snapshots of the functions



Concepts Used from OOP

1. **Data encapsulation**
2. **Inheritance**
3. **Friend classes and friend functions**
4. **Overloading of Functions**
5. **File management from fstream class**
6. **Time from system using time.h and chrono.h**
7. **Using terminal commands in C++**

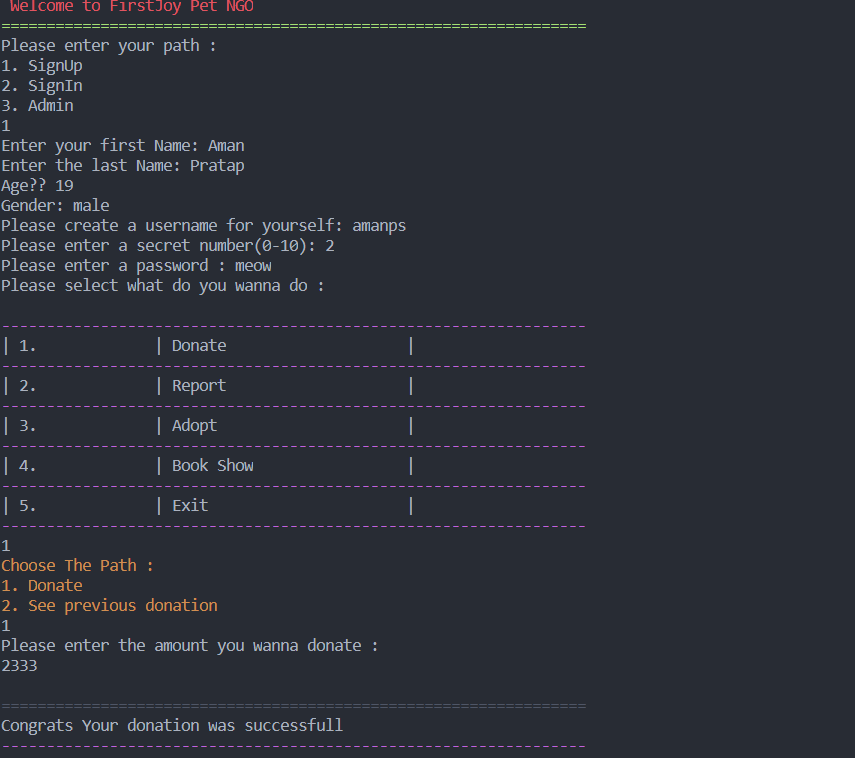
**Summary and future work**

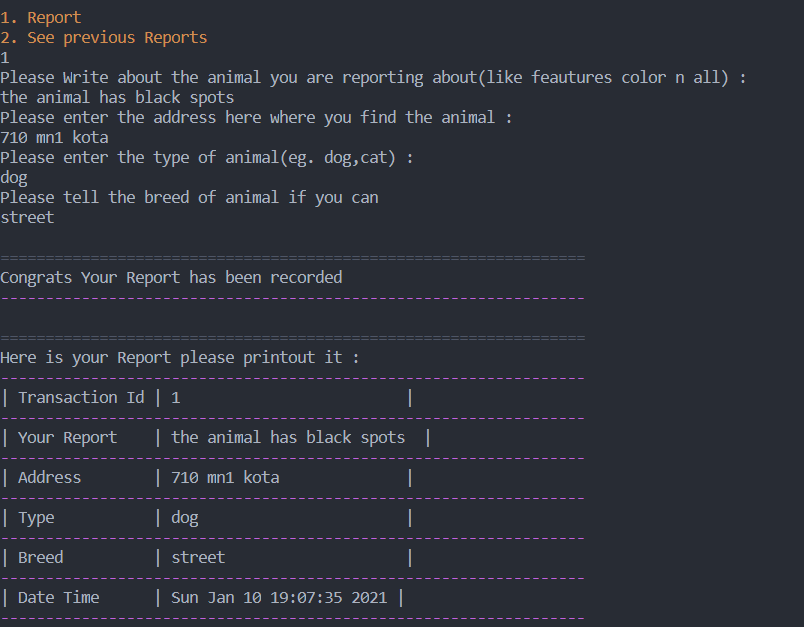
So we have implemented this whole software by following the above concepts.

Our future aim is to convert this terminal software into a graphical user interface with much more advance database now we have used file system as our database our aim is to make SQL as our database and then our data can be easily managed for a ton of people this program is good for the association in the initial stage later on when customer will increase we will make beautiful graphical user interface and better database for managing large amount of data

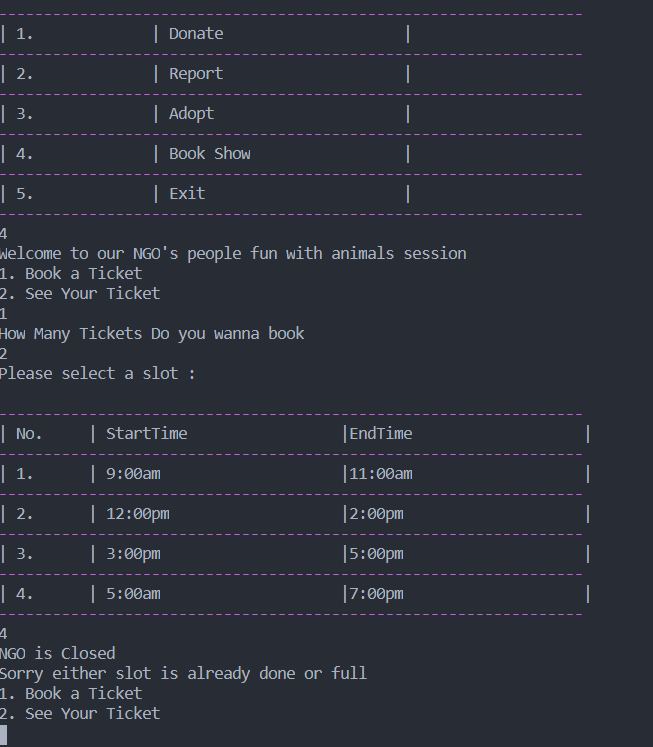
in a easy and reliable way and we will also make a website for this organizations so the user can donate things online and wont have to come to the headquarters .

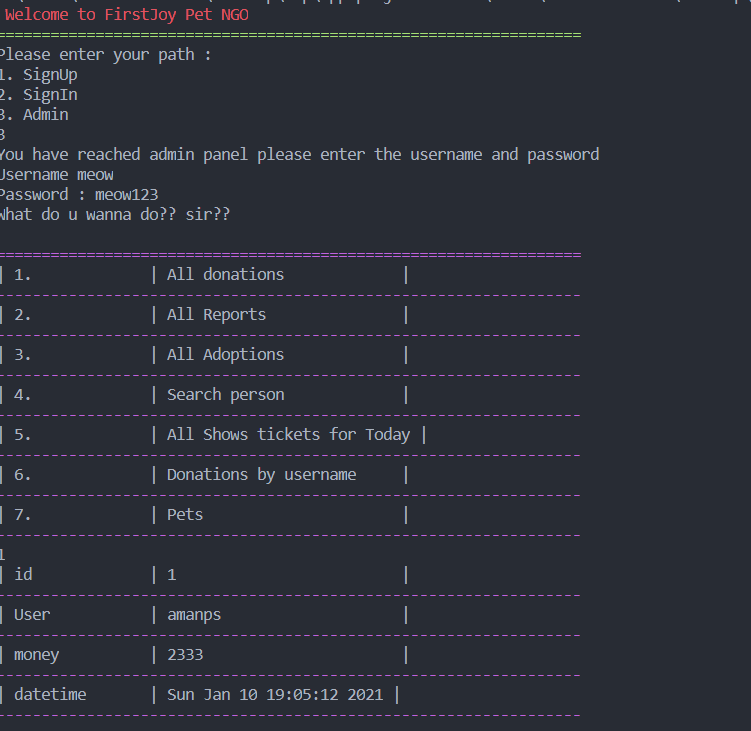
**Screenshots of the program**

****

****







**Bibliography**

**[1].The C++ Programming Language**

**Book by Bjarne Stroustrup**

**[2].C++ Primer**

**Book by Josée Lajoie and Stanley B. Lippman**

**[3].Object-oriented programming in Microsoft C++**

**Book by Robert Lafore**

**[4].Object-oriented programming with C++**

**Book by Raimund K. Ege**

**[5].A Tour of C++**

**Book by Bjarne Stroustrup**

**[6] Printed C++ notes**

**by Muthuraj V**