KARNATAK UNIVERSITY DHARWAD

Janata Shikshana Samiti's

Shri Manjunatheshwara Institute of UG & PG Studies
Vidyagiri, Dharwad - 580 004

A PROJECT REPORT ON

"ORPANAGE MANAGEMENT SYSTEM"

BACHELOR OF COMPUTER SCIENCE (B.Sc. (CS))
OF
KARNATAK UNIVERSITY, DHARWAD

PROJECT GUIDED BY

Smt. UJWALA N. GRAMPUROHIT

Submitted by:

AMULYA.K
B.Sc. (CS) VI Semester
REG:19M10107

E. SHIRISHA
B.Sc. (CS) VI Semester
REG:19M10129

DEPARTMENT OF COMPUTER SCIENCE - 2022

JANATA SHIKSHANA SAMITI'S

SHRI MANJUNATHESHWARA INSTITUTE OF UG & PG STUDIES VIDYAGIRI, DHARWAD - 580004

CERTIFICATE

This is to certify that project work entitled **"ORPHANAGE MANAGEMENT SYSTEM"** is a bonafied work, successfully completed by **Amulya k (19M10107)** and **E Shirisha(19M10129)** of B.Sc (CS) 6th semester of the Department of Computer Science in partial fulfillment of the requirements for the award of B.Sc (CS) as prescribed by the Karnatak University, Dharwad during the academic year 2021-2022.

Smt. Ujwala N. Grampurohit	**Shri. Vivek M. Laxmeshwar**	**Dr. Ajith Prasad**
[Project Guide]	[HOD-Computer Dept]	[Principle]

Examiners:

1) _____

2) _____

DECLARATION

We, **Amulya k** and **E Shirisha** students of VI Semester B.Sc.(CS), Department of Computer Science, JSS SHRI MANJUNATHESHWARA INSTITUTE OF UG & PG STUDIES, VIDYAGIRI, DHARWAD, affiliated to Karnatak University, Dharwad hereby declare that the Project entitled "**ORPHANAGE MANAGEMENT SYSTEM**" submitted in partial fulfilment of the course requirement for the award of degree in Bachelor of Computer Science, Karnatak University, Dharwad during the academic year 2021-2022. We have not submitted the matter embodied to any other university or institution for the award of any other degree.

Amulya k **Reg. No. : 19M10107**

E Shirisha **Reg. No. : 19M10129**

Date :

Place : Dharwad

Client letter

Registration No: P148/1911 Phone Number:9986978199

JESUS SHIKSHANA SAMSTHE

Near Sai water Plant,P.W.D. Camp,sindhanur,Rachiur,Karnataka-584128

Ref No. *Date:02.08.2022*

ACCEPTANCE LETTER

Dear Amulya k & E Shirisha,

We are pleased to inform you that we have accepted your proposal for providing the web application for our Orphanage (Light of Hope) with the objective of reducing manual work and computerization of the working process. We are quite impressed with your comprehensive understanding of the scope of this project and your confidence in being able to complete the project on or before the assigned deadline. We are confident about this project and you will provide us all the essential requirements in a professional manner in a time bound manner. Hope this project work will help you in enhancing your skills in developing many such web applications and have a bright future

Yours sincerely

Signature 2/8/22

ACKNOWLEDGEMENT

The successful presentation of this project is acknowledgement of the immense support extended by **JSS SMI UG and PG STUDIES, DHARWAD**, which was provided us opportunity to fulfill the most cherished & desired way to reach our goal.

We would like to express our heartfelt thanks to our President Shri Shri Vishwaprasanna Theertha Swamiji of Sri Pejavarmath, Udupi, Parama Pujya Dr. D. Veerendra Heggade, honourable Rajyasabha Member, Chairman of Janata Shikshana Samiti, Dharmadhikari of Dharmasthala and The Secretary of Janata Shikshana Samiti Dr. N. Vajrakumar.

We would like to express our sincere and hearty thanks of gratitude to our beloved Principal Dr. Ajith Prasad and our Head of the Computer Science Department Shri Vivek M. Laxmeshwar who gave us the golden opportunity to do this wonderful project on the topic **"ORPHANAGE MANAGEMENT SYSTEM"**, which also helped us in doing a lot of research and exposed to lot of new information that would help us in our mere future.

We would also take this opportunity to offer our sincere gratitude to our Project guide Smt. Ujwala N. Grampurohit for her excellent support throughout the development of this project and providing the necessary information on our demand at all the times.

Amulya k

E Shirisha

INDEX

1.PROJECT SYNOPSIS

PROBLEM DEFINITION

Orphanage was still using paper- based system to collect and store data since it was established. The paper based file system was faced with many problems in daily work such as loss of some documents. Other problem includes lack of security, due to poor storage of data so everyone could read it because there is no central automated security to secure data.

Data duplication, because everyone has to fill the data at its own time and departments and there was no administrator to check or control the data, redundancy that appeared several times. Separation and isolation of data, it was more difficult to access data which was isolated in separate files. Updating of files was difficult to update files since they are mixed up, this makes it difficult to get them from the shelves. Therefore a computerized system will be introduced to alleviate the mentioned problems.

OBJECTIVES

- The main objective of the project is to design, develop and test a reliable computerized information system which will orphanage to keep, maintain records of data.
- To promote the knowledge of Orphanage for purposes of income generation.
- Adoption applications is provided for the adoption of the child.
- Helping hands will increase through this web application.

BASIC MODULES

- Admin
- User
- Staff
- Donation
- Expenses
- Adoption

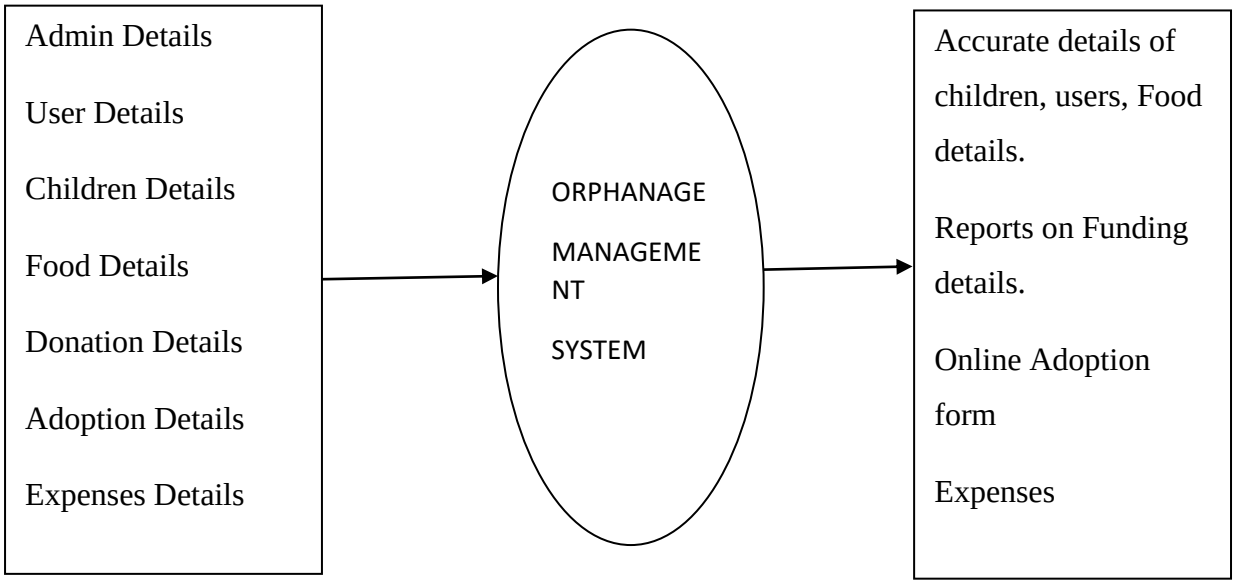
INPUT OF THE PROJECT

- Admin Details
- User Details
- Children Details
- Food Details
- Donation Details
- Adoption Details
- Expenses Details

OUTPUT OF THE PROJECT

- Accurate details of children, users, Food details.
- Reports on Funding details.
- Online Adoption form
- Expenses

PROCESS LOGIC



TOOLS OR PLATFORMS, LANGUAGES TO BE USED

Hardware Requirements:

Operating System	Windows 7 and above
Hard disk	Minimum 20GB
RAM	Minimum 4GB
Processor	Intel i3, i5.

Software Requirements:

IDE	Visual studio code
Front End	HTML, CSS, JavaScript
Middle ware	PHP
Back End	MYSQL
Web Browser	Google Chrome, FireFox, Microsoft Edge
Server	XAMP

ARE YOU DOING THIS PROJECT FOR ANY INDUSTRY/CLIENT?

Yes.

Client: Light of hope Orphanage.

Place: Sindhanur.

Name: JOSEPH CHRISTOPHER

DURATION OF THE PROJECT.

Two Months.

MEMBERS OF THE PROJECT

E. SHIRISHA (19M10129)

AMULYA. K (19M10107)

LIMITATIONS

- Online Adoption can't be done (Adoption form will be provided online).
- Online transaction can only done my the contact and account details of the admin.

SCOPE

- Orphanage management system will store all the details of the orphan's including, educational qualifications, personal details and medical reports all the information related to orphan's.

FUTURE

- It may get more support from NGO's and Government.
- The Online transactions can be done by UPI.

2.FRAME WORK

➢ PHP

PHP started out as a small open source project that evolved as more and more people found out how useful it was. Rasmus Lerdorf unleashed the first version of PHP way back in 1994.

- PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
- PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content, databases, session tracking, even build entire e-commerce sites.
- It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
- PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.
- PHP supports a large number of major protocols such as POP3, IMAP, and LDAP. PHP4 added support for Java and distributed object architectures (COM and CORBA), making n-tier development a possibility for the first time.
- PHP is forgiving: PHP language tries to be as forgiving as possible.
- PHP Syntax is C-Like.

Common uses of PHP

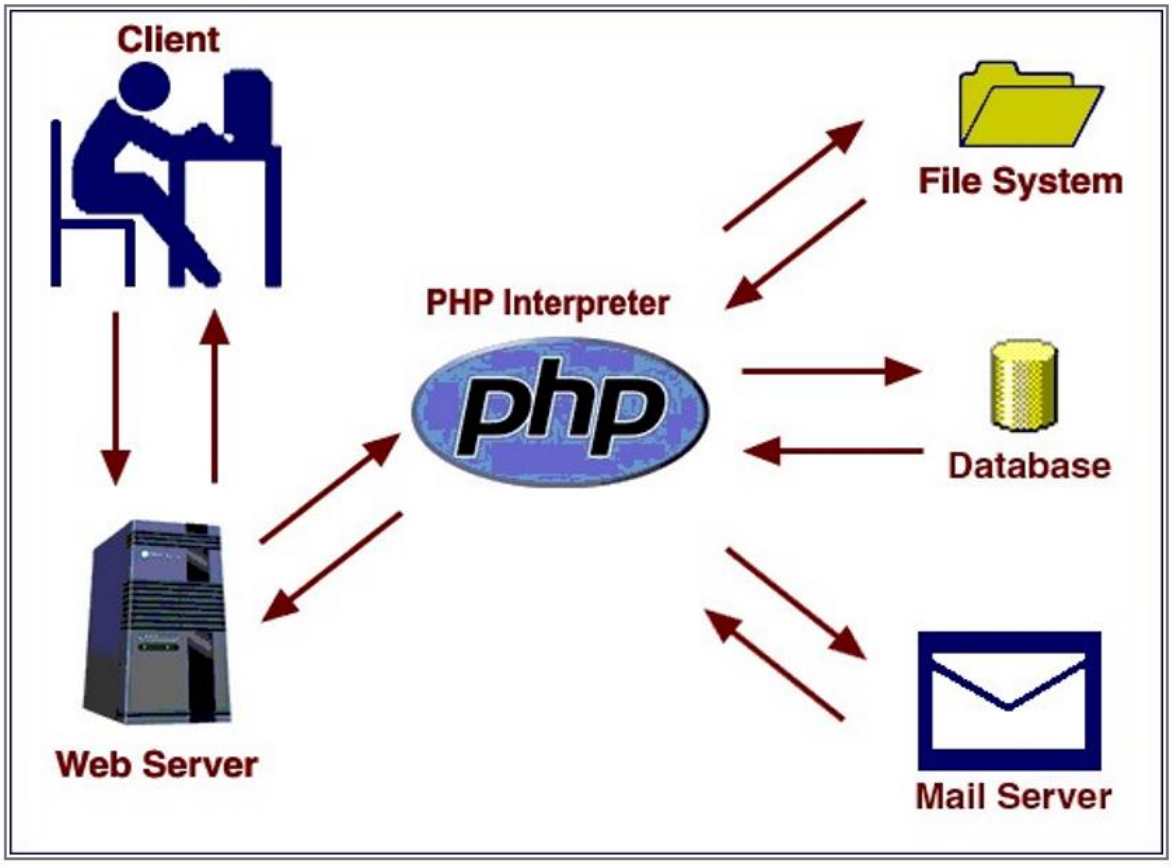
- PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
- PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
- You add, delete, modify elements within your database through PHP.
- Access cookies variables and set cookies.
- Using PHP, you can restrict users to access some pages of your website.
- It can encrypt data.

Characteristics of PHP

Five important characteristics make PHP's practical nature possible −

- Simplicity
- Efficiency
- Security
- Flexibility
- Familiarity

PHP ARCHITECTURE



➢ XAMPP

XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. It was developed by the **Apache Friends**, and its native source code can be revised or modified by the audience. It consists of **Apache HTTP Server, MariaDB, and interpreter** for the different programming languages like PHP and Perl. It is available in 11 languages and supported by different platforms such as the IA-32 package of Windows & x64 package of macOS and Linux.

XAMPP is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MYSQL, and the Ps stand for PHP and Perl, respectively. It is an open-source package of web solutions that includes Apache distribution for many servers and command-line executables along with modules such as Apache server, MariaDB, PHP, and Perl.

XAMPP helps a local host or server to test its website and clients via computers and laptops before releasing it to the main server. It is a platform that furnishes a suitable environment to test and verify the working of projects based on Apache, Perl, MySQL database, and PHP through the system of the host itself. Among these technologies ,Perl is a programming language used for web development, PHP is a backend scripting language, and MariaDB is the most vividly used database developed by MySQL.

➢ MYSQL

MySQL tutorial provides basic and advanced concepts of MySQL. Our MySQL tutorial is designed for beginners and professionals.

MySQL is a relational database management system based on the Structured Query Language, which is the popular language for accessing and managing the records in the database. MySQL is open-source and free software under the GNU license. It is supported by **Oracle Company**.

Our MySQL tutorial includes all topics of MySQL database that provides for how to manage database and to manipulate data with the help of various SQL queries. These queries are: insert records, update records, delete records, select records, create tables, drop tables, etc.

What is Database?

It is very important to understand the database before learning MySQL. A database is an application that stores the organized collection of records. It can be accessed and manage by the user very easily. It allows us to organize data into tables, rows, columns, and indexes to find the relevant information very quickly. Each database contains distinct API for performing database operations such as creating, managing, accessing, and searching the data it stores. Today, many databases available like MySQL, Sybase, ORACAL, MongoDB, PostgreSQL, SQL Server etc.

What is MySQL?

MySQL is currently the most popular database management system software used for managing the relational database. It is open-source database software, which is supported by Oracle Company. It is fast, scalable, and easy to use database management system in comparison with Microsoft SQL Server and Oracle Database. It is commonly used in conjunction with PHP scripts for creating powerful and dynamic server-side or web-based enterprise applications.

- **HTML**

HTML means Hypertext Markup Language. HTML is a method of describing the format of documents which allows them to be viewed on computer screens. HTML documents are displayed by web browsers, programs which can navigate across networks and display a wide variety of types of information. HTML pages can be developed to be simple text or to be complex multimedia extravaganzas containing sound, moving images, virtual reality, and Java applets.

The global publishing format of the Internet is HTML. It allows authors to use not only text but also format that text with headings, lists, and tables, and to include still images, video, and sound within text. Readers can access pages of information from anywhere in the world at the click of a mouse button. Information can be downloaded to the reader's own PC or workstation. HTML pages can also be used for entering data and as the front-end for commercial transactions.

Features of HTML

- It is not a programming language.
- It is not a data description language.
- It is simple to understand and implement.
- HTML constructs a very easy to comprehend, and can be used effectively by anybody.
- The methodology used by HTML to mark up information is independent of its representation on a particular hardware or software architecture.
- HTML syntax is a worldwide standard.

➢ CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content. achading layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate css file, and reduce complexity and repetition in the structural content

Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.

- **JAVA SCRIPT**

  - JavaScript is a very powerful client-side scripting language.

  - JavaScript is used mainly for enhancing the interaction of a user with the webpage.

  - In other words, you can make your webpage more lively and interactive, with the help of JavaScript.

  - The language was initially called Live Script and was later rename JavaScript. The syntax of JavaScript is mostly influenced by the programming language C

  - JavaScript is a cross-platform, object-oriented scripting language used to make WebPages interactive (eg, having complex animations,clickable buttons, popup menus, etc.).

  - JavaScript contains a standard library of objects, such as Array, Date,and Math, and a core set of language control structures, and statements. elements such as operators. Client-side JavaScript extends the core language bysupplying objects to control a browser and it's Document Object Model (DOM).

3.PROJECT SUBJECT

➢ MODULES

a. ADMIN:

- In this administration of light of hope orphanage is to be maintained.
- The details of children, donations, expenses, all under control of the admin.
- The admin will edit, update all the details as it is required.

b. USER:

- The user must have the personal account.
- The user is of two types

1. Donor's
2. Adopters

- Donor's will donate money, food, cloths, medicines etc for the children.
- If the user want to Adopt a child then the user has to fill the application form.

c. STAFF:

- His contact and service details management.
- Monthly reports of his attendance.
- Every Staff salary report.

d. DONATION:

- All the donated details are recorded under admin.
- The report on donations are managed.

e. ADOPTION

- The details of adopted child are managed.
- Application Details for adoption are also managed by the admin.

4.SYSTEM REQUIREMENTS AND ANALYSIS:

➢ Specification

Specification is used to specify the requirement for the initial implementation of system and update in the future. Software Requirement Specification bridges the gap between

Tools and Platforms used

Back end-MySQL10.4.24

Front end-PHP5.4.16

Software requirements

Platform : PHP

Database: My SQL

Tools: HTML, CSS & Java Script

Operating system: Windows 10

Designing tool: Visual Studio Code

Hardware requirement:

Processor : Intel CORE i3,i5

RAM: 4GB

Hard disk: Minimum 60MB

➢ System Analysis

Database Storage

Proposed Database is intended to store, retrieve, update, and manipulate details of library, which include tables.

· Admin Account.

Design Constraints

- Complete validation has been done no mandatory fields was to be left unfilled, if left unfilled then appropriate alert message will be displayed.
- Data in the database cannot be modified until and unless you are the proper authenticator.
- Efficient error handling capacities have been provided.
- The Admin can view the records whenever and wherever required as the system is easily available on the net.

➢ Feasibility Study :

Preliminary investigations examine project feasibility; the likelihood of how the system will be useful to the organization. The system is thinking of developing a common admission portal to facilitate transparent, speedy and cost effective way for public. The system will eliminate the need to visit each agency manually. Instead register and hire in a easy way. The system facilitates an orderly process. Three tests of feasibility-all equally important are studied.

➢ Operational feasibility:

This test of feasibility asks if the system will work when it is developed and is the current business methods acceptable to the user.

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system. The proposed system is very user friendly with good GUI.

- **Technical Feasibility:**

It is based on technical concept such as does the proposed system have the technical capacity to hold the data required to use the system.

The technical requirement for the system is economic and it does not use any other additional Hardware and software.

Financial Feasibility:

It is based on the financial concept such as is the college has sufficient budget to develop the proposed system or is it cost effective.

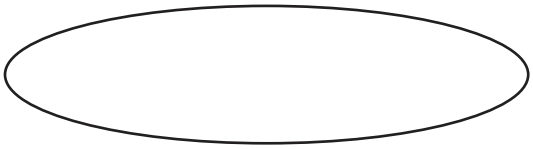
The system being developed is economic with respect to the public/agency point of view. It is cost effective in the sense that it has eliminated much of the paper work and for public no need of manually approaching/contacting the agencies for hiring the guards. The system is also time effective because the results are processed in quick time. The result obtained is highly accurate. As it is a common platform for both public and agency details of many agencies under one platform.

5.DATA FLOW DIAGRAM

Graphical description of system's data & how the processes transform the data is known as data flow diagram. It is also known as Bubble chart. DFD modules, system using external entities from which data flow to a process, which transforms the data, and create output data transforms which go to. Other processes or external entities like files. The main metric of DFD is that it can provide an overview of what data a system would process, what transformation of data are done, what files are used and where the result flows.

Notations Used In Data Flow Diagram :

Functional Processing: It is represented by an oval. The processing or main transactions are specified by this notation.

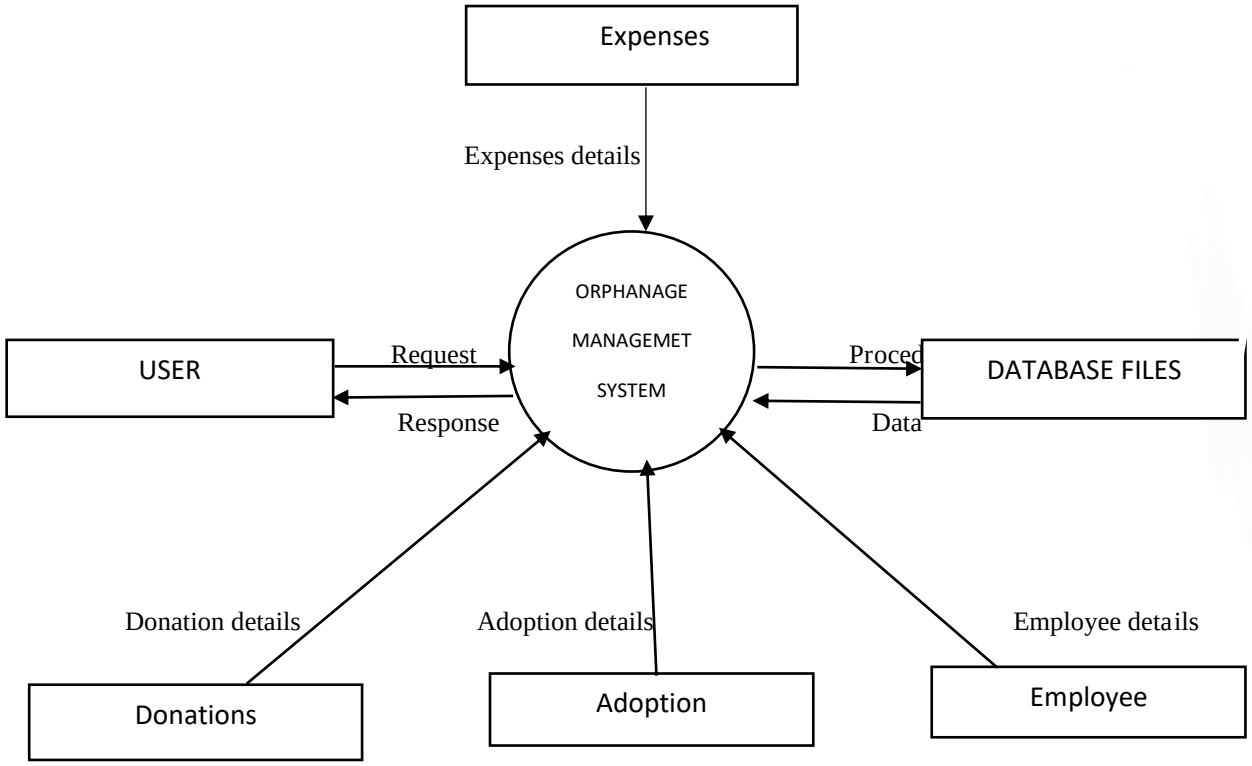


Data Flow: It is represented by an arrow line and name of the data is specified by the side of the line as label. This is used for data movement.

Data Store: It is represented by one open-end rectangle. The databases used in the system are specified by this notation.

Source or sink: It is represented by one open-end rectangle. It is used for specifying from where data comes and where it reaches.

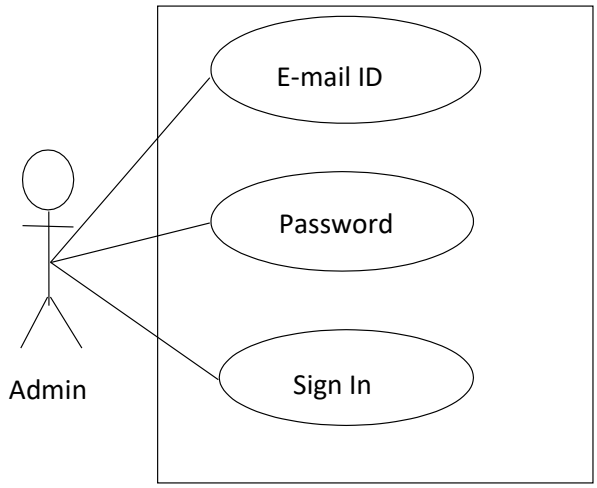
ZERO LEVEL DFD



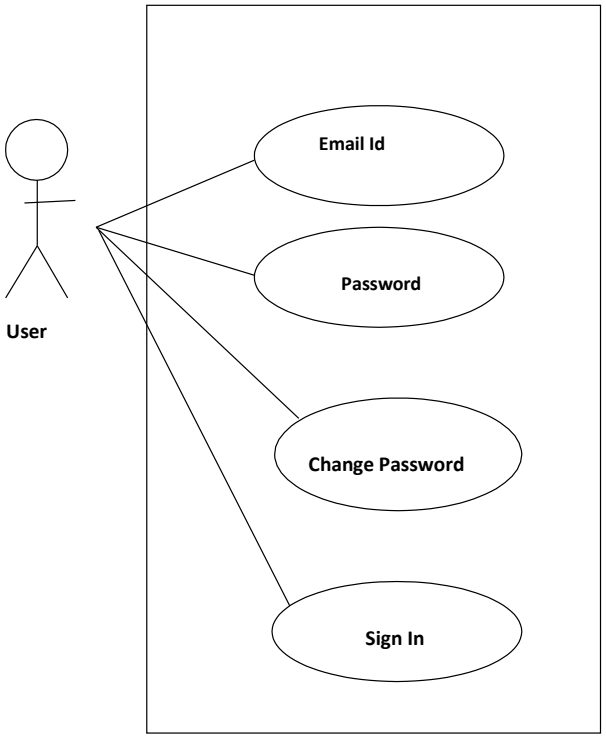
USE CASES DESCRIPTION:

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. Use case is a list of steps, typically defining interactions between a role (known in UML as an actor) and a system, to achieve a goal. The actor can be a human or an external system.

User case for Admin



User case for User



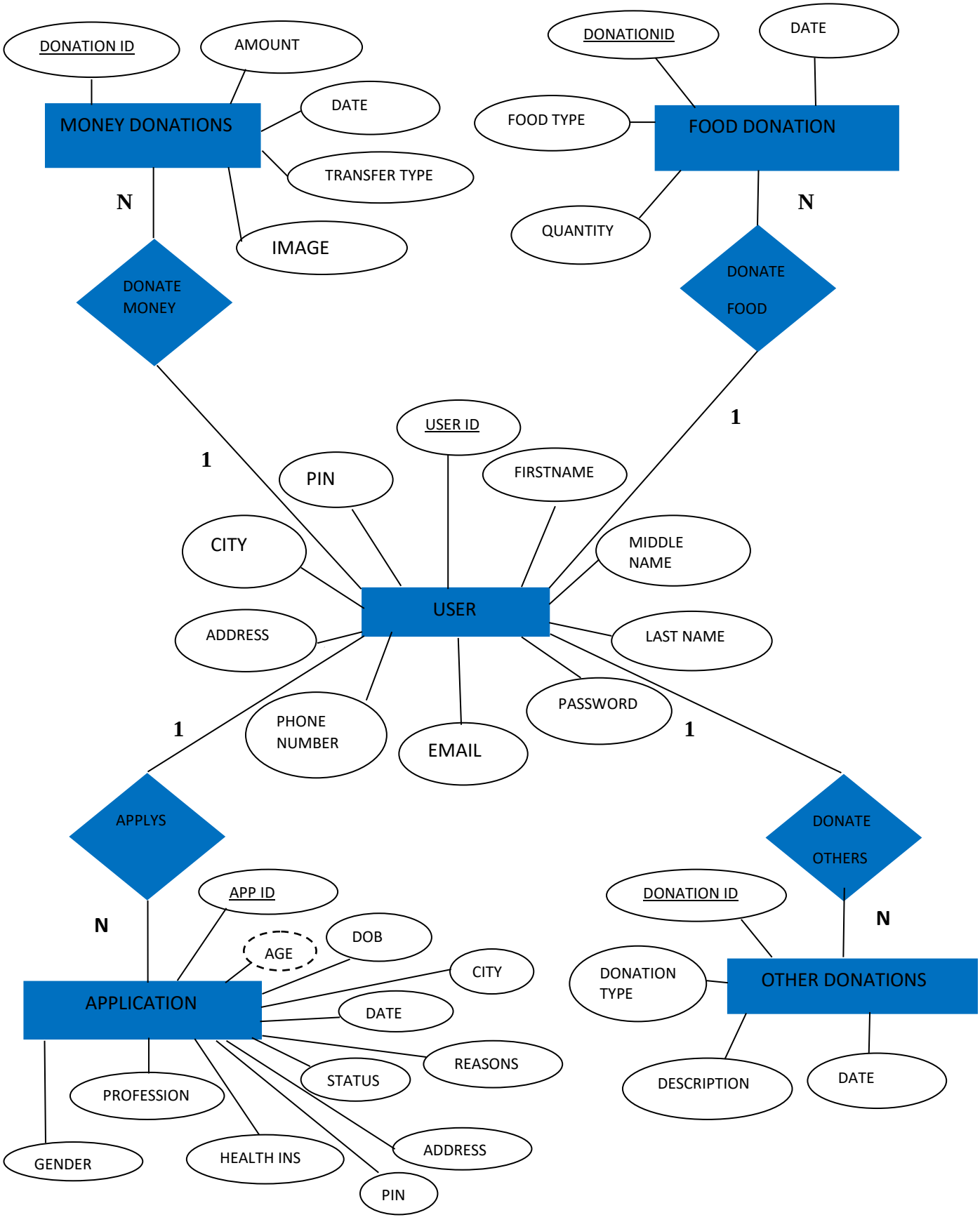
6.ENTITY-RELATIONSHIP MODEL

The entity relationship model is based on a perception of the world as consisting of a collection of basic objects (entities) and relationships among these objects.

- An entity is a distinguishable object that exists.
- Each entity has associated with it a set of attributes describing it.
- A relationship is an association among several entities .
- The set of entities or relationships of the same type is called the entity set relation set.
- Another important element of the E-R diagram is the mapping the cardinalities, which express the number of the entities to which another entity can be associated via a relation set.

The overall logical structure of a database can be expressed graphically by E-R diagram:

- Rectangle represent entity sets.
- Ellipses: represent attributes.
- Diamonds: represent relationships among entity sets.
- Lines: link attribute to entity sets and entity sets to relationships.


DONATION ID
AMOUNT
MONEY DONATIONS
DATE
TRANSFER TYPE
IMAGE
N
DONATE MONEY
DONATIONID
DATE
FOOD TYPE
FOOD DONATION
QUANTITY
N
DONATE FOOD
1
1
USER ID
PIN
FIRSTNAME
CITY
MIDDLE NAME
USER
ADDRESS
LAST NAME
PHONE NUMBER
EMAIL
PASSWORD
1
1
APPLYS
DONATE OTHERS
APP ID
N
DOB
AGE
CITY
APPLICATION
DATE
REASONS
STATUS
PROFESSION
GENDER
HEALTH INS
ADDRESS
PIN
DONATION ID
N
DONATION TYPE
OTHER DONATIONS
DESCRIPTION
DATE

7.TABLES USED IN DATA BASE

a) Admin

```
MariaDB [orphangedb]> desc admin;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| emailid  | varchar(50) | NO   | PRI | NULL    |       |
| password | varchar(10) | YES  |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
2 rows in set (0.015 sec)
```

b) Adopted child

```
MariaDB [orphangedb]> desc adoptedchild;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| adoptedid      | int(11)     | NO   | PRI | NULL    | auto_increment |
| childid        | int(11)     | YES  | MUL | NULL    |                |
| appid          | int(11)     | YES  | MUL | NULL    |                |
| age            | int(11)     | YES  |     | NULL    |                |
| guardianname   | varchar(10) | YES  |     | NULL    |                |
| email          | varchar(10) | YES  |     | NULL    |                |
| phonenumber    | varchar(13) | YES  |     | NULL    |                |
| address        | varchar(20) | YES  |     | NULL    |                |
| city           | varchar(20) | YES  |     | NULL    |                |
| pin            | varchar(6)  | YES  |     | NULL    |                |
| dateofadoption | date        | YES  |     | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
11 rows in set (0.015 sec)
```

c) Application

```
MariaDB [orphangedb]> desc application;
+-----------------+-------------+------+-----+---------+----------------+
| Field           | Type        | Null | Key | Default | Extra          |
+-----------------+-------------+------+-----+---------+----------------+
| Appid           | int(11)     | NO   | PRI | NULL    | auto_increment |
| userid          | int(11)     | YES  | MUL | NULL    |                |
| Birthdate       | date        | YES  |     | NULL    |                |
| Age             | int(11)     | YES  |     | NULL    |                |
| Appdate         | date        | YES  |     | NULL    |                |
| Status          | varchar(20) | YES  |     | NULL    |                |
| profession      | varchar(20) | YES  |     | NULL    |                |
| Address         | varchar(50) | YES  |     | NULL    |                |
| City            | varchar(10) | YES  |     | NULL    |                |
| pin             | varchar(6)  | YES  |     | NULL    |                |
| HealthInsurance | varchar(20) | YES  |     | NULL    |                |
| Reasons         | varchar(50) | YES  |     | NULL    |                |
| gender          | varchar(50) | YES  |     | NULL    |                |
| doc             | varchar(50) | YES  |     | NULL    |                |
+-----------------+-------------+------+-----+---------+----------------+
14 rows in set (0.013 sec)
```

d) Child photo

```
MariaDB [orphangedb]> desc childphoto;
+-----------+-------------+------+-----+---------+----------------+
| Field     | Type        | Null | Key | Default | Extra          |
+-----------+-------------+------+-----+---------+----------------+
| photoid   | int(11)     | NO   | PRI | NULL    | auto_increment |
| childid   | int(11)     | YES  | MUL | NULL    |                |
| image     | varchar(50) | YES  |     | NULL    |                |
| age       | int(11)     | YES  |     | NULL    |                |
| imagedate | date        | YES  |     | NULL    |                |
+-----------+-------------+------+-----+---------+----------------+
5 rows in set (0.013 sec)
```

e) Children

```
MariaDB [orphangedb]> desc children;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| childid      | int(11)     | NO   | PRI | NULL    | auto_increment |
| name         | varchar(20) | YES  |     | NULL    |                |
| birthdate    | date        | YES  |     | NULL    |                |
| age          | int(11)     | YES  |     | NULL    |                |
| gender       | varchar(10) | YES  |     | NULL    |                |
| fathername   | varchar(20) | YES  |     | NULL    |                |
| mothername   | varchar(20) | YES  |     | NULL    |                |
| addharnumber | varchar(12) | YES  |     | NULL    |                |
| education    | varchar(50) | YES  |     | NULL    |                |
| address      | varchar(20) | YES  |     | NULL    |                |
| city         | varchar(10) | YES  |     | NULL    |                |
| pin          | varchar(6)  | YES  |     | NULL    |                |
| Image        | varchar(50) | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
13 rows in set (0.013 sec)
```

f) Expenses

```
MariaDB [orphangedb]> desc expenses;
+--------------------+-------------+------+-----+---------+----------------+
| Field              | Type        | Null | Key | Default | Extra          |
+--------------------+-------------+------+-----+---------+----------------+
| expensesid         | int(11)     | NO   | PRI | NULL    | auto_increment |
| expensesdescrption | varchar(50) | YES  |     | NULL    |                |
| amount             | int(11)     | YES  |     | NULL    |                |
| expensesdate       | date        | YES  |     | NULL    |                |
| expensestype       | varchar(20) | YES  |     | NULL    |                |
+--------------------+-------------+------+-----+---------+----------------+
5 rows in set (0.013 sec)
```

g) Food Donations

```
MariaDB [orphangedb]> desc fooddonations;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| donationid     | int(11)     | NO   | PRI | NULL    | auto_increment |
| userid         | int(11)     | YES  | MUL | NULL    |                |
| foodtype       | varchar(20) | YES  |     | NULL    |                |
| quantity       | varchar(20) | YES  |     | NULL    |                |
| dateofdonation | date        | YES  |     | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
5 rows in set (0.015 sec)
```

h) Money Donations

```
MariaDB [orphangedb]> desc moneydonations;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| donationid   | int(11)     | NO   | PRI | NULL    | auto_increment |
| userid       | int(11)     | YES  | MUL | NULL    |                |
| amount       | int(11)     | YES  |     | NULL    |                |
| donateddate  | date        | YES  |     | NULL    |                |
| transfertype | varchar(20) | YES  |     | NULL    |                |
| image        | varchar(20) | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
6 rows in set (0.013 sec)
```

i) Other Donations

```
MariaDB [orphangedb]> desc otherdonations;
+--------------+-------------+------+-----+---------+----------------+
| Field        | Type        | Null | Key | Default | Extra          |
+--------------+-------------+------+-----+---------+----------------+
| donateid     | int(11)     | NO   | PRI | NULL    | auto_increment |
| Userid       | int(11)     | YES  | MUL | NULL    |                |
| donationtype | varchar(20) | YES  |     | NULL    |                |
| description  | varchar(50) | YES  |     | NULL    |                |
| donateddate  | date        | YES  |     | NULL    |                |
+--------------+-------------+------+-----+---------+----------------+
5 rows in set (0.013 sec)
```

j) Staff

```
MariaDB [orphangedb]> desc staff;
+----------------+-------------+------+-----+---------+----------------+
| Field          | Type        | Null | Key | Default | Extra          |
+----------------+-------------+------+-----+---------+----------------+
| staffid        | int(11)     | NO   | PRI | NULL    | auto_increment |
| firstname      | varchar(20) | YES  |     | NULL    |                |
| middlename     | varchar(20) | YES  |     | NULL    |                |
| lastname       | varchar(20) | YES  |     | NULL    |                |
| phonenumber    | varchar(13) | YES  |     | NULL    |                |
| email          | varchar(20) | YES  |     | NULL    |                |
| responsibility | varchar(20) | YES  |     | NULL    |                |
| salary         | int(11)     | YES  |     | NULL    |                |
| address        | varchar(50) | YES  |     | NULL    |                |
| city           | varchar(20) | YES  |     | NULL    |                |
| pin            | varchar(6)  | YES  |     | NULL    |                |
+----------------+-------------+------+-----+---------+----------------+
11 rows in set (0.014 sec)
```

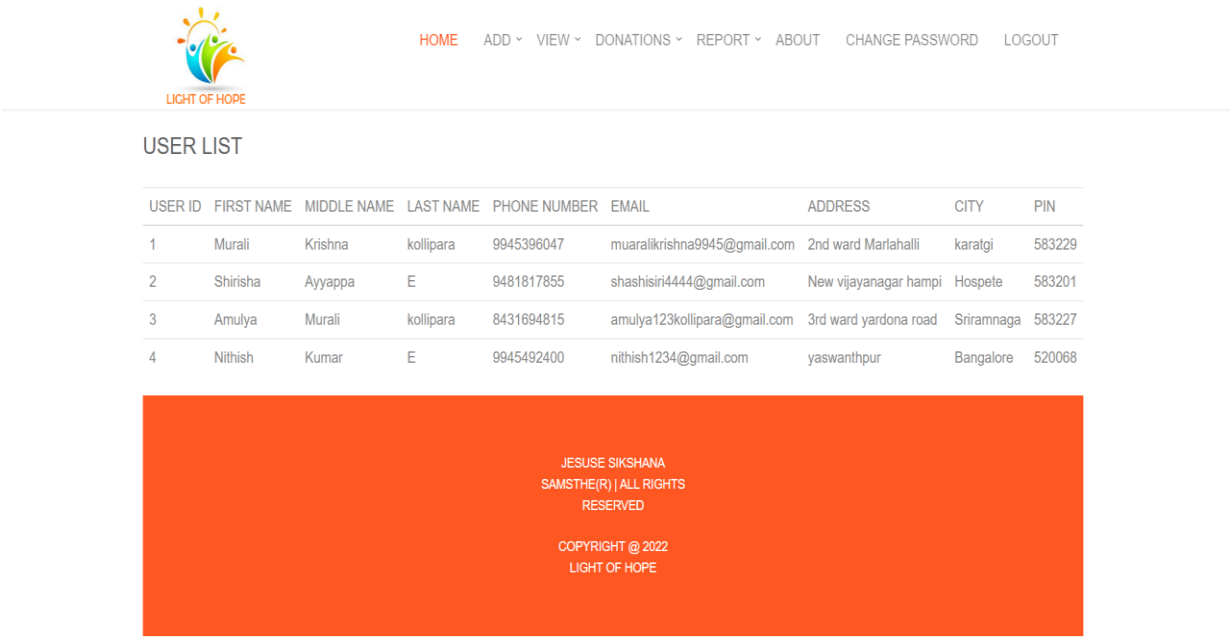
k) User

```
MariaDB [orphangedb]> desc user;
+-------------+-------------+------+-----+---------+----------------+
| Field       | Type        | Null | Key | Default | Extra          |
+-------------+-------------+------+-----+---------+----------------+
| userid      | int(11)     | NO   | PRI | NULL    | auto_increment |
| firstname   | varchar(20) | YES  |     | NULL    |                |
| middlename  | varchar(20) | YES  |     | NULL    |                |
| lastname    | varchar(20) | YES  |     | NULL    |                |
| phonenumber | varchar(10) | YES  |     | NULL    |                |
| email       | varchar(30) | YES  |     | NULL    |                |
| password    | varchar(10) | YES  |     | NULL    |                |
| address     | varchar(50) | YES  |     | NULL    |                |
| city        | varchar(20) | YES  |     | NULL    |                |
| pin         | varchar(6)  | YES  |     | NULL    |                |
+-------------+-------------+------+-----+---------+----------------+
10 rows in set (0.015 sec)
```

8.SAMPLE OUTPUT/FORM/SCREENSHOTS

1. Home page

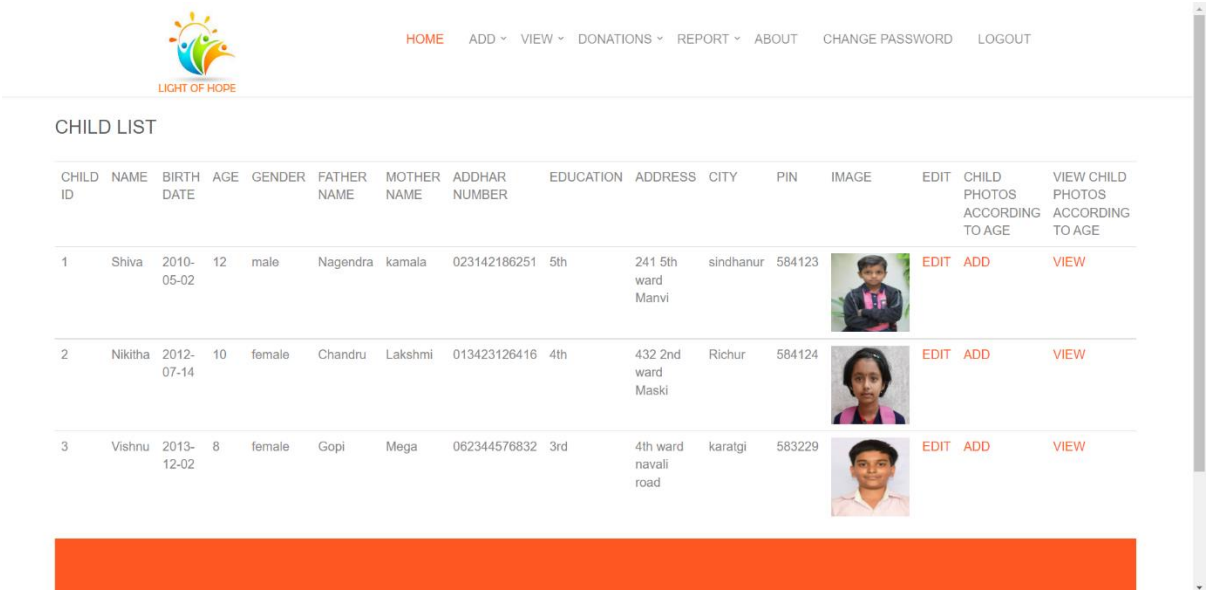
2. User details



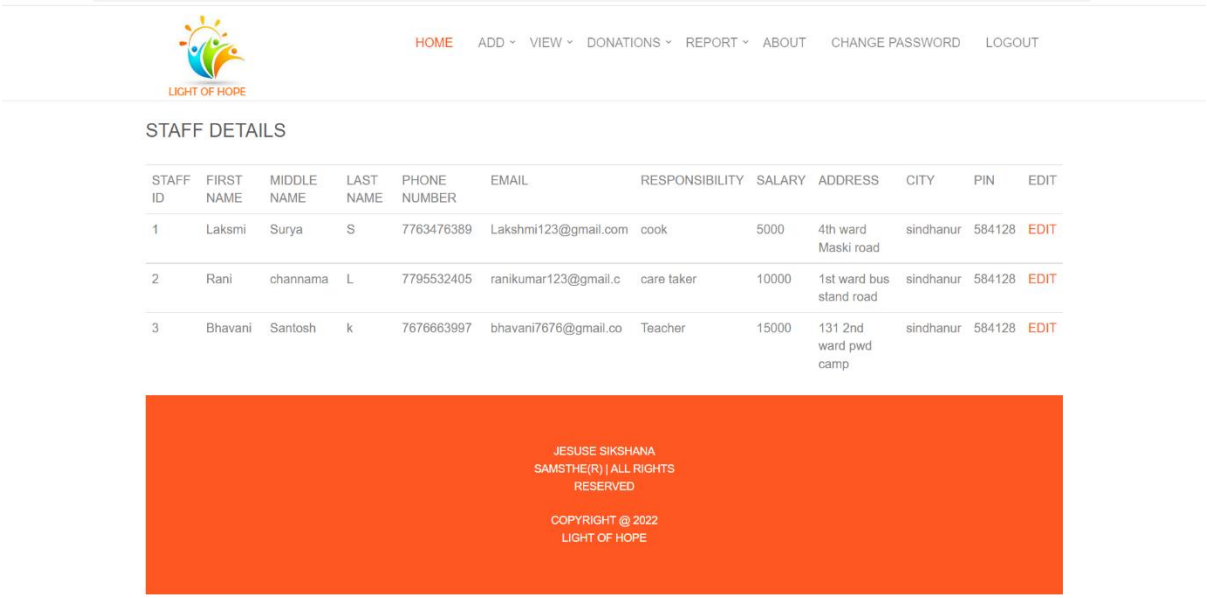
3. Login Home:

4. Admin login

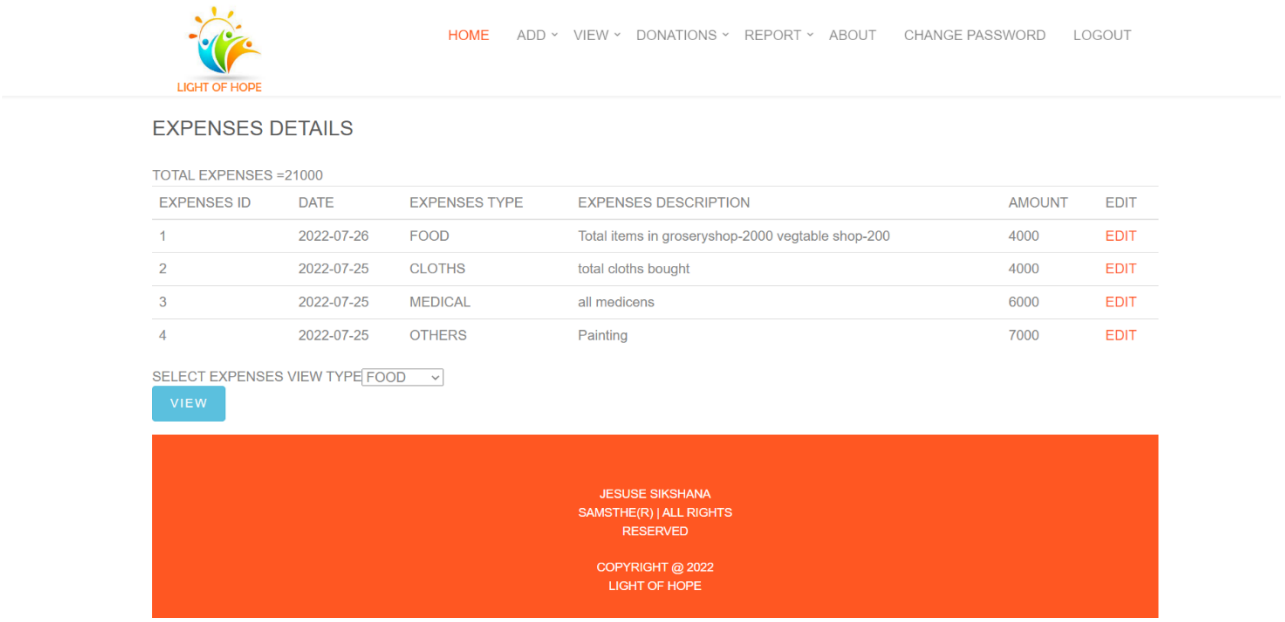
5. View child



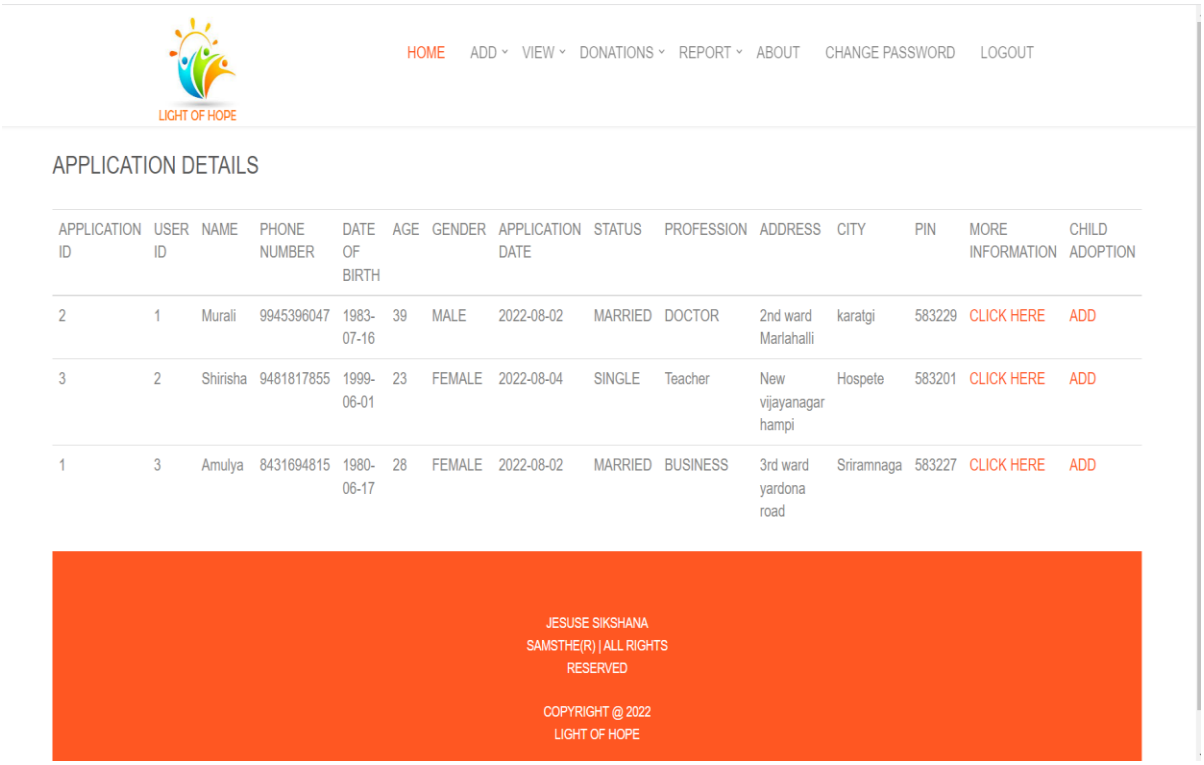
6. View Staff



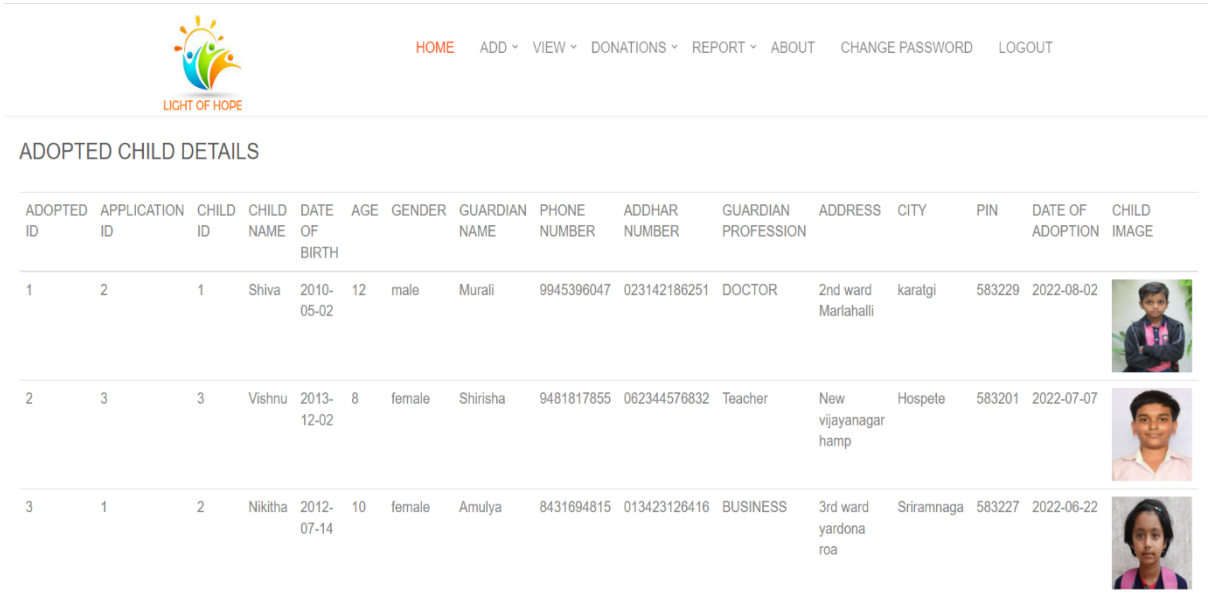
7. Expenses Details



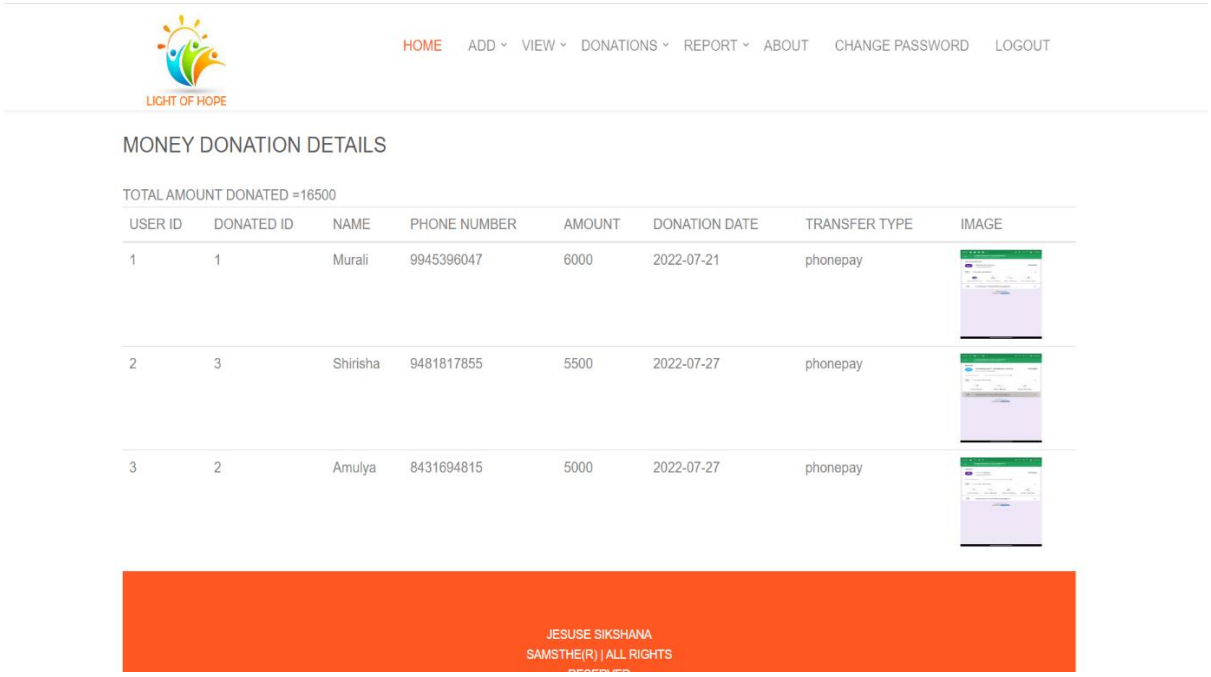
8. Application Details



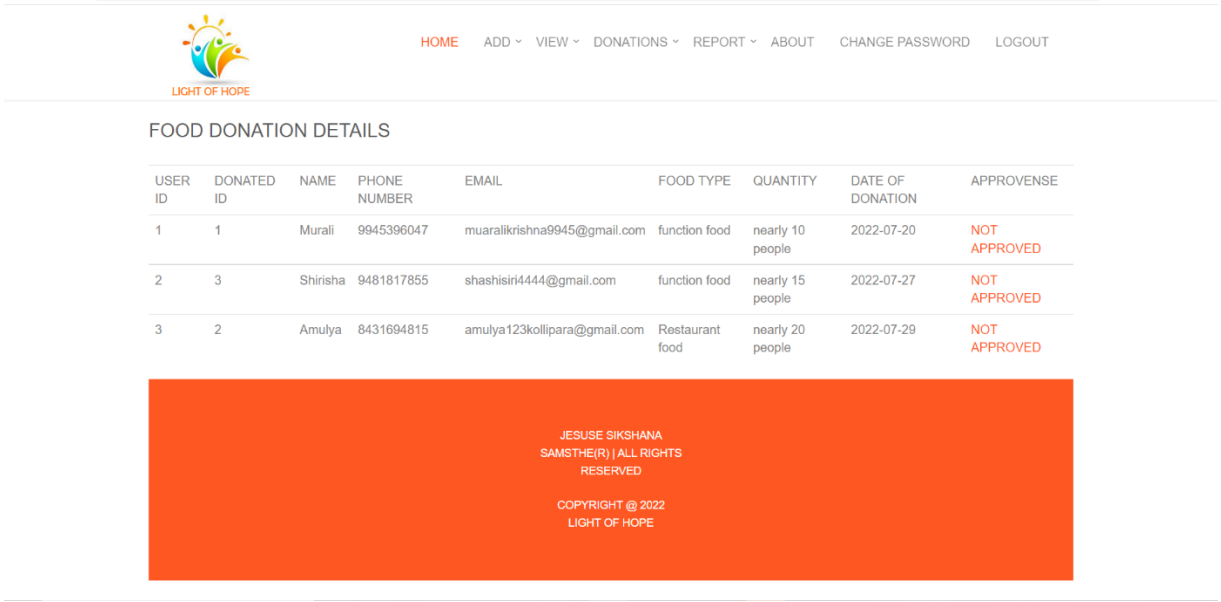
9. Adopted Child Details



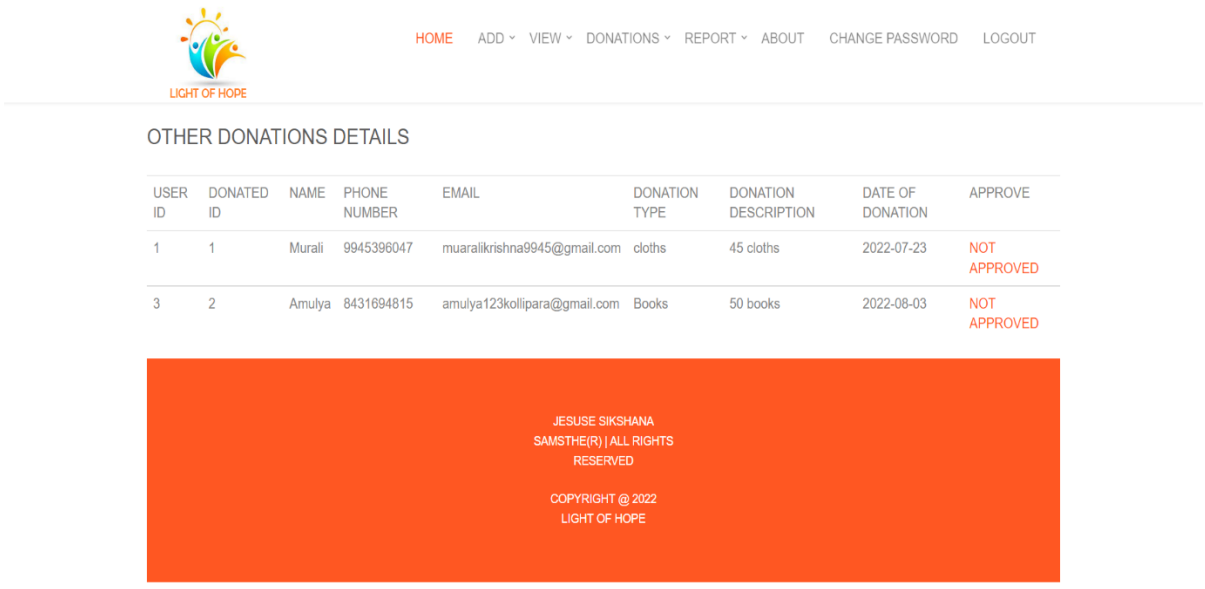
10. View Money Donation Details



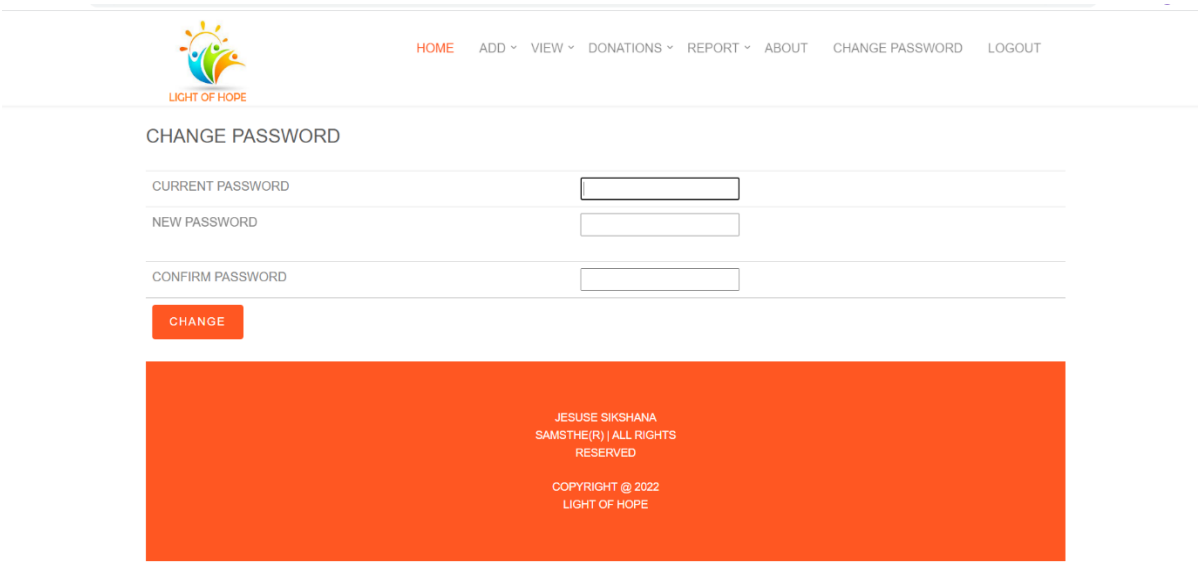
11. View Food Donation Details



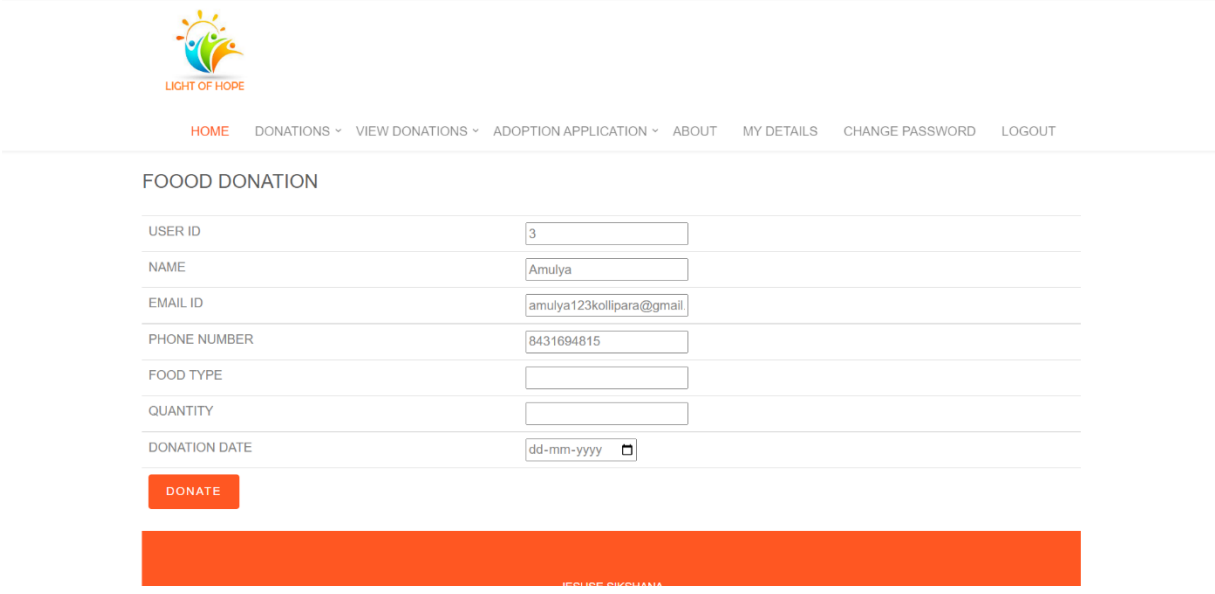
12. View Other Donation Details



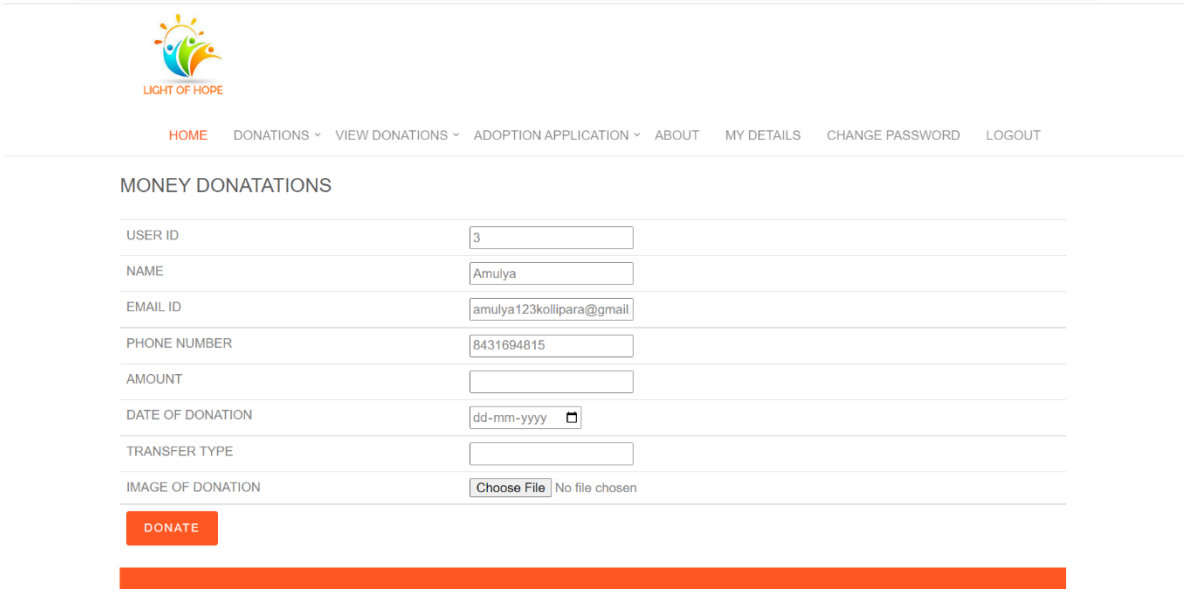
13.Change Password



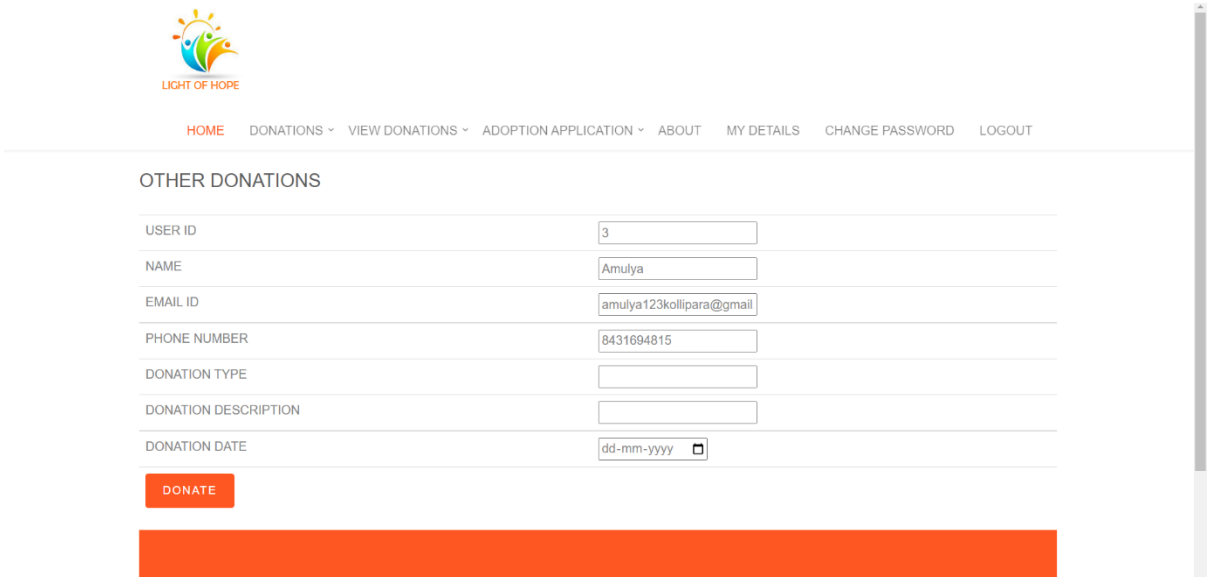
14. Food Donations



15.Money Donations

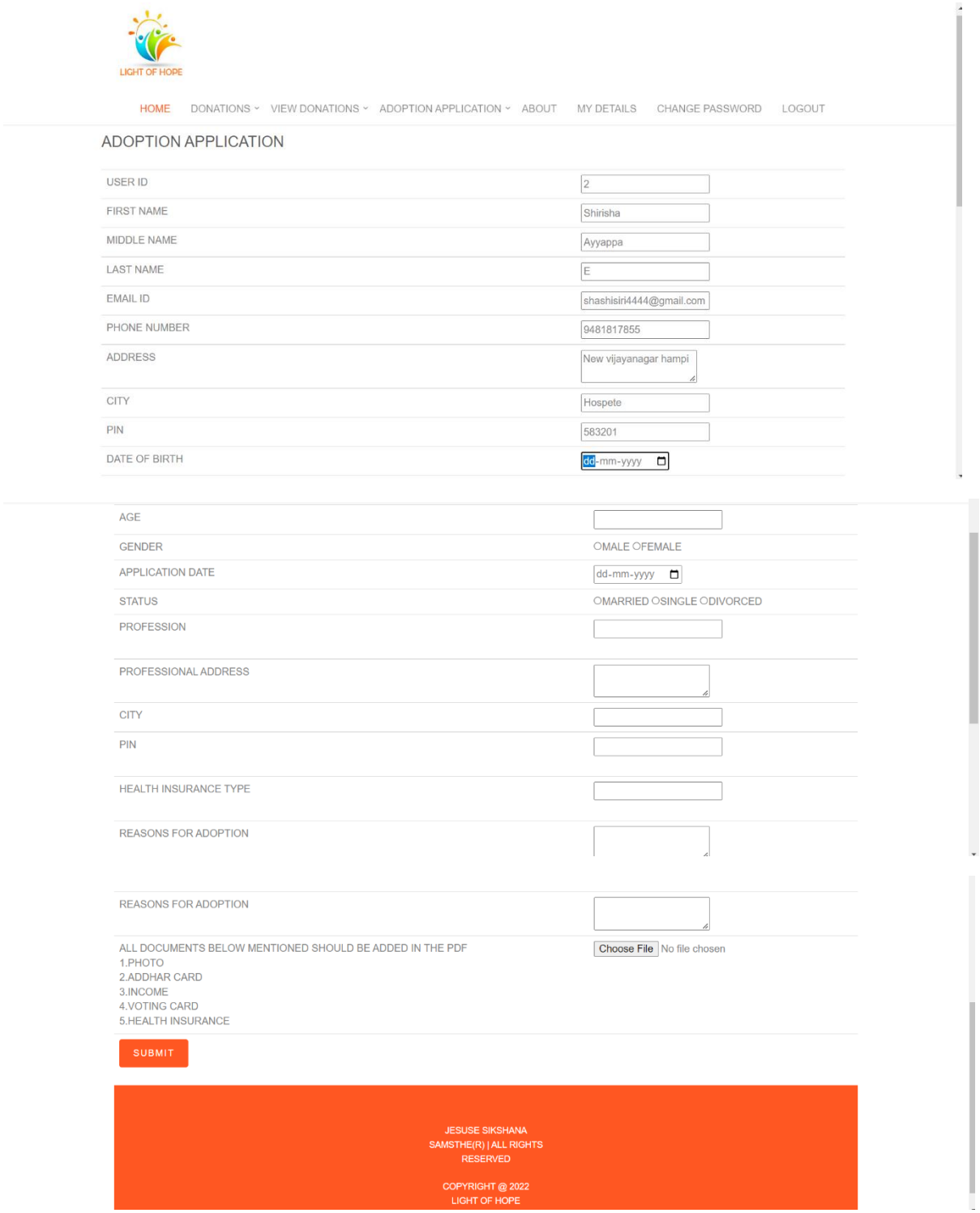


16.Other Donations



17.User Sign Up

18.Application



9. SOURCE CODE:

ADMIN SIDE

SOURCE CODE: SOURCE CODE OF ADMIN SIGNIN PAGE:

```
<?php
session_start();
//fetch input

$email=$_POST["textemail"];
$pwd=$_POST["textpassword"];
$msg=null;
//open conn
$conn=new PDO("mysql:host=localhost;dbname=orphangedb","root",null);
$conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);


try{
   //build the stmt to check
   $stmt=$conn->prepare("select * from user where  email=? and password=?");

   $stmt->bindParam(1,$email);
   $stmt->bindParam(2,$pwd);
   $stmt->execute();
   $c=$stmt->rowCount();

   //if found then goto adminhome.html else error msg
   if($c==1){
      //fetch customer code from row
      while($row=$stmt->fetch(PDO::FETCH_ASSOC))
      {
         $userid=$row["userid"];
      }
      //store admin details in session
      $_SESSION["userid"]=$userid;
      $_SESSION["email"]=$email;
      $_SESSION["password"]=$pwd;
      header('location:user/userhome.php');
```

```
   }
   else{
      $msg="Invalid login";
   }
}catch(Exception $e){
   $msg="Invalid login,".$e->getMessage();

}
?>
<html>
   <head>
      <title>Signin</title>
      <?php
       include("loginlink.php");
      ?>
   </head>
   <body>
       <?php
      include('header.php')
      ?>
      <table align="center">
      <tr>
      <td style="font-size:50px;"><?php
                echo $msg;

     ?></td>
      </tr>
      <tr>

<td> </td>
</tr>
      <tr>
            <td>
            <div class="group">
              <a href="usersigninform.php"><input type="button" class="button" value="SIGN
IN AGAIN"/></a>

            </div>
            </td>
         </tr>
      </table>
      <?php
      include('footer.php')
```

```
    ?>    </body>    </html>
```

SOURCE CODE TO ADD CHILD

```
<?php
session_start();
$name=$_POST["textname"];
$birthdate=$_POST["textcal"];
$age=$_POST["textage"];
$gender=$_POST["gender"];
$father=$_POST["textfather"];
$mother=$_POST["textmother"];
$addhar=$_POST["textadhar"];
$education=$_POST["texteducation"];

$address=$_POST["textaddress"];
$city=$_POST["textcity"];
$pin=$_POST["textpin"];
$status=null;

try{
   if(isset($_FILES["file1"]["type"])){
      $validextensions=array("jpeg","jpg","png");
      //split file extension and store into $temporary
      $temporary=explode(".",$_FILES["file1"]["name"]);
      //get file extension from $temporary variable
      $file_extension=end($temporary);
      //check the mime type provided by the browser
      if((($_FILES["file1"]["type"]=="image/png")
      ||($_FILES["file1"]["type"]=="image/jpg")
      ||($_FILES["file1"]["type"]=="image/jpeg"))
      &&($_FILES["file1"]["size"]<500000)&& in_array($file_extension,$validextensions)){
      //if file was not uploaded correctly or partially uploaded,returns 0 if valid
      if($_FILES["file1"]["error"]>0)
            $msg=$_FILES["file1"]["error"];
      //check is file is alerady uploaded
      else if(file_exists("../photo/".$_FILES["file1"]["name"]))
           $msg="THIS FILE ALREADY EXITS .";
      else{
         $sourcePath=$_FILES['file1']['tmp_name'];
         $targetPath="../photo/".$_FILES['file1']['name'];
         move_uploaded_file($sourcePath,$targetPath);
```

```
            $photos=$targetPath;
            $status="ok";
        }
    }else{
        $msg="INVALID  FILE SIZE OR TYPE";
    }


    }else{
        $msg="PLEASE SELECT FILE";
    }
    if($status=="ok"){

        $conn=new PDO("mysql:host=localhost;dbname=orphangedb","root",null);
        $conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
        $stmt=$conn->prepare("insert into
children(name,birthdate,age,gender,fathername,mothername,addharnumber,education,address
,city,pin,Image)values(?,?,?,?,?,?,?,?,?,?,?,?)");
        $stmt->bindParam(1,$name);
        $stmt->bindParam(2,$birthdate);
        $stmt->bindParam(3,$age);
        $stmt->bindParam(4,$gender);
        $stmt->bindParam(5,$father);
        $stmt->bindParam(6,$mother);
        $stmt->bindParam(7,$addhar);
        $stmt->bindParam(8,$education);
        $stmt->bindParam(9,$address);
        $stmt->bindParam(10,$city);
        $stmt->bindParam(11,$pin);
        $stmt->bindParam(12,$photos);
        $stmt->execute();
        $msg="CHILD ADD";
    }
}catch(Exception $e){
    $msg="CHILD NOT ADD".$e->getMessage();
}finally{
    $conn=null;
}


        if(empty($_SESSION))
        {
            header('location:../adminsigninform.php');
        }
```

```
      ?>

<html>
   <head>
      <title>ADD CHILD INFO</title>
      <link rel="stylesheet" href="styless.css">
      <?php
       include("headerlink.php");
      ?>

   </head>
   <body>
      <div class="container">
         <div class="item">
         <?php
          include('header.php');
         ?>
       <br>
      </div>
      <div class="item" align=center>
        <h4><?php echo $msg; ?></h4>
      </div>
      </div>
      <?php
       if(empty($_SESSION))
       {
          header('location:../adminsigninform.php');
       }
       ?>
       <?php
           include('footer.php');
         ?>
   </body>
</html>
```

SOURCE CODE TO VIEW A CHILD

```
<?php
$conn=new PDO("mysql:host=localhost;dbname=orphangedb","root",null);
$conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);

//crete arrays
$childid=array();
$name=array();
```

```
$birthdate=array();
$age=array();
$gender=array();
$fathername=array();
$mothername=array();
$addhar=array();
$education=array();

$address=array();
$city=array();
$pin=array();
$image=array();

//prepare select statements
$stmt=$conn->prepare("select * from children");
$stmt->execute();

//push rows into arrays
while($row = $stmt->fetch(PDO::FETCH_ASSOC)){
   array_push($childid,$row["childid"]);
   array_push($name,$row["name"]);
   array_push($birthdate,$row["birthdate"]);
   array_push($age,$row["age"]);
   array_push($gender,$row["gender"]);
   array_push($fathername,$row["fathername"]);
   array_push($mothername,$row["mothername"]);
   array_push($addhar,$row["addharnumber"]);
   array_push($education,$row["education"]);

   array_push($address,$row["address"]);
   array_push($city,$row["city"]);
   array_push($pin,$row["pin"]);
   array_push($image,$row["Image"]);

}
$conn=null;
?>
<html>
   <head>
      <title>VIEW products</title>
      <link rel="stylesheet" href="styless.css"/>
      <?php
      session_start();
        include("headerlink.php");
```

```
    ?>

</head>
<body>

   <div class="container">
      <div class="item">
         <?php
         include('header.php');
         ?>

      </div>
      <?php
    if(empty($_SESSION))
    {
       header('location:../adminsigninform.php');
    }
    ?>
    <br>
      <div class="item">
   <h3>CHILD LIST</h3>
      <table class="table">
         <tr>
            <td>CHILD ID</td>
            <td>NAME</td>
            <td>BIRTH DATE</td>
            <td>AGE</td>
            <td>GENDER</td>
            <td>FATHER NAME</td>
            <td>MOTHER NAME</td>
            <td>ADDHAR NUMBER</td>
            <td>EDUCATION</td>
            <td>ADDRESS</td>
            <td>CITY</td>
            <td>PIN</td>
            <td>IMAGE</td>
            <td>EDIT</td>
            <td>CHILD PHOTOS ACCORDING TO AGE</td>
            <td>VIEW CHILD PHOTOS ACCORDING TO AGE</td>
         </tr>
         <?php
         $len=count($childid);
         for($i=0;$i<$len;$i++)
         {
```

```
                echo "<tr>";
                echo "<td>".$childid[$i]."</td>";
                $cname=urlencode($name[$i]);
                echo "<td>".$name[$i]."</td>";
                echo "<td>".$birthdate[$i]."</td>";
                echo "<td>".$age[$i]."</td>";
                echo "<td>".$gender[$i]."</td>";
                echo "<td>".$fathername[$i]."</td>";
                $father=urlencode($fathername[$i]);
                echo "<td>".$mothername[$i]."</td>";
                $mother=urlencode($mothername[$i]);
                echo "<td>".$addhar[$i]."</td>";
                echo "<td>".$education[$i]."</td>";
                $edu=urlencode($education[$i]);

                echo "<td>".$address[$i]."</td>";
                $add=urlencode($address[$i]);
                echo "<td>".$city[$i]."</td>";
                echo "<td>".$pin[$i]."</td>";
                echo "<td><img src=".$image[$i]." height=100 width=100/></td>";
                echo "<td><a
href=editchildform.php?childid=$childid[$i]&name=$cname&birthdate=$birthdate[$i]&age
=$age[$i]&gender=$gender[$i]&fathername=$father&mothername=$mother&city=$city[$i]
&pin=$pin[$i]&address=$add&education=$edu&addhar=$addhar[$i]>EDIT</a></td>";
                echo "<td><a
href=addphotosform.php?childid=$childid[$i]&name=$cname&age=$age[$i]>ADD</a></td
>";
                echo "<td><a
href=viewchildphotos.php?childid=$childid[$i]&name=$cname>VIEW</a></td>";
                echo "</tr>";

            }


            ?>
        </table>
    </div>
    </div>
    <?php
        include('footer.php');
        ?>

  </body>
</html>
```

SOURCE CODE TO ADD ADOPTED CHILD

```
<?php
session_start();
$childid=$_POST["textid"];
$childname=$_POST["textname"];
$age=$_POST["textage"];
$appid=$_POST["textappid"];
$guardianname=$_POST["textfname"];
$email=$_POST["textemail"];
$phone=$_POST["textphone"];
$address=$_POST["textadd"];
$city=$_POST["textcity"];
$pin=$_POST["textpin"];
$dateofadoption=$_POST["textdate"];
$msg=null;

//connection
$conn=new PDO("mysql:host=localhost;dbname=orphangedb","root",null);
$conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);

//3.prepare statment to insert data into the category table
try{
   $stmt=$conn->prepare("insert into adoptedchild
(childid,appid,age,guardianname,email,phonenumber,address,city,pin,dateofadoption) values
(?,?,?,?,?,?,?,?,?,?)");
   $stmt->bindParam(1,$childid);
   $stmt->bindParam(2,$appid);
   $stmt->bindParam(3,$age);
   $stmt->bindParam(4,$guardianname);
   $stmt->bindParam(5,$email);
   $stmt->bindParam(6,$phone);
   $stmt->bindParam(7,$address);
   $stmt->bindParam(8,$city);
   $stmt->bindParam(9,$pin);
   $stmt->bindParam(10,$dateofadoption);
   $stmt->execute();
   $c=$stmt->rowCount();
   if($c>0){
      $msg="ADOPTION DETAILS ADDED";
   }
```

```
   else{
      $msg="ADOPTION DETAILS NOT ADDED";
   }
}catch(Exception $e){
  $msg=$e->getMessage();
}
//4.clos ethe connection to datebase
finally{

  $conn=null;
}
```

```
        if(empty($_SESSION))
        {
           header('location:../adminsigninform.php');
        }
        ?>

<html>
   <head>
      <title>adoption add</title>
      <?php
        include("headerlink.php");
      ?>
   </head>
   <body>
       <?php
       include('header.php')
       ?>
       <br>
       <div class="item" align="center">
       <h4><?php

          echo $msg;

      ?></h4>
       </div>
      <?php
       if(empty($_SESSION))
       {
          header('location:../adminsigninform.php');
       }
       ?>
```

```
<?php
include('footer.php')        ?>    </body>    </html>
```

SOURCE CODE TO VIEW APPLICATION

```
<?php

//set a connections
$conn=new PDO("mysql:host=localhost;dbname=orphangedb","root",null);
$conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);

//crete arrays
$firstname=array();
$phonenumber=array();
$email=array();
$address=array();
$city=array();
$pin=array();
$userid=array();
$birthdate=array();
$age=array();
$gender=array();
$appdate=array();
$status=array();
$profession=array();
$appid=array();

//prepare select statements
$stmt=$conn->prepare("select
application.userid,Appid,Birthdate,Age,Appdate,Status,profession,application.Address,applic
ation.City,application.Pin,HealthInsurance,Reasons,gender,firstname,middlename,lastname,p
honenumber,email,user.address,user.city,user.pin from application inner join user on
user.userid=application.userid;");
$stmt->execute();

//push rows into arrays
while($row = $stmt->fetch(PDO::FETCH_ASSOC)){
   array_push($appid,$row["Appid"]);
   array_push($userid,$row["userid"]);
   array_push($firstname,$row["firstname"]);
   array_push($phonenumber,$row["phonenumber"]);
   array_push($email,$row["email"]);
   array_push($birthdate,$row["Birthdate"]);
```

```
    array_push($age,$row["Age"]);
    array_push($gender,$row["gender"]);
    array_push($appdate,$row["Appdate"]);
    array_push($status,$row["Status"]);
    array_push($profession,$row["profession"]);
    array_push($address,$row["address"]);
    array_push($city,$row["city"]);
    array_push($pin,$row["pin"]);



}
$conn=null;
?>
<html>
    <head>
        <title>VIEW products</title>
        <link rel="stylesheet" href="styless.css"/>
        <?php

          include("headerlink.php");
         ?>

    </head>
    <body>

        <div class="container">
            <div class="item">
                <?php
                include('header.php');
                ?>
            <br>
            </div>
            <div class="item">
        <h3>APPLICATION DETAILS</h3>
            <table class="table">
                <tr>
                    <td>APPLICATION ID</td>
                    <td>USER ID</td>
                    <td>NAME</td>
                    <td>PHONE NUMBER</td>

                    <td>DATE OF BIRTH</td>
                    <td>AGE</td>
```

```
            <td>GENDER</td>
            <td>APPLICATION DATE</td>
            <td>STATUS</td>
            <td>PROFESSION</td>
            <td>ADDRESS</td>
            <td>CITY</td>
            <td>PIN</td>
            <td>MORE INFORMATION</td>
            <td>CHILD ADOPTION</td>


        </tr>
        <?php
        $len=count($appid);
        for($i=0;$i<$len;$i++)
        {
            echo "<tr>";
            echo "<td>".$appid[$i]."</td>";
            echo "<td>".$userid[$i]."</td>";
            echo "<td>".$firstname[$i]."</td>";
            echo "<td>".$phonenumber[$i]."</td>";

            echo "<td>".$birthdate[$i]."</td>";
            echo "<td>".$age[$i]."</td>";
            echo "<td>".$gender[$i]."</td>";
            echo "<td>".$appdate[$i]."</td>";
            echo "<td>".$status[$i]."</td>";
            echo "<td>".$profession[$i]."</td>";
            echo "<td>".$address[$i]."</td>";
            $add=urlencode($address[$i]);
            echo "<td>".$city[$i]."</td>";
            echo "<td>".$pin[$i]."</td>";
            echo "<td><a href=moreinformation.php?appid=$appid[$i]>CLICK
HERE</a></td>";
            echo "<td><a
href=viewchild.php?appid=$appid[$i]&name=$firstname[$i]&userid=$userid[$i]&email=$e
mail[$i]&phonenumber=$phonenumber[$i]&address=$add&city=$city[$i]&pin=$pin[$i]>A
DD</a></td>";

            echo "</tr>";



        }
```

```
            ?>
          </table>
       </div>
       </div>
       <?php
            include('footer.php');
            ?>
   </body>
</html>
```

SOURCE CODE FOR REPORT

```
<?php
$from=$_POST["textdate1"];
$to=$_POST["textdate2"];
$total=0;
$firstname=array();
$phonenumber=array();
$email=array();
$userid=array();
$donationid=array();
$amount=array();
$transfertype=array();
$image=array();

//set a connections
$conn=new PDO("mysql:host=localhost;dbname=orphangedb","root",null);
$conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);


//prepare select statements
$stmt=$conn->prepare("select
firstname,email,moneydonations.userid,donationid,amount,transfertype,image,phonenumber
from moneydonations inner join user on user.userid=moneydonations.userid where
donateddate between ? and ?");
$stmt->bindParam(1,$from);
$stmt->bindParam(2,$to);
$stmt->execute();

//push rows into arrays
while($row = $stmt->fetch(PDO::FETCH_ASSOC)){
   array_push($firstname,$row["firstname"]);
   array_push($email,$row["email"]);
   array_push($userid,$row["userid"]);
```

```
   array_push($phonenumber,$row["phonenumber"]);
   array_push($donationid,$row["donationid"]);
   array_push($amount,$row["amount"]);
   array_push($transfertype,$row["transfertype"]);
   array_push($image,$row["image"]);


}

$conn=null;
?>
<html>
   <head>
      <title>VIEW REPORT DETAILS</title>
      <link rel="stylesheet" href="styless.css"/>
      <?php
      session_start();
       include("headerlink.php");
       ?>

   </head>
   <body>
   <?php
       if(empty($_SESSION))
       {
          header('location:../adminsigninform.php');
       }
       ?>
      <div class="container">
         <div class="item">
            <?php
            include('header.php');
            ?>
         <br>
         </div>
         <div class="item">
      <h3>TOTAL AMOUNT SUMMARY</h3>
         <table class="table">
         <tr>

               <td>USER ID</td>
               <td>DONATED ID</td>
               <td>NAME</td>
               <td>PHONE NUMBER</td>
```

```
            <td>EMAIL</td>
            <td>AMOUNT</td>
            <td>TRANSFER TYPE</td>
            <td>IMAGE</td>
        </tr>
        <?php
            $len=count($donationid);
            for($i=0;$i<$len;$i++)
            {
                echo "<tr>";

                echo "<td>".$userid[$i]."</td>";
                echo "<td>".$donationid[$i]."</td>";
                echo "<td>".$firstname[$i]."</td>";
                echo "<td>".$phonenumber[$i]."</td>";
                echo "<td>".$email[$i]."</td>";
                echo "<td>".$amount[$i]."</td>";
                echo "<td>".$transfertype[$i]."</td>";
                echo "<td><img src=".$image[$i]." height=100 width=100/></td>";
                $total=$total+$amount[$i];

            }

            ?>
        <tr>

        <td>TOTAL AMOUNT OF DONATIONS</td>
        <td><?php echo $total; ?></td>
        </tr>
        </table>
      </div>
      </div>
      <?php
       include('footer.php');
       ?>
   </body>
</html>
```

USER SIDE

SOURCE CODE FOR USER SIGNUP

```
<?php
//fetch from html
$firstname=$_POST["textfname"];
$middlename=$_POST["textmname"];
$lastname=$_POST["textlname"];
$phone=$_POST["textphone"];
$email=$_POST["textemail"];
$new=$_POST["textpasword"];
$conform=$_POST["textcpassword"];
$add=$_POST["textarea"];
$city=$_POST["textcity"];
$pin=$_POST["textpin"];
$msg=null;
if($new==$conform)
{

//connection
$conn=new PDO("mysql:host=localhost;dbname=orphangedb","root",null);
$conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);

//3.prepare statment to insert data into the category table
try{
   $stmt=$conn->prepare("insert into
user(firstname,middlename,lastname,email,password,address,city,pin,phonenumber) values
(?,?,?,?,?,?,?,?,?)");
   $stmt->bindParam(1,$firstname);
   $stmt->bindParam(2,$middlename);
   $stmt->bindParam(3,$lastname);
   $stmt->bindParam(4,$email);
   $stmt->bindParam(5,$new);
   $stmt->bindParam(6,$add);
   $stmt->bindParam(7,$city);
   $stmt->bindParam(8,$pin);
   $stmt->bindParam(9,$phone);
   $stmt->execute();
```

```
   $c=$stmt->rowCount();
   if($c>0){
      $msg="sign up sucessfull";
   }
   else{
      $msg="sign up unsucessful";
   }
}catch(Exception $e){
 $msg=$e->getMessage();
}
//4.clos ethe connection to datebase
finally{

 $conn=null;
}
}
else{
   $msg="NEW  AND CONFORM PASSWORD DO NOT MATCH";

}

?>
<html>
   <head>
      <title>Signin</title>
      <?php
       include('loginlink.php')
       ?>
   </head>
   <body>
       <?php
       include('header.php')
       ?>
      <table align="center">
       <tr>
          <td style="font: size 50px;"><?php

          echo $msg;
      ?></td>
       </tr>
       <tr>

<td> </td>
</tr>
```

```
        <tr>
            <td>
            <div class="group">
              <a href="loginhome.php"><input type="button" class="button"
value="BACK"/></a>

            </div>
            </td>
          </tr>
        </table>
       <?php
        include('footer.php')
        ?>
   </body>
   </html>
```

SOURCE CODE FOR DONATIONS

```
<?php

//fetch from html
$firstname=$_POST["textfname"];
$middlename=$_POST["textmname"];
$lastname=$_POST["textlname"];
$phone=$_POST["textphone"];
$email=$_POST["textemail"];
$new=$_POST["textpasword"];
$conform=$_POST["textcpassword"];
$add=$_POST["textarea"];
$city=$_POST["textcity"];
$pin=$_POST["textpin"];
$msg=null;
if($new==$conform)
{

//connection
$conn=new PDO("mysql:host=localhost;dbname=orphangedb","root",null);
$conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);

//3.prepare statment to insert data into the category table
try{
   $stmt=$conn->prepare("insert into
user(firstname,middlename,lastname,email,password,address,city,pin,phonenumber) values
(?,?,?,?,?,?,?,?,?)");
   $stmt->bindParam(1,$firstname);
   $stmt->bindParam(2,$middlename);
```

```
   $stmt->bindParam(3,$lastname);
   $stmt->bindParam(4,$email);
   $stmt->bindParam(5,$new);
   $stmt->bindParam(6,$add);
   $stmt->bindParam(7,$city);
   $stmt->bindParam(8,$pin);
   $stmt->bindParam(9,$phone);
   $stmt->execute();
   $c=$stmt->rowCount();
   if($c>0){
      $msg="sign up sucessfull";
   }
   else{
      $msg="sign up unsucessful";
   }
}catch(Exception $e){
  $msg=$e->getMessage();
}
//4.clos ethe connection to datebase
finally{

  $conn=null;
}
}
else{
   $msg="NEW  AND CONFORM PASSWORD DO NOT MATCH";

}

?>
<html>
   <head>
      <title>Signin</title>
      <?php
       include('loginlink.php')
       ?>
   </head>
   <body>
       <?php
       include('header.php')
       ?>
      <table align="center">
       <tr>
          <td style="font: size 50px;"><?php

          echo $msg;
      ?></td>
       </tr>
```

```
        <tr>

<td> </td>
</tr>
        <tr>
            <td>
            <div class="group">
              <a href="loginhome.php"><input type="button" class="button"
value="BACK"/></a>

            </div>
            </td>
          </tr>
        </table>
        <?php
        include('footer.php')
        ?>
   </body>
   </html>
```

SOURCE CODE FOR APPLICATION

```
<?php
session_start();
$userid=$_POST["textuserid"];
$birthdate=$_POST["textcal"];
$age=$_POST["textage"];
$gender=$_POST["gender"];
$appdate=$_POST["textdate"];
$sta=$_POST["textstatus"];
$profession=$_POST["textprofession"];
$address=$_POST["proaddress"];
$health=$_POST["texthealth"];
$reasons=$_POST["textreasons"];
$city=$_POST["textcity"];
$pin=$_POST["textnpin"];
$status=null;

try{
   if(isset($_FILES["file2"]["type"]))
   {
      $validextensions=array("pdf");
       //split file,extension and store into $temporary
       $temporary=explode(".",$_FILES["file2"]["name"]);
       //get file extension from $temporary variable
       $file_extension=end($temporary);
       //check the mine type provided by the browser
       if((($_FILES["file2"]["type"]=="application/pdf"))
```

```
        &&in_array($file_extension,$validextensions))
        {
          //if file was not upload correctly or partially uploaded,returns 0 if valid
          if($_FILES["file2"]["error"]>0)
          $msg=$_FILES["file2"]["error"];
          //check if file is already uploaded
          else if(file_exists("../photo/".$_FILES["file2"]["name"]))
          $msg="THIS FILE ALREADY EXITS.";
          else
          {
            $sourcePath=$_FILES['file2']['tmp_name'];
            $targetPath="../photo/".$_FILES['file2']['name'];
            move_uploaded_file($sourcePath,$targetPath);
            $photos=$targetPath;
            $status="ok";
          }
        }
        else
        {
          $msg="INVALID FILE SIZE OR TYPE";
        }
      }
  else{
    $msg="PLEASE SELECT FILE";
  }
  if($status=="ok"){

    $conn=new PDO("mysql:host=localhost;dbname=orphangedb","root",null);
    $conn->setAttribute(PDO::ATTR_ERRMODE,PDO::ERRMODE_EXCEPTION);
    $stmt=$conn->prepare("insert into
application(userid,Birthdate,Age,Appdate,Status,profession,Address,City,Pin,HealthInsuranc
e,Reasons,doc,gender)values(?,?,?,?,?,?,?,?,?,?,?,?,?)");
    $stmt->bindParam(1,$userid);
    $stmt->bindParam(2,$birthdate);
    $stmt->bindParam(3,$age);
    $stmt->bindParam(4,$appdate);
    $stmt->bindParam(5,$sta);
    $stmt->bindParam(6,$profession);
    $stmt->bindParam(7,$address);
    $stmt->bindParam(8,$city);
    $stmt->bindParam(9,$pin);
    $stmt->bindParam(10,$health);
    $stmt->bindParam(11,$reasons);
    $stmt->bindParam(12,$photos);
    $stmt->bindParam(13,$gender);

    $stmt->execute();
    $msg="APPLICATION SUBMITED";
```

```
   }
}catch(Exception $e){
   $msg="APPLICATION NOT SUBMITED".$e->getMessage();
}finally{
   $conn=null;
}


        if(empty($_SESSION))
        {
           header('location:../adminsigninform.php');
        }
        ?>

<html>
   <head>
      <title>ADD CHILD INFO</title>
      <link rel="stylesheet" href="styless.css">
      <?php
       include("headerlink.php");
       ?>

   </head>
   <body>
   <?php
        if(empty($_SESSION))
        {
           header('location:../usersigninform.php');
        }
        ?>
      <div class="container">
         <div class="item">
         <?php
          include('header.php');
         ?>
         <br>
      </div>
      <div class="item" align="center">
        <h4><?php echo $msg; ?></h4>
      </div>
      </div>
      <?php
            include('footer.php');
            ?>

   </body>
</html>
```

10.TESTING:

Introduction:

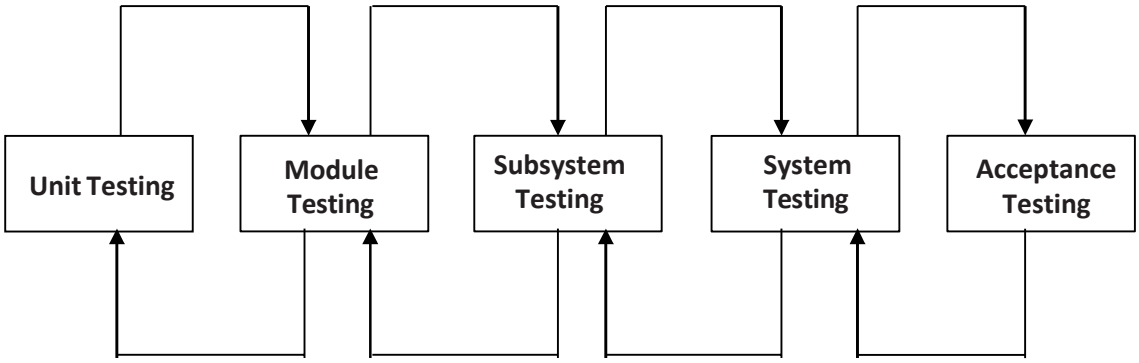
Software Testing is a process of executing program with an indent of finding error. Testing is vital to success of the system. Testing demonstrates that the software functions appear to be working according to the specifications and performance requirements appeared to have been met. If a test is conducted successfully, it will discover errors in the software.

Software Testing consists of all test life cycle activities like static and dynamic testing concerned with planning, preparation and evaluation of software products to determine that the software products satisfy customers requirements and are fit the customer use.

The various strategies that were used in testing this software are as follows:

- Unit Testing
- Integration Testing
- System Testing
    - Validation Testing
    - Black Box Testing
    - White Box Testing

Acceptance Testing

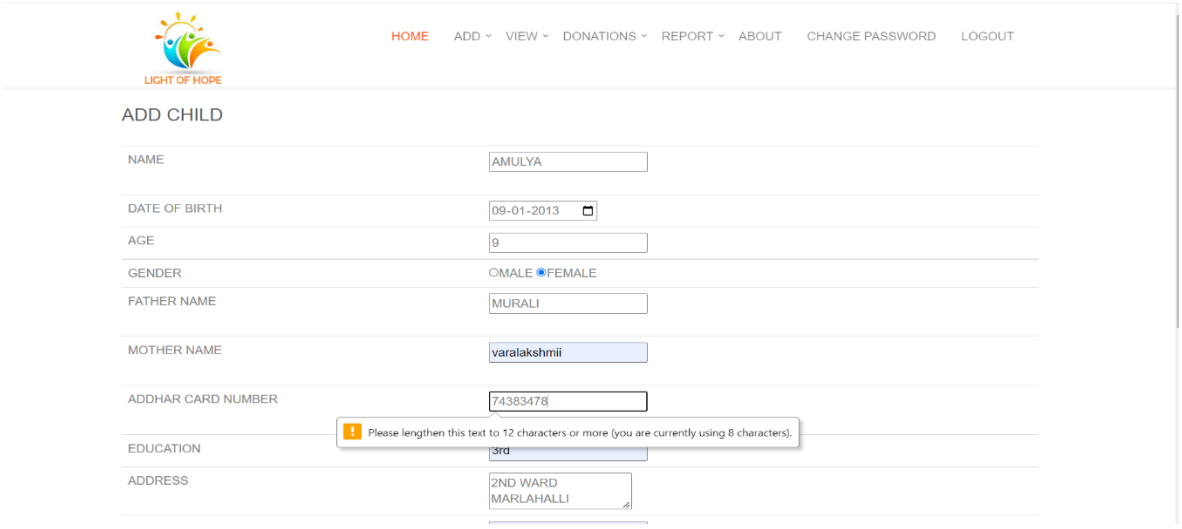


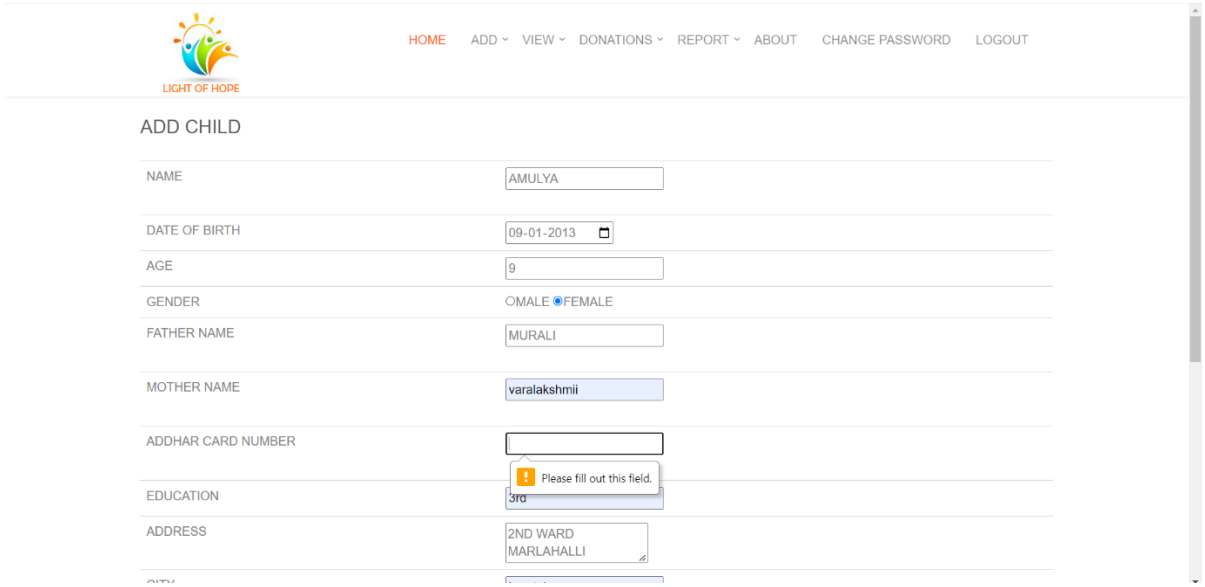
UNIT TESTING:

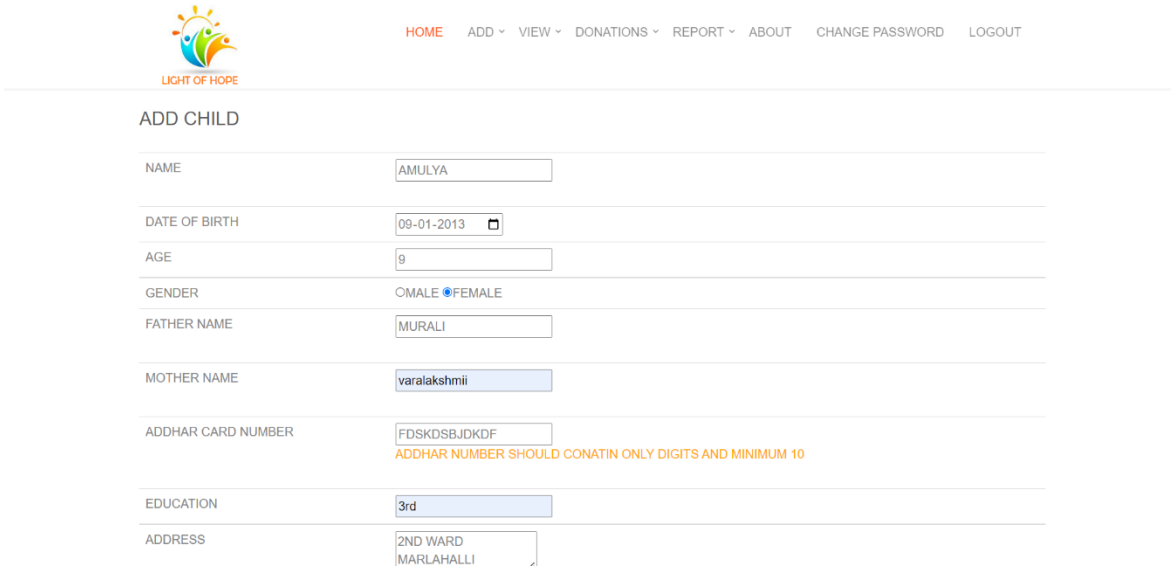
Unit testing is done on individual modules as they are completed and become executable. This system was tested with the set of proper test data for each module and results were checked with the expected output. Unit testing focuses on verification effort on the smallest unit of the software design module.

This is also known as MODULE TESTING. This testing is carried out during phases, each module is founded to be working satisfactory as regards to the expected result from the module.

Unit testing involves the design of the test cases that validate the internal program logic is functioning properly, and that program input produces valid output. All decision branches and internal code flow should be validated.


LIGHT OF HOPE
HOME ADD VIEW DONATIONS REPORT ABOUT CHANGE PASSWORD LOGOUT
ADD CHILD
NAME
AMULYA
DATE OF BIRTH
09-01-2013
AGE
9
GENDER
MALE FEMALE
FATHER NAME
MURALI
MOTHER NAME
varalakshmii
ADDHAR CARD NUMBER
74383478
Please lengthen this text to 12 characters or more (you are currently using 8 characters).
EDUCATION
3rd
ADDRESS
2ND WARD
MARLAHALLI


LIGHT OF HOPE
HOME ADD VIEW DONATIONS REPORT ABOUT CHANGE PASSWORD LOGOUT
ADD CHILD
NAME
AMULYA
DATE OF BIRTH
09-01-2013
AGE
9
GENDER
MALE FEMALE
FATHER NAME
MURALI
MOTHER NAME
varalakshmii
ADDHAR CARD NUMBER
Please fill out this field.
EDUCATION
3rd
ADDRESS
2ND WARD
MARLAHALLI


LIGHT OF HOPE
HOME ADD VIEW DONATIONS REPORT ABOUT CHANGE PASSWORD LOGOUT
ADD CHILD
NAME
AMULYA
DATE OF BIRTH
09-01-2013
AGE
9
GENDER
MALE FEMALE
FATHER NAME
MURALI
MOTHER NAME
varalakshmii
ADDHAR CARD NUMBER
FDSKDSBJDKDF
ADDHAR NUMBER SHOULD CONATIN ONLY DIGITS AND MINIMUM 10
EDUCATION
3rd
ADDRESS
2ND WARD
MARLAHALLI

INTEGRATION TESTING:

Integration testing ensures that software and subsystems work together as a whole. It tests the interface of all the modules to make sure that the modules behave properly when integrated together. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied as shown by the Unit testing, the combination of the components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components. Integration Testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests to aggregates and delivers as its output. The Integration Testing verifies functional, performance, and reliablility requirements placed on a major design items.

FUNCTIONAL TESTING:

Functional tests provide a systematic demonstration of the functions testedthat are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional Testing is centered on the following items:

Valid Input : Identified classes of valid input must be accepted. Invalid Input : Identified classes of invalid input must be accepted.Functions : Identified functions must be exercised.

Output : Identified classes of application outputs must be exercised.Systems/Procedures : Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows, data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of the current test is determined.

SYSTEM TESTING :

In this the entire software system is tested. The reference document for this process is the requirement document. Here the entire software is tested and the performance of the system was observed to see that it satisfies the requirement specification.

System testing tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process linksand integration points.

System testing involves in-house testing of the entire system before delivery to the user. Its aim is to satisfy the user. The system meets all requirements of the client's specifications.

The following are the types of system tests that were carried out for the system :

- **Validation Testing:** The system has been tested and implemented successfully and thus ensured that all requirements as listed in the software requirements specification are completely fulfilled. In case of erroneous input corresponding error messages are displayed.

- **Black Box Testing:**

  This method focuses on the functional requirements of the software. This testing enables to derive set input conditions that will fully exercise all functional requirements of the program. Black Box Testing attempts to find errors in the following category.

  - Incorrect or missing functions.
  - Interface errors.
  - Error in external database access.
  - Performance errors.
  - Initialization and Termination errors.

- **White Box Testing :**

  This is performed early in the testing process, while Black Box testing is applied during the last stage of testing. In this test cases are generated on the logic of eachmodule by drawing flow graphs of that module and logical decisions are tested onall the cases.

It has been used to generate the test case in the following test cases :

1. Guarantee that all independent paths have been executed.
2. Execute all logical decisions from their True and False side
3. Execute all loops at their boundaries and within their operational bounds.
4. Execute internal data structures to ensure their validity.
5. Ensure whether all the possible validity checks and validity lookups havebeen provided to validate data entry.

Performance Testing :

Performance Testing can serve different purpose. It can demonstrate that the system meets the performance criteria. It can compare two systems to find which performs better, or it can measure what parts of the system or workload cause the system to perform badly. In the diagnostic case, software engineers use tools such as profilers to measure what parts of a device or software contributes most to the poor performance.

It was a good idea to do our stress testing early on, because it gave us time to fix some of the unexpected deadlocks and stability problems that only occurred when components were exposed to very high transaction volumes.

Acceptance Testing :

It is a pre-delivery testing in which entire system is tested at client's site on real world data to find errors. It deals with successful satisfaction of user needs. This project is approved and accepted by the clients. The process flow and execution is 99% working with respect to system testing procedure

Test Objectives :

- The system is tested with variety of inputs. The System is tested for accuracy and correctness of the results obtained. Finally the system is tested for inter-operability.
- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Test cases

Example

Intake- Sign In Form :

TEST NO.	TEST CASE	EXPECTED RESULT	PASS
1.	Leave the Email ID and Password textbox blank and press Login	Message stating that provide Email ID and Password	Passed
2.	Enter the Email ID and Password	Homepage will display.	Passed
3.	Enter invalid Email ID and Password	Error message stating that invalid Email ID or Password	Passed
4.	Enter the invalid Password	Message stating that invalid Password	Passed
5.	Leave all textboxes blank	Message stating that provide enough values.	Passed

11.CONCLUSION:

- The software reduces manual work. Increases the reliability.
- Helps to increase Orphanage efficiency.
- Saves time and reduces overheads.

FUTURE SCOPE FOR THE SYSTEM:

Change is part of a project. So, we provide way to make some changes and enhance the projectfor future.

- Adoption process will be implemented further more through online.
- The Donation process will be taking place with the help of UPI

12.BIBLIOGRAPHY:

References

1. Fundamentals of database system-B.Navathe
2. Ian Sommerville “Software Engineering” Pearson Education Ltd 6th edition 2004

Websites

1. www.w3schools.com
2. www.google.com
3. www.wikepedia.com