**Chapter-1**

**SYNOPSIS**

**1.1 Title of the project**

“Crowd Funding”

* 1. **Introduction**

“Crowd Funding” system allows donors to donate to orphanages registered to the website. Users can contact admin to adopt children. The admin in turn forwards the request to the orphanage. Orphanages manage children and adoption details.

* 1. **Objective of the project**
* Provide an online portal to donate for orphanages.
* Orphanages can reach out to more donors.

**1.4 Project Category**

RDBMS (Web based Relational Database Management System)

* 1. **System Requirements:**

**1.5.1 Software Requirements**

* Operating System: Windows
* Text Editor: Sublime Text 3
* Web server: Apache
* Database server: MySQL
* Browsers: Chrome, Firefox or any other browsing application

**1..5.2 Hardware Requirements**

* Processor: Intel Dual Core or above
* Processor Speed: 2 GHz
* RAM: 1 GB
* Hard Disk: Minimum 40 GB
  1. **Languages Used**
     1. **Front End**
* Users Interface Design:

HTML: Hypertext Markup Language

CSS: Cascaded Style Sheet

JavaScript: Programming Language

* Server-side Scripting:

PHP: Hypertext Preprocessor

* + 1. **Back End**
* MySQL: Database Server
  1. **Project Modules**

**Admin:**

* Login Module
* Donor Module
* Orphanage Module
* User Module
* Feedback Module

**Orphanage:**

* Registration Module
* Login module
* Manage Children Detail
* Manage Adaption Request Module
* Donation Module
* Story Module

**Donor:**

* Registration Module
* Login Module
* View Children Details
* Request Adaption Module
* Donate Module
* Story Module
  1. **Future Scope**
* Develop an android application for the website.
* Develop an software version for the website.
* Develop chat application between donor and admin.

**Chapter-2**

**SOFTWARE REQUIREMENT SPECIFICATION**

**SOFTWARE REQUIREMENT SPECIFICATIONS (SRS)**

**2.1 INTRODUCTION**

The SRS typically contains a brief description of the project. It describes what the proposed software should do without describing how the software will for the do it. An SRS provides a reference for the validation of the final product, i.e, the SRS helps the client to determine if the software meets the requirements. SRS is the medium through which the client and user needs are accurately specified to the developer.

The basic objective of SRS is to specify the important requirements of the proposed system that are gathered during the system analysis. To satisfy the basic goals, an SRS should have some desirable characteristics: correctness, completeness, unambiguous, verifiable, modifiable and traceable, which we have observed throughout during the process in entirely.

The SRS of “Crowd Funding” is as follows:

**2.1.1 PURPOSE**

We make online donation and adoption request system which is easy, better and quick for everyone. The main purpose of developing this project is to provide a complete solution to all the problems that we face while making a donation. This system is designed to easily maintain the data of the donated and adopted persons. This system is made to keep the records about money got by donation and adopted persons details. Donor can donate money request adoption online.

**2.1.2 SCOPE**

* The main objective of the project is to get the money through online donation and send the adoption request online.
* Orphanages can post details of children online and the users can request for an adaption.
* Users can donate money through online and get a receipt in back.
* As the admin is present only genuine orphanages and donors are allowed.
* There will be manual verification of the orphanage details submitted to the admin.
* Administrator can view all the reports.

**2.1.3 DEFINITIONS, ACRONYMS AND ABBREVIATIONS**

* This program is menu driven program. When we click the menu options the different pages will be enabled.
* SRS: Software Requirement Specification
* GUI: Graphical User Interface
* DBNS: Database Management System
* RDBMS: Relational Database Management System
* ADMIN: The Administrator
* CPU: Central Processing Unit
* SQL: Structured Query Language
* HTML: Hypertext Markup Language
* CSS: Cascading Style Sheet
* Crowd Funding: Online donation and adaption request management
* PHP:- PHP: Hypertext pre-processor

PHP is a server scripting language, and is a powerful tool for making dynamic and interactive web pages quickly. PHP is a widely used, free and efficient alternative to competitors such as Microsoft’s ASP.

* MySQL: MySQL Database Server

MySQL is a most popular database system used on the web. It supports standard SQL Queries. The data in MySQL is stored in tables. A table is a collection of related data, ant it consists of columns and rows.

**2.1.4 REFERENCE**

**2.1.4.1 BOOK REFERENCES**

* Software engineering by Pankaj Jalote
* Learning PHP, MySQL, JavaScript and CSS: A Step-by-Step Guide to Creating Dynamic Websites by Robin Nixon

**2.1.4.2 INTERNET REFERENCES**

* [www.w3schools.com](http://www.w3schools.com)
* www.geeksforgeeks.com
* [www.php.net](http://www.php.net)
* [www.mysql.com](http://www.mysql.com)
* [www.tutorialspoint.com](http://www.tutorialspoint.com)

**2.1.5 OVERVIEW**

User Friendliness is provided with the application with various controls provided by system Rich User Interface. This is web application and its developed by PHP language. All the records stores in MySQL Database.

**2.2 OVERALL DESCRIPTION**

**2.2.1 PRODUCT PERSPECTIVE**

The product Crowd Funding web application will be developed completely by web application and accessed over the internet. The orphanages and donor can login to the system by entering Username and Password. After the login User can view orphanage details and donor details. The administrator is the main user of this web application and he can add orphanage details, add donor details edit, delete etc.

**2.2.2 PRODUCT FUNCTION**

The product functions will include the following areas:

The application is capable enough to store all the records and also perform some editing on them that is added. User Friendliness is provided in the application with various controls provided by the systems Rich User Interface.

* The administrator generates orphanages username and password.
* Donors can request for adoption without visiting the orphanage.
* Donors can donate through online.
* The system makes the overall project management much easier and flexible.
* The user information can be stored in centralized database which can be maintained by the system.

**2.2.3 USER CHARECTRISTICS**

* Administrator has the full authority over the website. He can view all the registered users and have the power to delete and edit them. He can edit the webpages and update them.
* A basic knowledge of internet usage may take benefits of the software.
* The user of this product is supposed to be fairly educated about the usage of the computers.
* A person who has no knowledge of computers will find it difficult to understand the system. But with a little knowledge it will be very easy to handle the project.

**2.2.4 GENERAL CONSTRAINTS**

* Designed to run on windows XP and further versions.
* Latest browsers like Internet Explorer or Mozilla Firefox or Google Chrome is required.
* Internet facility is essential to use this web application.

**2.2.5 ASSUMPTIONS AND DEPENDENCIES**

* Authentication is provided for this application only registered users can access.
* Every user must enter valid username and password to use this web web application.
* It can be accessed over the internet.

**2.3 SPECIFIC REQUIREMENTS**

**EXTERNAL INTERFACE REQUIREMENTS**

**2.3.1 USER INTERFACE**

The software provides good graphical interface for the front end which is self-explanatory. The website is designed using HTML, CSS and JavaScript. PHP is used as the frontend and MySQL server used as backend. The developer will have to study the designing of the product.

**2.3.2 HARDWARE INTERFACE (MINIMUM)**

* **Operating System:** Windows XP
* **Hard Disk:** 40 GB hard disk
* **RAM:** 1 GB RAM
* **Peripheral devices**
* **Processor:** Intel Pentium and above

**2.3.3 SOFTWARE INTERFACE (MINIMUM)**

* **Design Interface:** HTML, CSS, JavaScript
* **Front End:** PHP 5.2
* **Back End:** MySQL Server 5.5
* **Server:** Apache server 1.8.2
* **IDE:** Sublime Text 3

**2.4 FUNCTIONAL REQUIREMENTS (MODULES)**

**2.4.1 ADMIN MODULE**

This module stores orphanage and donor details. Admin module can add, edit, delete and block orphanages. this module gives username and password to the orphanages the further login. This module can view the donor details.

* **Login module:** Admin logs into the system with his username and password.
* **Donor module:** This module allows the admin to approve the registration, edit, delete and view details of donors.
* **Orphanage module:** This module allows the admin to approve the registration, edit, delete and view details of orphanages.
* **User module:** This module allows to view the user details
* **Feedback module:** This module allows the admin to view the feedbacks given by the donors.

**2.4.2 ORPHANAGE MODULE**

This module will first get approved by the admin and then gets password in order to login to the application in future. This module logs into the system and then posts the details of the child, who is to be get adopted.

* **Registration module:** The orphanage needs to give primary information to the system to approval.
* **Login module:** Once the admin approves the registration, orphanage can login with username and password.
* **Manage children detail:** This module allows orphanage to add, edit, delete and view children details.
* **Manage adoption request module:** The orphanage can confirm or reject adoption request.
* **Manage donation:** The orphanage can manage donations from the donor and also give receipt to donor.
* **Manage Story:** Orphanage can add story and view the added story.

**2.4.3 DONOR MODULE**

This module generates a username and password during the registration process and then uses it for the further login. This module uses a username and password to login and can adopt a child and make some donations.

* **Registration module:**  The orphanage needs to give primary information to the system for approval.
* **Login module:** Once the admin approves registration, donor can login with email and password.
* **View children detail:** Donor can view children details for adoption purposes.
* **Request adoption module:** This module allows the donor to contact the orphanage for the purpose of adoption.
* **Donate module:** Donor can donate to orphanages through online payment.
* **View Story:** Donor can view the story added by the orphanage.

**2.5 OTHER NON-FUNCTIONAL REQUIREMENTS**

**2.5.1 PERFORMANCE REQUIREMENTS**

The system is supposed to be having good memory space and RAM should be above 1 GB preferably. The sound card and graphics card will have to be of good quality and capacity.

**2.5.2 DESIGN CONSTRAINTS**

* Computer: INTEL compatible
* RAM: 256 MB or higher
* Operating System: Windows XP or above

**2.5.3 RELIABILIT**

This system is designed to have very simple database jut to cater the exact need of online career guidance system. It is tested for all the constraints at development stage.

**2.5.4 SECURITY**

This can give good security for the user information because data is not in client machine. Authentication is provided for this application only registered users can access.

**2.5.5 MAINTAINABILITY**

There will be no maintenance required for this web application. The database is provided by the end-user and therefore is maintained by this user.

**2.5.6 PORTABILITY**

Internet connection is required.

**2.5.7 OTHER REQUIREMENTS (IF ANY)**

None

**Chapter-3**

**SYSTEM ANALYSIS DESIGN**

**SYSTEM DESIGN**

**3.1. INTRODUCTION**

System design is the process or art of defining the architecture, components, modules, interfaces and data for a system to satisfy specify specified requirements. One could see it as the application of systems theory to product development. There is some overlap with the disciplines of system analysis, systems architecture and systems engineering.

In system design focus is on deciding which modules are needed for the system, the specifications of these modules should be interconnected is called **System Design**.

System design is also called top-level design. Here we consider a system to be set of components with clearly defined behaviour that interact with each other in a fixed manner to produce some behaviour. In a system design, the design consists of module definitions, with each module supporting a functional abstraction.

**3.2 CONTEXT FLOW DIAGRAM (CFD)**

A context flow diagram (CFD) is graphical representation of the “flow” of context through an information system.

CFDs can also be used for the visualization of data processing (structured design). On a CFD, data items flow from an external data source or an internal data source to an internal data store or an external data sink, via internal process. A CFD provides no information about the timing or ordering of the processes, or whether the processes will operate in sequence or in parallel. It is therefore quite different from a flowchart, which shows the control through an algorithm, allowing a reader to determine what operations will be performed, in that order, and under what circumstances.

In this context diagram, the information provided to and received from the movement control domain is identified. The arrows represent the information received or generated by the movement control domain. The closed boxes represent the set of sources and sinks of information.

A context flow diagram confines project scope to an agreed limit by obtaining agreement from the user on the data flows that are included in the scope of the project. The context diagram focuses on the relationships with the external entities and identifies the information that is exchanged between these external entities and the business system under review. The external entities themselves are outside the scope of the project but the data flows between the entities and the business system are within the project scope.

**3.2.1 LEVEL 0 DFD (CONTEXT FLOW DIAGRAM)**

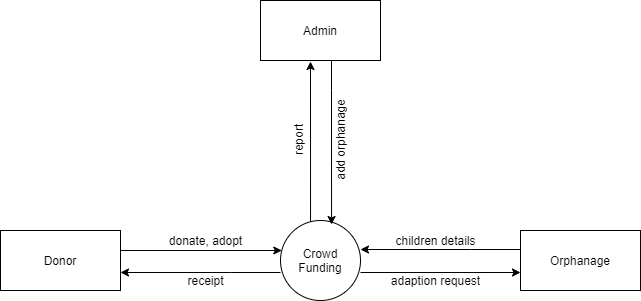


Figure 1.1.1: Level 0 DFD Context Flow Diagram

**3.3 DATA FLOW DIAGRAMS**

The DFD was first developed by Larry Constantine as a way of expressing system requirements in a graphical form; this leads a module design. A data flow diagram is a graph showing the flow of data values from their sources in objects through processes that transform them to their destinations in other objects.

A DFD also known as “bubble chart”, has the purpose of clarifying the system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. A DFD consists of a series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flows in the system. A DFD consists of processes that transform data, data flows that move data, actor objects that produce and consume data and data store objects that store data passively.

Data flow diagrams are used to describe how the system transforms information. They define how information is processed and stored and also identify how the information flows through the processes. It differs from flow chart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing (structured design).

**3.3.1 NOTATION OF THE DFD**

|  |  |
| --- | --- |
| **Notation** | **Definition** |
| **Process** | A process shows a transformation or manipulation of data flows within the system. A process transforms incoming data flow into outgoing data flow. |
| **Source/Sink** | The source or sink is represented as a rectangular box. The source or sink is the net originator of the consumer of the data that flows in the system. |
| **Data Flow** | A data flow shows flow of information from source to destination. A data flow is represented by a line with arrowhead showing the direction of flow. |
| **Database** | A data store indicates to which database a particular data is stored and from which database the data is retrieved. |

**3.3.2 TOP LEVEL DFD**

**3.3.2.1 LEVEL-1 ADMIN MODULE**

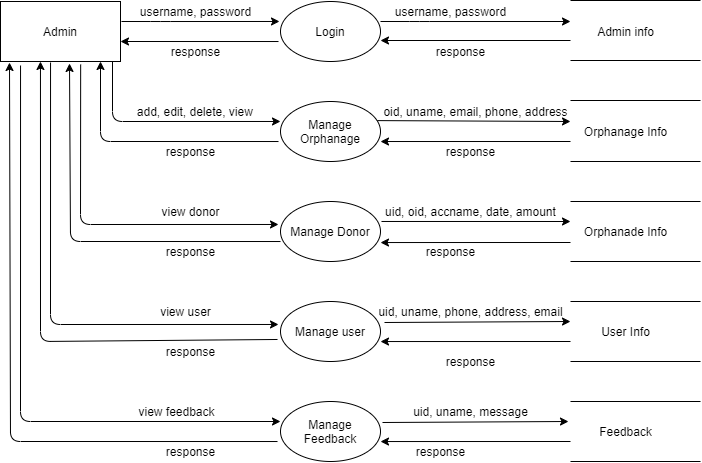


Figure 3.3.2: Level 1 Dataflow Diagram

**3..3.2.2 LEVEL-1 ORPHANAGE MODULE**

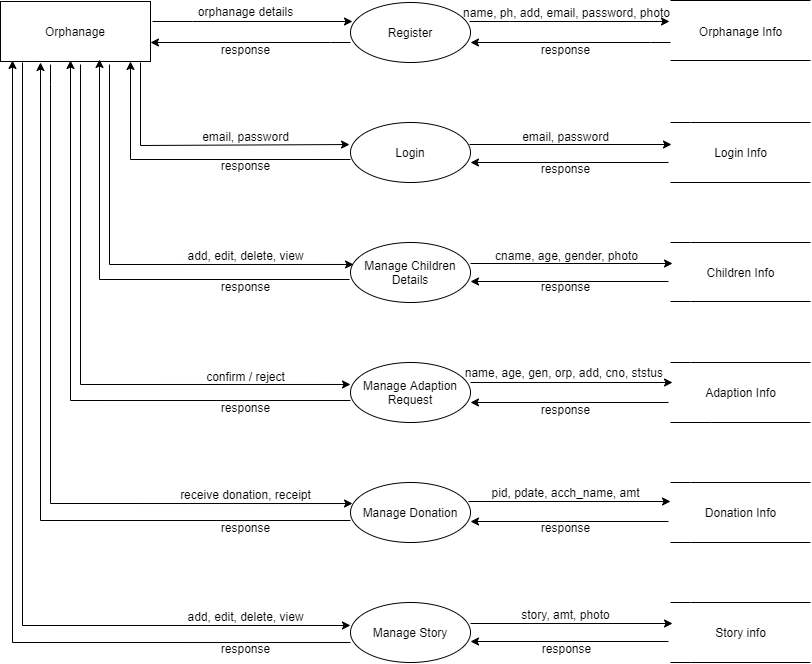
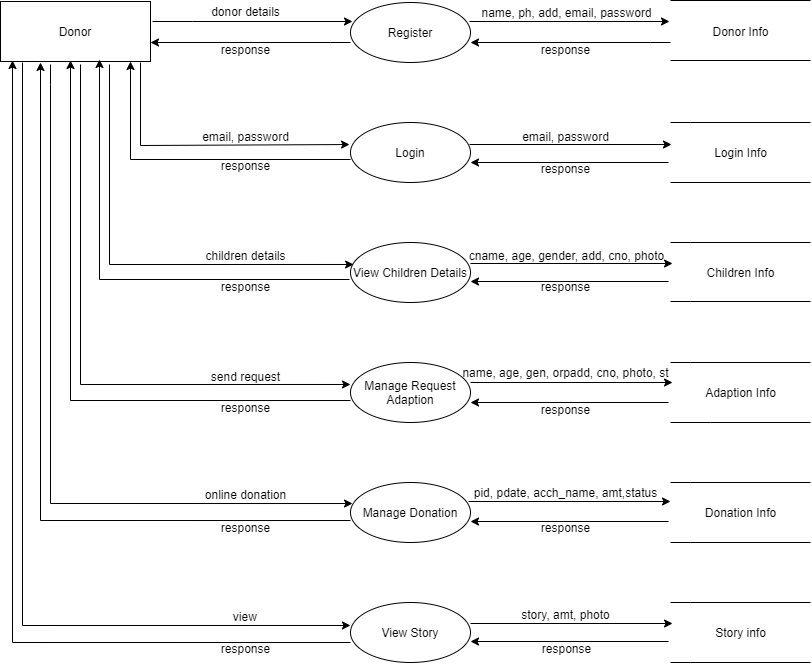


Figure 3.3.2.2 Level 1 Dataflow Diagram

**3.3.2.3 LEVEL-1 DONOR MODULE**



3.3.2.3 Level 1 Dataflow Diagram

**3.4 ENTITY RELATIONSHIP DIAGRAM**

Entity relationship diagram is used in modern database software. Software engineering is to illustrate logical structure of database. It is a relational schema database, modelling method, used to model a system and approach. This approach is commonly used in database design. The diagram created using this are called entity relationship diagram. The ER diagram depicts the various relationships among entities, considering each object as an entity. Relationship depicts the relationship between data objects. The ERD is the notation that is used to conduct the data modeling activity.

**3.4.1 ENTITY**

Entity is a thing, which we want to store information. It is an elementary basic building block of storing information about business process. An entity represents an object desired within the information system about which you want to store information.

**3.4.2 RELATIONSHIP**

A relationship is a named connection, associated between entities, or used to relate two or more entities with some common attributes or meaningful interaction between the object.

**3.4.3 ATTRIBUTES**

Attributes are the properties of entities and relationship. Description of the entity. Attributes are elementary pieces of information attached to an entity.

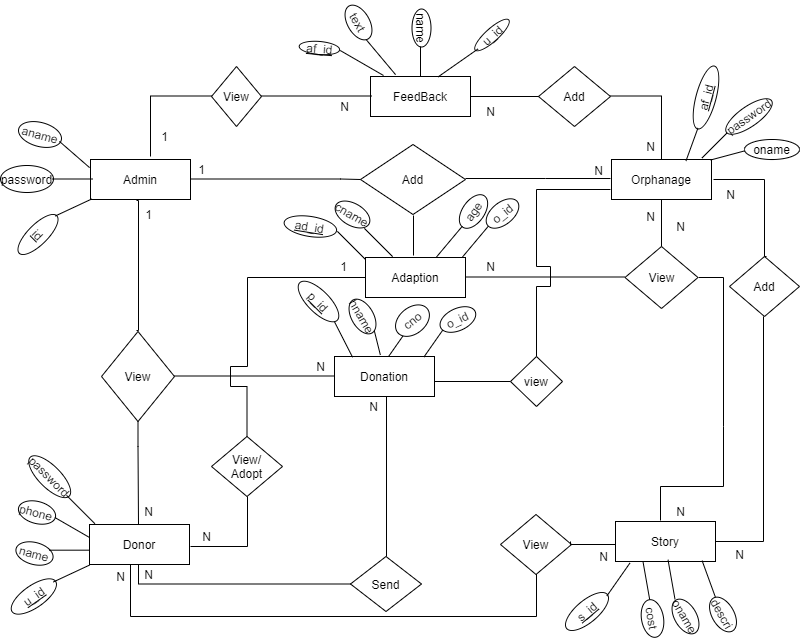
**ER (Entity Relationship Diagram) and Database design**

The database for our system and its design using ER diagrams is shown in the figure below.

Some of the conventions while designing the E-R Diagram are shown below.

|  |  |
| --- | --- |
| **SYMBOL** | **CONVENTION** |
|  | Entity |
|  | Relationship |
|  | Attribute |
|  | Key Attribute |
|  | Weak Entity |

**3.4.4 ER DIAGRAM**

****

**Chapter-4**

**DETAILED DESIGN**

**DETAILED DESIGN**

**4.1 INTRODUCTION**

Detailed design is the second level of design process. During detailed design, we specify how the module in the system interacts with each other and the internal logic of each of the modules specified during system during system design is decided, hence it is also called as logic design.

Detailed design essentially expands the system design and database design to contain a more detailed description of the processing logic and data structures so that the design is sufficiently complete for coding.

**4.2 APPLICABLE DOCUMENT**

The detailed design refines the system document. Hence the first applicable document here is system design. Also refer the data structure. Hence the second applicable document here is data base design.

* 1. **STRUCTURE OF SOFTWARE PACKAGE**

The Crowd Funding detailed design contains of following functional components:

* Functional component 1: Admin
* Functional Component 2: Orphanage
* Functional Component 3: Donor
  1. **MODULAR DECOMPOSITION OF COMPONENTS**
     1. **ADMIN MODULE**

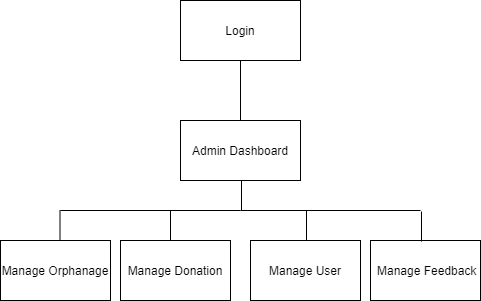
**4.4.1.1 DESIGN ASSUMPTION**

This module is designed with an intention to allow the admin in performing the specific activities displayed in the dashboard.

* + - 1. **IDENTIFICATION OF MODULES**
* Manage Orphanage
* Manage Donation
* Manage User
* Manage Feedback
  + - 1. **STRUCTURE CHART SHOWING THE HIERARCHY OF MODULES**

Structure chart is a graphical representation of the control logic of processing functions or modules representing a system.

It is one of the most commonly used methods for system design. In a structure chart, each program module is represented by a rectangular box. Modules at the top level of the structure chart call the modules at the lower level. The connections between modules are represented by lines between the rectangular boxes. The connection describes data flows between the called and calling modules.



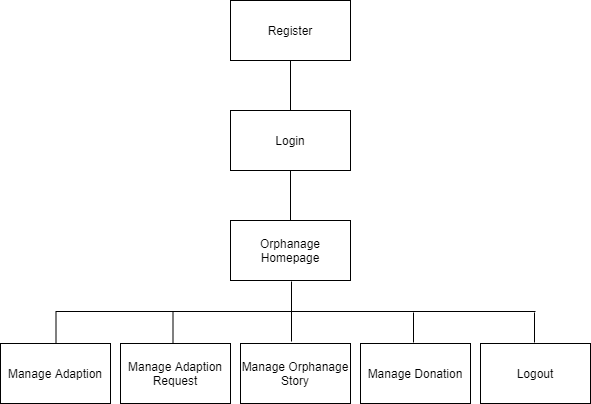
**4.4.2 ORPHANAGE MODULE**

* + - 1. **DESIGN ASSUMPTION**

This module is designed with an intention which allows the orphanage to perform all the activities specified in the dashboard.

**4.4.2.2 IDENTIFICATION OF MODULE**

* Manage Adaption
* Manage Adaption Request
* Manage Donation
* Manage Story
* Logout
  + - 1. **STRUCTURE CHART SHOWING THE HIERARCHY OF MODULES**



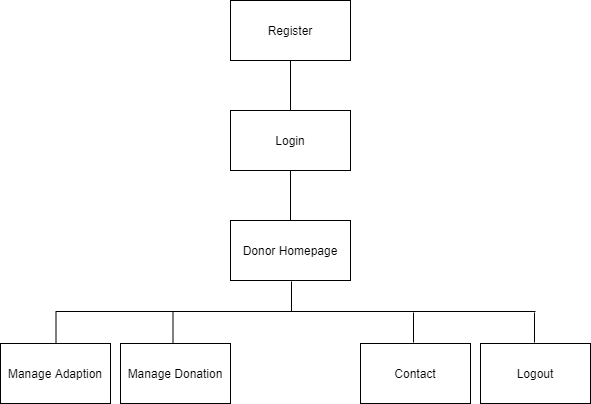
**4.4.3 DONOR MODULE**

**4.4.3.1 DESIGN ASSUMPTION**

This module is designed with an intention to allow the donor in performing the specified activities displayed in the homepage.

**4.4.3.2 IDENTIFICATION OF THE MODULE**

* Manage Children Details
* View Request Adaption
* Manage Donation
* Manage Story
  + - 1. **STRUCTURE CHART SHOWING THE HIERARCHY OF MODULES**



**4.5 DATA STRUCTURES SHARED AMONG MODULES**

Tables used are listed as follows:

* Login
* Children details
* Adaption
* Donation
* Rejected request
* Accepted request

**Chapter-5**

**DATABASE DESIGN**

**5.1 INTRODUCTION**

The word “database”, used to describe everything from a single set of data, to a complex set of tools, such as SQL server, and a whole lot in between. The term data model to mean the conceptual description of the problem space, this includes the definition of entities, their conceptual description of the problem space, this includes the definition of entities, their attributes, and the entity constraints. The data model also includes a description of the relationships between entities and any constraints on those relationships. It is a translation of the conceptual model into a physical representation, which shall be implemented using a database management system. The main advantage of this software is to reduce the manual work. The system also does the required calculations, maintenance error and free reports and validates the final report within a short period of time.

The DBMS provides flexibility in data storage and retrieval of data and production of information. It determines what type of data, is needed and how they are processed, and the operating system of the computer is responsible for placing data on the storage devices. A schema defines a database and the sub schema defines the portion of the database that a specific program will use:

* To retrieve data from database.
* The application program determines what data is needed and communicates the need to the DBMS.
* The DBMS system determines that the data requested are in fact stored in the database.
* The data must be defined in a subschema.
* The DBMS system instructs the OS to locate and retrieve the data from the specific location on the disk.
* The copy of the data is given to the OS for processing.
* A database, must be created before it can be used.

**5.2 DATABASE DESCRIPTION**

Database description describes all the database used in the software to store all the records. The database term is for describes the detail giving all the fields used with their data type, constraints available to them and description. Constraints include primary key, foreign key, etc. which allow the entities to be identified uniquely.

**5.3 TABLE DESIGN**

**Table Name:** Admin

**Description:** Table is used to store the details of the admin.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Constraints** | **Description** |
| a\_id | int(11) | Primary key | Admin ID |
| aname | varchar(100) | No | Admin name |
| password | varchar(100) | No | Password |

**Table Name:** Admin Feedback

**Description:** Table is used to store the feedback given by the users.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Constraints** | **Description** |
| af\_id | int(11) | Primary key | Admin feedback ID |
| name | varchar(100) | No | User name |
| text | varchar(100) | No | Feedback |
| u\_id | int(100) | No | User ID |

**Table Name:** Adoption

**Description:** Table is used to store the details of adaption.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Constraints** | **Description** |
| ad\_id | int(11) | Primary key | Adaption ID |
| cname | varchar(100) | No | Child name |
| age | int(11) | No | Age |
| phone | bigint(20) | No | Phone number |
| oname | varchar(100) | No | Orphanage name |
| o\_id | int(11) | No | Orphanage ID |
| address | varchar(100) | No | Address |
| status | varchar(100) | No | Status |
| image | varchar(100) | No | Image |
| gender | varchar(100) | No | Gender |

**Table Name:** Adoption Request

**Description:** Table is used to store the details of the user who requested for adaption.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Constraints** | **Description** |
| adr\_id | int(11) | Primary key | Adaption request ID |
| uname | varchar(100) | No | User name |
| uphone | bigint(20) | No | User phone number |
| phone | bigint(20) | No | Phone number |
| oname | varchar(100) | No | Orphanage name |
| o\_id | int(11) | No | Orphanage ID |
| address | varchar(100) | No | Address |
| status | varchar(100) | No | Adaption status |
| image | varchar(100) | No | Child Image |
| gender | varchar(100) | No | Child Gender |
| age | int(11) | No | Child Age |
| cname | varchar(100) | No | Child name |
| u\_id | int(11) | No | User ID |
| ad\_id | int(11) | No | Adaption ID |

**Table Name:** Orphanage

**Description:** Table is used to store the details of the orphanages.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Constraints** | **Description** |
| o\_id | int(11) | Primary key | Orphanage ID |
| oname | varchar(100) | No | Orphanage name |
| address | varchar(100) | No | Orphanage address |
| phone | varchar(100) | No | Phone number |
| email | varchar(100) | No | Orphanage email |
| password | varchar(100) | No | Password |
| status | varchar(100) | No | Orphanage status |
| image | varchar(100) | No | Orphanage image |

**Table Name:** Payment

**Description:** Table is used to store the details of donation donated by the donor.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Constraints** | **Description** |
| p\_id | int(11) | Primary key | Payment ID |
| hname | varchar(100) | No | Account holder name |
| cno | int(11) | No | Contact number |
| my | varchar(200) | No | Mon and year of expir |
| cvv | int(100) | No | Card number |
| o\_id | int(11) | No | Orphanage ID |
| s\_id | int(11) | No | Story ID |
| status | varchar(100) | No | Payment status |
| pamount | varchar(100) | No | Paid amount |
| u\_id | int(11) | No | User ID |
| pdate | varchar(100) | No | Payment date |

**Table Name:** Story

**Description:** Table is used to store the stories posted by the orphanage.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Constraints** | **Description** |
| s\_id | int(11) | Primary key | Story ID |
| description | varchar(1000) | No | Story |
| cost | int(11) | No | Targeted amount |
| phone | bigint(20) | No | Orphanage Contact |
| oname | varchar(100) | No | Orphanage name |
| o\_id | int(11) | No | Orphanage ID |
| address | varchar(100) | No | Orphanage address |
| image | varchar(100) | No | Child image |

**Table Name:** User

**Description:** Table is used to store the details of the user.

|  |  |  |  |
| --- | --- | --- | --- |
| **Column Name** | **Type** | **Constraints** | **Description** |
| u\_id | int(11) | Primary key | User ID |
| name | varchar(100) | No | User name |
| phone | varchar(100) | No | User contact |
| address | varchar(100) | No | User address |
| email | varchar(100) | No | User email |
| password | int(11) | No | User password |

**Chapter-6**

**CODING**

**CODING**

**6.1 DATABASE CONNECTION**

<?php

/\*\*

\* Created by PhpStorm.

\* User: your name

\* Date: todays date

\* Time: todays time

\*/

ob\_start();

session\_start();

//=========== database connection variables ====================

define('DB\_SERVER', "localhost"); // database host name eg. localhost or 127.0.0.1

define('DB\_USER', "root"); // database user name eg. root

define('DB\_DATABASE', "crowdfund"); //database name

define('DB\_PASSWORD', ""); //database user password

define('DB\_TYPE', 'mysql'); //database drive eg. mysql, pgsql, mongodb etc

//========== site details described here ========================

define('SITE\_TITLE', 'demo.com');

define('SITE\_TAG\_LINE', 'give your tag line of your project here');

//contact ifnormation

define('SITE\_CONTACT', 'your number');

//email information

define('SITE\_EMAIL\_INFO', 'your mail id');

//url information

define('BASE\_URL', 'http://localhost/cud opertaion/');

// included main class

require\_once 'app/Main.php';

/\*\*

\* @param $class

\*/

function \_\_autoload($class) {

require\_once 'app/' . $class.'.php';

}

**6.2 ADMIN**

**6.2.1 ADMIN LOGIN**

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

@import url('https://fonts.googleapis.com/css2?family=Fredoka+One&display=swap');

html

{

background: #fff;

}

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

font-family: 'Fredoka One', cursive;

color: black;

}

\*::selection{

background: purple;

color: #FFF;

}

section {

height: 60vh;

width: 100%;

background-color: blue;

position: relative;

animation: left .75s ease-out;

}

section::after {

background-color: orange;

width: 100%;

height: 175px;

position: absolute;

content: " ";

bottom: -75px;

transform: skewY(3.5deg);

}

.card {

background-color: #fff;

width: 400px;

height: 550px;

position: absolute;

right: 25%;

top: 19%;

z-index: 100;

border-radius: 40px;

box-shadow: 5px 5px 20px rgba(0, 0, 0, 0.25),

inset -5px -5px 20px rgba(0,0,0,.25),

inset 5px 5px 20px rgba(255,255,255,.25);

animation: right .75s ease-in;

}

.content {

display : flex;

flex-direction: column;

padding: 2em;

align-items: flex-center;

height: 100%;

justify-items: space-between;

}

.user, .pass {

position: relative;

}

.content i {

position: absolute;

top: 50%;

left: 20px;

transform: translateY(-50%);

color: #149ad1;

font-size: 1.125rem;

}

input[type="text"], input[type="password"] {

font-size: 1.125rem;

margin: .75em 0;

width: 100%;

border-radius: 100px;

border: none;

padding: .8em 1.5em .8em 2.75em;

color: #149ad1;

box-shadow: 1px 3px 10px #00000020,

inset -1px -3px 10px #00000020,

inset 1px 3px 10px #ffffff20;

}

input[type="text"]::placeholder,

input[type="password"]::placeholder{

color: #0e4c6690;

font-size: 1rem;

margin-left: 1em;

}

input[type="text"]:focus, input[type="password"]:focus {

outline: none;

}

h2 {

margin: 1.5em;

font-size: 2rem;

text-align: center;

}

.else {

display: flex;

align-items: center;

width: 85%;

margin: .5em auto;

}

.else span {

margin-left: auto;

}

.else span,

.else label {

font-size: .75rem;

text-align: right;

opacity: .65;

cursor: pointer;

transition: 250ms;

}

.else label {

margin-left: .5em;

}

span:hover,

label:hover{

opacity: 100%;

text-decoration: underline;

}

span:active,

label:active{

transform: scale(.99);

}

input[type="submit"] {

background-color: #149ad1;

border: none;

color: #fff;

padding: 1.1em 2.5em;

border-radius: 100px;

margin: 1.2em auto;

cursor: pointer;

box-shadow: 2px 3px 10px rgba(0,0,0,.25),

inset -2px -3px 10px rgba(0,0,0,.35),

inset 2px 3px 10px rgba(255,255,255,.5);

letter-spacing: 1.25px;

}

input[type="submit"]:hover {

background: #34a1ce;

}

input[type="submit"]:active {

box-shadow: 2px 3px 10px rgba(0,0,0,.25),

inset -2px -3px 10px rgba(255,255,255,.35),

inset 2px 3px 10px rgba(0,0,0,.25);

}

input[type="submit"]:focus {

outline: none;

}

section::before {

background-color: #fff;

width: 100%;

height: 250px;

position: absolute;

content: " ";

top: -100px;

transform: skewY(-3.5deg);

}

. fa-instagram {

position: absolute;

color: #149ad1;

bottom: 20%;

left: 20%;

font-size: 1.5rem;

transition: 300ms;

}

.instagram:hover .fa-instagram {

color: #149ad195;

}

@keyframes right{

0% {

transform: translateX(700%)

}

100% {

transform: translateX(0)

}

}

@keyframes left{

0% {

transform: translateX(-100%)

}

100% {

transform: translateX(0)

}

}

</style>

</head>

<body>

<script src="https://kit.fontawesome.com/783b858c89.js" crossorigin="anonymous"></script>

<section>

<div class="card">

<form action="../controller/adminlogin.php" method="post">

<div class="content">

<h2>CROWDFUND ADMIN</h2>

<div class="user">

<input type="text" placeholder="Username" name="a" required>

<i class="fas fa-user"></i>

</div>

<div class="pass">

<input type="password" placeholder="Password" name="b" required>

<i class="fas fa-lock"></i>

</div>

<input type="submit" value="Login" name="login">

</div>

</form>

</div>

</section>

</body>

</html>

**6.2.2 ADD ORPHANAGE**

<?php

define('DIR', '../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(!isset($\_SESSION["a\_id"])) {

header("location:adminlogin.php");

}

?>

<!DOCTYPE html>

<!--

Template Name: BRILLIANT Bootstrap Admin Template

Version: 4.5.6

Author: WebThemez

Website: http://www.webthemez.com/

-->

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta content="" name="description" />

<meta content="webthemez" name="author" />

<title>BRILLIANT Free Bootstrap Admin Template - WebThemez</title>

<!-- Bootstrap Styles-->

<link href="assets/css/bootstrap.css" rel="stylesheet" />

<!-- FontAwesome Styles-->

<link href="assets/css/font-awesome.css" rel="stylesheet" />

<!-- Morris Chart Styles-->

<link href="assets/js/morris/morris-0.4.3.min.css" rel="stylesheet" />

<!-- Custom Styles-->

<link href="assets/css/custom-styles.css" rel="stylesheet" />

<!-- Google Fonts-->

<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css' />

<link rel="stylesheet" href="assets/js/Lightweight-Chart/cssCharts.css">

</head>

<body>

<div id="wrapper">

<?php include "header.php";?>

<!--/. NAV TOP -->

<?php include "sidebar.php";?>

<!-- /. NAV SIDE -->

<div id="page-wrapper">

<div id="page-inner">

<!-- /. ROW -->

<div class="row">

<div class="col-xs-12">

<div class="panel panel-default">

<div class="panel-heading">

<div class="card-title">

<div class="title">Add all the Orphanage Center details</div>

</div>

</div>

<div class="panel-body">

<form action="../controller/orphange.php" method="post" enctype="multipart/form-data">

<div class="form-group">

<label for="exampleInputEmail1">Orphange center Name:</label>

<input type="text" name="a" class="form-control" id="exampleInputEmail1" placeholder="Enter Orphange center name" required>

</div>

<div class="form-group">

<label for="exampleInputEmail1">Email address:</label>

<input type="text" name="b" class="form-control" id="exampleInputEmail1" placeholder="Enter Email address" required>

</div>

<div class="form-group">

<label for="exampleInputEmail1">password:</label>

<input type="text" name="c" class="form-control" id="exampleInputEmail1" placeholder="Enter Paswword" required>

</div>

<div class="form-group">

<label for="exampleInputEmail1">Phone number</label>

<input type="text" name="d" class="form-control" pattern="^\d{10}$" title="10 numeric characters only" id="exampleInputEmail1" placeholder="Enter phone number" required>

</div>

<div class="form-group">

<label for="exampleInputPassword1">address details</label>

<input type="text" name="e" class="form-control" id="exampleInputPassword1" placeholder="Enter address details" required>

</div>

<div class="form-group">

<label for="exampleInputPassword1">upload photo details</label>

<input type="file" name="image" class="form-control" id="exampleInputPassword1" placeholder="Enter address details" required>

</div>

<button type="submit" name="addd" class="btn btn-warning">Submit</button>

</form>

</div>

</div>

</div>

<!-- /. ROW -->

</div>

<!-- /. PAGE INNER -->

</div>

<!-- /. PAGE WRAPPER -->

</div>

<!-- /. WRAPPER -->

<!-- JS Scripts-->

<!-- jQuery Js -->

<script src="assets/js/jquery-1.10.2.js"></script>

<!-- Bootstrap Js -->

<script src="assets/js/bootstrap.min.js"></script>

<!-- Metis Menu Js -->

<script src="assets/js/jquery.metisMenu.js"></script>

<!-- Morris Chart Js -->

<script src="assets/js/morris/raphael-2.1.0.min.js"></script>

<script src="assets/js/morris/morris.js"></script>

<script src="assets/js/easypiechart.js"></script>

<script src="assets/js/easypiechart-data.js"></script>

<script src="assets/js/Lightweight-Chart/jquery.chart.js"></script>

<!-- Custom Js -->

<script src="assets/js/custom-scripts.js"></script>

<!-- Chart Js -->

<script type="text/javascript" src="assets/js/Chart.min.js"></script>

<script type="text/javascript" src="assets/js/chartjs.js"></script>

</body>

</html>

**6.2.3 UPDATE ADD ORPHANAGE**

<?php

define('DIR', '../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(!isset($\_SESSION["a\_id"])) {

header("location:adminlogin.php");

}

$id=$\_GET['id'];

$stmtso = $admin -> ret("SELECT \* FROM `orphange` WHERE `o\_id` = '".$id."'");

$row = $stmtso -> fetch(PDO::FETCH\_ASSOC);

?>

<!DOCTYPE html>

<!--

Template Name: BRILLIANT Bootstrap Admin Template

Version: 4.5.6

Author: WebThemez

Website: http://www.webthemez.com/

-->

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta content="" name="description" />

<meta content="webthemez" name="author" />

<title>BRILLIANT Free Bootstrap Admin Template - WebThemez</title>

<!-- Bootstrap Styles-->

<link href="assets/css/bootstrap.css" rel="stylesheet" />

<!-- FontAwesome Styles-->

<link href="assets/css/font-awesome.css" rel="stylesheet" />

<!-- Morris Chart Styles-->

<link href="assets/js/morris/morris-0.4.3.min.css" rel="stylesheet" />

<!-- Custom Styles-->

<link href="assets/css/custom-styles.css" rel="stylesheet" />

<!-- Google Fonts-->

<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css' />

<link rel="stylesheet" href="assets/js/Lightweight-Chart/cssCharts.css">

</head>

<body>

<div id="wrapper">

<?php include "header.php";?>

<!--/. NAV TOP -->

<?php include "sidebar.php";?>

<!-- /. NAV SIDE -->

<div id="page-wrapper">

<div id="page-inner">

<!-- /. ROW -->

<div class="row">

<div class="col-xs-12">

<div class="panel panel-default">

<div class="panel-heading">

<div class="card-title">

<div class="title">update all orphange center details</div>

</div>

</div>

<div class="panel-body">

<form action="../controller/orphange.php" method="post" enctype="multipart/form-data">

<div class="form-group">

<label for="exampleInputEmail1">Orpahnge center Name:</label> <input type="text" name="a" class="form-control" id="exampleInputEmail1" placeholder="Enter Emission center name" value="<?php echo $row['oname'];?>"required>

</div>

<div class="form-group">

<label for="exampleInputEmail1">Email id:</label>

<input type="text" name="b" class="form-control" value="<?php echo $row['email'];?>" id="exampleInputEmail1" placeholder="Enter user name" required>

</div>

<div class="form-group">

<label for="exampleInputEmail1">password:</label>

<input type="text" name="c" class="form-control" value="<?php echo $row['password'];?>" id="exampleInputEmail1" placeholder="Enter Paswword" required>

</div>

<div class="form-group">

<label for="exampleInputEmail1">Phone number</label>

<input type="text" name="d" class="form-control" pattern="^\d{10}$" title="10 numeric characters only" id="exampleInputEmail1" value="<?php echo $row['phone'];?>" placeholder="Enter phone number" required>

</div>

<div class="form-group">

<label for="exampleInputPassword1">address details</label>

<input type="text" value="<?php echo $row['address'];?>" name="e" class="form-control" id="exampleInputPassword1" placeholder="Enter address details" required>

</div>

<div class="form-group">

<label for="exampleInputPassword1">upload photo details</label>

<input type="file" name="image" value="<?php echo $row['image'];?>" class="form-control" id="exampleInputPassword1" placeholder="Enter address details" required>

</div>

<input type="hidden" name="z" value="<?php echo $id; ?>">

<button type="submit" name="update" class="btn btn-warning">Submit</button>

</form>

</div>

</div>

</div>

</div>

<!-- /. ROW -->

</div>

<!-- /. PAGE INNER -->

</div>

<!-- /. PAGE WRAPPER -->

</div>

<!-- /. WRAPPER -->

<!-- JS Scripts-->

<!-- jQuery Js -->

<script src="assets/js/jquery-1.10.2.js"></script>

<!-- Bootstrap Js -->

<script src="assets/js/bootstrap.min.js"></script>

<!-- Metis Menu Js -->

<script src="assets/js/jquery.metisMenu.js"></script>

<!-- Morris Chart Js -->

<script src="assets/js/morris/raphael-2.1.0.min.js"></script>

<script src="assets/js/morris/morris.js"></script>

<script src="assets/js/easypiechart.js"></script>

<script src="assets/js/easypiechart-data.js"></script>

<script src="assets/js/Lightweight-Chart/jquery.chart.js"></script>

<!-- Custom Js -->

<script src="assets/js/custom-scripts.js"></script>

<!-- Chart Js -->

<script type="text/javascript" src="assets/js/Chart.min.js"></script>

<script type="text/javascript" src="assets/js/chartjs.js"></script>

</body>

</html>

**6.2.4 VIEW ORPHANAGE**

<?php

define('DIR', '../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(!isset($\_SESSION["a\_id"])) {

header("location:adminlogin.php");

}

?>

<!DOCTYPE html>

<!--

Template Name: BRILLIANT Bootstrap Admin Template

Version: 4.5.6

Author: WebThemez

Website: http://www.webthemez.com/

-->

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta content="" name="description" />

<meta content="webthemez" name="author" />

<title>BRILLIANT Free Bootstrap Admin Template - WebThemez</title>

<!-- Bootstrap Styles-->

<link href="assets/css/bootstrap.css" rel="stylesheet" />

<!-- FontAwesome Styles-->

<link href="assets/css/font-awesome.css" rel="stylesheet" />

<!-- Morris Chart Styles-->

<link href="assets/js/morris/morris-0.4.3.min.css" rel="stylesheet" />

<!-- Custom Styles-->

<link href="assets/css/custom-styles.css" rel="stylesheet" />

<!-- Google Fonts-->

<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css' />

<link rel="stylesheet" href="assets/js/Lightweight-Chart/cssCharts.css">

</head>

<body>

<div id="wrapper">

<?php include "header.php";?>

<!--/. NAV TOP -->

<?php include "sidebar.php";?>

<!-- /. NAV SIDE -->

<div id="page-wrapper">

<div id="page-inner">

<!-- /. ROW -->

<div class="col-md-12">

<!-- Context Classes -->

<div class="panel panel-default">

<div class="panel-heading">

View all Orphange center details

</div>

<div class="panel-body">

<div class="table-responsive">

<table class="table">

<tbody>

<tr class="primary">

<td></td>

<td>

Orphan Id

</td>

<td>

Orphan center name

</td>

<td>Email Id</td>

<td>password</td>

<td>phone number</td>

<td>address details</td>

<td>Status</td>

<td>

<span>Edit</span>

</td>

<td>

<span>Delete</span>

</td>

<td>

<span>Block Ac</span>

</td>

</tr>

<?php

$stmts=$admin -> ret("SELECT \* FROM `orphange`");

while($rows=$stmts->fetch(PDO::FETCH\_ASSOC))

{

?>

<tr class="success">

<td><img src="../controller/<?= $rows['image']; ?>" height="100px" width="100px"></td>

<td><?= $rows['o\_id']; ?></td>

<td><?= $rows['oname']; ?></td>

<td><?= $rows['email']; ?></td>

<td><?= $rows['password']; ?></td>

<td><?= $rows['phone']; ?></td>

<td><?= $rows['address']; ?></td>

<td style="color:red;"><?= $rows['status']; ?></td>

<td>

<a href="addorphangeupdate.php?id=<?= $rows['o\_id']; ?>" class="badge bg-primary" onclick="return confirm('Are you sure want to edit?')">Edit</a>

</td>

<td>

<a href="../controller/orphange.php?id=<?= $rows['o\_id']; ?>" class="badge bg-success" onclick="return confirm('Are you sure want to delete?')">Delete</a>

</td>

<td>

<a href="../controller/orphange.php?idd=<?= $rows['o\_id']; ?>" class="badge bg-success" onclick="return confirm('Are you sure want to block Ac?')">Block</a>

</td>

</tr>

<?php } ?>

</tbody>

</table>

</div>

</div>

</div>

</div>

<!-- /. ROW -->

</div>

<!-- /. PAGE INNER -->

</div>

<!-- /. PAGE WRAPPER -->

</div>

<!-- /. WRAPPER -->

<!-- JS Scripts-->

<!-- jQuery Js -->

<script src="assets/js/jquery-1.10.2.js"></script>

<!-- Bootstrap Js -->

<script src="assets/js/bootstrap.min.js"></script>

<!-- Metis Menu Js -->

<script src="assets/js/jquery.metisMenu.js"></script>

<!-- Morris Chart Js -->

<script src="assets/js/morris/raphael-2.1.0.min.js"></script>

<script src="assets/js/morris/morris.js"></script>

<script src="assets/js/easypiechart.js"></script>

<script src="assets/js/easypiechart-data.js"></script>

<script src="assets/js/Lightweight-Chart/jquery.chart.js"></script>

<!-- Custom Js -->

<script src="assets/js/custom-scripts.js"></script>

<!-- Chart Js -->

<script type="text/javascript" src="assets/js/Chart.min.js"></script>

<script type="text/javascript" src="assets/js/chartjs.js"></script>

</body>

</html>

**6.2.5 VIEW DONATION**

<?php

define('DIR', '../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(!isset($\_SESSION["a\_id"])) {

header("location:adminlogin.php");

}

?>

<!DOCTYPE html>

<!--

Template Name: BRILLIANT Bootstrap Admin Template

Version: 4.5.6

Author: WebThemez

Website: http://www.webthemez.com/

-->

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta content="" name="description" />

<meta content="webthemez" name="author" />

<title>BRILLIANT Free Bootstrap Admin Template - WebThemez</title>

<!-- Bootstrap Styles-->

<link href="assets/css/bootstrap.css" rel="stylesheet" />

<!-- FontAwesome Styles-->

<link href="assets/css/font-awesome.css" rel="stylesheet" />

<!-- Morris Chart Styles-->

<link href="assets/js/morris/morris-0.4.3.min.css" rel="stylesheet" />

<!-- Custom Styles-->

<link href="assets/css/custom-styles.css" rel="stylesheet" />

<!-- Google Fonts-->

<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css' />

<link rel="stylesheet" href="assets/js/Lightweight-Chart/cssCharts.css">

</head>

<body>

<div id="wrapper">

<?php include "header.php";?>

<!--/. NAV TOP -->

<?php include "sidebar.php";?>

<!-- /. NAV SIDE -->

<div id="page-wrapper">

<div id="page-inner">

<!-- /. ROW -->

<div class="col-md-12">

<!-- Context Classes -->

<div class="panel panel-default">

<div class="panel-heading">

View all Donated user details

</div>

<div class="panel-body">

<div class="table-responsive">

<table class="table">

<tbody>

<tr class="primary">

<td>

User Id

</td>

<td>Orpahange Id</td>

<td>

Ac holder name

</td>

<td>paid date</td>

<td>paid amount[Rs]</td>

</tr>

<?php

$stmts=$admin -> ret("SELECT \* FROM `payment`");

while($rows=$stmts->fetch(PDO::FETCH\_ASSOC))

{

?>

<tr class="success">

<td><?= $rows['u\_id']; ?></td>

<td><?= $rows['o\_id']; ?></td>

<td><?= $rows['hname']; ?></td>

<td><?= $rows['pdate']; ?></td>

<td><?= $rows['pamount']; ?></td>

</tr>

<?php } ?>

</tbody>

</table>

</div>

</div>

</div>

</div>

<!-- /. ROW -->

</div>

<!-- /. PAGE INNER -->

</div>

<!-- /. PAGE WRAPPER -->

</div>

<!-- /. WRAPPER -->

<!-- JS Scripts-->

<!-- jQuery Js -->

<script src="assets/js/jquery-1.10.2.js"></script>

<!-- Bootstrap Js -->

<script src="assets/js/bootstrap.min.js"></script>

<!-- Metis Menu Js -->

<script src="assets/js/jquery.metisMenu.js"></script>

<!-- Morris Chart Js -->

<script src="assets/js/morris/raphael-2.1.0.min.js"></script>

<script src="assets/js/morris/morris.js"></script>

<script src="assets/js/easypiechart.js"></script>

<script src="assets/js/easypiechart-data.js"></script>

<script src="assets/js/Lightweight-Chart/jquery.chart.js"></script>

<!-- Custom Js -->

<script src="assets/js/custom-scripts.js"></script>

<!-- Chart Js -->

<script type="text/javascript" src="assets/js/Chart.min.js"></script>

<script type="text/javascript" src="assets/js/chartjs.js"></script>

</body>

</html>

**6.2.6 VIEW USER**

<?php

define('DIR', '../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(!isset($\_SESSION["a\_id"])) {

header("location:adminlogin.php");

}

?>

<!DOCTYPE html>

<!--

Template Name: BRILLIANT Bootstrap Admin Template

Version: 4.5.6

Author: WebThemez

Website: http://www.webthemez.com/

-->

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta content="" name="description" />

<meta content="webthemez" name="author" />

<title>BRILLIANT Free Bootstrap Admin Template - WebThemez</title>

<!-- Bootstrap Styles-->

<link href="assets/css/bootstrap.css" rel="stylesheet" />

<!-- FontAwesome Styles-->

<link href="assets/css/font-awesome.css" rel="stylesheet" />

<!-- Morris Chart Styles-->

<link href="assets/js/morris/morris-0.4.3.min.css" rel="stylesheet" />

<!-- Custom Styles-->

<link href="assets/css/custom-styles.css" rel="stylesheet" />

<!-- Google Fonts-->

<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css' />

<link rel="stylesheet" href="assets/js/Lightweight-Chart/cssCharts.css">

</head>

<body>

<div id="wrapper">

<?php include "header.php";?>

<!--/. NAV TOP -->

<?php include "sidebar.php";?>

<!-- /. NAV SIDE -->

<div id="page-wrapper">

<div id="page-inner">

<!-- /. ROW -->

<div class="col-md-12">

<!-- Context Classes -->

<div class="panel panel-default">

<div class="panel-heading">

View all registered user details

</div>

<div class="panel-body">

<div class="table-responsive">

<table class="table">

<tbody>

<tr class="primary">

<td>

User Id

</td>

<td>

User name

</td>

<td>phone</td>

<td>address</td>

<td>email</td>

</tr>

<?php

$stmts=$admin -> ret("SELECT \* FROM `user`");

while($rows=$stmts->fetch(PDO::FETCH\_ASSOC))

{

?>

<tr class="success">

<td><?= $rows['u\_id']; ?></td>

<td><?= $rows['name']; ?></td>

<td><?= $rows['phone']; ?></td>

<td><?= $rows['address']; ?></td>

<td><?= $rows['email']; ?></td>

</tr>

<?php } ?>

</tbody>

</table>

</div>

</div>

</div>

</div>

<!-- /. ROW -->

</div>

<!-- /. PAGE INNER -->

</div>

<!-- /. PAGE WRAPPER -->

</div>

<!-- /. WRAPPER -->

<!-- JS Scripts-->

<!-- jQuery Js -->

<script src="assets/js/jquery-1.10.2.js"></script>

<!-- Bootstrap Js -->

<script src="assets/js/bootstrap.min.js"></script>

<!-- Metis Menu Js -->

<script src="assets/js/jquery.metisMenu.js"></script>

<!-- Morris Chart Js -->

<script src="assets/js/morris/raphael-2.1.0.min.js"></script>

<script src="assets/js/morris/morris.js"></script>

<script src="assets/js/easypiechart.js"></script>

<script src="assets/js/easypiechart-data.js"></script>

<script src="assets/js/Lightweight-Chart/jquery.chart.js"></script>

<!-- Custom Js -->

<script src="assets/js/custom-scripts.js"></script>

<!-- Chart Js -->

<script type="text/javascript" src="assets/js/Chart.min.js"></script>

<script type="text/javascript" src="assets/js/chartjs.js"></script>

</body>

</html>

**6.2.7 VIEW FEEDBACK**

<?php

define('DIR', '../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(!isset($\_SESSION["a\_id"])) {

header("location:adminlogin.php");

}

?>

<!DOCTYPE html>

<!--

Template Name: BRILLIANT Bootstrap Admin Template

Version: 4.5.6

Author: WebThemez

Website: http://www.webthemez.com/

-->

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<meta content="" name="description" />

<meta content="webthemez" name="author" />

<title>BRILLIANT Free Bootstrap Admin Template - WebThemez</title>

<!-- Bootstrap Styles-->

<link href="assets/css/bootstrap.css" rel="stylesheet" />

<!-- FontAwesome Styles-->

<link href="assets/css/font-awesome.css" rel="stylesheet" />

<!-- Morris Chart Styles-->

<link href="assets/js/morris/morris-0.4.3.min.css" rel="stylesheet" />

<!-- Custom Styles-->

<link href="assets/css/custom-styles.css" rel="stylesheet" />

<!-- Google Fonts-->

<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css' />

<link rel="stylesheet" href="assets/js/Lightweight-Chart/cssCharts.css">

</head>

<body>

<div id="wrapper">

<?php include "header.php";?>

<!--/. NAV TOP -->

<?php include "sidebar.php";?>

<!-- /. NAV SIDE -->

<div id="page-wrapper">

<div id="page-inner">

<!-- /. ROW -->

<div class="col-md-12">

<!-- Context Classes -->

<div class="panel panel-default">

<div class="panel-heading">

View all user feedback details

</div>

<div class="panel-body">

<div class="table-responsive">

<table class="table">

<tbody>

<tr class="primary">

<td>

User Id

</td>

<td>

User name

</td>

<td>Message</td>

</tr>

<?php

$stmts=$admin -> ret("SELECT \* FROM `adminfeedback`");

while($rows=$stmts->fetch(PDO::FETCH\_ASSOC))

{

?>

<tr class="success">

<td><?= $rows['u\_id']; ?></td>

<td><?= $rows['name']; ?></td>

<td><?= $rows['text']; ?></td>

</tr>

<?php } ?>

</tbody>

</table>

</div>

</div>

</div>

</div>

<!-- /. ROW -->

</div>

<!-- /. PAGE INNER -->

</div>

<!-- /. PAGE WRAPPER -->

</div>

<!-- /. WRAPPER -->

<!-- JS Scripts-->

<!-- jQuery Js -->

<script src="assets/js/jquery-1.10.2.js"></script>

<!-- Bootstrap Js -->

<script src="assets/js/bootstrap.min.js"></script>

<!-- Metis Menu Js -->

<script src="assets/js/jquery.metisMenu.js"></script>

<!-- Morris Chart Js -->

<script src="assets/js/morris/raphael-2.1.0.min.js"></script>

<script src="assets/js/morris/morris.js"></script>

<script src="assets/js/easypiechart.js"></script>

<script src="assets/js/easypiechart-data.js"></script>

<script src="assets/js/Lightweight-Chart/jquery.chart.js"></script>

<!-- Custom Js -->

<script src="assets/js/custom-scripts.js"></script>

<!-- Chart Js -->

<script type="text/javascript" src="assets/js/Chart.min.js"></script>

<script type="text/javascript" src="assets/js/chartjs.js"></script>

</body>

</html>

**6.3 ORPHANAGE**

**6.3.1 LOGIN**

<?php

define('DIR', '../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(isset($\_POST['login']))

{

$a=$\_POST['a'];

$b=$\_POST['b'];

$stmt=$admin->ret("SELECT \* from `admin` WHERE `aname`='$a' AND `password`='$b'");

$row = $stmt->fetch(PDO::FETCH\_ASSOC);

$num = $stmt->rowCount();

if($num>0){

$id=$row['a\_id'];

$\_SESSION['a\_id']=$id;

$admin->redirect('../admin/index');

}else{

echo"<script>

alert('you have entered wrong username or password');

window.location.href='../admin/adminlogin.php';

</script>";

}

}

?>

**6.3.2 UPDATE ORPHANAGE**

<?php

define('DIR','../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(isset($\_POST['addd'])){

$a = $\_POST['a'];

$b = $\_POST['b'];

$c = $\_POST['c'];

$d = $\_POST['d'] ;

$e = $\_POST['e'] ;

$status='accepted';

$targetDir = "../controller/photo/upload";

$image=$targetDir.basename($\_FILES["image"]["name"]); // @image = "upload/c.jpg"

move\_uploaded\_file($\_FILES['image']["tmp\_name"], $image);

$stmt = $admin -> cud("INSERT INTO `orphange` ( `oname`, `email`, `password`, `phone`,`address`,`status`,`image`) VALUES ('".$a."','".$b."','".$c."','".$d."','".$e."','".$status."','".$image."')","saved");

$admin -> redirect('../admin/vieworphange');

}

//delete

if(isset($\_GET['id']))

{

$id=$\_GET['id'];

$stmt=$admin -> cud("DELETE FROM `orphange` WHERE o\_id='$id'","deleted");

$admin -> redirect('../admin/vieworphange');

}

if(isset($\_POST['update'])){

$a = $\_POST['a'];

$b = $\_POST['b'];

$c = $\_POST['c'];

$d = $\_POST['d'] ;

$e = $\_POST['e'] ;

$status='accepted';

$z = $\_POST['z'] ;

$targetDir = "../controller/photo/upload";

$image=$targetDir.basename($\_FILES["image"]["name"]); // @image = "upload/c.jpg"

move\_uploaded\_file($\_FILES['image']["tmp\_name"], $image);

$stmt=$admin -> cud("UPDATE `orphange` SET `oname`='$a',`email`='$b',`password`='$c',`phone`='$d',`address`='$e',`status`='$status',`image`='$image' WHERE `o\_id`='$z'","updated");

$admin -> redirect('../admin/vieworphange');

}

if(isset($\_GET['idd'])){

$a = "blocked";

$id=$\_GET['idd'];

$stmt=$admin -> cud("UPDATE `orphange` SET `status`='$a' WHERE `o\_id`='$id'","updated");

$admin -> redirect('../admin/vieworphange');

}

?>

**6.3.3 ADOPTION**

<?php

define('DIR','../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(isset($\_POST['add'])){

$a = $\_POST['a'];

$b = $\_POST['b'];

$c = $\_POST['c'];

$e = $\_POST['e'] ;

$f = $\_POST['f'];

$g = $\_POST['g'] ;

$h = $\_POST['h'] ;

$status='show';

$targetDir = "../controller/photo/upload";

$image=$targetDir.basename($\_FILES["image"]["name"]); // @image = "upload/c.jpg"

move\_uploaded\_file($\_FILES['image']["tmp\_name"], $image);

$stmt = $admin -> cud("INSERT INTO `adopt` ( `cname`, `age`, `gender`, `o\_id`,`oname`,`phone`,`address`,`status`,`image`) VALUES ('".$a."','".$b."','".$c."','".$e."','".$f."','".$g."','".$h."','".$status."','".$image."')","saved");

$admin -> redirect('../user/orphan1');

}

//delete

if(isset($\_GET['id']))

{

$id=$\_GET['id'];

$stmt=$admin -> cud("DELETE FROM `adopt` WHERE `ad\_id`='$id'","deleted");

$admin -> redirect('../user/orphan2');

}

if(isset($\_POST['update'])){

$a = $\_POST['a'];

$b = $\_POST['b'];

$c = $\_POST['c'];

$e = $\_POST['e'] ;

$f = $\_POST['f'];

$g = $\_POST['g'] ;

$h = $\_POST['h'] ;

$status='show';

$z = $\_POST['z'] ;

$targetDir = "../controller/photo/upload";

$image=$targetDir.basename($\_FILES["image"]["name"]); // @image = "upload/c.jpg"

move\_uploaded\_file($\_FILES['image']["tmp\_name"], $image);

$stmt=$admin -> cud("UPDATE `adopt` SET `cname`='$a',`age`='$b',`gender`='$c',`o\_id`='$e',`oname`='$f',`phone`='$g',`address`='$h',`status`='$status',`image`='$image' WHERE `ad\_id`='$z'","updated");

$admin -> redirect('../user/orphan2');

}

?>

**6.3.4 ADOPTION ACCEPTED**

<?php

define('DIR','../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(isset($\_GET['idd'])){

$a = "accepted";

$id=$\_GET['idd'];

$b ="hide";

$ad=$\_GET['ad'];

$stmt=$admin -> cud("UPDATE `adoptreq` SET `status`='$a' WHERE `adr\_id`='$id'","updated");

$stmts=$admin -> cud("UPDATE `adopt` SET `status`='$b' WHERE `ad\_id`='$ad'","updated");

$admin -> redirect('../user/orphan3');

}

?>

**6.3.5 ADAPTION REJECTED**

<?php

define('DIR','../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(isset($\_GET['idd'])){

$a = "rejected";

$id=$\_GET['idd'];

$stmt=$admin -> cud("UPDATE `adoptreq` SET `status`='$a' WHERE `adr\_id`='$id'","updated");

$admin -> redirect('../user/orphan3');

}

?>

**6.3.6 ADOPTION REQUESTED**

<?php

define('DIR','../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(!isset($\_SESSION["u\_id"])) {

header("location:../user/logreg.php");

}

$a = $\_GET['a'];

$b = $\_GET['b'];

$c = $\_GET['c'];

$e = $\_GET['e'] ;

$f = $\_GET['f'];

$g = $\_GET['g'] ;

$h = $\_GET['h'] ;

$i = $\_GET['i'] ;

$status='pending';

$image=$\_GET['image'] ;

$u\_id=$\_SESSION['u\_id'];

$stmts=$admin -> ret("SELECT \* FROM `user` WHERE `u\_id`='$u\_id'");

$rows=$stmts->fetch(PDO::FETCH\_ASSOC);

$uphone=$rows['phone'];

$uname=$rows['name'];

$stmt = $admin -> cud("INSERT INTO `adoptreq` ( `cname`, `age`, `gender`, `o\_id`,`oname`,`phone`,`address`,`ad\_id`,`status`,`image`,`u\_id`,`uphone`,`uname`) VALUES ('".$a."','".$b."','".$c."','".$e."','".$f."','".$g."','".$h."','".$i."','".$status."','".$image."','".$u\_id."','".$uphone."','".$uname."')","saved");

// $admin -> redirect('../user/index');

echo"<script>

alert('your adoption request sent successfully');

window.location.href='../user/index.php';

</script>";

?>

**6.3.7 PAYMENT**

<?php

define('DIR','../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(isset($\_POST['add'])){

$a = $\_POST['a'];

$b = $\_POST['b'];

$c = $\_POST['c'];

$e = $\_POST['e'] ;

$f = $\_POST['f'];

$g = $\_POST['g'] ;

$h = $\_POST['h'] ;

$status='paid';

$u\_id=$\_SESSION['u\_id'];

$pdate=date('y-m-d');

$stmt = $admin -> cud("INSERT INTO `payment` ( `hname`, `cno`, `my`, `cvv`,`pamount`,`o\_id`,`s\_id`,`status`,`u\_id`,`pdate`) VALUES ('".$a."','".$b."','".$c."','".$e."','".$f."','".$g."','".$h."','".$status."','".$u\_id."','".$pdate."')","saved");

echo"<script>

alert('your payment sent successfully');

window.location.href='../user/index.php';

</script>";

}

?>

**6.3.8 UPDATE STORY**

<?php

define('DIR','../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(isset($\_POST['add'])){

$a = $\_POST['a'];

$b = $\_POST['b'];

$e = $\_POST['e'] ;

$f = $\_POST['f'];

$g = $\_POST['g'] ;

$h = $\_POST['h'] ;

$targetDir = "../controller/photo/upload";

$image=$targetDir.basename($\_FILES["image"]["name"]); // @image = "upload/c.jpg"

move\_uploaded\_file($\_FILES['image']["tmp\_name"], $image);

$stmt = $admin -> cud("INSERT INTO `story` ( `description`, `cost`, `o\_id`,`oname`,`phone`,`address`,`image`) VALUES ('".$a."','".$b."','".$e."','".$f."','".$g."','".$h."','".$image."')","saved");

$admin -> redirect('../user/orphan4');

}

//delete

if(isset($\_GET['id']))

{

$id=$\_GET['id'];

$stmt=$admin -> cud("DELETE FROM `story` WHERE `s\_id`='$id'","deleted");

$admin -> redirect('../user/orphan5');

}

if(isset($\_POST['update'])){

$a = $\_POST['a'];

$b = $\_POST['b'];

$e = $\_POST['e'] ;

$f = $\_POST['f'];

$g = $\_POST['g'] ;

$h = $\_POST['h'] ;

$z = $\_POST['z'] ;

$targetDir = "../controller/photo/upload";

$image=$targetDir.basename($\_FILES["image"]["name"]); // @image = "upload/c.jpg"

move\_uploaded\_file($\_FILES['image']["tmp\_name"], $image);

$stmt=$admin -> cud("UPDATE `story` SET `description`='$a',`cost`='$b',`o\_id`='$e',`oname`='$f',`phone`='$g',`address`='$h',`image`='$image' WHERE `s\_id`='$z'","updated");

$admin -> redirect('../user/orphan5');

}

?>

**6.3.9 FEEDBACK**

<?php

define('DIR','../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(!isset($\_SESSION["u\_id"])) {

header("location:../user/logreg.php");

}

$a = $\_POST['a'];

$b = $\_POST['b'];

$u\_id=$\_SESSION['u\_id'];

$stmt = $admin -> cud("INSERT INTO `adminfeedback` ( `name`, `text`, `u\_id`) VALUES ('".$a."','".$b."','".$u\_id."')","saved");

// $admin -> redirect('../user/index');

echo"<script>

alert('your feedback sent successfully');

window.location.href='../user/index.php';

</script>";

?>

**6.3.10 LOGOUT**

<?php

define('DIR','../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(isset($\_SESSION['a\_id'])){

session\_destroy();

unset($\_SESSION['a\_id']);

$admin -> redirect('../admin/adminlogin');

}

?>

**6.4 USER**

**6.4.1 REGISTRATION**

<!DOCTYPE html>

<html>

<head>

<title></title>

<style type="text/css">

:root{

--form-height:550px;

--form-width: 900px;

/\* Sea Green \*/

--left-color: #9fdeaf;

/\* Light Blue \*/

--right-color: #96dbe2;

}

body, html{

width: 100%;

height: 100%;

margin: 0;

font-family: 'Helvetica Neue', sans-serif;

letter-spacing: 0.5px;

}

.container{

width: var(--form-width);

height: var(--form-height);

position: relative;

margin: auto;

box-shadow: 2px 10px 40px rgba(22,20,19,0.4);

border-radius: 10px;

margin-top: 50px;

}

/\*

----------------------

Overlay

----------------------

\*/

.overlay{

width: 100%;

height: 100%;

position: absolute;

z-index: 100;

background-image: linear-gradient(to right, var(--left-color), var(--right-color));

border-radius: 10px;

color: white;

clip: rect(0, 385px, var(--form-height), 0);

}

.open-sign-up{

animation: slideleft 1s linear forwards;

}

.open-sign-in{

animation: slideright 1s linear forwards;

}

.overlay .sign-in, .overlay .sign-up{

/\* Width is 385px - padding \*/

--padding: 50px;

width: calc(385px - var(--padding) \* 2);

height: 100%;

display: flex;

justify-content: center;

flex-direction: column;

align-items: center;

padding: 0px var(--padding);

text-align: center;

}

.overlay .sign-in{

float: left;

}

.overlay-text-left-animation{

animation: text-slide-in-left 1s linear;

}

.overlay-text-left-animation-out{

animation: text-slide-out-left 1s linear;

}

.overlay .sign-up{

float:right;

}

.overlay-text-right-animation{

animation: text-slide-in-right 1s linear;

}

.overlay-text-right-animation-out{

animation: text-slide-out-right 1s linear;

}

.overlay h1{

margin: 0px 5px;

font-size: 2.1rem;

}

.overlay p{

margin: 20px 0px 30px;

font-weight: 200;

}

/\*

------------------------

Buttons

------------------------

\*/

.switch-button, .control-button{

cursor: pointer;

display: block;

margin-left: auto;

margin-right: auto;

width: 140px;

height: 40px;

font-size: 14px;

text-transform: uppercase;

background: none;

border-radius: 20px;

color: white;

}

.switch-button{

border: 2px solid;

}

.control-button{

border: none;

margin-top: 15px;

}

.switch-button:focus, .control-button:focus{

outline:none;

}

.control-button.up{

background-color: var(--left-color);

}

.control-button.in{

background-color: var(--right-color);

}

/\*

--------------------------

Forms

--------------------------

\*/

.form{

width: 100%;

height: 100%;

position: absolute;

border-radius: 10px;

}

.form .sign-in, .form .sign-up{

--padding: 50px;

position:absolute;

/\* Width is 100% - 385px - padding \*/

width: calc(var(--form-width) - 385px - var(--padding) \* 2);

height: 100%;

display: flex;

justify-content: center;

flex-direction: column;

align-items: center;

padding: 0px var(--padding);

text-align: center;

}

/\* Sign in is initially not displayed \*/

.form .sign-in{

display: none;

}

.form .sign-in{

left:0;

}

.form .sign-up{

right: 0;

}

.form-right-slide-in{

animation: form-slide-in-right 1s;

}

.form-right-slide-out{

animation: form-slide-out-right 1s;

}

.form-left-slide-in{

animation: form-slide-in-left 1s;

}

.form-left-slide-out{

animation: form-slide-out-left 1s;

}

.form .sign-in h1{

color: var(--right-color);

margin: 0;

}

.form .sign-up h1{

color: var(--left-color);

margin: 0;

}

.social-media-buttons{

display: flex;

justify-content: center;

width: 100%;

margin: 15px;

}

.social-media-buttons .icon{

width: 40px;

height: 40px;

border: 1px solid #dadada;

border-radius: 100%;

display: flex;

justify-content: center;

align-items: center;

margin: 10px 7px;

}

.small{

font-size: 13px;

color: grey;

font-weight: 200;

margin: 5px;

}

.social-media-buttons .icon svg{

width: 25px;

height: 25px;

}

#sign-in-form input, #sign-up-form input{

margin: 12px;

font-size: 14px;

padding: 15px;

width: 260px;

font-weight: 300;

border: none;

background-color: #e4e4e494;

font-family: 'Helvetica Neue', sans-serif;

letter-spacing: 1.5px;

padding-left: 20px;

}

#sign-in-form input::placeholder{

letter-spacing: 1px;

}

.forgot-password{

font-size: 12px;

display: inline-block;

border-bottom: 2px solid #efebeb;

padding-bottom: 3px;

}

.forgot-password:hover{

cursor: pointer;

}

/\*

---------------------------

Animation

---------------------------

\*/

@keyframes slideright{

0%{

clip: rect(0, 385px, var(--form-height), 0);

}

30%{

clip: rect(0, 480px, var(--form-height), 0);

}

/\* we want the width to be slightly larger here \*/

50%{

clip: rect(0px, calc(var(--form-width) / 2 + 480px / 2), var(--form-height), calc(var(--form-width) / 2 - 480px / 2));

}

80%{

clip: rect(0px, var(--form-width), var(--form-height), calc(var(--form-width) - 480px));

}

100%{

clip: rect(0px, var(--form-width), var(--form-height), calc(var(--form-width) - 385px));

}

}

@keyframes slideleft{

100%{

clip: rect(0, 385px, var(--form-height), 0);

}

70%{

clip: rect(0, 480px, var(--form-height), 0);

}

/\* we want the width to be slightly larger here \*/

50%{

clip: rect(0px, calc(var(--form-width) / 2 + 480px / 2), var(--form-height), calc(var(--form-width) / 2 - 480px / 2));

}

30%{

clip: rect(0px, var(--form-width), var(--form-height), calc(var(--form-width) - 480px));

}

0%{

clip: rect(0px, var(--form-width), var(--form-height), calc(var(--form-width) - 385px));

}

}

@keyframes text-slide-in-left{

0% {

padding-left: 20px;

}

100% {

padding-left: 50px;

}

}

@keyframes text-slide-in-right{

0% {

padding-right: 20px;

}

100% {

padding-right: 50px;

}

}

@keyframes text-slide-out-left{

0% {

padding-left: 50px;

}

100% {

padding-left: 20px;

}

}

@keyframes text-slide-out-right{

0% {

padding-right: 50px;

}

100% {

padding-right: 20px;

}

}

@keyframes form-slide-in-right{

0%{

padding-right: 100px;

}

100%{

padding-right: 50px;

}

}

@keyframes form-slide-in-left{

0%{

padding-left: 100px;

}

100%{

padding-left: 50px;

}

}

@keyframes form-slide-out-right{

0%{

padding-right: 50px;

}

100%{

padding-right: 80px;

}

}

@keyframes form-slide-out-left{

0%{

padding-left: 50px;

}

100%{

padding-left: 80px;

}

}

</style>

</head>

<body>

<div class="container">

<div class="overlay" id="overlay">

<div class="sign-in" id="sign-in">

<h1>Welcome Back!</h1>

<p>To keep connected with us please login with your personal info</p>

<button class="switch-button" id="slide-right-button">Sign In</button>

</div>

<div class="sign-up" id="sign-up">

<h1>Hello, Friend!</h1>

<p>Enter your personal details and start a journey with us</p>

<button class="switch-button" id="slide-left-button">Sign Up</button>

</div>

</div>

<div class="form">

<div class="sign-in" id="sign-in-info">

<h1>Sign In</h1>

<div class="social-media-buttons">

<div class="icon">

<svg viewBox="0 0 24 24">

<path fill="#000000" d="M17,2V2H17V6H15C14.31,6 14,6.81 14,7.5V10H14L17,10V14H14V22H10V14H7V10H10V6A4,4 0 0,1 14,2H17Z" />

</svg>

</div>

<div class="icon">

<svg viewBox="0 0 24 24">

<path fill="#000000" d="M23,11H21V9H19V11H17V13H19V15H21V13H23M8,11V13.4H12C11.8,14.4 10.8,16.4 8,16.4C5.6,16.4 3.7,14.4 3.7,12C3.7,9.6 5.6,7.6 8,7.6C9.4,7.6 10.3,8.2 10.8,8.7L12.7,6.9C11.5,5.7 9.9,5 8,5C4.1,5 1,8.1 1,12C1,15.9 4.1,19 8,19C12,19 14.7,16.2 14.7,12.2C14.7,11.7 14.7,11.4 14.6,11H8Z" />

</svg>

</div>

<div class="icon">

<svg viewBox="0 0 24 24">

<path fill="#000000" d="M21,21H17V14.25C17,13.19 15.81,12.31 14.75,12.31C13.69,12.31 13,13.19 13,14.25V21H9V9H13V11C13.66,9.93 15.36,9.24 16.5,9.24C19,9.24 21,11.28 21,13.75V21M7,21H3V9H7V21M5,3A2,2 0 0,1 7,5A2,2 0 0,1 5,7A2,2 0 0,1 3,5A2,2 0 0,1 5,3Z" />

</svg>

</div>

</div>

<p class="small">or use your email account:</p>

<form id="sign-in-form" action="../controller/logreg.php" method="post">

<input type="email" placeholder="Email" name="a" />

<input type="password" placeholder="Password" name="b" />

<input type="submit" name="login" value="Sign In">

</form>

</div>

<div class="sign-up" id="sign-up-info">

<h1>Create Account</h1>

<p class="small">new user register here:</p>

<form id="sign-up-form" action="../controller/logreg.php" method="post">

<input type="text" placeholder="Name" name="a"/>

<input type="text" placeholder="phone" name="b"/>

<input type="text" placeholder="address" name="c"/>

<input type="text" placeholder="email" name="d"/>

<input type="text" placeholder="password" name="e"/>

<input type="submit" name="register" value="Sign Up">

</form>

</div>

</div>

</div>

<script type="text/javascript">

var overlay = document.getElementById("overlay");

// Buttons to 'switch' the page

var openSignUpButton = document.getElementById("slide-left-button");

var openSignInButton = document.getElementById("slide-right-button");

// The sidebars

var leftText = document.getElementById("sign-in");

var rightText = document.getElementById("sign-up");

// The forms

var accountForm = document.getElementById("sign-in-info")

var signinForm = document.getElementById("sign-up-info");

// Open the Sign Up page

openSignUp = () =>{

// Remove classes so that animations can restart on the next 'switch'

leftText.classList.remove("overlay-text-left-animation-out");

overlay.classList.remove("open-sign-in");

rightText.classList.remove("overlay-text-right-animation");

// Add classes for animations

accountForm.className += " form-left-slide-out"

rightText.className += " overlay-text-right-animation-out";

overlay.className += " open-sign-up";

leftText.className += " overlay-text-left-animation";

// hide the sign up form once it is out of view

setTimeout(function(){

accountForm.classList.remove("form-left-slide-in");

accountForm.style.display = "none";

accountForm.classList.remove("form-left-slide-out");

}, 700);

// display the sign in form once the overlay begins moving right

setTimeout(function(){

signinForm.style.display = "flex";

signinForm.classList += " form-right-slide-in";

}, 200);

}

// Open the Sign In page

openSignIn = () =>{

// Remove classes so that animations can restart on the next 'switch'

leftText.classList.remove("overlay-text-left-animation");

overlay.classList.remove("open-sign-up");

rightText.classList.remove("overlay-text-right-animation-out");

// Add classes for animations

signinForm.classList += " form-right-slide-out";

leftText.className += " overlay-text-left-animation-out";

overlay.className += " open-sign-in";

rightText.className += " overlay-text-right-animation";

// hide the sign in form once it is out of view

setTimeout(function(){

signinForm.classList.remove("form-right-slide-in")

signinForm.style.display = "none";

signinForm.classList.remove("form-right-slide-out")

},700);

// display the sign up form once the overlay begins moving left

setTimeout(function(){

accountForm.style.display = "flex";

accountForm.classList += " form-left-slide-in";

},200);

}

// When a 'switch' button is pressed, switch page

openSignUpButton.addEventListener("click", openSignUp, false);

openSignInButton.addEventListener("click", openSignIn, false);

</script>

</body>

</html>

**6.4.2 ORPHANAGE HEADER**

<header>

<div class="header-area ">

<div class="header-top\_area">

<div class="container-fluid">

<div class="row">

<div class="col-xl-6 col-md-12 col-lg-8">

<div class="short\_contact\_list">

<ul>

<li><a href="#"> <i class="fa fa-phone"></i> +1 (454) 556-5656</a></li>

<li><a href="#"> <i class="fa fa-envelope"></i>Orpahange@gmail.com</a></li>

</ul>

</div>

</div>

<div class="col-xl-6 col-md-6 col-lg-4">

<div class="social\_media\_links d-none d-lg-block">

<a href="#">

<i class="fa fa-facebook"></i>

</a>

<a href="#">

<i class="fa fa-pinterest-p"></i>

</a>

<a href="#">

<i class="fa fa-linkedin"></i>

</a>

<a href="#">

<i class="fa fa-twitter"></i>

</a>

</div>

</div>

</div>

</div>

</div>

<div id="sticky-header" class="main-header-area">

<div class="container-fluid">

<div class="row align-items-center">

<div class="col-xl-3 col-lg-3">

<div class="logo">

<a href="orphan.php">

<img src="img/logo.png" alt="">

</a>

</div>

</div>

<div class="col-xl-9 col-lg-9">

<div class="main-menu">

<nav>

<ul id="navigation">

<li><a href="orphan.php">home</a></li>

<li><a href="#">Adoption <i class="ti-angle-down"></i></a>

<ul class="submenu">

<li><a href="orphan1.php">Add Adoption</a></li>

<li><a href="orphan2.php">View Adoption</a></li>

</ul>

</li>

<li><a href="#">Adoption Request <i class="ti-angle-down"></i></a>

<ul class="submenu">

<li><a href="orphan3.php">View Request</a></li>

</ul>

</li>

<li><a href="#">Orphanage Story <i class="ti-angle-down"></i></a>

<ul class="submenu">

<li><a href="orphan4.php">Add Story</a></li>

<li><a href="orphan5.php">View Story</a></li>

</ul>

</li>

<li><a href="#">Donation<i class="ti-angle-down"></i></a>

<ul class="submenu">

<li><a href="orphan6.php">View Donation</a></li>

</ul>

</li>

</ul>

</nav>

<div class="Appointment">

<div class="book\_btn d-none d-lg-block">

<a href="ologout.php">Logout</a>

</div>

</div>

</div>

</div>

<div class="col-12">

<div class="mobile\_menu d-block d-lg-none"></div>

</div>

</div>

</div>

</div>

</div>

</header>

**6.4.3 ORPHANAGE UPDATE**

<?php

define('DIR', '../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(!isset($\_SESSION["o\_id"])) {

header("location:logreg.php");

}

$id=$\_GET['id'];

$stmtss=$admin -> ret("SELECT \* FROM `story` WHERE `s\_id`='$id'");

$rowss=$stmtss->fetch(PDO::FETCH\_ASSOC);

?>

<!doctype html>

<html class="no-js" lang="zxx">

<head>

<meta charset="utf-8">

<meta http-equiv="x-ua-compatible" content="ie=edge">

<title>Charifit</title>

<meta name="description" content="">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- <link rel="manifest" href="site.webmanifest"> -->

<link rel="shortcut icon" type="image/x-icon" href="img/favicon.png">

<!-- Place favicon.ico in the root directory -->

<!-- CSS here -->

<link rel="stylesheet" href="css/bootstrap.min.css">

<link rel="stylesheet" href="css/owl.carousel.min.css">

<link rel="stylesheet" href="css/magnific-popup.css">

<link rel="stylesheet" href="css/font-awesome.min.css">

<link rel="stylesheet" href="css/themify-icons.css">

<link rel="stylesheet" href="css/nice-select.css">

<link rel="stylesheet" href="css/flaticon.css">

<link rel="stylesheet" href="css/gijgo.css">

<link rel="stylesheet" href="css/animate.css">

<link rel="stylesheet" href="css/slicknav.css">

<link rel="stylesheet" href="css/style.css">

<!-- <link rel="stylesheet" href="css/responsive.css"> -->

</head>

<body>

<!--[if lte IE 9]>

<p class="browserupgrade">You are using an <strong>outdated</strong> browser. Please <a href="https://browsehappy.com/">upgrade your browser</a> to improve your experience and security.</p>

<![endif]-->

<!-- header-start -->

<?php include"oheader.php"?>

<!-- header-end -->

<!-- slider\_area\_start -->

<!-- footer\_start -->

<section class="contact-section">

<div class="container">

<div class="row">

<div class="col-12">

<h2 class="contact-title">Update donation story information</h2>

</div>

<div class="col-lg-8">

<form action="../controller/story.php" method="post" enctype="multipart/form-data">

<div class="row">

<div class="col-12">

<div class="form-group">

<input class="form-control" name="a" value="<?php echo $rowss['description']; ?>" id="subject" type="text" >

</div>

</div>

<div class="col-12">

<div class="form-group">

<input class="form-control" name="b" value="<?php echo $rowss['cost']; ?>" id="subject" type="text" >

</div>

</div>

<div class="col-12">

<div class="form-group">

<label>upload picture</label>

<input class="form-control" name="image" id="subject" type="file" placeholder="Enter child age">

</div>

</div>

</div>

<input type="hidden" name="e" value="<?php echo $rowss['o\_id']; ?>">

<input type="hidden" name="f" value="<?php echo $rowss['oname']; ?>">

<input type="hidden" name="g" value="<?php echo $rowss['phone']; ?>">

<input type="hidden" name="h" value="<?php echo $rowss['address']; ?>">

<input type="hidden" name="z" value="<?php echo $id; ?>">

<input type="submit" name="update" class="btn btn-warning" value="submit">

</form>

</div>

</div>

</div>

</section>

<!-- footer\_end -->

<!-- link that opens popup -->

<!-- JS here -->

<script src="js/vendor/modernizr-3.5.0.min.js"></script>

<script src="js/vendor/jquery-1.12.4.min.js"></script>

<script src="js/popper.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/owl.carousel.min.js"></script>

<script src="js/isotope.pkgd.min.js"></script>

<script src="js/ajax-form.js"></script>

<script src="js/waypoints.min.js"></script>

<script src="js/jquery.counterup.min.js"></script>

<script src="js/imagesloaded.pkgd.min.js"></script>

<script src="js/scrollIt.js"></script>

<script src="js/jquery.scrollUp.min.js"></script>

<script src="js/wow.min.js"></script>

<script src="js/nice-select.min.js"></script>

<script src="js/jquery.slicknav.min.js"></script>

<script src="js/jquery.magnific-popup.min.js"></script>

<script src="js/plugins.js"></script>

<script src="js/gijgo.min.js"></script>

<!--contact js-->

<script src="js/contact.js"></script>

<script src="js/jquery.ajaxchimp.min.js"></script>

<script src="js/jquery.form.js"></script>

<script src="js/jquery.validate.min.js"></script>

<script src="js/mail-script.js"></script>

<script src="js/main.js"></script>

</body>

</html>

**6.4.4 VIEW STORY**

<?php

define('DIR', '../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(!isset($\_SESSION["u\_id"])) {

header("location:logreg.php");

}

$vid=$\_GET['vid'];

$stmts=$admin -> ret("SELECT \* FROM `story` WHERE `s\_id`='$vid'");

$rows=$stmts->fetch(PDO::FETCH\_ASSOC);

?>

<!doctype html>

<html class="no-js" lang="zxx">

<head>

<meta charset="utf-8">

<meta http-equiv="x-ua-compatible" content="ie=edge">

<title>Charifit</title>

<meta name="description" content="">

<meta name="viewport" content="width=device-width, initial-scale=1">

<!-- <link rel="manifest" href="site.webmanifest"> -->

<link rel="shortcut icon" type="image/x-icon" href="img/favicon.png">

<!-- Place favicon.ico in the root directory -->

<!-- CSS here -->

<link rel="stylesheet" href="css/bootstrap.min.css">

<link rel="stylesheet" href="css/owl.carousel.min.css">

<link rel="stylesheet" href="css/magnific-popup.css">

<link rel="stylesheet" href="css/font-awesome.min.css">

<link rel="stylesheet" href="css/themify-icons.css">

<link rel="stylesheet" href="css/nice-select.css">

<link rel="stylesheet" href="css/flaticon.css">

<link rel="stylesheet" href="css/gijgo.css">

<link rel="stylesheet" href="css/animate.css">

<link rel="stylesheet" href="css/slicknav.css">

<link rel="stylesheet" href="css/style.css">

<!-- <link rel="stylesheet" href="css/responsive.css"> -->

</head>

<body>

<!--[if lte IE 9]>

<p class="browserupgrade">You are using an <strong>outdated</strong> browser. Please <a href="https://browsehappy.com/">upgrade your browser</a> to improve your experience and security.</p>

<![endif]-->

<!-- header-start -->

<?php include"header.php"?>

<!-- header-end -->

<!-- slider\_area\_start -->

<div class="bradcam\_area breadcam\_bg overlay d-flex align-items-center justify-content-center">

<div class="container">

<div class="row">

<div class="col-xl-12">

<div class="bradcam\_text text-center">

<h3>Help us to Raise Fund</h3>

</div>

</div>

</div>

</div>

</div>

<!-- bradcam\_area\_end -->

<!-- popular\_causes\_area\_start -->

<div class="popular\_causes\_area pt-120 cause\_details ">

<div class="container">

<div class="row">

<div class="col-lg-12 col-md-12">

<div class="single\_cause">

<div class="thumb">

<img src="../controller/<?= $rows['image']; ?>">

</div>

<div class="causes\_content">

<div class="custom\_progress\_bar">

<div class="progress">

<div class="progress-bar" role="progressbar" style="width: 30%;" aria-valuenow="30" aria-valuemin="0" aria-valuemax="100">

<span class="progres\_count">

30%

</span>

</div>

</div>

</div>

<div class="balance d-flex justify-content-between align-items-center">

<span>Goal: Rs<?= $rows['cost']; ?> </span>

</div>

<h4>OUR requirement </h4>

<p><?= $rows['description']; ?></p>

</div>

</div>

</div>

</div>

</div>

</div>

<!-- popular\_causes\_area\_end -->

<div class="make\_donation\_area section\_padding">

<div class="container">

<div class="row justify-content-center">

<div class="col-lg-6">

<div class="section\_title text-center mb-55">

<h3><span>Make a Donation</span></h3>

</div>

</div>

</div>

<div class="row">

<div class="col-12">

<div class="donate\_now\_btn text-center">

<a href="payment.php?a=<?= $rows['o\_id']; ?> & b=<?= $rows['s\_id']; ?>" class="boxed-btn4">Donate Now</a>

</div>

</div>

</div>

</div>

</div>

<?php include"footer.php"?>

<!-- footer\_start -->

<!-- footer\_end -->

<!-- link that opens popup -->

<!-- JS here -->

<script src="js/vendor/modernizr-3.5.0.min.js"></script>

<script src="js/vendor/jquery-1.12.4.min.js"></script>

<script src="js/popper.min.js"></script>

<script src="js/bootstrap.min.js"></script>

<script src="js/owl.carousel.min.js"></script>

<script src="js/isotope.pkgd.min.js"></script>

<script src="js/ajax-form.js"></script>

<script src="js/waypoints.min.js"></script>

<script src="js/jquery.counterup.min.js"></script>

<script src="js/imagesloaded.pkgd.min.js"></script>

<script src="js/scrollIt.js"></script>

<script src="js/jquery.scrollUp.min.js"></script>

<script src="js/wow.min.js"></script>

<script src="js/nice-select.min.js"></script>

<script src="js/jquery.slicknav.min.js"></script>

<script src="js/jquery.magnific-popup.min.js"></script>

<script src="js/plugins.js"></script>

<script src="js/gijgo.min.js"></script>

<!--contact js-->

<script src="js/contact.js"></script>

<script src="js/jquery.ajaxchimp.min.js"></script>

<script src="js/jquery.form.js"></script>

<script src="js/jquery.validate.min.js"></script>

<script src="js/mail-script.js"></script>

<script src="js/main.js"></script>

</body>

</html>

**6.4.5 LOGOUT**

<?php

define('DIR','../');

require\_once DIR . 'config.php';

$control = new Controller();

$admin = new Admin();

if(isset($\_SESSION['o\_id'])){

session\_destroy();

unset($\_SESSION['o\_id']);

$admin -> redirect('index');

}

?>

**Chapter-7**

**TESTING**

**7. TESTING**

**7.1 INTRODUCTION**

Software testing is an investigation conducted to provide stack holders with information about the quality of the product or service under test. Testing has been defined as the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the futures of the software item. Software testing is the process used to access the quality of computer software.

It involves operation of a system or application under controlled conditions and evaluating the results. The controlled conditions include both normal and abnormal conditions. Testing should internationally attempt to make things go wrong to determine if things happen when they should. It is oriented to “detection”.

Software testing has three main purposes:

* The verification process confirms that the software meets its technical specifications. A “specification” is a description of a function in terms of a measurable output value given a specific input value under specific preconditions.
* The validation process confirms that the software meets the business requirements.
* A defect’s is a variance between the expected and actual result. The defects ultimate source may be traced to a fault introduced in the specification, design or development phase. Not all the defects will necessarily result in failure.

There are two types of software testing:

* Black box testing- Internal system design is not considered in this type of testing. Tests are based on requirements are functionality.
* White box testing- This testing is based on knowledge of thaw internal logic of an applications code. Also, known as glass box testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths and conditions.

A test case is a software testing documentation, which consists of events, action, input, output, expected result and actual result. Clinically defines a test case is an input and expected output result. This can be pragmatic as ‘for condition x your derived result is y’; where as other test cases described in more detail the input scenario and what result must be expected. It can occasionally be a series of steps but one with expected outcome. A test case should also contain a place for the actual result.

While box testing is applicable at the unit, integration and system levels of the software testing process.

**7.2 TESTING OBJECTIVES**

* Finding defects which may get created by the programmer while developing the software.
* Gaining confidence and providing information about the level of quality.
* To prevent defects.
* To make sure that the end result meets the business and user requirements.
* To endure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specification.

**7.3 TESTING METHODS**

System testing is the stage of implementation. This is to check whether system works accurately and efficiently before live operation commences. Testing is vital to the success of the system. The candidate system is a subject to a variety of tests: online response, volume, stress, recovery, security and usability tests. A series of tests are performed for the proposed system is ready for user acceptance testing.

**7.4 TESTING STEPS**

**7.4.1 UNIT TESTING**

Unit testing focuses efforts on the smallest unit of software design. This is known as module testing. The modules are tested separately. The test is carried out during programming stage itself. In this step, each module is found to be working satisfactory as regards to the expected output from the module.

**7.4.2 INTEGRATION TESTING**

Data can be lost across an interface. One module can have an adverse effect on another, sub functions, when combined, may not be linked in desired manner in major functions. Integration testing is a systematic approach for constructing the program structure, while at the same time conducting teat to uncover errors associated within the interface. The objective is to take unit tested modules and builds program structure. All the modules are combined and tested as a whole.

**7.4.3 VALIDATION**

* The function or performance characteristics confirm to application and are accepted.
* A deviation from specification is uncovered and a deficiency list is created.
* Proposed system under consideration has been tested by using validation test and found to be working satisfactory.

**7.4.4 OUTPUT TESTING**

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in a specific format. The output format on the screen is found to be correct. The format was designed in the system design time according to user needs. For the hard copy, also; the output comes as per the specified requirements by the user. Hence output testing did not result in any correction for the system.

**7.4.5 USER ACCEPTANCE TESTING**

User acceptance of a system is the key factor of the success of the system. The system under condition is tested for the user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes whenever required.

This is done in regarding to the following point:

* Input screen design.
* Output screen design.
* Online message should be guide to the user.
* Format of reports and other outputs.

**7.5 TEST CASES**

**7.5.1 ADMIN LOGIN FORM**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI.NO** | **TEST CONDITION** | **EXPECTED RESULT** | **RESULT** |
| 1. | If admin clicks on login button without entering username and password | Please fill out this field | Success |
| 2. | If only username is entered and password is not entered | Please fill out this field | Success |
| 3. | If only password is entered and username is not entered | Please fill out this field | Success |
| 4. | If curser moves over the field | Please fill out this field | Success |
| 5. | If username or password is incorrect | You have entered wrong username or password | Success |
| 6. | If the valid username and password is entered | System displays the main page | Success |

**7.5.2 ADD ORPHNAGE FORM**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI.NO** | **TEST CONDITION** | **EXPECTED RESULT** | **RESULT** |
| 1. | When orphanage center name is empty | Please fill out this field | Success |
| 2. | When email address is empty | Please fill out this field | Success |
| 3. | When password is empty | Please fill out this field | Success |
| 4. | When cursor is on phone number | Ten numeric characters only | Success |
| 5. | When phone number is less than or more than ten | Please match the requested format | Success |
| 6. | When address details is empty | Please fill out this field | Success |
| 7. | When upload photo details is empty | Please select a file | Success |

**7.5.3 ORPHANAGE LOGIN FORM**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI.NO** | **TEST CONDITION** | **EXCEPTED RESULTS** | **RESULT** |
| 1. | If email or password is incorrect | You have entered wrong email or password please try again | Success |
| 2. | If valid email id and password is entered | System displays the home page | Success |
| 3. | If @ is missing in email ID | Please include an ‘@’ in the email address. ‘email’ is missing and ‘@’. | Success |

**7.5.4 DONOR LOGIN FORM**

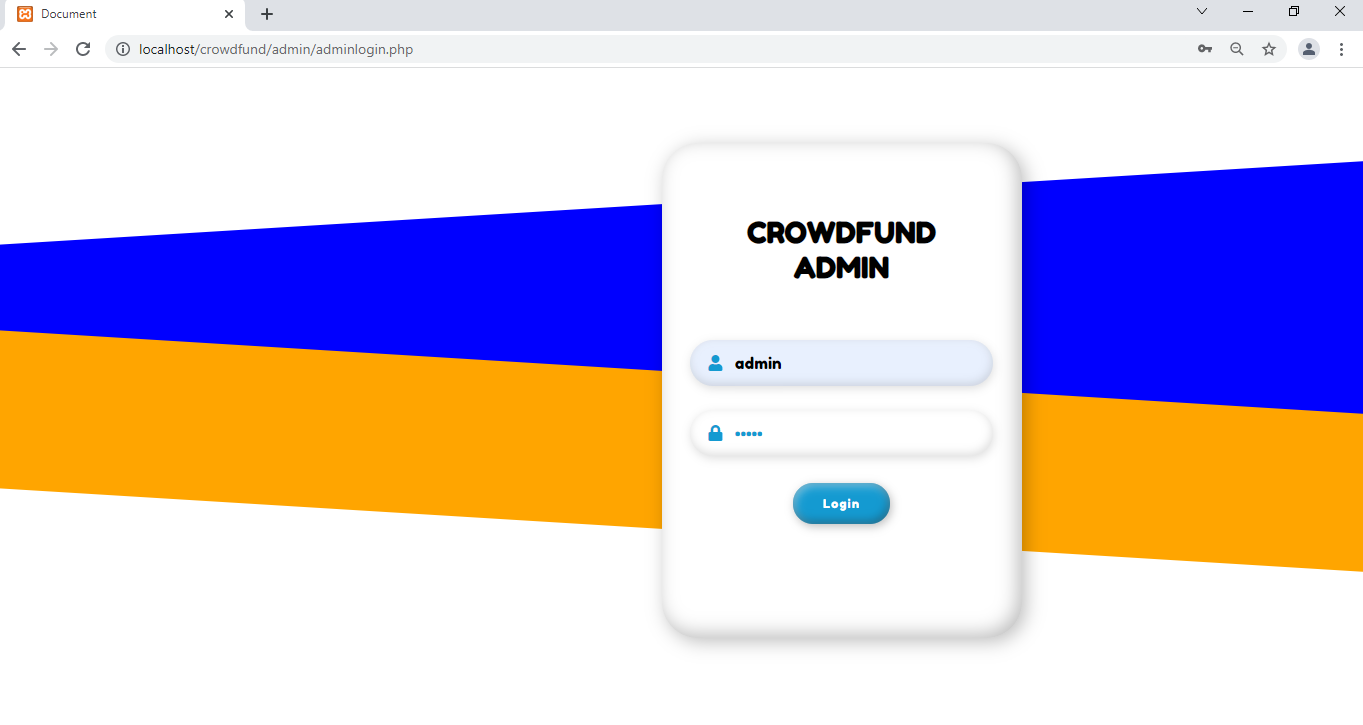
|  |  |  |  |
| --- | --- | --- | --- |
| **SI.NO** | **TEST CONDITION** | **EXCEPTED RESULTS** | **RESULT** |
| 1. | If email or password is incorrect | You have entered wrong email or password please try again | Success |
| 2. | If valid email id and password is entered | System displays the home page | Success |
| 3. | If @ is missing in email ID | Please include an ‘@’ in the email address. ‘email’ is missing and ‘@’. | Success |

**Chapter-8**

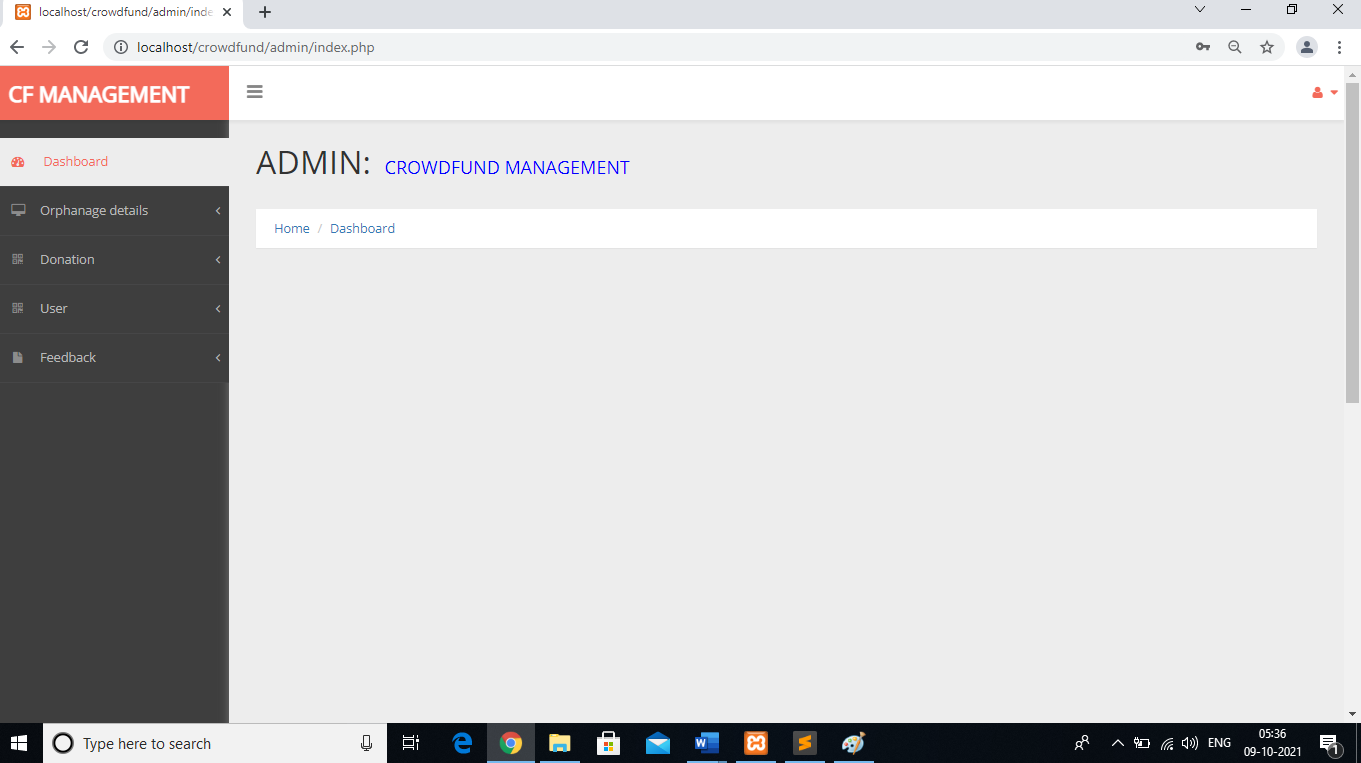
**USER INTERFACE**

**8.1 ADMIN MODULE**

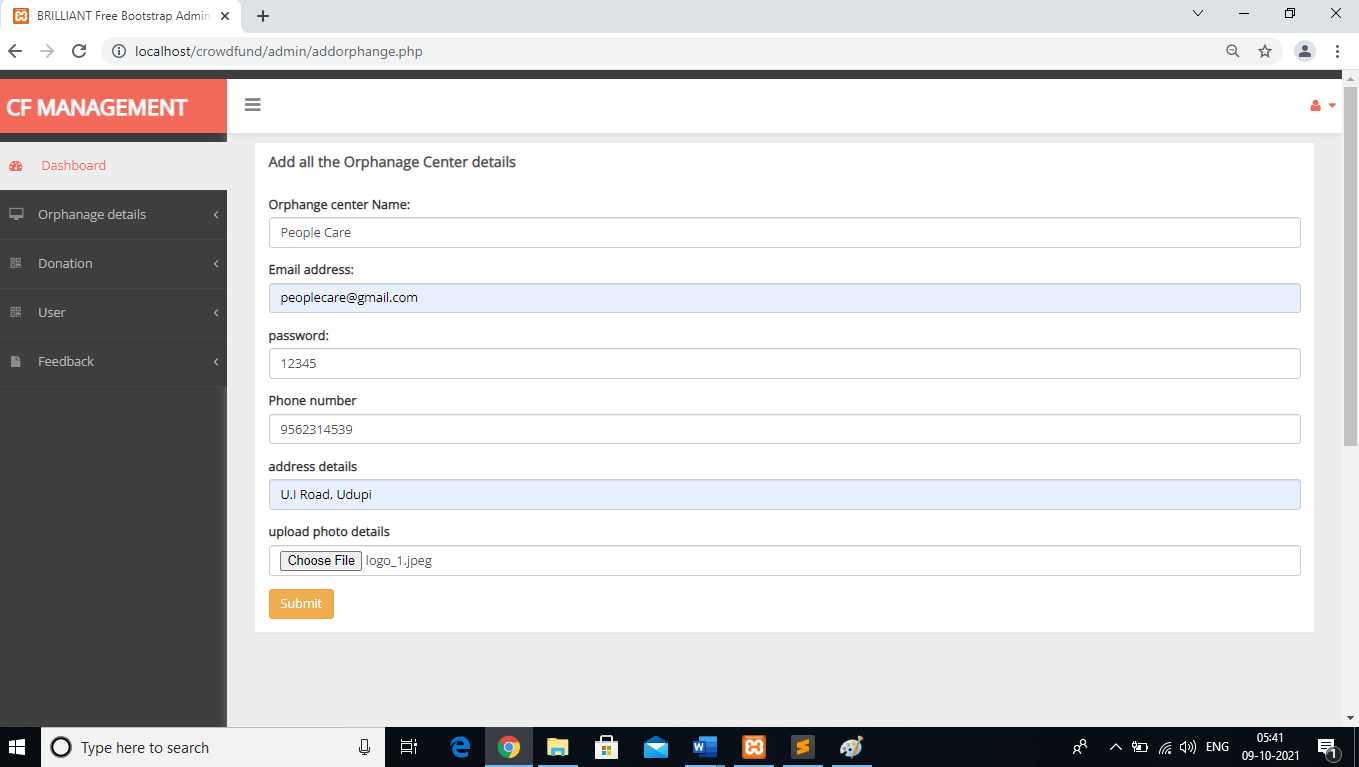
**8.1.1 LOGIN**

****

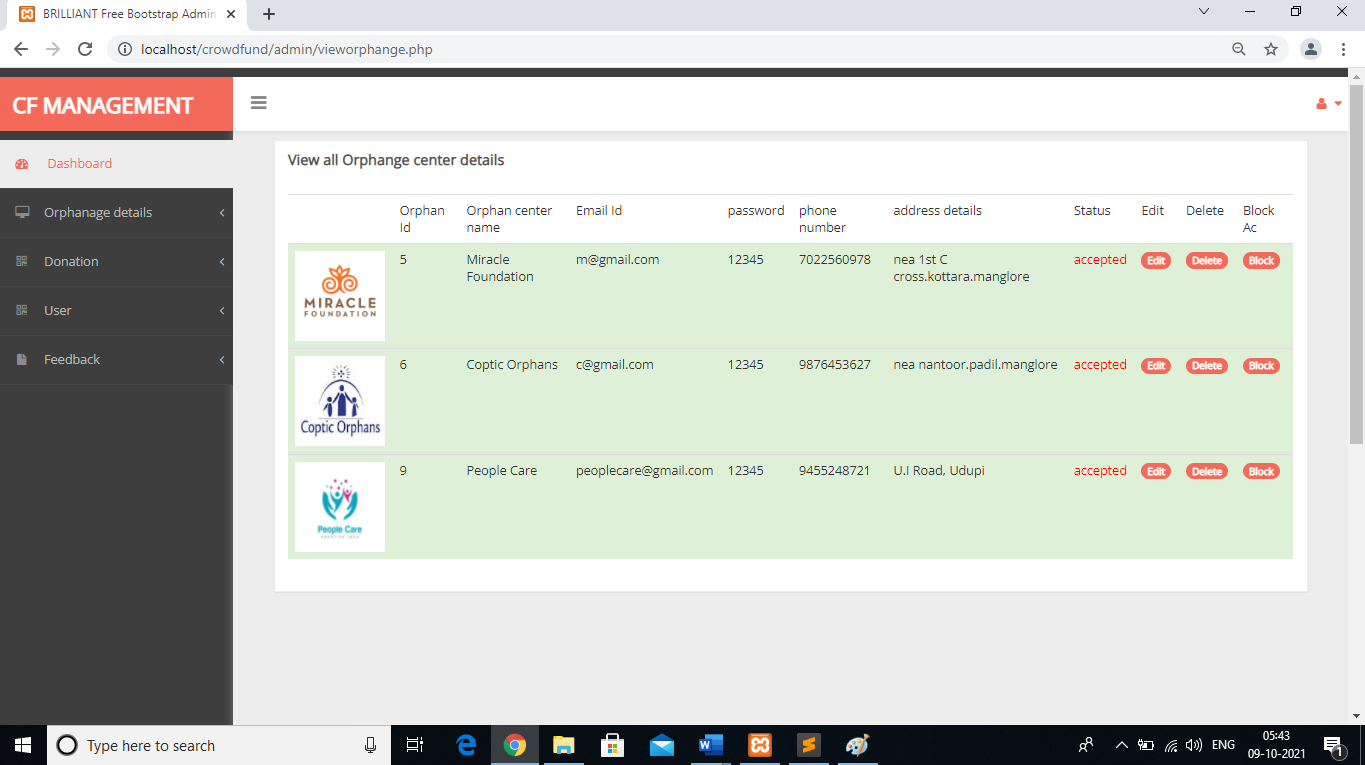
**8.1.2 HOME PAGE**

****

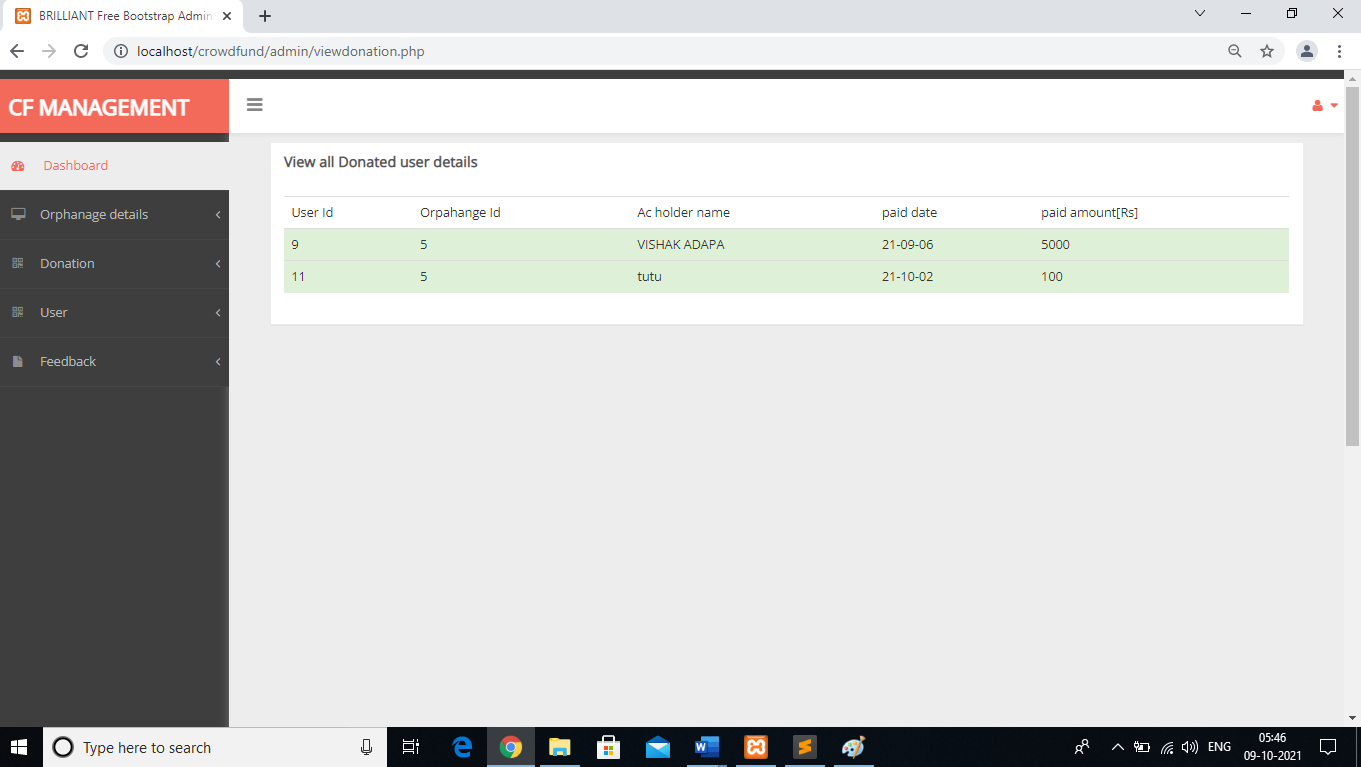
**8.1.3 ADD ORPHANAGE**

****

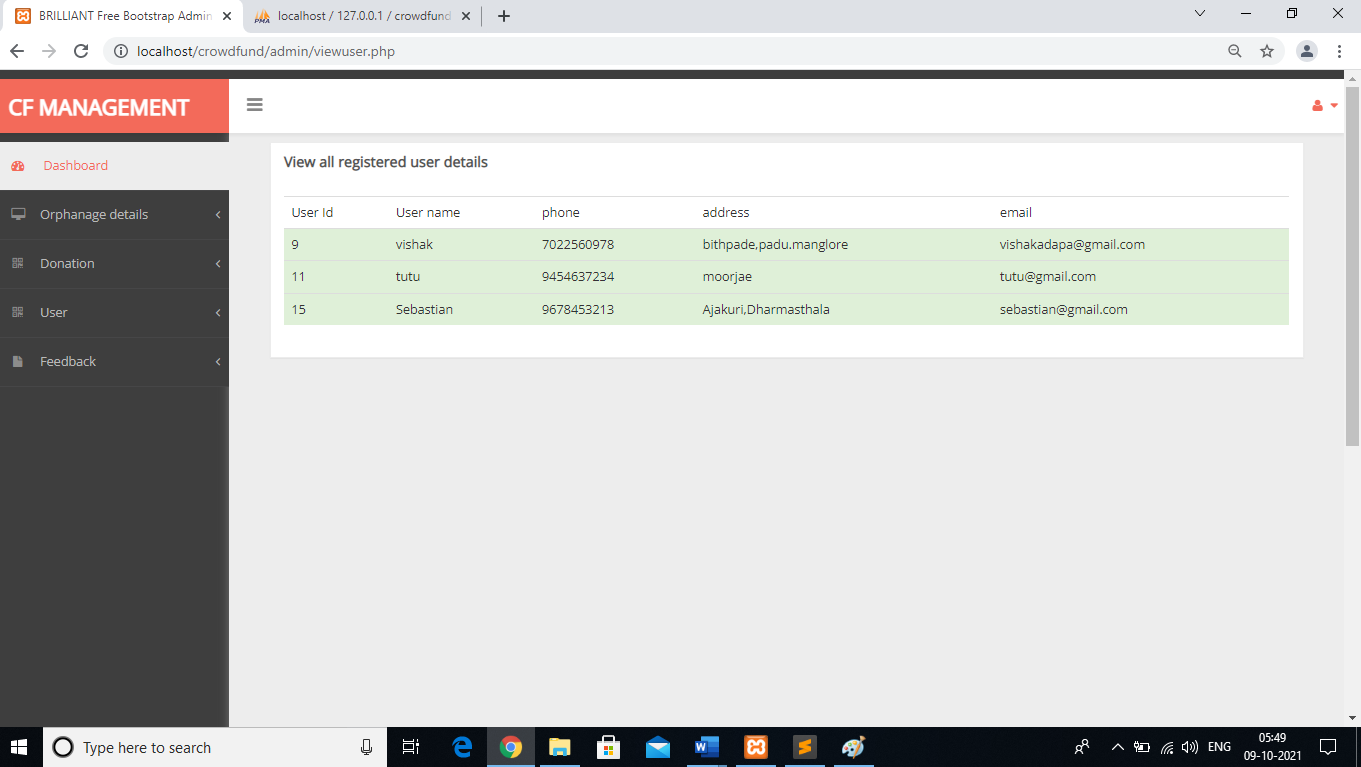
**8.1.4 VIEW ORPHANAGE**

****

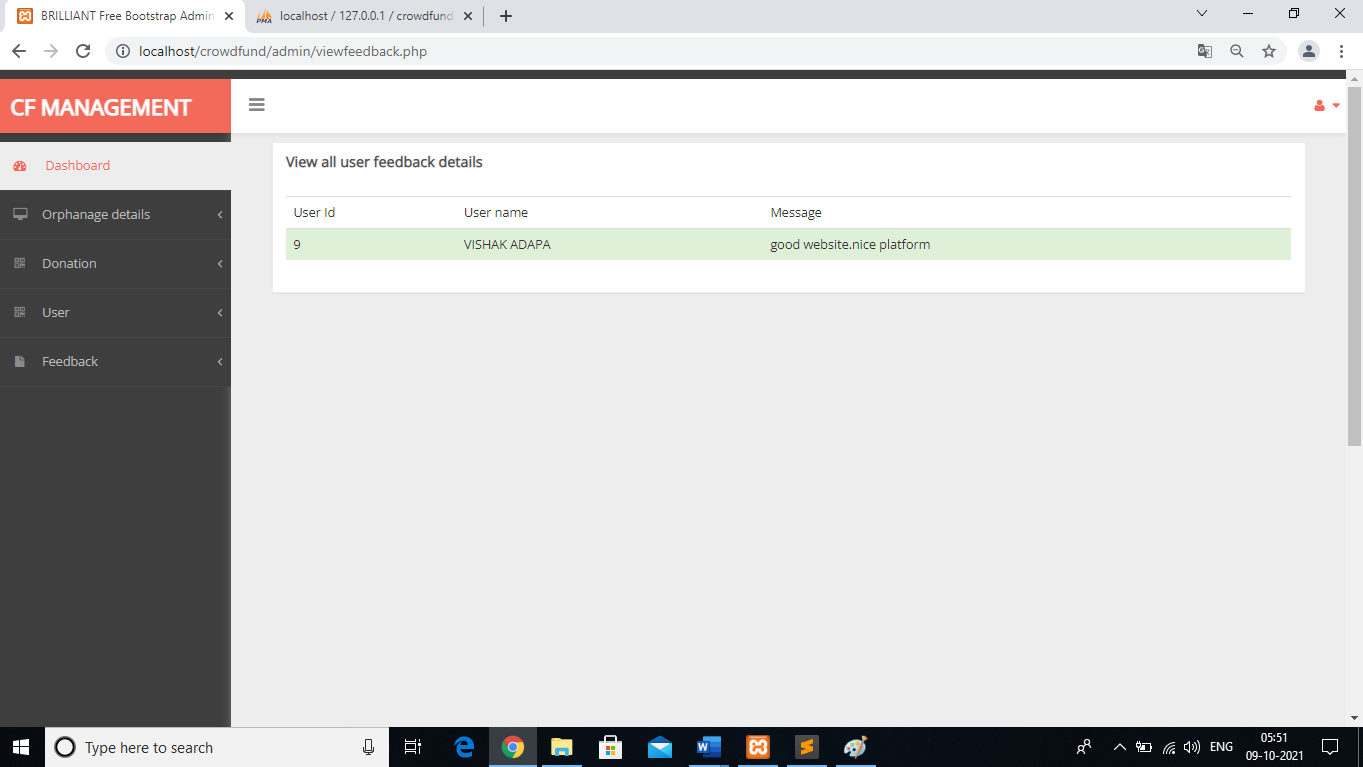
**8.1.5 VIEW DONATION**

****

**8.1.6 VIEW USER**

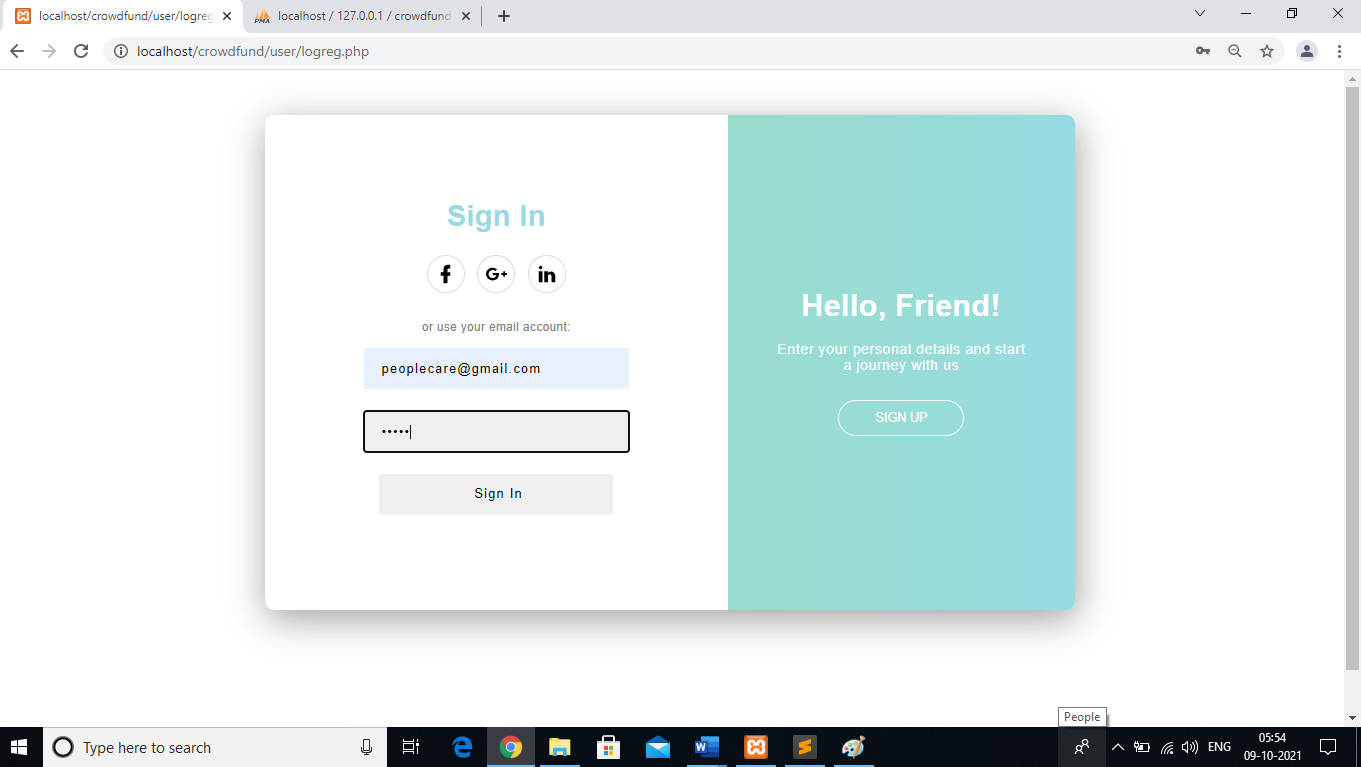
****

**8.1.7 VIEW FEEDBACK**

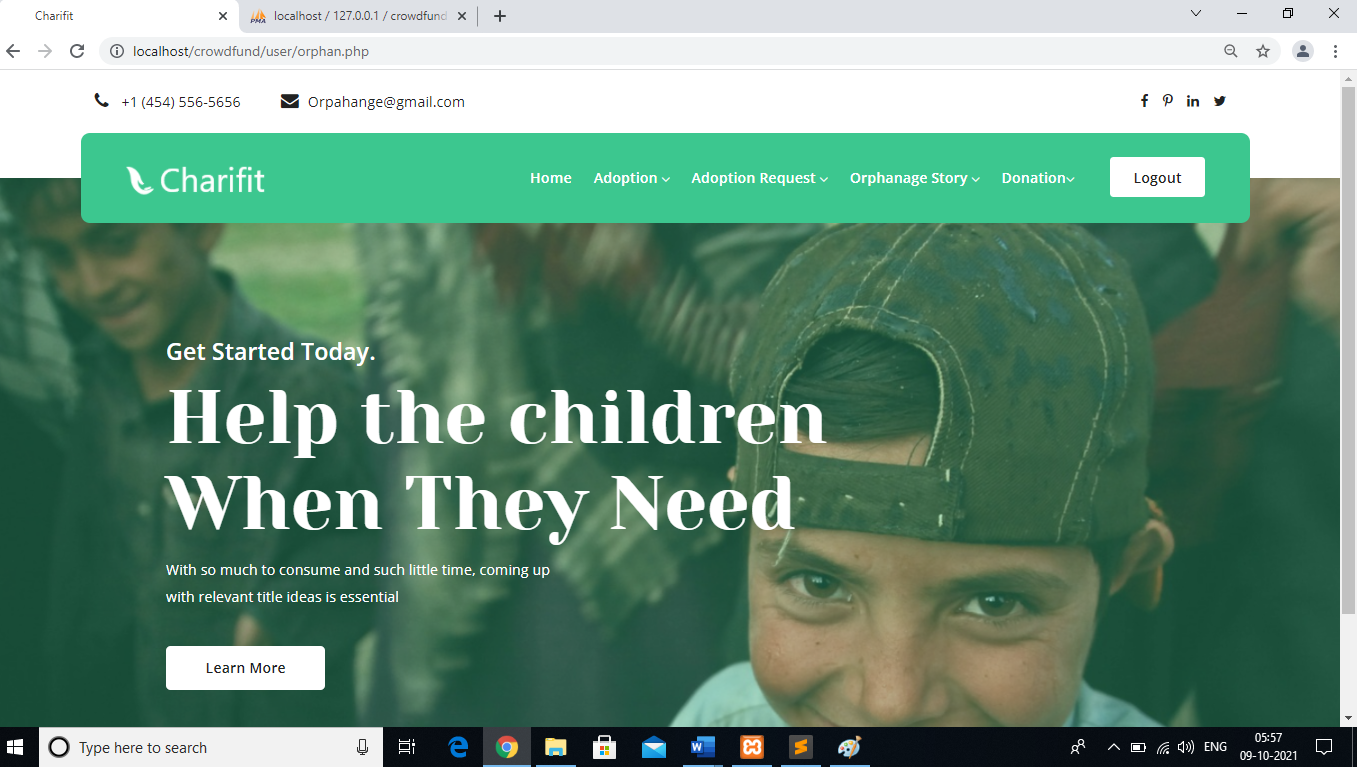
****

**8.2 ORPHANAGE MODULE**

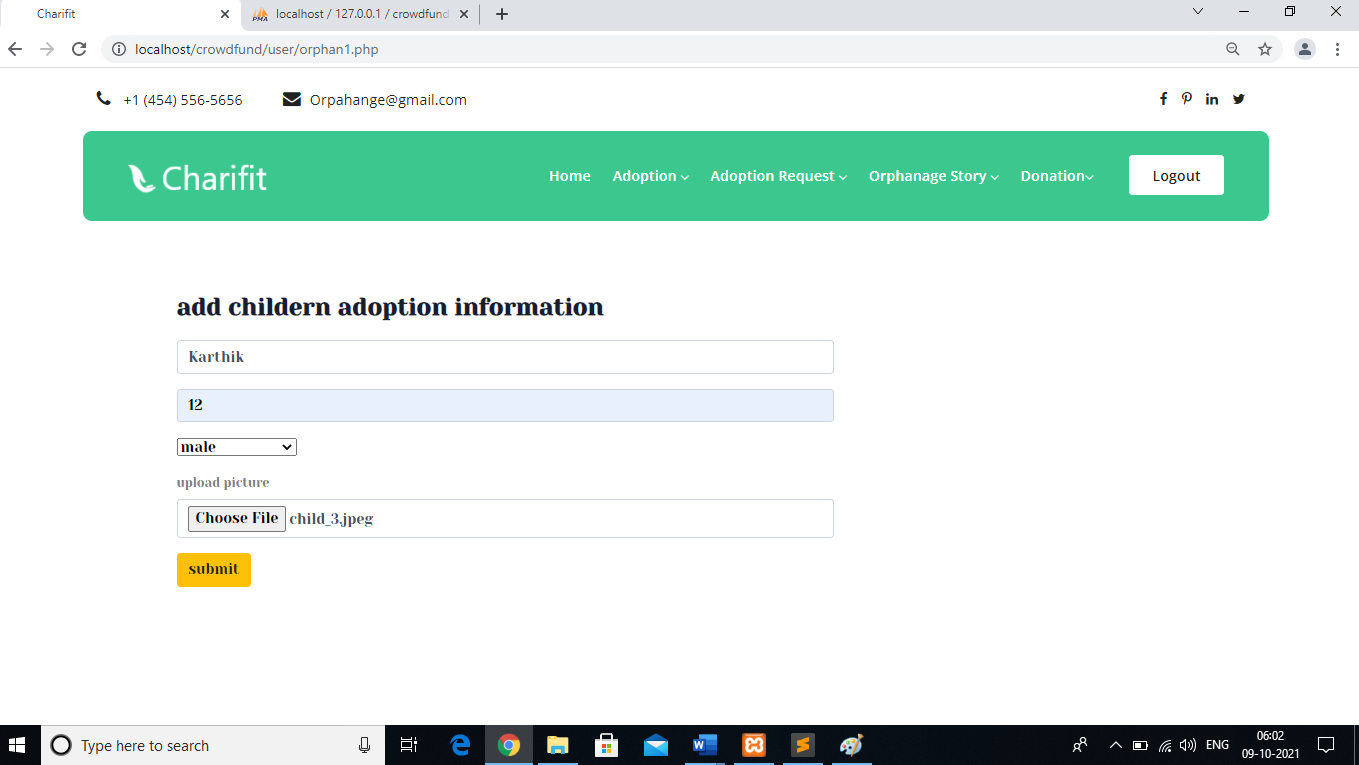
**8.2.1 SIGN IN**

****

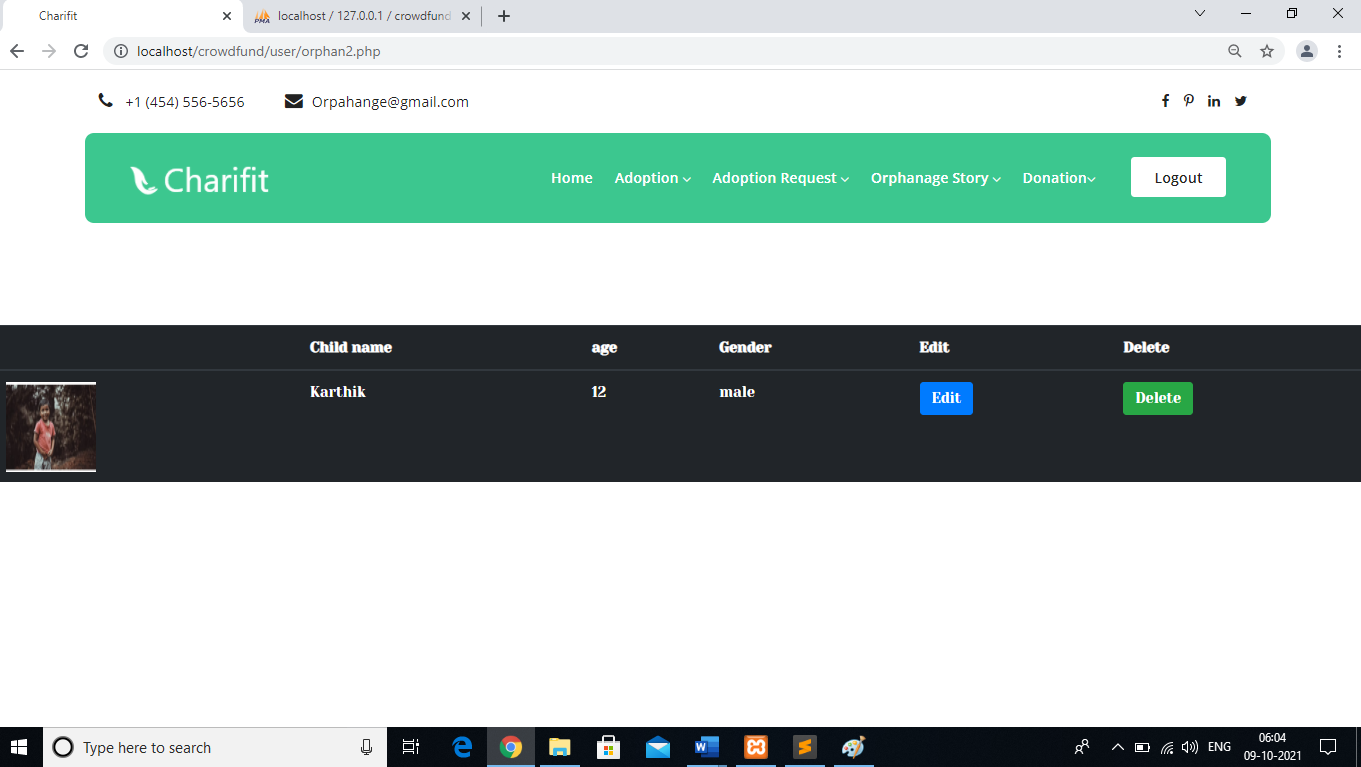
**8.2.2 HOME PAGE**

****

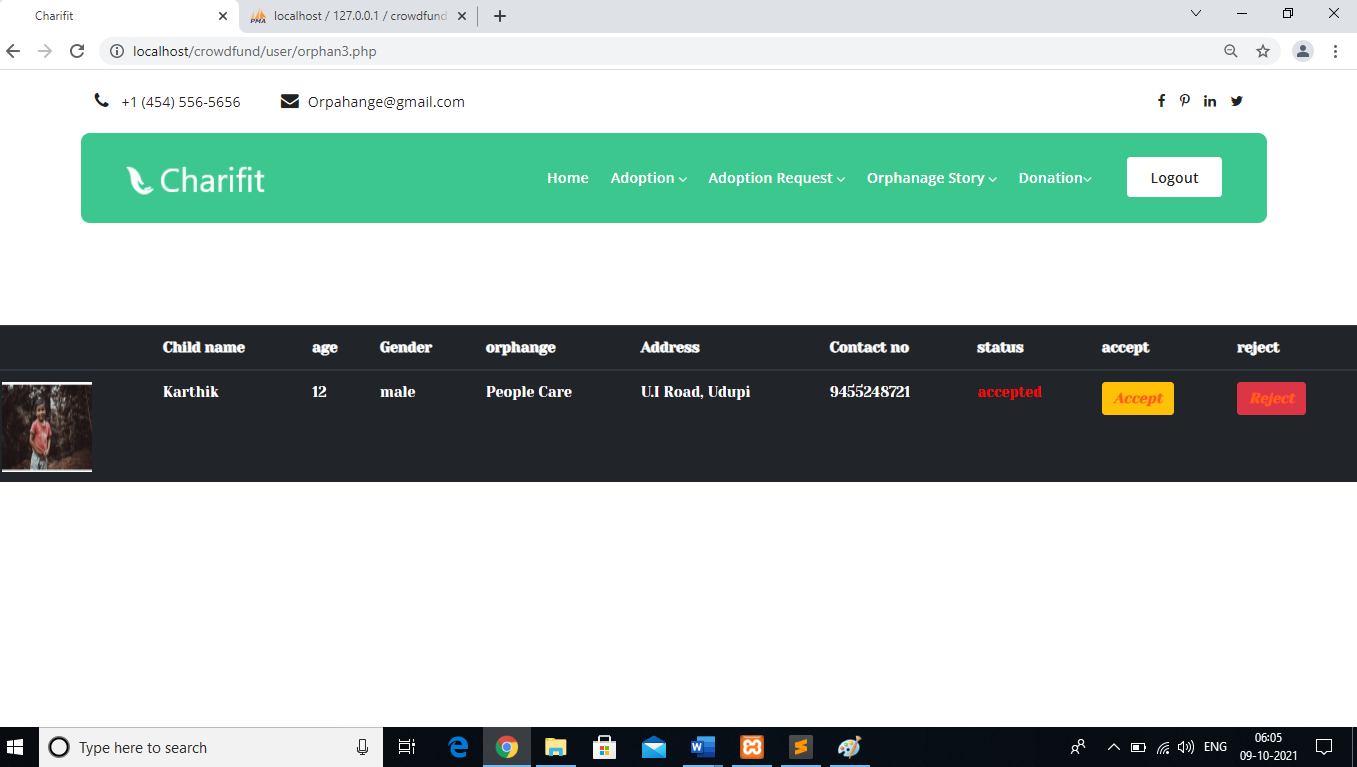
**8.2.3 ADD ADAPTION**

****

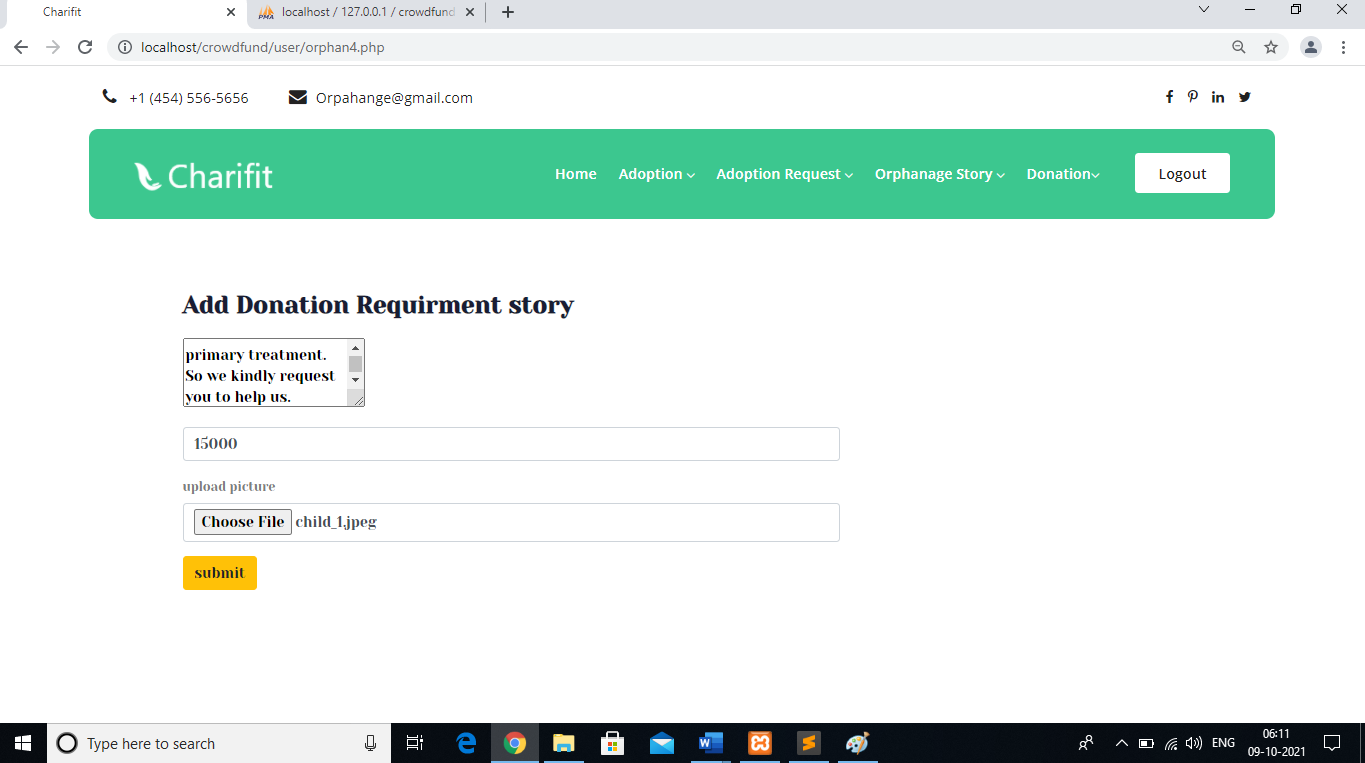
**8.2.4 VIEW ADAPTION**

****

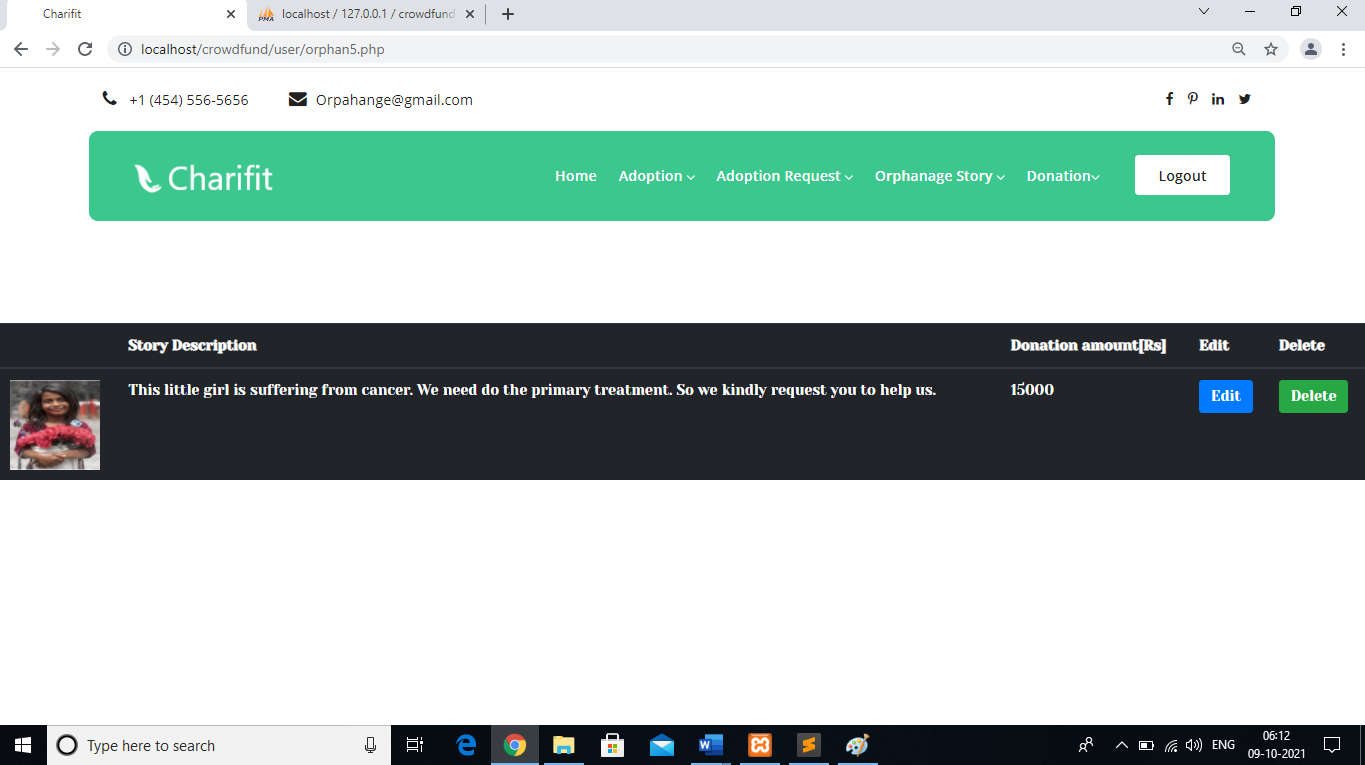
**8.2.5 VIEW ADAPTION REQUEST**

****

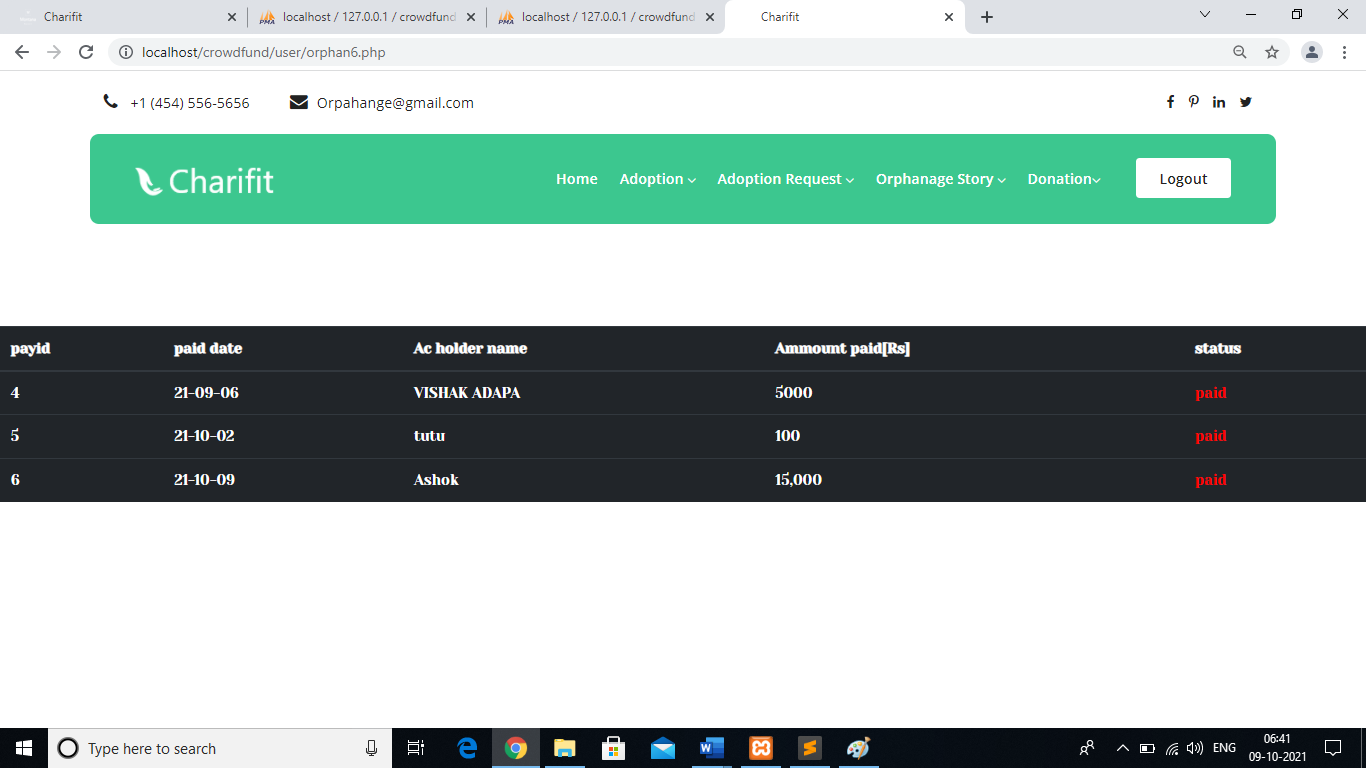
**8.2.6 ADD STORY**

****

**8.2.7 VIEW STORY**

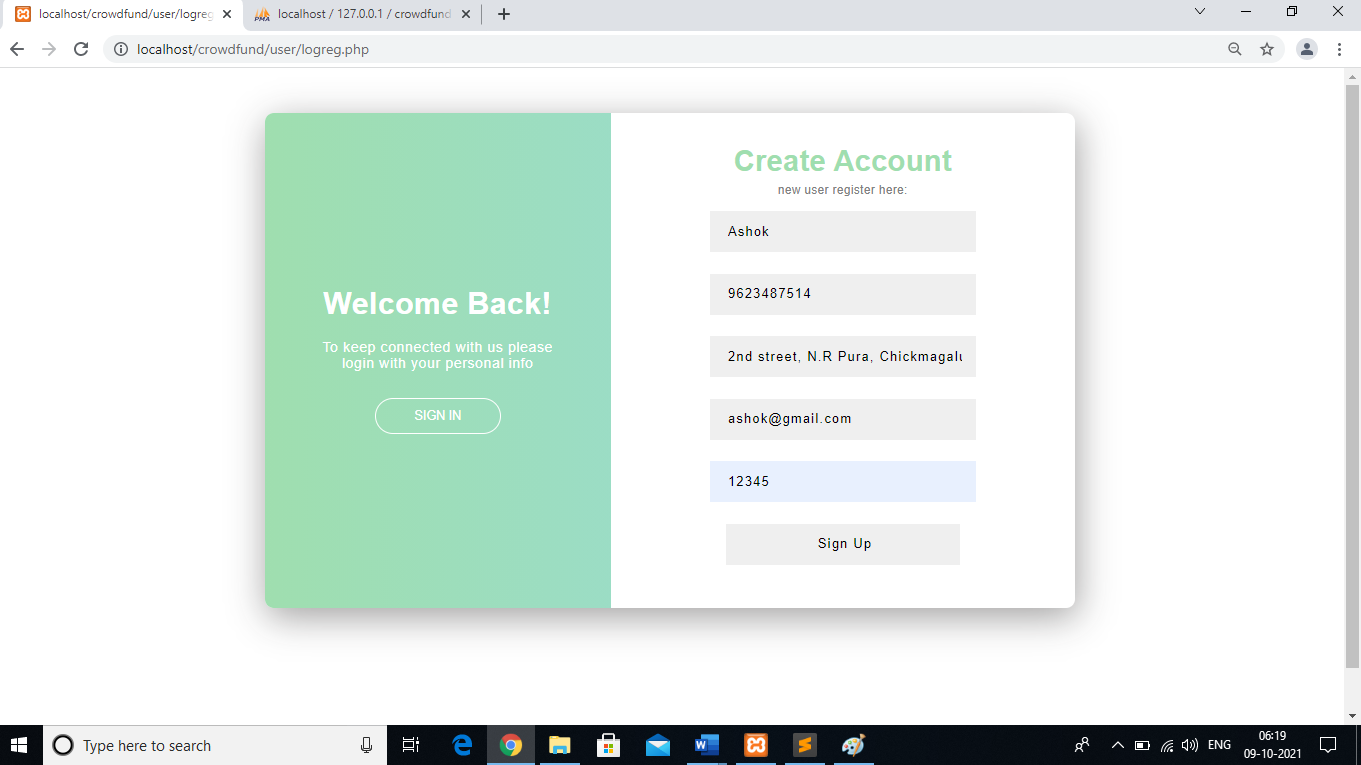
****

**8.2.8 VIEW DONATION**

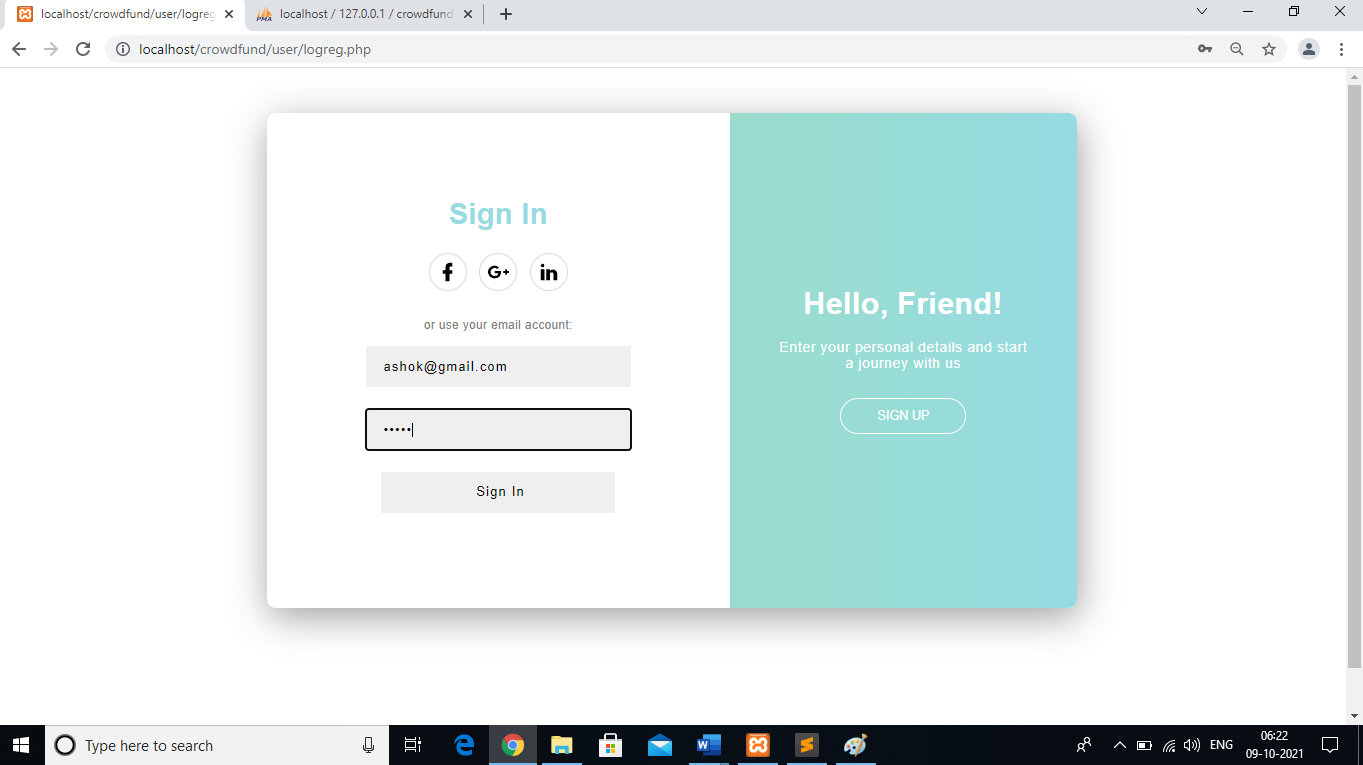
****

**8.3 DONOR MODULE**

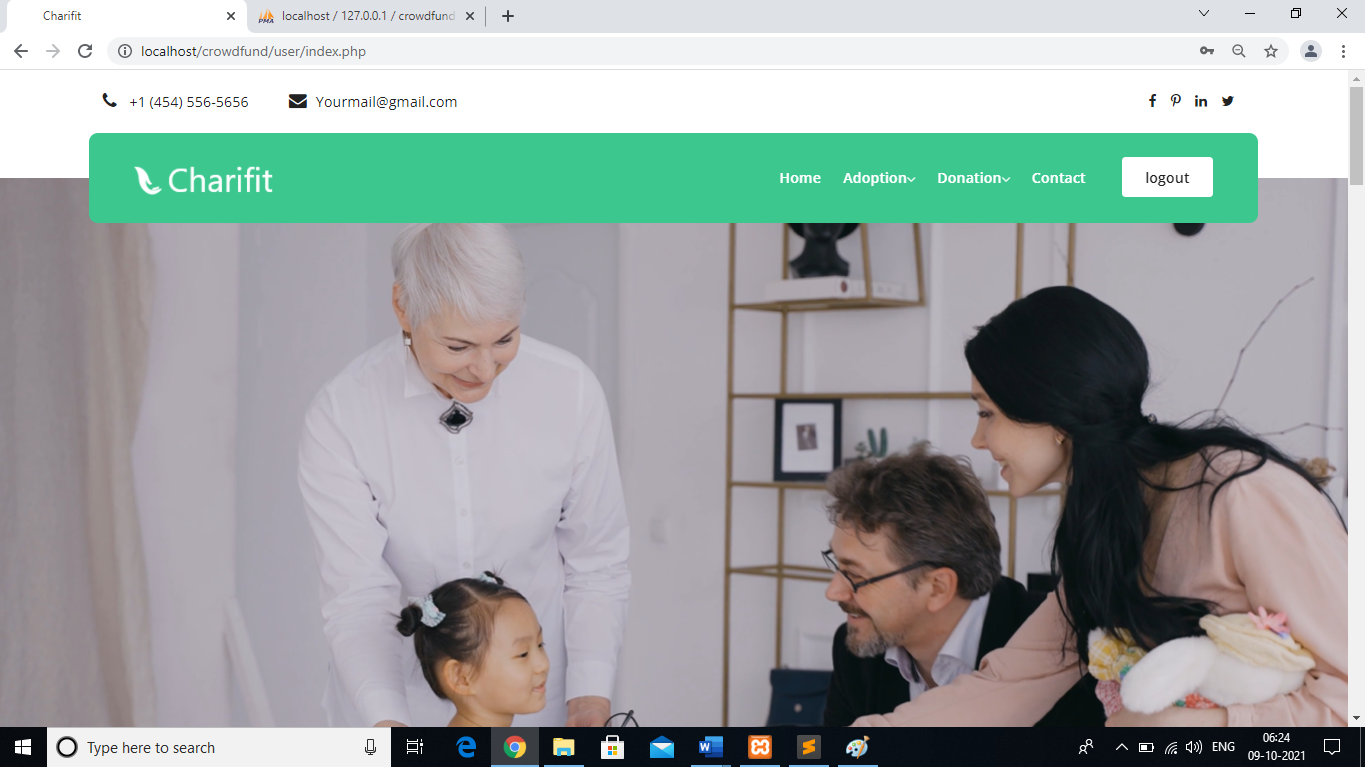
**8.3.1 DONOR REGISTRATION**

****

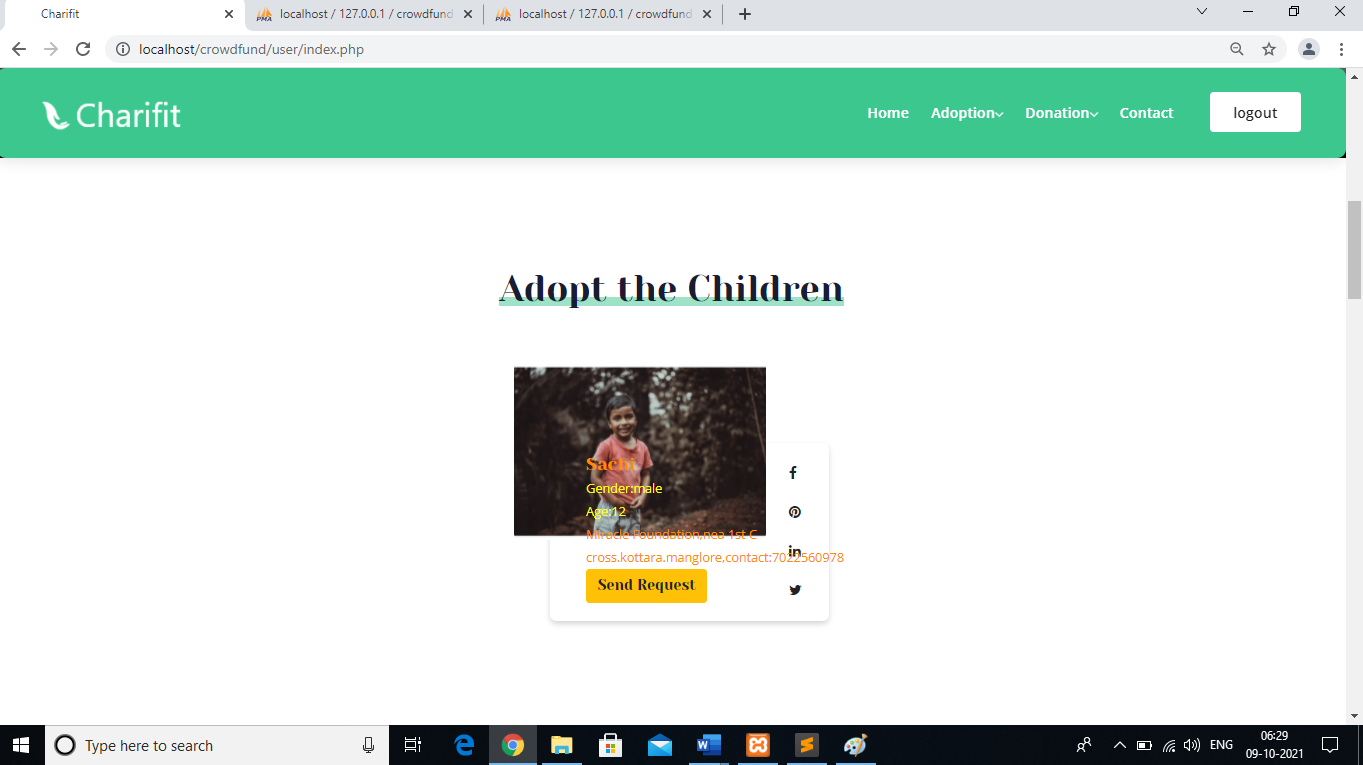
**8.3.2 LOGIN**

****

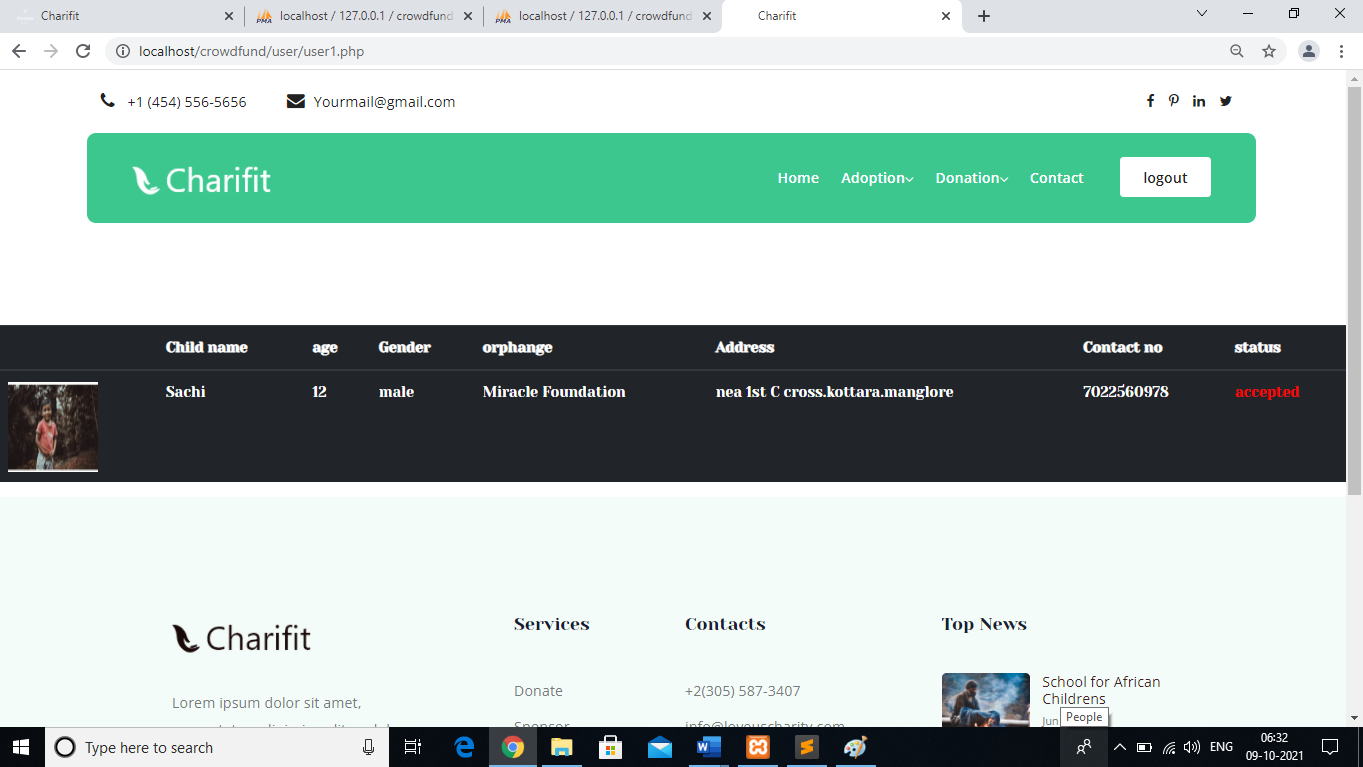
**8.3.3 HOMEPAGE**

****

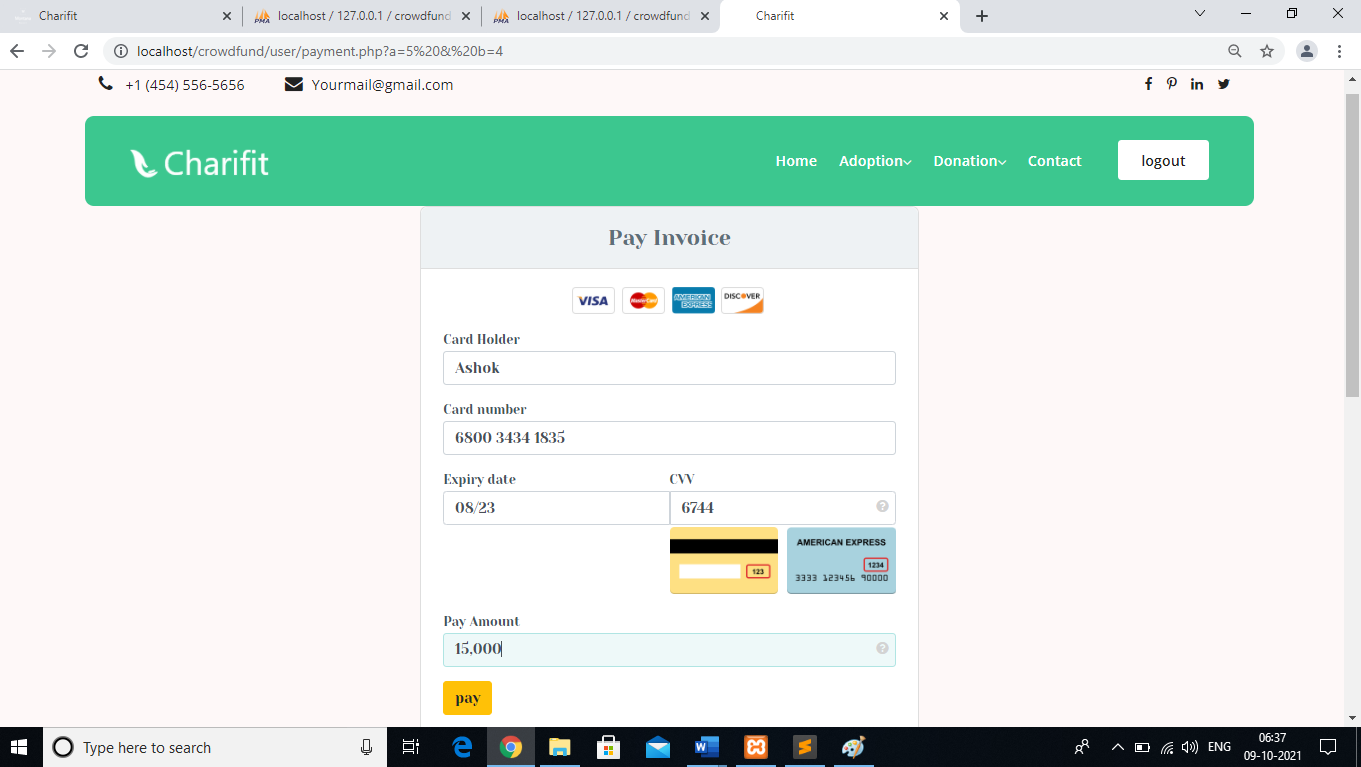
**8.3.4 ADOPTION**

****

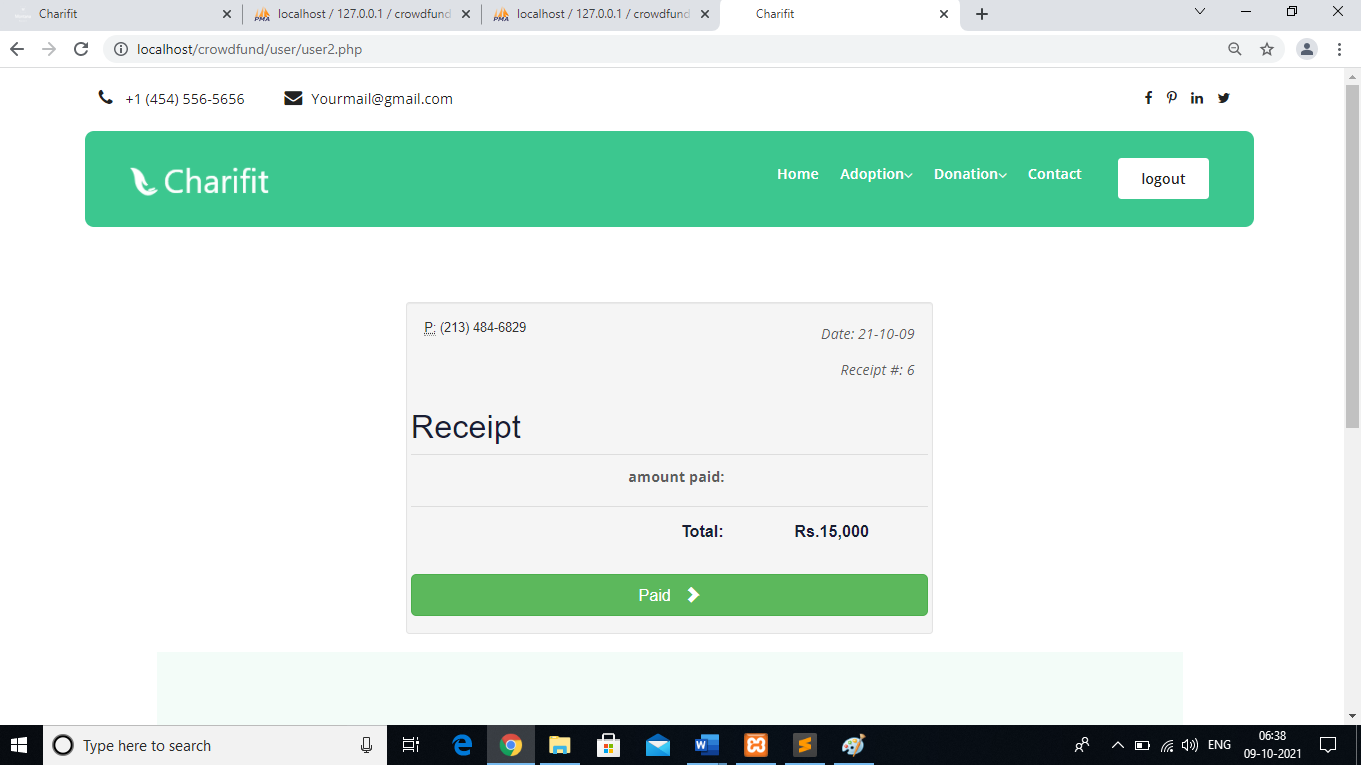
**8.3.5 VIEW ADAPTION**

****

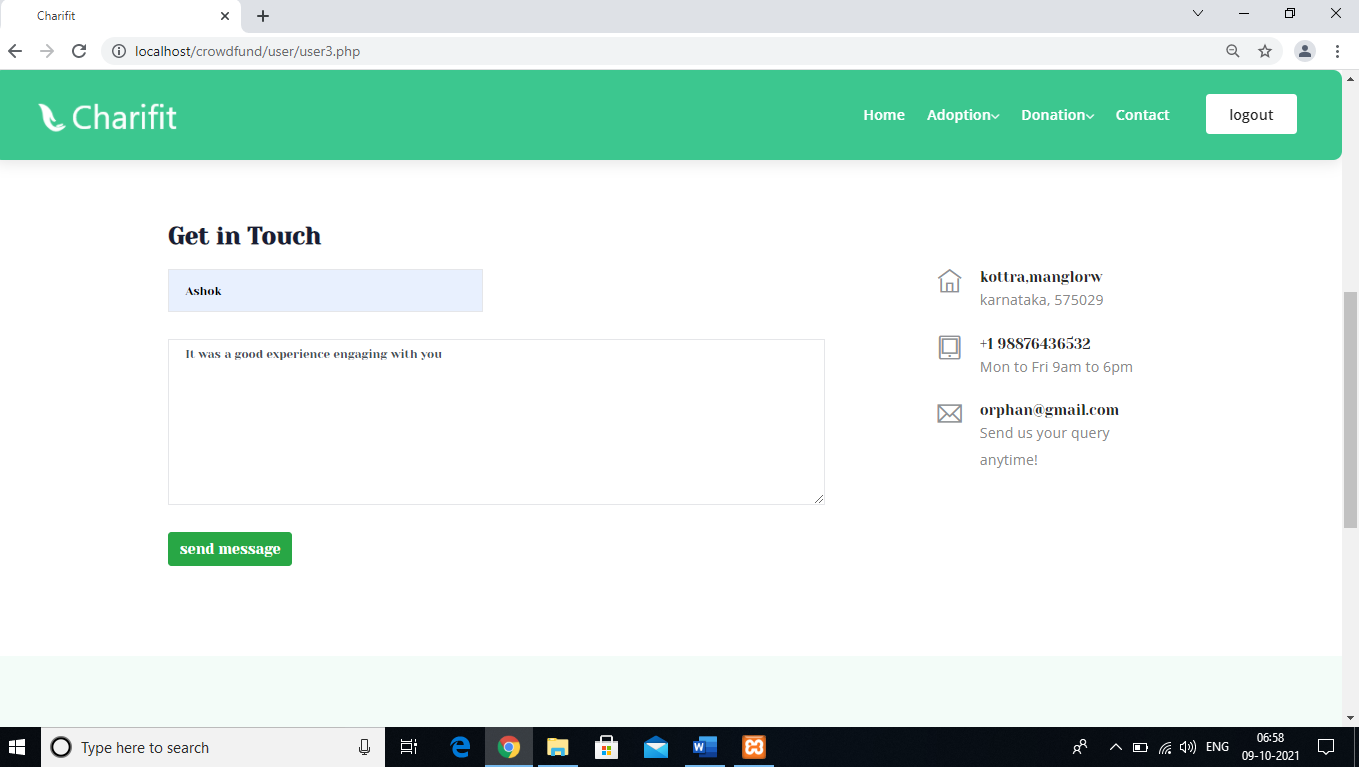
**8.3.6 DONATION**

****

**8.3.7 VIEW DONATION**

****

**8.3.8 CONTACT**

****

**Chapter-9**

**LIMITATIONS**

* Designed to run on Windows XP and further versions.
* Latest browsers like Internet explorer or Mozilla firefox or Google chrome required.
* Internet facility is essential to use this web application.

**Chapter-10**

**Future Scope**

* Develop an android application for the website.
* Develop a software version of the website.
* Develop chat application between donor and admin
* Add some additional feature to the system.

**Chapter-11**

**CONCLUSION**

**CONCLUSION**

The project “Crowd Funding” is a man-made project and, therefore, there may be mistakes and limitations. The ideas put up may be different. The terms and names may be different. However, our sincere effort was to give the best. The advanced techniques like sensor technology can be used in the future for measuring the quality of the product.

All the records stores in MySQL Database. This is totally self contained and works efficiently. It provides simple database rather than complex ones for high requirements and it provides good and easy graphical user interface new, naïve as well as experienced users of the computers.

This project has fulfilled all the objectives identified. The module has been developed in an attractive manner keeping in the mind about user convenience. So user with common computer language can handle it very easily.

**Chapter-12**

**BIBLIOGRAPHY**

**WEBSITES**

* PHP Tutorial -

- http://www.w3schools.com/php/default.asp

- http://www.tutorialspoint.com/php/

* MySQL Tutorial -   
  - http://www.w3schools.com/php/php\_mysql\_intro.asp  
  -http://www.tutorialspoint.com/mysql/index.htm
* JavaScript -

- http://www.w3schools.com/js/default.asp  
- http://www.tutorialspoint.com/javascript/index.htm

* CSS -  
  - http://www.w3schools.com/css/default.asp  
  - http://www.tutorialspoint.com/css/index.htm
* HTML -  
  - http://www.w3schools.com/html/default.asp  
  -http://www.tutorialspoint.com/html/index.htm  
  - <http://www.tutorialspoint.com/html5/index.htm>
* Question and answer site:   
  [www.stackoverflow.com](http://www.stackoverflow.com)

**BOOKS REFERRED**

* Sams Teach Yourself PHP, MySQL and Apache All in One (5th Edition)  
  Author: Julie Meloni
* Learning PHP, MySQL, JavaScript, and CSS: A Step-by-Step Guide   
  Author: Robin Nixon