

# **DEPARTMENT OF COMPUTER SCIENCE**

## **SURANA COLLEGE**



**A PROJECT REPORT ON**  
**WARM HEARTS**  
**WHICH IS BONAFIDE WORK OF**  
**SANGEETHA S [19KXSB7085]**  
**SHREYAS G BHARADWAJ [19KXSB7094]**

Students of Bachelor of Computer Science submitted in partial  
fulfillment of the requirement of the award of  
DEGREE IN B.C.A prescribed by BANGLORE UNIVERSITY 2022

Mrs. RASHMI ESHWAR

(PROJECT GUIDE)

EXAMINAR: \_\_\_\_\_

Dr. A SRINIVAS

HOD (COMPUTER SCIENCE)

\_\_\_\_\_



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

## DEPARTMENT OF COMPUTER SCIENCE SURANA COLLEGE



A PROJECT REPORT ON  
**WARM HEARTS**  
WHICH IS BONAFIDE WORK OF  
**SANGEETHA S [19KXSB7085]**  
**SHREYAS G BHARADWAJ [19KXSB7094]**

**Student of Bachelor of Computer Applications**

Submitted in partial fulfillment of the requirement of the completion  
of

DEGREE IN  
**BACHELOR OF COMPUTER APPLICATIONS**

Under the guidance of  
**Mrs. RASHMI ESHWAR**  
**(Asst. Prof)**

**DEPARTMENT OF COMPUTER SCIENCE**



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

## INDEX

SL.NO	CONTENT	PG.NO
1	Acknowledgement	3
2	Project Synopsis	4
3	Feasibility Report	7
4	Introduction To DBMS	8
5	Introduction To ASP.NET and C#	12
6	Requirement Analysis	13
7	SRS Document	15
8	System Design	20
9	Data Flow Diagram	22
10	Entity Relationship diagram	27
11	System Implementation (With Source Code)	29
12	Testing	64
13	Sample Out	67
14	Telegram Implementation	95
15	Future Enhancement And Conclusion	97
16	Bibliography	98



## ACKNOWLEDGEMENT

The successful completion of any work would be always incomplete unless we mention the valuable cooperation and assistance of those people who were a source of constant guidance and encouragement. The growth and completion of this project has been quite challenging and has been done with sincere effort.

It is not possible to develop a project and prepare a project report without the assistance and encouragement of many. We make use of this opportunity to thank all those who helped us in bringing out this project.

We would like to express our gratitude to **Dr Bhaskar Reddy** honorable principal, for giving us all the facilities for the completion of our project. We would like to express our gratitude to **Dr Srinivas A** (Head of Computer Science Department) for granting us permission to carry out this project work at college.

We sincerely thank our guide **Mrs. Rashmi Eshwar** and all teachers for their constant help for our project. A special thanks to **Mr. Sridhar Joshi** for his complete support and guidance. We also thank to all our family and friends for their valuable suggestions and encouragement.



# PROJECT SYNOPSIS

## OVERVIEW

Warm Hearts is based on web-based application and it has been developed using ASP NET and C#. The objective of this project is to avoid wastage of food and providing it to the orphans who are in need. This website will also connect donors and receivers i.e., Orphanage. Donor can donate food, money, books and clothes to orphanage that they wish. Donor can also visit to the orphanage and give their feedback.

The web portal will have an online interface in the form of a online classes that will allow orphans to attend classes and to gain extra curriculum activities provided by the volunteers.

## PROBLEM STATEMENT

Food wastage and problem of hunger are two huge-crisis arising in today's world. Tons of food is thrown as waste every year. Additionally, one-third of the food consumed is stated as surplus.

Nearly 40 per cent of the food produced in India is wasted every year due to fragmented food systems and inefficient supply chains.

## OBJECTIVE

The main objective of this project is to avoid wastage of food unnecessarily and providing it to the orphanage who are in need.

This may help orphanage to get sufficient needs. And also helps to get other activities to improve the skills of orphans rather than academics.

Donors get a clear description of every orphanage and they can donate to the particular orphanage that they wish to. They can donate food, money,



books, clothes or they can directly visit to the orphanage and help them. Visitors can also write blog on orphanage and can give feedback.

## COMPONENTS

This project will be divided into following separate components:

The e-commerce website/portal

The donor, receiver, food details, course details, blogs, books and clothes database.

Reporting of the uploading food by donor and accepting the food by receiver

The data security system.

## MODULAR DESCRIPTION

These are the main modules of the project:

### Donor Module:

- Donor module includes donating various categories namely food, money, books and clothes.
- Donor will add food details with expiry date for donation which will be received by the orphanage.
- Donor can give feedback on the orphanage.
- Donor can write blog on any orphanage they have visited.

### Receiver Module:

- Receiver will receive the donations.
- Receiver will get a notification when food is uploaded by donor.
- Orphanage can receive many courses from volunteers to gain extra skills for children excluding academics.
- Orphanage can see the visitors feedback which helps to improve more.



### **Food Donation Module:**

- This Module contains all the food details.
- It includes the type of food, number of people that can be feed and the expiry date of the food.
- Orphanage who are nearer to the donor address and who can receive the food before the expire of the food can accept the donation.

### **Admin Module:**

- Admin can see and control the overall activities of the website.
- Admin have all the access to modify or to delete the orphanage, blogs and feedback.
- Admin can see the accepted and awaited food, orphanage by their review in graphical representation.

### **Hardware requirements**

- Processor: Pentium IV, 2GHz and above.
- RAM: 2GB (minimum) or 4GB (recommended).
- Hard disk: 80GB or above.
- Monitor: 15" CRT or LCD monitor.
- Keyboard: Normal or Multimedia
- Mouse: Any compatible mouse.

### **Software requirements**

- Operating System: Windows 10/ Linux Kernel (4.0 or later) operating system
- Front End: BOOTSTRAP 5, ASP.NET Frame Work
- Back End: C#, PYTHON
- Scripting Language: JAVA SCRIPT
- Data base: My SQL
- Tools & IDE: Visual Studio 2022, Git Bash, XAMPP



## FEASIBILITY REPORT

In a feasibility study, a proposed plan or project is evaluated for its practicality. As part of a feasibility study, a project or venture is evaluated for its viability in order to determine whether it will be successful. As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable

- Our project has separate areas that a feasibility study examines, described below:

### Technical Feasibility Study

The system technically feasible because here we are using ASP. Net and C# which is free to use for educational purposes. From the user's point of view, it is feasible because it can fulfill user's requirements and maintain their records.

### Operational Feasibility Study

Operational Feasibility is the measure of how well the proposed system solves the problem and specifies the requirements in the system development. Degree of providing service to requirements is analyzed along with how much easy product will be to operate and maintenance after deployment.

This application is feasible as this as this system can be implemented in the organization there is adequate support from the management and users. Being developed in C# so that the necessary operations are carried out automatically and it is user friendly.

### Economic Feasibility Study

Economic Feasibility study cost and benefit of the project is analyzed. Means under this feasibility study a detail analysis is carried out what will be cost of the project for development which includes all required cost for final development like hardware and software resource required, design and development cost and operational cost and so on.



## INTRODUCTION TO DBMS

The database is a collection of inter-related data which is used to retrieve, insert and delete the data efficiently. It is also used to organize the data in the form of a table, schema, views, and reports, etc.

For example: The college Database organizes the data about the admin, staff, students and faculty etc.

Advantages:

Controls database redundancy

Data sharing

Easy Maintenance

Reduce time

Multiple user interface

## CONNECTION OF DATABASE

Connection: While working with a database, the first step is to establish a connection. To establish a connection, the following parameters are required:

Database name or data source: The name of the database to which the connection can be made. You can work with only one database at a time.

Creating stored procedure: Right-click Stored Procedures, and then click New Stored Procedure.

On the Query menu, we should replace with our query to create a new table or to modify the existing table.



**SURANA COLLEGE**  
**Department Of Computer Science**  
**VI Semester Project 2022**

The screenshot shows the Microsoft Visual Studio interface. In the center, there is a code editor window titled "dbo.Procedure.sql" containing the following SQL code:

```
1 CREATE PROCEDURE [dbo].[Procedure]
2     @param1 int = 0,
3     @param2 int
4 AS
5     SELECT @param1, @param2
6     RETURN 0
7
```

On the left side, the "Server Explorer" pane is open, showing a tree view of "Data Connections" and "Tables". Under "Tables", several tables are listed: admin, blog, blogs, book, booking, clothes, contact, donor, fdetails, orphanage, receiver, rdetails, and volunteers. There are also "Views", "Stored Procedure", "Functions", "Synonyms", and "Types" sections.

On the right side, the "Solution Explorer" pane shows a project named "We-care.sln" with a folder structure under "We-care" containing various files like "abalastrama.p", "ablog.aspx", "about.aspx", "admin.aspx", "adonordetails.aspx", "aeditpwd.aspx", "aerobics.aspx", "afood.aspx", "akfood.jpg", "acorphanage.aspx", "app.py", "arcdetails.asp", "art.aspx", "bfood.jpg", "blogindex.aspx", "blog.aspx", "blog1.aspx", "booking.aspx", "book.aspx", "choosing\_bg.jpg", "clothes.aspx", "contact\_bg.jpg", "contact.aspx", "convolunteer.aspx", "course.aspx", and "dance.aspx".

**Selecting data from the database:** Once the connection is established, data can be fetched from the database by executing SQL select command against the database. The SQL statement is used to fetch data from a specific table in the database.

- Select \* from table\_name;

**Inserting data into the database:** We can insert new records into the database and the values for each row that are to be inserted in the database should be specified.

- Insert into table\_name values(“,”);

**Updating data into the database:** Existing records can also be updated into the database. New values should be specified for each row that needs to be updated into the database.

- Update table\_name set attribute\_name=value;

**Deleting data from the database:** Records can also be deleted from the database.

- Delete from table\_name;



# Data Dictionary

## Donor Registration Table

id	name	address	phno	email	password	role
6	Vasuda	Hanumantha...	6362093374	vasuda@gmail...	Vasu123	Donor
7	Sangeetha	Srinagar	9900107168	sangeetha@gmail...	Sangu321	Donor
8	Shreesta G Bhu.	Jayanager	6362099374	shreesta@gmail...	Shreesta123	Donor
9	VenuGopala	Sarjapura	997248066	venugopala@g...	Venu@123	Donor
NULL	NULL	NULL	NULL	NULL	NULL	NULL

### Query:

```
create table donor(id varchar(100), name varchar(100), address varchar(100), phno  
varchar(100), email varchar(100), password varchar(100), role int, PRIMARY KEY(id));
```

## Receivers Registration Table

id	name	address	phno	emailid	password	role
1	Pooja	Srinagar	6362093374	pooja@gmail.c...	srikanth	Receiver
2	Srikanth	RR nagar	7038839322	srikanth@gmail...	sri123	Receiver
NULL	NULL	NULL	NULL	NULL	NULL	NULL



**SURANA COLLEGE**  
**Department Of Computer Science**  
**VI Semester Project 2022**

**Query:**

```
create table receiver(id int identity(1,1) NOT NULL, name varchar(100), address  
varchar(100), phno varchar(100), email varchar(100), password varchar(100), role  
int, PRIMARY KEY(id));
```

**Upload Food Details Table**

The screenshot shows the Microsoft Visual Studio interface. The Server Explorer pane on the left displays the database schema, including tables like 'dbo.details' and 'fdetails'. The Solution Explorer pane on the right shows the project structure for 'We-care', which includes files such as 'abakashrama.aspx', 'about.aspx', 'admin.aspx', and 'adordetails.aspx'. The main workspace shows a grid of data from the 'dbo.details' table.

fid	name	phno	email	address	ftype	number	fimage	pdate	expirydate	status	dem
100	Pooja K	9900107168	pooja@gmail.com	Thyaagarajanagar	Non Vegetarian	10	bfood.jpg	30-08-2022	2022-09-30	Awaited	sang
101	Pooja	6362093374	pooja@gmail.com	Shivajinagar	Vegetarian	30	akfood.jpg	01-09-2022	31-10-2022	Accepted	pooj
102	Vasuda	6362093374	vasuda@gmail.com	JP Nagar	Vegetarian	10	sflood.jpg	01-09-2022	2022-09-02	Awaited	
103	admin	9900107168		Sadashivnagar	Vegetarian	10	sflood.jpg	08-09-2022	2022-09-08	Awaited	sang
104	Shreyas G Bharadwaj	9900107168	sangeetha@gmail.com	Jayanagar	Vegetarian	20	akfood.jpg	08-09-2022	08-09-2022	Accepted	sang
105								08-09-2022	2022-09-09	Accepted	
106	Sangeetha	9900107168	sangeetha@gmail.com	Jaynagar	Vegetarian	20	akfood.jpg	09-09-2022	2022-09-10	Awaited	
107	Sangeetha	9900107168	sangeetha@gmail.com	JP NAGARA	Vegetarian	20	akfood.jpg	09-09-2022	2022-09-17	Awaited	
108	Pooja	6362093374	pooja@gmail.com	Srinagar	Vegetarian	12	kalasin.jpg	09-09-2022	2022-09-11	Accepted	sang
109	Sangeetha	9900107168	sangeetha@gmail.com	Sandhyanagar	Vegetarian	11	abakashrama.png	09-09-2022	2022-09-10	Awaited	sang
110	Sangeetha	9900107168	sangeetha@gmail.com	Rajahinagar	Non veg	20	bfood.jpg	10-09-2022	2022-09-11	Awaited	
111	Sangeetha	9900107168	sangeetha@gmail.com	ShivajiNagar	Non Vegetarian	10	bfood.jpg	10-09-2022	2022-09-11	Awaited	
112								NULL	NULL	NULL	NULL
113								NULL	NULL	NULL	NULL

**Query:**

```
create table fdetails(fid varchar(100), name varchar(100), phno varchar(100), email  
varchar(100), address varchar(100), ftype varchar(100), number varchar(100), fimage  
varchar(100), pdate varchar(100), expirydate varchar(100), status  
varchar(100), demail varchar(100), PRIMARY KEY(fid));
```



## ASP.NET FRAMEWORK

- ❖ ASP.NET is a web application framework developed and marketed by Microsoft to allow programmers to build dynamic web sites.
- ❖ It allows you to use a full featured programming language such as C# or VB.NET to build web applications easily.
- ❖ ASP.NET extends the .NET platform with tools and libraries specifically for building web apps.

## C# LANGUAGE

- ❖ C# is a standard choice for Windows applications because of the native support for .NET frameworks.
- ❖ It provides tons of libraries, components, UI class libraries and other resources that speed the development. Web services and apps.
- ❖ It supports all the basic object-oriented language features including encapsulation, polymorphism, and inheritance.
- ❖ Automatic memory management with the help of the garbage collector makes sure that program does not leak memory and automatically manages life cycles of objects and releases once objects are not required.



## REQUIREMENT ANALYSIS

All possible requirements of the system to be developed and captured in this phase. Requirements are a set of functions and constraints that the end user expects from the system. The requirements gathered from the end user at the start of the software development phase. These requirements are analyzed for their validity, and the possibility of incorporating the requirements in the system to be developed is also studied. Finally, a requirement specification document is created which serves the purpose of guideline for the next phase of the model.

Requirements Analysis is the process of defining the expectations of the users that is to be built or modified. It involves all the tasks that are conducted to identify the needs of different stakeholders. Therefore, requirements analysis means to analyze, document, validate and manage software or system requirements.

Requirements are descriptions of the services that a software system must provide and the constraints under which it must operate. Requirements can range from high-level abstract statements of services or system constraints to detailed mathematical functions specifications.

### **Requirement may serve a dual function:**

- ❖ May be the basis of for a bid for a contract- therefore must be open to interpretation
- ❖ May be the basis for the contract itself- therefore must be defined in detail both the statements may be called requirements.

## REQUIREMENTS

### **Requirement analysis phase is divided into:**

- Hardware Requirements
- Software Requirements



- Functional Requirements
- Non-Functional Requirements

### **Hardware Requirements:**

These requirements include the **minimum processor speed, memory, and disk space required to install Windows**. There is no rigorous restrictions on the machine configuration. The model should be capable of working on all machines capable of supporting recent versions of any version of Windows operating system.

### **Software Requirements:**

The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product. The requirements can be obvious or hidden, known or unknown, expected or unexpected from client's point of view.

### **Functional Requirements:**

Functional requirements capture the intended behavior of the system. This behavior may be expressed as services, tasks or functions the system is required to perform. It specifies actions that a system must be able to perform, without taking physical constraints into consideration. Functional requirements the specify the input and output behavior of a system.

### **Non-Functional Requirements:**

A Non-functional requirement is a statement of how a system must behave; it is a constraint upon the systems behavior. Non-functional requirements specify all the remaining requirements not covered by the functional requirements. They specify criteria that judge the operation of a system, rather than specific behaviors.



# Software Requirement Specification (SRS) Document

SRS document is a contract between the development team and the customer. Once the SRS document is approved by the customer, any subsequent controversies are settled by referring the SRS document. This should be carefully written.

It is a formal and official document.

It resolves the conflict between user and developer.

Based on SRS, further development of the system takes place.

It is the fundamental document, bridges the gap between user requirements and developers view.

IEEE standard suggests the following structure for requirements documents:

## **1. Introduction**

### **1.1 Purpose of the Requirements Document.**

Food donation is a mission to end hunger and no wasting of food to make a hungry-free world. According to the latest survey, 1.3 billion tons of food is thrown as a waste every year. Additionally, one-third of the food consumed is stated as leftovers. The focus of this project is to reduce the amount of food wasted and being used to the needy people.

Therefore, an web-based application is developed by which a person can donate food with their capacity

The purpose of this document is to unambiguously specify the software requirements for a web-based application WARM HEARTS providing interaction between Donors and Receivers i.e, Orphanage and provide information for Donor about orphanage so that it gives clarity for Donors to select the orphanage that they want to donate. Donor can directly contact and visit the orphanage.

### **Intended Audience**

The Intended Audience for this project can be broadly classified under four main categories, these are:



- i. Donor
- ii. Receiver

### **1.2 Scope of the product**

The scope of this project is to make easy to the people who wish to donate food, money, books and clothes. This helps them to save their time and choose any orphanage which they need in many options.

### **1.3 Definitions, acronyms and abbreviations**

<b>Terms</b>	<b>Definition</b>
SRS	Software Requirement Specifications.
ER Diagram	Entity-Relationship Diagram
DFD	Data Flow Diagram
HTML	Hyper Text Markup Language
OS	Operating System

<b>Terms</b>	<b>Definition</b>
ASP.NET Frame Work	ASP.NET is an opensource web framework, created by Microsoft, for building modern web apps and services with .NET. ASP.NET is cross platform and runs on Windows, Linux, macOS, and Docker.
C# Language	C# is an object-oriented, component-oriented programming language. C# provides language constructs to directly support these concepts, making C# a natural language in which to create and use software components.
Visual Studio 2022	Visual Studio 2022 can be classed as an integrated development environment (IDE), meaning that developers can write and test code at the same time.



## **1.4 References**

Basic CRUD operations and fundamentals of SQL

## **1.5 Overview of remainder of the document**

The next chapter, the general description section of this document gives an overview of the functionality of the product.

The third chapter describes External Interface Requirements that are hardware and software requirements.

The fourth and fifth chapters, Specific Requirements section of this document is written primarily for the developers and describes in technical terms, the details of the interfaces, functional and non-functional requirements of the system.

## **2. General Description**

### **2.1 Product perspective**

Warm Hearts will be a generic and OpenSource software that captures the information and handles database. The purpose of this project is to help people who are starving due to hungry. This project “FOOD DONATION CENTER” will help those people by accepting surplus amount of food from donors who are willing to donate rather than wasting it and other donation.

### **2.2 Product functions**

The complete data is stored in the database of details of Donors, Orphanage and Food details.

This will capture the details of the sub services specified in Chapter 1 under Scope section for the specified service components.

### **2.3 User characteristics**

- ❖ Admin: The administrator can view the donations. Admin have all the access to modify or to delete the orphanage, blogs and feedback. They can see the accepted and awaited food, orphanage by their review in graphical representation.
- ❖ Donor: The donor can specify the quantity and type of food that they will donate. They will give details of their donation.



- ❖ **Receiver:** The Receiver receives donation and when it's come to food donation Receiver will get a notification when food is uploaded by donor.

## **2.4 General constraints**

Telegram is accessed to stay connected with donors and receivers instead of normal text messages.

## **2.5 Assumptions and dependencies**

We assume that the computers that will use the software will be part of the college LAN.

We assume that the food uploaded by Donors are accepted within the expire of food by the receiver.

The end users of this software are assumed to have basic level of computer knowledge i.e., point and click.

## **3. Specific Requirements**

### **3.1 Functional Requirements**

#### **Intended audience**

- 1.Donors-Personal details, Food Details
- 2.Receiver-Personal details, Orphanage details

Courses are provided to the orphans in orphanages by the volunteers.

Donation behavior is considered in data visualization.

### **3.2 Non-functional requirements**

Accuracy: The system should perform its process in accuracy.

Modifiability: The system should be easy to modify.

Security: The system should be secure and saving student's privacy.

Usability: The system should be easy to deal with and simple to understand.

Maintainability: The maintenance group should be able to fix any problem occur suddenly.

Speed and responsiveness: Execution of operation should be fast.



### **3.3Interface requirements**

#### **Hardware requirements**

Processor: Pentium IV, 2GHz and above.

RAM: 2GB (minimum) or 4GB (recommended).

Hard disk: 80GB or above.

Monitor: 15" CRT or LCD monitor.

Keyboard: Normal or Multimedia

Mouse: Any compatible mouse.

#### **Software requirements**

Operating System: Windows 10/ Linux Kernel (4.0 or later) operating system

Front End: BOOTSTRAP 5, ASP.NET Frame Work

Back End: C#, PYTHON

Scripting Language: JAVA SCRIPT

Data base: My SQL

Tools & IDE: Visual Studio 2022, Git Bash, XAMPP

### **4. Appendices**

### **5. Index**



## SYSTEM DESIGN

System Design is the process of defining the architecture, modules, interfaces and data for the system to satisfy the specified requirements. System design could be seen as the application of the system analysis, system architecture and the system engineering. System design is the first design state in which the basic approach to solving the problem is selected. During the system design, the overall structure and the style are decided. System design is concerned with how the system functionality is to be provided by the components of the system.

**The activities involved in this process are:**

- Partition requirements.
- Identify sub-systems.
- Assign requirements to sub-system.
- Specify sub-system functionality.
- Define sub-system interface.

The system architecture provides the context in which more detailed decisions are made in later design stages. It is the process of defining, developing and designing systems which satisfies the specific needs and requirements of a business or organization.

A good system design is to organize the program modules in such a way that are easy to develop and change. Structured design techniques help developers to deal with the size and complexity of programs. Analysts create instructions for the developers about how code should be written and how pieces of code should fit together to form a program.

System design could be seen as the application of systems theory to product development. It provides a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The importance of this phase may be understood by reason of the fact that it involves identifying data sources, the nature and type of data that is available.



## SYSTEM ARCHITECTURE MODELING

The system requirements and design activity system may be modelled as a set of components and relationship between these components. These are normally illustrated graphically in a system architecture model that gives the reader an overview of the system organization. The system architecture may be presented as a block diagram. Showing the major sub-systems and inter-connection between the sub-systems.

System architecture is a conceptual model that defines the structure, behavior and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structure of the system which compromise system components, the externally visible properties of those components, the relationships between them and provides a plan from which procured and the systems developed that will work together to implement the overall system.



## DATA FLOW DIAGRAMS

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.

It shows how data enters and leaves the system, what changes the information, and where data is stored.

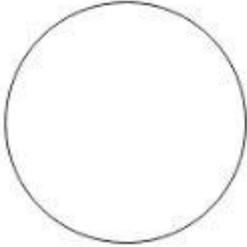
The objective of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for redesigning a system. The DFD is also called as a data flow graph or bubble chart.

**The following observations about DFDs are essential:**

1. All names should be unique. This makes it easier to refer to elements in the DFD.
2. Remember that DFD is not a flow chart. Arrows is a flow chart that represents the order of events; arrows in DFD represents flowing data. A DFD does not involve any order of events.
3. Suppress logical decisions. If we ever have the urge to draw a diamond-shaped box in a DFD, suppress that urge! A diamond-shaped box is used in flow charts to represents decision points with multiple exists paths of which the only one is taken. This implies an ordering of events, which makes no sense in a DFD.
4. Do not become bogged down with details. Defer error conditions and error handling until the end of the analysis.



## Common Flow Diagram Symbols and Meanings

Symbol	Name	Description
	Entity	It is represented by a rectangle and simply depicts a source or termination of the diagram by mapping real-world entities.
	Process	It is represented by a circle and depicts how the data is handled and processed in the system.
	Data Store	It is represented by two parallel lines and depicts a location where data is stored in the system.
	Data Flow	It is represented by directional lines and depicts the flow of data from one location to another



## **Entity**

Also known as eternal entities, they are often placed at the source or the terminal ends of the diagram. They provide a real-world representation of things in the diagram and are responsible for sending or receiving data. For instance, in the above data flow diagram, customers and warehouses are entities.

## **Process**

It is one of the most important data flow diagram symbols that depict the processing of data at any stage. A process would have incoming data, a management operation, and an output. It usually provides logical management or change in data. In the diagram, you can see that payment collection or order placement are some of the vital processes.

## **Data Store**

It simply depicts any location where data is stored in the system. It can be a document or an entire database that is used as a repository to store and handle data. In DFDs, Data Stores can have an input to collect data and an optional output to provide the stored data to any other entity or process. In our diagram, customers, invoices, and orders are some of its data storage units.

## **Data Flow**

Lastly, any data flow diagram would be incomplete without representing the flow of data from one entity/process to another. As you can see in the diagram, we use directional lines to depict how the data is moving from one place to another. At times, we also add captions on data flow arrows to make others understand these data flow diagram symbols and their meanings.

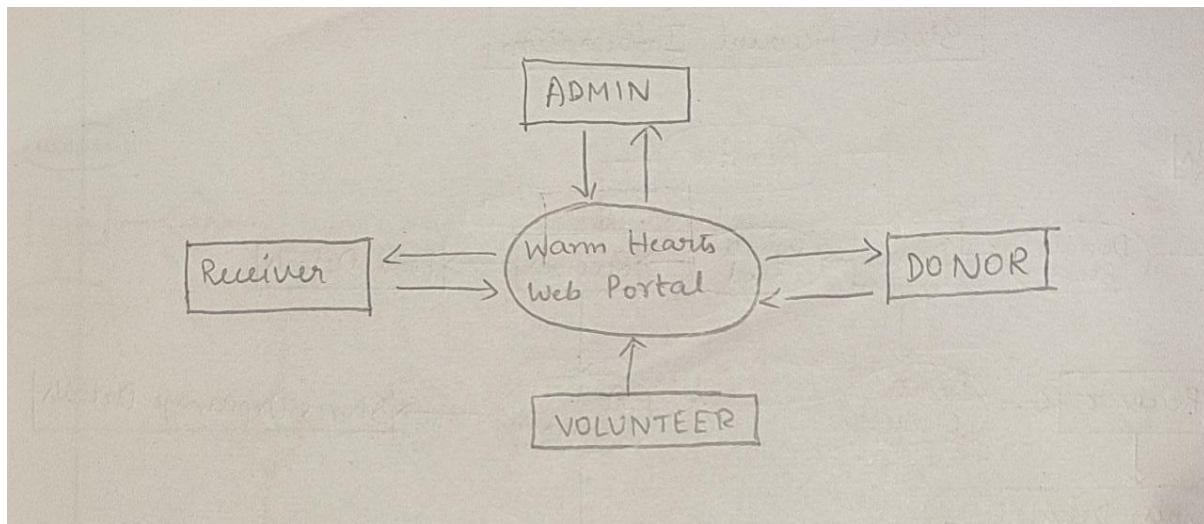


## Levels in Data Flow Diagrams (DFD)

The DFD may be used to perform a system or software at any level of abstraction. In fact, DFDs may be partitioned into levels that represent increasing information flow and functional detail. Levels in DFD are numbered 0, 1, 2 or beyond. Here, we will see primarily three levels in the data flow diagram, which are: 0-level DFD, 1-level DFD, and 2-level DFD.

### 0-level DFDM

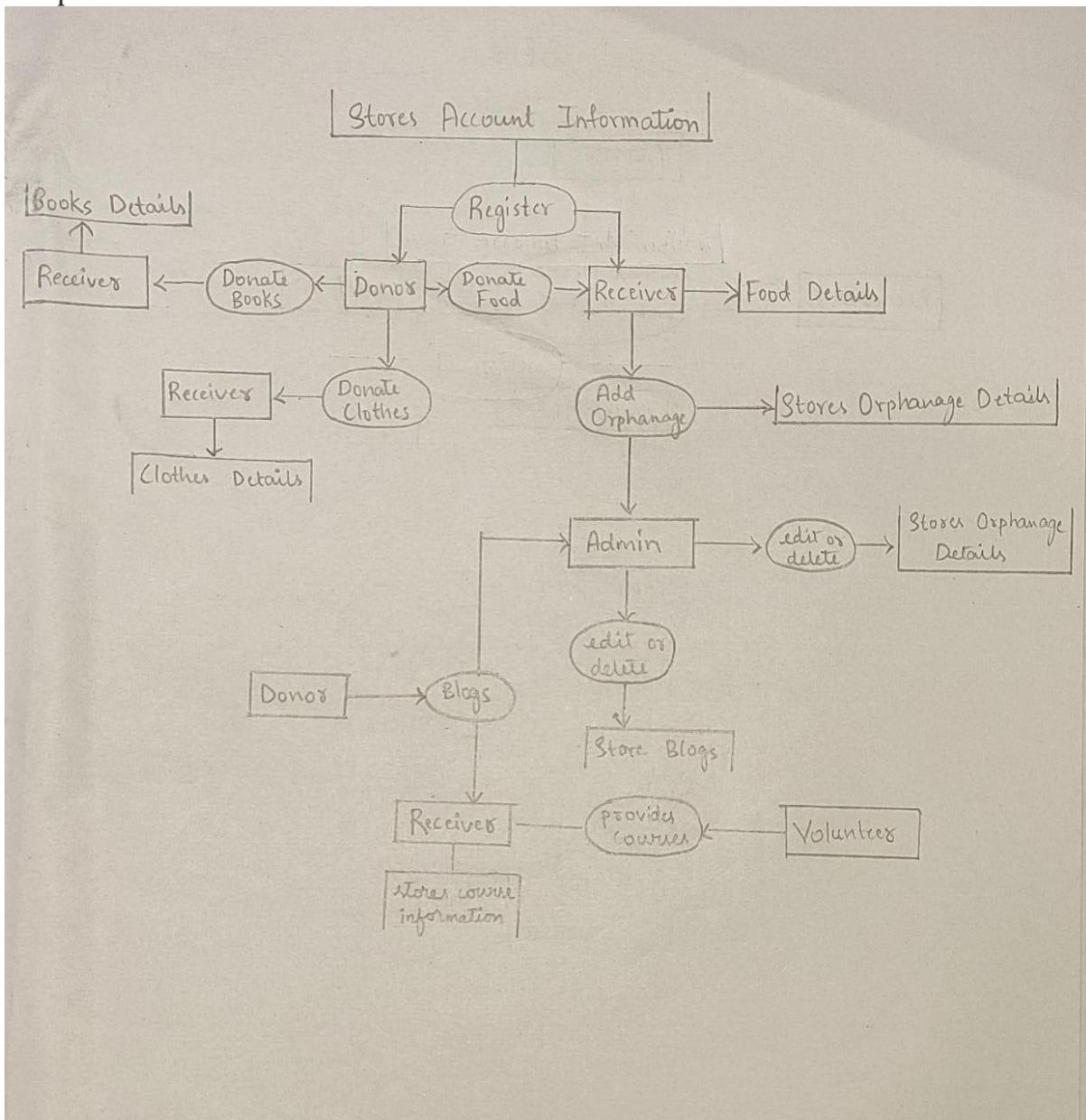
It is also known as fundamental system model, or context diagram represents the entire software requirement as a single bubble with input and output data denoted by incoming and outgoing arrows. Then the system is decomposed and described as a DFD with multiple bubbles. Parts of the system represented by each of these bubbles are then decomposed and documented as more and more detailed DFDs. This process may be repeated at as many levels as necessary until the program at hand is well understood. It is essential to preserve the number of inputs and outputs between levels, this concept is called levelling by DeMacro. The Level-0 DFD, also called context diagram of the result management-system.





## 1-level DFD

In 1-level DFD, a context diagram is decomposed into multiple bubbles/processes. In this level, we highlight the main objectives of the system and breakdown the high-level process of 0-level DFD into subprocesses.





## ENTITY RELATIONSHIP DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.

Uses of entity relationship diagrams

Database design

Database troubleshooting

Business information systems

Education

Research

### ->Entity

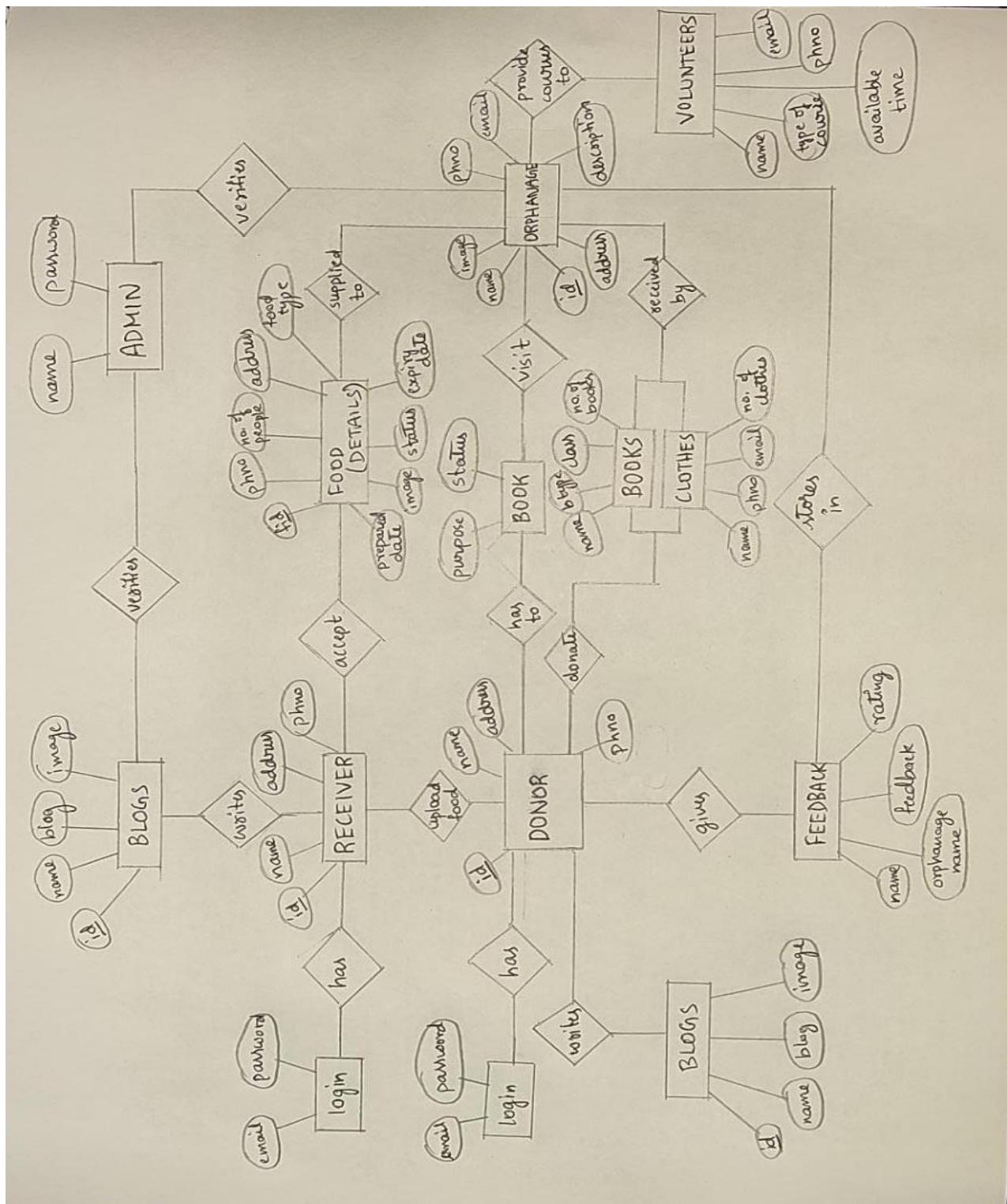
A definable thing—such as a person, object, concept or event—that can have data stored about it. Think of entities as nouns

### ->Relationship

How entities act upon each other or are associated with each other. Think of relationships as verbs. For example, the named student might register for a course.



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022





## SYSTEM IMPLEMENTATION

Implementation is a process of ensuring that the information system is operational. It involves –

- Constructing a new system from scratch
- Constructing a new system from the existing one.

Implementation allows the users to take over its operation for use and evaluation.

Implementation includes **user notification, user training, installation of hardware, installation of software onto production computers, and integration of the system into daily work processes**. This phase continues until the system is operating in production in accordance with the defined user requirements.

This is the process in which the developed system is handed over to the client. The old system is dispensed, new system is put into operation for use and all the end users are trained to manage and maintain the new system.

### Notification of uploaded food

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class notification : System.Web.UI.Page
{
    SqlConnection con;
    SqlConnection con1;
    protected void Page_Load(object sender, EventArgs e)
    {
        try
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
{  
  
    string strcon =  
ConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;  
    con = new SqlConnection(strcon);  
  
    String str = "select * from fdetails where status='Awaited'";  
  
    //String id = Request.QueryString.Get(0);  
  
  
    //string ln = Session["logname"].ToString();  
    // str = "select * from addmovies";  
    SqlCommand cm = new SqlCommand(str, con);  
    SqlDataReader reader;  
    con.Open();  
    // string cat = "", cat1 = "";  
    reader = cm.ExecuteReader();  
    Response.Write("<body style=background-  
image:url(assets/images/choosing-bg.jpg) > ");  
  
    Response.Write("<br><br><br><br><h1 style=color:white  
align=center>NOTIFICATIONS</h1><div id=menu> </div>");  
  
    Response.Write("<table style=color:white align=center  
border=5><tr><td>Id</td><td>Donor Name</td><td>Phone Number</td><td>Email  
Id</td><td>Address</td><td>Type Of Food</td><td>Number Of People Can Be Fed</td>  
<td>Image Of Food</td><td>Prepared Date</td>s<td>Expiry  
Date</td><td></td></tr><tr>");  
  
    String s = "", pid = "", d="",em="";  
    while (reader.Read())  
    {  
        for (int i = 0; i <= 9; i++)  
        {  
            s = reader.GetString(i).Trim();  
  
            if (i == 7)  
            {  
  
                //Response.Write(s);  
  
                //Response.Write("<td>" + s + " </td>");  
                Response.Write("<td></img></td>");  
                // Response.Write("<td><a href=addtocart.aspx?id=" + s  
+ " >Add to Cart</a></td>");  
            }  
            else if (i == 0)  
            {  
                pid = s;  
                Response.Write("<td>" + s + " </td>");  
  
            }  
            else if (i==3)  
        }  
    }  
}
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
{  
    em = s;  
    Response.Write("<td>" + s + " </td>");  
}  
else if (i==9)  
{  
    d = s;  
    Response.Write("<td>" + s + " </td>");  
}  
else  
{  
    Response.Write("<td>" + s + " </td>");  
}  
  
}  
DateTime date1 = DateTime.Parse(d);  
DateTime date2 = DateTime.Now;  
if (date1.Date >= date2.Date)  
{  
    Response.Write("<td><a href=notification1.aspx?fid=" + pid +  
"&expirydate=" + d + "&email=" + em + "><button  
style=background:green;color:white; border:green><b>ACCEPT</b></button></a></td>");  
;  
    // Response.Write("<td><a href=deletemovies1.aspx?id=" + pid  
+ " ><img src=delete.jpg height=50 width=50></a></td>");  
    // Response.Write("<td><a href>Editbook.aspx?id=" + class=btn  
btn-info btn-lg> <span class=glyphicon glyphicon-pencil></span> Edit  
</a></td>");  
    // Response.Write("<td><a href=deletebook.aspx?id=" + s + "  
>Delete</a></td>");  
    // Response.Write("<td><a  
href=Viewcommentsmainpage1.aspx?u=" + s + ">Download<a></td>");  
}  
else  
{  
    Response.Write("<td style=color:red;><b>EXPIRED</b></td>");  
}  
Response.Write("</tr>");  
}  
Response.Write("</table>");  
Response.Write("</body>");  
// Response.Write("<a href=homepage.aspx> back </a>");  
con.Close();  
  
}  
catch (Exception ex)  
{  
    Response.Write(ex);  
}  
}  
}
```



## Accepting the food

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class notification1 : System.Web.UI.Page
{
    SqlConnection con;
    SqlConnection con1;

    protected void Page_Load(object sender, EventArgs e)
    {
        String id = Request.QueryString.Get(0);
        String d = Request.QueryString.Get(1);
        String em = Request.QueryString.Get(2);
        TextBox1.Text = id;
        TextBox6.Text = d;
        TextBox7.Text = em;
        TextBox4.Text = (string)Session["emailid"];
        TextBox2.Text = (string)Session["rname"];
        TextBox3.Text = (string)Session["phoneno"];
    }

    protected void Button1_Click1(object sender, EventArgs e)
    {

        try
        {
            DateTime date1 = DateTime.Parse(TextBox6.Text);
            DateTime date2 = DateTime.Now;

            //Response.Write("Your Date:" + date1.ToShortDateString());
            // Response.Write("Now:" + date2.ToShortDateString());

            string strcon =
ConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
            con = new SqlConnection(strcon);
            if (date1.Date >= date2.Date)
            {

                String s = "select * from fdetails";
```



```
s = "update fdetails set
name=@n,phno=@ph,email=@em,address=@ad,status='Accepted',demail=@dem where fid="
+ "" + TextBox1.Text + "";
SqlCommand cmd11 = new SqlCommand(s, con);
cmd11.Parameters.Add("@n", TextBox2.Text);
cmd11.Parameters.Add("@ph", TextBox3.Text);
cmd11.Parameters.Add("@ad", TextBox5.Text);
cmd11.Parameters.Add("@em", TextBox4.Text);
cmd11.Parameters.Add("@dem", TextBox7.Text);
con.Open();
cmd11.ExecuteNonQuery();
con.Close();
Response.Write("<script>alert('Thank You for Accepting...Send A
Notification To
Donor');window.location='http://127.0.0.1:8000/admin/rsendmessage';</script>");
}
else
{
    Response.Write("<script>alert('Please Check The Expiry
Date');window.location='notification.aspx';</script>");
}

}
catch (Exception ex)
{
    Response.Write(ex.ToString());
}
}
}
```

## Donor registration

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class donorsignin : System.Web.UI.Page
{
    SqlConnection con;
    SqlConnection con1;
    protected void Page_Load(object sender, EventArgs e)
    {
        string strcon =
ConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
        con = new SqlConnection(strcon);
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
}

protected void Button1_Click(object sender, EventArgs e)
{
    try
    {

        string s = "select * from donor where email= " + "" + TextBox4.Text
+ "";;
        con.Open();

        SqlCommand cmd1 = new SqlCommand(s, con);
        SqlDataReader reader;
        reader = cmd1.ExecuteReader();
        int ctr = 0;
        while (reader.Read())
        {
            ctr++;

        }
        reader.Close();
        con.Close();
        if (ctr == 0)
        {

            s = "insert into donor
values(@name,@address,@phno,@email,@password,'Donor')";
            cmd1 = new SqlCommand(s, con);

            cmd1.Parameters.Add("@name", TextBox1.Text);
            cmd1.Parameters.Add("@address", TextBox2.Text);
            cmd1.Parameters.Add("@phno", TextBox3.Text);
            cmd1.Parameters.Add("@email", TextBox4.Text);
            cmd1.Parameters.Add("@password", TextBox5.Text);

            con.Open();
            cmd1.ExecuteNonQuery();
            con.Close();
            // Label1.Text = "Registered successfully";
            Response.Write("<script>alert('Registered
Successfully');window.location='donor.aspx';</script>");

            TextBox1.Text = "";
            TextBox2.Text = "";
            TextBox3.Text = "";
            TextBox4.Text = "";
            TextBox5.Text = "";
        }
    }
}
```



```
        }
        else
        {
            Response.Write("<script>alert('Sorry, you have Registered  
Already');window.location='index.html';</script>");
        }
        //Session["name"] = TextBox1.Text.Trim();
        //Session["address"] = TextBox6.Text.Trim();
        //Session["mobile"] = TextBox2.Text.Trim();

    }

    catch (Exception ex)
    {

        Label6.Text = ex.ToString();
    }
}
}
```

## Receiver Registration

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class receiversignin : System.Web.UI.Page
{
    SqlConnection con;
    SqlConnection con1;
    protected void Page_Load(object sender, EventArgs e)
    {
        string strcon =
ConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
        con = new SqlConnection(strcon);
    }

    protected void Button1_Click(object sender, EventArgs e)
    {
        try
        {

            string s = "select * from receiver where emailid= " + "" +
TextBox5.Text + "";
            con.Open();

            SqlCommand cmd1 = new SqlCommand(s, con);
            SqlDataReader reader;
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
reader = cmd1.ExecuteReader();
int ctr = 0;
while (reader.Read())
{
    ctr++;

}
reader.Close();
con.Close();
if (ctr == 0)
{

    s = "insert into receiver
values(@name,@address,@phno,@emailid,@password,'Receiver')";
cmd1 = new SqlCommand(s, con);

cmd1.Parameters.Add("@name", TextBox1.Text);
cmd1.Parameters.Add("@address", TextBox2.Text);
cmd1.Parameters.Add("@phno", TextBox3.Text);
cmd1.Parameters.Add("@emailid", TextBox5.Text);
cmd1.Parameters.Add("@password", TextBox6.Text);

con.Open();
cmd1.ExecuteNonQuery();
con.Close();
// Label1.Text = "Registered successfully";
Response.Write("<script>alert('Registered
Successfully');window.location='receiver.aspx';</script>");

TextBox1.Text = "";
TextBox2.Text = "";
TextBox3.Text = "";
TextBox5.Text = "";
TextBox6.Text = "";

}

else
{
    Response.Write("<script>alert('Sorry, you have Registered
Already');window.location='index.html';</script>");
}
//Session["name"] = TextBox1.Text.Trim();
//Session["address"] = TextBox6.Text.Trim();
//Session["mobile"] = TextBox2.Text.Trim();

}

catch (Exception ex)
{
    Label1.Text = ex.ToString();
}
```



```
}
```

## Forgot Password

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;
using System.Net.Mail;

public partial class Forgotpwd : System.Web.UI.Page
{
    SqlConnection conn;
    SqlConnection con1;

    protected void Button1_Click(object sender, EventArgs e)
    {
        try
        {
            if (DropDownList1.Text == "Donor")
            {
                String str = "select * from donor where email=" + "'" +
                TextBox1.Text + "'";
                SqlCommand cm = new SqlCommand(str, conn);
                SqlDataReader reader;
                conn.Open();
                reader = cm.ExecuteReader();

                int f1 = 0;
                while (reader.Read())
                {
                    f1++;

                    }

                conn.Close();
                if (f1 == 0)
                {
                    Response.Write("<script>alert('Sorry, Invalid User
Id');window.location='Forgotpwd.aspx';</script>");
                }
                else
                {

                    string allowedChars = "";

```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
    allowedChars =
"a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,";
    allowedChars += "A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,";
    allowedChars += "1,2,3,4,5,6,7,8,9,0,!,@,#,$,%,&,";

    char[] sep = { ',' };

    string[] arr = allowedChars.Split(sep);

    string passwordString = "";

    string temp = "";

    Random rand = new Random();
    string txt = TextBox1.Text;

    for (int i = 0; i < Convert.ToInt32(txt.Length); i++)
    {
        temp = arr[rand.Next(0, arr.Length)];
        passwordString += temp;
    }
    //    Label1.Text = passwordString;
    //
    string str1 = "update donor set password = @p where email= "
+ "" + TextBox1.Text.Trim() + "";
//Response.Write(str1);
conn.Open();
SqlCommand cmd = new SqlCommand(str1, conn);
cmd.Parameters.Add("@p", passwordString);
cmd.ExecuteNonQuery();
conn.Close();
Label3.Text = "Your New Password is:" + passwordString;

}

}

else if(DropDownList1.Text == "Receiver")
{
    String str = "select * from receiver where emailid=" + "" +
TextBox1.Text + "";
    SqlCommand cm = new SqlCommand(str, conn);
    SqlDataReader reader;
    conn.Open();
    reader = cm.ExecuteReader();

    int f1 = 0;
    while (reader.Read())
    {

        f1++;

    }

}
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
conn.Close();
if (f1 == 0)
{
    Response.Write("<script>alert('Sorry, Invalid User
Id');window.location='Forgotpwd.aspx';</script>");
}
else
{

    string allowedChars = "";
    allowedChars =
"a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u,v,w,x,y,z,";
    allowedChars += "A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,";
    allowedChars += "1,2,3,4,5,6,7,8,9,0,!,@,#,$,%,&?";

    char[] sep = { ',' };

    string[] arr = allowedChars.Split(sep);

    string passwordString = "";

    string temp = "";

    Random rand = new Random();
    string txt = TextBox1.Text;

    for (int i = 0; i < Convert.ToInt32(txt.Length); i++)
    {
        temp = arr[rand.Next(0, arr.Length)];
        passwordString += temp;
    }
    //    Label1.Text = passwordString;
    //
    string str1 = "update receiver set password = @p where
emailid= " + "" + TextBox1.Text.Trim() + "";
    //Response.Write(str1);
    conn.Open();
    SqlCommand cmd = new SqlCommand(str1, conn);
    cmd.Parameters.Add("@p", passwordString);
    cmd.ExecuteNonQuery();
    conn.Close();
    Label3.Text = "Your New Password is:      " + passwordString;

    }

}

catch (Exception ex)
{
    Response.Write(ex.ToString());
}

}
```



```
protected void Page_Load(object sender, EventArgs e)
{
    string strcon =
ConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
    //create new sqlconnection and connection to database by using connection
string from web.config file
    conn = new SqlConnection(strcon);
}
}
```

## Edit Password

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class editpwd : System.Web.UI.Page
{
    SqlConnection con;
    SqlConnection con1;
    String eid = "", s = "";
    protected void Page_Load(object sender, EventArgs e)
    {

    }
    protected void Button1_Click1(object sender, EventArgs e)
    {
        try
        {
            string strcon =
ConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
            con = new SqlConnection(strcon);
            eid = Session["email"].ToString();

            s = "select * from donor where email=" + "" + eid + "" + " and
password=" + "" + TextBox1.Text + "";
            //Response.Write(s);
            con.Open();

            SqlCommand cmd1 = new SqlCommand(s, con);
            SqlDataReader reader;
            reader = cmd1.ExecuteReader();
            int ctr = 0;
            //   string uname = "";
            while (reader.Read())
            {
                //   uname = reader.GetString(0);
                ctr++;
            }
        }
    }
}
```



```
reader.Close();
con.Close();
if (ctr == 1)
{
    s = "update donor set password=@p where email=@eid";
    SqlCommand cmd11 = new SqlCommand(s, con);
    cmd11.Parameters.AddWithValue("@p", TextBox3.Text);
    cmd11.Parameters.AddWithValue("@eid", eid);

    con.Open();
    cmd11.ExecuteNonQuery();
    con.Close();
    Response.Write("<script>alert('Password Updated
Successfully...');window.location='signin.aspx';</script>");
}
else
{
    Label1.Text = "Something has gone wrong plzzz check...";
}
catch (Exception ex)
{
    Label2.Text = ex.ToString();
}
}
}
```

## Sign-in

```
using System;
using System.Collections;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.HtmlControls;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Xml.Linq;
using System.Data.SqlClient;

public partial class signin : System.Web.UI.Page
{
    SqlConnection conn = new SqlConnection(@"Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\We-
care\App_Data\Database.mdf;Integrated Security=True");
    SqlDataReader r;
    int f = 0;
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
protected void Page_Load(object sender, EventArgs e)
{
}

protected void Button1_Click(object sender,EventArgs e)
{
    try
    {
        if (DropDownList1.Text == "Admin")
        {
            string s = "select * from admin";
            SqlCommand cmd = new SqlCommand(s, conn);
            conn.Open();
            int f = 0;
            r = cmd.ExecuteReader();
            String name, password, un1, pwd1;
            name = TextBox1.Text.Trim();
            password = TextBox2.Text.Trim();
            while (r.Read())
            {
                un1 = r.GetString(0).Trim();
                pwd1 = r.GetString(1).Trim();

                if (name == un1 && password == pwd1)
                {
                    // Label1.Text = "success";
                    Session["name"] = TextBox1.Text.Trim();
                    //Session["password"] = TextBox2.Text.Trim();
                    //Session["address"] = address;
                    f = 1;
                    Response.Redirect("admin.aspx");
                    break;
                }
            }
            conn.Close();
            r.Close();
            if (f == 0)
            {
                Label4.Text = "Invalid username or password!";
            }
        }
        else if (DropDownList1.Text == "Donor")
        {
            string s = "select * from donor";
            SqlCommand cmd = new SqlCommand(s, conn);
            conn.Open();
            r = cmd.ExecuteReader();
            string email, password, un1, pwd1,n="" ,phno="";
            email = TextBox1.Text.Trim();
            password = TextBox2.Text.Trim();
            while (r.Read())
            {
                un1 = r.GetString(4).Trim();
            }
        }
    }
}
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
pwd1 = r.GetString(5).Trim();
n = r.GetString(1).Trim();
phno = r.GetString(3).Trim();
if (email == un1 && password == pwd1)
{
    f = 1;
    Session["email"] = TextBox1.Text.Trim();
    Session["dname"] = n;
    Session["dphno"] = phno;
    Response.Redirect("donor.aspx");
    break;
}
conn.Close();
r.Close();
if (f == 0)
{
    Label4.Text = "Invalid username or password!";
}
else if (DropDownList1.Text == "Receiver")
{
    string ss = "select * from receiver";
    SqlCommand cmd1 = new SqlCommand(ss, conn);
    conn.Open();
    r = cmd1.ExecuteReader();
    string unr, pwdr, unr1, pwdr1, n = "", phno = "";
    unr = TextBox1.Text.Trim();
    pwdr = TextBox2.Text.Trim();
    while (r.Read())
    {
        unr1 = r.GetString(4).Trim();
        pwdr1 = r.GetString(5).Trim();
        n=r.GetString(1).Trim();
        phno=r.GetString(3).Trim();
        if (unr == unr1 && pwdr == pwdr1)
        {
            f = 1;
            Session["emailid"] = TextBox1.Text.Trim();
            Session["rname"] = n;
            Session["phoneno"] = phno;
            //Session["name"] = TextBox2.Text.Trim();
            break;
        }
    }
    conn.Close();
    r.Close();
    if (f == 0)
    {
        Label4.Text = "Invalid username or password!";
    }
    else
    {
        Response.Redirect("receiver.aspx");
    }
}
```



```
        }

    }
    catch (Exception ex)
    {
        Response.Write(ex.ToString());
    }
}

}
```

## Data Visualization

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="admin.aspx.cs"
Inherits="admin" %>

<!DOCTYPE html>
<html lang="en">

    <head>

        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
        <meta name="description" content="">
        <meta name="author" content="">
        <link
href="https://fonts.googleapis.com/css?family=Montserrat:100,200,300,400,500,600,
700,800,900" rel="stylesheet">

        <title>WARM HEARTS</title>

        <!-- Bootstrap core CSS -->
        <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

        <!-- Additional CSS Files -->
        <link rel="stylesheet" href="assets/css/fontawesome.css">
        <link rel="stylesheet" href="assets/css/templatemo-grad-school.css">
        <link rel="stylesheet" href="assets/css/owl.css">
        <link rel="stylesheet" href="assets/css/lightbox.css">
    <!--

TemplateMo 557 Grad School

https://templatemo.com/tm-557-grad-school

-->
    </head>

    <body style="background-image:url(assets/images/choosing-bg.jpg)">

        <form id="form1" runat="server">

            <!--header-->
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
<header class="main-header clearfix" >
    <div class="logo">
        <a href="#"><em>WARM</em>HEARTS</a>
    </div>
    <nav id="nav" class="main-nav" role="navigation">
        <ul class="main-menu">
            <li><a href="admin.aspx">Home</a></li>
            <li><a href="adonordetails.aspx">Donor's details</a></li>
            <li><a href="arecdetails.aspx">Receivers details</a></li>
            <li><a href="aeditpwd.aspx">Edit Password</a></li>
            <li><a href="ablog.aspx">Edit Blog</a></li>
            <li><a href="aorphanage.aspx">Edit Orphanage</a></li>
            <!-- <li><a href="#section5">Video</a></li> -->

            <li><a href="signin.aspx">Logout</a></li>
        </ul>
    </nav>
</header>

<!-- ***** Main Banner Area Start ***** -->
<!-- ***** Main Banner Area End ***** -->



<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="assets/js/isotope.min.js"></script>
<script src="assets/js/owl-carousel.js"></script>
<script src="assets/js/lightbox.js"></script>
<script src="assets/js/tabs.js"></script>
<script src="assets/js/video.js"></script>
<script src="assets/js/slick-slider.js"></script>
<script src="assets/js/custom.js"></script>
<script>
    //according to loftblog tut
    $('.nav li:first').addClass('active');

    var showSection = function showSection(section, isAnimate) {
        var
            direction = section.replace(/#/,''),
            reqSection = $('.section').filter('[data-section="' + direction
+ '"']),
            reqSectionPos = reqSection.offset().top - 0;

        if (isAnimate) {
            $('body, html').animate({
                scrollTop: reqSectionPos
            },
            800);
        } else {
            $('body, html').scrollTop(reqSectionPos);
        }
    }

```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
};

$(window).scroll(function () {
    checkSection();
});

</script><br /><br /><br /><br /><br />
<div align="center">
<h3 style="color:white">NUMBER OF USERS REGISTERED</h3>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionString %>">
SelectCommand="SELECT role, COUNT(*) AS count FROM donor GROUP BY role UNION ALL
SELECT role, COUNT(*) AS count FROM receiver GROUP BY role"></asp:SqlDataSource>
<asp:Chart ID="Chart1" runat="server" DataSourceID="SqlDataSource1"
LabelForeColor="White" BackColor="Transparent" Palette="None" CssClass="bg-
transparent" Height="438px" Width="698px" EnableViewState="True"
PaletteCustomColors="Maroon; 128, 64, 64" style="margin-top: 81px"
BorderlineWidth="2">
<Series>
    <asp:Series ChartType="Pie" LabelForeColor="White" Name="Count
of Users" XValueMember="role" YValueMembers="count" Legend="Donor">
        </asp:Series>
    </Series>
    <ChartAreas>
        <asp:ChartArea Name="ChartArea1" BorderColor="White">
            </asp:ChartArea>
    </ChartAreas>
    <Legends>
        <asp:Legend LegendStyle="Row" Name="Donor">
            </asp:Legend>
        <asp:Legend Name="Receiver">
            </asp:Legend>
        </Legends>
    <Titles>
        <asp:Title Name="Title1">
            </asp:Title>
    </Titles>
    <Annotations>
        <asp:TextAnnotation Name="TextAnnotation1" Text="Donor">
            </asp:TextAnnotation>
        <asp:TextAnnotation Name="TextAnnotation2"
Text="TextAnnotation2">
            </asp:TextAnnotation>
        </Annotations>
    </asp:Chart><br />
    <h3 style="color:white">FOOD STATUS</h3>
    <asp:Chart ID="Chart2" runat="server"
DataSourceID="SqlDataSource2" Height="250px" Width="638px" CssClass="auto-
style1">
        <Series>
            <asp:Series Name="Series1" XValueMember="status"
YValueMembers="count">
                </asp:Series>
        </Series>
        <ChartAreas>
            <asp:ChartArea Name="ChartArea1">
```



```
</asp:ChartArea>
</ChartAreas>
</asp:Chart>
<br /><br />
<asp:SqlDataSource ID="SqlDataSource2" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionString %>" 
SelectCommand="SELECT status, count(*) as count from fdetails group by
status;"></asp:SqlDataSource>
<h3 style="color:white">RATINGS OF ORPHANAGES</h3>
<asp:Chart ID="Chart3" runat="server"
DataSourceID="SqlDataSource3" Width="492px" Height="341px" CssClass="auto-style1"
>
<Series>
<asp:Series Name="Series1" XValueMember="oname"
YValueMembers="rating">
</asp:Series>
</Series>
<ChartAreas>
<asp:ChartArea Name="ChartArea1">
</asp:ChartArea>
</ChartAreas>
</asp:Chart>
<asp:SqlDataSource ID="SqlDataSource3" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionString %>" 
SelectCommand="SELECT oname,rating from blog group by
oname,rating;"></asp:SqlDataSource>
</div>

</form>

</body>
</html>
```

## Edit or Delete Blogs

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="ablog.aspx.cs"
Inherits="ablog" %>
<!DOCTYPE html>
<html lang="en">
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link
        href="https://fonts.googleapis.com/css?family=Montserrat:100,200,300,400,500,600,
        700,800,900" rel="stylesheet">

    <title>WARM HEARTS</title>
    <!-- Bootstrap core CSS -->
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- Additional CSS Files -->
    <link rel="stylesheet" href="assets/css/fontawesome.css">
    <link rel="stylesheet" href="assets/css/templatemo-grad-school.css">
    <link rel="stylesheet" href="assets/css/owl.css">
    <link rel="stylesheet" href="assets/css/lightbox.css">
<!--

TemplateMo 557 Grad School

https://templatemo.com/tm-557-grad-school

-->
    </head>

    <body style="background-image:url(assets/images/choosing-bg.jpg)">

        <form id="form1" runat="server">

            <!--header-->
            <header class="main-header clearfix" >
                <div class="logo">
                    <a href="#"><em>WARM</em>HEARTS</a>
                </div>
                <nav id="nav" class="main-nav" role="navigation">
                    <ul class="main-menu">
                        <li><a href="admin.aspx">Home</a></li>
                        <li><a href="aeditpwd.aspx">Edit password</a></li>

                        <li><a href="adonordetails.aspx">Donor's details</a></li>
                        <!-- <li><a href="#section5">Video</a></li> -->
                        <li><a href="arecdetails.aspx">Receivers details</a></li>
                        <li><a href="signin.aspx">Logout</a></li>
                    </ul>
                </nav>
            </header>

            <!-- ***** Main Banner Area Start ***** -->
            <!-- ***** Main Banner Area End ***** -->
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
<!-- Scripts -->
<!-- Bootstrap core JavaScript -->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="assets/js/isotope.min.js"></script>
<script src="assets/js/owl-carousel.js"></script>
<script src="assets/js/lightbox.js"></script>
<script src="assets/js/tabs.js"></script>
<script src="assets/js/video.js"></script>
<script src="assets/js/slick-slider.js"></script>
<script src="assets/js/custom.js"></script>
<script>
    //according to loftblog tut
    $('.nav li:first').addClass('active');

    var showSection = function showSection(section, isAnimate) {
        var direction = section.replace(/#/,''),
            reqSection = $('.section').filter('[data-section="' + direction
+ '"]');
            reqSectionPos = reqSection.offset().top - 0;

        if (isAnimate) {
            $('body, html').animate({
                scrollTop: reqSectionPos
            },
            800);
        } else {
            $('body, html').scrollTop(reqSectionPos);
        }
    };

    $(window).scroll(function () {
        checkSection();
    });
</script>

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
DataSourceID="SqlDataSource1"   CssClass="avatar"   Height="100px" Width="1250px"
AutoGenerateDeleteButton="True"  AutoGenerateEditButton="true" DataKeyNames="id"
EnableModelValidation="true" style="margin-left: 158px; margin-top: 329px" >
    <Columns>
        <asp:BoundField DataField="id" HeaderText="ID"
SortExpression="id" ItemStyle-ForeColor="White" HeaderStyle-ForeColor="White"/>
        <asp:BoundField DataField="name" HeaderText="NAME"
SortExpression="name" ControlStyle-ForeColor="White" ItemStyle-ForeColor="White"
HeaderStyle-ForeColor="White"/>
        <asp:BoundField DataField="blog" HeaderText="BLOG"
SortExpression="blog" ControlStyle-ForeColor="White" ItemStyle-ForeColor="White"
HeaderStyle-ForeColor="White" />
```



```
</Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionString %>" 
SelectCommand="SELECT * FROM [blogs]" DeleteCommand="DELETE FROM [blogs] where
[id]=@id" UpdateCommand="UPDATE [blogs] set [blog]=@blog,[image]=@image where
[id]=@id"></asp:SqlDataSource>

</form>

</body>

</html>
```

## Edit or Delete Orphanage

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="aorphanage.aspx.cs"
Inherits="aorphanage" %>

<!DOCTYPE html>
<html lang="en">

<head>

    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
    <meta name="description" content="">
    <meta name="author" content="">
    <link
    href="https://fonts.googleapis.com/css?family=Montserrat:100,200,300,400,500,600,
700,800,900" rel="stylesheet">

    <title>WARM HEARTS</title>
    <!-- Bootstrap core CSS -->
    <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- Additional CSS Files -->
    <link rel="stylesheet" href="assets/css/fontawesome.css">
    <link rel="stylesheet" href="assets/css/templatemo-grad-school.css">
    <link rel="stylesheet" href="assets/css/owl.css">
    <link rel="stylesheet" href="assets/css/lightbox.css">
<!--

TemplateMo 557 Grad School

https://templatemo.com/tm-557-grad-school

-->
</head>
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
<body style="background-image:url(assets/images/choosing-bg.jpg)">

<form id="form1" runat="server">

<!--header-->
<header class="main-header clearfix" >
    <div class="logo">
        <a href="#"><em>WARM</em>HEARTS</a>
    </div>
    <nav id="nav" class="main-nav" role="navigation">
        <ul class="main-menu">
            <li><a href="admin.aspx">Home</a></li>
            <li><a href="aeditpwd.aspx">Edit password</a></li>

            <li><a href="adonordetails.aspx">Donor's details</a></li>
            <!-- <li><a href="#section5">Video</a></li> -->
            <li><a href="arecdetails.aspx">Receivers details</a></li>
            <li><a href="signin.aspx">Logout</a></li>
        </ul>
    </nav>
</header>

<!-- ***** Main Banner Area Start ***** -->
<!-- ***** Main Banner Area End ***** -->

<!-- Scripts -->
<!-- Bootstrap core JavaScript -->
<script src="vendor/jquery/jquery.min.js"></script>
<script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

<script src="assets/js/isotope.min.js"></script>
<script src="assets/js/owl-carousel.js"></script>
<script src="assets/js/lightbox.js"></script>
<script src="assets/js/tabs.js"></script>
<script src="assets/js/video.js"></script>
<script src="assets/js/slick-slider.js"></script>
<script src="assets/js/custom.js"></script>
<script>
    //according to loftblog tut
    $('.nav li:first').addClass('active');

    var showSection = function showSection(section, isAnimate) {
        var
            direction = section.replace(/#/,''),
            reqSection = $('.section').filter('[data-section="' + direction
+ '"']),
            reqSectionPos = reqSection.offset().top - 0;

        if (isAnimate) {
            $('body, html').animate({
                scrollTop: reqSectionPos

```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
        },
        800);
    } else {
        $('body, html').scrollTop(reqSectionPos);
    }
};

$(window).scroll(function () {
    checkSection();
});
</script>

<asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
DataSourceID="SqlDataSource1" Height="100px" Width="1250px"
AutoGenerateDeleteButton="True" AutoGenerateEditButton="True" style="margin-left: 158px; margin-top: 329px" DataKeyNames="id" EnableModelValidation="true" >
    <Columns>
        <asp:BoundField DataField="id" HeaderText="ID"
SortExpression="id" HeaderStyle-ForeColor="White" ItemStyle-ForeColor="White" />
        <asp:BoundField DataField="name" HeaderText="NAME"
SortExpression="name" HeaderStyle-ForeColor="White" ItemStyle-ForeColor="White" />
        <asp:BoundField DataField="phno" HeaderText="PHONE NUMBER"
SortExpression="phno" HeaderStyle-ForeColor="White" ItemStyle-ForeColor="White" />
        <asp:BoundField DataField="email" HeaderText="EMAIL ID"
SortExpression="email" HeaderStyle-ForeColor="White" ItemStyle-ForeColor="White" />
        <asp:BoundField DataField="address" HeaderText="ADDRESSS"
SortExpression="address" HeaderStyle-ForeColor="White" ItemStyle-ForeColor="White" />
        <asp:BoundField DataField="descr" HeaderText="REQUIREMENTS"
SortExpression="descr" HeaderStyle-ForeColor="White" ItemStyle-ForeColor="White" />
        <asp:ImageField DataImageUrlField="oimage" ControlStyle-Height="100" HeaderText="IMAGE" SortExpression="oimage" HeaderStyle-ForeColor="White" ItemStyle-ForeColor="White" />
    </Columns>
</asp:GridView>
<asp:SqlDataSource ID="SqlDataSource1" runat="server"
ConnectionString="<%$ ConnectionStrings:ConnectionString %>" SelectCommand="SELECT * FROM [orphanage]" DeleteCommand="DELETE FROM [orphanage] where [id]=@id"></asp:SqlDataSource>

</form>
</body>
</html>
```



## Fetching Details of Orphanage from Receiver

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class orphanage1 : System.Web.UI.Page
{
    SqlConnection con;
    SqlConnection conn;

    protected void Page_Load(object sender, EventArgs e)
    {
        string n=Request.QueryString.Get(0);
        try
        {
            string strcon =
ConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
            con = new SqlConnection(strcon);
            String str = "select * from orphanage where name=" + "" + n + "";
            //String id = Request.QueryString.Get(0);

            //string ln = Session["logname"].ToString();
            // str = "select * from addmovies";
            SqlCommand cm = new SqlCommand(str, con);
            SqlDataReader reader;
            con.Open();
            // string cat = "", cat1 = "";
            reader = cm.ExecuteReader();

            Response.Write("<body style=background-
image:url(assets/images/choosing-bg.jpg) > ");
            Response.Write("<br><br/><br/><h1 style=color:white
align=center>ORPHANAGES</h1><br/><br/>");
            Response.Write("<div class=row>");


```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
Response.Write("<div class=col-md-12 my-2>");  
Response.Write("<div class=card style=background-  
image:url(assets/images/grey.jpg);>");  
  
String s = "", pid = "", em="";  
while (reader.Read())  
{  
    for (int i = 0; i <= 5; i++)  
    {  
        s = reader.GetString(i).Trim();  
        if (i == 1)  
        {  
  
            Response.Write("<center></img></center>");  
  
            //Response.Write(s);  
            //Response.Write("<td>" + s + " </td>");  
  
            //    Response.Write("<td><a href=addtocart.aspx?id=" + s  
+ " >Add to Cart</a></td>");  
        }  
        else if (i == 0)  
        {  
            pid = s;  
            Response.Write("<center><h1>" + s + "</h2></center>");  
        }  
        else if(i == 3)  
        {  
            em = s;  
            Response.Write("<center><h4>" + s + " </h4></center>");  
        }  
        else  
        {  
            Response.Write("<center><h4>" + s + " </h4></center>");  
        }  
    }  
  
    Response.Write("<center><p><a href=uploaddonation.aspx><button  
style=background:green;color:white>Donate</button>&nbsp;&nbsp;&nbsp;&nbsp;&  
nbsp;&nbsp;&nbsp;</a><a href=course.aspx><button  
style=background:green;color:white>Course</button>&nbsp;&nbsp;&nbsp;&nbsp;&  
nbsp;&nbsp;&nbsp;</a><a href=blogs.aspx?name=" + pid + " ><button  
style=background:green;color:white>Feedback</button>&nbsp;&nbsp;&nbsp;&nbsp;&  
nbsp;&nbsp;&nbsp;</a><a href=booking.aspx?=" + em + "&name=" + pid + "><button  
style=background:green;color:white>Booking</button></a></p><hr/>");  
  
}  
  
Response.Write("</div>");
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
Response.Write("</div>");  
Response.Write("</div>");  
con.Close();  
  
conn = new SqlConnection(strcon);  
  
String str1 = "select * from blog where oname='"+ "'"+ n +"'";  
  
//String id = Request.QueryString.Get(0);  
  
  
//string ln = Session["logname"].ToString();  
// str = "select * from addmovies";  
SqlCommand cm1 = new SqlCommand(str1, conn);  
SqlDataReader reader1;  
conn.Open();  
// string cat = "", cat1 = "";  
reader1 = cm1.ExecuteReader();  
Response.Write("<body style=background-  
image:url(assets/images/choosing-bg.jpg) > ");  
  
Response.Write("<br><br><h1 style=color:white  
align=center>FEEDBACK</h1>");  
  
Response.Write("<table style=color:white;font-size:20; align=center  
border=5><tr><td>Name</td><td>FeedBack</td><td>Rating</td></tr><tr>");  
  
String s1 = "";  
while (reader1.Read())  
{  
    for (int i = 0; i <= 3; i++)  
    {  
        s1 = reader1.GetString(i).Trim();  
  
        if (i == 0)  
        {  
            Response.Write("<td>" + s1 + " </td>");  
        }  
  
        else if (i == 2)  
        {  
  
            Response.Write("<td>" + s1 + " </td>");  
        }  
        else if (i == 3)  
        {  
            Response.Write("<td>" + s1 + " </td>");  
        }  
  
    }  
  
    Response.Write("</tr>");  
}
```



```
}

Response.Write("</table>");
Response.Write("</body>");
// Response.Write("<a href=homepage.aspx> back </a>");
conn.Close();

}

catch (Exception ex)
{
    Response.Write(ex);
}

}

}
```

## Taking feedback with star ratings(frontend)

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="blogs.aspx.cs"
Inherits="blogs" Debug="true" %>

<!DOCTYPE html>
<script runat="server">

</script>

<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, shrink-
to-fit=no">
        <meta name="description" content="">
        <meta name="author" content="">
        <link
            href="https://fonts.googleapis.com/css?family=Montserrat:100,200,300,400,500,600,
            700,800,900" rel="stylesheet">
        <title>WARM HEARTS</title>
        <!-- Bootstrap core CSS -->
        <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
        <!-- Additional CSS Files -->
        <link rel="stylesheet" href="assets/css/fontawesome.css">
        <link rel="stylesheet" href="assets/css/templatemo-grad-school.css">
    </head>
    <body>
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
<link rel="stylesheet" href="assets/css/owl.css">
<link rel="stylesheet" href="assets/css/lightbox.css">
<link href="assets/css/stylenew.css" rel="stylesheet">
<!--

TemplateMo 557 Grad School

https://templatemo.com/tm-557-grad-school

-->
<style type="text/css">
    .auto-style5 {
        height: 55px;
        width: 388px;
    }
    .auto-style6 {
        height: 55px;
    }
    .auto-style9 {
        height: 18px;
        width: 388px;
    }
</style>
<script type="text/javascript">
    function fetch() {
        var a;
        if (document.querySelector('input[id="star-10"]:checked')) {
            a = document.getElementById("star-10").value;
            document.getElementById("TextBox4").value = a;
        }
        else if (document.querySelector('input[id="star-9"]:checked')) {
            a = document.getElementById("star-9").value;
            document.getElementById("TextBox4").value = a;
        }
        else if (document.querySelector('input[id="star-8"]:checked')) {
            a = document.getElementById("star-8").value;
            document.getElementById("TextBox4").value = a;
        }
        else if (document.querySelector('input[id="star-7"]:checked')) {
            a = document.getElementById("star-7").value;
            document.getElementById("TextBox4").value = a;
        }
        else if (document.querySelector('input[id="star-6"]:checked')) {
            a = document.getElementById("star-6").value;
            document.getElementById("TextBox4").value = a;
        }
    }
}

</script>
</head>

<body>
```



```
<form id="form1" runat="server">

    <!--header-->
<header class="main-header clearfix">
    <div class="logo">
        <a href="#"><em>WARM</em>HEARTS</a>
    </div>
    <nav id="menu" class="main-nav" role="navigation">
        <ul class="main-menu">
            <li><a href="donor.aspx">Home</a></li>
            <li><a href="editpwd.aspx">Edit Password</a></li>
            <li><a href="blogs1.aspx">Blogs</a></li>
            <li><a href="fooddetails.aspx">Upload Food Details</a></li>
            <li><a href="uploaddonation.aspx">Upload Donations</a></li>
            <li><a href="signin.aspx">Logout</a></li>
        </ul>
    </nav>
    <hr/>
</header>

    <section class="section main-banner" id="top" data-section="section1">
        <video autoplay muted loop id="bg-video">
            <source src="assets/images/course-video.mp4" type="video/mp4" />
        </video>
        <!-- Scripts -->
        <!-- Bootstrap core JavaScript -->
        <script src="vendor/jquery/jquery.min.js"></script>
        <script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>

        <script src="assets/js/isotope.min.js"></script>
        <script src="assets/js/owl-carousel.js"></script>
        <script src="assets/js/lightbox.js"></script>
        <script src="assets/js/tabs.js"></script>
        <script src="assets/js/video.js"></script>
        <script src="assets/js/slick-slider.js"></script>
        <script src="assets/js/custom.js"></script>

        <div class="video-overlay header-text">
            <div class="caption">
                <p> &nbsp;</p>
                <p> &nbsp;</p>
                <p> &nbsp;</p>
                <center><h1 style="color:aliceblue">GIVE YOUR FEEDBACK HERE</h1>
                    <ContentTemplate><table style="font-family:Verdana;font-size:125%;width:702px;">
                        <tr>

```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
<td align="center" colspan="2" style="color:White;background-color:aliceblue;font-weight:bold;">
</td>
</tr>
<tr>
<td class="auto-style5">
<asp:Label ID="Label1" runat="server" Text="Name"
ForeColor="White"></asp:Label>
</td>
<td class="auto-style6">
<asp:TextBox ID="TextBox1" runat="server" ToolTip="Your Name"
Width="278px"></asp:TextBox>
</td>
</tr>
<tr>
<td class="auto-style5">
<asp:Label ID="Label6" runat="server" ForeColor="White"
Text="Orphanage Name"></asp:Label>
</td>
<td class="auto-style6">
<asp:TextBox ID="TextBox3" runat="server"></asp:TextBox>
</td>
</tr>
<tr>
<td class="auto-style5">
<asp:Label ID="Label5" runat="server" ForeColor="White"
Text="Feedback"></asp:Label>
</td>
<td class="auto-style6">
<asp:TextBox ID="TextBox2" runat="server" ToolTip="Few
opinions on the Orphanage" Width="285px" Columns="5" Rows="3"
TextMode="MultiLine"></asp:TextBox>
</td>
</tr>
<tr>
<td class="auto-style5">
<asp:Label ID="Label7" runat="server" ForeColor="White"
Text="Ratings"></asp:Label>
</td>
<td class="auto-style6">
<div class="stars">
<input name="star" type="radio" class="star-1"
id="star-6" value="1" />
<label class="star-1" for="star-6">1</label>
<input name="star" type="radio" class="star-2"
id="star-7" value="2" />
<label class="star-2" for="star-7">2</label>
<input name="star" type="radio" class="star-3"
id="star-8" value="3" />
<label class="star-3" for="star-8">3</label>
<input name="star" type="radio" class="star-4"
id="star-9" value="4" />
<label class="star-4" for="star-9">4</label>
<input name="star" type="radio" class="star-5"
id="star-10" value="5" />
<label class="star-5" for="star-10">5</label>
<span></span> </div>
```



```
<asp:TextBox runat="server" id="TextBox4"
Visible="false"></asp:TextBox>
    <center><button style="background-color:orange; border-
color:orange; color:white" onclick="fetch()">Take Rating</button></center>

    </td>

</tr>
<tr align="center">
    <td class="auto-style9" colspan="2">
        <asp:Button ID="Button1" align="center" runat="server"
Text="Submit" BackColor="#0099CC" ForeColor="Black" BorderColor="#003366"
OnClick="Button1_Click"/>
    </td>
</tr>

</table>
</center>

</div>
    </div>
</section>

</form>

</body>
</html>
```

### Taking feedback with star ratings(backend)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Configuration;

public partial class blogs : System.Web.UI.Page
{
    SqlConnection con;
    SqlConnection con1;
    protected void Page_Load(object sender, EventArgs e)
    {
        String n = Request.QueryString.Get(0);
        TextBox3.Text=n;
        TextBox1.Text = (string)Session["dname"];
    }
}
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
protected void Button1_Click(object sender, EventArgs e)
{
    try
    {

        string strcon =
ConfigurationManager.ConnectionStrings["dbconnection"].ConnectionString;
        con = new SqlConnection(strcon);
        String s = "insert into blog values(@name,@oname,@blog,@rating)";
        SqlCommand cmd1 = new SqlCommand(s, con);
        cmd1 = new SqlCommand(s, con);

        cmd1.Parameters.Add("@name", TextBox1.Text);
        cmd1.Parameters.Add("@oname", TextBox3.Text);
        cmd1.Parameters.Add("@blog", TextBox2.Text);
        cmd1.Parameters.Add("@rating", TextBox4.Text);

        con.Open();
        cmd1.ExecuteNonQuery();
        con.Close();
        //    Label1.Text = "Registered successfully";
        Response.Write("<script>alert('Feedback Uploaded
successfully!!!');window.location='donor.aspx';</script>");

        TextBox1.Text = "";
        TextBox3.Text = "";
        TextBox2.Text = "";
        //TextBox4.Text = "";

        //Session["name"] = TextBox1.Text.Trim();
        //Session["address"] = TextBox6.Text.Trim();
        //Session["mobile"] = TextBox2.Text.Trim();

    }
    catch (Exception ex)
    {
        Label7.Text = ex.ToString();
    }
}
```



## Database Connection Using Connection String

```
<?xml version="1.0"?>
<!--
  --
<configuration>
  <!--
  -->
  <appSettings>
    <add key="ChartImageHandler"
value="storage=file;timeout=20;dir=c:\TempImageFiles\;" />
  </appSettings>
  <system.webServer>
    <validation validateIntegratedModeConfiguration="false" />
    <handlers>
      <remove name="ChartImageHandler" />
      <add name="ChartImageHandler" preCondition="integratedMode"
verb="GET,HEAD,POST" path="ChartImg.axd"
type="System.Web.UI.DataVisualization.Charting.ChartHttpHandler,
System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" />
    </handlers>
  </system.webServer>
  <system.web>
    <httpHandlers>
      <add path="ChartImg.axd" verb="GET,HEAD,POST"
type="System.Web.UI.DataVisualization.Charting.ChartHttpHandler,
System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" validate="false" />
    </httpHandlers>
    <pages>
      <controls>
        <add tagPrefix="asp" namespace="System.Web.UI.DataVisualization.Charting"
assembly="System.Web.DataVisualization, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=31bf3856ad364e35" />
      </controls>
    </pages>
    <compilation debug="true" targetFramework="4.7.1">
      <assemblies>
        <add assembly="System.Core, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
        <add assembly="System.Data.Linq, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
        <add assembly="System.Xml.Linq, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
        <add assembly="System.Web.Entity, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
        <add assembly="System.Data.Entity, Version=4.0.0.0, Culture=neutral,
PublicKeyToken=B77A5C561934E089" />
        <add assembly="System.Web.DataVisualization, Version=4.0.0.0,
Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
      </assemblies>
    </compilation>
  </system.web>
  <connectionStrings>
```



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

```
<add name="dbconnection" connectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=C:\We-
care\App_Data\Database.mdf;Integrated Security=True"/>
<add name="ConnectionString" connectionString="Data
Source=(LocalDB)\MSSQLLocalDB;AttachDbFilename=|DataDirectory|\Database.mdf;Integ
rated Security=True" providerName="System.Data.SqlClient"/>
</connectionStrings>
</configuration>
```



## TESTING

Testing is the process of analyzing a software item to detect the differences between existing and required conditions and to evaluate the features of the software item.

System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS).

### Purpose of Testing:

- Testing is a process of executing a program with the intent of finding an error.
- To check whether the system is working as required or not.
- Testing is to cause failures in order to make faults visible, so that faults can be fixed and not be delivered in the code that goes to customer.
- To prove that the software system is free of any errors.
- Testing is used to establish the presence of defects and to show fitness for purpose.

### Testing Principles:

- All tests should be traceable to customer requirement.
- Testing should be planned long before testing begins.
- Exhaustive testing is not impossible.
- Testing by third party will be very effective.



## Test Process:

1. Unit Testing
2. Module Testing
3. Integration Testing
4. System Testing
5. Acceptance Testing

### 1. Unit Testing:

Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components.

### 2. Module Testing:

A module is a collection of dependent components such as an object class, an abstract data type or some collection of procedures and functions. A module encapsulates related components so that it can be tested without referring other system modules.

### 3. Integration Testing:

This phase involves testing collection of modules which have been integrated into sub-systems. Sub-systems may be independently designed and implemented. The most common problems which arise in large software system are subsystem interface mismatches. The subsystem test process should therefore concentrate on the detection of interface errors by rigorously exercising these interfaces.

### 4. System Testing:

The subsystems are integrated to make-up the entire system. The testing process is concerned with finding errors that result from unanticipated interactions between sub-systems and system components. It is also concerned with validating that the system meets its functional and non-functional requirements.



## 5. Acceptance Testing:

This is the final stage in the testing process before the system is accepted for operational use. The system is tested with data supplied by the client rather than simulated test data.

## TEST CASE

Project Name	WARM HEARTS		
Module Name	Sign-in		
Checked by	Shreyas G Bharadwaj		
Created date	07-Sept-2022		
Reviewed by	Sangeetha S		
Reviewed date	10-Sept-2022		



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

## SAMPLE OUT

**WARMHEARTS**

HOME ABOUT US COURSES DONOR RECEIVER CONTACT SIGN IN

WELCOME TO OUR WORLD

**WARMHEARTS**

JOIN WITH US

**BLOGS**  
Anybody can share their thoughts that they have experienced when they visited any Orphanage by writing the BLOGS  
[MORE INFO](#)

**DONATIONS**

**ONLINE CLASSES**

**WARMHEARTS**

HOME ABOUT US COURSES DONOR RECEIVER CONTACT SIGN IN

How can you help?

**DONATION**

**EDUCATION**

**VISIT**

**Donation**

Children face a variety of economic, mental, emotional, and physical challenges as they grow. Proper support helps children navigate through those struggles on their journey to adulthood. Without proper support, they face those difficulties alone. Our donors' support has made it so that thousands of children haven't had to face those challenges alone. Thank you.



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

localhost:63627/about.aspx

## WARMHEARTS

HOME ABOUT US COURSES DONOR RECEIVER CONTACT SIGN IN

KNOW HOW VOLUNTEERS WORK

**Watch the sample video to know how volunteers work**

There are already many volunteers working with us to provide proper knowledge, proper health awareness. They give their precious time to teach the orphans.

So if you are interested to join your hands with us, apply here.

[CLICK HERE](#)

Sample Video

Yoga For Beginners :Part 1 | Yoga Routine | Yoga At Home

**YOGA FOR BEGINNERS PART 1**

Watch on YouTube

© CONTACT US VIA EMAIL | SANGEETHA S: SANGEETHA29SANGU@GMAIL.COM | SHREYAS G BHARADWAJ: SHREYASBHARADWAJ002@GMAIL.COM

localhost:63627/course.aspx

## WARMHEARTS

HOME ABOUT US COURSES DONOR RECEIVER CONTACT SIGN IN

Choose Your Course

**YOGA CLASS**  
You can get free images and videos for your websites by visiting Unsplash, Pixabay, and Pexels.  
 Contact»

**ART AND CRAFT MAKING**  
elit dictum, quis voluptat enim blandit. Maecenas a lectus ac ipsum porta.  
 Contact»

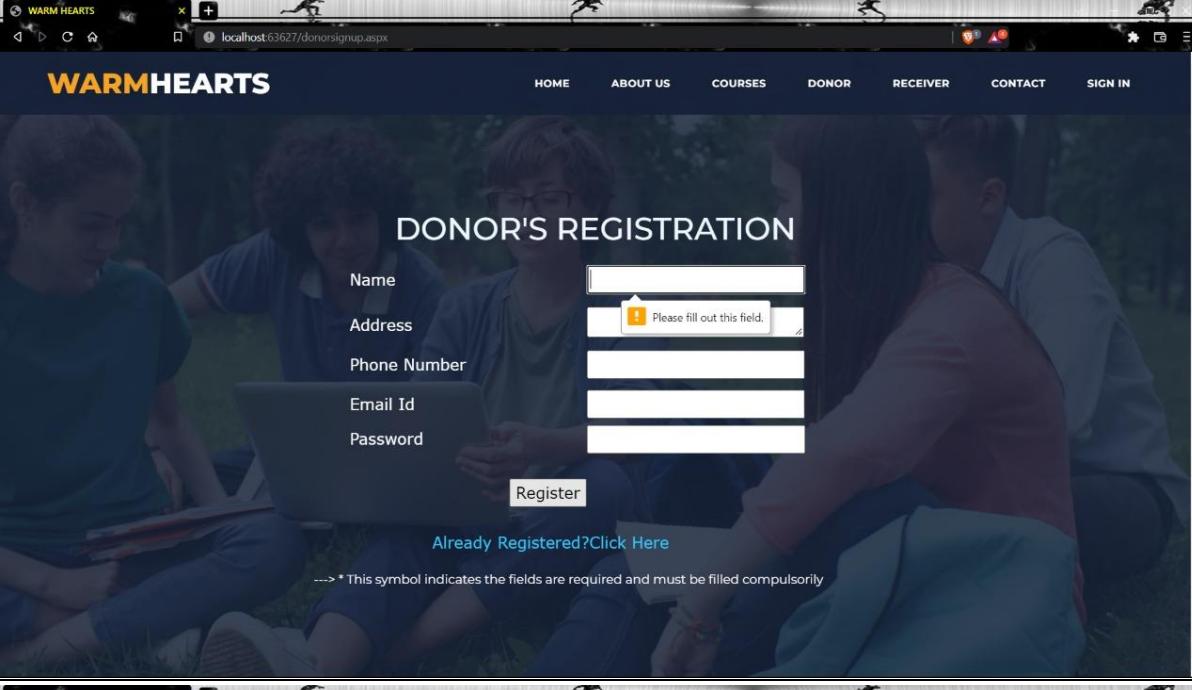
**AEROBICS**  
Pellentesque ultricies diam magna, auctor cursus lectus pretium nec. Maecenas finibus lobortis enim.  
 Contact»

**SANGEETHA CLASS**  
Pellentesque ultricies diam magna, auctor cursus lectus pretium nec. Maecenas finibus lobortis enim.  
 Contact»

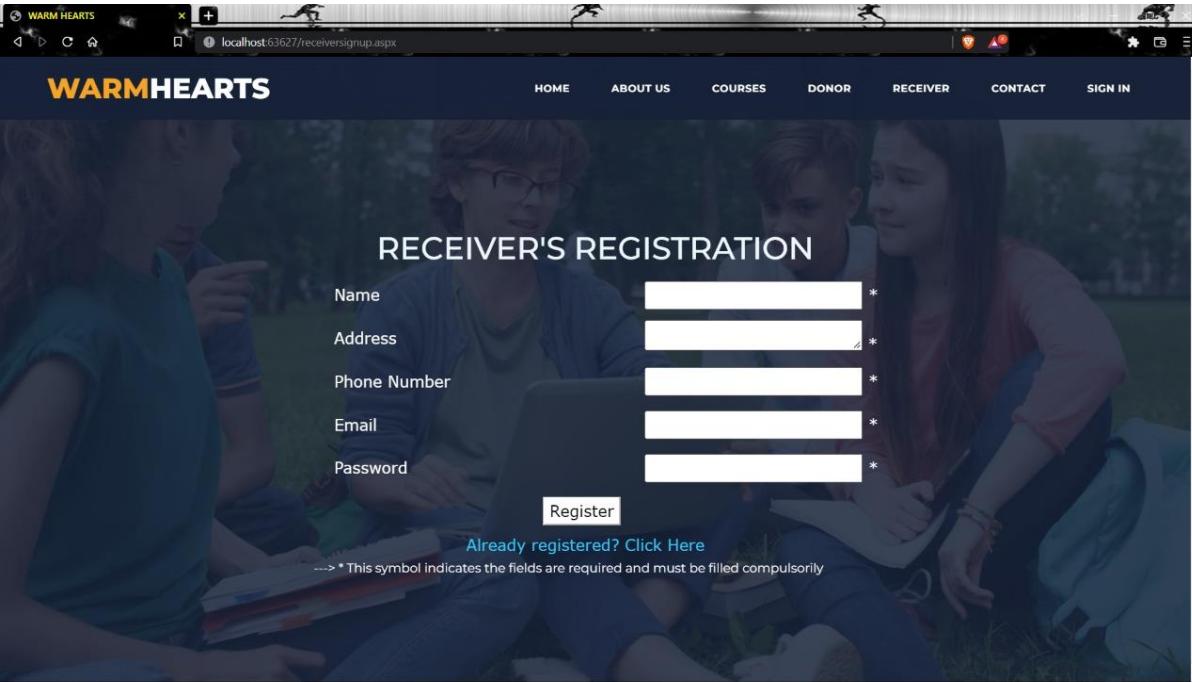
**DANCE CLASS**  
Download free images and videos for your websites by visiting Unsplash, Pixabay, and Pexels.  
 Contact»



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022



The screenshot shows the "DONOR'S REGISTRATION" page. At the top, there is a navigation bar with links for HOME, ABOUT US, COURSES, DONOR, RECEIVER, CONTACT, and SIGN IN. Below the navigation bar is a banner image of children sitting outdoors. The registration form consists of five input fields: Name, Address, Phone Number, Email Id, and Password. The "Name" field has a validation message: "Please fill out this field." A "Register" button is located at the bottom of the form. Below the form, there is a link "Already Registered? Click Here" and a note: "...> \* This symbol indicates the fields are required and must be filled compulsorily".



The screenshot shows the "RECEIVER'S REGISTRATION" page. The layout is identical to the donor registration page, with a navigation bar, a banner image of children, and a registration form with five input fields: Name, Address, Phone Number, Email, and Password. Each of these fields has an asterisk (\*) indicating they are required. A "Register" button is at the bottom, and a note below the form states: "...> \* This symbol indicates the fields are required and must be filled compulsorily".



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

The screenshot shows a web browser window with two tabs. The top tab is titled "WARM HEARTS" and the URL is "localhost:63627/blogindex.aspx". The bottom tab is titled "localhost:63627/admin/sendmessage" and the URL is "localhost:63627/blogindex.aspx".

The main content area has a dark background image of four students sitting outdoors. At the top left, it says "WARMHEARTS". At the top right, there are navigation links: HOME, ABOUT US, COURSES, DONOR, RECEIVER, CONTACT, and SIGN IN.

A form titled "BLOG HERE" is displayed. It includes fields for "ID" (101), "Name" (Shreyas), "Blog" (containing the text "had. Thank you for providing such wonderful opportunity. Keep growing"), and "Upload Image" (with a file input field showing "Choose File abalashrama.webp"). A "Submit" button is located below the image upload field.

The bottom tab shows a confirmation message: "localhost:63627 says Blog Uploaded successfully!!" with an "OK" button.



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

The screenshot shows the 'CONTACTUS' page of the WARMHEARTS website. It features a dark blue header with the 'WARMHEARTS' logo and a navigation menu with links to HOME, ABOUT US, COURSES, DONOR, RECEIVER, CONTACT, and SIGN IN. Below the header is a contact form with fields for 'Your Name', 'Your Email', and 'Your message...', and a 'Send Message Now' button. To the right of the form is a map of the Jayanagar area in Udupi, showing various landmarks like Sri Raghavendra Matha, Sri Kurnoor Childrens Hospital, Yediyur Lake Park, and Surana College.

The screenshot shows the 'TELEGRAM' page of the WARMHEARTS website. It features a dark blue header with the 'WARMHEARTS' logo and a navigation menu with links to HOME, ABOUT US, COURSES, DONOR, RECEIVER, CONTACT, and SIGN IN. Below the header are two large boxes, each containing a QR code and a link to join the corresponding Telegram channel. The left box is for the 'DONOR' channel (@WARM\_HEARTS1) and the right box is for the 'RECEIVER' channel (@WARM\_HEARTSS). Both boxes include the text 'Click Here To Join The Telegram Channel' at the bottom.



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

WARMHEARTS

HOME ABOUT US COURSES DONOR RECEIVER CONTACT SIGN IN

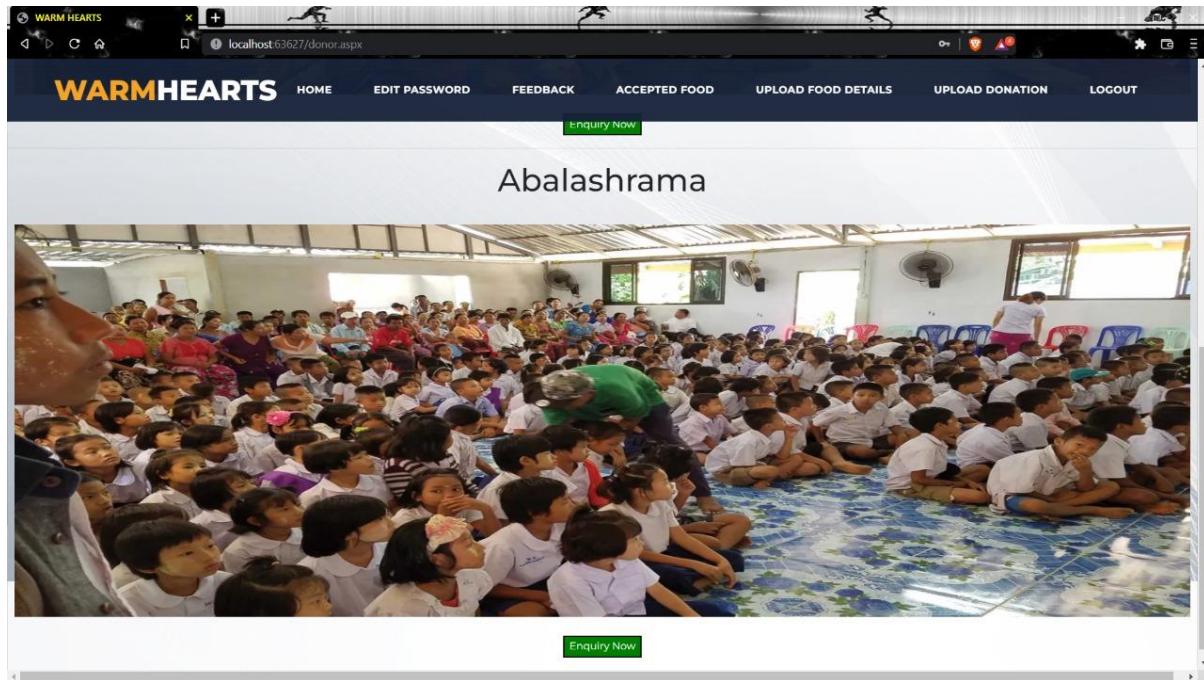
LOGIN HERE

Email-id

Password

Login As

[Forgot Password?](#)





SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

WARM HEARTS

HOME EDIT PASSWORD FEEDBACK ACCEPTED FOOD UPLOAD FOOD DETAILS UPLOAD DONATION LOGOUT

6362093374  
pooja@gmail.com  
Gandhi Bazar

60 students are in need of uniform, stationaries, books and school fees. We admits girls who have attained 16 years of age. Our vision is to make vulnerable women lead a self-reliant, honorable and dignified life which make them worthy members of the society.

Donate Course Feedback Booking

## FEEDBACK

Name	FeedBack	Rating
Sahana	I personally suggest you to visit once to this orphanage and do your needful because i liked the way they treat the children and taking care of them like their parents.	4

WARM HEARTS

HOME EDIT PASSWORD FEEDBACK ACCEPTED FOOD UPLOAD FOOD DETAILS UPLOAD DONATIONS LOGOUT

## Type of Donation

FOOD BOOKS MONEY CLOTHES

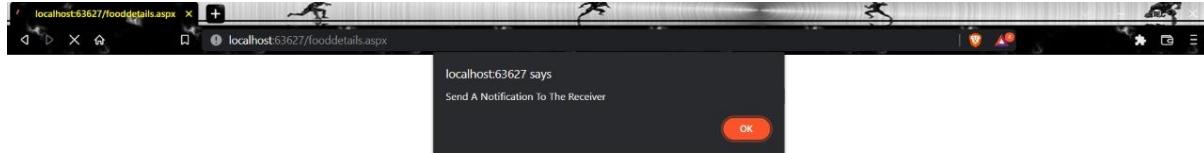


SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

The screenshot shows a web browser window with two tabs. The top tab displays the 'WARMHEARTS' website with a dark background image of people. The main content area is titled 'UPLOAD FOOD DETAILS'. The form fields are as follows:

Id	111
Name	Sangeetha
Phone Number	9900107168
Email Id	sangeetha@gmail.com
Address	(empty)
Food Type	(empty)
Number of people that can be fed	(empty)
Upload Image	Choose File No file chosen
Prepared Date	10-09-2022
Expiry Date	dd-mm-yyyy <input type="button" value="Calendar"/>

Below the form is a button labeled 'Upload'.





SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

TELEGRAM NOTIFICATION

Send A Notification To The Receiver

Enter Your Name

SEND

CLEAR

Food Id	Receiver's Name	Receiver's Phone Number	Donor's Email-Id	Receiver's Email-Id	Receiver's Address	Food Type	Number Of People That Can Be Fed	Food Image	Prepared Date	Expiry Date	Status
104	Shreyas G Bharadwaj	9900107168	sangeetha@gmail.com	sangeetha@gmail.com	Jayanagar	Vegetarian	20		08-09-2022	08-09-2022	Accepted
108	Pooja	6362093374	sangeetha@gmail.com	pooja@gmail.com	Srinagar	Vegetarian	12		09-09-2022	2022-09-11	Accepted



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

6:50 PM ⏱️ ⏳ ⏴ ⏵ ⏷

WARM HEARTS RECEIVERS 3 members

September 9

Pintuu created the group

Warm Hearts admin  
Sangeetha has uploaded a food, please check your notification. 9:51 PM

Ok thank you 9:53 PM ✓✓

You changed the group name to WARM\_HEARTS\_RECEIVER

Warm Hearts admin  
hii has uploaded a food, please check your notification. 10:34 PM

Vasuda has uploaded a food, please check your notification. 10:38 PM

Pintuu changed the group name to WARM HEARTS RECEIVER

Pintuu changed the group name to WARM HEARTS RECEIVERS

September 10

Warm Hearts admin  
Sandesh Mukherji has uploaded a food, please check your notification. 11:29 AM

Message

76 | WARM HEARTS



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

The screenshot shows a web application titled "WARMHEARTS" with a dark blue header bar. The header includes the logo, the word "WARMHEARTS" in white, and navigation links: HOME, ADD ORPHANAGE DETAILS, EDIT PASSWORD, NOTIFICATIONS, DONATION DETAILS, and LOGOUT. Below the header is a banner image of children sitting outdoors, one using a laptop. The main content area has a title "BOOKS" and a form for donations. The form fields are:

Name	Sangeetha
Phone Number	9900107168
Email Id	sangeetha@gmail.com
No of Clothes	20

A "Donate" button is at the bottom of the form. The URL in the browser address bar is "localhost:63627/clothes.aspx".

localhost:63627 says  
Donation is Confirmed. Please wait until receiver contacts you

OK



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

The screenshot shows a web browser window with two tabs. The top tab is titled "WARM HEARTS" and displays a form for "CLOTHES" donation. The form fields are: Name (Sangeetha), Phone Number (9900107168), Email Id (sangeetha@gmail.com), and No of Clothes (30). A "Donate" button is visible. The background of the page features a photograph of children sitting outdoors. The bottom tab is titled "localhost:63627/books.aspx" and shows a confirmation message: "localhost:63627 says Donation is Confirmed. Please wait until receiver contacts you" with an "OK" button.



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

The screenshot displays two pages of the WARMHEARTS website:

**Top Page (localhost:63627/money.aspx):**

Orphanage Name	Email	Phone Number
Kalasiri_Charitable_Trust	pooja@gmail.com	9900107168
Abalashrama	pooja@gmail.com	6362093374
BLESS_TRUST_FOR_RURAL	pooja@gmail.com	9972094387

**Bottom Page (localhost:63627/booking.aspx?&=pooja@gmail.com&name=Kalasiri\_Charitable\_Trust):**

BOOK AN ORPHANAGE

Orphanage Name: Kalasiri\_Charitable\_Trust

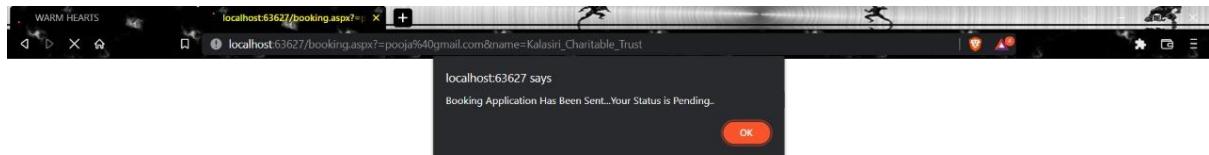
Enter Your Name: Sangeetha

Enter Purpose And Date Of Visit:  
Hello!  
This Sunday we would like to visit your orphanage i.e. 11th september!

SEND CLEAR



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022



**WARMHEARTS**   [HOME](#)   [EDIT PASSWORD](#)   [BOOKING](#)   [ACCEPTED FOOD](#)   [UPLOAD FOOD DETAILS](#)   [UPLOAD DONATION](#)   [LOGOUT](#)

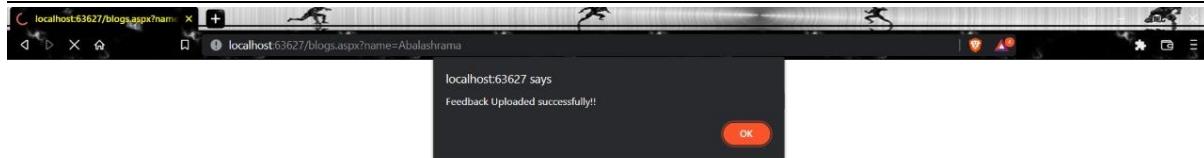
## BOOKED ORPHANAGES

Name	Orphanage Name	Purpose	Status
Sangeetha	Kalasiri_Charitable_Trust	Hello..!! This Sunday i.e. 11th August 2022 we wish to visit.	Booked



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

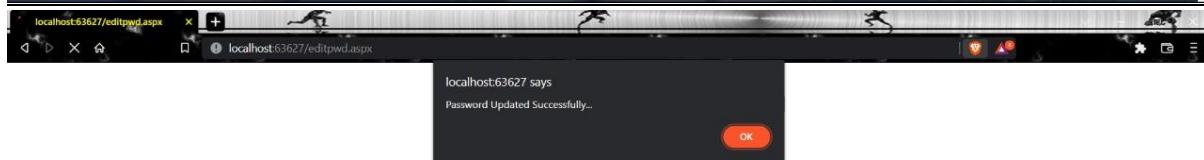
The screenshot shows a feedback form titled "GIVE YOUR FEEDBACK HERE". It includes fields for Name (Sangeetha), Orphanage Name (Kalasiri\_Charitable\_Trust), Feedback (empty), Ratings (5 stars), and a Submit button. The background features a photograph of children sitting outdoors.





SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

The screenshot shows a web browser window with the URL `localhost:63627/editpwd.aspx`. The page is titled "WARMHEARTS" and has a navigation bar with links for HOME, EDIT PASSWORD, FEEDBACK, ACCEPTED FOOD, UPLOAD FOOD DETAILS, UPLOAD DONATION, and LOGOUT. The main content area is titled "EDIT PASSWORD" and contains three input fields: "Old Password" (Sangu123), "New Password" (Sangu321), and "Confirm Password" (Sangu321). A blue "Submit" button is located below the fields. The background of the page features a photograph of four children sitting outdoors, looking at a laptop together.





SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

The image displays two screenshots of a web application named "WARMHEARTS".

The top screenshot shows the homepage with a dark blue background featuring school-related icons like books, pencils, and backpacks. It features a video thumbnail for "Heart of an orphan: Amy Eldridge at TEDxSHA" and another for "This Race Called Life - a beautiful inspirational short-sto...". The navigation bar includes links for HOME, ADD ORPHANAGE DETAILS, BOOKING, EDIT PASSWORD, NOTIFICATIONS, and LOGOUT.

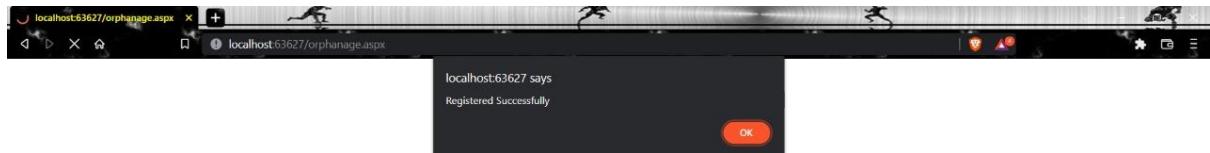
The bottom screenshot shows the "ORPHANAGE DETAILS" form. The background is a photograph of three children sitting outdoors, one using a laptop. The form fields are as follows:

ID	102
Orphanage Name	BLESS_TRUST_FOR_RURA
Image	Choose File new.jpg
Address	Penya
Your Requirements	60 students are in need of
Contact Number	9900107168
Email Id	pooja@gmail.com

A "Submit" button is located at the bottom of the form.



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022



The screenshot shows a web application titled "WARM HEARTS" with a dark blue header. The header includes links for "HOME", "EDIT PASSWORD", "FEEDBACK", "ACCEPTED FOOD", "UPLOAD FOOD DETAILS", "UPLOAD DONATION", and "LOGOUT". Below the header is a table with the following data:

Name	Oname	Email-id	purpose	Status	Action
Sangeetha	Kalasiri_Charitable_Trust	pooja@gmail.com	Hello!! This Sunday we would like to visit your orphanage i.e. 11th september.	Pending	<button>Accept</button>



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

ID	Donor Name	Phone Number	Email Id	Address	Type Of Food	Number Of People Can Be Fed	Image Of Food	Prepared Date	Expiry Date	Status
100	Pooja K	9900107168	pooja@gmail.com	Thyagarajanagara	Non Vegetarian	10		30-08-2022	2022-08-30	EXPIRED
101	Pooja K	6362099374	pooja@gmail.com	Srinagar	Vegetarian	30		01-09-2022	31-10-2022	ACCEPT
102	Vasuda	6362099374	vasuda@gmail.com	JP Nagara	Vegetarian	10		01-09-2022	2022-09-02	EXPIRED
103	admin	9900107168		Sadhashivnagara	Vegetarian	10		08-09-2022	2022-09-08	EXPIRED
106	Sangeetha	9900107168	sangeetha@gmail.com	Jaynagar	Vegetarian	20		09-09-2022	2022-09-10	ACCEPT
107	Sangeetha	9900107168	sangeetha@gmail.com	JP NAGARA	Vegetarian	20		09-09-2022	2022-09-17	ACCEPT

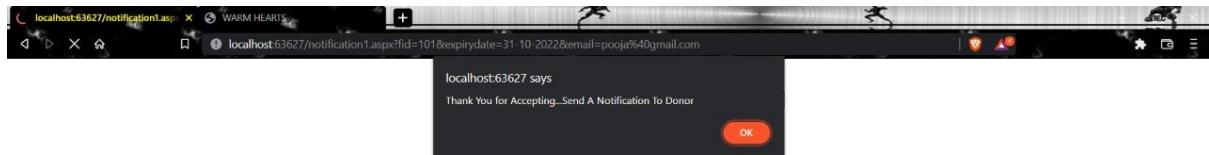
ACCEPT THE FOOD

<b>Id</b>	101
<b>Name</b>	Pooja
<b>Phone Number</b>	6362099374
<b>Donor's E-mail id</b>	pooja@gmail.com
<b>Your E-mail Id</b>	pooja@gmail.com
<b>Address</b>	Shivajinagar
<b>Acceptance Date</b>	31-10-2022

**Accept**



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022



A screenshot of a web application titled "WARMHEARTS". The top navigation bar includes links for HOME, EDIT PASSWORD, FEEDBACK, ACCEPTED FOOD (which is highlighted in yellow), UPLOAD FOOD DETAILS, UPLOAD DONATION, and LOGOUT. The main content area features a dark background image of children. Overlaid on this is a "TELEGRAM NOTIFICATION" form. The form has a title "Send A Notification To The Donor" and two input fields: "Enter Your Name" and "Enter Donor's Name". Below the fields are two buttons: a green "SEND" button and a red "CLEAR" button.



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

7:12 PM 4G LTE 58%

← WARM HEARTS DONORS 4 members :

September 10

WH Warm Hearts admin  
Hi 12:43 PM

WH Warm Hearts admin  
Pooja has accepted food donated by Sangeetha, please check the notification. 12:53 PM

Pintuu changed the group name to WARM HEARTS DONORS

Message



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

WARM HEARTS

CONTACT TRAINER FOR THE MEETING-LINK

Volunteer Name	Type of Course	Available Time	Phone Number	Email Id
Shreyas	Dance Class	4pm-5pm	9905483833	shreyasbharadwaj002@gmail.com

WARM HEARTS

CONTACT TRAINER FOR THE MEETING-LINK

Name	type of Course	Available time	Phone Number	Email Id
Sunitha	Art and Crafts	4pm-5pm	9663813444	sunitha@gmail.com



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

The screenshot shows a web application titled "WARMHEARTS" with a dark-themed header bar. The header includes the logo, the word "WARMHEARTS" in yellow, and navigation links: HOME, ADD ORPHANAGE DETAILS, EDIT PASSWORD, NOTIFICATIONS, DONATION DETAILS, and LOGOUT. Below the header, a banner features a group of people sitting outdoors, one person using a laptop. The main content area has a title "CONTACT TRAINER FOR THE MEETING-LINK". A table displays the following information:

Name	Type of Course	Available Time	Phone Number	Email Id
Shreesha	Sangeetha Class	6pm-7pm	8774849839	shreesha@gmail.com

The second screenshot shows the same application interface. The main content area has a title "CONTACT TRAINER FOR THE MEETING-LINK". A table displays the following information:

Name	Type of Course	Available Time	Phone Number	Email Id
Karthik	Aerobics	6am-7am	7099123877	karthi@gmail.com



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

WARMHEARTS

HOME ABOUT US COURSES DONOR RECEIVER CONTACT SIGN IN

LOGIN HERE

Email-id: pooja@gmail.com

Password: Enter Your Password

Login As: Admin

Invalid username or password!

Login

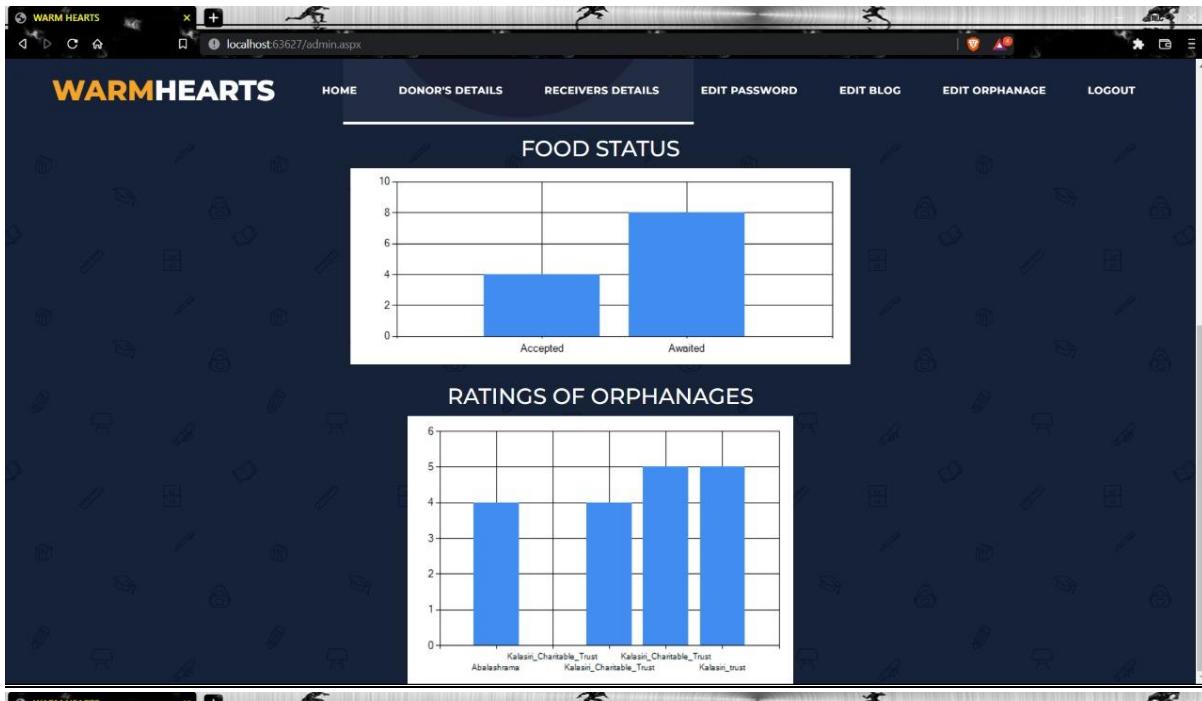
[Forgot Password?](#)

[SIGN IN](#)





SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022



The screenshot displays a table of donor details overlaid on a background image of children. The table includes columns for ID, Name, Email, Phone Number, and Address.

id	name	email	phno	address
6	Vasuda	vasuda@gmail.com	6362093374	Hanumanthanagar
7	Sangeetha	sangeetha@gmail.com	9900107168	Srinagar
8	Shreesha G Bharadwaj	shreesha@gmail.com	6362099374	Jayanagar
9	VenuGopala	venugopala@gmail.com	997248066	Sarjapura



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

The screenshot shows a web browser window titled "WARM HEARTS" with the URL "localhost:63627/arecdetails.aspx". The page header includes links for "HOME", "EDIT PASSWORD", "DONOR'S DETAILS", "RECEIVERS DETAILS", and "LOGOUT". The main content area is titled "RECEIVER DETAILS" and displays a table with the following data:

id	name	emailId	phno	address
1	Pooja	pooja@gmail.com	6362093374	Srinagar
2	Srinkanth	srikanth@gmail.com	7038839322	RR nagar

The screenshot shows a web browser window titled "WARM HEARTS" with the URL "localhost:63627/ablog.aspx". The page header includes links for "HOME", "EDIT PASSWORD", "DONOR'S DETAILS", "RECEIVERS DETAILS", and "LOGOUT". The main content area is titled "WARMHEARTS" and displays a table with the following data:

ID	NAME	BLOG
100SHREYAS	SHREYAS	This was very wonderful experience. With the help of this website i could connect with many orphanages and donte the needs they had. Thank you for providing such wondrful opportunity.Keep growing



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

WARM HEARTS

localhost:63627/ablog.aspx

HOME EDIT PASSWORD DONOR'S DETAILS RECEIVERS DETAILS LOGOUT

ID	NAME	BLOG
100		

Update Cancel

WARM HEARTS

localhost:63627/orphanage.aspx

HOME EDIT PASSWORD DONOR'S DETAILS RECEIVERS DETAILS LOGOUT

ID	NAME	PHONE NUMBER	EMAIL ID	ADDRESS	REQUIREMENTS	IMAGE
100	Kalasiri Charitable Trust	9900107168	pooja@gmail.com	Uttarahalli	A Trust with 200 childrens provided with sufficient facilities and Education	
101	Abakashrama	6362093374	pooja@gmail.com	Gandhi Bazar	60 students are in need of uniform, stationaries, books and school fees. We admits girls who have attained 16 years of age. Our vision is to make vulnerable women lead a self-reliant, honorable and dignified life which make them worthy members of the society.	
102	BLESS TRUST FOR RURAL			Penya	60 students are in need of uniform, stationaries, books and school fees. We admits girls who have attained 16 years of age. Our vision is to make vulnerable women lead a self-reliant, honorable and dignified life which make them worthy members of the society.	



## Select command for sessions and status using where clause

The screenshot shows two instances of the 'Configure Data Source - SqlDataSource' dialog box in Microsoft Visual Studio. In the top instance, the 'Specify columns from a table or view' option is selected, with 'Name' set to 'fdetails'. Under 'Columns', several checkboxes are checked, including 'fid', 'name', 'email', 'address', 'fptype', 'number', 'fimage', and 'pdate'. The 'SELECT statement' field contains the query: 'SELECT \* FROM [fdetails]'. In the bottom instance, the 'Add WHERE Clause' dialog is open, showing a table with two rows under 'WHERE clause': '[email] = @email' with 'Value' 'Session("email")' and '[status] = @status' with 'Value' 'Accepted'. The 'OK' button is visible at the bottom of this dialog.



## **TELEGRAM IMPLEMENTATION**

### **Telegram integrating from python**

```
import requests
class test2:
    SEND_MESSAGE_URL = ""
    def __init__(self, token, channelname) -> None:
        self.token = token
        self.channelname = channelname

    def rsendmessage(self, message,message1):
        SEND_MESSAGE_URL =
        'https://api.telegram.org/bot{bot_token}/sendMessage?chat_id=@{channel}&text={msg2} has accepted food donated by {msg1}, please check the notification.'.format(bot_token=self.token,channel=self.channelname,msg2=message,msg1=message1)
        requests.get(SEND_MESSAGE_URL)
```

### **API key and token for telegram**

```
t2=test2("5464240136: AAGKxznl8kTleWiiJthoqivQcB2-gKwrpFg","Warm_Hearts1")
```

### **Importing files for telegram**

```
from Htelegram import test2
```

### **GET and POST requests**

```
@app.get("/admin/rsendmessage", response_class=HTMLResponse)
def rsend(request : Request):
    return templates.TemplateResponse("/admin/send2.html", {"request": request})

@app.post("/admin/rsendmessage",response_class=HTMLResponse)
def resend(request : Request, message : str = Form(...),
           message1:str=Form(...)):
    t2.rsendmessage(message,message1)
```



```
return  
RedirectResponse("http://localhost:63627/receiver.aspx",status_code=sta  
tus.HTTP_302_FOUND)
```

### **Telegram integrating from python**

```
import requests  
class testt1:  
    SEND_MESSAGE_URL = ""  
    def __init__(self, token, channelname) -> None:  
        self.token = token  
        self.channelname = channelname  
    def rsendmessage(self, message):  
        SEND_MESSAGE_URL =  
        'https://api.telegram.org/bot{bot_token}/sendMessage?chat_id=@{channe  
l}&text={msg2} has accepted food donated by {msg1}, please check the  
notification.'.format(bot_token=self.token,channel=self.channelname,msg2  
=message)  
        requests.get(SEND_MESSAGE_URL)
```

### **GET and POST requests**

```
@app.get("/admin/rsendmessage", response_class=HTMLResponse)  
def rsend(request : Request):  
    return templates.TemplateResponse("/admin/send1.html",{"request" :  
request})  
  
@app.post("/admin/dsendmessage",response_class=HTMLResponse)  
def resend(request : Request, message : str = Form(...),  
message1:str=Form(...)):  
    t2.rsendmessage(message,message1)  
    return  
    RedirectResponse("http://localhost:63627/donor.aspx",status_code=sta  
tus.HTTP_302_FOUND)
```

### **API key and token for telegram**

```
t1=testt1("5727830001:AAHvu_9zDl6FcVZtKVbhOpfRbVXfFMLKVdo","war  
m_heartss")
```

### **Importing files for telegram**

```
from Wtelegram import testt1
```



## Future Enhancement

Connecting donor with every orphanage is the goal of the project. Making the website more dynamic by tracking the location from where the donation has been sent so that the nearest orphanage can receive the food before the expiry of the food.

Donating money to the orphanage using razor pay api methods is our further goal.

Providing the meeting links to the orphanage by the volunteers and making them more interactive is our goal where it helps orphans to learn extra curriculum activities.

Sending updated information about the website and orphanage through normal messages which can be accessed offline also is our further goal. As of now to achieve that we have integrated telegram which is free of cost.

## Conclusion

This project is developed with ASP. Net framework and C# programming language. The project is completely computerizing the existing system. It's not all about giving back to the community, though. Volunteering can actually teach us invaluable lessons.

When we help orphanages and orphans, you we have the opportunity to discover aspects of ourselves that we did not realize we had.

Let's take a break from our everyday routine to learn more about ourself, gain a renewed perspective about our life, and plan for our future. Live every moment to the fullest and create lifelong memories with people of all ages.

Helping orphanages and orphans can promote health, literacy, education, and children's empowerment.

"Your small contribution makes a huge difference on someone's life. So, Be the reason someone believes in the goodness of people...!"

We would also like to say that it was very overwhelming to work with C#. Thank you!!



SURANA COLLEGE  
Department Of Computer Science  
VI Semester Project 2022

## Bibliography

<https://stackoverflow.com>  
<https://getbootstrap.com>  
<https://www.w3schools.com>