**FOOD MANAGEMENT SYSTEM**

***FindIt – Don’t Waste***

**Submitted in partial fulfillment of the requirements**

**for the award of the degree**

**of**

**BACHELOR OF COMPUTER APPLICATIONS**

**Of**

**Bangalore University**

**By**

**BHAIRESH.M (15BUSB7009)**

****

**May 2018**

**SCHOOL OF GRADUATE STUDIES**

**DEPARTMENT OF COMPUTER APPLICATIONS**

**SSMRV Degree College**

**BENGALURU - 560041**

**Rashtreeya Shikshana Samithi Trust**

**SSMRV Degree College**

4th T Block, Jayanagar, Bengaluru-560041.

****

**Department of Computer Applications**

**CERTIFICATE**

This is to certify that this project entitled “FOOD MANAGEMENT SYSTEM ” is a bonafide work done by BHAIRESH.M (15BUSB7009), submitted to Bangalore University, Bangalore for the award of Bachelor of Computer Applications during the academic year 2017-18.

**Internal Project Guide Head of the Department**

**Mrs.Suprita Nayak Mrs.R. Shanthi Krishna**

**Submitted for the Viva-Voce Examination held on 9-05-18.**

**ACKNOWLEDGEMENT**

We are privileged to express our sincere gratitude to all those who helped us in various capacities in undertaking this project and devising the report.

We the student of SSMRV Degree College with reverence and profound gratitude, place on accord our indebtedness to esteem for having **Dr. S.Anil Kumar, Principal**, for giving us the support to do the Project. We also thank **Mrs.R.Shanthi Krishna,HOD, Department of BCA,** for having granted us the permission to proceed with our project.

We hereby take immense pleasure of thanking **Mrs. Suprita Nayak, Assistant Professor, Department of BCA ,** who presented us with his valuable and guidance during the development of the project work done. We also express our heartfelt thanks for their persistent encouragement throughout the progress of the work. We are greatly obliged to all the assistants for having made all the lab facilities available, to complete our project work on time.

Last but not the least, we thank all other faculties of our department who extended their helping hands at the difficult times.

**Food In Need Donate It – Don’t Waste**

**( FindIt – Don’t waste )**

**CONTENTS**

|  |  |  |
| --- | --- | --- |
| **Chapter** | **Title** | **Page No.** |
|  | **Title Page..** |  |
|  | **Bonafied Certificate** |  |
|  | **Acknowledgement..** |  |
| **1** | **INTRODUCTION** | **2** |
|  | **1.1 About the Project.** | **2** |
|  | **1.2 Existing System And its Limitations..** | **2** |
|  | **1.3 Proposed System.** | **2** |
| **2** | **SYSTEM REQUIREMENTS** | **3** |
|  | **2.1 Hardware Specification.** | **3** |
|  | **2.2 Software Specification.** | **3** |
|  | **2.2.1 Operating System.** | **3** |
|  | **2.2.2 Language / Packages.** | **6** |
| **3** | **SYSTEM DESIGN.** | **10** |
|  | **3.1 Code Design.** | **11** |
|  | **3.2 Input Design.** | **11** |
|  | **3.3 Output Design.** | **12** |
|  | **3.4 Database Design.** | **13** |
|  | **3.4.1 Tables.** | **13** |
|  | **3.4.2 E-R Diagram.** | **15** |
|  | **3.5 Data Flow Diagram.** | **18** |
| **4** | **SYSTEM IMPLEMENTATION AND TESTING.** | **21** |
|  | **4.1 System Implementation.** | **21** |
|  | **4.2 System Testing.** | **21** |
| **5** | **CONCLUSION.** | **24** |
| **6** | **FUTURE ENHANCEMENTS** | **25** |
|  | **BIBLIOGRAPHY.** | **25** |
|  | **APPENDIX-I : SOURCE CODE.** | **25** |
|  | **APPENDIX-II : SNAP SHOTS.** | **61** |

**SYNOPSIS**

**Title of the Project:**

**Food Management System**

**Team members:**

**Bhairesh M(15BUSB7009)**

**Introduction:**

**Main Objective: This project is used to donate Food, Cloth or Fund(Money) to orphanages and mainly the Leftout Food (i.e, Left out food in Hotels and Restuarants) is taken and delivered to the nearest Orphanages , So that the Food will help the people who are in need of it.**

**CHARACTERISTIC OF THE PROPOSED SYSTEM:**

1. **Easy to find Nearest Orphanages.**
2. **Application is user friendly.**
3. **Anyone can donate food, clothes or money which helps the people who are in need.**
4. **Can search the Orphanages in the Bangalore.**

**MODULES:**

1. **MAIN MODEL:**

**This module shows all the options like Donate Food, Cloths or Fund. Login, SignUp etc.**

1. **ORPHANAGES MODEL:**

**In this module we can see the list of registered Orphanages.**

1. **DONATE MODEL:**

**In this module we can Donate Food, Cloths and Money to a particular**

**Orphanages.**

1. **USERS MODEL:**

**This is a module where only admin can see all the Users who are registred and modify data.**

1. **ADMIN PANEL MODEL:**

**This is a admin(only he can) access tables. In this he can modify everything like Orphanages, Users and he can view all the messages sent by users and also reply them.**

**Chapter–1**

**Introduction**

***1.1 About the project***

The main objective of **Food Management System** is to allow the users to Donate Food, Clothes and Money to a particular Orphanage.

Basically **Food Management System** contains three modules they are:

1. Donate: Used to donate left out food, used clothes and ttransfer money.
2. Orphanages: In this anyone can request to add their Orphanage and can list all the registered Orphanages.
3. Admin Panel: This is admin side site where he can Add, Delete And Modify the Orphanages and Users and also he can view the messages sent by users and reply them.

**1.2 Existing System & Its Limitations**

The present system is complex and complicated to the users who use it for the first time. That is when the user is new to the Online Donation, and is required to operate the existing system it will be difficult for them.

* Complex and confusing for freshers.
* Time Consuming.

**1.3 Proposed System**

* Strain of manual labor can be reduced
* Keeps track of products donated.
* Easy to handle

**Chapter-2**

**System Requirement**

**2.1 Hardware Requirements(Min)/Configuration:**

Processor: Dual Core

RAM: 512MB

Hard Disk: 50MB

**2.2 Software Requirements/Configuration:**

Operating System: Windows 10.

Front-End: HTML, CSS, Javascript, Bootstrap.

Back-End: PHP, MYSQLi .

**2.2.1 OPERATING SYSTEM - WINDOWS 10:**

**Windows 10** is a [personal computer](https://en.wikipedia.org/wiki/Personal_computer) [operating system](https://en.wikipedia.org/wiki/Operating_system) developed and released by [Microsoft](https://en.wikipedia.org/wiki/Microsoft) as part of the [Windows NT](https://en.wikipedia.org/wiki/Windows_NT) family of operating systems. It was first released on July 29, 2015. Unlike previous versions of Windows, Microsoft has branded Windows 10 as a "service" that receives ongoing "feature updates". Devices in enterprise environments can receive these updates at a slower pace, or use long-term support milestones that only receive critical updates, such as security patches, over their ten-year lifespan of extended support.

USER INTERFACE AND DESKTOP: A new iteration of the [Start menu](https://en.wikipedia.org/wiki/Start_menu) is used on the Windows 10 desktop, with a list of places and other options on the left side, and tiles representing applications on the right. The menu can be resized, and expanded into a full-screen display, which is the default option in Tablet mode. A new virtual desktop system was added. A feature known as Task View displays all open windows and allows users to switch between them, or switch between multiple workspaces. Windows Store apps, which previously could be used only in full screen mode, can now be used in self-contained windows similarly to other programs. Program windows can now be snapped to quadrants of the screen by dragging them to the corner. When a window is snapped to one side of the screen, Task View appears and the user is prompted to choose a second window to fill the unused side of the screen (called "Snap Assist"). Windows' system icons were also changed.

SYSTEM SECURITY : The enterprise version of Windows 10 offers additional security features; administrators can set up policies for the automatic [encryption](https://en.wikipedia.org/wiki/Data_encryption) of sensitive data, selectively block applications from accessing encrypted data, and enable *Device Guard*‍—‌a system which allows administrators to enforce a high security environment by blocking the execution of software that is not digitally signed by a trusted vendor or Microsoft. Device Guard is designed to protect against [zero-day exploits](https://en.wikipedia.org/wiki/Zero-day_exploit), and runs inside a [hypervisor](https://en.wikipedia.org/wiki/Hypervisor) so that its operation remains separated from the operating system itself.

### STORAGE REQUIREMENTS:

To reduce the storage footprint of the operating system, Windows 10 automatically [compresses](https://en.wikipedia.org/wiki/Data_compression) system files. The system can reduce the storage footprint of Windows by approximately 1.5 GB for [32-bit](https://en.wikipedia.org/wiki/32-bit) systems and 2.6 GB for [64-bit](https://en.wikipedia.org/wiki/64-bit_computing) systems. The level of compression used is dependent on a performance assessment performed during installations or by [OEMs](https://en.wikipedia.org/wiki/Original_equipment_manufacturer), which tests how much compression can be used without harming operating system performance. Furthermore, the Refresh and Reset functions use runtime system files instead, making a separate recovery partition redundant, allowing patches and updates to remain installed following the operation, and further reducing the amount of space required for Windows 10 by up to 12 GB. These functions replace the [WIMBoot mode](https://en.wikipedia.org/wiki/Features_new_to_Windows_8#Installation) introduced on Windows 8.1 Update, which allowed OEMs to configure low-capacity devices with flash-based storage to use Windows system files out of the compressed [WIM image](https://en.wikipedia.org/wiki/Windows_Imaging_Format) typically used for installation and recovery. Windows 10 also includes a function in its Settings app that allows users to view a breakdown of how their device's storage capacity is being used by different types of files, and determine whether certain types of files are saved to internal storage or an [SD card](https://en.wikipedia.org/wiki/SD_card) by default.

SYSTEM REQUIREMENTS: The basic hardware [requirements](https://en.wikipedia.org/wiki/System_requirements) to install Windows 10 are the same as for Windows 8.1 and Windows 8, and only slightly higher than Windows 7. The 64-bit versions require a CPU that supports certain instructions. Devices with low storage capacity must provide a USB flash drive or SD card with sufficient storage for temporary files during upgrades.

**2.2.2 LANGUAGE/PACKAGES**

**HTML: Hyper Text Markup Language**

is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) for creating [web pages](https://en.wikipedia.org/wiki/Web_page) and [web applications](https://en.wikipedia.org/wiki/Web_application). With [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) and [JavaScript](https://en.wikipedia.org/wiki/JavaScript), it forms a triad of cornerstone technologies for the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web).

[Web browsers](https://en.wikipedia.org/wiki/Web_browser) receive HTML documents from a [web server](https://en.wikipedia.org/wiki/Web_server) or from local storage and render the documents into multimedia web pages. HTML describes the structure of a web page [semantically](https://en.wikipedia.org/wiki/Semantic_Web) and originally included cues for the appearance of the document.

[HTML elements](https://en.wikipedia.org/wiki/HTML_element) are the building blocks of HTML pages. With HTML constructs, [images](https://en.wikipedia.org/wiki/HTML_element#Images_and_objects) and other objects such as [interactive forms](https://en.wikipedia.org/wiki/Fieldset)may be embedded into the rendered page. HTML provides a means to create [structured documents](https://en.wikipedia.org/wiki/Structured_document) by denoting structural [semantics](https://en.wikipedia.org/wiki/Semantics" \o "Semantics)for text such as headings, paragraphs, lists, [links](https://en.wikipedia.org/wiki/Hyperlink), quotes and other items. HTML elements are delineated by *tags*, written using [angle brackets](https://en.wikipedia.org/wiki/Bracket#Angle_brackets). Tags such as <**img** /> and <**input** /> directly introduce content into the page. Other tags such as <**p**> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

HTML can embed programs written in a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript), which affects the behavior and content of web pages. Inclusion of CSS defines the look and layout of content. The [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

**CSS: Cascading Style Sheets**

is a [style sheet language](https://en.wikipedia.org/wiki/Style_sheet_language) used for describing the [presentation](https://en.wikipedia.org/wiki/Presentation_semantics) of a document written in a [markup language](https://en.wikipedia.org/wiki/Markup_language) like [HTML](https://en.wikipedia.org/wiki/HTML). CSS is a cornerstone technology of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web), alongside HTML and [JavaScript](https://en.wikipedia.org/wiki/JavaScript).

CSS is designed to enable the separation of presentation and content, including [layout](https://en.wikipedia.org/wiki/Page_layout), [colors](https://en.wikipedia.org/wiki/Color), and [fonts](https://en.wikipedia.org/wiki/Typeface). This separation can improve content [accessibility](https://en.wikipedia.org/wiki/Accessibility), provide more flexibility and control in the specification of presentation characteristics, enable multiple [web pages](https://en.wikipedia.org/wiki/Web_page) to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

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

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

The CSS specifications are maintained by the [World Wide Web Consortium](https://en.wikipedia.org/wiki/World_Wide_Web_Consortium) (W3C). Internet media type ([MIME type](https://en.wikipedia.org/wiki/MIME_media_type)) text/css is registered for use with CSS by [RFC 2318](https://tools.ietf.org/html/rfc2318) (March 1998). The W3C operates a free [CSS validation service](https://en.wikipedia.org/wiki/W3C_Markup_Validation_Service#CSS_validation) for CSS documents.

**JavaScript**

 Is often abbreviated as **JS**, is a [high-level](https://en.wikipedia.org/wiki/High-level_programming_language), [interpreted](https://en.wikipedia.org/wiki/Interpreted_language) [programming language](https://en.wikipedia.org/wiki/Programming_language). It is a language which is also characterized as [dynamic](https://en.wikipedia.org/wiki/Dynamic_programming_language), [weakly typed](https://en.wikipedia.org/wiki/Weak_typing), [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming) and [multi-paradigm](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language).

Alongside [HTML](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/CSS), JavaScript is one of the three core technologies of the [World Wide Web](https://en.wikipedia.org/wiki/World_Wide_Web). JavaScript enables interactive [web pages](https://en.wikipedia.org/wiki/Web_page) and thus is an essential part of [web applications](https://en.wikipedia.org/wiki/Web_application). The vast majority of [websites](https://en.wikipedia.org/wiki/Website) use it, and all major [web browsers](https://en.wikipedia.org/wiki/Web_browser) have a dedicated [JavaScript engine](https://en.wikipedia.org/wiki/JavaScript_engine) to execute it.

As a multi-paradigm language, JavaScript supports [event-driven](https://en.wikipedia.org/wiki/Event-driven_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming), and [imperative](https://en.wikipedia.org/wiki/Imperative_programming) (including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [prototype-based](https://en.wikipedia.org/wiki/Prototype-based_programming)) [programming styles](https://en.wikipedia.org/wiki/Programming_paradigm). It has an [API](https://en.wikipedia.org/wiki/Application_programming_interface) for working with text, [arrays](https://en.wikipedia.org/wiki/Array_data_type), dates, [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), and basic manipulation of the [DOM](https://en.wikipedia.org/wiki/Document_Object_Model), but the language itself does not include any [I/O](https://en.wikipedia.org/wiki/Input/output), such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded.

Initially only implemented [client-side](https://en.wikipedia.org/wiki/Client-side) in web browsers, JavaScript engines are now embedded in many other types of host software, including [server-side](https://en.wikipedia.org/wiki/Server-side) in web servers and databases, and in non-web programs such as word processors and [PDF](https://en.wikipedia.org/wiki/Portable_Document_Format) software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets.

Although there are strong outward similarities between JavaScript and [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), including language name, [syntax](https://en.wikipedia.org/wiki/Syntax_(programming_languages)), and respective [standard libraries](https://en.wikipedia.org/wiki/Standard_library), the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as [Self](https://en.wikipedia.org/wiki/Self_(programming_language)) and [Scheme](https://en.wikipedia.org/wiki/Scheme_(programming_language)).

**PHP: Hypertext Preprocessor** (or simply **PHP**)

is a [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting) language designed for [web development](https://en.wikipedia.org/wiki/Web_development) but also used as a [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). It was originally created by [Rasmus Lerdorf](https://en.wikipedia.org/wiki/Rasmus_Lerdorf" \o "Rasmus Lerdorf) in 1994, the PHP [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) is now produced by The PHP Group. PHP originally stood for *Personal Home Page*,[[3]](https://en.wikipedia.org/wiki/PHP" \l "cite_note-History_of_PHP-3) but it now stands for the[recursive acronym](https://en.wikipedia.org/wiki/Recursive_acronym) PHP*: Hypertext Preprocessor*.

PHP code may be embedded into [HTML](https://en.wikipedia.org/wiki/HTML) code, or it can be used in combination with various [web template systems](https://en.wikipedia.org/wiki/Web_template_system), web content management systems, and [web frameworks](https://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) implemented as a [module](https://en.wikipedia.org/wiki/Plugin_(computing)) in the web server or as a [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface) (CLI) and can be used to implement [standalone](https://en.wikipedia.org/wiki/Computer_software) [graphical applications](https://en.wikipedia.org/wiki/Graphical_user_interface).

The standard PHP interpreter, powered by the [Zend Engine](https://en.wikipedia.org/wiki/Zend_Engine), is [free software](https://en.wikipedia.org/wiki/Free_software) released under the [PHP License](https://en.wikipedia.org/wiki/PHP_License). PHP has been widely ported and can be deployed on most web servers on almost every [operating system](https://en.wikipedia.org/wiki/Operating_system) and [platform](https://en.wikipedia.org/wiki/Computing_platform), free of charge.

The PHP language evolved without a written [formal specification](https://en.wikipedia.org/wiki/Formal_specification) or standard until 2014, leaving the canonical PHP interpreter as a [*de facto*](https://en.wikipedia.org/wiki/De_facto) standard. Since 2014 work has gone on to create a formal PHP specification.

**Chapter 3 SYSTEM DESIGN**

The most creative and challenging face of the web development is Web Design. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Design goes through the logical and physical stages of development.

In designing a new system, the system analyst must have a clear understanding of the objectives, which the design is aiming to fulfill. The first step is to determine how the output is to be produced and in what format. Second, input data and master files have to be designed to meet the requirements of the proposed output. The operational phases are handled through program construction and testing.

Design of a system can be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Thus system design is a solution to “how to” approach to the creation of a new system. Thus important phase provides the understanding and the procedural details necessary for implementing the system recommended in the feasibility study. The design step provides a data design, architectural design, and a procedural design.

**3.1 CODE DESIGN**

First phase of implementation is coding. Coding can be done in two ways. One by automatic program code and other by programmer’s manually written code. A code generator is a suite of programs that matches the input to an appropriate code template and from these produces modules of code. The code is made simple in such a way that another programmer can easily understand and work on that in future. The crucial phase in the system development life cycle is the successful implementation of the new system design. The process of converting as new or revised system into an operational one is known as system implementation. This includes all those activities that take place to convert from an old system to a new system. The system can be implemented only after a through testing is done and if it is found to work according to the specifications. The most crucial stage in achieving a new successful system and giving confident on the new system for the users is that it will work effectively and efficiently. It involves careful planning, investigation of the current system and its constraints on implementation, design of methods to achieve the changeover.

**3.2 INPUT DESIGN**

In the input design, user-oriented inputs are converted into a computer based system format. It also includes determining the record media, method of input, speed of capture and entry on to the screen. Online data entry accepts commands and data through a keyboard. The major approach to input design is the menu and the prompt design. In each alternative, the user’s options are predefined. The data flow diagram indicates logical data flow, data stores, source and destination. Input data are collected and organized into a group of similar data. Once identified input media are selected for processing.

In this software, importance is given to develop Graphical User Interface (GUI), which is an important factor in developing efficient and user-friendly software. For inputting user data, attractive forms are designed. User can also select desired options from the menu, which provides all possible facilities.

Also the important input format is designed in such a way that accidental errors are avoided. The user has to input only just the minimum data required, which also helps in avoiding the errors that the users may make. Accurate designing of the input format is very important in developing efficient software. The goal or input design is to make entry as easy, logical and free from errors.

**3.3 OUTPUT DESIGN**

In the output design, the emphasis is on producing a hard copy of the information requested or displaying the output on the CRT screen in a predetermined format. Two of the most output media today are printers and the screen. Most users now access their reports from a hard copy or screen display. Computer’s output is the most important and direct source of information to the user, efficient, logical, output design should improve the systems relations with the user and help in decision-making.

As the outputs are the most important source of information to the user, better design should improve the system’s relation and also should help in decision-making. The output device’s capability, print capability, print capability, response time requirements etc should also be considered form design elaborates the way output is presented and layout available for capturing information. It’s very helpful to produce the clear, accurate and speedy information for end users.

**3.4 DATABASE DESIGN**

Database design is about the logically implied data. Each and every data in the form can be designed in such a manner to understand the meaning. Database designing should give a clear understanding and idea about the related data used to construct a form.

**3.4.1 TABLES**

**1: orphanageslist**

|  |  |
| --- | --- |
| Field name | Data type |
| id | Int(20) |
| Name | Varchar(64) |
| Email | Varchar(20) |
| Phone | Int(20) |
| Address | Varchar(128) |

**2: users**

|  |  |
| --- | --- |
| Field name | Data type |
| id | Int(20) |
| Name | Varchar(32) |
| Password | Varchar(6) |
| Phone | Int(20) |
| Email | Varchar(20) |
| Address | Varchar(128) |

* + 1. **E-R Diagrams**

E-R-Diagram is a diagrammatic representation of the logical design of the database. In E-R diagram entities, attributes and relationship is represented.

**Entity:** An entity is a real time object. In databases the entity refers to the tables.

**Attribute:** An attribute is an characteristic representation of a column in a table as known as Fields.

**Relationship:** It is the association between one entity to another.

Different symbols used in drawing E-R Diagrams are:

FindIt

Has

Has a list of

Has

Donate

Orphanages

Admin panel

o.

Donate

Orphanages List

Transfer

Food &Cloths

Fund

Orphanages List

Orphanages

Request to addOrphanages

has

has

**3.5 DATA FLOW DIAGRAM**

A Data Flow Diagram (DFD) is a diagram that describes the flow of data and the processes that change data throughout a system. It’s a structured analysis and design tool that can be used for flowcharting in place of or in association with information. Oriented and process oriented system flowcharts. When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply physical implementations. The Data Flow Diagram reviews the current physical system, prepares input and output specification,

specifies the implementation plan etc.

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called node.

**Context Level DFD**

Donor

Donate

0

Food Management system

**Context Level 1**

1.0

Donate

2.0

Food

3.0

Clothes

4.0

Money

D2 Fund Transfer

D1 Delivery or Call

Donor

Delivered to Orphanage

**Context Level2 DFD**

3.0

Admin Panel

3.1

Add User/Orphanage

3.2

Delete User/Orphanage

3.3

Update User/Orphanage

Admin

3.4

View&Reply User Queries

**Chapter 4**

**SYSTEM IMPLEMENTATION AND TESTING**

**4.1 SYSTEM IMPLEMENTATION**

Implementation is the stage, which is crucial in the life cycle of the new system designed. Planning, training and system testing are the main stages in the implementation. Converting a new or revised system into an operational one is called implementation.

Implementation includes all those activities involving the conversion of an old system into a new system. The new system may be in a totally new concept or a revision of the old one. A proper implementation is required for reliable system, but still does not guarantee a successful system. Chances are there that if implementation is not proper the whole system may become a failure.

Conversion an important aspect of implementation is the process of change from the old system to the new one. A review is conducted once implementations are over. Information required for maintenance is collected during this phase. The basic review method is data collection methods of interview, observation, sampling and record inspection.

**4.2 SYSTEM TESTING**

Testing is a process to show the correctness of the program. Testing is needed to show completeness, it improve the quality of the software and to provide the maintenance aid. Some testing standards are therefore necessary reduce the testing costs and operation time. Testing software extends throughout the coding phase and it represents the ultimate review of configurations, design and coding. Based on the way the software reacts to these testing, we can decide whether the configuration that has been built is study or not. All components of an application are tested, as the failure to do so many results in a series of bugs after the software is put to use.

**Black box Testing:**

Black box testing, also called behavioral testing, focuses on the functional requirements of software. This testing approach enables the software engineer to derive the input conditions that will fully exercise all requirements for a program. Blackbox testing attempts to find the errors like

* Incorrect or missing functions
* Interface errors
* Errors in data structures or external database access
* Behavior or performance errors
* Initialization and termination errors

In Black box testing software is exercised over a full range of inputs and outputs are observed for correctness.

**White box Testing:**

White box testing is also called Glass box testing is a test case design control; structure of the procedural design to derive test cases using White box testing method, the software engineer can derive the test cases that guarantee that all independent paths within the module have been exercised at least once. Exercise all logic decisions on their true or false sides. Execute all loops at their boundaries and within their operational bounds. Exercise internal data structure to ensure their validity.

**Software Testing Strategies:**

Testing involves

* Unit testing
* Integration testing
* Acceptance testing

**UNIT TESTING:**

The first level of test is unit testing. The purpose of unit testing is to ensure that each program is fully tested.

Unit testing focuses verification effort on the smallest unit of software design – the module. The unit test is always white box oriented. The tests that occur as part of unit testing are testing the module interface, examining the local data structures, testing the boundary conditions, executing all the independent paths and testing error-handling paths.

**INTEGRATION TESTING:**

The second step is integration testing. In this individual program units or programs are integrated and tested as a complete system to ensure that the software requirements are met.

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. Scope of testing summarizes the specific functional, performance, and internal design characteristics that are to be tested. It employs top-down testing and bottom-up testing methods for this case.

**ACCEPTANCE TESTING:**

This is a system testing done by the user of the applications. The only emphases is functionality testing as the user is not aware of the technical aspect of the system. The testing is also done in a controlled environmental with logging off all error based on the error found in the system, the user has to accept or reject the system.

**5. CONCLUSION**

This project entitled “Billing and Invoice Management” has been developed efficiently to meet the needs of the users. It is developed using VB 2012and MYSQL SERVER 5.7. The system can be easily modified or expanded. The software provides all facilities required in the finished system, and the reports are user-friendly. The software requires less effort to operate. The assistance of computers will reduce the work done by the staff. The system provides flexibility for incorporating new features, which may be necessary in future. The system has been tested and implemented. Finally, we have concluded that system has capacity to do what it was expected to do.

**6. SCOPE OF THE FURTHER DEVELOPMENT**

Now the organization staffs can bill the purchases very easily. The software is very easy to access. This project can further be implemented as the online system so that the billing can be done online.

The application can also have user accounts, that is, the customers can also be given the access to their particular account to view the products, their prices and bill accordingly.

**BIBILIOGRAPHY**

* Stackoverflow
* DreamInCode
* CodeProject

**Source Code:**

**Index page:**

<!DOCTYPE HTML>

<html>

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

<link rel="stylesheet" href="/FindIt-Don't Waste/Docs/bootstrap-3.3.7-dist/css/bootstrap.css">

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

<script type="text/javascript" src="Javascript/addOrphanage.js"></script>

<script type="text/javascript" src="Javascript/register.js"></script>

<link rel="stylesheet" href="CSS/loginPopup.css">

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

<link rel="stylesheet" href="CSS/w3(3).css">

<link rel="stylesheet" href="CSS/w3(4).css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<title>FindIt-Don't Waste</title>

<head>

<style type="text/css">

a:link, a:visited {

color: (internal value);

text-decoration: none;

}

a:link:active, a:visited:active {

color: white;

}

.contact input{

width:38%;

height:40px;

padding: 0px 5%;

margin-bottom: 4px;

border:none;

line-height: 40px;

background-color: #fff;

}

</style>

</head>

<!-- Body Starts Here -->

<body id="body" style="background-color: #f3f3f3;">

<!-- Navbar (sit on top) -->

<div class="">

<div class="w3-bar w3-white w3-padding w3-card" style="letter-spacing:4px;position: relative;background-color: #00b894;">

<a href="#home" class="w3-bar-item w3-button">FindIt</a>

<div class="w3-right">

<a href="#donate" class="w3-bar-item w3-button">Donate</a>

<a href="donateSomething.php" class="w3-bar-item w3-button">Orphanages</a>

<!-- <a href="#about" class="w3-bar-item w3-button">About</a>-->

<a href="#contact" class="w3-bar-item w3-button">Contact</a>

<span style=""><img src="Images/search.png" style="height: 30px;width: 43px;margin:0px 0px 1px 10px;" onclick="search.php"></span> <input type="combo" onclick ="search.php" name="search" placeholder="Search Orphanages.." style="border: none;">

<span style=""><img src="Images/login-face.png" style="height: 30px;width: 30px;margin:4px 1px 1px 10px;" onclick="document.getElementById('login').style.display='block'"></span>

</div>

</div>

</div>

<div class="mainPage" id="mainPage">

<!-- Log in -->

<div id="login" class="modal" style="text-align: center;background-color: #f3f3f3;">

<form class="" action="login.php" method="post">

<span onclick="document.getElementById('login').style.display='none'" class="close" title="Close Modal">&times;</span>

<div class="container">

<input type="text" placeholder="Enter User ID" name="uid" required style="width:50%;height:40px;padding: 0px 5%;margin-bottom: 4px;border:none;line-height: 40px;background-color: #fff;"><br><br>

<input type="password" placeholder="Enter Password" name="password" required style="width:50%;height:40px;padding: 0px 5%;margin-bottom: 4px;border:none;line-height: 40px;background-color: #fff;"><br><br>

<button type="submit" style="background-color: #0984e3;border-radius: 5px;">Login</button>

</div>

<span><a href="#">Forgot password..?</a></span>

<span style="margin-left: 20px;"><a href="signUp.php">Sign up</a></span>

</form>

</div>

<!-- Sign up -->

<div style="background-image: url(Imagesmainpage2.jpg);width: 100%;height:700px;background-size:100% 100%;background-repeat: no-repeat;">

<div id="donate" style="margin-top:0px;">

<br><br><br>

<div style="border-style: groove;border:none; width:70%;height: 150px;text-align: center;margin:0px 0px 15px 15%;background-color: #C4E538;border-radius:10px;" class="food" name="food" onclick="">

<p style="padding-top: 5px;margin: 0px 5px 0px 5px;"><a href="donateSomething.php"> <h1>Donate Food</h1>

<p>Kindness in giving creates love. It is one of the beautiful compensations of life that no man can sincerely try to help another without helping himself. I shall pass through this world but once. Any good, therefore,that I can do or any kindness that I can show to any fellow creature, let me do it now.</p></a></p>

</div>

<div style="border-style: groove;border:none;width:70%;height: 150px;text-align: center;margin:10px 0px 15px 15%;background-color: #A3CB38;border-radius:10px;" class="cloths" name="cloths">

<p style="padding-top: 10px;"><a href=""><h1>Donate Clothes</h1>

<p>The measure of life is not its duration, but its donation.</p></a></p>

</div>

<div style="border-style: groove;border:none;width:70%;height: 150px;text-align: center;margin:0px 0px 0px 15%;background-color: #009432;border-radius:10px;" name="fund" class="fund">

<p style="padding-top: 10px;"><a href=""><h1 style="">Donate Fund</h1>

<p>A man's true wealth is the good he does in this world.</p></a></p>

<!--<button >Account Details..</button>-->

</div>

</div>

</div>

<br>

<hr style="border-style: groove;width:70%;text-align: center;margin-left: 15%;">

<div style="margin: 5%">

<h5 style="margin-left: 15%;">FindIt's Food and Organic Waste Recycling helps you:</h5>

<p style="margin-left: 17%;">Raise your recycling rates<br>

Increase diversion rates<br>

Choose a cost-effective disposal option<br>

Foster a green image among your employees and customers<br>

Success made simple.

</p><br>

<h5 style="margin-left:15%;">Acceptable items for Food and Organic Waste Recycling can include:</h5>

<p style="margin-left: 17%;">

<ul style="margin-left: 15% " type="dot">

<li>Fruits and vegetables</li>

<li>Meat, poultry, seafood (bones and shells)</li>

<li>Bakery items and ingredients</li>

<li>Eggs and paper egg cartons</li>

<li>Plants, cut flowers, potting soil</li>

<li>Coffee grounds, filters, tea bags</li>

<li>Paper products (napkins, paper towels)</li>

<li>Ice cream, yogurt, cottage cheese</li>

</ul>

</p>

<h4 style="margin-left:15%;">Reuse (Donation)</h4>

<p style="margin-left:17%;margin-right: 15 %;">Instructions: Food shows, catered events, and similar special events generate significant amounts of usable food and other items (napkins, plants, decorations, etc.) that can be donated to food rescue organizations and others. Arrange with local organizations in advance of the event. Describe how vendors will be informed about the food donation opportunity, acceptable items, etc. Talk with vendors at food shows and similar events about their reuse practices; these might present a model for other vendors.</p>

<h4 style="margin-left:15%;">WASTE COMPOSITION, GENERATION, AND QUANTITIES</h4>

<p style="margin-left:17%;margin-right: 15%;">Instructions: Conduct a waste assessment using a vendor food waste assessment survey to track waste generation at the event. As part of the waste assessment, ask vendors and event organizers (or staff) questions about specific waste items, clarification of reuse, recycling, and disposal practices, and potential options for reuse and/or substitution to recyclable or compostable products. Talk with the vendors about composting and the potential implementation of composting at the event. Use the chart below to summarize the types of wastes generated.</p>

</div>

<hr style="border-style: groove;width:70%;text-align: center;margin-left: 15%;">

<!-- Contact starts Here -->

<div style="background-color:;width: 100%;height:450px;background-size:100% 100%;background-repeat: no-repeat;">

<div id="contact" style="text-align: center;width: 50%;float: right;">

<div class="w3-content w3-text-grey" id="contact" style=" ">

<h2 class="">Contact Me</h2>

<hr style="width:200px" class="w3-opacity">

<p>Send me a message:</p>

<form action="<?php $\_PHP\_SELF ?>" class="contact" method="POST" >

<p><input class="" type="text" placeholder="Name" required name="Name" style=""></p>

<p><input class="" type="text" placeholder="Email" required name="Email"></p>

<p><input class="" type="text" placeholder="Phone" required name="Phone"></p>

<p><input class="" type="text" placeholder="Message" required name="Message"></p>

<p><button class="w3-button w3-light-grey w3-padding-large" type="submit" name="submit" id="sendMsg"><i class="fa fa-paper-plane"></i>&nbsp SEND MESSAGE</button></p>

</form>

</div>

</div>

</div>

<!-- End Contact Section -->

</div>

</body>

<!-- Body Ends Here -->

<!-- Footer -->

<footer class="w3-center w3-light w3-padding-32" style="background-color: #05386B;">

<p></p>

<i class="fa fa-facebook-official w3-hover-opacity" style="color: white;"></i>&nbsp

<i class="fa fa-instagram w3-hover-opacity" style="color: white;"></i>&nbsp

<i class="fa fa-twitter w3-hover-opacity" style="color: white;"></i>&nbsp

<i class="fa fa-linkedin w3-hover-opacity" style="color: white;"></i>

<p style="text-align:center;color: white;">&copy 2018 Copyright</p>

</footer>

</html>

**Admin Page:**

<!DOCTYPE html>

<html>

<head>

<title>Admin Page</title>

<style type="text/css">

ul {

list-style-type: none;

margin: 0;

padding: 0;

overflow: hidden;

background-color: #333;

}

li {

float: left;

}

li a {

display: block;

color: white;

text-align: center;

padding: 14px 16px;

text-decoration: none;

}

li a:hover {

background-color: #111;

}

.btn {

background: #3498db;

background-image: -webkit-linear-gradient(top, #3498db, #2980b9);

background-image: -moz-linear-gradient(top, #3498db, #2980b9);

background-image: -ms-linear-gradient(top, #3498db, #2980b9);

background-image: -o-linear-gradient(top, #3498db, #2980b9);

background-image: linear-gradient(to bottom, #3498db, #2980b9);

-webkit-border-radius: 5;

-moz-border-radius: 5;

border-radius: 5px;

font-family: Courier New;

color: #ffffff;

font-size: 17px;

padding: 10px 20px 10px 20px;

text-decoration: none;

width: 300px;

height: 150px;

float: left;

}

.btn:hover {

background: #0c4e7a;

text-decoration: none;

}

</style>

</head>

<body>

<nav>

<ul>

<li><a class="active" href="#home">Admin Panel</a></li>

<li style="float: right;"><a href="LogOut.php">Logout</a></li>

<li style="float: right;"><a href="displayUsers.php">Users</a></li>

<li style="float: right;"><a href="displayOrphanages.php" >Orphanages</a></li>

<li style="float: right;"><a href="displayQueries.php">Queries</a></li>

</ul>

</nav>

<div class="home" style="text-align: center;">

<div class="btns" style="max-width: 50%;float: left;margin: 5% 0% 10% 20%;"><a href="displayUsers.php"><button class="btn">Users</button></a></div>

<div class="btns" style="max-width: 50%;float: left;margin: 5% 10% 10% 20%;"><a href="displayOrphanages.php"><button class="btn">Orphanages</button></a></div>

<div class="btns" style="max-width: 50%;float: left;margin: 0% 0% 10% 20%;"><a href="displayQueries.php"><button class="btn">Queries</button></a></div>

<div class="btns" style="max-width: 50%;float: left;margin: 0% 10% 10% 20%;"><a href=""><button class="btn">Something</button></a></div>

</div>

</body>

</html>

**donateSomething page:**

<?php

$conn = mysqli\_connect('localhost','root','','finditdb');

if (!$conn) {

die ('Failed to connect to MySQL: ' . mysqli\_connect\_error());

}

else{

$\_PHP\_SELF ;

}

$sql = 'SELECT \* FROM orphanageslist';

$query = mysqli\_query($conn, $sql);

if (!$query) {

die ('SQL Error: ' . mysqli\_error($conn));

}

?>

<!DOCTYPE html>

<html>

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

<link rel="stylesheet" href="/FindIt-Don't Waste/Docs/bootstrap-3.3.7-dist/css/bootstrap.css">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

<script src="/FindIt-Don't Waste/Javascript/script.js"></script>

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

<script type="text/javascript" src="Javascript/addOrphanage.js"></script>

<link rel="stylesheet" href="CSS/loginPopup.css">

<script type="text/javascript" src="Javascript/loginPopup.js"></script>

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

<link rel="stylesheet" href="CSS/w3(3).css">

<link rel="stylesheet" href="CSS/w3(4).css">

<title>FindIt-Don't Waste</title>

<head>

</head>

<!-- Body Starts Here -->

<body style="" id="body">

<!-- Navbar (sit on top) -->

<div class="" style="">

<div class="w3-bar w3-white w3-padding w3-card" style="letter-spacing:4px;position: relative;">

<a href="index.html.#home" class="w3-bar-item w3-button">FindIt</a>

<div class="w3-right">

<a href="index.html.#donate" class="w3-bar-item w3-button">Donate</a>

<a href="donateSomething.html" class="w3-bar-item w3-button">Orphanages</a>

<!-- <a href="#about" class="w3-bar-item w3-button">About</a>-->

<a href="index.html.#contact" class="w3-bar-item w3-button">Contact</a>

<a href="#" class=""><span style=""><img src="Images/search.png" style="height: 30px;width: 43px;margin:0px 0px 1px 10px;" onclick=""></span></a> <input type="text" name="search" placeholder="Search orphanages.." style="border: none;">

<a href="#" class=""><span style=""><img src="Images/login-face.png" style="height: 30px;width: 30px;margin:4px 1px 1px 10px;" onclick="display()"></span></a>

</div>

</div>

</div>

<button style="margin:10px 25px 0px 0px;float: right;border-radius: 4px;" onclick="addOrphanage()">Request to Add Orphanage</button>

<div>

<!-- Nearest Orphanages Block -->

<div id="orphanages" style="margin:5% 5% 5% 5%;text-align: center;top:">

<h2>Orphanages</h2><br><br>

<div style="width:100%;">

<?php

while ($row = mysqli\_fetch\_array($query))

{

?><div id="list" style="background-color: ;border-style: groove;width: 70%;max-height: 250px;float: left;margin:5px 5px 5px 15%;border-radius: 10px;"><?php

echo '

<h3>Name: &nbsp '.$row['Name'].'</h3>

<h3>Email: &nbsp '.$row['Email'].'</h3>

<h3>Website: &nbsp '.$row['Website'].'</h3>

<h3>Address: &nbsp '.$row['Address'].'</h3>

';

?>

<a><i class="fa fa-phone fa-3x" aria-hidden="true" onclick="displayContact()"></i></a> &nbsp &nbsp &nbsp

<a href="delivery.php"><i class="fa fa-truck fa-3x" aria-hidden="true" onclick=""></i></a>

</div>

<?php

}

?>

</div>

<div id="myModal1" class="modal">

<!-- Modal content orphanage button -->

<div class="modal-content">

<span class="close" onclick="closePopup1()">&times;</span>

<form action="addOrphanage.php" method="post"></br>

<p>Request to Add your Orphanage and get served</p>

<input type="text" name="Name" placeholder="Orphanage Name" required></br></br>

<input type="text" name="Email" placeholder="Email" required></br></br>

<input type="text" name="Phone" placeholder="Phone number" required></br></br>

<input type="text" name="Address" placeholder="Address..." required></br></br>

<input type="submit" value="Send Request" style="text-align: center;width: auto;"></br></br>

</form>

</div>

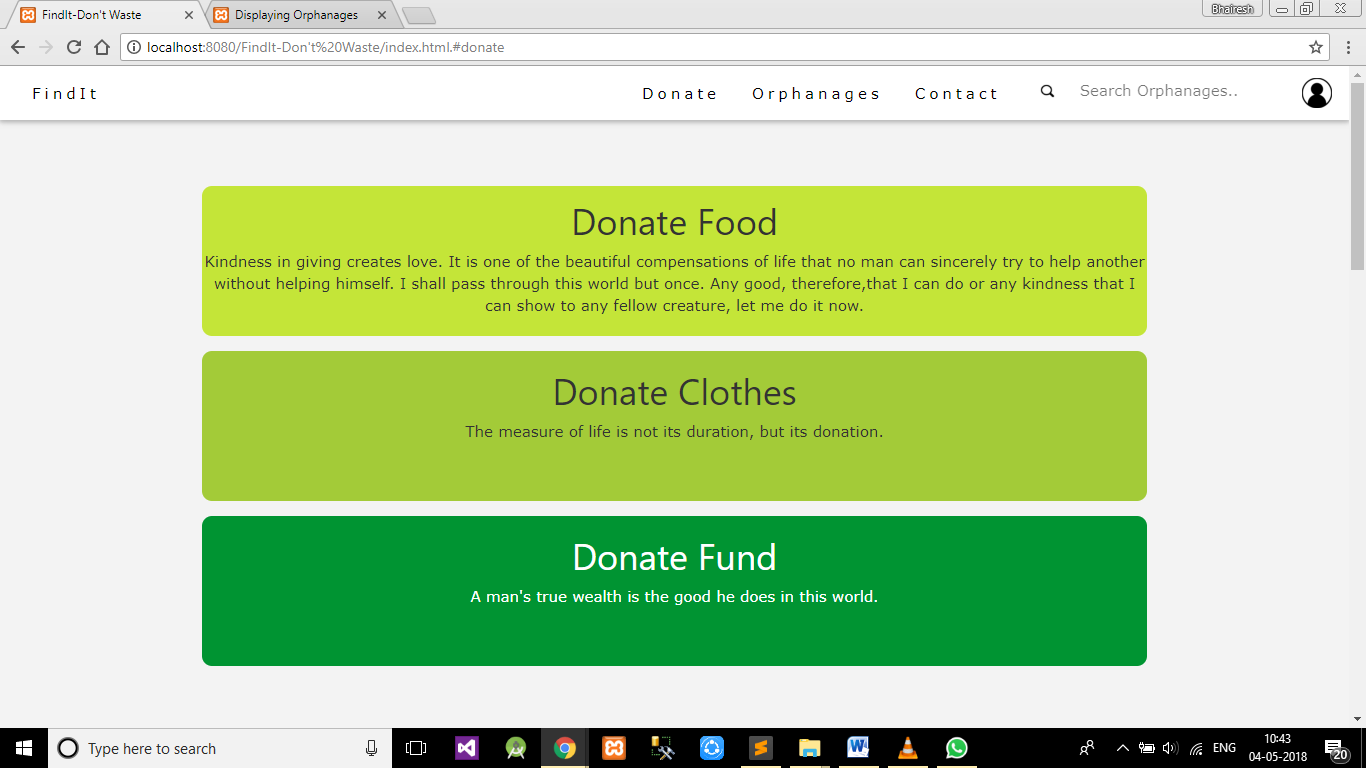
</div>

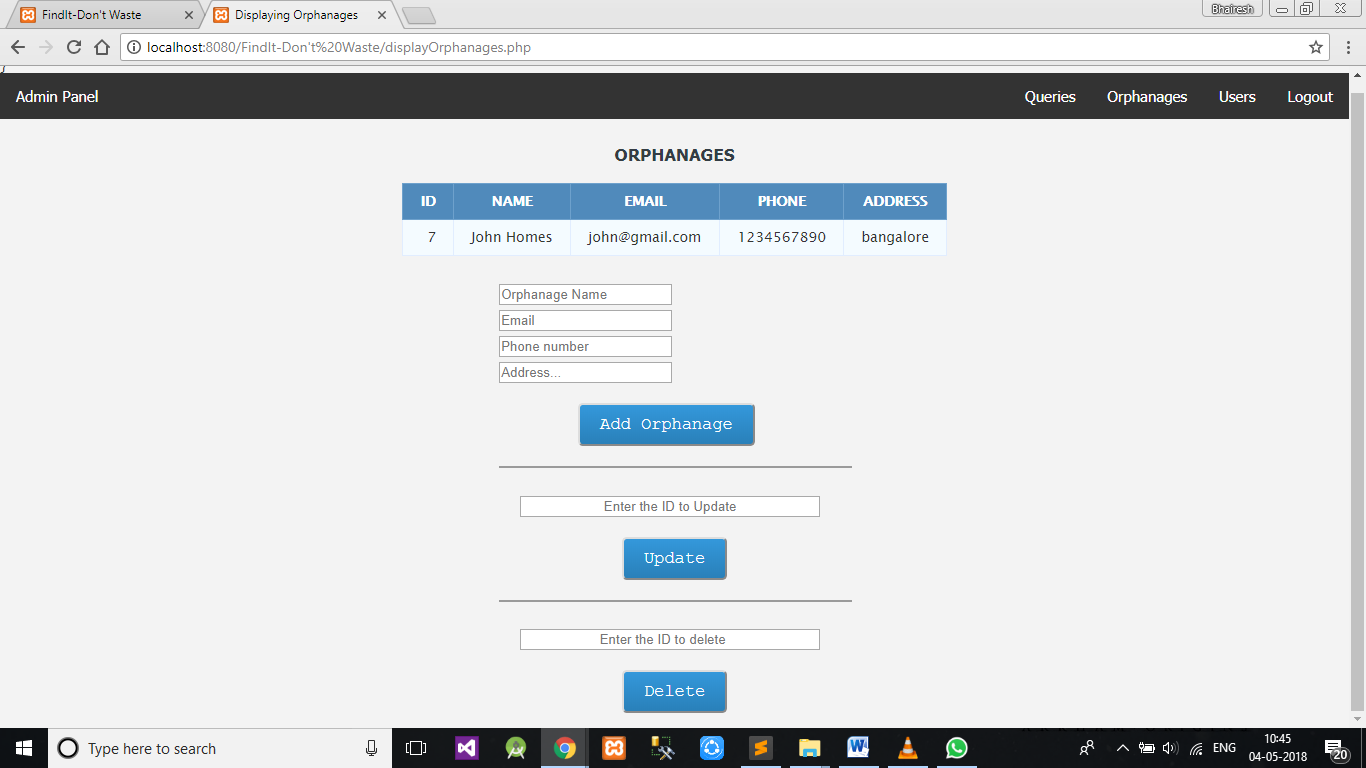
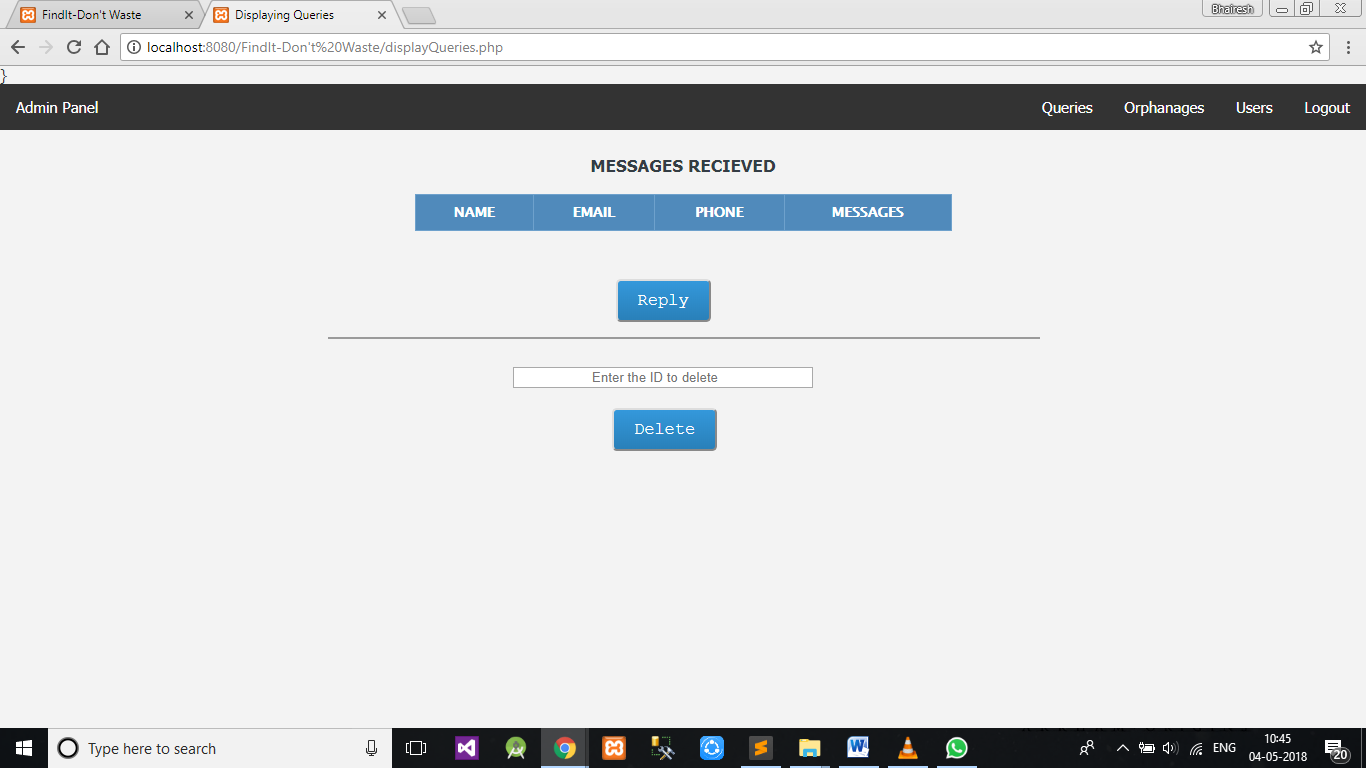
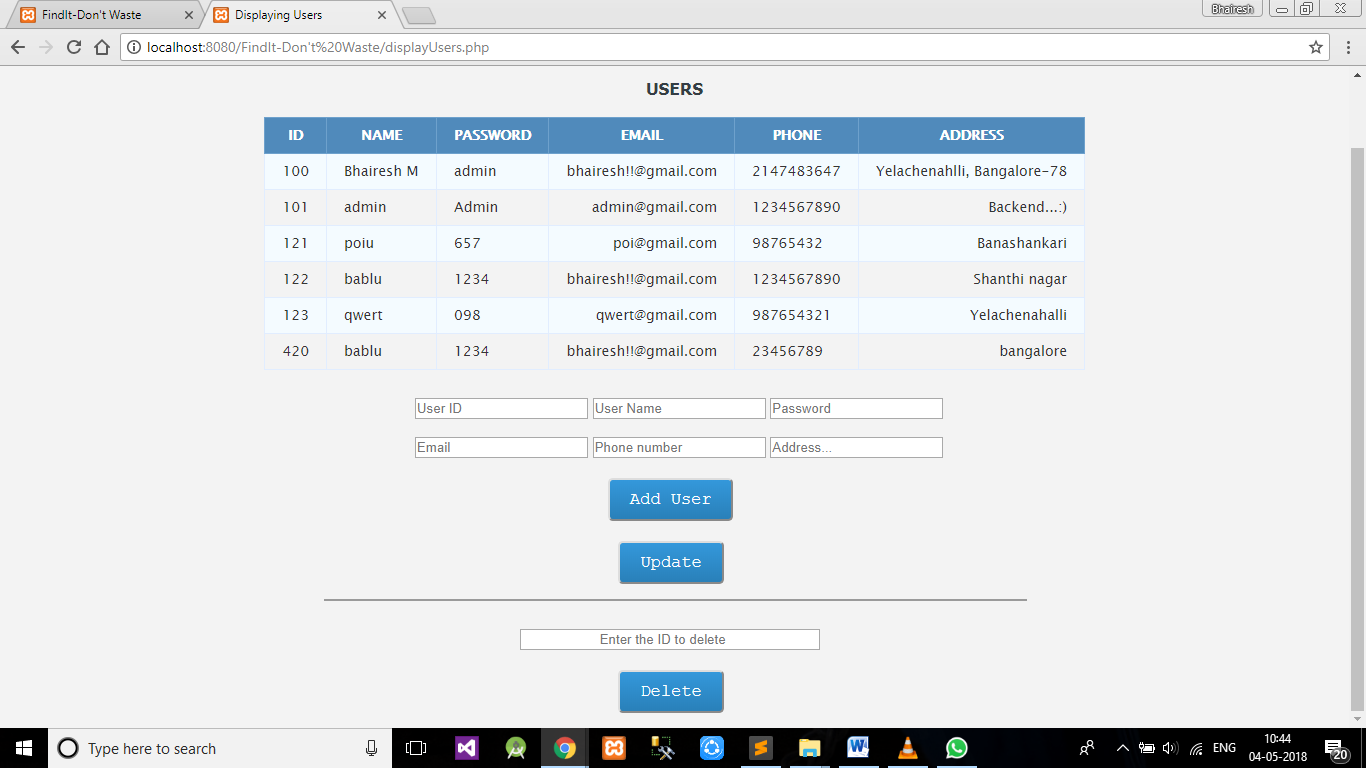
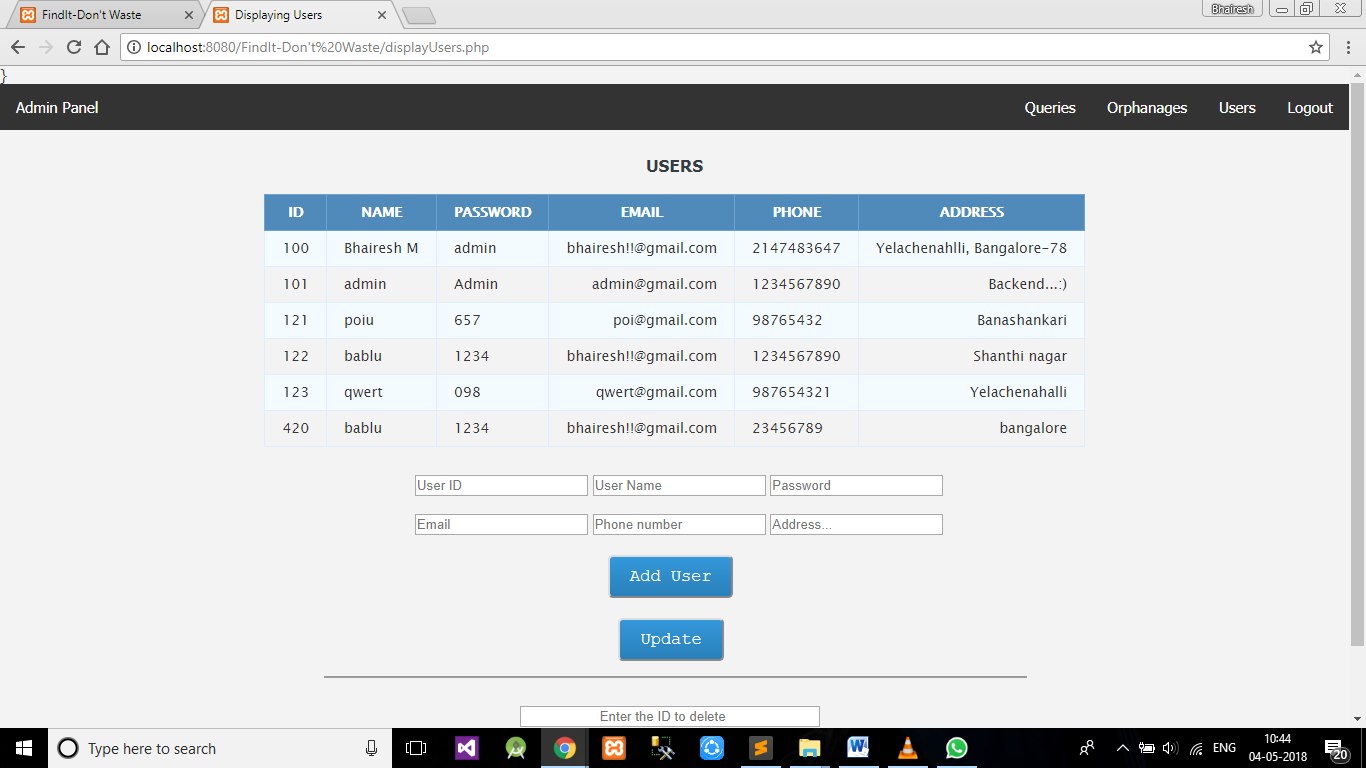
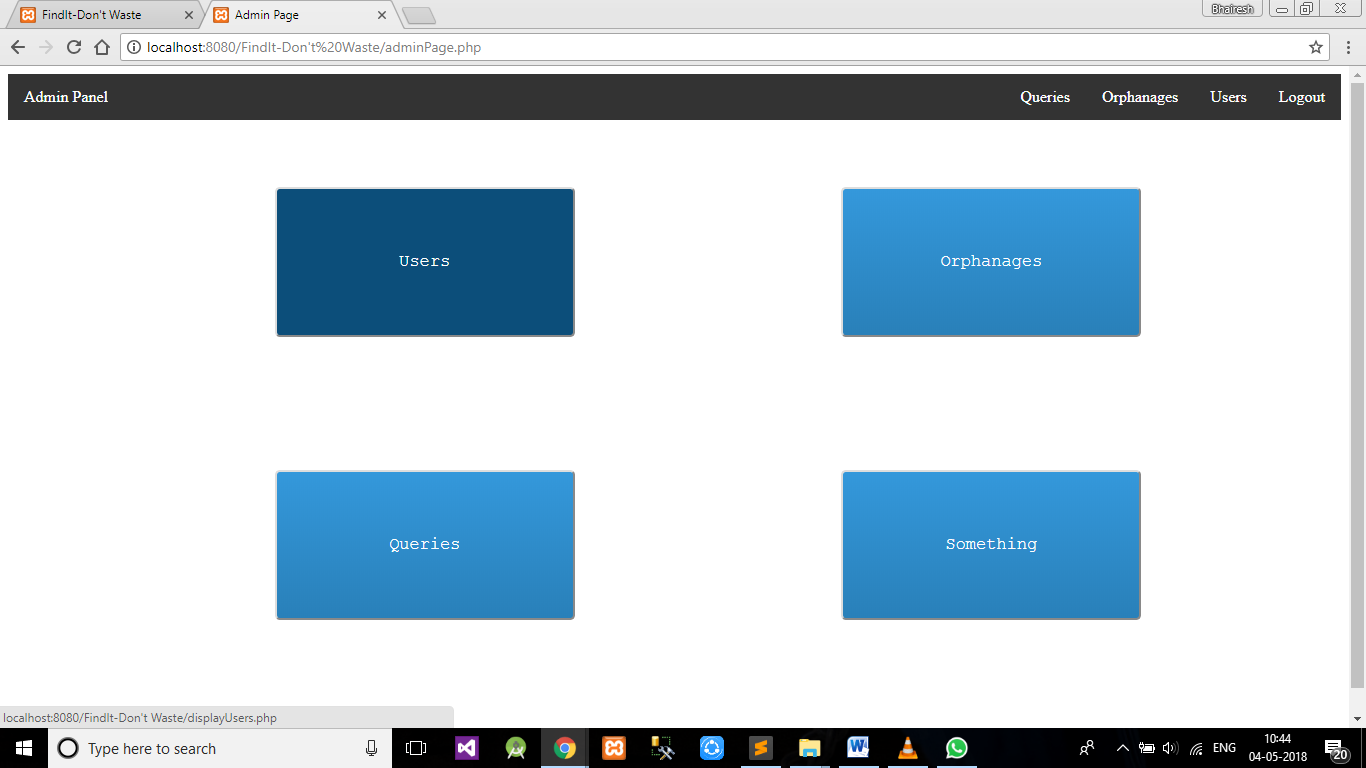
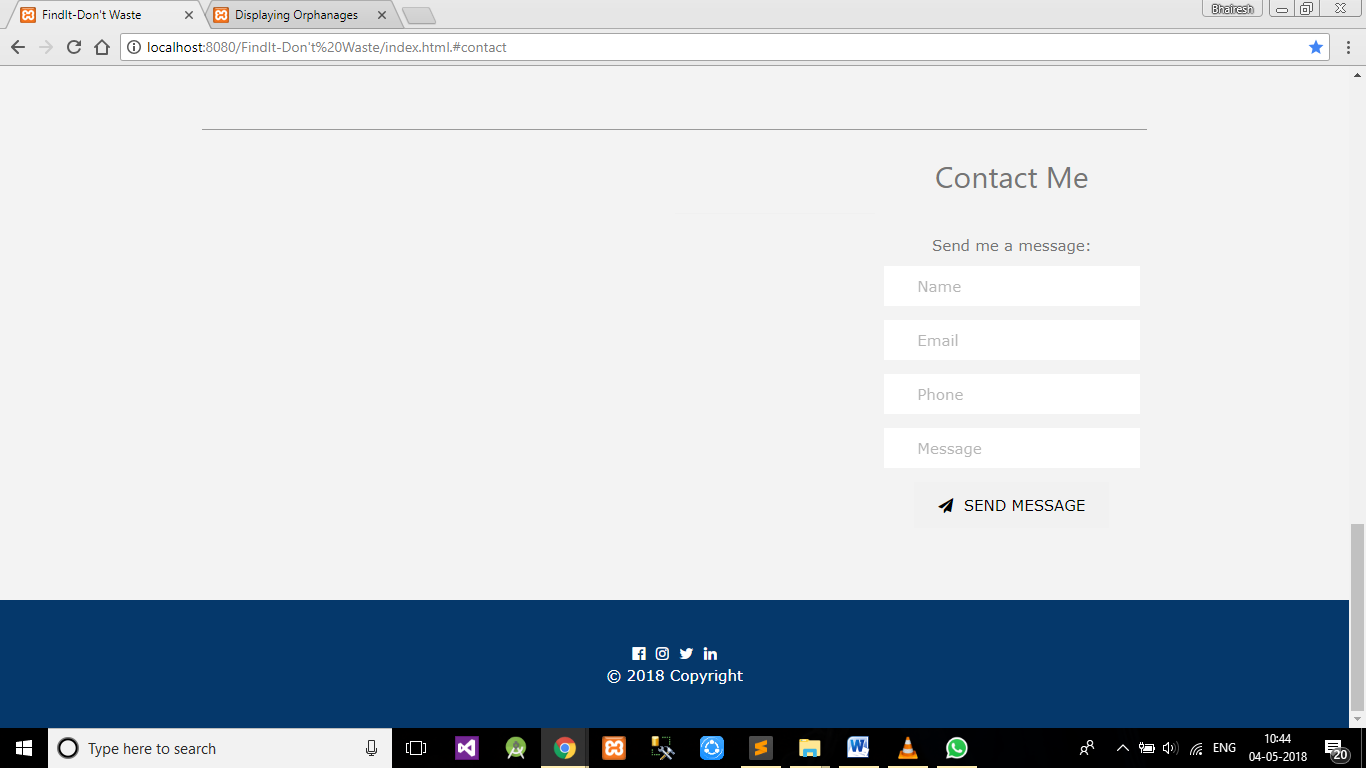
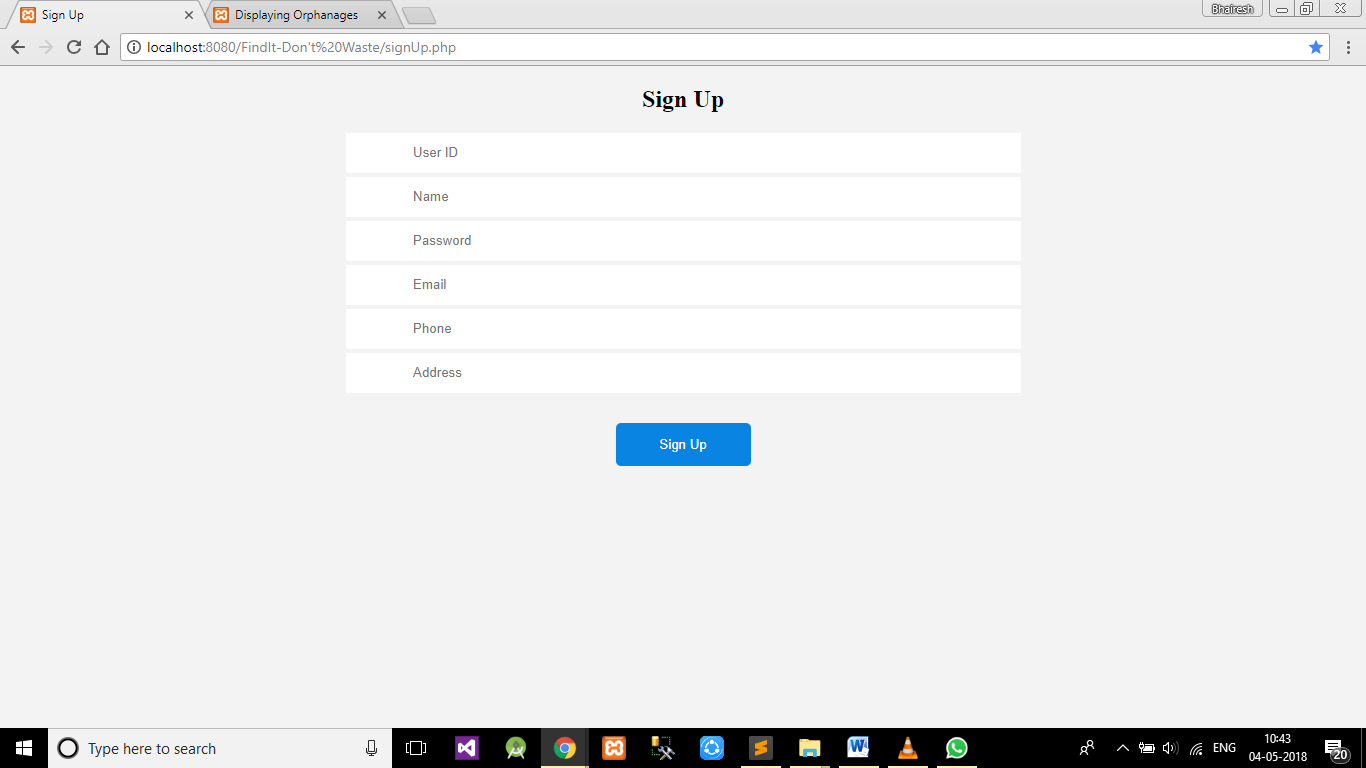
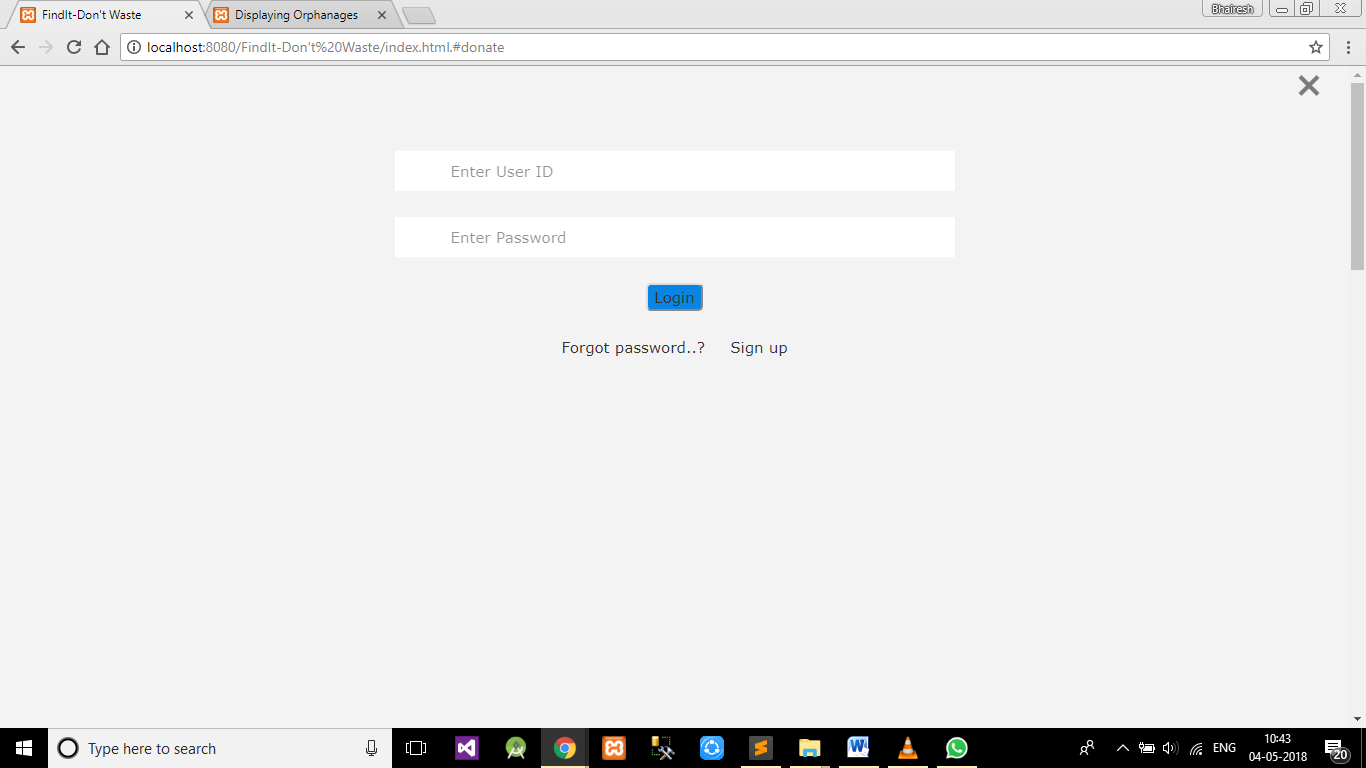
</div>

</body>

</html>

**SNAPSHOTS:**

****

****