DOCUMENTATION ON

"ONLINE CHARITY SYSTEM"

SUBMITTED IN

PARTIAL FULFILLMENT OF

DIPLOMA IN ADVANCED COMPUTING (PG-DAC)

SUBMITTED BY

GROUP BY: 96

Abhishek Sharma & Arpit Thanvi

PG-DAC: FEB-2020

UNDER THE MENTORING OF

Mrs. Sarika Kankaria & Mrs. Ashwini Thorat

Mr. Prashant Karhale
Center Coordinator

Mr. Akshay Parab
Project Guide

AT

Institute for Advanced Computing and Software Development

CERTIFICATE

This is to certify that the project

'ONLINE CHARITY SYSTEM'

Has been submitted by

Abhishek Sharma & Arpit Thanvi

In partial fulfillment of the requirement for the Course of **PG Diploma in Advanced Computing (PG-DAC FEB-2020)** as prescribed by The **CDAC** ACTS, PUNE.

Place: Pune Date: 30-JAN-2021

Mr. Akshay Parab
(Project Guide)

ACKNOWLEDGEMENT

The satisfaction that accompanies that the successful completion of any task would be incomplete without the mention of people whose ceaseless cooperation made it possible, whose constant guidance and encouragement crown all efforts with success.

We are grateful to our Alumni Mentor Guide project mentor Mrs. Sarika Kankaria mam mule and Mrs. Ashwini Thorat mam for the guidance, inspiration and constructive suggestions that helpful us in the preparation of this project. Lastly, we thank the entire faculty and the staff members of IACSD, pune for their support.

ABSTRACT

One of the key fundaments of building a society is common interest or shared aims of the group members. This research work is a try to analyze web-based services oriented towards money collection for various social and charity projects. The phenomenon of social founding is worth a closer look at because its success strongly depends on the ability to build an ad-hoc or persistent groups of people sharing their believes and willing to support external institutions or individuals. The paper presents a review of money collection sites, various models of donation and money collection process as well as ways how the projects' results are reported to their founders. There is also a proposal of money collection service, where donators are not charged until total declared help overheads required resources to complete the project. The risk of missing real donations for declared payments, after the collection is closed, can be assessed and minimized by building a social network.

INDEX

List of Figures

1. Introduction

1.1 Purpose:

This document is meant to illustrate the features of Online charity system, so as to serve as a guide to the developers on one hand and a software validation document for the prospective client on the other.

The purpose of OCS is to provide an easy way to help those needy people by doing charity. This project provides an online platform which helps people who want to do charity by a single click using internet. This project gives the list of that NGOS, Orphan Houses and Old Age Houses to person who want to do charity. This project has an online payment method by which beneficiaries directly take benefits. Online payment provides a transparency and there is no chance of corruption and now a days we are facing PANDAEMIC like **COVID** in which physical contact should be avoided so, in this situation our project also helps to beneficiaries.

1.2 Scope:

This system allows the philanthropist i.e. person who want to do charity is directly search people who are needy and help them out.

1.3 Definitions:

OCS- Online Charity System

SRS- Software Requirement Specification

GUI- Graphical User Interface

RDBMS- Relational Database Management System

Philanthropist-The person/company/organization who want to do charity.

Beneficiary- The person/company/organization who got charity.

1.4 Overview:

This system provides an easy way to philanthropist and beneficiaries without going any where and without any hindrance. This system is user friendly, no need of technical experience, and no need of any technology hands on. Its GUI is user-friendly end user only have to click and do anything which they want.

2.Overall Description:

The Online Charity System (OCS) application enables NGOs, Orphan Houses, Old Age Homes to register using their own information e.g. Bank Details, Location and Number of persons who lived over there. Philanthropist also can register in our online project and can easily access the list of beneficiaries and help them out without visiting over there.

The online charity system will use the internet as the sole method for giving a list of beneficiaries to philanthropist.

2.1 Product Perspective:

This project aimed towards to those people who want to help other person and for those who are actually needy because of PANDEMIC these peoples are unable to visit anywhere. This project provides an interface between donors and acceptors.

2.2 User Characteristics:

User should be familiar with the terms like login, register, payment system, accessing of list etc.

2.3 Principle Actors:

Two Principle Actors are Philanthropist, Beneficiary and Administrative who own this web project.

2.4 General Constraints:

A full internet connection is required for OCS.

2.5 Assumptions and Dependencies:

Working of OCS need Internet Connection.

3. Specific Requirements:

3.1 Functional Requirements:

This section provides requirement overview of the system. Various functional modules that can be implemented by the system will be –

3.1.1 Registration:

If beneficiary wants the benefit of charity and donor wants to donate then they must be registered otherwise they can’t do any kind of transaction.

3.1.2 Login:

Donor/Acceptor logins to the system by entering valid user id and password for the transaction.

3.1.3 Shortlist:

End User can browse list of NGOs, Orphan Houses, Old Age Homes they can do charity over there.

3.1.4 Payment:

In this system there are many of secure billing will be prepaid as debit or credit cart, Net Banking, Cash, UPI, Phone Wallets, etc.

3.1.5 Logout:

After doing transaction has to logout.

3.1.6 Report Generation:

After registering and transaction, system will be sent one receipt as an acknowledgment mail.

3.2 Non-Functional Requirements:

Following Non-Functional Requirements will be there in the insurance to the internet:

(i) Secure access to consumer's confidential data.

(ii) 24X7 availability.

(iii) Better component design to get better performance at peak time.

(iv) Flexible service-based architecture will be highly desirable for future extension. Non-Functional Requirements define system properties and constraints.

Various other Non-Functional Requirements are:

1.Security:

The System use SSL (Secure Socket Layer) in all transactions that include any confidential donors' information.

The system must automatically log out all donors after a period of inactivity.

The system should not leave any cookies on the donor's computer containing donor's password.

The system's back-end servers shall only be accessible to authenticated administrators.

Sensitive data will be encrypted before being sent over insecure connections like internet.

The proper firewalls should be developed to avoid intrusions from the internal or external sources.

2.Reliability:

The system provides storage of all databases on redundant computers with automatic switchover.

The main pillar of reliability of the system is the backup of the database

which is continuously maintained and update to reflect the most recent changes.

3: Availability:

The system should be available at all times. meaning the user can access it using web browser, only restricted by the down time of the server on which the system runs.

In case of a of a hardware failure or database corruption, a replacement page will be shown.

uptime: It mean 24 * 7 availability

4: Maintainability:

A commercial database is used for maintaining the database and application server takes care of the site.

The maintainability can be done efficiently.

5.Portability:

The application is HTML and scripting language based (JavaScript). So the end user part is fully portable and any system using any web browser should be able to use the features of the system, including any hardware platform that is available will be available in the future.

An end-user is used this system on an OS either it is Windows or Linux.

The System shall run on PC, Laptops and PDA etc.

The technology should be transferable to different environments easily.

6.Accessibility:

Only registered users should be allowed to process the orders after authentications.

Only GUI access of the system should be permitted to end users.

7.Policies:

The system should adhere to all the legal formalities of the particular countries.

The system should maintain security related to sensitive data.

8.Efficiency:

The system should provide good throughput and response to multiple users without burdening the system by using appropriate number of servers.

9.Safety:

Software should not harm ethical and environmental conditions of the end donors' machine.

10.Modulariy:

The system should have user friendly interface.

It should be easily updated, modified and reused.

3.3 Performance Requirements:

In order to maintain an acceptable speed at maximum number of uploads allowed from a particular customer as any number of users can access to the system at any time. Also, the connections to the servers will be based on the attributes of the user like his location and server will be working 24X7 times.

3.4 Technical Issues:

This system will work on client-server architecture. It will require an internet server. The system should support some commonly used browser such as IE, Mozilla Firefox, chrome etc.

3.5 Interface Requirement:

Various interfaces for the product could be

1. Welcome Page
2. Login Page
3. Registration form
4. List of DONORS/ACCEPTORS

5. Payment Details

6. Payment history

7.Payment Gateways

8.Mail for Acknowledgement

Hardware Interface:

The System must run over the internet.

All the hardware shall require to connect to internet will be hardware interface for the system e.g., modem, WAN, LAN.

Software Interface:

The system is on server so it requires the any scripting language like JSP.

Front end: Angular

Back end: Java

Database: MySQL

The system requires Data Base also for the store the any transaction of the system like MySQL or oracle, or SQL server etc. At the end-user need web browser for interact with the system.

Operating Environment:

- Server Side:
  - ❖ **Preprocessor:** Intel Xenon preprocess 3500 series
  - ❖ **HDD:** Minimum 500GB Disk Space
  - ❖ RAM: **Minimum 16GB**
  - ❖ OS: **Windows 8.1, Linux 6**
  - ❖ Database: **MySql**
- Client Side (minimum requirement)
  - ❖ Preprocessor: Intel Dual Core
  - ❖ **HDD:** Minimum 80GB Disk Space
  - ❖ **RAM:** Minimum 4GB
  - ❖ **OS:** Windows 7, Linux

Design and Implementation Constraints:

- HTTP and FTP protocols are used as communication protocols. FTP is used to upload the web application in live domain and the client can access it via HTTP protocol.
- Several types of validations make this web application a secured one and SQL Injections can also be prevented.
- Since Society Management system is a web-based application, internet connection must be established.
- The Charity Management System will be used on PCs and will function via internet or intranet in any web browser.

Specific Requirement

External Interface Requirements:

User Interfaces:

- All the users will see the same page when they enter in this website. This page asks the users a username and a password.
- After being authenticated by correct username and password, user will be redirect to their corresponding profile where they can do various activities.
- The user interface will be simple and consistence, using terminology commonly understood by intended users of the system. The system will have simple interface, consistence with standard interface, to eliminate need for user training of infrequent users.

Hardware Interfaces:

- No extra hardware interfaces are needed.
- The system will use the standard hardware and data communication resources.
- This includes, but not limited to, general network connection at the server/hosting site, network server and network management tools.

Application Interfaces:

- **OS:** Windows 7, Linux
- **Web Browser:**

The system is a web-based application; clients need a modern web

browser such as Mozilla Firebox, Internet Explorer, Opera, and Chrome. The computer must have an Internet connection in order to be able to access the system.

Communications Interfaces:

- This system uses communication resources which includes but not limited to, HTTP protocol for communication with the web browser and web server and TCP/IP network protocol with HTTP protocol.
- This application will communicate with the database that holds all the booking information. Users can contact with server side through HTTP protocol by means of a function that is called HTTP Service. This function allows the application to use the data retrieved by server to fulfil the request fired by the user.

System Design:

Activity Diagram

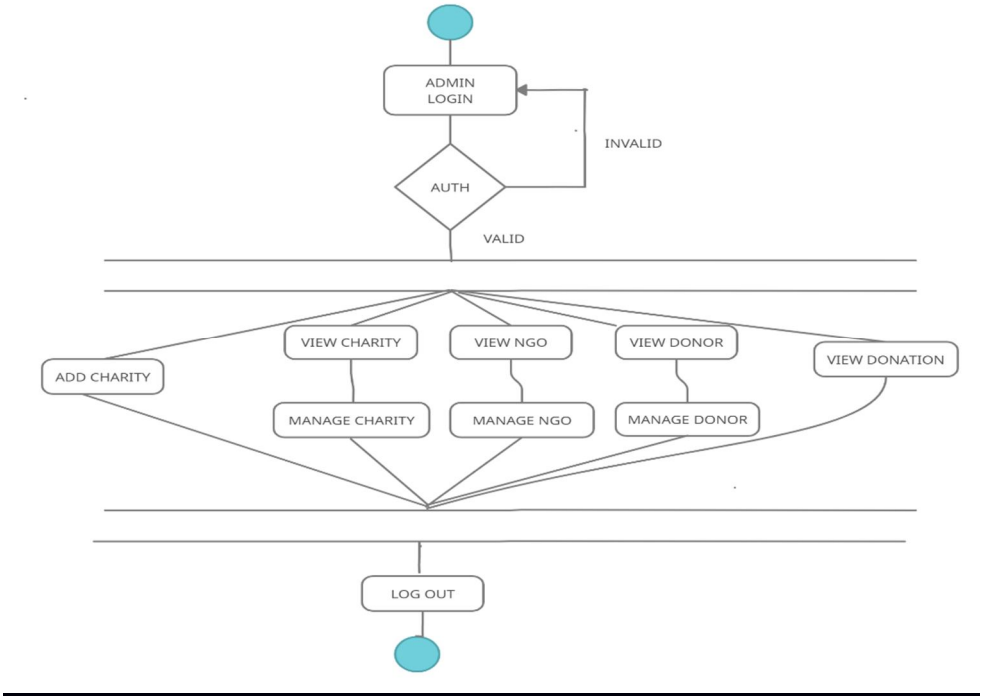


Fig 1: Admin Activity Diagram

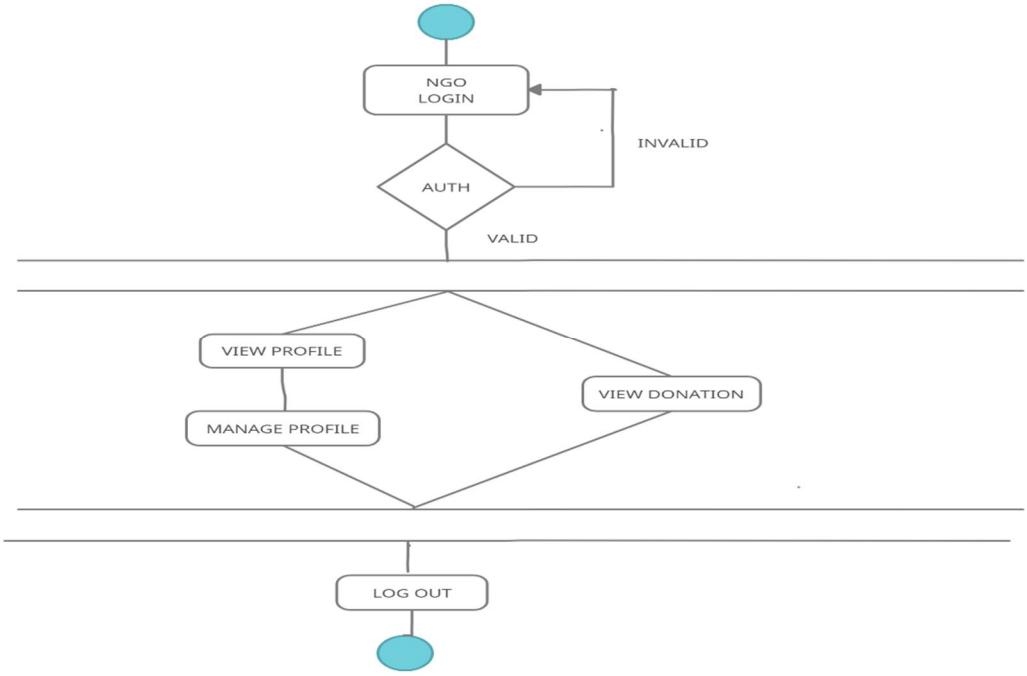


Fig 2: NGO Activity Diagram

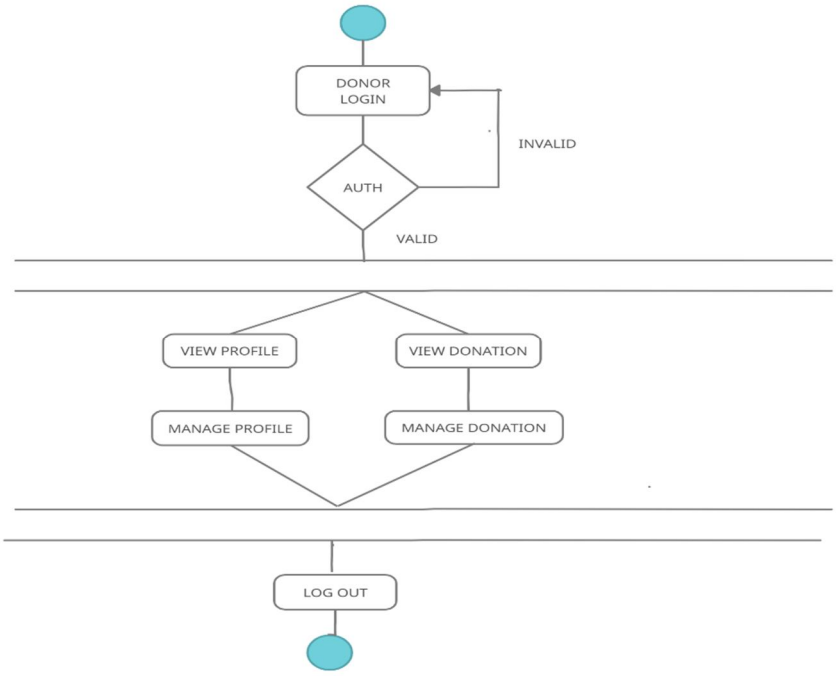


Fig 3: Donor Activity Diagram

Data Flow Diagram:

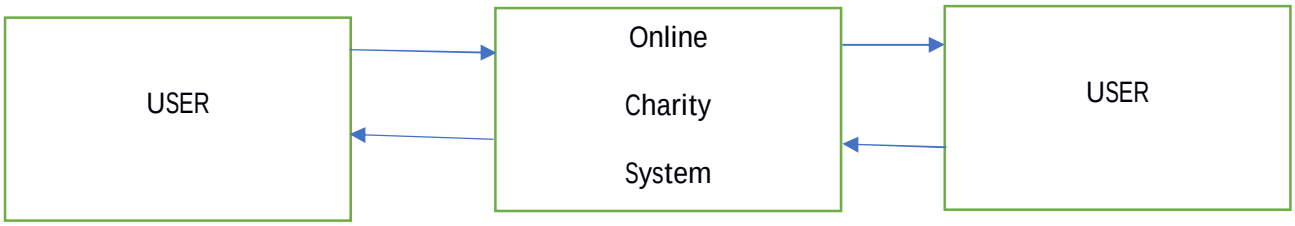


Fig 4: Level 0 Data Flow Diagram

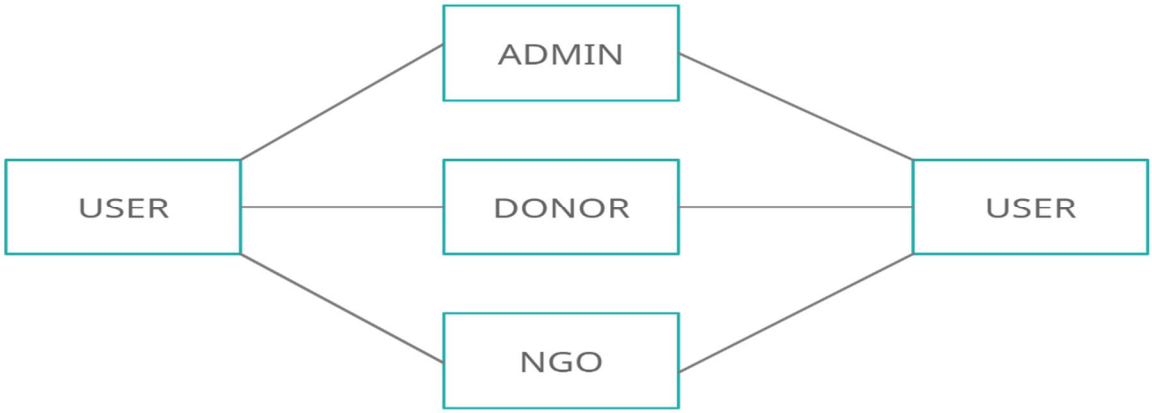


Fig 5: Level 1 DFD

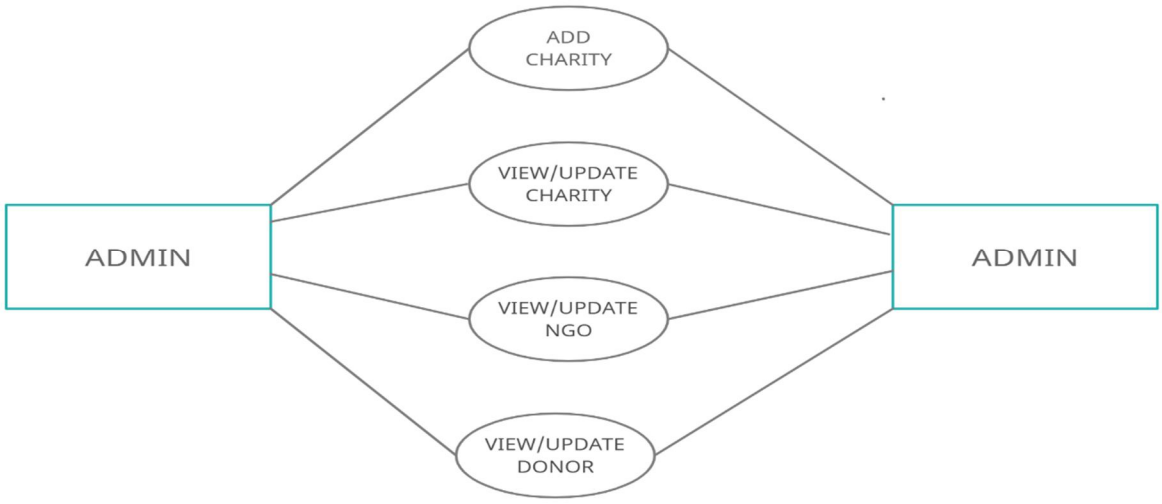


Fig 6: Level 2 DFD for Admin

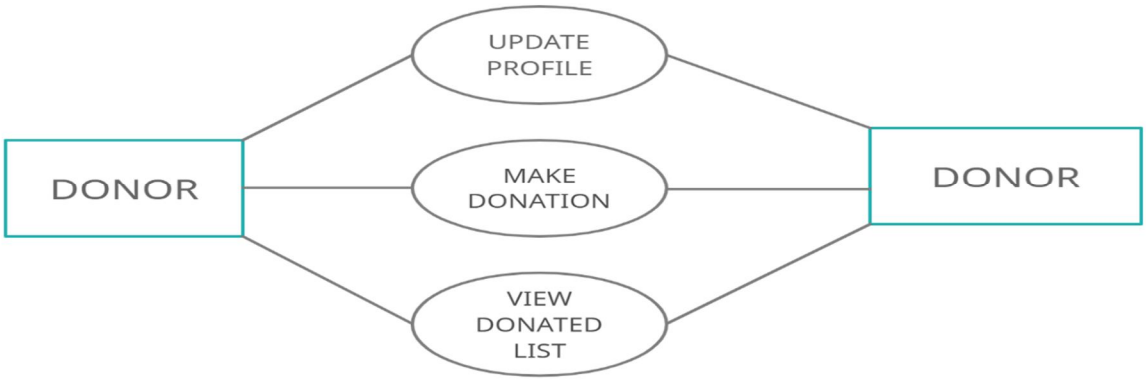


Fig 7: Level 2 DFD for Donor

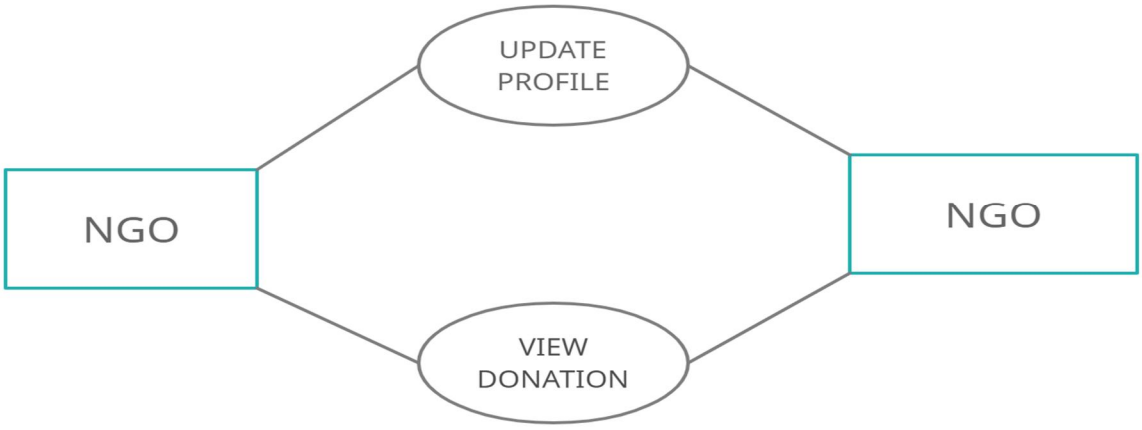


Fig 8: Level 2 DFD for NGO

Class Diagram:

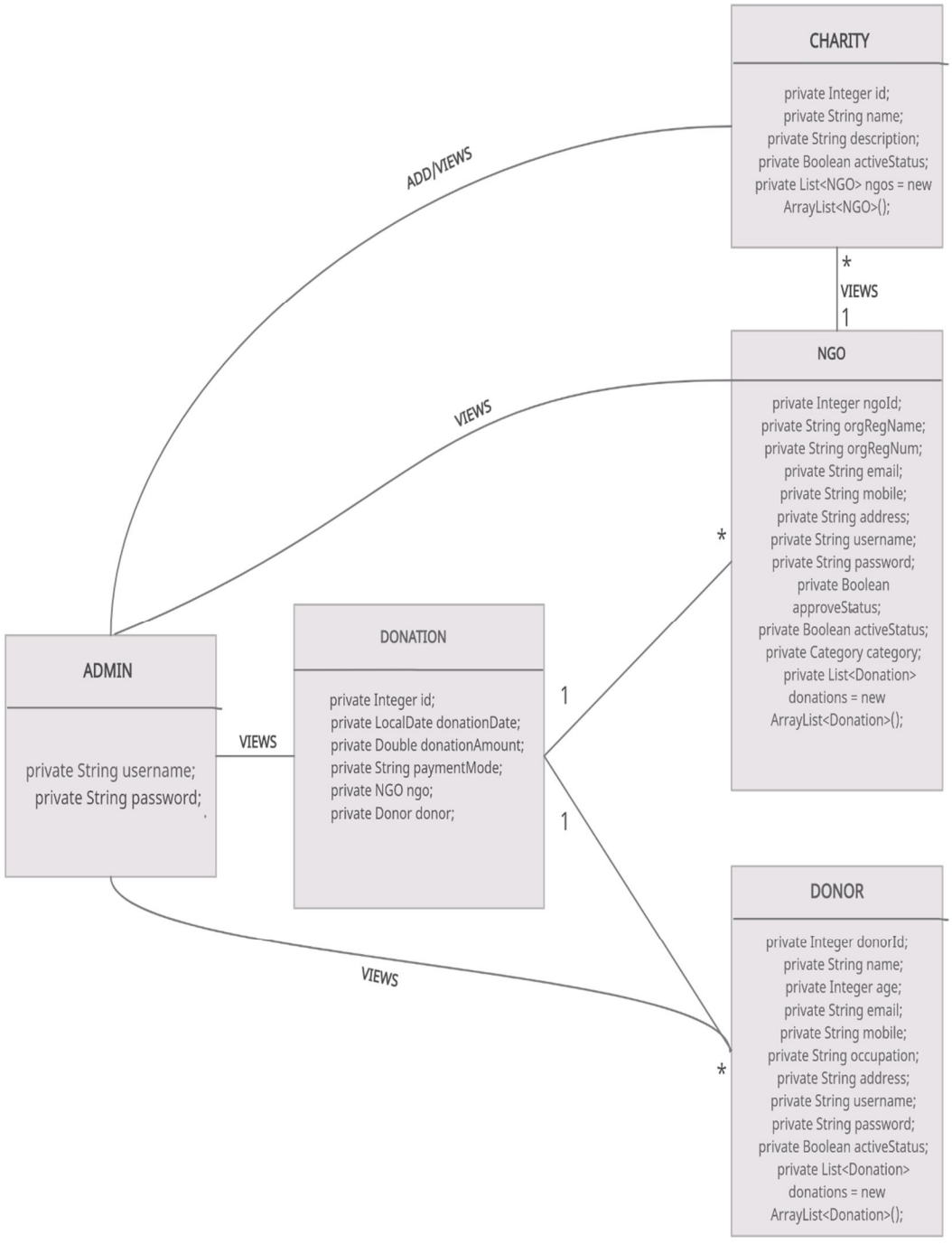


Fig 9: Class Diagram

UML Use Case Diagram:

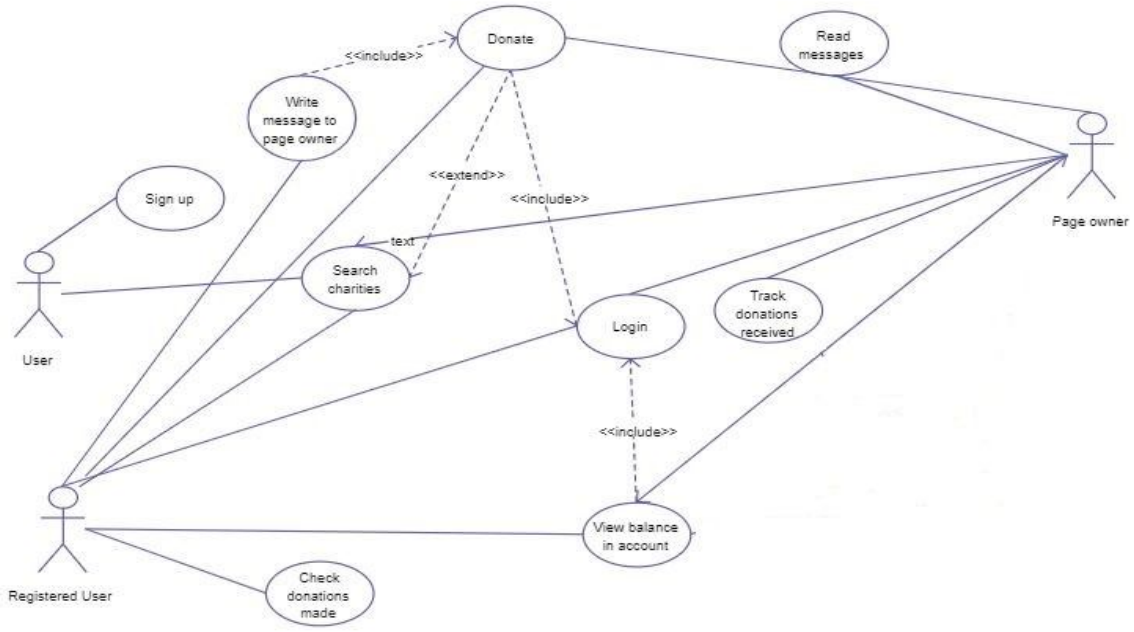


Fig 10: UML Use case diagram

Entity Relation Diagram:

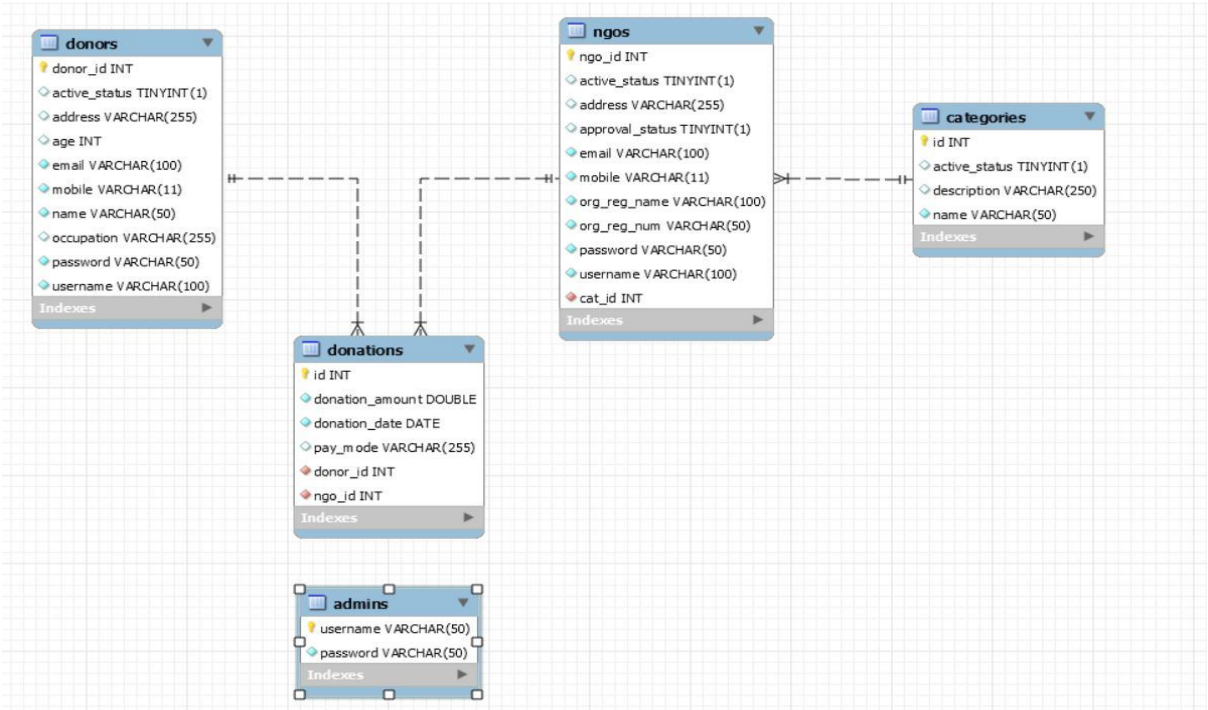


Fig 11 : ER Diagram

6.Software Development Life Cycle:

Any software development project goes through a process known as a software life cycle. The International Standard, ISO/IEC 12207:2008 have established a common framework outlining the standard that defines the activities required for software life cycle processes [21]. These activities contain tasks that are to be applied during project life cycle. The methods used for implementing the software is based on the development process model described in the next section

Somerville defines software process in [7] as a set of activities and processes that are involved in a production of a software product. Boehm states that the primary functions of a software process model are to determine the order of the stages involved in software development and the criteria for progressing from one stage to the next one [22]. Boehm further contents that software process models are important because the models guide the developer(s) on the order in which a project should be carried out. These days a wide variety of software process models have evolved, each with its own strengths and weaknesses. Some of these recent models today are the rapid throwaway prototypes, incremental development, evolutionary prototypes, reusable software, and automated software synthesis. In [7], Somerville discusses various software process models and identifies that one process model is not necessarily suitable for use by all projects. Each process model is best suited to specific kinds of projects, based on various techniques and organizational needs. Nevertheless, some fundamental stages are common to all the software process models, example of such stages are the requirements phase, the design phase, the implementation phase and the testing phase.

In the same work Somerville discusses the following software development models:

The Waterfall Model: This model is represented as a sequential design process where the development process progresses downwards. The waterfall model was defined in the 1970s by Royce and later refined by Boehm in 1976. Because of this transition from one phase to another, the model is called the waterfall model, just like a waterfall. The sequential tasks involve in this model includes the requirement specification, software design, implementation, testing and maintenance. This model represents a linear model which makes it the most simple to be understood. One of the advantages of this model is that documentation is produced after every stage which makes the understanding of the development procedure simpler by all stakeholders. One of the disadvantages of this model is that if any of the previous stage has gone wrong, things can get very complicated in later stages. There are other modified versions

of the waterfall model such as the V Waterfall model which aims to improve the shortcoming of the original model.

The spiral model: In order to overcome the disadvantages of waterfall model, spiral model was developed. This model is represented as a spiral where each phase incorporates the different stages of the waterfall model. The spiral model contains four phases which are: Planning, Evaluation, Risk Analysis and Engineering. As one move from one phase of the spiral to another, you repeat all the stages of the waterfall model. Some of the disadvantages of the spiral model are that it demands more time, it is complicated and requires highly skilled people in the area of planning, risk analysis and mitigation, development, customer relation etc.

Evolutionary development: This is model based on the idea of developing an initial implementation, presenting it to the stakeholders for feedback, development team responds to the feedback, often by refining the product until an adequate system has been developed. One of the advantages of this model is that the customers can see steady progress. One of the disadvantages is that it's impossible to know at the outset of the project how long it will take. There are two types of Evolutionary development models: 1. Exploratory development: Objective is to work with the customers to evolve a final system from an initial outline specification. Process starts with the wellunderstood requirements. 2. Throw-away prototyping: Objective is to understand the system requirements. Process starts with the poorly understood requirements. Component-based software engineering: This approach is based on already existing of reusable components that can be integrated into other components. After studying the different software development processes, the model chosen for this project will be a slightly modified version of the waterfall model. The pure waterfall model provides very clear phases and considering that there is only one developer implementing the software, the developer can focus on each part of the model during stages and come back previous stages if needed to. The modified waterfall adopted for this thesis work uses the same phases of the pure waterfall, but on iterative basis. This enables the phases to overlap when needed.

6.1 The Waterfall Model for Software Development

The waterfall model recommends software development to go sequentially and follow certain rules and stages. The methods used for implementing the WCMS are categorized into different phases following the recommendations of the waterfall model.

Planning phase: During the planning phase, development of the product is initiated. Stakeholders and goals are identified. Various existing technologies are being explored and learnt and how useful they are in producing a well-organized software system.

Requirements specification: This phase is about gathering information about what the owner needs and defining it in the clearest possible terms and the problem that the software is expected to solve. Techniques used to obtain these specifications include Scenario-driven description. The results of the analysis form the requirement specifications then to serve as a contract between partners. **Design:** This phase consists of designing the web layout interfaces to satisfy specified requirements. A mock-up layout is designed and presented to the stakeholders for approval. It also involves separating the administrative interfaces from all other users.

Implementation: This phase consists of actually constructing the software according to the design specification(s) developed in the previous phase.

System Testing and Integration: This phase is about integrating different components of the software and testing them in order to validate the requirement specifications.

Maintenance: This phase occurs after testing and deployment. It involves making modifications to the system to alter attributes or improve performance. However, this phase is

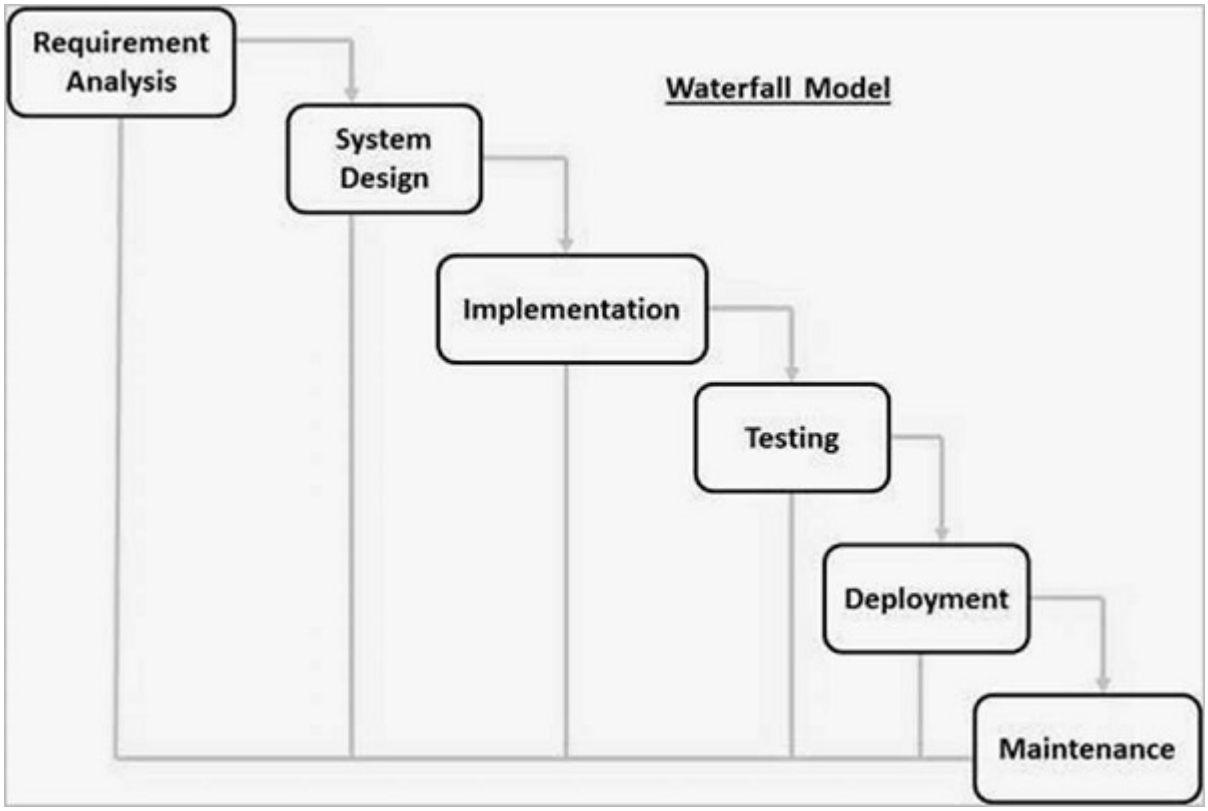


Table Structure:

All tables in CHARITY tables:

```
+--------------------+
| Tables_in_charity  |
+--------------------+
| admins             |
| categories         |
| donations          |
| donors             |
| ngos               |
+--------------------+
```

Admin Table:

```
mysql> desc admins;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| username | varchar(50) | NO   | PRI | NULL    |       |
| password | varchar(50) | NO   |     | NULL    |       |
+----------+-------------+------+-----+---------+-------+
```

```
+-----------+-----------+
| username  | password  |
+-----------+-----------+
| abhishek  | sharma    |
| arpit     | thanvi    |
+-----------+-----------+
2 rows in set (0.00 sec)
```

Category Table:

```
mysql> desc categories;
+---------------+--------------+------+-----+---------+----------------+
| Field         | Type         | Null | Key | Default | Extra          |
+---------------+--------------+------+-----+---------+----------------+
| id            | int          | NO   | PRI | NULL    | auto_increment |
| active_status | tinyint(1)   | YES  |     | 1       |                |
| description   | varchar(250) | YES  |     | NULL    |                |
| name          | varchar(50)  | NO   |     | NULL    |                |
+---------------+--------------+------+-----+---------+----------------+
```

```
mysql> select * from categories;
+----+---------------+-----------------------------+---------------+
| id | active_status | description                 | name          |
+----+---------------+-----------------------------+---------------+
|  1 |             0 | We are helping orphans      | Orphan Help   |
|  2 |             1 | We are helping old peoples  | Old Age Help  |
+----+---------------+-----------------------------+---------------+
```

NGO Table:

```
mysql> desc ngos;
+------------------+--------------+------+-----+---------+----------------+
| Field            | Type         | Null | Key | Default | Extra          |
+------------------+--------------+------+-----+---------+----------------+
| ngo_id           | int          | NO   | PRI | NULL    | auto_increment |
| active_status    | tinyint(1)   | YES  |     | 1       |                |
| address          | varchar(255) | YES  |     | NULL    |                |
| approval_status  | tinyint(1)   | YES  |     | 0       |                |
| email            | varchar(100) | NO   | UNI | NULL    |                |
| mobile           | varchar(11)  | NO   | UNI | NULL    |                |
| org_reg_name     | varchar(100) | NO   |     | NULL    |                |
| org_reg_num      | varchar(50)  | NO   |     | NULL    |                |
| password         | varchar(50)  | NO   |     | NULL    |                |
| username         | varchar(100) | NO   |     | NULL    |                |
| cat_id           | int          | NO   | MUL | NULL    |                |
+------------------+--------------+------+-----+---------+----------------+
```

```
mysql> select * from ngos;
+--------+---------------+-----------------------+-----------------+-------------------------+------------+-------------------------------+--------------+-----------+-----------+--------+
| ngo_id | active_status | address               | approval_status | email                   | mobile     | org_reg_name                  | org_reg_num  | password  | username  | cat_id |
+--------+---------------+-----------------------+-----------------+-------------------------+------------+-------------------------------+--------------+-----------+-----------+--------+
|      1 |             1 | Hyedrabad, Telangana  |               1 | heulness_edu@gmail.com  | 7376761774 | Helpful service Foundation    | ID9167781    | foundation | heartfulness |      1 |
|      2 |             1 | Nashik, Maharashtra   |               1 | oldpeops@gmail.com      | 8726641996 | Old Age service Foundation    | IM9167781    | 1234      | oldage    |      2 |
|      3 |             0 | Shimla, Himanchal Pradesh |           1 | smile@gmail.com         | 7275774563 | Smile Care Foundation         | IH91342100   | care      | smile     |      2 |
|      4 |             1 | mahad                 |               1 | old12@gmail.com         | 1234       | Old                           | old123       | 1234      | oldp      |      2 |
+--------+---------------+-----------------------+-----------------+-------------------------+------------+-------------------------------+--------------+-----------+-----------+--------+
```

Donor Table:

```
mysql> desc donors;
+----------------+--------------+------+-----+---------+----------------+
| Field          | Type         | Null | Key | Default | Extra          |
+----------------+--------------+------+-----+---------+----------------+
| donor_id       | int          | NO   | PRI | NULL    | auto_increment |
| active_status  | tinyint(1)   | YES  |     | 1       |                |
| address        | varchar(255) | YES  |     | NULL    |                |
| age            | int          | YES  |     | NULL    |                |
| email          | varchar(100) | NO   | UNI | NULL    |                |
| mobile         | varchar(11)  | NO   | UNI | NULL    |                |
| name           | varchar(50)  | NO   |     | NULL    |                |
| occupation     | varchar(255) | YES  |     | NULL    |                |
| password       | varchar(50)  | NO   |     | NULL    |                |
| username       | varchar(100) | NO   |     | NULL    |                |
+----------------+--------------+------+-----+---------+----------------+
```

```
mysql> select * from donors;
+----------+---------------+-----------+------+-------------------------+------------+-----------------+-----------------+----------+----------+
| donor_id | active_status | address   | age  | email                   | mobile     | name            | occupation      | password | username |
+----------+---------------+-----------+------+-------------------------+------------+-----------------+-----------------+----------+----------+
|        1 |             1 | Delhi     |   30 | avi@gmail.com           | 9161348111 | Avneesh Kumar   | Service         | 1234     | avi      |
|        2 |             0 | Student   |   25 | sharmaadarsh@gmail.com  | 8726641998 | Adarsh Sharma   | Lucknow, U.P.   | sharma   | adarsh   |
|        3 |             1 | Jankipuram|   22 | ankit@gmail.com         | 8914567412 | Ankit Singh     | Zumba Trainer   | 0987     | ankit    |
|        5 |             1 | M.P.      |   45 | priyu@gmail.com         | 7623451287 | priyanshi sohani| Student         | 5678     | priyu    |
|        6 |             1 | pune      |   25 | djhdjhd@gmail.com       | 75252511232| aishwarya       | Student         | 1234     | aish     |
+----------+---------------+-----------+------+-------------------------+------------+-----------------+-----------------+----------+----------+
```

Donation Table:

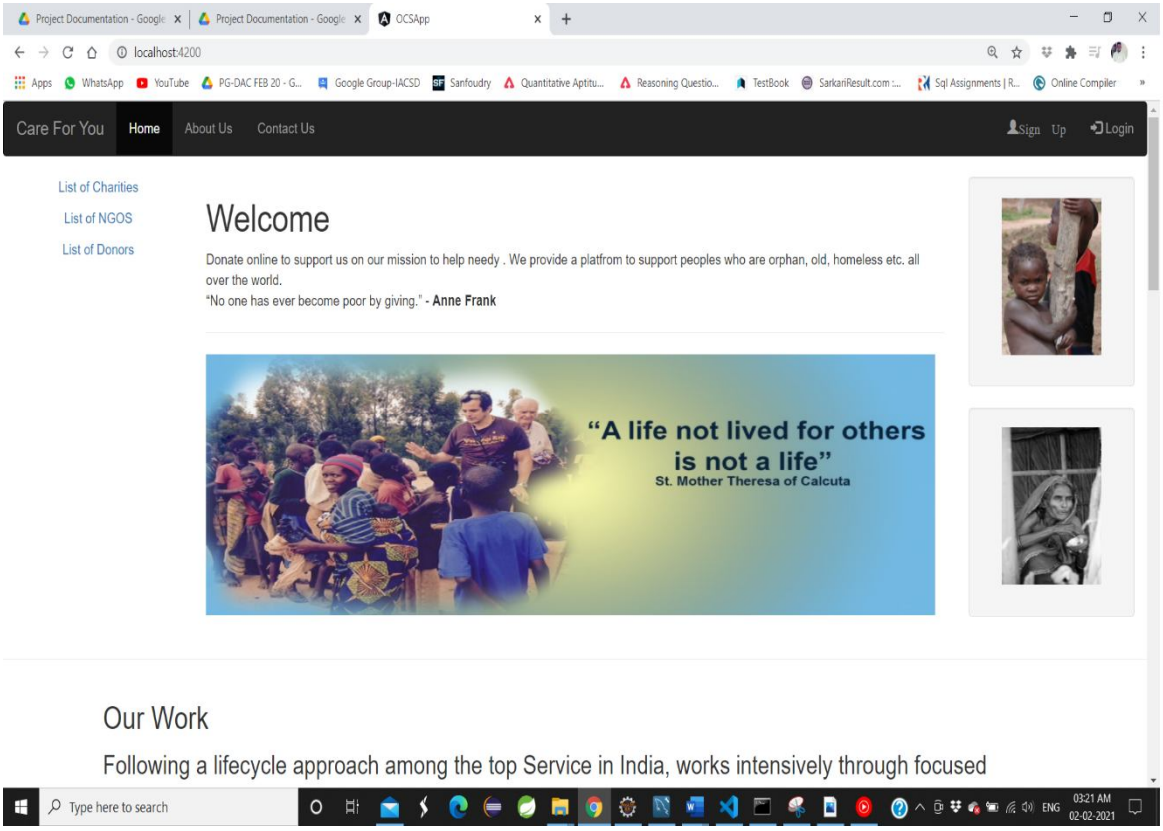
```
mysql> desc donations;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| id              | int          | NO   | PRI | NULL    | auto_increment |
| donation_amount | double       | NO   |     | NULL    |                |
| donation_date   | date         | NO   |     | NULL    |                |
| pay_mode        | varchar(255) | YES  |     | NULL    |                |
| donor_id        | int          | NO   | MUL | NULL    |                |
| ngo_id          | int          | NO   | MUL | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
```

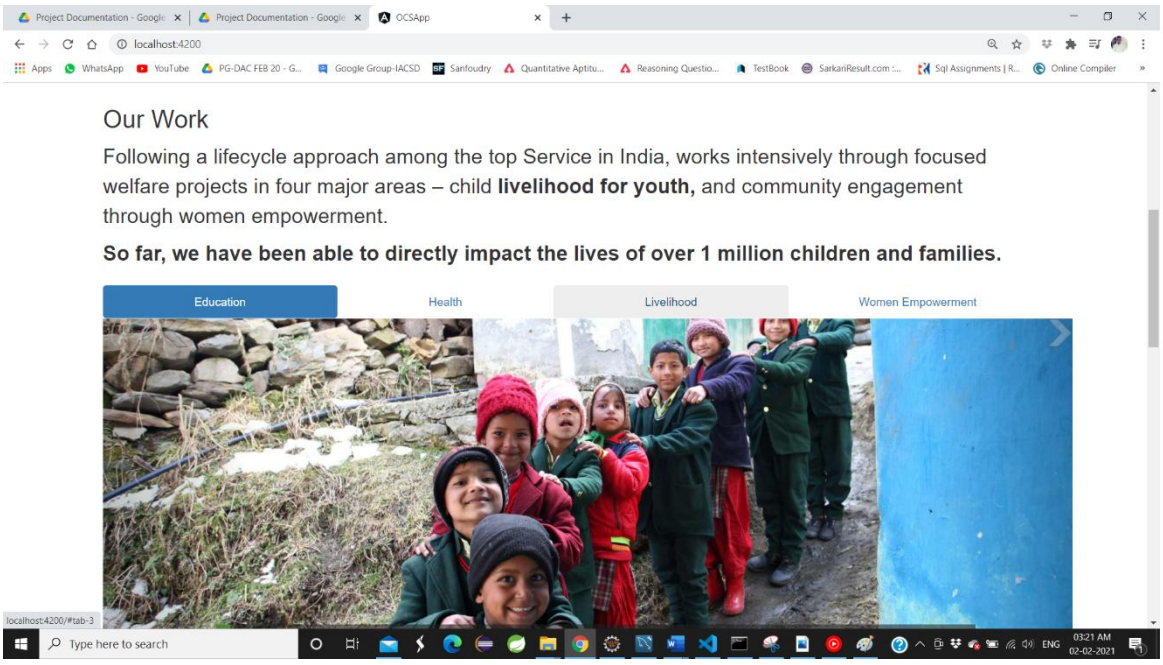
```
mysql> select * from donations;
+----+-----------------+---------------+---------------+----------+--------+
| id | donation_amount | donation_date | pay_mode      | donor_id | ngo_id |
+----+-----------------+---------------+---------------+----------+--------+
|  1 |           24350 | 2020-12-22    | Google Pay    |        1 |      1 |
|  2 |           15000 | 2021-01-23    | Bank Transfer |        3 |      3 |
|  3 |           24350 | 2020-12-22    | Google Pay    |        3 |      1 |
|  4 |           24350 | 2020-12-22    | Google Pay    |        1 |      3 |
|  5 |           25000 | 2021-01-08    | Online        |        2 |      4 |
|  6 |           13000 | 2021-01-03    | Pay           |        2 |      1 |
|  7 |           31999 | 2021-01-19    | Phone Pay     |        5 |      2 |
|  8 |           10000 | 2021-01-03    | online        |        5 |      1 |
|  9 |            7800 | 2021-01-02    | online        |        5 |      4 |
| 10 |           15000 | 2021-01-20    | bank          |        5 |      3 |
| 11 |            2100 | 2021-01-20    | google pay    |        1 |      4 |
| 12 |           12000 | 2021-01-21    | Phone Pay     |        1 |      2 |
| 13 |            3199 | 2021-01-20    | bank          |        2 |      3 |
| 14 |           31000 | 2021-01-15    | phone         |        2 |      2 |
| 15 |            1000 | 2021-01-15    | bank          |        3 |      4 |
| 16 |            5001 | 2021-01-22    | neft          |        2 |      2 |
| 17 |            1200 | 2021-01-16    | online        |        6 |      1 |
| 18 |            2100 | 2021-01-16    | google        |        6 |      4 |
| 19 |            1500 | 2021-01-16    | online        |        6 |      3 |
| 20 |            2100 | 2021-01-10    | gogle         |        6 |      2 |
| 21 |           99999 | 2021-01-03    | online        |        6 |      1 |
| 22 |            1100 | 2021-01-01    | online        |        5 |      1 |
| 23 |            1500 | 2021-01-31    | online        |        2 |      2 |
| 24 |            1000 | 2021-02-01    | online        |        2 |      2 |
+----+-----------------+---------------+---------------+----------+--------+
```

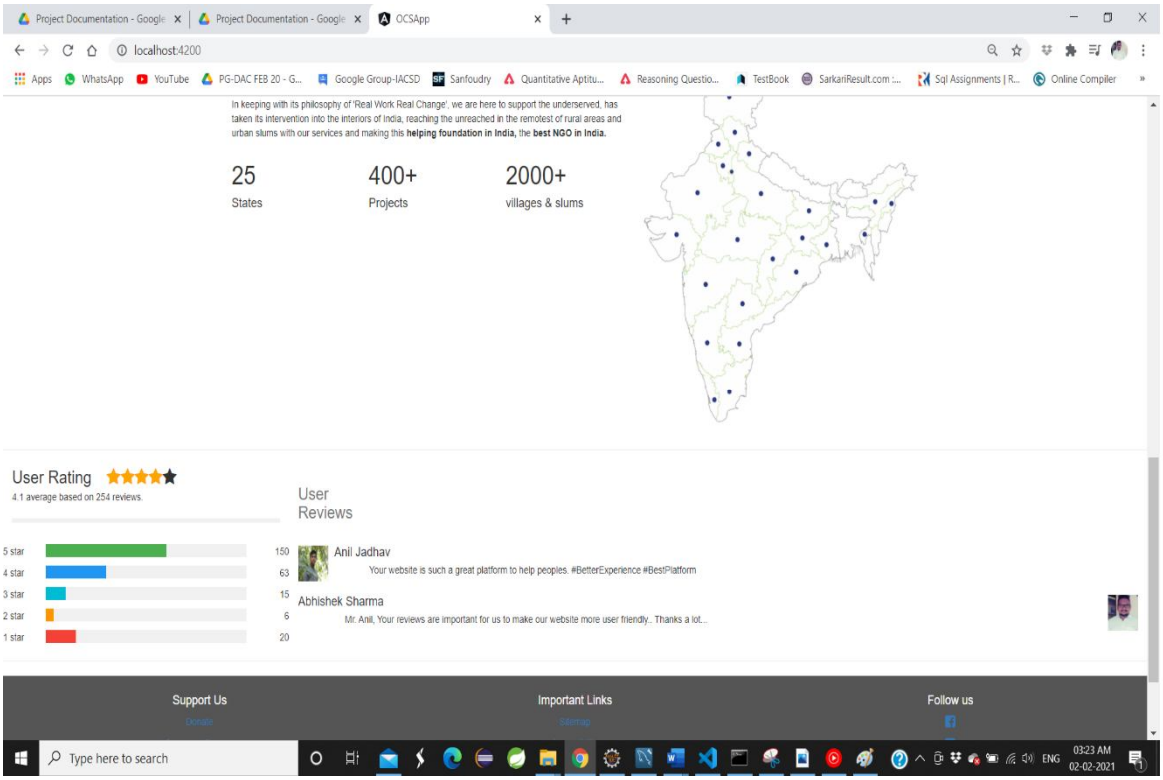
User Interface:

***All below pictures for normal visitor who are not logged in

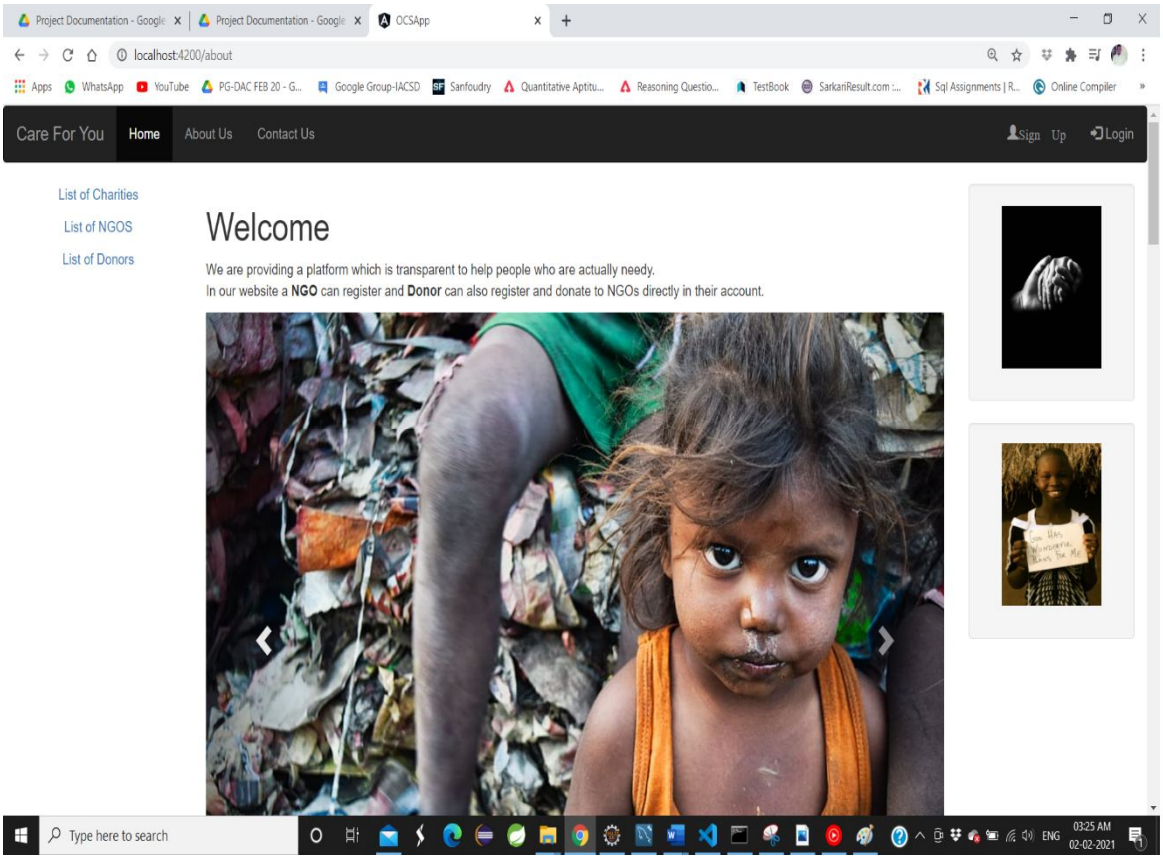
Home Page:

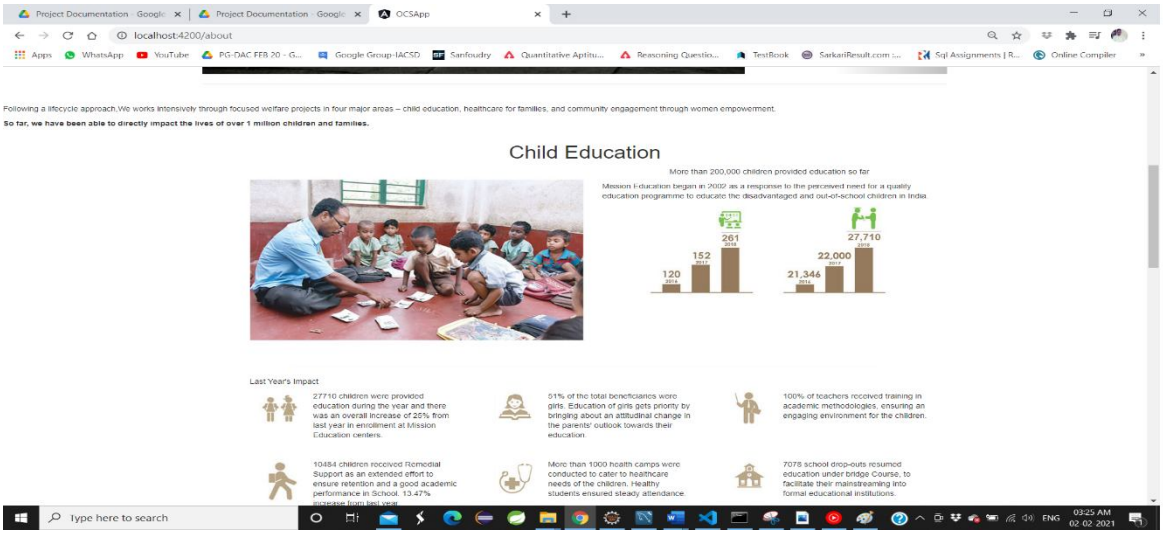


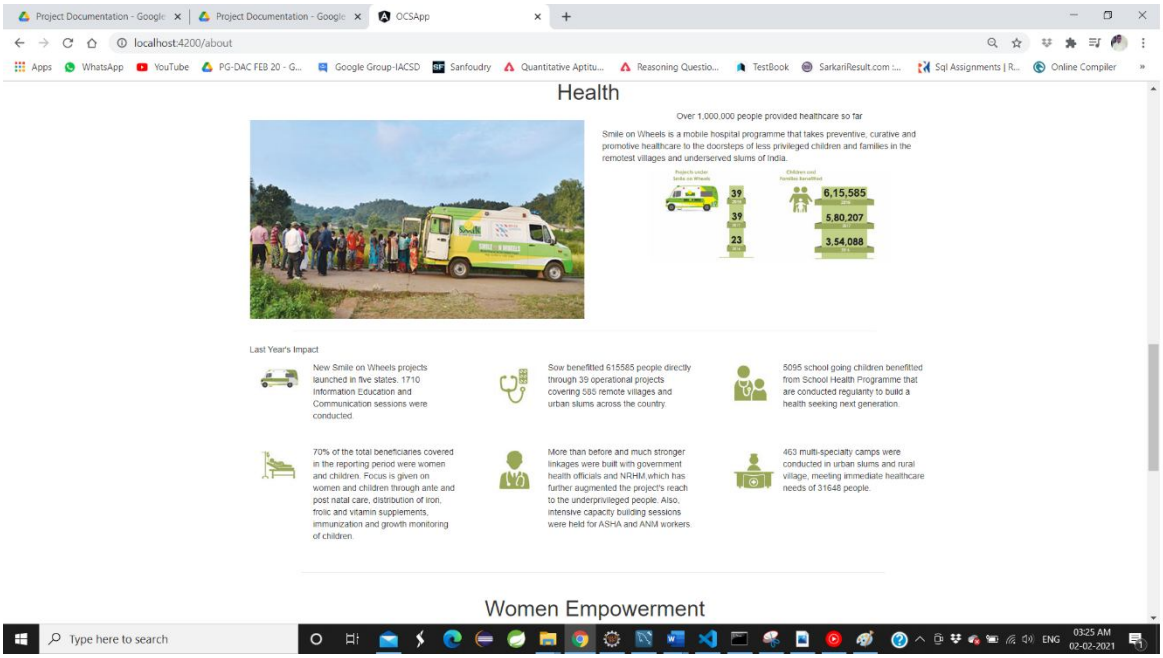


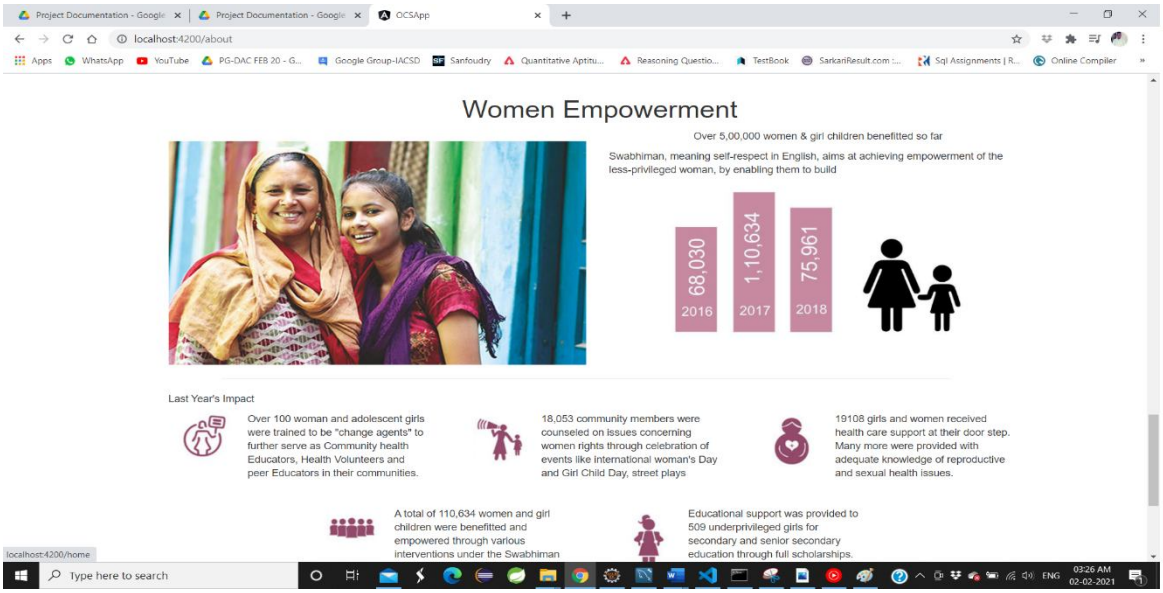


About Us page:

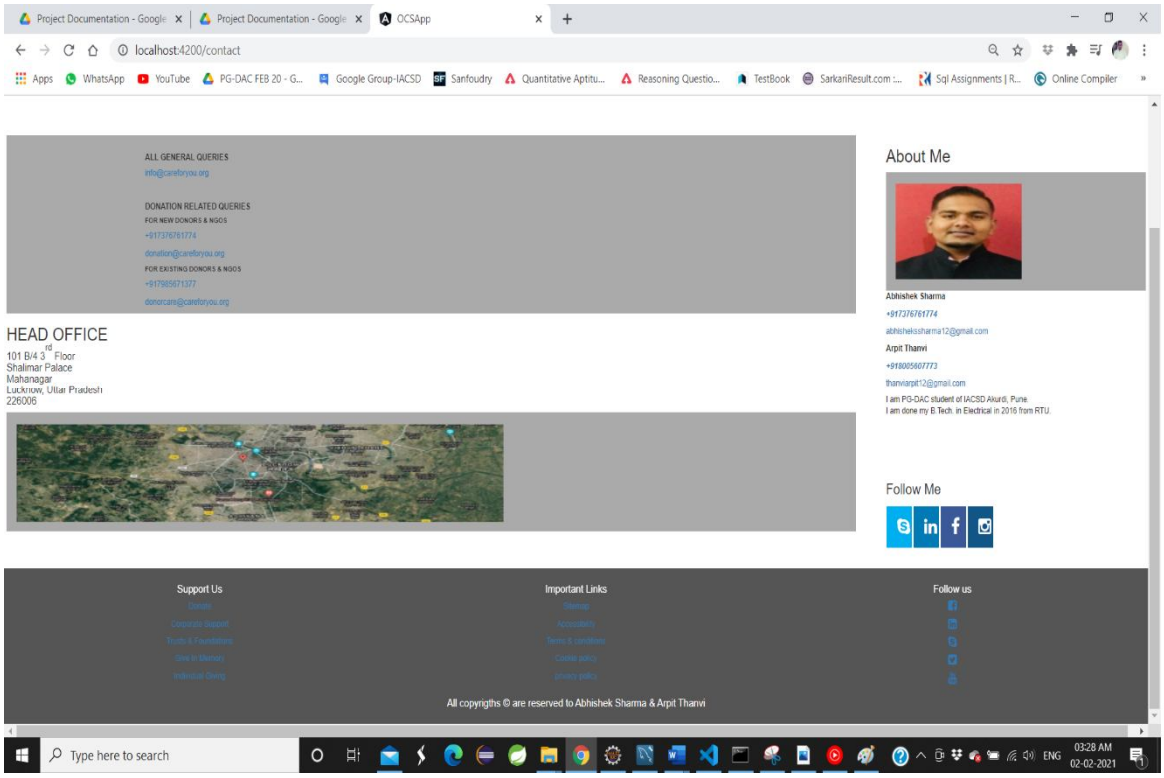



Project Documentation - Google
Project Documentation - Google
OCSApp
localhost:4200/about
Following a lifecycle approach,We works intensively through focused welfare projects in four major areas – child education, healthcare for families, and community engagement through women empowerment.
So far, we have been able to directly impact the lives of over 1 million children and families.
Child Education
More than 200,000 children provided education so far
Mission Education began in 2002 as a response to the perceived need for a quality education programme to educate the disadvantaged and out-of-school children in India.
120
152
261
21,344
22,000
27,710
Last Year's Impact
27710 children were provided education during the year and there was an overall increase of 25% from last year in enrollment at Mission Education centers.
51% of the total beneficiaries were girls. Education of girls gets priority by bringing about an attitudinal change in the parents' outlook towards their education.
100% of teachers received training in academic methodologies, ensuring an engaging environment for the children.
10484 children received Remedial Support as an extended effort to ensure retention and a good academic performance in School. 13.47% increases from last year.
More than 1000 health camps were conducted to cater to healthcare needs of the children. Healthy students ensured steady attendance.
7078 school drop-outs resumed education under bridge Course, to facilitate their mainstreaming into formal educational institutions.
Type here to search
ENG
03:25 AM
02-02-2021

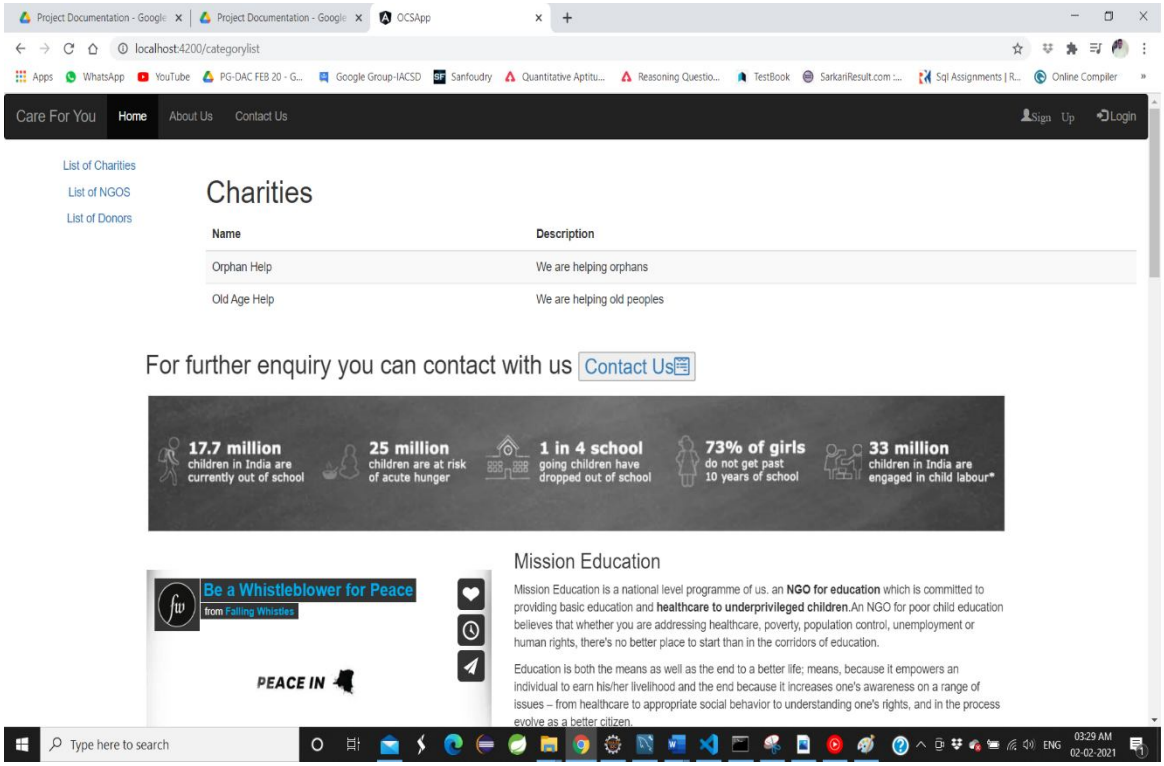

Project Documentation - Google
Project Documentation - Google
OCSApp
localhost:4200/about
Health
Over 1,000,000 people provided healthcare so far
Smile on Wheels is a mobile hospital programme that takes preventive, curative and promotive healthcare to the doorsteps of less privileged children and families in the remotest villages and underserved slums of India.
39
39
23
6,15,585
5,80,207
3,54,088
Last Year's Impact
New Smile on Wheels projects launched in five states. 1710 Information Education and Communication sessions were conducted.
Sow benefitted 615585 people directly through 39 operational projects covering 585 remote villages and urban slums across the country.
5095 school going children benefitted from School Health Programme that are conducted regularity to build a health seeking next generation.
70% of the total beneficiaries covered in the reporting period were women and children. Focus is given on women and children through ante and post natal care, distribution of iron, folic and vitamin supplements, immunization and growth monitoring of children.
More than before and much stronger linkages were built with government health officials and NRHM,which has further augmented the project's reach to the underprivileged people. Also, intensive capacity building sessions were held for ASHA and ANM workers.
463 multi-specialty camps were conducted in urban slums and rural village, meeting immediate healthcare needs of 31648 people.
Women Empowerment
Type here to search
ENG
03:25 AM
02-02-2021


Project Documentation - Google
Project Documentation - Google
OCSApp
localhost:4200/about
Women Empowerment
Over 5,00,000 women & girl children benefitted so far
Swabhiman, meaning self-respect in English, aims at achieving empowerment of the less-privileged woman, by enabling them to build
68,030
2016
1,10,634
2017
75,961
2018
Last Year's Impact
Over 100 woman and adolescent girls were trained to be "change agents" to further serve as Community health Educators, Health Volunteers and peer Educators in their communities.
18,053 community members were counseled on issues concerning women rights through celebration of events like International woman's Day and Girl Child Day, street plays
19108 girls and women received health care support at their door step. Many more were provided with adequate knowledge of reproductive and sexual health issues.
A total of 110,634 women and girl children were benefitted and empowered through various interventions under the Swabhiman
Educational support was provided to 509 underprivileged girls for secondary and senior secondary education through full scholarships.
localhost:4200/home
Type here to search
ENG
03:26 AM
02-02-2021

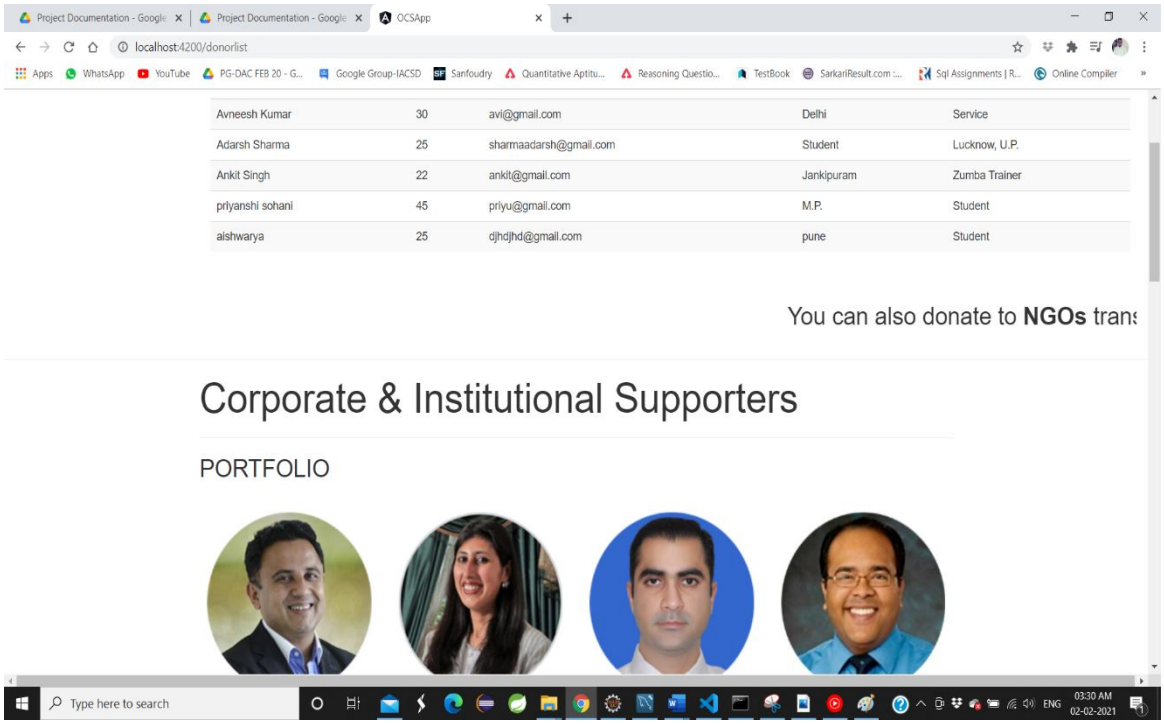
Contact Us Page:



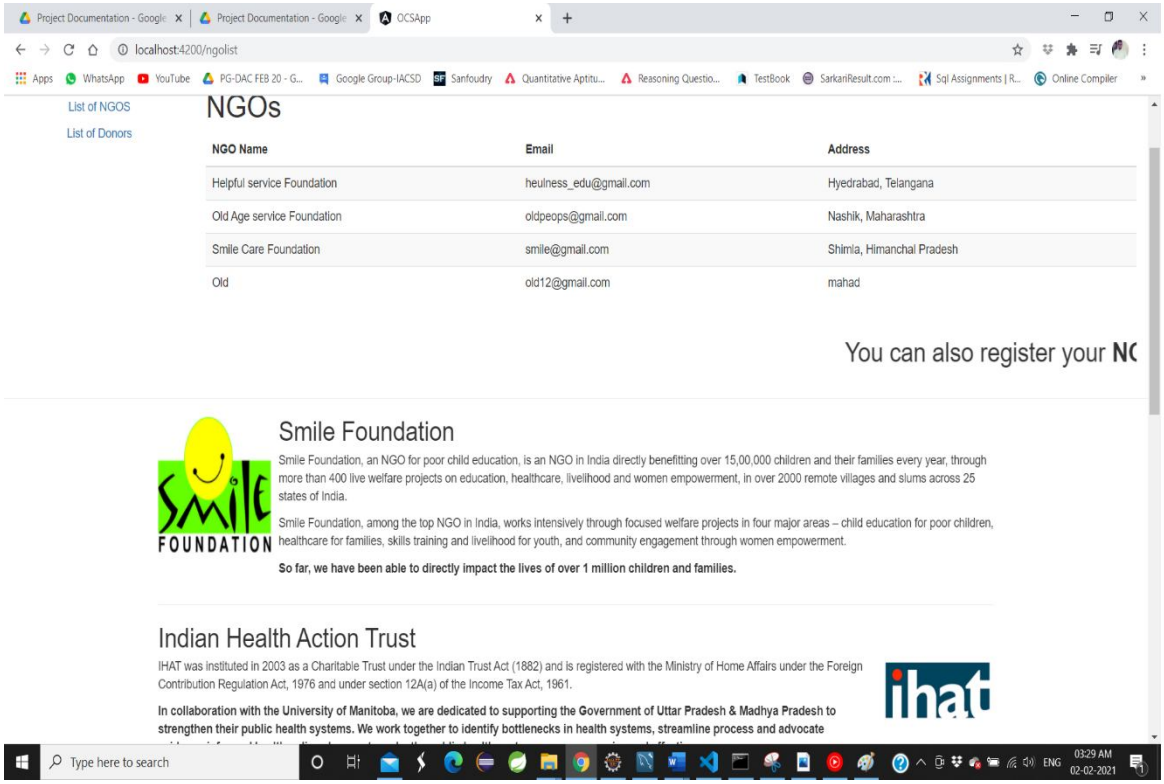
List of Category(Data From Database) :



List of Donor(Data From Database) :

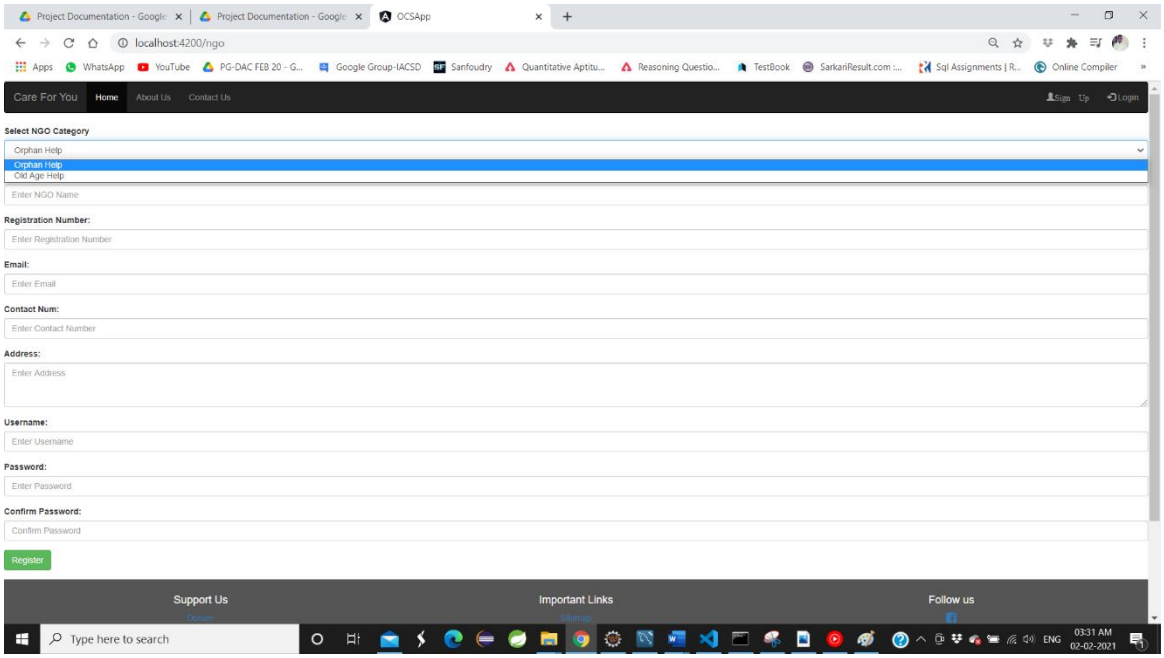


List of NGO(Data From Database) :

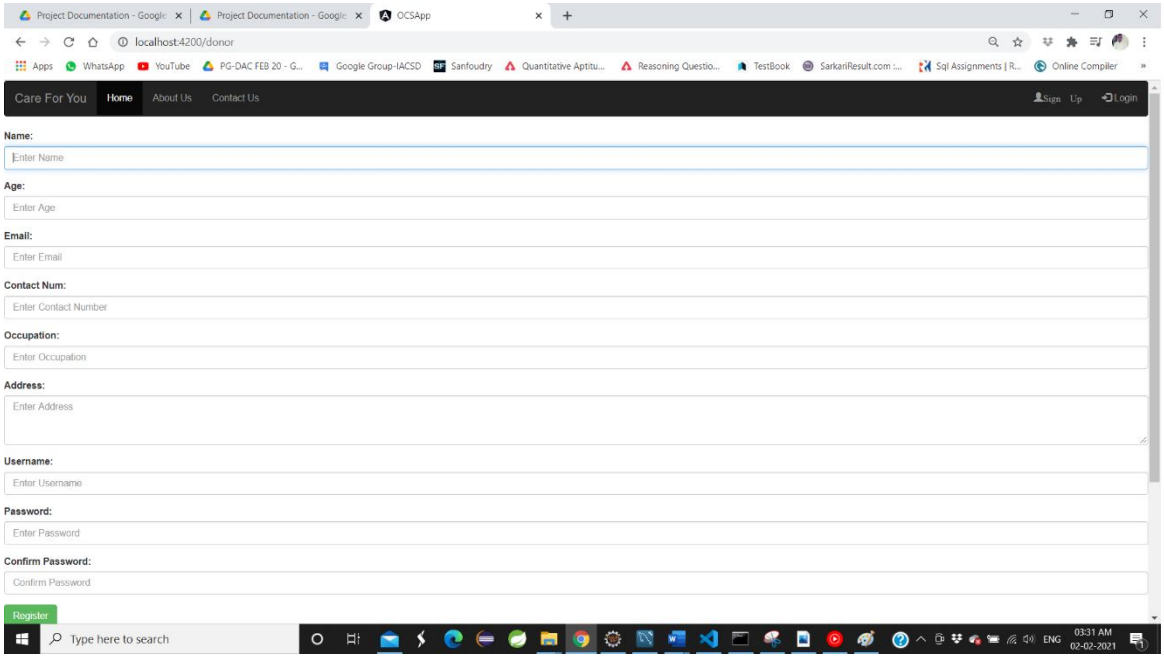


Registration:

NGO Registration:

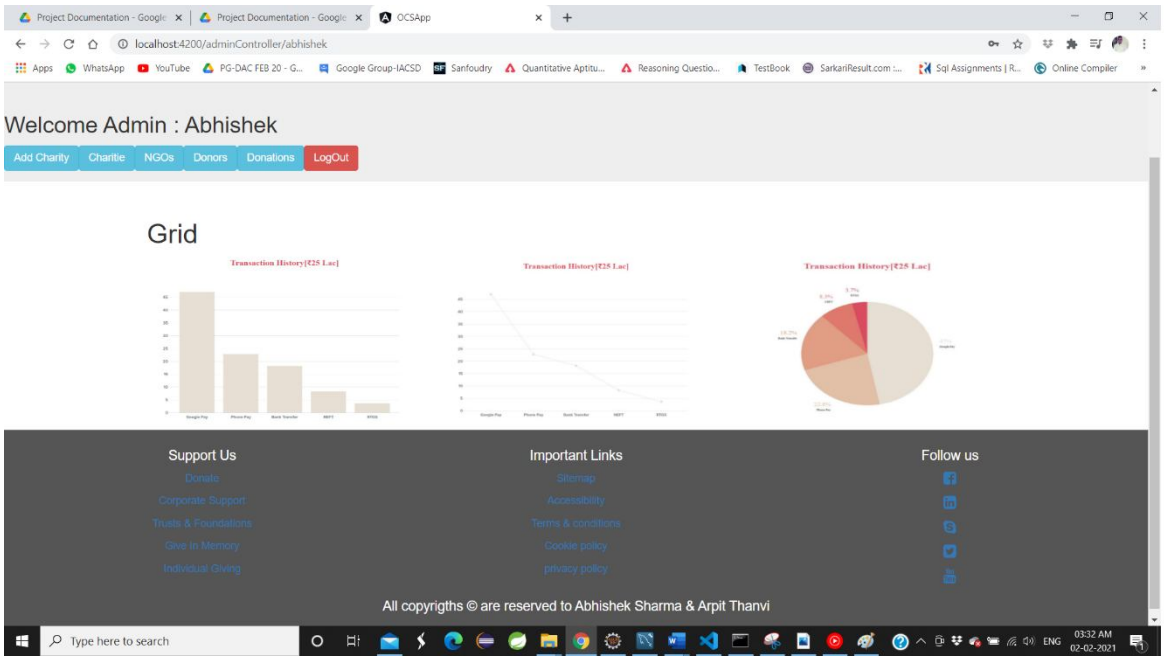


Donor Registration:

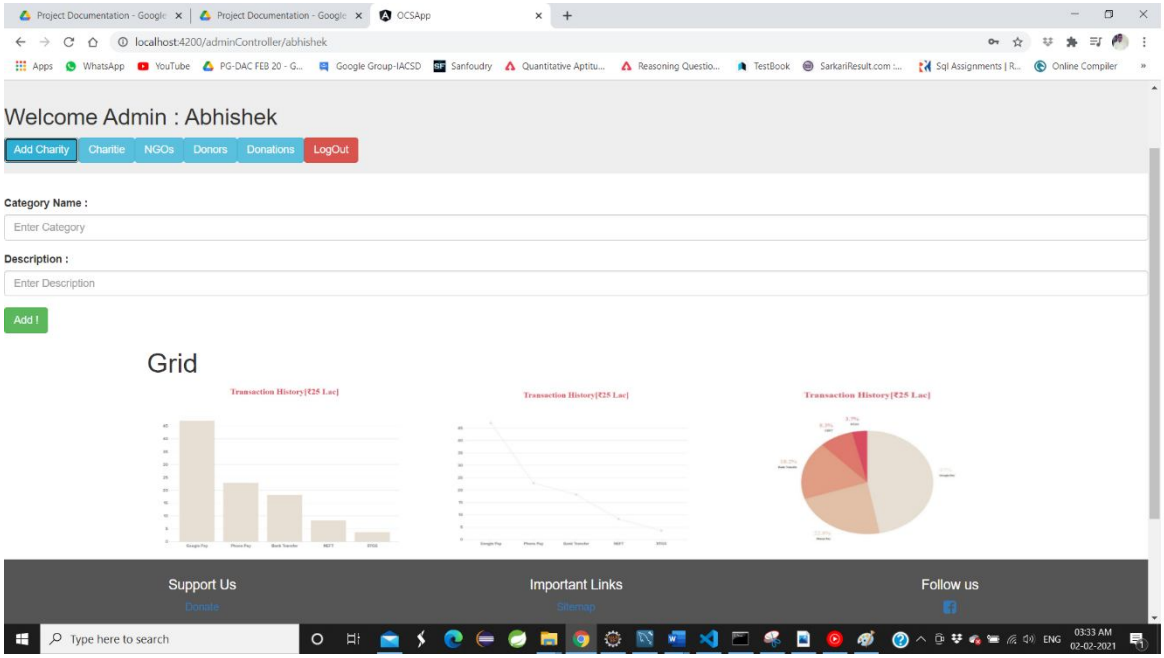


***All below pictures for normal visitor who are logged in

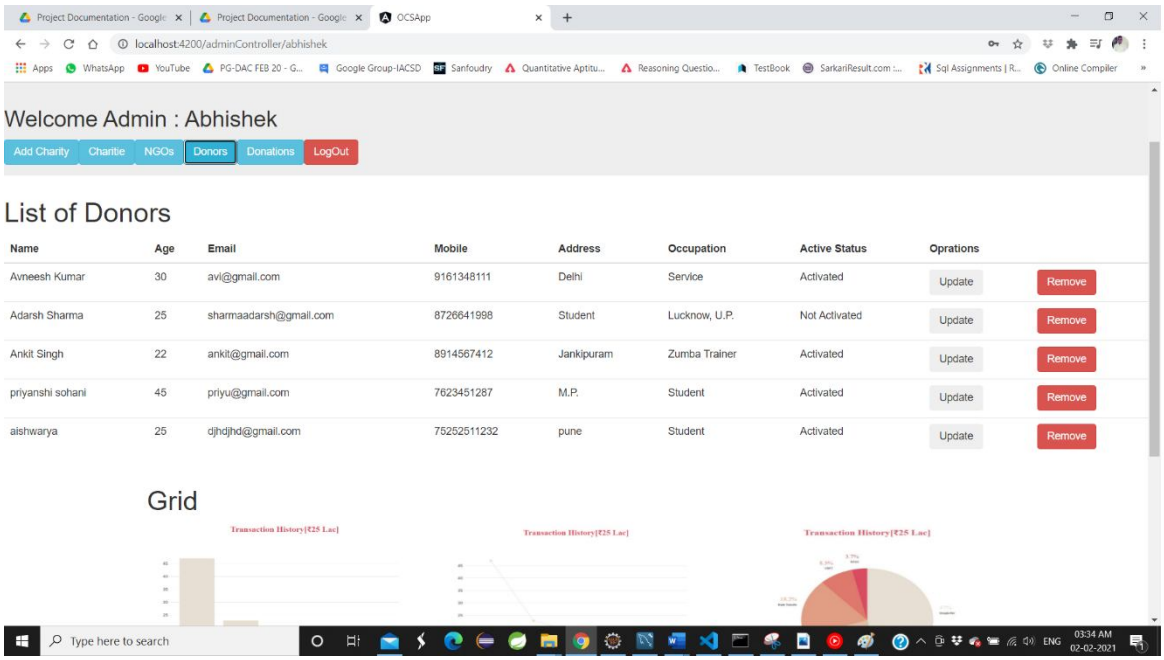
Admin Controller:



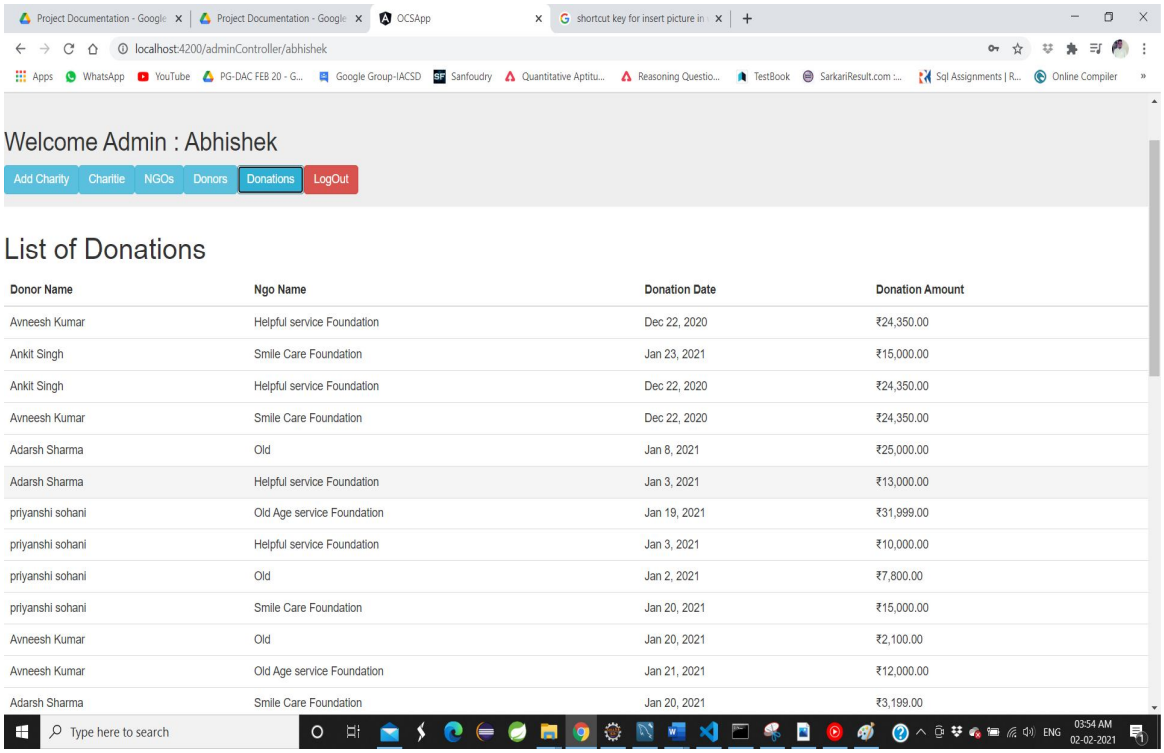
**Admin Can Add Charity Category



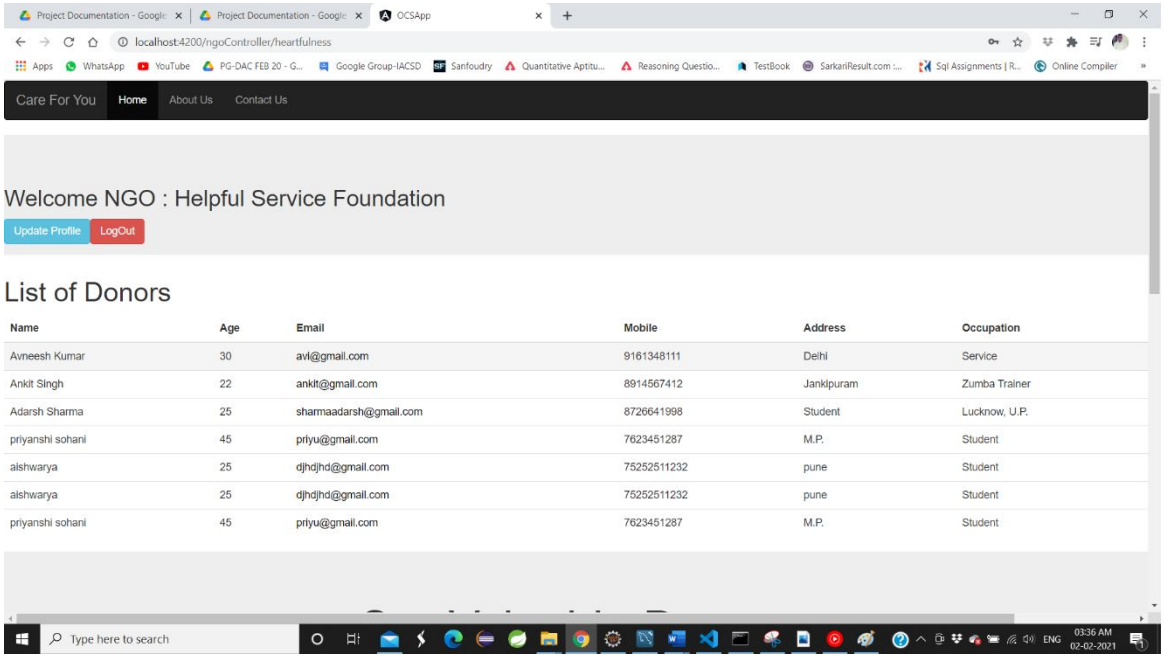
**Admin can UPDATE/REMOVE for charity category, NGO, Donor. I attached only one UI for category and donor is also same

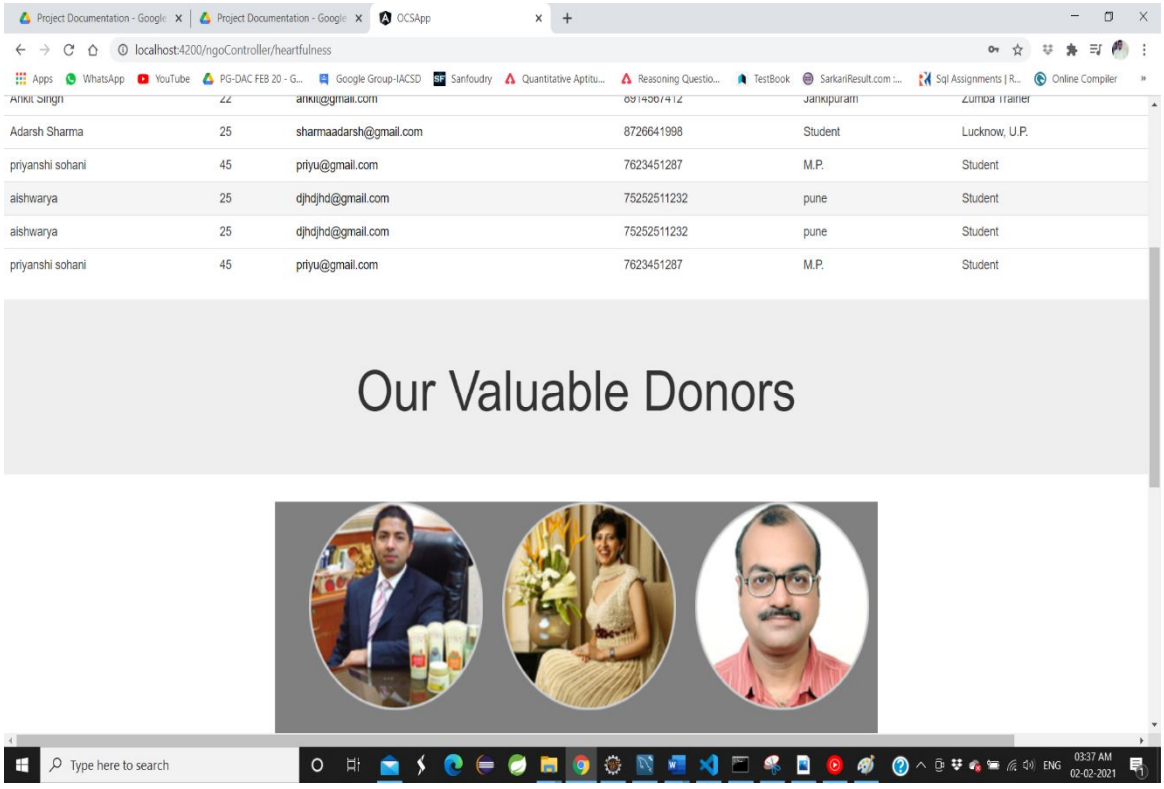


Donation View for Admin where he cant do any manipulation

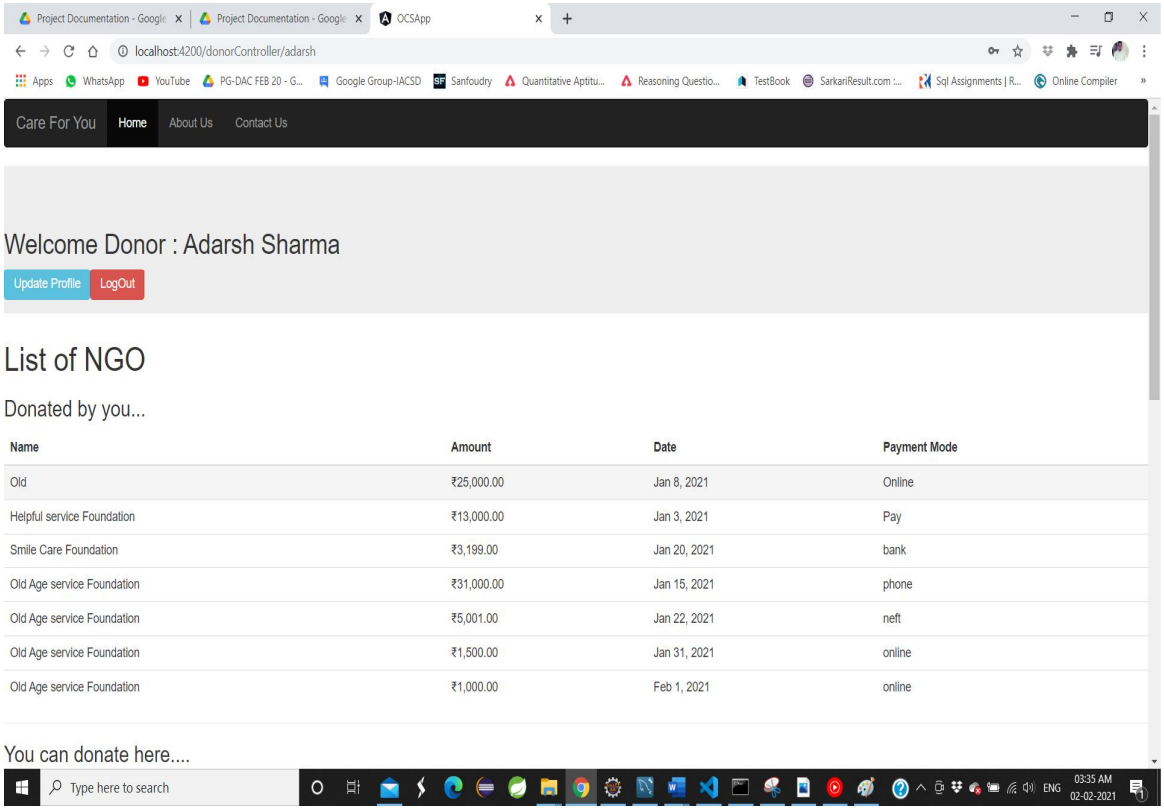


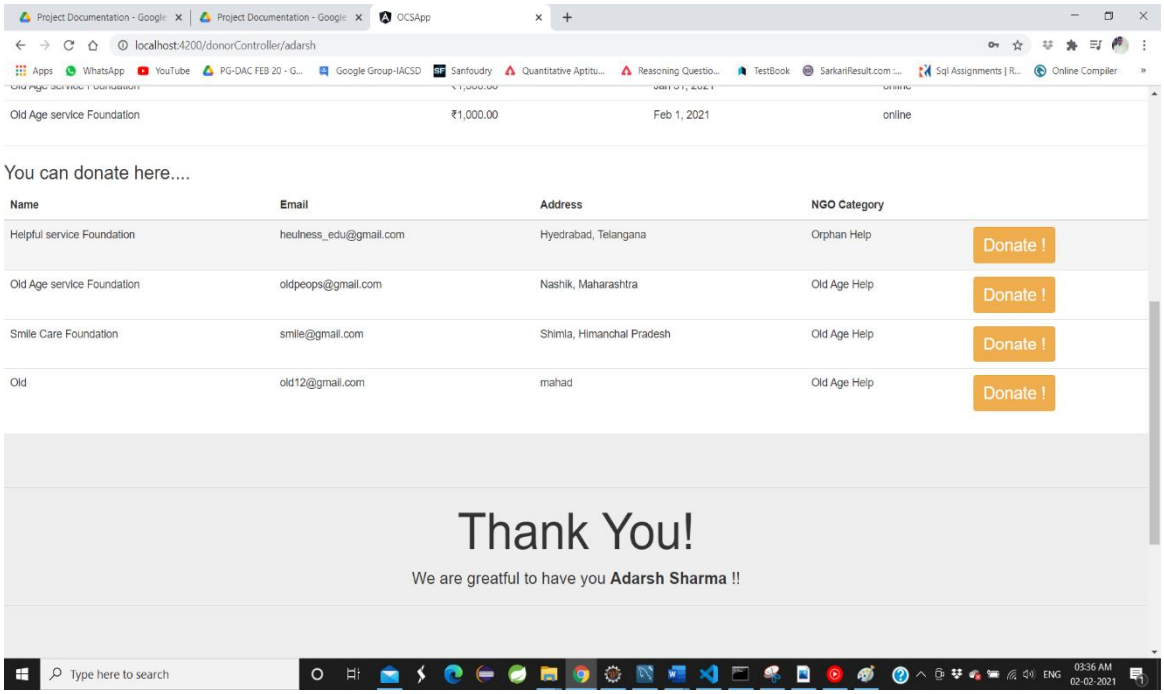
NGO Controller:





Donor Controller:





Conclusion:

Help directly. Donors shy away from spending on overhead and they may overestimate what portion of their donations goes toward fundraising and salaries (Bennett 2002). Charities can try to convince donors that their donations will help people directly by guaranteeing that their specific donations will not pay for overhead. They can also give donors the power to choose which cause their donation supports, which may increase donors' confidence in the impact of their gift (e.g. Li et al. 2012).. Make a meaningful contribution to solving a problem. Donors tend to avoid contributing to needs that are large in scope, or that will last a long time (Warren and Walker, 1991). They like giving to charitable campaigns that are very close to succeeding already (e.g. List and Price, 2003). But even if a charity is dealing with a difficult long-term issue that affects millions of people, they could still benefit by offering "small victories" to their donors.

Future Scope:

This system is developed mainly to target a charity group and its end-users but with just few changes the system can also be utilized in many real world applications. For our stakeholders the system has delivered its requirements but there are lots of improvements that can be done. One of those improvements is to add multiple payment methods for the donors. The system can also provide more privileges for members to make comments on an article. Many nonprofits websites advertise their web contents to other websites or clients using RSS (Really Simple Syndication). Our website can also be improved to implement RSS to provide newsfeed to those websites interested in promoting non-profit works. The system can also be improved to add forums and videos to attract more donors on the website.

References:

1. https://www.tutorialspoint.com/java/
2. https://www.javatpoint.com/java-tutorial
3. https://www.guru99.com/java-tutorial.html
4. http://www.w3resource.com/java-tutorial/
5. http://www.journaldev.com/7153/core-java-tutorial
6. https://www.udemy.com/java-tutorial/
7. https://www.w3schools.in/java-tutorial/intro/