

ABSTRACT

Our application revolves around helping the needy by connecting NGOs and common people. The donors shall be able to see a plurality of options by which they can donate. The NGOs will get the details of the persons wishing to donate via our application and thus a network is established between donors, people who aid the donors in donating (NGOs) and the actual needy people to whom the donated item is sent. Our application aims to bring about transparency, clarity and swiftness in the process of donation thus aiming to mitigate prevailing issues in whatever zone it is possible for us to do so.

Goals: -

1. To make a responsive, easy to operate, fast and efficient retrieval of information as per the user's convenience.
2. To make a database that is consistent, reliable and secure.
3. To provide correct, complete, ongoing information.
4. To develop a well-organized information storage system.
5. To make good documentation so as to facilitate possible future enhancements.

TABLE OF CONTENTS

S.No	Chapters	Page No.
1.	Introduction	2
2.	Problem Description	5
3.	Literature Review	10
4.	Research Methods	13
5.	System Analysis	16
6.	System Design	20
7.	Coding	30
8.	Testing	67
9.	Conclusion	70
10.	References	72

TABLE OF FIGURES

S.No	Figure	Page No.
1	Key Features	4
2	Hardware System	8
3	Software System	8
4	Resource Execution	9
5	Preliminary Search	11
6	Registration DFD	25
7	Login DFD	25
8	Details DFD	26
9	Architecture Diagram	27
10	Login Screen Snap:App	29
11	Drawer and Make Donation Screen Snap:App	29
12	Home Page:Website	30
13	Choose User Type:Website	30
14	Login Screen:Website	31
15	Signup Screen:Website	31
16	Type of Testing Diagram	70
17	Type of Testing Description Table	71

CHAPTER 1: INTRODUCTION

Donation System

Project Abstract

Our application revolves around helping the needy by connecting NGOs and common people. The donors shall be able to see a plurality of options by which they can donate. The NGOs will get the details of the persons wishing to donate via our application and thus a network is established between donors, people who aid the donors in donating (NGOs) and the actual needy people to whom the donated item is sent. Our application aims to bring about transparency, clarity and swiftness in the process of donation thus aiming to mitigate prevailing issues in whatever zone it is possible for us to do so.

Goals:-

1. To make a responsive, easy to operate, fast and efficient retrieval of information as per the user's convenience.
2. To make a database that is consistent, reliable and secure.
3. To provide correct, complete, ongoing information.
4. To develop a well-organized information storage system.
5. To make good documentation so as to facilitate possible future enhancements

Problem Definition

The 18% of NGOs lack in strategies and planning. 25% of Loss is caused by the lack of effective networking and communication system. 29% by inadequate financial provision and 22% by limitation on raising funds from gov. and other financial institutions. NGOs lack in maintenance so this application with help.

Why do we need this system, and what is the rationale behind it?

We have reviewed many websites and concluded that some websites are paid and some websites are specified for their own NGO's whereas some NGO's deals in specific product donation only. We are creating a platform where anyone can donate their things without paying any delivery amount or any extra amount enumerating to it they can deliver to any NGO's nearest to them also they can donate multiple things.

I was part of three NGOs earlier and worked as a volunteer there , we conduct frequent drives such as clothes drive food drive and many more . we are requested by our seniors in the NGO to bring old clothes food shoes to distribute in the drive me and my team members tried our best but were not able to collect a good amount of items to distribute at that time I thought about this system which can help poor as well as I know there are always volunteers for an NGO but they lack in collecting items for the drive so this platform will help to resolve this problem and this was our motivation to bring a revolution for the poor by distributing old items as I know NGO volunteers can distribute the items in a more systematic way so that it can fulfil the requirement of as many people as possible..

、 、 、 、 、

Key Features

Features	Application
Performance	High
User-Experience	High
Frameworks Used	React JS
Backend	MongoDB
Cross-Platform Support	Yes

Fig1

Benefits

- **Strategic Planning:** Although some organizations may already be experienced and their employees are familiar with their operations, coaching and training every now and then can still be helpful. In every situation, employees and volunteers will inevitably pick something up and be able to apply it for further development of the organization.
- **Maintenance:** Given the number of communities in developing countries and the limited funding sources for non-profit organizations, it's quite tempting to move on to the next project to provide aid for other communities who are equally in desperate need of help as well. Thus, it would be best to make sure there is enough resources and manpower to both maintain an existing project and begin a new one. It will be better to successfully accomplish just a few projects with lasting impact rather than accomplishing a lot but with temporary results. In the end, a slow but sure process is still more effective as compared to handing more than your capacity

Scope of the project

The scope of a Donation System can involve, among other things, valuing NGO's, building a network, and bring about transparency, clarity and swiftness in the process of donation thus aiming to mitigate prevailing issues in whatever zone it is possible for us to do so.

Conclusion

Chapter 1 contains the system's background research. The system's purpose, as well as its goals and objectives, have been determined through study. After identifying the system's aim, the developer came up with the benefits of the system's intended end-user and how the system may assist its users. Following that, the developer cultivated the ideal features that would be incorporated in the suggested system.

CHAPTER 2: PROBLEM DESCRIPTION

2.1 Current problem Description

The 18% of NGO's lack in strategies and planning. 25% of Loss is caused by the lack of effective networking and communication system. 29% by inadequate financial provision and 22% by limitation on raising funds from gov. and other financial institutions.

2.2 Problem Importance and Justification

First, we will create an application for NGO's where they can store their data and NGO details digitally at one place. Since it is a mobile application so it can be run on any (Android or IOS).

2.3 Proposed Solution

2.3.1 Strategic Planning: Although some organizations may already be experienced and their employees are familiar with their operations, coaching and training every now and then can still be helpful. In every situation, employees and volunteers will inevitably pick something up and be able to apply it for further development of the organization.

2.3.2 Maintenance: Given the number of communities in developing countries and the limited funding sources for non-profit organizations, it's quite tempting to move on to the next project to provide aid for other communities who are equally in desperate need of help as well. Thus, it would be best to make sure there is enough resources and manpower to both maintain an existing project and begin a new one. It will be better to successfully accomplish just a few projects with lasting impact rather than accomplishing a lot

2.4 Nature of Challenge

During the development of this software, we will face a number of significant problems, including:

- **Database:** To put the MongoDB principles and ideas into practice in order to store data.
- **React JS:** The most difficult aspect of constructing this software is that we must start from scratch.
- **Save data:** To save NGO's data with security is also a major challenge.

2.5 Feasibility Study

A feasibility study investigates and assesses a proposed project to see whether it is technically, economically, timely, and operationally feasible. The goal of a project feasibility study is to see if the proposed project can be completed in the time provided, with the resources available, and within the budget allotted. After the feasibility study is completed, a decision is taken on whether or not to proceed with the project. The feasibility study is broken down into four stages:

2.5.1 Technical Feasibility

The purpose of technical feasibility testing is to see if a project can be completed within the restrictions of available resources. The technical viability of a project is determined by taking into account the project's technical requirements as well as the technical resources required for its effective and efficient implementation.

Hardware System

S. No.	Hardware System
1.	Processor- core i3
2.	RAM (Memory) - 4 GB and more
3.	Monitor –Lenovo
4.	Pen drive , Mouse, Keyboard
5.	System Type- 64bit OS
6.	Hard Drive– More than 10 GB
7.	Accessories – Internet connection.

Fig2

Software System

S. No.	Software System
1.	Visual Studio Code
2.	Database Technology – MongoDB
3.	Operating System- Windows 10
4.	Documentation and Presentation tool - Microsoft Office 2010

Fig 3

Resource Execution

S. No.	Resources for Execution
1.	Operating system - windows 10
2.	CPU: 600 MHZ
3.	RAM: 100 MB
4.	Disk space: 50 MB
5.	A good data/internet connection for customers

Fig4

2.5.2 Economic Feasibility

The primary benefits and costs of a project's development. The project is regarded economically feasible if the anticipated benefits equal or surpass the system's expected expenses. The development component will be less expensive since the recommended system programming environment is developed on the web, which is an open source platform. In addition, every piece of software or programme that is generated must be tested, which may be done for a low cost. The only outlay required would be for a physical device to test the system.

2.5.3 Schedule Feasibility

A project's timetable feasibility is checked to see if it can be finished within the allotted time. Dates are set for each step of the project, and it is necessary to guarantee that the project is completed on time and according to the schedule. The planned feasibility of the project is high if it can be finished on time.

Operational Feasibility

Because the project's main purpose is to ease the issues that people face in today's environment, it will almost certainly be operationally possible.

2.6 Conclusion

In Chapter 2, the issues found in the current area of study, notably the challenging environment, are examined. All of the challenges that the ordinary user faces have been identified by the developer. The problems have been documented, as well as the explanations behind each problem. The developer's next major goal was to provide adequate solutions to all of the problems that had been uncovered. Each problem's solution, as well as the thinking behind it, has been made public for the same reason. The feasibility study was successful, allowing the developer to move on with the project

CHAPTER 3: LITERATURE REVIEW

Introduction

A literature review is a summary of prior study on the topic conducted by other academics. It is based on previously published books, journals, and other research papers and is intended to serve as a starting point for further research.

Preliminary Research

The creator had the concept while trying to solve the problem of discovering information about personnel, goods, and sales in companies or industries.

Topic	Book Referred
Donation System	1. https://www.udayfoundation.org/delhi-donate-old-clothes/ , The Uday Foundation (NGO). 2. International Journal of Scientific and Research Publications, 3(3), 520–530
Database	Using Scaling MongoDB by Kristina Chodorow, Released August 2011 Publisher(s): O'Reilly Media, Inc
React JS	Learning React: Functional Web Development with React and Redu by AlexBanks 2017

Fig 5

3.4 Domain Research

Because the project domain changes depending on the circumstances, domain research is essential, and after this is accomplished for a specific module, we may work on the analysis and design of that module to make it easier to use. Throughout the investigation, the researchers saw that students face a variety of obstacles when it comes to getting projects.

Researchers must deploy technology to lessen difficulties experienced by businesses in order to have a thorough understanding of user needs and critical features of the system.

Software

A Windows-based application is software that runs on a computer that runs the Microsoft Windows operating system. These apps are meant to be installed and operated locally on a Windows computer, and they are typically cross-platform compatible.

Market Analysis

The application may be used as a complete retail software package since it contains two user interfaces (User and Admin), each with its own set of functionality. As a consequence, market analysis is for those who have an issue to solve.

Examining the Literature Review Critically

The creation of the suggested software for users is investigated in order to determine what characteristics are required by the end user in order to improve their firm's finding information. The literature review lays the framework for future study by providing an overview of the proposed system. The systems that are comparable to the proposed system have been researched to establish a baseline for the system; what challenges the developer could encounter, and what features are impossible to implement.

Important Success Factor

The suggested system's important capabilities are the Backup Facility in Remote Database, Push Notification, and Save Projects features that it would provide to its customers. To provide a useful system to the end user , the developer must complete all of the features discovered during the literature review , secondary research that must be implemented

Conclusion

The third chapter examines existing research to establish a baseline for the intended project. As part of the literature research, existing systems were studied to obtain experience in the study's subject; what features are new on the market, and what features the developer may integrate in his new system. The chapter also includes data and resources from the literature study that are now being explored or will be studied in the future to obtain data and information about the top

CHAPTER 4: RESEARCH METHODS

Primary Research

Primary research is the study from the actual world directly on a certain subject. It's done to clear up any misunderstanding or uncertainty that a developer could have after conducting secondary research. This is accomplished by directly posing such queries to forum participants and taking into account their responses. Create a network and utilise this research criterion to further the project's progress.

Secondary Research

Secondary research is essentially an improved version of primary search with additional elements. It refers to the gathering of information from previous records, books, and research papers that have been verified by a third party. It is carried out in order to learn from mistakes through a process and by offerings made on the connected topic.

Technical Research

This refers to the research's technical aspects, and it simply indicates that the study is concerned with the programming language, database, and technique. Technical research for the projected project is one of the significant and crucial aspects that might impact the project's implementation to various phases, and it is done before the system design and implementation phase begins. How common is research that analyses and evaluates several platforms, databases and technologies in order to determine which would be picked for the platform.

4.4 Hardware

The most typical criteria for the proposed system or hardware, including extra quality elements that do not rely on a lengthy procedure. Hardware is made up of physical components that are needed to manage operating systems.

The hardware requirements of the system is discussed below:

Processor	i3 or above
Hardware	4 GB or above
Memory	256 MB RAM or above
Cache Memory	128 KB or above
Pen Drive	5 GB

4.4 Software

Software is not a physical part but it is a soft part that is set of instructions and we cannot feel it. It is used to provide the optimal functioning of a software to the hardware and operating system.

Operating System	Windows
Framework used	React JS
DataBase	MongoDB
Basic Text Editor	Visual Studio Code

Back-end

- We have use this as it provides and efficient solution for major databasetechnology.
- Space management and a large database.
- Industry accepted standards.
- Manageable Security.
- High availability.

4.8 Functional requirements:

4.8.1 Data Collection

Data collecting, often known as information gathering, is the most important part of research. The most essential purpose of data collection is to guarantee that the details are rich in informative scheme and that reliable data is extracted from the whole data set for statistical analysis.

4.8.2 Data Visualisation

Data visualisation is the graphic representation of information and data depending on the required information so that data driven decisions can be made for research. By using visual elements like charts for statistics, graph for up down values and maps for details data visualisation tools that view the information that provide an accessible way to see the pattern of data and basic data to understand trends.

4.8.3 Data Labelling

Data labelling maps the data in the better format which takes much time and effort to build a purposeful document as data sets is sufficient for machine learning thousands of records to be labelled.

4.8.4 Data Selection

Data integrity may be obtained by the process of picking all information vegetable Tata for a research endeavour. Following data gathering, the data analyst divides the data into a number of categories with comparable qualities in order to answer the given issue. The data from the whole set was chosen since it has properties that must be considered while creating a prediction model in order to get a good picture of it.

4.8.5 Modelling

After pre-processing of the information the collected data and split it into three subsets, which can improve data for proceeding with model training. This process is to get credit analysis with entails “feeding” algorithm. Not an algorithm choice is given by supervised and unsupervised learning that give information about the style of grouping of complete data which will process accordingly.

4.8.6 Conclusion

The research methodologies are the focus of Chapter 4. In this project, the research is performed through a primary search and secondary research, in which technical research was combined with content to arrive at a final conclusion of user needs, technologies, and some of the challenges that were encountered, as well as the tools that were utilized.

CHAPTER 5: SYSTEM ANALYSIS

5.1 System Analysis

System analysis refers to research that is carried out with the goal of examining a system and its components in relation to an issue in order to determine its goals. It's a problem-solving strategy. It significantly improves system performance and guarantees that all of the system's components work together to achieve the goal for which the system was created. Analyses define what the system should be able to achieve.

5.2 Principles

- Understand the issue because it is where the analytical model was born.
- Use numerous perspectives of requirements, such as building data, function, and behavioural models, to ensure that different elements of analysis are completed without mistakes.
- Create a prototype that allows a user to visualize how human-machine interaction will work.
- Work to eliminate ambiguity.

5.2 The components of system analysis

In general, the system is separated into multiple portions dependent on how it functions, but every system has three primary components: input, processing, and output. After the analysis stage, which is a separation of a substance into simple and many tiny components to simplify the research and make the implementation and comprehensive examination easier, system analysis begins during the feasibility study of the system. The comprehensive inspection gives the exact data needed throughout project design to guarantee that all of the client's needs are met, as well as the developer's pleasure. The feasibility study is the foundation for the inquiry or research undertaken throughout the analysis phase of the entire project.

5.3 Feasibility Study

The feasibility study presents a set of solutions that can achieve the project's aim even while addressing one or more of the project's difficulties conceptually. In reality, the solutions are generally descriptions of the System and its assessment of whether or not it is beneficial to continue with the project. The practicality of such solutions is assessed. Such an assessment reveals flaws in the initial objectives. This process is repeated when goals are tweaked and existing solutions are assessed. Typically, a feasibility analysis evaluates a number of project alternatives, one of which is chosen as the most suitable option and the other of which is pursued. These options must also be considered in a Broadway-style manner without devoting too much resources to the entire process.

5.4.1 Economic Feasibility

The cost-benefit analysis is carried out in economic visibility, which reflects the actual costs and benefits based on the production. An assessment of development costs in relation to the eventual benefit received from the product development system.

5.4.2 Cost and benefit analysis:

The feasibility study of the system follows the analyses stage following the development of the software, and Alice starts during the feasibility study of the system after the analyses step. Profits may be monitored, which aids in providing a concise overview of the costs involved in evaluating other investments in the form of a better working environment.

These are various kinds of costs and benefits:

1 Supply cost- This is an important component that is defined as the variable cost plus the real cost that is proportional to the number of use items.

2 Personal expenses- This category includes money spent on people, salaries, and security costs associated with the construction of a system at a fair cost.

3 Operating Costs- These are all of the expenditures, both selling and purchasing, that are necessary for the system's day-to-day operations, services, and functioning.

5.4.3 Operational Feasibility

Operational feasibility is primarily concerned with questions such as whether the system will be used at various phases of development or implementation. The most important question that arises throughout the system's testing for operational feasibility is as follows:

- Is the project supported by management?
- Will there be a loss of information accessibility?
- Will the system have an impact on the customers?
- Are the users satisfied with the projects?
- Is there an improvement in overall performance?

5.4.4 Legal Feasibility

The practicality of the system is directly tied to the determination of any contravention, violation, or obligation that might come from its development. Legal feasibility states that the software used in the project should be original, obtained from legal authorities in accordance with security regulations, and that they have the right to use it because pirated software.

5.5 Feasibility Analysis

5.5.1 Technical feasibility

The project is technically possible since it is a formal procedure that can be constructed using current technology before going live. It's a window-based software that makes use of VScode to provide relevant data on various topics. Stockpile's streaming system requires existing technology to track projects, therefore it is theoretically possible.

5.5.2 Economic Feasibility

The project is financially possible since it is a step toward the business growth of a model, and the project's cost is solely engaged in the hosting of the project. As the number of data samples grows, so does the amount of time it takes to process them. In that case, a more powerful processor may be required.

5.5.3 Operational Feasibility

As a user with minimal computer and internet expertise, the project is operationally apparent.

5.5.4 System Life Cycle

System life-cycle refers to the international process of establishing and sustaining systems. The term "system development life-cycle" refers to the grouping of diverse operations in various ways. It is a system that we build during various phases aids in the creation of a system project plan that contains system details, as it provides a list of processes and sub processes required for system development, as well as information on how to handle problems that may arise during the process. To put it another way, numerous processes are grouped together and referred to as a system development life-cycle. The system development life-cycle refers to the software development life-cycle in terms of system analysis and design vocabulary.

The following are the major stages of the life cycle:

- Feasibility study
- System study
- Implementation
- Maintenance
- Coding
- System analysis
- System design Testing

CHAPTER 6: SYSTEM DESIGN

Introduction

The design phase begins when the developer completes the research and analysis phase, after which the system design is developed to continue the process. It includes physical modelling utilising software engineering tools and processes, as well as the creation of the system's design. The transformation of the usage requirements

System Design

The blueprint of the software is the design document for the system that we will construct during this phase. It explains how the customer's problem will be solved when the blueprint steps are completed. Change the answer to a difficult issue is seldom found on the first try; hydration is almost always necessary since multiple designs are formed before all of the processes are completed. Any approach or design must support and guide the partitioning process so that the results are as independent as feasible from one another and can be readily merged for the overall answer to the given challenge.

This is also true for software design before beginning the process and working, we need a blueprint. As a result, any design approach, method, or language that is used must be adaptable. The problem is subdivided into a problem that is a self-contained and simple combination of their answers that lessen the problem's complexity.

The specifications for the system to be developed and taken into account for implementation are the input to the design face, and these specifications describe the good system design. Both functional and object-oriented elements are interconnected through a series of processes to produce a successful outcome. Specification languages are used to specify the design.

Input Design

The input design is the connection that connects the information system to the user's reality. Determine the inputs, validate the data, minimise data entry, and give multi-user facility are all part of the input design. The most prevalent source of data processing failures is inaccurate inputs. Input design can be used to control the errors made by data entry operators. In the input design, user-generated inputs are translated to a computer-based format. The input data is gathered and arranged into groupings of data that are related. Following that, the proper input media for processing is chosen. All input data is checked, and if any data violates any requirements, a notification is displayed to the user. If the data meets all of the criteria, it is moved to the proper database tables. Employee information will be entered at the time of registration for this project. For this goal, a user-friendly and simple-to-use page has been created. When exceptions arise, the design is done in such a way that consumers receive suitable notifications.

Output Design

The most essential and immediate source of information for the user is computer output. Because the output must be efficient, the output design step is particularly crucial. A user-friendly and efficient output design increases the system's interaction with the user and aids decision-making. Allowing the user to see a sample screen is critical since the user is the final arbiter of output quality. The specified notifications are the system's output module.

Database

MongoDB is an open-source document-oriented database that is designed to store a large scale of data and also allows you to work with that data very efficiently. It is categorized under the NoSQL (Not only SQL) database because the storage and retrieval of data in the MongoDB are not in the form of tables.

System tools

This chapter discusses the many system tools that were employed throughout the project's development.

React JS

The main objective of ReactJS is to develop User Interfaces (UI) that improves the speed of the apps. It uses virtual DOM (JavaScript object), which improves the performance of the app. The JavaScript virtual DOM is faster than the regular DOM. We can use ReactJS on the client and server-side as well as with other frameworks. It uses component and data patterns that improve readability and helps to maintain larger apps.

Data Flow Diagram

A Data Flow Diagram (DFD) is a flowcharting tool that may be used for organised analysis and design. A data flow diagram (DFD) is a diagram that depicts the movement of data as well as the processes that modify or convert it within a system. This network is built with a collection of symbols that don't imply a physical implementation. Its goal is to identify important transitions and explain system needs. As a result, it is the beginning point of the design phase, when the required specifications are functionally decomposed down to the smallest degree of detail. The logic of an information-oriented or a process-oriented system flow-chart may

be abstracted using DFD. DFDs are often referred to as logical data flow diagrams because of this.

External entity

A data flow's source or destination is an external entity. On a data flow diagram, only the entities that generate or receive data are shown. A rectangular box is used as the emblem.

Process

Within the system, a process depicts the transformation or modification of data flow. An oval shape is employed as the emblem.

Dataflow

The data flow diagram depicts the flow of data from a source to a destination. A line represents data flow, with arrowheads indicating flow direction. Written, spoken, or electronic information always flows to or from a process. Each data flow can be identified by its head and tail processes or data storage.

Registration DFD:-

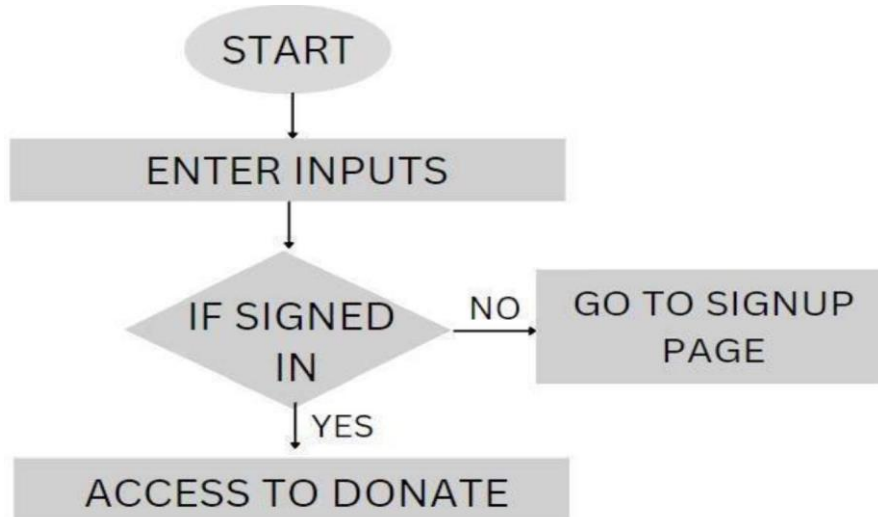


Fig 6

Login DFD:-

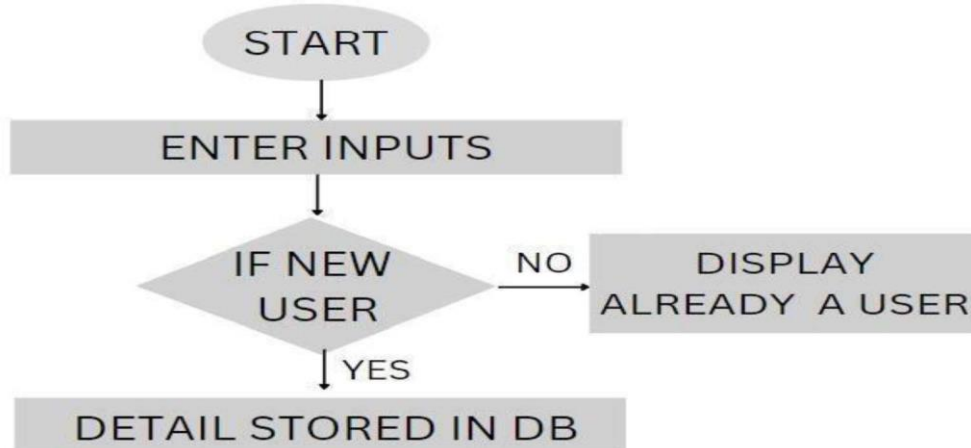


Fig 7

Details DFD:-

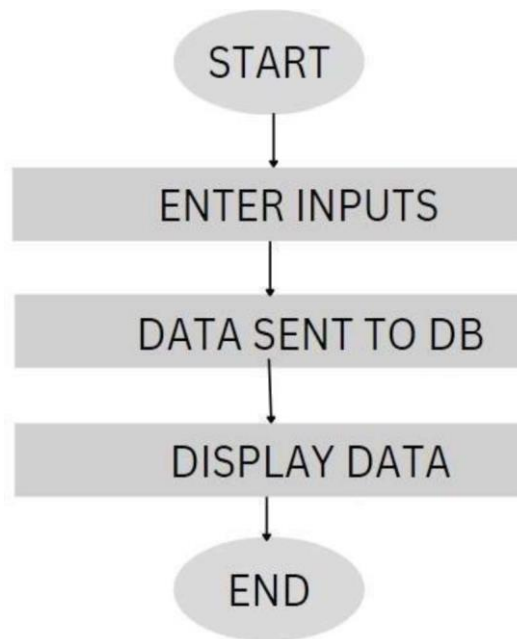


Fig 8

CHAPTER 7: IMPLEMENTATION AND ARCHITECTURE DIAGRAM

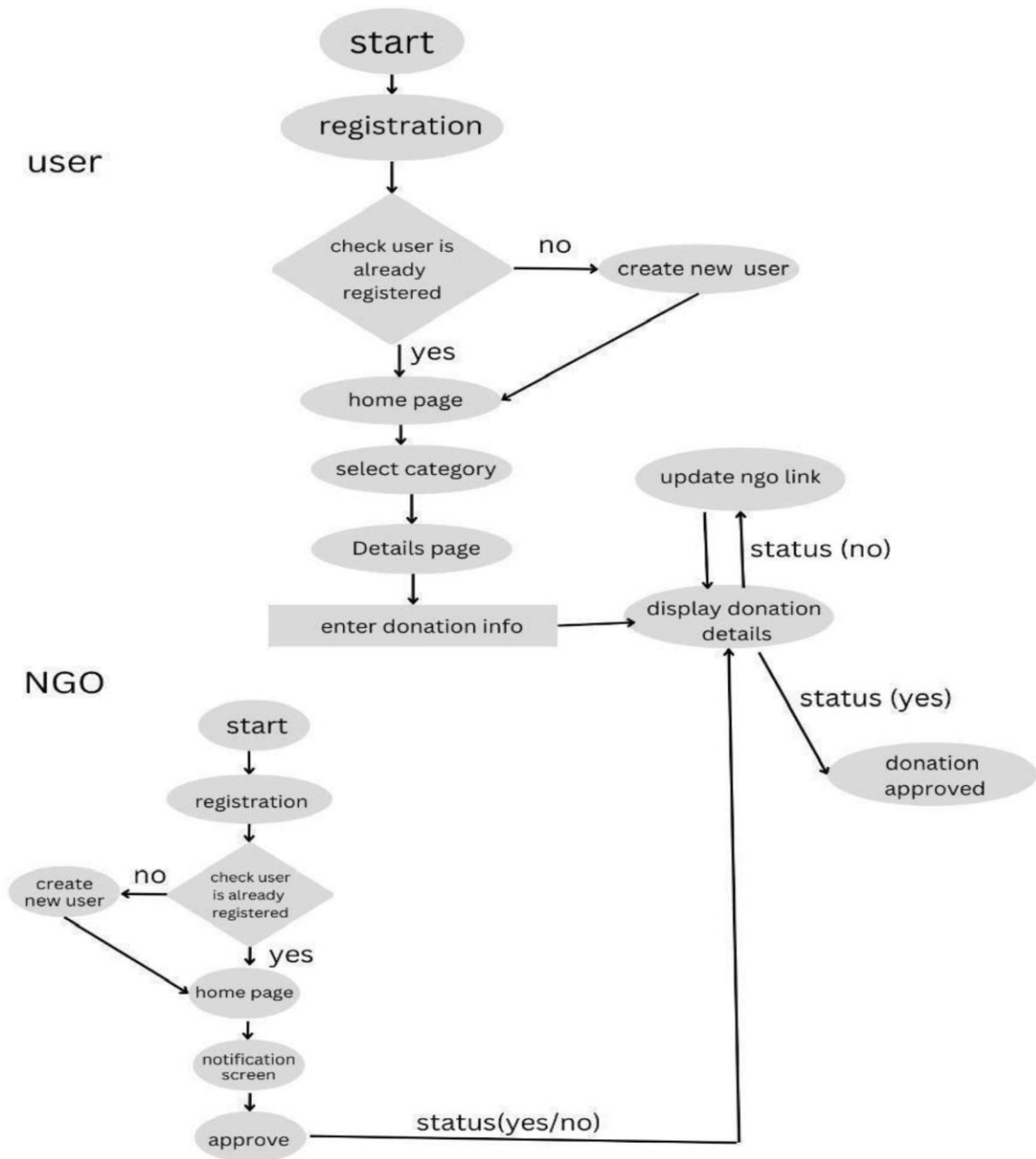


Fig 9



Fig 12

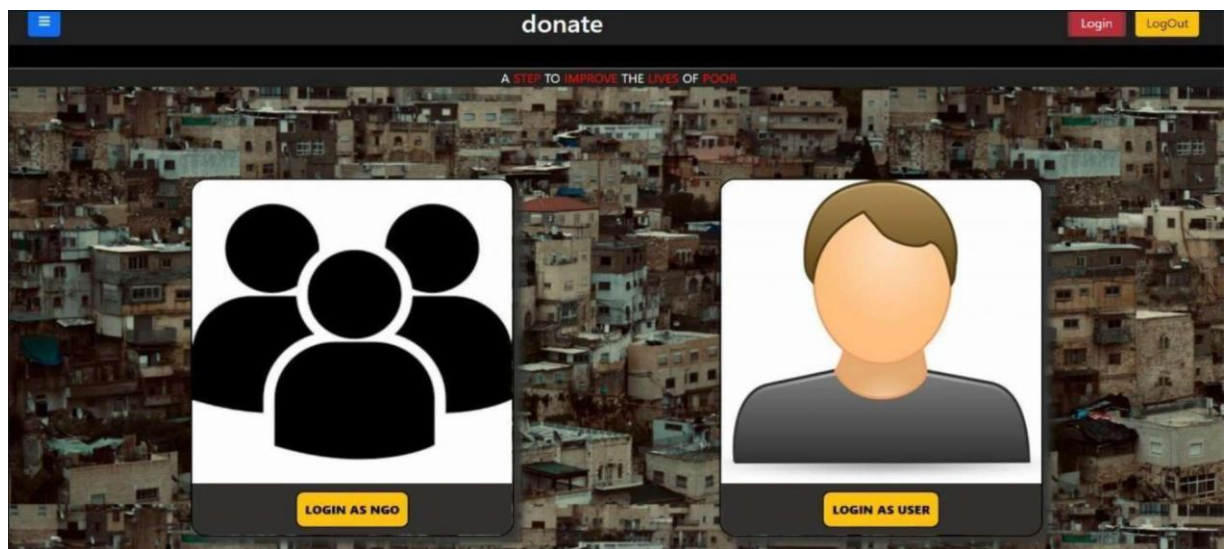


Fig 13

donate

LoginLogout

A STEP TO IMPROVE THE LIVES OF POOR

Login

Welcome to Donation

Camp

Enter your credentials and start journey with us.

Ayush Jain

Forgot Password ?

Login

New User ? [Create Account](#)




Fig 14

donate

LoginLogout

A STEP TO IMPROVE THE LIVES OF POOR

WELCOME

Sign Up

Please fill in this form to create an account!

Ayush Jain

Email

Create Account

Already have an account ? [Login here](#)

Fig 15

Website Code

HomePage.JS

```
<Slider {...settings}>
  <div>
    
  </div>
  <div>
    
  </div>
  <div>
    
  </div>
  <div>
    
  </div>
  <div>
    
```



```

        </div>
        <div>
          
        </div>
      </Slider>
    </div>
  );
}
}
import React from "react";
import styles from
'./Donation.module.css' import {
useNavigate } from 'react-router-
dom'

const Donation = () => {
  const navigate =
    useNavigate()
  const
    handleSubmit =
    () =>{
      navigate('/images')
    }
  const
    handleSubmit
    _1 = () =>{
      navigate('/im
agesFootwear
')
    }
  return (
    <>
      <div className={` ${styles.DonateContainer} d-flex justify-
content-center align-items-center container-fluid`}>
        <div className={` ${styles.sentence} text-white text-center fw-
bold`}>YOU CAN DONATE OLD ITEMS IN THESE
        CATEGORIES
      </div>
    </div>
  )
}

```

```

    <div onClick={handleSubmit_1}
className={styles.DonateBoxContainer} container-fluid>
    <div className={` ${styles.row} row`}>
    <div className={` ${styles.col} ${styles.bboxes} col-6 col-md-3`}>
    <div className={` ${styles.cards} card border-0`}>
    { /*  */ }
    <div className="card-body p-0">
    <h5 className={` ${styles.title} card-title d-flex justify-
content-center align-items-center w-100 text-white`}>FOOTWEAR</h5>
    </div>
    </div>
    <div className={` ${styles.cards} card border-0`}>
    { /*  */ }
    <div className="card-body p-0">
    <h5 className={` ${styles.title} card-title d-flex justify-
content-center align-items-center w-100 text-white`}>CLOTHES</h5>
    </div>
    </div>
    { /* <div onClick={handleSubmit} className={` ${styles.col} col-6 col-
md-3`}>
    <div className={` ${styles.cards} card border-0`}>
    
    <div className="card-body p-0">
    <h5 className={` ${styles.title} card-title d-flex justify-
content-center align-items-center w-100 text-
white`}>BOOKS/NOTEBOOKS</h5>
    </div>
    </div>

```

```

    </div>
    <div onClick={handleSubmit} className={` ${styles.col} col-6 col-md-3`}>
      <div className={` ${styles.cards} card border-0`>
        
        <div className="card-body p-0">
          <h5 className={` ${styles.title} card-title d-flex justify-content-center align-items-center w-100 text-white`>MONEY</h5>
        </div>
      </div>
    </div>
  </div>

const
footer = ()
=> {return
(
  <div className="container-fluid">
    <div className="row">
      <div className={` ${styles.col} col-6`>
        <div className={` ${styles.box} d-flex justify-content-center align-items-center text-white text-center`>
          <div className={` ${styles.footcontain} row`>
            <div className="col-12">XYZ@ALL RIGHTS ARE RESERVED</div>
            <div className="col-12">XYZ@GMAIL.COM</div>
          </div>
        </div>
      </div>
      <div className={` ${styles.col} col-6`>
        <div className={` ${styles.box} d-flex justify-content-center align-items-center text-white text-center`>
          <div className={` ${styles.footcontain} row`>
            <div className="col-12">XYZ COMPANY</div>
            <div className="col-12">ABC, NEW DLHI-110089</div>
          </div>
        </div>
      </div>
    </div>
  </div>
)
}

```

```

        </div>
      </>
    );
  };

```

```
export default footer;
```

```
import React from "react";
import styles from "./Shoutout.module.css";
```

```
const
  Shoutou
  t = () =>
  { return (
    <
      <div className="container">

        </div>
      </div>
    </div>
  </div>
    </>
  );
};
```

```
export default Shoutout;
```

```
import React from 'react';
import styles from './Testimaonials.module.css'
```

```
const
  Testimaonia
  ls = () => {
    return (
      <div className={` ${styles.bg} container mt-5`} >
        <div className={` ${styles.test} row m-5`} >
          <div className='col-md-4 p-4'>
            <div className='w-100'>
              
            </div>
          </div>
          <div className={` ${styles.text} col-md-8 d-flex justify-
```

content-center align-items-center p-5` }>

 Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio ducimus quo dolore accusamus veniam temporibus doloremque recusandae cum adipiscidignissimos ipsam aliquid placeat, tempora corporis eaque quisquam molestiae culpamagni saepe. Consequatur, asperiores voluptates. Lorem ipsum, dolor sit amet consectetur adipisicing elit. Ut, quis eius velit odio deserunt porro quisquam abperferendis! Alias in voluptates architecto ab eligendi quod repellat beatae explicabo, quia iusto suscipit, consequatur minus illum.

 </div>

 </div>

 <div className={` \${styles.divider} row m-5` }></div>

 <div className={` \${styles.test} row m-5` }>

 <div className={` \${styles.text} col-md-8 d-flex justify-

content-center align-items-center p-5` }>

 Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio ducimus quo dolore accusamus veniam temporibus doloremque recusandae cum adipiscidignissimos ipsam aliquid placeat, tempora corporis eaque quisquam molestiae culpamagni saepe. Consequatur, asperiores voluptates. Lorem ipsum, dolor sit amet consectetur adipisicing elit. Ut, quis eius velit odio deserunt porro quisquam abperferendis! Alias in voluptates architecto ab eligendi quod repellat beatae explicabo, quia iusto suscipit, consequatur minus illum.

 </div>

 <div className='col-md-4 p-4'>

```

</div>
</div>
{/* <div className={` ${styles.test} row m-5`} >
<div className='col-8 d-flex justify-content-center align-items-
center p-5'>
Lorem ipsum dolor sit amet consectetur adipisicing elit. Odio
ducimus quo dolore accusamus veniam temporibus doloremque
recusandae cum adipisci dignissimos ipsam aliquid placeat,
tempora corporis eaque quisquam molestiae culpa magni saepe.
Consequatur, asperiores voluptates. Lorem ipsum dolor sit,
amet consectetur adipisicing elit. Quas hic dicta explicabo,
accusamus nemo, soluta cumque voluptatibus dolor at
repudiandae alias rerum, aperiam corrupti. Natus laborum eum
repellat excepturi, ratione a id? Ad, numquam!
</div>
<div className='col-4 p-4'>

</div>
</div> */}
</div>
)}}

```

Login Code:

```

import React, { useState } from "react"; import styles from "./Login.module.css";
import { useNavigate } from "react-router-dom"; import { CgLogIn } from "react-
icons/cg"; import { Formik, Form, Field } from "formik";
const Login = (props) => {
const navigate = useNavigate()
const [email, setemail] = useState("")
const [password, setpassword] = useState("") const handleSubmit = async () => {
// e.preventDefault();
// const { name, email, password } = credentials;
const response = await fetch("http://localhost:5000/api/auth/login", { method: 'POST',
headers: {
'Content-Type': 'application/json',
},
body: JSON.stringify({ email, password })
});

```

```

const json = await
response.json()
console.log(json);
if (json.success) {
  // Save the auth token and redirect
  props.showAlert("Account Created Successfully",
    "success")localStorage.setItem('token',
    json.authtoken); navigate('/');
}
else {
  props.showAlert("Invalid Credential", "danger")
}
}

const validateEmail =
(value) => {let
error;
if (!value) {
  error = "*This field is Required";
} else if (!/^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-
Z]{2,4}$/i.test(value)) {error = "*Invalid email address";
}
};

const validatePassword =
(value) => {let error;
if (!value) {
  error = "*This field is Required";
} else if (!/^[0-9a-zA-
Z]{0,10}$/i.test(value)) {error =
  "*Invalid Password";
}
err
or;
};

return (
  <
    <div className="container mt-5">
      <div className="row">
        <div className="col-md-6 m-auto">
          <div className="card border-dark h-100">
            <div className="card-body">
              <h3 className="text-dark">Login</h3>

```

```

<h5 className="card-title text-center mt-5">
  Welcome to Donation
  <span className="btn btn-success mx-2">Camp</span>
</h5>
<p className="text-center text-muted mt-3 mb-5"> Enter
  your credentials and start journey with us.
</p>
<Formik
  initialValues={{
    email: "",
    password: "",
  }}
  onSubmit={handleSubmit}
>
  ({ errors, touched }) => (
    <Form>
      <div className="mb-4">
        <Field
          type="email"
          className={`form-control ${errors.email && touched.email ?
'border-danger': ""}`}
          id="email"
          name="email"
          placeholder="Email"
          validate={validateEmail}
        />

```



```

        {errors.email && touched.email && <div
className="form-texttext-danger">{errors.email}</div>}
    </div>
    <div className="mb-3">
        <Field
            type="password"
            className={`form-control
${errors.password &&touched.password ? 'border-danger' : ''}`}
            id="password"
            "
            name="passw
            ord"
            placeholder=
            "Password"
            validate={ val
            idatePasswor
            d}

</p>
<div className="mb-5 col-3 d-grid mx-auto">
    <button type="submit" className="btn btn-dark">
    <CgLogIn className="me-1" /> Login
    </button>
</div>
<p className="text-center"> New User ?{ " "}
    <span
        onClick={() => navigate("/signup")}
        className="ms-1 text-dark fw-bold text-decoration-
        underline"
        style={{ cursor: "pointer" }}
    >
    Create Account
    </span>
</p>
</Form>
)}
</Formik>
</div>
</div>
</div>
<div className={` ${styles.right} col-md-6 m-auto`}>
    
</div>

```

```

</div>
</div>
</>
);
};

export default Login;

```

NGO Login Code:

```

import React, { useState } from "react"; import styles from
"./NgoLogin.module.css";
import { useNavigate } from "react-router-dom"; import { CgLogIn } from
"react-icons/cg";
import { Formik, Form, Field } from "formik";

```

```

const NgoLogin =
  (props) => {
    const navigate =
      useNavigate()
    const [email, setemail] = useState("")
    const [password, setpassword] =
      useState("")const handleSubmit =
      async () => {
        // e.preventDefault();
        // const { name, email, password } = credentials;
        const response = await
          fetch("http://localhost:5000/api/ngo/login", { method:
            'POST',
            headers: {
              'Content-Type': 'application/json',
            },
            body: JSON.stringify({ email, password })
          });

        const json = await
          response.json()
        console.log(json);
        if (json.success) {
          // Save the auth token and redirect
          props.showAlert("Account Created Successfully",
            "success")localStorage.setItem('token',
              json.authtoken); navigate('/Ngo');
        }
      }
  }

```

```

return (
  <div className="container mt-5">
    <div className="row">
      <div className="col-md-6 m-auto">
        <div className="card border-dark h-100">
          <div className="card-body">
            <h3 className="text-dark">Ngo Login</h3>
            <hr
              style={{
                color:
                  "#000",
                borderT
                  op: "2px
                    solid",
                    opacity:
                      "1",
              }}
            />
            <h5 className="card-title text-center
              mt-5">Welcome to Donation
            <span className="btn btn-success mx-
              2">Camp
            </span>
            </h5>
            <p className="text-center text-muted mt-3 mb-
              5"> Enter your credentials and start journey
              with us.
            </p>
            <Formik
              initialV
                alu
                es
                ={
                  {
                    em
                    ail:
                      "",
                    password: "",
                  }}
              onSubmit={handleSubmit}
            >
              {{({ errors, touched }) => (
                <Form>

```

```

        className={`form-
        control
        ${errors.password && touched.password ? 'border-danger' : ''}`}
        id="password"
        name="password"
        placeholder="Password"
        validate={validatePassword}
      />
    {errors.password &&
    touched.password}&& <div className="form-text text-
    danger">{errors.password}</div>
  </div>
  <p
    onClick={() =>
      navigate("/resetPassword")}
    id="forgetPassword"
    className={` ${styles.forgotPass
    word}
    style={{ cursor: "pointer" }}
  >
    Forgot Password ?
  </p>
  <div className="mb-5 col-3 d-grid mx-
  <button type="submit"
    className="btn
    <CgLogIn className="me-
    Login
  </button>
</div>
<p className="text-center">
  New User ?{ " "}
  <span
    onClick={() =>
      className="ms-1 text-dark s

```

Sign Up Code:

```
import React, { useState }
from "react"; import styles
from "./SignUp.module.css";
import { useNavigate } from "react-router-dom";

import { IoIosLock } from "react-icons/io";

// import welcome from "../assets/Images/signup.png";

import { Formik, Form, Field } from "formik";

const SignUp = (props) => {
  const [name, setname] =
    useState("") const [email,
    setemail] = useState("")
  const [password, setpassword] =
    useState("")const navigate =
    useNavigate()
  const handleSubmit = async () => {
    // e.preventDefault();
    // const { name, email, password } = credentials;
    const response = await
      fetch("http://localhost:5000/api/auth/createuser", { method:
        'POST',
        headers: {
          'Content-Type': 'application/json',
        },
        body: JSON.stringify({ name, email, password })
      });

    const json = await
    response.json()
    console.log(json);
    if (json.success) {
      // Save the auth token and redirect
      props.showAlert("Account Created Successfully",
        "success")localStorage.setItem('token',
        json.authtoken); navigate('/');
    }
    else {
```

```

const validateUserName
  = (value) => {let
    error;
    if (!value) {
      error = "*This field is required";
    }
  };

const validateEmail =
  (value) => {let
    error;
    if (!value) {
      error = "*This field is required";
    } else if (!/^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-
      Z]{2,4}$/i.test(value)) {error = "*Invalid email address";
    }
  };

const validatePassword =
  (value) => {let error;
    if (!value) {
      error = "*This field is Required";
    } else if (!/^[0-9a-zA-Z]{9,100}$/i.test(value))
      { error = "*Invalid Password must be
        greater than 8";
    }
  }

return (
  <
    <div className="container mt-5">
      <div className="row">
        <div
          className={` ${styles.left} col-md-6 d-flex justify-
            content-

```

CHAPTER 8: TESTING

Introduction

Testing is a step in which the final system is verified and validated. Testing's goal is to find various types of mistakes with the least amount of time and effort possible. Furthermore, the system must go through a testing and assessment process for each module and capability.

Testing is needed:

The testing process is important because it allows the final system to be verified and validated. So that it will be run and presented as intended, and that all of the functions will work as specified in the project definition form. As a result, testing is carried out in order to improve the quality of the system generated so that the intended user is satisfied with it.

Type of testing

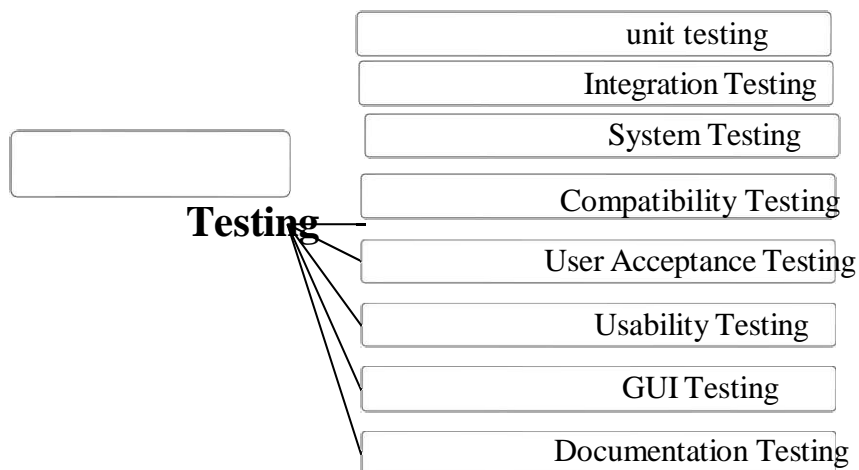


Fig 16

1.	Unit testing	Individual module testing, as well as the type of testing, is only done by developers because it necessitates in-depth understanding. It comprises black-box and white-box testing.
2.	Integration testing	Various modules are combined and then tested by the system's developer to ensure that they function properly.
3.	System testing	The entire system is created in accordance with the specifications.
4.	Compatibility testing	If the system causes any problems with the way it functions in relation to the operating system.
5.	GUI testing	Testing is required to ensure that the system's design is correct. This is accomplished by analysing a product by seeing real people at work.
6.	Usability testing	The system is tested by the end user
7.	Documentation testing	Documentation testing is used to see if the system documentation is clear and easy to understand.

Fig 17

Unit Testing

Unit testing is the first level of software testing where the smallest testable parts of a software are tested. This is used to validate that each unit of the Software performs as designed. The unit test framework is python's unit style framework.

Method:

White Box Testing method is used for Unit testing. OOP concepts supported by unit test framework:

Test fixture:

A test fixture is used as a baseline for running tests to ensure that there is a fixed environment in which tests are run so that results are repeatable.

Examples :

Creating temporary databases and creating test cases

Test case:

A test case is a set of conditions which is used to determine whether a system under test works correctly.

Test suite:

Test suite is a collection of test cases that are used to test a software program to show that it has some specified set of behaviours by executing the aggregated tests together.

Test runner:

A test runner is a component which sets up the execution of tests and provides the outcome to the user.

CHAPTER 9: CONCLUSION

Success criteria

The proposed system, named "Donation System," has shown to be incredibly helpful to the end user, since it fits all of the customer's needs as well as the system's quality and victory criteria. It is regarded effective when the proposed system meets the success criteria and identifies what it should do. The system has addressed the concerns raised in Chapter 2 in terms of obstacles faced and solutions provided. Following the evaluation, it was revealed that the proposed system effectively safeguarded the end user's reservation and control in a real-time context, thereby resolving the problem.

Limitation.

- A user must login into the system for accessing all the features. Learning experience gather

Research work taught me a lot: During the early stages of the project, the developer was unsure of what features to put in the system. As a result, the developer did secondary research by examining other authors' research papers and pointing out the mistakes they had made. I helped to acquire in-depth knowledge of the subject matter and to critically analyze comparable systems in terms of their advantages and limitations. Finally, the developer conducted primary research to get real-time consumer desires.

Learned to manage stress and time: During the various stages of the SDLC, the developer took on several tasks, including Database and Mobile Application Analyzer, Researcher, Developer, and Tester. This gave the developer the ability to efficiently manage both stress and time. The developer was able to complete the project within the planned time frame, overcoming all the stress and time constraints encountered during the development phase.

Gained technical expertise: Not only did the system implementation necessitate the application of the course's learning goals, but it also necessitated a thorough grasp of various mobile technologies. To learn the development principles of Android applications, numerous services, MySQL database concepts, web services, and other topics, you'll need to read a variety of books and articles. Finally, it aided in the development of technical knowledge and the mastery of many languages and technologies such as Java, XML, JSON, Android SDK, and others.

How to resolve errors and learn from them: Several faults and exceptions were found throughout the system's implementation, which were resolved utilizing various approaches. This offered me an idea of how to use a top-down method to debug and solve errors. In addition, the developer has learnt how to discover and resolve flaws, a skill that is essential to create a bug-free system.

Finally, this 4th year project provided a great learning experience and enjoyable experience to work on projects in real time and sparked a deep interest in the development of mobile apps that will bring benefits for a career in the future.

Conclusion:

Concerns uncovered in the current study topic, notably the obstacles in the ticket environment. All of the challenges that the ordinary user faces have been identified by the developer. The queries have been documented, as well as the elaboration behind each query. The developer's next goal was to provide perfect solutions to all of the issues that had been identified. Each problem's query, as well as the thinking behind it, has been made public for the same reason. The feasibility study was successfully completed, allowing the developer to move on with the project.

CHAPTER 10: REFERENCES

1. <https://www.umeedngo.org/> -:

(The Umeed NGO is an NGO we all must have heard of and they also have their website to get donations. But they only take donations in clothes and food. We are filling the gap here by increasing the number of categories, the donors can donate to. Not only this, through our platform, donors can donate to any NGO registered on ourPlatform.)

2. Bromideh, A. A. (2011). The widespread challenges of NGOs in developing countries: Case studies from Iran. International NGO Journal, 6(9),
<http://www.academicjournals.org/journal/INGOJ/articleabstract/117960141095>

fl

3. Using Scaling MongoDB by Kristina Chodorow, Released August 2011
Publisher(s):O'Reilly Media, Inc.
4. <https://www.indiadonated.org>
5. <https://www.donatekart.com>
6. <https://www.givedo.com>
7. <https://www.globalgiving.org>
8. <https://www.smilefoundationindia.org>
9. <https://www.akshaypatra.org>
10. Dart Apprentice By Jonathon Sande
11. https://www.youtube.com/channel/UCois2_Va_dAq4UowZ5Mxy9Q

12. Pro React 16 By Adam Freeman
13. React Cookbook By Carlos Santana Roldan
14. Getting Started with React By Danilo Corvalan
15. Learning react By Kirupa Chinnathambi
16. Mastering React By Adam Horton
17. Javascript Everywhere By Adam D. Scott
18. Scaling MongoDB By Kristina Chodorow
19. MongoDB Basics By David Hows
20. MongoDB Cookbook By Amol Nayak
21. Express In Action By Evan Hahn
22. Beginning Node.js By Basarat Ali Syed
23. Node.js Web Development By David Herron
24. Mastering Node.js By Sabdro Pasquali
25. Node.js: Novice to Ninja By Craig Buckler
26. Node.js By Dhruti Shah
27. <https://www.butterfliesngo.org>
28. <https://www.proudindian.ngo>
29. <https://www.swadesfoundation.org>
30. <https://www.aarohanngo.org>
31. <https://www.goonj.org>
32. <https://www.ketto.org>