**Table of Contents**

### Page No.

### [Chapter 1. Introduction](#chap2)

1. Introduction to the System 4
2. Problem Definition 4
3. Aim 5
4. Objective 5
5. Goal 6
6. Need of System 6

### [Chapter 2. Hardware and Software requirement](#chap3)

1. Introduction 8
2. System environment 8
3. Software requirement 9
4. Hardware requirements 10

### [Chapter 3. System Analysis](#chap4)

1. Purpose 12
2. Project Scope 12
3. Existing System 13
4. Proposed System 14
5. System Description 15

### [Chapter 4.Implementation issues](#chap 8)

1. Python 17
2. HTML 18
3. Cascading style sheet(CSS) 19
4. Sqlite 20

[**Chapter 5. System Design**](#chap5)

1. Introduction 21
2. Conceptual Design 22
3. ER-Diagram 23
4. Logical Design 24
5. Physical Design 25
6. Data Flow Diagram of **“Donation Management System”** 26

### [Chapter 6.User Screens](#chap 8) 27-60

### [Chapter 7.Coding](#chap 8) 61-94

[**Chapter 8. Conclusion**](#chap 9)

1. Features of “**Donation Management System**” 95
2. Benefits Accrued from “**Donation Management System**” 96
3. Limitations of “**Donation Management System**” 97

[Bibliography](#bibliography)

**CHAPTER # 1**

**Introduction**

*Contents:*

* Introduction
* Problem Definition
* Aim
* Objective
* Goal
* Need of System

**Introduction to the System:**

* Donation Management Project System is a mission to end hunger and no wasting of food and other needy items to make a hungry-free world.
* In the current working scenario, many NGOs are struggling with some issues mainly communication with their member, heads and volunteer while NGO are donating.
* The focus of this project is to reduce the amount of wastage of needy items and being used to the needy people.

**Problem Definition:**

Presently people who wish to donate items need to personally visit the organizations and donate foods or other items. Otherwise, they have to search for some websites to donate needy items. In general, the large manufacturers, wholesalers, and organized community provide food items to food banks or waste tons of foods daily. They have to search for some organization that needs food. This process involves a lot of time to contact the organization to check the requirement. If they do not need the food, then the person has to contact another organization. This makes the donor tired and exhausted.

**Objective:**

* Reduce lack of awareness.
* Enable easy interaction between donor and volunteer.
* It is very much faster than manual system.
* Easy and fastest record finding technique.
* It is very much flexible to work.
* It is very user oriented.
* Data can be stored for a longer period.

**Need of the System:**

There is always a need of a system that will provide an interactive platform between donors and NGOs (Volunteers). This system will reduce the manual operation required to collect donations. And also generates the various reports for analysis.

Thus, there is a big need of an online donation management system, which provides all the above- mentioned facilities and many more.

**CHAPTER # 2**

**Hardware and Software Requirements**

*Contents:*

##### Introduction

* System environment
* Software requirement
* Hardware requirements

**Introduction:**

In this chapter we mentioned the software and hardware requirements, which are necessary for successfully running this system. The major element in building systems is selecting compatible hardware and software. The system analyst has to determine what software package is best for the **“Donation Management System”** and, where software is not an issue, the kind of hardware and peripherals needed for the final conversion.

**System Environment:**

After analysis, some resources are required to convert the abstract system into the real one.

The hardware and software selection begins with requirement analysis, followed by a request for proposal and vendor evaluation.

Software and real system are identified. According to the provided functional specification all the technologies and its capacities are identified. Basic functions and procedures and methodologies are prepared to implement. Some of the Basic requirements such as hardware and software are described as follows: -

**Hardware and Software Specification**

**Software Requirements:**

* Technology: Python Django
* IDE : Pycharm/Atom
* Client Side Technologies: HTML, CSS, JavaScript , Bootstrap
* Server Side Technologies: Python
* Data Base Server: Sqlite
* Operating System: Microsoft Windows/Linux

**Hardware Requirements:**

* Processor: Pentium-III (or) Higher
* Ram: 64MB (or) Higher
* Hard disk: 80GB (or) Higher

**CHAPTER # 3**

**System Analysis**

*Contents:*

##### Purpose

* Project Scope
* Existing System
* Proposed System
* System Overview

### Purpose:

1. This project is aimed at developing a web based Donation Management Tool, which is of importance to either an NGO or an organization. This system can be used to automate the workflow and keep track of donation collection and donation delivery. There are features like cancellation of accept or reject donation, allocate volunteer as per location, report generators etc. in this Tool.

**Project Scope:**

The project has a wide scope, as it is not intended to a particular organization. This project is going to develop generic software, which can be applied by any businesses organization. More over it provides facility to its users. Also the software is going to provide a huge amount of summary data.

**Proposed System:**

The development of the new system contains the following activities, which try to automate the entire process keeping in view of the database integration approach.

1. User friendliness is provided in the application with various controls.
2. The system makes the overall project management much easier and flexible.
3. There is no risk of data mismanagement at any level while the project development is under process.
4. It provides high level of security with different level of authentication.

**System Overview:**

Donation management system project divided in three modules:

1. **Donor module**
2. **Volunteer module**
3. **Admin module**

### **Donor Module details**

* Add Donation Detail
* View Donation History
* Edit Profile
* Change Password

### Volunteer Module

* View Donation Collection Request
* Update Donation Status and Remark
* View Donation History
* Edit Profile
* Change Password

Admin Modules:

* Admin is the super user of the website who can manage everything on the website. Admin can log in through the login page
* Dashboard: In this section, admin can see all detail in brief like the total new donation request, Total accepted donation, Total donation delivered, Total donors , Total volunteers and Total donation area.
* View All New Donation Request.
* Accept or Reject Donation
* Assign Volunteer and Donation Area
* Manage Donors.
* View New Volunteer Registration Request.
* Accept or Reject Volunteer.
* Manage Volunteer.
* Admin can also update the password.

**CHAPTER # 4**

**Implementation issues**

**Python**

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

**HTML**

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

**CASCADING STYLE SHEET (CSS)**

Cascading Style Sheets (CSS) are a collection of rules we use to define and modify web pages. CSS are similar to styles in Word. CSS allow Web designers to have much more control over their pages look and layout. For instance, you could create a style that defines the body text to be Verdana, 10 point. Later on, you may easily change the body text to Times New Roman, 12 point by just changing the rule in the CSS. Instead of having to change the font on each page of your website, all you need to do is redefine the style on the style sheet, and it will instantly change on all of the pages that the style sheet has been applied to. With HTML styles, the font change would be applied to each instance of that font and have to be changed in each spot.

CSS can control the placement of text and objects on your pages as well as the look of those objects.

HTML information creates the objects (or gives objects meaning), but styles describe how the objects should appear. The HTML gives your page structure, while the CSS creates the “presentation”. An external CSS is really just a text file with a .css extension. These files can be created with Dreamweaver, a CSS editor, or even Notepad.

The best practice is to design your web page on paper first so you know where you will want to use styles on your page. Then you can create the styles and apply them to your page.

**Javascript**

JavaScript is a programming language commonly used in web development. It was originally developed by Netscape as a means to add dynamic and interactive elements to websites. While JavaScript is influenced byJava, the syntax is more similar to C and is based on ECMAScript, a scripting language developed by Sun Microsystems.

JavaScript is a client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server. This means JavaScript functions can run after a webpage has loaded without COMMUNICATING with the server. For example, a JavaScript function may check a web form before it is submitted to make sure all the required fields have been filled out. The JavaScript code can produce an error message before any information is actually transmitted to the server.

Like server-side scripting languages, such as PHP and ASP, JavaScript code can be inserted anywhere within the HTML of a webpage. However, only the output of server-side code is displayed in the HTML, while JavaScript code remains fully visible in the source of the webpage. It can also be referenced in a separate .JS file, which may also be viewed in a browser.

**Django**

Django is a web application framework written in Python programming language. It is based on MVT (Model View Template) design pattern. The Django is very demanding due to its rapid development feature. It takes less time to build application after collecting client requirement.

This framework uses a famous tag line: **The web framework for perfectionists with deadlines.**

**CHAPTER # 5**

**System Design**

*Contents:*

* Use case diagram
* Class Diagram
* Sequence Diagram
* Data flow diagram

**Use Case Diagram:**

* Use case diagram consists of use cases and actors and shows the interaction between them. The key points are:
* The main purpose is to show the interaction between the use cases and the actor.
* To represent the system requirement from user’s perspective.
* The use cases are the functions that are to be performed in the module.

**Use Case Diagram - Admin**

**Use Case Diagram – Donor**

**Use Case Diagram – Volunteer**

**FLOW CHART**

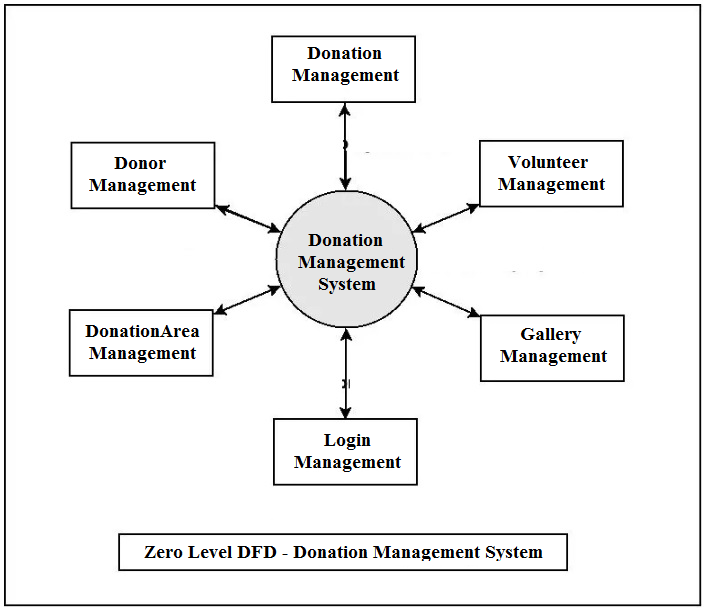
**Admin Flow Chart -**

**Donor flow chart**

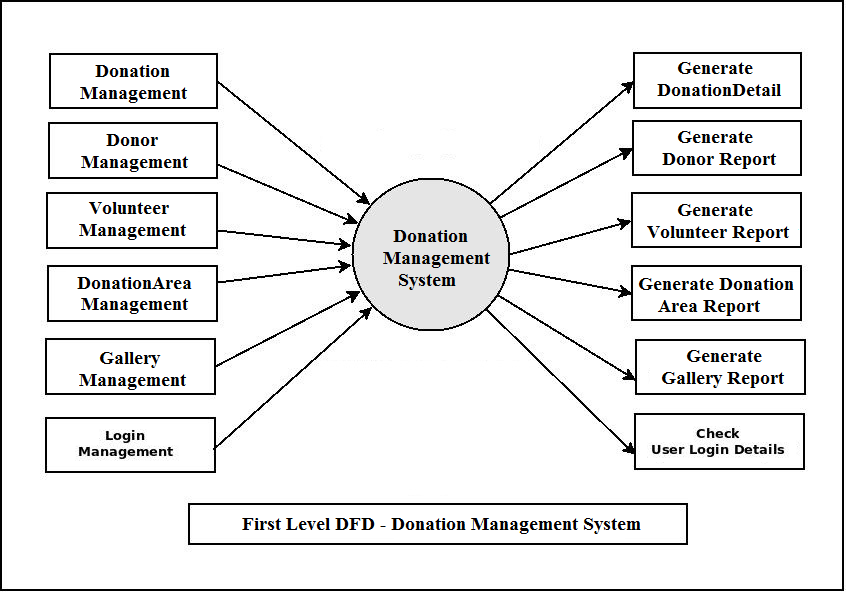
**Volunteer Flow Chart**

**DFD (Data Flow Diagram)**

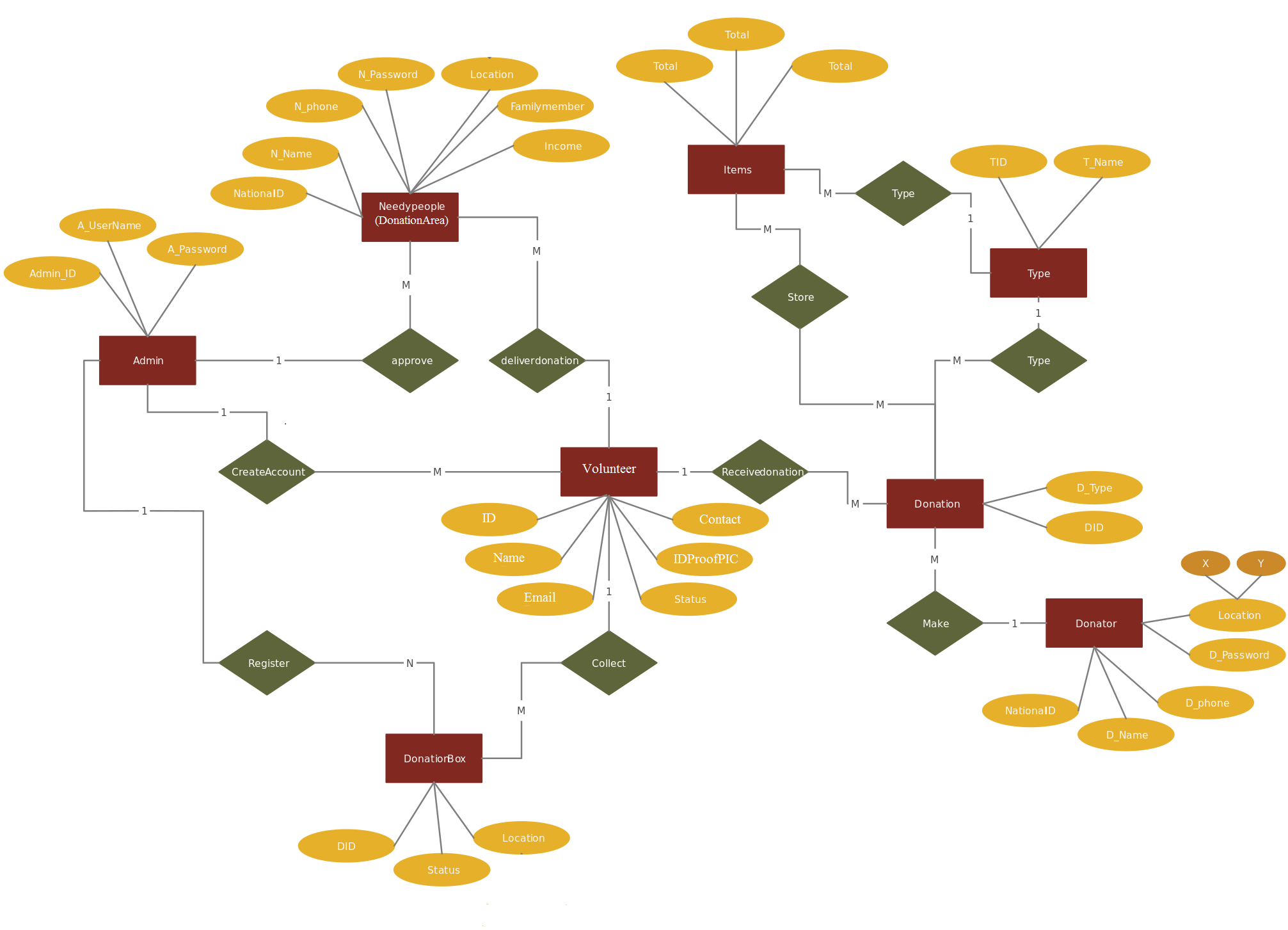
**DFD Level 1**

****

**DFD Level 2**

****

**ER DIAGRAM**

****

**Sequence Diagram For Administrator:-**

**Administrator**

**Success:hide()**

**Login**

**Application**

**Database**

**Login**

**:Request**

**:Validate()**

**:executeQuery()**

**Response**

**Show Result**

**Failed:show()**

Fig.5.4

**Sequence Diagram For User:-**

**User**

**Success:hide()**

**Login**

**Application**

**Database**

**Login**

**:Request**

**:Validate()**

**:executeQuery()**

**Response**

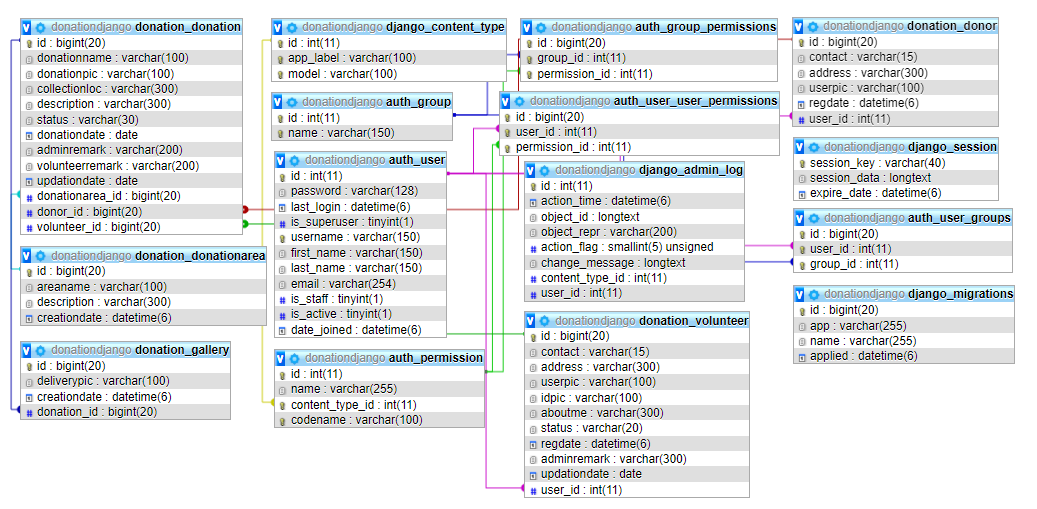
**Show Result**

**Failed:show()**

**:**

Fig.5.5

**Tables Relationship (Class Diagram)**

****

**CHAPTER # 6**

**Output screens**

**CHAPTER # 7**

***Coding***

**CHAPTER # 8**

**Advantages & Limitations**

**Advantages of “Donation Management System”:**

“Donation Management System” provides various features, which complement the information system and increase the productivity of the system. These features make the system easily usable and convenient. Some of the important features included are listed as follows:

* Intelligent User Forms Design
  + - Data access and manipulation through same forms
    - Access to most required information
* Data Security
* Restrictive data access, as per login assigned only.
* Organized and structured storage of facts.
* Strategic Planning made easy.
* No decay of old Records.
* Exact financial position of the business.

**Limitations of “Donation Management System”:**

Besides the above achievements and the successful completion of the project, we still feel the project has some limitations, listed as below:

* + It is not a large scale system.
  + Only limited information provided by this system.
  + Since it is an online project, user needs internet connection to use the software.
  + People who are not familiar with computers can’t use this software.

**CHAPTER # 8**

**Future Scope**

**FUTURE SCOPE**

Thisweb application involves almost all the basic features of the online donation management system. The future implementation will be online help for the users and chatting with website administrator.

**CONCLUSION**

It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in Python and Sqlite web based application. It also provides knowledge about the latest technology used in developing web enabled application and client server technology that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

**Bibliography**

**BIBLIOGRAPHY**

* Wikipedia
* <https://www.geeksforgeeks.org/python-django/>
* <https://www.javatpoint.com>
* <https://www.python.org/>
* <https://www.tutorialspoint/>
* **REFERENCE BOOKS**

### Two scoops of Django for 1.11 by Daniel Greenfeld’s and Audrey Greenfield

### Lightweight Django by Elman and Mark Lavin