CROWD FUNDING USING BLOCKCHAIN

A PROJECT REPORT

Submitted by

ATHIRA V

CUAWMCS006

In Partial Fulfillment of the Requirements for the Degree of

MSc. COMPUTER SCIENCE

CENTRE FOR COMPUTER SCIENCE AND

INFORMATION TECHNOLOGY (CCSIT)

UNIVERSITY OF CALICUT

JUNE 2024

ACKNOWLEDGEMENT

Let me take this opportunity to thank all those who have been directly and indirectly involved in making this project a success. First of all, thank you Universe for making this possible.

I express my gratitude to Mrs. Seema S, Associate Coordinator, Centre for Computer Science and Information Technology, University of Calicut .

I am extremely grateful to Mrs. Teena G Nath, Assistant Professor, Centre for Computer Science and Information Technology, University of Calicut, my Project-in-Charge for the encouragement, valuable suggestions and guidance rendered.

I express my sincere thanks to my family and friends who helped me in overcoming all the hurdles thus gearing me towards the fulfillment of the task that I had overtaken.

I also express my profound gratitude to all others who, in some way or another, helped me with this project till the end.

Place: **CU CAMPUS**   **ATHIRA V**

Date: **25/06/2024**   **CUAWMCS006**

DECLARATION

I **ATHIRA V,** hereby declare that the project work entitled **"CROWD FUNDING USING BLOCKCHAIN "** is original work done by me, under the guidance of **Mrs. Teena G Nath,** Assistant Professor**,** Centre for Computer Science and Information Technology, University of Calicut. This project report is submitted in partial fulfillment of the requirement for the award of the degree of MSc. Computer Science during the period of study at University of Calicut.

Place: **CU CAMPUS**

Date:25/06/24

ATHIRA V
CUAWMCS006

CERTIFICATE

    Certified that this is a bonafide record of the project work entitled **"CROWD FUNDING USING BLOCKCHAIN "** *submitted by* **ATHIRA V (Reg No: CUAWMCS006)** *in partial fulfillment of the requirement for the*   *award of the degree of* **MSc. COMPUTER SCIENCE** *from University of Calicut.*

Mrs. TEENA G NATH
Assistant Professor
CCSIT, CU Campus

Mrs. SEEMA S
Associate Coordinator
CCSIT, CU Campus

Place: CU CAMPUS

Date : 25/06/24

Certified that the candidate was examined by us in the Project Viva Voice Examination held

on …..…………….. and her Register Number is

……………………………………………………………………………………… …….

Examiners

……………………………….

……………………………….

CROWD FUNDING USING BLOCKCHAIN

ABSTRACT

Blockchain technology offers a clear advantage for the crowdfunding industry, bringing transparency and security to the space and protecting both creators and donors in these interactions.

Many traditional industries are moving their operations onto the blockchain to leverage the potential of this technology. It is only a matter of time before the crowdfunding industry explores the blockchain, opening the path for more funds to flow into the space

Blockchain is a promising technology for charities as they look to promote transparency and at the same time cut the risk of fraud. But too often it is being ignored by charities through a lack of understanding, which can be due to a lack of technical expertise, as well as difficulty in explaining how it can offer charities tangible benefits.

INDEX

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

INTRODUCTION

This application is used for charity funding. If someone needs money, they can use it, but only a certain amount can be taken. This is a transparent application. We can do it with block chain, Also we can do it normally Crowd funding is a way to raise money from a large number of individual investors or companies. In this, investors can contribute to any project they are interested in and can gain the profit if that project gets successful. Now a days, many crowd funding platforms already exist and they take huge chunk of money from investors and contributes and leave them with empty promises. Crowd funding using block chain changes the traditional way to deal with business funding. Generally, when people need to raise a cash to begin a business, they have to design strategy, statistical surveying, and models, and afterward present the thoughts around to attract people or organizations. These subsidizing sources included banks, angel investors, venture capital firms. The present day crowd funding model depends on three kinds of on-screen characters: the task initiator who proposes the thought or venture to be funded, people or investors who invests in the thought, and a platform which puts these two characters together to make the venture successful. It is used to finance a wide scope of start-ups, pioneering ideas, for example, innovative activities, medical advances, travel and social business enterprise ventures

2. SYSTEM STUDY

SYSTEM STUDY

EXISTING SYSTEM

There is not have been a widely recognized or prominent platform that exactly matches the detailed combination of this features – a blockchain-powered charity funding platform with voting, transparent tracking, and accountability measures.

However, the technology landscape is constantly evolving, and new platforms and projects are developed all the time. It's possible that developments in the blockchain and crowdfunding spaces since then have led to the creation of platforms that share similarities with your concept

PROPOSED SYSTEM

The proposed system provides a comprehensive solution for transparent charity funding using both traditional methods and blockchain technology. It ensures user authentication, secure donation transactions, complaint management, and user management while incorporating the benefits of blockchain for transparency and security. Implementation details, technology stack, and specific smart contract design would need to be further developed based on the requirements and resources available.

3. SYSTEM ANALYSIS

SYSTEM ANALYSIS

PRELIMINARY INVESTIGATION

The main aim of preliminary analysis is to identify the problem. First, need for the new or enhanced system is established. Only after the recognition of need, for the processed system done then further analysis is possible.

Once the initial investigation is done and the need for new or improved system is established, all possible alternate solutions are chalked out. All those systems are known as "candidate system". All the candidate systems are then weighed and the best alternative of all those is selected as the solution system, which is termed as the "proposed system". The proposed system is then evaluated of its feasibility. Feasibility for a system means whether it is practical and beneficial to build that system.

The system has been designed in such a way that it can be modified with very little effort when such a need arises in the future. The system has been found to work efficiently and effectively. The administrator controls the entry system. He can add new authorities and update the authorities.

FEASIBILITY STUDY

The feasibility study is defined as the practical extent to which a project can be performed successfully. When the client approaches the organization for getting the desired product developed, it comes up with a rough idea about what all functions the software must perform and which all features are expected from the software. This feasibility study is focused on the goal of the organization. This study analyzes whether the software product can be practically materialized in terms of implementation, the contribution of the project to organization, cost constraints and as per values and objectives of the organization. It explores technical aspects of the project and product such as usability, maintainability, productivity and integration ability.

Three key considerations involved in the feasibility analysis are:

1. Technical feasibility
2. Economic feasibility
3. Operational feasibility

1. TECHNICAL FEASIBILITY

Technical feasibility assesses the current resources (includes the hardware and software) and technology. Its center on the existing computer system and to what extent it can support the proposed addition. Since the minimum requirements of the system like internet connection, a web server in the server system is met by an average user.

2. ECONOMICAL FEASIBILITY

The economic analysis is to determine the benefits and savings with the current system and the proposed system that are compared with costs. The system is economically feasible, as the organization possesses the hardware and software resources required for the functioning of the system. Any additional resources, if required, can also be easily acquired. The proposed system was developed with available resources. Since cost input for the software is almost nil the output of the software is always a profit. Hence Software is economically feasible.

3. OPERATIONAL FEASIBILITY

Operational feasibility assesses the extent to which the required software system performs a series of steps to solve business problems and user requirements. This is the longest phase in the development life cycle of a system. So, operational feasibility should be given much importance. The users of the application don't need thorough training on the system. It has a user-friendly interface.

4. SYSTEM SPECIFICATION

HARDWARE SPECIFICATION

The selection of hardware is very important in the existence and proper working of any software. Then selection hardware, the size and capacity requirements are also important.

Processor	:	Intel Pentium Core i3 and above
Primary Memory	:	4GB RAM and above
Storage	:	320 GB hard disk and above
Monitor	:	VGA Color Monitor
Keyboard	:	Windows compatible
Mouse	:	Windows compatible

SOFTWARE SPECIFICATION

One of the most difficult task is selecting software for the system, once the system requirements is found out then we have to determine whether a particular software package fits for those system requirements. The application requirement:

Front end   :   HTML

Language   :   Python

Back End   :   MySQL

Other Software   :   Node JS, Ganache, Truffle Suite

Operating System   :   Windows 8 or above

IDE   :   Pycharm

MODULES

ADMIN

- ❖ Login
- ❖ Change password
- ❖ View user
- ❖ View Publicity added charity news accept and reject
- ❖ Add donation request
- ❖ View donation request
- ❖ View user donation
- ❖ View complaint and send reply
- ❖ View organization
- ❖ View status at each donation title

ORGANIZATION

- ❖ Sign
- ❖ Login
- ❖ Change password
- ❖ View profile
- ❖ Edit profile
- ❖ Add charity related fund raising request and forward in to admin for approval
- ❖ View approved charity funds request
- ❖ View current status charity request

USER

❖ Registration

❖ Login

❖ Change password

❖ View profile

❖ Edit profile

❖ View donation request

❖ Add donation(block chain)

❖ View donation status

❖ View previous donation

❖ Send complaint and view reply

PUBLIC

❖ Forward charity related news to admin

❖ View current status of all donation

5. SYSTEM DESIGN

SYSTEM DESIGN

The detailed design of the system selected in the study phase is accomplished in the design phase and the user-oriented performance specification is covered into a technical design specification. The principle activities performed by the design phase are general system design, input design and the design of the database.

INPUT DESIGN

Input design is the process of converting the user-oriented input data is to make the automation is easy and free from errors. The design of handling input specifies how data are accepted for computer processing. Input design is art of overall system design that needs careful attention and if includes specifying the means by which actions are taken. A system user interacting through a work station must be able to tell the system whether to accept input produce a report or end processing.

The collection of input data is considered to be the most expensive part of the system design. Since the inputs have to be planned in such a manner so as to get the relevant information extreme care is taken to obtain the information if the data going into the system is incorrect then the processing and outputs will magnify these errors. Input design is a part of overall system design, which requires careful attention.

The major objective of the input design is to make the data entry easier, logical and error free. With this objective the screen for the system is developed. The input design requirement such user friendliness, consistent format and interactive dialogue boxes for giving the development of the project. The data entry operator needs to know the space allocated for each field, the field sequence, which must match with source document and the format in which the data is entered.

OUTPUT DESIGN

A quality output is the one, which meets the requirements of the end user and presents the information early. In any system, the results of processing are given to the user through the output. In the output design it is determined how the information is to be displayed for immediate need.

Output design should improve the relationship of the system with user and help in decision making. The objective of the output design is to define the format of all printed documents and of the screens that will be produced by the system. The output has been designed as per the needs of the institution. The suggestions of the user are also taken into consideration while designing the layout and the fields that are to be included in the project.

Operations are required primarily to communicate about the results of the processing to the user and to the management. They also provide the hardcopy of these results for later consultations and rectification. In this system operations are of different from like messages, input boxes and reports. All recent reports can be generated by the system Menu will help the user to select the required outputs.

DATA FLOW DIAGRAM

Data flow diagram is used is to define the flow of the system and the its resources such as information. As the name suggests DFD is a representation of the data flow within the system. It includes processes, data store and external interfaces to the system. External interfaces may be treated as external entities. DFD's can by explode to sub-processes. Any two entities data store must be connected to each other by some process. The process cannot be directly connected to each other. The following are the four major components of the DFD:

- **Process**
- **Data store**
- **Data flow**
- **External Entity**

It is also known as the bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to system, various processing carried out in these data and output data generated by the system. The main reason why this DFD technique is so popular is probably because of facts that DFD is very simple formalism. It is simple to understand and use. A DFD model uses a very limited number of primitive symbols to represent the functions performed by a system and the data flow among these systems. Starting with a set of high-level functions that a system performance of DFD model in hierarchically it represents various sub functions. The data flow diagramming technique also follows a simple set of intuitive concepts and rules.

Process

Process shows the work of the system. Each process has one or more data inputs and produce one or more data outputs. Process is represented by rounded rectangles, circles or ovals in DFD's. Each process has a unique name and number. This name and number appear inside the rectangle that represents the process in a DFD.

Data store

Data store is a repository of data. Processes can enter data into or retrieve the data from data store. Each data has a unique name.

Data flow

Data flow show the passage of data in the system and represented by lines joining system components. An arrow indicates the direction of flow and the line is labeled by name of the data flow.

External Entity

External Entities are outside the system but they either supply input data into the system or use other system output. They are represented by rectangles. They are entities on which the designer has control. They may be any natural person, corporation, partnership, sole proprietorship, association, organization, company, government agency etc.
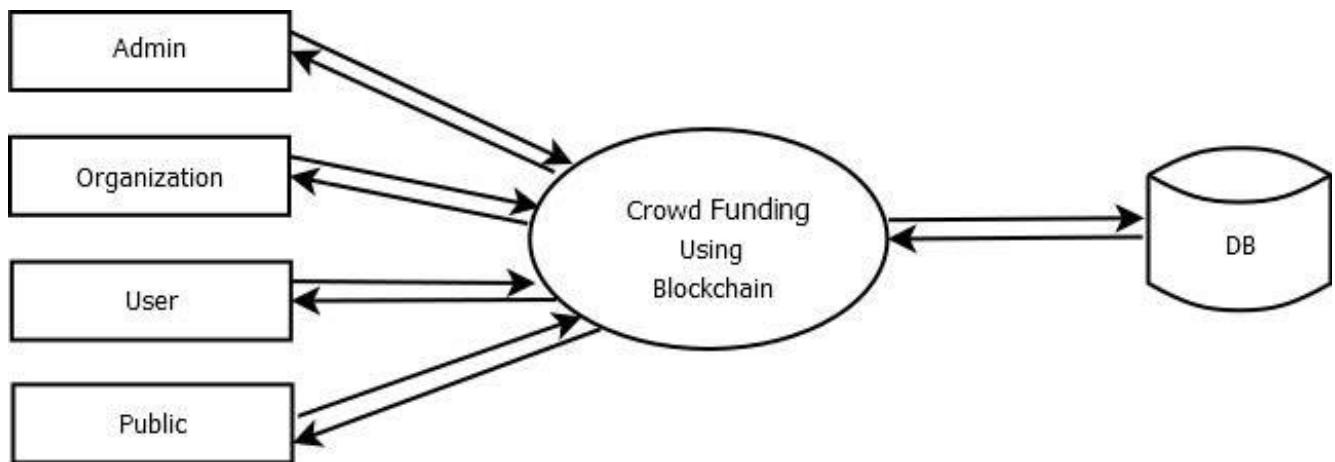
DFDs

LEVEL 0
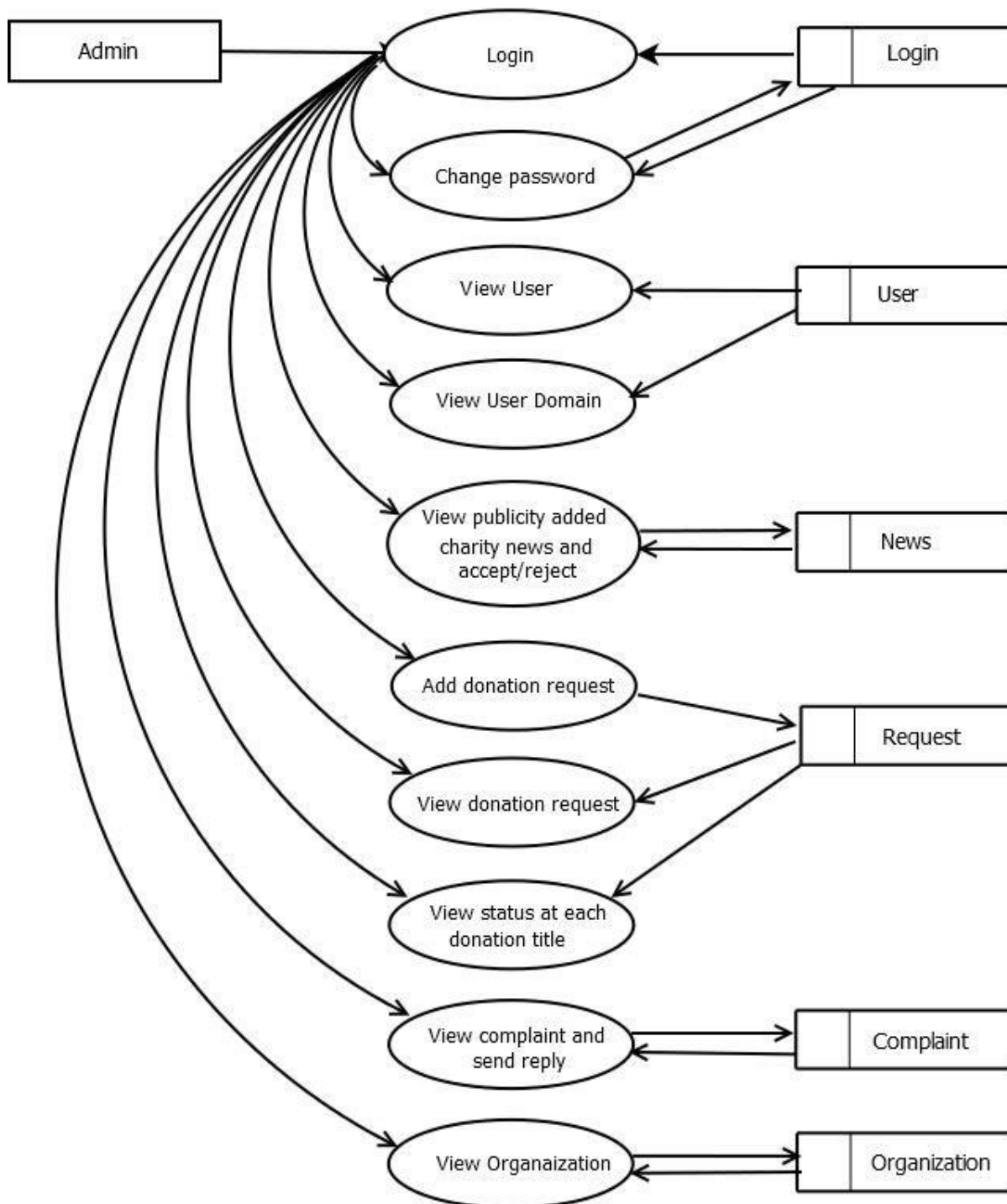

Figure 1 (Level 0 Diagram)
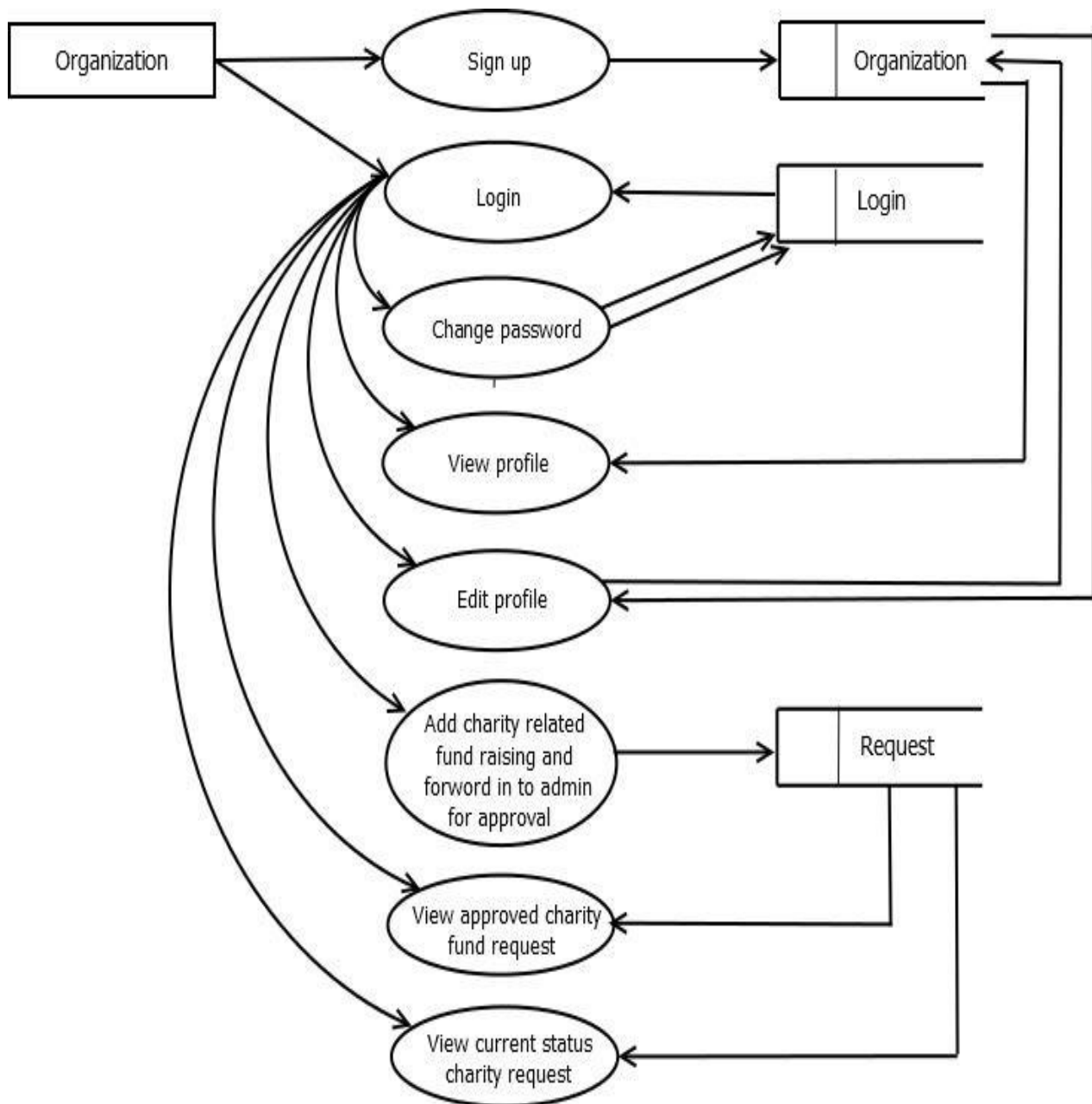
LEVEL 1

Figure 2 (Level 1 Diagram)

LEVEL 2

Figure 3 (Level 2 Diagram)
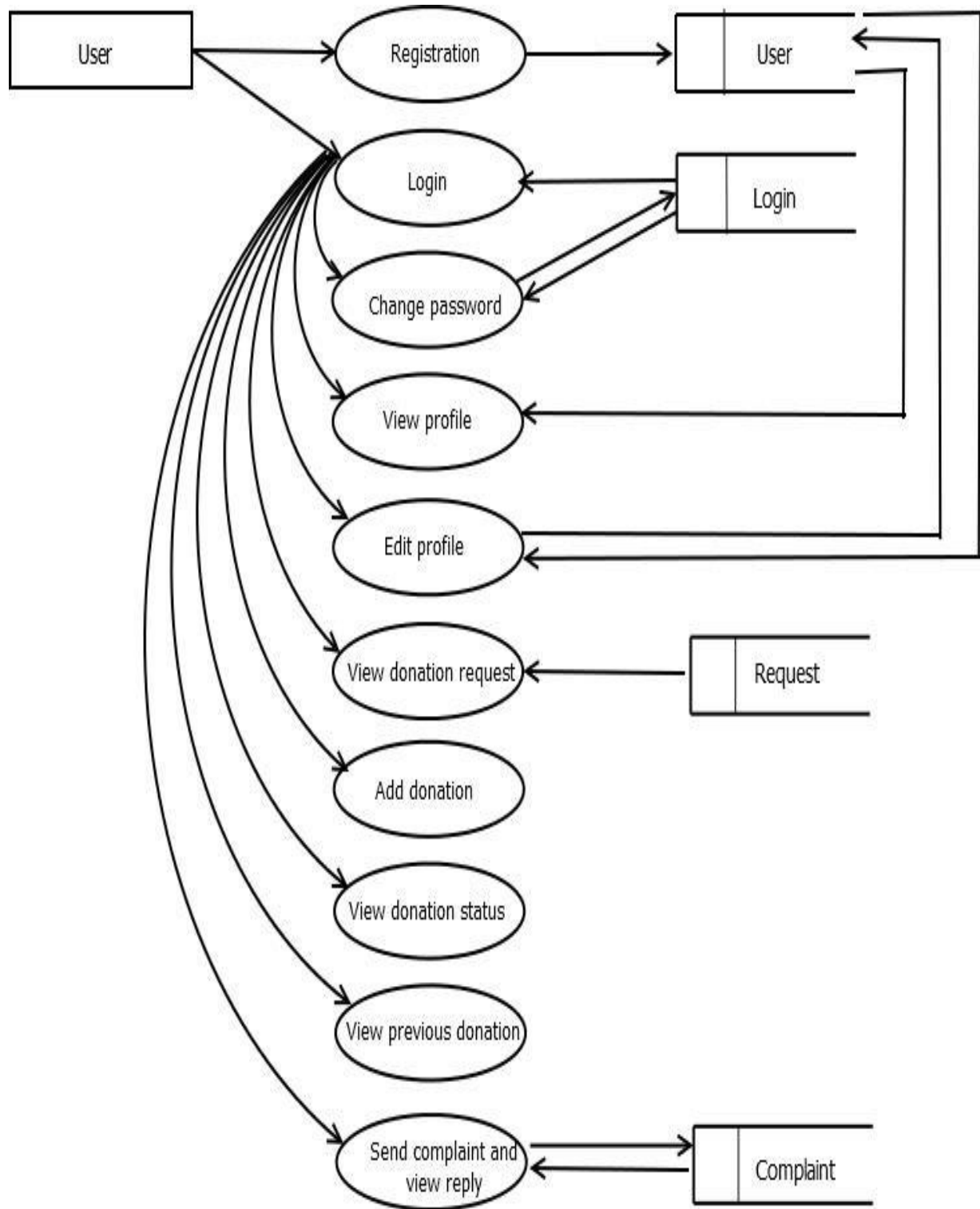
LEVEL 3

Figure 4 (Level 3 Diagram)

LEVEL 4

Figure 5 (Level 4 Diagram)

DATABASE DESIGN

      Database design is one of the most important parts of the system design phase. In a database environment, common data are available and are used by several users. Instead of each program managing its own data, authorized users share data across the application with the database software managing the data as an entity. The primary objective of database design is fast response time to enquiries, more information at low cost, control of redundancy, clarity and ease of use, date and program independence, accuracy and integrity of the system, fast recovery and availability of powerful end-user languages. The theme behind a database is to handle information as an integrated whole, thus the main objective is to make the information as to access easy, quick, inexpensive and flexible for the users. Data directory specifies the major element in the system, and care should be taken while designing, in order to avoid unnecessary duplication of data. The entire package depends on, how the data are maintained in the system. Several tables are maintained in the system to store data that are required for the processing of various data as well as storing intermediate or final processed results.

      Database design mainly aims at handling large volumes of information, involving the definitions for the structure of storage and provisions for the manipulation of information, providing safety of information despite system crashes due to unauthorized access.

Some conditions are satisfied in the database design stage:

- Control redundancy.
- Ease of use.
- Data independence.
- Accuracy and integrity.
- Recovery from failures.
- Security and privacy.
- Performance.

NORMALISATION

It is a process of efficiency organizing data in database. Normalization has two main goals. One is to eliminating redundant data, ensuring data dependencies make sense. They reduce the amount of space a database consumes, and ensure that is logically stored.

▪ **First normal form**:

First normal form eliminates duplicate columns from the same table. It creates separate table for each group of related data and identify each row with a unique column or set of columns (primary key).

▪ **Second normal form**:

Second normal form meets all the requirements of the first normal form. It removes subset of data that apply to multiple rows of a table and place them in separate tables. Create relationship between these new tables and their predecessors through the use of foreign keys.

▪ **Third normal form**:

Third normal form meets all the requirements of the second normal form. It removes columns that are not dependent upon the primary key.

TABLE

Table 1 (Login)

FIELD NAME	DATA TYPE	LENGTH	CONSTRAINTS
login_id	int	10	Primary key
username	varchar	20	
password	varchar	20	
type	varchar	20	

Table 2 (User)

FIELD NAME	DATA TYPE	LENGTH	CONSTRAINTS
login_id	int	10	Primary key
login_id	int	10	Foreign key
name	varchar	20	
phone_no	bigint	20	
place	varchar	20	
post	varchar	20	
pin	varchar	20	
email	varchar	20	
gender	varchar	20	
photo	varchar	20	
Dob	date	10	

Table 3 (News)

FIELD NAME	DATA TYPE	LENGTH	CONSTRAINTS
login_id	int	10	Primary key
date	date	20	
title_of_news	varchar	20	
news_file	varchar	20	
status	varchar	20	

Table 4 (Requester)

FIELD NAME	DATA TYPE	LENGTH	CONSTRAINTS
login_id	int	10	Primary key
date	date	20	
user_id	int	20	
amount	int	20	
status	varchar	20	
organization	varchar	20	Foreign key

Table 5 (Complaints)

FIELD NAME	DATA TYPE	LENGTH	CONSTRAINTS
login_id	int	10	Primary key
date	date	20	
user_id	int	20	Foreign key
status	varchar	20	
reply	varchar	20	

Table 6 (Organization)

FIELD NAME	DATA TYPE	LENGTH	CONSTRAINTS
login_id	int	10	Primary key
login_id	int	10	Foreign key
org_name	varchar	20	
phone_no	big int	20	
place	varchar	20	
post	varchar	20	
pin	varchar	20	
email	varchar	20	
photo	varchar	20	

6. SYSTEM   DEVELOPMENT

BRIEF DESCRIPTION ABOUT THE
TOOLS/SCRIPTS FOR IMPLEMENTATION

Front-end Development:

BOOTSRAP

Bootstrap is a popular open-source framework for developing responsive and mobile-first websites. It provides a comprehensive toolkit of pre-designed HTML, CSS, and JavaScript components, making it easier for developers to create modern web interfaces. With a grid system, extensive component library, and powerful customization options, Bootstrap allows for consistent design and efficient development. Its ease of use and robust documentation make it a go-to choice for both beginners and experienced developers.

HTML

HTML (Hyper Text Markup Language) is the standard language used to create and structure content on the web. It provides the foundation for web pages, allowing developers to define elements such as headings, paragraphs, links, images, and forms. HTML uses a system of tags and attributes to organize content, making it interpretable by web browsers. As a cornerstone of web development, HTML works in conjunction with CSS and JavaScript to create visually appealing and interactive websites.

Back-end Development with Python Django:

Tools:

Django: A high-level Python web framework that encourages rapid development and clean, pragmatic design. Django includes an ORM, an admin panel, and various built-in functionalities for building robust web applications.

Django REST Framework (DRF): A powerful and flexible toolkit for building Web APIs in Django. DRF makes it easy to create RESTful APIs with Django.

Scripts:

Django CLI: Command-line tools provided by Django for managing the backend application. Common commands include:

django-admin startproject <project_name>: Creates a new Django project.

python manage.py startapp <app_name>: Creates a new app within the Django project.

python manage.py migrate: Applies database migrations.

python manage.py runserver: Runs the Django development server.

python manage.py createsuperuser: Creates an admin user for the Django admin panel.

SOURCE CODE

```python
import datetime

from django.core.files.storage import FileSystemStorage
from django.http import HttpResponse, JsonResponse
from django.shortcuts import render


# Create your views here.
from myapp import admin
from myapp.models import *



import json
from web3 import Web3, HTTPProvider
blockchain_address = 'HTTP://127.0.0.1:7545'
try:
    web3 = Web3(HTTPProvider(blockchain_address))
    web3.eth.defaultAccount = web3.eth.accounts[0]
    deployed_contract_address = "0xB8fdf5Ac9e3DdB6a41127AEa0e563E9ce85270F8"
    compiled_contract_path =   'C:\\Users\\athir\\crowdfundingbc\\build\\contracts\\Crowdfunding.json'
    deployed_contract_addressa = web3.eth.accounts[4]
    with open(compiled_contract_path)as file:
        contract_json = json.load(file)
        contract_abi = contract_json['abi']
    contract = web3.eth.contract(address=deployed_contract_address, abi=contract_abi)
except:pass
```

Home page

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title</title>
</head>
<body>
{#<a href="/myapp/org_registration/">org_registraion</a><br>#}
<a href="/myapp/view_profile_org/">view_profile_org</a><br>
 <a href="/myapp/edit_org/">edit_org</a><br>
{#<a href="/myapp/complaint/">complaint</a><br>#}
<a href="/myapp/view_profile_user/">view_profile_user</a><br>
</body>
</html>

<!DOCTYPE html>

  <html lang="en">

  <head>

 <meta charset="UTF-8">

  <title>Title</title>

 </head>

 <body>

 <a href="/myapp/change_password/">change_password</a><br>

 <a href="/myapp/view_user/">view_user</a><br>

 <a href="/myapp/view_news/">view_news</a><br>

  <a href="/myapp/view_complaint/">view_complaint</a><br>

  <a href="/myapp/view_organization/">view_organization</a><br>

  <a href="/myapp/view_request/">view_request</a><br>

 </body>

  </html>
```

```dart
import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';

import 'package:http/http.dart' as http;

import 'dart:convert';
import 'package:shared_preferences/shared_preferences.dart';


import 'package:shared_preferences/shared_preferences.dart';

import 'Change password.dart';
import 'Edit profile.dart';
import 'View donation status.dart';
import 'View profile.dart';
import 'login.dart';


void main() {
 runApp(const HomeNew());
}

class HomeNew extends StatelessWidget {
const HomeNew({super.key});

@override
Widget build(BuildContext context) {
 return MaterialApp(
  title: 'Home',
  theme: ThemeData(

   colorScheme: ColorScheme.fromSeed(seedColor: Color.fromARGB(255, 18, 82, 98)),
   useMaterial3: true,
  ),
  home: const HomeNewPage(title: 'Home'),
 );
}
}

 class HomeNewPage extends StatefulWidget {
const HomeNewPage({super.key, required this.title});

final String title;

@override
State<HomeNewPage> createState() => _HomeNewPageState();
}
```

MODULE HIERARCHY

Admin:

The admin module hierarchy includes User Management (Login, Change Password, View User), Charity News Management (Approve/Reject News), Donation Management (Add/View Requests, View User Donations), Complaint Management (View/Reply to Complaints), Organization Management (View Organizations), and Status Management (View Status at Each Donation Title.

Organaization:

The organization module includes registration (Sign), account access (Login, Change Password), profile management (View/Edit Profile), fundraising (Add Charity Fundraising Request, Forward for Admin Approval), and status tracking (View Approved Requests, View Current Status of Requests), streamlining charity activity management for organizations.

User:

The user module includes functionalities for Registration, Login, Change Password, View/Edit Profile, View Donation Requests, Add Donations, View Donation Status, View Previous Donations, and Send Complaints and View Replies. This module ensures users can manage their accounts, track donations, and communicate issues effectively.

Public:

The public module includes functionalities for submitting charity-related news for admin review and approval, ensuring transparency and dissemination of charitable activities to the wider community.

PROBLEM ENCOUNDED

Crowdfunding using blockchain technology offers several advantages, such as transparency, security, and lower transaction costs. However, it also presents a unique set of challenges. Here are some common problems encountered with blockchain-based crowdfunding

Regulatory Issues: The regulatory environment for blockchain and cryptocurrencies is still evolving. Different countries have varying regulations, and compliance can be complex and costly. It is the first level of testing. Each module is tested individually and focus is given for finding errors limited to each individual module and correcting them. The different modules of the system are tested individually and corrected all errors. Each module is focused to work satisfactorily with regard to the expected output from the module.

Security Concerns: Despite blockchain's inherent security features, smart contracts and wallets can be vulnerable to hacks and exploits. High-profile breaches can undermine trust in the platform. It is the first level of testing. Each module is tested individually and focus is given for finding errors limited to each individual module and correcting them. The different modules of the system are tested individually and corrected all errors. Each module is focused to work satisfactorily with regard to the expected output from the module.

Technical Complexity: Blockchain technology can be complex for average users. This complexity can deter potential backers who are not familiar with cryptocurrency and blockchain concepts. It is the first level of testing. Each module is tested individually and focus is given for finding errors limited to each individual module and correcting them. The different modules of the system are tested individually and corrected all errors. Each module is focused to work satisfactorily with regard to the expected output from the module.

7.SYSTEM TESTING AND IMPLEMENTATION

SYSTEM TESTING

In general, testing is finding out how well software or a product works. Testing is a set of activity that can be planned and conducted systematically. Testing begins at the module level and work towards the integration of entire computers-based system. Nothing is complete without testing, as it is vital success of the system.

Software testing is a process of executing a program or application with the intent of finding the software bugs. Testing is for evaluating a system or its components to find whether it satisfies the specified requirements or not. Testing cannot show the absence of defects, it can also be stated as the process of validating and verifying that a software program or software or product.

TESTING TECHNIQUES

- Unit testing
- Integration Testing
- Validation Testing
- System Testing
- Output Testing
- User Acceptance Testing

UNIT TESTING

It is the first level of testing. Each module is tested individually and focus is given for finding errors limited to each individual module and correcting them. The different modules of the system are tested individually and corrected all errors. Each module is focused to work satisfactorily with regard to the expected output from the module.

INTEGRATION TESTING

Integration testing is a systematic testing for construction the program structure while as the same time conducting tests to uncover Instead of testing the system as a whole, unit testing focuses on the module that make up the system. Each module is taken up individually and tested for correctness in coding and logic. Error resulting from interaction of module is initially avoided. It tests for the errors resulting from integration of modules. One specifies target of integration testing is the interface, whether errors associated with interfacing. The objective is to take until- tested modules and build a program structure that has been dictated by design.

VALIDATION TESTING

Administrator has to specify the username and password. When the user enters username and password, checking it with the already registered username and password in the database will validate it. If they do not match, user is defined access there by providing a strong security.

SYSTEM TESTING

System testing was performed to verify that all system elements have been properly integrated and perform allocated function. Security testing was done to check the security mechanisms built into the system, which will protect it from improper penetration, performance testing was done to test the runtime performance of the software. For user acceptance testing the system was given to the end user to use.

OUTPUT TESTING

After performing the validation testing, the next step is the output testing of the enhanced system. No system could be useful if it does not produce the required output in the required format. The outputs generated or the displayed by the system are tested by asking the users about the format required by them.

USER ACCEPTANCE TESTING

User acceptance testing comprises a completed and successful end-to-end system test with review of the results by one or more users with specific knowledge. Users may apply a variety of validation techniques. For example: generate a report from the new system and compare the results with the same report from the current system, data inspection and others.

Prior to this testing the system was delivered to the clients along with the steps for implementation. In user acceptance testing the system was run in the client environment by system users. The users were allowed to test the system and raise any issues for a specific period after the system is assumed to be complete and error free.

SYSTEM IMPLEMENTATION

A software implementation method is integrating software-based service systematically. This is the phase in the software life cycle where the actual software is implemented. The result of this phase consists of source code, together with documentation to make the code more readable. Implementation is the stage of a project where the theoretical design is turned into a working system. If the implementation is not carefully planned and controlled, it may cause confusion. Implementation is done only after testing is done. The most crucial stage is achieving a new successful system and giving confidence about the new system for the user that it will work according to the requirements. It involves careful planning, investigation of the current system and its constraints on implementation. The more complex the system is, the more effort is needed to implement the system. In other words, implementation is the process of bringing a developed system into operational use and turning over it to the user implementation activities extend from planning through the old system to the new one.

The implementation involves the following formalities:

- Careful planning.
- Investigation of the systems and constraints.
- Design the methods to achieve the changes.
- Training the staffs in the changed phase.
- Evaluation of the changes over method.

8.SYSTEM MAINTENANCE

SYSTEM MAINTENANCE

A process of modifying a software system or component after delivery to correct faults, to improve performance is known as software maintenance. A common perception of maintenance is that it merely involves fixing defects. However, one study indicated that the majority, over 80%, of the maintenance effort is used for non-corrective actions. Software maintenance is a very broad activity that includes error correction, enhancements of capabilities, deletion of obsolete capabilities, and optimization. Because change is inevitable, mechanisms must be developed for evaluation, controlling and making modifications. So, any work done to change the software after it is in operation is considered to be maintenance work. The purpose is to preserve the value of software over the time. The value can be enhanced by expanding the customer base, meeting additional requirements, becoming easier to use, more efficient and employing newer technology. Maintenance may span for 20 years, whereas development may be 1-2 years.

Corrective maintenance is done to repair the faults or defects found in day –to – day system functions, that is software design, logic and coding errors. Adaptive maintenance is the implementation of changes in a part of the system which has been affected by changes that occurred in some other part of the system. The objective of perspective maintenance should be to prevent failures and optimize the software. Minor adaptive changes should be handled by normal maintenance process. Major adaptive changes should be carried out as a separate development project.

9. CONCLUSION

CONCLUSION

Crowdfunding using blockchain technology represents a transformative approach to fundraising, offering unprecedented levels of transparency, security, and accessibility. By leveraging blockchain's decentralized ledger, crowdfunding platforms can eliminate intermediaries, reduce transaction costs, and enhance trust among participants

One of the key advantages of blockchain in crowdfunding is transparency. Every transaction and donation is recorded on the blockchain, visible to all participants in real-time. This transparency builds trust, as donors can verify where their contributions are allocated and ensure they align with the stated goals of the crowdfunding campaign. Moreover, smart contracts can automate the release of funds when predefined conditions are met, reducing the risk of fraud or mismanagement.

Security is another critical benefit. Blockchain's cryptographic features protect transaction data from tampering, ensuring that funds are securely transferred and stored. This reduces the risk of hacks or unauthorized access, which is particularly crucial in online fundraising where data breaches can undermine donor confidence. User acceptance testing comprises a completed and successful end-to-end system test with review of the results by one or more users with specific knowledge.

Blockchain also enhances accessibility by enabling global participation without geographical restrictions or reliance on traditional banking infrastructure. Anyone with internet access can contribute to crowdfunding campaigns, empowering individuals and communities worldwide to support causes they believe in.

In conclusion, blockchain-based crowdfunding revolutionizes fundraising by promoting transparency, enhancing security, and broadening access. As this technology continues to evolve, its potential to reshape how charitable donations are collected, managed, and distributed holds promise for a more inclusive and accountable approach to philanthropy.

SCOPE FOR FUTURE ENHANCEMENT

The scope for future enhancements of crowd funding using blockchain is vast and promising, aiming to further refine user experience and expand the platform's capabilities. Here are several potential areas for future development: The future enhancements could be as follows:

❖ Scalability: Improving blockchain scalability to handle a larger volume of transactions efficiently, thereby accommodating widespread adoption without congestion or delays.

❖ Interoperability: Enhancing interoperability between different blockchain platforms and cryptocurrencies to facilitate seamless cross-platform donations and transactions.

❖ Enhanced Smart Contracts: Developing more sophisticated smart contracts that can automate complex conditions and disburse funds based on verifiable milestones or achievements, increasing trust and accountability.

❖ Tokenization: Exploring tokenization of assets to represent ownership or rights within crowdfunding campaigns, enabling fractional ownership and diverse investment opportunities.

❖ Governance and Transparency: Implementing decentralized governance mechanisms where stakeholders can participate in decision-making processes and transparently audit fund utilization through immutable blockchain records.

❖ Integration with IoT and AI: Leveraging Internet of Things (IoT) devices and Artificial Intelligence (AI) to track project milestones in real-time and provide predictive analytics on campaign success and impact.

❖ Enhanced Security Measures: Continuously improving security protocols to protect against evolving cyber threats and ensure donor data privacy.

❖ Community Engagement: Facilitating community engagement through blockchain-based voting mechanisms, allowing stakeholders to prioritize and fund projects democratically.

❖ Global Accessibility: Expanding access to blockchain-based crowdfunding platforms in underserved regions, empowering diverse communities to participate in global philanthropic efforts.

In essence, future enhancements in blockchain-based crowdfunding aim to foster greater efficiency, transparency, and inclusivity in fundraising while addressing technological, regulatory, and societal challenges to realize its full potential.
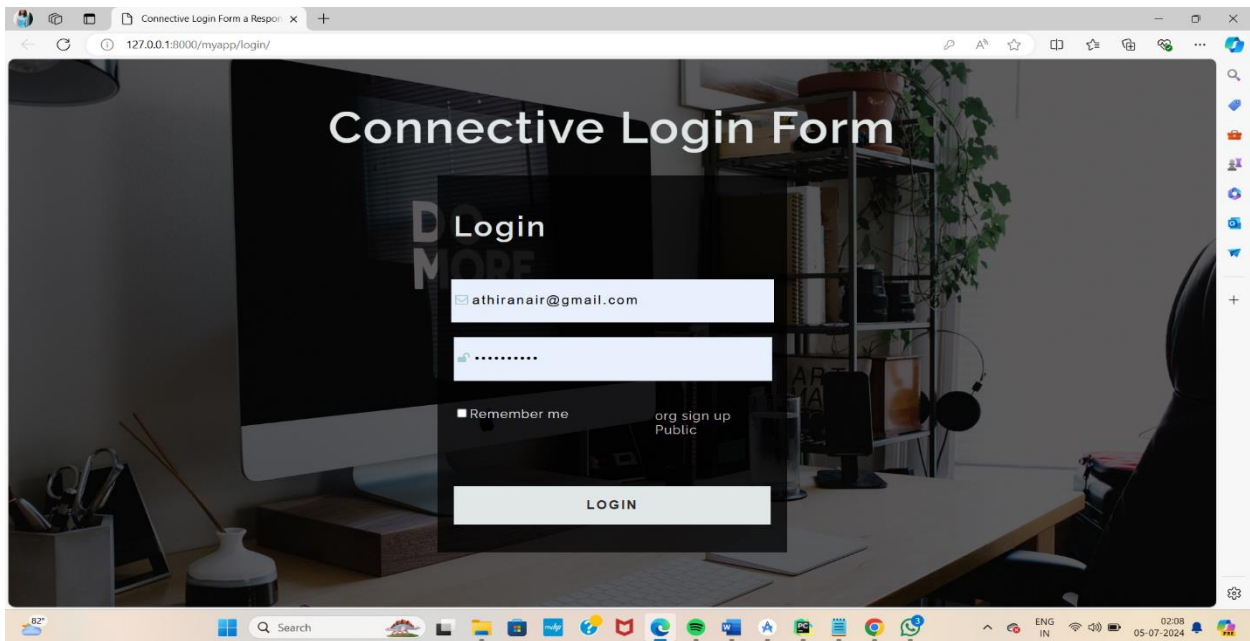
10.BIBLIOGRAPHY

BIBLIOGRAPHY

WESBITES

△ **www.fullstackpython.com**

△ **http://realpython.com/**

△ **http://www.mysql.com/**

△ **http://www.w3schools.com/**

11.APPENDIX

SCREENSHOTS

LOGIN:

ADMIN HOME PAGE:
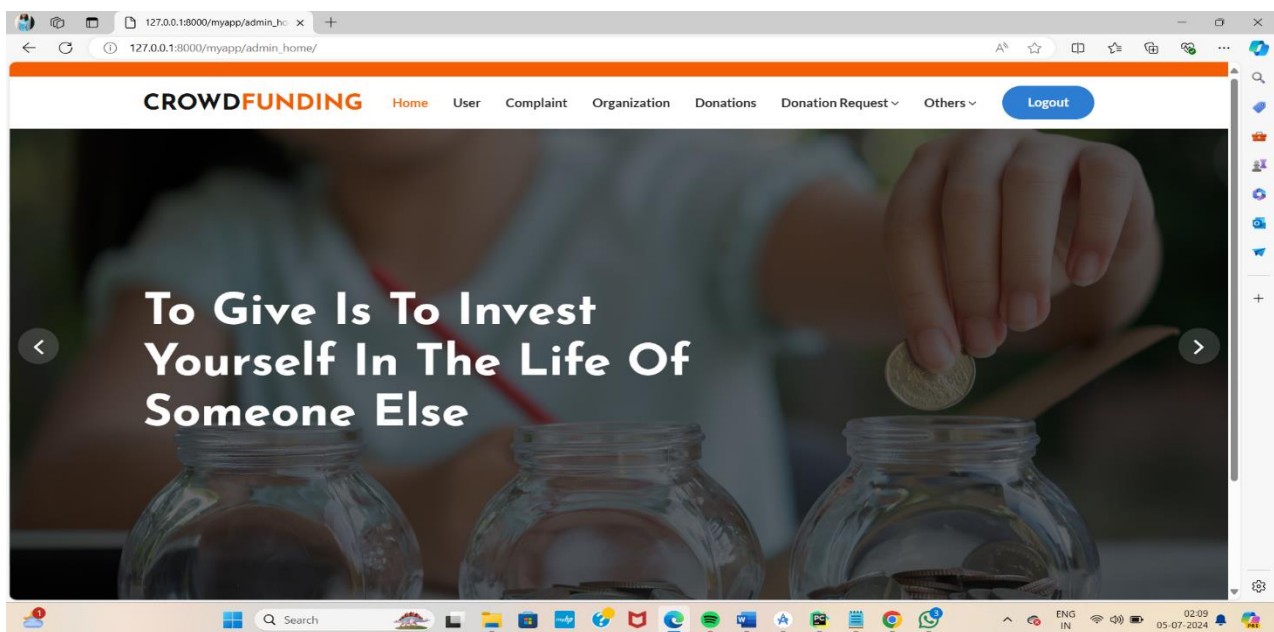
ORGANIZATION REGISTER:

ORGANIZATION HOME:
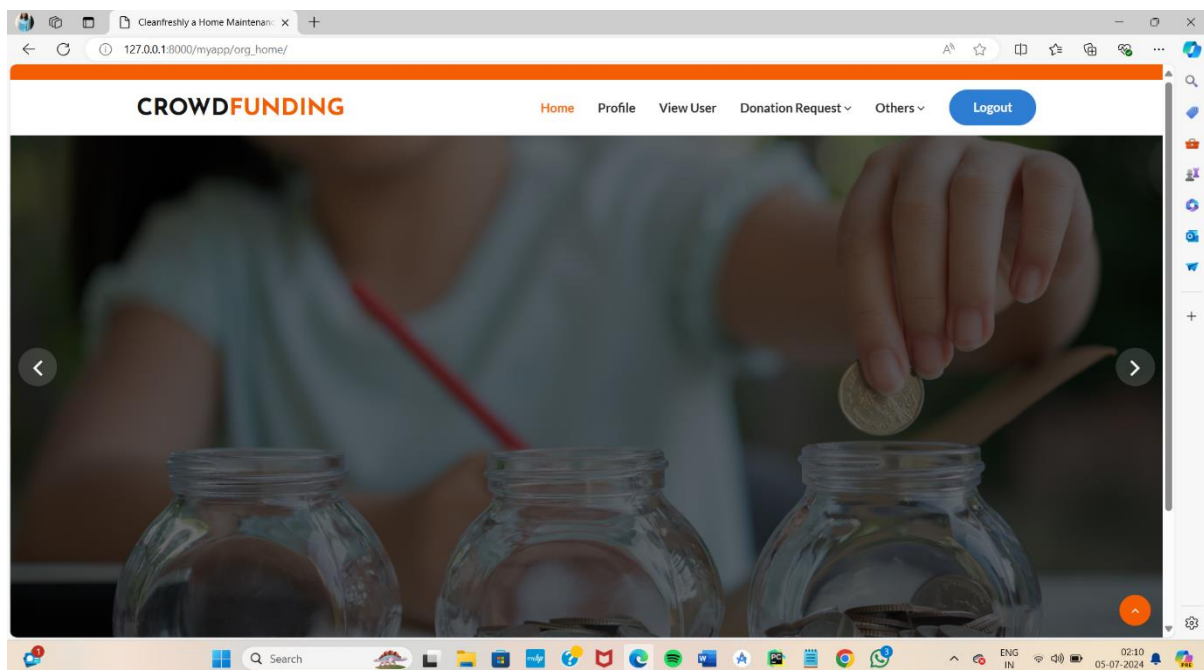
PUBLIC HOME:
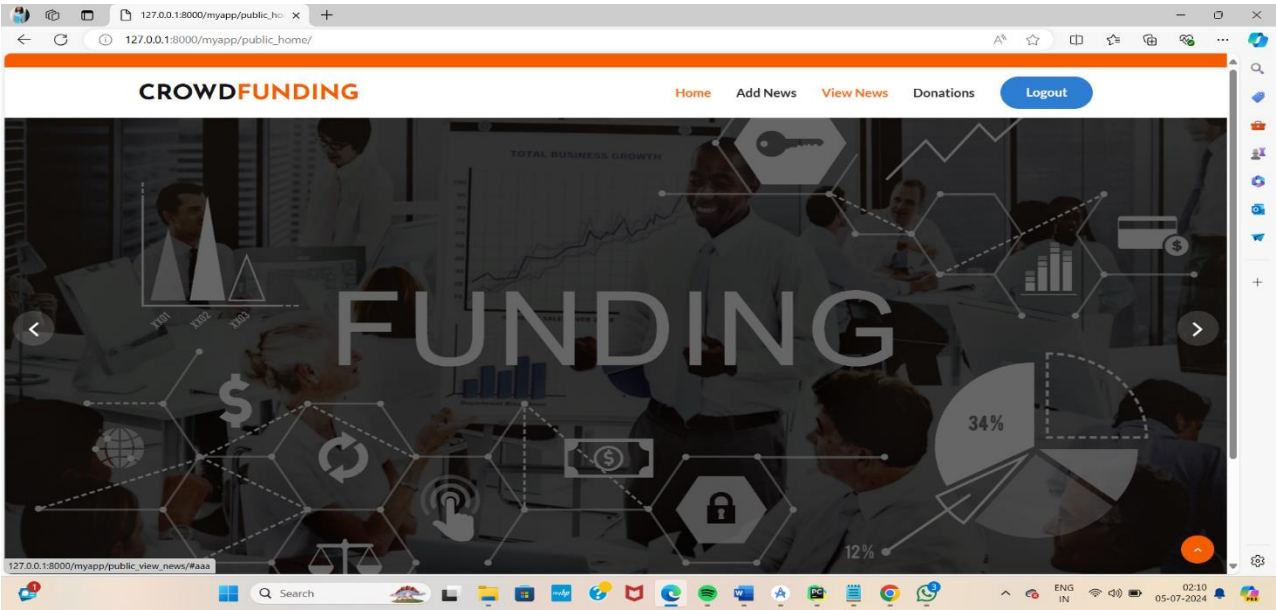
ADMIN SHOT:

USER PAGE SHOTS: