

# **PREVENTION AND DETECTION OF FAKE CHEQUE SCAMS USING BLOCK CHAIN**

**A PROJECT REPORT**

*Submitted by*

**SRIMANJARI P [211418104262]  
SUBRINI K B [211418104270]  
SWETHA A R [211418104283]**

*in partial fulfillment for the award of the degree*

*of*

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**



**PANIMALAR ENGINEERING COLLEGE**

**(An Autonomous Institution, Affiliated to Anna University,  
Chennai)**

**MAY 2022**

**PANIMALAR ENGINEERING COLLEGE**  
**(An Autonomous Institution, Affiliated to Anna University, Chennai)**

**BONAFIDE CERTIFICATE**

Certified that this project report **“PREVENTION AND DETECTION OF FAKE CHEQUE SCAMS USING BLOCK CHAIN”** is the bonafide work of **“SRIMANJARI P [REGISTER NO: 211418104262], SUBRINI K B [REGISTER NO: 211418104270], SWETHA A R [REGISTER NO: 211418104283]”** who carried out the project work under my supervision.

**SIGNATURE**

**Dr.S. MURUGAVALLI, M.E., Ph.D.,**  
**HEAD OF THE DEPARTMENT**

DEPARTMENT OF CSE,  
PANIMALAR ENGINEERING COLLEGE,  
NASARATHPETTAI,  
POONAMALLEE,  
CHENNAI-600 123.

**SIGNATURE**

**Mrs.S.T. SANTHANALAKSHMI**  
**SUPERVISOR**  
**ASSISTANT PROFESSOR**  
**(GRADE I)**

DEPARTMENT OF CSE,  
PANIMALAR ENGINEERING COLLEGE,  
NASARATHPETTAI,  
POONAMALLEE,  
CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project

Viva-Voce Examination held on.....

**INTERNAL EXAMINER**

**EXTERNAL EXAMINER**

## **DECLARATION BY THE STUDENT**

We **SRIMANJARI P [211418104262]**, **SUBRINI K B [211418104270]**, **SWETHA A R[211418104283]** hereby declare that this project report titled **“PREVENTION AND DETECTION OF FAKE CHEQUE SCAMS USING BLOCK CHAIN”** , under the guidance of **Mrs.S.T.SANTHANALAKSHMI ,ASSISTANT PROFESSOR (GRADE I)** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

**1. SRIMANJARI P**

**2. SUBRINI K B**

**3. SWETHA A R**

## ACKNOWLEDGEMENT

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P. CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI, Dr.C.SAKTHIKUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.**, for providing us with the necessary facilities to undertake this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** who facilitated us in completing the project.

We thank the Head of the CSE Department, **Dr. S. MURUGAVALLI, M.E., Ph.D.**, for the support extended throughout the project.

We would like to thank my **Project Guide Mrs.S.T. SANTHANALAKSHMI, ASSISTANT PROFESSOR (GRADE I)** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

**SRIMANJARI P  
SUBRINI K B  
SWETHA A R**

## **ABSTRACT**

Fake cheque scam is one of the most common attacks used to commit fraud against consumers. Currently, there is no existing solution to authenticate cheques and detect fake ones instantly. Instead, banks must wait for a period of more time and date to detect the scam. More precisely, our approach helps the banks to share information about provided cheques and used ones, without exposing the banks' customers' personal data. Fake cheques come in many forms. They might look like business or personal cheques, cashier's cheques, money orders, or a cheque delivered electronically. These scams work because fake cheques generally look just like real cheques, even to bank employees. They are often printed with the names and addresses of legitimate financial institutions. Banks are maintaining security services for money transaction to customer security purpose directly cheque with customer and requires their permission. It also charity trust their have Register login and approved by main branch and charity needed send to user and if user need to help charity, they can help by give cheque to charity which passed to bank and amount is transaction to charity and the details about charity amount transaction will send to main branch and maintained by main branch. If response received from concern customer only, transaction will be happened.

# TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	<b>ABSTRACT</b>	v
	<b>LIST OF TABLES</b>	viii
	<b>LIST OF FIGURES</b>	ix
	<b>LIST OF SYMBOLS, ABBREVIATIONS</b>	xi
<b>1.</b>	<b>INTRODUCTION</b>	
	1.1 Problem Definition	2
	1.2 Scope of the project	2
<b>2.</b>	<b>LITERATURE SURVEY</b>	3
<b>3.</b>	<b>SYSTEM ANALYSIS</b>	
	3.1 Existing System	10
	3.2 Proposed system	10
	3.3 Feasibility Study	11
	3.4 Hardware Environment	12
	3.5 Software Environment	12
<b>4.</b>	<b>SYSTEM DESIGN</b>	
	4.1. ER diagram	14
	4.2 Data dictionary	15
	4.3 Data Flow Diagram	18
	4.4 UML Diagram	20

<b>CHAPTER NO.</b>	<b>TITLE</b>	<b>PAGE NO.</b>
<b>5.</b>	<b>SYSTEM ARCHITECTURE</b>	
	5.1 System Architecture	27
	5.2 Module Design Specification	28
	5.3 Algorithms	33
<b>6.</b>	<b>SYSTEM IMPLEMENTATION</b>	
	6.1 Client-side coding	36
	6.2 Server-side coding	54
<b>7.</b>	<b>SYSTEM TESTING</b>	
	7.1 Test Cases & Reports	72
<b>8.</b>	<b>CONCLUSION</b>	
	8.1 Conclusion	76
	8.2 Future Enhancements	76
	<b>APPENDICES</b>	
	A.1 Sample Screens	77
	<b>REFERENCES</b>	84

## LIST OF TABLES

TABLE NO.	TABLE DESCRIPTION	PAGE NO.
4.1	CHARITY REGISTER	15
4.2	PUBLIC REGISTER	15
4.3	CHARITY REQUEST	16
4.4	CHEQUE DETAILS	16
4.5	MONEY TRANSFERED	17
4.6	MONEY TRANSFERRING	17
7.1	CHARITY MANAGEMENT	72
7.2	CHARITY	72
7.3	PUBLIC	73
7.4	BANK	74

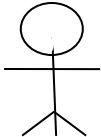


## LIST OF FIGURES

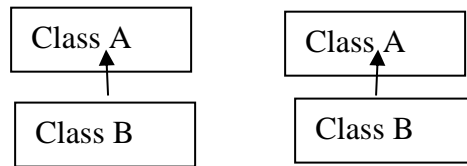
FIG NO.	FIGURE DESCRIPTION	PAGE NO.
4.1	ER DIAGRAM	14
4.2	DFD LEVEL 0	18
4.3	DFD LEVEL 1	18
4.4	DFD LEVEL 2	19
4.5	USE CASE DIAGRAM	20
4.6	STATE DIAGRAM	21
4.7	ACTIVITY DIAGRAM	22
4.8	CLASS DIAGRAM	23
4.9	SEQUENCE DIAGRAM	24
4.10	COLLABORATION DIAGRAM	25
5.1	SYSTEM ARCHITECTURE	27
5.2	CHARITY REGISTER	28
5.3	CHARITY LOGIN	28
5.4	CHARITY REQUEST	29
5.5	CHARITY REQUEST VIEW	29
5.6	CHARITY DONATION VIEW	29
5.7	BANK RESPONSE VIEW	30
5.8	VIEW ACCOUNT BALANCE	30
5.9	PUBLIC REGISTER	31
5.10	PUBLIC LOGIN	31
5.11	RESPONSE	31
5.12	ADMIN LOGIN	32
5.13	RESPONSE (BANK)	32
A.1	MAIN PAGE	77

A.2	CHARITY MANAGEMENT LOGIN	77
A.3	CHARITY MANAGEMENT MAIN PAGE	78
A.4	CHARITY APPROVE PAGE	78
A.5	CHARITY COMPLAINT VIEW	79
A.6	CHARITY LOGIN PAGE	79
A.7	CHARITY REGISTER PAGE	80
A.8	CHARITY MAIN PAGE	80
A.9	CHARITY ACCOUNT VIEW	81
A.10	PUBLIC LOGIN PAGE	81
A.11	REGISTER PAGE	82
A.12	PUBLIC MAIN PAGE	82
A.13	PUBLIC REQUEST VIEW	83
A.14	CHEQUE VIEW PAGE	83

## LIST OF SYMBOLS

S.NO	NOTATION NAME	NOTATION	DESCRIPTION
1.	Class	<div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 20px;"> <i>+ public</i>  <i>-private</i>  <i># Protected</i> </div> <div style="border: 1px solid black; padding: 5px; text-align: center;"> <i>Class Name</i>  <hr/> <i>-attribute</i>  <hr/> <i>+operation</i>  <i>+operation</i>  <i>+operation</i> </div> </div>	Represents a collection of similar entities grouped together.
2.	Association	<div style="display: flex; align-items: center; justify-content: center; margin-bottom: 10px;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Class A</div> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">NAM</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;">Class B</div> </div> <div style="display: flex; align-items: center; justify-content: center;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Class A</div> <div style="border: 1px solid black; padding: 5px; margin-left: 10px;">Class B</div> </div>	Associations represents static relationships between classes. Roles represents the way the two classes see each other.
3.	Actor		It aggregates several classes into a single class.

4. Aggregation



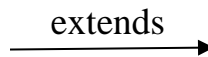
Interaction between the system and external environment

5. **Relation**  
(uses)

uses

Used for additional process communication.

6. Relation  
(extends)



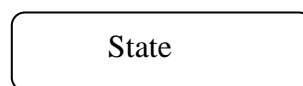
Extends relationship is used when one use case is similar to another use case but does a bit more.

7. Communication



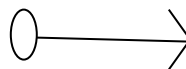
Communication between various use cases.

8. State



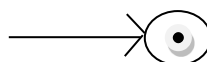
State of the process.

9. Initial State



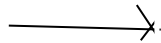
Initial state of the object

10. Final state



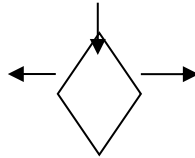
Final state of the object

11. Control flow



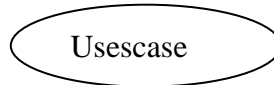
Represents various control flow between the states.

12. Decision box



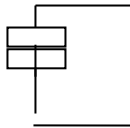
Represents decision making process from a constraint

13.  
Usecase



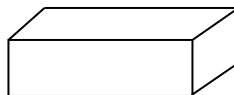
Interact ion between the system and external environment.

14. Component



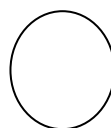
Represents physical modules which is a collection of components.

15. Node



Represents physical modules which are a collection of components.

16. Data  
Process/State



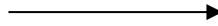
A circle in DFD represents a state or process which has been triggered due to some event or action.

17. External entity



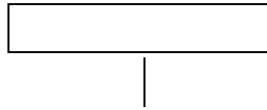
Represents external entities such as keyboard, sensors, etc.

18. Transition



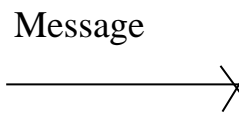
Represents communication that occurs between processes.

19. Object Lifeline



Represents the vertical dimensions that the object communications.

20. Message



Represents the message exchanged.

## **LIST OF ABBREVIATION**

<b>S.NO</b>	<b>ABBREVIATION</b>	<b>EXPANSION</b>
1.	DB	Data Base
2.	SMC	Secure Multiparty Computation
3.	MDA	Medical Admin
4.	DBC	Data Base Confidentiality
5.	JVM	Java Virtual Machine
6.	JSP	Java Server Page

# **1.INTRODUCTION**



# **1.INTRODUCTION**

## **1.1 PROBLEM DEFINITION**

In our current society, cheques represent one of the dominant payment methods. A cheque is an order written by the depositor instructing the bank to pay a specific amount to a recipient from the depositor's bank account. Unfortunately, numerous malicious scammers exploit some flaws in the banking system to commit frauds. Indeed, frauds employing fake cheques are growing rapidly and cost billions of money. we focus on fake cheque scams. This fraud is achieved by getting people mainly through some email scam, establishing a relationship a business relationship most of the time sending them overpaid counterfeit paycheque and finally asking for the overpayment.

## **1.2 SCOPE OF THE PROJECT**

The threat of online fraud has spurred many charity companies and financial institutions to alert consumers when potentially fraudulent transactions are made. Blockchain can help to reduce and even prevent fraud in the supply chain through greater transparency and improved traceability of products. It's very difficult to manipulate the blockchain. In this project bank will return the cheque leaf to the customer when bank find the charity is fake.

## **2.LITERATURE SURVEY**

## **2.LITERATURE SURVEY**

1. **Badis Hammi, Sherali Zeadally, Yves Christian Elloh Adja, Manlio Del Giudice, and Jamel Nebhen, 2021, “Blockchain-Based Solution for Detecting and preventing Fake Cheque Scams” Published in IEEE Transaction on Engineering Management**

This paper uses the methodology Digital Signature Algorithm. In this context, we propose a blockchain-based scheme to authenticate checks and detect fake check scams. Moreover, our approach allows the revocation of used checks. The merit is the signature provides authenticity and ensures that the signature is verified and the demerit is it requires a lot of time to authenticate the verification process.

2. **Emart 77, Beirut, Lebanon, 2021, “The Role of Blockchain in Reducing the Cost of Financial Transactions in the Retail Industry” Published in WCNC**

This paper uses the methodology Consumer Package Goods. The aim of this paper was to assess the role of blockchain in reducing the cost of financial transactions in the retail industry. The merits is Consumer packaged goods (CPG) – and to some extent, services is arguably one of the most challenging industries to be successful in. And the demerit is Being a consumer limited resource that they are forced to accept. This includes the limitations on income and the way these decisions influence consumer interest in a product.

3. **Dilip Kumar Sharma, Sonal Garg, Priya Shrivastava, 2021, “Evaluation of Tools and Extension for Fake News Detection” Published in IEEE International Conference on Innovation Practice in Technology Management (ICIPTM)**

This paper uses the methodology Bi-LSTM classifier. Merit in this It is a sequence prediction model. It is a discriminative classifier which model the decision boundary between different classes and demerit is Since Bi- LSTM has double LSTM cells so it is costly. Not Good fit for Speech Recognition.

4. **Vikash Kumar Aggarwal, Nikhil Sharma, Ila Kaushik, Bharat Bhushan, Himanshu, 2020, “Integration of Blockchain and IoT (B-IoT): Architecture, Solution, & Future Research Direction” Published in 1<sup>st</sup> International Conference on Computational Research and Data Analytics (ICCRDA).**

This paper uses the methodology Integration of Blockchain and IoT. In this paper, we present introductory part of IoT enabled with blockchain, their key features, architecture layout, characteristic features of both the technologies, their futuristic solutions for different real-world problems, different communicational models. The merit is the smart contract is what allows a smart device to function on its own without the need for a centralized authority and the demerit is one of the notable weaknesses of blockchain is scalability while blockchain is not indestructible.

5. **Jackie Jones, Damon McCopy, 2020, “The Cheque is in the Mail: Monetization of Craigslist Buyer Scams” Published in IEEE APWG Symposium.**

This paper uses the Methodology Conservation Classification Strategy. In this paper we extend on previous works about fake payment scams targeting Craigslist. To grow our understanding of scammer methods and how they monetize these scams. The merit is Compared to manual data entry automatic data entry greatly reduced errors. Company documents can be filled out, stored,

retrieved and used more accurately and the demerit is Automated equipment include the high capital expenditure required to invest in automation.

**6. Abiola, Idowu, 2019, “An Assessment of Fraud and its Management in Nigeria Commercial Banks” Published in IEEE Paper European Journal of Social Science.**

This paper uses the Methodology Pearson Product Moment Correlation Coefficient. This aims at finding practical means of minimizing the incidence of fraud in Nigerian banks. During the course of the investigation efforts were made to identify various means employed in defrauding banks and at the same time determine the effects of fraud on the banking services. Merit in this is a correlation coefficient such as Pearson  $r$  is that it provides effect size information (in unit free terms) and the demerit is a key limitation of Pearson's  $r$  is that it cannot distinguish between independent and dependent variables. Therefore, also if a relationship between two variables is found.

**7. Sumeet Kumar, 2018, “Simulating DDOS attacks on the us fiber-optics internet infrastructure” Published in Proceedings of the 2017 Winter Simulation Conference.**

This paper uses the Methodology Cyber-attacks. In this research, we have designed a test-bed that mirrors the Internet infrastructure of the US and can simulate the Internet traffic flow patterns for different attack targets. We also estimate the degradation in the quality-of-service and the number of users impacted in two attack scenarios. The merit is a network simulation model to understand the Internet traffic flow pattern in a DDOS attack situation and the demerit is Altering data through remote access or damage the system causing data loss.

8. **Nazli Ismail Nawang , 2017,“ Combating anonymous offenders in the cyberspace: An overview of the legal approach in Malaysia” Published in conference**

This feature is most commonly cherished by Internet users as they are empowered to publish online content anonymously; or in the alternative to conceal their true identities behind fictitious names (pseudonyms). Merit is It identifies the anonymous offender in the cyber world and demerit is the user must know the various laws that have been enacted to govern publication of illegal content.

9. **Haris S emic, Sasa Mrdovic, 2017, “IoT honeypot: A multi-component solution for handling manual and Mirai-based attacks” Published in: IEEE 25<sup>th</sup> Telecommunication forum TELFOR 2017.**

This paper uses the Methodology Internet-of-things (IoT) device. This paper proposes an implementation of honeypot that detects and reports telnet attacks on Internet-of-Things (IoT) devices. The honeypot operates with manual and Mirai-based attacks. Merit in this It is used to attain sufficient exposure to malicious traffic and security of collected data and demerit is the plank of security has left IoT devices vulnerable to various attacks that aim to take control of said devices and utilize them for various malicious purposes.

10. **Bernie S.Fabito,Angelique D.Lacasandile ,Emeliza R.Yabut ,2017 ,“Leveraging crime reporing in Metro Manila using unsupervised crowd-sourced data: A case for the Report framework” Published in International Conference on control, electronics, renewable energy and communication.**

This paper uses the Methodology Ping ER Monitoring Agent. It aims to provide a venue for victims of crimes to report their experiences without having to go directly to police stations. Those who have experienced the same offense in the

area can link their reports with those previously reported offenses which refer to the same case. Merit is In this layer, existing ping ER scripts which generate hourly, monthly and yearly reports can work seamlessly. This is because, in this framework, contents are addressed through hashes which is a widely used means of connecting data in a distributed network and demerit is the downside of using an always-free tool is that it may be missing key features that you need. And with always-free tools, what you get is what you get.

**11. Wendy Baker-Smemoe, Chair David Eddington William G. Eggington, 2015, “The Language and Cross-Culture Perceptions of Deception” Published in: Brigham young university.**

This paper uses the Methodology Qualtrics Survey Block. research has shown that some linguistic features can indicate a person is lying, this line of research has led to conflicting results. Furthermore, very little research has been done to verify that these supposed linguistic features of deception are universal. In addition, few studies have researched the cross-cultural perceptions of deception, which knowledge could greatly improve the detection of deception across cultures. Merit in this is Rather than creating two surveys, you could create two blocks of questions within one survey and randomly assign participants to one block or the other and the demerit is Qualtrics support could use some work. It can be difficult to find help in the community. It almost has to reach out to someone on the online chat or via email to get help.

### **3.SYSTEM ANALYSIS**



### **3.SYSTEM ANALYSIS**

#### **3.1 EXISTING SYSTEM**

**Concept:**

Fake cheque scam has more disastrous consequences on the victims than many other attacks. In this context, we believe that the best solution to protect users is the detection of fake cheques well before they are cashed.

**Technique:**

Digital Signature Algorithm.

**Disadvantage:**

- It requires lot of time for computation.
- Verification of a given take a look at and, for this reason, avoid the modern drift duration of greater than 48 h.

#### **3.2 PROPOSED SYSTEM**

**Concept:**

To verify the authenticity of a given cheque, without exposing the banks' customers' personal data. To evaluate the performance of our proposed approach, we also deployed our cheque's authentication scheme based on the blockchain.

**Technique:**

SHA algorithm, AES algorithm.

**Advantage:**

- It requires less time than Digital Signature algorithm.
- Bank asking validated consumer and client give the permission the financial institution may be send the cash in charity.

### **3.3 FEASIBILITY STUDY**

#### **TECHNICAL FEASIBILITY**

##### **HARDWARE REQUIREMENTS**

- PROCESSOR : DUAL CORE 2 DUOS
- RAM : 2 GB DD RAM
- HARD DISK : 250 GB

##### **SOFTWARE REQUIREMENTS**

- FRONT END : J2EE (JSP, SERVLET)
- BACK END : MY SQL 5.5
- OPERATING SYSTEM : WINDOWS 7
- LANGUAGE : JAVA

##### **ECONOMIC FEASIBILITY**

- This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated.
- It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

##### **SOCIAL FEASIBILITY**

- It is helpful for people to save their money from hackers
- It helps the bank to detect the fake cheque instantly

### **3.4 HARDWARE ENVIRONMENT**

- PROCESSOR : DUAL CORE 2 DUOS
- RAM : 2 GB DD RAM
- HARD DISK : 250 GB

### **3.5 SOFTWARE ENVIRONMENT**

- FRONT END : J2EE (JSP, SERVLET)
- BACK END : MY SQL 5.5
- OPERATING SYSTEM : WINDOWS 7
- IDE : ECLIPSE

## **4.SYSTEM DESIGN**

## 4.SYSTEM DESIGN

### 4.1 ER DIAGRAM

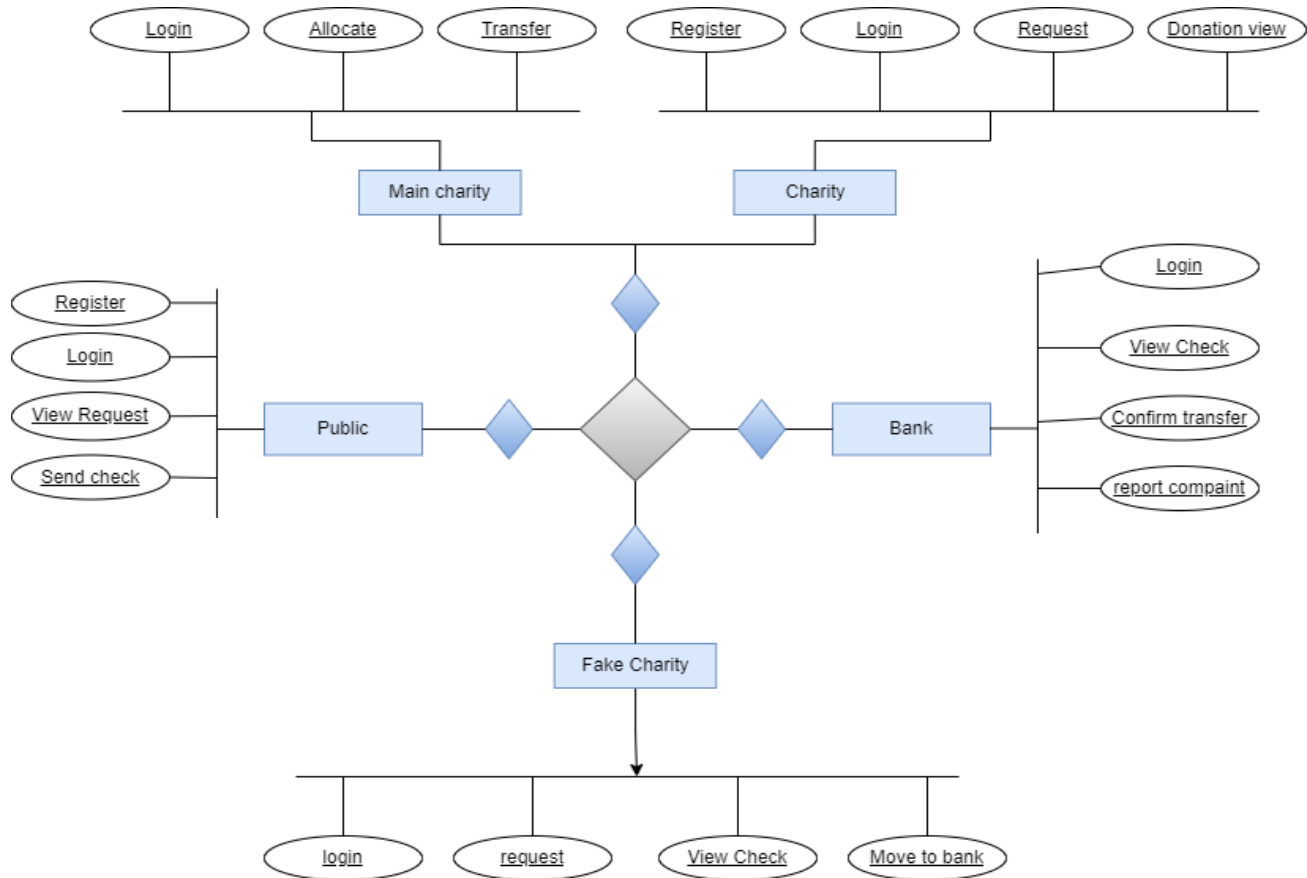


FIG NO. 4. 1 ER DIAGRAM

An entity is represented as rectangle in an ER diagram. For example: In the following ER diagram we have two entities Student and College and these two entities have many to one relationship as many students' studies in a single college. We will read more about relationships later, for now focus on entities.

## 4.2 DATA DICTIONARY

### 4.2.1 CHARITY REGISTRATION





















Column Name	Datatype	NOT NULL	AUTO INC	Flags
 id	 INT(10)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL
 cname	 VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
 cemail	 VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
 contact	 VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
 about	 VARCHAR(500)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
 city	 VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
 state	 VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
 password	 VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
 pic	 VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
 Status	 VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY

TABLE NO. 4.1 CHARITY REGISTRATION

### 4.2.2 PUBLIC REGISTRATION



















Column Name	Datatype	NOT NULL	AUTO INC	Flags
 id	 INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL
 name	 VARCHAR(45)			<input type="checkbox"/> BINARY
 email	 VARCHAR(45)	<input checked="" type="checkbox"/>		<input type="checkbox"/> BINARY
 mobile	 VARCHAR(45)			<input type="checkbox"/> BINARY
 password	 VARCHAR(45)			<input type="checkbox"/> BINARY
 compass	 VARCHAR(45)			<input type="checkbox"/> BINARY
 gender	 VARCHAR(45)			<input type="checkbox"/> BINARY
 pic	 VARCHAR(45)			<input type="checkbox"/> BINARY
 status	 VARCHAR(45)			<input type="checkbox"/> BINARY

TABLE NO .4.2 PUBLIC REGISTRATION

### 4.2.3 CHARITY REQUEST



















Column Name	Datatype	NOT NULL	AUTO INC	Flags
 id	 INT(11)			<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL
 name	 VARCHAR(45)			<input type="checkbox"/> BINARY
 remail	 VARCHAR(45)			<input type="checkbox"/> BINARY
 cemail	 VARCHAR(45)			<input type="checkbox"/> BINARY
 purposee	 VARCHAR(200)			<input type="checkbox"/> BINARY
 contactno	 VARCHAR(45)			<input type="checkbox"/> BINARY
 charity	 VARCHAR(45)			<input type="checkbox"/> BINARY
 status	 VARCHAR(45)			<input type="checkbox"/> BINARY

TABLE NO. 4.3 CHARITY REQUEST

### 4.2.4 CHEQUE DETAILS































Column Name	Datatype	NOT NULL	AUTO INC	Flags
 id	 INT(11)			<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL
 checkno	 VARCHAR(45)			<input type="checkbox"/> BINARY
 checkword	 VARCHAR(300)			<input type="checkbox"/> BINARY
 date	 VARCHAR(45)			<input type="checkbox"/> BINARY
 charityname	 VARCHAR(100)			<input type="checkbox"/> BINARY
 decription	 VARCHAR(100)			<input type="checkbox"/> BINARY
 amount	 VARCHAR(45)			<input type="checkbox"/> BINARY
 Charityemail	 VARCHAR(45)			<input type="checkbox"/> BINARY
 city	 VARCHAR(100)			<input type="checkbox"/> BINARY
 pic	 VARCHAR(45)			<input type="checkbox"/> BINARY
 encrypted	 VARCHAR(45)			<input type="checkbox"/> BINARY
 status	 VARCHAR(45)			<input type="checkbox"/> BINARY
 pemail	 VARCHAR(45)			<input type="checkbox"/> BINARY
 address	 VARCHAR(45)			<input type="checkbox"/> BINARY

TABLE NO .4.4 CHEQUE DETAILS

#### 4.2.5 MONEY TRANSFERED













Column Name	Datatype	NOT NULL	AUTO INC	Flags
 pemail	 VARCHAR(45)			<input type="checkbox"/> BINARY
 cemail	 VARCHAR(45)			<input type="checkbox"/> BINARY
 checkno	 VARCHAR(45)			<input type="checkbox"/> BINARY
 amountinword	 VARCHAR(150)			<input type="checkbox"/> BINARY
 date	 VARCHAR(45)			<input type="checkbox"/> BINARY
 amount	 INT(10)			<input checked="" type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL

TABLE NO.4.5 MONEY TRANSFERED

#### 4.2.6 MONEY TRANSFERRING



















Column Name	Datatype	NOT NULL	AUTO INC	Flags
 id	 INT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/> UNSIGNED <input type="checkbox"/> ZEROFILL
 pemail	 VARCHAR(45)			<input type="checkbox"/> BINARY
 cemail	 VARCHAR(45)			<input type="checkbox"/> BINARY
 checkno	 VARCHAR(45)			<input type="checkbox"/> BINARY
 amountinword	 VARCHAR(45)			<input type="checkbox"/> BINARY
 date	 VARCHAR(45)			<input type="checkbox"/> BINARY
 amount	 VARCHAR(45)			<input type="checkbox"/> BINARY
 phash	 VARCHAR(200)			<input type="checkbox"/> BINARY
 ahash	 VARCHAR(200)			<input type="checkbox"/> BINARY

TABLE NO.4.6 MONEY TRANSFERRING



### 4.3 DATA FLOW DIAGRAM

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system. It differs from the flowchart as it shows the data flow instead of the control flow of the program. A data flow diagram can also be used for the visualization of data processing. The DFD is designed to show how a system is divided into smaller portions and to highlight the flow of data between those parts.

#### LEVEL 0:



FIG NO.4.2 DFD LEVEL 0

#### LEVEL 1:

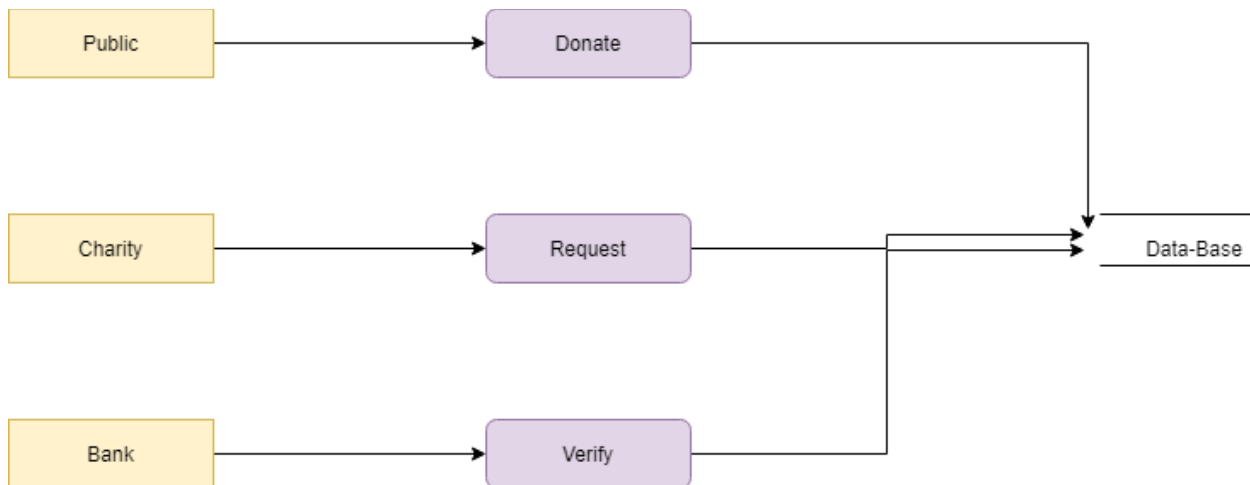


FIG NO .4.3 DFD LEVEL 1

**LEVEL 2:**

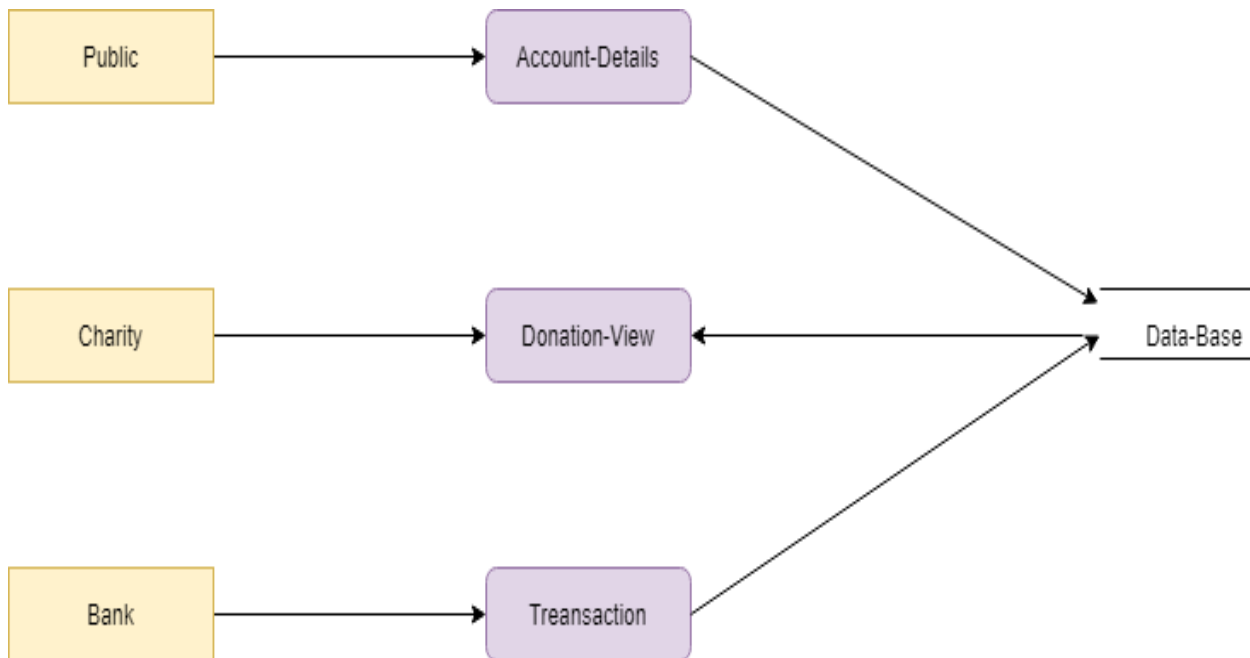


FIG NO.4.4 DFD LEVEL 2

## 4.5 UML DIAGRAM

### 4.5.1 USE-CASE DIGRAM:

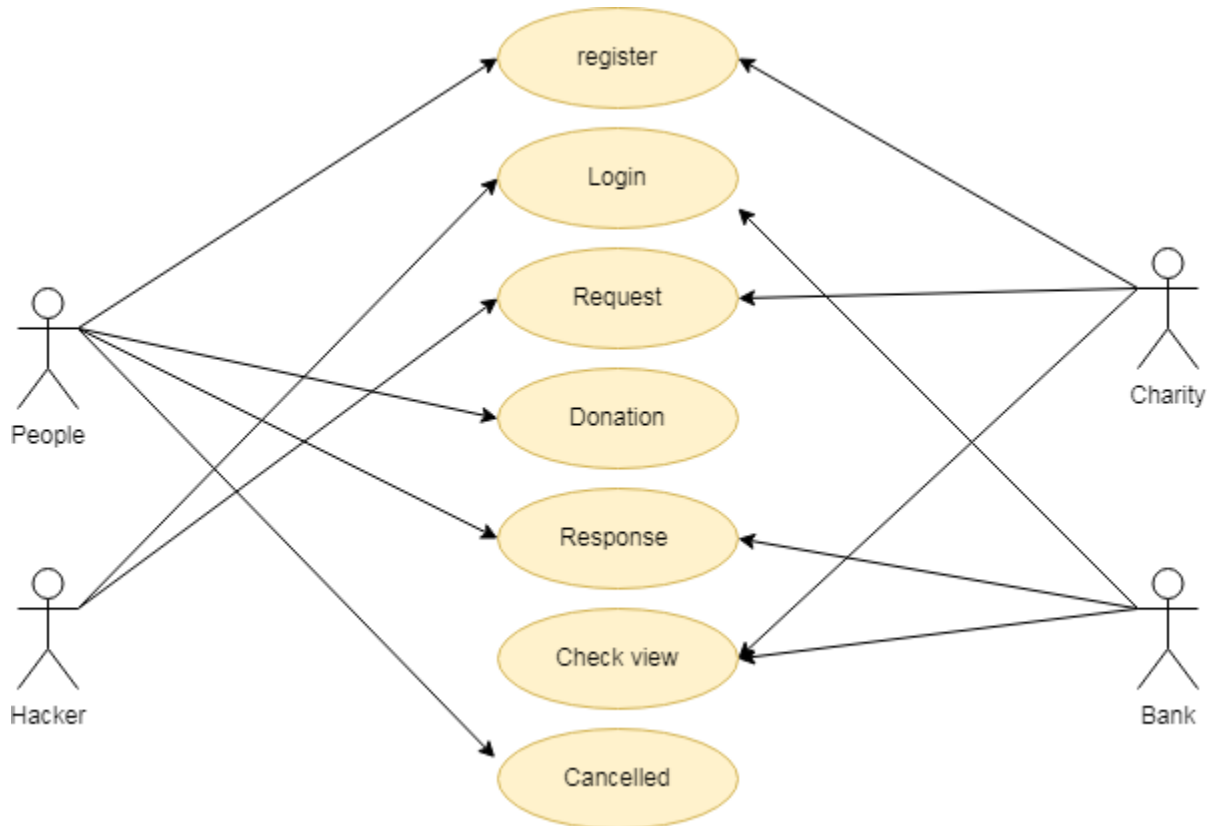


FIG NO.4.5 USE-CASE DIAGRAM

### EXPLANATION:

The use case diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. For this in our component diagram first propose a data. In this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.

#### 4.5.2 STATE DIAGRAM:

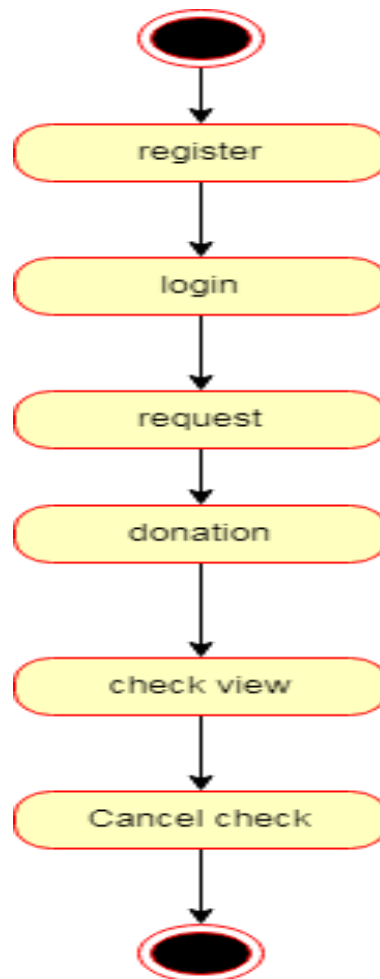


FIG NO.4.6 STATE DIAGRAM

#### EXPLANATION:

State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics. In our state diagram first propose for this in our component diagram first propose a data in this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.

### 4.5.3 ACTIVITY DIAGRAM:

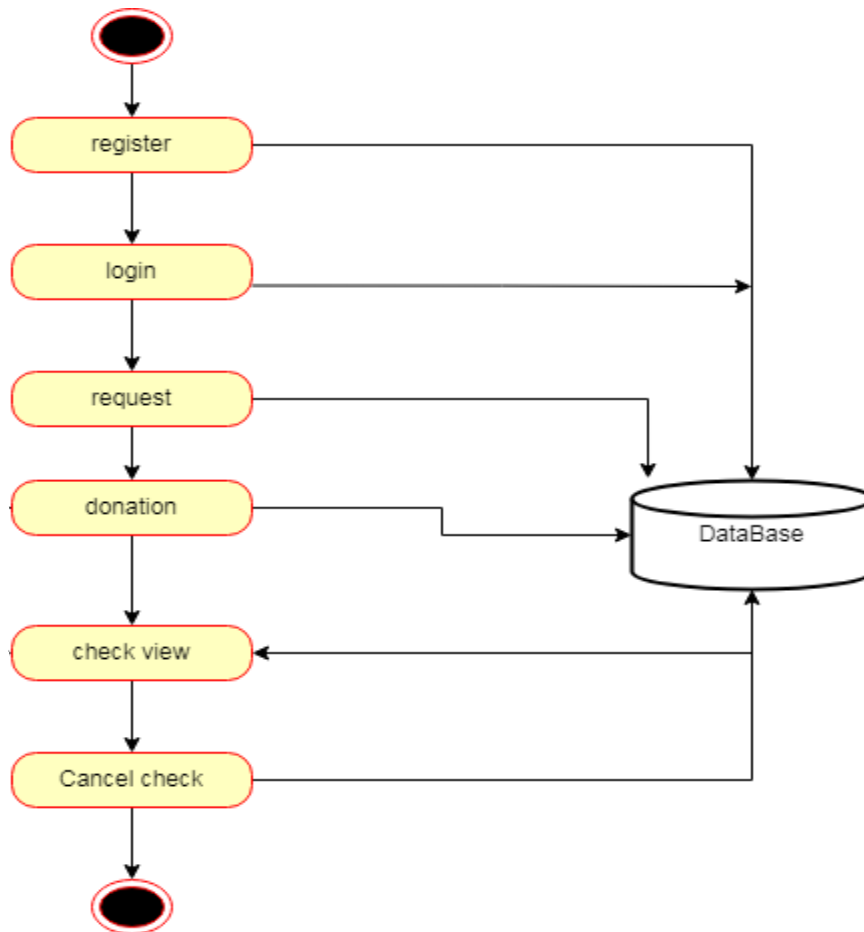


FIG NO.4.7 ACATIVITY DIAGRAM

### EXPLANATION:

Activity diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. UML, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. UML activity diagrams could potentially model the internal logic of a complex operation. In many ways UML activity diagrams are the object-oriented equivalent of flow charts and data flow diagrams(DFDs)from structural development.

#### 4.5.4 CLASS DIGRAM:

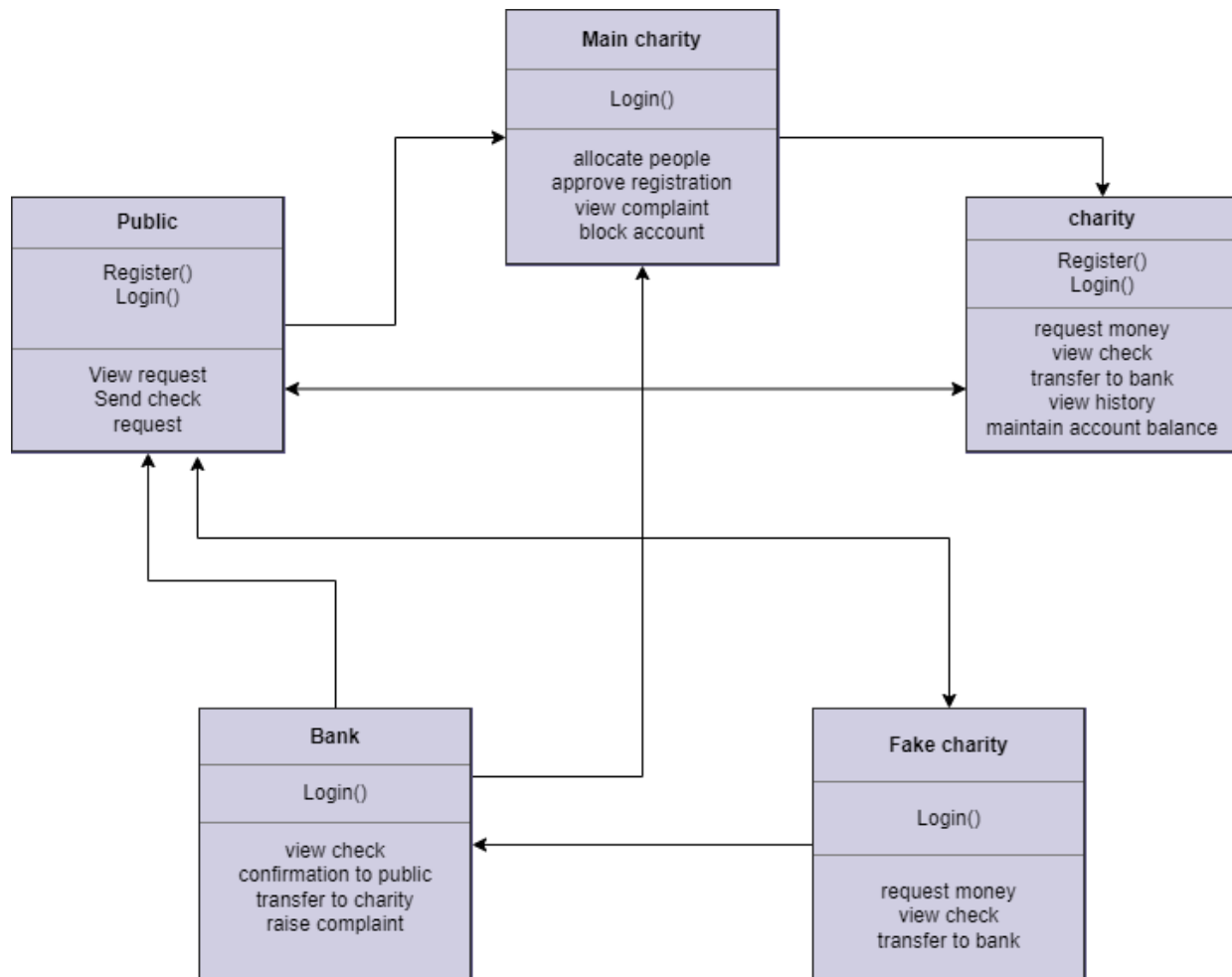


FIG NO.4.8 CLASS DIAGRAM

#### EXPLANATION:

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, and the relationships between the classes. The classes in a class diagram represent both the main objects and or interactions in the application and the objects.

#### 4.5.5 SEQUENCE DIAGRAM:

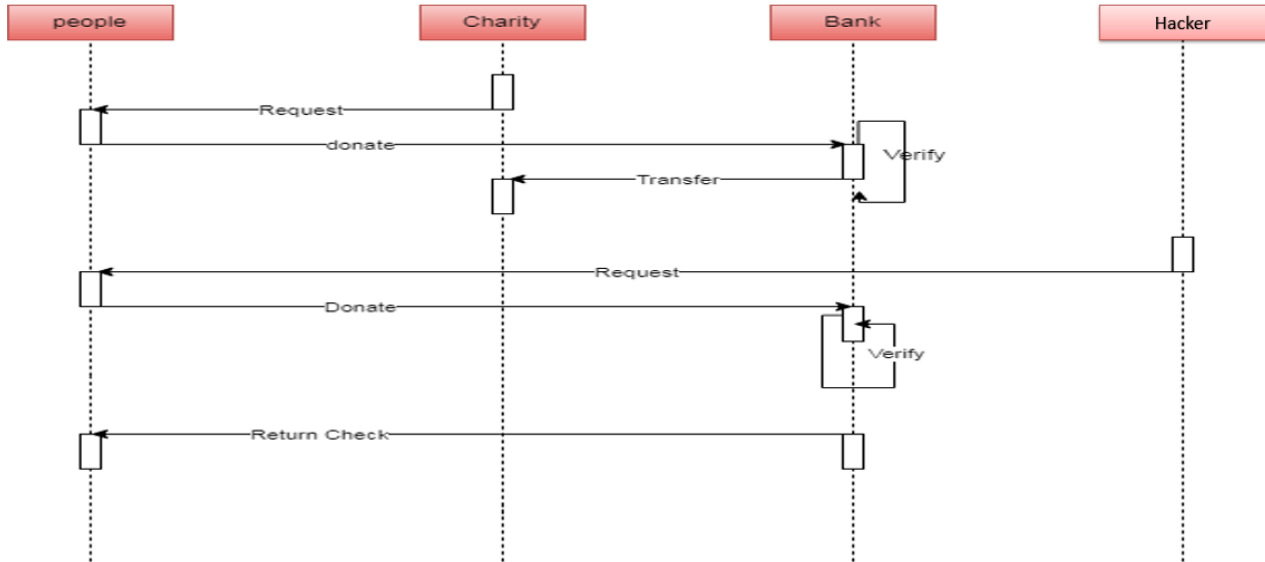


FIG NO.4.9 SEQUENCE DIAGRAM

#### EXPLANATION:

In our sequence diagram specifying processes operate with one another and in order. In our sequence diagram first propose a data. In this proposed method we are using Hash-Solomon Code Algorithm to encrypt the data.

## 4.5.6 COLLABORATION DIAGRAM

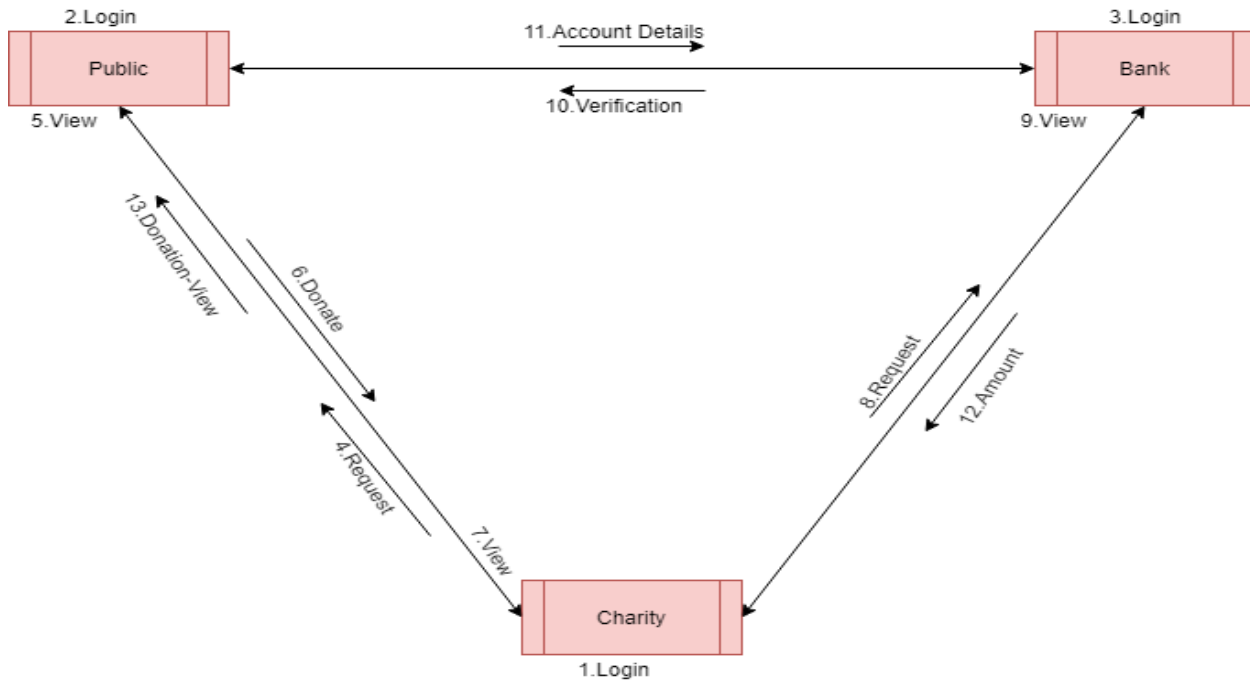


FIG NO.4.10 COLLABORATION DIAGRAM

### EXPLANATION:

A collaboration diagram shows the objects and relationships involved in an interaction, and the sequence of messages exchanged among the objects during the interaction. The collaboration diagram can be a decomposition of a class, class diagram, or part of a class diagram. it can be the decomposition of a use case, use case diagram, or part of a use case diagram. The collaboration diagram shows messages being sent between classes and object(instances). A diagram is created for each system operation that relates to the current development cycle(iteration).



## **5.SYSTEM ARCHITECTURE**

## 5.SYSTEM ARCHITECTURE

### 5.1 SYSTEM ARCHITECTURE DIAGRAM

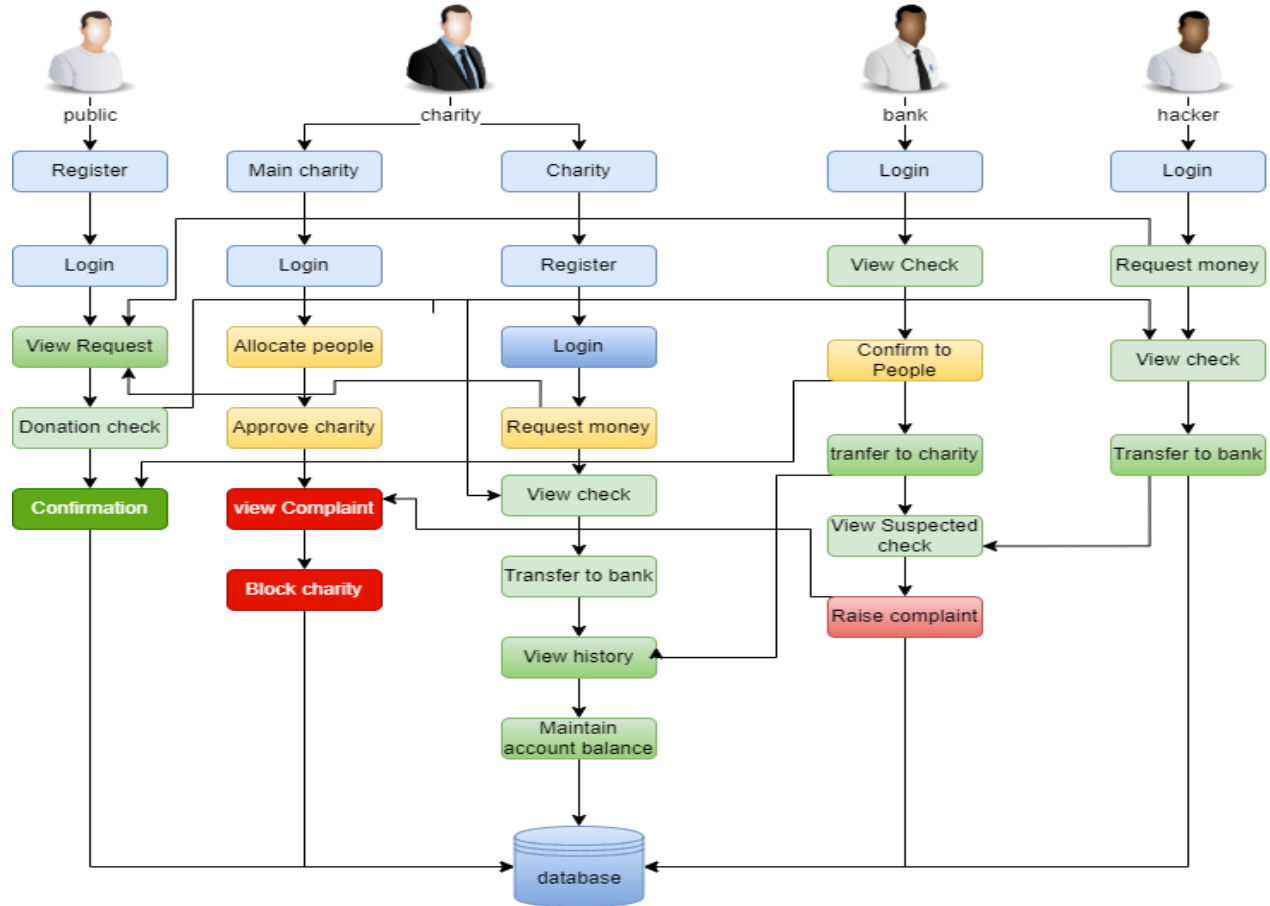


FIG NO 5.1 SYSTEM ARCHITECTURE DIAGRAM

#### EXPLANATION:

The systems architect establishes the basic structure of the system, we propose a Hash code Solomon algorithm and a we can put a small part of data in local machine and fog server in order to protect the privacy. Moreover, based on computational intelligence, this algorithm can compute the distribution proportion stored in cloud, fog, and local machine, respectively. Through the theoretical safety analysis and experimental evaluation, the feasibility of our scheme has been validated, which is really a powerful supplement to existing cloud storage scheme.

## 5.2 MODULE DESIGN SPECIFICATION

### 5.2.1 CHARITY REGISTER

The register module provides a conceptual framework for entering data on those charity in a way that: eases data entry & accuracy by matching the charity entry to the data source (usually paper files created at point of care), ties easily back to individual charity records to connect registers to charity data, and collects data elements to enable better supervision of donation programs.



FIG NO. 5.2 CHARITY REGISTER

### 5.2.2 CHARITY LOGIN

Add a subheading in this module in our project, here symbolizes a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database user will transfer the amount to provider.



FIG NO. 5.3 CHARITY LOGIN

### 5.2.3 CHARITY REQUEST

In this module is used to help to the user to Request for donation with the land longitude and the user will update the report along with their opinion and they will be stored the database.

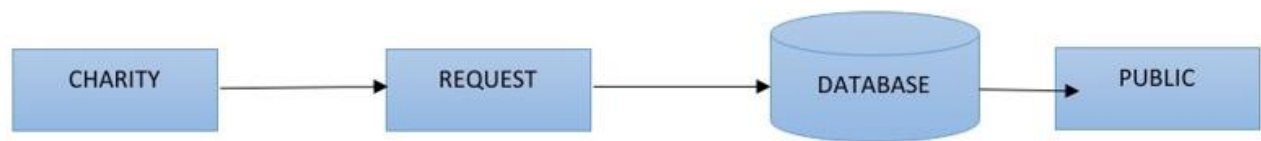


FIG NO.5.4 CHARITY REQUEST

### 5.2.4 CHARITY REQUEST VIEW

In this module the charity will also view the request. And analysis the details will be responsible for your file stored in database.

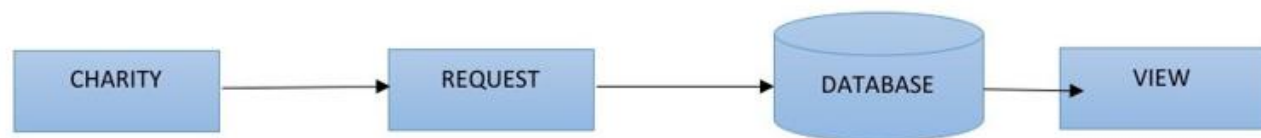


FIG NO.5.5 CHARITY REQUEST VIEW

### 5.2.5 CHARITY DONATION VIEW

In this module the charity will also view the donation. And analysis the details will be responsible for your file stored in database.

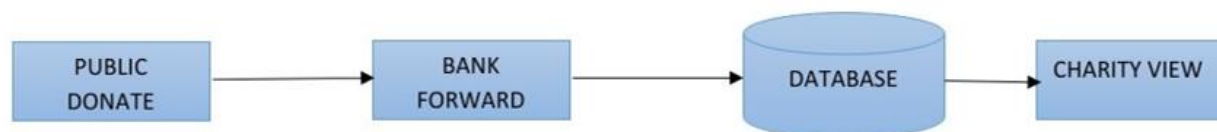


FIG NO.5.6 CHARITY DONATION VIEW

### 5.2.6 BANK RESPONSE VIEW

In this module the charity will also view the donation. And analysis the details will be responsible for your file stored in database.

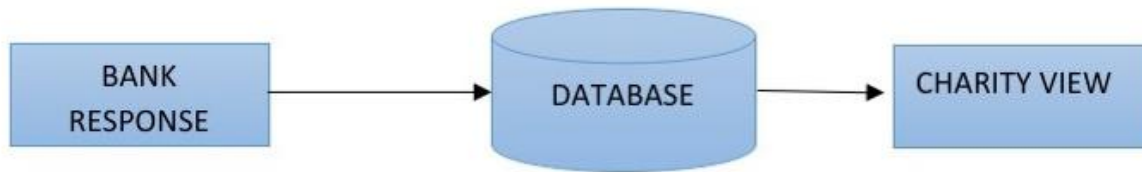


FIG NO. 5.7 BANK RESPONSE VIEW

### 5.2.7 VIEW ACCOUNT BALANCE

In this module the charity will also view the account balance. And analysis the details will be responsible for your file stored in database.

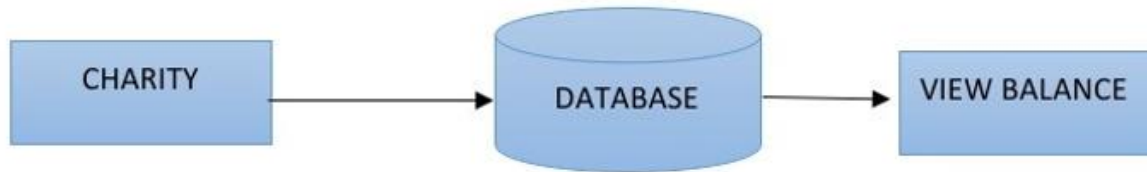


FIG NO. 5.8 VIWE ACCOUNT BALANCE

### 5.2.8 PUBLIC REGISTER

The register module provides a conceptual framework for entering data on those charity in a way that: eases data entry & accuracy by matching the charity entry to the data source (usually paper files created at point of care), ties easily back to individual charity records to connect registers to charity data, and collects data elements to enable better supervision of donation programs.



FIG NO 5.9 PUBLIC REGISTER

### 5.2.9 PUBLIC LOGIN

In this module in our project, here symbolizes a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database user will transfer the amount to provider.



FIG NO. 5.10 PUBLIC LOGIN

### 5.2.10 RESPONSE

In this module the bank will response the data file fully analyzed data in category wise view Bank will be responsible for your file stored in database.

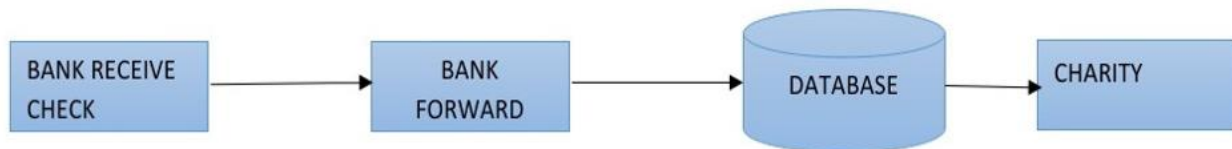


FIG NO. 5.11 RESPONSE

### 5.2.11 ADMIN LOGIN

In this module in our project, here symbolizes a unit of work performed within a database management system (or similar system) against a database, and treated in a coherent and reliable way independent of other transactions. A transaction generally represents any change in database user will transfer the amount to provider.



FIG NO 5.12 ADMIN LOGIN

### 5.2.12 RESPONSE (BANK)

In this module is used to help to the public to donation the amount with the land longitude and the public will update the report along with their opinion and the will be stored the database.



FIG NO 5.13 RESPONSE (BANK)

## 5.3 ALGORITHM

### SHA ALGORITHM

- In the field of cryptography and crypt analytics, the SHA-1 algorithm is a crypt-formatted hash function that is used to take a smaller input and produces a string that is 160 bits, also known as 20-byte hash value long.
- The hash value therefore generated, is known as a message digest which is typically rendered and produced as a hexadecimal number which is specifically 40 digits long.

### CHARACTERISTICS

- The cryptographic hash functions are utilized and used to keep and store the secured form of data by providing three different kinds of characteristics such as pre-image resistance, which is also known as the first level of image resistance, the second level of pre-image resistance and collision resistance.
- The cornerstone lies in the fact that the pre-image crypt resistance technique makes it hard and more time consuming for the hacker or the attacker to find the original intended message by providing the respective hash value.
- The security, therefore, is provided by the nature of a one way that has a function that is mostly the key component of the SHA algorithm. The pre-image resistance is important to clear off brute force attacks from a set of huge and powerful machines.
- Similarly, the second resistance technique is applied where the attacker has to go through a hard time decoding the next error message even when the first level of the message has been decrypted. The last and most difficult to crack is the collision resistance, making it extremely hard for the attacker to find two completely different messages which hash to the same hash value.



- Therefore, the ratio to the number of inputs and the outputs should be similar in fashion to comply with the pigeonhole principle. The collision resistance implies that finding two different sets of inputs that hash to the same hash is extremely difficult and therefore marks its safety.

## **USES OF SHA ALGORITHM**

- These SHA algorithms are widely used in security protocols and applications, including the ones such as TLS, PGP, SSL, IPsec, and S/MiME.
- These also find their place in all the majority of cryptanalytic techniques and coding standards which is mainly aimed to see the functioning and working of majorly all governmental as well as private organizations and institutions.
- Major giants today such as Google, Microsoft, or Mozilla have started to recommend the use of SHA-3 and stop the usage of the SHA-1 algorithm.

## **6.SYSTEM IMPLEMENTATION**

## 6.SYSTEM IMPLEMENTATION

### 6.1 CLIENT-SIDE CODING

#### Web Content:

#### Homepage.jsp:

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Homepage</title>

<style>

ul {

    list-style-type: none;

    margin: 0;

    padding: 0;

    overflow: hidden;

    background: linear-gradient(to right, #0000ff 0%, #00ffff 100%);

}

li {

    float: right;
```

```

padding-right:185px;
}
li a {
display: block;
color: white;
text-align: center;
text-style:bold;
padding: 14px 16px;
text-decoration: none;
}
body {
background-image: url("bgimages/charity1.jpg");
background-repeat: no-repeat;
background-size: cover;
}
</style>
</head>
<body>
<ul>
<li><a href="banklogin.jsp">Bank</a></li>
<li><a href="publiclogin.jsp">Public</a></li>
<li><a href="charitylogin.jsp">Charity</a></li>

```

```
</ul>

</body>

</html>
```

## **Publicregister.jsp;**

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

    <%-- <% @page import="dbconn.Dbconn"%> --%>

    <% @page import="java.sql.ResultSet"%>

    <% @page import="java.sql.PreparedStatement" %>

    <% @page import="java.sql.*" %>

    <% @page import="java.util.*" %>

    <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

    <html>

    <head>

    <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

    <title>Insert title here</title>

    <style>

    .myDiv {

        border: 5px outset #ff0000;

        background-color: lightyellow;
```

```

border-radius: 10px;

width:600px;

height:400px;

margin: auto;

padding-top:30px;

box-shadow: 25px 20px 20px #888888;

}

.myDiv2 {

font-size:25px;

font-style: italic;

font-weight: bold;

color: #ffffff;

}

body{

background-image: linear-gradient(to top, lightyellow,maroon);

}

</style>

</head>

<body>

<br><br>

<center>

<div class="myDiv2">

```

## Public Register

```
</div>

</center>

<br>

<div class="myDiv">

<center>

  <form action="Publicregservlet" method="post">

    <input type="text" name="username" placeholder="Username"
    style="width:280px;height:40px;border-radius: 10px;text-align:center;"><br><br>

    <input type="text" name="phonenum" placeholder="Phonenumber"
    style="width:280px;height:40px;border-radius: 10px;text-align:center;"><br><br>

    <input type="text" name="email" placeholder="Emailid"
    style="width:280px;height:40px;border-radius: 10px;text-align:center;"><br><br>

    <input type="text" name="password" placeholder="Password"
    style="width:280px;height:40px;border-radius: 10px;text-align:center;"><br><br>

    <input type="submit" value="Submit" style="width:100px;height:40px;border-
    radius: 10px;"><br><br>

  </form>

</center>

</div>

</body>

</html>
```

## Publicmainpage.jsp

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

<style>
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: #4bc99b;
}
li {
    float: right;
    padding-right: 185px;
}
li a {
    display: block;

```



```

color: white;

text-align: center;

text-style:bold;

padding: 14px 16px;

text-decoration: none;

}

.myDiv2 {

font-size:25px;

font-style: italic;

font-weight: bold;

color:green;

}

body {

background-image: url("bgimages/charity3.jpg");

background-repeat: no-repeat;

background-size: cover;


/* background: linear-gradient(to bottom, #996633 0%, #ff66cc 100%); */

/* background-image: linear-gradient(to right, red , yellow); */

}

</style>

</head>

```

```

<body>

<ul>

    <li><a href="homepage.jsp">Logout</a></li>

    <li><a href="publicresptobank.jsp">Response(Bank)</a></li>

    <li><a href="publicrequview.jsp">Request(Charity)</a></li>

    <li><a href="homepage.jsp">Home</a></li>

</ul>

<br><br>

<center>

<div class="myDiv2">

Public Mainpage

</div></center>

</body>

</html>

```

### **Charitylogin.jsp:**

```

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
    pageEncoding="ISO-8859-1"%>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Insert title here</title>

```

```
<style>

.myDiv {

    border: 5px outset red;

    background-color: lightgreen;

    border-radius: 10px;

    width:600px;

    height:200px;

    margin: auto;

    padding-top:30px;

    /* box-shadow: 25px 20px 20px #888888; */

}

.myDiv2 {

    font-size:25px;

    font-style: italic;

    font-weight: bold;

    color:white;

}

body{

    /* background-image: linear-gradient(to bottom, #00ccff 0%, #33cc33 100%); */

    background-image: linear-gradient(to top, lightblue, blue);

}
```

```

</style>

</head>

<body>

<br><br><br>

<center>

<div class="myDiv2">

Charity Login Page

</div>

</center>

<br><br>

<form action="Charityloginservlet" method="post">

<div class="myDiv">

<center>

    <input type="text" name="username" placeholder="Username"
    style="width:280px;height:40px;border-radius: 10px;text-align:center;"><br><br>

    <input type="text" name="password" placeholder="Password"
    style="width:280px;height:40px;border-radius: 10px;text-align:center;"><br><br>

    <input type="submit" value="Submit" style="width:100px;height:40px;border-
    radius: 10px;background-color:green;"><br><br>

</center>

</div>

</form>

</body>

```

</html>

## **Charitymainpage.jsp:**

```
<% @ page language="java" contentType="text/html; charset=ISO-8859-1"
```

```
    pageEncoding="ISO-8859-1"%>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

```
<title>Insert title here</title>
```

```
<style>
```

```
ul {
```

```
    list-style-type: none;
```

```
    margin: 0;
```

```
    padding: 0;
```

```
    overflow: hidden;
```

```
    background-color: #cccc00;
```

```
}
```

```
li {
```

```
    float: right;
```

```
    padding-right: 85px;
```

```
}  
  
li a {  
    display: block;  
    color: white;  
    text-align: center;  
    text-style:bold;  
    padding: 14px 16px;  
    text-decoration: none;  
}  
  
.myDiv2 {  
    font-size:25px;  
    font-style: italic;  
    font-weight: bold;  
    color:green;  
}  
  
body {  
    background-image: url("bgimages/charity4.jpg");  
    background-repeat: no-repeat;  
    background-size: cover;  
  
    /* background: linear-gradient(to bottom, #996633 0%, #ff66cc 100%); */  
  
    /* background-image: linear-gradient(to right, red , yellow); */
```

```

}

</style>

</head>

<body>

<ul>

    <li><a href="homepage.jsp">Logout</a></li>

    <li><a href="charityaccount.jsp">Account</a></li>

    <li><a href="bankapprovtocharity.jsp">Response(Bank)</a></li>

    <li><a href="charitydonationview.jsp">Donation</a></li>

    <li><a href="charitypubreqview.jsp">RequestView</a></li>

    <li><a href="charityrequest.jsp">Request</a></li>

    <li><a href="homepage.jsp">Home</a></li>

</ul>

<br><br>

<center>

<div class="myDiv2">

Charity Mainpage

</div></center>

</body>

</html>

Charityrequest.jsp;

<% @ page language="java" contentType="text/html; charset=ISO-8859-1"

```

```
pageEncoding="ISO-8859-1"%>
```

```
<% @page import="dbconn.Dbconn"%>
```

```
<% @page import="java.sql.ResultSet"%>
```

```
<% @page import="java.sql.PreparedStatement" %>
```

```
<% @page import="java.sql.*" %>
```

```
<% @page import="java.util.*" %>
```

```
<% @page import="javax.servlet.http.HttpSession" %>
```

```
<% @page import="imple.Imple" %>
```

```
<% @page import="inter.Inter" %>
```

```
<% @page import="bean.Charityreqbean" %>
```

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
```

```
<script type="text/javascript"  
src="https://cdn.zingchart.com/zingchart.min.js"></script>
```

```
<title>Insert title here</title>
```

```
<style>
```

```
.myDiv2 {
```

```
font-size:25px;
```



```
    font-style: italic;
font-weight: bold;
color:red;
}
table,td,th {
    border: 2px solid black;

}
table {
    border-collapse: collapse;
    width: 90%;
}
td {
    text-align: center;
    height: 40px;
}
th{
    height: 30px;
    color: red;
}
.backtag{
float:right;
```

```

margin-right:20px;

font-size:20px;

}

</style>

</head>

<body>

<div class="backtag">

<a href="charitymainpage.jsp">Back</a>

</div>

<br>

<center>

<div class="myDiv2">

Public Details

</div>

<%

String uname=session.getAttribute("emailkey").toString();

%>

<br><br><br>

<table>

<th>Name</th>

<th>Phone No.</th>

<th>Email</th>

```

<th>Remarks</th>

<%

String idlist="";

String namelist="";

String phnolist="";

String emailist="";

String unnameemail="";

%>

<%

Connection d = Dbconn.create();

PreparedStatement p = d.prepareStatement("SELECT \* FROM  
`chequebc01`.`publicregtbl`");

//PreparedStatement p = d.prepareStatement("SELECT \* FROM  
`admission`.`adminaccepttbl`");

ResultSet rp = p.executeQuery();

while (rp.next()){

idlist=rp.getString(1);

namelist=rp.getString(2);

phnolist=rp.getString(3);

emailist=rp.getString(4);

%>

```

        <tr>

        <td><%=namelist%></td>

        <td><%=phnolist%></td>

        <td><%=emaillist%></td>


        <%-- <td class="text-right"><a href="delete?id=<%=id%>"><button class="btn
btn-sm btn-danger"><I class="fa fa-trash"></i> </button> </a></td> --%>

        <%-- <td><a href='Deleteservletnew?id=<%=idlist%>' class="btn btn-
danger">Remove</a></td> --%>

        <%-- <td class="text-right"><a href="delete?id=<%=id%>"><button class="btn
btn-sm btn-danger"><I class="fa fa-trash"></i> </button> </a></td> --%>

        <%-- <td>

        </td> --%>

        <%-- <td><a href="Deleteservletnew?id=<%=idlist%>"><button class="btn btn-
danger">Remove</button></a></td> --%>

        <td><a href="charityreqtopublic.jsp?unaml=<%=emaillist%>"><button
class="btn btn-danger">Request</button></a></td>

    </tr>

    <%

    }

    %>

</table>

</center>

```

</body>

</html>

## 6.2 SERVER-SIDE CODING

**Java Database code;**

**Dbcon;**

```
package dbconn;
```

```
import java.sql.Connection;
```

```
import java.sql.DriverManager;
```

```
public class Dbconn {
```

```
    static Connection con;
```

```
    public static Connection create(){
```

```
        try{
```

```
            Class.forName("com.mysql.jdbc.Driver");
```

```
            con=DriverManager.getConnection("jdbc:mysql://localhost:3306/chequebc01",  
            "root","root");
```

```
        }catch(Exception ex)
```

```
        {
```

```
            ex.printStackTrace();
```

```
        }
```

```

        return con;
    }
}

```

### **Charitylog.java:**

```

package servlet;

import imple.imple;
import inter.inter;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import bean.Charitybean;

/**
 * Servlet implementation class Charitylog
 */
@WebServlet("/Charitylog")
public class Charitylog extends HttpServlet {

```

```
private static final long serialVersionUID = 1L;
```

```
/**
```

```
 * @see HttpServlet#HttpServlet()
```

```
 */
```

```
public Charitylog() {
```

```
    super();
```

```
    // TODO Auto-generated constructor stub
```

```
}
```

```
/**
```

```
 * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse  
response)
```

```
 */
```

```
protected void doGet(HttpServletRequest request, HttpServletResponse  
response) throws ServletException, IOException {
```

```
    // TODO Auto-generated method stub
```

```
}
```

```
/**
```

```
 * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse  
response)
```

```
 */
```

```
protected void doPost(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
```

```
    // TODO Auto-generated method stub
```

```
    String cemail=request.getParameter("username");
```

```
    System.out.println("Username :"+ceemail);
```

```
    HttpSession sess=request.getSession();
```

```
    sess.setAttribute( "ceemail",ceemail);
```

```
    System.out.println("email: "+ceemail);
```

```
    String pass=request.getParameter("password");
```

```
    System.out.println("Password :"+pass);
```

```
    Charitybean cb=new Charitybean();
```

```
    cb.setCemail(ceemail);
```

```
    cb.setPassword(pass);
```

```
    inter in=new imple();
```

```
    57boolean log=in.clog(cb);
```

```
    if(log==true)
```

```
    {
```

```
        response.sendRedirect("charitymainpage.jsp");
```

```
    }
```

```
    else
```

```
    {
```



```

        response.sendRedirect("error.jsp");
    }
}
}

```

### **Imple.java**

```

package imple;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import dbcon.Dbconn;
import bean.Charitybean;
import bean.charity;
import bean.charityrequest;
import bean.fakecheque;
import bean.transfer;
import bean.publicreg;
import inter.inter;

public class imple implements inter{

```

```

    Connection con;

```

@Override

```
public int creg(Charitybean cb) {
```

```
    int reg=0;
```

```
    con=Dbconn.create();
```

```
        try {
```

```
            PreparedStatement
```

```
ps=con.prepareStatement("INSERT INTO `scam`.`charityreg`  
VALUES(id,?,?,?,?,?,?,?,?)");
```

```
            ps.setString(1,cb.getCname());
```

```
            ps.setString(2,cb.getCemail());
```

```
            ps.setString(3,cb.getContact());
```

```
            ps.setString(4,cb.getAbout());
```

```
            ps.setString(5,cb.getCity());
```

```
            ps.setString(6,cb.getState());
```

```
            ps.setString(7,cb.getPassword());
```

```
            ps.setString(8,cb.getPicture());
```

```
            ps.setString(9,"Verification");
```

```
            reg=ps.executeUpdate();
```

```

    } catch (SQLException e) {

        // TODO Auto-generated catch block

        e.printStackTrace();

    }

    return reg;

}

@Override

public int preg(publicreg pb) {

    int reg=0;

    con=Dbconn.create();

    try {

        PreparedStatement
ps=con.prepareStatement("INSERT INTO `scam`.`pubreg`
VALUES(id,?,?,?,?,?,?,?)");

        ps.setString(1,pb.getPname());

        ps.setString(2,pb.getPemail());

        ps.setString(3,pb.getMobilenumber());

        ps.setString(4,pb.getPassword());

        ps.setString(5,pb.getConpass());

```

```
ps.setString(6,pb.getGender());
```

```
ps.setString(7,pb.getPic());
```

```
ps.setString(8,"Register");
```

```
reg=ps.executeUpdate();
```

```
    } catch (SQLException e) {
```

```
        // TODO Auto-generated catch block
```

```
        e.printStackTrace();
```

```
    }
```

```
    return reg;
```

```
}
```

```
@Override
```

```
public boolean plog(publicreg pl) {
```

```
    boolean log=false;
```

```
    con=Dbconn.create();
```

```
    try {
```

```
        PreparedStatement ps=con.prepareStatement("SELECT *  
FROM `scam`.`pubreg` where email=? And password=?");
```

```

        ps.setString(1,pl.getPemail());

        ps.setString(2,pl.getPassword());

        ResultSet rs=ps.executeQuery();

        log=rs.next();

    } catch (SQLException e) {

        // TODO Auto-generated catch block

        e.printStackTrace();

    }

    return log;

}

@Override

public int creq(charityrequest cr) {

    int reg=0;


    con=Dbconn.create();


    try {


        PreparedStatement
ps=con.prepareStatement("INSERT INTO `scam`.`Charityreq`
VALUES(id,?,?,?,?,?,?)");


        ps.setString(1,cr.getRname());

```

```
ps.setString(2,cr.getRemail());  
ps.setString(3,cr.getCemail());  
ps.setString(4,cr.getPurpose());  
ps.setString(5,cr.getContact());  
ps.setString(6,cr.getCharityname());  
ps.setString(7, "Request");
```

```
reg=ps.executeUpdate();
```

```
    } catch (SQLException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
    }
```

```
    return reg;
```

```
}
```

```
@Override
```

```
public int ches(charity ck) {  
    int reg=0;  
    con=Dbconn.create();  
    try {
```

### PreparedStatement

```
ps=con.prepareStatement("INSERT INTO `scam`.`chequedetails`  
VALUES(id,?,?,?,?,?,?,?,?,?,?,?)");
```

```
ps.setString(1,ck.getChequeno());
```

```
ps.setString(2,ck.getChequeword());
```

```
ps.setString(3,ck.getDate());
```

```
ps.setString(4,ck.getCharityname());
```

```
ps.setString(5,ck.getDescription());
```

```
ps.setString(6,ck.getAmount());
```

```
ps.setString(7,ck.getCemail());
```

```
ps.setString(8,ck.getCity());
```

```
ps.setString(9,ck.getPic());
```

```
ps.setString(10,ck.getEncryptedchecno());
```

```
ps.setString(11,"Donation");
```

```
ps.setString(12,ck.getPemail());
```

```
ps.setString(13, ck.getAddress());
```

```
reg=ps.executeUpdate();
```

```
} catch (SQLException e) {
```

```
    // TODO Auto-generated catch block
```

```
    e.printStackTrace();
```

```
}
```

```

        return reg;
    }

    @Override

    public boolean clog(Charitybean cl) {

        // TODO Auto-generated method stub

        boolean log=false;

        con=Dbconn.create();

        try {

            PreparedStatement ps=con.prepareStatement("SELECT *
FROM `scam`.`charityreg` where cemail=? And password=? And status='Verified'
");

            ps.setString(1, cl.getCemail());

            ps.setString(2, cl.getPassword());

            ResultSet rs=ps.executeQuery();

            log=rs.next();

        } catch (SQLException e) {

            // TODO Auto-generated catch block

            e.printStackTrace();

        }

        return log;
    }

```



@Override

public int trans(transfer tr) {

    // TODO Auto-generated method stub

    int reg=0;

    con=Dbconn.create();

        try {

                PreparedStatement

ps=con.prepareStatement("INSERT INTO `scam`.`transmoney`  
VALUES(id,?,?,?,?,?,?,?)");

                ps.setString(1, tr.getPemail());

                ps.setString(2, tr.getCemail());

                ps.setString(3, tr.getChequeno());

                ps.setString(4, tr.getAmountinword());

                ps.setString(5, tr.getDate());

                ps.setString(6, tr.getAmount());

                ps.setString(7, tr.getPhash());

                ps.setString(8, tr.getAhash());

                reg=ps.executeUpdate();

    } catch (SQLException e) {

```

        // TODO Auto-generated catch block

        e.printStackTrace();

    }

    return reg;

}

@Override

public boolean clog1(Charitybean fc) {

    // TODO Auto-generated method stub

    boolean log=false;

    con=Dbconn.create();

    try {

        PreparedStatement ps=con.prepareStatement("SELECT *
FROM `scam`.`charityreg` where cemail=?");

        ps.setString(1,fc.getCemail());

        ResultSet rs=ps.executeQuery();

        log=rs.next();

    } catch (SQLException e) {

        // TODO Auto-generated catch block

        e.printStackTrace();

    }

```

```
        return log;
    }
}
```

### **Publicloginservlet.java**

```
package servlet;

import imple.imple;
import inter.inter;
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import bean.publicreg;

/**
 * Servlet implementation class Publicloginservlet
 */
@WebServlet("/Publicloginservlet")
public class Publicloginservlet extends HttpServlet {

    private static final long serialVersionUID = 1L;
```

```

    /**
     * @see HttpServlet#HttpServlet()
     */
    public Publicloginservlet() {
        super();
        // TODO Auto-generated constructor stub
    }

    /**
     * @see HttpServlet#doGet(HttpServletRequest request, HttpServletResponse
    response)
     */
    protected void doGet(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }

    /**
     * @see HttpServlet#doPost(HttpServletRequest request, HttpServletResponse
    response)
     */
    protected void doPost(HttpServletRequest request, HttpServletResponse
    response) throws ServletException, IOException {
        // TODO Auto-generated method stub
    }

```

```

String name=request.getParameter("username");

System.out.println("username:"+name);

String pass=request.getParameter("password");

System.out.println("password:"+pass);

HttpSession sess=request.getSession();

sess.setAttribute( "pemail",name);

System.out.println("email: "+name);

publicreg ur=new publicreg();

ur.setPemail(name);

ur.setPassword(pass);

inter in=new imple();

70oolean log=in.plog(ur);

if(log==true){

    response.sendRedirect("publicmainpage.jsp");

}

else{

    response.sendRedirect("error.jsp");

}

}

}

```

## **7.SYSTEM TESTING**

## 7.SYSTEM TESTING

### 7.1 TEST CASE & REPORT

#### CHARITY MANGEMENT

S.NO	ACTION TO PERFORM	EXCEPECTED RESULTS	ACTUAL RESULTS	RESULT(PASS/FAIL)
1	Login	Fill the user name and password	As Excepted	Pass
2	Charity approved	Verify the details of charity	As Excepted	Pass
3	Allocate public	Verify the details of public	As Excepted	Pass

TABLE NO. 7.1 CHARITY MANAGEMENT

#### CHARITY

S.NO	ACTION TO PERFORM	EXCEPECTED RESULTS	ACTUAL RESULTS	RESULT(PASS/FAIL)
1	Register	Fill the given details	As Excepted	Pass
2	Login	Fill the user name and password	As Excepted	Pass
3	Charity request	Request for donation	As Excepted	Pass
4	View cheque	View the cheque and transfer to the bank	As Excepted	Pass

TABLE NO. 7.2 CHARITY

## PUBLIC

S.NO	ACTION TO PERFORM	EXCEPECTED RESULTS	ACTUAL RESULTS	RESULT(PASS/FAIL)
1	Register	Fill the given details	As Excepted	Pass
2	Login	Fill the user name and password	As Excepted	Pass
3	View the request	Public views the request by the charity	As Excepted	Pass
4	Willing to donate	Public make the donation	As Excepted	Pass
5	Response from bank	Verification from the bank	As Excepted	Pass

TABLE NO.7.3 PUBLIC



## BANK

S.NO	ACTION TO PERFORM	EXCEPECTED RESULTS	ACTUAL RESULTS	RESULT(PASS/FAIL)
1	Login	Fill the user name and password	As Excepted	Pass
2	View and verify the cheque	Cheque will be view by bank and verified from public	As Excepted	Pass
3	Verified	Amount will be transferred to charity	As Excepted	Pass
4	Not verified	Cheque will be encrypted	As Excepted	Pass

TABLE NO.7.4 BANK

## **8. CONCLUSION**

## **8. CONCLUSION**

### **8.1 CONCLUSION:**

Banking scams involve attempts to access your bank account. Use this information to recognize, report, and protect yourself from them. These scams work because fake cheques generally look just like real cheques, even to bank employees. They are often printed with the names and addresses of legitimate financial institutions. They may even be real cheques written on bank accounts that belong to identity theft victims. It can take for a bank to figure out that the cheque is a fake.

### **8.2 FUTURE ENHANCEMENTS:**

1. Implementing a real-world database system.
2. Improving the efficiency of protocols, in terms of number of messages exchanged and in terms of their sizes, as well.
3. Implement using two or more algorithms.

## APPENDICES

### A.1 SAMPLE SCREENS

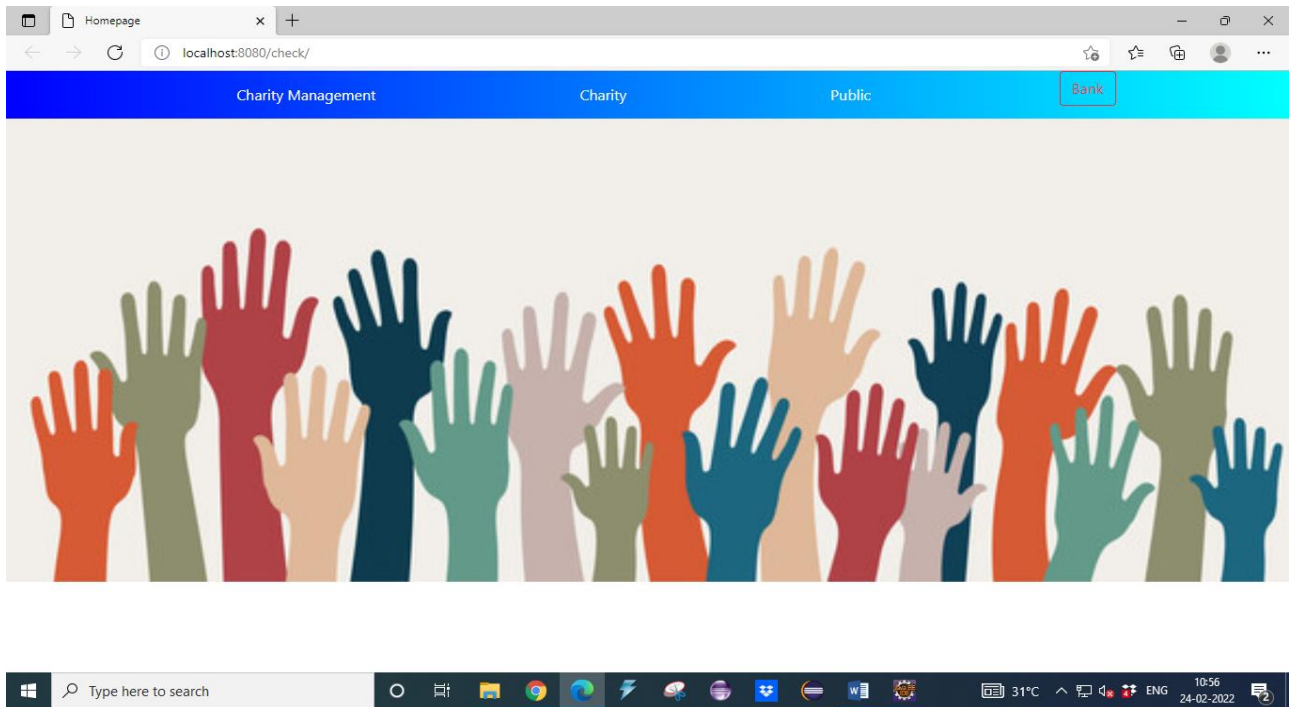


FIG NO. A.1 MAIN PAGE

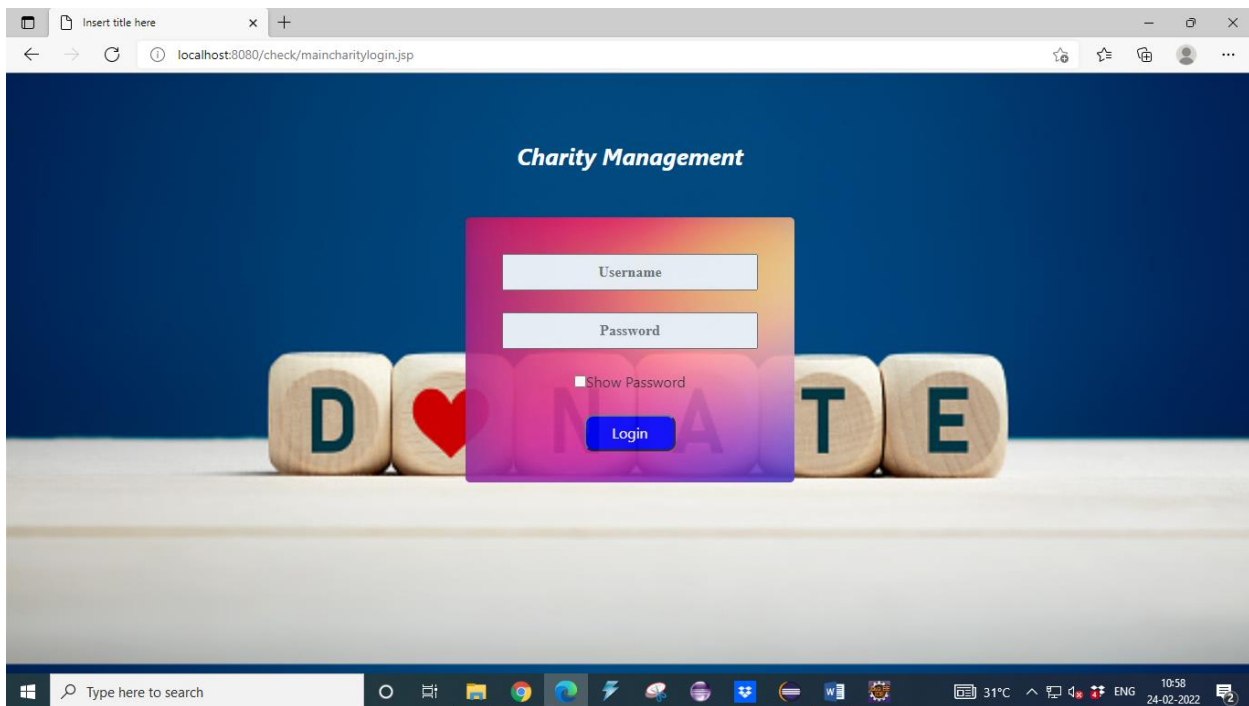


FIG NO. A.2 CHARITY MANAGEMENT LOGIN

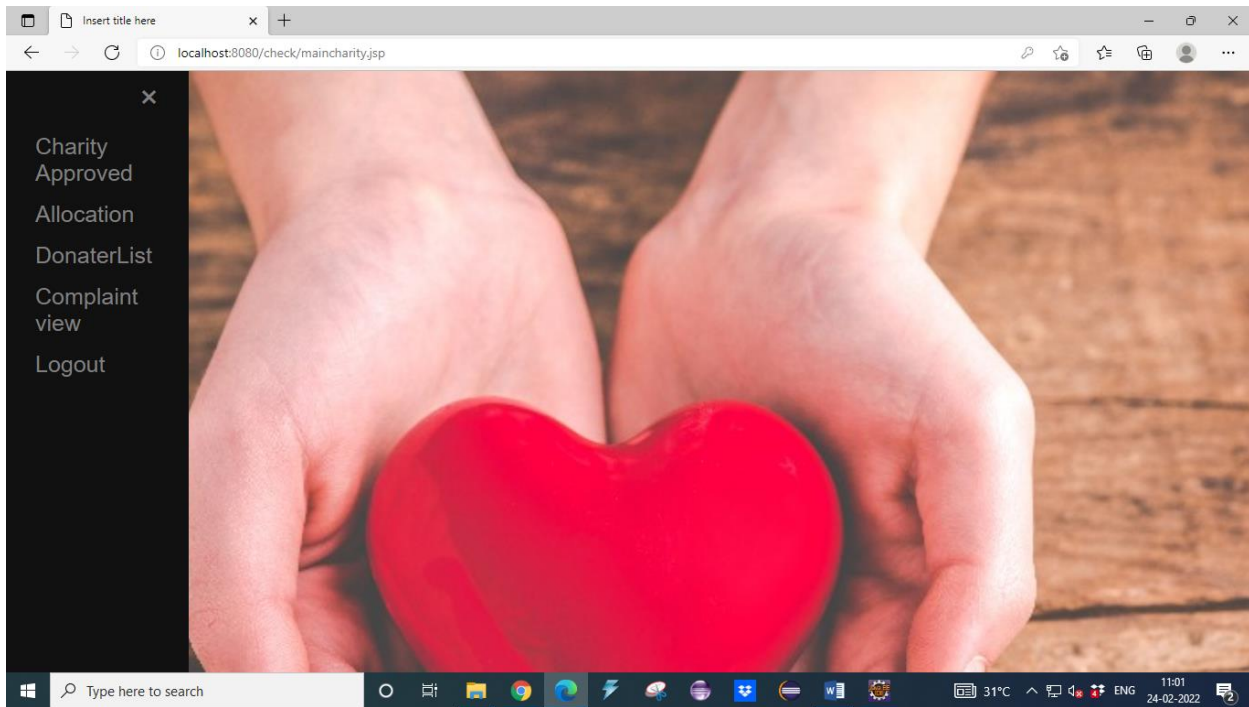


FIG NO. A.3 CHARITY MANAGEMENT MAIN PAGE

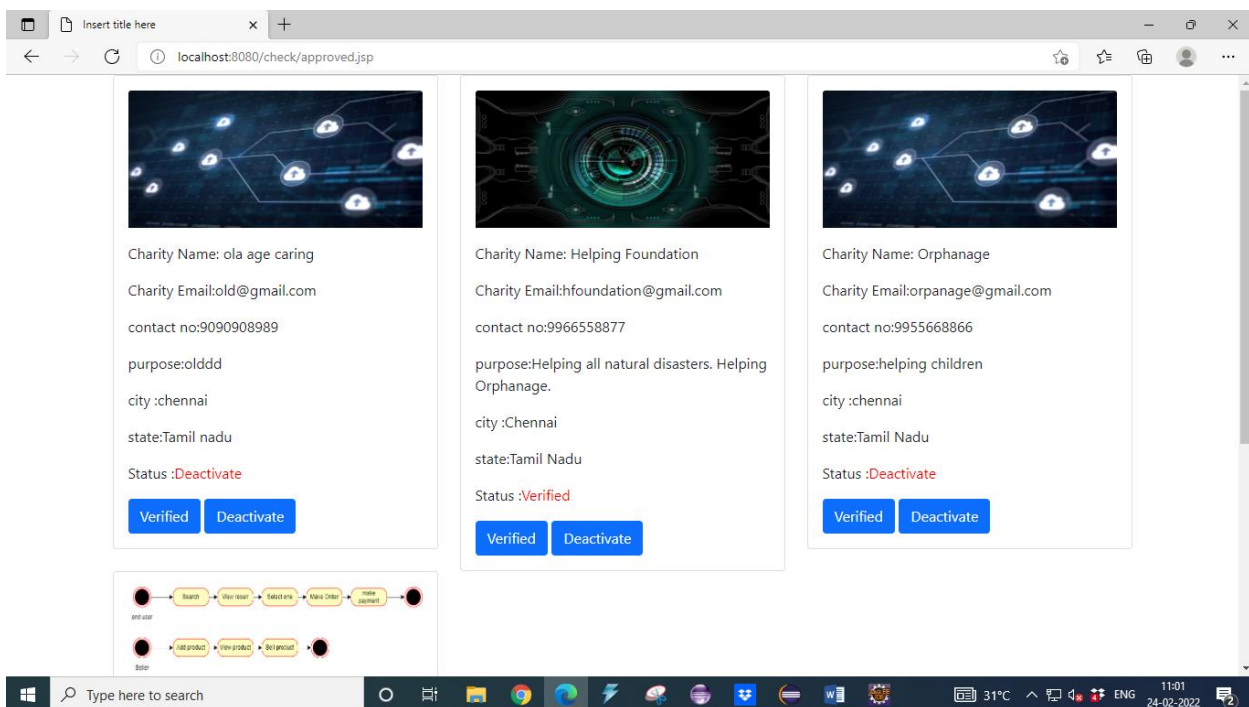


FIG NO. A.4 CHARITY APPROVE PAGE

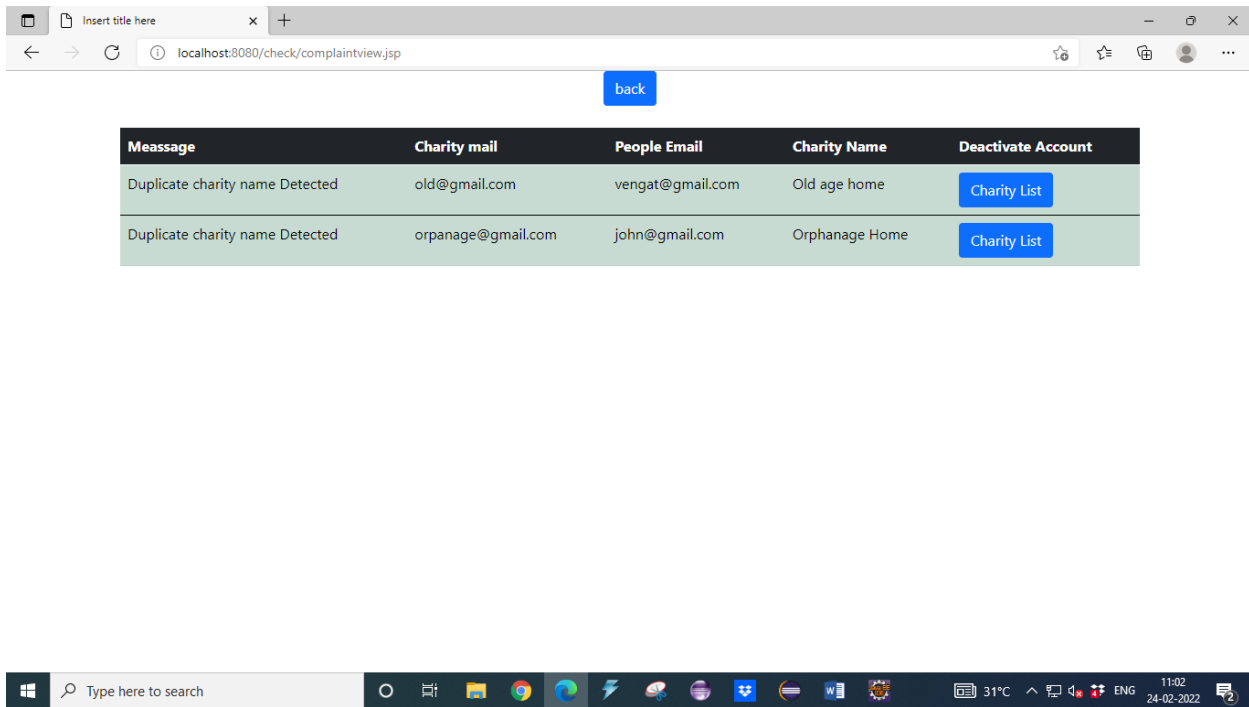


FIG NO. A.5 CHARITY COMPLAINT VIEW

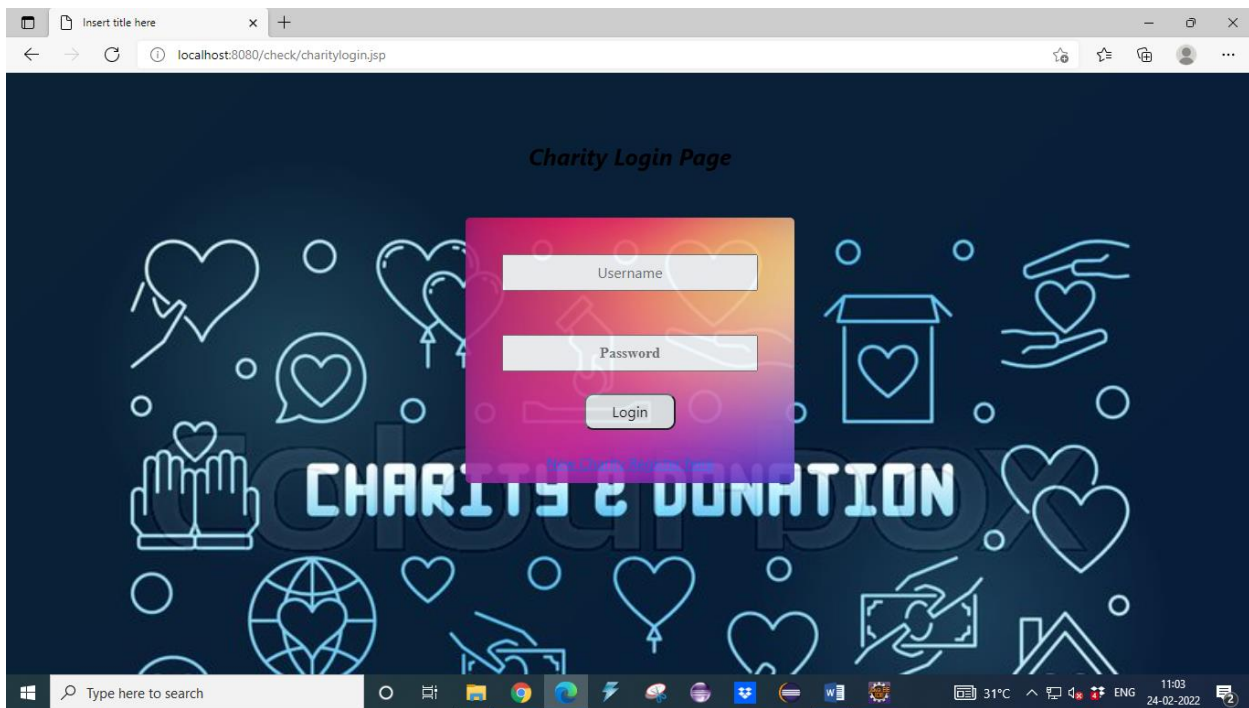


FIG NO. A.6 CHARITY LOGIN PAGE

Charity Register Details

Charity Name:

Enter Charity Name

Charity Email:

Charity Email

Contact Number

Contact Number

ABOUT INFO

City

City

State

State

password

Enter Password

Charity Upload photo:

Choose File

No file chosen

Your selected File:

FIG NO. A.7 CHARITY REGISTER PAGE:



FIG NO. A.8 CHARITY MAIN PAGE

CharityBank
Personal Account

hulk@gmail.com  
ID: 2014112

ACCOUNT DETAILS	BALANCE ( in Indian Rupees )
ACCOUNT HOLDER hulk@gmail.com ACCOUNT NO 211-XXX-3546-XXX	₹ Rs.10000.0

FIG NO. A.9 CHARITY ACCOUNT VIEW

Public Login

Username

Password

☐ Show Password

Login

New User [Signup](#)

FIG NO. A.10 PUBLIC LOGIN PAGE



**Register Details**

Name:

Email:

Contact Number

Password

Confirm Password

Gender ☐ male ☐ female

Charity Upload photo:

Your selected File:

FIG NO. A.11 REGISTER PAGE

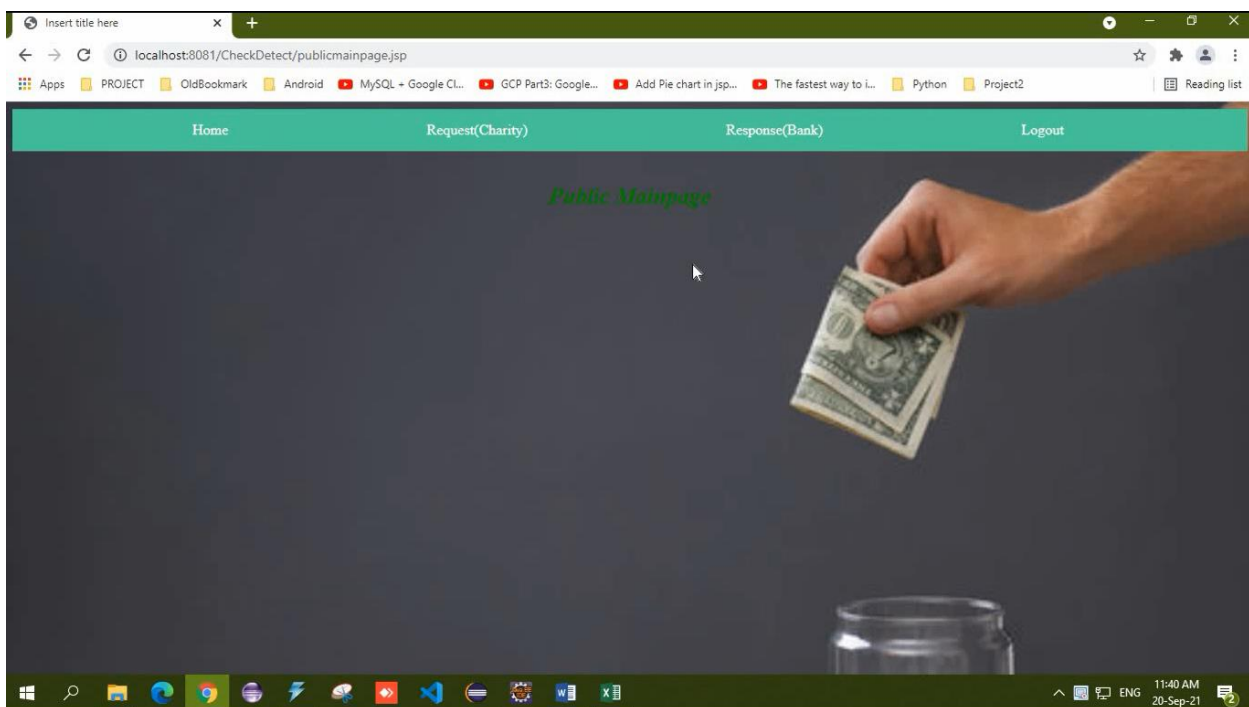


FIG NO. A.12 PUBLIC MAIN PAGE

<div>back</div>					
Name	Charity Name	charity Email	Purpose	Charity Contact Number	Donation
Batman	Hulk foundation	hulk@gmail.com	need some fund fore kaja cyclone affected people	9988774455	<div>Donation</div>

FIG NO. A.13 PUBLIC REQUEST VIEW

THIS CHECK IS IN PAYMENT OF THE FOLLOWING		CHECK NO: 724204196
PAY	FIVE HUNRED	RUPPEES
DATE 03-01-2022	TO THE ORDER OF ORPHANAGE HOME	DESCRIPTION NEED TO HELP
		CHECK AMOUNT 500
TAMIL NADU		
Bank Name, N.A.		
SUBMIT		
SIGNATURE		

FIG NO. A.14 CHEQUE VIEW PAGE

## REFERENCES

- [1] Badis Hammi, Sherali Zeadally, Yves Christian Elloh Adja, Manlio Del Giudice, and Jamel Nebhen. (2021) “Blockchain-Based Solution for Detecting and preventing Fake Cheque Scams”.
- [2] Emart 77, Beirut, Lebanon. (2021) “The Role of Blockchain in Reducing the Cost of Financial Transactions in the Retail Industry”.
- [3] Dilip Kumar Sharma, Sonal Garg, Priya Shrivastava. (2021) “Evaluation of Tools and Extension for Fake News Detection”.
- [4] Vikash Kumar Aggarwal, Nikhil Sharma, Ila Kaushik, Bharat Bhushan, Himanshu. (2020) “Integration of Blockchain and IoT (B-IoT): Architecture, Solution, & Future Research Direction”.
- [5] Jackie Jones, Damon McCopy. (2020) “The Cheque is in the Mail: Monetization of Craigslist Buyer Scams”.
- [6] Abiola, Idowu. (2019) “An Assessment of Fraud and its Management in Nigeria Commercial Banks”.
- [7] Sumeet Kumar. (2018) “Simulating DDOS attacks on the us fiber-optics internet infrastructure”.
- [8] Nazli Ismail Nawang . (2017) “Combating anonymous offenders in the cyberspace: An overview of the legal approach in Malaysia”.
- [9] Haris Semic ,Sasa Mrdovic. (2017) “IoT honeypot: A multi-component solution for handling manual and Mirai-based attacks”.
- [10] Bernie S. Fabito, Angelique D. Lacasandile ,Emeliza R. Yabut .(2017) “Leveraging crime reporting in Metro Manila using unsupervised crowd-sourced data: A case for the Report framework”.
- [11] Wendy Baker-Smemoe, Chair David Eddington William G. Eggington. (2015) “The Language and Cross-Culture Perceptions of Deception”.

- [12] S.Baker .(2018) “Don’t cash that cheque: BBB study shows how fake cheque scams bait consumers”,Tech Rep., Better Bus Bureau ,Arlington Country,VA,USA.
- [13] L. M. Rose. (2018) “Modernizing cheque fraud detection with machine learning”,Phd.D. dissertation, Dept. Financial Crime Compliance Manage., Utica College ,Utica, NY, USA.
- [14] Federal Trade Commission.(2018),“Consumer Sentinel network data book 2017”, Federal Trade Commission ,Washington, DC, USA, Tech. Rep.
- [15] C. Tressler.(2020), “ FTC: The bottom-line on fake cheques scams”, Federal Trade Commission ,Washington , DC ,USA , Tech. Rep.
- [16] “2017 Internet crime report ”.(2018) Federal Bureau of Investigation/Internet Crime Complaint Center, Washington ,DC ,USA, Tech. Rep.
- [17] K. Pak and D. Shadel. (2011)“AARP Foundation national fraud victim study ”, AARP Foundation , Washington , DC, USA, Tech.Rep.
- [18] C.-D. Chen and L.-T.Huang. (2011) “Online deception investigation: Content analysis and cross-culture comparision”, Int. J. Bus. Inf., vol. 6, no.1, pp.91-111.
- [19] K. Christidis and M . Devetsikiotis. (2016) “Blockchains and smart contracts for the Internet of things” , IEEE Access, vol.4 , pp.2292-2303.
- [20] A.Reyna , C.Martin, J. Chen ,E. Soler ,and M. Diaz .(2018) “On blockchain and its integration with Iot: Challenges and opportunities”, Future Gener . Comput .Syst., vol.88 ,pp.173-190.
- [21] M.T. Hammi, B. Hammi, P. Bellot, and A. Serhrouchni.(2018) “Bubbles of trust: A decentralized blockchain-based authentication system for IoT”, Comput. Secure., vol. 78, pp. 126 - 142.
- [22] R. M. Factora. (2014)“ Financial and legal methods to protect individuals from financial exploitation in Aging and Money” , Newyork, NY , USA: Springer, pp .109 - 122.

- [23] C. W. Smith.(2001) “Defence to a payor bank’s liability for kate returns”, CCH Deposit Law Notes , vol. 2, no.6, p. 8.
- [24] A. T. Riggs and P. M. Podrazik. (2014) “Financial exploitation of the elderly: Review of the epidemic - Its victims, national impact and legislative solutions”, in Aging and Money. New York, NY, USA: Springer, 2014, pp.1 – 18.
- [25] J. Jones and D. McCoy.(2014)“The cheque is in the mail:Monetization of Craigslist buyer scams”, in Proc. APWGSymp. Electron. Crime Res., 2014, pp. 25 – 35 .
- [26] I.Abiola. (2009) “An assessment of fraud and its management in Nigeria commercial banks”, Eur. J. Social Sci., vol.10, no.4, pp.628 – 640.
- [27] J. A. Ojo, 2008, “Effects of bank frauds on banking operations in Nigeria”, Int. J. Investment Finance, vol. 1, no. 1, p, 103.
- [28]S. Chhabra, G. Gupta, M. Gupta, and G.Gupta.(2017) “Detecting fraudulent bank Cheques”, in Proc. IFIP Int. Conf. Digit. Forensics, pp.245 – 266.
- [29] R. Kumar and G. Gupta. (2016)“ Forensic authentication of bank cheques”, in Proc. IFIP Int. Conf. Digit. Forensics, pp. 311 – 322 .