

**DECENTRALIZED FUNDING PLATFORM BASED ON BLOCKCHAIN TECHNOLOGY**

##### A PROJECT REPORT

###### ***Submitted by***

##### BENETA GETZIE M [REGISTER NO:211417104034]

**DEEPIKA T [REGISTER NO:211417104048]**

**HARINI B [REGISTER NO:211417104077]**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

IN

# COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

##### AUGUST 2021

**BONAFIDE CERTIFICATE**

##### Certified that this project report “DECENTRALIZED FUNDING PLATFORM BASED ON BLOCKCHAIN TECHNOLOGY” is the bonafide work of “BENETA GETZIE M [REGISTER NO:211417104034], DEEPIKA T [REGISTER NO:211417104048], HARINI B [REGISTERNO:211417104077]” who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr.S.MURUGAVALLI,M.E.,Ph.D., ANITHA MOSES V, ME**

**HEAD OF THE DEPARTMENT SUPERVISOR**

**ASSOCIATE PROFESSOR**

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE,

NASARATHPETTAI, NASARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on 05/08/2021.

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We 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 would like to extend our heartfelt and sincere thanks to our

Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHIKUMAR,M.E.,Ph.D.,** and **Tmt.SARANYASREESAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of 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.V.ANITHA MOSES**, **ME**. and

all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

BENETA GETZIE M

DEEPIKA T

HARINI B

**ABSTRACT**

Crowdfunding is a method of raising funds from a large number of funders to start a new business or for charitable purposes using the internet. An important factor for the people involved in raising these funds is trust and the temporary funds of the recipient is stored in the fundraising organization, so to attract funders to donate their funds to the recipient, trust is the important capital for the fundraising organization. In the existing method of online crowdfunding, the contributor does not have any control over the money that they have contributed. Since in the existing method the fundraising organization has all the control over the money contributed, they can very easily perform malicious activities. The problem faced by this existing system can be addressed by using blockchain concept. Blockchain in crowdfunding allows decentralization which suggests that nobody within the network as control over the blocks which makes it transparent to everyone within the blockchain. In this proposed method all the activities performed in a crowdfunding campaign are managed by using the blockchain concept. Each transaction is recorded in the blockchain network to ensure that the amount is received by the valid recipient.

TABLE OF CONTENTS

| **CHAPTERNO.** | **TITLE** | **PAGE NO.** |
| --- | --- | --- |
|  | **ABSTRACT** | ii |
|  | **LIST OF TABLES** | V |
|  | **LIST OF FIGURES** | Vi |
| **1.** | **INTRODUCTION** | 1 |
|  | 1.1 Overview | 1 |
|  | 1.2 Problem Definition | 1 |
| **2.** | **LITERATURE SURVEY** | 2 |
| **3.** | **SYSTEM ANALYSIS** | 8 |
|  | 3.1 Existing System | 8 |
|  | 3.2 Proposed system | 8 |
|  | 3.3 Requirement Analysis and Specification | 10 |
|  | 3.3.1 Input Requirements | 10 |
|  | 3.3.2 Output Requirements | 10 |
|  | 3.3.3 Functional Requirements | 10 |
|  | 3.4 Hardware Environment | 11 |
|  | 3.5 Software Environment | 11 |
| **4.** | **SYSTEM DESIGN** | 16 |
|  | 4.1 Data dictionary | 16 |
|  | 4.2 Data Flow Diagram | 17 |
|  | 4.3 UML Diagrams | 19 |
|  |  |  |
|  |  |  |
| **5.** | **SYSTEM ARCHITECTURE** | 25 |
|  | 5.1 Architecture Overview | 25 |
|  | 5.2 Module Design Specification | 26 |
|  |  |  |
| **6.** | **SYSTEM IMPLEMENTATION** | 29 |
|  | 6.1 Client-side coding | 29 |
|  | 6.2 Server-side coding | 32 |
| **7.** | **SYSTEM TESTING** | 39 |
|  | 7.1 Unit Testing | 40 |
|  | 7.2 Integration Testing | 45 |
|  | 7.3 Test Cases & Reports / Performance Analysis | 46 |
| **8.** | **CONCLUSION** | 50 |
|  | 8.1 Conclusion and Future Enhancements | 50 |
|  | **APPENDICES** | 51 |
|  | A.1 Sample Screens | 51 |
|  | A.2 Publications | 57 |
|  | **REFERENCES** | 58 |

**LIST OF TABLES**

4.1.1 Registration DB for crowdfunding 16

4.1.2 Fundraiser DB for crowdfunding 17

7.3.1 Test case report for Fundraiser for registration 47

7.3.2 Test case report for Funder registration 48

7.3.3 Test case report for Post creation 49

**LIST OF FIGURES**

3.5 An overview of the software development process 12

4.2.1 Data Flow Diagram for Crowdfunding 18

4.2.2 Data Flow Diagram for Crowdfunding 18

4.3.1 Use case Diagram for Crowdfunding 20

4.3.2 Class Diagram for Crowdfunding 21

4.3.3 Activity Diagram for Crowdfunding 22

4.3.4 Swimlane Diagram for Crowdfunding 23

4.3.5 Sequence Diagram for Crowdfunding 24

5.1 Architecture Diagram for Crowdfunding 25

7.1.1 Fundraiser register page to validate mobile number 43

7.1.2 Fundraiser register page to validate account number 43

7.1.3 Fundraiser transaction page to validate account number 44

7.1.4 Invalid page 44

A.1.1 Home page 51

A.1.2 Fundraiser register and login page 51

A.1.3 Fundraiser register page 52

A.1.4 Fundraiser login page 52

A.1.5 Charity home page 53

A.1.6 Fund raise page 53

A.1.7 Donor home page 54

A.1.8 Fund information page 54

A.1.9 Payment page 55

A.1.10 Fund transaction page 55

A.1.11 Transaction history page 56

A.1.12 Admin page 56

1. **INTRODUCTION**
   1. **OVERVIEW**

Crowdfunding is a method of raising funds from a large number of funders to start a new business or for charitable purposes using the internet. In crowdfunding we need to handle lot of transaction and document it accordingly. So here the blockchain concept is used to document these transactions and all the transacted amount are stored in blocks that form a blockchain, so that there is no need for any third party. By using the blockchain concept each transaction is recorded so that funder and fundraiser can keep track of their transaction.

* 1. **PROBLEM DEFINITIOIN**

In present day crowdfunding platform, the money donated by the funder is temporarily stored in fundraising platform so the funder will have no control over the money that they have donated. The funder will have no idea if the money that he has donated has reached the fundraiser or not. The third-party i.e., the fundraising agency will have complete control over the money and hence they can perform malicious activities. The third-party involvement may also cause additional cost. As this platform is centralized by someone there is a chance to use the money the funders have donated in an illegal way.

1. **LITERATURE SURVEY**

**Nikhil Yadav and Sarasvathi V** presented “Venturing Crowdfunding using Smart Contracts in Blockchain” in Third International Conference on Smart Systems and Inventive Technology (ICSSIT 2020). As there are a lot of transactions involved in crowdfunding the actions should be documented legally. A smart contract which is a transaction protocol is used to automatically control, execute and document the actions automatically according to the agreement created by the project developer. This paper includes two contracts. One contract is used to store the transactions and another contract is used to handle the transactions. There are 3 stages in crowdfunding platform. Project creation, 2. Sending request, 3. Voting system. The disadvantages are it can be only used for funding projects. It is based on voting system maximum votes is needed for a project to be approved.[2]

**FirmansyahAshari, TetukoCatonsukmoro, WilyuMahendra Bad, Sfenranto, Gunawan Wangpresented** presented“Smart Contract and Blockchain for Crowdfunding Platform” in International Journal of Advanced Trends in Computer Science and Engineering June 2020. The dominant schemes in this crowdfunding platform are all or nothing – if the goal is not reached, the funds will be returned to the funders. Keep-it-all – Same as all or nothing but the funds are returned based on the policies of fundraising. Stretched goal scheme – The fundraising goal is expanding and are bound by a statement that can add more values to services and products. The disadvantages are it follows all or nothing scheme so the project is funded only when the goal is reached.[6]

**Ms. S. Benila, V. Ajay, K. Hrishikesh, R. Karthick** presented “Crowdfunding using Blockchain” in Global Research and Development Journal for Engineering. In this smart contract is written in solitary language. The campaign factory is used to create new campaigns. A gas fees is specified when a campaign is deployed. The campaign created contains the idea, minimum contribution and detailed description of project. The modules of the project are campaign creation, create request model, request approval model and finalize modules. The disadvantages are during campaign creation minimum fees is specified. The request form must be approved.[3]

**Hasnan Baber** presented “Blockchain-Based Crowdfunding: A ‘Pay-it-Forward’ Model of WHIRL” in International Journal of Recent Technology and Engineering (IJRTE). Whirl is a crowdfunding platform developed on pay it forward theory. The owner starts the campaign. The backers is used to back the project by providing cryptocurrencies and in turn they receive karma points. The karma points is used to start their own campaign. The disadvantages are whirl-app want you to first provide help to others in the community, then the Whirl community will help you in turn.[4]

**Vikas Hassija, Vinay Chamola, and SheraliZeadally** presented “BitFund :A Blockchain-based Crowd Funding Platform for Future Smart and Connected Nation” in Sustainable Cities and Society**.** In this a smart contract is established between the investors and the developer. Investors add new project on the decentralized network. A new block contains the timestamp, expected cost, project details, time and developer reputation. The block is delivered to all the nodes in the developer's network, and they start bidding their values for time, cost, support period and votes to win the project. The auctioning algorithm is deployed by the smart contract and it searches for the best developer for the project. The block is then verified and approved. The block is added in the blockchain and linked with the previously existing blocks. The disadvantages are the developer need votes to win the project.[1]

**Er.WaheedaDhokley , Saurabh Gupta , Ganesh Pawar , Abrar Shaikh** presented “Crowdsourcing and Crowdfunding Platform using Blockchain and Collective Intelligence” in International Journal of Computer Sciences and Engineering**.** Users can create a project or back up an project. It follows all or nothing, i.e. the funders should get all his funds back if the funding is not completed within specific time. The creator posts the project and the funder donates. In crowdsourcing the creator will post tasks. A rule is set by the creator to find qualified workers. Workers are selected on the basis of reputation points. The workers should submit their solutions for the project within the specified time and the workers will receive reward points in return. Collective Intelligence and Collaborative Filtering are used to find new campaign and qualified workers. The disadvantages are in crowdfunding it follows all or nothing scheme so the project is funded only when the goal is reached within the specific time. In crowdsourcing the workers will earn reputation points only when he submits a solution.[5]

**André Schweizer, Vincent Schlatt, Nils Urbach, Gilbert Fridgen** presented “Unchaining Social Businesses – Blockchain as the Basic Technology of a Crowdlending Platform”in International Conference on Information Systems (ICIS), Seoul, South Korea, December 2017. Crowdfunding allows investors to fund various projects and is usually based on the Internet. In crowdlending, The investors provide loans to the seekers for their projects instead of providing donation or equity. The variables of our fundraising auction smart contract were fundingGoal, minimumFunding, currentFundingAmount, currentFundraisingStatus. These describes the current funding status and the funding amount. The typeOfProject describes the student project type. startOfPaybackPeriodInMinutes ,paybackCycleTimeInMinutes stores the detail about the payback conditions. If the payback sum is paid then no more actions is taken by the smart contract and the status of the smart contract is set as closed. The smart contract contains the history and related transactions, these transactions will reside immutably in the blockchain. The disadvantages are the investors can reclaim their amounts if the minimum funding is not reached within the specified time. The students can withdraw the invested amount only if the fundraising is successful.[7]

**Zhu, Huasheng; Zhou, Zach Zhizhong** presented “Analysis and outlook of applications of blockchain technology to equity crowdfunding in China” in Financial Innovation, Springer, Heidelberg, 2016. Equity crowdfunding is a type of crowdfunding in which the fundraiser launches a project with shares of the business as reward. Funders invest in the project in return for equity. This paper examines the problems of equity crowdfunding in china and how to overcome it using blockchain technology. In registration and confirmation of shareholders the company paper are stored in centralized storage. There is no security. The blockchain provides alternate solution which is decentralized storage and it provides security. For the cost of registration blockchain provides digitalized solution and human cost is eliminated. The issues in the equity transaction is that we need a third party system for equity transactions. Blockchain achieves point-to-point i.e. direct money transfers between the users. The equity owned by the investors are provided as proof on the blockchain technology. Blockchain traceability features helps to track all the funds that are transferred and prevents anti-money laundering. The disadvantages arethe problems are underdeveloped credit rating system and lack of qualification management of the investor.[8]

**Zhao Hongjiang, Cephas P.K Coffie** presented “The Applications of Blockchain Technology in Crowdfunding Contract” Article *in* SSRN Electronic Journal · January 2018. This paper discusses about how the blockchain technology handle the relationship between the fundraiser and the funder. The blockchain helps the platform to record the description of the details provided by the fundraiser and it is visible to the funder. The institutions serves as a mediator. It is very important to provide trust to the investor and serves secure communication. With blockchain technology it provides trust and keeps the transaction secure. The investors can donate in the platform. Blockchain provides security and protection. It helps in peer to peer communication. Reward based and voting based systems can also be used. Blockchain based crowdfunding is used when there are issues related to security, trust, payment, voting, fund management and communication. The disadvantages arewhile using reward based, first we have to provide help in turn the community will help us.While using voting system we need to obtain maximum votes then only our project will be approved.[9]

**HissuHyvarinen, Marten Risius, Gustav Friis** presented“A Blockchain-Based Approach Towards Overcoming Financial Fraud in Public Sector Services” Business and Information systems Engineering, November 2017.This paper focuses on overcoming the taxation issues faced by the government related to double spending. When one person applies for the same tax refund without being deducted or if the digital currency is spend twice then it is known as double spending problem. The main reason for double spending is that the activities are not monitored. Blockchain traces the payment activities and prevents double spending problem. Blockchain provides data privacy and helps in managing funds. The main advantage of blockchain is the presence of immutable blocks. It is very easy for the tax authorities to track the funds from the start point to the end point and it helps to prevent the fraudulence. The disadvantages areit depends on the trust of the company that offers the tokens and depends on the policies of the bank.It requires a greater number of service providers and customers in order to establish a closed system.[10]

**Claudia Gabriela Bîzderea** presented“ANALYSIS OF FUNDING ALTERNATIVES THROUGH FINANCIAL TECHNOLOGY SERVICES – FINTECH – THE PHENOMENON OFCROWDFUNDING” in Review of Financial Studies November 2017.This paper monitors the financial systems. Fintech(“Financial Technology”) introduces a new way in which people can fund, i.e. send money, investments and borrowing. Fintech is an emerging business that uses technology to automate process in finance. It changes the global finance. Fintech provides more transparency, more efficiency and cost reduction. The financing crowdfunding also known as participatory multifinancing. In multifinacing there are several actors involved. The fundraiser proposes the project that needs amount. The funders who financially support the project. The platform acts as an intermediate between fundraiser and funder. Multifinancing establishes stronger relationship between fundraisers and funders, in which the funder later will become voluntary lawyers of the project. The disadvantages are if the project is launched without generating interest from the public could cause loss. At least a preliminary period of 3-4 months is recommended before launching a new project.[11]

**Fridgen, Gilbert; Regner, Ferdinand; Schweizer, André; and Urbach, Nils** presented“DON’T SLIP ON THE ICO – A TAXONOMY FOR A BLOCKCHAIN-ENABLED FORM OFCROWDFUNDING” in Proceedings at AIS Electronic Library (AISeL),11-28-2018.ICO (Initial Coin Offering) is equivalent to IPO (Initial Public Offering). A new business model in which the funders can participate in funding process by using tokens. The project developers first posts a whitepaper which contains the project description. A presale is conducted with limited number of investors. A date is announced in which the public can purchase tokens to participate in project donation or can buy shares of the project. The disadvantages areit depends on the trust of the company that offers the tokens.[12]

1. **SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM**

In Present Crowdfunding system, the fundraiser and the funder will have to log in to the fundraising platform. And if they are a new user then they have to register in the fundraising platform. Here the fundraising platform is monitored by the fundraising agency(third-party). The fundraiser posts the need to raise funds in the fundraising platform. The funders who wish to donate can donate in this platform. Once the fundraiser received the entire amount that he has specified the post will be removed by the fundraising agency. But the funders will have no control over the money they have donated. The funder will have no idea if the money that he has donated has reached the fundraiser or not. The third-party i.e., the fundraising agency will have complete control over the money and hence they can perform malicious activities. The third-party involvement may also cause additional cost. As this platform is centralized by someone there is a chance to use the money the funders have donated in an illegal way.

**3.2 PROPOSED SYSTEM**

In crowdfunding we need to handle lot of transaction and document it accordingly. So here the blockchain concept is used to document these transactions which is executed automatically when it meets the certain condition. In this project we use blockchain concept to handle the transactions. The main entities in this project are fundraiser, funder, service provider platform and blockchain concept for tracking transactions. The funder have to register in the service provider platform if they are new user by providing Username, Password, E-mail ID, Phone number, CVV number, and Account number. The fundraiser have to register in the service provider platform if they are new user by providing Username, Password, E-mail ID, and Phone number. Admin will login using the Username and Password. For storing these meta details about the fundraiser and funder we use the MySQL database. Also, every time scanned the immutable entry added in the table.

CREATION:

This is the first stage in which the fundraiser will raise their funds by providing their Name, Account details, Mobile number, Description, and Amount details. Now these details will be posted on the service provided platform. Then the funder can view the details posted by the fundraiser and if they wishes to donate then they can proceed with that.

DONATION:

If the funder wishes to donate for any cause they can donate the money to the particular fundraiser by clicking on the particular post. Then it asks for CVV number and account number if these numbers matches with number that they have provided during the registration, they can proceed with payment process. In the payment process they have to provide Fundraiser name, Funder name, Amount to be transacted, Time, E-mail ID, and Account Number of fundraiser. Now the details of Fundraiser name, Funder Name, Account Number of fundraiser, Amount transacted, E-mail ID, Date and Time of the Transaction are recorded in the blockchain and it can be viewed in the Transaction history. Then this money will be added to the account of the fundraiser which they can further use for their need. All these transactions received can be viewed by the funder and fundraiser in the Transaction history. As the funder donate the money the amount posted in the service provider platform will reduce accordingly. The admin can view the transaction history, funder and fundraiser registration details. So, both the fundraiser and funder can trust this platform as this platform enables the fundraiser to receive all the amounts of the funder without any involvement of the third party and the funder can also verify that their amount is received by the valid fundraiser.

* 1. **REQUIREMENT ANALYSIS AND SPECIFICATION**

**3.3.1 INPUT REQUIREMENTS**

**1. NETBEANS 8.0.**

Netbeans is an open-source Integrated Development Environment (IDE). It is used to develop applications with **Java**, **PHP**, C++, HTML and many other programming languages. The applications are developed using modules in **Java**. NetBeans can run on any operating system such as Windows, MacOS, Linux etc.

**2. APACHE TOMCAT SERVER**

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the Java Server Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server.

1. **SQLyog**

SQLyog is a GUI tool for the RDBMS MySQL. **It** is the most powerful manager, admin and GUI tool for MySQL, combining the features of MySQL Query Browser, Administrator, phpMyAdmin and other MySQL Front Ends and MySQL GUI tools in a single intuitive interface.

* + 1. **OUTPUT REQUIREMENTS**

System with64 bit distribution capable of running 32 bit application and 1200\*800 minimum screen resolution with stable internet connection

* + 1. **FUNCTIONAL REQUIREMENTS**

As a user, I must be able to register and enter the website

1. Sign-up and register page
2. Logging in to website.

As a funder, I must be able to view posted information and explore them.

1. A page with a list of the posted requirements or needs.
2. A feature that enables funder to explore different posted information.
3. A feature that enables funder to transfer the money to fundraiser and features that enables to store these transaction in blockchain.

As a fundraiser, I must be able to post information in the home page.

1. Feature that enables them to view the transaction history and track the amount.

**3.4 HARDWARE ENVIRONMENT**

* Memory of 4 GB RAM.
* 64 bit distribution capable of running 32 bit application.
* 1200\*800 minimum screen resolution.
* 2 GB of available disk space minimum 4 GB recommended.

**3.5 SOFTWARE ENVIRONMENT**

* Language: Java
* Technology: Blockchain
* Platform(IDE): NetBeans 8.0.2
* Database Tool: SQLyog tool
* Operating System: Windows 10

### **THE JAVA PROGRAMMING LANGUAGE:**

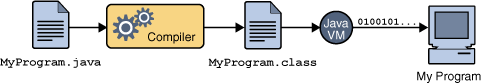
The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

* Simple
* Architecture neutral
* Object oriented
* Portable
* Distributed
* High performance

Each of the preceding buzzwords is explained in [The Java Language Environment](http://java.sun.com/docs/white/langenv/) , a white paper written by James Gosling and Henry McGilton.

In the Java programming language, all source code is first written in plain text files ending with the .java extension. Those source files are then compiled into .class files by the javac compiler.

A .class file does not contain code that is native to your processor; it instead contains byte codes — the machine language of the Java Virtual Machine[1](http://java.sun.com/docs/books/tutorial/getStarted/intro/definition.html#FOOT#FOOT) (Java VM). The java launcher tool then runs your application with an instance of the Java Virtual Machine.



**Fig 3.5 An overview of the software development process**

Because the Java VM is available on many different operating systems, the same .class files are capable of running on Microsoft Windows, the Solaris TM Operating System (Solaris OS), Linux, or Mac OS. Some virtual machines, such as the Java Hot Spot virtual machineperform additional steps at runtime to give your application a performance boost. This include various tasks such as finding performance bottlenecks and recompiling (to native code) frequently used sections of code.

**APACHE TOMCAT SERVER**

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

**Environment**  
 Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments. Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

**BLOCKCHAIN**

With the emergence of Digital Currency (aka Crypto currency), several enterprises or financial institutions are experimenting with the Distributed Ledger system as a trusted way to track the ownership of the assets without any central authority. The core system behind the new currency system is Blockchain technology. A walkthrough of the basic building blocks of the Blockchain technology is described below.

A Blockchain is basically a chain of Blocks. Blocks are hashed using SHA-256 hashing algorithm to generate the signature of the data associated with it.

Imagine a Blockchain as a linked-list whose node contains below attributes:

1. Block number – a sequence number (monotonically increasing) assigned to the block
2. Nonce – a random number which is used to generate Hash (as in #5) value which starts with 4 zeroes (0000). The process of generating this Nonce is called Mining.
3. Data – the actual user data associated with the block
4. Prev – contains the Hash of the previous block (e.g. current block # -1). The value for the first block in the chain is 64 zeroes (0000000000000000000000000000000000000000000000000000000000000000).
5. Hash – current block’s Hash value (generated using SHA-256). All of the above attributes excluding Hash e.g. Block #, Nonce, data, Prev are used to calculate the Hash of this block.

[#=1, Nonce=3409, Data=x, Prev=00..0, Hash=0000ffgr5rg67j] <- [#=2, Nonce=4986, Data=x, Prev=0000ffgr5rg67j, Hash=000045tggr5rg..77yh] <-……and the chain goes on…

e.g. in above block #1, the value for Hash=0000ffgr5rg67j is generated using the values 1,3409,x,00..0. In case value for any of these 4 attributes changes, it will change the Hash value of this block. Once the Hash value of this Block changes (e.g. from 0000ffgr5rg67j to 34sdffgr5rg67j), it will break the next Block (#2) as its Prev field will point to invalid Hash (0000ffgr5rg67j doesn’t exist anymore). This leads to a ripple effect and turns whole chain as invalid/tampered.

One way to fix it is to run mining and recalculate the Hash value of Block #1 which basically will generate new value for Nonce and hence leading to a valid Hash value which starts with 4 zeroes. Copying this to next Block #2’s Prev field will fix these 2 Blocks. However in order to fix the whole Blockchain, we need to continue with this process for all the Blocks in the chain so that all Blocks point to new & valid Hash codes of their previous blocks.

The cost of fixing the tampered Blockchain as described in above process is very high. Because we have to go and fix the Chain from the starting Block to the last one. In case the Chain is large, it becomes costly operation. In case of Distributed Blockchain where several Peers are involved in the process and keeping the copy of the Blockchain, the repairing the Blocks becomes even more costly operation.

The other and more efficient process is to come up with the compensating data and add this Block at the end of the Chain. E.g. In case your Chain contains the financial transaction (money movement) in Data field of the Block, then instead of fixing each of the Block’s Data with corrected financial transaction, come up with the adjusted financial transaction (aka compensating transaction) and create a Block (with Data=adjusted transaction record) and add this Block to the Blockchain (adds to the end of the Chain).

**SQL**

SQLyog is a [GUI](https://en.wikipedia.org/wiki/Graphical_user_interface) tool for the [RDBMS](https://en.wikipedia.org/wiki/RDBMS) [MySQL](https://en.wikipedia.org/wiki/MySQL). It is developed by Webyog, Inc. SQLyog v0.9 was first released to the public in 2001 after eight months of development. SQLyog was available free of charge, but with closed source code, until v3.0 when it was made a fully commercial software. This makes it possible to specify and test "scheduled jobs" on a Windows environment and port execution parameters seamlessly to a Linux environment.

**4. SYSTEM DESIGN**

* 1. **DATA DICTIONARY**

**4.1.1. REGISTRATION DB FOR CROWDFUNDING:**

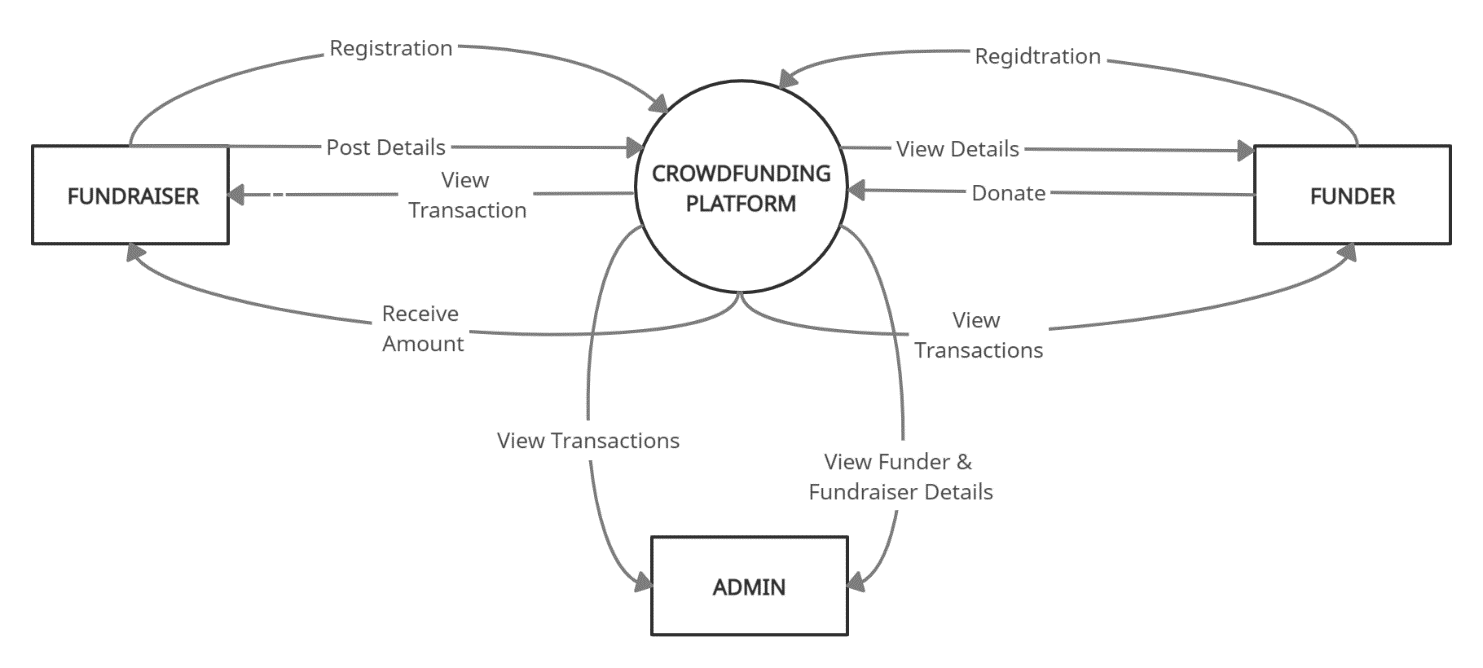
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **LENGTH** | **DESCRIPTION** | **CONSTRAINTS** |
| Username | Varchar | 200 | Name of the user | Primary Key |
| Password | Varchar | 200 | Password should contains capital letter, small letter, special character and number | Not Null |
| Email | Varchar | 200 | Mail id of the user | Primary Key |
| Mobile | Varchar | 200 | Mobile number of the user | Not Null |

**4.1.2. FUNDRAISER DB FOR CROWDFUNDING:**

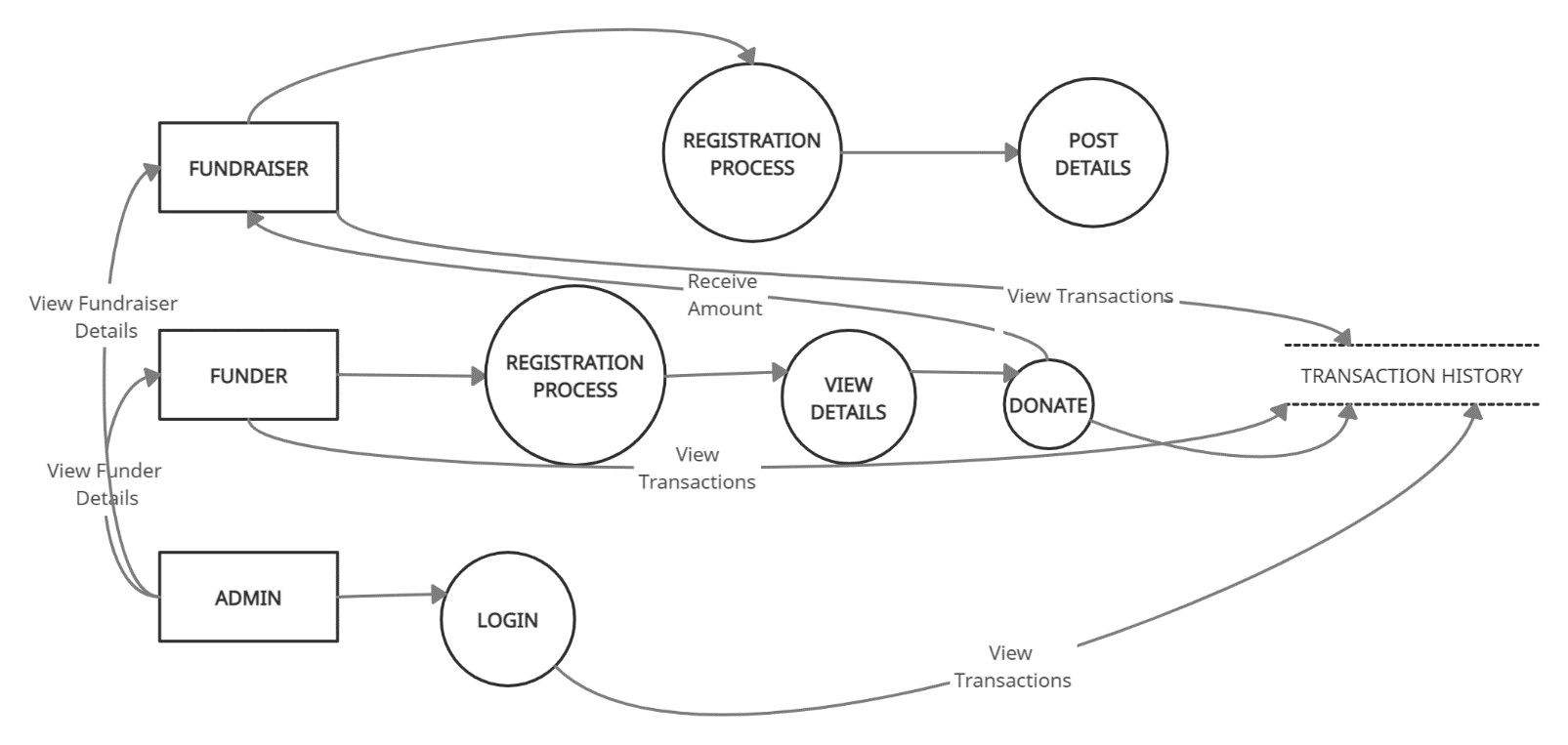
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FIELD NAME** | **DATATYPE** | **LENGTH** | **DESCRIPTION** | **CONSTRAINTS** |
| Username | Varchar | 200 | Name of the fundraiser | Primary Key |
| Account Number | Varchar | 12 | Account number of the fundraiser | Primary Key |
| Amount | Varchar | 200 | Amount that they are expecting | Not Null |
| Mobile | Varchar | 200 | Mobile number of the fundraiser | Not Null |
| Description | Varchar | 200 | Reason for their fundraising | Not Null |

* 1. **DATA FLOW DIAGRAM FOR CROWDFUNDING:**

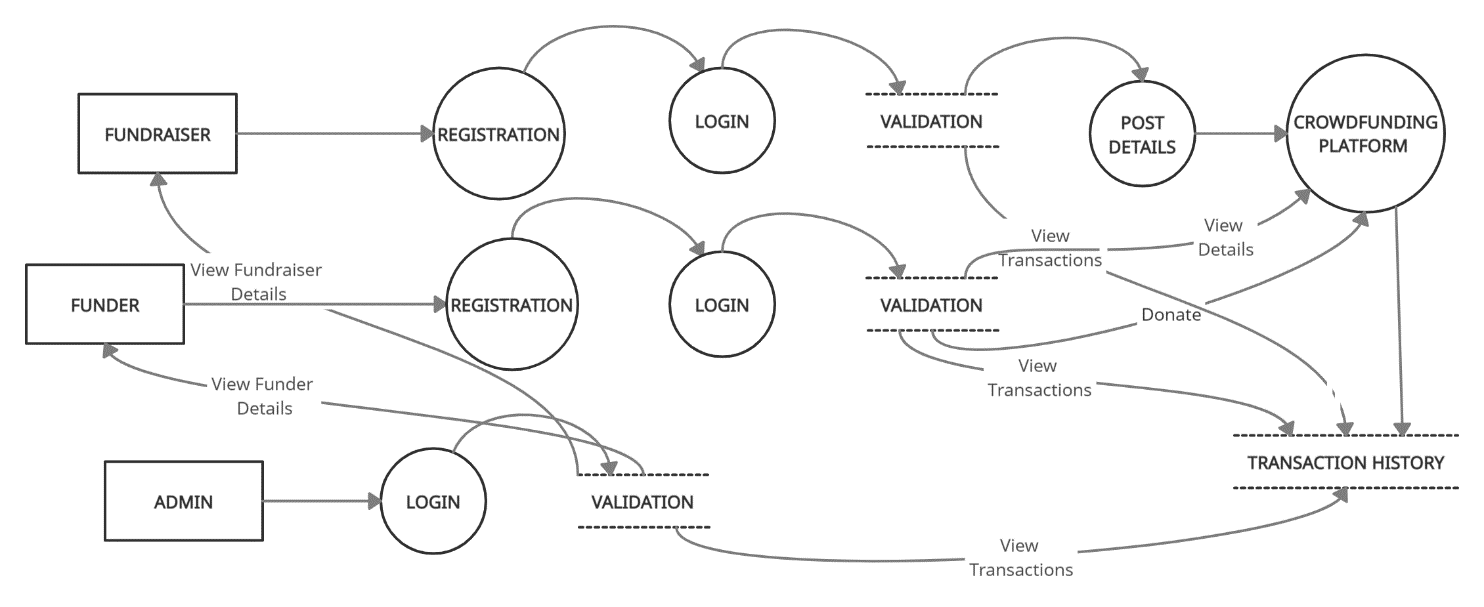
A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

****

**Fig-4.2.1 Data flow diagram level 0 for crowdfunding**

****

**Fig-4.2.2 Data flow diagram level 1 for crowdfunding**

****

**Fig-4.2.3 Data flow diagram level 2 for crowdfunding**

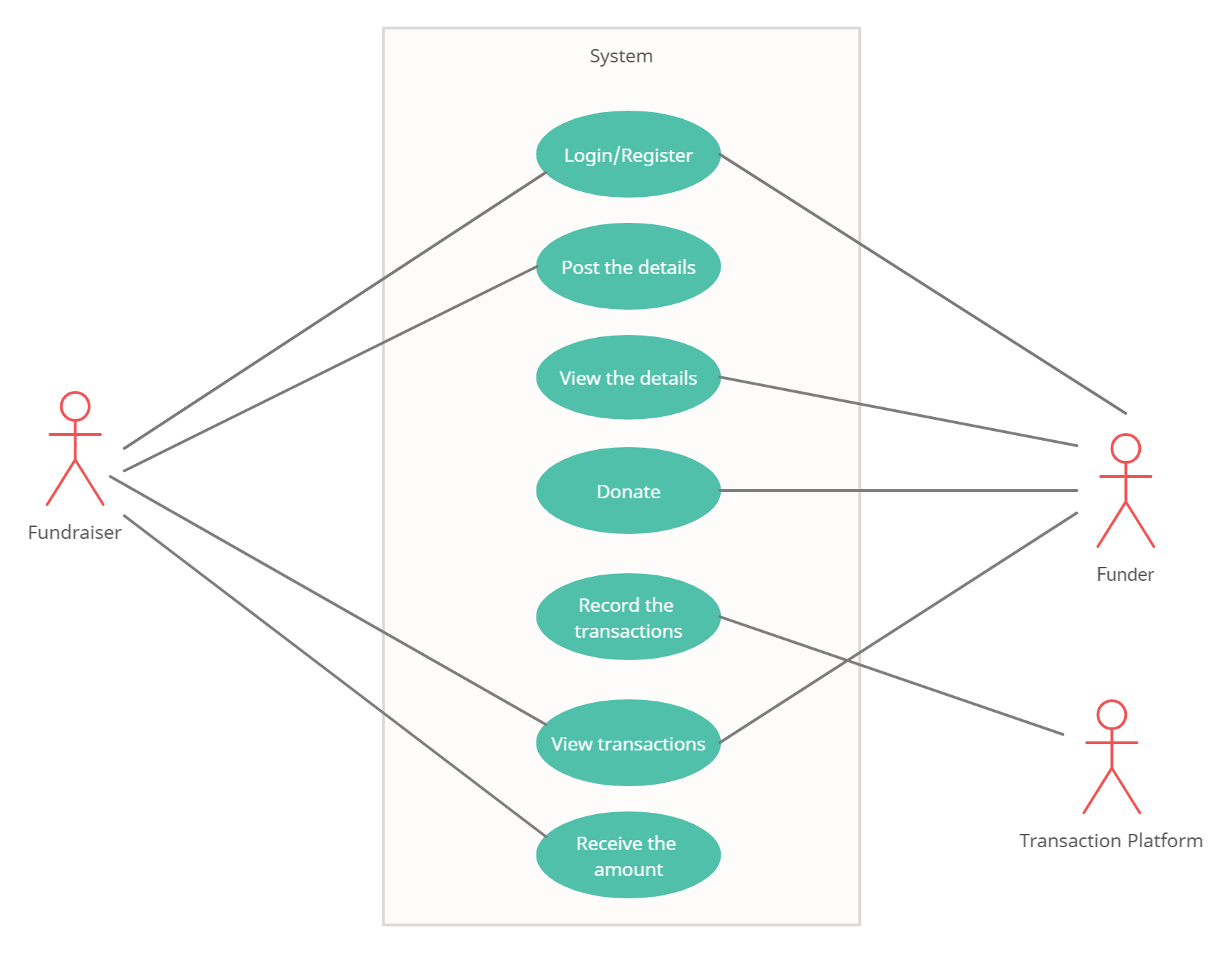
**4.3 UML DIAGRAMS**

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

* + 1. **USE CASE FOR CROWDFUNDING:**

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases.A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

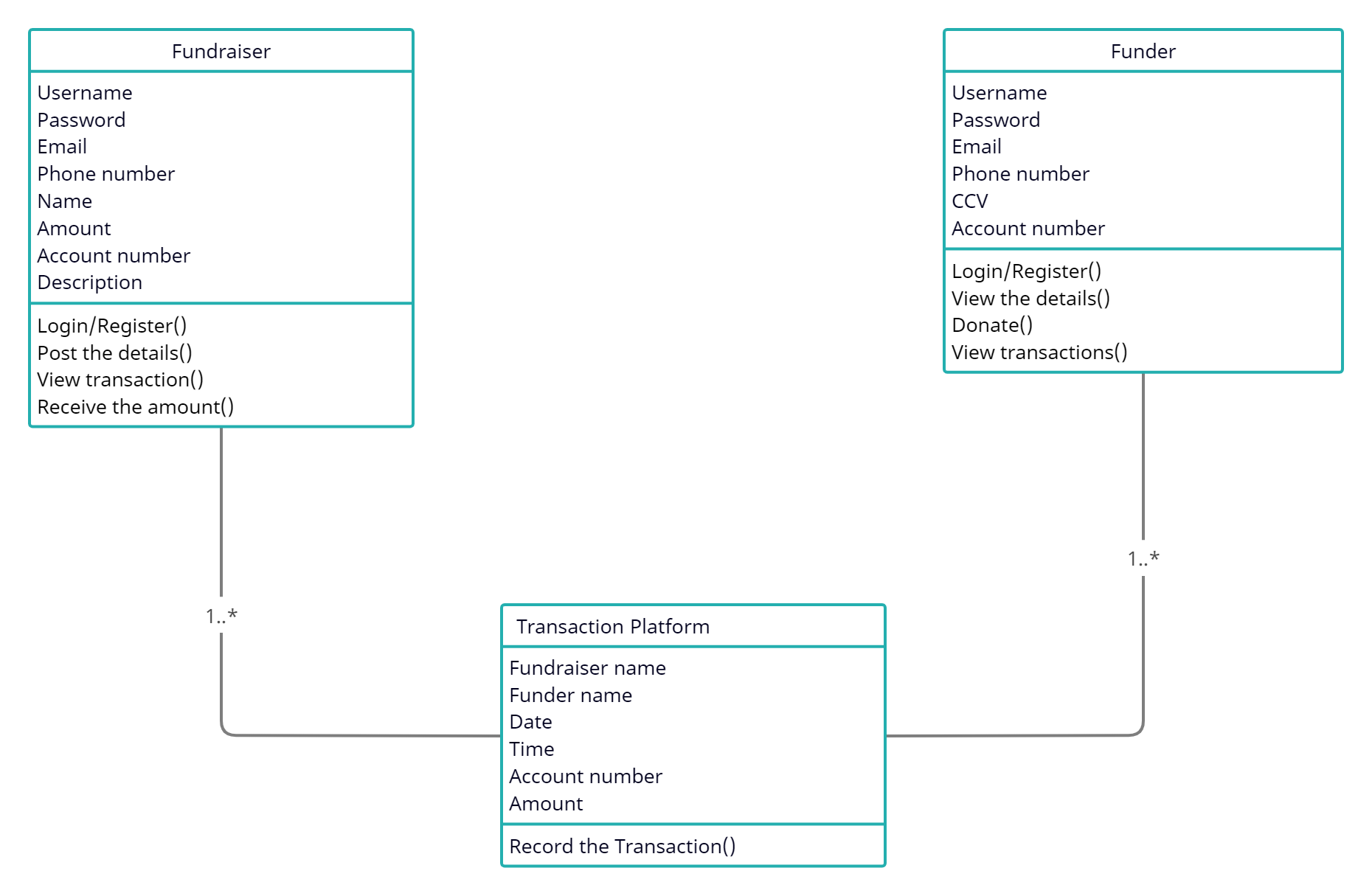
.



**Fig-4.3.1 Use case diagram for crowdfunding**

* + 1. **CLASS DIAGRAM FOR CROWDFUNDING:**

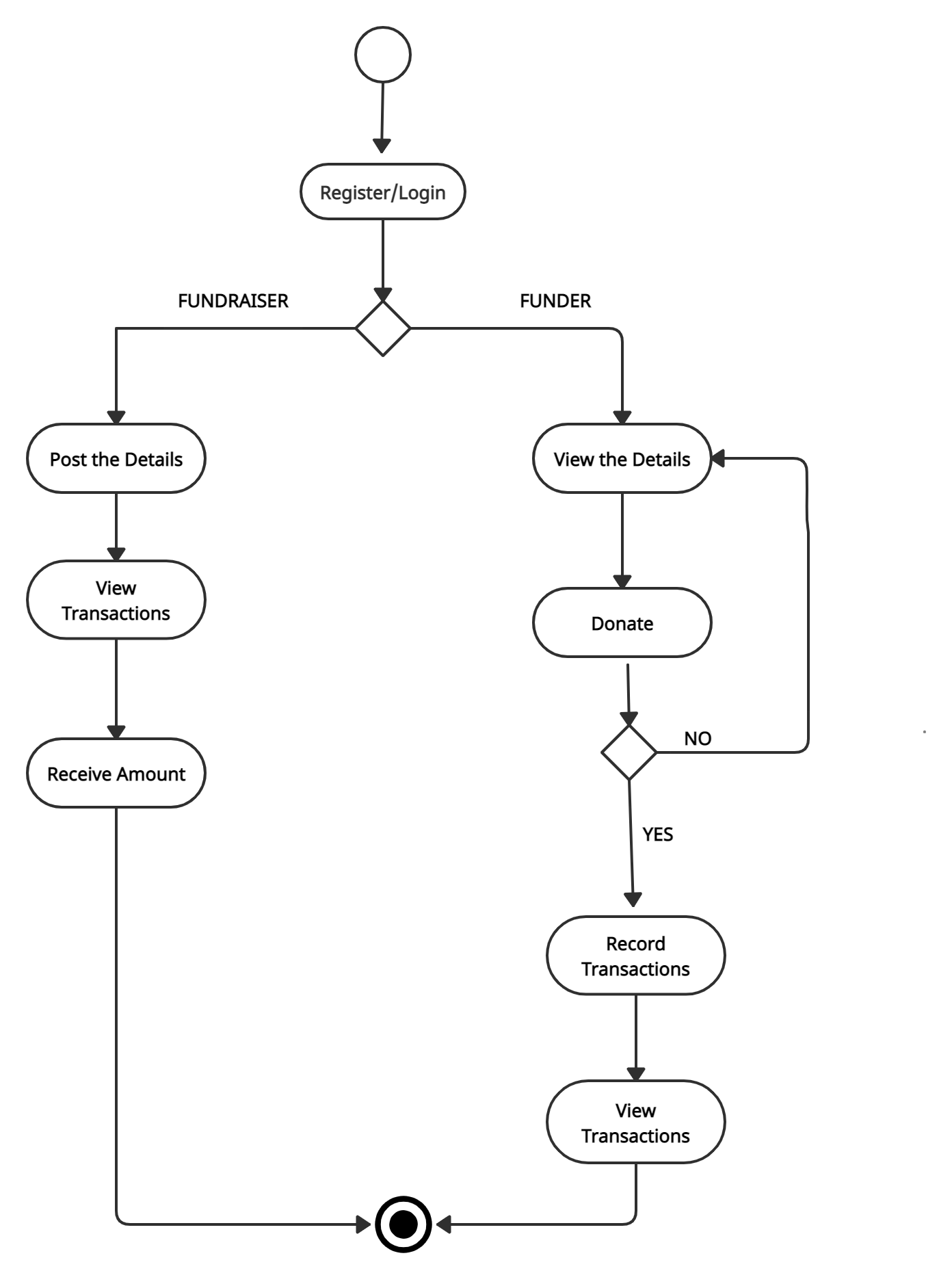
A Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

****

**Fig-4.3.2 Class diagram for crowdfunding**

* + 1. **ACTIVITY DIAGRAM FOR CROWDFUNDING:**

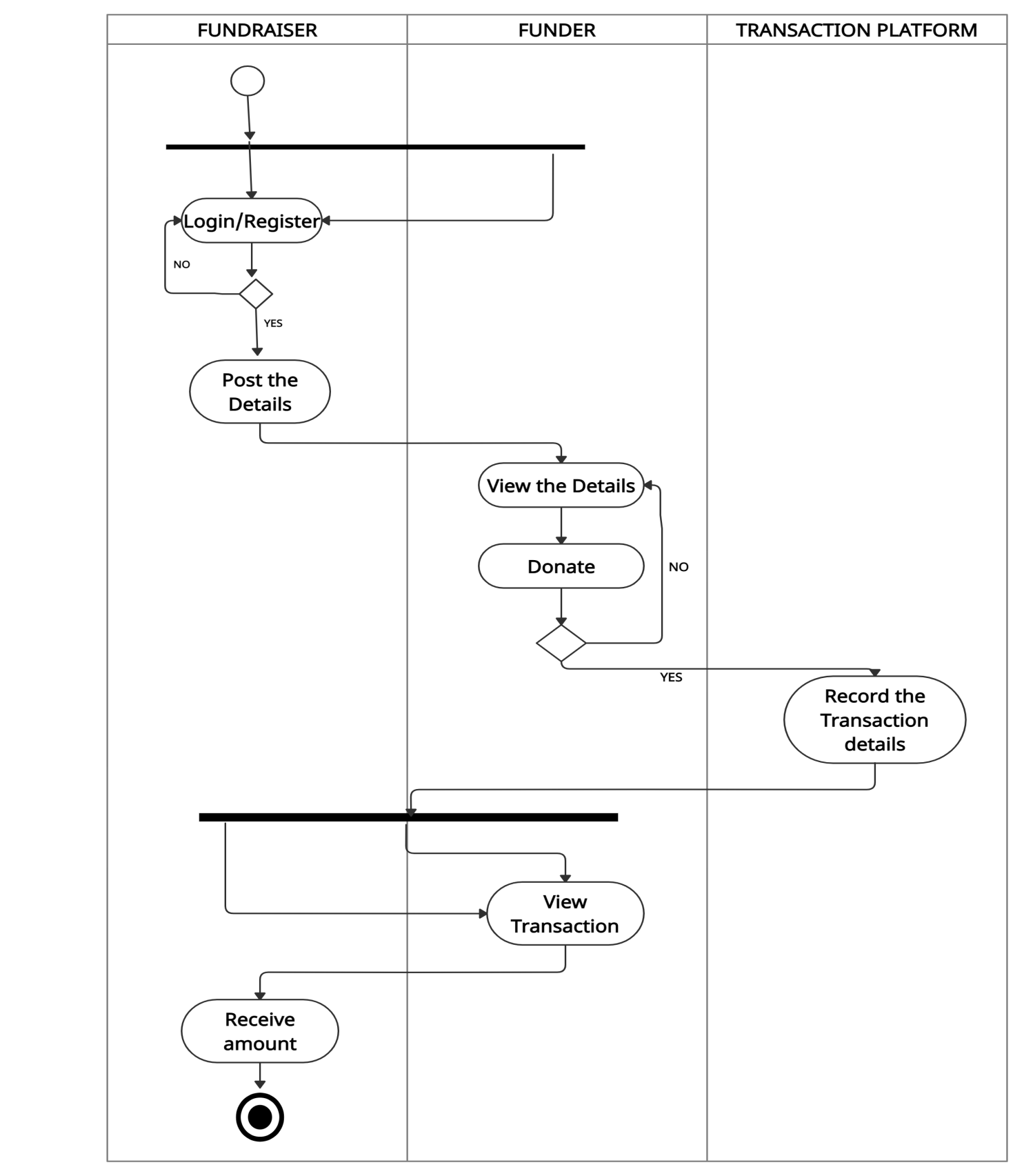
Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.



**Fig-4.3.3 Activity diagram for crowdfunding**

* + 1. **SWIMLANE DIAGRAM FOR CROWDFUNDING:**

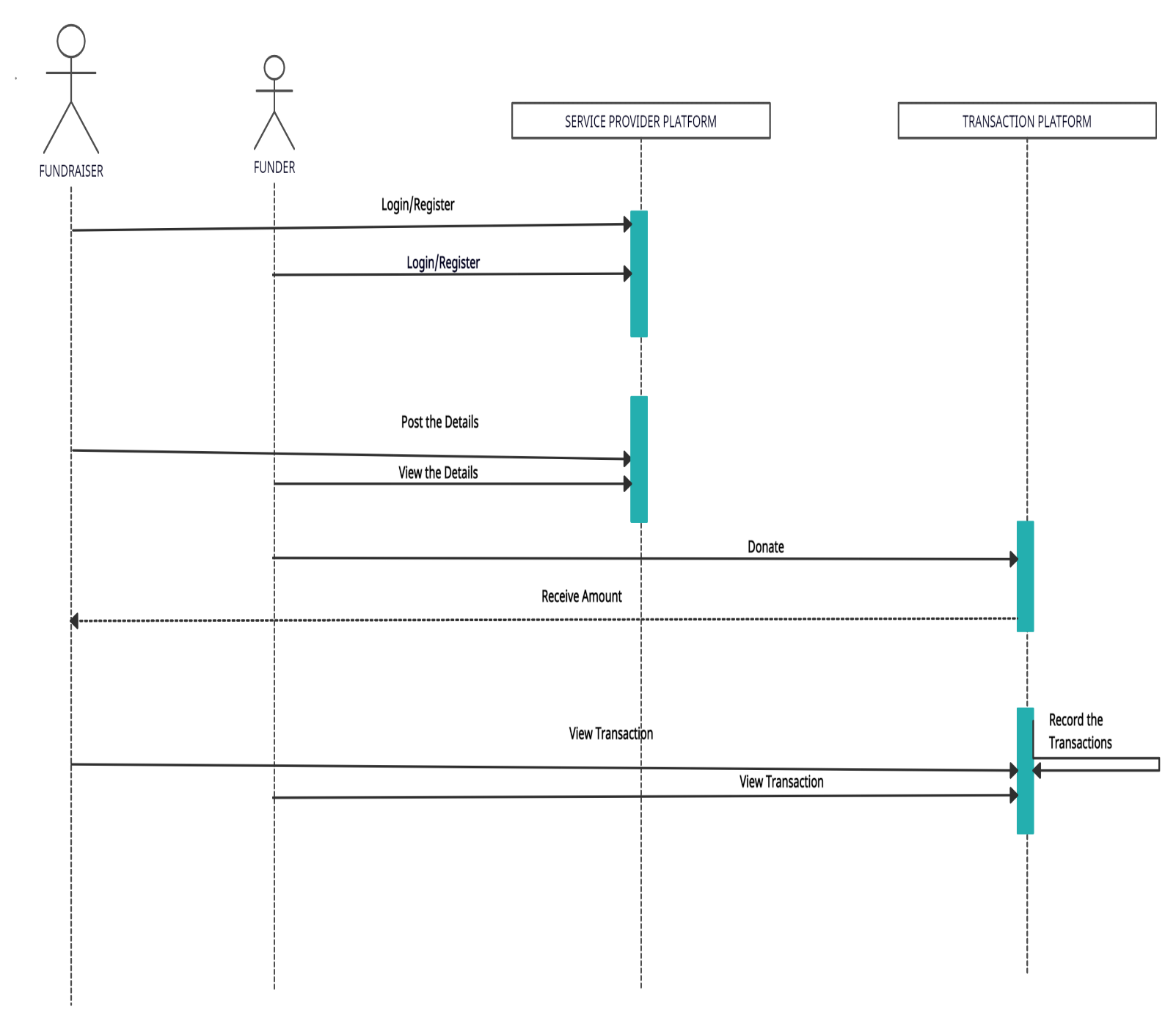
A swim lane diagram is a type of flowchart that delineates who does what in a process. Using the metaphor of lanes in a pool, a swim lane diagram provides clarity and accountability by placing process steps within the horizontal or vertical “swim lanes” of a particular employee, work group or department.

****

**Fig-4.3.4 Swim lane diagram for crowdfunding**

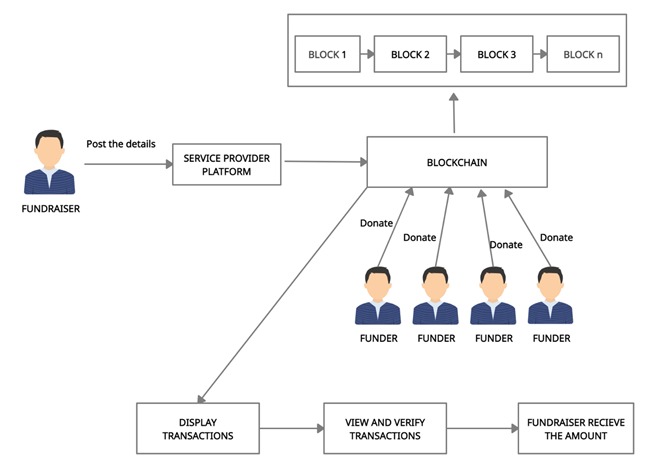
**4.3.5 SEQUENCE DIAGRAM FOR CROWDFUNDING:**

A sequence diagram shows object interactions arranged in time sequence. It depicts the object and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

**Fig-4.3.5 Sequence diagram for crowdfunding**

1. **SYSTEM ARCHITECTURE FOR CROWDFUNDING:**
   1. **ARCHITECTURE OVERVIEW**

This is a website-based system in which the fundraiser and funder have to register in the service provider platform. The fundraiser will raise their funds by describing the need. Then the funder can view all the information posted by various fundraiser. If the funder wishes to donate for any particular cause they can proceed with it. Then the money that is donated by the funder will be added to the account of the fundraiser. The sum of the amount that has been donated by all the funders will also be displayed, with the help of this we can identify whether the specified amount has been reached or not. Once they donated the money to the fundraiser the information of the transaction is stored in blockchain and that information can be viewed by both the fundraiser and funder. The funder can track their money until the transacted amount reaches the valid recipient.



**Fig-5.1 Architecture Diagram For Crowdfunding**

**5.2 MODULE DESIGN SPECIFICATION**

1.Registration – Fundraiser

1. Registration – Funder
2. Post Creation
3. Service provider Platform
4. Payment platform
5. Transaction history
6. Admin Platform
7. Blockchain creating and mining
   1. **REGISTRATION – FUNDRAISER**

The fundraiser have to register in the service provider platform if they are new user by providing Username, Password, E-mail ID, and Phone number. For storing these meta details about the fundraiser we use the MySQL database. Also every time scanned the immutable entry added in the table.

* 1. **REGISTRATION – FUNDER**

The funder have to register in the service provider platform if they are new user by providing Username, Password, E-mail ID, Phone number, CVV number, and Account number. For storing the meta details about the funder we use the MySQL database. Also every time scanned the immutable entry added in the table.

* 1. **POST CREATION**

In this module the fundraiser will raise their funds by providing their Name, Account details, Mobile number, Description, and Amount details. Now these details will be posted on the service provided platform.

* 1. **SERVICE PROVIDER PLATFORM**

The details given during the post creation by the fundraiser will be posted here. Then the funder can view the details posted by the fundraiser and if they wishes to donate then they can proceed with that. If the funder wishes to donate for any cause they can donate the money to the particular fundraiser by clicking on the particular post. Then it asks for CVV number and account number if these numbers matches with number that they have provided during the registration, they can proceed with payment process.

* 1. **PAYMENT PLATFORM**

In the payment process they have to provide Fundraiser name, Funder name, Amount to be transacted, Time, E-mail ID, and Account Number of fundraiser. Now the details of Fundraiser name, Funder Name, Account Number of fundraiser, Amount transacted, E-mail ID, Date and Time of the Transaction are recorded in the blockchain.

* 1. **TRANSACTION PLATFORM**

Now the details of Fundraiser name, Funder Name, Account Number of fundraiser, Amount transacted, E-mail ID, Date and Time with be displayed in the transaction history. This can be viewed by funder, fundraiser and admin. They can keep track of their money through this transaction platform.

* 1. **ADMIN PLATFORM**

Here the admin will login using username and password. Once they login they can view funder details, fundraiser details and transaction history.

* 1. **BLOCKCHAIN CREATING AND MINING**

On startup, create the genesis block then server initiates creation of new block and send the required data Create a new block with the scan data. Mine the block.

1. **SYSTEM IMPLEMENTATION**
   1. **CLIENT SIDE CODING**

**Block.java**

import java.util.Date;

public class Block {

public String hash;

public String previousHash;

private String data; //our data will be a simple message.

private long timeStamp; //as number of milliseconds since 1/1/1970.

private int nonce;

//Block Constructor.

public Block(String data,String previousHash ) {

this.data = data;

this.previousHash = previousHash;

this.timeStamp = new Date().getTime();

this.hash = calculateHash(); //Making sure we do this after we set the other values.

}

Block(String string) {

throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templates.

}

//Calculate new hash based on blocks contents

public String calculateHash() {

String calculatedhash = StringUtil.applySha256(

previousHash +

Long.toString(timeStamp) +

Integer.toString(nonce) +

data

);

return calculatedhash;

}

//Increases nonce value until hash target is reached.

public void mineBlock(int difficulty) {

String target = StringUtil.getDificultyString(difficulty); //Create a string with difficulty \* "0"

while(!hash.substring( 0, difficulty).equals(target)) {

nonce ++;

hash = calculateHash();

}

System.out.println("Block Mined!!! : " + hash);

}

}

**Noobchain.java**

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

import java.util.ArrayList;

public class NoobChain {

public static ArrayList<Block> blockchain = new ArrayList<>();

public static int difficulty = 5;

static int i =5;

public static void main(String[] args) {

//add our blocks to the blockchain ArrayList:

try{

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

Connection con=DriverManager.getConnection("jdbc:mysql://localhost:3306/crowdfund","root","password");

Statement st=con.createStatement();

ResultSet rs=st.executeQuery("Select \* from transaction");

addBlock(new Block("Genius block","0"));

while(rs.next()){

String qrvalue = rs.getString("fname");

String username = rs.getString("frname");

String currentdate = rs.getString("accno");

String amount = rs.getString("amount");

String date = rs.getString("date");

String time = rs.getString("time");

String mail = rs.getString("email");

String all = qrvalue+username+currentdate+amount+date+time+mail;

while(true)

{

addBlock(new Block(all, blockchain.get(blockchain.size()-1).hash));

String blockchainJson = StringUtil.getJson(blockchain);

System.out.println("\nThe block chain: ");

System.out.println(blockchainJson);

}

}

}

catch(ClassNotFoundException | SQLException e){

System.out.println(e);

}

}

public static Boolean isChainValid() {

Block currentBlock;

Block previousBlock;

String hashTarget = new String(new char[difficulty]).replace('\0', '0');

//loop through blockchain to check hashes:

for(int i=1; i < blockchain.size(); i++) {

currentBlock = blockchain.get(i);

previousBlock = blockchain.get(i-1);

//compare registered hash and calculated hash:

if(!currentBlock.hash.equals(currentBlock.calculateHash()) ){

System.out.println("Current Hashes not equal");

return false;

}

//compare previous hash and registered previous hash

if(!previousBlock.hash.equals(currentBlock.previousHash) ) {

System.out.println("Previous Hashes not equal");

return false;

}

//check if hash is solved

if(!currentBlock.hash.substring( 0, difficulty).equals(hashTarget)) {

System.out.println("This block hasn't been mined");

return false;

}

}

return true;

}

public static void addBlock(Block newBlock) {

newBlock.mineBlock(difficulty);

blockchain.add(newBlock);

}

}

* 1. **SERVER SIDE CODING**

**Charityhome.jsp**

<%--

Document : adminhome

Created on : 4 Nov, 2020, 3:47:54 PM

Author : h

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<link rel="stylesheet" href="https://www.w3schools.com/w3css/4/w3.css">

<title>Donor Home\_page</title>

<style>

body {

background-image: url("images/cover\_bg\_2.jpg");

background-color: #cccccc;

}

</style>

</head>

<body align ="center">

<div class="w3-container">

<hr/>

<br/><br/><br/>

<a href="fundraise.jsp" class="w3-btn w3-black">Fund Raise</a>

<a href="fundhistn.jsp" class="w3-btn w3-black">Fund History</a>

<br/><br/><br/>

<hr/>

</div>

</body>

</html>

**Fundraise.jsp**

<%--

Document : buyer

Created on : 28 Nov, 2020, 10:21:22 AM

Author : h

--%>

<%@page contentType="text/html" pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Buyer</title>

<style>

body{

margin:0;

color:#6a6f8c;

background:#c8c8c8;

font:600 16px/18px 'Open Sans',sans-serif;

}

\*,:after,:before{box-sizing:border-box}

.clearfix:after,.clearfix:before{content:'';display:table}

.clearfix:after{clear:both;display:block}

a{color:inherit;text-decoration:none}

.login-wrap{

width:100%;

margin:auto;

max-width:525px;

min-height:670px;

position:relative;

background:url(https://raw.githubusercontent.com/khadkamhn/day-01-login-form/master/img/bg.jpg) no-repeat center;

box-shadow:0 12px 15px 0 rgba(0,0,0,.24),0 17px 50px 0 rgba(0,0,0,.19);

}

.login-html{

width:100%;

height:100%;

position:absolute;

padding:90px 70px 50px 70px;

background:rgba(40,57,101,.9);

}

.login-html .sign-in-htm,

.login-html .sign-up-htm{

top:0;

left:0;

right:0;

bottom:0;

position:absolute;

transform:rotateY(180deg);

backface-visibility:hidden;

transition:all .4s linear;

}

.login-html .sign-in,

.login-html .sign-up,

.login-form .group .check{

display:none;

}

.login-html .tab,

.login-form .group .label,

.login-form .group .button{

text-transform:uppercase;

}

.login-html .tab{

font-size:22px;

margin-right:15px;

padding-bottom:5px;

margin:0 15px 10px 0;

display:inline-block;

border-bottom:2px solid transparent;

}

.login-html .sign-in:checked + .tab,

.login-html .sign-up:checked + .tab{

color:#fff;

border-color:#1161ee;

}

.login-form{

min-height:345px;

position:relative;

perspective:1000px;

transform-style:preserve-3d;

}

.login-form .group{

margin-bottom:15px;

}

.login-form .group .label,

.login-form .group .input,

.login-form .group .button{

width:100%;

color:#fff;

display:block;

}

.login-form .group .input,

.login-form .group .button{

border:none;

padding:15px 20px;

border-radius:25px;

background:rgba(255,255,255,.1);

}

.login-form .group input[data-type="password"]{

text-security:circle;

-webkit-text-security:circle;

}

.login-form .group .label{

color:#aaa;

font-size:12px;

}

.login-form .group .button{

background:#1161ee;

}

.login-form .group label .icon{

width:15px;

height:15px;

border-radius:2px;

position:relative;

display:inline-block;

background:rgba(255,255,255,.1);

}

.login-form .group label .icon:before,

.login-form .group label .icon:after{

content:'';

width:10px;

height:2px;

background:#fff;

position:absolute;

transition:all .2s ease-in-out 0s;

}

.login-form .group label .icon:before{

left:3px;

width:5px;

bottom:6px;

transform:scale(0) rotate(0);

}

.login-form .group label .icon:after{

top:6px;

right:0;

transform:scale(0) rotate(0);

}

.login-form .group .check:checked + label{

color:#fff;

}

.login-form .group .check:checked + label .icon{

background:#1161ee;

}

.login-form .group .check:checked + label .icon:before{

transform:scale(1) rotate(45deg);

}

.login-form .group .check:checked + label .icon:after{

transform:scale(1) rotate(-45deg);

}

.login-html .sign-in:checked + .tab + .sign-up + .tab + .login-form .sign-in-htm{

transform:rotate(0);

}

.login-html .sign-up:checked + .tab + .login-form .sign-up-htm{

transform:rotate(0);

}

.hr{

height:2px;

margin:60px 0 50px 0;

background:rgba(255,255,255,.2);

}

.foot-lnk{

text-align:center;

}

</style>

</head>

<body>

<div class="login-wrap">

<div class="login-html">

<input id="tab-1" type="radio" name="tab" class="sign-in" checked><label for="tab-1" class="tab">Raise Fund</label>

<input id="tab-2" type="radio" name="tab" class="sign-up"><label for="tab-2" class="tab"></label>

<div class="login-form">

<form action="fundraise" method="POST">

<div class="sign-in-htm">

<div class="group">

<label for="pass" class="label">Amount</label>

<input id="pass" type="text" name="amount" class="input" data-type="text">

</div>

<div class="group">

<label for="pass" class="label">Description</label>

<input id="pass" type="text" name="description" class="input" data-type="text">

</div>

<div class="group">

<label for="pass" class="label">Name</label>

<input id="pass" type="text" name="name" class="input" data-type="text">

</div>

<div class="group">

<label for="pass" class="label">Mobile number</label>

<input id="pass" type="text" name="phonenumber" class="input" data-type="text">

</div>

<div class="group">

<label for="pass" class="label">Account Details</label>

<input id="pass" type="text" name="ad" class="input" data-type="text">

</div>

<div class="group">

<input type="submit" class="button" value="Post\_Details">

</div>

</div>

</form>

</div>

</div>

</div>

</body>

</html>

1. **SYSTEM TESTING**

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consisting of several key activities and steps for running a program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing. Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design be detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough. Testing is one of the important steps in the software development phase. Testing check for the errors, as a whole of the project testing involves the following test cases: Static analysis is used to investigate the structural properties of the Source. Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

* 1. **UNIT TESTING**

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures are tested to determine if they are fit for use. Intuitively, one can view a unit as the smallest testable part of an application. In procedural programming, a unit could be an entire module, but is more commonly an individual function or procedure. In object-oriented programming, a unit is often an entire interface, such as a class, but could be an individual method. Unit tests are created by programmers or occasionally by white box testers during the development process. Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

In this project in the module of registration fundraiser the fundraiser should provide username, password, email and phone number. Each input is denoted with certain constraint and it is checked for violation of the constraint. For example, the email should end with @gmail.com and the phone number should contain 10 digits, violation of these constraints will display error. Similarly, for login the fundraiser should specify username and password if it is not registered then it will specify invalid username and password.

**Result:** All the test cases mentioned above passed successfully. No defects encountered

Similarly, in the module of registration of funder the funder should provide username, password, email, phone number, account number and cvv number. Each input is denoted with certain constraint and it is checked for violation of the constraint. For example, the email should end with @gmail.com, the phone number should contain 10 digits, the account number should contain 12 digits and the cvv number should contain 3 digits, violation of these properties will display error. Similarly, for login the funder should specify username and password if it is not registered then it will specify invalid username and password.

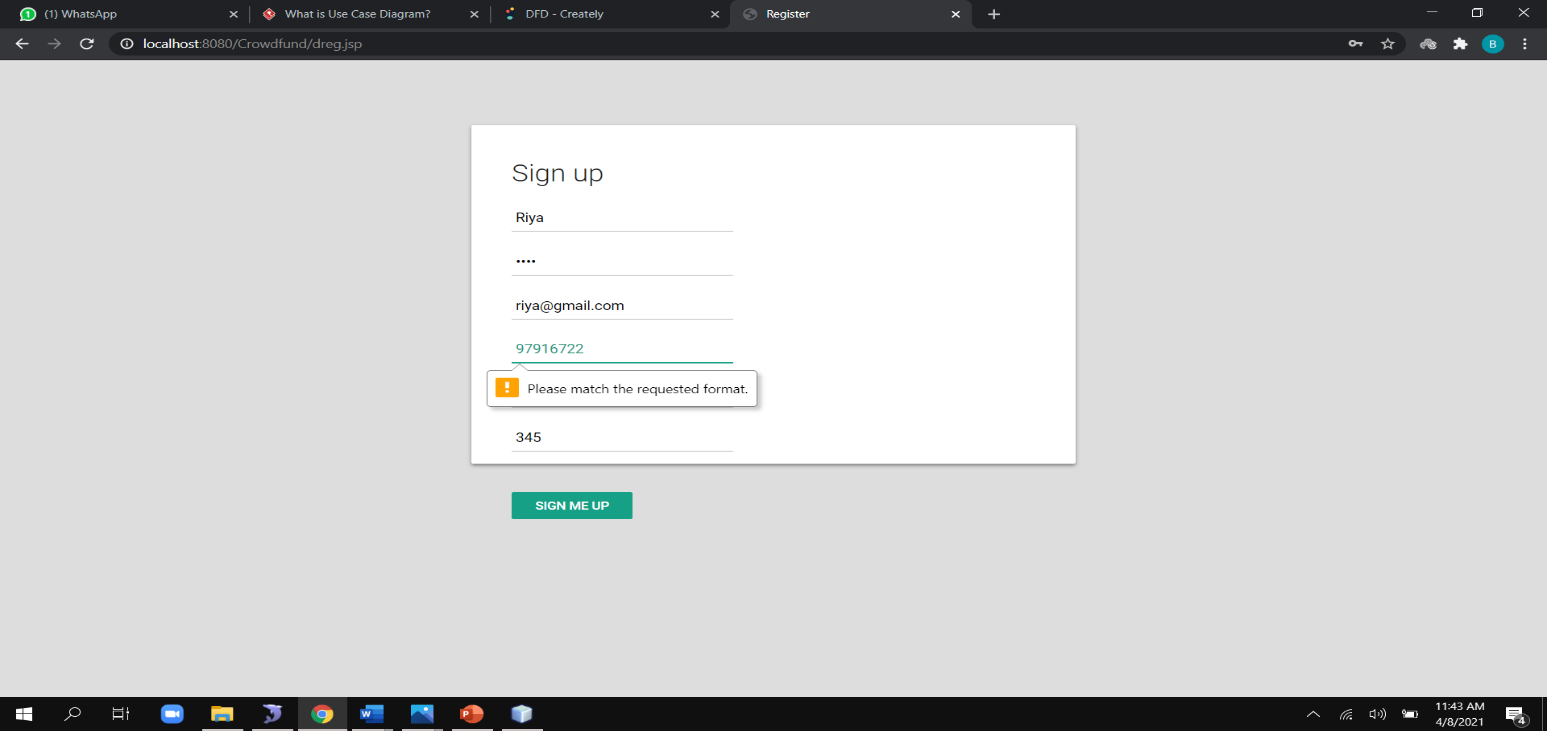
**Result:** All the test cases mentioned above passed successfully. No defects encountered

In the service provider platform, the details specified by the fundraiser are visible. The fundraiser should specify name, amount, phone number, description and account number. Each input is denoted with certain constraint and it is checked for violation of the constraint. In this the phone number should contain 10 digits and account number should contain 12 digits.

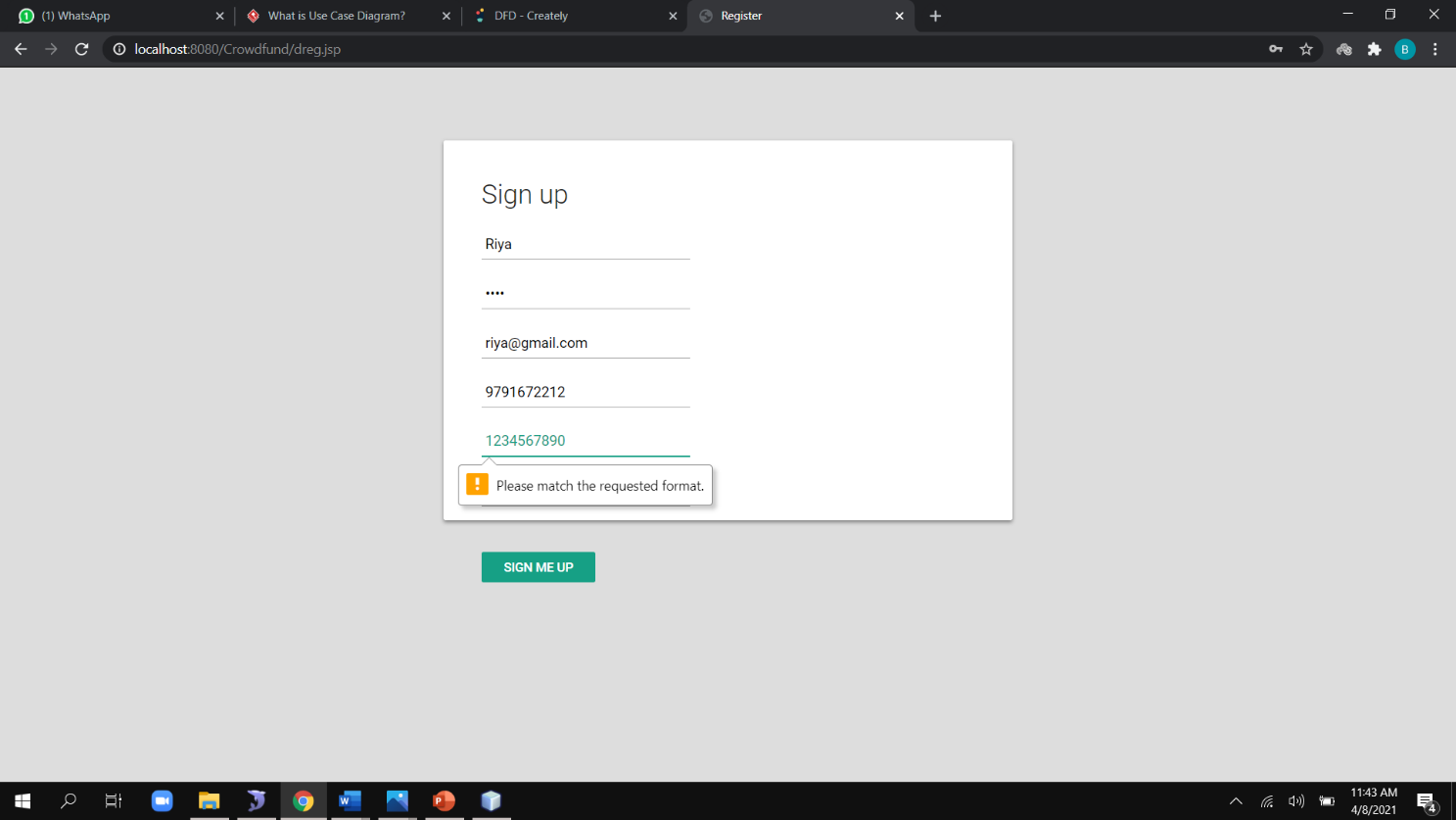
**Result:** All the test cases mentioned above passed successfully. No defects encountered

In the transaction platform module, the transaction which occurs are all stored in blockchain. The donor should specify the account number and cvv number which they have registered. Then they should specify funder name, fundraiser name, amount, email, account number and time. Each input is denoted with certain constraint and it is checked for violation of the constraint. In this the phone number should contain 10 digits, account number should contain 12 digits and cvv number should contain 3 digits.

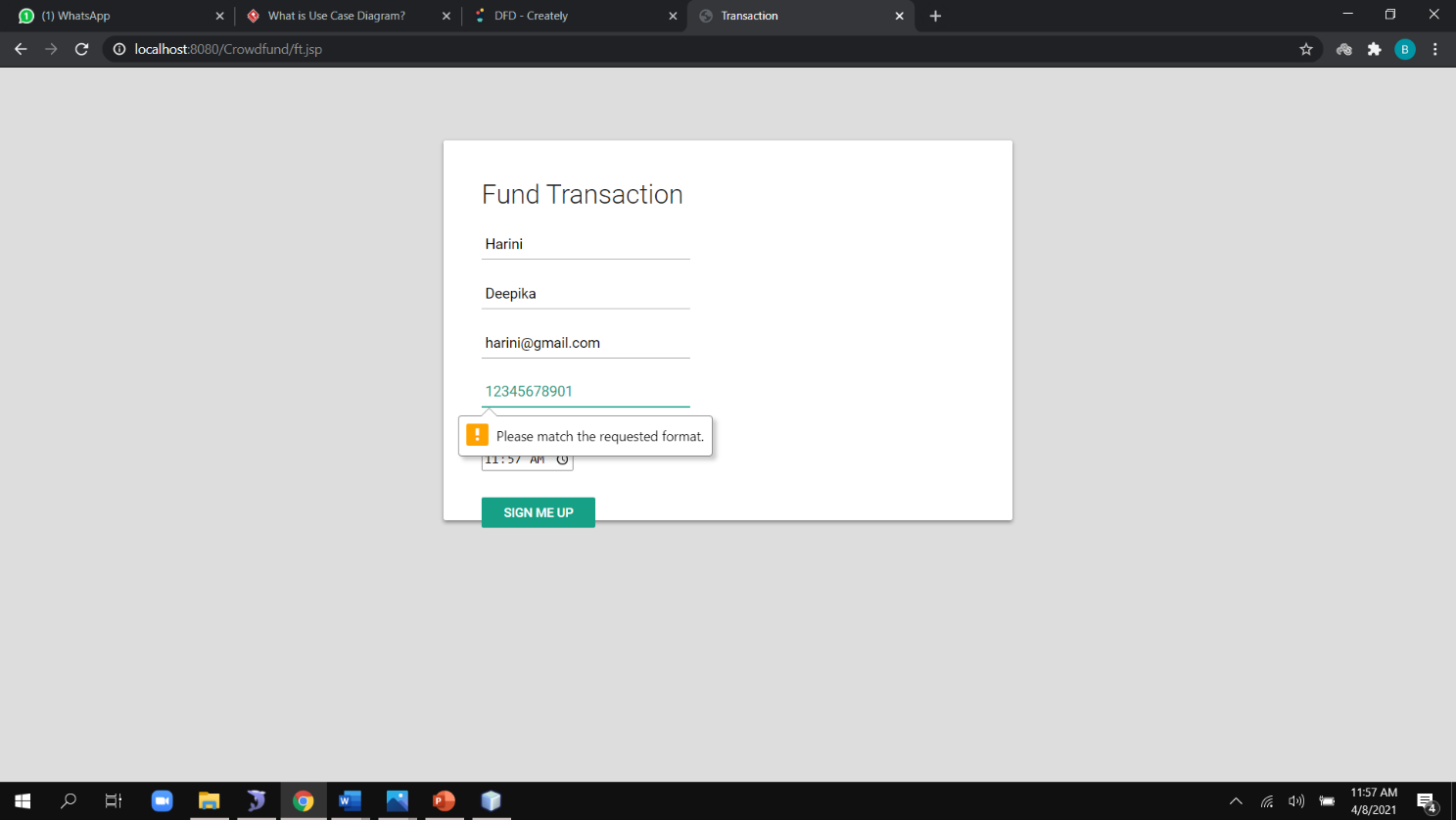
**Result:** All the test cases mentioned above passed successfully. No defects encountered

****

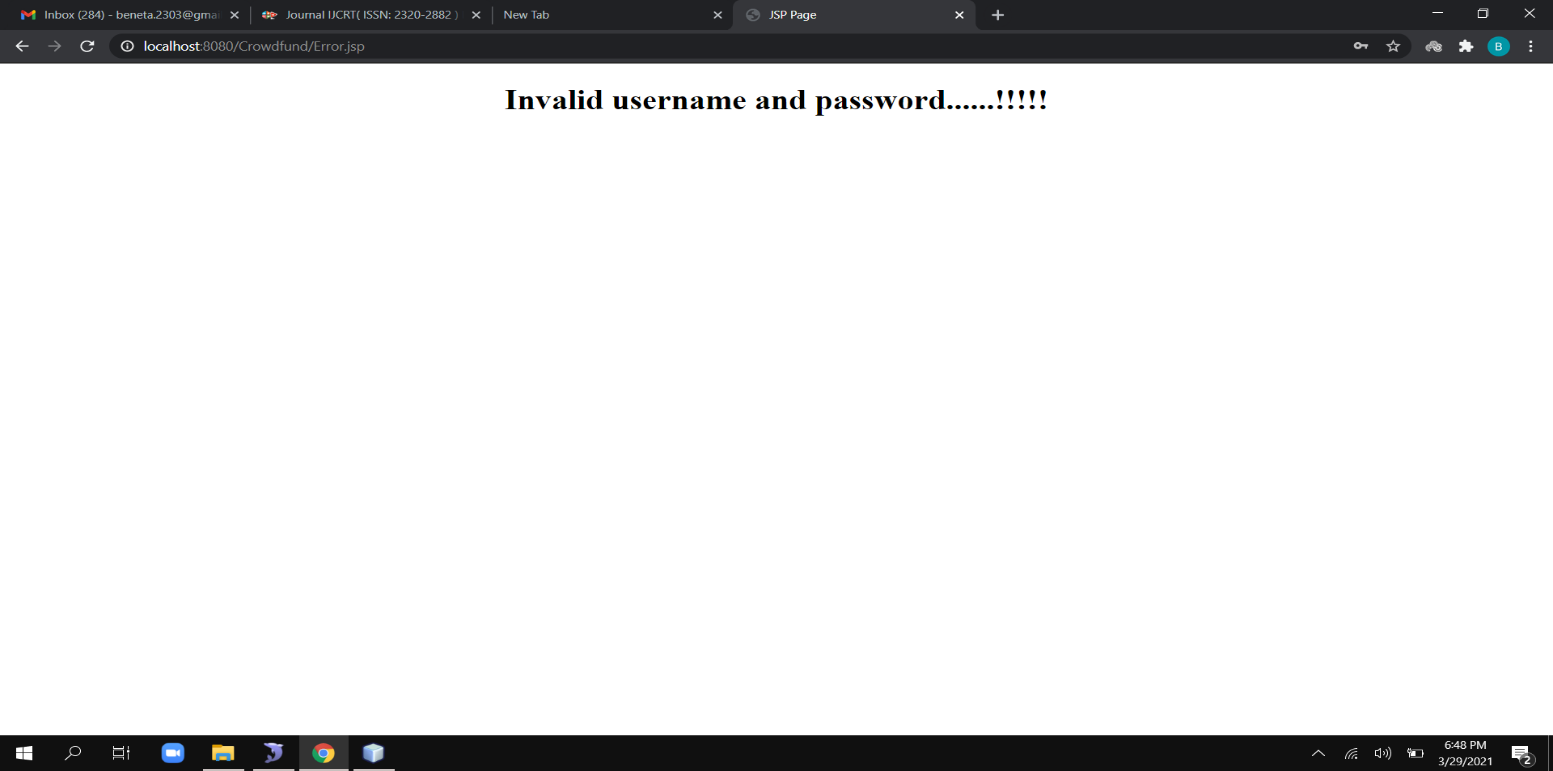
**Fig 7.1.1 Fundraiser register page to validate mobile number**

****

**Fig 7.1.2 Fundraiser register page to validate account number**

****

**Fig 7.1.3 Fund transaction page to validate account number**

****

**Fig 7.1.4 Invalid page**

**7.2 INTEGRATION TESTING**

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

In the fundraiser register module we first have to register and then login when we login it redirect to charityhome.jsp page. In charityhome page there are 2 options either to check fund history or to fund raise. When we click fund raise it redirects to fundraise.jsp page. In fundraise page the fundraiser specifies the details and click post details. Now all the information posted will be available in the service provider platform.

**Result:** All the test cases mentioned above passed successfully. No defects encountered

Similarly, In the funder registration module the funder can register and login the funder will be directed to the donorhomenew.jsp page. In donorhomenew page 2 options will be available. The funder can either check the fund history or he can donate by clicking fund request. When he clicks on fund request he will be directed to donorhome.jsp page. The funder can click on the person he wants to donate and he will be redirected to payment.jsp page. In payment page the funder has to provide the account number and cvv number through which the funder has registered then they will be redirected to ft.jsp page. In this page the funder has to give some details in order to donate. All the details will be stored in Blockchain.

**Result:** All the test cases mentioned above passed successfully. No defects encountered

Now if the funder and fundraiser wants to check the transaction history they can click on fund history and they will be redirected to fundhistn.jsp page.

**Result:** All the test cases mentioned above passed successfully. No defects encountered

* 1. **TEST CASES AND REPORT**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **ACTION** | **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **TEST RESULT** |
| 1. | Enter the username, password, email, phone number for registration | Username: XXX  Password: \*\*\*  Email id: [abc@gmail.com](mailto:abc@gmail.com)  Phone number: 9876543210 | XXX  \*\*\*  [abc@gmail.com](mailto:abc@gmail.com)  9876543210 | XXX  \*\*\*  [abc@gmail.com](mailto:abc@gmail.com)  9876543210 | Passed |
| 2. | Compare username and password with registered field | Username: XXX  Password: \*\*\* | It redirects to charity home.jsp | It redirects to charity home.js | Passed |
| 3. | Compare username and password with registered field | Username: XXy  Password: \*\*\* | Invalid username and password | Invalid username and password | Passed |

**7.3.1 Test case report for Fundraiser registration**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **ACTION** | **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **TEST RESULT** |
| 1. | Funder should provide account number, amount, mail id | Account number: 789012345645  Amount:  10000  Mail id:  abc@gmail.com | These details will be stored in fund history | These details will be stored in fund history | Passed |
| 2. | Funder should provide account number, amount, mail id | Account number: 7890123456  Amount:  10000  Mail id:  abc@gmail.com | Please match the requested format for account number | Please match the requested format for account number | Passed |

**7.3.2 Test case report for Funder registration**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **ACTION** | **INPUT** | **EXPECTED OUTPUT** | **ACTUAL OUTPUT** | **TEST RESULT** |
| 1. | Fundraiser should provide name, mobile number, amount, description, account number | Name: john  Mobile Number: 9876543210  Amount:  15000  Description:  (reason for the fundraising )  Account number:  789012345673 | It will be posted in the service provided platform and it can be viewed by the funder and admin. | It will be posted in the service provided platform and it can be viewed by the funder and admin. | Passed |
| 2. | Fundraiser should provide name, mobile number, amount, description, account number | Name: john  Mobile number: 98765432  Amount:  15000  Description:  (reason for the fundraising )  Account number:  7890123444 | Please match the requested format for account number and mobile number | Please match the requested format for account number and mobile number | Passed |

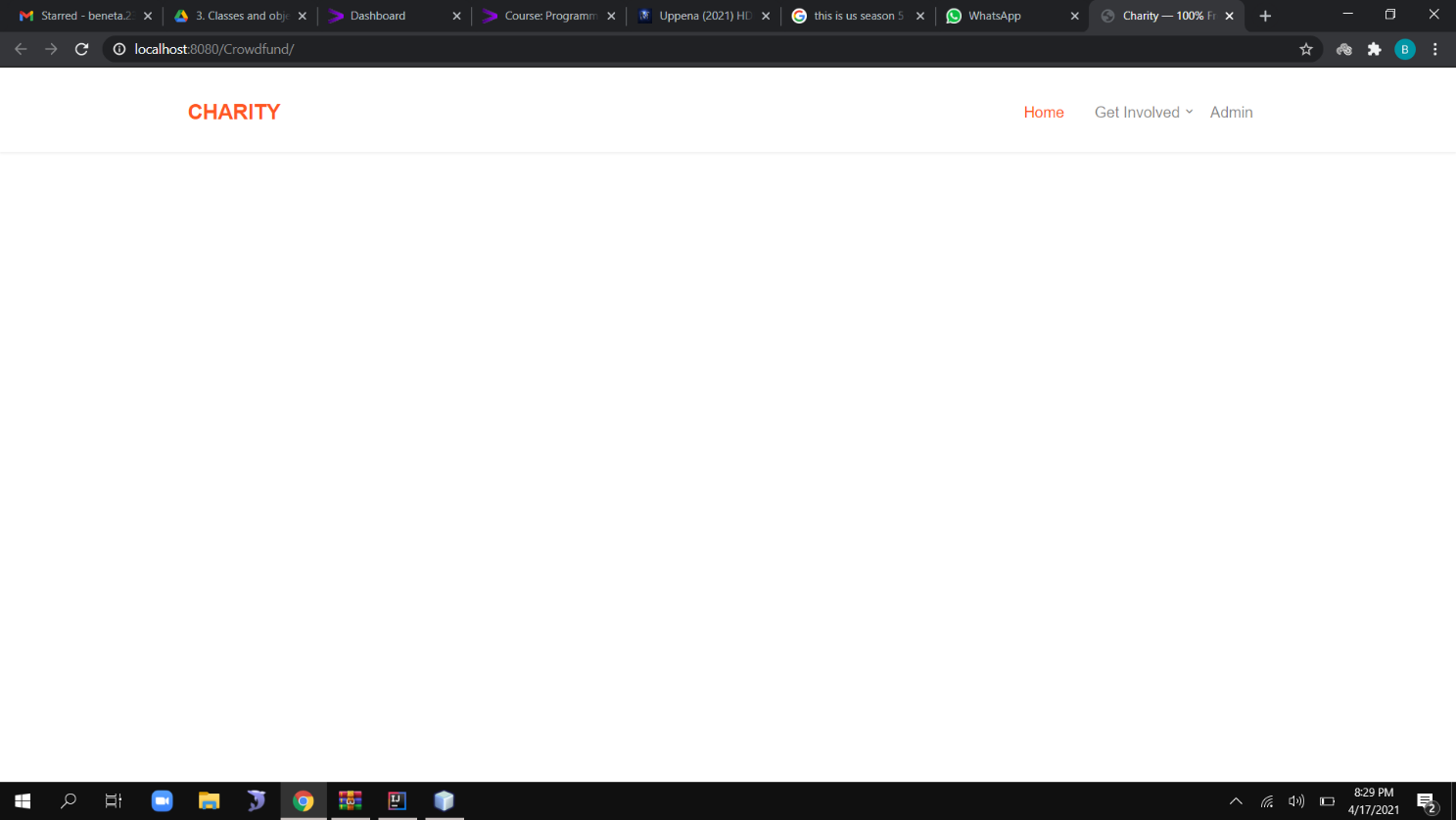
**7.3.3 Test case report for Post creation**

1. **CONCLUSION**
   1. **CONCLUSION AND FUTURE ENHANCEMENTS**

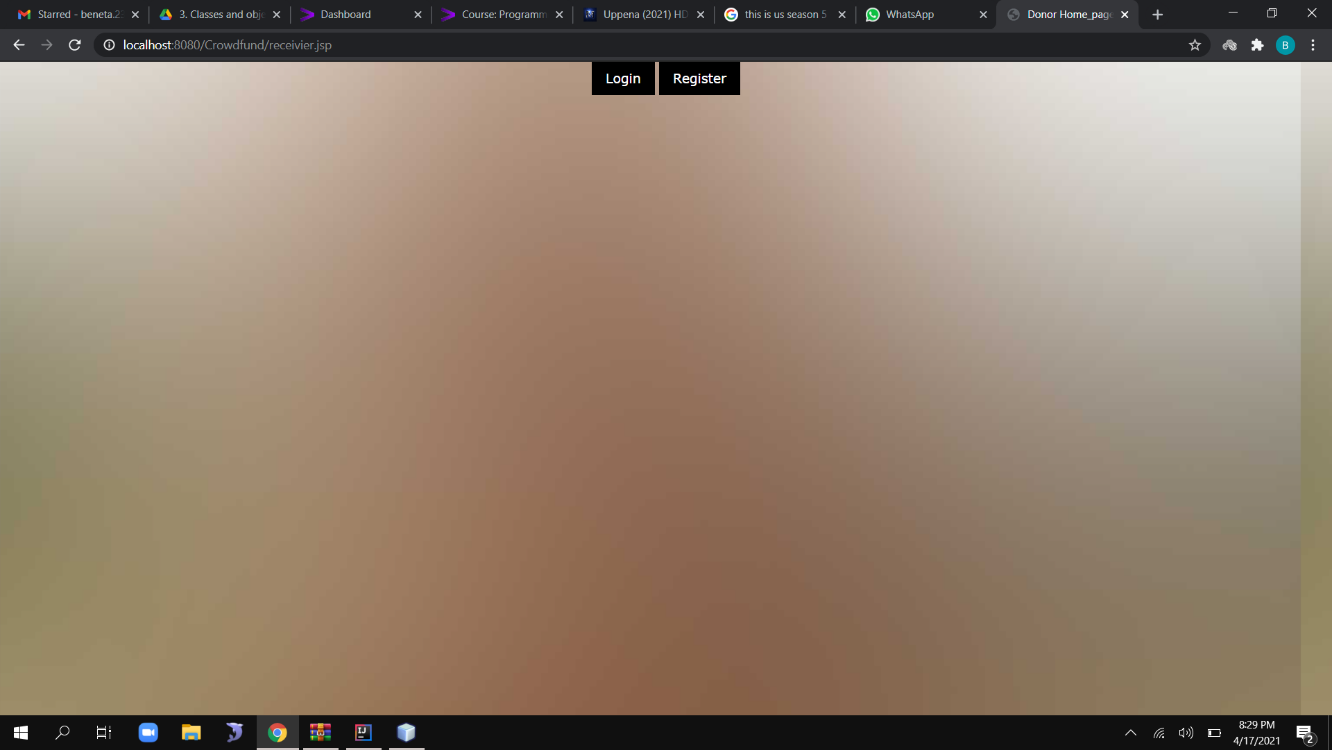
Finally, the disadvantages faced by the traditional crowdfunding has been overthrown with the help of blockchain technology in crowdfunding. An important factor for the people involved in raising these funds is trust. With the use of the blockchain concept in the crowdfunding, ensures trust of this crowdfunding platform and also reduces the cost of the third party. In future our work can be further enhanced by adding timestamp to the platform i.e., if the funder donates some amount to fundraiser and if the amount doesn’t reach the fundraiser within the timestamp specified, then the amount is returned back to the funder and fundraiser have to upload the file as a proof of their need. With the help of blockchain technology in the crowdfunding platform people trust the crowdfunding platform and be a part in this crowdfunding network.

**APPENDICES**

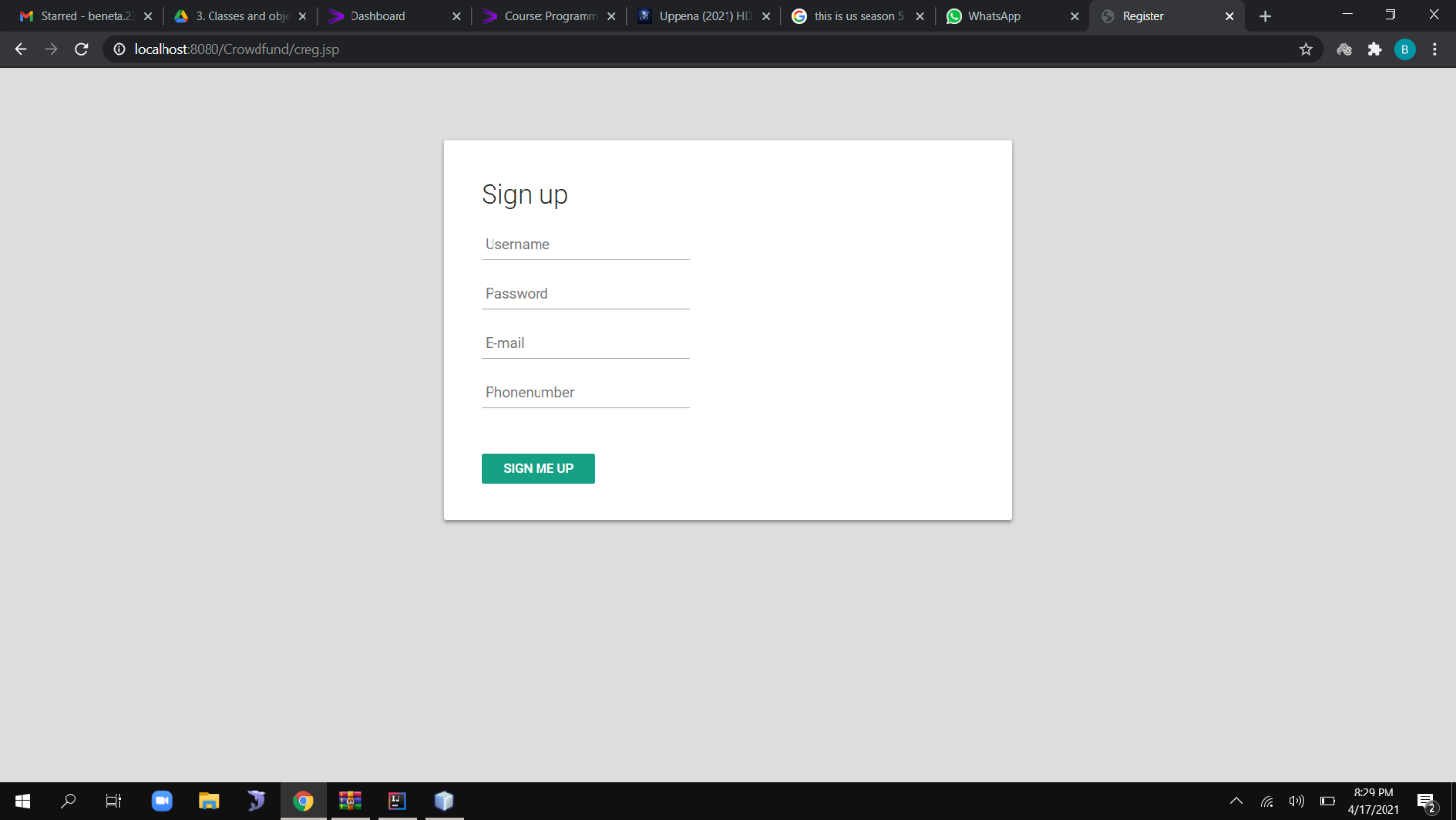
**A.1 SAMPLE SCREENS**

****

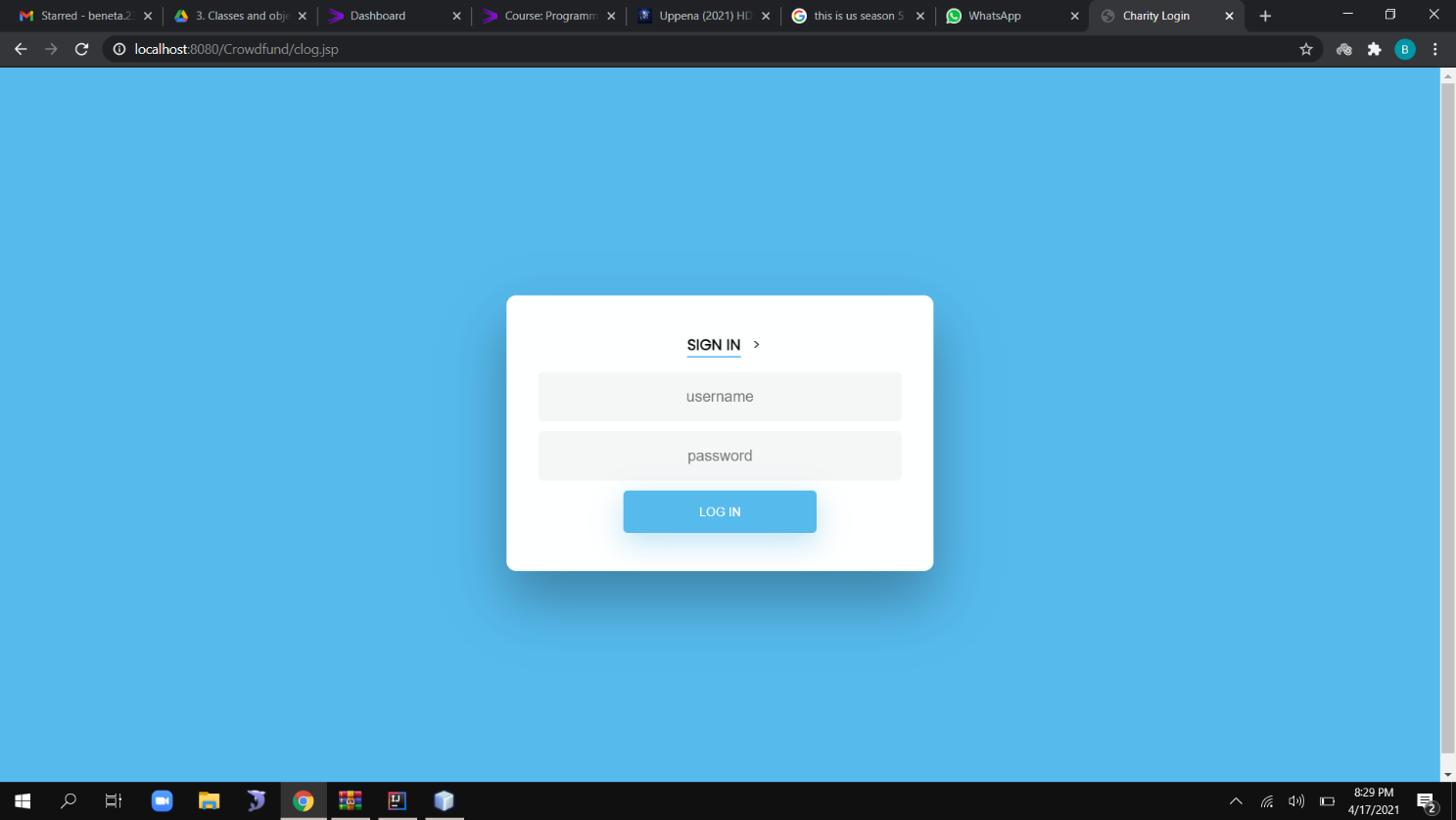
**Fig A.1.1 Home page**

****

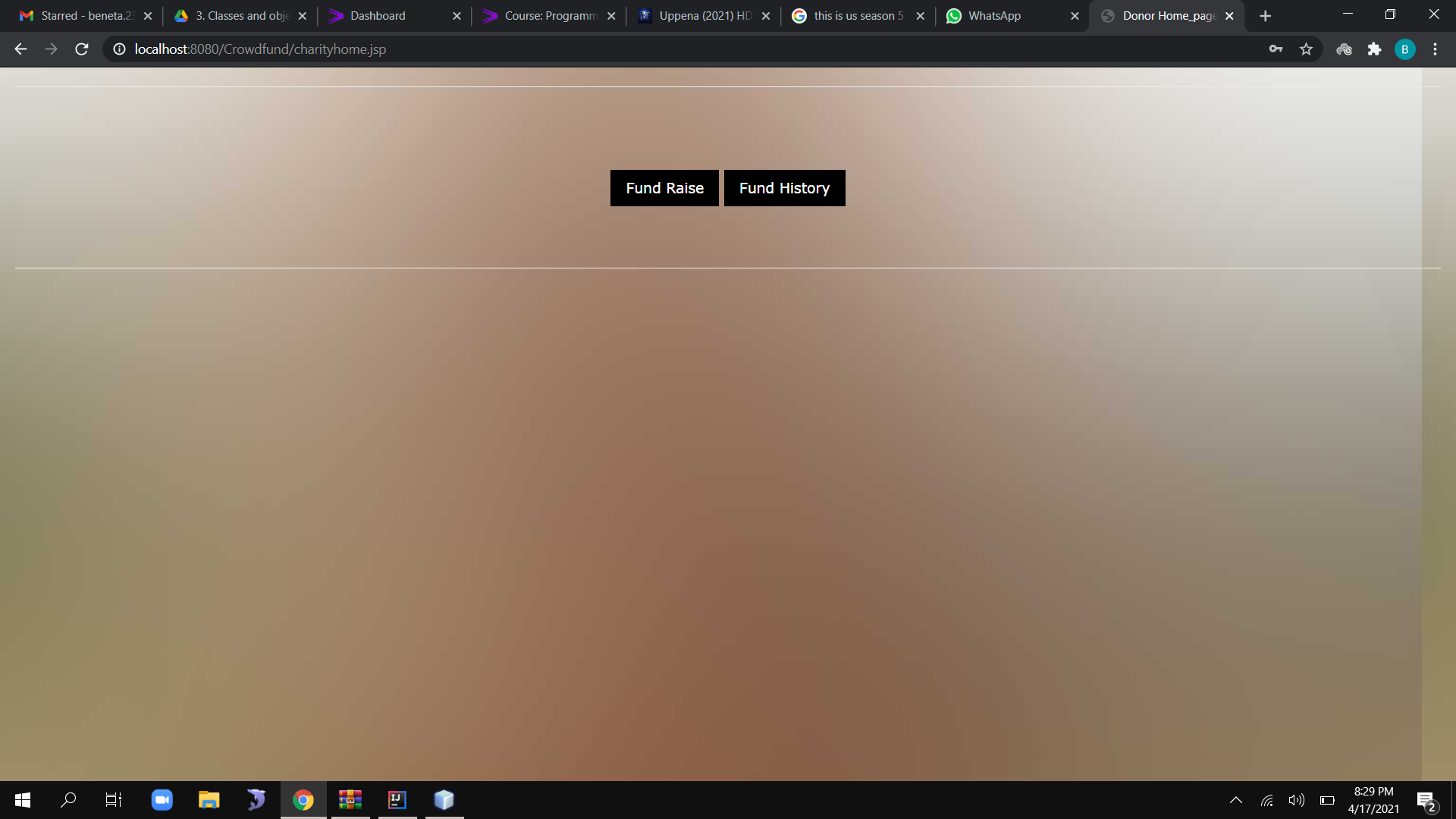
**Fig A.1.2 Fundraiser register and login page**

****

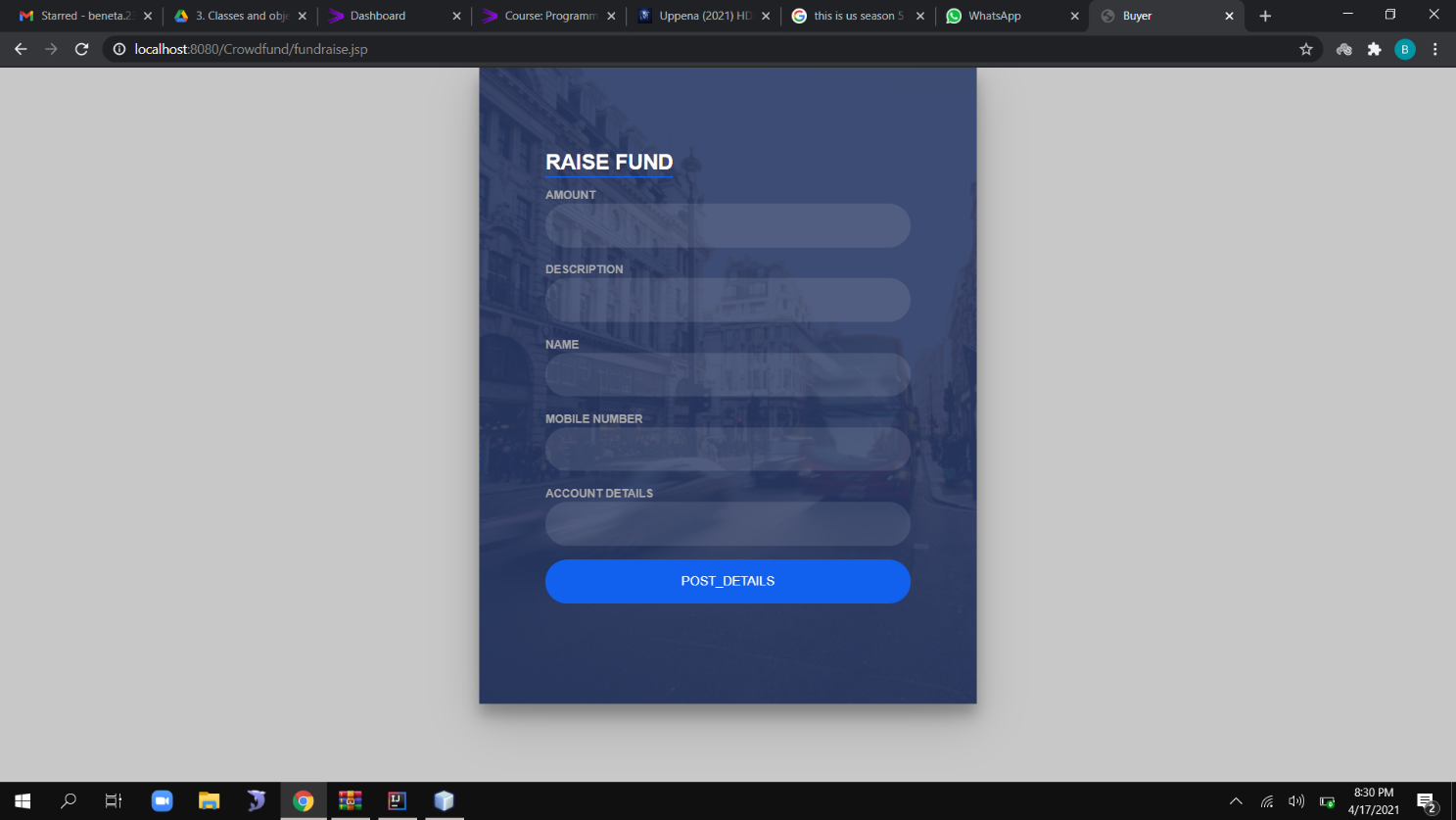
**A.1.3 Fundraiser register page**

****

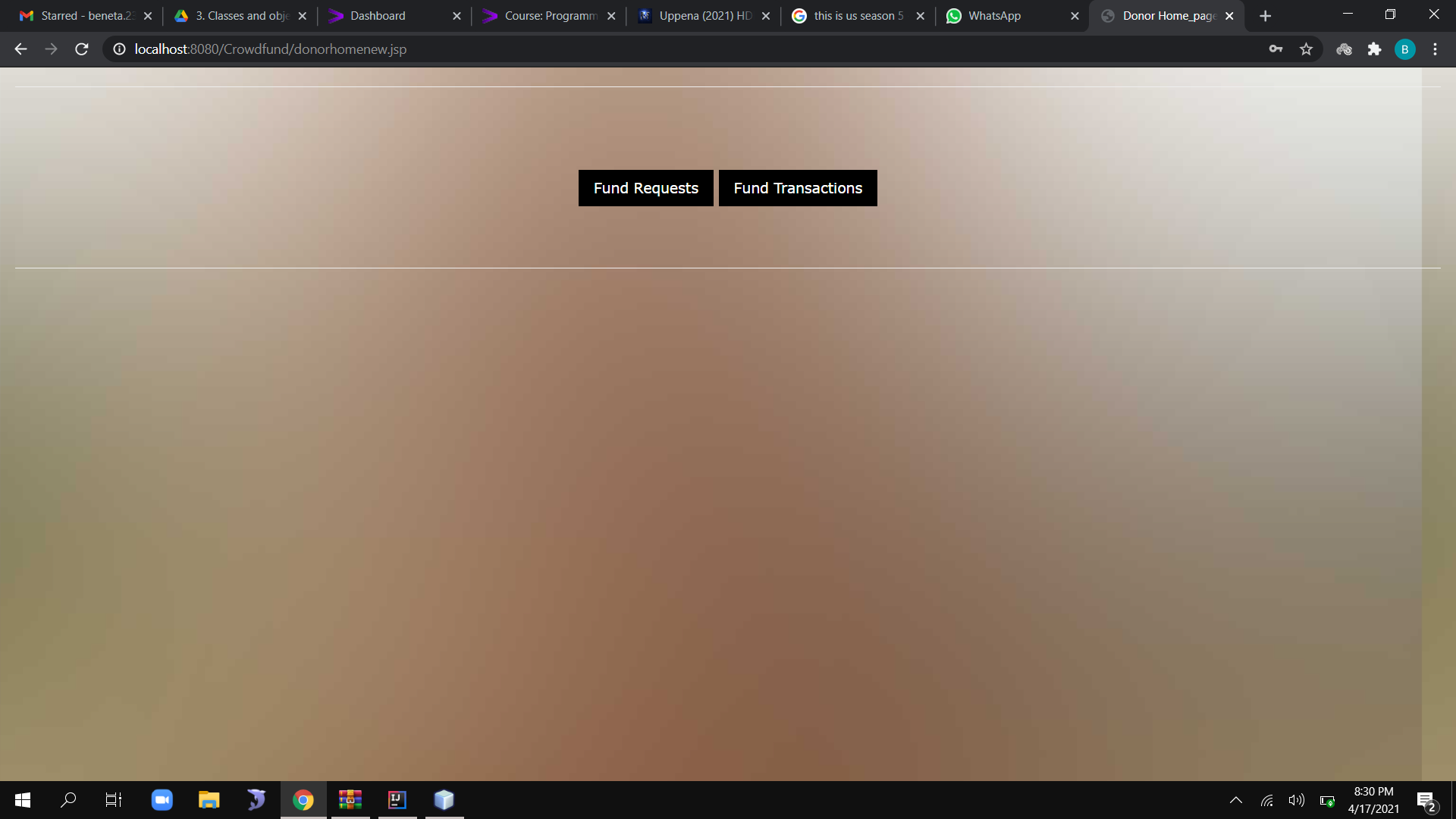
**A.1.4 Fundraiser login page**

****

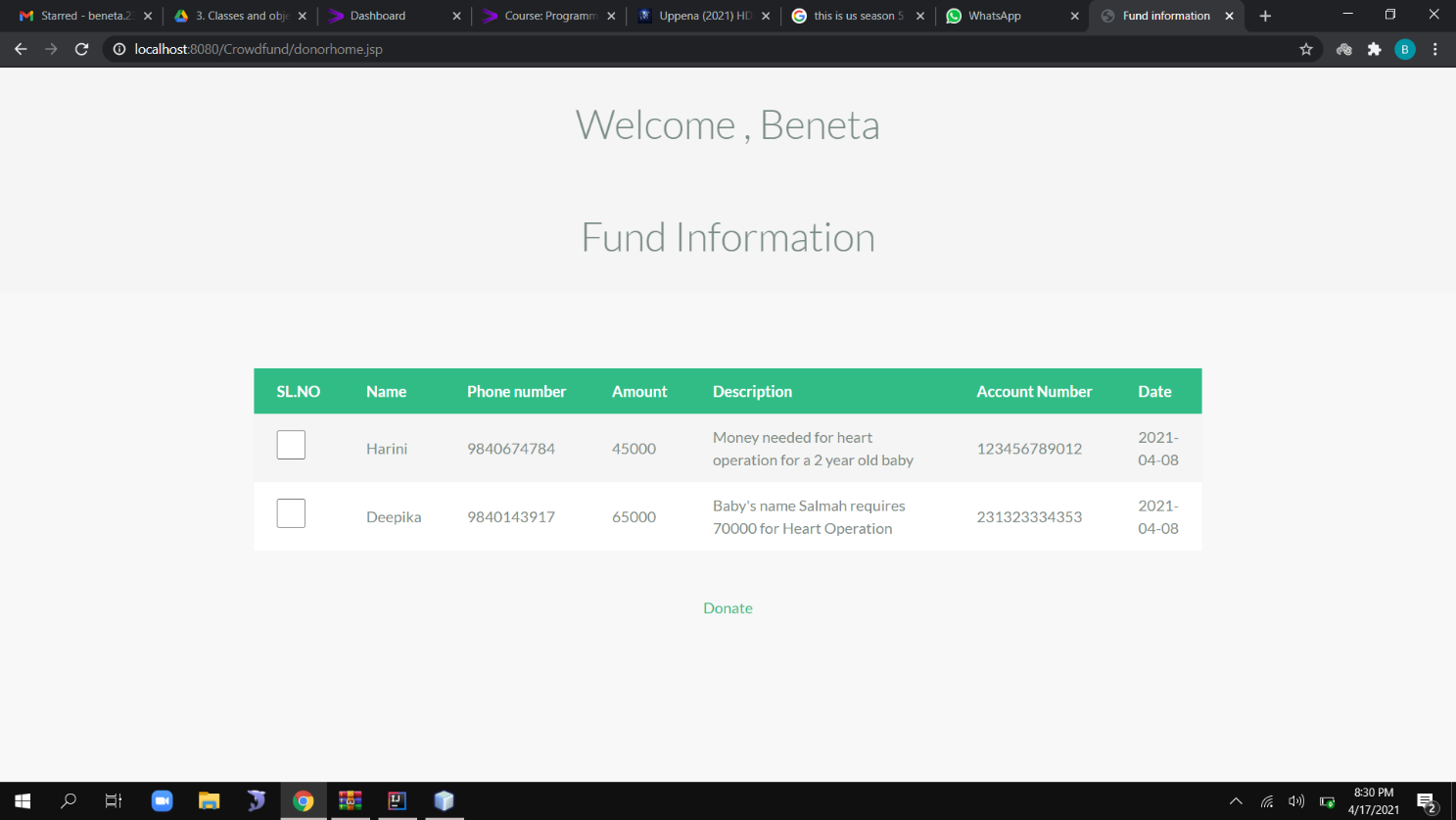
**A.1.5 Charity home page**

****

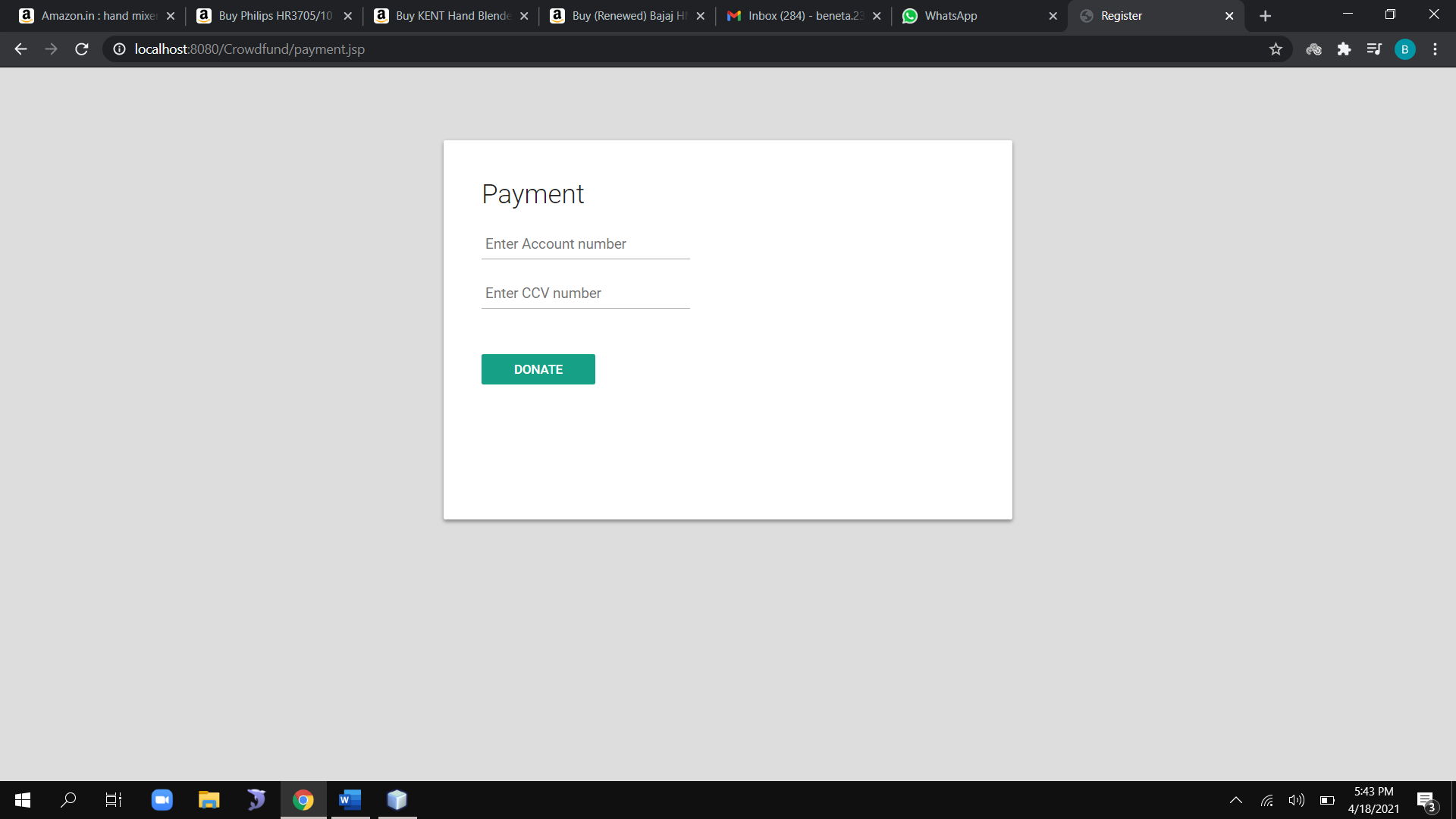
**A.1.6 Fund raise page**

****

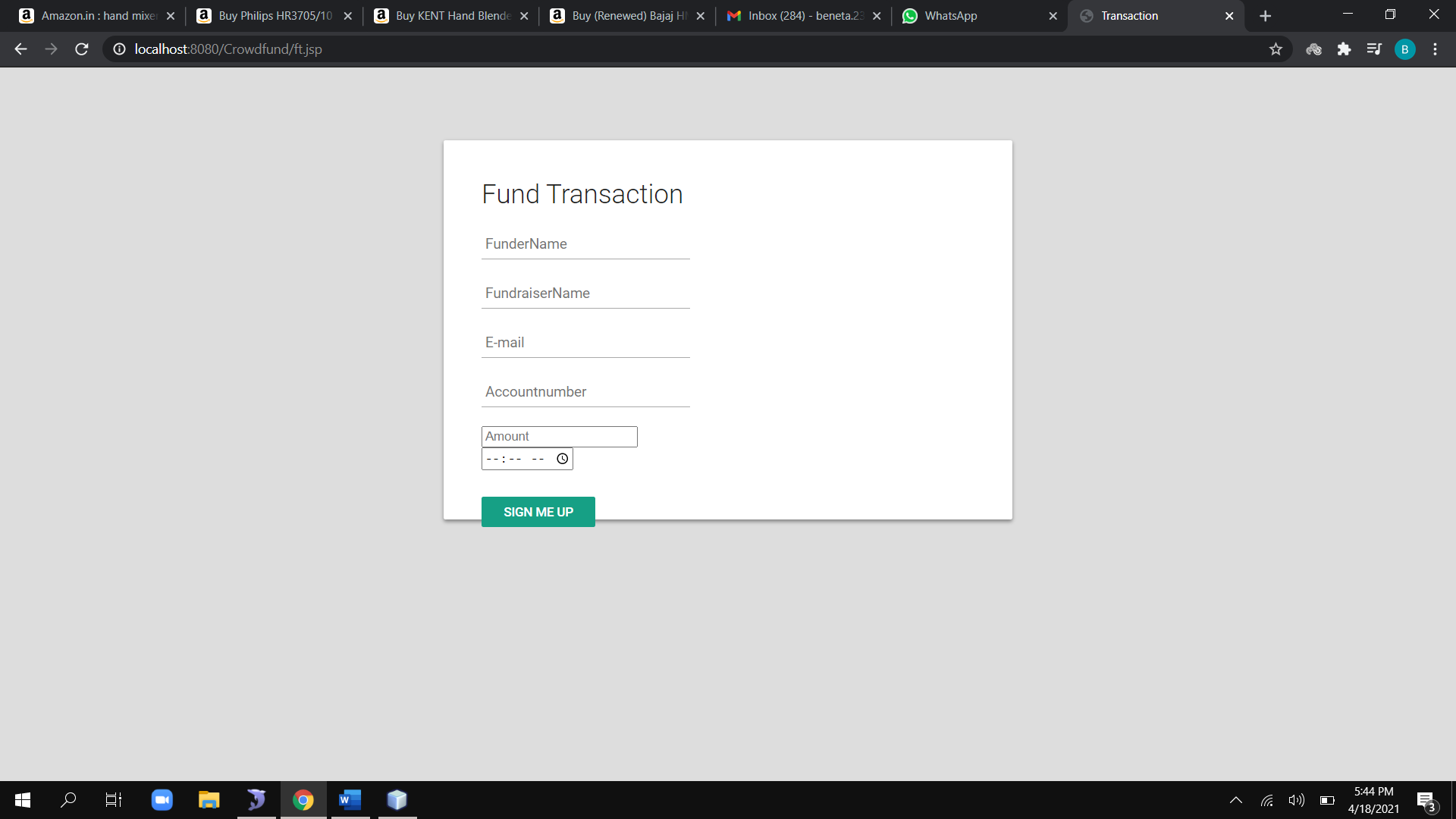
**A.1.7 Donor home page**

****

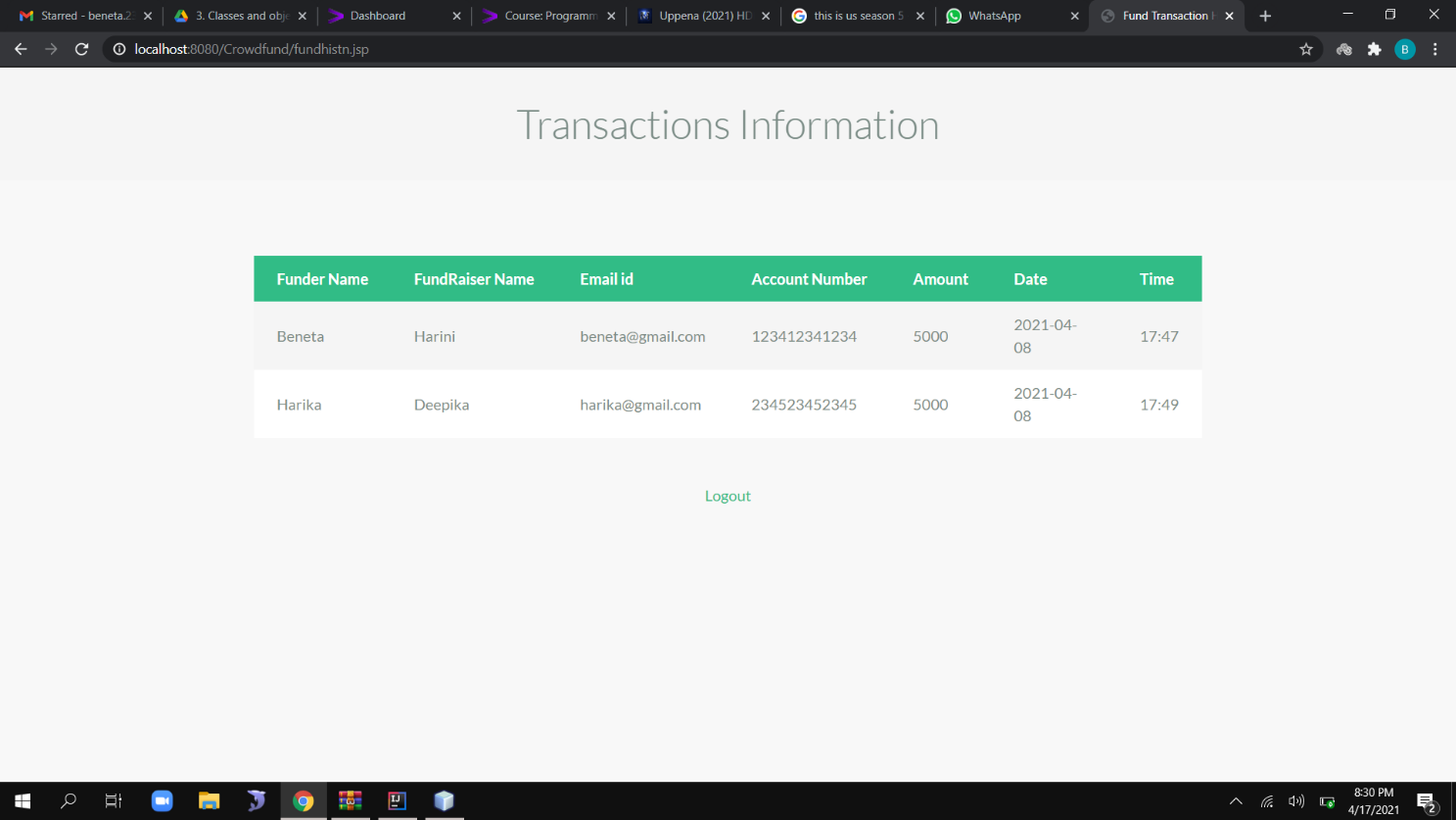
**A.1.8 Fund Information page**

****

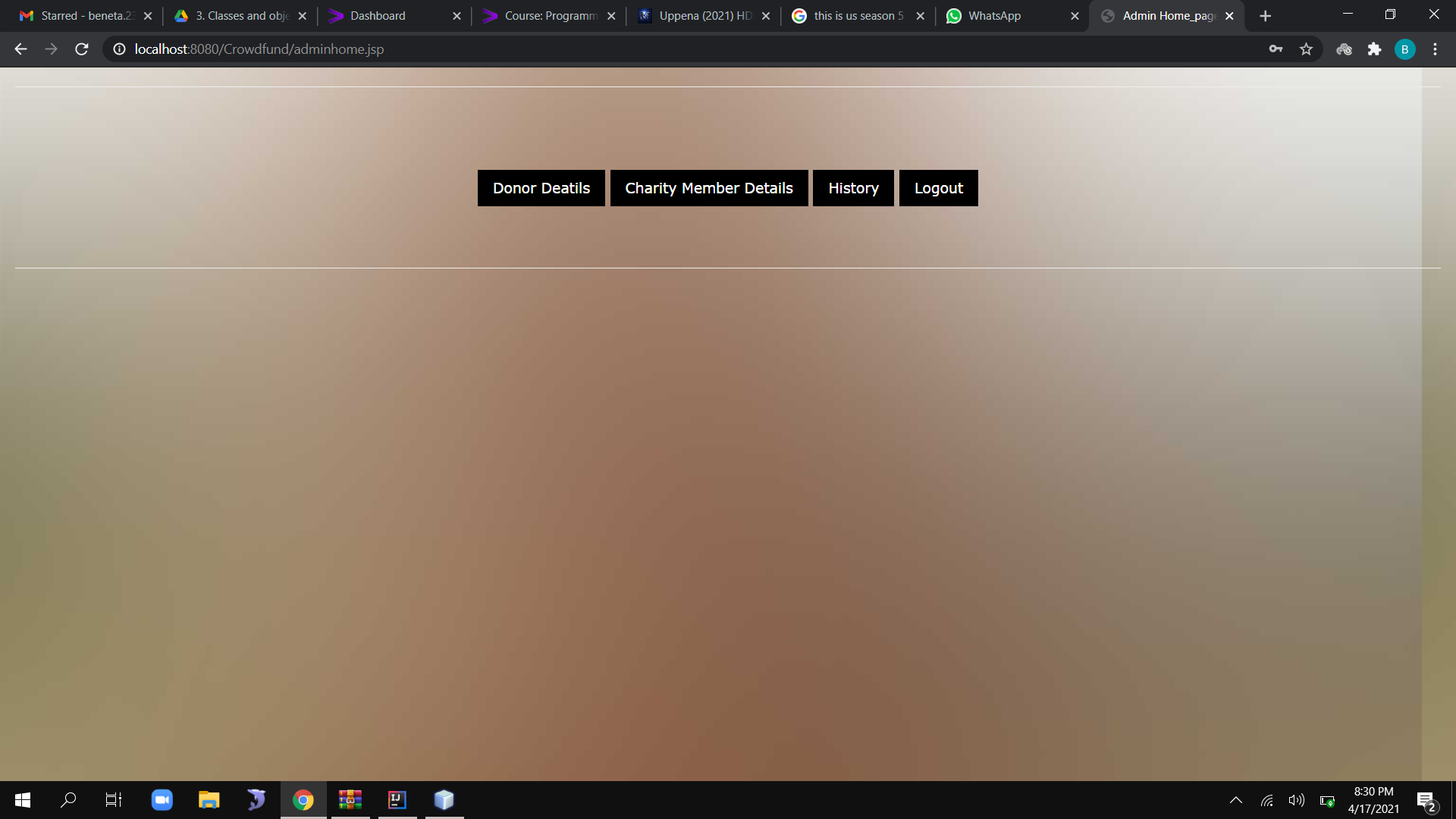
**A.1.9 Payment page**

****

**A.1.10 Fund transaction page**

****

**A.1.11 Transaction history page**

****

**A.1.12 Admin page**

**A.2 PUBLICATIONS**

BenetaGetzie M, Deepika T, Harini B, Anitha Moses V,   **"DECENTRALIZED FUNDING PLATFORM BASED ON BLOCKCHAIN TECHNOLOGY"**, International Journal of Creative Research Thoughts (IJCRT), ISSN:2320-2882, Volume.9, Issue 3, pp.4507-4511, March 2021.

**DOI Details**: <http://doi.one/10.1729/Journal.26363>

**REFERENCES:**

[1]. Hassija, Vikas, Chamola, Vinay & Zeadally Sherali. “BitFund: A Blockchain-based Crowd Funding Platform for Future Smart and Connected Nation”. Sustainable Cities and Society. 60. 102145. 10.1016/j.scs.2020.102145, May 2020.

[2]. Nikhil Yadav and Sarasvathi V. “Venturing Crowdfunding using Smart Contracts in Blockchain”. Third International Conference on Smart Systems and Inventive Technology (ICSSIT 2020)), IEEE Xplore Part Number: CFP20P17-ART; ISBN: 978-1-7281-5821-1, November 2020.

[3]. Ms. S. Benila, V. Ajay, K. Hrishikesh, R. Karthick. “Crowdfunding using Blockchain”. Global Research and Development Journal for Engineering, Volume 4, Issue 4, ISSN: 2455-5703, March 2019.

[4]. Hasnan Baber. “Blockchain-Based Crowdfunding: A ‘Pay-it-Forward’ Model of WHIRL”. International Journal of Recent Technology and Engineering (IJRTE),

ISSN: 2277-3878, Volume-8 Issue-3, September 2019.

[5]. Er. Waheeda Dhokley, Saurabh Gupta, Ganesh Pawar, Abrar Shaikh, “Crowdsourcing and Crowdfunding Platform using Blockchain and Collective Intelligence”.  International Journal of Computer Sciences and Engineering*, Vol.7*, Issue.2, pp.668-673, February 2019.

[6]. Firmansyah Ashari, Tetuko Catonsukmoro, Wilyu Mahendra Bad, Sfenranto, Gunawan Wang (2020). “Smart Contract and Blockchain for Crowdfunding Platform”. International Journal of Advanced Trends in Computer Science and Engineering. 9.3036-3041.10.30534/ijatcse/ 2020/83932020, June 2020.

[7]. André Schweizer, Vincent Schlatt, Nils Urbach, Gilbert Fridgen. “Unchaining Social Businesses - Blockchain as the Basic Technology of a Crowdlending Platform”. Proceedings of the 38th International Conference on Information Systems (ICIS), Seoul, South Korea, December 2017.

[8]. Zhu, Huasheng; Zhou, Zach Zhizhong (2016). “Analysis and outlook of applications of blockchain technology to equity crowdfunding in China”. Financial Innovation, ISSN 2199-4730, Springer, Heidelberg, Vol. 2, Iss. 29, pp. 1-11, 2016.

[9]. Zhao Hongjiang & Coffie Cephas. (2018). The Applications of Blockchain Technology in Crowdfunding Contract. SSRN Electronic Journal. 10.2139/ssrn.3133176, January 2018.

[10]. [Hissu Hyvärinen](https://link.springer.com/article/10.1007/s12599-017-0502-4#auth-Hissu-Hyv_rinen), [Marten Risius](https://link.springer.com/article/10.1007/s12599-017-0502-4#auth-Marten-Risius) & [Gustav Friis](https://link.springer.com/article/10.1007/s12599-017-0502-4#auth-Gustav-Friis). “A Blockchain-Based Approach Towards Overcoming Financial Fraud in Public Sector Services”. Bus Inf Syst Eng 59, 441–456, 2017.

[11]. Claudia Gabriela Bîzderea. "ANALYSIS OF FUNDING ALTERNATIVES THROUGH FINANCIAL TECHNOLOGY SERVICES – FINTECH – THE PHENOMENON OF CROWDFUNDING". Revista de Studii Financiare 3:109-127, November 2017.

[12]. Fridgen, Gilbert; Regner, Ferdinand; Schweizer, André; and Urbach, Nils, "DON’T SLIP ON THE ICO – A TAXONOMY FOR A BLOCKCHAIN-ENABLED FORM OF CROWDFUNDING" (2018). Research Papers. 83, November 2018.