JUSTCHAT

Project Report Submitted by

JUSTIN V KALAPPURA

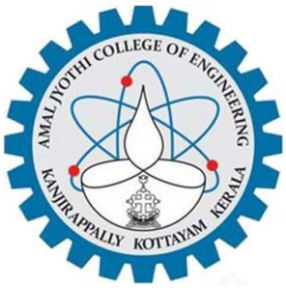
Reg. No: AJC21MCA-2068

In Partial fulfillment for the Award of the Degree of

MASTER OF COMPUTER APPLICATIONS

(MCA TWO YEAR)

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY



AMAL JYOTHI COLLEGE OF ENGINEERING
KANJIRAPPALLY

[Affiliated to APJ Abdul Kalam Technological University, Kerala. Approved by AICTE, Accredited by NAAC with 'A' grade. Koovappally, Kanjirappally, Kottayam, Kerala – 686518]

2021-2023

DEPARTMENT OF COMPUTER APPLICATIONS

AMAL JYOTHI COLLEGE OF ENGINEERING

KANJIRAPPALLY

CERTIFICATE

This is to certify that the Project report, **JUSTCHAT** is the bona fide work of **JUSTIN V KALAPPURA (Reg.no: AJC21MCA-2068)** in partial fulfillment of the requirements for the award of the Degree of Master of ComputerApplications under APJ Abdul Kalam Technological University during the year 2022-23.

Mr. Ajith G.S

Internal Guide

Ms. Meera Rose Mathew

Coordinator

Rev. Fr. Dr. Rubin Thottupurathu Jose

Head of the Department

External Examiner

AMAL JYOTHI
COLLEGE OF ENGINEERING

Department of Computer Applications

CERTIFICATE ON PLAGIARISM CHECK

1	Name of the Scholar	Justin V Kalappura
2	Title of the Publication	JUSTCHAT
3	Name of the Guide	Mr. Ajith G.S
4	Similar Content (%) identified	
5	Acceptable Maximum Limit	25%
6	Software Used	TURNITIN
8	No.of pages verified	

*Report on plagiarism check result with % of similarity shall be attached.

Name & Signature of the Scholar:

Name & Signature of the Guide:

Name & Signature of the Head of the Department:

Dept. Seal

DECLARATION

I hereby declare that the project report **JUSTCHAT** is a bona fide work done at Amal Jyothi College of Engineering, towards the partial fulfilment of the requirements for the award of the Master of Computer Applications (MCA) from APJ Abdul Kalam Technological University, during the academic year 2022-2023.

Date: **JUSTIN V KALAPPURA**

KANJIRAPPALLY **Reg.no: AJC21MCA-2068**

ACKNOWLEDGEMENT

First and foremost, I thank God Almighty for his eternal love and protection throughout the project. I take this opportunity to express my gratitude to all who helped me in completingthis project successfully. It has been said that gratitude is the memory of the heart. I wish to express my sincere gratitude to our manager **Rev. Fr. Dr. Mathew Paikatt** and Principal **Dr. Lillykutty Jacob** for providing good faculty for guidance.

I owe a gratitude towards our Head of the Department **Rev.Fr.Dr. Rubin Thottupurathu Jose** for helping us. I extend my whole hearted thanks to the project coordinator **Ms. Meera Rose Mathew** for her valuable suggestions and for overwhelming concern and guidance from the beginning to the end of the project. I would also express sincere gratitude to my guide **Mr. Ajith G.S** for his inspiration and helping hand.

I thank our beloved teachers for their cooperation and suggestions that helped me throughout the project. I express my thanks to all my friends and classmates for their interest, dedication, and encouragement shown towards the project. I convey my hearty thanks to my family for the moral support, suggestions, and encouragement to make this venture a success.

JUSTIN V KALAPPURA

ABSTRACT

The traditional news industry is facing challenges with declining revenues and increasing pressure to deliver quality content. Social media platforms have disrupted the traditional news model by offering a more engaging and personalized experience. However, social media platforms have also been criticized for promoting sensationalism and fake news. On the other hand, crowdfunding has emerged as a promising alternative funding model for independent journalism. In this project, we propose a novel approach to sustainable journalism by building a news platform that combines social media with crowdfunding functionality. Here news platform aims to provide a seamless user experience by integrating social media features such as sharing, commenting, and liking with crowdfunding functionality such as donations and subscriptions. The platform will allow users to interact with news content in a more engaging and participatory way, while also enabling them to support independent journalism financially. Here platform will also prioritize transparency and accountability by providing clear information about the sources of funding and the distribution of revenues. We will leverage the power of blockchain technology to ensure the security and transparency of transactions and to create a trust less environment. Here platform will also use machine learning algorithms to personalize content recommendations and to filter out fake news and clickbait. We will collaborate with journalists, media organizations, and crowdfunding experts to co-create the platform and to ensure that it meets the needs of both content creators and consumers. The potential impact of our news platform is significant. By combining social media with crowdfunding functionality, we aim to create a sustainable funding model for independent journalism and to promote the production of high-quality, reliable news content. The platform will also contribute to building a more informed and engaged society by providing a trustworthy source of news and promoting a culture of civic participation. The project is intended for anyone interested in the future of journalism, social media, and crowdfunding, including journalists, media organizations, technologists, and policymakers.

CONTENT

List of Abbreviation

IDE	- Integrated Development Environment
HTML	- Hyper Text Markup Language.
CSS	- Cascading Style Sheet
SASS	- Syntactically Awesome Style
SheetsSQL	- Structured Query
UML	- Unified Modeling Language

CHAPTER 1

INTRODUCTION

1.1 PROJECT OVERVIEW

The project 'JUSTCHAT' is defined as a social media website that aims to share our opinions, views, and outlooks in the public. This system can be used as a normal social media platform too. For the news head users, the administrator creates login id & password, using these the corresponding news head can access the system to add and update the news in the application. In brief, this website has third users i.e., normal user, admin, crowdfund user which allows correspondingly special credits to the users. Here the administrator will manage the accounts, and check the genuinely of the news uploaded by the news heads. And the normal users can share their view on them. And as a guest user they can just view everything. The social media application can be used to express our opinion on the present social issues which are presently existing in the society. And also in the present situation, social media usages are also very high. So, to make the society updated JUSTCHAT will be very much helpful.

1.2 PROJECT SPECIFICATION

In this proposed system, it is a normal social media with the functionality of crowdfunding added to it, which too is implemented in blockchain. In this we can perform the normal basics functionality like in a social media and in addition to that have a crowdfunding functionality added to it. And also, it is very much necessary for the project. The major users added to this project are listed below:

ADMIN:

This module has overall control of this website. All the details of the News head are added by the admin. Admin can handle news head and users. And also, can view the whole system. If needed can block or delete the corresponding users or their details. And checking will be going under a continuous suspects checking.

NORMAL USERS:

This module has the control over the user's panel. In this they can view the whole feed by other users and news updates by the corresponding news teams. They too can post the feed to the platform and chat with other users using our platform.

CROWDFUND USERS:

This module is purely a web3 implemented system. In this all the users are allowed to use this particular feature of the social media. But can do any transaction only with having separate account in MetaMask or only with a wallet with corresponding blockchain coins. They need the corresponding coins in their wallet for the transactions that should be made in the whole blockchain module of crowdfund. This is the first time that we implement a web3 module in to a web 2 module. Here, the web 3 module used is done in the blockchain technology and the frontend for this module is designed in the react language.

CHAPTER 2
SYSTEM STUDY

2.1 INTRODUCTION

A social network service which focuses on the building and verifying of online social networks on which we can personally share their own interests on common social issues, which will can be seen by all the users of this platform. And also, they can be updated with the recent headlines of the day. This application will be primarily web-based application and provide a collection of various ways for users to interact. The traditional news industry is facing challenges with declining revenues and increasing pressure to deliver quality content. Social media platforms have disrupted the traditional news model by offering a more engaging and personalized experience.

2.2 EXISTING SYSTEM

Social networks, which have almost become part of our daily lives, have established new communication structures and behaviors in society. While citizens and businesses have already extensively used social networks for years, governments continuously increase their interest in the new communication technologies. Sites such as Facebook, Twitter, and Linked In provide a mechanism for individuals to come together based on a variety of factors such as existing friendships, common interests, or work.

2.3 DRAWBACKS OF EXISTING SYSTEM

The main purpose of the system is to replace the existing manual system. Some of the main problems encountered in existing system are follows.

- Making them less aware of the things happening around.
- Making people living in a virtual world by not having any opinion.
- Pressurizing the new generation to follow the old methods.
- Not having a good transparent system for helping others financially.

2.4 PROPOSED SYSTEM

As already mentioned, the existing social media applications are mostly and exclusively meant for either entertainment that makes the users mainly not updated. And also in the present situation, social media usages are also very high. This particular problem is being solved in our proposed social media application in which you can express our opinion on the present social issues openly. And also in the present situation, social media usages are also very high.

2.5 ADVANTAGES OF PROPOSED SYSTEM

- Make new generation aware about the things happening around.
- Promotes the people having their own opinion.
- Along with the entertainment can also make them updated.
- Good feature to have a transparent crowdfunding.

CHAPTER 3
REQUIREMENT ANALYSIS

3.1 FEASIBILITY STUDY

Planning, organizing, and managing resources to ensure the achievement of particular project goals and objectives is the process of project management. A feasibility study is a preliminary examination of a prospective project or end to determine its merits and viability. A feasibility study aims to provide an objective assessment of the technical, economic, financial, legal, and environmental elements of a proposed project. The information can then be used by decision-makers to decide whether to proceed with the project or not. The findings of the feasibility study can also be used to develop a practical project plan and budget. It cannot be simple to determine whether or not a proposed project is worthwhile pursuing without a feasibility study. Various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic and Operational feasibility.

3.1.1 Economical Feasibility

Cost and benefit analyses are required to support the developing system. criteria to make sure that focus is on the project that will yield the best results and return the earliest. The price that would be involved in developing a new system is one of the variables. Some significant financial queries raised during the initial investigation include the following:

- The costs conduct a full system investigation?
  - The proposed system is developed as part of project work, there is no manual cost to spend for the making.
- The cost of the hardware and software?
  - Also, all the resources are already available.

3.1.2 Technical Feasibility

Technical feasibility refers to the assessment of whether a proposed project or solution can be successfully implemented from a technological standpoint. The system needs to be assessed first from a technical standpoint. The outline design of the system requirement in terms of input, output, programs, and procedures must serve as the foundation for the assessment of this feasibility. After determining an outline investigation must continue to identify the necessary equipment kind. Once the system has been designed, there are several ways to run it.

- Is the project feasible within the limits of current technology.
  - YES
- Technical issues raised during the investigation are:
  - NOTHING
- Can the technology be easily applied to current problems?
  - YES
- Does the technology have the capacity to handle the solution?
  - YES

3.1.3 Behavioral Feasibility

A behavioral study for social media involves examining and analyzing user behavior, interactions, and patterns within a social media platform or across multiple platforms. It aims to understand how individuals engage with social media, their motivations, preferences, and the impact of their behaviors on the platform and its ecosystem. The study provides valuable insights for social media companies, marketers, and researchers to optimize user experiences of the users. The proposed system includes the following questions:

- Is there sufficient support for the users?
  - YES
- Will the proposed system cause harm?
  - NO

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible

3.1.4 Feasibility Study Questionnaire

1.To what extend the system is proposed for?

The social media application can be a used to express our opinion on the present social issues which are presently existing in the society. And also in the present situation, social media usages are also very high. So, in order to use the social media usage time more fruitful we introduce a session to get updated.

2.Specify the Viewers/Public which is to be involved in the System?

The General public are involved in the system.

3.Identify the users in your project?

The users are the General public and the agents.

4.Who owns the system?

The Administrator who owns the social media company.

5.Is it necessary to build automated system?

Yes, Automated system helps us to reduce paper documentations.

6.What is the important aim of the system?

To reduce paper works and all jobs done by automated.

7.Do you provide encryption for security?

Yes.

8.Is it a government-organized system?

No, It’s a privately organized system.

9.Is this system is scalable or not?

This system can able to scale.

10.What is the possibility of the system in the present scenario?

Very much useful. As the new generation is so efficient in using the online systems.

3.2 SYSTEM SPECIFICATION

3.2.1 Hardware Specification

Processor - Intel core i5

RAM - 8 GB

Hard disk - 512 SSD

3.2.2 Software Specification

Front End - HTML, CSS, React

Back End - POSTGRE SQL,

Client on PC - Windows 10 and above.

Technologies used - JAVA SCRIPT, jQuery, PYTHON, DJANGO, BLOCKCHAIN

3.3 SOFTWARE DESCRIPTION

3.3.1 PYTHON

Python is a high-level programming language that is frequently used to create a wide variety of software applications, from web development and data analysis to scientific computing and artificial intelligence. Python is a well-liked option for project development because of its simplicity, usability, and versatility. Python is the perfect language for project development because of its enormous standard library and ecosystem of third-party libraries, tools, and frameworks. Python's syntax is simple to understand and read, and because its code is frequently shorter than that of other programming languages, projects can be developed more quickly. Python also provides a wide range of debugging, profiling, and testing tools that make it simpler for developers to find and fix problems rapidly.

3.3.2 MySQL

MySQL is an open-source relational database management system (RDBMS) that allows users to store, organize, and manage structured data. It is widely used for a range of applications, from small websites to large-scale enterprise systems. With its emphasis on performance, scalability, and ease of use, MySQL has become one of the most popular databases in the world. It supports SQL for data manipulation and retrieval, offers robust security features, and provides options for replication and high availability. It is frequently referred to as the most popular database. MySQL's active open-source community ensures continuous development, support, and a wealth of resources, making it a reliable and flexible choice for developers and organizations alike. MySQL is a relational database management system (RDBMS) built on the structured query language (SQL) and created by Oracle. The project makes use of the WAMP (Windows, Apache, MySQL, and PHP) server stack, which implements the MySQL database management system. This technological stack offers a local server environment for creating and testing PHP and MySQL web applications.

3.3.3 DJANGO

Django is a high-level Python web framework that simplifies and accelerates web development. It provides a robust set of tools, libraries, and patterns for building secure, scalable, and maintainable web applications. Django follows the Model-View-Controller (MVC) architectural pattern, emphasizing the separation of concerns and promoting code reusability. One of Django's key features is its powerful Object-Relational Mapping (ORM) layer, which abstracts the database access and allows developers to interact with the database using Python objects and methods. This makes it easier to work with databases and perform common operations without writing complex SQL queries. Django also includes a built-in administration interface that automatically generates forms, views, and CRUD (Create, Read, Update, Delete) functionality for database models. This feature accelerates the development process by providing a ready-to-use backend for managing data. Additionally, Django promotes the use of reusable components through its application structure. Developers can create modular and pluggable apps, which can be easily integrated into multiple projects, reducing code duplication and increasing productivity.

3.3.4 BLOCKCHAIN

Blockchain is a decentralized and immutable digital ledger that records transactions across multiple computers or nodes. It enables secure and transparent peer-to-peer transactions without the need for intermediaries like banks or governments. Blockchain technology is based on cryptographic principles and consensus algorithms that ensure the integrity and trustworthiness of the recorded data. The fundamental concept of blockchain revolves around the creation of blocks, each containing a set of transactions. These blocks are linked together in a chronological chain, forming a permanent and unalterable record of all transactions. The distributed nature of blockchain ensures that every participant has a copy of the entire ledger, making it resistant to tampering and single points of failure. Blockchain has gained significant attention due to its potential applications beyond cryptocurrencies. It can be used for various purposes, such as supply chain management, healthcare records, voting systems, smart contracts, and decentralized finance. Its transparency, security, and decentralization characteristics offer increased efficiency, reduced costs, and enhanced trust in various industries. Blockchain also offers increased efficiency by eliminating the need for intermediaries or centralized authorities. This reduces costs, speeds up processes, and minimizes the potential for human error or manipulation.

Beyond its association with cryptocurrencies, blockchain has a wide range of applications. It can be utilized in supply chain management to track and authenticate products, in healthcare to securely store and share patient data, in voting systems to enhance transparency and eliminate voter fraud, and in energy grids to facilitate peer-to-peer energy trading, among many other possibilities.

CHAPTER 4
SYSTEM DESIGN

4.1 INTRODUCTION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to effective system. The term “design” is defined as “the process of applying various techniques and principles for the purpose of defining a process or a system in sufficient detail to permit its physical realization”. It may be defined as a process of applying various techniques and principles for the purpose of defining a device, a process or a system in sufficient detail to permit its physical realization. Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm that is used. The system design develops the architectural detail required to build a system or product. As in the case of any systematic approach, this software too has undergone the best possible design phase fine tuning all efficiency, performance and accuracy levels. The design phase is a transition from a user-oriented document to a document to the programmers or database personnel. System design goes through two phases of development: Logical and Physical Design.

4.2 UML DIAGRAM

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design. All the elements, relationships are used to make a complete UML diagram and the diagram represents a system. The visual effect of the UML diagram is the most important part of the entire process. All the other elements are used to make it complete. They enhance communication by providing a visual representation that is easily understood by technical and non-technical stakeholders alike. They aid in analysis and design by capturing system requirements, relationships, and interactions. UML diagrams also facilitate software development by serving as a blueprint for developers to implement and test systems.

UML includes the following nine diagrams.

- Use case diagram
- Sequence diagram
- State chart diagram
- Activity diagram
- Class diagram
- Object diagram
- Component diagram
- Deployment diagram

4.2.1 USE CASE DIAGRAM

A use case diagram is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements. In this context, the term "system" refers to something being developed or operated, such as a mail-order product sales and service Web site. Use case diagrams are employed in UML (Unified Modeling Language), a standard notation for the modeling of real-world objects and systems. System objectives can include planning overall requirements, validating a hardware design, testing and debugging a software product under development, creating an online help reference, or performing a consumer-service-oriented task. For example, use cases in a product sales environment would include item ordering, catalog updating, payment processing, and customer relations. A use case diagram contains four components.

- The boundary, which defines the system of interest in relation to the world around it.
- The actors, usually individuals involved with the system defined according to their roles.
- The use cases, which are the specific roles are played by the actors within and around the system.
- The relationships between and among the actors and the use cases.

Fig 1: Use case diagram for ‘JUSTCHAT ‘

4.2.2 SEQUENCE DIAGRAM

A sequence diagram m simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event diagrams or event scenarios to refer to a sequence diagram. Sequence diagrams describe how and in what order the objects in a system function. These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems. In a sequence diagram, objects are represented as vertical lifelines, and messages are depicted as horizontal arrows between the lifelines. The arrows indicate the flow of communication, including method calls, responses, and signals. The sequence of messages helps to depict the dynamic behavior of the system and how different objects collaborate to achieve specific functionality. Sequence diagrams are useful for understanding and analyzing the interactions within a system, especially in scenarios where multiple objects or actors are involved. They allow software developers and designers to visualize the chronological order of events, identify potential bottlenecks or issues, and ensure that the system behaves as intended. Overall, sequence diagrams are a valuable tool in software development for modeling and understanding the dynamic behavior of systems, enabling efficient design, analysis, and communication throughout the development process.

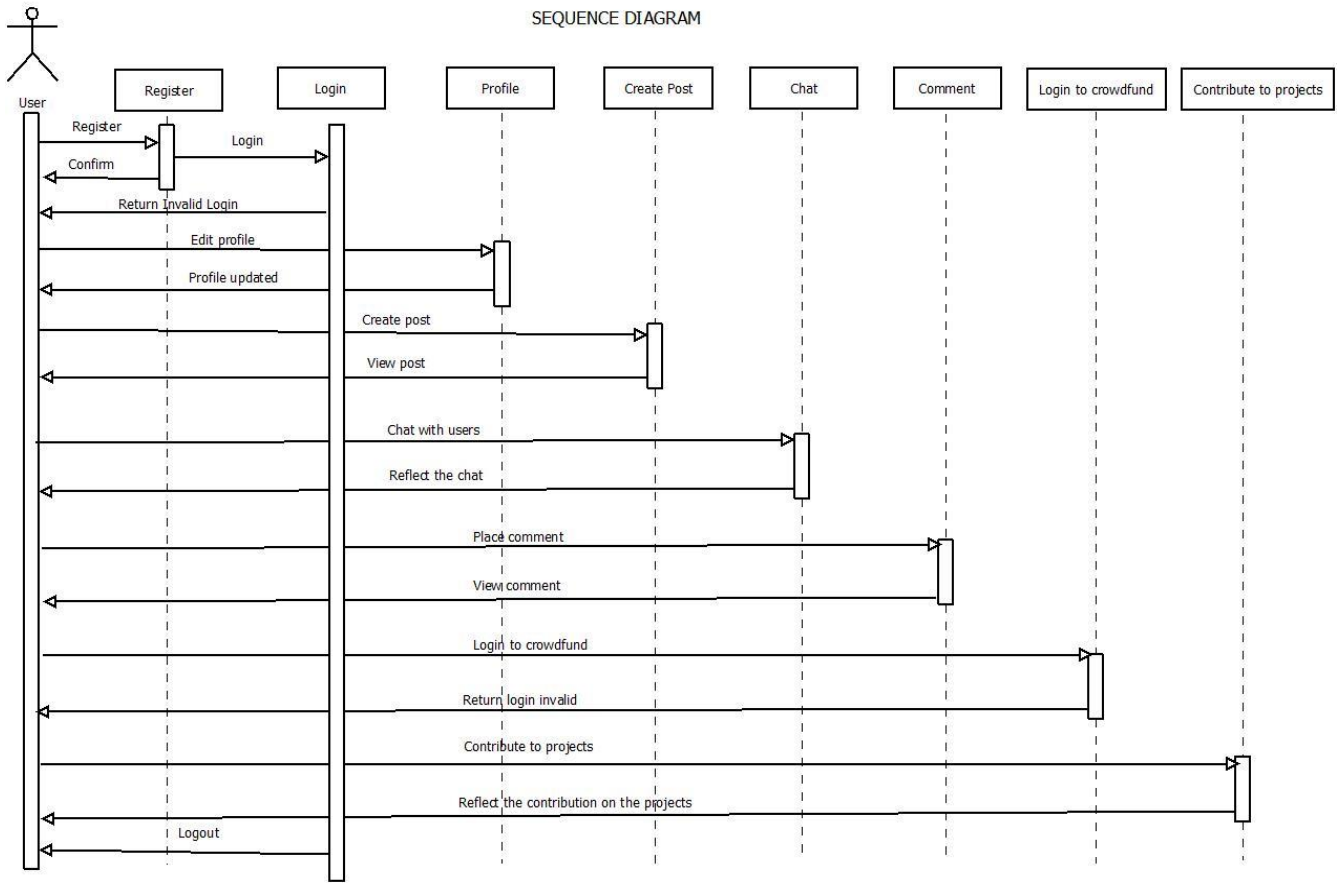


Fig 2: Sequence diagram for 'JUSTCHAT'

4.2.3 State Chart Diagram

A State Chart Diagram is a type of UML (Unified Modeling Language) diagram used to model the behavior of a system or object over time. A State Chart Diagram shows the various states that an object or system can be in, and how it transitions from one state to another. Each state represents a particular condition or mode that the object or system can be in, and may have associated actionsor behaviors. Transitions between states are triggered by events, which may be external (such as a user input) or internal (such as a timer).A transition may also have a guard condition, which must be true for the transition to occur. State Chart Diagrams can be used to model complex systems with many states and transitions, such as user interfaces, control systems, or business processes. State Chart Diagrams can also show hierarchical states, where a state may contain sub-states with their own transitions and behaviors. State Chart Diagrams are often used in conjunction with other UML diagrams, such as Use Case Diagrams or Class Diagrams. State Chart Diagrams can be usedto communicate system behavior to stakeholders, including developers, testers, and users. State Chart Diagrams can be implemented using various programming languages and tools, such as finite state machines or event-driven programming. State Chart Diagrams can help improve system reliability and maintainability by providing a clear and concise model of system behavior that canbe used for testing, debugging, and maintenance. In a state chart diagram, states are represented as rounded rectangles, and transitions between states are depicted as arrows. Events trigger these transitions, causing the system or object to move from one state to another. Actions, conditions, and triggers associated with each transition can also be specified.

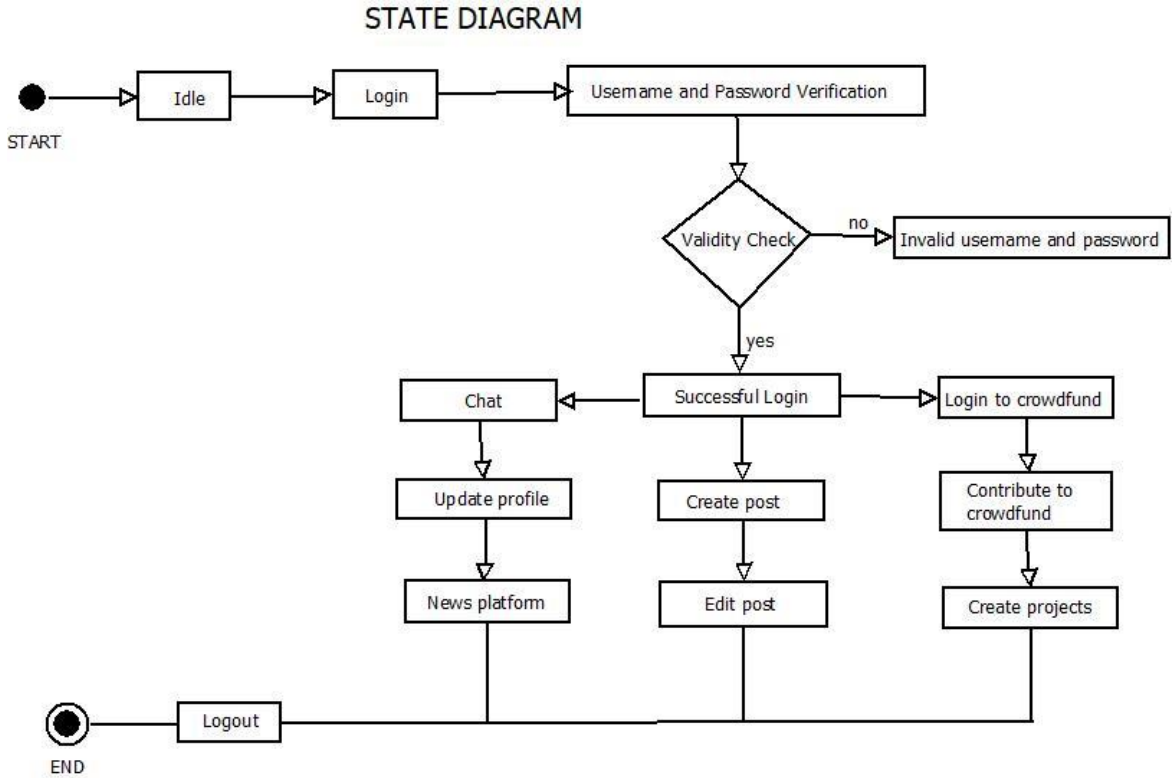


Fig 3: State chart diagram for 'JUSTCHAT'

4.2.4 Activity Diagram

Activity diagrams depict how different levels of abstraction of activities are linked to provide a service. Typically, an event should be completed by some activities, particularly when the activity is intended to do multiple separate goals that need coordination. Another typical requirement is how the events in a single use case interact with one another, particularly in use cases where operations may overlap and require coordination. It may also be used to show how a collection of interrelated use cases interacts to reflect business operations. In an activity diagram, activities are represented as rounded rectangles, and arrows indicate the flow of control between activities. Decision points are depicted as diamonds, where different paths or branches can be taken based on conditions or criteria. Forks and joins represent parallel or concurrent execution of activities. Swim lanes can also be used to group related activities performed by specific actors or roles. Activity diagrams are valuable for modelling and analysing complex processes or workflows. They help to visualize the sequence of steps, dependencies, and decision points within a system, providing a clear understanding of how the process operates. This aids in identifying potential bottlenecks, inefficiencies, or areas for optimization.

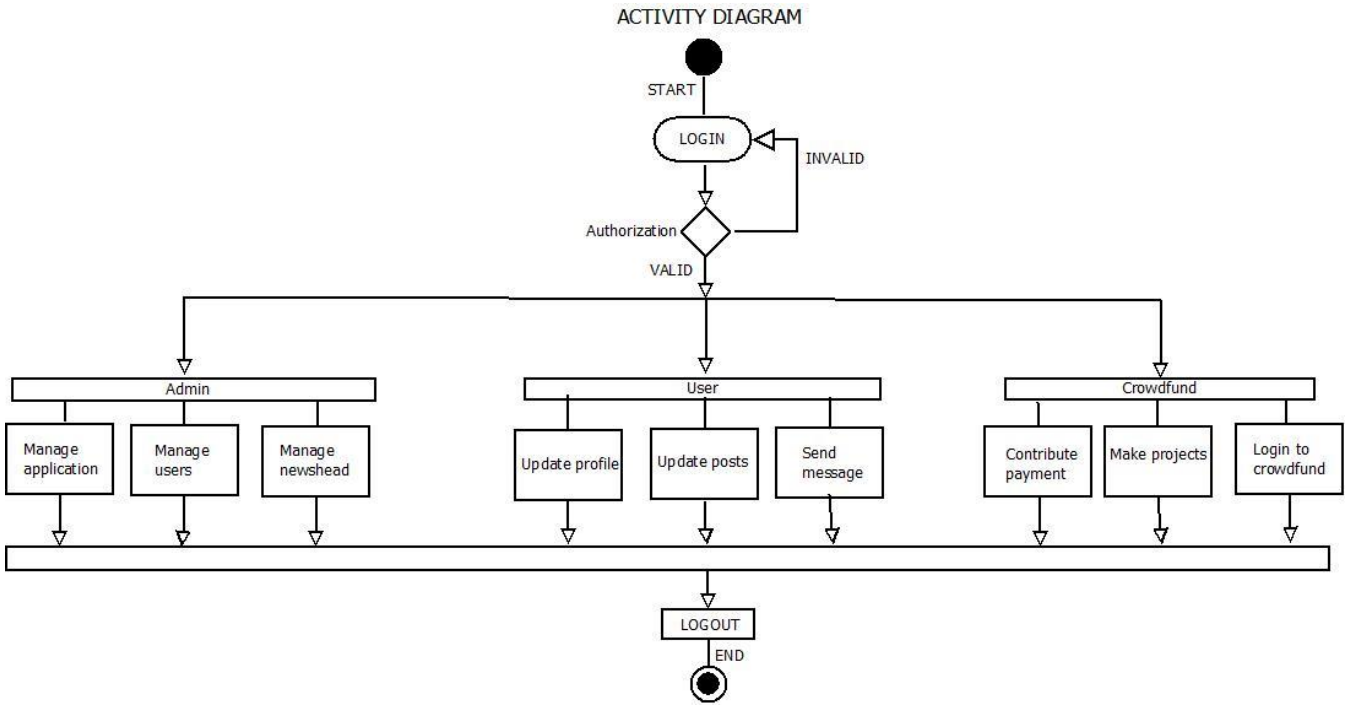


Fig 4: Activity diagram for 'JUSTCHAT'

4.2.5 Class Diagram

Class diagram is a static diagram. It represents the static view of the application. Class diagrams are useful for visualizing, describing, and documenting various system components as well as for writing executable code for software applications. A class diagram describes the constraints imposed on the system together with the properties and operations of a class. The only UML diagrams that can be directly converted into object-oriented languages are class diagrams, which are extensively utilized in the designing of object-oriented systems. An assortment of classes, interfaces, affiliations, partnerships, and limitations are displayed in a class diagram. It also goes by the name "structural diagram". Class diagrams are widely used in software development to design and document the structure of object-oriented systems. They help to identify and organize classes, their attributes, and behaviours, providing a blueprint for software implementation. Class diagrams also assist in understanding and communicating the relationships and interactions between different classes. Key elements in a class diagram include associations, which represent relationships between classes, such as one-to-one, one-to-many, or many-to-many associations. Inheritance is depicted using arrows to indicate the "is-a" relationship, where one class inherits attributes and methods from another. Aggregation and composition relationships are used to represent whole-part relationships between classes. Class diagrams are an essential tool for visualizing the static structure of a system and capturing its essential elements. They aid in system design, modelling, and documentation, allowing software developers and stakeholders to understand the system's architecture, dependencies, and responsibilities of different classes.

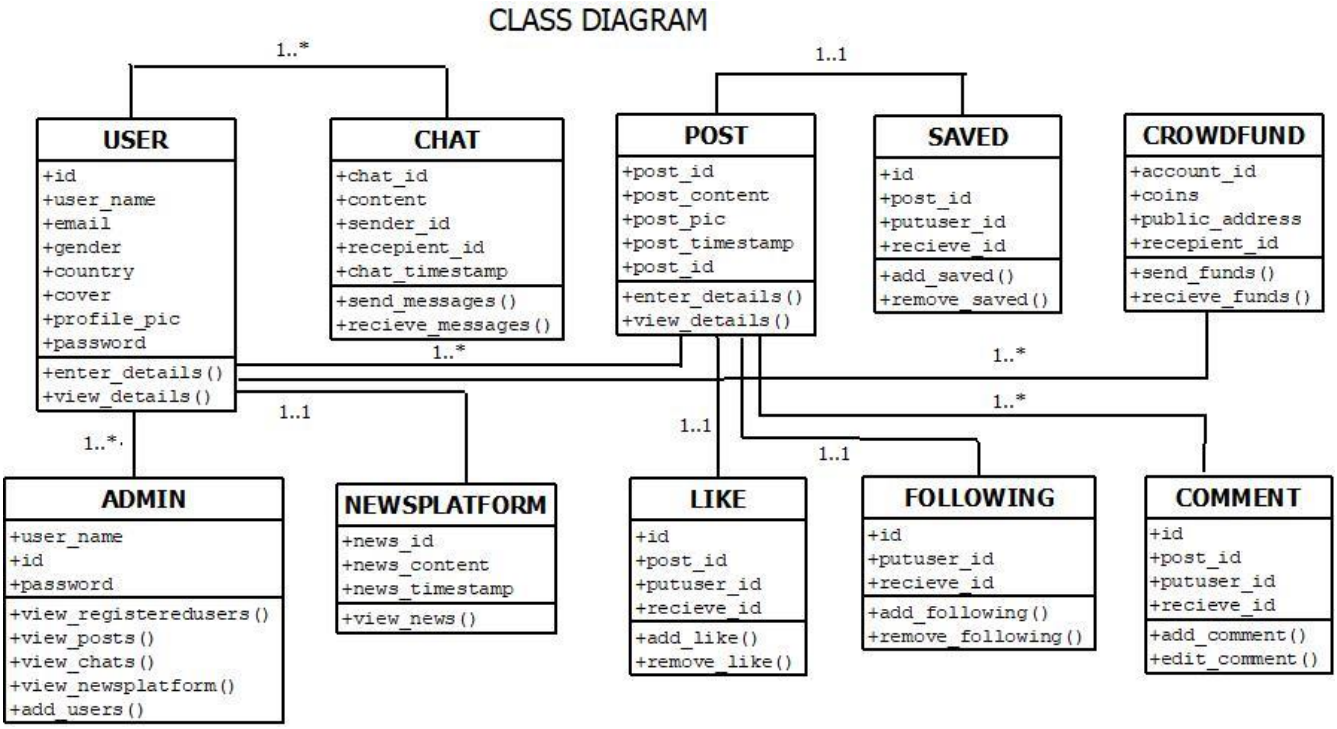


Fig 5: Class diagram for 'JUSTCHAT'

4.2.6 Object Diagram

A chart represents an instance of a chart. The basic concepts of art and art are the same. The diagramalso represents a static view of the system, but this static view is a snapshot of the system at a particular time. A diagram is used to illustrate a group of objects and their relationships as an example. The intended use of the image should be clearly understood. Charts are used in the same way as charts. The difference is that the diagram represents an abstract model of classes and their relationships. However, a painting represents a sample of a particular time and is authentic in nature.This means that the graph is close to the actual system behavior. The goal is to capture a static view of the system at a given time. In an object diagram, objects are depicted as individual instances of classes, usually with their attributes and current values. The relationships between objects are shown using lines and arrows, representing associations, dependencies, or other types of connections. Object diagrams are useful for visualizing and understanding the structure and relationships between objects within a system. They help to depict real-world scenarios or specific instances of a system, enabling software developers and stakeholders to gain insights into how objects collaborate and interact. One of the key benefits of object diagrams is that they provide a concrete representation of objects and their states. This allows for clearer communication and validation of system behavior, as well as identification of potential issues or inconsistencies. Object diagrams are particularly valuable for testing and debugging, as they help to analyze and understand the behavior of specific instances or scenarios.

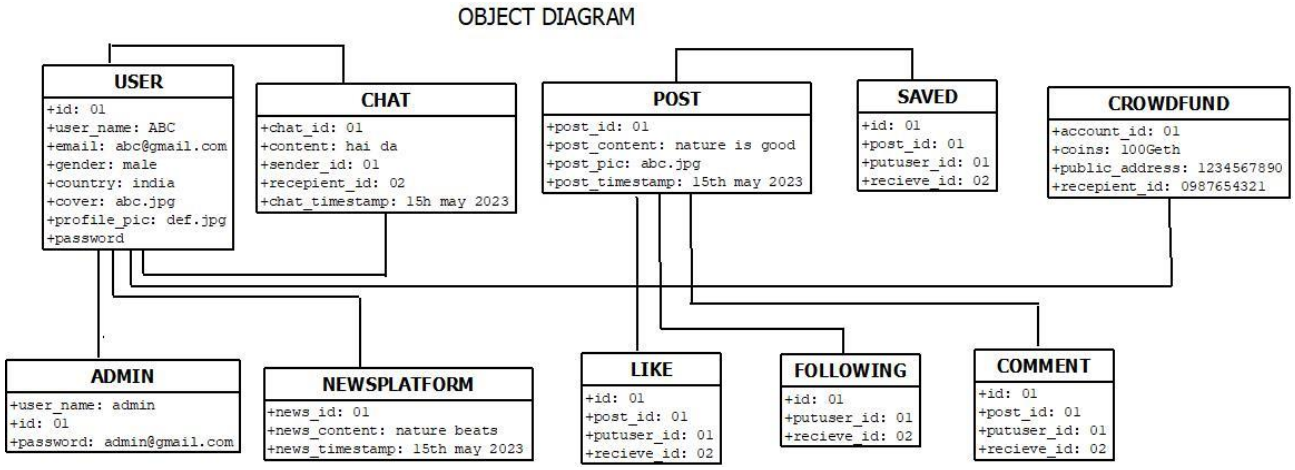


Fig 6: Object diagram for ‘JUSTCHAT’

4.2.7 Component Diagram

Component diagrams have different behaviors and personalities. The physical parts of the system are represented using component diagrams. Executables, libraries, files, documents, and other items that are physically present in a node are just a few examples. Component diagrams are used to show how the components of a system are connected and arranged. These diagrams may also be used to construct systems that can be run. Key elements in a component diagram include interfaces, which define the contract or communication protocol between components. Dependencies represent the reliance of one component on another, indicating that changes in one component may impact others. Component diagrams also show the provided and required interfaces of each component, specifying the services or functionalities it offers or requires. Component diagrams are commonly used in software development and system design to depict the overall structure and organization of a system.

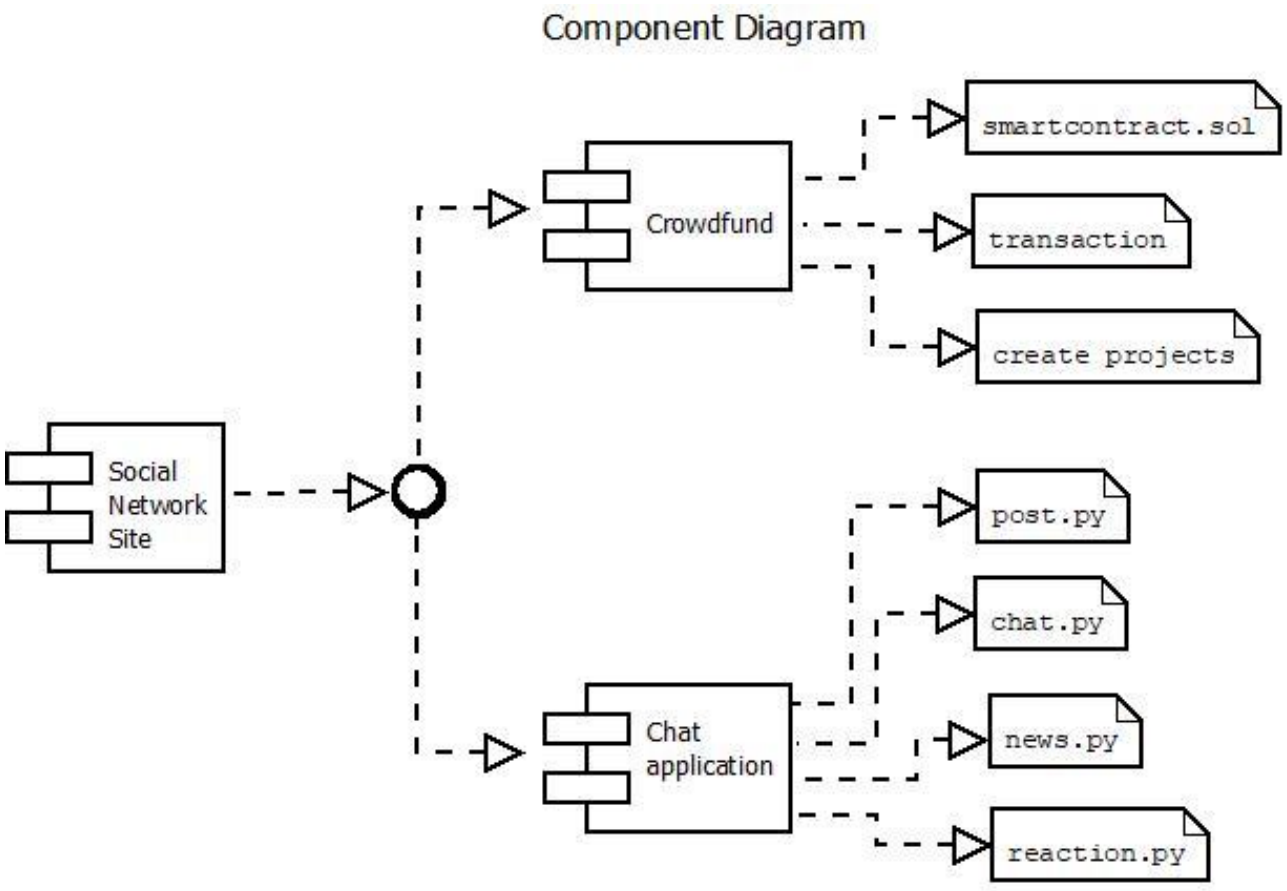


Fig 7: Component diagram for 'JUSTCHAT'

4.2.8 Deployment Diagram

An execution architecture of a system, containing nodes like hardware or software execution environments, and the middleware linking them, is shown in a deployment diagram, a form of UML diagram. Typically, deployment diagrams are used to represent the actual hardware and software of a system. By using it, you can comprehend how the hardware will physically deliver the system. In contrast to other UML diagram types, which primarily depict the logical Key elements in a deployment diagram include nodes, which represent the hardware or software execution environments, and their relationships with components and artifacts. Communication paths, such as network connections or communication protocols, are also depicted to illustrate the interaction and data flow between nodes. Deployment diagrams are commonly used in software development and system architecture to design, communicate, and document the deployment strategy of a system. They assist in planning the allocation of resources, identifying potential points of failure, and ensuring the scalability and efficiency of the system's deployment. In summary, deployment diagrams provide a visual representation of how components and artifacts are deployed on nodes within a system.

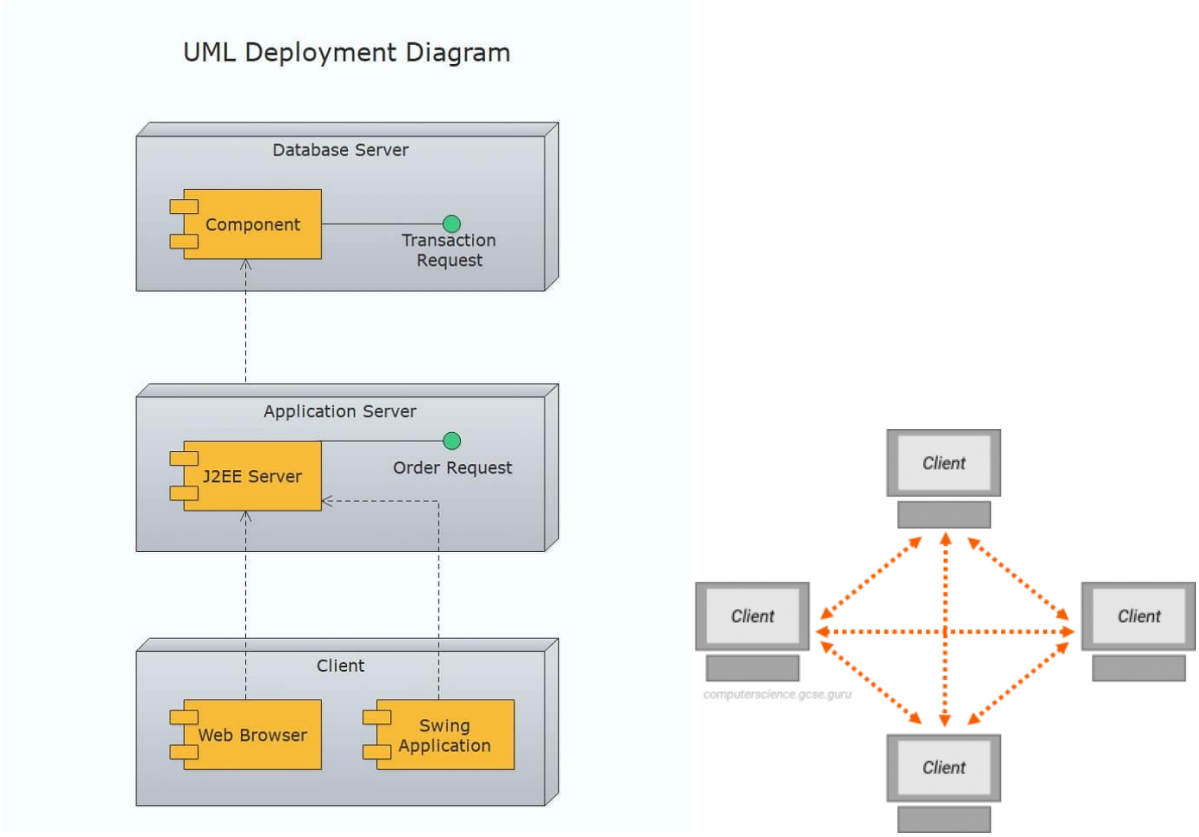
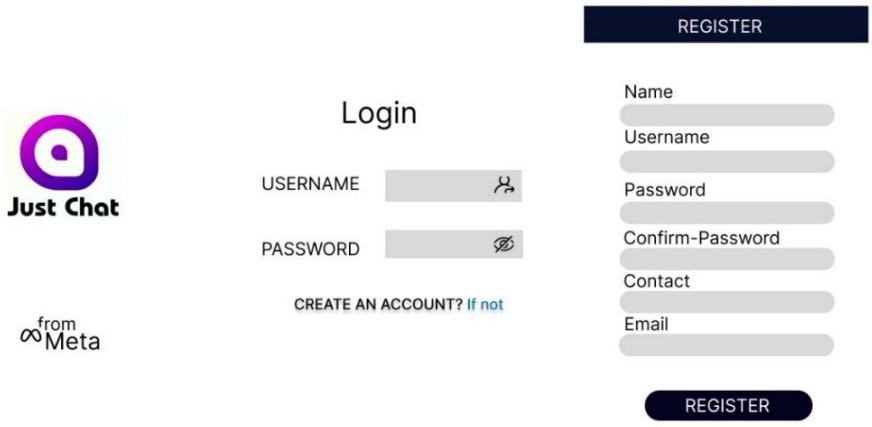


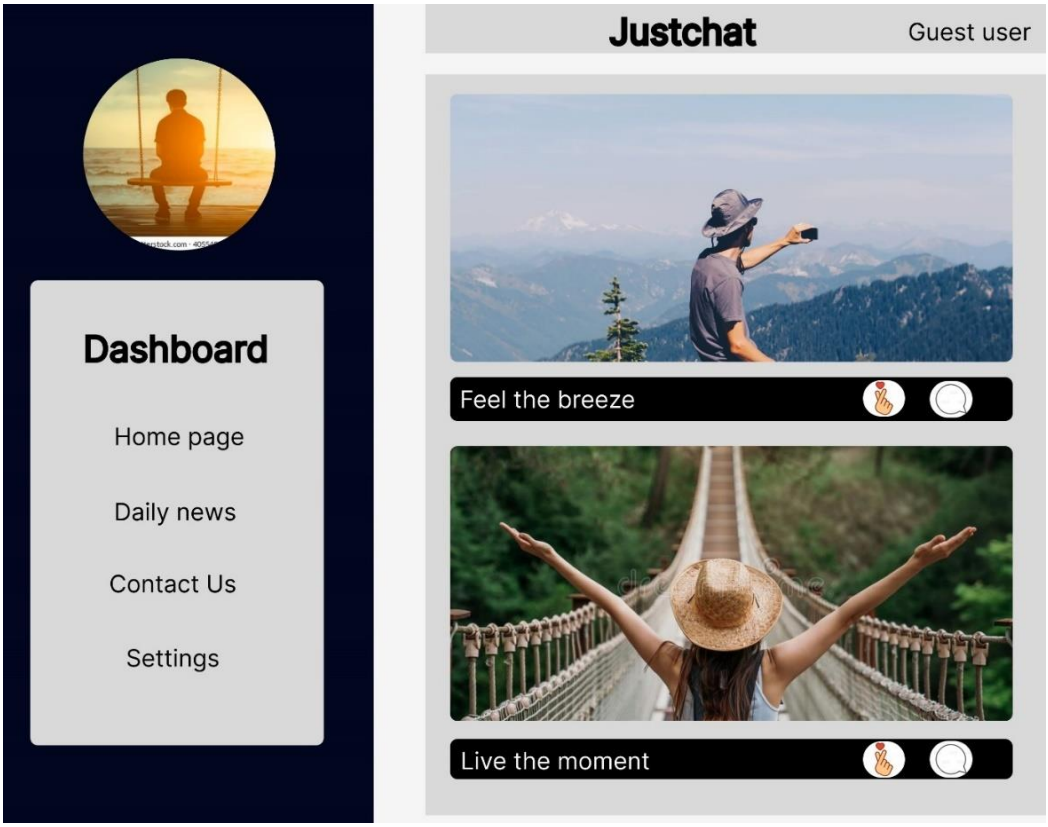
Fig 8: Deployment diagram for ‘JUSTCHAT’

4.3 USER INTERFACE DESIGN USING FIGMA

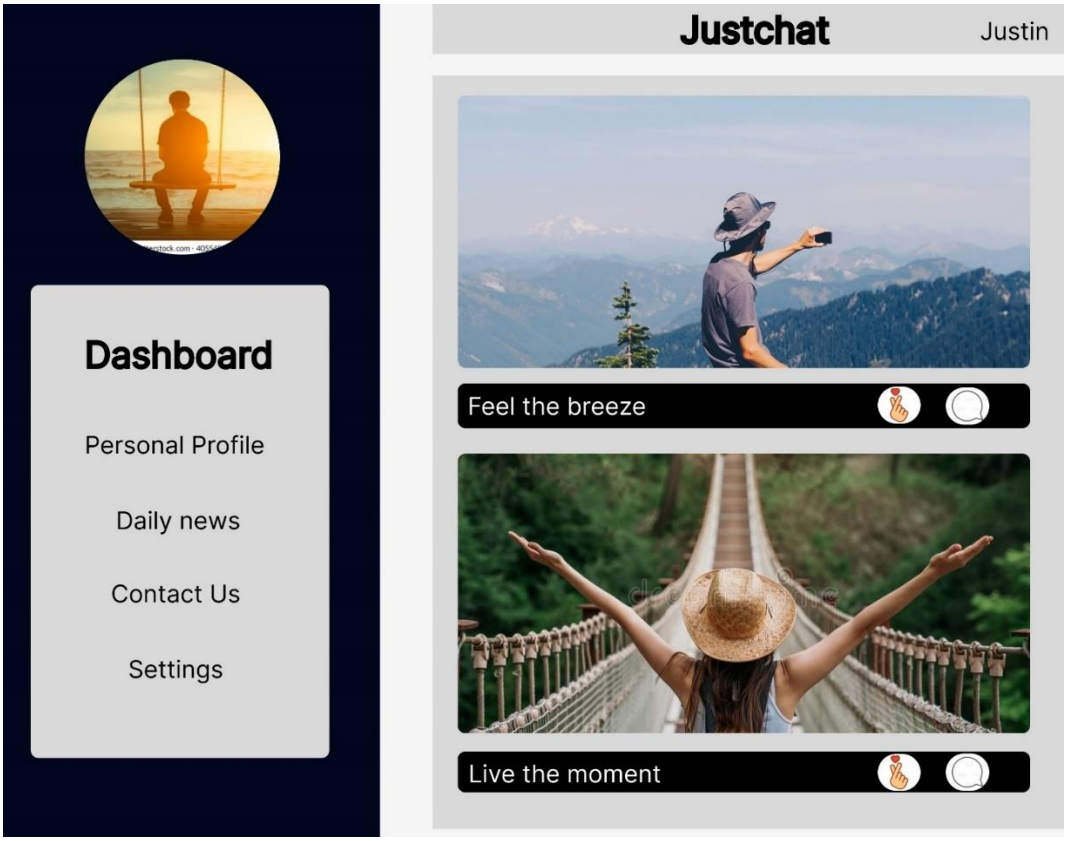
Front Page, Login Page, Register Page:



Guest Home Page:



User Home Page:



4.4 DATABASE DESIGN

A database is a process with the ability to store information to retrieve stored information fromusers in a quantity and quality. Data is the purpose of any database and must be protected. A database is created in two steps. Write out the user's needs in order to build a document that satisfies them as precisely as you can. This process, known as data-level creation, is separate from each DBMS. This data-level design is converted into a design for the specific DBMS being used to carry out the queries in the second stage. The details of the exact DBMS to be used are included in this step, which is referred to as physical layer design. System design and database design are complementary processes.

4.4.1 Relational Database Management System (RDBMS)

Data is represented via a relational model as a set of relationships. Every relationship is like a table of values or datasets. In relational model terminology, rows are called tuples, column headings are called attributes, and tables are relations. A relational file consists of tables, each of which is givena unique name. A row in the chart represents a group of related values.

4.4.2 Normalization

The table is a relationship. Rows in a table are called tuples. A tuple is an ordered collection of n elements. Lines are called attributes. Relationships are established between all tables in the database. This ensures the integrity of information and the relationship between organizations. Domain D is a set of atomic charges. One way to clarify the author is to determine the file type by extracting the main file that created the name. It's also helpful to give a field a name to help defineits value.

First Normal Form (1NF)

If a table's atomicity is 1, it is said to be in its first normal form. Atomicity dictates that a single cell cannot contain multiple values in this instance. It can only include one attribute with a single value. The multi-valued attribute, composite attribute, and their combinations are forbidden by the first normal form.

Example: The table of students' records below includes details on each student's age, course, course number, and roll number. You can see that the course column in the students record table has two values. As a result, it deviates from the First Normal Form.

rollno	name	course	age
1	Rahul	c/c++	22
2	Harsh	java	18
3	Sahil	c/c++	23
4	Adam	c/c++	22
5	Lisa	java	24
6	James	c/c++	19
NULL	NULL	NULL	NULL

rollno	name	course	age
1	Rahul	c	22
1	Rahul	c++	22
2	Harsh	java	18
3	Sahil	c	23
3	Sahil	c++	23
4	Adam	c	22
4	Adam	c++	22
5	Lisa	java	24
6	James	c	19
6	James	c++	19

Second Normal Form (2NF)

The first requirement for a table to be in Second Normal Form is that it must be in First NormalForm; the table should also not have partial dependency, which means that the correct subset of the candidate key should give a non-prime attribute. Now, let's understand the Second Normal Form using an example. You must divide the table into two pieces in order to get it to Second Normal Form. You will then the tables given below:

cust_id	storeid	store_location
1	D1	Toronto
2	D3	Miami
3	T1	California
4	F2	Florida
5	H3	Texas

cust_id	storeid
1	D1
2	D3
3	T1
4	F2
5	H3

storeid	store_location
D1	Toronto
D3	Miami
T1	California
F2	Florida
H3	Texas

The column store location is completely dependent on the main key of that table, storied, as you have removed the partial functional reliance from the location table.

Third Normal Form (3NF)

The table must first be in the Second Normal Form in order for it to be in the Third Normal Form. Non-prime attributes that are not a part of the candidate key should not depend on other non- prime characteristics in a table, according to the second requirement, which states that there should be no transitive reliance for non-prime attributes. As a result, a transitive dependency is a functional dependency in which A determines C indirectly due to A B and B C (where B A is not true).

A student table with their student ID, name, subject ID, and subject is provided below. In the student table shown above, sub_id and sub_id are determined by stu_id. As a result, sub is determined by stu_id via sub_id. As a result, the table is said to have a transitive functional dependency and does not meet the third normal form requirement. Now, split the table as indicated below to convert it to the third normal form:

stu_id	name	subid	sub	address
1	Arun	11	SQL	Delhi
2	Varun	12	Java	Bangalore
3	Harsh	13	C++	Delhi
4	Keshav	12	Java	Kochi

stu_id	name	subid	address
1	Arun	11	Delhi
2	Varun	12	Bangalore
3	Harsh	13	Delhi
4	Keshav	12	Kochi

subid	subject
11	SQL
12	java
13	C++
12	Java

All non-key attributes are now completely functioning and solely rely on the primary key, as you can see in both tables. Name, sub_id, and addresses are the only fields in the first table that depend on stu_id. The sub solely depends on sub_id in the second table.

4.4.3 Sanitization

Data sanitization entails removing or erasing data from a storage device on purpose and permanently to make sure it cannot be restored. Normally, when data is removed from storage media, the medium is not truly wiped, and an attackerwho gains access to the device can recover the data. Serious questions about security and data privacy are raised by this. Sanitization involves cleaning storage media so that no data remains on the device and that no data can be recovered even with cutting-edge forensic techniques.

4.4.4 Indexing

By reducing the number of disc accesses needed when a query is completed, indexing helps a database perform better. It is a data structure method used to locate and access data in a database rapidly. Several database columns are used to generate indexes.

- The search key, which is the first column, contains a duplicate of the table's primary key or potential primary key. These values are kept in sorted order so that it is easy to get the corresponding data.

  Note: The data may or may not be kept in sorted order.
- The address of the disc block on which that specific key value is stored is held by a series ofpointers in the second column, which is designated as the Data Reference or Pointer.

4.5 TABLE DESIGN

1. tbl_profileapp_user

Primary Key: **id**

No:	Fieldname	Key constraints	Data type	Size	Description
1	id	Primary key	int	20	Unique id of the table
2	password	Not null	int	50	Password
3	last_login	Not null	varchar	20	Last_login
4	is_superuser	Not null	varchar	20	Checking superuser
5	username	Not null	int	50	User name
6	first_name	Not null	varchar	50	First name
7	last_name	Not null	varchar	50	Last name
8	email	Not null	int	100	Email
9	is_active	Not null	varchar	20	Checking whether active
10	date_joined	Not null	int	20	Date
11	profile_pic	Not null	varchar	200	Profile image
12	cover	Not null	varchar	200	Cover
13	country	Not null	varchar	50	Country
14	gender	Not null	varchar	50	Gender

2. tbl_chat_message

Primary Key: **id**

Foreign Key: **id references table tbl_profileapp_user**

No:	Fieldname	Key constraints	Data type	Size	Description
1	id	Primary key	int	20	Unique id of table
2	message	Not null	int	100	content
3	timestamp	Not Null	int	30	date
4	recipient_id	Not Null	int	20	Recipient id

5	sender_id	Not Null	int	20	Sender id

3. tbl_profileapp_comment

Primary Key: **id**

Foreign Key: **id references table tbl_profileapp_post**

No:	Fieldname	Key constraints	Data type	Size	Description
1	id	Primary key	int	20	Unique id for table
2	comment_content	Not null	int	100	Comment content
3	comment_time	Not null	varchar	20	Time of Comment
4	commenter_id	Not null	varchar	20	Comment id
5	post_id	Not null	varchar	20	Post id

4. tbl_profileapp_followers

Primary Key: **id**

Foreign Key: **id references table tbl_profileapp_user**

No:	Fieldname	Key constraints	Data type	Size	Description
1	id	Primary key	int	20	Unique id for table
2	user_id	Not null	int	20	User id

5. tbl_profileapp_follower_followers

Primary Key: **id**

Foreign Key: **id references table tbl_profileapp_user**

No:	Fieldname	Key constraints	Data type	Size	Description
1	id	Primary key	int	20	Unique id for table
2	followers_id	Not null	varchar	20	Followers id
3	user_id	Not null	varchar	20	User id

6. tbl_profileapp_post:

Primary Key: **id**
Foreign Key: **id references table tbl_profileapp_user**

No:	Fieldname	Key constraints	Data type	Size	Description
1	id	Primary key	int	20	Unique id for the table
2	date_created	Not null	int	20	date created
3	context_text	Not null	int	100	Text context
4	context_image	Not null	date	200	context image
5	comment_count	Not null	int	100	comment count
6	creater_id	Not null	int	20	creator_id

7. tbl_profileapp_post_likers:

Primary Key: **id**
Foreign Key: **id references table tbl_profileapp_post**

No:	Fieldname	Key constraints	Data type	Size	Description
1	id	Primary key	int	20	Unique id for table
2	post_id	Not null	int	20	id of post
3	user_id	Not null	varchar	20	id of user

8. tbl_profileapp_post_savers:

Primary Key: **id**
Foreign Key: **id references table tbl_profileapp_post**

No:	Fieldname	Key constraints	Data type	Size	Description
1	id	Primary key	int	20	Unique Id for the table
2	post_id	Not null	int	20	Post id
3	user_id	Not null	varchar	20	User id

CHAPTER 5
SYSTEM TESTING

5.1 INTRODUCTION

Software Testing is the process of executing software in a controlled manner, in order to answer the question - Does the software behave as specified? Software testing is often used in association with the terms verification and validation. Validation is the checking or testing of items, includes software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, analysis, inspections, and walkthroughs. Validation is the process of checking that what has been specified is what the user actually wanted.

5.2 TEST PLAN

An extensive document that describes the testing strategy and approach for a specific project or product is called a test plan. Selenium is one of the most popular tools for automating tests when it comes to web application testing. A test plan suggests a number of required steps that need be taken in order to complete various testing methodologies. Therefore, the test plan should include information on the mean time to failure, the cost to locateand correct the flaws, the residual defect density or frequency of occurrence, and the number of test work hours required for each regression test.

The testing levels include:

- Unit testing
- Integration Testing
- Data validation Testing
- Output Testing

Overall, defining the scope, identifying the test environment, defining the test cases, generating the test scripts, running the tests, recording the results, analyzing the results, and reporting the results are all steps in the Selenium test plan creation process. By doing the following actions, you can make sure that your testing is thorough and efficient and that you are able to spot and fix any issues before they become serious ones.

5.2.1 Unit Testing

Unit testing concentrates verification efforts on the software component or module, which is the smallest unit of software design. The component level design description is used as a guide when testing crucial control paths to find faults inside the module's perimeter. The unit testing's disclosed scope and the tests' relative complexity. Unit testing is white-box focused, and numerous components may be tested simultaneously. To guarantee that data centers and exits the program unit under test properly, the modular interface is tested. To make sure that data temporarily stored retains its integrity during all phases of an algorithm's execution, the local data structure is inspected. To confirm that each statement in a module has been executed at least once, boundary conditionsare evaluated. Finally, each path for managing errors is examined. Before starting any other test, tests of data flow over a module interface must be completed.

5.2.2 Integration Testing

Integration testing is systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. The objective is to take unit tested components and build a program structure that has been dictated by design. The entire program is tested as whole. Correction is difficult because isolation of causes is complicated by vast expanse of entire program. Once these errors are corrected, new ones appear and the process continues in a seemingly endless loop. After performing unit testing in the System all the modules were integrated to test for any inconsistencies in the interfaces. Moreover, differences in program structures were removed and a unique program structure was evolved.

5.2.3 Validation Testing or System Testing

The testing process comes to an end here. This involved testing the entire system in its entirety, including all forms, code, modules, and class modules. Popular names for this type of testing include system tests and black box testing. The functional requirements of the software are the main emphasis of the black box testing approach. That example, using Black Box testing, a software engineer can create sets of input conditions that will fully test every program requirement. Errors in data structures or external data access, erroneous or missing functions, interface errors, performance issues, initialization issues, and termination issues are all types of faults that black box testing looks for.

5.2.4 Output Testing or User Acceptance Testing

User approval of the system under consideration is tested; in this case, it must meet the needs of the company. When developing, the program should stay in touch with the user and perspective system to make modifications as needed. With regard to the following points, this was done:

- Input Screen Designs,
- Output Screen Designs,

The aforementioned testing is carried out using a variety of test data. The preparation of test data is essential to the system testing process. The system under investigation is then put to the test using the prepared test data. When testing the system, test data issues are found again and fixed using the testing procedures described above. The fixes are also logged for use in the future.

5.2.5 Automation Testing

A test case suite is executed using specialized automated testing software tools as part of the software testing technique known as automation testing. The test stages are meticulously carried out by a human performing manual testing while seated in front of a computer. Additionally, the automation testing software may generate thorough test reports, compare expected and actual findings, and enter test data into the System Under Test. Software test automation necessitates significant financial and material inputs. Repeated execution of the same test suite will be necessary during subsequent development cycles.

5.2.6 Selenium Testing

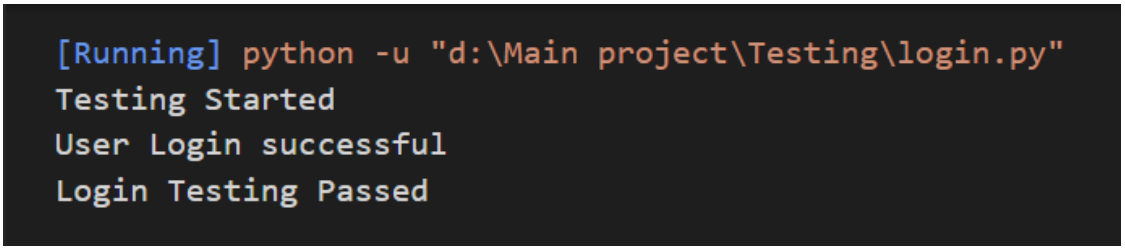
Selenium is a free (open-source) automated testing framework used to verify web applications across different browsers and platforms. You can use multiple programming languages to create Selenium test scripts like Java, C#, Python, etc. Testing done using Selenium testing tool is usually referred to as Selenium testing Since Selenium is a collection of different tools, it also had different developers. Below are the key people who have made significant contributions to the Selenium project. Moreover, Selenium's extensibility allows integration with other testing frameworks, tools, and technologies. It can be combined with testing frameworks like TestNG or JUnit for advanced test management and reporting. It integrates well with popular build tools, source control systems, and defect tracking systems.

Test Case 1: Guest - Login

Code:

```
from selenium import webdriver
import time
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import Select

options=webdriver.ChromeOptions()
options.add_experimental_option('excludeSwitches',['enable-logging'])
driver = webdriver.Chrome(options=options)
driver.maximize_window()

print("Testing Started")
driver.get("http://127.0.0.1:8000/n/login")
driver.find_element("id", "email").send_keys("malavika")
time.sleep(1)
driver.find_element("id", "password").send_keys("malavika")
time.sleep(1)
driver.find_element("xpath", "/html/body/div/div/form/center/input").click()
time.sleep(1)

if driver.current_url == 'http://127.0.0.1:8000/':
    print('User Login successful')
else:
    print('User Login failed')
print("Login Testing Passed")

driver.quit()
```

Screenshot:



Test report:

Test Case 1

Project Name: JUSTCHAT					
Login Test Case					
Test Case ID: 1			**Test Designed By:** Justin V Kalappura		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 12-05-2023		
Module Name: Login Screen			**Test Executed By:** Mr. Ajith G.S		
Test Title: Guest Login			**Test Execution Date:** 12-05-2023		
Description: Verify login with valid username and password					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to Login Page		Index features should be displayed	Login page displayed	Pass
2	Provide Valid Username	User Name: malavika	User should be able to Login	User Logged in and navigated to their corresponding Dashboard	Pass
3	Provide Valid Password	Password: Malavika@123			
4	Click on Login button				
Post-Condition: User is validated with database and successfully login into account. The Account session details are logged in database.					

Text case 2 : Following option

Code:

```
from selenium import webdriver
import time
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import Select

print("Testing Started")
options=webdriver.ChromeOptions()
options.add_experimental_option('excludeSwitches',['enable-logging'])
driver = webdriver.Chrome(options=options)
driver.maximize_window()

driver.get("http://127.0.0.1:8000/n/login")
driver.find_element("id", "email").send_keys("malavika")
time.sleep(1)
driver.find_element("id", "password").send_keys("malavika")
time.sleep(1)
driver.find_element("xpath", "/html/body/div/div/form/center/input").click()
time.sleep(1)
driver.find_element("xpath",
"/html/body/div[2]/div[3]/div/div[2]/div[2]/div[3]/button").click()
time.sleep(1)
driver.find_element("xpath",
"/html/body/div[2]/div[1]/div/div[1]/div/ul/li[2]/a/div").click()
time.sleep(1)

if driver.current_url == 'http://127.0.0.1:8000/n/following':
    print('Following option is working')
else:
    print('Following option failed working')

print("Following Testing Passed")

driver.quit()
```

Screenshot:

```
[Running] python -u "d:\Main project\Testing\Following.py"
Testing Started
Following option is working
Following Testing Passed
```

Test report:

Project Name: JUSTCHAT					
Update Following Test Case					
Test Case ID: 2			**Test Designed By:** Justin V Kalappura		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 12-05-2023		
Module Name: Following/Unfollow			**Test Executed By:** Mr. Ajith G.S		
Test Title: Update Following			**Test Execution Date:** 12-05-2023		
Description: Following functionality working or not					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to Login Page		Login page should be displayed	Login page displayed	Pass
2	Provide Valid Username	User name: malavika	User should be able to Login	User able to login to the panel	Pass
3	Provide Valid Password	Password: Malavika@123			
4	Click on Login button				
5	Click on Following button				
6	Check whether following is working or not		User started following the other user.	User able to start following.	Pass
Post-Condition: User is validated with database and successfully login into account and the Following option is checked whether working or not.					

Test Case 3: Post option

Code :

```
from selenium import webdriver
import time
import os
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import Select
```

```
options=webdriver.ChromeOptions()
options.add_experimental_option('excludeSwitches',['enable-logging'])
driver = webdriver.Chrome(options=options)
driver.maximize_window()

print("Testing Started")
driver.get("http://127.0.0.1:8000/n/login")
driver.find_element("id", "email").send_keys("malavika")
time.sleep(1)
driver.find_element("id", "password").send_keys("malavika")
time.sleep(1)
driver.find_element("xpath", "/html/body/div/div/form/center/input").click()
time.sleep(1)
driver.find_element("xpath", "/html/body/div[2]/div[1]/div/div[1]/button").click()
time.sleep(1)
driver.execute_script("createpost()")
driver.find_element("id", "post-text").send_keys("This is my test post.")
time.sleep(1)
image_filename = "spoon.JPG"
image_path = os.path.abspath(os.path.join(os.getcwd(), "static", "assets",
"images", image_filename))
file_input = driver.find_element("id", "insert-img")
file_input.send_keys(image_path)
time.sleep(1)
submit_button = driver.find_element(By.CLASS_NAME, "submit-btn")
submit_button.click()
time.sleep(1)

if driver.current_url == 'http://127.0.0.1:8000/':
    print("Post created successfully")
else:
    print('Post creation failed')
print("Post Testing Passed")

driver.quit()
```

Screenshot:

```
[Running] python -u "d:\Main project\Testing\post.py"
Testing Started
Post created successfully
Post Testing Passed
```

Test Case 3

Project Name: JUSTCHAT					
Guest Post creation					
Test Case ID: 3			**Test Designed By:** Justin V Kalappura		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 10-05-2023		
Module Name: Post creation			**Test Executed By:** Mr. Ajith G.S		
Test Title: Guest Post creation			**Test Execution Date:** 10-05-2023		
Description: Creation of a post by a user					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to Login Page		Login page should be displayed	Login page displayed	Pass
2	Provide Valid Username	Username: malavika	User should be able to Login	User able to login to the panel	Pass
3	Provide Valid Password	Password: Malavika@123		User Able to select the contents for the post	Pass
4	Click on Login button				
5	Click on create post				
6	Select picture and content	Text content: Hai Picture content: eye.jpg	User should able to select a picture and content		Pass
7.	Click on Post Button		User should able to put a post	Post created successfully	Pass

Post-Condition: New Post created Successfully.

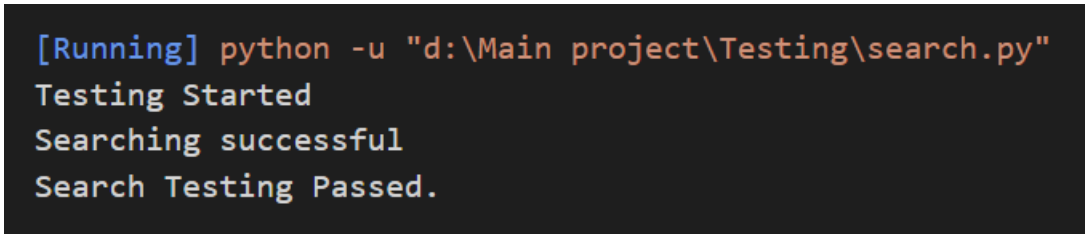
Test Case 4: Search Testing

Code:

```
from selenium import webdriver
import time
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import Select

options=webdriver.ChromeOptions()
options.add_experimental_option('excludeSwitches',['enable-logging'])
driver = webdriver.Chrome(options=options)
```

```
driver.maximize_window()

print("Testing Started")
driver.get("http://127.0.0.1:8000/n/login")
driver.find_element("id", "email").send_keys("malavika")
time.sleep(1)
driver.find_element("id", "password").send_keys("malavika")
time.sleep(1)
driver.find_element("xpath", "/html/body/div/div/form/center/input").click()
time.sleep(1)
search_box = driver.find_element("id", "search-box")
search_box.send_keys("sajo")
search_box.send_keys(Keys.ENTER)
time.sleep(1)
driver.find_element("xpath",
"/html/body/div[2]/div[2]/div/div/div/div/div[2]/div/div/div/div/div[1]/center/a/di
v").click()
time.sleep(1)

if driver.current_url == 'http://127.0.0.1:8000/sajo':
    print("Searching successful")
else:
    print('Searching failed')
print("Search Testing Passed")

driver.quit()
```

Screenshot:



Test Case 4

Project Name: JUSTCHAT					
User Searching					
Test Case ID: 4			**Test Designed By:** Justin V Kalappura		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 10-05-2023		
Module Name: Searching			**Test Executed By:** Mr. Ajith G.S		
Test Title: User Searching			**Test Execution Date:** 10-05-2023		
Description: Searching implemented for users					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to Login Page		Login page should be displayed	Login page displayed	Pass
2	Provide Valid Username	User Name: malavika	User should be able to Login	User able to login to the panel	Pass
3	Provide Valid Password	Password: Malavika@123			
4	Click on Login Button				
5	Search a user	Sajo	Should get the searched result	Got the searched value	Pass
Post-Condition: Searching is successfully implemented.					

Test Case 5: Update password

Code

```
from selenium import webdriver
import time
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.action_chains import ActionChains
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import Select, WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
```

```
from selenium.common.exceptions import TimeoutException

options=webdriver.ChromeOptions()
options.add_experimental_option('excludeSwitches',['enable-logging'])
driver = webdriver.Chrome(options=options)
driver.maximize_window()
wait = WebDriverWait(driver, 30)

print("Testing Started")
driver.get("http://127.0.0.1:8000/n/login")
driver.find_element("id", "email").send_keys("sreelakshmi")
time.sleep(1)
driver.find_element("id", "password").send_keys("Sreelakshmi@123")
time.sleep(1)
driver.find_element("xpath", "/html/body/div/div/form/center/input").click()
time.sleep(1)
driver.find_element("xpath",
"/html/body/div[2]/div[1]/div/div[1]/div/ul/li[4]/a/div").click()
time.sleep(1)
driver.find_element("xpath",
"/html/body/div[2]/div[2]/div/div/div[1]/div[3]/div[1]/div/button").click()
time.sleep(1)
menu_button = driver.find_element("id", "menu")
menu_button.click()
driver.find_element("xpath",
"/html/body/div[2]/div[2]/div/div/div[1]/div[3]/div[1]/div/div/a[2]").click()
time.sleep(1)
current_password = driver.find_element("id", "currentpassword")
current_password.send_keys("Sreelakshmi@123")
time.sleep(1)
new_password = driver.find_element("id", "newpassword")
new_password.send_keys("Sreelakshmi@1234")
time.sleep(1)
confirm_password = driver.find_element("id", "confirmpassword")
confirm_password.send_keys("Sreelakshmi@1234")
time.sleep(1)
driver.find_element("xpath",
"/html/body/div[2]/div[2]/div/div/div/div/div/div/form/div[4]/div/button").click()
time.sleep(1)

if driver.current_url == 'http://127.0.0.1:8000/n/login':
    print("UpdatePassword successful")
else:
    print('UpdatePassword failed')
print("UpdatePassword Testing Passed")
driver.quit()
```

Screenshot:

```
[Running] python -u "d:\Main project\Testing\updatepassword.py"
Testing Started
UpdatePassword successful
UpdatePassword Testing Passed
```

Test Case 5

Project Name: JUSTCHAT					
Update Password					
Test Case ID: 5			**Test Designed By:** Justin V Kalappura		
Test Priority (Low/Medium/High): High			**Test Designed Date:** 10-05-2023		
Module Name: Update Password			**Test Executed By:** Mr. Ajith G.S		
Test Title: User Update Password			**Test Execution Date:** 10-05-2023		
Description: User can update their own password					
Pre-Condition: User has valid username and password					
Step	**Test Step**	**Test Data**	**Expected Result**	**Actual Result**	**Status (Pass/Fail)**
1	Navigation to Login Page		Dashboard shouldbe displayed	Donation page displayed	Pass
2	Provide valid Username	User Name: malavika	User should be able to Login	User abled to login	Pass
3	Provide valid Password	Password: Malavika@123			
4	Click on Login Button				
5	Click on the update password option				
6	Provide correct password	Malavika@123	Able to enter the old password.	Can enter old password.	Pass
7	Provide new and confirm password	Malavika@1234	Able to enter the new and confirm password.	Can enter new and confirm password.	Pass
8	Click update button		Given password has to be updated properly.	Password update implemented successfully	Pass
Post-Condition: Password updated Successfully.					

CHAPTER 6
IMPLEMENTATION

6.1 INTRODUCTION

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate. It is primarily concerned with user training and documentation. Conversion usually takes place about the same time the user is being trained or later. Implementation simply means convening a new system design into operation, which is the process of converting a new revised system design into an operational one. At this stage the main work load, the greatest upheaval and the major impact on the existing system shifts to the user department. If the implementation is not carefully planned or controlled, it can create chaos and confusion.

Implementation includes all those activities that take place to convert from the existing system to the new system. The new system may be a totally new, replacing an existing manual or automated system or it may be a modification to an existing system. Proper implementation is essential to provide a reliable system to meet organization requirements. The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specifications. The system personnel check the feasibility of the system. The more complex the system being implemented, the more involved will be the system analysis and design effort required to implement the three main aspects: education and training, system testing and changeover.

The implementation state involves the following tasks:

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.

6.2 IMPLEMENTATION PROCEDURES

Implementation of software refers to the final installation of the package in its real environment, to the satisfaction of the intended uses and the operation of the system. In many organizations someone who will not be operating it, will commission the software development project. In the initial stage people doubt about the software but we have to ensure that the resistance does not build up, as one has to make sure that:

- The active user must be aware of the benefits of using the new system.
- Their confidence in the software is built up.
- Proper guidance is imparted to the user so that he is comfortable in using the application.

Before going ahead and viewing the system, the user must know that for the result, the server program should be running in the server. If the server object is not up running on the server, the process won't take place.

6.2.1 User Training

User training is designed to prepare the user for testing and converting the system. To achieve the objective and benefits expected from computer based system, it is essential for the people who will be involved to be confident of their role in the new system. As system becomes more complex, the need for training is more important. By user training the user comes to know how to enter data, respond to error messages, interrogate the database and call up routine that will produce reports and perform other necessary functions.

6.2.2 System Maintenance

Maintenance is the enigma of system development. The maintenance phase of the software cycle is the time in which a software product performs useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect in the software development life cycle. The need for system maintenance is for it to make adaptable to the changes in the system environment. Software maintenance is of course, far more than "Finding Mistakes".

6.2.3 Training on the Application Software

After providing the necessary basic training on computer awareness the user will have to be trained on the new application software. This will give the underlying philosophy of the use of the new system such as the screen flow, screen design type of help on the screen, type of errors while entering the data, the corresponding validation check at each entry and the ways to correct the date entered. It should then cover information needed by the specific user/ group to use the system or part of the system while imparting the training of the program on the application. This training may be different across different user groups and across different levels of hierarchy.

6.2.4 Hosting

Hosting is the process of providing a platform or service for storing, serving, and managing websites, applications, or data on the internet. Web hosting is a common form of hosting that allows individuals and organizations to make their websites accessible on the internet. There are different types of web hosting, including shared hosting, VPS hosting dedicated hosting, cloud hosting, and managed hosting. Shared hosting is the most affordable option, as it involves sharing server resources with other websites. PS hosting offers more flexibility and control over server resources, while dedicated hosting provides exclusive use of a server.

Managed hosting involves outsourcing the technical aspects of hosting to a third-party provider, allowing businesses to focus on their core activities. Choosing the right hosting provider is crucial,as it affects website performance, security, and uptime. Factors to consider when selecting a hosting provider include cost, reliability, support, scalability, and features. Overall, hosting plays a critical role in enabling businesses and individuals to establish a presence on the internet and reach a global audience.

AWS Web hosting

The needs of companies of all sizes are met by a variety of web hosting services provided by Amazon Web Services (AWS). Among other things, AWS offers a scalable and dependable infrastructure for hosting websites, web apps, and e-commerce platforms. Amazon Elastic Compute Cloud (EC2), AWS Elastic Beanstalk, AWS Lambda, Amazon Lightsail, and Amazon S3 are justa few of the hosting options available to users. While Elastic Beanstalk offers a simple-to-use infrastructure for delivering and scaling web applications, EC2 enables complete control over the virtual servers. Serverless computing services like Lambda let customers run code without having to provision or manage servers. While Amazon S3 is an object storage service that can be used to store and retrieve files and static website content, Lightsail is a straightforward, affordable choice for those who only require a basic website or application. AWS web hosting services provide cutting-edge security tools like SSL/TLS encryption, network firewalls, and distributed denial-of-service (DDoS) prevention, ensuring the safety and high availability of the online applications hosted on AWS.

EC2 (Elastic cloud compute)

Renting virtual computing resources, such as virtual machines (VMs) or instances, so that users can run their own applications is made possible via the Amazon Elastic Compute Cloud (EC2) web service, which is offered by Amazon Web Services (AWS). Users have flexibility and scalability with EC2 instances since they may be built up with different hardware configurations, operating systems, and networking configurations. EC2 instances are a cost-effective choice for companies and individuals who need on-demand computing power since they are simple to launch and terminate as needed and users just pay for the resources they use. In addition, EC2 works in conjunction with other AWS services to deliver a whole cloud computing platform.

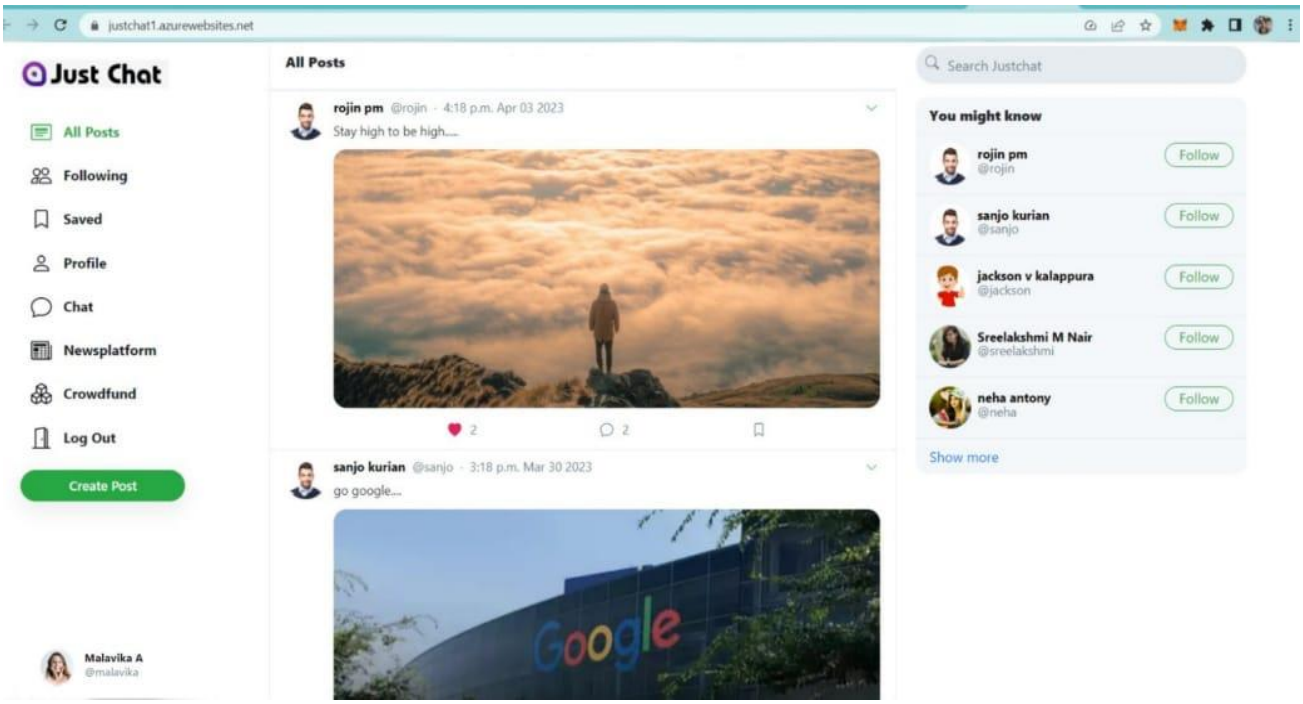
Steps to create instance:

1. Open the EC2 dashboard in your AWS Management Console and log in. • Select the "Launch Instance" option to start building a new instance. Click on the "Launch Instance" button to create new instance.
2. Decide which Amazon Machine Image (AMI) will serve as the foundation of your instance.Usually, this will be a pre-configured image of a certain software stack or operating system.
3. Choose the instance type you want to employ. Your instance's CPU, memory, storage, and network capabilities are determined by this.
4. Set up the instance's specifics, including the quantity of instances you wish to run, network configurations, and security groups.
5. Include any additional volumes or storage devices that your instance requires.

Examine and set up any necessary advanced settings, such as placement groups, user datascripts, or IAM roles. Lastly, launch your instance after reviewing its launch configuration, which includes theinstance type, storage, network configuration, and security groups.

Hosted website:

Hosted link: https://justchat1.azurewebsites.net



CHAPTER 7
CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION

The analysis shows that our system “JUSTCHAT (social media platform)” is better product for doing social upliftment in a society. The existing systems are very poor and inefficient in this concern. The new system that be introduced is effective, efficient and simple in managing and molding a socially aware community or society. It makes everyone equally important and give a common space to express their views on the present events around. This system tested with sample data and output obtained in according the requirements. the performance of the system is evaluated, and is found to be much efficient than the existing system. Though it could not be claimed that is an ideal project, it will meet the primary requirements of the concern. So the project has to be proved by having modifications as and when necessity arises in due course. The system was implemented and tested with real as per the facilities available.

7.2 FUTURE SCOPE

- Additional security features can be implemented ie by making the same application into a web 3.0 application using technologies like blockchain.
- A mobile application can be developed for this system so that to make a socially aware society around.
- Real time news can be made visible.
- Can be made efficient or profitable by making a space to show the admin our application.

CHAPTER 8
BIBLIOGRAPHY

REFERENCES:

- Gary B. Shelly, Harry J. Rosenblatt, “System Analysis and Design”, 2009.
- Roger S Pressman, “Software Engineering”, 1994.
- PankajJalote, “Software engineering: a precise approach”, 2006.
- CSS Cookbook by Christopher Schmitt
- IEEE Std 1016 Recommended Practice for Software Design Descriptions.

WEBSITES:

- www.w3schools.com
- www.jquery.com
- https://www.lovelycoding.org/construction-work-management-system/
- https://app.diagrams.net
- http://homepages.dcc.ufmg.br/~rodolfo/es-1-03/IEEE-Std-830-1998.pdf
- www.agilemodeling.com/artifacts/useCaseDiagram.html
- https://chat.openai.com/

CHAPTER 9

APPENDIX

9.1 Sample Code:

1.Login.html

```
{% extends "profileapp/layout2.html" %}

{% load static %}

{% block title %}

   Sign In | Network

{% endblock %}

{% block body %}

   <div class="user-box">

      <div>  <center>

<img src="{% static 'profileapp/Images/imagebg.png' %}" height="45em" style="margin-bottom: 2vh;">

            <h3>Log in to Network</h3><br>

         </center>

      </div>

      <form onsubmit="return validatee()" action="{% url 'login' %}" method="post" id="signin-form" style="width: 100%;margin-bottom: 5vh;">

         {% include 'profileapp/notification.html' %}

         {% csrf_token %}

         <div class="form-group">

            <input id="email" class="form-control inp usrnm" type="text" name="username" placeholder="Username" autocomplete="off" autofocus>

         </div>

         <div class="form-group">

            <input id="password" class="form-control inp pswd" type="password" name="password" placeholder="Password" autocomplete="off">

            <span class="star"><small>{{message}}</small></span>

         </div>

         <!-- <div class="g-recaptcha" data-sitekey="6Lfvib0kAAAAAHOzRQ2EdtYmDlSMDm3_CJ5NikSP"></div> -->

            <div class="g-recaptcha" data-sitekey="6Lc4e_0kAAAAALhzAszCaniQDW3aFE1c6nD-hH1F"></div>

            <div id="g-recaptcha-error"></div>

            <div class="form-check">

               <!-- <input type="checkbox" class="form-check-input" id="exampleCheck1"> -->
```

```
                <!-- <label class="form-check-label" for="exampleCheck1">Check me out</label> -->

                <span class="ml-auto"><a href="/forgotPassword/" style="margin-left: 10px;" class="forgot-pass">Forgot Password</a></span>

            </div>

            <!-- <span class="ml-auto"><a href="/forgotPassword/" class="forgot-pass">Forgot Password</a></span> -->

          <center>

            <input class="btn btn-primary" type="submit" value="Log in" style="margin-top: 10px; width: 95%;" disabled>

          </center>

       </form> <div>

          <center>

            Don't have an account?&amp;nbsp;&amp;nbsp;<a href="{% url 'register' %}">Sign Up</a>

          </center>

       </div></div>

   <script type="text/javascript">

      function verifyCaptcha() {

       document.getElementById('g-recaptcha-error').innerHTML = '';

      }

{% endblock %}
```

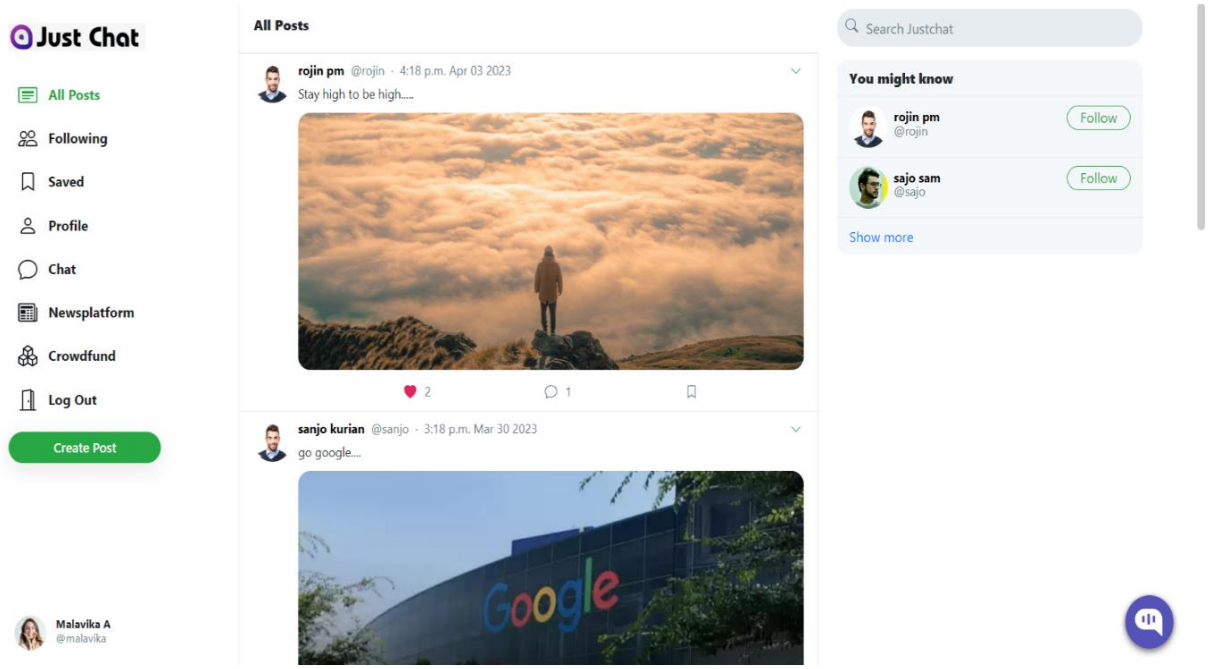
2.Smart_Contract:

```
// SPDX-License-Identifier: UNLICENSED
pragma solidity ^0.8.9;

contract CrowdFunding {
   struct Campaign {
      address owner;
      string title;
      string description;
      uint256 target;
      uint256 deadline;
      uint256 amountCollected;
      string image;
      address[] donators;
      uint256[] donations;
   }

   mapping(uint256 => Campaign) public campaigns;

   uint256 public numberOfCampaigns = 0;
```

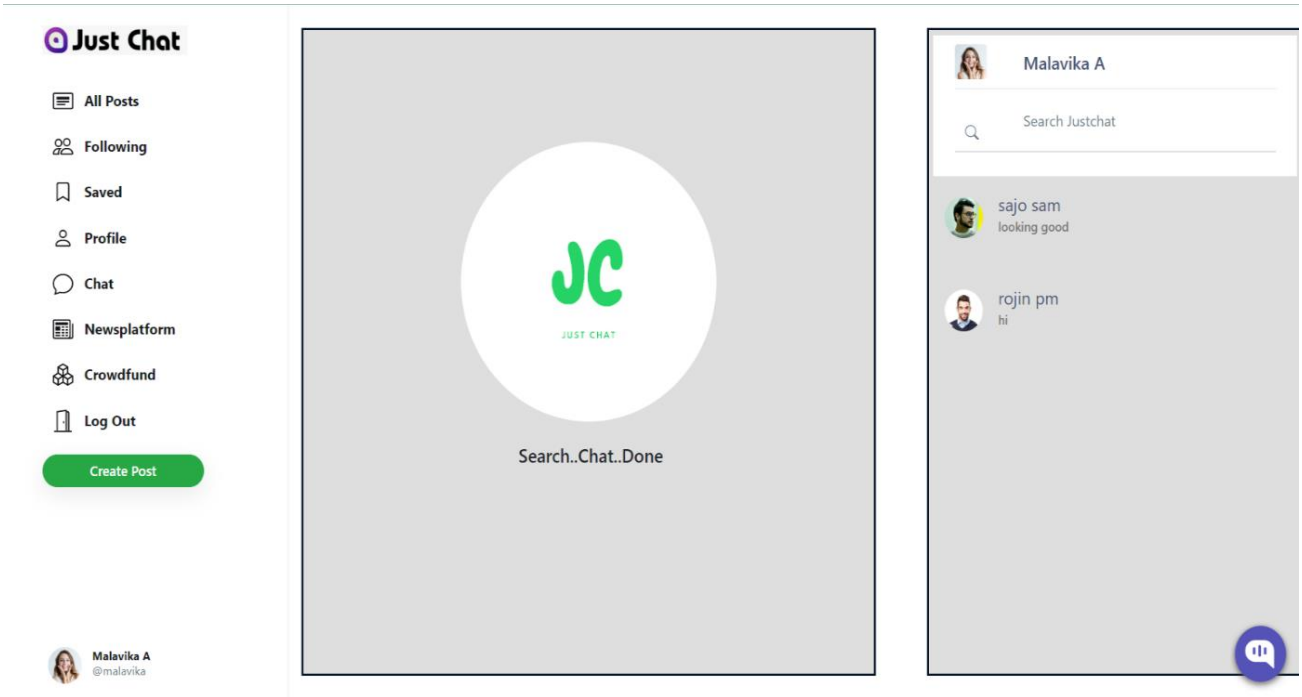
```
    function createCampaign(address _owner, string memory _title, string memory _description, uint256
_target,
     uint256 _deadline, string memory _image) public returns (uint256) {
        Campaign storage campaign = campaigns[numberOfCampaigns];

        require(campaign.deadline < block.timestamp, "The deadline should be a date in the future.");

        campaign.owner = _owner;
        campaign.title = _title;
        campaign.description = _description;
        campaign.target = _target;
        campaign.deadline = _deadline;
        campaign.amountCollected = 0;
        campaign.image = _image;

        numberOfCampaigns++;

        return numberOfCampaigns - 1;
    }

    function donateToCampaign(uint256 _id) public payable {
        uint256 amount = msg.value;

        Campaign storage campaign = campaigns[_id];

        campaign.donators.push(msg.sender);
        campaign.donations.push(amount);

        (bool sent,) = payable(campaign.owner).call{value: amount}("");

        if(sent) {
            campaign.amountCollected = campaign.amountCollected + amount;
        }
    }

    function getDonators(uint256 _id) view public returns (address[] memory, uint256[] memory) {
        return (campaigns[_id].donators, campaigns[_id].donations);
    }

    function getCampaigns() public view returns (Campaign[] memory) {
        Campaign[] memory allCampaigns = new Campaign[](numberOfCampaigns);

        for(uint i = 0; i < numberOfCampaigns; i++) {
            Campaign storage item = campaigns[i];

            allCampaigns[i] = item;
        }

        return allCampaigns;
    }
}
```

9.2 Screen Shots:

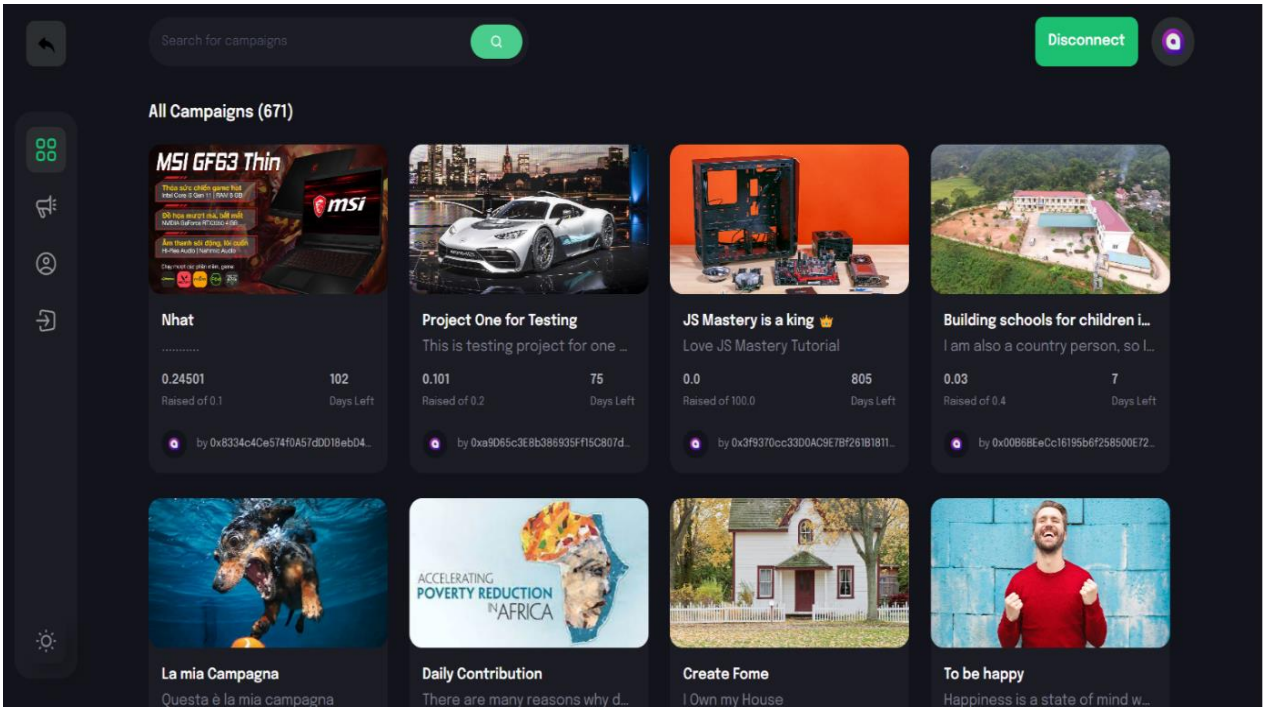
Home Page:



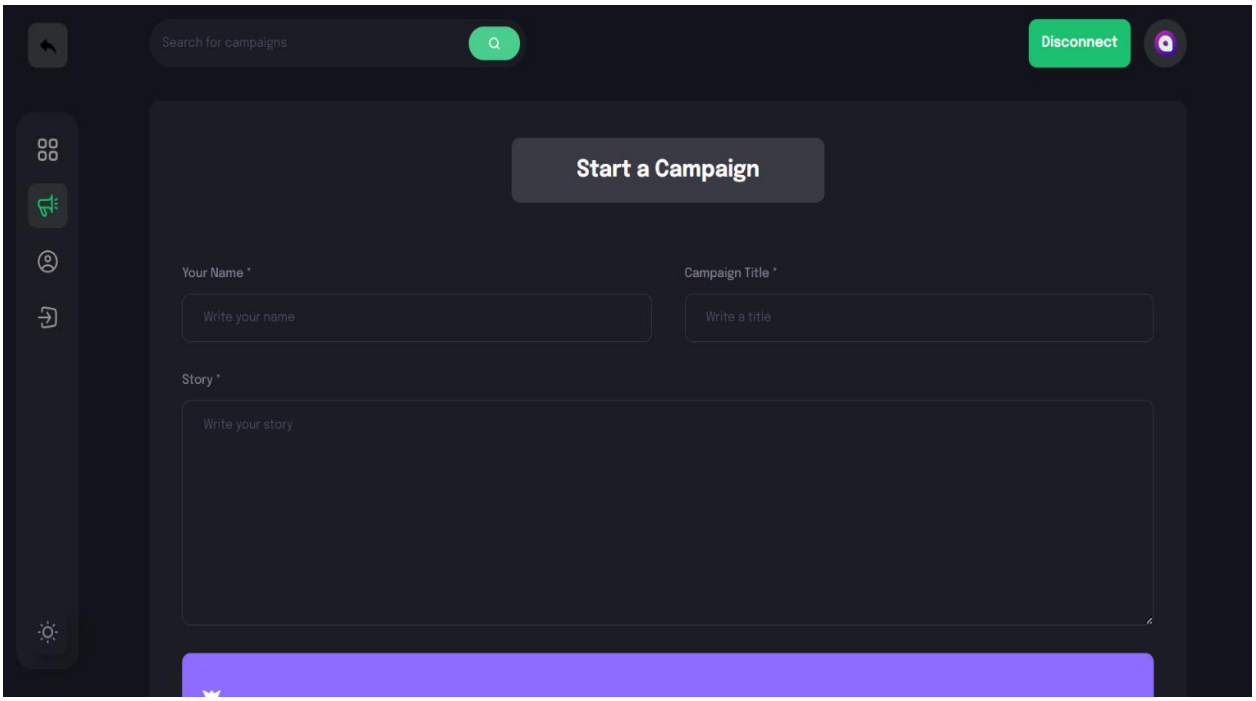
Chat Page:



Crowdfund page:



Donation to campaigns:



PAPER NAME

S4B_10_Justin v kalappura (2).pdf

WORD COUNT

11498 Words

CHARACTER COUNT

64677 Characters

PAGE COUNT

52 Pages

FILE SIZE

1.4MB

SUBMISSION DATE

May 23, 2023 4:07 PM GMT+5:30

REPORT DATE

May 23, 2023 4:08 PM GMT+5:30

23% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 14% Internet database
- Crossref database
- 22% Submitted Works database
- 4% Publications database
- Crossref Posted Content database

Excluded from Similarity Report

- Bibliographic material

23% Overall Similarity

Top sources found in the following databases:

- 14% Internet database
- Crossref database
- 22% Submitted Works database
- 4% Publications database
- Crossref Posted Content database

TOP SOURCES

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	slideshare.net Internet	2%
2	St. Ignatius High School on 2022-07-19 Submitted works	2%
3	St. Ignatius High School on 2022-07-19 Submitted works	2%
4	simplilearn.com Internet	1%
5	coursehero.com Internet	<1%
6	stackoverflow.com Internet	<1%
7	Singapore Institute of Technology on 2022-07-28 Submitted works	<1%
8	docs.huihoo.com Internet	<1%

9	College of Banking and Financial Studies on 2023-04-19 Submitted works	<1%
10	St. Ignatius High School on 2022-07-19 Submitted works	<1%
11	University of Makati on 2023-05-15 Submitted works	<1%
12	Southern New Hampshire University - Continuing Education on 2022-0... Submitted works	<1%
13	The State University of Zanzibar on 2023-01-31 Submitted works	<1%
14	iGlobal University on 2022-11-09 Submitted works	<1%
15	University of Greenwich on 2022-04-24 Submitted works	<1%
16	androidmegaflow.blogspot.com Internet	<1%
17	dspace.daffodilvarsity.edu.bd:8080 Internet	<1%
18	First City University College on 2023-05-04 Submitted works	<1%
19	Florida Institute of Technology on 2021-02-11 Submitted works	<1%
20	Perth Institute of Business and Technology (PIBT) on 2019-04-08 Submitted works	<1%

21	HELP UNIVERSITY on 2022-01-05 Submitted works	<1%
22	University of Sydney on 2021-05-13 Submitted works	<1%
23	The University of the South Pacific on 2016-03-24 Submitted works	<1%
24	CSU, San Jose State University on 2015-05-18 Submitted works	<1%
25	barabansheg.blogspot.com Internet	<1%
26	Sir George Monoux College on 2023-01-09 Submitted works	<1%
27	Colorado Technical University Online on 2016-10-29 Submitted works	<1%
28	Yaqiong He, Hanjie Dong, Huaiguang Wu, Qianheng Duan. "Formal Anal... Crossref	<1%
29	Sriwijaya University on 2022-02-21 Submitted works	<1%
30	University of East London on 2023-05-12 Submitted works	<1%
31	deped-pasig on 2023-05-23 Submitted works	<1%
32	qiita.com Internet	<1%

33	Kampala International University on 2019-12-12 Submitted works	<1%
34	jultika.oulu.fi Internet	<1%
35	Higher Education Commission Pakistan on 2016-12-15 Submitted works	<1%
36	Kampala International University on 2019-12-13 Submitted works	<1%
37	dev.to Internet	<1%
38	if-statement21.blogspot.com Internet	<1%
39	Middlesex University on 2010-03-01 Submitted works	<1%
40	National College of Ireland on 2023-04-19 Submitted works	<1%
41	University of Colombo on 2022-10-04 Submitted works	<1%
42	University of Hertfordshire on 2022-09-23 Submitted works	<1%
43	University of Sydney on 2021-05-13 Submitted works	<1%
44	City Unity College on 2023-04-06 Submitted works	<1%

45	Southampton Solent University on 2020-05-27 Submitted works	<1%
46	examtopics.com Internet	<1%
47	Camarines Sur Polytechnic Colleges on 2023-03-19 Submitted works	<1%
48	University of Canberra on 2006-09-05 Submitted works	<1%
49	University of Greenwich on 2016-01-05 Submitted works	<1%
50	docs.microsoft.com Internet	<1%
51	pt.scribd.com Internet	<1%
52	tudr.thapar.edu:8080 Internet	<1%
53	Asia Pacific International College on 2023-04-28 Submitted works	<1%
54	Berlin School of Business and Innovation on 2022-10-10 Submitted works	<1%
55	Glyndwr University on 2022-01-30 Submitted works	<1%
56	NCC Education on 2023-01-11 Submitted works	<1%

57	SRM University on 2022-05-05 Submitted works	<1%
58	University of Edinburgh on 2023-02-27 Submitted works	<1%
59	University of Greenwich on 2022-11-21 Submitted works	<1%
60	University of Sydney on 2023-04-26 Submitted works	<1%
61	higithub.com Internet	<1%
62	Glyndwr University on 2022-02-07 Submitted works	<1%
63	Asia Pacific University College of Technology and Innovation (UCTI) on... Submitted works	<1%
64	Federal University of Technology on 2019-11-22 Submitted works	<1%
65	University of Greenwich on 2021-04-01 Submitted works	<1%
66	gitlab.stud.idi.ntnu.no Internet	<1%
67	City University on 2020-08-06 Submitted works	<1%
68	Jeffrey Palermo. ".NET DevOps for Azure", Springer Science and Busin... Crossref	<1%

69	Queen Mary and Westfield College on 2023-05-09 Submitted works	<1%
70	University of Bedfordshire on 2021-09-05 Submitted works	<1%
71	University of Warwick on 2022-04-19 Submitted works	<1%
72	github.com Internet	<1%
73	studentsrepo.um.edu.my Internet	<1%
74	Colorado Technical University Online on 2015-06-25 Submitted works	<1%
75	Icon College of Technology and Management on 2020-07-11 Submitted works	<1%
76	KMD Computer Center on 2014-12-01 Submitted works	<1%
77	Tower Hamlets College on 2023-03-31 Submitted works	<1%
78	Universiti Sains Islam Malaysia on 2017-12-11 Submitted works	<1%
79	cse.anits.edu.in Internet	<1%
80	eml.ou.edu Internet	<1%

81	krex.k-state.edu Internet	<1%
82	thesai.org Internet	<1%
83	Birkbeck College on 2022-04-11 Submitted works	<1%
84	Otto-von-Guericke-Universität Magdeburg on 2017-10-26 Submitted works	<1%
85	Rivier University on 2022-05-10 Submitted works	<1%
86	University of West London on 2023-04-30 Submitted works	<1%
87	Anish Francis. "Investigation of Channel Capacity of Railway Power Lin... Crossref	<1%
88	CTI Education Group on 2018-08-22 Submitted works	<1%
89	Colorado Technical University Online on 2015-10-06 Submitted works	<1%