# 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 four users i.e., guest user, admin, normal user, news head 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. The major modules 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 continous suspects checking.

**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:**

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.

# CHAPTER 2

**SYSTEM STUDY**

#### 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

#### The ubiquitous presence of social networks has established novel modes of communication and behavior within contemporary society. Over the years, social networks have been frequently utilized by citizens and businesses, yet governments exhibit a consistent inclination towards incorporating novel communication technologies. Online social networking platforms, namely Facebook, Twitter, and Linked In, offer a platform for individuals to convene, predicated upon a range of influences, including pre-existing relationships, shared interests, or professional affiliations.

#### 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.

#### 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. So, instead of just preventing the usage and make them updated through the old method we make them updated through our proposed JUSTCHAT. 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.

#### ADVANTAGES OF PROPOSED SYSTEM

• Can make a major change with less effort.

• 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

### FEASIBILITY STUDY

#### The process of project management encompasses the planning, coordination, and appropriation of resources with the aim of attaining specific goals and objectives for a given project. A feasibility study is a preliminary inquiry conducted to evaluate the viability and feasibility of a potential project or endpoint in terms of its overall worth and suitability. The primary objective of conducting a feasibility study is to furnish an impartial evaluation of the various technical, economic, financial, legal, and environmental facets of a proposed project. The data provided can subsequently enable decision-makers to discern whether to advance the project or refrain from doing so. The results of the feasibility analysis may be deployed to create an implementable project blueprint and corresponding financial plan. Determining the viability of a proposed project is a complex task that necessitates the conduction of a feasibility study, ascertaining the project's worthiness and its feasibility for pursuit. A comprehensive evaluation was conducted during the feasibility analysis of this project, with due consideration given to several critical areas including technical, economic and operational feasibility.

#### 3.1.1 Economical Feasibility

#### In order to facilitate the development of the system, it is imperative to conduct comprehensive cost and benefit analyses. The establishment of criteria aimed at ensuring that emphasis is placed on the endeavor that is most likely to render optimal outcomes and prompt returns. The cost associated with the creation of a new system is among the variables under consideration. Several noteworthy financial inquiries were raised during the preliminary inquiry process, such as the following:

#### The system in question has been conceptualized and subsequently developed in the context of a project, and as such, no manual expenditure is required for its implementation.

#### 1. What is the cost associated with the procurement of hardware and software?

#### 2. Moreover, it should be noted that all resources have already been made accessible.

#### 3.1.2 Technical Feasibility

Prior to evaluating the system, it is imperative to conduct a thorough technical assessment. The assessment of feasibility hinges on the fundamental basis of the system's requirement outline, particularly its input, output, programs, and procedures. In an academic context, it is essential to use sophisticated language and structure to enhance clarity and precision, allowing for a more comprehensive understanding of the content. After establishing an outline, it is imperative to perform further inquiry to ascertain the requisite equipment type. 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

#### Behavioral Feasibility

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 prove advantageous as it effectively fulfills the designated objectives upon its successful development and installation. Upon careful consideration of all behavioral factors, it has been determined that the project is behaviorally viable.

#### 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 know the news update.

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.

### SYSTEM SPECIFICATION

#### Hardware Specification for the project:

Processor needed - Intel core i5

RAM needed - 8 GB

Hard disk needed - 512 SSD

#### Software Specification for the project:

Front End - HTML, CSS, React

Back End - POSTGRE SQL,

Client on PC - Windows 10 and above.

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

### SOFTWARE DESCRIPTION

#### 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. Additionally, Python has a vibrant and helpful community that frequently creates new libraries and tools, which makes it simpler for developers to find answers to their issues and keep up with the most recent trends and best practises.

#### MySQL

MySQL, the premier open-source SQL database management system, is developed, distributed, and backed by Oracle Corporation. The official website of MySQL serves as a primary source of up-to-date information on MySQL software.

The MySQL is a type of software that facilitates the management and organization of an enterprise's data through the administration of databases.

A database comprises an organized and systematic assembly of information. It encompasses a wide spectrum of data ranging from the rudimentary compilation of grocery items to the intricate assembly of a visual commemoration, or even the vast reservoir of knowledge traversing a commercial enterprise's information system. In order to efficiently handle data retrieval, insertion and manipulation of computerized data sets, it is essential to have a specialized software application known as a Database Management System (DBMS). MySQL Server is one such system that enables efficient access and manipulation of data stored in a computer database. Given their exceptional capacity to handle vast quantities of data, database management systems hold a commanding position in the realm of computing, functioning either as self-sufficient utilities or as integral components of other applications.

The relational nature of MySQL databases is a defining characteristic.

A relational database employs a table structure approach for storing data instead of a singular, undifferentiated repository. The database structures are systematically arranged within physical files that are meticulously optimized for enhanced speed. The logical model presents a malleable programming setting through the presence of various entities such as databases, tables, views, rows, and columns. Rules that govern relationships among various data fields, such as one-to-one, one-to-many, unique, required, or optional, as well as pointers between different tables, are established. This is accomplished by setting up regulations that define the manner in which data fields interact. The database implements regulations to ensure that in the case of a well-structured database, the application will not encounter any instances of contradictory, replicated, abandoned, outdated, or absent data. The acronym "MySQL" implies the involvement of the Structured Query Language (SQL). SQL is the predominant standardized language employed for database access. Depending on the specific programming environment employed, one may engage in the direct input of SQL commands, such as for report generation purposes, integrate SQL statements within code written in a different programming language, or employ a language-specific Application Programming Interface (API) that obscures the underlying SQL syntax. The definition of SQL derives from the ANSI/ISO SQL Standard. The SQL standard has exhibited a continuous process of development since 1986, yielding multiple iterations. In the present document, the term "SQL92" pertains to the standard version that was released in 1992. Similarly, "SQL: 1999" denotes the version of the standard that was launched in 1999, while "SQL: 2003" refers to the present iteration of the standard. The term "the SQL standard" is understood to denote the most up-to-date iteration of the SQL Standard during a given timeframe within scholarly discourse.

The MySQL software is an open-source application.

The concept of Open Source pertains to the unrestricted accessibility and malleability of computer software, thereby permitting any individual to employ and alter it as deemed necessary. The MySQL software can be procured free-of-charge from the Internet, enabling it to be utilized by any individual sans payment. Should you desire, the source code may be studied and customized accordingly to fulfill one's requirements. The MySQL software employs the GNU General Public License (GPL) to regulate permissible actions and restrictions regarding the utilization of the software under varying circumstances. If the General Public License (GPL) makes one feel uneasy or if it becomes necessary to incorporate MySQL code into a commercially-oriented application, then the option of purchasing a commercially licensed version from our organization is available. Please refer to the "MySQL Licensing Overview" document for further details.

The MySQL Database Server is widely acclaimed for its remarkable speed, unwavering reliability, exceptional scalability, and user-friendliness.

If such is the pursuit that incites one's interest, it is recommended to execute an attempt. The MySQL Server has the capacity to operate smoothly on a personal computer or portable computer, in conjunction with other software applications, online servers, and related systems, with minimal or negligible need for maintenance. If one allocates a complete computing system for MySQL, it is feasible to optimize the parameters in order to harness the entirety of the system's memory, processing unit potential, and input/output bandwidth capabilities.

The MySQL Server operates within the context of client/server architectures, as well as in embedded systems.

The MySQL Database Software is a sophisticated client/server system, comprised of a highly efficient and scalable multithreaded SQL server that provides support for diverse backends, various client programs and libraries, administrative tools, as well as a plethora of application programming interfaces (APIs). Furthermore, our services encompass the supply of MySQL Server as an embedded multi-threaded library, which can be efficiently incorporated into your application, resulting in a streamlined, expedient, and more manageable autonomous offering.

#### DJANGO

Django is a high-level web framework for building web applications using the Python programming language. It follows the model-view-controller (MVC) architectural pattern, emphasizing the principle of "Don't Repeat Yourself" (DRY). Overall, Django simplifies and accelerates the process of building web applications by providing a comprehensive set of tools, conventions, and best practices, allowing developers to focus on the application's logic rather than the underlying infrastructure.

#### WSGI

A WSGI server in Django acts as a middle layer between the web server and the Django application. It receives HTTP requests from the web server, translates them into a standardized format (WSGI), and forwards them to the Django application for further processing. Similarly, it takes the response generated by the Django application, converts it into the appropriate HTTP format, and sends it back to the web server to be delivered to the client.

#### ASGI

ASGI (Asynchronous Server Gateway Interface) is a specification that allows Django to handle both synchronous and asynchronous web applications. In the context of Django, an ASGI server is responsible for handling incoming HTTP requests and passing them to the Django application, which can handle them asynchronously if configured to do so.

**3.3.4 BLOCKCHAIN:**

Blockchain is a decentralized and distributed ledger technology that allows multiple parties to maintain a shared record of transactions in a secure and transparent manner. While Django is a web framework for building web applications, it can be used in combination with blockchain technology to develop applications that leverage the benefits of blockchain.

Here's a short description of how blockchain can be integrated with Django:

1. Smart Contracts: Blockchain platforms like Ethereum allow the execution of smart contracts, which are self-executing contracts with predefined rules and conditions. Django can be used to build web interfaces that interact with smart contracts deployed on a blockchain network. These interfaces can enable users to interact with the blockchain, execute transactions, and view contract states.
2. Decentralized Applications (DApps): Django can be utilized to build decentralized applications (DApps) that leverage blockchain technology. DApps are applications that run on a peer-to-peer network of computers, where the blockchain acts as the underlying infrastructure. Django can handle the front-end and back-end development of the DApp, while the blockchain network stores and validates the data.
3. Data Storage and Authentication: Django can be used to create applications that store data on the blockchain. The blockchain provides an immutable and tamper-proof data storage mechanism, ensuring the integrity and security of the data. Django can also handle user authentication and authorization for accessing the blockchain-based application.
4. Blockchain Integration: Django can be integrated with various blockchain networks and protocols, such as Ethereum, Hyperledger, or Bitcoin. This integration allows Django applications to interact with the blockchain, retrieve and submit data, and perform transactions.
5. Blockchain-based Identity Systems: Django can be used to build identity systems that leverage blockchain for secure identity management. By using blockchain's decentralized nature and cryptographic techniques, Django applications can implement robust identity verification and authentication mechanisms.

It's important to note that while Django can facilitate the development of blockchain applications, the core blockchain functionality and infrastructure are typically provided by the underlying blockchain platform or protocol. Django serves as a framework to build the application layer on top of the blockchain technology, enabling developers to create user-friendly interfaces and business logic.

# CHAPTER 4 SYSTEM DESIGN

### INTRODUCTION

Design represents the inaugural stage in the development process of any engineered product or system. The act of designing involves the utilization of creativity in the process. Effective system performance is strongly contingent upon the integration of a sound design. The concept of "design" is explicated as the procedure of utilizing diverse methodologies and precepts in order to outline a process or system with adequate specificity and detail to enable its material manifestation. The present concept may be explicated as a systematic procedure of implementing diverse methodologies and fundamental precepts with the ultimate aim of precisely delineating a specific device, process or system such that its material embodiment can be achieved. At the technical core of the software engineering process lies software design, which is employed independent of the developmental paradigm employed. The process of system design is integral to the development of a detailed architecture needed for construction of a given system or product. Like any systematic approach, this software has undergone a meticulous design phase focused on optimizing its efficacy, performance, and precision. The design phase involves a shift from a document that caters to the needs and preferences of the users towards a document that is more tailored to the programming and database personnel. The design of a system involves a two-stage development process consisting of distinct phases, namely the logical and physical design.

### UML DIAGRAM

The Unified Modeling Language, commonly referred to as UML, is a universally recognized standard language employed in the specification, visualization, construction, and documentation of software system artifacts. The Object Management Group (OMG) developed the Unified Modeling Language (UML), with its initial iteration - UML 1.0 specification draft - being submitted to the OMG in January 1997. The Unified Modeling Language (UML) is a visual representation tool utilized for the creation of software architectural plans. The Unified Modeling Language (UML) serves as a versatile visual modeling tool that strives to facilitate the holistic representation, definition, development, and documentation of various software systems. Unified Modeling Language (UML) is not classified as a programming language, but it serves as a valuable instrument for code generation in multiple programming languages by applying UML diagrams. The Unified Modeling Language (UML) exhibits a significant correlation with the domain of object-oriented analysis and design. In order to create a comprehensive UML diagram, various components and their associated dependencies are employed, resulting in a holistic representation of the relevant system. The paramount aspect of the entire process is the visual element portrayed by the UML diagram. The integration of all other constituent elements is imperative for the attainment of comprehensiveness.

The Unified Modeling Language (UML) encompasses a collection of nine distinct diagrams, which are widely recognized and utilized in the field of software engineering.

The class diagram is a graphical representation that models the structural organization of a system by illustrating the types of objects, attributes, and behaviors that are linked through relationships within the system. It is commonly utilized in the field of software engineering to aid in the design and analysis of a system's structure. By depicting the various classes within the system and their associations with one another, a class diagram provides a high-level view of the system that facilitates communication and collaboration among design team members.

The object diagram is a widely used diagrammatic representation in computer science and software engineering. It is commonly used to depict a specific instance or snapshot of an object-oriented system, illustrating the objects' relationships and interactions at a particular point in time. This diagram serves as a valuable tool for software developers, aiding in the understanding and visualization of complex software systems, and facilitating effective communication between team members. As such, the object diagram holds significant importance in the field of computer science and software engineering.

The utilization of a diagram intended for showcasing the system requirements and interactions, referred to as a use case diagram, is commonly observed in various industries and fields. Its purpose is to depict the potential user interactions with a system and demonstrate the possible outcomes of such interactions. The outlines of actors, use cases, and their associations are illustrated in a use case diagram, making it a useful tool for system development and analysis.

### USE CASE DIAGRAM

A use case diagram represents the various functionalities and interrelationships of a system's components through a visual medium. The utilization of a use case represents a systematic approach employed in the realm of system analysis, designed to effectively identify, clarify, and organize system requirements. Within this particular context, the designation "system" pertains to an entity that is currently undergoing development or is in operation, such as a web-based platform designed to facilitate the sales and provision of services associated with mail-order products. The utilization of use case diagrams is widespread in the Unified Modeling Language (UML) - a widely accepted and standardized notation that facilitates the modeling of physical entities and systems. The objectives of a system may encompass a range of tasks, such as strategizing overarching requirements, validating the design of hardware components, testing and identifying defects in software products that are undergoing development, generating a comprehensive online reference to provide assistance to users, or fulfilling service-oriented responsibilities aimed at accommodating the needs and preferences of consumers. An instance of employment scenarios in the context of selling a product would pertain to actions such as the request of an item, the refinement of a product catalog, the handling of payment procedures, and the maintenance of interactions with the consumers. A use case diagram comprises four fundamental components, as stipulated by established academic conventions.

The boundary delimitates the system under investigation with respect to the environment in which it operates.

The individuals involved in a given system are commonly referred to as actors, and they are typically defined according to the specific roles that they fulfill within said system.

The utilization scenarios, which delineate the precise functions undertaken by the actors both within and in conjunction with the system. The interrelationships among the actors and the utilization cases are of paramount importance.

Use case diagrams are typically created to capture and represent the functional specifications and requirements of a given system. Upon identification of the aforementioned components, it is imperative to adhere to a set of established principles in order to construct a highly effective use case diagram. The nomenclature of a utilization instance holds significant importance within the domain of academic writing. The selection of a name ought to be conducted in a manner that ensures its ability to accurately distinguish and characterize the various tasks and capabilities executed. A suitable nomenclature for actors is required. Clearly indicate the relationships and dependencies in the diagram. Avoid attempting to comprehensively incorporate all forms of relationships, for the principal objective of the diagram is to ascertain the necessary provisions.

It is recommended that notes be utilized as necessary to explicate pertinent information.

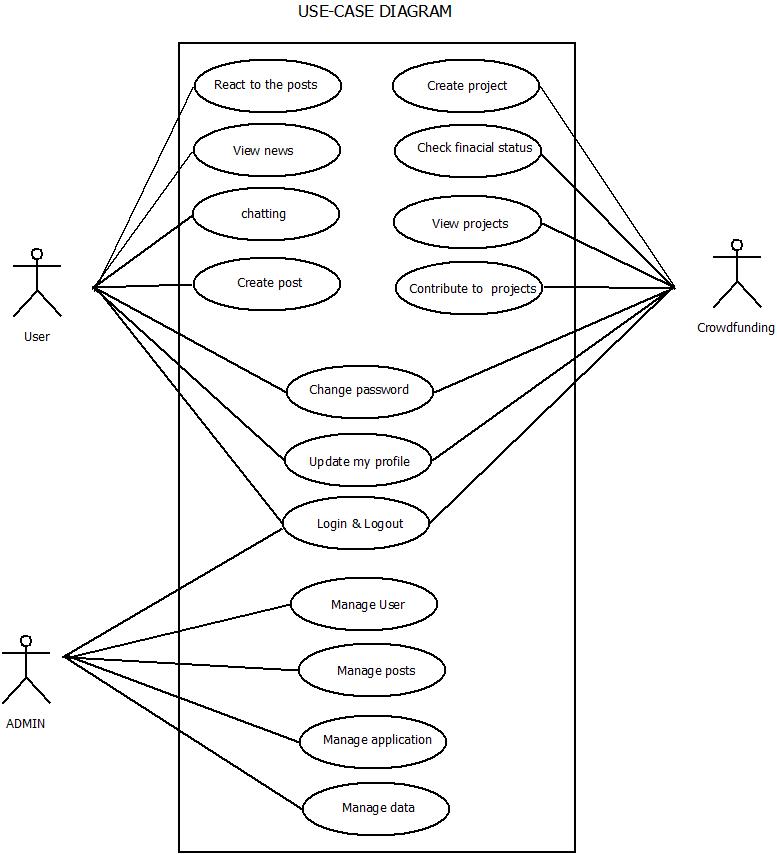


Fig 1: Use case diagram for ‘JUSTCHAT ‘ System

### 4.2.2 SEQUENCE DIAGRAM

A arrangement chart speaks to the successive interaction among objects. The arrangement of occasions in which these intuitive happen. The arrangement graph can be on the other hand alluded to as occasion charts or occasion scenarios. Arrangement charts outline the working of the objects inside a framework by delineating the particular arrange in which they work. The utilization of charts for the reason of recording and comprehending necessities relating to novel or show frameworks may be a predominant hone among business visionaries and software developers.

The image utilized within the representation of arrangement graphs is alluded to as the Arrangement Chart Image. It is basic for understudies to create solid think about propensities in order to succeed in their scholarly interests. In this manner, it is prescribed to set up a reliable consider plan, looking for bolster from teachers or mentors, and utilizing successful ponder techniques such as taking breaks and dynamic engagement with the fabric. Based on inquire about, students' scholastic execution is profoundly connected with their consider propensities and behaviors. Hence, contributing time and exertion into creating compelling think about propensities can result in made strides scholarly results. In UML graphs, the performing artist component indicates a particular sort of part that locks in in intuitive with the framework and its different objects. It is critical to recognize that the part of an on-screen character is reliably situated exterior the domain of the framework we proposed to speak to through the utilization of the UML graph. On-screen characters are utilized for the reason of depicting differing personalities, including human clients and other unessential substances. In UML charts, the portrayal of an performing artist is ordinarily done through the utilization of a adhere figure documentation.

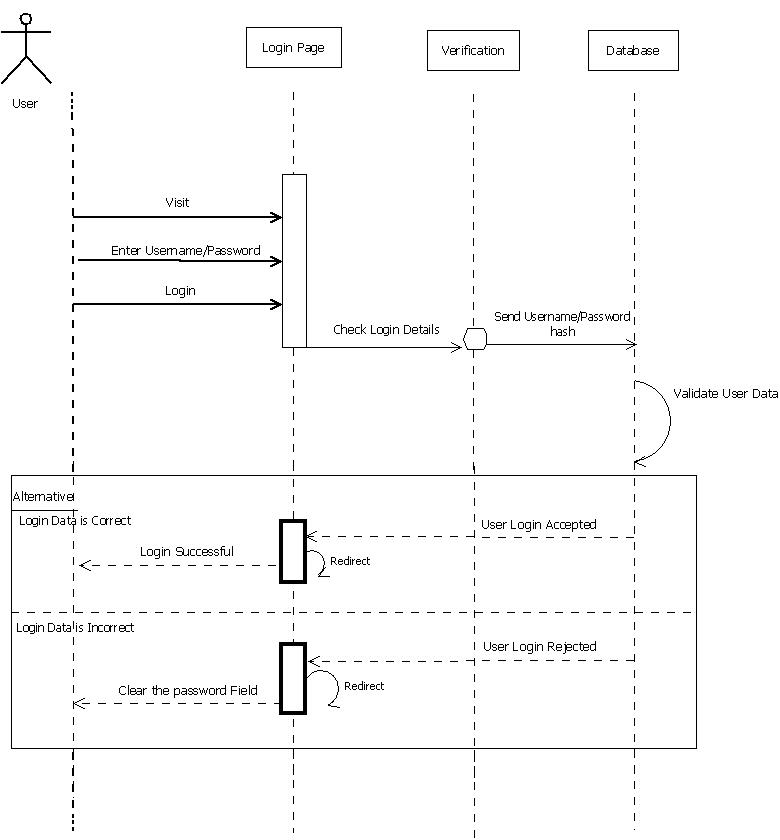
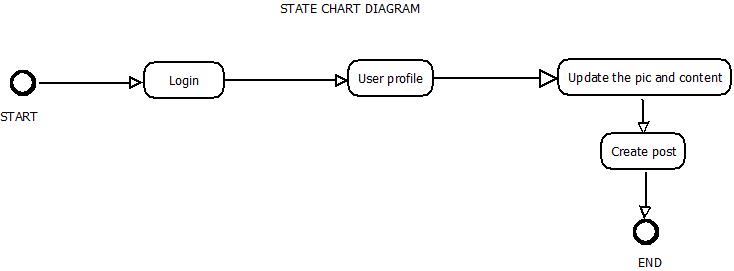


Fig 2: Sequence diagram for ‘JUSTCHAT’ Login page

### State Chart Diagram

A State Chart Graph could be a sort of UML (Bound together Modeling Dialect) graph utilized to demonstrate the behavior of a framework or protest over time. A State Chart Chart appears the different states that an question or framework can be in, and how it moves from one state to another. Each state speaks to a specific condition or mode that the protest or framework can be in, and may have related activities or behaviors. Moves between states are activated by occasions, which may be outside (such as a client input) or inner (such as a timer).A move may too have a protect condition, which must be genuine for the move to happen. State Chart Graphs can be utilized to demonstrate complex frameworks with numerous states and moves, such as client interfacing, control frameworks, or trade forms. State Chart Charts can moreover appear progressive states, where a state may contain sub-states with their possess moves and behaviors.

Fig 1: State chart diagram for ‘JUSTCHAT’



### Activity Diagram

Activity diagrams illustrate the interconnectedness of distinct levels of activity abstraction in facilitating a service. It is customary for an event to entail a series of activities, especially when said activities aim to achieve multiple distinct objectives that necessitate coordination. A typical prerequisite involves elucidating the interplay among events within a given use case, particularly in scenarios where operations may coincide and necessitate synchronization. The utilization of this method may additionally serve the purpose of demonstrating the interconnectivity between a set of correlated use cases that exemplify the operational functionalities of a given business entity.

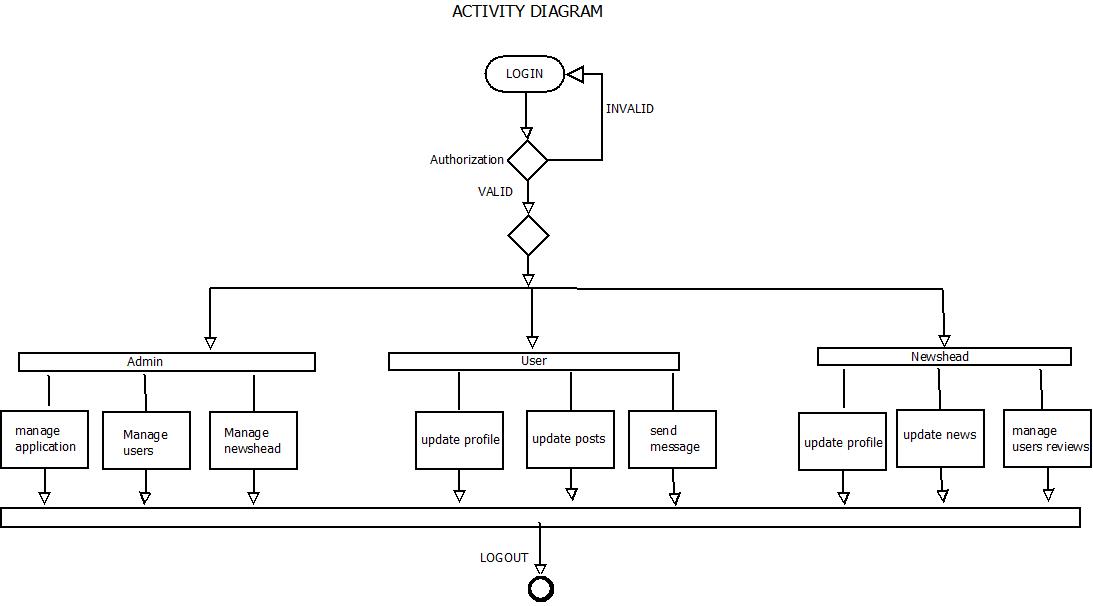


Fig 3: Activity diagram for ‘JUSTCHAT’

### Class Diagram

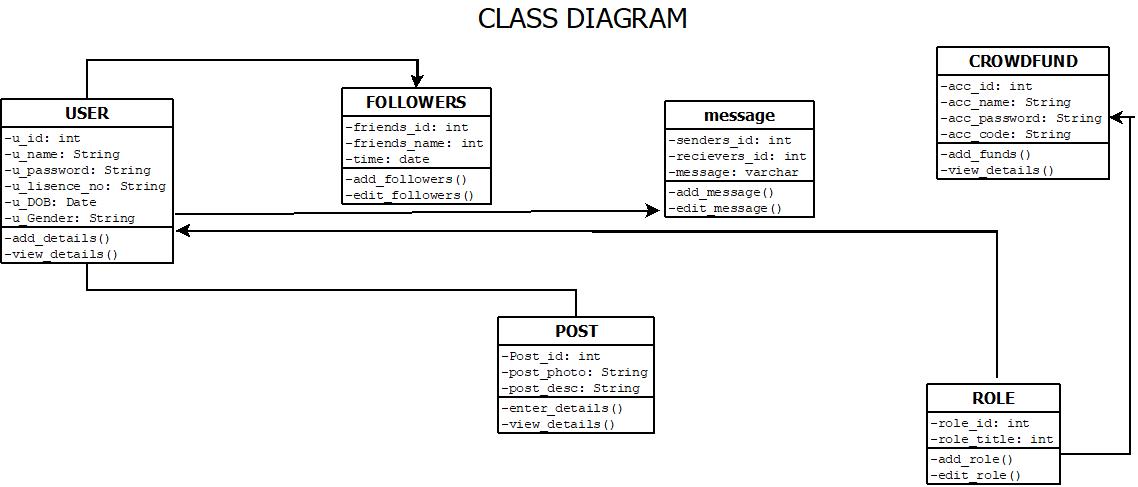
The lesson graph can be classified as a inactive chart, which shows the structure of a framework through the portrayal of classes, their qualities, and interconnects. This signifies the application's inactive point of view. Course graphs are a profitable device for the purposive visualization, depiction and documentation of differing framework constituents, in conjunction with encouraging the composition of executable code for computer program applications. A portrayal of the confinements forced on a framework, beside the properties and functionalities of a course, is displayed through the utilization of a lesson chart. Course charts are the sole UML graphs that can be direct translated into object-oriented programming dialects. These charts are broadly utilized within the prepare of planning object-oriented frameworks. A assorted collection of classes, interfacing, affiliations, collaborations, and imperatives are delineated in a representation of course chains of command. The previously mentioned phrasing, i.e., "auxiliary chart," is similarly utilized in this respect. 

Fig 4: Class diagram for ‘JUSTCHAT’

### Object Diagram

One can perceive a chart as an exemplification or an instance of graphically depicted data. The fundamental principles underlying both art and artistic expressions remain identical. The diagram depicts a static aspect of the system; however, it captures a momentary snapshot of the system state at a specific point in time. The utilization of a diagram serves to exemplify a collection of entities and their interconnectedness. It is imperative to have a clear comprehension of the intended utilization of the image. Diagrams are employed in a similarly analogous manner as that of charts. The disparity between the two entities lies in the fact that the diagram typically serves as a conceptual depiction of classes and their interconnections. However, it should be noted that a painting serves as a specimen of a specific era and possesses inherent authenticity. Thus, it can be inferred that the graph exhibits a strong resemblance to the behavior of the system in question. The objective is to obtain a stagnant portrayal of the system at a designated moment.

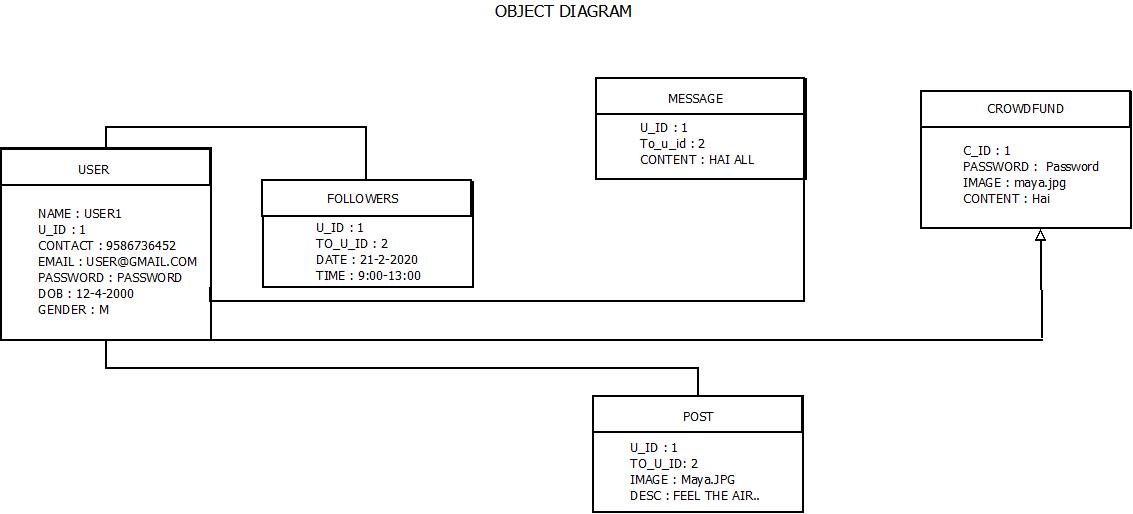


Fig 5: Object diagram for ‘JUSTCHAT’

### Component Diagram

Component diagrams exhibit divergent characteristics and idiosyncrasies. Component diagrams are utilized to symbolize the physical aspects of the system. A number of tangible objects, such as executables, libraries, files, documents, and various other items, can be found within a given node. Component diagrams serve as a graphical representation that elucidates the interconnection and arrangement of the various constituent parts within a given system. These diagrams can potentially serve as a means for constructing operational systems.

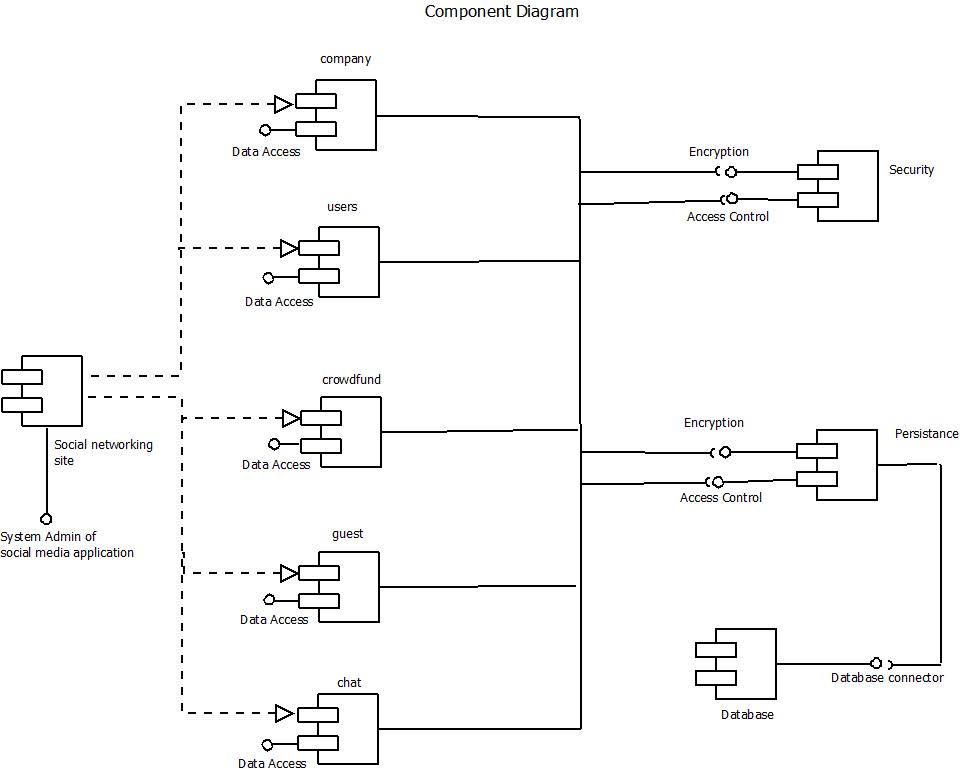


Fig 6: Component diagram for ‘CAREHAND’

### Deployment Diagram

The deployment diagram, which is a variant of the Unified Modeling Language (UML), portrays the system's execution architecture, comprising nodes such as hardware and software execution environments, as well as the intermediate layer that connects them. Deployment diagrams are frequently employed in order to depict the tangible components of a system, including both its hardware and software. Through utilization of this tool, one is able to gain a comprehensive understanding of the physical delivery of the system by the hardware. In comparison to other categories of Unified Modeling Language (UML) diagrams that predominantly portray the rational elements of a system, deployment diagrams aid in expounding the hardware configuration of a system.

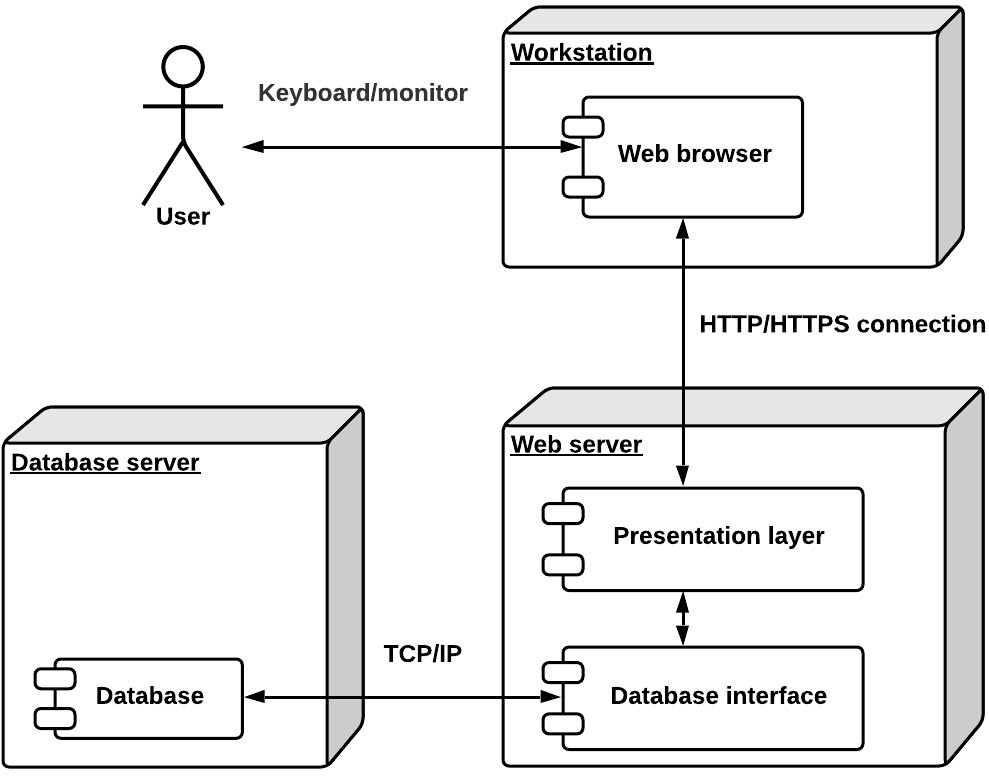
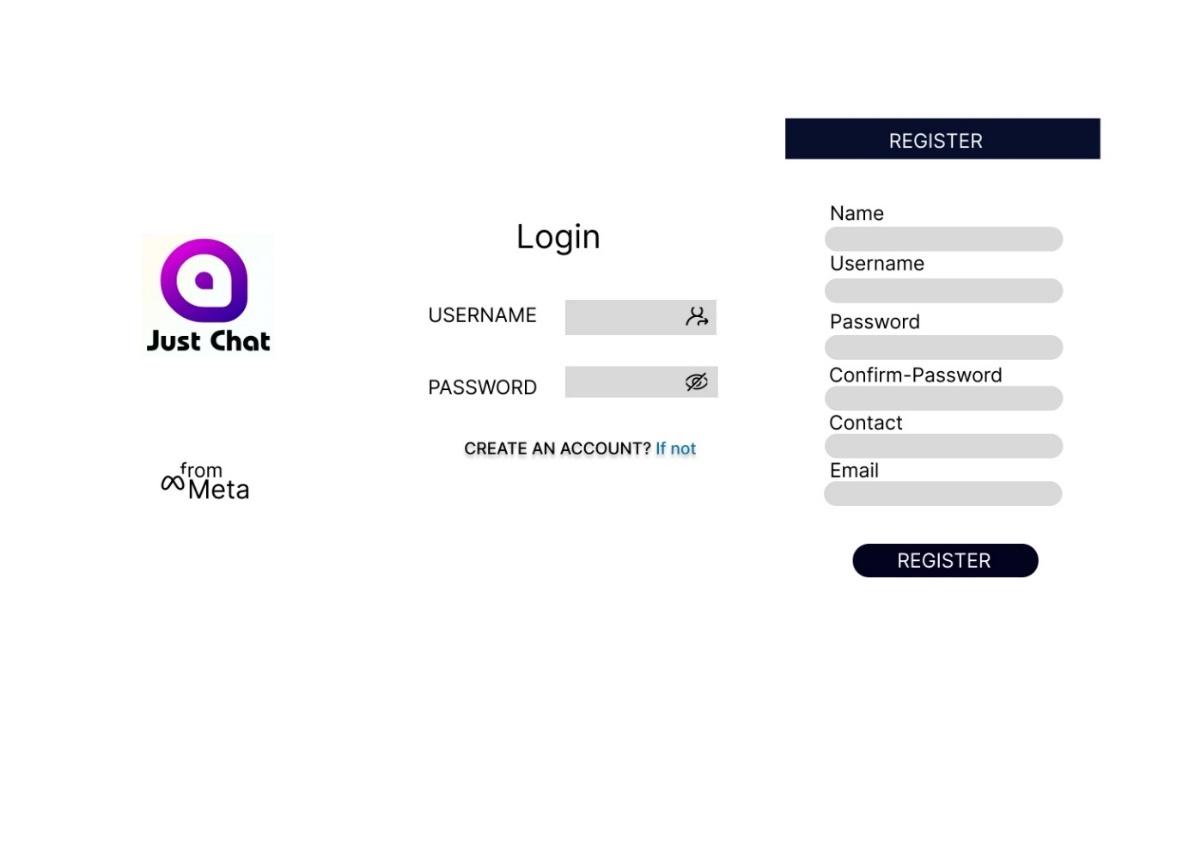


Fig 7: Deployment diagram for ‘JUSTCHAT’

#### USER INTERFACE DESIGN USING FIGMA

**Front Page, Login Page, Register Page:**



**Guest Home Page:**

#### 

#### User Home Page:

#### 

### DATABASE DESIGN

A database is a system that possesses the capability to store information and subsequently retrieve the stored data from users in both quantity and quality. The preservation and safeguarding of data are indispensable objectives for any database. The establishment of a database involves a dual-phase process. In order to develop a document that satisfactorily fulfills the user's requirements, it is imperative to comprehensively delineate their specific demands. The mechanism of data-level creation is distinct from that of every database management system (DBMS). The particularities regarding the specific DBMS that shall be employed are incorporated within the physical layer design process. The process of system design and database design are interdependent and act as complementary in nature. The informational structure of a database fulfills two fundamental objectives in a scholarly setting.

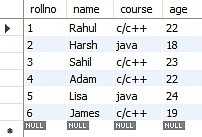
#### Relational Database Management System (RDBMS)

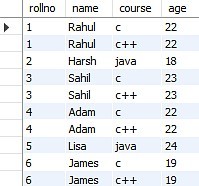
Data is represented via a relational model as a set of relationships. Every relationship is like a table of values or datasets. In relational model terminology, rows are called tuples, column headings are called attributes, and tables are relations. A relational file consists of tables, each of which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

#### 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 define its value. All values in a relationship are atomic and cannot be parsed.

#### 1st Normal Form (1NF)

"If a table has a degree of atomicity that equals 1, it is commonly referred to as being in the first normal form."Atomicity is a fundamental principle that stipulates that in this particular case, a singular cell is incapable of accommodating multiple values. The inclusion of a sole attribute with a singular value is limited to this context.The prohibition of the multi-valued attribute, composite attribute, as well as their amalgamation, by the first normal form is a established principle in academic writing. E.g. 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. The outcome of applying the First Normal Form to the preceding table is the table below.

By applying the First Normal Form, you achieve atomicity, and also every column has unique values.

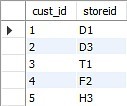
#### Second Normal Form (2NF)

The first requirement for a table to be in Second Normal Form is that it must be in First Normal Form; 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.

The primary key for the Location database is a composite cust ID called storied. The location of the store is a non-key feature. In this instance, the primary key's field storied is the sole factor affecting store location. This table thus fails to conform to the second normal form.

You must divide the table into two pieces in order to get it to Second Normal Form. You will then

have the tables below:

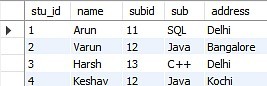


The column store location is completely dependent on the main key of that table, storied, as you have removed the partial functional reliance from the location table.

#### Third Normal Form (3NF)

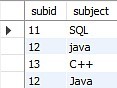
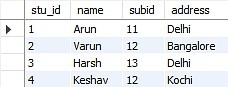
In order for a table to be in the third normal form, it is a prerequisite that the table is in the second normal form. The second normal form dictates that non-prime attributes that do not form part of the candidate key should not exhibit dependencies on other non-prime attributes. This requirement further specifies that no transitive dependencies should exist for such non-prime attributes. Consequently, a transitive dependency can be defined as a functional relationship between two attributes, in which attribute A indirectly determines attribute C, via intermediary attribute B, where the functional dependency between attributes B and A does not exist.

The third Normal Form (3NF) ensures the minimization of data duplication. The preservation of the accuracy and consistency of data is ensured by its utilization. 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:



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.

#### Sanitization

#### The process of data sanitization encompasses a deliberate and permanent removal or erasure of data from a storage device, with the ultimate goal of ensuring its irrevocable elimination and subsequent unavailability for restoration.

#### Typically, when data is deleted from storage media, the process does not completely erase the information, thereby exposing it for recovery by an unauthorized individual who may have gained access to the device. This issue can potentially engender critical inquiries regarding security and confidentiality of data. The process of sanitization entails purging storage media to ensure complete eradication of any lingering data on the device, rendering it unobtainable by advanced forensic methodologies.

#### Indexing

The deployment of indexing has proven to be an effective solution for enhancing the performance of a database by reducing the frequency of disc accesses required during query completion. A widely employed method in data structuring for rapid data retrieval and access within a database is database indexing. Multiple database columns are utilized in the creation of indices.

The initial column of the table, denoted as the search key, comprises a replicated copy of the primary key or potential primary key. The arrangement of these values in a sorted order facilitates the retrieval of the corresponding data with ease.

It should be noted that the preservation of sorted order in the data is variable and may not be present.

The location of the disc block containing the designated key value is retained by a sequence of pointers present in the second column, identified as the Data Reference or Pointer.

#### TABLE DESIGN

1. **Profileapp\_user:**

Primary key: **id**

Description: User Registration

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Key constraints** | **Data type** | **Size** | **Description** |
| 1 | id | Primary key | Int | 11 | id |
| 2 | password | Not null | int | 11 | password |
| 3 | last\_login | Not null | varchar | 25 | Last\_login |
| 4 | is\_superuser | Not null | varchar | 50 | Checking superuser |
| 5 | username | Not null | int | 10 | name |
| 6 | first\_name | Not null | varchar | 25 | first\_name |
| 7 | last\_name | Not null | varchar | 25 | last\_name |
| 8 | email | Not null | int | 11 | email |
| 9 | is\_staff | Not null | varchar | 25 | Checking staff |
| 10 | is\_active | Not null | varchar | 300 | Checking whether active |
| 11 | date\_joined | Not null | int | 25 | date |
| 12 | profile\_pic | Not null | varchar | 100 | profile image |
| 13 | cover | Not null | varchar | 100 | cover |
| 14 | country | Not null | varchar | 25 | country |
| 15 | gender | Not null | varchar | 25 | gender |
| 16 | phone | Not null | int | 25 | phone |

1. **Chat\_message:**

Primary key: id

Description: Chat

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Key constraints** | **Data type** | **Size** | **Description** |
| 1 | id | Primary key | int | 11 | Id of message |
| 2 | message | Not null | int | 11 | content |
| 3 | timestamp | Not Null | int | 11 | date |
| 4 | recipient\_id | Not Null | Not Null | 25 | Active/Not |
| 5 | sender\_id | Not Null | Not Null | 25 | Active/Not |

1. **Profileapp\_comment:**

Primary key: **id**

Description: Comment table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Key constraints** | **Data type** | **Size** | **Description** |
| 1 | id | Primary key | Int | 11 | Id for Manager |
| 2 | comment\_content | Not null | Int | 11 | Id to Admin |
| 3 | comment\_time | Not null | Varchar | 25 | Name of Manager |
| 4 | commenter\_id | Not null | Varchar | 25 | Date of birth |
| 5 | post\_id | Not null | Varchar | 25 | Gender |

1. **Profileapp\_followers:**

Primary key: **id**

Description: followers table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Key constraints** | **Data type** | **Size** | **Description** |
| 1 | id | Primary key | int | 11 | RC allocate id |
| 2 | user\_id | Not null | int | 11 | Manager Id |

1. **Profileapp\_follower\_followers:**

Primary key: **id**

Description: Information about relief center

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Key constraints** | **Data type** | **Size** | **Description** |
| 1 | id | Primary key | int | 11 | Id if Relief center |
| 2 | followers\_id | Not null | Varchar | 25 | Name of relief center |
| 3 | user\_id | Not null | Varchar | 25 | Place |

1. **Profileapp\_post:**

Primary key: **id**

Description: post descripition

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Key constraints** | **Data type** | **Size** | **Description** |
| 1 | id | Primary key | int | 11 | id |
| 2 | date\_created | Not null | int | 11 | date created |
| 3 | context\_text | Not null | int | 11 | Text context |
| 4 | context\_image | Not null | date | 10 | context image |
| 5 | comment\_count | Not null | int | 10 | comment count |
| 6 | creater\_id | Not null | int | 10 | creator\_id |

1. **Profileapp\_post\_likers:**

Primary key: **id**

Description: likes of the post

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Key constraints** | **Data type** | **Size** | **Description** |
| 1 | Id | Primary key | int | 50 | Primary key |
| 2 | post\_id | Not null | int | 20 | id of post |
| 3 | user\_id | Not null | Varchar | 20 | id of user |

1. **Profileapp\_post\_savers:**

Primary key: **id**

Description: post savers

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No:** | **Fieldname** | **Key constraints** | **Data type** | **Size** | **Description** |
| 1 | id | Primary key | int | 11 | Id |
| 2 | post\_id | Not null | int | 11 | post id |
| 3 | user\_id | Not null | varchar | 200 | User\_id |

# CHAPTER 5

**SYSTEM TESTING**

### INTRODUCTION

Software testing is a systematic procedure of executing software in a regulated manner for the purpose of determining whether the software performs in accordance with the prescribed requirements. The assessment of software functionality, as ascertained by software testing, is frequently associated with the terms verification and validation. Validation is the process of scrutinizing or evaluating the conformity and coherence of various items, which may include software, with respect to an associated specification. The verification process entails several techniques, including reviews, analysis, inspections, and walkthroughs; among them is software testing. The validation process entails verifying that the specifications align with the user's intended expectations. "Static analysis and dynamic analysis are frequently linked activities in the domain of software testing." The methodology of static analysis involves the examination of software source code in order to detect and identify potential issues and gather relevant metrics, without the need for actual code execution. Dynamic analysis is concerned with examining the behavior of software during its execution phase in order to obtain valuable information, including execution traces, timing profiles, and test coverage information.

Testing comprises a range of activities that can be premeditated and executed in a methodical manner. The testing process commences at the module level and progresses towards the integration of the complete system comprising computers. The indispensable nature of testing in determining the success of a system necessitates the establishment of testing objectives. To this end, several rules exist which can function as effective testing objectives.

### TEST PLAN

A test plan refers to a comprehensive document that outlines the testing strategy and approach tailored towards a particular project or product. Selenium has gained widespread popularity as a choice tool for the automation of tests pertaining to web application testing.

A comprehensive test plan prescribes a series of essential procedures that must be executed to achieve successful testing outcomes across a range of methodological approaches. The proposed course of action is delineated in the testing protocol. The development of a computer program, its accompanying documentation, and corresponding data structures, resides within the purview of software developers. In the field of software development, it is incumbent upon the software developers to undertake thorough testing procedures on every individual component of a program, ensuring that it meets the intended objectives. An independent test group (ITG) has been established in order to tackle the issues arising from the practice of allowing the creators of a particular item to conduct its testing. The establishment of specific objectives for testing should be articulated in measurable discourse.

Consequently, the test plan must comprise pertinent details concerning the average length of time until failure, the expenses involved in identifying and remedying defects, residual defect density or frequency of incidence, as well as the total number of work hours imperative for conducting each regression test.

#### Unit Testing

The practice of unit testing is oriented towards the verification of software components or modules, which represent the most fundamental elements of software design. The description pertaining to the design at the component level is utilized as a framework during the process of evaluating critical control pathways in order to identify inaccuracies or defects that may be situated within the boundaries of the module. The revealed extent of unit testing and the corresponding level of difficulty of the tests.

The practice of unit testing places emphasis on scrutinizing the internal workings of software components, and often involves assessing multiple elements concurrently. In order to ensure proper functioning of the program unit under test and its departure from data centers, the modular interface undergoes rigorous testing. In order to ensure the integrity of temporarily stored data throughout all stages of algorithmic execution, scrutiny is placed on the local data structure. In order to verify the execution of all statements within a given module, an assessment of boundary conditions is conducted. Subsequently, a comprehensive analysis is conducted on individual methods of error management. Prior to commencing any other testing, it is imperative to perform the data flow evaluations on a module interface.

#### Integration Testing

#### Integration testing is a well-organized and methodical technique for constructing the program structure. Simultaneously, it also involves conducting a series of tests to reveal any manifestations of errors that might be related to interfacing. The aim of this process is to construct a program framework based on predetermined design principles utilizing rigorously tested unit components. The comprehensive evaluation of the program occurs in its entirety. Corrections pose significant challenges as identifying the causal factors is a complex undertaking due to the expansive nature of the underlying program. Upon rectification of such errors, a subsequent emergence of novel errors becomes evident, thereby initiating a seemingly perpetual cycle. Upon completion of unit testing in the system, the various modules were subsequently integrated and tested for any potential incongruities within the interfaces. Furthermore, program structures that exhibited disparities were eliminated, and a singular program structure underwent evolution.

#### Validation Testing or System Testing

#### The procedure of experimentation is concluded at this juncture. A comprehensive evaluation of the complete system was conducted, encompassing all forms, code, modules, and class modules. This variety of testing is commonly referred to as system tests or black box testing in the domain of software engineering. The primary focus of the black box testing method lies in the functional requisites of the software. By employing the Black Box testing methodology, a software engineer can assemble a collection of input conditions that comprehensively evaluate each and every program requirement.

#### The identification of faults within software systems is a crucial task in software engineering. One of the widely-employed techniques in this regard is black box testing, which targets a range of such faults, including but not limited to errors in data structures or external data access, defective or incomplete functions, interface discrepancies, performance impairments, issues in system initialization, and problems with system termination.

#### 5.2.4 Output Testing or User Acceptance Testing

The evaluation of the system in question is contingent upon user acknowledgment, with the additional criterion of satisfying the company's requirements. During development, it is essential for the program to establish frequent communication with both the user and the underlying system in order to implement necessary modifications. In consideration of the subsequent aspects, action was taken:

The task of generating output screen designs can be accomplished through a systematic approach that adheres to the principles of effective communication. Such an approach requires the utilization of relevant design principles, methodologies, and tools to ensure the utmost clarity, organization, and visual appeal of the resultant output screens. By executing this task in a structured manner, one can enhance the user experience and facilitate the effective transmission of information from the system to its users.

#### Automation Testing

#### The automation testing technique involves the utilization of specialized automated testing software tools to execute a suite of test cases. The test phases are executed with great attention to detail by an individual conducting manual testing, from a seated position in front of a computer. Moreover, the automated testing software has the capability to produce comprehensive test reports, perform a comparative analysis of the anticipated outcomes versus actual results, and input test data into the System Under Test (SUT). The process of automating software testing demands considerable financial and material resources.

#### Selenium Testing

Selenium is an open-source automated testing framework that is utilized to verify web applications across a diverse range of platforms and browsers, free of cost. Various programming languages can be utilized for crafting Selenium test scripts, such as Java, C#, Python, amongst others. The process of conducting evaluations through the utilization of the Selenium testing tool is commonly referred to as Selenium testing. Given that Selenium is an amalgamation of diverse tools, it is also developed by a varied group of developers. Herein presented are the primary contributors who have made noteworthy contributions to the Selenium project.

#### 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 |  | Dashboard 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 Admin Dashboard with  records | Pass |
| 3 | Provide Valid Password | Password: malavika |
| 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:

#### 

**Test report:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |
| **Project Name: JUSTCHAT** | | | | | | |
| **Update Following** | | | | | | |
| **Test Case ID:** 2 | | | | **Test Designed By:** Justin V Kalappura | | |
| **TestPriority(Low/Medium/High):**High | | | | **Test Designed Date:** 12-05-2023 | | |
| **Module Name**: Following/Unfollow | | | | **Test Executed By: :** Mr. Ajith G.S | | |
| **Test Title:** Guest 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** | **ExpectedResult** | **Actual**  **Result** | **Status(Pass/**  **Fail)** |
| 1 | | Navigation to Login Page |  | Dashboard should be displayed | Login page displayed | Pass |
|  | | Provide | User Name: | User should be able | User Logged |  |
| 2 | | Valid | malavika | to Login | in and |  |
|  | | Username |  |  | navigated to  Dashboard |  |
|  | with Records  Following option is working | Pass |
| 3 | | Provide Valid Password | Password: 9847344754 |
| 4 | | Click on Login button |  |
|  | |  |  |  |  |
| 5 | | Click on Following button |  |  |
| 6 | | Check whether following is working or not |  |  |  |  |
| **Post-Condition:** User is validated with database and successfully login into account and the password of the user updated successfully. The Following option is working. | | | | | | |

#### 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")

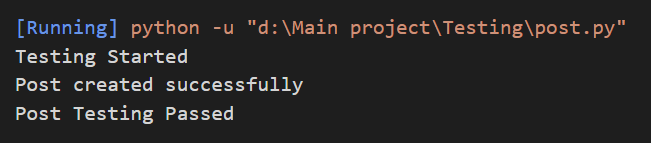
else:

    print('Post creation failed')

print("Post Testing Passed")

driver.quit()

#### Screenshot:



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **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 Back details | | | | | |
| **Step** | **Test Step** | **Test Data** | **Expected Result** | **Actual Result** | **Status (Pass/Fail)** |
| 1 | Navigation to Login Page |  | Dashboard should be displayed | Login page displayed | Pass |
| 2 | Provide  Valid Username | Username: malavika |  |  |  |
| 3 | Provide  Valid Password | Password: 9847344754 |  | User Logged  in and  navigated to  Dashboard  with Records  Post created successfully |  |
| 4 | Click on  Login button |  |
| 5 | Click on create post |  |  |
| 6 | Select pic and content |  |  |
| 7. | Click On Post Button |  |  |  |  |

|  |
| --- |
| **Post-Condition:** New Post created Succefully. |

**Test Case 4: Search Testing**

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/div").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 :** Searching properly implemented. | | | | | |
| **Step** | **Test Step** | **Test Data** | **ExpectedResult** | **Actual**  **Result** | **Status(Pass/**  **Fail)** |
| 1 | Navigation to Login Page |  | Dashboard should be displayed | Login page displayed | Pass |
| 2 | Provide Valid Username | User Name: admin@gmail.com | User should be able to Login | Admin Logged in and navigated to Dashboard  Search implemented | Pass |
| 3 | Provide Valid  Password | Password:1234 |
| 4 | Search a user |  |
| **Post-Condition:** Searching is successfully depicted. | | | | | |

# CHAPTER 6 IMPLEMENTATION

### INTRODUCTION

### The implementation phase is the pivotal stage of a project that involves the conversion of the theoretical design into an operational system. The attainment of users' confidence in the functionality, effectiveness, and accuracy of a newly developed system is widely perceived as a pivotal stage in its successful establishment. The primary focus of this study centers on user education and documentation. The process of conversion generally occurs either concurrently with the user's training or at a subsequently later point in time. During this stage, the primary burden, the most significant disturbance, and the principal influence on the current system are transferred to the department responsible for utilizing it. Insufficiently strategized or regulated implementation may result in disorder and disorientation.

### The process of implementation encompasses all tasks and operations involved in transitioning from the current system to the newly-introduced system. The novel system possesses the potential to be a complete departure from an established manual or automated system, or it may undertake modification of a preexisting system. The proficient execution of a system is of utmost importance in ensuring its dependability and its ability to fulfill the requisites of an organization. The act of deploying a developed system into practical use is commonly referred to as system implementation within academic discourse. This encompasses all the procedures and operations that are carried out in order to facilitate the transition from the old system to the new system. The implementation of the system is contingent on thorough testing to ensure conformity with the predetermined specifications. The personnel responsible for administering the system assess its feasibility. As the complexity of the implemented system increases, the system analysis and design effort required to execute the three fundamental components namely, education and training, system testing, and changeover, increases correspondingly.

### IMPLEMENTATION PROCEDURES

The implementation of software pertains to the conclusive incorporation of a software package into its actual operating environment, ensuring the fulfillment of its intended purposes and the effective functioning of the system. In numerous organizational contexts, the task of commissioning a software development project is often delegated to an individual who will not be directly involved in its operational aspects. It is imperative that the active user is cognizant of the advantageous features of utilizing the novel system. The software has instilled confidence within the users.

Adequate instruction is provided to the user in order to facilitate comfort and competence when operating the application. Prior to accessing the system, it is imperative that the user possesses knowledge of the server program's prerequisite operation for result generation to occur. The occurrence of the process is contingent upon the operational status of the server object on the server. Failure to activate the server object will result in the absence of the process.

#### User Training

The purpose of user training is to equip users with the necessary skills and knowledge to effectively evaluate and transition into a new system. In order to attain the anticipated aims and advantages of a computer-based system, it is imperative that the individuals who will be engaged are assured of their respective roles within the novel system. The increasing complexity of systems necessitates the primacy of training. Through comprehensive user training, individuals gain knowledge on how to effectively enter data into the designated system, accurately respond to error notifications, efficiently investigate the database, and promptly initiate routine processes that generate critical reports and carry out various essential functions.

#### System Maintenance

#### The maintenance phase of the software development life cycle is primarily characterized by the period during which a software product is utilized to carry out practical tasks. Upon successful implementation of a system, it ought to be duly maintained in an appropriate fashion. The maintenance of software systems is a crucial element within the software development life cycle. The necessity of system maintenance lies in its capacity to remain adaptable to alterations in the system's surrounding surroundings. The realm of software maintenance extends beyond the mere identification of errors.

#### Training on the Application Software

#### Subsequent to equipping the user with the essential rudimentary training regarding computer literacy, it will be imperative to impart instruction respecting the novel application software. This exposition presents the fundamental tenets that underlie the employment of the new system, encompassing the screen flow, screen design typology in conjunction with on-screen assistance, potential data entry errors, the corresponding validity assessments with each entry, and remedial measures to rectify any inaccuracies. This statement posits that the provision of pertinent information to a given user or group is a necessary precursor to effective utilization of a system or one of its constituent parts. In addition, the training for the application program must also be imparted to facilitate the user/ group's proficiency in its operation. Variations in the training approach may occur amongst diverse user groups and hierarchical levels.

**6.2.4 Hosting**

Hosting your Mini-Hospital Management System on Azure App Service using a GitHub Actions CI pipeline offers a streamlined and automated deployment process. With Azure App Service's managed platform, you can easily deploy your web application without worrying about infrastructure management. Leveraging GitHub Actions allows you to define a workflow that triggers deployment whenever there are changes to your repository. Azure App Service provides scalability, high availability, and security features, while integrating with other Azure services for additional capabilities. This combination simplifies the hosting process and ensures a reliable platform for delivering your application to users.

Eg: Azure App Service

Azure App Service is a fully managed platform provided by Microsoft Azure that allows developers to easily deploy and host web applications without the need for infrastructure management. With App Service, developers can focus on building and deploying their applications while Azure takes care of the underlying infrastructure, including scalability, high availability, and security. It supports various programming languages and frameworks, making it versatile for different application types. App Service also integrates seamlessly with other Azure services, enabling developers to leverage additional functionalities such as database storage, monitoring, and authentication. With its ease of use, flexibility, and robust features, Azure App Service provides a reliable and efficient hosting solution for web applications.

**Procedure for hosting a website on Azure App Service:**

Step 1: Containerize your Django app: Create a Dockerfile in the root directory of your Django project to define the container image. Configure the Dockerfile to include the necessary dependencies, packages, and settings for your app.

Step 2: Set up Docker Hub: Create an account on Docker Hub (hub.docker.com) and set up a repository to store your container image. Make sure to note down the repository name and tag.

Step 3: Configure GitHub Actions: In your GitHub repository, create a workflow file (e.g., .github/workflows/main.yml) to define your deployment workflow. Use the appropriate GitHub Actions for building and pushing the Docker image to Docker Hub. This workflow should trigger on push events to the repository.

Step 4: Build and push Docker image: Commit and push your code changes to trigger the GitHub Actions workflow. The workflow will build the Docker image using the Dockerfile and push it to the specified Docker Hub repository.

Step 5: Deploy to Azure App Service: In the Azure portal, create an Azure App Service instance or use an existing one. Configure the App Service to pull the Docker image from Docker Hub by specifying the repository name and tag. Once configured, Azure App Service will automatically deploy and run your Django app using the Docker image from Docker Hub..

.

By following these steps, you can automate the process of containerizing and deploying your Django app to Azure App Service using Docker, Docker Hub, and GitHub Actions. This approach simplifies the deployment process and ensures a consistent and scalable hosting environment for your app.

**Hosted Website:**

Hosted Link: Mini-Hospital (www.freeforu.in)

# CHAPTER 7 CONCLUSION AND FUTURE SCOPE

### CONCLUSION

The present study reveals that the "JUSTCHAT" system, as a social media platform, represents a superior product for promoting social progress within a given community. The current systems exhibit significant inadequacies and inefficiencies concerning this matter. The novel system which is set to be implemented offers notable effectiveness, efficiency, and ease in the management and formation of a socially cognizant community or society. This phenomenon engenders a sense of egalitarianism and provides a shared platform for individuals to articulate their perspectives on contemporary occurrences. The aforementioned system has been subjected to testing using a set of experimental data in order to ensure compliance with specified criteria, thereby yielding results that align with the established standards. The evaluation of the system's performance indicates a significantly higher level of efficiency compared to the current system. While it cannot be asserted that this project is an exemplar, it will adequately fulfill the primary demands of the issue. "In order to ensure the success of the project, it is imperative that modifications be made in response to evolving needs as they arise over time." The system was deployed and evaluated using actual resources in accordance with the available facilities.

### 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 us.
* Real time news can be made visible.
* Can be made efficient or profitable by making a space to show the admin our application.