**ELECTRONIC-STORE**

#### AN INTERNSHIP REPORT

***Submitted by***

### Patel Anshul Jayeshbhai

#### GTU Enrollment Number : 210670107047

***In fulfilment for the award of the degree of***

## BACHELOR OF ENGINEERING

***in***

### Computer Engineering

**SAL Institute of Technology and Engineering Research, Ahmedabad**

****

### Gujarat Technological University, Ahmedabad

**May, 2025**

#### SAL Institute of Technology and Engineering Research

**Opposite Science City, Sola, Ahmedabad, Gujarat – 380060**

# CERTIFICATE

This is to certify that the project report submitted along with the project entitled “**ELECTRONIC-STORE”** has been carried out by **Patel Anshul Jayeshbhai (210670107047)** under my guidance in partial fulfilment for the degree of Bachelor of Engineering in Computer Engineering, 8th Semester of Gujarat Technological University, Ahmedabad during the academic year2024- 25.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Dr. Krishna Hingrajiya Dr. Nimisha Patel

Internal Guide Head of the Department

**TO WHOM IT MAY CONCERN**

****

#### SAL Institute of Technology and Engineering Research

**Opposite Science City, Sola, Ahmedabad, Gujarat – 380060**

## DECLARATION

We hereby declare that the Internship report submitted along with the IT Organization Management System entitled **Electronic-Store** submitted in partial fulfilment for the degree of Bachelor of Engineering in Computer Engineering to Gujarat Technological University, Ahmedabad, is a bonafide record of original project work carried out by me at SAL Institute of Technology and Engineering Research under the supervision of **Prof.Krishna Hingrajiya** and that no part of this report has been directly copied from any students’ reports or taken from any other source, without providing due reference.

Name of Student

**Patel Anshul Jayeshbhi**

Sign of Student

…………………………..

## ACKNOWLEDGEMENT

I would like to express my sincere gratitude to several individuals for supporting me throughout my internship project. First, I wish to express my sincere gratitude to my Internal Guide, **Prof. Varsha Tank** for his enthusiasm, patience, insightful comments, helpful information, practical advice and unceasing ideas that have helped me tremendously at all times in my project. Without his support and guidance, this project would not have been possible.

I would like to express my endless thanks to my external guide **Mr. Ayaz Vhora** at **Qrious Tech Team LLP** their sincere and dedicated guidance throughout the project development.

I also wish to express my sincere thanks to Principal of **SAL institute of Technology and Engineering Research** , **Prof. Rupesh Vashani** and to HOD of Computer Engineering **Prof. Neha Minocha** for providing the needful support throughout the timeline. Additionally, I would acknowledge all the support staff and my friends for their constant help and mental support.

## ABSTRACT

The industrial training is as important as theoretical knowledge, considering the rapidly changing technologies in the field of computer and information technology.

I completed the internship in the field of Web Development, at Qrious Tech Team LLP. The project definition that I had been assigned to is titled: ‘Incognito Messenger’ which is Developed using the MERN stack and Socket.IO, Incognito Messenger is a secure messaging application that prioritizes user privacy. It ensures confidentiality through encrypted messages and discreet profiles, featuring centralized data storage, automated data collection, and real-time analysis capabilities for timely insights while safeguarding user identity.

With scalable infrastructure and advanced analytical tools, Incognito Messenger delivers efficient performance and meaningful agricultural insights. The project follows a systematic implementation process to establish a secure and intuitive platform for confidential dialogue and collaboration, setting a new standard for private online interactions

## LIST OF FIGURES

Fig 1.1Organization Chart 1

Fig 2.1Schematic Layout. 4

Fig 3.7.1Gantt Chart 24

Fig 5.1.1 System Design… 29

Fig 5.1.2 Software Methodology 29

Fig 5.2.1 UserActivity Diagram 31

Fig 5.2.2Admin Activity Diagram 31

Fig 5.2.3 ER Diagram 36

Fig5.2.4 DFD Diagram… 37

Fig 5.3.1 Home Page 39

Fig 5.3.2Login 39

Fig 5.3.3Sign up… 40

Fig 5.3.4 Chat Page 40

Fig 8.1 Meeting with Mentor 45

## LIST OF TABLES

Table3.7.2Internship Summary 24

Table 5.2.1: Symbols and components of ER Diagram 36

Table 5.2.2: Symbols and components of DFD Diagram 37

Table 7.2.1.1: Test cases for User 44

Table 7.2.1.2: Test cases for Admin 44

### Abbreviations

**HTML CSS JS URL API OOP ORM CRUD DFD ER**

**UX ERP CRM**

**HyperText Markup Language Cascading Style Sheets JavaScript**

**Uniform Resource Locator Application Programming Interface Object Oriented Programming Object-Relational Mapping**

**Create, Read, Update, Delete Data Flow Diagram**

**Entity Relationship User Experience**

**Enterprise Resource Planning Customer Relationship Management**

**Table of Content**

[Acknowledgement i](#_bookmark0)

[Abstract ii](#_bookmark1)

[List of Figures… iii](#_bookmark2)

[List of Tables iv](#_bookmark3)

List of Abbreviations v

Table ofContents vi

[Chapter 1 Overview of the Company 1](#_bookmark4)

* 1. [History 1](#_bookmark5)
  2. [Scope of work 2](#_bookmark6)
  3. [Capacity of Company 2](#_bookmark7)

Chapter 2 Overview Of Different Department Of The Organization 3

* 1. Includes the details of the work being carried out in each department 3
  2. Technical Specification of Major Equipment Used… 4
  3. [Schematic Layout of Sequence of Operation 4](#_bookmark8)
  4. Explain in details about each stage of production… 7

[Chapter 3 IntroductiontoProject 8](#_bookmark9)

* 1. [Project Summary… 8](#_bookmark10)
  2. [Purpose 8](#_bookmark11)
  3. [Objective 9](#_bookmark12)
  4. [Scope 9](#_bookmark13)
  5. [Technology and Literature Review 11](#_bookmark14)
     1. [Technology Review 11](#_bookmark15)
     2. Justification of Review 12
     3. [Literature Review 15](#_bookmark16)
  6. Project Planning 15
     1. Project / Internship Development Approach and Justification 16
     2. Project / Internship Effort and Time, Cost Estimation 19
     3. [Roles and Responsibilities 22](#_bookmark17)
     4. [Group Dependencies… 23](#_bookmark18)
  7. [Project Scheduling 24](#_bookmark19)
     1. Gantt Chart 24

[Chapter 4 System Analysis 26](#_bookmark20)

* 1. [S](#_bookmark21)[tudy of Current System… 26](#_bookmark20)
  2. [Problem and Weaknesses of Current System 26](#_bookmark22)
  3. [Requirements of New System 27](#_bookmark23)
  4. [System Feasibility 27](#_bookmark24)
  5. [Activity / Process in New System / Proposed System………………… 27](#_bookmark25)
  6. [Features of New System 28](#_bookmark26)
  7. Main Components of Proposed System 28

[Chapter 5 SystemDesign 29](#_bookmark27)

* 1. [System Design & Methodology 29](#_bookmark28)
  2. [Process Design 30](#_bookmark29)
  3. [Interface Design 39](#_bookmark30)

[Chapter 6 Implementation 41](#_bookmark31)

* 1. [Implementation Platform / Environment 41](#_bookmark32)
  2. [Process / Program / Technology / Modules Specification… 41](#_bookmark33)

[Chapter 7 Testing 43](#_bookmark34)

* 1. [Testing Strategy… 43](#_bookmark35)
  2. [Test Results and Analysis 44](#_bookmark36)
     1. [Test Cases 44](#_bookmark37)

[Chapter 8ConclusionandDiscussion 45](#_bookmark38)

* 1. [Overall Analysis of Internship Viabilities 45](#_bookmark39)
  2. [Meeting with Institute Mentor 45](#_bookmark40)
  3. [Problem Encountered and Possible Solutions 46](#_bookmark41)
  4. [Summary of Internship work 46](#_bookmark42)
  5. [Limitation and Future Enhancement 47](#_bookmark43)

[References 48](#_bookmark44)

## CHAPTER 1: OVERVIEWOFTHECOMPANY

#### HISTORY

Qrious Tech Team LLP was founded with a vision to address growing businesses' needs of reducing the time to market and cost effectiveness required to develop and maintain unique and customized web and mobile solutions. We are uniquely and strategically positioned to partner with startups and leading brands to help them expand their business and offer the most effective and cost-efficient solutions that provide revenues and value to their business needs.

Qrious Tech Team LLP is a software development company focusing primarily on Mobile Application Development, Web Development, Salesforce servicing and many more, providing top- quality solutions according to client needs. Qrious Tech Team LLP’s development team expertise in different technologies allows us to propose the ideal technical solution for each client specific needs. Our combination of client-side scripting, backend programming, database development and usable interface design produce simple, intuitive and productivity- enabling application solutions.

Qrious Tech Team LLP succeeds because we have respect for each client’s business goals, bring proactive perspectives, solve problems through critical thinking and collaboration, apply best practices and technical expertise, and continually foster the professional growth and learning of our consultants.

#### Scope of Work

Qrious Tech Team LLP is a Customer-Centric Professional Service Company. A few of the services provide are IT Consultations, Customized Web Solutions, Mobile Development, Cloud Development, Systems Integration, and Server Management.

#### Capacity of Company

Qrious Tech Team LLP offer a wide range of services like app development for Android, iOS, Windows, web development, enterprise solutions, SEO services, digital marketing solutions, UI/UX design, Data science and Cloud Development.

It believes in working with less but highly qualified and experienced developers. With a team of 80+ developers, they follow standard software development process. They are wellknown for on-time delivery of high quality and secure products.

## CHAPTER 2: OVERVIEW OF DIFFERENT PROCESSES OF THE ORGANIZATION

#### Include the details about the work being carried out in each department

There are different departments in the company: Project management, Technical Management, Software Development, Web Development, App Development and Testing.

Project planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment. Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure.

Project planning is often used to organize different areas of a project, including project plans, workloads and the management of teams and individuals.

Project Management System has six major modules of Admin, Manage Application, Test Management, Process Management, Manage Comment, Reports. We analysed the overall complexity of each of these modules and it was found that the project will require approximately 12 weeks to get completed, so we planned accordingly.

Web and app developers work on different project definitions to build a fully working model. Testing department is in charge of testing the working of different elements and functionalities in the project built by developers and provide feedback status to developers pointing out errors if any.

#### Technical Specifications of Major Equipment Used

* + - **Hardware :-**
      * Minimum screen resolution: 1024\*768
      * Processor minimum: 500 mhz Intel Core i5 workstation or equivalent.
      * Memory minimum :512mb recommended :2gb
      * Disk space :850 MB of free disk space recommended: 1gb

#### Software :-

* + - * Front End Tool: - HTML, CSS Bootstrap, React , MUI
      * Back End Tool: - Node JS, Java
      * Frame work: Express JS, Angular
      * Database – MongoDB, SQL

#### Schematic Layout of Sequence of Operation

Project Scheduling is the culmination of a planning activity that is primary component of software project management. When combined with estimation methods and risk analysis, scheduling establishes a road map for the project management.

Scheduling begins with the process composition. The characteristics of the project are used to adapt an appropriate task set for the work to be done. The task network is used to compute the critical project path, a time line chart and a variety of project information. When creating a software project schedule, the planner begins with a set of tasks. If automated tools are used, the work breakdown is input as a task network or task outline. Effort, duration, and start date are then input for each task. In addition, tasks may be assigned to specific individuals. As a consequence of this input, a timeline chart, also called a Gantt chart is generating. A Timeline Chart can be developed for the entire project. Timeline Charts depict a part of a software project schedule

The incremental build model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the waterfall model with the iterative philosophy of prototyping.

Iterative and Incremental development is a combination of both iterative design or iterative method and incremental build model for development. "During software development, more than one iteration of the software development cycle may be in progress at the same time." and "This process may be described as an "evolutionary acquisition" or "incremental build" approach."

In incremental model the whole requirement is divided into various builds. During each iteration, the development module goes through the requirements, design, implementation and testing phases. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is ready as per the requirement.

The key to successful use of an iterative software development life-cycle is rigorous validation of requirements, and verification & testing of each version of the software against those requirements within each cycle of the model. As the software evolves through successive cycles, tests have to be repeated and extended to verify each version of the software.

Like other SDLC models, Iterative and incremental development has some specific applications in the software industry. This model is most often used in the following scenarios:

* Requirements of the complete system are clearly defined and understood.
* Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
* There is a time to the market constraint.
* A new technology is being used and is being learns by the development team while working on the project.

Resources with needed skill set are not available and are planned to be used on contract basis for specific iterations

#### Details About Production

Project Management System has six major modules of Admin, Manage Application, Test Management, Process Management, Manage Comment, Reports. We analysed the overall complexity of each of these modules and it was found that the project will require approximately 14 weeks completing, so we planned accordingly.

We decided to follow the SDLC i.e. Software Development Life Cycle while planning various phases of our project. This method consists of following activities:

1. Determination of system requirements
2. System Analysis
3. Design of system
4. Development of software
5. System Testing
6. Implementation and Evaluation

There are some high-risk features and goals which may change in the future.

* + Integration and Testing: All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
  + Deployment of system: Once the functional and nonfunctional testing is done, the product is deployed in the customer environment or released into the market.

Maintenance: There are some issues which come up in the client environment. To fix those issues patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment**.**

### Chapter 3 : Introduction to Project

#### Project Summary

During my internship at Qrious Tech Team LLP, I had the opportunity to immerse myself in various aspects of web development within the software development field. Over the course of 3 Months, I gained valuable hands-on experience and insight into the software development industry, contributing to the team's objectives while honing my skills and expanding my knowledge.

Throughout the internship, I was involved in a range of web development tasks and projects, collaborating with colleagues to achieve project milestones and deliverables. This included front-end and back-end development tasks, such as designing user interfaces, implementing functionality, and troubleshooting issues to ensure optimal performance.

Additionally, I participated in training programs and workshops organized by Qrious Tech Team LLP, where I learned about the latest technologies and best practices in web development. This helped me develop a deeper understanding of the software development field, and I appreciated the opportunity to network with professionals in the industry.

Overall, my internship experience at Qrious Tech Team LLP was incredibly rewarding. It provided me with valuable insights, practical skills, and professional connections that will undoubtedly benefit my future endeavors in web development and software development.

#### Purpose

* To Provide a Secure Communication Platform: The primary purpose of Incognito Messenger is to offer users a secure and confidential platform for communication, where they can freely express themselves without the fear of privacy breaches.
* To Foster Meaningful Interactions: By prioritizing anonymity and privacy, the platform aims to facilitate genuine and open dialogue among users, fostering meaningful interactions and connections.
* To Fill a Gap in the Market: The purpose of the platform is to address the growing need for privacy-focused communication tools in an era where online privacy concerns are becoming increasingly prominent.
* To Empower Users: Incognito Messenger aims to empower users by giving them control over their online interactions, allowing them to engage with others on their own terms, without the pressure of revealing their identities.
* To Promote Collaboration: Through features such as group creation and collaboration tools, the platform seeks to promote collaboration among users, enabling them to work together on projects or share ideas in a secure environment.

#### Objective

* Privacy and Security: Ensure that users can communicate securely and anonymously, prioritizing their privacy and confidentiality.
* User Engagement: Foster meaningful interactions among users through features such as group creation, facilitating discussions, collaboration, etc.
* Ease of Use: Develop an intuitive user interface and user experience to make the platform accessible and user-friendly.
* Community Building: Encourage the formation of communities and networks within the platform, allowing users to connect with like-minded individuals or professionals.
* Feedback Mechanism: Establish channels for user feedback to continuously improve the platform based on user suggestions and requirements.
* Market Differentiation: Position the platform as a unique offering in the market, distinguishing it from other messaging platforms through its emphasis on privacy and anonymity.

#### Scope

* Platform Overview: Provide an overview of the Incognito Messenger platform, highlighting its key features such as secure and anonymous communication, group creation, discreet profiles, etc.
* Target Audience: Define the target audience for the platform, such as individuals who prioritize privacy in their online interactions, professionals needing confidential communication channels, etc.
* Technical Specifications: Detail the technical aspects of the platform, including the technologies used, encryption methods employed for secure communication, compatibility across devices and operating systems, etc.
* Regulatory Compliance: Discuss any regulatory compliance requirements that the platform adheres to, such as data protection laws, privacy regulations, etc.
* Development Timeline: Outline the development timeline for the platform, including milestones, key deliverables, and anticipated launch dates.
* Scaling Plans: Describe plans for scaling the platform to accommodate growth in user base and feature enhancements in the future.

#### Technology and Literature Review

#### Technology Review

* + - * Frontend Development:
        + React: React served as the primary framework for building the frontend of

Silent Exchange. Its component-based architecture allowed for the creation of dynamic and responsive user interfaces, enhancing user interaction and experience.

CSS Styling: CSS played a crucial role in styling the frontend of Silent

Exchange. It was used in conjunction with React components to design and customize the appearance of user interface elements, ensuring a visually appealing and cohesive design across the application.

* + - * Backend Development:
        + Node.js: Node.js formed the backbone of the backend infrastructure for Silent

Exchange. Its event-driven architecture and non-blocking I/O operations facilitated the development of scalable and efficient server-side applications, enabling smooth communication between clients and servers.

* MongoDB: MongoDB was chosen as the backend database management system

for Silent Exchange. Its flexible document-based data model allowed for the storage and retrieval of user data and messages in a scalable and efficient manner, supporting the platform's growth and expansion.

* + - * Real-time Communication:
* Socket.IO: Socket.IO was utilized to implement real-time communication features in

Silent Exchange. It enabled bidirectional, event-based communication between clients and servers, facilitating instant messaging and notifications between users in a seamless and efficient manner.

#### Justification of Technology

* + - * React.js:
        + React.js was chosen for the frontend development due to its component- based

architecture, which allows for reusable and modular UI elements. This facilitates the development of a dynamic and responsive user interface, enhancing the overall user experience.

* + - * + Its virtual DOM (Document Object Model) and efficient rendering mechanism

optimize performance, ensuring smooth interactions and fast page loading times, which are essential for modern web applications.

* + - * + The large and active community surrounding React.js provides extensive

documentation, libraries, and resources, making it easier to troubleshoot issues, find solutions, and stay updated on best practices.

* + - * CSS (Cascading Style Sheets):
        + CSS is indispensable for styling and layout purposes, allowing for the

customization of the visual appearance of the web application. It enables the creation of visually appealing and user-friendly interfaces, enhancing user engagement and satisfaction.

* + - * + With CSS preprocessors like Sass or Less, developers can leverage features such

as variables, mixins, and nesting to streamline styling workflows and maintain consistency across the application.

* + - * + CSS Grid and Flexbox layouts offer powerful tools for creating responsive

designs, ensuring optimal viewing experiences across various devices and screen sizes, which is essential for modern web applications.

* + - * JavaScript (ES6+):
        + JavaScript is the backbone of web development, enabling dynamic and interactive

functionalities on the client-side. With ES6+ features such as arrow functions, template literals, and destructuring, developers can write cleaner, more expressive code.

* + - * + Its asynchronous nature and event-driven model make JavaScript well- suited for

handling user interactions, form validations, and DOM manipulation, providing a seamless and engaging user experience.

* + - * + The availability of modern JavaScript frameworks and libraries, along with

extensive browser support, ensures compatibility and accessibility across different platforms, enhancing the reach and usability of the web application.

* + - * Node.js:
        + Node.js is chosen for backend development due to its lightweight and scalable

nature, making it ideal for building server-side applications that can handle concurrent requests efficiently.

* + - * + Its non-blocking I/O model and event-driven architecture enable high

performance and responsiveness, ensuring fast data processing and delivery to the client-side.

* + - * + With a rich ecosystem of npm (Node Package Manager)

modules and libraries,

Node.js provides a vast selection of tools and resources for backend development, allowing developers to rapidly build and deploy robust web services and APIs.

* + - * Express.js:
        + Express.js complements Node.js by providing a minimalist and flexible web

application framework, simplifying the process of building server-side logic, handling routes, and managing middleware.

* + - * + Its simplicity and unopinionated nature allow for greater flexibility and

customization, empowering developers to structure their applications according to specific requirements and preferences.

* + - * + The streamlined development experience offered by Express.js, coupled with its

performance and scalability benefits, makes it a popular choice for building backend services and RESTful APIs in Node.js environments.

* + - * MongoDB:
        + MongoDB is selected as the database solution for its flexibility, scalability, and

ease of use, particularly in scenarios where the data structure is dynamic or subject to frequent changes.

* + - * + Its document-oriented data model and schema-less design eliminate the need for

predefined schemas, allowing for agile development and accommodating evolving data requirements.

* + - * + MongoDB's scalability features, including sharding and replication, ensure high

availability and fault tolerance, making it well-suited for handling large volumes of data and scaling horizontally as the application grows.

#### Literature Review

* A review of existing literature underscores the critical importance of privacy and anonymity in online communication platforms. Research papers and articles emphasize the need for secure communication channels to safeguard user data and foster trust among users in today's digital landscape.
* Comparative analysis with other similar platforms highlights the unique features and advantages of Silent Exchange. By prioritizing privacy and anonymity, it distinguishes itself as a secure and confidential communication platform, addressing the growing concerns of users regarding data privacy and security online.
* Insights gleaned from the literature review informed the development approach and feature set of Silent Exchange, ensuring alignment with user expectations and industry standards in secure communication platforms. Challenges and Opportunities: Identify common challenges faced in agricultural data analysis projects, such as data quality issues, scalability concerns, and the need for domain expertise. Discuss opportunities for innovation and improvement.
* User Experience and Adoption: Look into studies or reports on the user experience of data analysis tools in the agricultural domain. Understand factors influencing user adoption and satisfaction.

#### Internship Planning

* Internship planning is part of project management, which relates to the use of schedules such as Gantt charts to plan and subsequently report progress within the project environment.
* Initially, the project scope is defined and the appropriate methods for completing the project are determined. Following this step, the durations for the various tasks necessary to complete the work are listed and grouped into a work breakdown structure.
* Internship planning is often used to organize different areas of a project, including project plans, workloads and the management of teams and individuals.
* After completion of initial training, different modules were assigned to us individually which we had to work on, under the guidance or our industry mentor and internal guide.

#### Project Development Approach and Justification

Planning before any activity is very much important. And if it is planned nicely, then success is guaranteed.

Project Management System has six major modules of Admin, Manage Application, Test Management, Process Management, Manage Comment, Reports. We analyzed the overall complexity of each of these modules and it was found that the project will require approximately 14 weeks completing, so we planned accordingly.

We decided to follow the SDLC i.e. Software Development Life Cycle while planning various phases of our project. This method consists of following activities:

1. Determination of system requirements
2. System Analysis
3. Design of system
4. Development of software
5. System Testing
6. Implementation and Evaluation

The incremental build model is a method of software development where the model is designed, implemented and tested incrementally (a little more is added each time) until the product is finished. It involves both development and maintenance. The product is defined as finished when it satisfies all of its requirements. This model combines the elements of the

waterfall model with the iterative philosophy of prototyping.

Following is the pictorial representation of Iterative and Incremental model:



*Fig 3.1 Iterative And Incremental Model*

Iterative and Incremental development is a combination of both iterative design or iterative method and incremental build model for development. "During software development, more than one iteration of the software development cycle may be in progress at the same time." and "This process may be described as an "evolutionary acquisition" or "incremental build" approach."

In incremental model the whole requirement is divided into various builds. During each iteration, the development module goes through the requirements, design, implementation and testing phases. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is ready as per the requirement.

The key to successful use of an iterative software development life-cycle is rigorous validation of requirements, and verification & testing of each version of the software against those requirements within each cycle of the model. As the software evolves through successive cycles, tests have to be repeated and extended to verify each version of the software

Like other SDLC models, Iterative and incremental development has some specific applications in the software industry. This model is most often used in the following scenarios:

* Requirements of the complete system are clearly defined and understood.
* Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.
* There is a time to the market constraint.
* A new technology is being used and is being learns by the development team while working on the project.

Resources with needed skill set are not available and are planned to be used on contract basis for specific iterations.

* There are some high-risk features and goals which may change in the future.
* Integration and Testing: All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
* Deployment of system: Once the functional and non functional testing is done, the product is deployed in the customer environment or released into the market.
* Maintenance: There are some issues which come up in the client environment. To fix those issues patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

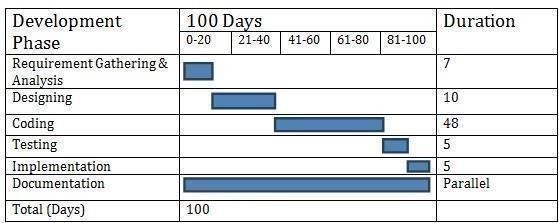


Fig 3.2 Development Stages

#### Project Effort and Time, Cost Estimation

**Effort Estimation:**

Each company determines the output it expects from its team members. Let us call the average output of a team member per man-hour as the unit output. Assume that one has to deliver an end-to-end login module's functionality for an application. The time spent on the login functionality should include the corresponding time required for gathering the requirements, doing a requirement analysis, architecture inputs, form design, object/class design, implementing the business rules, data validation and storage, framework (i.e., code for login module's constants, enumerations, utilities), testing, debugging, deployment up to user acceptance, etc.

Now, the estimator has to figure out how many man-hours it would take to complete the login module, keeping all these factors in mind. The sequence of work and dependencies should be considered as they do cause delays in completion. For example, form design should be done first (all the way up to acceptance by the customer), then object design (up to acceptance by the architect), followed by coding (for business rules, calculations, and data validations), internal testing, and user acceptance testing. A wise estimator would always take support from other people to understand the scope of work to do a given task.

#### Cost Estimation:

**The COCOMO Model**

* Like all estimation models for software, the COCOMO models require sizing information.
* Three different sizing options are available as part of the model hierarchy: object points, function points, and lines of source code.
* Like function points, the object point is indirect software that is computed using counts of the number of:
  1. Screens (at the user interface),
  2. Reports,
  3. Components likely to be required to build the application.
* Once complexity is determined, the number of screens, reports, and components are weighted according to Table above.
* The object point count is then determined by multiplying the original number of object instances by the weighting factor in table above and summing to obtain a total object point count.
* Once complexity is determined, the number of screens, reports, and components are weighted according to Table above.
* The object point count is then determined by multiplying the original number of object instances by the weighting factor in table above and summing to obtain a total object point count.
* When component-based development or general software reuse is to be applied, the percent of reuse (%reuse) is estimated and the object point count is adjusted:
* **NOP = (object points) X [(100 - %reuse) / 100].**
* Where NOP is defined as new object points. To derive an estimate of effort based on the computed NOP value, a “productivity rate” must be derived.
* **PROD=NOP / person-month**
* For different levels of developer experience and development environment maturity. Once the productivity rate has been determined, an estimate of project effort can bederived as Estimated effort = NOP/PROD.

There are three types of software project: Organic, semi-detached and embedded project.

Cost required to develop project=effort\*rs/month

* **Effort Estimation (E):**

In Organic=2.4 (KLOC)1.05 PM

In semidetached=3.0(KLOC)1.12 PM In Embedded=3.6(KLOC)1.20PM

* **Duration Estimation (D):** In Organic=2.5(effort)0.38months

In semidetached=2.5(effort)0.35 months In Embedded=2.5((effort)0.32months

* **Person Estimation:**

P=E/D

**Advantages of COCOMO:**

* COCOMO is factual and easy to interpret.
* One can clearly understand how it works.
* Accounts for various factors that affect cost of the project. •Works on historical data and hence is **more** predictable and accurate.

**Disadvantages:**

* COCOMO model ignores requirements and all documentation.
* It ignores **customer skills**, cooperation, knowledge and other parameters.
* It oversimplifies the impact of safety/security aspects.
* It ignores hardware issues.
* It ignores personnel turnover levels.

It is dependent on the amount of time spent in each phase.

#### Roles and Responsibilities

* Understand and undergo extensive training given by mentors in the field of Python with Data Science
* Fulfill tasks given by supervisors.
* Attend meetings and take minutes
* Conduct research work for the implementation of project
* Backend development
* Front-end development
* Data visualization module using Pandas and Plotly library.
* Perform testing for the project under the supervision of mentors

#### Group Dependencies

The project that I was assigned under, is titled ‘Protection Against Domestic Violence & Guidance System’. The modules where I had to work on included: quiz module, implementing data science modules which included data visualization, matplotlib libraries, pandas library, etc.

I also worked on the ‘Help’ Page that will help find the users, specific details about police stations in a particular area. I implemented the ‘legal resources’ page, that includes information about different laws, crimes and their punishments. I also worked towards building a ‘Story Page’ that showcases experiences of different women.

#### Project Scheduling

* Project Scheduling is the culmination of a planning activity that is primary component of software project management.
* When combined with estimation methods and risk analysis, scheduling, establishes a road map for the project management.
* Scheduling begins with the process composition. The characteristics of the project are used to adapt an appropriate task set for the work to be done.
* The task network is used to compute the critical project path, a time line chart and a variety of project information.
* When creating a software project schedule, the planner begins with a set of tasks. If automated tools are used, the work breakdown is input as a task network or task outline. Effort, duration, and start date are then input for each task. In addition, tasks may be assigned to specific individuals.
* As a consequence of this input, a timeline chart, also called a Gantt chart is generating. A Timeline Chart can be developed for the entire project. Timeline Charts depict a part of a software project schedule
* All project tasks are listed in the left-hand column. The horizontal bars indicate the duration of each task. When multiple bars occur at the same time on the calendar, task concurrency is implied. The diamonds indicate milestones, which indicate the place where our project reach.

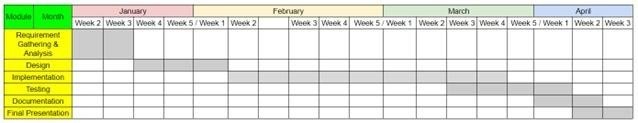


Fig 3.4 Gannt Chart



|  |  |
| --- | --- |
| Week 1 | Understood the basics of web development |
| Week 2 | Installed and configured the necessary tools for frontend and  backend development, including Node.js, React.js, Express.js, and MongoDB. |
| Week 3 | Learning Frontend technologies like HTML CSS |
| Week 4 | Learning Javascript which is base of web development |
| Week 5 | Fundamentals of OOP, file handling and exception handling and tasks on these topics. |
| Week 6 | A variety of tasks on random module and RegEx module. |
| Week 7 | Learnt about HTML, CSS, JavaScript, Bootstrap and responsive design in detail. |
| Week 8 | Learnt about basic React.js and How React Work behind the scene |
| Week 9 | Learnt about Custom hooks and react router dom |
| Week 10 | Learnt about Redux , Redux Toolkit and React Query |
| Week 11 | Learn about Node.js, express.js and basics of socket.io |
| Week 12 | Database Connectivity, Testing, Evaluation and Report Making. |

*Table 3.1 Summary Table*

### Chapter 4: System Analysis

#### Study of Current System

* + - The current system lacks anonymity and privacy features, compromising user confidentiality.
    - It may suffer from security vulnerabilities, risking data breaches and unauthorized access.
    - User feedback indicates dissatisfaction with the lack of secure communication channels.

#### Problem and Weaknesses of Current System

* + - Inadequate privacy measures lead to concerns over data privacy and security.
    - Absence of encryption and anonymity features exposes users to potential risks.
    - Limited functionality inhibits effective collaboration and communication among users.

#### Requirements of New System

* + - Incorporate robust encryption mechanisms to ensure secure communication.
    - Implement anonymity features to protect user identities and promote openness.
    - Provide intuitive user interfaces for seamless navigation and enhanced user experience

#### System Feasibility

* + - Technical feasibility: The proposed system can be implemented using modern technologies like the MERN stack and Socket.IO.
    - Economic feasibility: The project budget is within reasonable limits, considering development costs and potential ROI.
    - Operational feasibility: The new system aligns with organizational goals and can be integrated into existing workflows.

#### Activity / Process in New System / Proposed System

Features of the proposed system are:

1. Secure end-to-end encryption for all communications.
2. Anonymity options allowing users to engage without revealing identities.
3. Group creation and collaboration features for effective teamwork.
4. Intuitive user interfaces for ease of use and accessibility.
5. Real-time messaging and notification capabilities using Socket.IO.

#### Features of New System

The web application has different modules like:

* + - User Authentication and Authorization
    - Messaging System
    - Privacy and Security
    - Notification System
    - Integration with External Systems

#### Selection of Hardware & Software

* + - **Hardware: -**
      * Minimum screen resolution: 1024\*768
      * Processor minimum: 500mhz Intel Core i5 workstation or equivalent.
      * Memory minimum :512mb recommended :2gb
      * Disk space :850 MB of free disk space recommended: 1gb
    - **Software: -**
      * Front End Tool: CSS, React JS
      * Back End Tool: - Node JS
      * Frame work: Express JS
      * Database : Mongo DB

## CHAPTER 5: SYSTEM DESIGN

#### System Design & Methodology

* **User Authentication:**
  + When a user navigates to the login page, the system presents a form requesting username and password.
  + Upon form submission, the system validates the credentials against the

stored user data in the database.

* + If the credentials are valid, the system grants access to the home page; otherwise, it displays an error message.
* **User Registration:**
  + When a user chooses the sign-up option, the system presents a registration form requesting username, email, and password.
  + After the user fills in the required details and submits the form, the system

validates the data and checks for duplicate usernames or emails.

* + If the data is valid and unique, the system creates a new user record in the database and automatically logs in the user.
* **Home Page:**
  + Upon successful authentication, the system redirects the user to the home page.
  + The home page displays options for creating new groups, viewing existing

groups, and initiating chats.

* **Group Creation:**
  + When a user chooses to create a new group, the system prompts them to enter a group name and invite other users by username or email.
  + After submitting the group creation form, the system validates the input

and creates a new group record in the database, associating the creator as the group administrator.

* + The system sends invitations to the invited users, allowing them to join the group.
* **Messaging:**
  + Within a group, when a user initiates a chat, the system presents a chat interface where users can exchange messages in real-time.
  + Messages are encrypted before transmission to ensure security and

privacy.

* + The system stores message history in the database for future retrieval.
* **Logout:**
  + When a user chooses to log out, the system terminates the current session and redirects the user to the login page.
  + The system clears any session-related data and ensures that the user is

required to re-authenticate upon returning to the application.

#### Process Design

**Database Design:**

User Schema (User Model):

* Fields: \_id, username, email, password, status, new Messages
* Stores user information such as username, email, password (hashed), current status (online/offline), and the number of new messages.

Message Schema (Message Model):

* Fields: \_id, content, from, time, date, to
* Stores messages exchanged between users in different rooms.

Each message contains content, sender (from), timestamp (time and date), and the room it belongs to.

* + 1. **Process And Structure Designs**

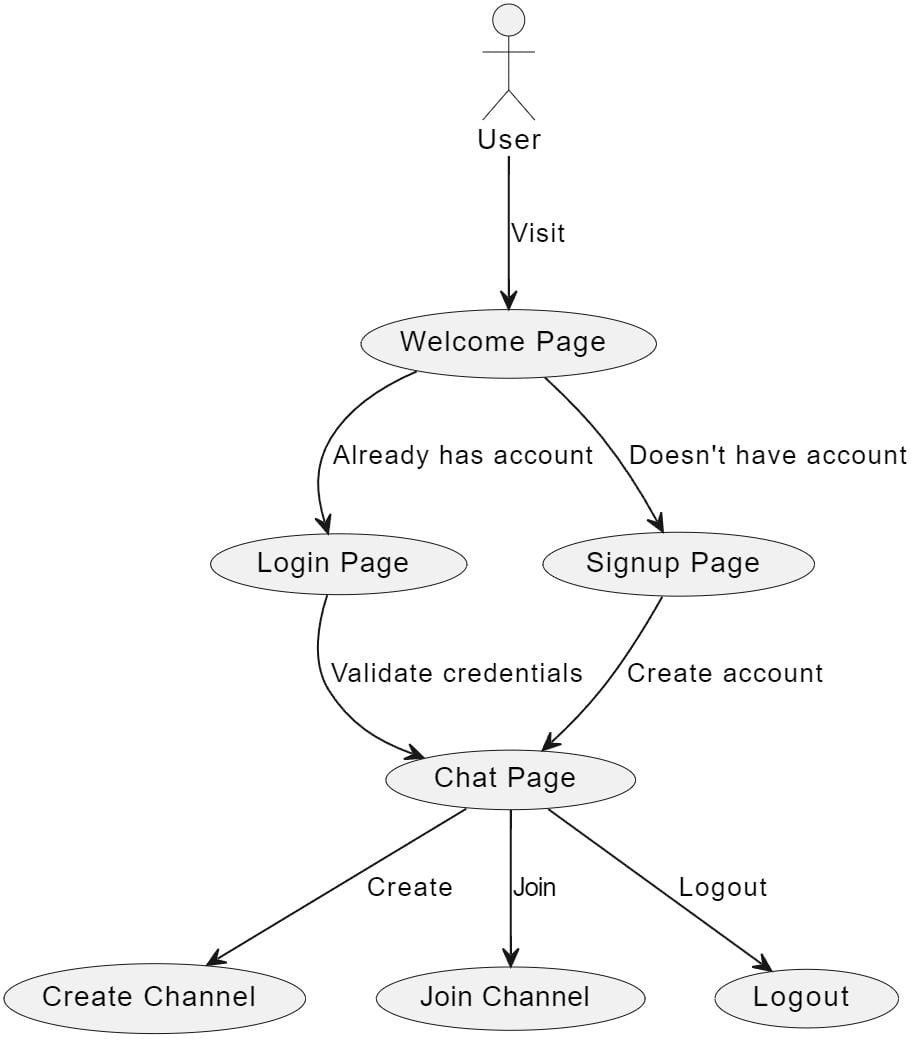
****

Fig 5.1 User Use Case Diagram

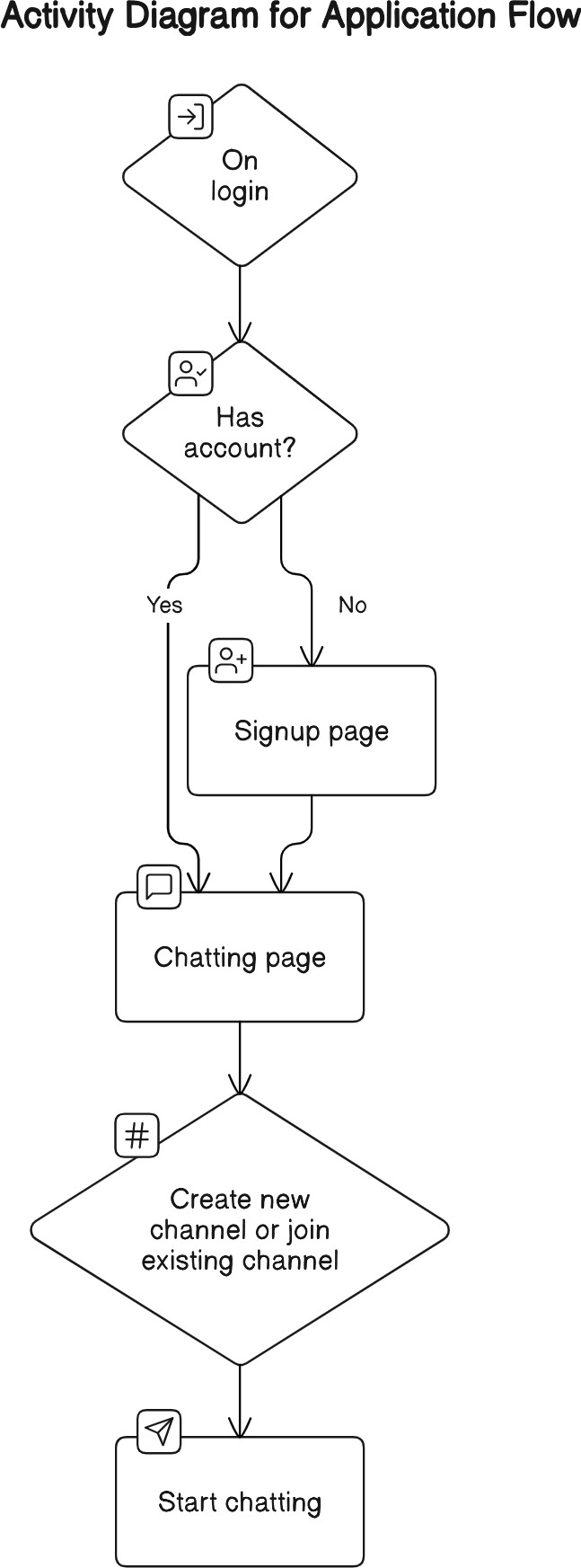
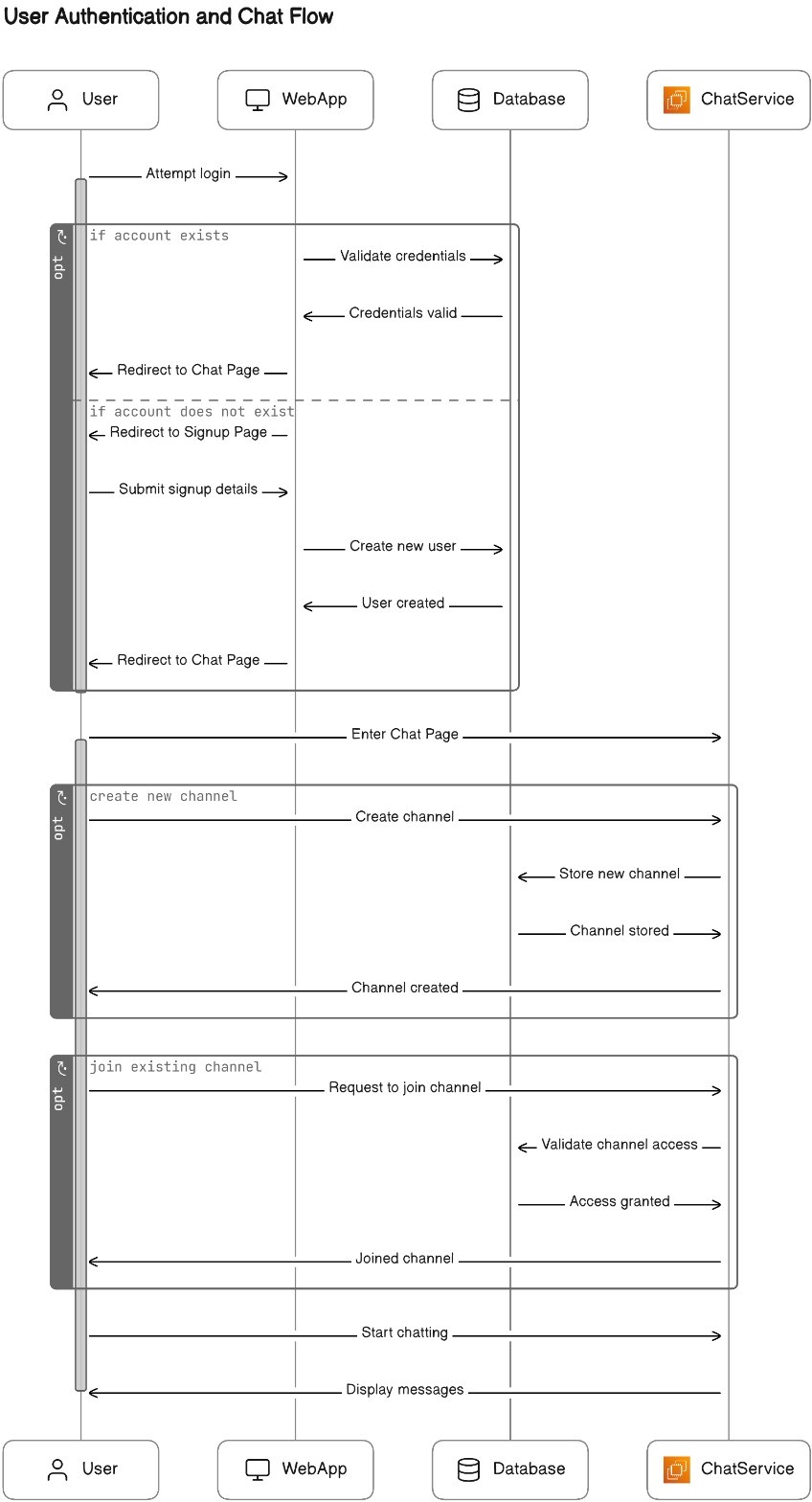
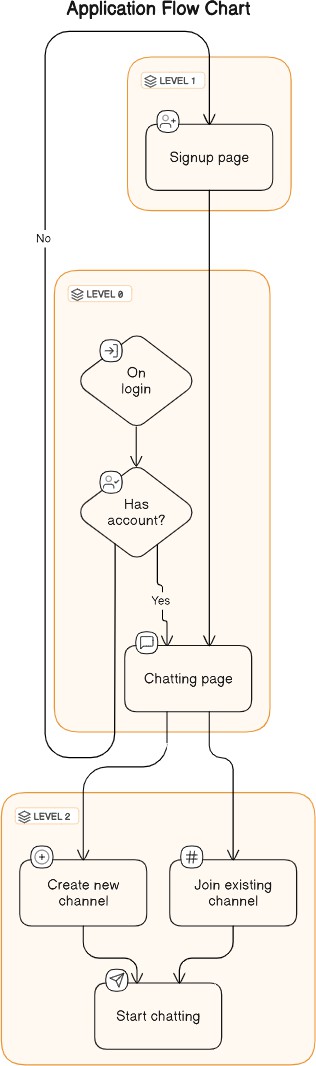


Fig 5.4 Activity Diagram 2



*Fig 5.5 Sequence Diagram 1*



*Fig 5.5 Flow Diagram*

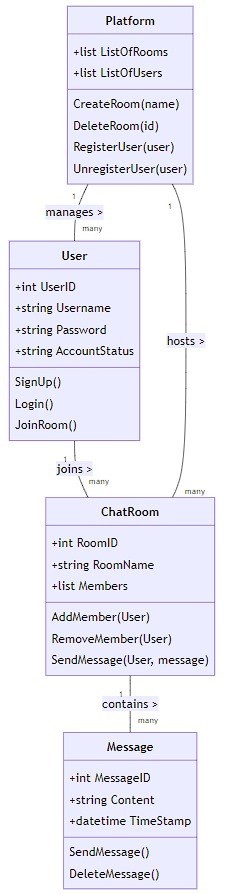
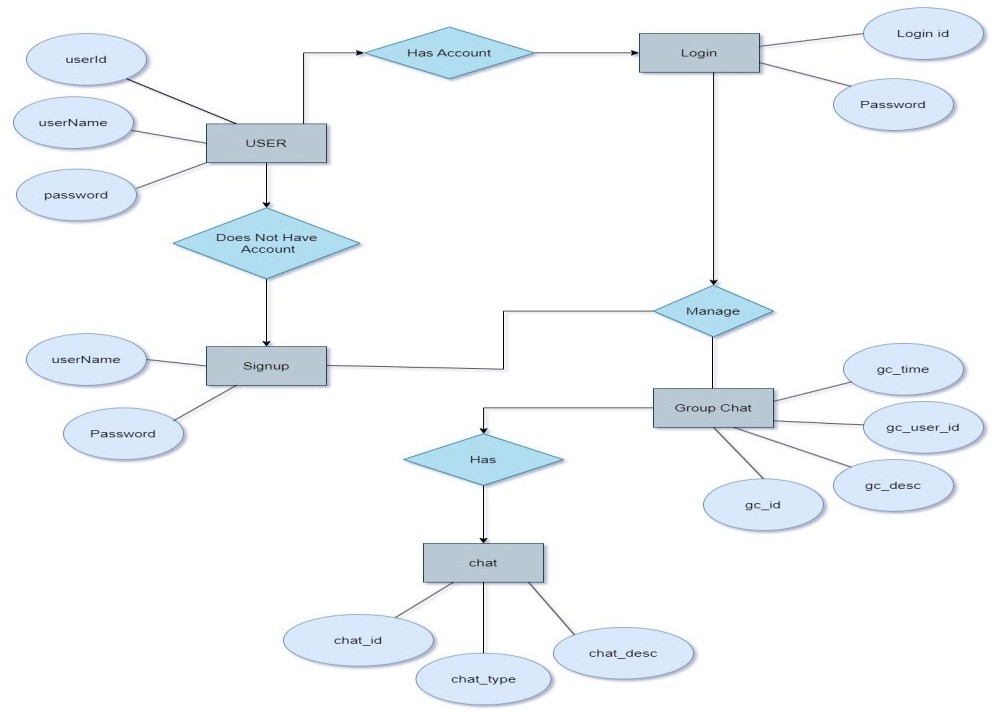


Fig 5.9 Class Diagram



*Fig 5.10 ER Diagram*

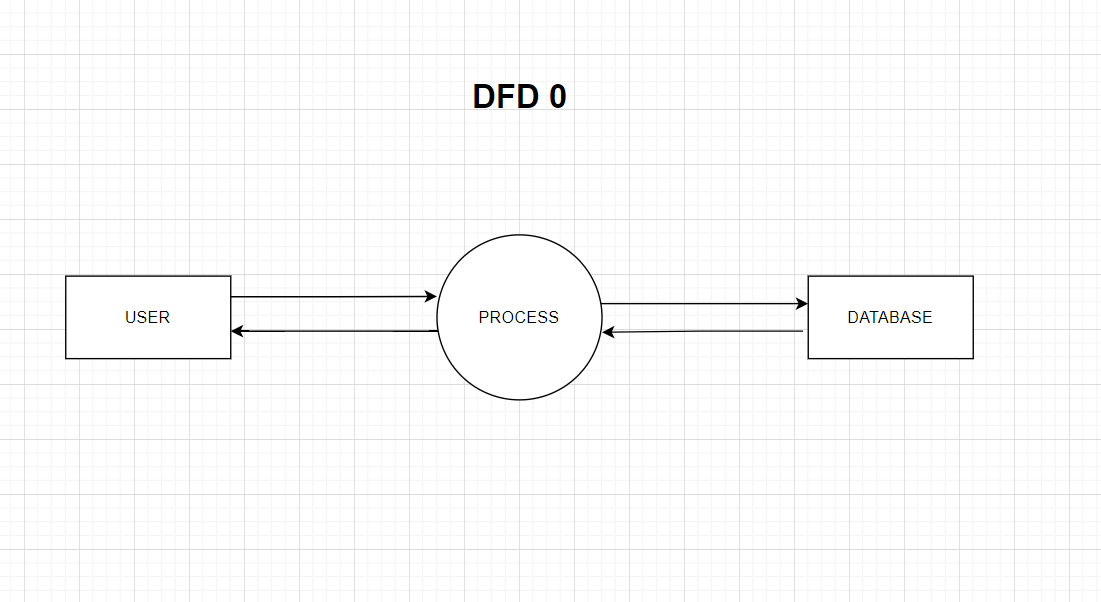


Fig 5.11 DFD Level 0

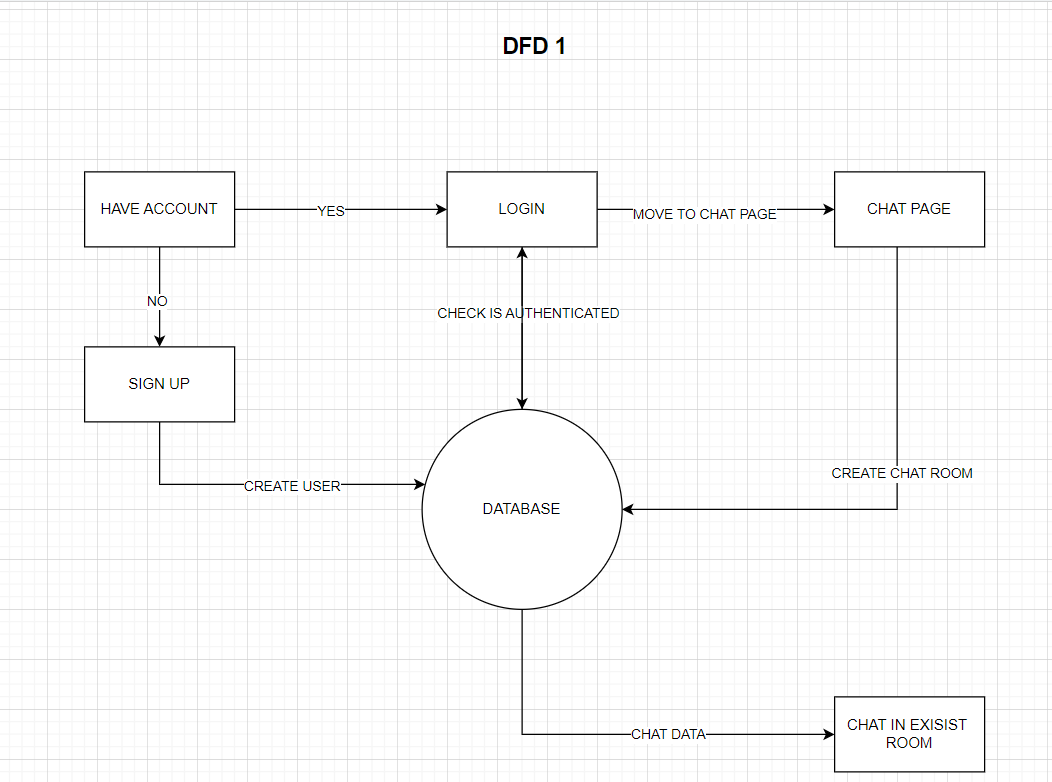


Fig 5.12 DFD Level 1

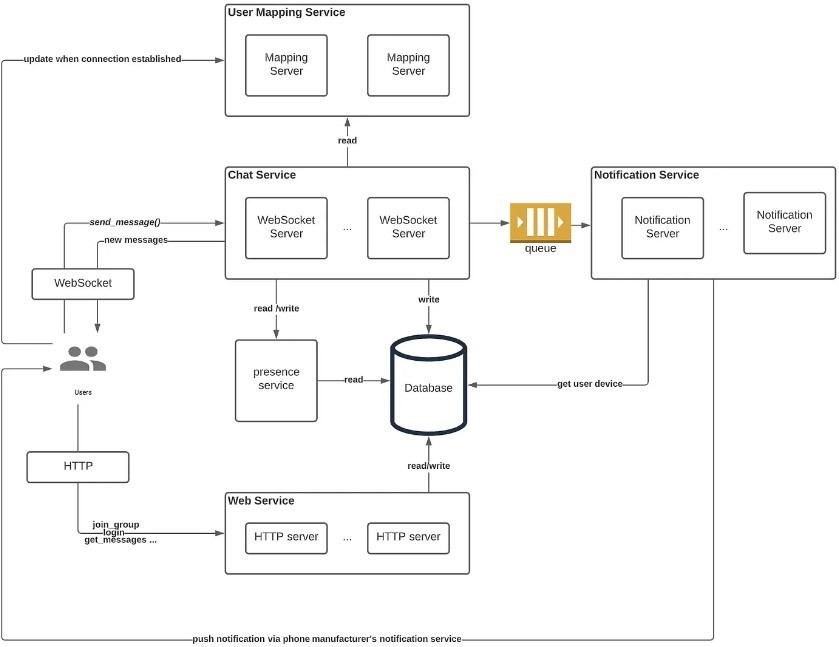


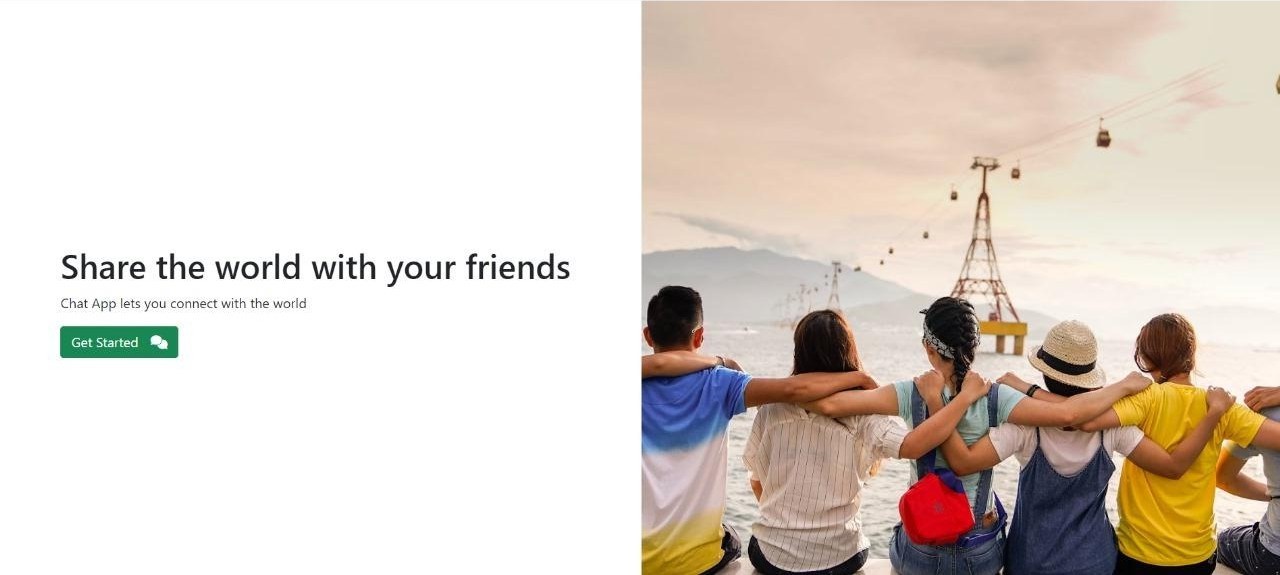
Fig 5.13 DFD Level 1

#### Interface Design

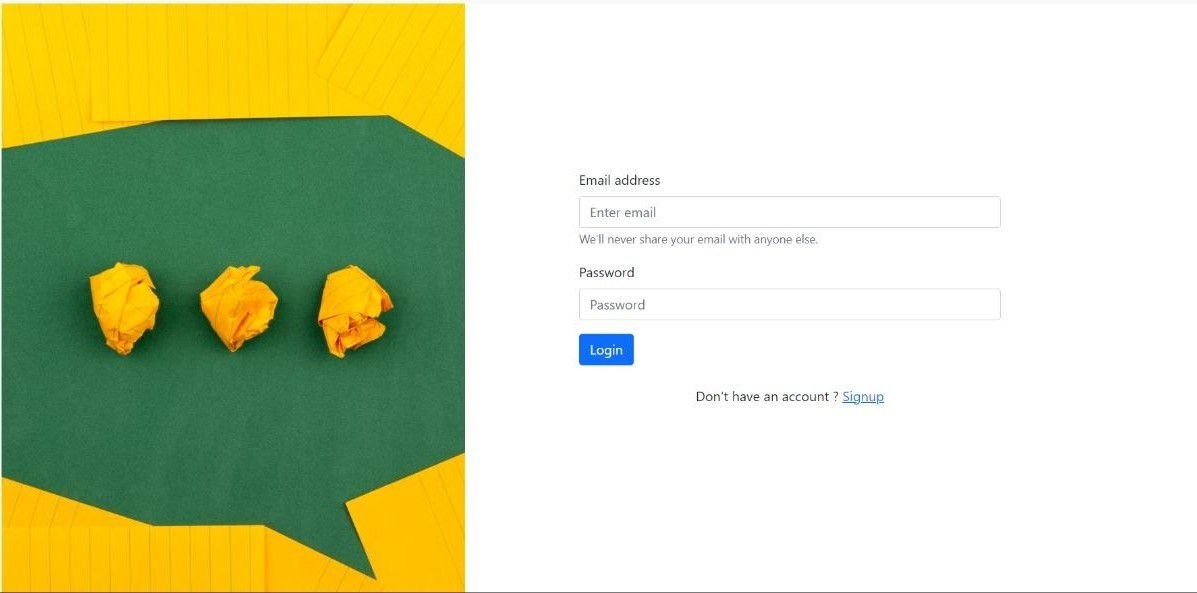
* + 1. **Sample of Forms and Interface**

Here are some snapshots of the interface and different user modules of our application.

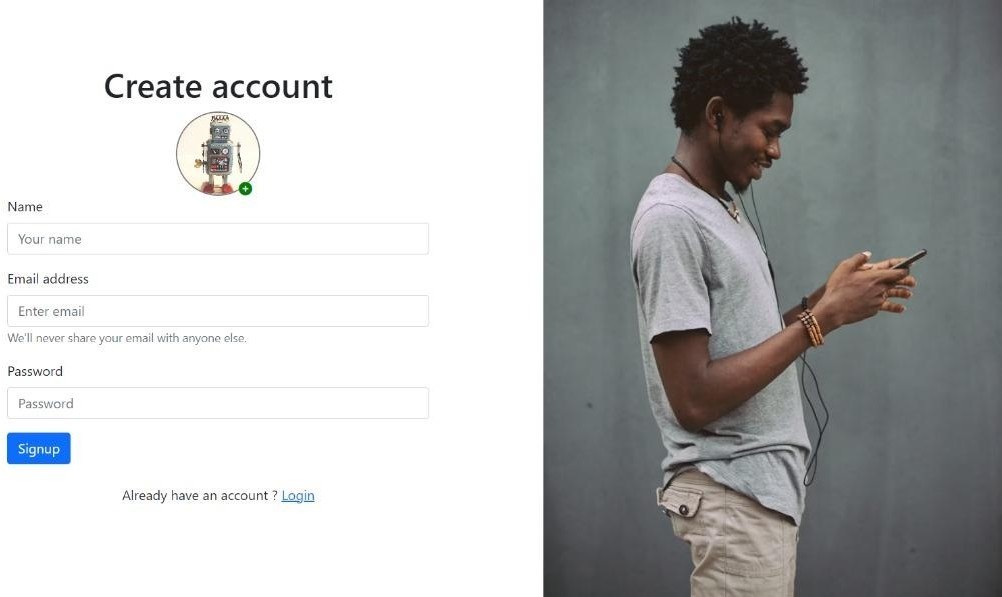
**1.Home Page:**

****

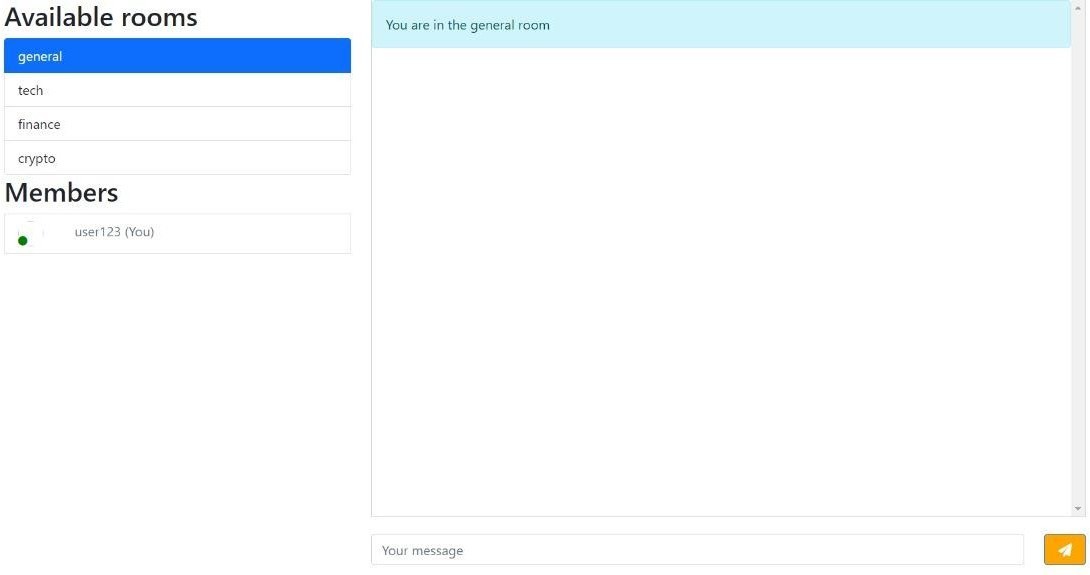
**Fig:5.3.1 Home Page**

****

**Fig:5.3.2 Login Page**



**Fig:5.3.3 Signup Page**

****

**Fig:5.3.4 chat Page**

## CHAPTER 6: IMPLEMENTATION

#### Implementation Platform / Environment

* + - Operating System: windows was used for development.
    - Text Editor: Visual Studio Code was the chosen code editor for development tasks.
    - Node.js: Node.js runtime environment was employed for server-side JavaScript execution.
    - Express.js: Express.js framework facilitated server-side application development and routing.
    - Socket.io: Socket.io library enabled real-time bidirectional communication between clients and the server.
    - Mongoose: Mongoose library was utilized for MongoDB interaction and data modelling.
    - HTML, CSS, JavaScript: These fundamental web technologies were used for building the client-side application.

#### Process / Program / Technology / Modules Specification

* Module: User Authentication
  + Manages user login, registration, password hashing, and JWT token generation.
* Module: Group Management
  + Handles creation, joining, listing, and administration of chat groups.
* Module: Real-time Messaging
* Facilitates instant messaging, message history storage, and end-to-end encryption.

## CHAPTER 7: TESTING

#### Testing Strategy

* A test plan is the cornerstone of a successful testing implementation. The testing plan represents the overall approach to the test. In many ways, the test plan serves as a summary of the test activities that will be performed. It shows how the tests will be organized, and outlines all of the tester’s needs that must be met in order to properly carry out the test. The goal of test planning is to establish the list of tasks that, if performed, will identify all of the requirements that have not been met in the software. There are many standards that can be used for developing test plans. Early in the deployment planning phase, the testing effort, and identifies the methodology that your team will use to conduct tests.
* It also identifies the hardware, software, and tools required for testing and the features and functions that will be tested. A well-rounded test plan notes any risk factors that jeopardize testing and includes a testing schedule. So, we can say that Test Planning details the activities, dependencies and effort required to conducting the system test.

#### Test Results and Analysis

Test analysis involves execution and implementation of the software with test data and examining the outputs of the software and its operational behavior to check that it is performing as required. Defect Testing is intended to find inconsistencies between a program and its specification. These inconsistencies are usually due to program faults or defects.

#### Test Cases

A test case is a document, which has a set of test data, preconditions, expected results and post conditions, developed for a particular test scenario in order to verify compliance against a specific requirement.

Test Case acts as the starting point for the test execution, and after applying a set of input values; the application has a definitive outcome and leaves the system at some end point or also known as execution post condition.

Testing was done by us under the guidance of our mentors. If an error occurred, we were supposed to note it down and make changes that would reflect positively on the website.

|  |  |  |
| --- | --- | --- |
| **Sr. No.** | **Test Case** | **Status** |
| 1 | To check if functionalities of Home page is working properly | Pass |
| 2 | To check if login functionality page is working properly | Pass |
| 3 | To check if signup functionality page is working properly. | Pass |
| 4 | To check if logout functionality page is working properly | Pass |
| 5 | To check if login messages are working properly | Pass |

**Table 7.2.1.1: Test cases for User**

## CHAPTER 8: CONCLUSION AND DISCUSSION

#### Overall Analysis of Internship Viabilities

The internship demonstrated viability through skill development, industry exposure, networking, personal growth, and project experience. It provided hands-on learning in frontend and backend development, database management, and project coordination. Collaborating with professionals offered insights into industry practices, while networking opportunities expanded connections. Feedback aided personal growth, and project involvement enhanced understanding of project lifecycles. Overall, the internship aligned with career goals, offering valuable experiences and potential career pathways.

The working model of the project/web application will undergo some change.

#### Meeting with Institute Mentor

****

**Fig 8.1 Meeting with Mentor**

#### Problem Encountered and Possible Solutions

Throughout the internship, challenges arose, such as managing tight deadlines, navigating technical complexities, and communication barriers among team members. Solutions include implementing effective time management strategies, allocating dedicated time for learning, seeking guidance from experienced colleagues, and fostering open communication channels and a culture of transparency within the team. These measures can help mitigate stress, improve project coordination, and enhance overall productivity and collaboration.

#### Summary of Internship work

The internship provided a comprehensive learning experience, encompassing various aspects of frontend and backend development, database management, and project coordination. Engaging in real-world projects allowed for practical application of theoretical knowledge and honing of practical skills. Challenges such as managing tight deadlines and navigating technical complexities were encountered, yet addressed through effective time management, continuous learning, and fostering open communication within the team. The experience fostered personal growth, expanded professional networks, and provided valuable insights into industry practices and dynamics. Overall, the internship served as a valuable stepping stone towards achieving career goals and paved the way for future opportunities in the field.

#### Limitation and Future Enhancement

##### Limitations

* + - * Limited Features: Incognito Messenger may lack certain functionalities compared to established platforms.
      * Scalability Challenges: Growing user base may lead to performance issues.
      * Security Concerns: Potential vulnerabilities may compromise user data.
      * Compatibility Issues: Application may face challenges across devices and systems.
      * User Experience Limitations: Design or technical constraints may impact usability.
      * Integration Constraints: Difficulty integrating with external services may limit functionality.
      * Limited Support: Resource constraints may affect customer support and maintenance.
      * Development Constraints: Technical challenges or resource limitations may lead to delays.

##### Future Work/Future Enhancement

* + - * Feature Expansion: Introducing new features such as voice and video calling, group chats, message scheduling, or file sharing to enhance user experience and functionality.
      * Improved Security Measures: Implementing end-to-end encryption, two-factor authentication, or biometric authentication to enhance user privacy and security.
      * Enhanced User Interface: Updating the user interface with modern design principles, customizable themes, and intuitive navigation for an improved user experience.
      * Performance Optimization: Optimizing application performance to ensure faster load times, smoother interactions, and efficient use of system resources.

# REFERENCES

1. Anonymous Messaging and Privacy <https://blog.kraden.com/anonymous-messaging>
2. web dev anonymity

<https://web.dev/articles/digging-into-the-privacy-sandbox>

1. Databases for Web Applications

<https://www.mongodb.com/>

1. Chat Application Development

<https://www.cometchat.com/tutorials/building-chat-app>

1. Web Security and Privacy

<https://developer.mozilla.org/en-US/docs/Web/Privacy>