SYNOPSIS

ON

"health Care Sector"

Submitted In the Partial Fulfilment for the Award of The

Degree Of

MASTER OF COMPUTER APPLICATION

UTTARANCHAL UNIVERSITY
NAAC GRADE A+

Under The Supervision of: -

Dr. Amarjeet Rawat Sir

Mr. Parminder Singh

Submitted By: -

Saurabh, Satish Kumar,

Badal Kushwaha, Aditya,

Saumyaditya

MCA 1st Semester

Sec: - B

TABLE OF CONTENT

LIST OF FIGURES

1. INTRODUCTION

Transforming our world: the 2030 agenda for sustainable development outlines a transformative vision with 17 sustainable development goals (SDGs) for economic, social and environmental development.

[1] While only SDG 3, to ensure healthy lives and promote well-being for all at all ages, focuses on human health, all goals are interrelated. This issue of the Bulletin of the World Health Organization examines the relationship between health and the SDGs.

Implementing the 2030 agenda requires a multistakeholder, multi-actor response. Innovations and development in policy, technology and research must include dialogue between governments, the private sector, civil society organizations and nongovernmental organizations; most importantly, strong community involvement is needed

.[2] The focus of the 2030 agenda on addressing country-level needs is based on the engagement of all actors and sectors, as opposed to the traditional top-down, single-sector approach.

Intersectoral governance is at the core of how to approach the social determinants of health,

[3] and how to address antimicrobial resistance through a one health approach.

[4] More specifically, to reach target 3.8, to achieve universal health coverage, Member States must ensure access to health services by introducing protection against catastrophic health expenditure

[5] Obstacles and financial hardships associated with weak health systems and inadequate financing mechanisms not only exacerbate health inequities, but also jeopardize the achievement of other SDGs

[6] For instance, catastrophic health spending

[7] has pushed almost 100 million people into poverty annually, while leaving over half of the world's population without access to essential health services

.[5] Most out-of-pocket payments, the cause of catastrophic expenditure, arise from high medication costs, which are likely to increase in line with the global increase in noncommunicable diseases and the need for long-term treatment.

2. OBJECTIVE

- To Enhance access to healthcare services in underserved communities.
- Implement eco-friendly practices in healthcare facilities.
- Promote preventive healthcare measures.
- Raise awareness about healthy lifestyles and disease prevention.
- Engage community members in decision-making processes regarding healthcare initiatives.

  2. Search functionality for easy retrieval of past conversations.

  3. Integration with productivity tools or calendar for better organization.

3. SYSTEM ANALYSIS

A feasibility study for a Local Area Network (LAN) Live Chat Application is crucial to assess the viability and potential success of the project. The study should cover various aspects to ensure that the proposed system is technically, economically, and operationally feasible. Below are key components typically included in a feasibility study

System analysis for a Local Area Network (LAN) Live Chat Application involves a detailed examination of the requirements, objectives, and constraints of the proposed system. This process helps define the scope of the project and ensures that the developed system meets the needs of users and stakeholders

3.1) Identification of Need:

1. Enhanced Communication: The LAN Chat Application aims to provide a dedicated and secure platform for users to communicate effectively within the LAN, fostering collaboration and information exchange.

2. Security and Privacy: It addresses security concerns by ensuring that communications stay within the LAN, reducing the risk of unauthorized access or data leakage.

3. Centralized Control: The application allows administrators to monitor and manage user interactions, ensuring compliance with company policies.

4. Customization: The LAN Chat Application can be customized to meet the specific needs of the LAN, including the integration of features like file sharing, voice and video calls, and user authentication.

5. Scalability: The system is designed to accommodate an increasing number of users and adapt to the growing needs of the LAN.

3.2) Preliminary Investigation

The "Preliminary Investigation of Live Chat App System Analysis" is a comprehensive exploration into the intricacies of Live Chat Apps, focusing on system analysis to uncover insights into functionality, efficiency, and user experience. The investigation aims to lay the foundation for understanding the system architecture, user interactions, and potential areas for improvement within Live Chat Apps.

1. Introduction:

Brief overview of the significance of Live Chat Apps in modern communication.
Introduction to the need for a systematic analysis to enhance functionality and user experience.

2. Objective of the Investigation:

Clearly defined objectives, including the assessment of system architecture, user interface, and overall system efficiency.
Identification of potential bottlenecks or areas of improvement in existing Live Chat App systems.

3. Methodology:

Description of the research methods, including a combination of system logs analysis, user feedback surveys, and interviews with system administrators and users.
Selection criteria for the Live Chat Apps under investigation and the rationale behind their selection.

4. System Architecture:

In-depth exploration of the system architecture of selected Live Chat Apps.
Analysis of server-client interactions, data storage mechanisms, and scalability considerations.

5. User Interface Analysis:

Evaluation of the user interface design, focusing on user-friendliness, accessibility, and intuitiveness.
Identification of potential areas for interface optimization based on user behaviour and feedback.

6. Functionality and Features:

Examination of the core functionalities and features offered by Live Chat Apps.
Assessment of feature usability, relevance, and the overall impact on user satisfaction.

7. Performance Metrics:

Definition and analysis of key performance metrics, including response time, uptime, and system reliability.
Comparative analysis of performance across different Live Chat Apps.

8. Security and Privacy Considerations:

Investigation into the security measures implemented within Live Chat Apps to safeguard user data and communications.
Evaluation of privacy policies and their alignment with user expectations and regulatory standards.

9. Integration with Emerging Technologies:

Exploration of how Live Chat Apps integrate with emerging technologies such as AI, machine learning, and natural language processing.
Assessment of the impact of these integrations on overall system performance and user experience.

10. Recommendations and Future Considerations:

Proposals for system improvements based on the analysis conducted.
Consideration of emerging trends in technology and communication that could shape the future development of Live Chat Apps.

11. Conclusion:

Summary of key findings from the system analysis.

Implications for developers, system administrators, and stakeholders in enhancing the performance and user experience of Live Chat Apps.

This preliminary investigation serves as a foundation for a more in-depth exploration into the Live Chat App system, offering valuable insights that can inform the development, optimization, and innovation within this dynamic realm of digital communication.

3.3) FEASIBILITY STUDY

A feasibility study for a Local Area Network (LAN) Live Chat Application is crucial to assess the viability and potential success of the project. The study should cover various aspects to ensure that the proposed system is technically, economically, and operationally feasible. Below are key components typically included in a feasibility study:

1. Technical Feasibility:

Technical Requirements: Evaluate whether the technology needed for the LAN live chat application is readily available or needs significant development.

Expertise: Assess if the required technical skills are available within the team or if external expertise is necessary.

Compatibility: Check the compatibility of the proposed system with existing hardware, software, and network infrastructure.

2. Economic Feasibility:

Cost-Benefit Analysis: Evaluate the estimated costs associated with development, deployment, and maintenance against the expected benefits.

Return on Investment (ROI): Assess the potential financial gains and determine if the project is economically viable.

Budget Constraints: Consider budget limitations and evaluate if the project can be completed within the allocated funds.

3. Operational Feasibility:

User Acceptance: Assess the willingness of end-users to adopt the new LAN live chat application.

Training Needs: Identify the training requirements for users and administrators.

Impact on Operations: Analyses how the new system will affect existing business processes and workflow.

4. Legal and Regulatory Feasibility:

Compliance: Ensure that the proposed system complies with local, national, and international laws and regulations.

Privacy and Security: Address data privacy and security concerns to meet legal requirements and user expectations.

5. Schedule Feasibility:

Timeline: Develop a realistic project timeline considering the complexity of the application and other dependencies.

Dependencies: Identify and account for dependencies that could impact the project schedule.

3.4) Existing System:

An existing system for a Local Area Network (LAN) Live Chat Application may have various components and features. Here's a general description of what such a system could include:

1. User Authentication:

Allows users to log in with unique credentials to access the chat application securely.

2. User Management:

Admin capabilities to manage users, roles, and permissions.
User profiles with customizable settings.

3. Chat Interface:

Real-time chat interface for users to send and receive messages.
Support for different types of messages (text, images, files, etc.).
Group chat functionality for multiple users to communicate simultaneously.

- The specifics of an existing LAN Live Chat Application will depend on the software or platform you are using. Different applications may have unique features or focus on particular aspects of communication and collaboration within a local network.

3.5) PROPOSED SYSTEM

When proposing a new system for a Local Area Network (LAN) Live Chat Application, it's essential to consider both the shortcomings of existing systems and the evolving needs of users. Below are key components and features for a proposed LAN Live Chat Application:

1. Enhanced User Interface:

- A modern and intuitive interface for an improved user experience.
- Customizable themes and layouts to cater to diverse user preferences.

2. Advanced Multimedia Support:

- Expanded support for multimedia content, including GIFs, videos, and other file types.
- Integration with third-party services for seamless multimedia sharing.

3. Real-time Collaboration Features:

- Collaborative tools such as document sharing, screen sharing, and real-time editing for increased productivity.

4. Cross-Platform Synchronization:

- Seamless synchronization across multiple devices (desktop, mobile, tablet) for uninterrupted communication.

5. Geotagging and Location Sharing:

- Geotagging capabilities to share location information for improved coordination.

3.6) Functional Requirement

Functional requirements for a Local Area Network (LAN) live chat application would define the features and capabilities that the system must have to meet its objectives. Here are some functional requirements for a LAN live chat application:

1. User Authentication and Authorization:

- Users should be able to log in securely using unique credentials.
- Access levels should be defined for different user roles (e.g., admin, regular user).

2. User Presence and Status:

- Users should be able to set their online, offline, or busy status.
- The system should display the real-time presence of users.

3. Chat Rooms:

- Users should be able to create and join chat rooms.
- Different chat rooms should support various topics or purposes.

4. One-on-One Messaging:

- Users should be able to send private messages to each other.
- The application should support real-time delivery and receipt of messages.

5. Group Messaging:

- Users should be able to create and participate in group chats.
- Group chat functionality should include the ability to add/remove participants.

3.7) Functional Requirement

Non-functional requirements define the qualities or attributes that describe how the system should behave, rather than specifying specific behaviour's. Here are some non-functional requirements for a Local Area Network (LAN) live chat application:

1. Performance:

- The application should respond to user actions (e.g., message sending, receiving, loading) within a specified timeframe.
- It should be capable of handling a concurrent user load without significant degradation in performance.

2. Scalability:

- The system should be scalable to accommodate an increase in the number of users and messages over time.
- Scalability should not compromise performance.

3. Reliability:

- The application should be available for use during agreed-upon hours of operation.
- It should have minimal downtime for maintenance or updates.

4. Availability:

- The system should have a high level of availability, ensuring users can access the chat application when needed.
- Redundancy and failover mechanisms may be implemented to enhance availability.

3.8) Hardware & Software Requirement

A. Being Used:

Hardware Used:

- AMD Ryzen 5 5600H with Radeon Graphics 3.30 GHz
- Installed RAM- 8GB
- SDD- 512GB.
- Graphics- Radeon Graphics 3.30 GHz

Software Used:

- Operating system: Windows 11 Home Single Language
- IDE: Visual Studio Code
- Web server and Database: Firebase
- Python

B. Minimum Requirement:

- Processor: AMD Ryzen 5 5600H
- RAM: 512MB
- Disk space: 800MB

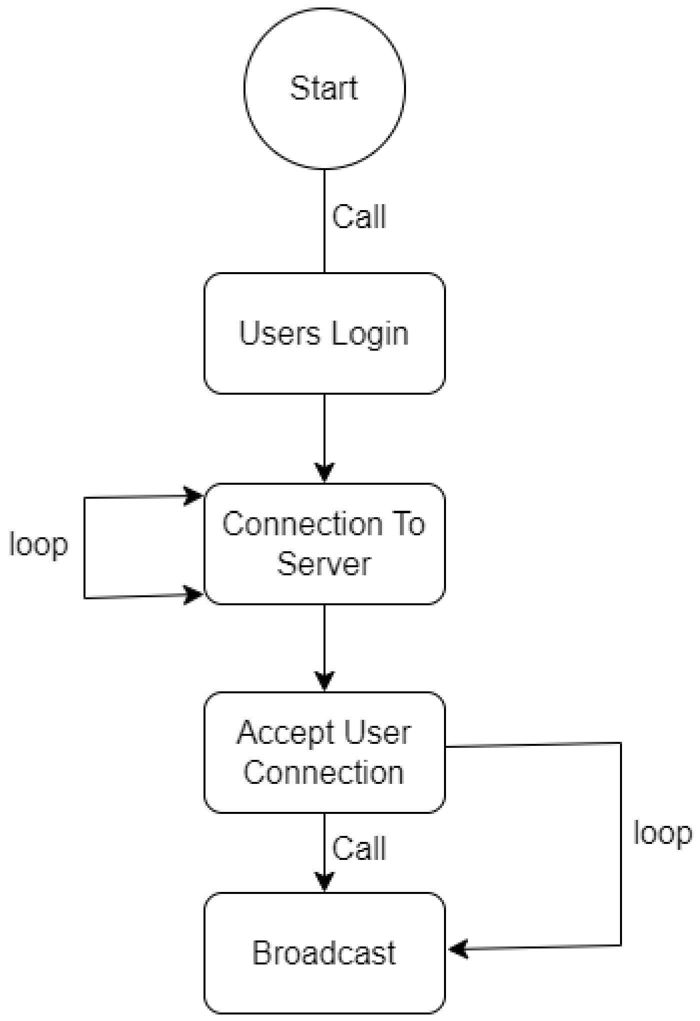
3.9) Project Scheduling

Gantt Chart

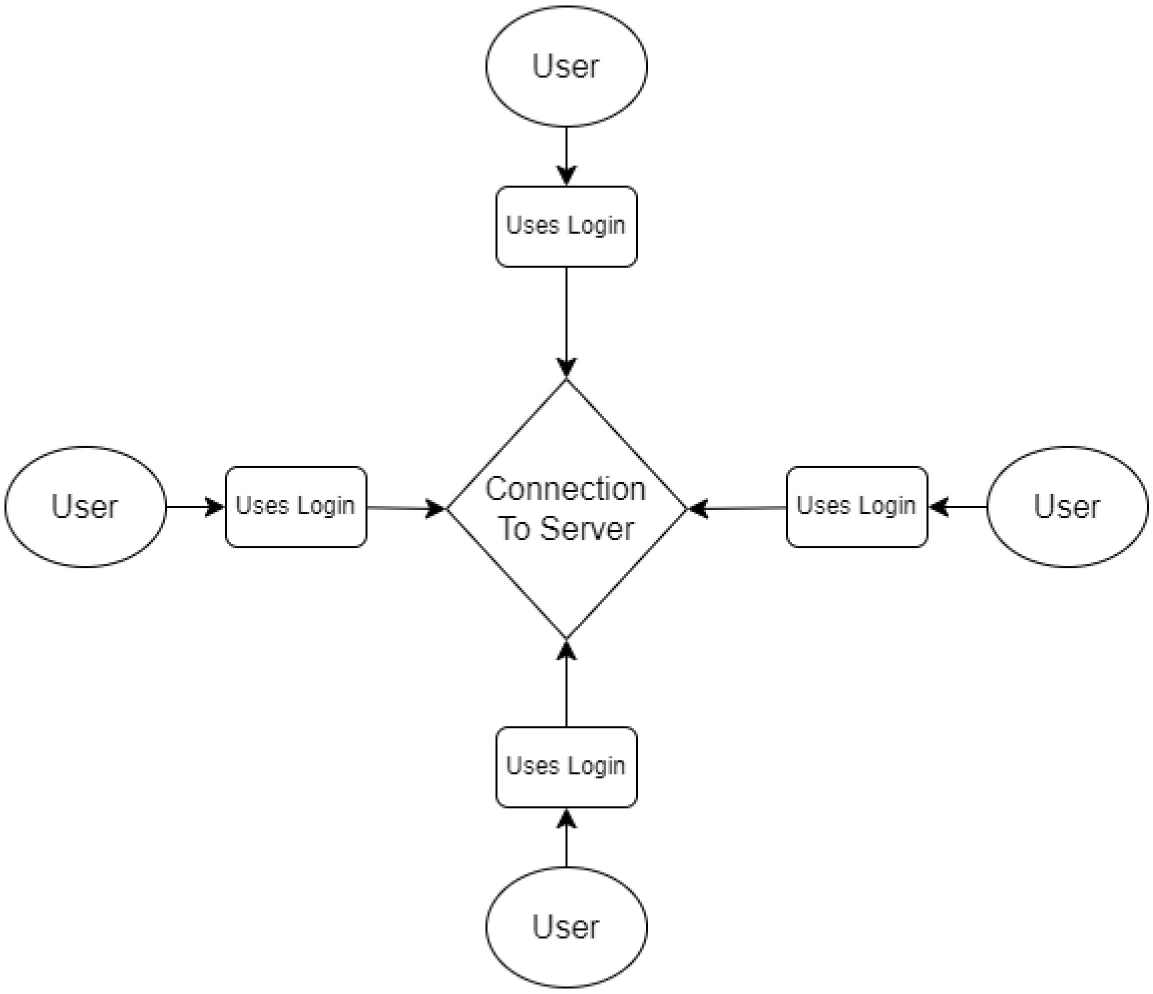
A	B	C	D	E
SDLC PHASE	**PROJECT ACTIVITY**	**START DATE**	**DURATION**	**END DATE**
PHASE 1	Requirement Analysis	15/02/2024	29 Days	15/03/2024
PHASE 2	Feasibility Study	16/03/2024	15 Days	31/03/2024
PHASE 3	Design	01/04/2023	13 Days	13/04/2024
PHASE 4	Coding	14/04/2024	20 Days	04/05/2024
PHASE 5	Testing	06/05/2024	10 Days	16/05/2024

4. System Design

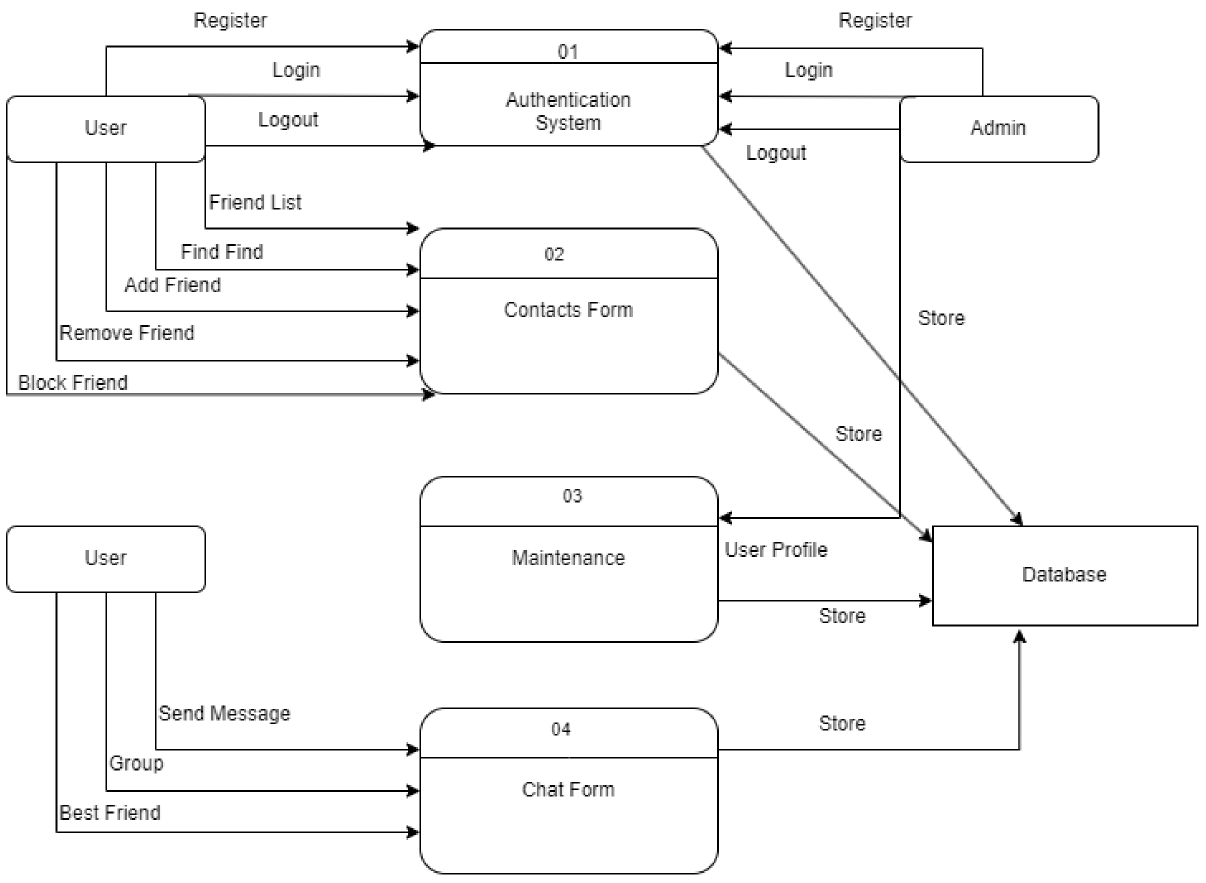
Flow Chart



ER Diagram

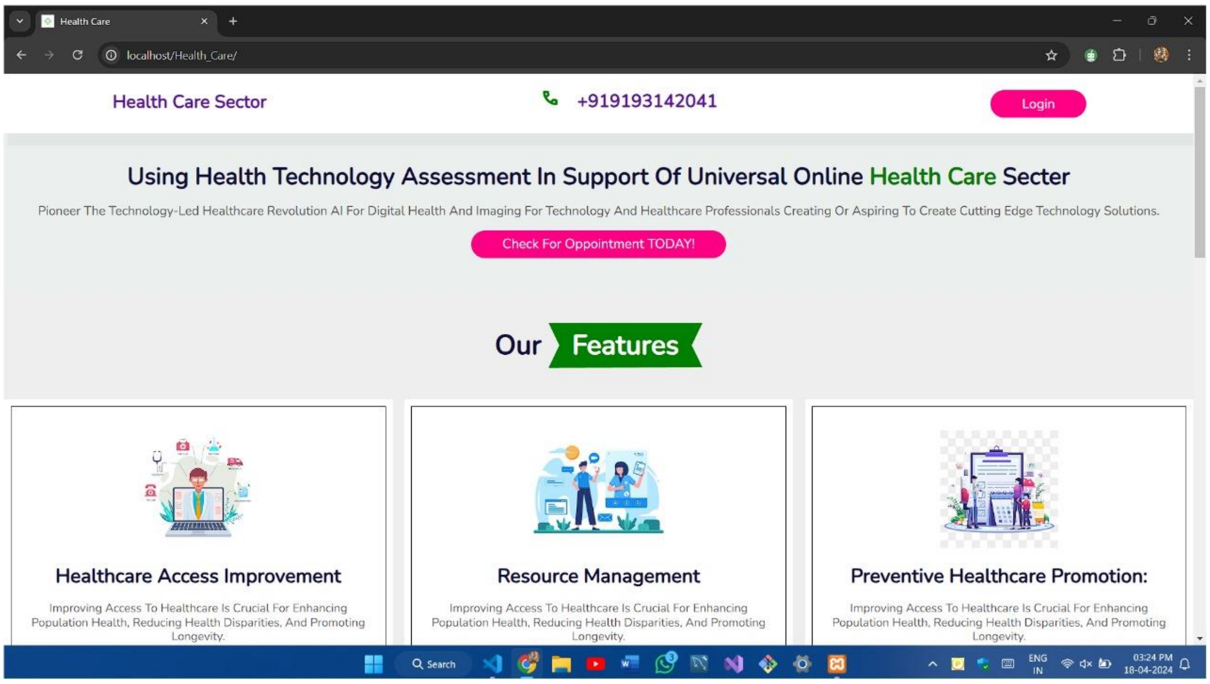

User
Uses Login
User
Uses Login
Connection To Server
Uses Login
User
Uses Login
User

Data Flow Diagram



User Interface Design

Home Page



Health Care Sector

+919193142041

Using Health Technology Assessment In Support Of Universal Online Health Care Secter

Pioneer The Technology-Led Healthcare Revolution AI For Digital Health And Imaging For Technology And Healthcare Professionals Creating Or Aspiring To Create Cutting Edge Technology Solutions.

Our Features

Healthcare Access Improvement

Improving Access To Healthcare Is Crucial For Enhancing Population Health, Reducing Health Disparities, And Promoting Longevity.

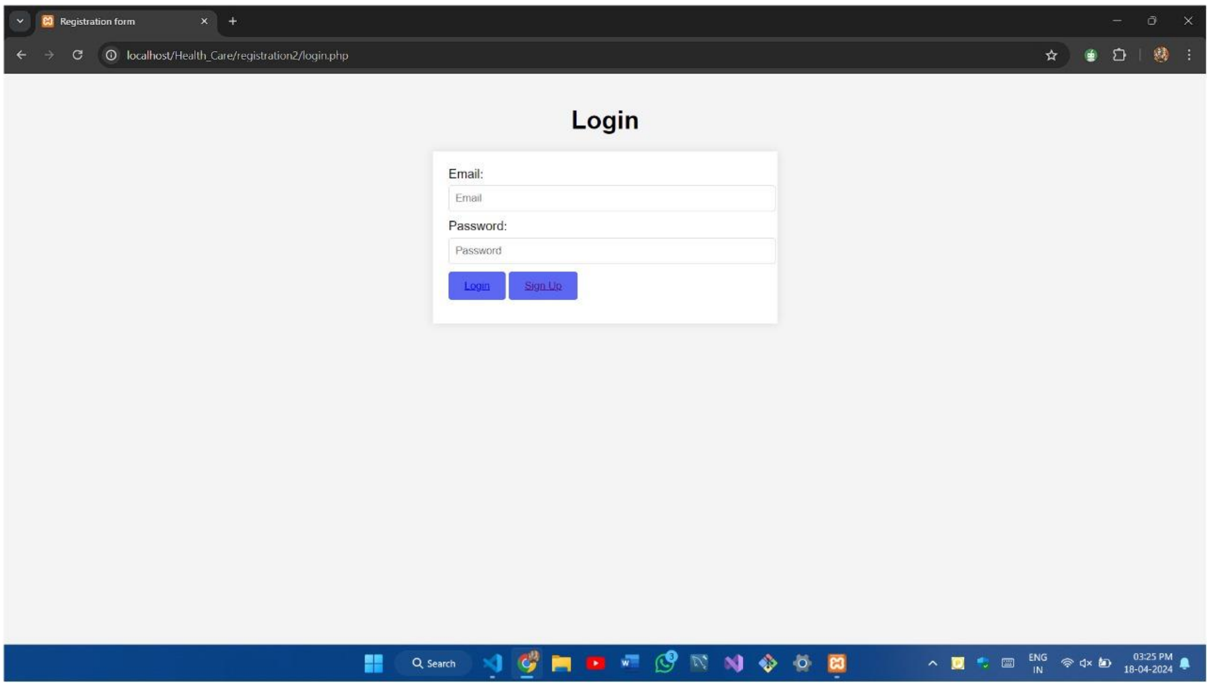
Resource Management

Improving Access To Healthcare Is Crucial For Enhancing Population Health, Reducing Health Disparities, And Promoting Longevity.

Preventive Healthcare Promotion:

Improving Access To Healthcare Is Crucial For Enhancing Population Health, Reducing Health Disparities, And Promoting Longevity.

Login Page


Registration form

localhost/Health_Care/registration2/login.php

Login

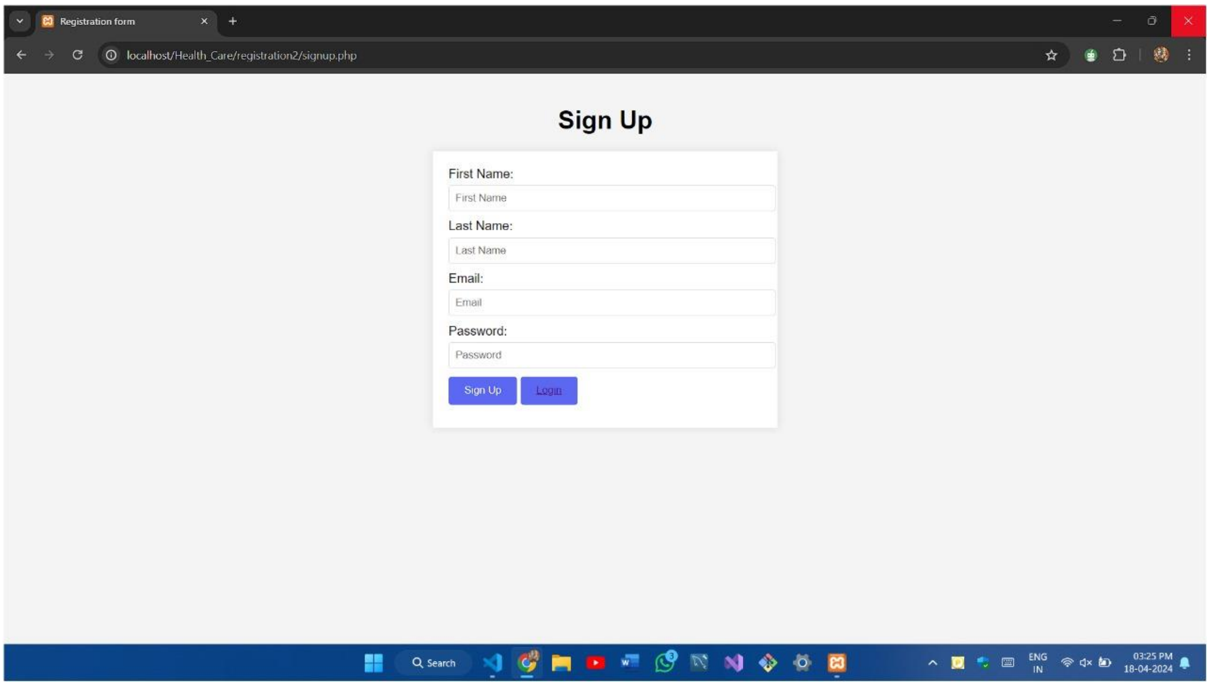
Email:

Email

Password:

Password

Sign Up Page


Registration form

localhost/Health_Care/registration2/signup.php

Sign Up

First Name:

First Name

Last Name:

Last Name

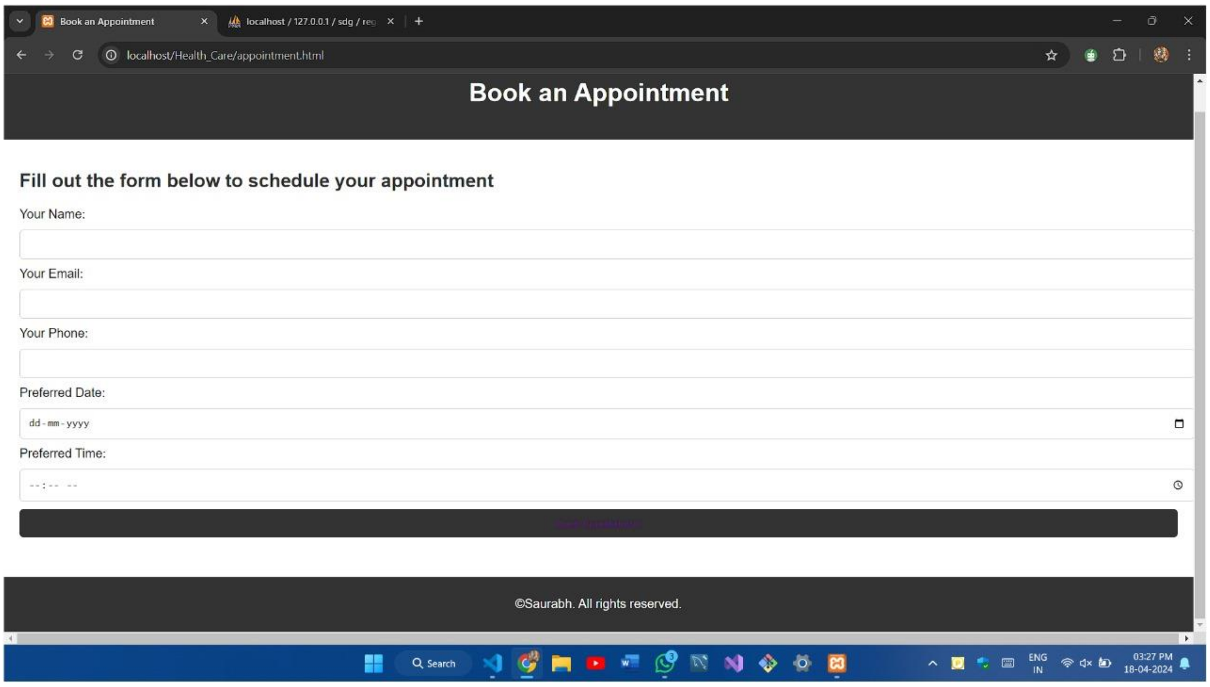
Email:

Email

Password:

Password

Booking Appointment


Book an Appointment

localhost/Health_Care/appointment.html

Book an Appointment

Fill out the form below to schedule your appointment

Your Name:

Your Email:

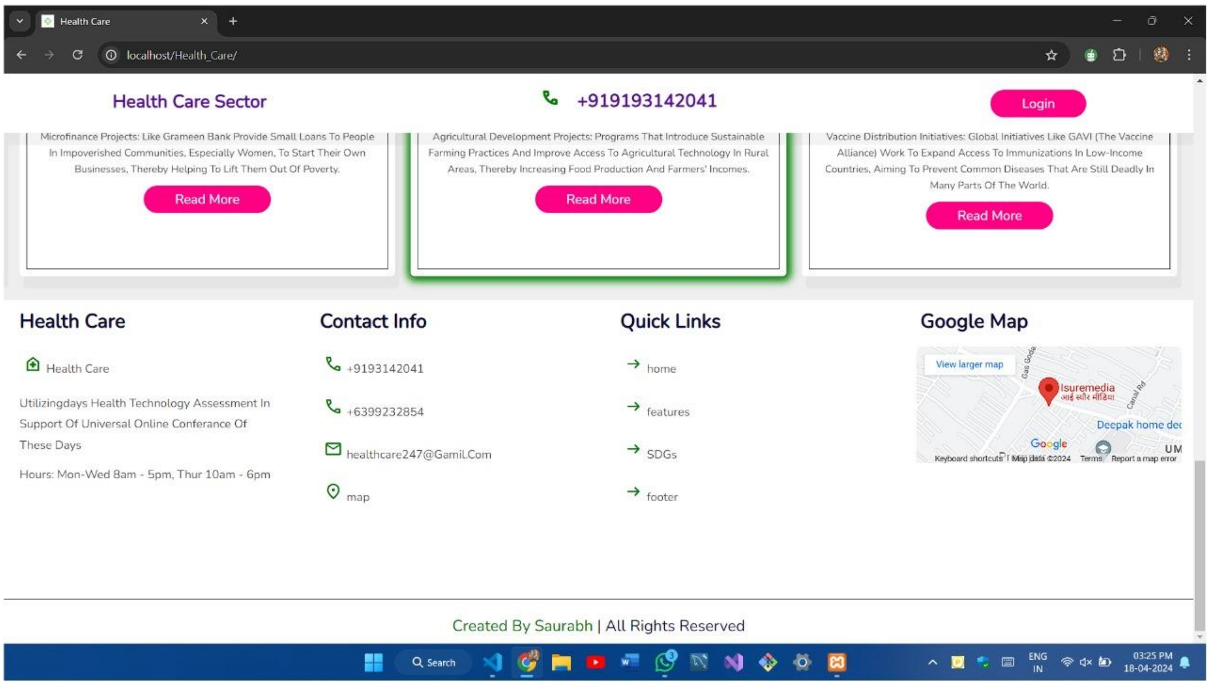
Your Phone:

Preferred Date: dd-mm-yyyy

Preferred Time: --:-- --

©Saurabh. All rights reserved.

Footer Page



Health Care Sector +919193142041 Login

Microfinance Projects: Like Grameen Bank Provide Small Loans To People In Impoverished Communities, Especially Women, To Start Their Own Businesses, Thereby Helping To Lift Them Out Of Poverty.

Read More

Agricultural Development Projects: Programs That Introduce Sustainable Farming Practices And Improve Access To Agricultural Technology In Rural Areas, Thereby Increasing Food Production And Farmers' Incomes.

Read More

Vaccine Distribution Initiatives: Global Initiatives Like GAVI (The Vaccine Alliance) Work To Expand Access To Immunizations In Low-Income Countries, Aiming To Prevent Common Diseases That Are Still Deadly In Many Parts Of The World.

Read More

Health Care

Health Care

Utilizingdays Health Technology Assessment In Support Of Universal Online Conferance Of These Days

Hours: Mon-Wed 8am - 5pm, Thur 10am - 6pm

Contact Info

+9193142041

+6399232854

healthcare247@Gamil.Com

map

Quick Links

home

features

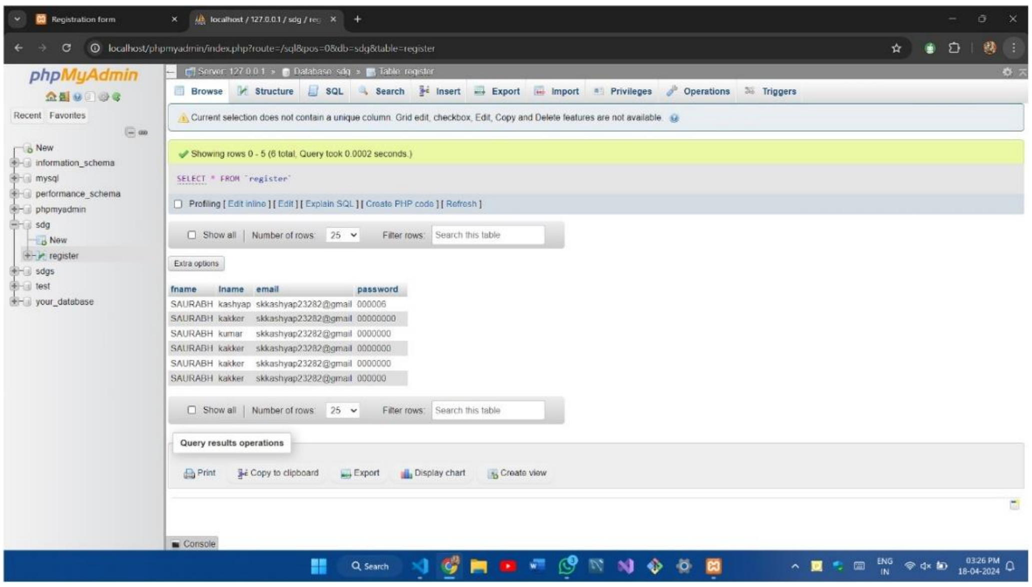
SDGs

footer

Google Map

Created By Saurabh | All Rights Reserved

Database SQL



⚠ Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

✓ Showing rows 0 - 5 (6 total, Query took 0.0002 seconds.)

```
SELECT * FROM `register`
```

fname	lname	email	password
SAURABH	kashyap	skkashyap23282@gmail	000006
SAURABH	kakker	skkashyap23282@gmail	00000000
SAURABH	kumar	skkashyap23282@gmail	0000000
SAURABH	kakker	skkashyap23282@gmail	0000000
SAURABH	kakker	skkashyap23282@gmail	0000000
SAURABH	kakker	skkashyap23282@gmail	000000

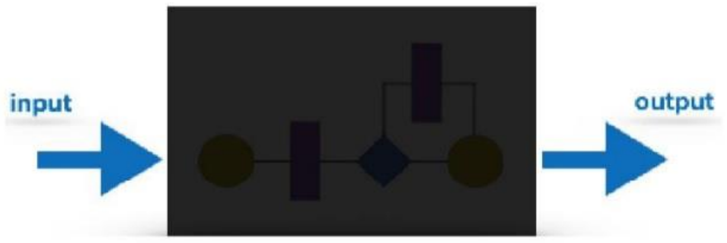
5.Testing

The "Testing for LAN Live Chat Application" project aims to ensure the robustness, functionality, and security of a Local Area Network (LAN) live chat application. In a rapidly evolving digital landscape, effective communication tools within a closed network environment are crucial. This testing initiative focuses on identifying and rectifying potential issues to deliver a seamless and secure user experience.

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

Black-box testing

It is carried out to test functionality of the program. It is also called 'Behavioral' testing. The tester in this case, has a set of input values and respective desired results. On providing input, if the output matches with the desired results, the program is tested 'ok', and problematic otherwise. Black box testing is designed to validate functional requirements without regard to the internal workings of a program.

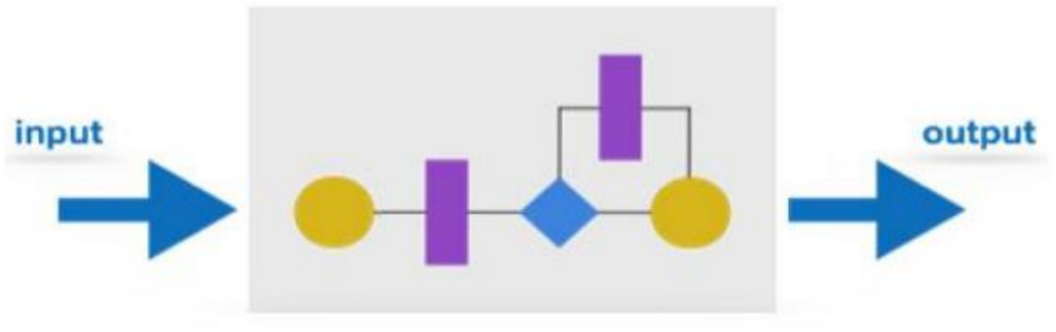


Black-box testing techniques:

- **Equivalence class –** The input is divided into similar classes. If one element of a class passes the test, it is assumed that all the class is passed.
- **Boundary values –** The input is divided into higher and lower end values. If these values pass the test, it is assumed that all values in between may pass too.
- **Cause-effect graphing –** In both previous methods, only one input value at a time is tested. Cause (input) – Effect (output) is a testing technique where combinations of input values are tested in a systematic way.
- **Pair-wise Testing –** The behavior of software depends on multiple parameters. In pair-wise testing, the multiple parameters are tested pair-wise for their different values.
- **State-based testing –** The system changes state on provision of input. These systems are tested based on their states and input.

<u>White-box testing</u>

It is conducted to test program and its implementation, in order to improve code efficiency or structure. It is also known as 'Structural' testing. White box testing focus on the program control structure. Test cases are derived to ensure that all statements in the program have been executed at least once during testing and that all logical conditions have been executed. In this testing method, the design and structure of the code are known to the tester.



Programmers of the code conduct this test on the code. The below are some White-box testing techniques:

Control-flow testing - The purpose of the control-flow testing to set up test cases which covers all statements and branch conditions. The branch conditions are tested for both being true and false, so that all statements can be covered.

Data-flow testing - This testing technique emphasis to cover all the data variables included in the program. It tests where the variables were declared and defined and where they were used or changed.

Testing Strategies - Testing separately is done just to make sure that there are no hidden bugs or issues left in the software. Software is tested on various levels –

Unit Testing- While coding, the programmer performs some tests on that unit of program to know if it is error free. Testing is performed under white-box testing approach. Unit testing helps developers decide that individual units of the program are working as per requirement and are error free.

Integration Testing- Even if the units of software are working fine individually, there is a need to find out if the units if integrated together would also work without errors. For example, argument passes and data updating etc.

System Testing- The software is compiled as product and then it is tested as a whole. This can be accomplished using one or more of the following tests:-

- **Functionality testing -** Tests all functionalities of the software against the requirement.

- **Performance testing -** This test proves how efficient the software is. It tests the effectiveness and average time taken by the software to do desired task. Performance testing is done by means of load testing and stress testing where the software is put under high user and data load under various environment conditions.
- **Security & Portability -** These tests are done when the software is meant to work on various platforms and accessed by number of persons.
- **Acceptance Testing-** When the software is ready to hand over to the customer it has to go through last phase of testing where it is tested for user-interaction and response. This is important because even if the software matches all user requirements and if user does not like the way it appears or works, it may be rejected.
- **Alpha testing -** The team of developer themselves perform alpha testing by using the system as if it is being used in work environment. They try to find out how user would react to some action in software and how the system should respond to inputs.
- **Beta testing -** After the software is tested internally, it is handed over to the users to use it under their production environment only for testing purpose. This is not as yet the delivered product. Developers expect that users at this stage will bring minute problems, which were skipped to attend.
- **Regression Testing-** Whenever a software product is updated with new code, feature or functionality, it is tested thoroughly to detect if there is any negative impact of the added code. This is known as regression testing.
-

Key Objectives:

1. Functionality Testing:

- Verify that basic chat functionalities work as intended, including sending and receiving messages, creating chat groups, and managing user profiles.
- Conduct tests to ensure the application supports multimedia elements such as images, files, and emojis.

2. Performance Testing:

- Evaluate the application's performance under varying network conditions, including high and low traffic scenarios.
- Measure response times for message delivery and assess the application's scalability to handle a growing number of users.

3. Security Testing:

- Identify and address potential security vulnerabilities, such as unauthorized access, data breaches, and encryption issues.
- Implement penetration testing to simulate potential attacks and ensure the application's resilience against security threats.

4. Compatibility Testing:

- Test the application on different devices and operating systems commonly found in a LAN environment to ensure compatibility.
- Verify that the application works seamlessly with popular web browsers and network configurations.

5. User Experience (UX) Testing:

- Gather feedback from potential end-users to assess the application's usability and user interface design.
- Make improvements based on user feedback to enhance the overall user experience.

6. System Security Measures

Securing a Local Area Network (LAN) live chat application is crucial to protect sensitive information, ensure privacy, and prevent unauthorized access. Here are several system security measures that can be implemented for a LAN live chat application:

1. Encryption:

- Implement end-to-end encryption for messages exchanged within the chat application. This ensures that data is secure during transmission and can only be decrypted by the intended recipients.

2. User Authentication:

- Enforce strong user authentication mechanisms, such as username/password combinations, two-factor authentication (2FA), or biometric authentication, to prevent unauthorized access to the application.

3. Access Control:

- Implement role-based access control (RBAC) to restrict users' permissions based on their roles. Regularly review and update access permissions to ensure that users have the minimum required privileges.

4. Secure Login Sessions:

Use secure login sessions and tokens to authenticate users and manage their sessions securely. Implement session timeout mechanisms to automatically log users out after a period of inactivity.

5. Secure Data Storage:

- Protect user data stored on the server by using encryption mechanisms. Employ secure hashing algorithms for storing passwords to prevent unauthorized access even in the event of a data breach.

7. Cost Estimation

1. The first step is to calculate the size of the project, which is measured in lines of code (LOC). Theproject has more than 290 lines of code, so we'll use that number as the project size.

2. Next, we need to estimate the effort required to complete the project. The effort is measured in person- months (PM). Since I worked part-time on the project for 3 months, we can estimate the total effort as 1.5PM (0.5 PM per month).

3. We also need to estimate the team size. Since I worked on the project alone, the team size is 1.

4. Now, we can use the COCOMO cost estimation model to calculate the total cost of the project. COCOMO has three different modes: Basic, Intermediate, and Advanced. For this project, we'll use the Intermediate mode since it's suitable for projects with a moderate level of complexity.

5. Using the Intermediate mode, we can calculate the total cost of the project as follows:

   Effort = 1.5 PM

Project Size = 290 LOC

Team Size = 1

Duration = 3 months

Using the COCOMO formula, we get:

Effort = 2.4 * (KSLOC)^1.05 * (Capers Factor)Duration = 2.5 *

(Effort)^0.38

Staff = Effort / Duration

Cost = Effort * Staff * Average Programmer Salary

6. Determine the average programmer salary, which can vary based on location and experience level for this example, we'll assume an average salary of ₹200 per hour.

7. Estimate the Capers Factor, which depends on the project's complexity. we'll use a Capers Factor of 1.15.

8. Plugging in the values and calculations, we can estimate the cost of the project:Effort = 2.4 * (2.0)^1.05 * 1.15 = 5.52 PM

Duration = 2.5 * (5.52)^0.38 = 5.8 months Staff

= 5.52 / 5.8 = 0.95

Cost = 100 * 0.1 * ₹10 * 100 = ₹10,000

So, the estimated cost of our is **Local Area Network Live Chat App ₹10,000**

Cost estimation for developing a **Local Area Network (LAN) live chat application** can vary based on several factors, including the complexity of features, the technology stack used, development time, and labor costs. Below is a breakdown of potential cost factors to consider:

8. Future scope

The future scope for a local area network (LAN) live chat application can be influenced by technological advancements, changing user preferences, and evolving business needs. Here are some potential areas of growth and development:

1. Security Enhancements:

- End-to-End Encryption: Implement robust security measures, such as end-to-end encryption, to ensure the privacy and security of user communications.
- Authentication and Authorization: Strengthen user authentication and authorization mechanisms to prevent unauthorized access and protect sensitive information.

2. Mobile Optimization:

- Cross-Platform Compatibility: Ensure that the application is optimized for various devices and operating systems to accommodate the increasing use of mobile devices.
- Mobile App Development: Develop dedicated mobile applications to enhance user experience and accessibility.

3. Collaboration and Productivity Features:

- File Sharing: Enable seamless file sharing within the chat application to enhance collaboration among users.
- Task Management: Integrate task management features to help teams coordinate and manage projects directly within the chat interface.

• Bibliography

- https://www.w3schools.com
- https://www.geeksforgeeks.org