CHAPTER TWO

2.1 **Definition of areas related.**

This section looks at a detailed de­finitions related to the project topic, aiming to make things cleare­r and help you better unde­rstand the project.

2.1.1 **WEBRTC**

WebRTC, which stands for Web Real-Time Communication, empowers peer-to-peer associations without the require for middle person servers.Information sharing over WebRTC is straightforward, supported by WebRTC APIs, and accessible at many stages thanks to its simplicity.

At first, servers are required for clients to set up associations. That being said, once the association is set up, clients can communicate straightforwardly without server mediation.

2.1.1.1H**ow it works:**

When a client sends data to the signaling server, the server transfers this data to the other client. After accepting this information, the client stores it locally. In this way, this client sends its claim information to the server, which transfers it to the other client. This trade guarantees both clients are mindful of each other, encouraging an association foundation.

Signaling servers, utilized for this information trade, are standard servers with the sole reason of encouraging client information trade. The method of exchanging information from one client to another in WebRTC is called signaling. This server can be any standard server able of encouraging information trade.

The primary step in a fruitful WebRTC association is the creation of offers. On the off chance that a client needs to put through with another, it makes an offer, regularly within the shape of a JavaScript protest containing session portrayal convention (SDP) information. This SDP incorporates media arrangements such as video and sound inclinations.

The client sends this offer to the signaling server, which advances it to the other client. Upon getting the offer, the other client stores it locally. It at that point produces an answer, containing its possess SDP information, and sends it back to the initial client by means of the signaling server.

While both clients presently know each other's media setups, they still need organize data to set up coordinate association. To address this, ICE candidates are created from Stagger and TURN servers. Clients give URLs of these servers to the WebRTC API to get ICE candidates.

As the client makes an offer, ICE candidates are recovered from the servers. These candidates are at that point sent to the other client by means of the signaling server. The same handle happens when creating answers.

Once both clients have each other's SDP information and ICE candidates, they can build up a coordinate peer-to-peer association. Information transmission through WebRTC APIs from that point happens straight forwardly between the clients, empowering efficient and secure real-time communication.

**2.1.2 Server side rendering**

Server-side rendering (SSR) could be a procedure that renders a web page on the server instead of within the browser. When a website's JavaScript is rendered on the website's server, a completely rendered page is sent to the client and the client's JavaScript bundle locks in and empowers the Single Page Application system to function.

server side rendering (SSR) is a method employed in web development where the server dynamically creates the HTML content of a web-page and sends it to the clients browser. This differs from client side rendering, where the browser uses JavaScript to generate the HTML content after receiving HTML from the server.

**2.1.2.1 here’s how it works**

In SSR, when a user asks for a web page, the server handles the request, collects the data from a database or external APIs, and then generates the HTML content incorporating this data. The entire HTML page is then transmitted to the client's browser for display without any processing. This approach can lead to quicker initial page loading times. Improved search engine optimization (SEO) since search engines can easily scan and index the HTML content.

SSR is frequently utilized in web applications developed with server side technologies like Node.js, Django, Ruby on Rails and PHP. It proves advantageous for websites with content or dynamically generated content where SEO and performance play vital roles. Nonetheless implementing SSR may add complexity to development and up keep efforts as it necessitates handling of server side and client side logic to ensure consistency and optimal performance, across platforms and devices.

**2.1.3** **Success of e-learning platforms**

Success of e-learning platforms The popularity of online learning platforms can be credited to reasons. One significant factor is their convenience enabling students to reach materials at any place and time. Another aspect is the range of content offered accommodating learning preferences and styles. also, the engaging features of online learning platforms such, as quizzes, videos and simulations boost involvement and memory retention. finally, the flexibility of learning platforms allows them to cater to an audience and adjust to different educational needs and settings.

**2.1.4 Scalable DBMS (MongoDB)**

MongoDB stands out as a liked No-SQL database recognized for its ability to scale effectively. Its distributed architecture is the key, to this scalability allowing it to manage data volumes and heavy traffic loads efficiently. MongoDB utilizes sharing to distribute data among servers facilitating scaling. This implies that as data and traffic increase you can expand the number of servers in the MongoDB cluster to accommodate the growing demand.

The adaptability of MongoDB schema also plays a role in its scalability. Unlike databases MongoDB does not mandate a predefined schema making it simple to introduce new fields or modify existing ones without any downtime. This adaptability proves beneficial in environments where requirements are subject to frequent changes.

Apart from scaling MongoDB also supports scaling by enabling you to enhance individual server resources (such as CPU and RAM) within the cluster to manage higher workloads. The combination of vertical scaling capabilities makes MongoDB a scalable database solution suitable for various applications ranging from small startups, to large enterprises.

**2.1.5 Containerization(docker)**

Docker, an example of containerization transforms the landscape of software development and deployment by packaging applications and their requirements, into self contained units known as containers. These containers operate reliably across settings spanning from development, to production guaranteeing behavior of applications regardless of their deployment location. Containerization offers application segregation, flexibility and optimized resource usage simplifying the development and deployment workflow while enhancing resource efficiency compared to machines.

**2.1.6 System architecture to avoid SPOF (single point of failure)**

It's really important to make sure that a our system doesn't rely on one part that could cause the whole thing to fail. When a single point of failure occurs it means that if one component breaks the entire system will stop working. To prevent this methods, like having backups and being able to handle faults are used to keep the system running even if something goes wrong with one or more parts. This is particularly crucial, in systems where any downtime could lead to losses or affect how users interact with it. Ultimately avoiding points of failure helps boost the reliability, availability and resilience of a system.

**2.1.7 Web sockets for bi directional Communication**

Web Sockets allow for two way communication, between a client like a web browser and a server through a connection., Unlike HTTP requests where the client typically initiates communication. Web Sockets enable both the client and server to exchange messages at any time. This back and forth communication is ideal for applications that need real time updates, such as chat platforms, online games or collaborative editing tools. By using a handshake process to establish connections, Web Sockets can efficiently receive data in both directions without the need for repeated setup. This efficiency makes Web Sockets more effective for real time communication compared to methods like polling or long polling. In summary Web Sockets offer a solution, for creating web applications that demand immediate responsiveness and minimal delays.

**2.2 REVIEW OF RELATED AREAS**

**2.2.1 RESEARCH IMPLEMENTATION OF WEBRTC SIGNALING VIA WEB-SOCKET-BASED FOR REAL-TIME MULTIMEDIA COMMUNICATION by (CUI JIAN, ZHUYING LIN 2021)**

This research paper focuses on the research and implementation of WebRTC signaling via Web Socket-based for real-time multimedia communications. The paper analyzes the core architecture and related technologies of WebRTC, including video input and output, multimedia transmission, peer-to-peer connection establishment, and signaling mechanism. It presents a signaling exchange mechanism via Web-socket and researches the WebRTC peer-to-peer connection based on Web-socket in detail. The paper also describes the design and implementation of a real-time multimedia communication system on the mobile internet. The research provides an academic and practical foundation for WebRTC signaling work.

**Methodology**

The document provides information on the methodology used for the research and implementation of a real-time multimedia communication system based on WebRTC. It mentions that the system supports peer-to-peer connections between browsers after signaling negotiation. The methodology involves SDP and ICE information negotiation to establish the peer-to-peer connection. Additionally, the document describes the use of the WebRTC API to obtain local audio and video input, which is then transmitted over the established peer-to-peer connection. The HTML tag is utilized for video output. The implementation is mobile internet-based, supporting multimedia real-time interaction between smart mobile devices.

**STRENGTH**

The document noted out several strength of using webrtc which are;

1. PEER TO PEER COMMUNICATION: webrtc makes direct p2p communication possible between browsers without the need of third party plugins.
2. Cost effective. They mentioned that webrtc minimized hardware and technology costs associated with developing real time multimedia interactive systems by leveraging browser based communication capabilities
3. Browser support; main browsers such as PC chrome, Firefox, and also android chrome already support webrtc, ensuring large compatibility and accessibility

Weaknesses

1. Security concerns: webrtc works in a decentralized manner, this causes the raise in their security concerns if not implemented properly. So ensuring secure data transmission and encryption is important
2. Resource intensive; real time communication applications using webrtc can be resource intensive, especially for devices with limited processing power or memory

**Success of e-learning platforms**

**2.2.2 THE IMPACT AND EFFECTIVENESS OF E-LEARNING ON TEACHING AND LEARNING (DR. RIAH F. ELCULLADA ENCANACIONI,2020)**

Review by Dr. Riah F. Elcullada EncanacionI, a Senior Lecturer at Oman Tourism College presented an Article, “The Impact and Effectiveness of E-Learning on Teaching and Learning,” she presented the study results of the effectiveness and the impact of E-Learning on the Undergraduate Program and the General Foundation Program in the Oman Tourism College.

**METHODOLOGY**

The research methodology used was a combination of two methods, a quantitative online survey of the teachers and students experience and success utilizing in E-Learning as the primary research method. The study’s effectiveness criteria were based on five parameters and included Content Quality, Assessment, Collaborative Environment, System Quality, and Technical Support. The result showed that the teachers and students see e-learning favourable and provided higher ratings on the importance of e-learning. The study also discovered that E-Learning improved instructional strategies and student learning preferences. The researchers came to the conclusion that one of the finest teaching and learning tactics is e-learning, and they suggested doing more study that would involve other higher education institutions in Oman in order to create effective e-learning strategies.

**Shortcomings**  
limited Scope: Because the study is exclusively focused on Oman Tourism College, it's possible that the conclusions won't apply to other organizations or situations. To ensure that the findings are more broadly applicable, future research might strive for a more varied sample.  
Absence of Comparison: It appears that the study does not include a comparison group to evaluate the relative efficacy of e-learning, such as traditional face-to-face training. A better understanding of the distinct contributions that e-learning makes to teaching and learning outcomes could be obtained by including a control group.

**Literature review**

**Sever side rendering**

**2.2.3 COMPARISON BETWEEN CLIENT SIDE RENDERING AND SERVER SIDE RENDERING IN THE WEB DEVELOPMENT(MUHARMAN LUBIS, ARIF RIDHO LUBIS,TIEN KUSUMASARI, 2020)**

this academic article comparing client-side and server-side rendering in web development. explores the differences and advantages of both approaches in the context of web development. The article delves into the technical aspects and performance implications of client-side and server-side rendering, providing insights for developers and researchers in the field.

**Methodology**

The methodology used in the article involves a comparative analysis of client-side and server-side rendering in web development. The authors examines the technical aspects, performance implications, and advantages of both approaches. The author likely conducted a thorough review of existing literature, analyzed case studies, and possibly performed experiments to gather data and insights for the comparison. The methodology may also include a discussion of best practices and recommendations for developers based on the findings. For specific details on the methodology used.

**STRENGTH AND WEAKNESSES**

The key differences between client-side and server-side rendering in web development lie in where the rendering process takes place and how it impacts performance. Client-side rendering involves rendering web content in the user's browser using JavaScript, while server-side rendering involves generating the HTML on the server before sending it to the client's browser. Client-side rendering can provide a more interactive user experience but may lead to slower initial page load times, especially for content-heavy websites. On the other hand, server-side rendering can result in faster initial page loads but may limit interactivity and responsiveness. The author also mentioned how developers need to consider factors such as SEO, performance, and user experience when choosing between client-side and server-side rendering approaches.

**Review on Scalable DBMS (MongoDB)**

**2.2.4 THE NEW ERA OF DATABASE MANAGEMENT SYSTEM USING MONGO DB ( MARMISH VERMA, 2024)**

An article by Marmish Verma titled ' stated, MongoDB plays a crucial role in modern database technology due to its innovative features and capabilities. The article discusses the attributes, benefits, and practical implementations of non-relational databases, with a focus on MongoDB. It highlights MongoDB's superiority over traditional relational databases, particularly in handling vast datasets. The article explains MongoDB's features, such as adaptability, scalability, automatic sharing, and replication, which have made it a leader in the field. It also compares MongoDB with MySQL, highlighting the differences between the two database systems. The article concludes by discussing the advantages of MongoDB, its applications, and its role in modern database technology

**Shortcomings**

The document mentions a drawback of MongoDB related to document embedding, which can lead to larger documents after creation, potentially affecting database performance. However, it does not delve into other specific disadvantages of MongoDB. Despite this, MongoDB is widely recognized for its numerous advantages and is preferred over traditional relational databases in many scenarios.

**REVIEW ON WEBSOCKETS FO BI-DIRECTIONAL COMMUNICTION**

**ENHANCING REAL TIME COMMUNICATION AND EFFICIENCY WITH WEBSOCKET(ALOK DUBEY, AUGUST 2023)**

Web Socket, as discussed by Mr. Alok Ashok Dubey in April 2023, is a technology that enhances real-time communication and efficiency in web applications. It provides a persistent connection between the client and server, reducing latency and overhead. It also offers low latency, making it ideal for applications requiring instant data updates and interactions.Web Socket also facilitates push-based communication, allowing servers to send data to clients without client requests, making it beneficial for live data streaming, real-time collaboration, and instant messaging. It optimizes server resource utilization by eliminating the need for frequent polling or keeping numerous connections open, leading to better scalability and performance. Furthermore, Web Socket's real-time capabilities enhance user experience by enabling instant updates, notifications, and interactions, making web applications more engaging and interactive. Overall, Web Socket's capabilities make it a valuable technology for improving web application efficiency.

**Shortcomings**

Security consideration.

The author mention that, Security flaws with Web-socket connections include cross-origin resource sharing (CORS) problems, cross-site scripting (XSS) attacks, and unapproved access. To reduce these dangers and guarantee the integrity and confidentiality of data sent across Web-socket connections, developers must put strong security mechanisms in place, such as encryption, authentication, and authorization.

Web-Sockets do not automatically recover when connections are terminated.This is something you need to do yourself, which is part of the reason there are so many Web-socket client-side libraries available.

Implementation Complexity: Web-socket technology adds a layer of complexity to the creation of online applications, especially when it comes to resolving connection failures, maintaining persistent connections, and putting real-time communication protocols into practice.