**ABSTRACT**

Callify Meet is built using a modern tech stack that enhances its functionality and user experience. The primary framework utilized is **Next.js**, a powerful React-based framework that enables server-side rendering and static site generation, ensuring fast load times and improved SEO. This framework allows developers to create dynamic web applications with ease, making it ideal for a video conferencing platform.

To style the application, **Tailwind CSS** is employed, providing a utility-first approach to CSS. This allows for rapid design and customization, enabling developers to create responsive and visually appealing interfaces without the need for extensive custom CSS. Tailwind's flexibility ensures that the UI is not only functional but also aesthetically pleasing, enhancing user engagement.

For user authentication, **Clerk** is integrated into Callify Meet. Clerk simplifies the authentication process by offering features such as social sign-in, multi-factor authentication, and magic links. This ensures a secure and user-friendly experience, allowing users to sign up and log in effortlessly while maintaining high security standards.

Lastly, **GetStream** is utilized for real-time communication features, including video conferencing and chat functionalities. GetStream's robust API allows for seamless integration of real-time messaging and video capabilities, ensuring that users can communicate effectively during meetings. With its high uptime and reliability, GetStream enhances the overall performance of Callify Meet, making it a dependable choice for virtual collaboration.

Together, these technologies create a powerful and efficient platform that meets the demands of modern remote communication, providing users with a seamless and engaging experience.

CONTENTS

**Sl. No. Chapter Name Page No.**

**i. Abstract i**

**ii. Contents ii**

**iii. List of figures iv**

**iv. List of tables v**

1. **Chapter 1 - Introduction 1**
   1. Definition 1
   2. Technology Stack 2
2. **Chapter 2 – Literature Survey 3**
   1. Definition 3
3. **Chapter 3 – Requirement Specifications 6**
   1. Functional Requirements 6
   2. Non- Functional Requirements 9
   3. Software Requirements 11
   4. Hardware Requirements 13
4. **Chapter 4 – System Design 16**
   1. Definition 16
   2. Overview of Callify 16
   3. Key Features 17
   4. Technology Stack 18
   5. User Interface Design 18
   6. Security Measures 19
   7. Development and Deployment 19
   8. Future Enhancements 19
5. **Chapter 5- Implementation 20**
   1. Definition 20
   2. Project Setup 20
   3. Environment Configuration 20
   4. User Authentication with Clerk 21
   5. Integrating Getstream 22
   6. Building The User Interface 22
   7. Implementing Meeting Functionality 23
   8. Styling with Tailwind CSS 23
   9. Deployment 24
6. **Chapter 6- Testing 25**
   1. Definition 25
   2. Testing Goals and User Needs: 25
   3. Testing Environment and Equipment: 25
   4. Testing Process: 26
   5. Assumptions and Limitations: 26
   6. Key Findings 26
7. **Chapter 7 – Results 27**
   1. Definition 27
   2. Key Features and Benefits 27
   3. Applications Across Industries 27
   4. Challenges and Considerations 28
   5. Future Trends in Video Conferencing 28
8. **Chapter 8- Conclusion & Future scope 29**
   1. Conclusion 29
   2. Future Scope 29

**References 31**

**LIST OF FIGURES**

**Figure. No Figure Name Page. No**

**Figure 1.2.1**  Next Js 02

**Figure 1.2.2** Tailwind CSS 02

**Figure 1.2.3.**  Clerk 02

**Figure 1.2.4.**  Get Stream 02

**Figure 4.2.1**  Dataflow Diagram 16

**Figure 5.3.1**  ER Diagram 21

**Figure 5.4.1** Clerk Authentication 21

**Figure 5.5.1** Get Stream Chat 22

**Figure 5.6.1.** Create Link Model 23

**Figure 5.7.1.** Create and Joining Model 23

**Figure 5.7.2.** Joining Call 23

**LIST OF TABLES**

**Table.No Table Name Page. No**

**Table 2.1.1 : Literature Review**  03-05