**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 iii**

**iv. List of tables iv**

1. **Chapter 1 - Introduction 1**
   1. Heading 2
   2. Heading 3
   3. Heading 4
   4. Heading 5
      1. Sub Heading 5
      2. Sub Heading 6
      3. Sub Heading 7
   5. Heading 8
   6. Heading 9
2. **Chapter 2 – Literature Survey 10**
   1. Heading 12
   2. Heading 13
   3. Heading 14
3. **Chapter 3 – Requirement Specifications 15**
   1. Heading 16
   2. Heading 17
   3. Heading 18
   4. Heading 19
4. **Chapter 4 – System Design 21**
5. **Chapter 5- Implementation 23**
6. **Chapter 6- Testing 24**
7. **Chapter 7 – Results 25**
8. **Chapter 8- Conclusion & Future scope 27**

**References 28**

**LIST OF FIGURES**

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

**Figure 1.1.** Image Enhancement 02

**Figure 1.2.** Image Restoration 03

**Figure 1.3.** Image Compression 04

**Figure 1.4.** Binary Images 05

**Figure 1.5.** Gray Scale Images 05

**Figure 1.6.** Color Image 06

**Figure 1.7.** Digital Image Processing 11

**Figure 1.8.** Video Retrieval 13

**Figure 4.1.** Partial Occlusions 27

**Figure 5.1.** Examples of Positive and Negative Samples 33

**Figure 5.2.** Flowchart of PL-SVM Training 34

**Figure 5.3.** Flowchart of Human Detection 34

**Figure 5.4.** Feature Representation 36

**Figure 5.5.** Cascade Detector 36

**Figure 5.6.** Cascade Classification 37

**Figure 5.7.** Sliding Window 37

**LIST OF TABLES**

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

**Table 1.1 : Description of Table 1**  41

**Table 1.2 : Description of Table 2**  41

**Table 2.3 :** 41

**Table 3.4:** 42

**Table 5.1:**  43