PROJECT SYNOPSIS REPORT

ON

Code Live Test (CLT)

(Video Calling

Interview Platform

With Integrated Coding Platform)

TO

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FOR

Integrated Project

Submitted To:

Dr.Lekha Rani

Submitted By :

Arsh Verma (2210991363)
Arshpreet Singh (2210991370)

Saksham Garg (2210992221)

Index

Objective & Key Learnings:

The primary objective of this project is to create an intuitive and secure video calling interview platform that enables seamless communication for online interviews. The platform will allow users to conduct and attend interviews with features like video calls, integrated coding platform while ensuring a smooth and dynamic user experience.

Key learnings will include:

- **Building Real-Time Communication:** Integrating video calls and coding platform into the platform using **Stream** API.
- **Authentication & Authorization:** Implementing secure login and user management using **Clerk** for authentication and authorization.
- **Optimizing User Experience:** Using **Next.js** and **TypeScript** for building a performant and scalable application that caters to both client and server-side components.
- **Dynamic Routing:** Handling dynamic routing for interviews, including unique interview sessions, user authentication, and related tasks.
- **State Management:** Implementing **Server Actions** for smooth interactions between client and serverside logic.

Options Available to Execute the Project:

Several technologies can be leveraged to execute this project. However, the selected tech stack provides the most robust and scalable options:

1. **Next.js & TypeScript:** For building a modern, highly performant web app, including static and dynamic routes, server-side rendering, and API routes.
2. **Stream API:** For real-time video calling and screen sharing functionality.
3. **Convex:** A backend service that allows for managing application logic and data persistence.
4. **Clerk:** For handling secure user authentication and authorization with easy integrations.
5. **Tailwind CSS & Shadcn:** For styling the application with a highly customizable, responsive design.

6. **Server Actions & Layouts:** To manage server-side logic and perform actions dynamically without overwhelming the client.

Advantages/Disadvantages:

Advantages:

- **Real-time Features:** Stream API provides low-latency video calling enhancing the interview experience.
- **Security:** Clerk ensures secure authentication and authorization, providing peace of mind for both employers and candidates.
- **Scalable Architecture:** The use of **Next.js** allows for both server-side and static rendering, ensuring fast page loads and smooth transitions between components.
- **Tailored UI/UX: Tailwind CSS** and **Shadcn** ensure that the platform has an attractive, responsive design.
- **Seamless Integration:** Combining dynamic routes, server actions, and API calls provides an efficient architecture for handling the complex flows involved in an interview process.

Disadvantages:

- **Complexity:** Building a robust real-time video platform can be complex and may require more development time, especially integrating different APIs and handling edge cases.
- **Performance:** Managing both video calls and screen sharing in real-time can strain resources, particularly for larger-scale implementations.
- **Learning Curve:** Technologies like **Convex** and **Clerk** may have a learning curve for developers unfamiliar with them.
- **Cost:** Real-time video calling and server management via external APIs (like **Stream** and **Convex**) could incur additional operational costs.

References:

- Stream API Documentation
- Next.js Documentation
- Tailwind CSS Documentation
- Shadcn UI Documentation
- Clerk Documentation
- Convex Documentation