Yash Tekwani

FULL-STACK DEVELOPment internship REPORT – youtube clone

## Introduction

This project involved enhancing a you-tube inspired web application with three primary features: a points allocation system for watching videos, a custom video player with gesture-based controls, and a VoIP feature for video calls and screen sharing. The objective was to create an engaging and interactive user experience while maintaining functionality and ease of use.

## Background

The web application is designed to provide users with a rich media experience. It provides the users with an option to create a channel, upload the videos and view videos similar to the ones in you-tube. The main tasks involved in the project included enhancing the app with a points system, a custom video player with gesture based controls, and VoIP capabilities which aims to increase user engagement and provide additional functionalities that cater to modern digital consumption habits.

## Learning Objectives

1. Understand the implementation of a points system based on user activity.
2. Integrate a custom video player with advanced gesture-based controls.
3. Implement a VoIP feature with screen sharing and recording capabilities.
4. Host the application on Netlify or Vercel.

## Activities and Tasks

**1.** Points System

* **Task:** Implement a system to allocate points for watching videos.
* **Activity:** Modify the user profile to include a points section that updates based on the number of videos watched. This involved adding a field to the user model named viewedVideos which contained a list of ids of the videos the user had already watched. Once the user viewed a video its id was checked against that list and if the user had not already watched it, it was added to the list. The points were calculated by multiplying the length of list by 5.

**2.** Custom Video Player

* **Task:** Develop a custom video player with gesture-based controls.
* **Activity:** The following controls were implemented in the video player:
  1. **Double Tap on Right Side:** Playback moves 10 seconds forward
  2. **Double Tap on Left Side:** Playback moves 10 seconds backwards
  3. **Single Tap in Middle:** Video Paused
  4. **Three Taps in Middle:** Move to next video
  5. **Three Taps on Right Side:** Close the website
  6. **Three Taps on Left Side:** Show the comment section
  7. **Single Tap on Top Right Corner:** Display current location and temperature of user
  8. **Hold Right Side:** Video Speed two times faster
  9. **Hold Left Side:** Video Speed two times slower

**3.** VoIP Feature

* **Task:** Add video calling capabilities with screen sharing and recording.
* **Activity:** Integrate a VoIP library, enable screen sharing, and implement recording functionalities. Restrict the video call feature to specific hours. This involved implementing sockets to connect with the peers using their specific ids. It also involved creating an interactive user interface to start call, end call, start recording and stop recording.

## Skills and Competencies

* Proficiency in React for building interactive user interfaces.
* Experience with video player libraries and custom gesture controls.
* Knowledge of WebRTC or similar technologies for implementing VoIP features.
* Understanding of hosting web applications on platforms like Netlify or Vercel.

## Feedback and Evidence

Feedback was collected through user testing sessions, focusing on the usability and responsiveness of the new features. Users appreciated the intuitive gesture controls and the seamless integration of the video call feature.

## Challenges and Solutions

### Challenge 1: Implementing Gesture Controls

### Challenge: Implementing gesture control was a bit challenging because for different number and location of taps different functionalities were to be displayed. Moreover, the video frame had a default behavior of pausing video on single tap and entering/exiting full screen mode on double tap. This behavior was not possible to escape and therefore the rest of the taps were adjusted according to it.

* **Solution:** Implemented gesture controls by adding reference to the video element so that it detects tap on video frame object. Moreover, separate functions were implemented to handle single, double and triple tap with conditional statements to check the location of tap on video frame in order to implement the required functionality.

### Challenge 2: VoIP Feature

### Challenge: Implementing VoIP feature was a bit challenging because it involved the use of WebRTC. Moreover, initially the recording being downloaded was empty.

* **Solution:** In order to provide a robust solution, I used React Media Recorder for recording the entire meeting. Both video and audio recorder had to be enabled to ensure that meeting recording covered the entire user screen instead of just camera streams of both users in the call.

### Challenge 3: Implementing Video Stream on Vercel

* **Challenge:** The user’s own stream was not visible when the project was first deployed on Vercel although the functionality was working correctly on local host.
* **Solution:** UseEffect hooks were used to ensure that stream renders correctly on loading.

### Challenge 4: Hosting and Deployment

* **Challenge:** The backend deployment was first tried on Vercel which was unsuccessful as Vercel does not support static file serving and therefore the video upload feature was not working correctly.
* **Solution:** As a solution the backend was deployed on Render and frontend was deployed on vercel.

## Outcomes and Impact

The project successfully enhanced the web application by introducing a points system, an advanced custom video player, and a VoIP feature. These improvements resulted in increased user engagement and satisfaction, providing a more interactive and rewarding experience.

## Conclusion

The project achieved its objectives of enhancing user engagement and functionality. The implementation of the points system, custom video player, and VoIP feature has significantly improved the user experience. The deployment on Vercel ensured accessibility and reliability. This project demonstrates the potential for further enhancements and the continuous evolution of user-centric web applications.