**NullClass Internship Report**

**Introduction**

This report summarizes my work and learning experience during my internship at NullClass, where I contributed to developing a fully functional YouTube clone with advanced, real-world features. The internship focused on practical full-stack development, simulating real production environments.

**Background**

Before joining the internship, I had a foundational understanding of web development technologies such as HTML, CSS, JavaScript, React.js, Node.js, and MongoDB. This internship provided the opportunity to deepen my skills by building a complex video platform from scratch, incorporating video streaming, payments, VoIP, and geolocation logic.

**Learning Objectives**

* Strengthen full-stack development skills using the MERN stack (MongoDB, Express, React, Node.js)
* Learn video processing techniques for adaptive streaming
* Implement production-ready features such as premium upgrades, payment gateways, and download restrictions
* Gain experience with third-party APIs (e.g., Firebase, Razorpay, ipinfo.io, Google Cloud Storage)
* Understand how to deploy and maintain scalable web applications

**Activities and Tasks**

Over the course of the internship, I completed the following key tasks:

* **Multi-Quality Video Player**: Implemented FFmpeg-powered server-side conversion to HLS format with 360p, 480p, 720p, and 1080p streams; used Video.js for seamless quality switching
* **GCS Integration**: Uploaded processed videos and thumbnails to Google Cloud Storage with public URLs stored in MongoDB
* **Video Downloads & Premium Logic**:
  + Limited free users to one download per day (tracked via backend)
  + Issued signed URLs for secure downloads from GCS
  + Integrated Razorpay test payment gateway to unlock unlimited downloads for premium users
* **Comment System Enhancements**:
  + Added translation support using Google Translate API
  + Displayed user city name based on IP geolocation
  + Auto-deleted comments that received two or more dislikes
* **VoIP with Recording & Screen Share**: Implemented WebRTC-based video calling with screen sharing and local recording features
* **OTP Verification**:
  + Used Nodemailer for email-based OTP
  + Integrated Firebase for mobile OTP verification
  + Switched logic dynamically based on user location (South India = email OTP; other states = mobile OTP)
* **Theme & UI Adaptation**:
  + Applied dark or light themes based on user’s location and time
  + Used ipinfo.io for geolocation to handle location-based logic reliably

**Skills and Competencies Developed**

* Adaptive streaming & media conversion (FFmpeg, HLS)
* React.js & Redux for scalable UI development
* RESTful API development with Express.js
* MongoDB schema design and data management
* Secure media upload and access (Google Cloud Storage, signed URLs)
* Payment integration using Razorpay
* Real-time communication (WebRTC, socket.io)
* Firebase Authentication for mobile OTP
* UX logic based on user location and time
* Error handling, testing, and production debugging

**Feedback and Evidence**

* Code and backend logic were pushed regularly to GitHub
* Deployment completed via Netlify (frontend) and Render (backend)
* Regular progress updates submitted via NullClass portal
* Working project links:
  + GitHub: <https://github.com/AqdasAhmed/you_tube_clone>
  + Live App: [https://play-tube-clone.netlify.app](https://play-tube-clone.netlify.app/)

**Challenges and Solutions**

| **Challenge** | **Solution** |
| --- | --- |
| Video conversion errors on deployment | Ensured cross-platform FFmpeg path handling and used async/await for stability |
| GCS file overwrites and invalid URLs | Used safe naming conventions and normalized paths before upload |
| Geolocation API rate limits | Switched from ipapi.co to ipinfo.io for more reliable results |
| OTP routing issues | Implemented dynamic logic for email/mobile OTP based on region |
| Razorpay test flow confusion | Created a custom PremiumModal with clear test instructions |
| Download count tracking | Used MongoDB’s Download model to track daily usage and restrict accordingly |

**Outcomes and Impact**

This internship helped me grow significantly as a developer. I learned to work with a full-stack codebase, build advanced real-world features, and debug deployment issues independently. It strengthened my portfolio and confidence in applying for full-stack developer roles, especially those requiring media handling, payment systems, or third-party integrations.

**Conclusion**

The NullClass internship was an impactful and enriching experience. The YouTube clone project simulated a real-world product environment with complex requirements, enabling me to apply theoretical knowledge into meaningful, functional features. I now feel more prepared to take on production-level responsibilities in any full-stack development role.