CHAPTER V: CONCLUSION AND RECOMMENDATION

5.1. Conclusion

In conclusion, the content management system (Gyanshristi) solves the problems of traditional content creation, sharing, and teamwork by providing an easy-to-use, efficient, and accessible digital platform. This system makes managing content simpler, reduces the workload for educators, and ensures students can quickly access relevant learning materials. Although there are challenges like relying on technology and internet access, Gyanshristi overcomes these with features such as offline access, voice-based input, and tools for better collaboration.

With advanced features like speech-to-text and text-to-speech, Gyanshristi makes content more accessible, especially for people with different learning needs. Its tools for organizing content, managing users, and supporting different roles make it useful for many educational and professional needs. By focusing on inclusive design and user engagement, Gyanshristi helps foster collaboration and self-paced learning. Gyanshristi is set to improve how content is managed by promoting inclusivity, interaction, and accessibility, creating a better learning experience for people from all backgrounds.

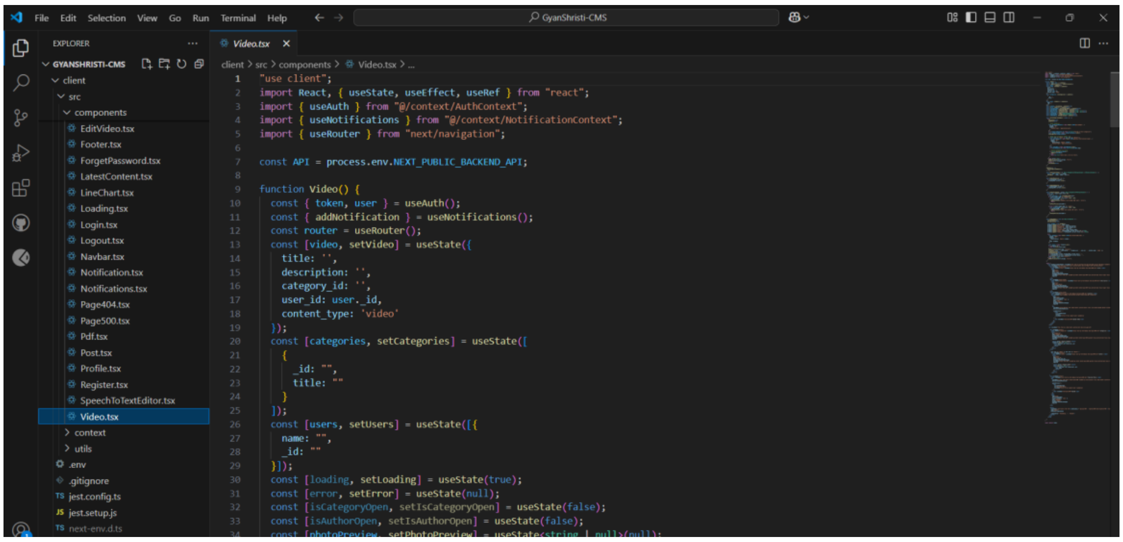
5.2. Recommendation

Once system is fully developed, it will offer powerful tools to manage content creation, organization, and collaboration while being easy to access for all users. We highly recommend that educators and institutions use system to simplify content management and make learning more engaging and inclusive. Teachers can save time by using features like voice-based content creation and speech-to-text tools, allowing them to focus on delivering quality educational materials. Administrators should regularly update user roles and permissions to keep the platform secure and well-organized. Authors and contributors are encouraged to use the platform’s collaboration tools to interact with users, boost engagement, and gather valuable feedback. To make the most of the platform, users should explore features like offline access and accessibility tools, such as the text-to-speech converter. Institutions should also provide training for teachers and students to help them understand and effectively use the system. By using Gyanshristi’s innovative features, educators, students, and contributors can create a more interactive and accessible learning experience, supporting the system’s goal of improving content management in modern education.

5.3. References

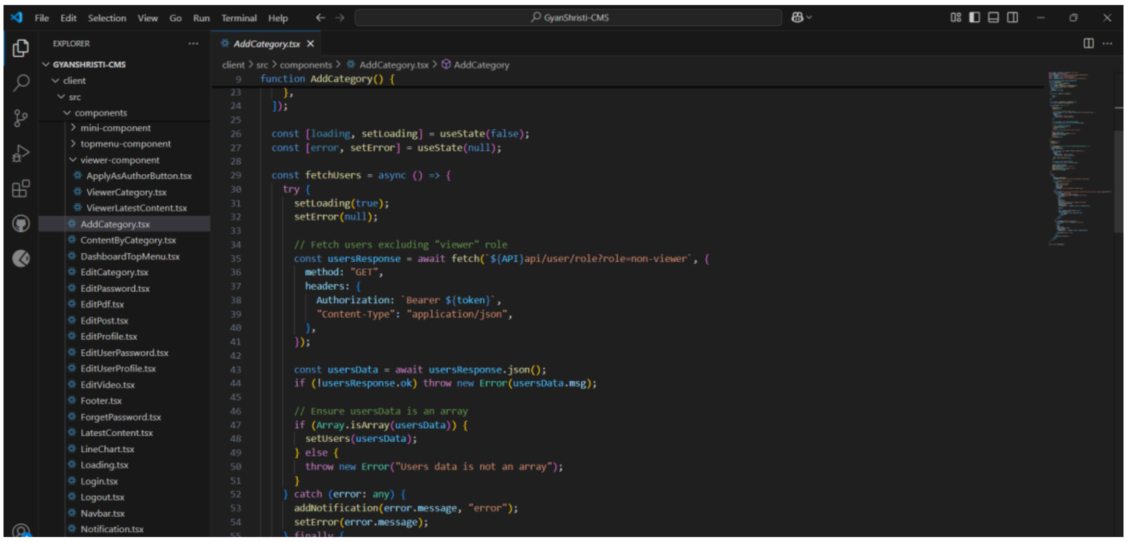
[1] Patel, S. K., Rathod, V. R., & Parikh, S. (2011, December 9). *A statistical comparison of Open Source CMS - IEEE Xplore*. Joomla, Drupal, and WordPress - a statistical comparison of open source CMS. https://ieeexplore.ieee.org/abstract/document/6169111
[2] Patel, S. K., Rathod, V. R., & Parikh, S. (2011, December 9). *A statistical comparison of Open Source CMS - IEEE Xplore*. Joomla, Drupal, and WordPress - a statistical comparison of open source CMS. https://ieeexplore.ieee.org/abstract/document/6169111
[3] Mohammed Ali Mohammed, Dr. Karim Q. Hussein, & Dr. Mustafa Dhiaa Al-Hassani. (2021, August 7). Real-Time Mobile Cloud Audio Reading System for Blind Persons. https://www.webology.org/data-cms/articles/20220122114149amWEB19024.pdf
[4] ComeConnect. (2023, November 8). *Reasons why scholar’s, researchers and students need listening.io*. Medium. https://medium.com/@enahjayjohn/reasons-why-scholars-researchers-and-students-need-listening-io-2c4ccc28ee3c
[5] ForbesIndia. (n.d.). *Paving the way for growth: GeeksforGeeks is making learning accessible and affordable for programmers*. Forbes India. https://www.forbesindia.com/article/brand-connect/paving-the-way-for-growth-geeksforgeeks-is-making-learning-accessible-and-affordable-for-programmers/75199/1
[6] Peetz, C. (2023, August 11). *Creating inclusive classrooms for blind students can benefit everyone. here’s how*. Education Week. https://www.edweek.org/teaching-learning/creating-inclusive-classrooms-for-blind-students-can-benefit-everyone-heres-how/2023/08
[7] *What is Data Flow Diagram?* (n.d.). https://www.visualparadigm.com/guide/data-flow-diagram/what-is-data-flow-diagram/;WWWSESSIONID=1715BFE147C14CA986ECC278C44815E7.www1

5.4. Appendix



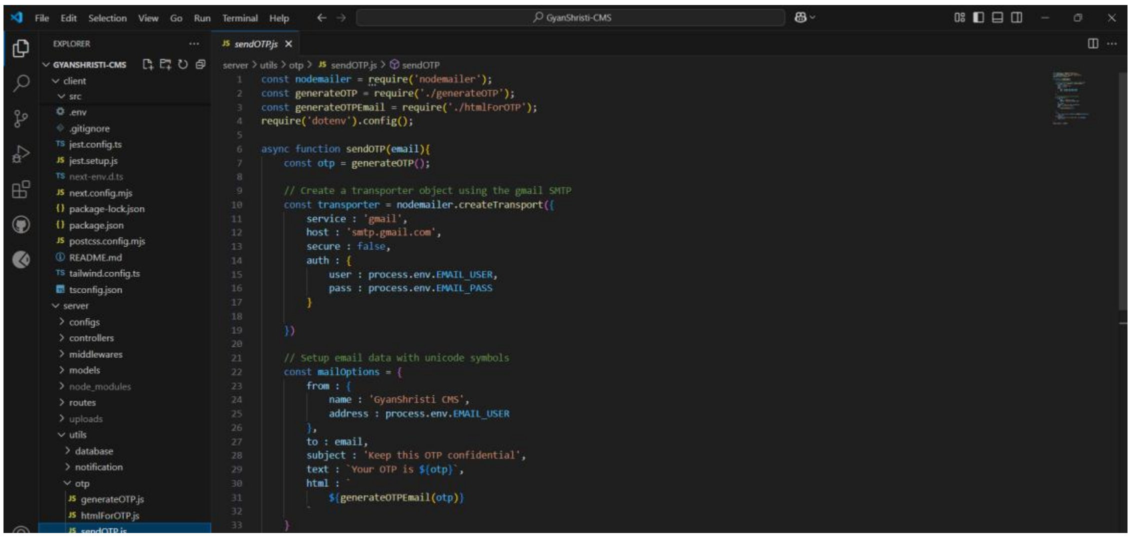
```
"use client";
import React, { useState, useEffect, useRef } from "react";
import { useAuth } from "@/context/AuthContext";
import { useNotifications } from "@/context/NotificationContext";
import { useRouter } from "next/navigation";

const API = process.env.NEXT_PUBLIC_BACKEND_API;

function Video() {
  const { token, user } = useAuth();
  const { addNotification } = useNotifications();
  const router = useRouter();
  const [video, setVideo] = useState({
    title: '',
    description: '',
    category_id: '',
    user_id: user._id,
    content_type: 'video'
  });
  const [categories, setCategories] = useState([
    {
      _id: "",
      title: ""
    }
  ]);
  const [users, setUsers] = useState([{
    name: "",
    _id: ""
  }]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);
  const [isCategoryOpen, setIsCategoryOpen] = useState(false);
  const [isAuthorOpen, setIsAuthorOpen] = useState(false);
  const [photoPreview, setPhotoPreview] = useState<string | null>(null);
```

Figure 1: Code for Uploading Video



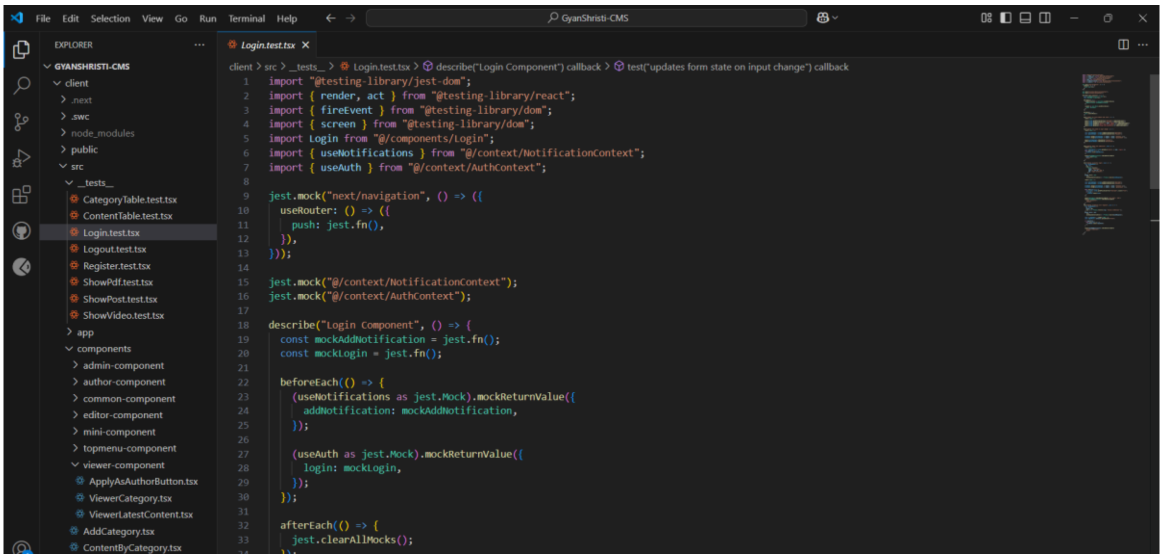
```
function AddCategory() {
    },
  ]);

  const [loading, setLoading] = useState(false);
  const [error, setError] = useState(null);

  const fetchUsers = async () => {
    try {
      setLoading(true);
      setError(null);

      // Fetch users excluding "viewer" role
      const usersResponse = await fetch(`${API}api/user/role?role=non-viewer`, {
        method: "GET",
        headers: {
          Authorization: `Bearer ${token}`,
          "Content-Type": "application/json",
        },
      });

      const usersData = await usersResponse.json();
      if (!usersResponse.ok) throw new Error(usersData.msg);

      // Ensure usersData is an array
      if (Array.isArray(usersData)) {
        setUsers(usersData);
      } else {
        throw new Error("Users data is not an array");
      }
    } catch (error: any) {
      addNotification(error.message, "error");
      setError(error.message);
    } finally {
```

Figure 2: Code for Adding Category



```javascript
const nodemailer = require('nodemailer');
const generateOTP = require('./generateOTP');
const generateOTPEmail = require('./htmlForOTP');
require('dotenv').config();

async function sendOTP(email){
    const otp = generateOTP();

    // Create a transporter object using the gmail SMTP
    const transporter = nodemailer.createTransport({
        service : 'gmail',
        host : 'smtp.gmail.com',
        secure : false,
        auth : {
            user : process.env.EMAIL_USER,
            pass : process.env.EMAIL_PASS
        }

    })

    // Setup email data with unicode symbols
    const mailOptions = {
        from : {
            name : 'GyanShristi CMS',
            address : process.env.EMAIL_USER
        },
        to : email,
        subject : 'Keep this OTP confidential',
        text : `Your OTP is ${otp}`,
        html : `
            ${generateOTPEmail(otp)}
        `

    }
```

Fig 3: Code for Sending OTP



```typescript
import "@testing-library/jest-dom";
import { render, act } from "@testing-library/react";
import { fireEvent } from "@testing-library/dom";
import { screen } from "@testing-library/dom";
import Login from "@/components/Login";
import { useNotifications } from "@/context/NotificationContext";
import { useAuth } from "@/context/AuthContext";

jest.mock("next/navigation", () => ({
  useRouter: () => ({
    push: jest.fn(),
  }),
}));

jest.mock("@/context/NotificationContext");
jest.mock("@/context/AuthContext");

describe("Login Component", () => {
  const mockAddNotification = jest.fn();
  const mockLogin = jest.fn();

  beforeEach(() => {
    (useNotifications as jest.Mock).mockReturnValue({
      addNotification: mockAddNotification,
    });

    (useAuth as jest.Mock).mockReturnValue({
      login: mockLogin,
    });
  });

  afterEach(() => {
    jest.clearAllMocks();
```

Fig 4: Test Code for Login Component