Varuvan Vadivelan Institute of Technology Dharmapuri.

Naan Mudhalvan:

IBM

TECHNOLOGY:

CLOUD APPLICATION DEVELOPMENT

PROJECT:

Media Streaming With IBM Cloud Video Streaming

Let’s Focus on the development aspects of your video streaming platform project.

Development :

1. User Management and Authentication:

- ❖ Develop a user registration system with fields for usernames, email addresses, and passwords.
- ❖ Implement secure password hashing and encryption.
- ❖ Create user profiles with customizable avatars and personal details.
- ❖ Design an authentication system with login, logout, and password reset functionality.
- ❖ Implement role-based authorization controls to manage user access.

2. Video Upload and Storage:

- ❖ Design and develop a video upload feature with a user-friendly interface.
- ❖ Implement validation for video format, size, and user permissions.
- ❖ Set up server-side storage for the uploaded videos. Consider using cloud storage services for scalability and redundancy.
- ❖ Create a database schema for video metadata, including title, description, and user references.

3. IBM Cloud Video Streaming Integration:

- ❖ Sign up for IBM Cloud Video Streaming services and obtain API credentials.
- ❖ Develop server-side code to interact with IBM Cloud Video Streaming APIs.
- ❖ Integrate video streaming functionality into your platform’s video player.
- ❖ Configure video settings, such as bitrate, resolution, and adaptive streaming options.
- ❖ Implement error handling and reporting for any issues related to the IBM Cloud Video Streaming service.

4. Video Playback System:

- ❖ Develop a video player component with features like play, pause, seek, volume control, and quality selection.
- ❖ Implement adaptive streaming to deliver the best video quality based on users’ network conditions.
- ❖ Consider support for closed captions and subtitles.
- ❖ Ensure a smooth and uninterrupted playback experience.

5. User Interface Development:

- ❖ Design and create a responsive and user-friendly web interface.

- ❖ Implement pages for video discovery, search, and video categorization.
- ❖ Develop user profiles with viewing history, uploaded videos, and user interactions (likes, comments, etc.).
- ❖ Ensure compatibility with various devices and browsers.

6. Monetization Features (Optional):

- ❖ If applicable, implement monetization features such as subscription plans, pay-per-view options, or advertising integration.
- ❖ Develop payment gateways and user subscription management systems.

7. Content Moderation and Security:

- ❖ Implement content moderation algorithms or integrate third-party moderation services to ensure content adherence to platform guidelines.
- ❖ Enforce security measures, including HTTPS, data encryption, and user data protection.

8. Quality Assurance and Testing:

- ❖ Conduct extensive testing, including functional testing, performance testing, and security testing.

- ❖ Test video playback under various network conditions and devices.
- ❖ Collaborate with QA testers to identify and resolve issues.

9. Soft Launch and User Feedback:

- ❖ Roll out the platform to a limited audience for a soft launch.
- ❖ Gather user feedback on the user experience, functionality, and performance.
- ❖ Address any issues and make improvements based on feedback.

10. Public Launch and Ongoing Maintenance:

- ❖ After successful testing and improvements, launch the platform to the public.
- ❖ Establish an ongoing maintenance plan to regularly update the platform and address any issues that arise.

These development tasks provide a detailed outline of the work involved in creating your video streaming platform with video upload and IBM Cloud Video Streaming integration. The development phase is a critical part of the project, and it's essential to follow best practices, conduct thorough testing, and maintain a responsive and secure platform for users.

Implementation of video streaming

Program:

```
# Import necessary libraries and frameworks

From flask import Flask, request, render_template, redirect, url_for

Import ibm_boto3

From ibm_botocore.client import Config


# Initialize Flask app

App = Flask(__name)


# Configure IBM Cloud Video Streaming

Api_key = 'YOUR_API_KEY'

Service_instance_id = 'YOUR_SERVICE_INSTANCE_ID'
```

```
Auth_endpoint =
'https://iam.cloud.ibm.com/identity/token'

Service_endpoint = 'https://api.video.cloud.ibm.com'


# Initialize the IBM Cloud Video Streaming client

Cos = ibm_boto3.resource("s3",

   Ibm_api_key_id=api_key,

   Ibm_service_instance_id=service_instance_id,

   Config=Config(signature_version="oauth"),

   Endpoint_url=service_endpoint

)


# Create a route for video upload

@app.route('/upload', methods=['GET', 'POST'])

Def upload_video():

   If request.method == 'POST':

      # Get the uploaded video file

      Video_file = request.files['video']
```

```
        # Validate and save the video to the cloud storage

        If video_file:

            Object_name = video_file.filename

            Cos.Object('bucket_name',
object_name).upload_fileobj(video_file)

            # Save video metadata and user information in the
database



            # Redirect to a success page

            Return redirect(url_for('success'))



    # Render the video upload form

    Return render_template('upload.html')



# Create a route for streaming videos

@app.route('/stream/<video_id>')

Def stream_video(video_id):

    # Retrieve video metadata and access permissions from
the database
```

```
    # Check if the user has permission to access the video


    # Generate a video playback URL from IBM Cloud Video Streaming


    # Render a video player page with the playback URL

    Return render_template('player.html', video_url=playback_url)


If __name__ == '__main__':

    App.run()
```

Output :

Video Upload Page:

- When you access the /upload route, you'll see an HTML form that allows you to select and upload a video file.

Upload Successful:

- After successfully uploading a video, you would be redirected to a success page.

Video Streaming Page:

- When you access a specific video’s URL (e.g., /stream/video123), you would see an HTML page with a video player embedded.
- The video player would use the playback URL provided by IBM Cloud Video Streaming services to stream the video.