MEDIA STREAMING WITH IBM CLOUD VIDEO STREAMING

PHASE-4: DEVELOPMENT PART – 2

Continue building the platform by integrating video streaming services and enabling on-demand playback. Implement the functionality for users to upload their movies and videos to the platform. Integrate IBM Cloud Video Streaming services to enable smooth and high-quality video playback.

To enable on-demand playback and allow users to upload their movies and videos, along with integrating IBM Cloud Video Streaming services for smooth playback, you'd typically involve both frontend and backend components. Here's a simplified guide:

Frontend Integration for Uploading Videos:

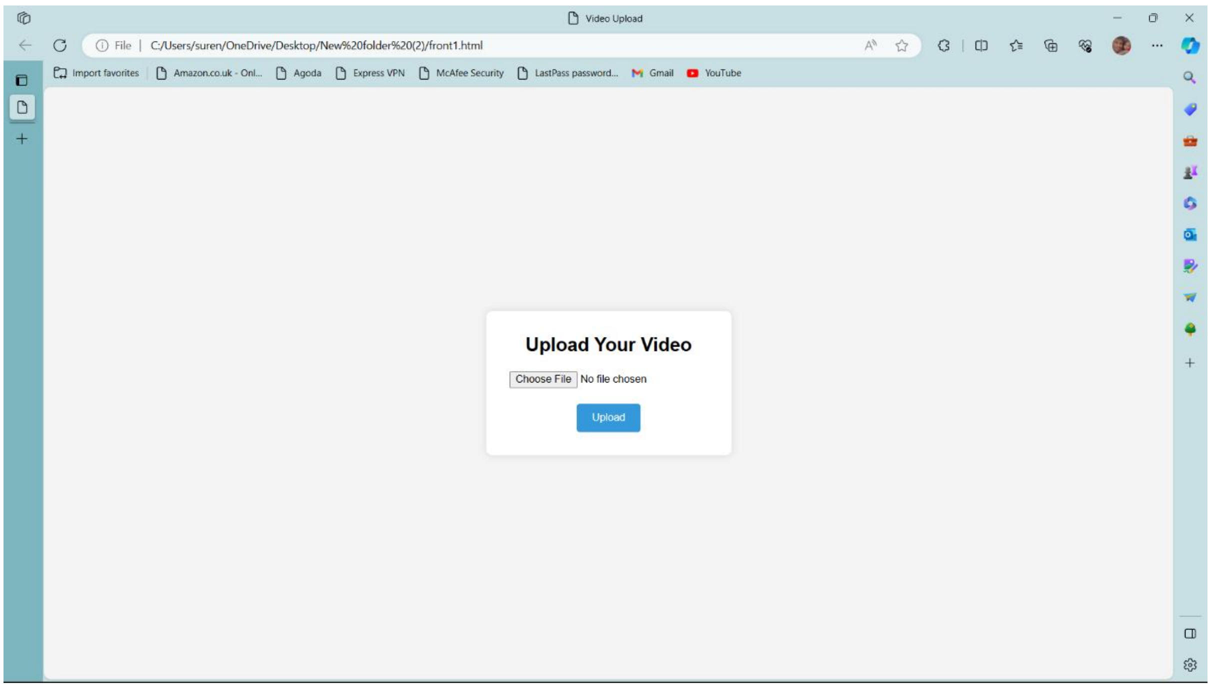
HTML (Upload Form):

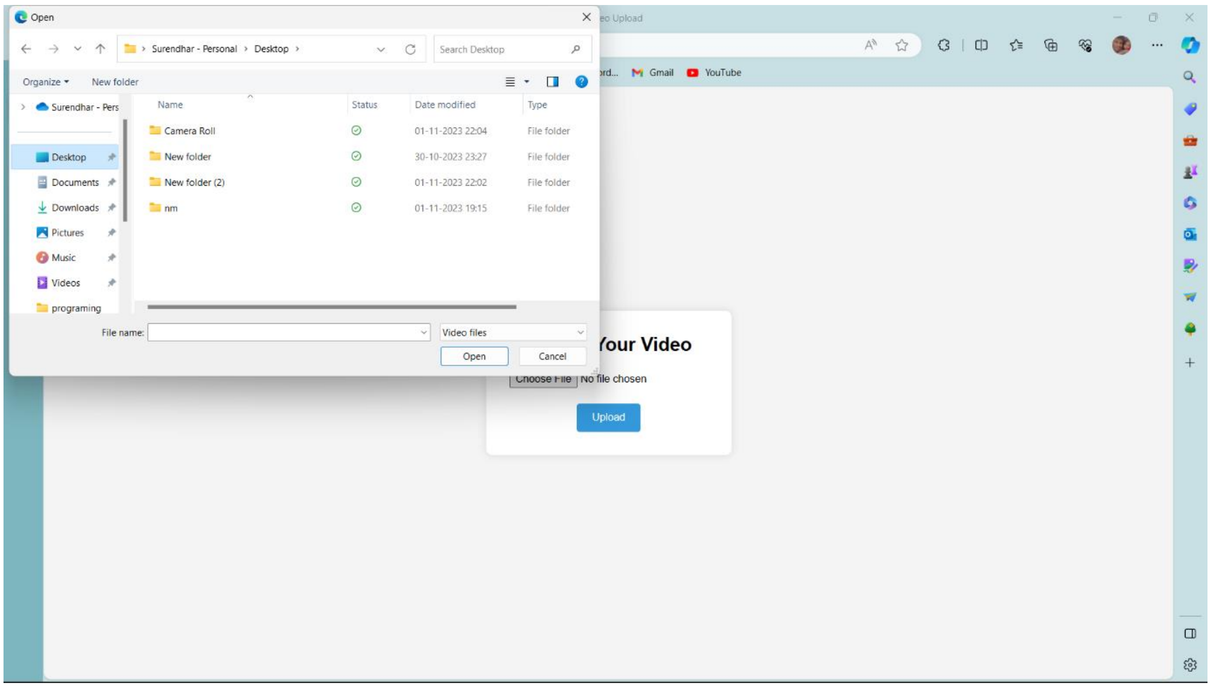
```
<!DOCTYPE html>

<html>

<head>

  <title>Video Upload</title>

  <style>

    body {

      font-family: Arial, sans-serif;

      background-color: #f4f4f4;

      margin: 0;

      padding: 0;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}
.upload-container {
  background-color: #fff;
  padding: 30px;
  border-radius: 8px;
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
}
.upload-container h2 {
  margin-top: 0;
  text-align: center;
}
.upload-form {
  display: flex;
  flex-direction: column;
  align-items: center;
}
.upload-form input[type="file"] {
      margin-bottom: 20px;
    }
    .upload-form button {
      padding: 10px 20px;
      background-color: #3498db;
      color: #fff;
      border: none;
      border-radius: 4px;
      cursor: pointer;
      transition: background-color 0.3s;
    }
    .upload-form button:hover {
      background-color: #2980b9;
    }
  </style>
</head>
<body>
  <div class="upload-container">
    <h2>Upload Your Video</h2>
    <form class="upload-form" id="videoUploadForm"
enctype="multipart/form-data">
      <input type="file" id="videoFile" accept="video/*" required>
      <button type="submit">Upload</button>

    </form>

  </div>


  <script>

document.getElementById('videoUploadForm').addEventListener('su
bmit', function(event) {

      event.preventDefault();

      const videoFile = document.getElementById('videoFile').files[0];


      // Simulate video upload by logging file details

      console.log('Uploaded Video:', videoFile);


      // Additional steps: Use Fetch API or other means to send the
video file to the backend

      // For this example, we simulate file upload and log the file
details.

    });

  </script>

</body>

</html>
```

Backend Integration for Video Storage and IBM Cloud Video Streaming:

1. Backend Server: Set up a backend server (using Node.js, Python, etc.) to handle file uploads, process the videos, and store them securely. Use a framework like Express.js (for Node.js) to create endpoints for handling file uploads.
2. IBM Cloud Video Integration: Use IBM Cloud Video Streaming API to manage and deliver videos. The IBM Cloud Video API allows you to create channels, upload videos, manage video metadata, and retrieve playback URLs.
3. Steps for Backend:
   - Receive uploaded video files from the frontend.
   - Store these files securely (consider using a cloud storage service like IBM Cloud Object Storage).
   - Utilize IBM Cloud Video API to create video assets, upload videos, and generate playback URLs.

```
const express = require('express');
const multer = require('multer');
const ibmVideoService = require('ibm-video-service-library'); // Import the library for IBM Video Service


const app = express();
const upload = multer({ dest: 'uploads/' });


// Endpoint to handle video uploads
app.post('/upload-video', upload.single('video'), async (req, res) => {
  const videoFile = req.file;
  // Handle storing the file securely and then use IBM Video API to upload the video
  const ibmVideo = new ibmVideoService();
  const videoAsset = await ibmVideo.uploadVideo(videoFile.path);

  const playbackURL = videoAsset.playbackURL;

  res.json({ playbackURL });
});

app.listen(3000, () => {
  console.log('Server is running on port 3000');
});
```





Name	Status	Date modified	Type
Camera Roll	✓	01-11-2023 22:04	File folder
New folder	✓	30-10-2023 23:27	File folder
New folder (2)	✓	01-11-2023 22:02	File folder
nm	✓	01-11-2023 19:15	File folder