```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>4-Video Editor</title>
    <!-- Firebase & FFmpeg CDN -->
    <script src="https://www.gstatic.com/firebasejs/9.6.10/firebase-app.js"></script>
    <script src="https://www.gstatic.com/firebasejs/9.6.10/firebase-auth.js"></script>
    <script src="https://www.gstatic.com/firebasejs/9.6.10/firebase-storage.js"></script>
    <script src="https://unpkg.com/@ffmpeg/ffmpeg@0.11.0/dist/ffmpeg.min.js"></script>

    <style>
        body { font-family: Arial; padding: 20px; }
        .container {
            display: grid;
            grid-template-columns: repeat(2, 1fr);
            gap: 10px;
            margin: 20px 0;
        }
        .video-box {
            border: 2px dashed #ccc;
            height: 200px;
            display: flex;
            align-items: center;
            justify-content: center;
            cursor: pointer;
        }
        #textInput {
            width: 300px;
            padding: 10px;
            margin: 10px 0;
        }
        #previewVideo { width: 100%; margin-top: 20px; }
    </style>
</head>
<body>
    <!-- Login Button -->
    <button id="googleLogin">Google से Login करें</button>

    <!-- Video Upload Grid -->
    <div class="container">
        <div class="video-box" data-id="0">Video 1 Upload</div>
        <div class="video-box" data-id="1">Video 2 Upload</div>
        <div class="video-box" data-id="2">Video 3 Upload</div>
```

```html
    <div class="video-box" data-id="3">Video 4 Upload</div>
</div>

<!-- Text Overlay Input -->
<input type="text" id="textInput" placeholder="Center में Text लिखें">

<!-- Create Video Button -->
<button onclick="processVideos()">Create Video</button>

<!-- Preview Video -->
<video id="previewVideo" controls></video>

<script>
    // Firebase Configuration (Replace with your keys)
    const firebaseConfig = {
        apiKey: "YOUR_API_KEY",
        authDomain: "YOUR_PROJECT_ID.firebaseapp.com",
        projectId: "YOUR_PROJECT_ID",
        storageBucket: "YOUR_PROJECT_ID.appspot.com"
    };
    firebase.initializeApp(firebaseConfig);
    const auth = firebase.auth();

    // Google Login
    document.getElementById('googleLogin').addEventListener('click', () => {
        const provider = new firebase.auth.GoogleAuthProvider();
        auth.signInWithPopup(provider).then((user) => {
            alert('Login Success: ' + user.user.email);
        });
    });

    // Video Upload Logic
    let uploadedVideos = [];
    document.querySelectorAll('.video-box').forEach(box => {
        box.addEventListener('click', () => {
            const input = document.createElement('input');
            input.type = 'file';
            input.accept = 'video/*';
            input.onchange = (e) => {
                const file = e.target.files[0];
                const videoURL = URL.createObjectURL(file);
                box.innerHTML = `<video src="${videoURL}"
style="width:100%;height:100%;"></video>`;
                uploadedVideos[box.dataset.id] = file;
```

```
        };
        input.click();
      });
    });

    // Auto-Pause Other Videos
    document.addEventListener('play', (e) => {
      const videos = document.querySelectorAll('video');
      videos.forEach(video => {
        if (video !== e.target) video.pause();
      });
    }, true);

    // Process Videos with FFmpeg
    async function processVideos() {
      const ffmpeg = new FFmpeg();
      await ffmpeg.load();

      // Upload Videos to FFmpeg
      for (let i=0; i<uploadedVideos.length; i++) {
        const data = await uploadedVideos[i].arrayBuffer();
        ffmpeg.writeFile(`input${i}.mp4`, new Uint8Array(data));
      }

      // Merge Videos & Add Text
      await ffmpeg.exec([
        '-i', 'input0.mp4', '-i', 'input1.mp4',
        '-filter_complex',
        `concat=n=${uploadedVideos.length}:v=1:a=1[outv][outa];

[outv]drawtext=text='${document.getElementById('textInput').value}':x=(w-tw)/2:y=(h-th)/2:fontsiz
e=24:fontcolor=white[outv]`,
        '-map', '[outv]', '-map', '[outa]', 'output.mp4'
      ]);

      // Download Final Video
      const output = await ffmpeg.readFile('output.mp4');
      const blob = new Blob([output], { type: 'video/mp4' });
      const url = URL.createObjectURL(blob);
      document.getElementById('previewVideo').src = url;
    }
  </script>
</body>
</html>
```