This is a fantastic video asset. The vertical aspect ratio (9:16) and the dark, cinematic lighting are perfect for a mobile-first "story" style loading screen. The steampunk/hologram aesthetic adds a very distinct, high-tech flavor to the app.
Since you are acting as both the designer and developer here, I have synthesized the design brief into a concrete implementation.

The Design Direction: "Cinematic Immersion"

Given the video is vertical and visually rich, we shouldn't hide it in a small card.

1. **Full-Screen Background:** The video will take up the entire screen (object-cover), creating an immersive experience.
2. **Glassmorphism Overlay:** We will place the text in a "glass" container at the bottom third of the screen. This ensures the text is legible against the busy background without blocking the robot's face or the holographic display.
3. **Progressive Loading:** A slim, high-tech progress bar at the very bottom to indicate the system is working, rather than a generic spinner which might clash with the custom video.

Here is the React component using Tailwind CSS.

LoadingPearls.tsx

```
import React, { useState, useEffect } from 'react';

// Sample pearls - replace with your actual UroRads data source
const RADIOLOGY_PEARLS = [
  "Look for the 'rim sign' in ureteral stones: soft tissue thickening
around the calcification distinguishes it from a phlebolith.",
  "In renal trauma, the presence of contrast extravasation implies
active bleeding (vascular injury) or urine leak (collecting system
injury).",
  "Xanthogranulomatous pyelonephritis (XGP) often presents with the
'bear paw' sign on CT: dilated calyces mimicking paw pads.",
  "Always check the lung bases on a KUB CT. Roughly 5% of incidental
findings in urologic scans are pulmonary.",
  "Bladder rupture: Intraperitoneal rupture shows contrast outlining
bowel loops; extraperitoneal shows the 'molar tooth' sign."
];

interface LoadingPearlsProps {
  videoSrc: string; // Pass the imported video file path here
}

const LoadingPearls: React.FC<LoadingPearlsProps> = ({ videoSrc }) =>
{
  const [currentPearlIndex, setCurrentPearlIndex] = useState(0);
  const [isVisible, setIsVisible] = useState(true);
  const [progress, setProgress] = useState(0);

  // Cycle pearls every 9 seconds
  useEffect(() => {
    const interval = setInterval(() => {
      // Fade out
      setIsVisible(false);

      // Wait for fade out to finish, then swap text and fade in
      setTimeout(() => {
        setCurrentPearlIndex((prev) => (prev + 1) %
RADIOLOGY_PEARLS.length);
        setIsVisible(true);
      }, 500); // 500ms fade out duration

    }, 9000);

    return () => clearInterval(interval);
  }, []);

  // Fake progress bar for demonstration (Remove if you have real
upload progress props)
  useEffect(() => {
    const timer = setInterval(() => {
      setProgress((oldProgress) => {
        if (oldProgress === 100) return 0;
        const diff = Math.random() * 10;
        return Math.min(oldProgress + diff, 100);
      });
    }, 500);
    return () => clearInterval(timer);
  }, []);

  return (
    <div className="relative w-full h-full min-h-screen bg-black
overflow-hidden flex flex-col items-center justify-end">

      {/* 1. Background Video Layer */}
      <div className="absolute inset-0 z-0">
        <video
          autoPlay
          loop
          muted
          playsInline
          className="w-full h-full object-cover opacity-90"
        >
          <source src={videoSrc} type="video/mp4" />
        </video>
        {/* Gradient Overlay: Darkens the bottom so text is readable
*/}
        <div className="absolute inset-0 bg-gradient-to-t from-black
```

```
via-black/40 to-transparent" />
      </div>

      {/* 2. Content Layer */}
      <div className="relative z-10 w-full max-w-md px-6 pb-12
text-center">

        {/* Status Label */}
        <div className="mb-4 flex items-center justify-center gap-2">
          <span className="relative flex h-3 w-3">
            <span className="animate-ping absolute inline-flex h-full
w-full rounded-full bg-teal-400 opacity-75"></span>
            <span className="relative inline-flex rounded-full h-3 w-3
bg-teal-500"></span>
          </span>
          <p className="text-teal-400 font-mono text-xs
tracking-widest uppercase">
            Analyzing Dicom Data
          </p>
        </div>

        {/* The Pearl Text */}
        <div className="min-h-[120px] flex items-center
justify-center">
          <p
            className={`
              text-lg md:text-xl font-medium text-slate-100
leading-relaxed drop-shadow-md
              transition-opacity duration-500 ease-in-out
              ${isVisible ? 'opacity-100 translate-y-0' : 'opacity-0
translate-y-2'}
            `}
          >
            "{RADIOLOGY_PEARLS[currentPearlIndex]}"
          </p>
        </div>

        {/* Branding / Footer */}
        <p className="text-slate-500 text-sm mt-4 font-light">
          UroRads AI • Educational Pearl
        </p>

        {/* 3. Progress Bar (Optional) */}
        <div className="mt-8 w-full bg-gray-800/50 rounded-full h-1.5
backdrop-blur-sm overflow-hidden">
          <div
            className="bg-teal-500 h-1.5 rounded-full transition-all
duration-300 ease-out box-shadow-glow"
```

```
            style={{ width: `${progress}%`, boxShadow: '0 0 10px 
#14b8a6' }}
          ></div>
        </div>

      </div>
    </div>
  );
};

export default LoadingPearls;
```

Implementation Notes for You:

1. **Video Placement:** I utilized playsInline in the video tag. This is critical for iOS mobile browsers; otherwise, the video might try to force full-screen native playback mode, breaking your UI overlay.
2. **Text Contrast:** I added a CSS gradient overlay (bg-gradient-to-t) starting from the bottom. This ensures that even if the video has bright spots (like the holographic chest scan), the white text remains perfectly readable.
3. **Typography:** I used standard Tailwind fonts, but this looks best with a clean sans-serif like Inter. I added a drop-shadow-md to the text to separate it from the 3D rendered background.
4. **Audio:** The video you uploaded has sound (typing/robot noises). I added the muted attribute to the video tag. Browsers generally block autoplay video unless it is muted. If you *want* sound, you will need to add a "Unmute" button, but for a loading screen, silent is usually standard.

Would you like me to adjust the transition timing or the styling of the progress bar?