



**Fig 1 : Video Core Component**

## Code

```
import React, { useState, useEffect } from "react";
import { Image, mergeClasses } from "@fluentui/react-components";
import { useStyles } from "../ZVideocss";
import { FullScreenMaximizeFilled } from "@fluentui/react-icons";

interface ZVideoProps {
  videourl: string;
  className: string;
  height?: string;
  width?: string;
  minHeight?: string;
  minWidth?: string;
  maxHeight?: string;
  maxWidth?: string;
}

const ZVideo: React.FC<ZVideoProps> = ({
  videourl,
  className,
  height,
  width,
  minHeight,
  minWidth,
  maxHeight,
  maxWidth,
  ...props
}) => {
  const classes = useStyles();
  const [videoPlaying, setVideoPlaying] = useState(false);

  useEffect(() => {
    const handleKeyDown = (event: KeyboardEvent) => {
      if (event.key === " ") {
        toggleVideo();
      }
    };
  });
}
```

```

    }
  };

  document.addEventListener("keydown", handleKeyDown);
  return () => {
    document.removeEventListener("keydown", handleKeyDown);
  };
}, []);

const handlePlay = () => {
  setVideoPlaying(true);
};

const toggleFullScreen = () => {
  const videoElement = document.querySelector(
    ".video-container video"
  ) as HTMLVideoElement;
  if (videoElement) {
    if (!document.fullscreenElement) {
      videoElement.requestFullscreen().then(() => {
        setVideoPlaying(true);
      });
    } else {
      document.exitFullscreen().then(() => {
        setVideoPlaying(false);
      });
    }
  }
};

const toggleVideo = () => {
  setVideoPlaying(prevState => !prevState);
};

return (
  <div
    className={mergeClasses(classes.ZVideoContainer, className)}
    {...props}
  >
    {!videoPlaying && (
      <div className={classes.ZVideoContainer_wrapper} onClick={handlePlay}>
        <Image src="https://s3.ap-south-
1.amazonaws.com/dev.zitrics.com/media/core/assets/image/zvideoplaylogo.svg" />

```

```

        <div
          className={classes.ZVideoFullScreenButton}
          onClick={toggleFullScreen}
        ></div>
      </div>
    )}
    {videoPlaying && (
      <div
        className={` ${classes.ZVideoContainer_video} video-container`}
        style={{ height, width, minHeight, minWidth, maxHeight, maxWidth }}
      >
        <video
          className={classes.ZVideoContainer_videocontainer}
          autoPlay
          onClick={() => setVideoPlaying(false)}
        >
          <source src={videourl} />
        </video>
      </div></div>
      <div
        className={classes.ZVideoFullScreenButton}
        onClick={toggleFullScreen}
      >
        <div className={classes.ZBottomRightContainer}>
          <FullScreenMaximizeFilled
            primaryFill="white"
            className={classes.fullscreenbtn}
          />
        </div>
      </div>
    </div>
  )}
</div>
);
};

export default ZVideo;

import { makeStyles } from "@fluentui/react-components";

export const useStyles = makeStyles({

```

```
ZVideoContainer: {
  width: '100%',
  height: '207px',
  borderRadius: '10px',
  background: 'linear-gradient(115deg, #969696 9.36%, #3E3E3E 95.32%)',
  position: 'relative',
},
ZVideoContainer_wrapper: {
  position: 'absolute',
  top: '50%',
  left: '50%',
  transform: 'translate(-50%,-50%)',
  cursor: 'pointer',
},
ZVideoContainer_videocontainer: {
  width: '100%',
  height: '100%',
  objectFit: 'cover',
  borderRadius: '10px',
},
ZVideoFullScreenButton: {
  position: 'absolute',
  bottom: '2px',
  right: '10px',
  cursor: 'pointer',
  padding: '0px 8px'
},
fullscreenbtn: {
  width: "25px",
  height: "50px",
  display: "flex",

},
ZBottomRightContainer: {
  display: "flex",
  gap: "20px",
},
ZVideoContainer_video:{
  width: "100%",
  height: "100%",
  top: '0%',
  left: '0%',
  objectFit: 'cover',
},
```

```
});
```

App.tsx

```
<ZVideo  
    videourl={"http://commondatastorage.googleapis.com/gtv-videos-  
bucket/sample/BigBuckBunny.mp4"}  
    className="" />
```

## \* Props For the Video Component

```
* videourl,    -----> To Add a VideoUrl is Compulsory.....  
* className,  
* height,  
* width,  
* minHeight,  
* minWidth,  
* maxHeight,  
* maxWidth,
```

## \* How to use this component

First up All Render That Component Whenever It is Required and then add a props of VideoUrl in ZVideo Component which is a mandatory Props... and all other Props are Secondary.....

possible ways to improve this component....?

> We Can add a Live Streaming on This Video Using our Platform Own Emmeded Code for example jio Cinema Has Its own Code Embeeded Platform for Live Streaming....