

Virtual Escape : The Cursed Mansion

Fun Event Gaming Website where participants must solve different puzzles in each level to escape and win. It consists of 6 levels. It is a 4 to 5 hr long group game.

Built Using :

ReactJS, MaterialUI for styled components, Lots of AI generated assets for horror experience.

Scary Effects :

The element fades in and out at random times, creating an unpredictable, eerie effect that keeps the viewer on edge. The fonts, colors, typewriter effects, backgrounds, and assets all contribute to creating an eerie atmosphere.

About Game :

Used Local Storage to store Health of Player

Used React Context to set a global variable Health and manage it from every component.

Used UseTimeOut function to make Hint Visible for 60 seconds

Protected URL so user can not jump to next level by changing the URL

Score System :

Initial Health = 500,

Failing to give correct answer reduces the Health by 50

Taking hint for 60 sec reduces the Health by 100

Giving Up in Warning Page reduces to Health to 0

Mission - To escape all the rooms with Health > 0

Full Screen Mode :

When the Full Screen button is clicked, a function is called, which either requests the document to enter or exit full-screen mode depending on the current state.

```

if (!isFullScreen) {
  if (document.documentElement.requestFullscreen) {
    document.documentElement.requestFullscreen().catch((err) => {
      console.error('Error attempting to enable full-screen mode:', err);
    });
  }
} else {
  if (document.exitFullscreen) {
    document.exitFullscreen();
  }
}

```

An event triggers another function whenever the document's full-screen state changes. This function updates the `isFullScreen` state to reflect the current full-screen status.

```

document.addEventListener('fullscreenchange',
setIsFullScreen(document.fullscreenElement !== null));

```

React re-renders the component based on the updated state, changing the button's text accordingly.

This setup ensures that the component is aware of whether it is in full-screen mode and can toggle this mode on user interaction.

Sound Effect :

Users can toggle an audio file on and off with a button. The button's label changes based on the audio's status, and the audio loops while playing.

```
<audio ref={audioRef} loop><source src={sound_effect}></audio>
```

```

const [notPlaying, setnotPlaying] = useState(true);
const audioRef = React.createRef();
const toggleAudio = () => {
  if (notPlaying) {
    audioRef.current.play();
  } else {
    audioRef.current.pause();
  }
  setnotPlaying(!notPlaying);
};

```

Typeewriter Effect :

```

const [text, setText] = useState('');
const [textIndex, setTextIndex] = useState(0);
const panelText = "You're all set, so get ready for a";
useEffect(() => {
  const interval = setInterval(() => {
    if (textIndex < panelText.length) {
      setText(panelText.substring(0, textIndex + 1));
      setTextIndex(textIndex + 1);
    } else {
      clearInterval(interval);
    }
  }, 20);
  return () => {
    clearInterval(interval);
  };
}, [panelText, textIndex]);

```

designed to create a typing animation effect, where text is progressively revealed on the screen.

Authentication Check :

Initial Check: When a user accesses any page, useEffect checks if they are authenticated. If not, they are sent back to the homepage.

Conditional Navigation: If the user provides a correct inputValue, they are redirected to "/specific_page" with auth: true in the state, allowing them access to that page.

Eg :

```

useEffect(() => {
  if (!location.state || !location.state.auth) {
    | navigate("/");
  }
}, []);

```

This ensures that only authenticated users (those who have auth: true in their state) can access the current page. If not, they are redirected to the homepage.

```

if (inputValue === "123489") {
  navigate("/page4", {
    state: {
      auth: true,
    },
  });
} else {

```

This indicates that the user is authenticated, as auth: true is stored in the state. When the user is redirected to "/page4", the state will contain auth: true, which allows them to access "/page4".



