

## Exercise 2: create a basic calculator using react js

### KeyPadComponents.js

```
import React, { Component } from 'react';

class KeyPadComponent extends Component {
  render() {
    return (
      <div className="button">
        <button name="(" onClick={e =>
this.props.onClick(e.target.name)}>(</button>
        <button name="CE" onClick={e =>
this.props.onClick(e.target.name)}>CE</button>
        <button name=")" onClick={e =>
this.props.onClick(e.target.name)}>></button>
        <button name="C" onClick={e =>
this.props.onClick(e.target.name)}>C</button><br/>

        <button name="1" onClick={e =>
this.props.onClick(e.target.name)}>1</button>
        <button name="2" onClick={e =>
this.props.onClick(e.target.name)}>2</button>
        <button name="3" onClick={e =>
this.props.onClick(e.target.name)}>3</button>
        <button name="+" onClick={e =>
this.props.onClick(e.target.name)}>+</button><br/>

        <button name="4" onClick={e =>
this.props.onClick(e.target.name)}>4</button>
        <button name="5" onClick={e =>
this.props.onClick(e.target.name)}>5</button>
        <button name="6" onClick={e =>
this.props.onClick(e.target.name)}>6</button>
        <button name="-" onClick={e => this.props.onClick(e.target.name)}>-
</button><br/>

        <button name="7" onClick={e =>
this.props.onClick(e.target.name)}>7</button>
        <button name="8" onClick={e =>
this.props.onClick(e.target.name)}>8</button>
        <button name="9" onClick={e =>
this.props.onClick(e.target.name)}>9</button>
        <button name="*" onClick={e =>
this.props.onClick(e.target.name)}>x</button><br/>
      </div>
    );
  }
}
```

```

        <button name="." onClick={e =>
this.props.onClick(e.target.name)}>.</button>
        <button name="0" onClick={e =>
this.props.onClick(e.target.name)}>0</button>
        <button name="=" onClick={e =>
this.props.onClick(e.target.name)}>=</button>
        <button name="/" onClick={e =>
this.props.onClick(e.target.name)}>÷</button><br/>
    </div>
  )
}
}

export default KeyPadComponent;

```

## ResultComponent.js

```

import React, { Component } from 'react';

class ResultComponent extends Component {
  render() {
    let { result } = this.props;

    return (
      <div className="result">
        <p>{ result }</p>
      </div>
    )
  }
}

export default ResultComponent;

```

## App.js

```

import React, { Component } from 'react';
import './styles.css';
import ResultComponent from './components/ResultComponent';
import KeyPadComponent from './components/KeyPadComponent';

```

```

class App extends Component {
  state = {
    result: ""
  }

  onClick = button => {
    if(button === "=") {
      this.calculate();
    }

    else if(button === "C") {
      this.reset();
    }

    else if(button === "CE") {
      this.backspace();
    }

    else {
      this.setState({
        result: this.state.result + button
      })
    }
  };

  calculate = () => {
    var checkResult = ''
    if(this.state.result.includes('--')) {
      checkResult = this.state.result.replace('--', '+')
    } else {
      checkResult = this.state.result;
    }

    try {
      this.setState({
        result: (eval(checkResult) || "") + ""
      })
    } catch(e) {
      this.setState({
        result: "error"
      })
    }
  };

  reset = () => {
    this.setState({
      result: ""
    })
  }
}

```

```

};

backspace = () => {
  this.setState({
    result: this.state.result.slice(0, -1)
  })
};

render() {
  return (
    <div>
      <div className="calculator-body">
        <h1>Simple Calculator</h1>
        <ResultComponent result={this.state.result} />
        <KeyPadComponent onClick={this.onClick} />
      </div>
    </div>
  )
}
}

export default App;

```

## index.js

```

import React from "react";
import ReactDOM from "react-dom";

import App from "./App";

const rootElement = document.getElementById("root");
ReactDOM.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>,
  rootElement
);

```

## Style.js

```

.result {

```

```

    height: 60px;
    background-color: #bbb;
    width: 100%;
}

.result p {
    font-size: 40px;
    margin: 5px;
}

.calculator-body {
    max-width: 400px;
    margin: auto;
}

.button {
    display: block;
    background-color: #bbb;
}

button {
    width: 25%;
    height: 60px;
    font-size: 30px;
}

```

## OUTPUT

### Simple Calculator

3*3			
(	CE	)	C
1	2	3	+
4	5	6	-
7	8	9	x
.	0	=	÷

### Exercise 3: Build a music store application using react components

#### AudioControls.jsx

```
import React from "react";
import { ReactComponent as Play } from "../assets/play.svg";
import { ReactComponent as Pause } from "../assets/pause.svg";
import { ReactComponent as Next } from "../assets/next.svg";
import { ReactComponent as Prev } from "../assets/prev.svg";

const AudioControls = ({
  isPlaying,
  onPlayPauseClick,
  onPrevClick,
  onNextClick
}) => (
  <div className="audio-controls">
    <button
      type="button"
      className="prev"
      aria-label="Previous"
      onClick={onPrevClick}
    >
      <Prev />
    </button>
    {isPlaying ? (
      <button
        type="button"
        className="pause"
        onClick={() => onPlayPauseClick(false)}
        aria-label="Pause"
      >
        <Pause />
      </button>
    ) : (
      <button
        type="button"
        className="play"
        onClick={() => onPlayPauseClick(true)}
        aria-label="Play"
      >
        <Play />
      </button>
    )}
    <button
      type="button"
      className="next"
      aria-label="Next"
      onClick={onNextClick}
    >
```

```

    >
    <Next />
  </button>
</div>
);

export default AudioControls;

```

### AudioPlayer.jsx

```

import React, { useState, useEffect, useRef } from "react";
import AudioControls from "../AudioControls";
import Backdrop from "../Backdrop";
import "../styles.css";

/*
 * Read the blog post here:
 * https://letsbuildui.dev/articles/building-an-audio-player-with-react-hooks
 */
const AudioPlayer = ({ tracks }) => {
  // State
  const [trackIndex, setTrackIndex] = useState(0);
  const [trackProgress, setTrackProgress] = useState(0);
  const [isPlaying, setIsPlaying] = useState(false);

  // Destructure for conciseness
  const { title, artist, color, image, audioSrc } = tracks[trackIndex];

  // Refs
  const audioRef = useRef(new Audio(audioSrc));
  const intervalRef = useRef();
  const isReady = useRef(false);

  // Destructure for conciseness
  const { duration } = audioRef.current;

  const currentPercentage = duration
    ? `${(trackProgress / duration) * 100}%`
    : "0%";
  const trackStyling = `
    -webkit-gradient(linear, 0% 0%, 100% 0%, color-stop(${currentPercentage},
    #fff), color-stop(${currentPercentage}, #777))
  `;

  const startTimer = () => {

```

```

// Clear any timers already running
clearInterval(intervalRef.current);

intervalRef.current = setInterval(() => {
  if (audioRef.current.ended) {
    toNextTrack();
  } else {
    setTrackProgress(audioRef.current.currentTime);
  }
}, [1000]);
};

const onScrub = (value) => {
  // Clear any timers already running
  clearInterval(intervalRef.current);
  audioRef.current.currentTime = value;
  setTrackProgress(audioRef.current.currentTime);
};

const onScrubEnd = () => {
  // If not already playing, start
  if (!isPlaying) {
    setIsPlaying(true);
  }
  startTimer();
};

const toPrevTrack = () => {
  if (trackIndex - 1 < 0) {
    setTrackIndex(tracks.length - 1);
  } else {
    setTrackIndex(trackIndex - 1);
  }
};

const toNextTrack = () => {
  if (trackIndex < tracks.length - 1) {
    setTrackIndex(trackIndex + 1);
  } else {
    setTrackIndex(0);
  }
};

useEffect(() => {
  if (isPlaying) {
    audioRef.current.play();
    startTimer();
  } else {

```



```

        audioRef.current.pause();
    }
}, [isPlaying]);

// Handles cleanup and setup when changing tracks
useEffect(() => {
    audioRef.current.pause();

    audioRef.current = new Audio(audioSrc);
    setTrackProgress(audioRef.current.currentTime);

    if (isReady.current) {
        audioRef.current.play();
        setIsPlaying(true);
        startTimer();
    } else {
        // Set the isReady ref as true for the next pass
        isReady.current = true;
    }
}, [trackIndex]);

useEffect(() => {
    // Pause and clean up on unmount
    return () => {
        audioRef.current.pause();
        clearInterval(intervalRef.current);
    };
}, []);

return (
    <div className="audio-player">
        <div className="track-info">
            <img
                className="artwork"
                src={image}
                alt={`track artwork for ${title} by ${artist}`}
            />
            <h2 className="title">{title}</h2>
            <h3 className="artist">{artist}</h3>
            <AudioControls
                isPlaying={isPlaying}
                onPrevClick={toPrevTrack}
                onNextClick={toNextTrack}
                onPlayPauseClick={setIsPlaying}
            />
            <input
                type="range"
                value={trackProgress}

```

```

        step="1"
        min="0"
        max={duration ? duration : `${duration}`}
        className="progress"
        onChange={(e) => onScrub(e.target.value)}
        onMouseUp={onScrubEnd}
        onKeyUp={onScrubEnd}
        style={{ background: trackStyling }}
      />
    </div>
    <Backdrop
      trackIndex={trackIndex}
      activeColor={color}
      isPlaying={isPlaying}
    />
  </div>
);
};

```

```
export default AudioPlayer;
```

#### **Backdrop.jsx**

```

import React, { useEffect } from "react";

const Backdrop = ({ activeColor, trackIndex, isPlaying }) => {
  useEffect(() => {
    document.documentElement.style.setProperty("--active-color", activeColor);
  }, [trackIndex, activeColor]);

  return <div className={`color-backdrop ${isPlaying ? "playing" : "idle"} `>
/>;
};

export default Backdrop;

```

#### **index.Js**

```

import React from "react";
import ReactDOM from "react-dom";
import AudioPlayer from "../AudioPlayer";
import tracks from "../tracks";

const rootElement = document.getElementById("root");
ReactDOM.render(
  <React.StrictMode>

```

```

    <AudioPlayer tracks={tracks} />
  </React.StrictMode>,
  rootElement
);

```

### styles.css

```

:root {
  --white: #fff;
  --active-color: #00aeb0;
}

* {
  box-sizing: border-box;
}

html {
  font-family: Arial, Helvetica, sans-serif;
  height: 100%;
  background: var(--active-color);
  transition: background 0.4s ease;
}

button {
  background: none;
  border: none;
  cursor: pointer;
}

input[type="range"] {
  height: 5px;
  -webkit-appearance: none;
  width: 100%;
  margin-bottom: 10px;
  border-radius: 8px;
  background: #3b7677;
  transition: background 0.2s ease;
  cursor: pointer;
}

.audio-player {
  max-width: 350px;
  padding: 24px;
  border-radius: 20px;
  box-shadow: 0 28px 28px rgba(0, 0, 0, 0.2);
  margin: auto;
}

```

```
    color: var(--white);
}

.artwork {
  border-radius: 120px;
  display: block;
  margin: auto;
  height: 200px;
  width: 200px;
}

.track-info {
  text-align: center;
  z-index: 1;
  position: relative;
}

.title {
  font-weight: 700;
  margin-bottom: 4px;
}

.artist {
  font-weight: 300;
  margin-top: 0;
}

.audio-controls {
  display: flex;
  justify-content: space-between;
  width: 75%;
  margin: 0 auto 15px;
}

.audio-controls .prev svg,
.audio-controls .next svg {
  width: 35px;
  height: 35px;
}

.audio-controls .play svg,
.audio-controls .pause svg {
  height: 40px;
  width: 40px;
}

.audio-controls path {
  fill: var(--white);
}
```

```

}

.color-backdrop {
  background: linear-gradient(45deg, var(--active-color) 20%, transparent
100%)
  no-repeat;
  position: absolute;
  top: 0;
  left: 0;
  right: 0;
  bottom: 0;
  z-index: -1;
}

.color-backdrop.playing {
  animation: colorChange 20s alternate infinite;
}

@keyframes colorChange {
  from {
    filter: hue-rotate(0deg);
  }
  to {
    filter: hue-rotate(360deg);
  }
}

```

### tracks.js

```

import imgSrc from "./assets/artwork.jpg";
import imgSrc2 from "./assets/artwork2.jpg";
import imgSrc3 from "./assets/artwork3.jpg";
import cali from "./assets/california-wataboi.mp3";
import fifty from "./assets/50-tobylane.mp3";
import iwonder from "./assets/iwonder-dreamheaven.mp3";

// All of these artists are at
https://pixabay.com/music/search/mood/laid%20back/
export default [
  {
    title: "Cali",
    artist: "Wataboi",
    audioSrc: cali,
    image: imgSrc,
    color: "#00aeb0"
  },

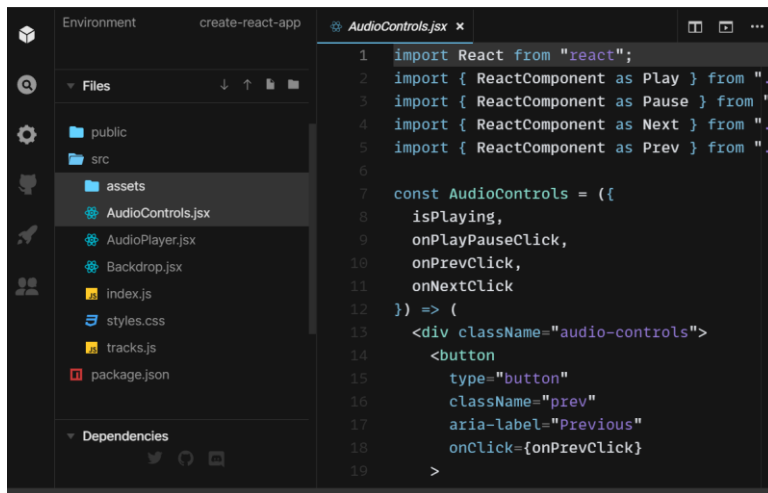
```

```

{
  title: "50",
  artist: "tobylane",
  audioSrc: fifty,
  image: imgSrc2,
  color: "#ffb77a"
},
{
  title: "I Wonder",
  artist: "DreamHeaven",
  audioSrc: iwonder,
  image: imgSrc3,
  color: "#5f9fff"
}
];

```

## Output



## Output

