Dileep Gedela

Case Study 2: Utilize framework for developing a React app from scratch

Code :

App.js

```
import React from 'react';
import Stopwatch from './components/Stopwatch';
import './App.css';
const App = () => {
  return (
    <div className="App">
      <Stopwatch />
    </div>
  );
};
export default App;
```

StopWatch.js

```
import React, { useState, useRef } from 'react';
import TimerDisplay from './TimerDisplay';
import Controls from './Controls';
import History from './History';
import './Stopwatch.css';

const Stopwatch = () => {
  const [time, setTime] = useState(0);
const [isRunning, setIsRunning] = useState(false);
const [history, setHistory] = useState([]);
const intervalRef = useRef(null);

const startTimer = () => {
  if (!isRunning) {
    setIsRunning(true);
    intervalRef.current = setInterval(() => {
      setTime(prevTime => prevTime + 10);
    }, 10);
  }
};

const pauseTimer = () => {
  if (isRunning) {
    clearInterval(intervalRef.current);
    setIsRunning(false);
  }
};

const lapTimer = () => {
  setHistory(prevHistory => [...prevHistory, time]);
};

const resetTimer = () => {
    if (isRunning) {
      clearInterval(intervalRef.current);
      setIsRunning(false);
    }
    setHistory(prevHistory => [...prevHistory, time]);
    setTime(0);
  };

  return (
    <div className="stopwatch-container">
      <div className="stopwatch">
        <TimerDisplay time={time} />
        <Controls
          isRunning={isRunning}
          startTimer={startTimer}
          pauseTimer={pauseTimer}
          lapTimer={lapTimer}
          resetTimer={resetTimer}
        />
        <History history={history} />
      </div>
    </div>
  );
};
```

```
export default Stopwatch;
```

TimerDisplay.js

```
import React from 'react';
import PropTypes from 'prop-types';
import './TimerDisplay.css';

const TimerDisplay = ({ time }) => {
  const secondDegree = (time / 1000) * 6;
  const minuteDegree = (time / 60000) * 6;
  const hourDegree = (time / 3600000) * 30;

  return (
    <div className="clock">
      <div className="hand hour" style={{ transform: `rotate(${hourDegree + 90}deg)` }}></div>
      <div className="hand minute" style={{ transform: `rotate(${minuteDegree + 90}deg)` }}></div>
      <div className="hand second" style={{ transform: `rotate(${secondDegree + 90}deg)` }}></div>
      <div className="center-point"></div>
      <div className="number number-12">12</div>
      <div className="number number-1">1</div>
      <div className="number number-2">2</div>
      <div className="number number-3">3</div>
      <div className="number number-4">4</div>
      <div className="number number-5">12</div>
      <div className="number number-6">6</div>
      <div className="number number-7">7</div>
      <div className="number number-8">8</div>
      <div className="number number-9">9</div>
      <div className="number number-10">10</div>
      <div className="number number-11">11</div>
    </div>
  );
};

TimerDisplay.propTypes = {
  time: PropTypes.number.isRequired,
};

export default TimerDisplay;
```

Controls.js

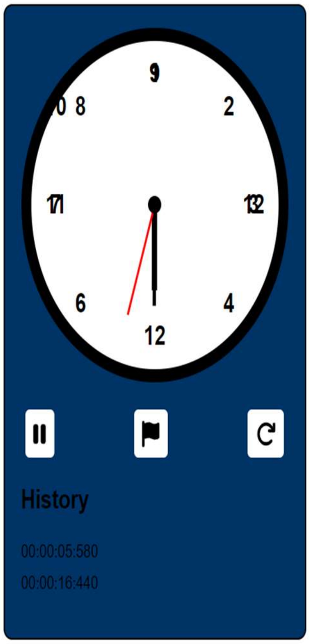
```
import React from 'react';
import PropTypes from 'prop-types';
import { FontAwesomeIcon } from '@fortawesome/react-fontawesome';
import { faPlay, faPause, faRedo, faFlag } from '@fortawesome/free-solid-svg-icons';
import './Controls.css';

const Controls = ({ isRunning, startTimer, pauseTimer, lapTimer, resetTimer }) =>
{
  return (
    <div className="controls">
      {!isRunning ? (
        <button onClick={startTimer} className="icon-button">
          <FontAwesomeIcon icon={faPlay} />
        </button>
      ) : (
        <button onClick={pauseTimer} className="icon-button">
          <FontAwesomeIcon icon={faPause} />
        </button>
      )}
      {isRunning && (
        <button onClick={lapTimer} className="icon-button">
          <FontAwesomeIcon icon={faFlag} />
        </button>
      )}
      <button onClick={resetTimer} className="icon-button">
        <FontAwesomeIcon icon={faRedo} />
      </button>
    </div>
  );
};
```

```
Controls.propTypes = {
  isRunning: PropTypes.bool.isRequired,
  startTimer: PropTypes.func.isRequired,
  pauseTimer: PropTypes.func.isRequired,
  lapTimer: PropTypes.func.isRequired,
  resetTimer: PropTypes.func.isRequired,
};

export default Controls;
```

History.js

```
import React from 'react';
import PropTypes from 'prop-types';
import './History.css';

const History = ({ history }) => {
  const formatTime = (time) => {
    const milliseconds = `0${time % 1000}`.slice(-3);
    const seconds = `0${Math.floor((time / 1000) % 60)}`.slice(-2);
    const minutes = `0${Math.floor((time / 60000) % 60)}`.slice(-2);
    const hours = `0${Math.floor(time / 3600000)}`.slice(-2);
    return `${hours}:${minutes}:${seconds}:${milliseconds}`;
  };

  return (
    <div className="history">
      <h2>History</h2>
      <ul>
        {history.map((time, index) => (
          <li key={index}>{formatTime(time)}</li>
        ))}
      </ul>
    </div>
  );
};

History.propTypes = {
  history: PropTypes.arrayOf(PropTypes.number).isRequired,
};

export default History;
```

Output:



```
History

00:00:05:580
00:00:16:440
```