

App.js

File Edit View

```
// App.js
import { useEffect, useState } from 'react';
import Axios from 'axios';
import Dropdown from 'react-dropdown';
import { HiSwitchHorizontal } from 'react-icons/hi';
import 'react-dropdown/style.css';
import './App.css';

function App() {

  // Initializing all the state variables
  const [info, setInfo] = useState({});
  const [input, setInput] = useState(0);
  const [from, setFrom] = useState("usd");
  const [to, setTo] = useState("inr");
  const [options, setOptions] = useState([]);
  const [output, setOutput] = useState(0);

  // Calling the api whenever the dependency changes
  useEffect(() => {
    Axios.get(
'https://cdn.jsdelivr.net/gh/fawazahmed0/currency-api@1/latest/currencies/${from}.json')
      .then((res) => {
        setInfo(res.data[from]);
      })
    }, [from]);

  // Calling the convert function whenever
  // a user switches the currency
  useEffect(() => {
    setOptions(Object.keys(info));
    convert();
  }, [info])

  // Function to convert the currency
  function convert() {
    var rate = info[to];
    setOutput(input * rate);
  }
}
```

Ln 88, Col 12,175 characters100%Windows (CRLF)UTF-8

33°C
Haze

ENG
IN19:48
05-04-2024

```
App.js
File Edit View
setOptions(Object.keys(info));
convert();
}, [info])

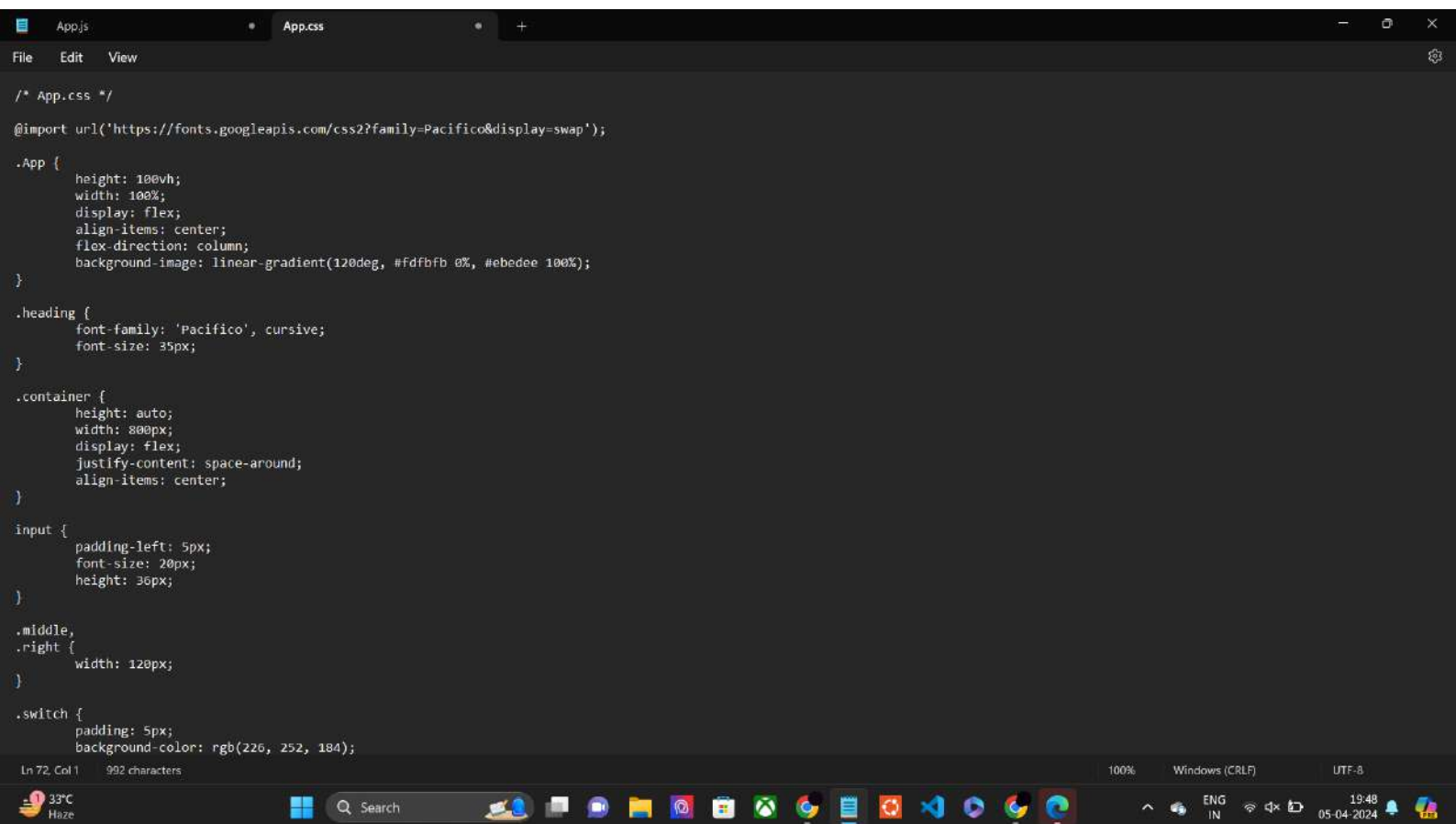
// Function to convert the currency
function convert() {
  var rate = info[to];
  setOutput(input * rate);
}

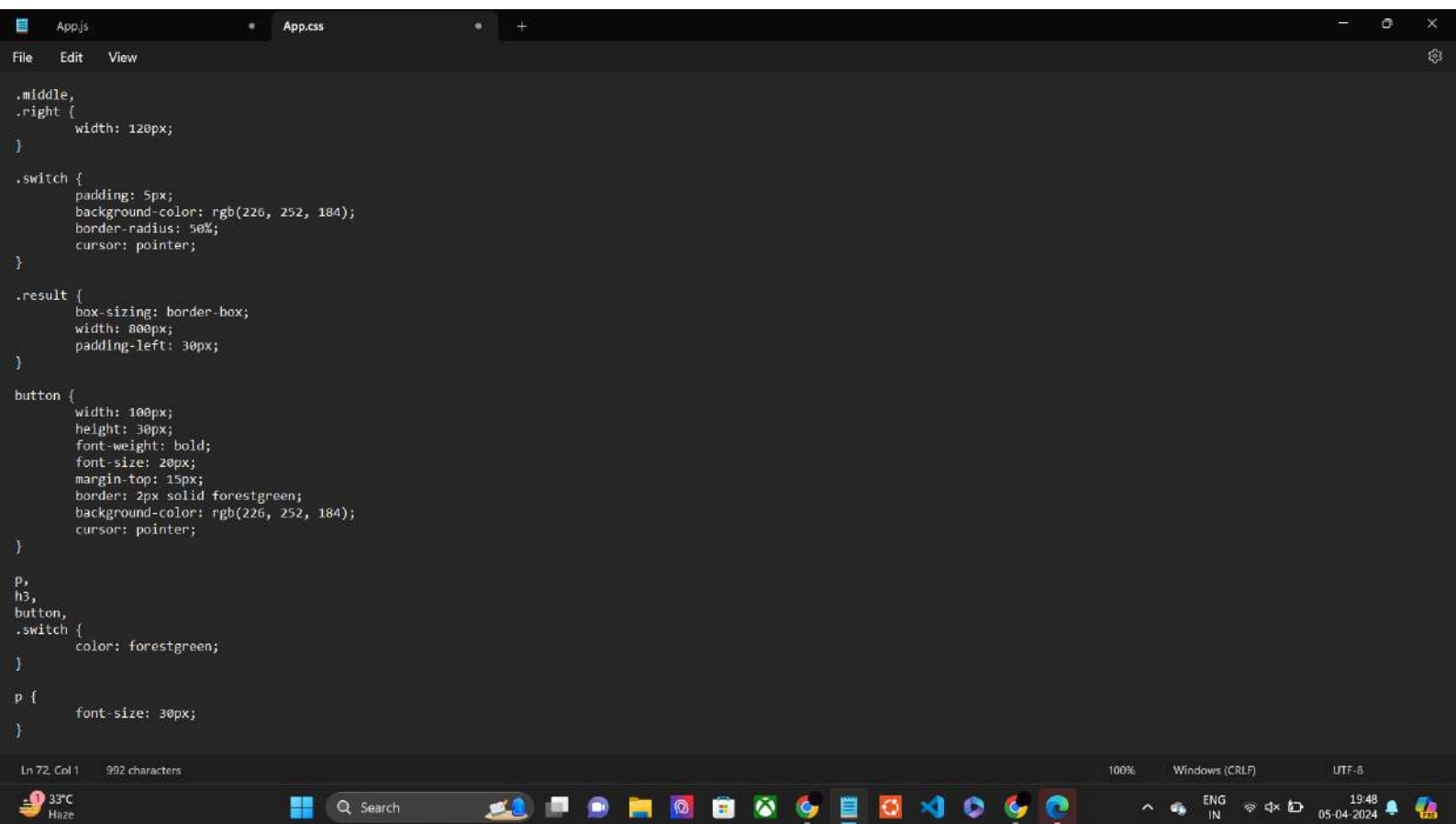
// Function to switch between two currency
function flip() {
  var temp = from;
  setFrom(to);
  setTo(temp);
}

return (
  <div className="App">
    <div className="heading">
      <h1>Currency converter</h1>
    </div>
    <div className="container">
      <div className="left">
        <h3>Amount</h3>
        <input type="text"
          placeholder="Enter the amount"
          onChange={(e) => setInput(e.target.value)} />
      </div>
      <div className="middle">
        <h3>From</h3>
        <Dropdown options={options}
          onChange={(e) => { setFrom(e.value) }}
          value={from} placeholder="From" />
      </div>
      <div className="switch">
        <HiSwitchHorizontal size="30px"
          onClick={() => { flip() }} />
      </div>
      <div className="right">
        <h3>To</h3>
      </div>
    </div>
  </div>
);
```

Ln 88, Col 1 2,175 characters 100% Windows (CRLF) UTF-8

33°C Haze Search 19:48 05-04-2024





Currency converter

Amount

From

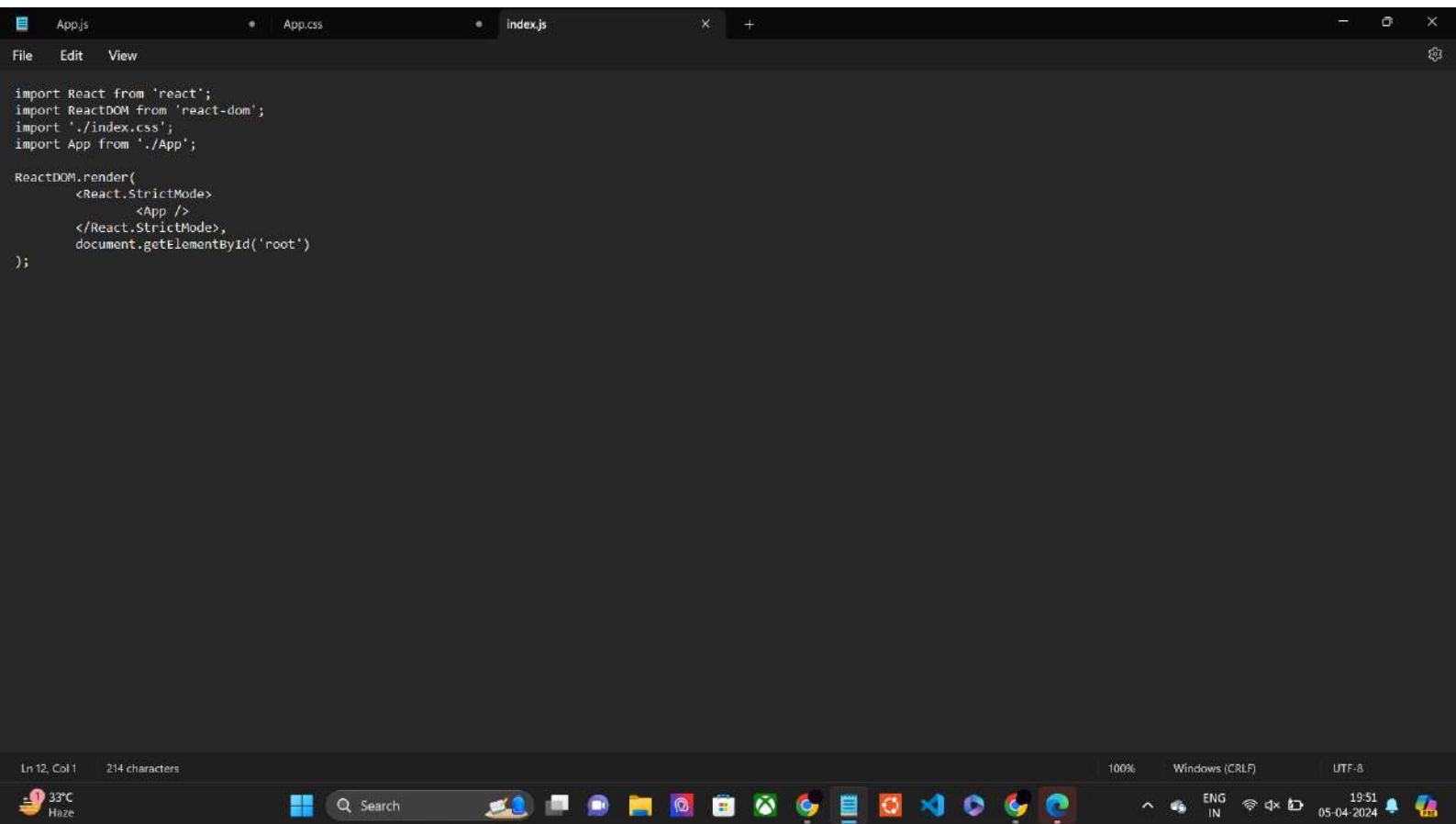


To

Convert

Converted Amount:

1 inr = 2.17 djf



App.jsApp.cssindex.jsstopwatch.jsx

FileEditView

```
import React, { useState } from "react";
import "./StopWatch.css";
import Timer from "../Timer/Timer";
import ControlButtons from "../ControlButtons/ControlButtons";

function Stopwatch() {
  const [isActive, setIsActive] = useState(false);
  const [isPaused, setIsPaused] = useState(true);
  const [time, setTime] = useState(0);

  React.useEffect(() => {
    let interval = null;

    if (isActive && isPaused === false) {
      interval = setInterval(() => {
        setTime((time) => time + 10);
      }, 10);
    } else {
      clearInterval(interval);
    }
    return () => {
      clearInterval(interval);
    };
  }, [isActive, isPaused]);

  const handleStart = () => {
    setIsActive(true);
    setIsPaused(false);
  };

  const handlePauseResume = () => {
    setIsPaused(!isPaused);
  };

  const handleReset = () => {
    setIsActive(false);
    setTime(0);
  };

  return (
```

Ln 55, Col 11,111 characters100%Windows (CRLF)UTF-8

33°C
Haze

Search

ENG
IN

19:51
05-04-2024

App.jsApp.cssindex.jsstopwatch.jsx

FileEditView

```
        interval = setInterval(() => {
            setTime((time) => time + 10);
        }, 10);
    } else {
        clearInterval(interval);
    }
    return () => {
        clearInterval(interval);
    };
}, [isActive, isPaused]);

const handleStart = () => {
    setIsActive(true);
    setIsPaused(false);
};

const handlePauseResume = () => {
    setIsPaused(!isPaused);
};

const handleReset = () => {
    setIsActive(false);
    setTime(0);
};

return (
    <div className="stop-watch">
        <Timer time={time} />
        <controlButtons
            active={isActive}
            isPaused={isPaused}
            handleStart={handleStart}
            handlePauseResume={handlePauseResume}
            handleReset={handleReset}
        />
    </div>
);
}

export default Stopwatch;
```

Ln 55, Col 11,111 characters100%Windows (CRLF)UTF-8

33°C HazeSearch19:51 05-04-2024

00:02.73

Reset

Resume