



Mini Project Report - 09

Master of Computer Application – General

Semester – I

Sub: FRONT_END FRAMEWORKS & TECHNOLOGIES

Topic: Simple_Intrest Calculator using React

By

Name: SUHAS D J

Reg no.: PROV/ASAC/MCA/25/7/050

Faculty Name: VEERA RAGHAV K

Faculty Signature: _____

Department of Computer Application

Alliance University

Chandapura - Anekal Main Road, Anekal

Bengaluru - 562 106

October 2025

LIST OF TABLES

FIG NO	NAME OF TABLE	PG NO
1	INTRODUCTION	1
2	INPUT CODE	2-7
3	OUTPUT CODE	8-9
4	CONCLUSION	10

INTRODUCTION

The Simple Interest Calculator project is a React-based web application developed to demonstrate the use of state management and event handling in modern front-end development. The main objective of this project is to compute the simple interest based on the user's input for the principal amount, rate of interest, and time period. It helps users instantly find out the interest value without manual calculations, offering both accuracy and simplicity.

The application uses the React useState Hook to store and update input values dynamically as the user types. When the "Calculate Interest" button is clicked, the simple interest is calculated using the standard formula:

$$\text{Simple Interest} = \frac{(\text{Principal} \times \text{Rate} \times \text{Time})}{100}$$

The result is then displayed immediately on the screen without any page reloads, demonstrating React's powerful re-rendering mechanism. The project's design is enhanced using CSS styling, which provides a user-friendly interface with a responsive layout, smooth hover effects, and color gradients.

This project not only improves understanding of fundamental React concepts such as components, hooks, and JSX but also helps in applying these concepts to solve real-world computational problems through an attractive, interactive user interface.

INPUT CODE

REACT CODE

```
import React, { useState } from "react";
import "../src/coloring.css";

function App() {
  const [price, setPrice] = useState(0);
  const [rate, setRate] = useState(0);
  const [time, setTime] = useState(0);
  const [interest, setInterest] = useState(0);

  const simple_interest = () => {
    const interest = (price * rate * time) / 100;
    setInterest(interest);
  };

  return (
    <div className="app">
      <div className="calculator">
        <h1 className="title"> Simple Interest Calculator</h1>

        <div className="input">
          <label>Principal Amount:</label>
          <input
            type="number"
```

```

        placeholder="Enter principal amount"
        value={price}
        onChange={(e) => setPrice(e.target.value)}
    />
</div>

<div className="input">
    <label>Rate of Interest :</label>
    <input
        type="number"
        placeholder="Enter rate of interest"
        value={rate}
        onChange={(e) => setRate(e.target.value)}
    />
</div>

<div className="input">
    <label>Time Period (in years):</label>
    <input
        type="number"
        placeholder="Enter time period"
        value={time}
        onChange={(e) => setTime(e.target.value)}
    />
</div>

<button className="calculate" onClick={simple_interest}>

```

```

        Calculate Interest
    </button>

    <p className="result">
        Simple Interest: <span>{interest}</span>
    </p>
</div>
</div>
);
}

export default App;

```

CSS CODE

```

/* --- Global Styles --- */
body {
    font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto,
    Helvetica,
        Arial, sans-serif;
    background-color: #a7d9f7; /* Light blue from your image */
    display: flex;
    justify-content: center;
    align-items: center;
    min-height: 100vh;
    margin: 0;
    overflow: hidden; /* Prevent scrollbars if content slightly exceeds viewport */
}

/* --- Calculator Container --- */
.calculator-container {
    max-width: 450px;
    width: 100%;
    margin: 20px;
    padding: 30px;

```

```

background-color: #f7a7a7; /* Coral/pinkish from your image */
border-radius: 12px;
box-shadow: 0 6px 20px rgba(0, 0, 0, 0.15); /* Slightly stronger shadow */
}

h1 {
text-align: center;
color: #551a1a; /* Darker red/brown for contrast on light pink */
font-size: 1.8rem;
margin-top: 0;
margin-bottom: 30px;
}

/* --- Input Groups --- */
.input-group {
margin-bottom: 20px;
}

.input-group label {
display: block;
font-weight: 600;
color: #551a1a; /* Darker red/brown for contrast */
margin-bottom: 8px;
text-align: center; /* <-- Added */
}

.input-group input[type='number'] {
width: 100%;
padding: 12px 15px;
font-size: 1rem;
border: 1px solid #d46a6a; /* Reddish border */
border-radius: 8px;
box-sizing: border-box;
background-color: #fcebeb; /* Very light pink background for input */
color: #551a1a; /* Dark text in input */
transition: border-color 0.3s ease, box-shadow 0.3s ease;
text-align: center; /* <-- Added */
}

/* Placeholder color */
.input-group input[type='number']::placeholder {
color: #a05252;
text-align: center; /* <-- Added */
}

```

```

/* Style for when the input is focused */
.input-group input[type='number']:focus {
  outline: none;
  border-color: #8c2a2a; /* Darker reddish border on focus */
  box-shadow: 0 0 5px rgba(140, 42, 42, 0.4);
}

/* --- Button --- */
button {
  width: 100%;
  padding: 15px;
  font-size: 1.1rem;
  font-weight: bold;
  color: #ffffff; /* White text for the button */
  /* Gradient background from your image */
  background: linear-gradient(
    to right,
    #00bcd4,
    #4ed1e0
  ); /* Light blue to turquoise gradient */
  border: 1px solid #0097a7; /* Darker turquoise border for definition */
  border-radius: 8px;
  cursor: pointer;
  transition: background 0.3s ease, transform 0.2s ease, box-shadow 0.3s ease;
  margin-top: 10px;
  box-shadow: 0 4px 10px rgba(0, 188, 212, 0.3); /* Soft shadow for the button */
}

button:hover {
  background: linear-gradient(
    to right,
    #0097a7,
    #3bc0cb
  ); /* Slightly darker gradient on hover */
  transform: translateY(-2px); /* Slight lift */
  box-shadow: 0 6px 15px rgba(0, 188, 212, 0.4); /* Enhanced shadow on hover */
}

/* --- Result Paragraph --- */
.result {
  font-size: 1.75rem;

```

```

font-weight: bold;
text-align: center;
color: #551a1a; /* Darker red/brown for contrast */
margin-top: 30px;
background-color: #fcebeb; /* Very light pink background */
padding: 15px;
border-radius: 8px;
border: 1px solid #d46a6a; /* Reddish border */
}

#i1, #i2, #i3 {
    margin-left: 70px;
}

#l1 {
    margin-left: 20px;
}
#l2 {
    margin-left: 84px;
}
#l3 {
    margin-left: 5px;
}

#p {
    margin-left: 180px;
}

```

OUTPUT

Simple Interest Calculator using React

Price Amount:
Rate:
Time in Months:

Calculate Simple Interest

Simple Interest is: 0

Simple Interest Calculator using React

Price Amount:

Rate:

Time in Months:

Calculate Simple Interest

Simple Interest is: 24000

CONCLUSION

In conclusion, the **Simple Interest Calculator** project successfully showcases the implementation of **React Hooks (useState)**, **event handling**, and **dynamic rendering** in a practical web application. Through this project, we learned how React efficiently updates the user interface in real time based on user interactions. The program accurately calculates simple interest, providing a quick and convenient way for users to obtain results.

The project also highlights the importance of combining **logic and design**—while React manages the functionality and interactivity, CSS adds to the aesthetic appeal with clean styling and smooth transitions. Overall, this project strengthens the understanding of how React can be used to create interactive, data-driven applications and serves as a solid foundation for building more complex financial or calculator-based web tools in the future.