Mini Project Report - 09

Master of Computer Application – General

Semester – I

Sub: FRONT_END FRAMEWORKS & TECHNOLOGIES

Topic: Simple_Intrest Calculator using React

By
Name: MANOJ M P
Reg no.: PROV/ASAC/MCA/25/7/041

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

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:

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

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;
```

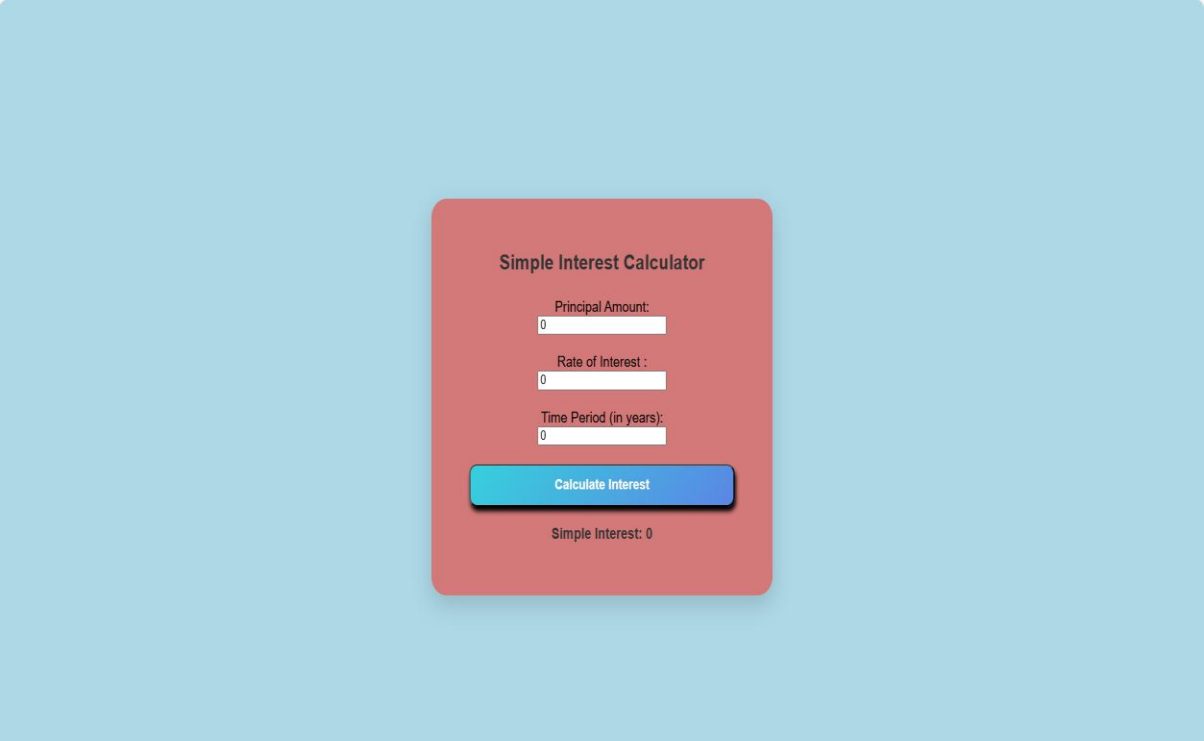
CSS CODE

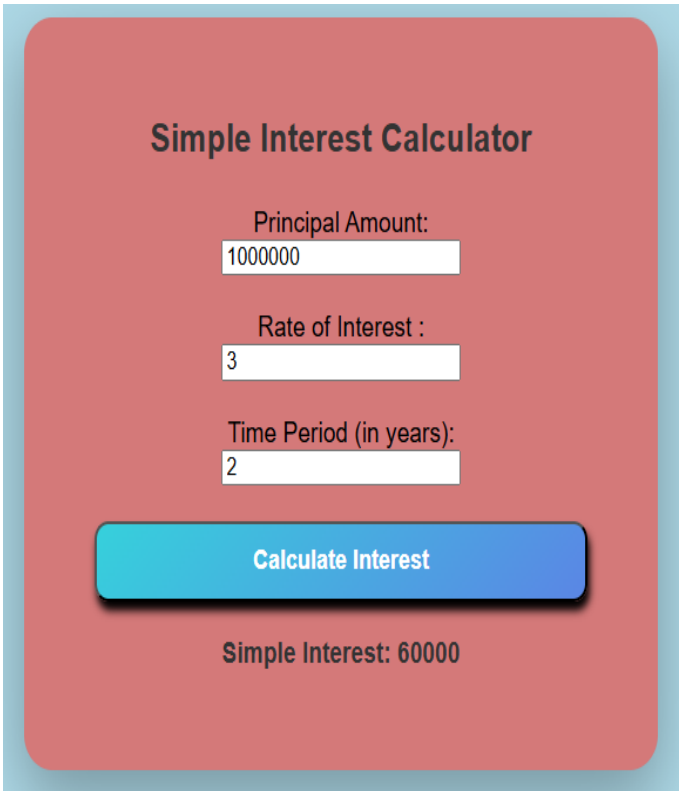
```
. .app {
  height: 100vh;
  display: flex;
  align-items: center;
  justify-content: center;
  background: #99c2ea;
  font-family: 'Poppins', sans-serif;
  box-shadow: 2px 5px 5px black;
  background-color: lightblue;
}

.calculator {
```

```
  background: #d47979;
  padding: 40px 50px;
  border-radius: 20px;
  box-shadow: 0 10px 25px rgba(0, 0, 0, 0.15);
  width: 350px;
  text-align: center;
  transition: all 0.3s ease;
}

.calculator:hover {
  box-shadow: 0 15px 40px rgba(0, 0, 0, 0.3);
  transform: translateY(-5px);
}

.title {
  font-size: 22px;
  color: #333;
  margin-bottom: 25px;
}

.input {
  display: flex;
  flex-direction: column;
  align-items: center;
  margin-bottom: 20px;
}
```

```
.calculate {
  background: linear-gradient(135deg, #36d1dc, #5b86e5);
  color: white;
  padding: 12px 18px;
  border-radius: 10px;
  font-size: 15px;
  font-weight: 600;
  cursor: pointer;
  width: 100%;
  box-shadow: 2px 5px 5px black;



}

.result {
  margin-top: 20px;
  font-size: 16px;
  font-weight: 600;
  color: #333;
}
```

OUTPUT


Simple Interest Calculator
Principal Amount:
0
Rate of Interest :
0
Time Period (in years):
0
Calculate Interest
Simple Interest: 0


Simple Interest Calculator
Principal Amount:
1000000
Rate of Interest :
3
Time Period (in years):
2
Calculate Interest
Simple Interest: 60000

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.