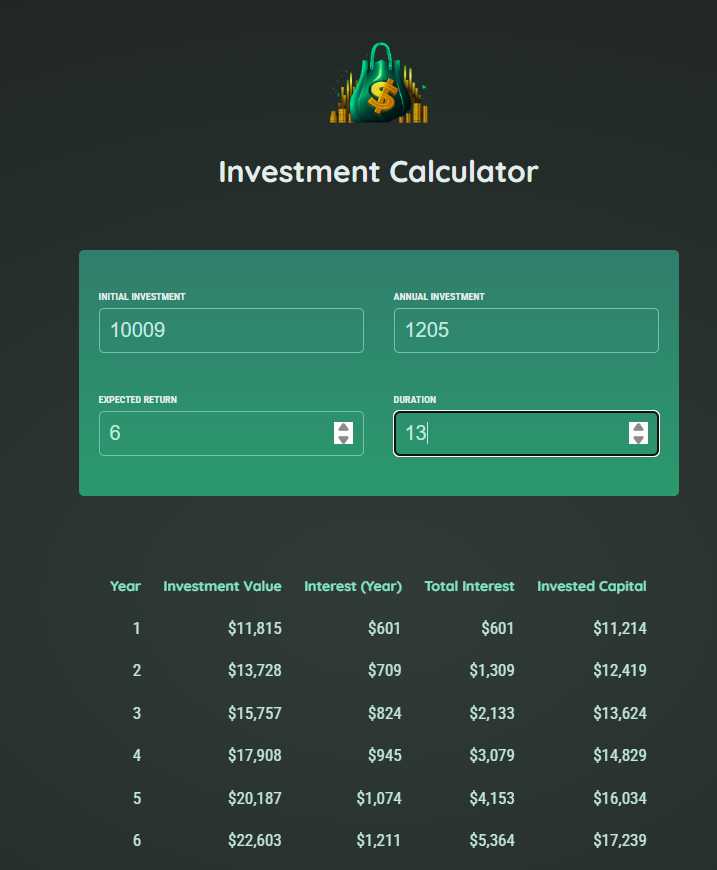
## Контролно №2 / Вариант D

Здравейте, контролното ще се проведе по следният начин:

1. В .zip файлът ще намерите първоначалният код, който трябва да довършите.
2. След като сте готови със задачата, моля, прикачете файловете (сорс кодът) в системата, архивирани в **.zip**
3. **ВАЖНО Е ДА ИЗТРИЕТЕ папка node\_modules.**

## Инвестиционен калкулатор



## Описание

Приложението стартира с предварително зададени инвестиционни параметри: първоначална инвестиция, годишна инвестиция, очаквана норма на възвръщаемост и продължителност. Потребителят може да променя тези параметри чрез компонента **UserInput**, като промените се обработват от функцията **handleChange**, която актуализира състоянието. Важно е да се провери дали продължителността на инвестицията е поне една година. Ако не е, да се покаже съобщение.

Основната функция **calculateInvestmentResults** изчислява растежа на инвестицията за определен период. За всяка година се изчислява спечелената лихва и се актуализира общата стойност на инвестицията. Данните за всяка година, включително спечелената лихва, инвестиционната стойност, годишния идентификатор и годишната инвестиция, се съхраняват в масив.

Ако въведените данни са валидни, изчислените резултати се показват в табличен формат чрез компонента **Results**. Числата да са форматирани като валута с помощта на **Intl.NumberFormat**.

## Използвайте началния проект, като основа за вашето уеб приложение (чрез команди *npm install & npm run dev*).

## Довършете компонента за показване на заглавната част на приложението, като добавите <header> елемент с текст „Инвестиционен калкулатор“.

## Довършете компонента за извличане на потребителско въвеждане на инвестиционни параметри (начална сума, процентна ставка, продължителност и др.)

### HTML & JavaScript Code

|  |
| --- |
| Част от UserInput.jsx |
| export default function UserInput({ onChange, userInput }) {  return (  <section id="user-input">  <div className="input-group">  <p>  <label>Initial Investment</label>  <input  type="number"  required  value={userInput.initialInvestment}  onChange={(event) =>  onChange('initialInvestment', event.target.value)  }  />  </p>  <p>  <label>Annual Investment</label>  <input  // userInput.annualInvestment  />  </p>  </div>  <div className="input-group">  <p>  <label>Expected Return</label>  <input  // userInput.expectedReturn  />  </p>  <p>  <label>Duration</label>  <input  // userInput.duration  />  </p>  </div>  </section>  );  } |

## Създайте функции за извличане на въведените инвестиционни параметри.

|  |
| --- |
| Част от App.jsx |
| import { useState } from 'react';  import Header from './components/Header.jsx';  import UserInput from './components/UserInput.jsx';  import Results from './components/Results.jsx';  function App() {  const [userInput, setUserInput] = useState({  initialInvestment: 10000,  annualInvestment: 1200,  expectedReturn: 6,  duration: 10,  });  const inputIsValid = userInput.duration >= 1;  function handleChange(inputIdentifier, newValue) {  setUserInput((prevUserInput) => {  return {  ...prevUserInput,  [inputIdentifier]: +newValue,  }; |

## Използвайте предоставената функция utility function от началния проект за изчисляване на резултатите от инвестицията, базирайки се на въведените потребителски данни.

|  |
| --- |
| Част от investment.js |
| export function calculateInvestmentResults({  initialInvestment,  annualInvestment,  expectedReturn,  duration,  }) {  const annualData = [];  let investmentValue = initialInvestment;  for (let i = 0; i < duration; i++) {  const interestEarnedInYear = investmentValue \* (expectedReturn / 100);  investmentValue += interestEarnedInYear + annualInvestment;  annualData.push({  year: i + 1, // year identifier  interest: interestEarnedInYear, // the amount of interest earned in this year  valueEndOfYear: investmentValue, // investment value at end of year  annualInvestment: annualInvestment, // investment added in this year  });  } |

## След изчисляването на резултатите, покажете ги в HTML таблица, използвайки елементи <table>, <thead>, <tbody>, <tr>, <th>, <td>.

|  |
| --- |
| Част от Results.jsx |
| import { calculateInvestmentResults, formatter } from '../util/investment.js';  export default function Results({ input }) {  const resultsData = calculateInvestmentResults(input);  const initialInvestment =  resultsData[0].valueEndOfYear -  resultsData[0].interest -  resultsData[0].annualInvestment;  return (  <table id="result">  <thead>  <tr>  <th>Year</th>  <th>Investment Value</th>  <th>Interest (Year)</th>  <th>Total Interest</th>  <th>Invested Capital</th>  </tr>  </thead> |

## Създайте условие за показване на информационно съобщение, ако е въведена невалидна продължителност (например, по-малко от 1 месец)

## Свържете всички описани компоненти