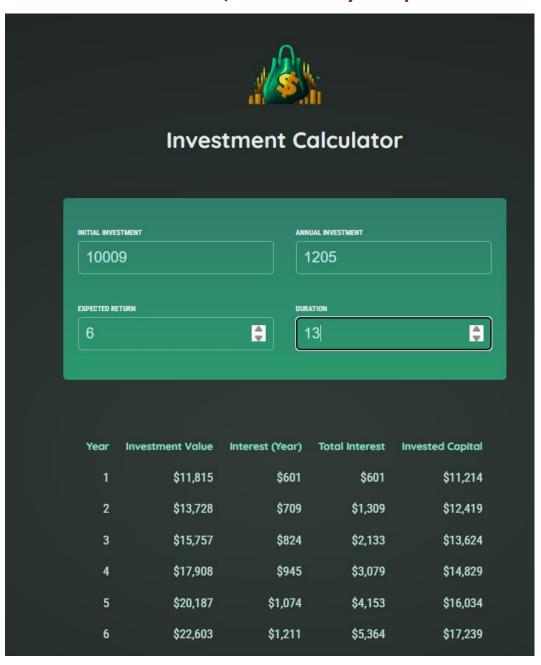
# Контролно №2 / Вариант D

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



#### 1. 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**.

- 2. Използвайте началния проект, като основа за вашето уеб приложение (чрез команди *npm install & npm run dev*).
- 3. Довършете компонента за показване на заглавната част на приложението, като добавите <header> елемент с текст "Инвестиционен калкулатор".
- 4. Довършете компонента за извличане на потребителско въвеждане на инвестиционни параметри (начална сума, процентна ставка, продължителност и др.)

## **HTML & JavaScript Code**

```
>
         <label>Annual Investment</label>
         <input
         // userInput.annualInvestment
         />
       </div>
      <div className="input-group">
         <label>Expected Return</label>
         <input
         // userInput.expectedReturn
         />
       >
         <label>Duration</label>
         <input
// userInput.duration
         />
       </div>
   </section>
  );
}
```

# 5. Създайте функции за извличане на въведените инвестиционни параметри.

```
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,
};
```

6. 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++) {</pre>
    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
    });
  }
```

7. След изчисляването на резултатите, покажете ги в HTML таблица, използвайки елементи , <thead>, , >, , .

- 8. Създайте условие за показване на информационно съобщение, ако е въведена невалидна продължителност (например, помалко от 1 месец)
- 9. Свържете всички описани компоненти