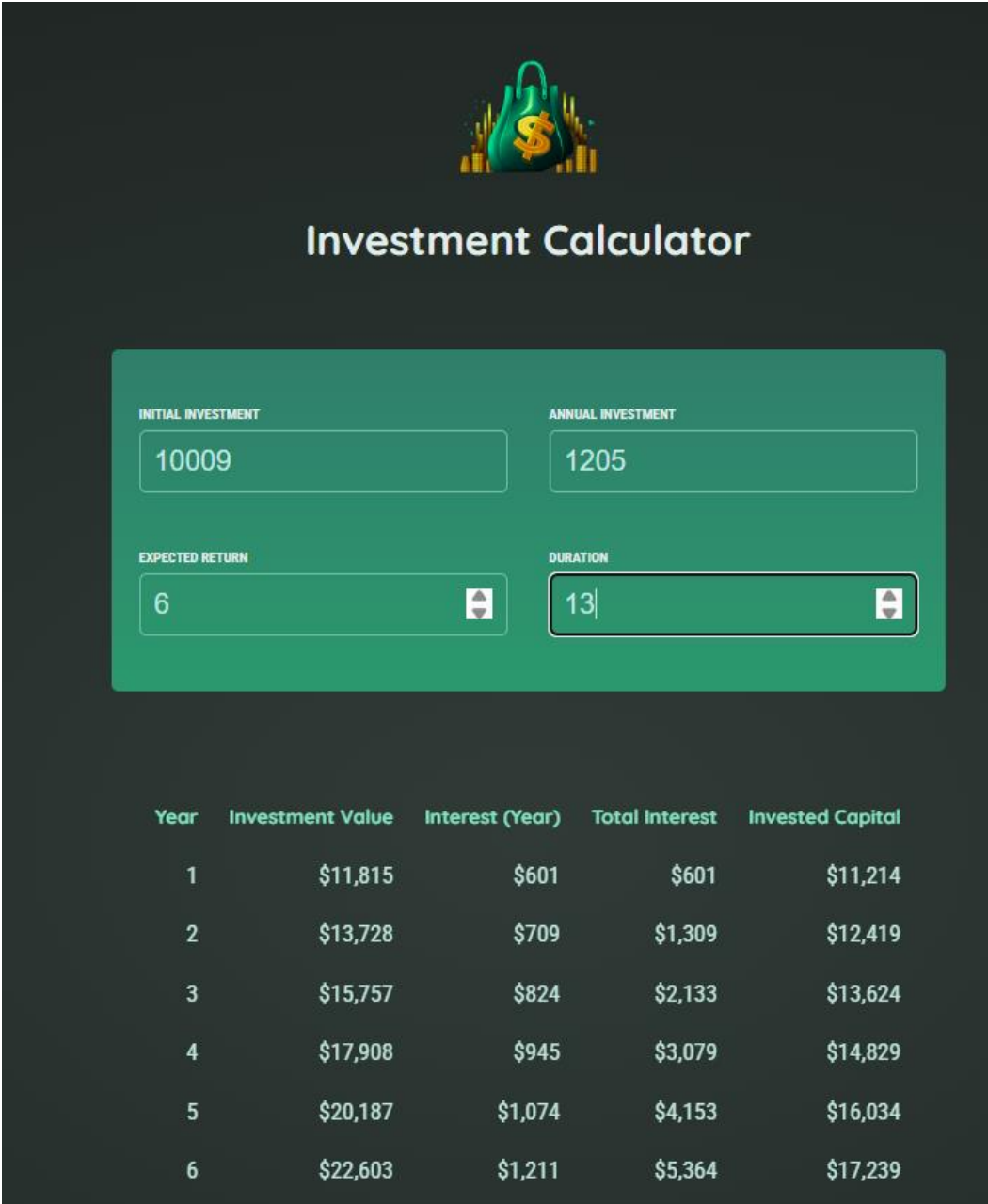


## Контролно №2 / Вариант D

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.

### 1. Създайте нов React проект чрез vite: iCalculator



The image shows a web application titled "Investment Calculator". At the top, there is a green icon of a money bag with a dollar sign. Below the title, there are four input fields arranged in a 2x2 grid. The first row contains "INITIAL INVESTMENT" with a value of 10009 and "ANNUAL INVESTMENT" with a value of 1205. The second row contains "EXPECTED RETURN" with a value of 6 and "DURATION" with a value of 13. Below the input fields, there is a table with 5 columns: "Year", "Investment Value", "Interest (Year)", "Total Interest", and "Invested Capital". The table contains 6 rows of data, showing the progression of an investment over 6 years.

Year	Investment Value	Interest (Year)	Total Interest	Invested Capital
1	\$11,815	\$601	\$601	\$11,214
2	\$13,728	\$709	\$1,309	\$12,419
3	\$15,757	\$824	\$2,133	\$13,624
4	\$17,908	\$945	\$3,079	\$14,829
5	\$20,187	\$1,074	\$4,153	\$16,034
6	\$22,603	\$1,211	\$5,364	\$17,239

2. Използвайте началния проект, като основа за вашето уеб приложение.
3. Създайте компонент за показване на заглавната част на приложението, като например `<header>` елемент с текст „Инвестиционен калкулатор“.
4. Добавете компонент за извличане на потребителско въвеждане на инвестиционни параметри (начална сума, процентна ставка, продължителност и др.)

### HTML and 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
            type="number"
            required
            value={userInput.annualInvestment}
            onChange={(event) =>
              onChange('annualInvestment', event.target.value)
            }
          />
        </p>
      </div>
      <div className="input-group">
        <p>
          <label>Expected Return</label>
          <input
```

```

        type="number"
        required
        value={userInput.expectedReturn}
        onChange={(event) =>
            onChange('expectedReturn', event.target.value)
        }
    />
</p>
<p>
    <label>Duration</label>
    <input
        type="number"
        required
        value={userInput.duration}
        onChange={(event) => onChange('duration', event.target.value)}
    />
</p>
</div>
</section>
);
}

```

## 5. 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,
            };
        });
    }
}

```

**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++) {
    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 <table>, <thead>, <tbody>, <tr>, <th>, <td>.**

Част от Results.jsx
<pre>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 (     &lt;table id="result"&gt;       &lt;thead&gt;         &lt;tr&gt;           &lt;th&gt;Year&lt;/th&gt;           &lt;th&gt;Investment Value&lt;/th&gt;           &lt;th&gt;Interest (Year)&lt;/th&gt;           &lt;th&gt;Total Interest&lt;/th&gt;           &lt;th&gt;Invested Capital&lt;/th&gt;         &lt;/tr&gt;       &lt;/thead&gt;</pre>

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

**9. Свържете всички описани компоненти**