

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



Investment Calculator

INITIAL INVESTMENT	ANNUAL INVESTMENT
<input type="text" value="10009"/>	<input type="text" value="1205"/>
EXPECTED RETURN	DURATION
<input type="text" value="6"/>	<input type="text" value="13"/>

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

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 форматираны 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

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>
      </div>
    </section>
  );
}
```

```

    </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>
);
}

```

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

```
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>
```

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

9. Свържете всички описани компоненти