Nama : Muhammad Luthfi

Prodi : MI02NR

Nim : 202202009

Kelas : Front end

1. Instalasi Kerangka UI pada Node.js

Untuk menginstal kerangka UI seperti Bootstrap

**1.npm init -y**

**2.npm install bootstrap**

**3.salin untuk di file HTML Anda, sertakan CSS dan JavaScript Bootstrap**

**<link rel="stylesheet" href="node\_modules/bootstrap/dist/css/bootstrap.min.css">**

**<script src="node\_modules/bootstrap/dist/js/bootstrap.bundle.min.js"></script>**

2. Fitur Terbaik React.js Library

**Arsitektur Berbasis Komponen**: React mendorong pembuatan komponen yang dapat digunakan kembali, sehingga kode lebih modular dan mudah dipelihara.

**Virtual DOM**: React menggunakan virtual DOM untuk mengoptimalkan rendering, hanya memperbarui bagian DOM yang perlu diubah.

**UI Deklaratif: React** memungkinkan Anda untuk mendeskripsikan bagaimana UI seharusnya terlihat berdasarkan state saat ini, dan React akan menangani pembaruan dengan efisien.

**Aliran Data Satu Arah:** Data mengalir dalam satu arah, memudahkan pemahaman bagaimana perubahan data mempengaruhi UI.

**Sintaks JSX:** React menggunakan JSX, ekstensi sintaks yang memungkinkan Anda menulis kode mirip HTML dalam JavaScript, meningkatkan keterbacaan dan kecepatan pengembangan.

**Ekosistem Kuat:** React memiliki ekosistem alat, pustaka, dan dukungan komunitas yang luas, termasuk React Router untuk routing dan Redux untuk manajemen state.

3. Perbedaan antara Hooks dan State di React.js

**State:** State adalah objek bawaan React yang menyimpan data yang dapat berubah selama masa hidup komponen. State bisa diinisialisasi di konstruktor (untuk komponen kelas) atau langsung dalam komponen (untuk komponen fungsional menggunakan hooks).

**Hooks**: Hooks adalah fungsi yang memungkinkan Anda menggunakan state dan fitur React lainnya dalam komponen fungsional. Hook useState digunakan untuk mengelola state dalam komponen fungsional, sementara hooks lain seperti useEffect mengelola efek samping, useContext menyediakan konteks, dan useReducer mengelola logika state yang kompleks

4.

function findMissingWords(input1, input2) {

const words1 = input1.split(/[\s,?]+/);

const words2 = new Set(input2.split(/[\s,?]+/));

const missingWords = words1.filter(word => !words2.has(word));

return missingWords;

}

const input1 = "Hello word, how are you today? Are you fine?";

const input2 = "word, are you Are fine?";

console.log(findMissingWords(input1, input2)); // Output: [ 'Hello', 'how', 'today?' ]

5. function findShortestSubstringAndMissingLetters(s) {

const uniqueChars = Array.from(new Set(s)).join('');

let minLength = uniqueChars.length;

let shortestSubstring = s;

for (let start = 0; start <= s.length - minLength; start++) {

for (let end = start + minLength; end <= s.length; end++) {

const substring = s.slice(start, end);

const charSet = new Set(substring);

if (charSet.size === uniqueChars.length) {

if (substring.length < shortestSubstring.length) {

shortestSubstring = substring;

}

break;

}

}

}

const charCounts = {};

for (const char of s) {

if (charCounts[char]) {

charCounts[char]++;

} else {

charCounts[char] = 1;

}

}

for (const char of shortestSubstring) {

charCounts[char]--;

}

let missingCount = 0;

for (const char in charCounts) {

if (charCounts[char] > 0) {

missingCount += charCounts[char];

}

}

return missingCount;

}

const s = "imagination";

console.log(findShortestSubstringAndMissingLetters(s)); // Output: 4