Ostad

MERN

ইন্টারভিউ কোয়েশ্চেন

1 MERN Stack কী? বা কাকে বলে?

Mern Stack হলো ৪টি জাভাস্ক্রিপ্ট-ভিত্তিক ফ্রেমওয়ার্ক নিয়ে তৈরি একটি ডেভেলপমেন্ট টেকনোলজি। এটি ব্যবহার করে বিভিন্ন ওয়েব অ্যাপ্লিকেশন বিল্ড করা হয়।

MEAN এর পূর্ণরূপ হলো, M — Mongo.DB, E — Express.js, R — React .js, N — Node.js

- এখানে, Mongo.DB হলো একটি জনপ্রিয়, ওপেন সোর্স SQL ডাটাবেজ। এটি ডেটা সঞ্চয় ও পুনরুদ্ধার করতে ব্যবহৃত হয়।
- Express.js হলো একটি ওয়েব ফ্রেমওয়ার্ক যা Node.js নিয়ে কাজ করে, এটি সার্ভার-সাইড লজিক পরিচালনা করতে ব্যবহৃত হয়।
- ReactJS বর্তমানে frontend এর সবথেকে বহুল প্রচালিতো framework যা দ্বারা আপনি Single Page Application তৈরি করা হয় ।
- সর্বশেষ, Node.js হলো একটি JavaScript রানটাইম যা ডেভেলপারদের সার্ভার-সাইডে জাভাস্ক্রিপ্ট চালানোর অনুমতি দেয়।

এটি এই ৪টি প্রযুক্তির কালেকশন, যা সাধারণত ওয়েব অ্যাপ্লিকেশন তৈরি করতে একসাথে ব্যবহার করা হয়।

2

আপনি কীভাবে একটি MERN অ্যাপ্লিকেশনে ব্যবহারকারীর অথেনটিকেশন এবং অথোরাইজেশন হ্যান্ডেল করবেন?

MERN এপ্লিকেশনে অথেনটিকেশন ও অথরাইজেশন হ্যান্ডেল করার উপায়।

- **JWT অথেনটিকেশন:** নোড জেএস দিয়ে JWT এর মাধ্যমে টোকেন বেইজড অথেনটিকেশন করা সম্ভব।
- **থার্ড পার্টি অথেনটিকেশন সিস্টেম:** চাইলে নিজেরা অথেনটিকেশন সিস্টেম ইমপ্লিমেন্ট না করে থার্ডপার্টি কোন সিস্টেম দিয়ে হ্যান্ডেল করা যেতে পারে। বাজারে অনেক গুলো সার্ভিস রয়েছে যারা এমন অথেনটিকেশন সিস্টেম প্রভাইড করে থাকে। যেমন: oauth0, firebase, cognito ইত্যাদি।
- **লাইব্রেরি ব্যবহার:** অথেনটিকেশন লাইব্রেরি ব্যবহার করে ইমপ্লিমেন্ট করা যায়। যেমন পাসপোর্ট।

3

আপনি কীভাবে একটি MERN অ্যাপ্লিকেশনের পারফরমেন্স অপ্টিমাইজ করবেন তার একটি উদাহরণ দিতে পারেন?

মার্ন স্ট্যাক এপ্লিকেশন এর পারফরম্যান্স অপ্টিমাইজড করার জন্য কীছু কাজ করা যেতে পারে। যেমন:

- **CDN ইউজ করা:** world wide সিডিএন ইউজ করলে লোড টাইম কমে আসবে।
- **ক্যাশিং করা:** যে ডেটাগুলো বেশি বেশি ব্যবহার হয় সেগুলো ক্যাশিং করে রাখলে দ্রুত রেসপন্স করা যাবে। এতে করে পারফরম্যান্স বৃদ্ধি পাবে।
- **লেজি লোডিং:** রিএক্ট এ লেজিলোডিং এড করলে যে কনটেন্ট গুলো পরে দেখবে সেগুলো পরে লোড করে ইনিশিয়াল লোডিং টাইম কমানো যেতে পারে।
- **মিনিফাই এসেটস:** কনটেন্ট গুলো মিনিফাই এবং কমপ্রেসড হলে লোড টাইম কমে আসবে।
- **ইফিসিয়েন্ট কুয়েরী:** যে কুয়েরী গুলো করা হয় ডাটাবেজে সেগুলো অপ্টিমাইজড করা প্রয়োজন।
- **ইউজ লোডব্যালেন্সার:** লোড ব্যালেন্সার ইউজ করে সার্ভারের উপর লোড কমিয়ে পারফরম্যান্স বাড়ানো যেতে পারে।

4

আপনি কীভাবে একটি প্রোডাকশন এনভাইরনমেন্টে একটি MERN অ্যাপ্লিকেশন স্থাপন করবেন?

মার্নস্ট্যাক এপ্লিকেশনে মুলত সাধারণত দুটি সার্ভার থাকে। যেখানে একটি সার্ভার এপিআই প্রভাইড করে এবং বাকী একটি ফ্রন্টএন্ড সাইট রেন্ডার করে।

>প্রথমে আমাদের কে ব্যাকএন্ড এপলিকেশন টি কোন সার্ভারে হোস্ট করতে হবে।

>এটি এপিআই এন্ডপয়েন্ট দিবে।

>ফ্রন্টএন্ডে সেই এপিআই গুলো কনজিউম করতে হবে।

>ফ্রন্টএন্ড এপ্লিকেশন টি কোন একটি সার্ভারে ডেপলয় করতে হবে।

5

আপনি কীভাবে একটি MERN অ্যাপ্লিকেশনের কম্পোনেন্টস এবং ফাংশনগুলো পরীক্ষা করবেন?

আমরা চাইলে ইউনিট টেস্ট দিয়ে এই সমস্যার সমাধান করতে পারি। ইউনিট টেস্ট হলো এমন টেস্ট যেখানে কোডের ছোট একটি মডিউল কে টেস্ট করি। এখানে এগুলো হতে পারে ফাংশন কিংবা পৃথক পৃথক কম্পোনেন্ট।

Ostad

6

আপনি কী কখনও MERN স্ট্যাক ব্যবহারের সময় কোন সমস্যার সম্মুখীন হয়েছেন? হয়ে থাকলে পরবর্তীতে আপনি কীভাবে এটি সমাধান করেছেন?

[আপনার নিজের এক্সপেরিয়েন্স সম্পর্কে আলোকপাত করুন। কী ধরনের সমস্যা আপনি আপনার নিজস্ব প্রজেক্ট তৈরি করতে গিয়ে ফেইস করেছেন।]

7

আপনি কীভাবে একটি MERN অ্যাপ্লিকেশনে স্টেট ম্যানেজ করবেন এবং এর জন্য আপনি কোন টুলসগুলি ব্যবহার করবেন?

স্টেট ম্যানেজ করার জন্য ফন্টএন্ড ই যথেষ্ট। অর্থাৎ আমাদের কে রিএক্ট এ স্টেট ম্যানেজমেন্ট করতে হবে। এর জন্য কয়েকটি সমাধান পাওয়া যায়।

১. Context API:

কনটেক্সট এপিআই ব্যবহার করে সবগুলো স্টেট একটি গ্লোবাল কনটেইনারে ট্রান্সফারের মাধ্যমে এটি পসিবল।

২. State Management Libraries:

ম্যানেজমেন্ট লাইব্রেরি যেমন রিডাক্স বা রিকয়েল ব্যবহার করা যেতে পারে। এতে করে এপ্লিকেশন এর সবগুলো স্টেট লাইব্রেরির মাধ্যমে ব্যবহার করা যাবে।

8

MERN Stack কী ? MERN-এর ফুল ফর্ম কী আপনার জানা আছে?

MERN Stack হলো একটি টেকনোলজি স্ট্যাক যেখানে;

- M দিয়ে MongoBD
- E দিয়ে Express
- R দিয়ে React
- N দিয়ে NodeJS

বুঝানো হয়।

9

আপনি কীভাবে একটি MERN অ্যাপ্লিকেশনে ব্যাকএন্ড রাউটিং পরিচালনা করবেন?

MERN স্ট্যাক এপ্লিকেশনে ব্যাকএন্ড রাউটিং করার জন্য এক্সপ্রেস এর router() মেথড কল করতে হবে। এই মেথড টি দুটি আর্গুমেন্ট নিবে। প্রথমে রাউটিং URL এবং সেকেন্ড টি একটি ফাংশন যেটি রিকুয়েষ্ট টি হ্যান্ডেল করবে।

যখন আমার রাউটিং কমপ্লিট হবে তখন সেই router টি এক্সপ্রেস এপ্লিকেশন এর সাথে মিডলওয়্যার এর মাধ্যমে বাইন্ড করে দিতে হবে app.use মেথডের মাধ্যমে।

10

React-এর কাজ কী?

React হলো একটি জাভাস্ক্রিপ্ট লাইব্রেরি যা ইউজার ইন্টারফেস (UI) তৈরি করার জন্য ব্যবহৃত হয়। এটি একটি ভার্চুয়াল DOM (Document Object Model) ব্যবহার করে UI আপডেটগুলি ম্যানেজ করে থাকে। DOM (Document Object Model) এর পার্টসগুলোকে রি-রেন্ডার করে এটি UI আপডেটগুলি সম্পাদন করে।

React একটি কম্পোনেন্ট-ভিত্তিক আর্কিটেকচার ব্যবহার করে। এর প্রতিটি উপাদান UI এর একটি নির্দিষ্ট অংশ রেন্ডার করার জন্য কার্যকরী। কোনো জটিল UI বানানোর ক্ষেত্রে এর উপাদানগুলি পুনরায় ব্যবহার এবং ক্রিয়েট করা যেতে পারে।

11

React কীভাবে কাজ করে?

রিএক্ট মুলত ভার্চুয়াল ডমের মাধ্যমে কাজ করে থাকে। এটি রিয়েল ডমের একটি লাইটওয়েট রিপ্রেজেন্টেশন। ইউজার যখন এপ্লিকেশনে কোন কীছু পরিবর্তন করে কিংবা কোন ইভেন্ট ঘটায় তখন সেটি সরাসরি ভার্চুয়াল ডমে অপারেশন করে। এবং সেখান থেকে রিকনসিলিয়েশন এর মাধ্যমে সবচেয়ে কম অপারেশন দিয়ে রিয়েল ডম আপডেট করে। এভাবে রিএক্ট কাজ করে থাকে।

12

React.js কেন এবং এর সুবিধা?

Ostad

React.js হচ্ছে জাভাস্ক্রিপ্ট এর একটি ওপেন-সোর্স লাইব্রেরী। যেটা মূলত ব্যবহার করা হয় ইউজার ইন্টারফেস এবং স্পেশালি Single Page Application (SPA) বানানোর জন্য। ওয়েব এবং মোবাইল এর ভিউ লেয়ার হ্যান্ডেলিং করাই React এর উদ্দেশ্য । React.js আমাদের UI Component গুলোকে রি-ইউজ করার সুবিধা দিয়ে থাকে। React ব্যবহার করার মূল উদ্দেশ্য, এটি অনেক সিম্পেল, ফাস্ট এবং স্কেলেবল একটি লাইব্রেরী।

React.js এর সুবিধাবলী-

- React এ রয়েছে প্রচুর পরিমানের অনলাইন রিসোর্স, বিশাল বড় ডেভেলপার কমিউনিটিসহ আকর্ষণীয় সব ডেভেলপার টুলস।
- React এর অন্যতম বৈশিষ্ট্য এর সিম্প্লিসিটি (Simplicity)। এটি Core-JavaScript এর পাশাপাশি JSX নামের একটা স্পেশাল Syntax ব্যাবহার করে। এর মাধ্যমে আমরা জাভাস্ক্রিপ্ট এর ভিতর শতভাগ HTML লিখার সুবিধা পেয়ে থাকী ।
- খুব সহজেই ডাইনামিক অ্যাপ্লিকেশন তৈরি করতে সাহায্য করে।
- একবার Component বানিয়ে একাধিকবার সেই Component গুলো রি-ইউজ করা হয়। এটা React.js এর একটি অন্যতম সুবিধাজনক ফিচার।
- React সর্বদাই একমুখী Data ফ্লো করে থাকে।

13

React-এর সীমাবদ্ধতাগুলো কী কী?

- React অ্যাপ্লিকেশনের কেবলমাত্র ভিউ লেয়ার কভার করে, তাই বাকী কাজ সম্পাদনের ক্ষেত্রে অন্যান্য প্রযুক্তির সাহায্য নিতে হয়।
- React ইনলাইন টেম্প্লেটিং এবং জেএসএক্স টেকনোলজি ব্যবহার করে থাকে, যা কীছু ডেভেলপারের কাছে গ্রহণযোগ্য নাও হতে পারে।
- নতুনদের কাছে এটি শিখা যথেষ্ট কষ্টসাধ্য।
- ক্লায়েন্ট-সাইড রেন্ডারিং-এর জন্য SEO সাপোর্ট খুবই সীমিত।
- React-এর নতুন ভার্সনে আপগ্রেড করার সময় কোডবেইস (codebase)-এ ব্রেকিং বা পরিবর্তন ঘটতে পারে।
- পুরো ফাংশনালিটির জন্য এডিশনাল সাপোর্ট দরকার, যেমন: স্টেইট ম্যানাজমেন্টের জন্য Redux.

14

React Hooks কী ?

React Hooks হলো রিঅ্যাক্ট ভার্শন 16.8-এ আপডেট একটি ফিচার। এটি একটি সাধারণ জাভাস্ক্রিপ্ট ফাংশন। এর মাধ্যমে আমরা একটি ফাংশনাল কম্পোন্যান্ট থেকে রিইউজেবল কম্পোন্যান্ট-গুলোকে আলাদা করে থাকী । ক্লাস কম্পোন্যান্ট ব্যবহার ছাড়াই Hooks লাইফসাইকেল এবং কম্পোন্যান্ট -এর স্টেট ম্যানেজ করে থাকে। বিভিন্ন ধরনের Hooks-গুলো হলো useState, useEffect, useContext এবং useReducer।

15

React-এ Super(props) এর মূল উদ্দেশ্য সম্পর্কে আপনার ধারণা দিন।

মুলত এটি একটি ক্লাস বেইজড কম্পোনেন্ট এর ইমপ্লিমেন্টেশন। যখন কোন একটি ক্লাস অন্য একটি ক্লাসকে এক্সটেন্ড করে তখন constructor এর ভিতর এই মেথড টি কল করতে হয়। যাতে এক্সটেন্ড কৃত ক্লাসে প্যারেন্ট ক্লাসের সব মেথড এবং প্রপস এভেইলএভল হয়।

16

Hooks কী ? Render Props এবং Higher-order-components (HOC)-কে রিপ্লেস করতে পারে? আপনার মতামত কী ?

হুক হলো এমন একটি স্টেটফুল ফাংশন যা শুধুমাত্র ফাংশনাল কম্পোনেন্টে ব্যবহার করা যায়। এবং এটি রিএক্ট ভার্শন ১৬.৮ থেকে সূচনা হয়। যে ফাংশন টির নাম শুরু হয় use দিয়ে। মুলত দুই ধরনের হুক আমরা ব্যবহার করি। বিল্ট-ইন হুক যেমন useState, useEffect, useReducer, useContext, useMomo, useCallback, useLayoutEffect ইত্যাদি। আর এই হুক গুলো ব্যবহার করে আমরা কাষ্টম হুক-ও তৈরি করতে পারি।

HOC কে রেন্ডার প্রপস দিয়ে রিপ্লেস করা সম্ভব।

17 React Hooks-এর সুবিধাগুলো কী কী?

React Hooks এর বিভিন্ন সুবিধাগুলোর মধ্যে অন্যতম:

- কম্পোনেন্টগুলো লিখার পদ্ধতিতে রেভ্যুলেশনাইজ করে।
- তুলনামূলক সংক্ষিপ্ত এবং পরিষ্কার কোড লিখতে সহায়ক।
- কোড পড়তে সহজসাধ্য করে তোলে।
- কোডকে পুনরায় ব্যবহারযোগ্য করে তোলে।
- ডিবাগিংগ সহ সহজেই রান বা পরীক্ষা করার সুবিধা দেয়।
- হুকগুলো ফাংশনাল কম্পোনেন্টকে স্টেট এবং লাইফসাইকেল মেথোডের অনুমতি দেয়, যা পূর্বে কেবলমাত্র ক্লাস কম্পোনেন্টের মধ্যে উপস্থিত ছিল।

18 React এর ক্ষেত্রে Props-এর অর্থ কী?

Props মূলত properties এর সংক্ষিপ্ত রুপ। Props একটি React কম্পোন্যান্ট থেকে আরেকটিতে ডেটা প্রেরণ করার জন্যে ব্যবহৃত হয়। একটি Parent কম্পোন্যান্ট থেকে Child কম্পোন্যান্টে Props স্থানান্তরিত হয়। একটি কম্পোন্যান্ট তার ফাংশন আর্গুমেন্ট হিসাবে props গ্রহণ করার পাশাপাশি dynamic কন্টেন্ট রেন্ডারেও এর ব্যবহার করতে পারে। Prop গুলো শুধুমাত্র রিডেবল হওয়ায় child কম্পোন্যান্ট দ্বারা মডিফাইড হতে পারে না।

19

Props কী এবং React-এ আপনি এটি কীভাবে ব্যবহার করেত পারেন?

প্রপস হলো এমন একটি ইমিউটিবল মেমরি যেটি একটি কম্পোনেন্ট থেকে আরেকটি কম্পোনেন্ট এ পাস করা হয়ে থাকে। যেমন আমার দুটি কম্পোনেন্ট রয়েছে যেখানে একটি স্টেট করেছে কাউন্ট। এবং সেটি কোন একটি ক্লিকের মাধ্যমে এক এক করে বৃদ্ধি পেতে থাকে। সেই স্টেট টি যদি আমার চাইল্ড কম্পোনেন্টে দেখাতে চাই তাহলে সেটি প্রপস এর ভিতর দিয়ে চাইল্ড কম্পোনেন্টে পাস করিয়ে আমাকে দেখাতে হবে।

20

ReactJS-এ আপনি কী কী উপায়ে প্রপস-এ ভ্যালিডেশন এপ্লাই করতে পারেন?

আমরা propTyoes এর মাধ্যমে প্রপস ভ্যালিডেশন করতে পারি। অথবা আমরা যদি প্রজেক্টে টাইপস্ক্রিপ্ট ব্যবহার করি তাহলে ইন্টারফেস বা টাইপ দিয়েও ভ্যালিডেশন করতে পারি।

21

Prop Drilling কী ? এটি কীভাবে পরিহার করা যায় এ সম্পর্কে বলুন।

প্রপস ড্রিলিং বলতে বুঝায় একটি স্টেট মাল্টিপল কম্পোনেন্ট এর ভিতর দিয়ে পাস করিয়ে নেস্টেড এবং ডিপ নেস্টেড কম্পোনেন্টে ব্যবহার করা। যখন কোন একটি ডেটা ডিপলি নেস্টেড চাইল্ড কম্পোনেন্ট এ ব্যবহার করার প্রয়োজন পরে তখন প্রপস ড্রিলিং করা হয়।

কয়েকটি উপায়ে এটি পরিহার করা যেতে পারে।

১. **Context API**: কনটেক্সট এপিআই ব্যবহার করে সবগুলো স্টেট একটি গ্লোবাল কনটেইনারে ট্রান্সফারের মাধ্যমে এটি পরি করা পসিবল।

২. **State Management Libraries:** স্টেট? ম্যানেজমেন্ট লাইব্রেরি যেমন রিডাক্স বা রিকয়েল ব্যবহার করা যেতে পারে। এতে করে এপ্লিকেশন এর সবগুলো স্টেট লাইব্রেরির মাধ্যমে ব্যবহার করা যাবে।

৩. **Higher Order Component:** হায়ার ওর্ডার কম্পোনেন্ট এর মাধ্যমে এই প্রপস ড্রিলিং পরিহার করা পসিবল।

৪. **রেন্ডার প্রপস** বা **useReducer** বা **useContext** এর মাধ্যমেও সমাধান সম্ভব।

22

ReactJS-এর কয়েকটি সুবিধা সম্পর্কে বলুন।

রিএক্ট জেএস ব্যবহারের অনেকগুলো সুবিধা রয়েছে। যেমন

>**পারফরম্যান্স**: রিএক্ট জেএস অনেক বেশি পারফরম্যান্ট। এটি ভার্চুয়াল ডম ব্যবহার করে রিয়েল ডমের অপারেশন কমিয়ে নিয়ে আসে।

>রিইউজিবল কম্পোনেন্ট: রিএক্ট রিইউজিবল কম্পোনেন্ট সাপোর্ট করে। ফলে আমরা একটি কম্পোনেন্ট ডিজাইন করে সেটি ই মাল্টিপল স্থানে ইউজ করতে পারি।

>ফ্ল্যাক্সিবল: রিএক্ট হলো ফ্ল্যাক্সিবল লাইব্রেরি। ছোট এপ্লিকেশন থেকে শুরু করে লার্জ এন্টারপ্রাইজ এপ্লিকেশন পর্যন্ত সব ধরনের এপ্লিকেশনে এই লাইব্রেরি ব্যবহার করা যায়।

>লার্জ কমিউনিটি: রিএক্ট এর লার্জ কমিউনিটি রয়েছে। ফলে এখান থেকে একটা বেনিফিট পাওয়া যেতে পারে। যদি কোন প্রবলেম ফেইস করে তাহলে দ্রুত তার সলিউশন কমিউনিটি থেকে নেওয়া যেতে পারে।

23 React ফাংশনের সীমাবদ্ধতাগুলো কী কী বলে আপনি মনে করেন?

রিএক্ট ১৬.৮ ইন্ট্রডিউস হওয়ার পর থেকে ফাংশনের সীমাবদ্ধতা চলে গিয়েছে। কারণ এখানে ফাংশনাল কম্পোনেন্টে হুক এব ব্যবহার এসেছে। এর আগে আমরা ফাংশনাল কম্পোনেন্টে কোন স্টেট ব্যবহার করতে পারতাম না। ফলে ফাংশন কম্পোনেন্ট শুধুমাত্র রিপ্রেজেন্টেশনাল কম্পোনেন্ট ছিলো। কী স্তু যখন ১৬.৮ ভার্সন রিলিজ হলো তখন এটি সবধরনের কম্পোনেন্ট হিসেবে ব্যবহার করা শুরু হলো। বর্তমানে ক্লাস কম্পোনেন্ট এর চেয়ে ফাংশনাল কম্পোনেন্ট লিখার প্রবনতা বেড়েছে।

24 আপনার জানা মতে React-এ useRef এবং createRef এর মধ্যকার পার্থক্যগুলো কী কী ?

createRef মুলত ক্লাস কম্পোনেন্ট এর জন্য তৈরী করা হয়েছে। যেখানে useRef শুধুমাত্র ফাংশনাল কম্পোনেন্ট এ ব্যবহার করা যায়।

cresteRef ক্লাসের ইন্সট্যান্স এর প্রপার্টি হিসেবে ডিফাইন করা হয়ে থাকে। আর useRef একটি আলাদা ফাংশন। useRef এ চাইলে DOM এর বহির্ভূত ভ্যালু স্টোর করা সম্ভব হয়।

25

Reacting Hooks কী এবং এর কাজ সম্পর্কে আপনি কতটুকু জানেন?

26 Super(Props)-এর উদ্দেশ্য কী ?

মুলত এটি একটি ক্লাস বেইজড কম্পোনেন্ট এর ইমপ্লিমেন্টেশন। যখন কোন একটি ক্লাস অন্য একটি ক্লাসকে এক্সটেন্ড করে তখন constructor এর ভিতর এই মেথড টি কল করতে হয়। যাতে এক্সটেন্ড কৃত ক্লাসে প্যারেন্ট ক্লাসের সব মেথড এবং প্রপস এভেইলএভল হয়।

27

React List-এ Keys প্রয়োজনীয় কেনো? এর সুবিধাগুলো কী কী?

লিস্টে key এট্রিবিউট ব্যবহার করা গুরুত্বপূর্ণ। কেননা আমরা যদি এটি ব্যবহার না করি তাহলে লিস্ট আপডেটের ক্ষেত্রে রিএক্ট এর ডিফল্ট Reconciliation অ্যালগরিদম রান হবে এবং key এট্রিবিউট না থাকলে রিএক্ট কম্পেয়ার করতে না পেরে সম্পুর্ণ লিস্টটিকেই রিরেন্ডার করবে ফলে এপ্লিকেশন স্লো পারফর্ম করবে।

28

React-এর ক্ষেত্রে পিউর-কম্পোনেন্টস কী ?

রিএক্টএ pure components হলো এমন টাইপ কম্পোনেন্ট যেগুলো শুধুমাত্র প্রপস এবং স্টেট এর ডেটা পরিবর্তন হলে রিরেন্ডার করে। অন্যথায় এটি অপ্রয়োজনীয় রিরেন্ডার হতে বিরত থাকে।

স্বভাবতই কম্পোনেন্টের প্রপস এবং স্টেট আপডেট হলে কম্পোনেন্ট রি রেন্ডার হয়। এতে প্রিভিয়াস ডেটা পরিবর্তন হোক কিংবা না হোক। কী ন্তু পিউর কম্পোনেন্ট এর ক্ষেত্রে বিষয়টি ভিন্ন। এক্ষেত্রে শুধু প্রপস এবং স্টেট আপডেট ই নয় বরং পরির্বতন হলেই শুধুমাত্র কম্পোনেন্ট পুনরায় রেন্ডার হবে অন্যথায় রেন্ডার হবে না।

29 ক্লাস কম্পোনেন্ট ও ফাংশনাল কম্পোনেন্টের মধ্যের বৈসাদৃশ্যগুলো কী?

React-এ ক্লাস কম্পোনেন্টস এবং ফাংশনাল কম্পোনেন্টস-এর মধ্যে বেশ কিছু বৈসাদৃশ্য দেখা যায়, যেমন:

	ক্লাস কম্পোনেন্টস	ফাংশনাল কম্পোনেন্টস
Syntax	ক্লাস কম্পোনেন্টস ক্লাস-সিনট্যাক্স ব্যবহার করে।	ফাংশনাল কম্পোনেন্টস ফাংশন-সিনট্যাক্স ব্যবহার করে।
State	ক্লাসের কম্পোনেন্টসগুলোর স্টেট থাকে।	রিয়েক্ট হুকের আগে ফাংশনাল কম্পোনেন্টস-এর স্টেট ছিল না।
Lifecycle Methods	ক্লাস কম্পোনেন্টস-এ লাইফসাইকেল মেথড থাকে যেমন: componentDidMount এবং componentDidUpdate।	এদিকে রিঅ্যাক্ট Hook দিয়ে ফাংশনাল কম্পোনেন্ট লাইফসাইকেল মেথড ব্যবহার করে থাকে।
Performance	ফাংশনাল কম্পোনেন্টগুলো সাধারণত ক্লাসের কম্পোনেন্টস এর চেয়ে দ্রুত হয়।	তুলনামূলক সময় বেশি প্রয়োজন হয়।

30

একটি Smart Component এবং একটি Dumb Component এর মৌলিক পার্থক্যগুলো কী কী?

স্মার্ট কম্পোনেন্ট অনেক সময় কনটেইনার কম্পোনেন্ট বলা হয়। অপর দিকে ডাম্ব কম্পোনেন্ট কে রিপ্রেজেন্টেশনাল কম্পোনেন্ট বলা হয়।

স্মার্ট কম্পোনেন্ট স্টেট ধারণ করে এবং স্টেট এর উপর ডিপেন্ড করে বিভিন্ন কাজ সম্পাদনা করে থাকে। অনেক সময় এটি সার্ভার থেকে ডেটা ফেচ করে থাকে। সেটি ব্রাউজারে শো করে থাকে। অপরদিকে ডাম্ব কম্পোনেন্ট এমন কোন কাজ করে না। বরং এটি শুধুমাত্র UI রেন্ডার করার জন্য ব্যবহার করা হয়ে থাকে। এটি স্টেটলেস হয়ে থাকে। তবে এটি প্রপস কনটেইন করতে পারে।

31

Higher-Order-Component কী এবং এরা কীভাবে কাজ করে?

Higher-Order-Component (HOC) হলো React-এর একটি এডভান্সড টেকনিক যা কম্পোনেন্ট লজিক পুনরায় ব্যবহার করার জন্য ইউজ করা হয়। সুনির্দিষ্টভাবে, HOC এমন একটি ফাংশন যা একটি কম্পোনেন্ট নেয় এবং অন্য একটি কম্পোনেন্ট রিটার্ন করে।

এখানে, নতুন কম্পোনেন্টকে একটি enhanced-component বলা হয় যেটিতে অতিরিক্ত কার্যকারিতা যুক্ত করা থাকে। authorization, logging, and caching মতো ক্রস-কাটিং সমস্যাগুলোর জন্য HOCs ব্যবহার করা হয়। এটি মুল কম্পোনেন্টটিকে একটি ফাংশনে জড়ানোর মাধ্যমে কাজ করে, যা অতিরিক্ত প্রপস প্রদান করে বা মূল কম্পোনেন্টটির আচরণ পরিবর্তন করে।

32

Pure Components কী এবং এ সম্পর্কে আপনার ধারণা কতটুকু?

পিউর কম্পোনেন্ট হলো এমন কম্পোনেন্ট যা কোন স্টেট ধারণ করে না। এটি মুলত ইউআই রেন্ডার করতে ব্যবহার করা হয়। এটি প্রপস হিসেবে কীছু ডেটা ইনপুট নিতে পারে এবং সে অনুযায়ী সে UI রেন্ডার করে। কীন্তু সে নিজে কোন স্টেট ধরে রাখে না। সেটিই হলো পিউর কম্পোনেন্ট।

33 Higher-Order-Components (HOC)-এর সংক্ষিপ্ত ধারণা দিন।

Higher Order Component বা HOC হলো এমন এক ধরনের কম্পোনেন্ট যেখানে কম্পোনেন্ট টি আরেকটি কম্পোনেন্ট কে ইনপুট হিসেবে গ্রহণ করে থাকে এবং সেটি নতুন একটি কম্পোনেন্ট কে রিটার্ন করে থাকে। HOC যে কম্পোনেন্ট কে ইনপুট হিসেবে গ্রহণ করে তার ভিতর সে মডিফাই করে থাকে। যেমন চাইলে সেখানে আরো কী ছু প্রপস পাঠানো যেতে পারে। কোন একটি প্রপস কে ক্যালকুলেশন করে নতুন একটি ভ্যালু এড করা যেতে পারে। ইত্যাদি কাজ সম্পুর্ণ করে নতুন একটি কম্পোনেন্ট রিটার্ন করা হয়।

34 আপনি কী ReactJS এর ক্ষেত্রে রিকনসিলিয়েশন (Reconciliation) ব্যাখ্যা করতে পারবেন?

React js এ Reconciliation হলো এমন একটি প্রসেস যা dom ম্যানুপুলেশন এর কাজে ব্যবহার হয়. এর মাধ্যমে সিদ্ধান্ত নেয়া হয় কোনো একটি কম্পোনেন্ট আপডেট হবে নাকী হবে না। যখন একটি স্টেট চেঞ্জ হয় তখন রিএক্ট কে চিন্তা করতে হয় যে কম্পোনেন্ট আপডেট হবে নাকী হবে না. এটা রিএক্ট ভার্চুয়াল ডোমের মাধ্যমে ক্যালকুলেট করে থাকে।

যখন রিএক্ট কম্পোনেন্ট এর কোনো স্টেট চেঞ্জ হয় তখন dom এক টাইপ থেকে অন্য টাইপ এ চেঞ্জ হতে হয়। তখন রিএক্ট কম্পোনেন্ট componentWillUnmount কল হয় এবং ইনার যত এলিমেন্ট আছে সব renernder হয়। এই প্রসেস কেই মূলত react reconciliation বলা হয়।

35

লিস্টে “Key” এট্রিব্যিউট-টি ব্যবহারের গুরুত্ব কতটুকু এবং এর সুবিধাগুলো কী কী?

লিস্টে key এট্রিবিউট ব্যবহার করা গুরুত্বপূর্ণ। কেননা আমরা যদি এটি ব্যবহার না করি তাহলে লিস্ট আপডেটের ক্ষেত্রে রিএক্ট এর ডিফল্ট Reconciliation অ্যালগরিদম রান হবে। এবং key এট্রিবিউট না থাকলে রিএক্ট কম্পেয়ার করতে না পেরে সম্পুর্ণ লিস্ট টি কে ই রি রেন্ডার করবে ফলে এপ্লিকেশন স্লো পারফর্ম করবে।

36

VirtualDOM এবং ShadowDOM সম্পর্কে আপনার ধারণা কতটুকু?

ভার্চুয়াল ডম হলো রিএক্ট এর একটি ফিচার। যেখানে ভার্চুয়াল ডম রিয়েল ডমের একটি কপি রাখে এবং সেটি রিয়েল ডমের সাথে সিঙ্ক করা থাকে। যখন ইউজার ইউ আই তে কোন আপডেট করে তখন সেটি সরাসরি রিয়েল ডমে আপডেট না হয়ে প্রথমে ভার্চুয়াল ডমে হয়। এবং সেটি রিয়েল ডমের সাথে কম্পেয়ার করে যে কত কম অপারেশন এ রিয়েল ডম আপডেট করা যায়। সে হিসেবে সবচেয়ে কম অপারেশন করে শুধুমাত্র রিয়েল ডমে যতটুকু আপডেট করার প্রয়োজন ঠিক ততোটুকুই আপডেট করে থাকে।

অন্য দিকে স্যাডো ডম হলো ব্রাউজারের একটি ফিচার। যেখানে স্যাডো ডম আইসোলেটেড একটি ইনভায়রনমেন্টে রান করে থাকে। উদাহরণ স্বরুপ বলা যায়। আমরা যখন কোন একটি ভিডিও <video/> ট্যাগের মাধ্যমে ব্রাউজারে শো করাই তখন দেখা যায় যে সেখানে কী ছু কন্ট্রোলার চলে আসে ভিডিও টার উপর। সেখানে আমার ভিডিও টানতে পারি, প্লে বা বন্ধ করতে পারি। এখন আমরা যদি ডেভ টুলে গিয়ে সে কন্ট্রোলার এর মার্কআপ দেখতে চাই সেগুলো দেখতে পারবো না। কারণ এটা স্যাডো ডম দিয়ে ব্রাউজার স্বভাবতই হ্যান্ডেল করে থাকে। এখানের কোন স্টাইল সরাসরি CSS লিখে চেইঞ্জ করতে পারি না। কারণ এগুলো আইসোলেটেড এনভায়রনমেন্টে রান হয়।

37 JSX সম্পর্কে কীছু বলুন।

JSX বা JavaScript XML হলো জাভাস্ক্রিপ্ট এর একটি এক্সটেনশন যা রিএক্ট এ ব্যবহার করা হয়। JSX ব্যবহার করে ডেভেলপাররা HTML এর মতো কোড লিখতে পারে।

তবুও এটি React.createElement এর একটি সিনথেটিক সুগার। আমরা চাইলে JSX ব্যবহার না করে React.createElement ব্যবহার করেও কম্পোনেন্ট তৈরি করতে পারি।

38

ReactDOM কী ?

রিএক্টডম হলো একটি লাইব্রেরি যা রিএক্ট এর সাথে ব্যবহার করা হয়। মুলত ডম ম্যানোপুলন করার কাজে ব্যবহার হয়ে থাকে।

39

Express Js কী ?

এক্সপ্রেস হলো Node.js-এর জন্য তৈরি একটি লাইট-ওয়েটেট (Lightweight) ওয়েব ফ্রেমওয়ার্ক। এটি সিংগেল পেইজ, মাল্টিপেজ এবং হাইব্রিড ওয়েব অ্যাপ্লিকেশন তৈরি করতে ব্যবহৃত হয়। এটি নোড জেএস-এর উপরে নির্মিত একটি স্টেজ যা সার্ভার এবং রুট পরিচালনা করতে সহায়তা করে।

39 Express কীভাবে ইনস্টল করবেন?

এক্সপ্রেস হলো Node.js-এর জন্য তৈরি একটি লাইট-ওয়েটেট (Lightweight) ওয়েব ফ্রেমওয়ার্ক। Express ইনস্টলেশনের জন্য নিচের কমান্ডটি সঠিকভাবে চালাতে হবে:

```
npm install express --save
```

40

Express.js-এ Routing কী ?

একটি URL/এন্ডপয়েন্ট সার্ভার দ্বারা কীভাবে রেসপন্স বা হ্যান্ডেলড করা হচ্ছে তা বোঝার জন্য Routing ব্যবহার করা হয়। Express একটি চমৎকার উপায় প্রদান করে থাকে অ্যাপ্লিকেশন Routing হ্যান্ডেল করার জন্য। Express-এ Routing পরিচালনা করার জন্য প্রাথমিক কোডটি হলো:

```
var express = require('express')
var app = express()
// respond with "hello world" when a GET request is
made to the homepage
app.get('/', function (req, res) {
  res.send('hello world')
});
```

41

Express.js-এ Scaffolding নিয়ে আপনার ধারণা কতটুকু?

একটি এক্সপ্রেস এপ্লিকেশন স্ক্যাফোল্ড করতে হলে কীছু স্টেপ বাই স্টেপ কাজ করতে হয়। যথা:

- কম্পিউটারে নতুন একটি ফোল্ডারে npm init করতে হয়।
- তারপরের কাজ হলো express ইন্সটল করা। npm install express এর মাধ্যমে এই কাজ টি করা হয়।
- app.js নামে একটি ফাইল তৈরি করতে হবে রুট ফোল্ডারে।
- ফাইলে এক্সপ্রেসের ইন্সট্যান্স তৈরি করতে হবে।
- রাউটিং ডিক্লেয়ার করতে হবে।
- Views নামে একটি ফোল্ডার করতে হবে যেখানে HTML থাকবে।
- public নামে একটি ডিরেক্টরি তৈরি করতে হবে যেখানে স্ট্যাটিক ফাইল থাকবে।
- সর্বশেষ সবকী ছু কনফিগার ও সেটআপ করে pm2 বা nodemon দিয়ে সার্ভার স্টার্ট দিতে হবে।

42 Express-এ Middleware নিয়ে সংক্ষিপ্ত বর্ণনা দিন।

মিডলওয়্যার এমন একটি ফাংশন যেটি মুল রিকুয়েষ্ট কন্ট্রোলার ফাংশনে যাওয়ার আগে রান হয়। এবং এটি রিকুয়েষ্ট, রেসপন্স এবং নেক্সট কে প্যারামিটার আকারে গ্রহণ করে। মিডলওয়্যার কয়েকটি কাজ করতে পারে যেমন।

- কোন কোড এক্সিকী উট করতে পারে।
- রিকুয়েষ্ট এবং রেসপন্স অবজেক্ট চেইঞ্জ করতে পারে।
- রিকুয়েষ্ট রেসপন্স লাইফসাইকেল কমপ্লিট করে দিতে পারে।
- নেক্সট মিডলওয়্যার কে কল করতে পারে।
- এরর মিডলওয়্যার কে কল করতে পারে।

43

Express.js-এ রাউট হ্যান্ডলারদের কাছে কোন ফাংশন আর্গুমেন্টগুলো পাওয়া যায়?

রাউট হ্যান্ডেলার ফাংশন বা কোন মিডলওয়্যার এর ফাংশন মুলত নেক্সট নামে একটি ফাংশন কে আর্গুমেন্ট হিসেবে পেতে পারি।

Ostad

44

এক্সপ্রেস-এ ইউজার অথেনটিকেশনের উপায়গুলো বিস্তারিতভাবে বর্ণনা করুন।

কয়েকটি উপায়ে অথেনটিকেশন করা যায়।

- **সেশন বেইজড অথেনটিকেশন:** এর মাধ্যমে অথেনটিকেশন করলে সেশন কুকী র ভিতর লগইন ক্রেডেনশিয়াল থাকে। ইউজার যখন ব্রাউজার ক্লোজ করে দেয় তখন অটোমেটিক ভাবে লগআউট হয়ে যায়।
- **কুকী বেইজড অথেনটিকেশন :** এর মাধ্যমে কুকী কে লগইন ক্রেডেনশিয়াল গুলো স্টোর থাকে। এর মাধ্যমে ইউজার লম্বা সময় লগইন থাকতে পারে।
- **JWT বেইজড অথেনটিকেশন:** এটি একটি নতুন টেকনোলজি। এর মাধ্যমে অথেনটিকেশন করলে সার্ভার একটি টোকেন প্রভাইড করে থাকে। এবং টোকেন টি চাইলে ব্রাউজারের লোকাল স্টোরে রাখা যায়। প্রতিটি রিকুয়েষ্ট এর সাথে এই টোকেন টি এটাচ করে দেওয়া হয়। সার্ভারে যেটি ভ্যালিডেশন হয়ে থাকে।

45

Express-এ আপনি কীভাবে প্লেইন HTML রেন্ডার করবেন?

রাউটিং হ্যান্ডেলার ফাংশন টি দুটি প্যারামিটার নিয়ে থাকে। একটি req অন্যটি res। res এর সাথে একটি মেথড রয়েছে send। যেটি একটি স্ট্রিং আর্গুমেন্ট হিসেবে গ্রহণ করে। সেই স্ট্রিং টি যদি আমরা HTML প্রবাইড করি তাহলেই প্লেইন HTML রেন্ডার করবে।

46

Express কোন Template Engine সাপোর্ট করে?

এক্সপ্রেস একাধিক টেমপ্লেট ইঞ্জিন সাপোর্ট করে। তার ভিতর অন্যতম হলো:

EJS, PUG, Handlebars, mustache ইত্যাদি।

47

ExpressJS-এর কাজ এবং উদ্দেশ্যকে আপনি কীভাবে সংজ্ঞায়িত করবেন?

এক্সপ্রেস হলো মিনিমালিস্ট, ফাস্ট, পপুলার ওয়েব ফ্রেমওয়ার্ক। এটি ইজি টু লার্ন। ইজি টু ইমপ্লিমেন্ট। এটির সিনটেক্স গুলোও স্ট্রেইটফরওয়ার্ড। এটির অনেক গুলো বিল্ট-ইন ফিচার রয়েছে যা ডেভেলপমেন্ট কে আরো ফাস্টার করে থাকে।

এক্সপ্রেস এপ্লিকেশন খুবই ফ্ল্যাক্সিবল। এটি দিয়ে যেকোন ধরনের এপ্লিকেশন বানানো পসিবল।
এটির পারফরম্যান্স খুবই ফাস্ট। এটি নোডজেএস এর উপর বেইজ করে হওয়ায় খুবই ইফিসিয়েন্ট।
এর অনেক বড় একটি ডেভেলপার কমিউনিটি রয়েছে। হিউজ পরিমাণ রিসোর্স রয়েছে, যার কারণে এটির যেকোন প্রবলেম অতিদ্রুত সমাধান করা সম্ভব হয়।

48

MongoDB-তে Document বলতে কী বোঝায়?

মংগোডিবি তে ডকুমেন্ট হলো একটি স্ট্রাকচার্ড ডেটা যা দেখতে JSON এর মতো লাগে। SQL ডেটাবেস এ যাকে টেবিল রো বলা হয় এখানে সেটাই ডকুমেন্ট। এটি কী ভ্যালু পেয়ার কালেকশন। যেখানে কী গুলো হলো স্ট্রিং আর ভ্যালো কয়েক ধরনের ডেটা টাইপ হতে পারে।

যেমন

```
{
        "nama": "shahjalal",
        "age": 25
}
```

এটি একটি ডকুমেন্ট। যা দেখতে JSON ডেটার মতো কীন্তু এটি একটি BSON ডেটা।

49

Mongo Shell কী ?

মংগো শেল হলো একটি জাভাস্ক্রিপ্ট ইন্টাফেস যা মংগোডিবি ডেটাবেস এক্সেস এবং ডেটা ম্যানিপুলেট করার কাজে ব্যবহার করা হয়ে থাকে। মংগোশেল মংগো সার্ভারের সাথে বিল্ট-ইন থাকে। এবং চাইলে সেপারেটলি ডাউনলোড ও করা সম্ভব।

মংগোশেল ইউজ করতে হলে প্রথমে মংগো সার্ভার স্টার্ট করে তারপর

>mongo

এই কমান্ড টি দিলেই মংগো শেল ওপেন হবে।

মংগো শেল দিয়ে আমরা ডেটা ইনসার্ট করতে পারি। আপডেট করতে পারি। ডিলিট করতে পারি এমন কী রিড ও করতে পারি। জাভাস্ক্রিপ্ট দিয়ে কুয়েরী লিখে সেটা এক্সিকী উট করতে পারি।

50

MongoDB-তে আপনি কীভাবে ট্রানজেকশন লক অর্জন করতে পারেন?

ট্রানজেকশন হলো এমন একটি ফিচার যার মাধ্যমে একটি বা একাধিক অপারেশন সেট যা সিঙ্গেল লজিক্যাল ইউনিটে এক্সিকী উট হয়। মংগোডিবি ট্রানজেকশন এর ক্ষেত্রে ACID (Atomicity, Consistency, Isolation, Durability) প্রপার্টি ফলো করে থাকে।

ট্রানজেকশন দুই ধরনের হতে পারে।

অটোমেটিক ট্রানজেকশনঃ

মংগোডিবির বাই ডিফল্ট ট্রানজেকশন হলো অটোমেটিক ট্রানজেকশন। এটি ব্যবহার করা বেশ সহজ। এবং এটি আইসোলেশন গ্যারান্টি প্রভাইড করে না।

এক্সপ্লিসিট ট্রানজেকশনঃ

এক্সপ্লিসিট ট্রানজেকশন পাওয়ারফুল আইসোলেটেড এনভায়রনমেন্ট প্রভাইড করে থাকে। এবং এটি অটোমেটিক ট্রানজেকশন থেকে পাওয়ারফুল। এটি ব্যবহার করা বেশ জটিল কী স্তু এটি হাইলি ডেটা কনসিসট্রেন্সি প্রভাইড করে থাকে।

অপর দিকে লকিং ফিচার হলো এমন একটি ফিচার যাতে একাধিক ইউজার কে একই সময়ে একই ডেটা এক্সেস করতে বাঁধা প্রদান করা হয়। ডেটা করাপশন কমাতে এই নীতি অনুসরণ করা হয়।

মংগোডিবি দুই ধরনের লকিং সিস্টেম প্রভাইড করে থাকে।

শেয়ার্ড লকিংঃ

শেয়ার্ড লকিং এ মাল্টিপল ইউজার সেইম টাইমে সেম ডেটা এক্সেস করতে পারে। এটি ডেটা করাপশন প্রতিরোধে ব্যবহার করা হয়।

এক্সক্লুসিভ লকিংঃ

এখানে লক করা ডেটাগুলো কে অন্য ইউজার যদি এক্সেস করতে চায় তখন বাঁধা প্রদান করে থাকে।

51 MongoDB-তে Replication কী ?

Replication হলো ডেটা সিঙ্ক্রোনাইজ করার একটি প্রক্রিয়া। এর উদ্দেশ্য হলো রিডান্ডেসি প্রদান, অ্যাভেইলেবিলিটি এবং স্কেলিবিলিটি বৃদ্ধি করা। একাধিক MongoDB সার্ভারে একই ডেটা সেটের কপি তৈরি করার প্রক্রিয়াই হলো MongoDB-তে Replication।

একটি রেপ্লিকেটেড MongoDB সার্ভারে, একটি সার্ভার প্রাইমারি সার্ভার হিসাবে কাজ করে এবং অন্যগুলি সেকেন্ডারি সার্ভার হিসাবে কাজ করে। প্রাইমারি সার্ভার সমস্ত ডেটার কার্যাদিসমূহ গ্রহণ করে এবং সেকেন্ডারি সার্ভারগুলিতে পরিবর্তনগুলি প্রচার করে। প্রাইমারি সার্ভার কোনো কারণে বিফল হলে, সেকেন্ডারি সার্ভারগুলির মধ্যে একটি স্বয়ংক্রিয়ভাবে নতুন প্রাইমারি সার্ভার হিসাবে নির্বাচিত হয়। ব্যাকআপ, রিকভারি, লোড ব্যালেন্সিং এবং রিডেবিলিটি উন্নত করার জন্য Replication ব্যবহার করা যেতে পারে।

52

CAP থিওরেম কী ? CAP থিওরেমে MongoDB-এর অবস্থান কোথায়?

CAP থিওরেম হলো এমন একটি ডিস্ট্রিবিউটেড সিস্টেম যেখানে নিম্নোক্ত ৩টি বৈশিষ্ট্যের দুটি পাওয়া সম্ভব।

- **Consistency**:

  সবাই সেইম টাইমে সেম ডেটা দেখতে পারবে।

- **Availability:**

  সব সিস্টেম সবসময় এভেইলএভল থাকবে এবং সবগুলো রিকুয়েষ্ট এবং রেসপন্স জেনারেট করতে পারবে।

- **Partition tolerance:**

  যদিও কী ছু নোড অন্যান্য নোড থেকে আলাদা থাকে তবুও সিস্টেমের অপারেশন কন্টিনিউ থাকবে।

এখানে মংগোডিবি হলো CP সিস্টেম। ফলে এটি Consistency এবং Partition tolerance কে Availability থেকে গুরুত্ব প্রদান করে।

53

MongoDB-তে Sharding কী এবং এটি কীভাবে কাজ করে?

Sharding এমন একটি পদ্ধতি যার মাধ্যমে mongodb ডাটাবেস কে horizontally স্কেল করা হয়।ট্রেডিশনাল ভার্টিক্যাল স্কেলিং করা বা সিপিইউ র্যাম বা মেমোরি বর্ধিত করে স্কেলিং করার চেয়ে দুই বা ততোধিক মেশিন একসাথে করে MongoDB ডেটাবেস কে এগ্রিগেট করার মাধ্যমে sharding করা হয়।

Sharding করার ক্ষেত্রে বার্ডিং

54

MongoDB-তে আপনার কখন একটি ডকুমেন্ট অন্যটির মধ্যে এম্বেড করা উচিত?

Ostad

এটা মুলতঃ নির্ভর করে এপ্লিকেশন এর আর্কিটেকচার এর উপর। তারপরও কী ছু সাধারণ ব্যাপারে এটা করা যায়। যেমনঃ

- যখন ডেটা গুলো প্রায় কাছাকাছি ধরনের হয়।
- যখন প্রায় সময় সবগুলো ডেটা একসাথে এক্সেস করার প্রয়োজন হয়।
- যখন ফ্লেক্সিবল স্কিমা ডিজাইন করা হয়।
- যখন আমরা NoSQL ডেটাবেস এর সুবিধা গ্রহণ করতে চাই।
- যখন রেয়ারলি ডেটা আপডেটের প্রয়োজন হয়। সে ডেটাগুলো একসাথে রাখা যায়।

55

MongoDB কী Foreign Key constraints সমর্থন করে?

না MongoDB তে Foreign Key Constraints সাপোর্ট করে না।

56

আপনি কী MongoDB-তে JSON এর চেয়ে BSON ব্যবহারের সুবিধাগুলো ব্যখ্যা করতে পারবেন?

১. লাইটওয়েট এবং ট্রাভার্সেবল। BSON ফরমেটে অনেক বড় ধরনের ডেটাসেট খুব সহজেই স্টোর করা যায়।এবং ট্রান্সফার সুবিধা বেশি। নেটওয়ার্কে ভালোভাবে কাজ করার জন্য উত্তম।

২. ইফিসিয়েন্ট ফরমেট। BSON ডেটা ফরমেটে দ্রুত কোয়েরি রান করা যায়। দ্রুত বড় ডেটাসেট স্টোর করা যায়। দ্রুত কম্পেয়ার এবং ক্যালকুলেট করা যায়।

৩. অন্যান্য ডেটাটাইপ সাপোর্ট। BSON Bindata, maxkey, minkey, Binary Data, ObjectID, Regular expression টাইপ গুলো সাপোর্ট করে যেখানে JSON এ এগুলো সম্ভব নয়।

57

MongoDB-তে ট্রানজেকশনগুলো ইম্পলিমেন্টের উপায়গুলো কী?

ট্রানজেকশন হলো এমন একটি ফিচার যার মাধ্যমে একটি বা একাধিক অপারেশন সেট যা সিঙ্গেল লজিক্যাল ইউনিটে এক্সিকী উট হয়। মংগোডিবি ট্রানজেকশন এর ক্ষেত্রে **ACID (Atomicity, Consistency, Isolation, Durability)** প্রপার্টি ফলো করে থাকে।
ট্রানজেকশন দুই ধরনের হতে পারে।

অটোমেটিক ট্রানজেকশনঃ

মংগোডিবির বাই ডিফল্ট ট্রানজেকশন হলো অটোমেটিক ট্রানজেকশন। এটি ব্যবহার করা বেশ সহজ। এবং এটি আইসোলেশন গ্যারান্টি প্রভাইড করে না।

এক্সপ্লিসিট ট্রানজেকশনঃ

এক্সপ্লিসিট ট্রানজেকশন পাওয়ারফুল আইসোলেটেড এনভায়রনমেন্ট প্রভাইড করে থাকে। এবং এটি অটোমেটিক ট্রানজেকশন থেকে পাওয়ারফুল। এটি ব্যবহার করা বেশ জটিল কী স্তু এটি হাইলি ডেটা কনসিসট্রেন্সি প্রভাইড করে থাকে।

58

আপনি লাইক অপারেটর ব্যবহার করে কীভাবে MongoDB-কে কোয়্যারি(Query) করবেন?

আমরা সরাসরি like অপারেটর মংগোডিবিতে ব্যবহার করতে পারি না। কীন্তু চাইলে আমরা $regex অপারেটর ব্যবহার করে লাইক অপারেটর এর মতো আউটপুট পেতে পারি। যেমন:

```
db.collection.find({ field: { $regex: /pattern/}})
```

উপরের এই প্যাটার্নের মাধ্যমে আমরা ম্যাচিং করে ডেটা পেতে পারি।

59

আপনি কী MongoDB-এর Aggregation Pipeline সম্পর্কে বিস্তারিত বলতে পারবেন?

Aggregation Pipeline হলো একটি পাওয়ারফুল ফ্রেমওয়ার্ক। যার মাধ্যমে আমরা কোন ডেটাসেট কে মডিফাই করতে পারি। যেমন: ফিল্টারিং, সর্টিং, গ্রুপিং প্রজেক্টিং, ট্রান্সফর্মিং ইত্যাদি করতে পারি।

Aggregation Pipeline একধিক স্টেজ নিয়ে গঠিত। যেখানে মাল্টিপল অপারেশন সম্ভব এবং একটি অপারেশন যা আউটপুট দিবে সেটি পরবর্তী অপারেশন এর ইনপুট হিসেবে ব্যবহার হবে। Aggregation Pipeline এর অনেকগুলো মেথড রয়েছে যেমন:

$match:

কোন একটি ক্রাইটেরিয়ার মাধ্যমে ডেটা match করে সেই ডেটা গুলো রিটার্ন করে।

$project:

কোন ডেটা রাখা বা বাদ দেওয়া কিংবা কোন ফিল্ড কে রিনেম করা কোন কম্পিউটেড প্রপার্টি যুক্ত করার কাজ করা হয এই মেথড দিয়ে

$group:

এর মাধ্যমে কোন ডেটা গ্রুপ করা হয়। এবং এগ্রেগেশন অপারেশন চালানো হয় যেমন sum, average, count ইত্যাদি।

$sort:

কোন একটা ক্রাইটেরিয়ার মাধ্যমে ডেটা সাজানো হয়।

$limit $skip:

এর মাধ্যমে ডেটার সংখ্যা বলে দেওয়া হয় এবং আগের কতগুলো ডেটা স্কিপ করতে হবে সেটা বলে দেওয়া হয়।

$lookup:

এর মাধ্যমে লেফট আউটার জয়েন করা হয়। এবং অন্য কোন কালেকশনে ডেটা সার্চ করা হয়।

60

MongoDB-তে রেপ্লিকেশন প্রক্রিয়াটি যেভাবে সম্পাদিত হয় এ সম্পর্কে আপনি কতটুকু ধারণা রাখেন?

MongoDB তে সচকরাচর রেপ্লিকেশন তখন করা হয় যখন কোন রাইট বা আপডেট অপারেশন সম্পুর্ন হয়। এমনিতেও করা সম্ভব। রেপ্লিকেশন এর উদ্দেশ্য থাকে ডেটা যাতে কখনো হারিয়ে না যায়, কিংবা ডেটা যাতে মাল্টিপল সার্ভার থেকে সবসময় পাওয়া যায়।

ডেটা রেপ্লিকেশন প্রসেস সম্পুর্ন করার জন্য কী ছু বিষয় মাথায় রাখা দরকার। যেমন, কতগুলো সার্ভার রানিং, সার্ভার গুলোর লোকেশন কোথায় কিংবা কোন ধরনের সার্ভার ইত্যাদি বিষয়।
ডেটা রেপ্লিকেশন এর জন্য কয়েকটি ধাপ অতিক্রম করতে হয়।

প্রথমে কয়েকটি ইন্সট্যান্স মাল্টিপল প্রসেসে স্টার্ট করতে হবে। তারপর rs.initiate() কমান্ডের মাধ্যমে ইনিশিয়েট করতে হবে। এবং এর ভিতর আমাদের অপশন প্রভাইড করতে হবে। যেখানে থাকবে ডেটার ইনফর্মেশন এবং প্রসেস আইডি।

61

আপনি কীভাবে অন্য একটি ফিল্ডের ভ্যালু ব্যবহার করে MongoDB-এর ফিল্ড আপডেট করতে পারবেন?

$set method দিয়ে কাজটা করা সম্ভব।

যেমন:

```
db.update({
        _id: 1,
}, {
        $set: {
                email: $nam
        }
})
```

৪৯

MongoDB-তে Covered Query গুরুত্বপূর্ণ কেনো এসম্পর্কে আপনার মতামত দিন।

কয়েকটি কারণে Covered Query গুরুত্বপূর্ণ।

1. পারফরম্যান্সঃ

পারফরম্যান্স বৃদ্ধির জন্য এটি ব্যবহার করা যেতে পারে। কারণ এই কুয়েরী চলতে হলে কোন স্ক্যানিং করতে হয় না।

২. ডেটাবেস লোড রিডিউসঃ

ডেটাবেস এ লোড কমাতে কভারড কোয়েরি সুন্দর সমাধান হতে পারে। কারণ এতে কোন স্ক্যানিং অপারেশন করতে হয় না ফলে একটি টাইম কন্সিং অপারেশন থেকে রক্ষা পেতে পারি।

৩. স্কেলেবিলিটিঃ

এটা যেহেতু দ্রুত রান হয় তাই বেশি ট্রাফিক এর মাধ্যমে সুবিধা পেতে পারে।

62

MongoDB-তে Replication এর ব্যবহার সম্পর্কে আপনি কতটুকু ধারণা রাখেন?

MongoDB তে সচকরাচর রেপ্লিকেশন তখন করা হয় যখন কোন রাইট বা আপডেট অপারেশন সম্পুর্ন হয়। এমনিতেও করা সম্ভব। রেপ্লিকেশন এর উদ্দেশ্য থাকে ডেটা যাতে কখনো হারিয়ে না যায়, কিংবা ডেটা যাতে মাল্টিপল সার্ভার থেকে সবসময় পাওয়া যায়।
ডেটা রেপ্লিকেশন প্রসেস সম্পুর্ন করার জন্য কীছু বিষয় মাথায় রাখা দরকার। যেমন, কতগুলো সার্ভার রানিং, সার্ভার গুলোর লোকেশন কোথায় কিংবা কোন ধরনের সার্ভার ইত্যাদি বিষয়।

ডেটা রেপ্লিকেশন এর জন্য কয়েকটি ধাপ অতিক্রম করতে হয়।
প্রথমে কয়েকটি ইন্সট্যান্স মাল্টিপল প্রসেসে স্টার্ট করতে হবে। তারপর rs.initiate() কমান্ডের মাধ্যমে ইনিশিয়েট করতে হবে। এবং এর ভিতর আমাদের অপশন প্রভাইড করতে হবে। যেখানে থাকবে ডেটার ইনফর্মেশন এবং প্রসেস আইডি।

63

ট্রানজেকশন ও লকিং কী ? MongoDB-তে এর ধারণাগুলো আপনি কীভাবে অর্জন করতে পারেন?

ট্রানজেকশন হলো এমন একটি ফিচার যার মাধ্যমে একটি বা একাধিক অপারেশন সেট যা সিঙ্গেল লজিক্যাল ইউনিটে এক্সিকী উট হয়। মংগোডিবি ট্রানজেকশন এর ক্ষেত্রে ACID (Atomicity, Consistency, Isolation, Durability) প্রপার্টি ফলো করে থাকে।

ট্রানজেকশন দুই ধরনের হতে পারে।

অটোমেটিক ট্রানজেকশনঃ

মংগোডিবির বাই ডিফল্ট ট্রানজেকশন হলো অটোমেটিক ট্রানজেকশন। এটি ব্যবহার করা বেশ সহজ। এবং এটি আইসোলেশন গ্যারান্টি প্রভাইড করে না।

এক্সপ্লিসিট ট্রানজেকশনঃ

এক্সপ্লিসিট ট্রানজেকশন পাওয়ারফুল আইসোলেটেড এনভায়রনমেন্ট প্রভাইড করে থাকে। এবং এটি অটোমেটিক ট্রানজেকশন থেকে পাওয়ারফুল। এটি ব্যবহার করা বেশ জটিল কী স্তু এটি হাইলি ডেটা কনসিসট্রেন্সি প্রভাইড করে থাকে।

অপর দিকে লকিং ফিচার হলো এমন একটি ফিচার যাতে একাধিক ইউজার কে একই সময়ে একই ডেটা এক্সেস করতে বাঁধা প্রদান করা হয়।
ডেটা করাপশন কমাতে এই নীতি অনুসরণ করা হয়।
মংগোডিবি দুই ধরনের লকিং সিস্টেম প্রভাইড করে থাকে।

শেয়ার্ড লকিংঃ

শেয়ার্ড লকিং এ মাল্টিপল ইউজার সেইম টাইমে সেম ডেটা এক্সেস করতে পারে। এটি ডেটা করাপশন প্রতিরোধে ব্যবহার করা হয়।

এক্সক্লুসিভ লকিংঃ

এখানে লক করা ডেটাগুলো কে অন্য ইউজার যদি এক্সেস করতে চায় তখন বাঁধা প্রদান করে থাকে।

64

আপনার কোন সময় MongoDB এবং কোন সময় Redis ব্যবহার করা উচিত বলে আপনি মনে করেন?

মংগোডিবি হলো ডকুমেন্ট অরিয়েন্টেড NoSQL ডেটাবেজ। যখন ফ্লেক্সিবল এবং স্কেলেবল ডেটা স্টোর করার প্রয়োজন পরে তখন আমরা এটি ব্যবহার করতে পারি। এটি স্কিমালেস ডেটা রিড রাইটের প্রয়োজনে ব্যবহার করা যেতে পারে। কিংবা কমনলি কোন CMS, e-commerce, networking application এর জন্য এটি ব্যাবহার করা যেতে পারে।

অপর দিকে, রেডিস হলো ইন-মেমোরি ডেটাস্টোর। এটি ক্যাশিং, রিয়েলটাইম ডেটা প্রসেসিং কিংবা হাইস্পিড রিড রাইটের জন্য ব্যাবহার করা হয়। যখন কোন ডেটা ফ্রিকুয়েন্টলি ব্যাবহারের প্রয়োজন হয় এবং দ্রুত রেসপন্স দিতে হয় তখন আমরা রেডিস ব্যবহার করতে পারি। অনেকসময় আমরা এটি ডেটাবেসের সাথে ইন্টিগ্রেট করে দেই ফাস্ট পারফরম্যান্স এর জন্য।

65

MongoDB-তে আপনি কীভাবে একটি ডকুমেন্ট ডিলিট করবেন?

দুই ভাবে আমরা ডিলিট অপারেশন করতে পারি।

১. Using Mongo Shell:

নিচের কমান্ড টি লিখে একটি ডকুমেন্ট ডিলিট করতে পারি।

```
db.collection.deleteOne({
    "_id": 1
});
```

এটি দিয়ে যে কালেকশন এর আইডি 1 সেটি ডিলিট হয়ে যাবে।

২. Using MongoDB driver:

নিচের কোড টি লিখে ডিলিট করতে পারি

```
const client = new mongo.MongoClient()

    client.connect((err, db) => {
        const db = client.test
        const collection = db.products
        collection.deleteOne({
            "_id": 1
        })
    })
```

এটি দিয়েও ডেটা ডিলিট করা যাবে।

66

MongoDB-তে রেপ্লিকা সেট বলতে আপনি কী বোঝেন?

MongoDB তে সচকরাচর রেপ্লিকেশন তখন করা হয় যখন কোন রাইট বা আপডেট অপারেশন সম্পুর্ন হয়। এমনিতেও করা সম্ভব। রেপ্লিকেশন এর উদ্দেশ্য থাকে ডেটা যাতে কখনো হারিয়ে না যায়, কিংবা ডেটা যাতে মাল্টিপল সার্ভার থেকে সবসময় পাওয়া যায়।

ডেটা রেপ্লিকেশন প্রসেস সম্পুর্ন করার জন্য কী ছু বিষয় মাথায় রাখা দরকার। যেমন, কতগুলো সার্ভার রানিং, সার্ভার গুলোর লোকেশন কোথায় কিংবা কোন ধরনের সার্ভার ইত্যাদি বিষয়।

ডেটা রেপ্লিকেশন এর জন্য কয়েকটি ধাপ অতিক্রম করতে হয়।

প্রথমে কয়েকটি ইন্সট্যান্স মাল্টিপল প্রসেসে স্টার্ট করতে হবে। তারপর rs.initiate() কমান্ডের মাধ্যমে ইনিশিয়েট করতে হবে। এবং এর ভিতর আমাদের অপশন প্রভাইড করতে হবে। যেখানে থাকবে ডেটার ইনফর্মেশন এবং প্রসেস আইডি।

67

ডাটা মডেলিং কী?

ডেটা মডেলিং হলো এমন একটি টার্ম যেখানে ডেটার স্ট্রাকচার এবং রিলেশনশিপ সম্পর্কে বলা হয়। ডেটা মডেলিং করা হয় যাতে ডেটা ইফিসিয়েন্ট হয় এবং ডেটা সহজেই ব্যবহার করা যায়।

মুলত ৩ ধরনের ডেটা মডেল হয়।

- লজিক্যাল ডেটা মডেল
- ফিজিক্যাল ডেটা মডেল
- এনটিটি রিলেশনশিপ ডেটা মডেল।

68 MongoDB-তে Indexing-এর উদ্দেশ্যগুলো কী কী ?

মংগোডিবি তে কুয়েরী আরো দ্রুত করার জন্য ইনডেক্সিং করা হয়। যদি ইনডেক্সিং করা না হতো তাহলে কুয়েরীর সময় প্রতিটা ডকুমেন্ট স্ক্যান করতে হতো ফলে এটি অনেক টাইম কনজিউমিং হতো।

মংগোডিবিতে ইনডেক্সিং করার কারণে অনেকগুলো সুবিধা পাওয়া যায়। যেমন:

১. কুয়েরীর পারফরম্যান্স বৃদ্ধি পায়।

২. ডিস্ক I/O কমে আসে।

৩. মেমোরি ইউজেস কমে আসে।

মিডলওয়্যার কয়েক ধরনের হয়।

- **এপ্লিকেশন লেভেল মিডলওয়্যারঃ** এই মিডলওয়্যার সব রিকুয়েষ্ট এ কল হয়।
- **রাউটার লেভেল মিডলওয়্যারঃ** এই মিডলওয়্যার শুধু মাত্র স্পেসিফিক রাউটে হিট করলে এক্সিকী উট হয়।
- **এরর হ্যান্ডেলার মিডলওয়্যারঃ** এই মিডলওয়্যার কোন এরর হলে রান হয়।
- **থার্ড পার্টি মিডলওয়্যারঃ** চাইলে থার্ডপার্টি মিডলওয়্যারও ইউজ করতে পারি। যেমন কুকী পার্সার মিডলওয়্যার।

69

MongoDB-এর মূল উদ্দেশ্য কী?

মংগোডিবি মুল উদ্দেশ্য গুলো হলো।

- **স্কিমালেস ডেটাবেস:** মংগোডিবি তে ডেটা সংরক্ষণ করার জন্য স্কিমার প্রয়োজন হয় না।
- **ফ্ল্যাক্সিবল:** মংগোডিবি ডকুমেন্ট যেকোন সাইজের হতে পারে। যেকোন ডেটা টাইপ সাপোর্ট করে।
- **স্কেলেবল:** মংগোডিবি কে বড় পরিসরে ডেটা হ্যান্ডেল করতে চাইলে হরিজন্টালি স্কেল করা সম্ভব।
- **ইফিসিয়েন্সি:** মংগোডিবি এমন ভাবে ডিজাইন করা যাতে দ্রুত রিড রাইট করা যায়। ফলে এটি ইফিসিয়েন্স একটি ডেটাবেস।
- **হাই ভলিওম:** মংগোডিবি তে হাই ভলিওম ডেটা রাখা সম্ভব।
- **রিয়েল টাইম:** মংগোডিবি তে ডেটা রিড টাইম খুবই কম। ফলে রিয়েলটাইম ডেটা পাওয়া সম্ভব।

70

MongoDB-এর ফিচারগুলোর ৫টি উদাহরণ দিন।

মংগোডিবি হলো ডকুমেন্ট বেইজড নো এসকী উএল ডেটাবেস। যেখানে আমরা ট্র্যাডিশনাল টেবিলে ডেটা না রেখে ডকুমেন্ট আকারে ডেটা রাখা হয়। এর অনেকগুলো ফিচার রয়েছে।

যেমন

>স্কিমালেস ডেটাবেস: এখানে ডেটা স্টোর করতে হলে আমাদের স্কিমা লাগে না। কোন স্কিমা ছাড়াই ডেটাবেজে ডেটা রাখতে পারি।

>ডকুমেন্ট অরিয়েন্টেড ডেটা: মংগোডিবি তে ডেটা রাখতে কোন টেবিল ব্যবহার করা হয় না। বরং এখানে আমাদের ব্যাবহার করা হয় কী ভ্যালু পেয়ার ডকুমেন্ট।

>ফ্ল্যাক্সিবল প্রোগ্রামিং মডেল: এই ডেটাবেস অনেকগুলো প্রোগ্রামিং ল্যাঙ্গুয়েজ সাপোর্ট করে।যেমন পাইথন, জাভাস্ক্রিপ্ট, জাভা ইত্যাদি।

>সিকী উরিটি: এটি স্ট্রং সিকী উরিটি প্রভাইড করে।

>সাপোর্ট রেপ্লিকেশন: মংগোডিবি রেপ্লিকেশন সাপোর্ট করে। তাই কোন একটি ডেটাবেস সার্ভারে ডেটা রাইট করলে সেটা ভিন্ন ভিন্ন সার্ভারে রেপ্লিকেশন হয়।

71

MongoDB-তে BSON ব্যবহারের সুবিধাগুলো তুলে ধরুন।

১. **লাইটওয়েট এবং ট্রাভার্সেবল**। BSON ফরমেটে অনেক বড় ধরনের ডেটাসেট খুব সহজেই স্টোর করা যায়।এবং ট্রান্সফার সুবিধা বেশি। নেটওয়ার্কে ভালোভাবে কাজ করার জন্য উত্তম।

২. **ইফিসিয়েন্ট ফরমেট**। BSON ডেটা ফরমেটে দ্রুত কোয়েরি রান করা যায়। দ্রুত বড় ডেটাসেট স্টোর করা যায়। দ্রুত কম্পেয়ার এবং ক্যালকুলেট করা যায়।

৩. **অন্যান্য ডেটাটাইপ সাপোর্ট**। BSON Bindata, maxkey, minkey, Binary Data, ObjectID, Regular expression টাইপ গুলো সাপোর্ট করে যেখানে JSON এ এগুলো সম্ভব নয়।

72 আপনার কখন এবং কেনো Express app ও server আলাদা করা উচিত?

অনেকগুলো কারণে এক্সপ্রেস এপ্লিকেশন এবং সার্ভার পৃথক হতে পারে। যেমন:

Separation of concerns:

এটাই মোষ্ট কমন রিজন সার্ভার এবং এপ্লিকেশন আলাদা করার। এতে করে সার্ভার এবং এপ্লিকেশন দুটি আলাদা ভাবে থাকতে পারে। ট্রাবল শুট করতে সহজ হয়। কোন ফিচার ইমপ্লিমেন্ট ইজি হয়।

Scalability:

এপ্লিকেশন যখন বড় হতে থাকে তখন এটি ম্যানেজ করতে কষ্টকর হয়ে দাড়ায়। তখন যদি সার্ভার এবং এপ্লিকেশন দুটি আলাদা ভাবে থাকে তাহলে তাদের কে স্কেল করা সহজ হয়।

Deployment:

যদি এপ্লিকেশন কোন ক্লাউড প্লাটফর্মে ডেপলয় করার চিন্তা করা হয় তখন সার্ভার এবং এপ্লিকেশন দুটি আলাদা ভাবে রাখলে কীছু সুবিধা পাওয়া যায়।

73

MongoDB-তে Foreign Keys বলতে আপনি কী বোঝেন? এখানে Foreign Keys-এর অল্টারনেটিভগুলো কী কী?

মংগোডিবি তে ফরেইন কী সাপোর্ট করে না। তবে আমরা চাইলে বিকল্প কীছু কাজ করে এই সমস্যা সমাধান করতে পারি। যেমন:

>এমবেড করা:

চাইলে ডকুমেন্ট এ নতুন ডেটা ফিল্ড এড করে ডেটা এমবেড করতে পারি। ফলে আমাদের ফরেন কী প্রয়োজন হবে না।

>রেফারেন্স:

আমরা চাইলে রেফারেন্স প্রভাইড করতে পারি অন্যকোন কালেকশন ও ডকুমেন্ট এর ফলে সেখান থেকে ডেটা কুয়েরী করতে পারি। এতে করে ফরেন কী এর প্রয়োজনীয়তা আসবে না।

>জয়েন করা:

আমরা জয়েন করেও এই সমস্যার সমাধান করতে পারি।

74

আপনি WebSockets-এর মতো কীভাবে একটি MERN অ্যাপ্লিকেশনে রিয়েল-টাইম কমিউনিকেশন বাস্তবায়ন করবেন?

আমরা Socket.IO ব্যবহার করে রিয়েলটাইম কমিউনিকেশন তৈরি করতে পারি।

প্রথমে ব্যাকএন্ডে আমাদের কে socket.io ইন্সটল করতে হবে।

তারপর সার্ভারের সাথে কনেক্ট করে কমিউনিকেশন ওপেন করতে হবে।

```
const express = require('express');
const socketio = require('socket.io');


const app = express();
const server = app.listen(3000);
```

```
const io = socketio(server);

io.on('connection', (socket) => {
  // Do something when a client connects
});
socket.start()
```

ক্লায়েন্ট সাইটে নিচের কোড টুকু লিখতে হবে।

```
const socket = new WebSocket('ws://localhost:3000');

socket.on('message', (message) => {
  // Do something when a message is received from the server
});

socket.on('connect', () => {
  // Do something when the client connects to the server
});
```

এভাবে আমরা কমিউনিকেশন করতে পারি।

75

Mongoose এর কাজ কী ?

Mongoose হলো একটি MongoDB অবজেক্ট ডেটা মডেলিং (ODM) লাইব্রেরি যা MongoDB এবং Node JS-কে টার্গেট করে বিল্ড করা হয়েছে। এটি একটি Straight-forward এবং Schema-based সমাধান দিয়ে থাকে অ্যাপ্লিকেশন ডেটাগুলি মডেল করার জন্যে। এতে বিল্ডইন টাইপকাস্টিং, ভ্যালিডেশন, কুয়েরি বিল্ডিং, বিজনেস লজিক হুক সহ আরও সুবিধাবলী রয়েছে।

76 Mongoose-এর কার্যাবলী নিয়ে সংক্ষিপ্ত ধারণা দিন।

মংগোডিবির Object Document Mapping (ODM) হলো মংগোজ। এটি দিয়ে জাভাস্ক্রিপ্ট এর মাধ্যমে সহজেই মংগোডিবির সাথে ইনটারেকশন করা যায়। এটি অনেকগুলো কাজ করে। যেমন:

- **স্কিমা ডেফিনেশন:** মংগোজ দিয়ে স্কিমা ডেফিনেশন তৈরি করা হয়। ফিল্ড, ভ্যালু, টাইপ ভ্যালিডেশন ইত্যাদি এড করা যায় স্কিমা ডেফিনেশন দিয়ে।
- **মডেল ক্রিয়েশন:** স্কিমা ডেফিনেশন দিয়ে চাইলে আমরা মডেল তৈরি করতে পারি। যা কালেকশন কে রিপ্রেজেন্ট করে থাকে।
- **ডেটা ভ্যালিডেশন:** মংগোজ ইনবিল্ট ভ্যালিডেশন ম্যাকানিজম প্রভাইড করে থাকে। ফলে আমরা ডেটার সবধরনের ভ্যালিডেশন করতে পারি।
- **কুয়েরী ও ম্যানিপুলেশন:** মংগোজ দিয়ে কুয়েরী এবং ডেটা ম্যানিপুলেশন করা যায়।
- **কানেকশন ম্যানেজমেন্ট:** মংগোজ দিয়ে সরাসরি ডেটাবেস কানেকশন তৈরি করা এবং রিমুভ করা যায়।

77

Node.JS কীভাবে চাইল্ড-থ্রেডগুলোকে ম্যানেজ করে?

আমরা জানি নোডজেএস সিঙ্গেল থ্রেটেড এপ্লিকেশন। সুতরাং কোন সিঙ্ক্রোনাস কাজ সে একটি থ্রেডেই করে থাকে। কীন্তু ইন্টারেস্টিং বিষয় হলো আমরা চাইলে নোড জেএস কে মাল্টিথ্রেডেট বানাতে পারি child_process বিল্ডইন মডিউল দিয়ে। চাইল্ড প্রসেস আমাদের কে কয়েকটি মেথড প্রভাইড করে যেমন: exec, execFile, spawn, fork ইত্যাদি মেথড।

এই মেথড গুলো মুলত আপনাকে এক্সার্নাল কমান্ড কল করতে সাহায্য করে। এর মাধ্যমে আপনি চাইল্ড প্রসেসে কমান্ড গুলো রান করতে পারবেন।

১. exec মেথড কল করে প্রথম প্যারামিটারে একটি কমান্ড এবং পরে একটি কলব্যাক ফাংশন দিবেন। যেখানে ফাংশন টি error, stdout, stderr নামে ৩টি প্যারামিটার নিবে। সেখানে আপনি কাঙ্খিত ফলাফল দেখতে পারবেন।

২. execFile মেথডটিও exec মেথডের মতো সেইম ভাবে কাজ করে। শুধু প্রথম প্যারামিটারে ফাইলের পাথ টা বলে দিতে হয়। কেননা এটা ফাইল এক্সিকী উট করে থাকে।

৩. spawn মেথড স্ট্রিম নিয়ে কাজ করে। এবং এই মেথড টা একটি চাইল্ড প্রসেস অবজেক্ট রিটার্ন করে থাকে। যেখানে ইভেন্ট ট্রিগারের মাধ্যমে কাজ সম্পুর্ণ করা হয়ে থাকে।

৪. fork মেথডটি স্পেশাল মেথড যেটি IPC(Inter-process Communication) এর মাধ্যমে কোন একটি স্পেসিফিক মডিউল রান হয়।

78

Node.js এর উল্লেখযোগ্য বৈশিষ্ট্যগুলো কী কী ?

Node.js হচ্ছে একটি ওপেন সোর্স ও ক্রস প্ল্যাটফর্ম। এটি একটি জাভাস্ক্রিপ্ট রান-টাইম এনভাইরোনমেন্ট, যা ব্রাউজারের বাইরে টার্মিনালে সংগ্রহ বা জমা করা হয়। এটি মূলত সার্ভার সাইড স্ক্রিপ্টিং ও কমান্ড লাইন টুল এর জন্য ব্যাবহৃত হয়ে থাকে। এক্ষেত্রে ওয়েব সাইট লোড হওয়ার আগে স্ক্রিপ্ট সার্ভার সাইড এর কম্পিউটারে রান হয় ও ডাইনামিকভাবে ওয়েব পেজ লোড করে। এভাবে ক্লায়েন্ট ও সার্ভার উভয় সাইডেই জাভাস্ক্রিপ্ট ব্যবহার করার মাধ্যমে Node.js "সর্বত্রই জাভাস্ক্রিপ্ট" প্যারাডিজম প্রকাশ করে।

নিম্নলিখিত কিছু গুরুত্বপূর্ণ বৈশিষ্ট্যগুলো Node.js কে সফ্টওয়্যার আর্কিটেক্টদের প্রথম পছন্দ করে তোলে:

- Asynchronous & Event Driven – Node.js লাইব্রেরির সমস্ত API গুলো অ্যাসিঙ্ক্রোনাস, অর্থাৎ, নন-ব্লকিং। এর অর্থ দাঁড়ায়, একটি Node.js ভিত্তিক সার্ভার কখনই ডেটা ফেরত দেওয়ার জন্য API এর জন্য অপেক্ষা করে না। সার্ভারটি কল করার পর পরবর্তী API-এ চলে যায়। Node.js এর ইভেন্টগুলির একটি নোটিফিকেশন মেকানিজম সার্ভারকে পূর্ববর্তী API কল থেকে রেস্পন্স পেতে সহায়তা করে।

79

আপনি কী একটি Node.js-এর ব্যাকএন্ডের সাথে একটি React-এর ফ্রন্ট-এন্ড সংযোগ করার প্রক্রিয়া বর্ণনা করতে পারবেন?

নোডজেএস ব্যাকএন্ড এর সাথে রিএক্ট সংযোগ করার প্রক্রিয়া নিম্নরুপ।

- প্রথমে আমাদের ব্যাকএন্ড এপ্লিকেশন কে কোন সার্ভারে হোস্ট করতে হবে। এবং এপিআই এন্ডপয়েন্ট পাবো।
- রিএক্টএ useEffect হুকের ভিতর আমাদের সেই এপিআই এন্ডপয়েন্ট কে কল করে ডেটা ফেচ করতে হবে।
- ফেচকৃত ডেটা UI তে শো করতে হবে।

80

Node.JS-এ আপনি কীভাবে কলব্যাক হেল পরিহার করবেন?

Ostad

কলব্যাক হেল বলতে বুঝায় যেখানে একটি ফাংশন কল করে তার প্যারামিটার হিসেবে আরেকটি ফাংশন পাস করা হয়। এবং এটি চলতেই থাকে।

তখন একটি কলব্যাক হেল তৈরি হয়। যেমন

```
fs.readdir(source, function (err, files) {
  if (err) {
    console.log('Error finding files: ' + err)
  } else {
    files.forEach(function (filename, fileIndex) {
      console.log(filename)
      gm(source + filename).size(function (err, values) {
        if (err) {
          console.log('Error identifying file size: ' + err)
        } else {
          console.log(filename + ' : ' + values)
          aspect = (values.width / values.height)
          widths.forEach(function (width, widthIndex) {
            height = Math.round(width / aspect)
            console.log('resizing ' + filename + 'to ' + height + 'x' + height)
            this.resize(width, height).write(dest + 'w' + width + '_' + filename, function(err) {
              if (err) console.log('Error writing file: ' + err)
              })
            }.bind(this))
```

```
})
        }.bind(this))
      }
    })
  })
 }
})
```

কলব্যাক হেল পরিহার করার জন্য আমরা প্রমিস এপিআই ব্যাবহার করতে পারি। এক্ষেত্রে দুটি পদ্ধতি গ্রহণ করতে পারি।

> then catch এর ব্যবহার এর মাধ্যমে কলব্যাক হেল পরিহার করতে পারি। যেমন

```
MyAsyncFunc()
        .then(data => console.log(data))
        .catch(err => console.log(err))
```

> async await ব্যবহার করার মাধ্যমে ফাংশন কে এসিঙ্ক বানাতে পারি।

```
async function myFunc() {
        const data = await promiseAPI()
        console.log(data)
}
```

81

Node.JS-এর ক্ষেত্রে কনকারেন্সি কীভাবে কাজ করে?

Ostad

কনকারেন্সি মুলত ইভেন্ট ড্রিভেন আর্কিটেকচার, নন ব্লকিং I/O এর উপর ভিত্তি করে মাল্টিপল অপারেশন একসাথে করে থাকে। যেখানে মেইনথ্রেড কে ব্লক করা হয় না।

সিঙ্গেলথ্রেডেড নোডজেএস ইভেন্ট লুপের মাধ্যমে রিকুয়েষ্ট হ্যান্ডেল করে থাকে। এবং কলব্যাক এক্সিকী উট করার মাধ্যমে সম্পুর্ণ হওয়া I/O অপারেশন গুলোকে মেইনথ্রেডে রান করে।

- Event Loop : ইভেন্ট লুপ হলো নোড জেএস এর কোর বিষয়। এটি এসিঙ্ক্রোনাস টাস্ক কনকারেন্টলি হ্যান্ডেল করা হয়।
- Non blocking I/O : যখন কোন I/O অপারেশন করা হয়। যেমন কোন ফাইল রিড করা। তখন নোড জেএস মেইনথ্রেডে কাজটি না করে কলব্যাক রেজিস্ট্রার করে তার মাধ্যমে হ্যান্ডেল করা হয়।
- Asynchronous API: নোড জেএস এডভান্স সেট অফ এসিঙ্ক্রোনাস API প্রভাইড করে। যেমন fs module, http module, setTimeout ইত্যাদি। এর মাধ্যমে কনকারেন্সি হ্যান্ডেল করা হয়।
- Callback & Promises: প্রমিসের মাধ্যমে এসিঙ্ক্রোনাস কাজ সম্পুর্ণ করা যায়। এতে করে মাল্টিপল কাজ একসাথে করা সম্ভব হয়।
- worker threads : এর মাধ্যমে নোড জেএস কে মাল্টিথ্রেড করা যায়। ফলে অনেকগুলো কাজ একসাথে সম্পুর্ণ সম্ভব হয়।

82

একটি Promise-Based Node.JS অ্যাপ্লিকেশনকে Async/Await-এ রুপান্তরিত করুন।

কয়েটি স্টেপের মাধ্যমে আমরা প্রমিজ বেইজড এপ্লিকেশন কে Async Await করতে পারি।

১. রিপ্লেস .then() and .catch() with async/await.

২. একটি ফাংশন লিখুন এবং তার আগে Async যুক্ত করুন। এবং এর ভিতর প্রমিজ ফাংশন টি কল করে একটি ভ্যারিয়েবলে রাখুন।

অবশ্যই ফাংশন টির আগে await দিতে ভুলবেন না।

৩. ইরর হ্যান্ডেলের জন্য try catch ব্লক নিন। এবং প্রমিজ কলটি try ব্লকের ভিতর ঢুকান। catch ব্লকের ভিতর এরর টি হ্যান্ডেল করুন।

৪. এখন async ফাংশন টি কল করুন। ব্যাস আমাদের কাজ শেষ।

83

Node.JS-এ Stream কী এবং এর বিভিন্নতা সম্পর্কে আপনি কতটুকু জানেন?

Stream হলো এমন একটি ডেটা হ্যান্ডেলিং পদ্ধতি যার মাধ্যমে আমরা কোন একটি লার্জ ফাইল কে ছোট ছোট অংশে বিভক্ত করে ব্যবহার করতে পারি। এতে করে মেমোরি ইফিসিয়েন্সি এবং টাইম ইফিসিয়েন্সি বাড়ে। কেননা সম্পুর্ণ ডেটা রিড করে মেমোরি তে রাখতে টাইম এবং মেমোরি দুটোই বেশি ব্যবহার হতো।

মুলতঃ ৪ধরনের স্ট্রিম হয়ে থাকে।

১. **readable stream**: রিডেবল স্ট্রীম দিয়ে একটি সিকুয়েন্সিয়াল ম্যানারের মাধ্যমে ডেটা রিড বা পড়া যায়।

২. **writable stream**: রিডেবল স্ট্রীমের মতোই রাইটেবল স্ট্রীম দিয়ে সিকুয়েন্সিয়াল ম্যানারের মাধ্যমে ডেটা রাইট করা যায়।

৩. **duplex stream**: ডুপ্লেক্স স্ট্রীম দিয়ে ডেটা রিড এবং রাইড উভয় ই করা সম্ভব হয়।

৪. **transform stream**: এটি স্পেশাল টাইপ ডুপ্লেক্স স্ট্রীম যা দিয়ে কম্পারিজন, এনক্রিপশন বা পার্সিং করা পসিবল হয়।

84 Node.JS-এ Blocking Code কী?

আমরা জানি নোড জেএস সিঙ্গেল থ্রেড এপ্লিকেশন। তার মানে নোড জেএস এর কোড একটি মাত্র থ্রেডে রান হয়। কীন্তু নোড জেএস কে মাল্টিথ্রেডের মতো ব্যাবহার করা যায় এসিঙ্কক্রোনাস কোডের মাধ্যমে। কী ন্তু এর পরও যখন কোন কোড সরাসরি মেইন থ্রেডে রান হয় তখন সেটা কে ব্লকিং কোড বলা হয়।

85

Event Loop কী এবং এটি কীভাবে কাজ করে?

Ostad

জাভাস্ক্রিপ্টএর একদম বেসিক জিনিসপত্র গুলোর ভিতর ইভেন্ট লুপ হলো অন্যতম। ইভেন্ট লুপের মাধ্যমে সিঙ্গেল থ্রেডেট এপ্লিকেশন মাল্টি থ্রেড এর মতো কাজ করতে পারে। ইভেন্ট লুপ কয়েকটি কম্পোনেন্ট নিয়ে গঠিত। যেমন:

১.কল স্ট্যাকঃ

এটি একটি LIFO (Last In First Out) স্ট্যাক। এটির মাধ্যমে ফাংশন কল ট্র্যাক রাখা হয়। যেখানে সর্বশেষ অপারেশন টি সর্ব প্রথম হয়ে থাকে।

২. ইভেন্ট কী উঃ

যেখানে ওয়েব এপিআই দিয়ে সম্পুর্ণ কাজ গুলো সারিবদ্ধ ভাবে থাকে। এটি একটি FIFO (First In First Out) কী উ। যেখানে যে সর্বপ্রথম শেষ হয়েছে তাকে সর্বপ্রথম এক্সিকী উট করা হয়।

ইভেন্ট কী উ আবার দুটি রয়েছে। টাস্ক কী উ এবং মাইক্রো টাস্ক কী উ। যেখানে মাইক্রো টাস্ক কী উ এর প্রায়োরিটি বেশি থাকে এবং এখানে প্রমিজ গুলো রিজল্ভ হয়ে আসে।

৩. ইভেন্ট লুপঃ

ইভেন্ট লুপ হলো একটি চলমান ক্রিয়া। যে সবসময় মেইন থ্রেড বা কলস্ট্যাক কে পর্যবেক্ষণ করে যখন কলস্ট্যাক খালি থাকে তখন ইভেন্ট কী উ থেকে ইভেন্ট নিয়ে কলস্ট্যাক এ পাঠিয়ে দেয়। এবং কলস্ট্যাক তা ফায়ার করে।

86 Node.JS কী পুরোপুরি সিংগেল থ্রেডেড? এ সম্পর্কে আপনার মতামত কী?

Node.js সিঙ্গেল থ্রেডেড। কী ন্তু আমরা চাইলে কী ছু স্পেসিফিক থ্রেট তৈরি করতে পারি দুটি উপায়ে। child_process এবং worker_threads দিয়ে। চাইল্ড প্রসেস এর আলাদা মেমোরি স্পেস রয়েছে। এটি প্যারালাল রান করতে পারে এবং এটি আইসোলেটেড। অন্যদিকে ওয়ার্কার থ্রেড দিয়ে মেইন থ্রেডের মতো আরেকটি থ্রেড তৈরি করা যেতে পারে।

87

আপনার কী মনে হয় NodeJS-এ Class-এর ব্যবহার সম্ভব?

ব্যবহার সম্ভব। তবে একটি কথা মাথায় রাখতে হবে যে নোড জেএস এর ক্লাস জাভা কিংবা সি++ এর ক্লাসে মতো নয়। এটি মুলত প্রটোপাইপ চেইনের মাধ্যমে কাজ করে থাকে।

ES6 সর্বপ্রথম ক্লাস কম্পোনেন্ট এর ইমপ্লিমেন্টেশন নিয়ে আসে। যদিও এটি একটি সিনথ্যাকটিক্যাল এবস্ট্রাকশন।

৪৪ কোন কোন ক্ষেত্রে আপনার NodeJS ব্যবহার করা উচিত না এবং কেনো?

১. CPU intensive task:

Nodejs ডিজাইন করা হয়েছে I/O বেইজড কাজ করার জন্য। যখন CPU ইনটেনসিভ কাজ করা হবে তখন এটি তার বেস্ট আউটপুট দিতে পারবে না। এক্ষেত্রে পাইথন কিংবা জাভা ভালো আউটপুট দিবে।

২. রিয়েলটাইম এপ্লিকেশন:

রিয়েলটাইম এপ্লিকেশনের ক্ষেত্রে NodeJS ভালো অপশন নয়। হাই কনকারেন্সি সিনারিও তে এটি ব্লকিং বিহেভিয়ার করতে পারে।

৩. কমপ্লেক্স মাল্টিথ্রেডিং এপ্লিকেশন:

নোডজেএস যেহেতু সিঙ্গেল থ্রেড এপ্লিকেশন, তাই কমপ্লেক্স মাল্টিথ্রেডিং এপ্লিকেশন এর ক্ষেত্রে এটি ভালো নির্বাচন হবে না।

89

NodeJs ব্যবহৃত হয় ৫টি ক্ষেত্রের নাম বলুন।

নোডজেএস ক্রস প্ল্যাটফর্ম ওপেনসোর্স জাভাস্ক্রিপ্ট রানটাইম। এটি অনেক কাজে ব্যবহার করা হয়। যেমন:

>ওয়েব এপ্লিকেশন:

ওয়েব এপ্লিকেশন তৈরি করার কাজে নোডজেএস ব্যবহার করা হয়।

>রিয়েলটাইম এপ্লিকেশন:

যখন আমাদের কোন রিয়েলটাইম এপ্লিকেশন দরকার হয়। যেমন চ্যাটএ্যাপ। তখন আমরা হাই কনকারেন্সি হ্যান্ডেল করতে পারি নোড জেএস দিয়ে।

>API ডেভেলপমেন্ট:

আমরা কোন এপিআই ডেভেলপমেন্ট করতে নোডজেএস ব্যবহার করতে পারি।

>ব্যাকএন্ড এপ্লিকেশন:

কোন ফুলস্ট্যাক এপ্লিকেশন এর ব্যাকএন্ড হিসেবে নোডজেএস ব্যবহার করতে পারি।

90

NodeJS-এ কলব্যাক হেল পরিহার করার জন্য কী কী উপায় অবলম্বন করা যেতে পারে বলে আপনি মনে করেন?

কলব্যাক হেল বলতে বুঝায় যেখানে একটি ফাংশন কল করে তার প্যারামিটার হিসেবে আরেকটি ফাংশন পাস করা হয়। এবং এটি চলতেই থাকে। তখন একটি কলব্যাক হেল তৈরি হয়।

কলব্যাক হেল পরিহার করার জন্য আমরা প্রমিস এপিআই ব্যাবহার করতে পারি। এক্ষেত্রে দুটি পদ্ধতি গ্রহণ করতে পারি।

> then catch এর ব্যবহার এর মাধ্যমে কলব্যাক হেল পরিহার করতে পারি। যেমন:

```
MyAsyncFunc()
    .then(data => console.log(data))
    .catch(err => console.log(err))
```

> async await ব্যবহার করার মাধ্যমে ফাংশন কে এসিঙ্ক বানাতে পারি।

```
async function myFunc() {
    const data = await promiseAPI()
    console.log(data)
}
```

91

Node.js-এ async কী উ ইনপুট হিসাবে নেওয়া দুটি আর্গুমেন্টের তালিকা করুন।

নোডজেএস এ async.queue হলো একটি ইউটিলিটি ফাংশন। যেটি প্যারালাল একাধিক এসিঙ্ক্রোনাস ফাংশন রান করতে পারে।

এই ফাংশন টি মুলত দুটি আর্গুমেন্ট গ্রহণ করে। প্রথমটি হলো একটি ফাংশন যেটি রান হতে থাকবে এবং অপরটি হলো একটি সংখ্যা যা দিয়ে বলা হয় এটি কতবার রান হবে।

92

Node.js-এ স্ট্যাটিক রিসোর্স সার্ভের জন্য কোন মডিউল ব্যবহার করা হয়?

স্ট্যাটিক ফাইল সার্ভ করার জন্য কয়েকটি লাইব্রেরি ব্যবহার করা যেতে পারে। যেমন

Express:

এটি একটি ব্যাকএন্ড ফ্রেমওয়ার্ক। এর মাধ্যমে স্ট্যাটিক ফাইল সার্ভ করা যেতে পারে। আমাদের কে মিডলওয়্যার ব্যবহার করে এই কাজটি সম্পুর্ন করতে হবে।

node-static:

এটি একটি লাইটওয়েট এবং সিম্পল লাইব্রেরি যা দিয়ে স্ট্যাটিক ফাইল সার্ভ করতে পারি।

serve-static:

এটি আরেকটি স্ট্যাটিক ফাইল সার্ভ করার জন্য লাইব্রেরি। এটি অনেকগুলো ফিচার প্রভাইড করে। যেমন ক্যাশিং, কমপ্রেশন ইত্যাদি।

93

REPL-এর কয়টি কাজ বা ফাংশন সম্পর্কে বলুন।

REPL হলো একটি প্রোগ্রামিং ল্যাঙ্গুয়েজ এনভায়রনমেন্ট। সোজা বাংলায় বলা যায় একটি কনসোল উইন্ডো। এটি সিঙ্গেল লাইন কোড ইনপুট নেয় এবং সেটির ফলাফল রিটার্ন করে থাকে। এটি মুলত ৪টি পর্যায়ে কাজ করে থাকে।

১. **রিড**: ইউজার থেকে ইনপুট রিড করে।

২. **ইভ্যাল**: এই পর্যায়ে ইনপুট কে ইভ্যালুয়েশন করে থাকে।

৩. **প্রিন্ট**: আউটপুট প্রিন্ট করে কনসোলে।

৪. **লুপ**: এই প্রসেস টি লুপ করতে থাকে।

94

NodeJS-এ Event Emitter কী এবং এর ফাংশন নিয়ে আপনার ধারণা কতটুকু?

Ostad

ইভেন্ট এমিটর হলো একটি অবজেক্ট যেটি কোন একটা ইভেন্ট ইমিট করতে পারে। ইভেন্ট গুলো নোটিফিকেশন এর মতো কাজ করে। যেখানে কোন একটি ইভেন্ট ট্রিগার হলে বাকী সব অবজেক্ট এর কাছে এলার্ট চলে যায় যে কোন ইমিট ট্রিগার হয়েছে।

একটি এমিট ক্রিয়েট করতে EventEmittor ক্লাস ব্যবহার করতে পারি। যেটি events মডিউলে পাওয়া যায়।

emit() মেথডটি ইভেন্ট তৈরি করতে ব্যবহার করা হয়। এটি দুটি আর্গুমেন্ট গ্রহণ করে থাকে। প্রথম টি হলো ইভেন্টের নাম এবং পরেরটি হলো একটি কলব্যাক ফাংশন। যখন এই নামের ইভেন্ট টি ট্রিগার হবে তখন এই কলব্যাক ফাংশন টি রান হবে।

```
const emitter = new EventEmitter();
emitter.on('event', () => {
  console.log('Event emitted');
});
emitter.emit('event');
```

95

NodeJs-এ NPM এর সম্পূর্ণ অর্থ নিয়ে কী আপনার ধারণা আছে?

NPM হলো Node Package Manager. এটি দিয়ে আমরা চাইলে আমাদের প্রজেক্টে এক্সটার্নাল লাইব্রেরি ইউজ করতে পারি।

96

NodeJS-এ Libuv কীভাবে কাজ করে?

লিভইউভি একটি পাওয়ারফুল লাইব্রেরি যেটি Nodejs কে এসিঙ্ক্রোনাস করতে সাহায্য করে। মুলত ইভেন্ট লুপ এই লাইব্রেরি কে ব্যবহার করে মেইনথ্রেড কে ফ্রী রেখে কলব্যাকের মাধ্যমে ইভেন্ট ট্রিগার করে থাকে। টাইমার, চাইল্ড প্রসেস, ফাইল সিস্টেম এক্সেস নেটওয়ার্ক I/O ইত্যাদি হলে এই LibUV এর ফিচার।

প্রথমে কোন একটি এসিঙ্ক্রোনাস ইভেন্ট সংঘটিত হলে ইভেন্ট লুপ অপারেটিং সিস্টেম কে সেই ইভেন্ট টি পাঠিয়ে দেয়। অপারেটিং সিস্টেম কর্নেলের মাধ্যমে সেই অপারেশন সম্পুর্ন করে। তারপর ইভেন লুপে কলব্যাক পাঠায়। ইভেন্ট লুপ মেইন থ্রেড যখন খালি থাকে তখন তাকে কল করে। এভাবেই LivUV কাজ করে থাকে।

97

Node.js অ্যাপ্লিকেশনের ক্ষেত্রে ডিফল্ট স্কোপ কী?

নোডজেএস এর ডিফল্ট স্কোপ হলো লোকাল স্কোপ। যদি কোন ফাংশন এর ভিতর কোন ভ্যারিয়েবল ডিক্লেয়ার করা হয় সেটি শুধু মাত্র ওই ফাংশন এর ভিতর থেকেই এক্সেস করা সম্ভব।

যেমন

```
function greet() {
  var name = "John Doe";
  console.log("Hello, " + name);
}
greet();
```

এখানে যদি name ভ্যারিয়েবল টি ফাংশন এর বাহির থেকে এক্সেস করতে চাই তাহলে এরর থ্রো করবে।

98

NodeJS-এ Occasion Circle বলতে কী বোঝায়?

নোডজেএস এ Occasion Circle হলো একটি ডেটা স্ট্রাকচার যেটি ইভেন্ট প্রসেসের কাজে ব্যবহার করা হয়। যেমন কোন ইউজার যদি কোন ক্লিক করে তাহলে সেই ইভেন্ট টি Occasion Circle এ এড হবে। Occasion circle তারপর এটি রান করাবে।

99

NodeJS ব্যবহারের সুবিধা এবং অসুবিধাগুলো কী কী ? এর একটি তালিকা তৈরী করুন।

সুবিধা:

- Asynchronous I/O ব্যবহারের মাধ্যমে অনেক বেশি পরিমাণে ইউজার রিকোয়েস্ট নির্ভর অ্যাপ্লিকেশন ডেভেলপমেন্ট সম্ভব।
- যেহেতু Node.js তৈরি হয়েছে জাভাস্ক্রিপ্ট এর উপর ভিত্তি করেই তাই ব্যাক এন্ড ফ্রন্ট এন্ড সবখানেই একটা ল্যাঙ্গুয়েজ ব্যবহার করেই খুব দ্রুত ডেভেলপমেন্ট সম্ভব। টেকনোলজি সুইচের দরকার পরে না।
- অন্যান্য স্ট্যাকের মত আলাদা করে সার্ভার টুল, তার সঙ্গে ডেপ্লয়মেন্ট ভিত্তিক কনফিগারেশন জানার প্রয়োজন পরে না কারণ Node.js এর সাথেই সার্ভার টেকনোলজি বিল্ট ইন এবং ডেপ্লয়মেন্টও সহজ।
- অ্যাক্টিভ কমিউনিটির অবদানের কারনে npm এর মত রিপজিটরিতে যুক্ত হচ্ছে অনেক অনেক রেডি প্যাকেজ যেগুলো ব্যবহার করে প্রায় সব রকম টুলস, সফটওয়্যার ডেভেলপ করা সম্ভব

অসুবিধা:

- CPU ইন্টেন্সিভ টাস্ক যেমন রিপোর্ট জেনারেট বা অ্যানালাইটিকস ভিত্তিক সফটওয়্যার ডেভেলপমেন্টে অসুবিধা (যদিও থ্রেড গুলোকে মাল্টিপল কোরে ডিস্ট্রিবিউট করে এই সমস্যা সমাধান করা যায়)
- ইভেন্ট ড্রাইভেন মেথডোলজি নিয়ে স্পষ্ট ধারনা না থাকলে এবং এর জন্য খারাপ মানের কোডিং এর কারনে উল্টো নিম্ন মানের সফটওয়্যার তৈরি হতে পারে। যেমন — সঠিকভাবে কলব্যাক হেল ম্যানেজ করতে না পারা।

100 NodeJs-এ কয়ধরনের API কাজ করে? এগুলো কী কী?

নোডজেএস অনেক ধরনের API নিয়ে কাজ করে থাকে। যেমন

›REST API:

সবচেয়ে জনপ্রিয় এবং কমন এপিআই হচ্ছে REST API. Representational State Transfer (REST) আর্কিটেকচার স্টাইল, যা HTTP ভার্ব গুলো কে পারফর্ম করার জন্য ব্যবহার করা হয়।

›GraphQL APIs:

এটি নতুন ধরনের এপিআই। যেখানে GraphQL সার্ভার এবং ক্লায়েন্টের সাথে কমিউনিকেশন হয়।

›SOAP API:

Simple Object Access Protocol (SOAP) একটি লেস পপুলার একটি এপিআই। যার জনপ্রিয়তা বর্তমানে অনেক কম। এটিও নোডজেএস সাপোর্ট করে।

›WebSocket:

বাই ডিরেকশনাল কমিউনিকেশন স্থাপন করার জন্য এটি অনেক জনপ্রিয়। চ্যাটিং এবং নোটিফিকেশন সিস্টেমে এই ইপিআই ব্যবহার করা হয়।

Ostad

101

NodeJs-এ non-hindering কী? ব্যখ্যা করুন।

102

NodeJS কীভাবে Child Thread হ্যান্ডেল করে এ সম্পর্কে বিস্তারিত বলুন।

আমরা জানি নোডজেএস সিঙ্গেল থ্রেটেড এপ্লিকেশন। সুতরাং কোন সিঙ্ক্রোনাস কাজ সে একটি থ্রেডেই করে থাকে। কীন্তু ইন্টারেস্টিং বিষয় হলো আমরা চাইলে নোড জেএস কে মাল্টিথ্রেডেট বানাতে পারি child_process বিল্ডইন মডিউল দিয়ে। চাইল্ড প্রসেস আমাদের কে কয়েকটি মেথড প্রভাইড করে যেমন: exec, execFile, spawn, fork ইত্যাদি মেথড।

এই মেথড গুলো মুলত আপনাকে এক্স্যার্নার কমান্ড কল করতে সাহায্য করে। এর মাধ্যমে আপনি চাইল্ড প্রসেসে কমান্ড গুলো রান করতে পারবেন।

১. **exec** মেথড কল করে প্রথম প্যারামিটারে একটি কমান্ড এবং পরে একটি কলব্যাক ফাংশন দিবেন। যেখানে ফাংশন টি error, stdout, stderr নামে ৩টি প্যারামিটার নিবে। সেখানে আপনি কাঙ্খিত ফলাফল দেখতে পারবেন।

২. **execFile** মেথডটিও exec মেথডের মতো সেইম ভাবে কাজ করে। শুধু প্রথম প্যারামিটারে ফাইলের পাথ টা বলে দিতে হয়। কেননা এটা ফাইল এক্সিকী উট করে থাকে।

৩. **spawn** মেথড স্ট্রিম নিয়ে কাজ করে। এবং এই মেথড টা একটি চাইল্ড প্রসেস অবজেক্ট রিটার্ন করে থাকে। যেখানে ইভেন্ট ট্রিগারের মাধ্যমে কাজ সম্পুর্ণ করা হয়ে থাকে।

৪. **fork** মেথডটি স্পেশাল মেথড যেটি IPC(Inter-process Communication) এর মাধ্যমে কোন একটি স্পেসিফিক মডিউল রান হয়।

103

যদি NodeJS সিংগেল থ্রেডেড হয়ে থাকে তাহলে এটি কীভাবে Concurrency হ্যান্ডেল করে?

নোডজেএস ইভেন্ট লুপের মাধ্যমে কনকারেন্সি হ্যান্ডেল করে থাকে। যখন কোন রিকুয়েষ্ট আসে সেটি চেক করা হয় এসিঙ্ক্রোনাস কী না। যদি হয় তাহলে সেটি এপিআই কে দিয়ে দেওয়া হয় কাজটি সম্পুর্ন করার জন্য। সেটি কাজ সম্পুর্ণ করে কী উতে জমা হয়। কী উ থেকে মেইন থ্রেডে গিয়ে কলব্যাক কল হয়। এভাবে যত রিকুয়েষ্ট আসে সবগুলো মেইনথ্রেডে হ্যান্ডেল হয়ে ওয়ার্কার দের কে ডিস্ট্রিবিউশন করে দেওয়া হয়। এভাবেই কনকারেন্সি হ্যান্ডেল করা হয়।

104 NodeJs কী সম্পূর্ণভাবে একটি সিংগেল থ্রেডের উপর নির্ভর করে?

Node.js সিঙ্গেল থ্রেডেড। কীন্তু আমরা চাইলে কীছু স্পেসিফিক থ্রেট তৈরি করতে পারি দুটি উপায়ে। child_process এবং worker_threads দিয়ে। চাইল্ড প্রসেস এর আলাদা মেমোরি স্পেস রয়েছে। এটি প্যারালাল রান করতে পারে এবং এটি আইসোলেটেড। অন্যদিকে ওয়ার্কার থ্রেড দিয়ে মেইন থ্রেডের মতো আরেকটি থ্রেড তৈরি করা যেতে পারে।

105

Blocking Code কী এবং Node কীভাবে Blocking Code প্রিভেন্ট করে?

যে কোড এসিঙ্ক্রোনাস নয়। সে কোড গুলো ব্লকিং কোড। ব্লকিং কোড সরাসরি মেইন থ্রেডে রান করে ফলে সে অন্য আর কোন কাজ করতে পারে না। যেমন কোন এটি ফরলুপ। যদি ফরলুপ টি কোন বড় ধরনের ক্যালকুলেশন করে তাহলে তখন তার মেইন থ্রেড ব্যাস্ত থাকে। কেননা সেটি মেইন থ্রেড রান হয়। এমন কোড করা ডেভেলপার দের একদম ই অনুচিত। বরং এসিঙ্ক্রোনাস কোড করে মেইন থ্রেড কে সবসময় ফ্রী রাখা উচিত। এবং এতে করে ব্লকিং কোড প্রিভেন্ট হয়।

106

NodeJS-এ Read Eval Print Loop (REPL) কী এবং এটি কীভাবে কাজ করে?

REPL হলো একটি প্রোগ্রামিং ল্যাঙ্গুয়েজ এনভায়রনমেন্ট। সোজা বাংলায় বলা যায় একটি কনসোল উইন্ডো। এটি সিঙ্গেল লাইন কোড ইনপুট নেয় এবং সেটির ফলাফল রিটার্ন করে থাকে। এটি মুলত ৪টি পর্যায়ে কাজ করে থাকে।

১. **রিড**: ইউজার থেকে ইনপুট রিড করে।

২. **ইভ্যাল**: এই পর্যায়ে ইনপুট কে ইভ্যালুয়েশন করে থাকে।

৩. **প্রিন্ট**: আউটপুট প্রিন্ট করে কনসোলে।

৪. **লুপ**: এই প্রসেস টি লুপ করতে থাকে।

107

Node.js-এ async.queue ইনপুট হিসাবে যে দুটি আর্গুমেন্ট গ্রহণ করে তা তালিকাভুক্ত করুন।

নোডজেএস এ async.queue হলো একটি ইউটিলিটি ফাংশন। যেটি প্যারালাল একাধিক এসিঙ্ক্রোনাস ফাংশন রান করতে পারে।

এই ফাংশন টি মুলত দুটি আর্গুমেন্ট গ্রহণ করে। প্রথমটি হলো একটি ফাংশন যেটি রান হতে থাকবে এবং অপরটি হলো একটি সংখ্যা যা দিয়ে বলা হয় এটি কতবার রান হবে।

108 Node.js এর ক্ষেত্রে Buffers কী? উদাহরণসহ বুঝিয়ে বলুন।

বাফার হলো নোড জেএস এর একটি ফিচার। যা পিউর, ফিক্সলেন্থ সিকুয়েন্স বাইট। এটি V8 হিপে মেমোরি এলোকেট করে। এটি বাফার এবং এরের ভিতর সিমিলারিটি রয়েছে। কীন্তু টাইপ ভিন্ন।

বাফার ফাংশনালিটি করার জন্য নোড জেএস এর বিল্ট-ইন লাইব্রেরি রয়েছে। যেটি অনেক গুলো মেথড প্রভাইড করে থাকে। যেমন

Buffer.alloc(size):

এটি বাফার ক্রিয়েশন এর কাজে ব্যবহার করা হয়। এবং সাইজ এলোকেট করা হয়।

Buffer.form():

এটি কোন একটি ডেটা থেকে বাফার ক্রিয়েট করে থাকে।

Buffer.write:

এটি বাফারে ডেটা রাইট করে থাকে।

toString:

বাফার থেকে ডেটা রিড করতে পারে এই মেথড কল করে।

Buffer.sBuffer():

এটি চেক করে থাকে কোন অবজেক্ট বাফার কী না।

Buffer.length:

এটি বাফারের লেন্থ বের করতে ব্যবহার হয়।

Buffer.copy:

একটি বাফার থেকে অন্য বাফারে ডেটা কপি করতে ব্যবহার করা হয়।

Buffer.concat()

দুটি বাফার কে একসাথে জয়েন করে থাকে এই মেথড।

Ostad

109

Node.js-এ Stub বলতে কী বোঝেন এবং এ সম্পর্কে আপনার ধারণা কতটুকু?

110 Node.js-এ Thread Pool কী এবং এটি কোন লাইব্রেরি দ্বারা পরিচালিত হয়?

নোডজেএস থ্রেডপুল মুলত ইউজ হয় এসিঙ্কক্রোনাস টাস্ক হ্যান্ডেল করার জন্য। থ্রেডপুল হলো কয়েকটি থ্রেডের একটি গ্রুপ। যেখানে কোন একটি টাস্ক সাবমিট হলে কী উতে প্রসেস করে। তারপর নেক্সট ফ্রী পুল সেটি এক্সিকী উট করে।

থ্রেডপুল I/O অপারেশন যেমন, ফাইল রিডিং, নেটওয়ার্ক রিকুয়েষ্ট ইত্যাদি করে থাকে।

CPU ইনটেনসিভ টাস্ক যেমন ক্রিপ্টো, কমপ্রেশন ইত্যাদি করে।

মুলত থ্রেটপুল নোডজেএস এর পারফরম্যান্স উন্নত করে। থ্রেটপুল থাকার কারনে নোডজেএস এসিঙ্কক্রোনাসলি কাজ করতে পারে।

111

Node মডিউলগুলো কীভাবে এক্সটার্নালি এভেইলেবল করা যায়?

আমরা চাইলে কোন মডিউল NPM (Node Package Manager) এ পাবলিশ করে অন্যান্য ডেভেলপার দের জন্য এভেইলএভল করতে পারি। ফলে তারাও সেইম মডিউল NPM থেকে ইন্সটল করে ব্যবহার করতে পারবে।

112

Node.js-এ স্ট্যাটিক ফাইলগুলো সার্ভ করার ক্ষেত্রে কোন মডিউল ব্যবহার করা হয় বলে আপনি মনে করেন?

স্ট্যাটিক ফাইল সার্ভ করার জন্য কয়েকটি লাইব্রেরি ব্যবহার করা যেতে পারে। যেমন:

>Express:

এটি একটি ব্যাকএন্ড ফ্রেমওয়ার্ক। এর মাধ্যমে স্ট্যাটিক ফাইল সার্ভ করা যেতে পারে। আমাদের কে মিডলওয়্যার ব্যবহার করে এই কাজটি সম্পুর্ন করতে হবে।

>node-static:

এটি একটি লাইটওয়েট এবং সিম্পল লাইব্রেরি যা দিয়ে স্ট্যাটিক ফাইল সার্ভ করতে পারি।

>serve-static:

এটি আরেকটি স্ট্যাটিক ফাইল সার্ভ করার জন্য লাইব্রেরি। এটি অনেকগুলো ফিচার প্রভাইড করে। যেমন ক্যাশিং, কমপ্রেশন ইত্যাদি।

113 AngularJS কীভাবে NodeJS থেকে আলাদা?

AngularJS এবং NodeJS হলো Mean Stack ফ্রেমওয়ার্কেরই একটি অংশ। কীন্তু, এদের মধ্যকার কাজ এবং উদ্দেশ্য সম্পূর্ণ ভিন্ন। এদের পার্থক্যগুলো হলো,

AngularJS	NodeJS
AngularJS জাভাস্ক্রিপ্টে লিখা ফ্রন্ট-এন্ড ফ্রেমওয়ার্ক যা একটি অ্যাপ্লিকেশনের UI এবং ক্লায়েন্ট-সাইড নিয়ে কাজ করে।	অন্যদিকে, NodeJS হল একটি runtime environment, যা সার্ভার-সাইড অ্যাপ্লিকেশন তৈরিতে কার্যকর।
AngularJS হলো একটি ক্লায়েন্ট-সাইড জাভাস্ক্রিপ্ট ফ্রেমওয়ার্ক। এটি ডেভেলপারদের Model-View-Controller (MVC) আর্কিটেকচারাল প্যাটার্নের উপর ভিত্তি করে Dynamic ওয়েব অ্যাপ্লিকেশন তৈরি করার সুবিধা দিয়ে থাকে। এটি টেমপ্লেট ভাষা হিসাবে HTML ব্যবহার করারও এক্সেস দেয়।	NodeJS প্রোগ্রামারা সার্ভার-সাইড ডেভেলপমেন্টের জন্য প্রোগ্রামিং ভাষা হিসেবে জাভাস্ক্রিপ্ট ব্যবহার করে থাকে।

এসকল ভিন্নতার কারণে আর্কিটেকচার, ফাংশনালিটি, পারফরম্যান্স এবং অ্যাপ্লিকেশনের মতো জিনিসগুলির তুলনা করার সময় NodeJS এবং AngularJS এর মধ্যে স্যুইচ করা সম্ভব হয়না।

114

AngularJS কীভাবে ReactJS থেকে আলাদা?

Angular হলো একটি Google দ্বারা তৈরি এবং রক্ষণাবেক্ষণকৃত ওয়েব ফ্রেমওয়ার্ক যা ২০১০ সালে AngularJS নামে প্রথম প্রকাশিত হয়েছিল। এর যাবতীয় সুবিধাবলীর জন্যে এটি রাতারাতি সবচেয়ে জনপ্রিয় ওয়েব ফ্রেমওয়ার্ক হয়ে ওঠে।

React ফেইসবুক দ্বারা ব্যবহৃত একটি জাভাস্ক্রিপ্ট লাইব্রেরি যা ২০১৩ সালে ওপেন-সোর্স করা হয়। এই ফ্রেমওয়ার্কটি একটি ওয়েব ডেভেলপমেন্ট ধারণাকে জনপ্রিয় করে তোলে যা কম্পোনেন্ট-বেসড আর্কিটেকচার নামে পরিচিত। মোবাইল ডেভেলপমেন্টের ক্ষেত্রে React ব্যবহার করা হয়।

AngularJS ও ReactJS দুটিই জনপ্রিয় জাভাস্ক্রিপ্ট ফ্রেমওয়ার্ক হলেও তাদের বেসিক কীছু পার্থক্য রয়েছে। যেমন:

AngularJS	ReactJS
AngularJS হল একটি পূর্ণাঙ্গ MVC (Model-View-Controller) ফ্রেমওয়ার্ক।	ReactJS হল একটি লাইট ওয়েটেট লাইব্রেরি যা ভিউ লেয়ারের উপর ফোকাস করে।
AngularJS দিয়ে দ্বি-মুখী ডেটা বাইন্ডিং হয়।	React দিয়ে কেবল একমুখী ডেটা প্রবাহ হয়।
AngularJS ডিরেক্টিভ বা নির্দেশের উপর নির্ভর করে।	এদিকে React একটি কম্পোন্যান্ট-বেইসড আর্কিটেকচার ব্যবহার করে।
AngularJS এর জন্য প্রচুর Boilerplate কোড প্রয়োজন এবং এটি শেখা কঠিন হতে পারে।	React এর একটি তুলনামূলক সহজ API রয়েছে যা শিখাও সহজসাধ্য।

- Very Fast- গুগল ক্রোমের V8 জাভাস্ক্রিপ্ট ইঞ্জিনে তৈরি হওয়ায় Node.js লাইব্রেরি কোড এক্সিকী উশনে খুব দ্রুত কাজ করে।
- Highly Scalable- Node.js ইভেন্ট লুপিং-এ একটি সিংগেল থ্রেডেড মডেল ব্যবহার করে। ইভেন্ট মেকানিজম সার্ভারকে নন-ব্লকিং উপায়ে সাড়া দিতে সাহায্য করে। এর সাথে সার্ভারকে ট্রেডিশনাল সার্ভারের বিপরীতে অত্যন্ত স্কেলেবল করে তোলে যা রিকোয়েস্টগুলো পরিচালনা করার জন্য লিমিটেড থ্রেড তৈরি করে। Node.js একটি সিংগেল থ্রেডেড প্রোগ্রাম ব্যবহার করে এবং সেই প্রোগ্রাম Apache HTTP সার্ভারের মতো ট্রেডিশনাল সার্ভারের তুলনায় অনেক বেশি সংখ্যক রিকোয়েস্টে সার্ভিস প্রদান করতে পারে।
- No Buffering- Node.js অ্যাপ্লিকেশন কখনই কোনো ডেটা বাফার করে না।
- Cross-platform: Node.js Windows, macOS এবং Linux এর মতো একাধিক অপারেটিং সিস্টেমে চলতে পারে।
- Extensible: npm রেজিস্ট্রিতে উপলব্ধ প্যাকেজগুলি ব্যবহার করে Node.js এক্সটেনশন করা যেতে পারে।

115 অ্যাসিঙ্ক্রোনাস জাভাস্ক্রিপ্ট কী?

জাভাস্ক্রিপ্ট হলো একটি সিঙ্গেল-থ্রেডেড প্রোগ্রামিং ল্যাঙ্গুয়েজ, যার মানে একক সময়ে শুধুমাত্র একটি কাজ করতে পারে। অর্থাৎ, জাভাস্ক্রিপ্ট ইঞ্জিন একটি সিঙ্গেল থ্রেডে একবারে একটিমাত্র স্টেটমেন্টকে প্রসেস করতে পারে।

সিঙ্গেল-থ্রেডেড ল্যাঙ্গুয়েজগুলি কোড লেখাকে সহজ করে। কারণ এক্ষেত্রে concurrency সংক্রান্ত সমস্যাগুলি নিয়ে চিন্তা করতে হয় না। যার অর্থ হলো মেইন থ্রেডটি ব্লক না করে নেটওয়ার্ক অ্যাক্সেসের মতো দীর্ঘ সময় সাপেক্ষ কাজগুলি সম্পাদন করা সম্ভব না। অর্থাৎ, কোনো একটি দীর্ঘ বা সময় সাপেক্ষ কাজ শেষ না হওয়া পর্যন্ত আপনি মেইন থ্রেডকে দিয়ে অন্য কোনো কাজ করাতে পারবেন না।

ধরা যাক, একটি API থেকে কীছু ডেটার জন্য আপনি রিকোয়েস্ট করলেন। বিভিন্ন পরিস্থিতির উপর নির্ভর করে সার্ভারটি রিকোয়েস্টটি প্রসেস করতে কিছু সময় নিতে পারে সেক্ষেত্রে মেইন থ্রেডটি ব্লক হয়ে থাকবে সার্ভার থেকে রেসপন্স আসার জন্য। ফলে সম্পূর্ণ ওয়েব পেজটিই প্রতিক্রিয়াহীন হয়ে থাকবে।

ঠিক এই ক্ষেত্রেই অ্যাসিঙ্ক্রোনাস জাভাস্ক্রিপ্টের প্রয়োজন হয়। অ্যাসিঙ্ক্রোনাস জাভাস্ক্রিপ্ট ব্যবহার করে (যেমন: কলব্যাক ফাংশন, প্রমিজ , এবং অ্যাসিঙ্ক/এওয়েট), আপনি প্রধান থ্রেড ব্লক না করে দীর্ঘ নেটওয়ার্ক রিকোয়েস্টগুলি বা API কলের মতো সময় সাপেক্ষ কাজগুলো সম্পাদন করতে পারবেন। অর্থাৎ, জাভাস্ক্রিপ্টের মতো সিঙ্গেল থ্রেডেড প্রোগ্রামিং ল্যাঙ্গুয়েজ দিয়ে মাল্টি থ্রেডেড ল্যাঙ্গুয়েজের মতো করে কাজ করাতে পারবেন।

116

কলব্যাক হেল কী ? কোন কারণে কলব্যাক হেল দেখা দেয়?

কলব্যাক হেল হলো এমন একটি শব্দ যেখানে নেস্টেড কলব্যাকগুলি খুব জটিল এবং পড়া এবং বজায় রাখা কঠিন হয়ে পড়ে এমন পরিস্থিতি বর্ণনা করতে ব্যবহৃত হয়। এটি ঘটে যখন একটি কলব্যাক ফাংশনকে অন্য একটি কলব্যাক ফাংশনের মধ্যে কল করা হয় এবং কোডটি নেস্টেড হতে শুরু করে এবং বোঝা কঠিন হয়। এই পরিস্থিতির উদ্ভব হয় যখন সঠিক পরিকল্পনা ছাড়াই কোড লেখা হয় এবং কলব্যাকের নেস্টিং খুব গভীর হয়ে যায়, যা কোডটিকে অপঠনযোগ্য করে তোলে।

117

কলিং এবং কলব্যাক রিটার্নিং এর মধ্যের পার্থক্য কী ?

Ostad

জাভাস্ক্রিপ্টের সুন্দর কয়েকটি ফিচারের ভিতর কলব্যাক একটি। কল ব্যাকের মাধ্যমে আরো এফিসিয়েন্ট ওয়েতে আরো ডায়নামিক ভাবে কোন একটি নির্দিষ্ট কাজ সম্পাদন করা সম্ভব।

কলব্যাক কলিংঃ

কলব্যাক কলিং হলো কোন একটি ফাংশন থেকে অন্য একটি ফাংশন কে কল করা হয়। যে ফাংশন টি মুলত প্যারেন্ট ফাংশনের আর্গুমেন্ট হিসেবে পাস হয়ে আসে এবং বডিতে কল করা হয়। মুলতঃ কোন একটি নির্দিষ্ট কাজ শেষ করার পর যদি অন্য একটি কাজ করার জন্য ইচ্ছে হয় তখনই কলব্যাক কলিং এর মাধ্যমে কাজটা সম্পুর্ন করা হয়।

```
const myFunc = (anotherFunc) => {
  console.log("calling myFunc")
anotherFunc()
}
const anotherFunc =() => {
  console.log("calling anotherFunc")
}
myFunc(anotherFunc)
```

কলব্যাক রিটার্নঃ

কলব্যাক রিটার্ন হলো এমন একটি পদ্ধতি যার মাধ্যমে একটি ফাংশন কল করা হয় এবং সেই ফাংশনের কাজ শেষে নতুন একটা ফাংশন রিটার্ন করা হয়। যেমন

```
const myFunc = () => {
  console.log("calling myFunc")
    return () => {
        console.log("returned function")
    }
}
const innerFunction = myFunc()
innerFunction()
```

Ostad

118 একটি নির্দিষ্ট স্যাম্পল কোড রান করার ফলাফলগুলো কী হতে পারে ব্যখ্যা করুন।

নির্দিষ্ট স্যাম্পল কোড রান করার ফলাফলগুলো হতে পারে কয়েকটি।

১. এটি কোন কীছু আউটপুট দিতে পারে। বা এপ্লিকেশনের কোন স্টেট চেইঞ্জ হতে পারে

২. এটির যদি কোন ভুল থাকে তাহলে এটি একটি ইরর জেনারেট করতে পারে।

119

V8 কীভাবে Javascript Code কম্পাইল করে?

V8 ইঞ্জিন কয়েকটি ধাপে কোড কম্পাইল করে থাকে। ভি৮ প্রথমে জাভাস্ক্রিপ্ট কোড পার্স করে AST (Abstract Syntax Tree) তে রুপান্তর করে। তারপর সেই AST মেশিন কোডে রুপান্তর হয়। সেখান থেকে সিপিইউ কোড নিয়ে এক্সিকী উট করে থাকে।

এটি পারফর্মেন্স উন্নত করতে আরো কীছু স্টেপ ফলো করে থাকে যেমন JIT (Just In Time) compile, কোড ক্যাশিং, গার্বেজ কালেকশন ইত্যাদি।

120 Event Loop কী এবং এর বিস্তারিত বলুন।

জাভাস্ক্রিপ্টএর একদম বেসিক জিনিসপত্র গুলোর ভিতর ইভেন্ট লুপ হলো অন্যতম। ইভেন্ট লুপের মাধ্যমে সিঙ্গেল থ্রেডেট এপ্লিকেশন মাল্টি থ্রেড এর মতো কাজ করতে পারে। ইভেন্ট লুপ কয়েকটি কম্পোনেন্ট নিয়ে গঠিত। যেমন :

১.কল স্ট্যাকঃ এটি একটি LIFO (Last In First Out) স্ট্যাক। এটির মাধ্যমে ফাংশন কল ট্র্যাক রাখা হয়। যেখানে সর্বশেষ অপারেশন টি সর্ব প্রথম হয়ে থাকে।

২. ইভেন্ট কীউঃ যেখানে ওয়েব এপিআই দিয়ে সম্পুর্ণ কাজ গুলো সারিবদ্ধ ভাবে থাকে। এটি একটি FIFO (First In First Out) কী উ। যেখানে যে সর্বপ্রথম শেষ হয়েছে তাকে সর্বপ্রথম এক্সিকী উট করা হয়।
ইভেন্ট কী উ আবার দুটি রয়েছে। টাস্ক কী উ এবং মাইক্রো টাস্ক কী উ। যেখানে মাইক্রো টাস্ক কী উ এর প্রায়োরিটি বেশি থাকে এবং এখানে প্রমিজ গুলো রিজল্ভ হয়ে আসে।

৩. ইভেন্ট লুপঃ ইভেন্ট লুপ হলো একটি চলমান ক্রিয়া। যে সবসময় মেইন থ্রেড বা কলস্ট্যাক কে পর্যবেক্ষণ করে যখন কলস্ট্যাক খালি থাকে তখন ইভেন্ট কী উ থেকে ইভেন্ট নিয়ে কলস্ট্যাক এ পাঠিয়ে দেয়। এবং কলস্ট্যাক তা ফায়ার করে।

121 জাভাস্ক্রিপ্টে একটি বস্তু array নাকী না তা কীভাবে পরীক্ষা করবেন?

জাভাস্ক্রিপ্ট এ এ্যারে কেও অবজেক্ট হিসেবে রিপ্রেজেন্ট করে থাকে। তাই typeof দিয়ে চেক করলে এটি অবজেক্ট রিটার্ন করে। কোন একটি আইটেম এ্যারে কী না সেটি পরীক্ষা করতে হলে Array.isArray() মেথড কল করে আর্গুমেন্ট হিসেবে সেই এলিমেন্ট কে পাস করতে হয়। এটি একটি বুলিয়ান রিটার্ন করে।

122 Javascript-এর ক্ষেত্রে Scope কী ?

স্কোপ বলতে বুঝায় একটি ভ্যারিয়েবল কোথায় থেকে এক্সেস করা সম্ভব। মুলত জাভাস্ক্রিপ্ট এ ৪ ধরনের স্কোপ রয়েছে।

- **গ্লোবাল স্কোপ:** এখানের যেকোন ভ্যারিয়েবল একটি ফাইলের সব স্থান থেকে এক্সেসেবল।
- **ফাংশন স্কোপ:** যদি কোন ফাংশনের ভিতর কোন ভ্যারিয়েবল ডিক্লেয়ার করা হয় তার সেই ভ্যারিয়েবল কে শুধুমাত্র সেই ফাংশন থেকেই এক্সেস করা যায়। এর বাহির থেকে করা যায় না।
- **ব্লক স্কোপ:** এটি ES6 এ নতুন ব্লক স্কোপ ফিচার। এতে কোন ব্লক(দুটি {} এর ভিতর যা থাকে তাই একটি ব্লক) এর ভিতর কোন ভ্যারিয়েবল ডিক্লেয়ার করা হলে সেটি শুধু মাত্র সেই ব্লকেই এক্সেসিবিলিটি থাকে। এর বাহিরে থাকে না।
- **লেক্সিকাল স্কোপ:** প্যারেন্টে যদি কোন ভ্যারিয়েবল থাকে সেটি সবসময় তার চাইল্ড ব্লক থেকে এক্সেসেবল। এজন্য এটি লেক্সিকাল স্কোপ।

123 NodeJS-এর ফাংশনগুলোর একটি তালিকা করুন।

নোড জেএস এর অনেক গুলো গ্লোবাল ফাংশন রয়েছে। নিচে কীছু লিস্ট দেওয়া হলো।

setTimeout: এটি কোন একটি নির্দিষ্ট সময় পর কোন ইভেন্ট ট্রিগার করতে ব্যবহার হয়।

setInterval: কোন নির্দিষ্ট সময় পর পর কোন একটি নির্দিষ্ট ইভেন্ট ট্রিগার করতে এটি ব্যবহার করা হয়।

setImmediate: কোন ইভেন্ট যদি ইভেন্ট লুপের নেক্সট ইটারেশন এ রান করাতে হয় তাহলে এই ফাংশন ব্যবহার করা হয়।

clearTimeout: যদি setTimeout দিয়ে তৈরি কোন ইভেন্ট ট্রিগার হওয়ার আগেই আপনি সেটি রিমুভ করতে চান তাহলে এই ফাংশন ব্যবহার করা হয়।

clearInterval: এটি clearTimeout এর মতোই। setInterval দিয়ে যদি কোন ইভেন্ট ক্রিয়েট করা হয় সেটি ক্লিয়ার করার জন্য এই ফাংশন টি ব্যবহার করা হয়।

clearImmediate: setImmediate দিয়ে যদি কোন ইভেন্ট কে রিমুভ করতে চাই তাহলে এই ফাংশন ব্যবহার করা হয়।

console.log: কনসোলে কোনকী ছু প্রিন্ট করতে হলে এই ফাংশন ব্যবহার করা হয়।

console.error: কনসোলে এরর প্রিন্ট করতে ব্যবহার হয়।

console.warn: কনসোলে ওয়ার্নিং দিতে এটি ব্যবহার করা হয়।

console.info: কোন ইনফরমেশন প্রিন্ট করতে হলে এই ফাংশন ব্যবহার করা হয়।

console.dir: জাভাস্ক্রিপ্ট অবজেক্ট রিটার্ন করে। এটি দিয়ে ডিবাগ করা হয়।

Buffer: কোন বাফার ক্রিয়েট করতে হলে এই মেথড ব্যবহার করা হয়।

require: কোন মডিউল ইমপোর্ট করতে হলে এই মেথড টি ব্যবহার করা হয়।

process.exit: এই মেথড দিয়ে কোড থেকে এক্সিট করা যায়। নির্দিষ্ট কোন এক্সিট কোড দিয়ে

124

আপনার জানা মতে NodeJS-এর ডেভেলপমেন্টে ব্যবহৃত IDE-গুলো কী কী?

নোড জেএস ডেভেলপমেন্ট এর পপুলার IDE গুলো হচ্ছে।

- **VS Code:** সবচেয়ে পপুলার নোড জেএস ডেভেলপমেন্ট IDE. এটি লাইটওয়েট, এটি প্রচুর পরিমাণ এক্সটেনশন সাপোর্ট দেয়।
- **WebStrom:** ফুল ফিচারড IDE, যা ওয়েব ডেভেলাপমেন্ট করার উদ্দেশ্য তৈরি করা হয়েছে। এটি এডভান্স কোন কম্প্লিশন, ডিবাগিং সাপোর্ট দেয়।
- **Atom:** এটম হলো হ্যাকেবল টেক্সট এডিটর। যা হাইলি কাস্টমাইজেবল।
- **Sublime text:** লাইটওয়েট এবং ফাস্ট টেক্সট এডিটর। অনেকগুলো প্লাগইন ও প্যাকেজের মাধ্যমে এটি নোডজেএস ডেভেলপমেন্ট ফিচারাইজড করে।
- **IntelliJ IDEA:** এটি JetBrains ডেভেলপড একটি IDE। এটাতে এডভান্স ডিবাগিং, কোড এনালাইসিস যুক্ত রয়েছে।

আরো অনেকগুলো IDE রয়েছে। যেমন: NetBeans, Brackets, cloud9, codeAnywhere ইত্যাদি।

125

AngularJS-এর ব্যবহার কাজকে কতটুকু সহজ করে বলে আপনি মনে করেন?

যেহেতু এঙ্গুলার জেএস একটি লার্জ ফ্রেমওয়ার্ক তাই এটি কাজকে সহজ করে।

- এঙ্গুলার জেএস টাইপস্ক্রিপ্টে তৈরি করা। ফলে এটির ডেভেলপার ইফিসিয়েন্সি বেশি।
- এই এপ্লিকেশনের একটি ওয়েল ডিফাইন এবং ওয়েল অর্গানাইজ স্ট্রাকচার রয়েছে। ফলে কাজ অনেক সহজ হয়।
- এটির লার্জ একটি কমিউনিটি রয়েছে। ফলে যেকোন প্রবলেম দ্রুতই সমাধান সম্ভব।

126

Linear Search এবং Binary Search-এর মধ্যকার পার্থক্যগুলো তুলে ধরুন।

লিনিয়ার সার্চ:

লিনিয়ার সার্চ হলো এমন একটি এলগরিদম যেখানে ডিজেয়ার্ড ইলিমেন্ট না পাওয়া পর্যন্ত একী এ্যারের প্রতিটি এলিমেন্ট কে চেক করে থাকে। এটি লিস্টের শুরু থেকে টার্গেট ইলিমেন্ট এর সাথে প্রতিটি এলিমেন্ট কম্পেয়ার করতে থাকে। যদি টার্গেট ইলিমেন্ট পাওয়া যায় তখন সার্চ বন্ধ হয়ে যায়। এর কম্প্লিক্সিটি হলো O(n)

বাইনারি সার্চ:

বাইনারি সার্চ একটি ইফিসিয়েন্ট সার্চ এলগরিদম।তবে এর জন্য এ্যারেটি এসেন্ডিং কিংবা ডিসেন্ডিং অর্ডারে সাজানো থাকতে হয়। এটি বারবার সাজানো এ্যারেকে দুটি ভাগে ভাগ করতে থাকে এবং তুলনা করতে থাকে। যদি মাঝের ইলিমেন্ট টি টার্গেট ইলিমেন্টের সাথে মিলে তখন এটির সার্চ বন্ধ হয়ে যায় এবং শুরুতে চলে আসে। যদি মাঝের ইলিমেন্ট টি টার্গেট ইলিমেন্টেররদ চেয়ে বড় হয় তাহলে সেটি বাম ট্রিতে চলতে থাকে। যদি ছোট হয় তাহলে ডানের টা তে চলতে থাকে। এটি সার্চের সময় যে অর্ধেক অপ্রয়োজনীয় সেটি তে ইরেজ করে দেয় ফলে পারফর্মেন্স অনেক ভালো হয়।

127

আপনার জানা মতে NodeJs, AJAX এবং jQuery এর মধ্যকার পার্থক্যগুলো কী কী? একটি ধারণা দিন।

নোডজেএস হলো একটি জাভাস্ক্রিপ্ট রানটাইম। জাভাস্ক্রিপ্ট কোড যখন শুধুমাত্র ব্রাউজার ছাড়া ব্যবহার করা যেতো না তখন রায়ান ডাল নামে এক ব্যাক্তি এটার সূচনা করেন। যেখানে জাভাস্ক্রিপ্ট ব্রাউজার এর বাহিরেও রান করা সম্ভব হয়।

অন্যদিকে AJAX হলো সার্ভারে রিকুয়েষ্ট পাঠানোর একটি টেকনিক । এর ফলে ওয়েব পেজ রিলোড করা ছাড়াই সার্ভার হতে ডেটা এনে শো করা যায়। এতো করে ওয়েব পেজ আরো ইন্টারেক্টিভ হয়ে উঠে।

আর জেকুয়েরী হলো একটি জাভাস্ক্রিপ্ট লাইব্রেরি যার মাধ্যমে ডম ম্যানিপুলেশন এবং ইউটিলিটি ফাংশন পাওয়া যায়।

128

Containerization কী? এ সম্পর্কে আপনি কতটুকু অবগত?

Containerization বলতে বুঝায় একটি সফটওয়্যার প্যাকেজিং টেকনিক। কোন একটি এপ্লিকেশন এর সমস্ত ডিপেন্ডেন্সি গুলো কে একসাথে করে একটি ইউনিটে নিয়ে আসা হয় যাকে বলা হয় কনটেইনার। এই কনটেইনার টি যেকোন মেশিনে চলতে পারে। রিগার্ডলেস হার্ডওয়্যার এবং অপারেটিং সিস্টেম।

কনটেইনার গুলো ভার্চুয়াল মেশিনের মতো এটি একটি আইসোলেটেড এনভায়রনমেন্টে রান করা হয়। কী স্তু এই কনটেইনার গুলো ভার্চুয়াল মেশিনের চেয়ে লাইটওয়েট এবং ফাস্ট।

বর্তমানে ক্লাউড কম্পিউটিং এ কনটেইনার ধারণা টি ব্যাপকভাবে ব্যবহার করা হচ্ছে। এতে করে অনেকগুলো সুবিধা পাওয়া যায়। যেমন।

- **ইফিসিয়েন্সি:** ভার্চুয়াল মেশিনের চেয়ে এই কনটেইনার গুলো লাইটওয়েট এবং ফাস্ট। ফলে রিসোর্স কম ব্যবহার করে বেশি কাজ সম্পুর্ণ করা সম্ভব হয়।
- **স্কেলেবিলিটি:** কনটেইনার গুলো প্রয়োজন অনুসারে যেকোন দিকে আপগ্রেড বা ডাউনগ্রেড করা সম্ভব হয়।
- **ম্যানেজেবিলিটি:** কনটেইনার ব্যবহার করার ফলে এগুলো ম্যানেজ, আপডেট, ট্রান্সফার ইত্যাদি করা সুবিধা হয়। যেহেতু এই কনটেইনার সব এনভায়রনমেন্টে চলতে পারে।

কীছু জনপ্রিয় কনটেইনারাইজেশন টেকনোলজিস :

- **ডকার:** এটি হলো সবচেয়ে জনপ্রিয় একটি কনটেইনার টেকনোলজি। এটি ওপেনসোর্স এবং অনেক বড় কমিউনিটি রয়েছে। ফলে এটি পরিচালনা সহজ।
- **কুবারনেটিস:** এটিও ওপেনসোর্স কনটেইনার অর্কেস্ট্রেশন সিস্টেম। যা গুগল দ্বারা তৈরি।
- **আমাজন ইলাস্টিক কনটেইনার ECS:** এটি AWS এর একটি সার্ভিস যাতে ডকার চালানো খুবই সহজ।

129

Test Pyramid কী এবং এর মূল কাজগুলো কী কী? HTTP APIs পরীক্ষা করার সময় আপনি কীভাবে টেস্ট পিরামিড বাস্তবায়িত করবেন?

টেস্ট পিরামিড হলো একটি মেথোডোলজি। যেখানে কয়েক ধরনের টেস্টিং একসাথে করা হয়। টেস্ট পিরামিড ৪টি টেস্টিং ফেজ এর ভিতর দিয়ে যায়। যেখানে

- **Unit Test:** ইউনিট টেস্ট হলো কোড লেভেলের টেস্ট। যেখানে প্রতিটি কোডের অংশ গুলো পৃথকভাবে টেস্ট করা হয়। এটি সবচেয়ে নিচের লেয়ার এবং সবচেয়ে বেশি কেস থাকে।
- **Integration Test:** ইন্টেগ্রেশন টেস্ট হলো ইউনিট টেস্টের উপরের লেয়ার। যেখানে কয়েকটি মডিউল একসাথে করে টেস্ট করা হয়।
- **End to End Test:** এটি হলো থার্ড লেয়ার। এখানে পুরো এপ্লিকেশন কে টেস্ট করা হয় যাতে সম্পুর্ন সিস্টেম যেভাবে চাওয়া হয় সেভাবে রান করে কী না।
- **UI Test:** UI টেস্ট বা ইন্টারফেস টেস্ট হলো গ্রাফিকাল ইউজার ইন্টারফেসে যেভাবে চাই সেভাবে বিহেব করে কী না। এটি সবচেয়ে উপারের লেয়ার এখানেই টেস্ট ফেজ সমাপ্তি ঘটে।

130 TypeScript-এ Classes এবং Interfaces-এর মধ্যকার পার্থক্য সম্পর্কে বলুন

টাইপস্ক্রিপ্টে ক্লাস হলো কোন একটি অবজেক্ট এর ব্লুপ্রিন্ট। কোন একটি অবজেক্ট কীভাবে তৈরি হবে তার সমস্ত কীছু থাকে একটি ক্লাসের ভিতর।

অপর দিকে ইন্টারফেস হলো ভ্যালিডেটর। যেখানে বলা হয়ে থাকে কী কী প্রপার্টি এবং মেথড কোন একটি অবজেক্ট এ অবশ্যই থাকতে হবে।

ক্লাস দিয়ে আমরা অবজেক্ট তৈরি করি। কীন্তু ইন্টারফেস দিয়ে অবজেক্ট তৈরি করতে পারি না।

ক্লাস কে এক্সটেন্ডেড করা সম্ভব কী ন্তু ইন্টারফেস কে তা করা সম্ভব হয় না।

131

TypeScript-এ Decorator বলতে আপনি কী বোঝেন?

ডেকোরেটর হলো স্পেশাল টাইপ একটি ফাংশন যেটি কোন ক্লাস, মেথড, প্রপার্টি ইত্যাদির বিহেভিয়ার মডিফাই করতে পারে। এটি @ সিম্বল দিয়ে শুরু করা হয়।

ডেকোরেটর যে কাজ গুলো করতে পারে।

- কোন ক্লাসের মেটাডেটা এড করতে পারে।
- কোন ক্লাসের বিহেভিয়ার পরিবর্তন করতে পারে।
- কোন কোড জেনারেট করতে পারে।

132 Cross-Site Scripting ব্যখ্যা করুন।

ক্রস সাইট স্ক্রিপ্টিং হলো একটি সিকী উরিটি ভালনারিবিলিটি। যা কোন হ্যাকারের ম্যালিশিয়াস কোড রান করতে সাহায্য করে থাকে। ক্রসসাইট স্ক্রিপটিং দিয়ে কুকী , সেশন চুরি করতে পারে। এমন কী পুরো ব্রাউজার টিও নিয়ন্ত্রণে নিতে পারে।

এই এটাক দুই ধরনের হতে পারে।

›reflected XSS:

হ্যাকার URL এর মাধ্যমে ম্যলিশিয়াস কোড এড করে ইউজার কে দেখায়। ইউজার সেখানে ব্রাউজ করলে XSS রান হয়।

›stored XSS:

এই ম্যলিসিয়াস কোড সার্ভারে থাকে। ইউজার যখন ব্রাউজ করে তখন সেটি রান হয় এবং তথ্য ছিনিয়ে নেয়।

133 AOT কী? এটি ব্যবহারের কয়েকটি সুবিধা বলুন।

AOT (Ahead of Time) একটি কমপাইলর টেকনিক যেখানে কোড রান হওয়ার আগেই মেশিন কোডে রুপান্তর হয়। এটি ট্রেডিশনাল JIT কম্পাইলার এর বিপরীত। যেখানে কোড রানটাইমে কম্পাইল হয়।

এর কয়েকটি সুবিধা রয়েছে:

- এটি পারফরম্যান্স বৃদ্ধি করে। কারণ এটি কোড কম্পাইল করতে বেশি সময় পেয়ে থাকে JIT কম্পাইলারের তুলনায়।
- যেতেতু কোড একবারে রান হয় তাই ম্যালিসিয়াস কোড রান হওয়ার পসিবিলিটি কম।
- মেমোরি ব্যবহার কমায়। কেননা এটি ইনলাইন ফাংশন কল করতে পারে। এবং আনইউজড কোড স্কিপ করে থাকে।

```
function handleRequest(req) {
  // Perform a non-blocking operation, such as reading a file from disk.
  fs.readFile('file.txt', (err, data) => {
    if (err) {
      // Handle the error.
    } else {
      // Do something with the data.
    }
  });

  // Continue handling other requests.
  nextRequest();
}
```

134 CSS-এ Grid System সম্পর্কে আপনি কতটুকু জানেন?

CSS গ্রিড লে-আউট হলো একটি 2D ডিজাইন সিস্টেম। যা দিয়ে কোন কমপ্লেক্স ডিজাইন দ্রুত এবং সহজেই করা যায়। গ্রিড লে-আউট ব্যবহার করতে হলে আমাদের মার্কআপ দুটি ভাগে ভাগ করতে হবে। একটি কনটেইনার ও কী ছু চাইল্ড আইটেম। কনটেইনার কে display প্রপার্টি grid দিতে হবে এবং grid template column ও grid template row বলে দিতে হবে। তাহলেই চিলড্রেন গুলো সেই প্লেসে বসে যাবে।

135

ওয়েব ডিজাইনে গ্রিড সিস্টেম কী ? আপনার জানা মতে সবচেয়ে বেশি ব্যবহৃত গ্রিড সিস্টেম কোনগুলো?

CSS গ্রিড লে-আউট হলো একটি 2D ডিজাইন সিস্টেম। যা দিয়ে কোন কমপ্লেক্স ডিজাইন দ্রুত এবং সহজেই করা যায়। গ্রিড লে-আউট ব্যবহার করতে হলে আমাদের মার্কআপ দুটি ভাগে ভাগ করতে হবে। একটি কনটেইনার ও কীছু চাইল্ড আইটেম। কনটেইনার কে display প্রপার্টি grid দিতে হবে এবং grid template column ও grid template row বলে দিতে হবে। তাহলেই চিলড্রেন গুলো সেই প্লেসে বসে যাবে।

136

Non-obstructing বা Non-blocking ফিচারটি সম্পর্কে বিস্তারিত বলুন।

নন ব্লকিং ফিচার বলতে বুঝায় নোডজেএস রানটাইম একসাথে মাল্টিপল রিকুয়েষ্ট হ্যান্ডেল করতে পারে এক্সিকী উশন কনটেক্সট কে ব্লক করা ছাড়াই। এটি ইভেন্ট লুপ দিয়ে সম্ভব করা হয়েছে। যেটি সিঙ্গেল থ্রেডেড যা এসিঙ্ক্রোনাস টাস্ক ম্যানেজ করে থাকে।

যখন নন ব্লকিং অপারেশন রান হয় তখন এসিঙ্ক্রোনাস টাস্ক টি কী উতে চলে যায়। এবং এক্সিকী উশন কনটেক্সট এ অন্যান্য টাস্ক রান হয়। যখন কনটেক্সট খালি থাকে তখন ইভেন্ট লুপ সিম্পলি মেইন থ্রেডে কলব্যাক টি রান করিয়ে দেয়।

নন ব্লকিং এর জন্য নোড জেএস হাই ভলিওম রিকুয়েষ্ট হ্যান্ডেল ও অপারেশন করতে পারে। যেখানে পারফরম্যান্স এ কোন হ্যাম্পার হয় না।

137 I/O বলতে কী বোঝায় ব্যখ্যা করুন।

বেশিরভাগ সফটওয়্যার সিস্টেমে যেকোনো রকম সিস্টেম কল যেমন — ফাইল অপারেশন, ডাটাবেজ কুয়েরী এসব ব্লকিং অর্থাৎ এরকম অপারেশন চলা অবস্থায় মুল প্রোগ্রামের এক্সিকী উশন বন্ধ থাকে যতক্ষণ না পর্যন্ত ওই কলের অপর প্রান্তের কাজ শেষ হয় এবং একটা রেজাল্ট রিটার্ন হয়। এটাকে বলতে পারেন ব্লকিং এবং synchronous প্রোগ্রামিং। Node.js এর একটা অন্যতম উদ্দেশ্য হচ্ছে যতটা সম্ভব নন-ব্লকিং I/O অপারেশন ইউটিলাইজ করা। এর জন্য সে ইভেন্ট লুপ এবং আলাদা থ্রেড পুল ব্যবহার করে।

নোডজেএস এ যখন রিকোয়েস্ট আসে সেগুলোকে একটি থ্রেড থেকেই সার্ভ করে। এই মুল থ্রেডে চলা আপনার প্রোগ্রামের কোড কীন্তু ঠিকী synchronously এক্সিকী উট হয় কীন্তু যখনই সেখানে একটি সিস্টেম কল ঘটে তখনি Node.js সেটাকে ইভেন্ট লুপে পাঠিয়ে দেয় যার সাথে থাকে একটি কলব্যাক ফাংশনও। এতে করে মুল থ্রেড কখনই ফ্রিজ হয়ে থাকবে না এবং নতুন নতুন রিকোয়েস্ট নিতেই থাকবে। যখনি ইভেন্ট লুপের দায়িত্বে থাকা ব্লকিং অপারেশনের কাজ শেষ হয়ে যাবে তখনি ইভেন্ট লুপ সেই কলব্যাক ফাংশনকে এক্সিকী উট করার জন্য V8 এর কাছে পাঠিয়ে দিবে। এই কলব্যাকের কাছে অপারেশন এর রেজাল্টও থাকবে। এতে করে মুল থ্রেড বা মুল প্রোগ্রাম ব্লকিং অপারেশনের কারনে ব্লক হয় না। এটিই হচ্ছে I/O অপারেশন।

138 Control stream work কী?

নোডজেএস এ কন্ট্রোল স্ট্রিম হলো এমন টাইপ স্ট্রিম যা দিয়ে আমরা ডেটা ফ্লো কনট্রোল করতে পারি। ডেটা ট্রান্সফার এর ক্ষেত্রে থ্রোটোলিং স্পিড, বাফারিং ডেটা, পুশিং ইত্যাদি করার জন্য ব্যবহার করা হয়ে থাকে। আমরা stream.ControlStream দিয়ে এই স্ট্রিম ক্রিয়েট করতে পারি।

139 Occasion Driven প্রোগ্রামিং কী?

Occasion driven programming হলো একটি সফটওয়্যার ডেভেলাপমেন্ট মেথোডোলজি যেখানে একটা নির্দিষ্ট সময়ে ইউজারের নিড এবং গোলের দিকে ফোকাস করা হয়।

এই মেথোডোলজি তে ভিন্ন ভিন্ন সময়ে ভিন্ন গোল এর উপর ডিপেন্ড করে। সফটওয়্যার ডেভেলাপমেন্টে এটি একটি নিউ মেথোডোলজি। কীন্তু ইউজার পার্সপেক্টিভ থেকে এটি একটি গুরুত্বপূর্ণ মেথড।

এর অনেকগুলো বেনিফিট রয়েছে। যেমন

- ইউজারের এক্সপেরিয়েন্স বৃদ্ধি করে।
- ইউজারের প্রোডাক্টিভিটি স্যাটিসফোশন, এনগেজমেন্ট বৃদ্ধি পায়।
- ইভেক্টিভনেস বৃদ্ধি পায়। কেননা সফটওয়্যার ইউজার এর উপর ডিপেন্ড করে এটি তৈরি করা হয়।

140 Sharding-এর প্রসেসগুলো ব্যাখা করুন।

Sharding হলো মংগোডিবি ডেটা ডিস্ট্রিবিউটিং একটি মেথড।

মংগোডিবি তে sharding এর কয়েকটি প্রসেস রয়েছে।

› Setup shard cluster:

ডেটা সার্ডিং এর জন্য প্রথমে ক্লাস্টার সেটআপ করতে হবে। সার্ড সার্ভার, কুয়েরি রাউটারস, কনফিগারেশন ইত্যাদি এর মুল কাজ।

›Configure Sharding:

ক্লাস্টার সেটআপ শেষ হলে সার্ড কী চুজ করতে হবে। সার্ড কী হলো এক বা একাধিক ফিল্ডের কম্বিনেশন যা দিয়ে ডিটারমাইন করা যাবে কীভাবে সার্ড গুলোর ভিতরে ডেটা ডিস্ট্রিবিউট হবে।

›Enable Sharding for DB:

sh.enableSharding() কমান্ডের মাধ্যমে সার্ডিং এনাবল করতে হবে।

›Create Shard Collection:

এরপর কাজ হলো সার্ডিং কালেকশন তৈরি করা যেখানে ডেটা স্টোর হবে। sh.shardCollection() মেথডের মাধ্যমে এটি করা হয়।

›Data distribution:

ডেটা ইনসার্ট করার পর মংগোডিবি ব্যালেন্সার ডেটা ডিস্ট্রিবিউট করবে ডিপেন্ডস অন সার্ড কী।

141 ডিপেন্ডেন্সি ইনজেকশন সম্পর্কে বিস্তারিত ধারণা দিন।

ডিপেন্ডেন্সি ইনজেকশন হলো একটি ডিজাইন প্যাটার্ন। যেখানে কোন ডিপেন্ডেন্সি সরাসরি ক্লাসের ভিতর তৈরি না করে একটি এবস্ট্রাকশন ক্লাস তৈরী করা হয়। সেখানে ইনজেকশন করা হয়। এটি করার কারণে যে সুবিধা পেতে পারি। যেমন:

- ফ্ল্যাক্সিবল ক্লাস লিখতে সুবিধাজনক।
- সহজে কোড টেস্ট করা সম্ভব হয়।
- বয়লারপ্লেট কোড কমে আসে।
- রিডেবিলিটি বাড়ে।