2024年12月04日

🔵 #DPM/UM Check resource

#DPP/Nekonote

-

? Cần resource ở những vị trí nào?

#DPM/UM Presales - Estimate WP

#DPM/UM Internal tool - Sửa file resources 2024

Note

- Bị lỗi khi fetch database

#DPM/UM Phỏng vấn ReactJS

Note

- Chiều thứ 4 (4/12/2024)
- Câu hỏi:

  - null và undefined
  - Viết css để tạo container chứa content ở LP
  - Các cách để lấy giá trị từ một object
  - Code kinh nghiệm chuẩn pixel
  - closure và ứng dụng
  - JSX là gì? khác với HTML như thế nào?
  - props vs state
  - Hooks, tại sao cần hooks
  - Câu hỏi 11: Functional components có lifecycle không?

  Trả lời:

Functional components không có lifecycle methods như class, nhưng có thể sử dụng useEffect để mô phỏng các hành vi tương tự.

- Làm thế nào để xử lý sự kiện trong React?
- key, tại sao dùng key
- Các phương pháp quản lý input data trong form, các cách để validate chúng
- ContextAPI là gì? Khi nào dùng Context, khi nào dùng Redux
- Virtual DOM hoạt động như thế nào? Điều gì xảy ra khi bạn gọi `setState` hoặc `useState`?
- setState là đồng bộ hay bất đồng bộ? tại sao? Nếu cần thực hiện một hành động sau khi setState?
- Dependency Array trong `useEffect`
- `props.children` là gì? Nó hoạt động như thế nào?
- React's `StrictMode` là gì? Điều gì xảy ra nếu bạn sử dụng nó?
- CSS Positioning (absolute, relative, fixed)
  - Viết một hàm tính tổng các số từ 1 đến n
  - Outsource nên có những lúc dự án sẽ yêu cầu ngôn ngữ/framework trái tay, quan điểm của em như thế nào
- git pull và git fetch
- Trường hợp push nhầm 1 commit lên remote
- Khi nào dùng git merge, khi nào dùng git rebase, khi nào không nên dùng git rebase
- SSR (Server-Side Rendering) và SSG (Static Site Generation)
- Thế nào là CI/CD
- Tại sao cần review code
- Trường hợp phát hiện lỗi production thì em làm gì?
- Waterfall Yêu cầu → Thiết kế → Phát triển → Kiểm thử → Triển khai → Bảo trì.
  - Giai đoạn nào trong Waterfall dễ gây ra lỗi lớn nhất? Tại sao?
    - Giai đoạn dễ gây lỗi nhất là **Requirement Analysis**.
    - Lý do: Nếu yêu cầu không được thu thập hoặc hiểu đúng, toàn bộ thiết kế và phát triển sau này có thể sai lệch, dẫn đến chi phí sửa chữa rất cao.

- Em quản lý team dự án như thế nào trong vị trí frontend leader
- Bạn đã áp dụng nguyên tắc OOP hoặc mẫu thiết kế nào trong các dự án không?
- Image những hiểu biết của em về IMAGE trong NextJS
- Trong CV bạn đề cập "Research and develop advanced functions". Bạn có thể mô tả một chức năng nâng cao mà bạn đã phát triển không?

Khi làm việc nhóm, bạn sẽ làm gì nếu có xung đột về ưu tiên công việc?

- Trong trường hợp nhiều deadline trùng lặp, bạn đã ưu tiên công việc thế nào?

Note

Câu hỏi: Debouncing và Throttling là gì? Bạn đã áp dụng chúng như thế nào trong ReactJS?
Trả lời mẫu:

- **Debouncing**: Giới hạn số lần gọi hàm bằng cách chỉ thực hiện hàm sau một khoảng thời gian. Dùng trong search bar để giảm số lần gọi API.
- **Throttling**: Giới hạn số lần thực hiện hàm trong một khoảng thời gian. Dùng trong scroll events hoặc resize events.

Note

Câu hỏi: Làm thế nào bạn tối ưu hóa việc render trong React khi sử dụng thuật toán?
Trả lời mẫu:
Sử dụng React.memo để tránh render lại không cần thiết.
Kết hợp useMemo và useCallback để tối ưu hóa các hàm xử lý phức tạp.

#DPM/UM Daily MTG

Note

- Mọi người báo cáo:

- Hôm qua làm task gì
- Hôm nay làm task gì
- Issues

- Team leader
  - Issues về mặt kỹ thuật
    - Assign cho ThắngLV, cùng Cường phân tích vấn đề và đưa ra hướng giải quyết, sau đó assign cho các bạn khác đối ứng. Trường hợp có vấn đề nằm ngoài khả năng team tự giải quyết như (cần hỏi hay request gì ở KH) thì mình assign lại cho PM
  - Review code
    - ThắnglV làm checklist
      - Tham khảo google style guide
    - Thắng Review chính, CườngNH với Thắng review chéo (Suggest như thế, còn lại tuỳ mn define)
    - Có comment thì mình comment trực tiếp vào pull request

Tasks