OK. Can you tell me the main Difference between react and angular?

Which is the landing page.in react So as soon as you open a create React app project.

Which is that one component that will render at the beginning which is that one page.

 Can you tell me the different functionalities of a package dot Jason file? What in all does the package dot Jason file contain?

Can you tell me how you go about routing in React? From one page to another page. And which is the package used for that (React router Dom, react router Dom, we use a browser router.

Attribute in app Dot JS itself. We like you know, we specify the parent routes and the child routes as well inside that OK, no no problem with the with the paths we specify.

)

Can you tell me what hooks are?

Immutable means.

What is the use of use effect hook?

What do you mean by JSX?

In just just tell me in one line there is a state called number how do you define state in react, In terms of coding how to would you define a number as a state

If number is a state can I update number directly by assigning a value like number= 5

Can you tell me the package that is used to make API calls in react?

What do you mean by lazy loading?

Which is the default one. (lazy loading or eager loading)

So for example now there is a file that needs to be uploaded by the user So for example, it is like a photo of me, OK, one for some. For some application I need to upload my photo. Just tell me the high level development steps. Including react,node, whatever. You can use whatever services you want to use, Ok So just tell me for file upload, how will you go about it?

(What will you store in DB? What will you store in DB?,  I'll tell you. So I upload a file, I upload my photo and I just need to show back my photo. I just need one more page. I need to show that uploaded photo. That's all.)

If there are 100 photos In s3 buckeet now ? There are hundred photos. How will you look at my photo?

How to involve in the database

How Will you send raw image files?  I mean, I'm just talking about image files, but whatever it is, if it's apdf file, text file, whatever PPT file, video file, whatever, right, there'll be multiple different format the file. Can you send that in an API? And if so? How?

Where is you are where you are. I don't have the URL. I have a raw image file in my local laptop or desktop. There is no URL.

Imagine the application is not about photos. What if it's about one movie I need to upload a movie, one big movie, 2 1/2 hours movie. It's like some 3 GB file. Can I send that in an API?

And how we upload files to S3?

I'm telling you to wait. I am not. There are no 1000. Users are not there 1 user. I am the only user in the entire application. If there are 1000 users, OK agreed. Subsub might be of little help.

 there is something called an unsigned URL which you can give which you can configure in S3 do you know about it ?

Unsigned URL means you do not have to upload the files directly. I mean you don't have to from the front end, it should it. It does not need to go to the back end and then from back end to S3 not not required from front end directly. It can go to S3.

**React Core Concepts**

1. What are the main differences between React and Angular? What are the advantages and disadvantages of each?
2. In a Create React App project, which component renders first and how is it configured?
3. Can you explain the purpose and contents of package.json? What configurations and dependencies does it typically contain?
4. Explain React Router DOM implementation:
   * How do you set up routing in a React application?
   * How do you configure parent and child routes?
   * What is BrowserRouter and how is it implemented?
5. Hooks in React:
   * What are React hooks and why were they introduced?
   * Explain useState hook with an example
   * What is the purpose of useEffect hook?
   * What are the different dependency scenarios in useEffect?
6. State Management:
   * How do you define state using useState hook?
   * Why can't we update state directly by assignment?
   * What is immutability and why is it important in React?
7. JSX:
   * What is JSX and how does it work?
   * What are the rules and limitations of JSX?

**API and Data Handling**

1. Which packages are commonly used for making API calls in React?
2. Explain lazy loading:
   * What is lazy loading and how is it implemented?
   * What's the difference between lazy loading and eager loading?
   * When should you use lazy loading?

**File Upload Implementation**

1. Explain the complete flow of file upload functionality:
   * How would you implement file upload from frontend to backend?
   * What considerations should be made for large file uploads?
   * How would you handle different file types?
2. AWS S3 Related:
   * How do you integrate S3 for file storage?
   * What is a pre-signed URL and when would you use it?
   * What is an unsigned URL in S3 and how does it differ from pre-signed URLs?
3. Database Design for File Storage:
   * What information would you store in the database for uploaded files?
   * How would you handle file metadata?
   * How would you implement file retrieval?
4. Performance Considerations:
   * How would you handle large file uploads (like video files)?
   * What strategies would you use for optimal file transfer?
   * How would you implement progress indicators for file uploads?

**System Design**

1. Design questions:
   * How would you design a system to handle multiple file uploads?
   * How would you implement file type validation?
   * How would you handle error cases in file uploads?
2. Security Considerations:
   * What security measures would you implement for file uploads?
   * How would you handle file access permissions?
   * How would you prevent malicious file uploads?

**Follow-up Questions**

1. Advanced Concepts:
   * How would you implement chunked file upload?
   * What's the difference between multipart/form-data and base64 encoding?
   * How would you handle file compression before upload?
2. Scaling Considerations:
   * How would your implementation change with 1000+ concurrent users?
   * What caching strategies would you implement?
   * How would you handle rate limiting?

**React Framework Fundamentals**

1. Compare React and Angular:
   * What are the core architectural differences?
   * How does the learning curve compare?
   * Which scenarios would you choose one over the other?
2. React Project Structure:
   * In a Create React App, which is the landing page/component?
   * What is the entry point of a React application?
   * How is the initial rendering configured?
3. Package.json Deep Dive:
   * What are the essential fields in package.json?
   * Explain the difference between dependencies and devDependencies
   * What do the version numbers (^, ~) indicate?
   * How do scripts work in package.json?
   * What is the purpose of the "browserslist" configuration?

**Routing**

1. React Router Implementation:
   * How do you set up React Router DOM?
   * What is BrowserRouter and where should it be placed?
   * How do you define:
     + Parent routes?
     + Child routes?
     + Nested routes?
   * What are route parameters and how do you handle them?
   * How do you implement protected routes?

**React Hooks and State Management**

1. React Hooks Overview:
   * What problem did hooks solve?
   * Why were class components replaced with hooks?
   * What are the rules of hooks?
2. State Management:
   * How do you define a state variable using useState?

javascript

Copy

*// Code example: How would you write this?*

const [number, setNumber] = useState(0);

* + Why can't you update state directly (number = 5)?
  + What is the correct way to update state?
  + When do you need useState vs useReducer?

1. UseEffect Deep Dive:
   * What is the purpose of useEffect?
   * What are the different dependency array scenarios?
   * How do you handle cleanup?
   * What are common useEffect pitfalls?
2. Immutability:
   * What is immutability in React?
   * Why is immutability important?
   * How do you maintain immutability with arrays and objects?

**JSX and Components**

1. JSX Understanding:
   * What is JSX and how does it work?
   * What are JSX limitations?
   * How does JSX differ from HTML?
   * What are the rules for writing JSX?

**API Integration**

1. API Implementation:
   * Which packages are commonly used for API calls?
   * Compare axios vs fetch
   * How do you handle API errors?
   * How do you implement API caching?
2. Lazy Loading:
   * What is lazy loading?
   * How does it differ from eager loading?
   * Which is the default and why?
   * When should you implement lazy loading?
   * How do you implement code splitting with lazy loading?

**File Upload Implementation**

1. Basic File Upload Flow:
   * How do you handle file input in React?
   * How do you validate file types?
   * How do you handle multiple file uploads?
2. File Upload Architecture: Scenario: Implement a profile photo upload feature
   * Frontend implementation steps
   * Backend handling
   * Storage solutions (S3 vs local storage)
   * Database design considerations
3. Database Design for Files:
   * What information should be stored in the database?
   * How do you store file metadata?
   * How do you handle file relationships?
4. AWS S3 Integration:
   * How do you upload files to S3?
   * What is a pre-signed URL?
   * What is an unsigned URL?
   * When would you use each approach?
5. Large File Handling: Scenario: Uploading a 3GB movie file
   * How would you handle large file uploads?
   * What are the API limitations?
   * How would you implement chunked upload?
   * How would you show upload progress?
6. File Transfer Methods:
   * How do you send raw image files via API?
   * What are the different approaches to file transfer?
   * Compare:
     + multipart/form-data
     + base64 encoding
     + binary transfer
   * When would you use each approach?

**Advanced Scenarios**

1. Performance Optimization:
   * How would you optimize file uploads for:
     + Multiple concurrent users?
     + Large files?
     + Different file types?
2. Security Considerations:
   * How do you implement secure file uploads?
   * How do you prevent malicious file uploads?
   * How do you handle file access permissions?
3. Error Handling:
   * How do you handle upload failures?
   * How do you implement retry mechanisms?
   * How do you handle network interruptions?

**System Design Scenarios**

1. Scale Considerations: Scenario: System needs to handle 1000+ concurrent users
   * How would you modify your implementation?
   * What caching strategies would you implement?
   * How would you handle rate limiting?
2. File Management System: Design a complete file management system with:
   * Upload capabilities
   * File organization
   * Search functionality
   * Access control
   * Version control

**Best Practices**

1. Development Guidelines:
   * What are React best practices for:
     + Component structure?
     + State management?
     + Performance optimization?
     + Error handling?
     + Testing?
2. Production Considerations:
   * What factors should be considered before deploying to production?
   * How do you handle environment variables?
   * How do you implement monitoring and logging?