# Project-management-workflow/

**Step 1: Create a list of jobs to be done**

No matter what the scale of the job or process we’re creating a workflow for, we need to start by defining the boundaries. In large-scale projects, this can look like determining all the major steps of a project.

In smaller-scale situations, that work will have already been completed. Once we have a rough framework in place, it’s time to create a list of jobs to be done.

This list should include every task and step (in order) and should leave nothing out. Make sure to include the work that comes before and after the “main'' work, like getting stakeholder approval or doing final post-production checks.

Ideally, we’ll start noting which individuals or departments these jobs and tasks belong to as we create the list. We also want to make sure the tasks are listed in order — though we’ll have plenty of chances to refine this as we go.

**Step 2: Identify what tools are needed to execute tasks**

Once we’ve created a detailed list of jobs and tasks, it’s time to expand search. Next, we need to identify the tools and resources that the team needs in order to get the job done.

Consider all types of resources we currently don't have in place (software, templates, files, collateral, etc.). Then make a plan to get the hands on those missing resources.

**Step 3: Delegate tasks and establish roles**

No project management workflow is complete without assigning tasks and steps to individuals (for one-off workflows) or departments. If we’re designing a reusable workflow, one that will be repeated for similar tasks or iterative steps, we’ll want to leave this general for now — list the department, but not the individual.

**Step 4: Test and improve workflow over time**

No project management workflow will be perfect the first time around, so consider the first use of every workflow to be a trial run. Keep on testing and refining the workflows, making improvements and adjustments over time.

We can also run a new workflow past the team that will be using it. Most of the time, some of the people on the team have done some of these tasks before and can offer valuable insights.

Even if the workflow is brand new, showing it to a group can help uncover bottlenecks, dependencies, or even logical failures that we didn’t see ourself.

# What did you like most about your job?

I worked with a really dynamic team of talented people. They’re what I will miss most about the job.

I had an opportunity to learn a lot of new skills and be exposed to varied approaches to business. It was certainly an educational experience.

I loved working one-on-one with customers, which is, incidentally, one of the things that attracted me to this role.

# What interested you about our position?

I am interested in this job because I thought it has many things for me to do.

Before facing this time, I had so many things to investigate.

First, the roles of your company are React and Javascript.

I have rich experiences with those skills.

9 years React experiences and 10 years Javascript experiences.

So I can complete any task that you give me and I can bring a lot of profit to your company.

Second, I guess there will be so many talented web developers and engineers who have high skills and technologies. I just want to learn a lot from them and compare my skills with them.

Third, there will be great managers who have high skills and leadership including you.

I am interested in business. I just want to learn a lot from you.

I mean if you accept me, I will do my best for you and me.

These are all things that I am interested in your job.

# What are you looking for next job

Three things that I'm looking for in my next job are a collaborative, team-focused culture, opportunities to learn and grow my skills from a technical standpoint, and a chance to learn more leadership skills over time

# How do you handle conflict

In most cases, I handle conflict well.

I value diversity and understand that different people have different opinions, which may lead to conflict.

When faced with conflict, I work to collaborate with others to resolve the issue in a way that is mutually beneficial for everyone involved. Sometimes, I can become defensive when trying to express my opinion.

I am practicing tactics to manage this behavior, such as pausing to take a deep breath and carefully considering my words before responding.

# Describe a situation when you had a conflict at work and how you handled it

While working on a project for a previous employer, one of my team members regularly challenged every solution I presented. He also had a tendency to interrupt and talk over others without listening to their input. I experienced a challenge in maintaining my patience when he interrupted others without listening. It reached a point where our respective managers counseled both of us on our behavior.

To resolve this conflict, I had to recognize that I cannot change or control his behavior. I also acknowledged that this behavior, from both of us, was likely a result of stress due to the heavy workload of the project. Therefore, I adjusted my own communication style to increase empathy, avoid triggers and build patience with interruptions. We were able to complete the project and maintain polite correspondence whenever we needed to work together after that.

# Explain a situation in which you disagreed with your manager and how you handled it

When preparing data about the success of past projects to request funding for a new project in our department, a supervisor asked me to delete data records for projects that were less successful. I knew that deleting these records would skew the results and representation of our past performance. I expressed this concern to my supervisor, but she insisted that I remove the data.

After careful consideration, I decided to discuss the issue with the next level of management, which dismissed my concerns and supported the decision to remove the data. Ultimately, I removed data as directed, but I also changed the language in the report to clearly communicate that the reported data represented successful projects to ensure I accurately represented the information. I also documented the situation, including data records removed from reported results, in the event of questioning or an audit.

I understood my supervisor’s decision for the request, but I could not compromise my values. I tried to create a compromise in removing the data and adding the language. We did eventually receive funding for our project based on the reports.

# WHAT WOULD YOU DO IF YOUR BOSS ASKED YOU TO DO SOMETHING THAT YOU DISAGREED WITH?

I would always respect my boss’s decisions. However, if I disagreed with something they wanted me to do, I would say something. I would give my reasons for disagreeing, and try to suggest a potential alternative way of doing things. If, after I had expressed my concerns to my boss, they still wanted me to carry out the task, then I would do it to the best of my ability.

# WHAT WOULD YOU DO IN THE FIRST WEEK OF STARTING WORK HERE

My priority would be to fit into the team quickly. In the very first week, I would introduce myself to my co-workers, I would let them know my strengths and the areas that I could perhaps help the team in, I would get a brief from my manager on what he or she needs me to focus on, and I would then set to work to show everyone what I was capable of. It would also be my priority in the first week of starting work here to build good relationships with my co-workers, and to learn any company policies or procedures that were essential in my role.

# HOW WOULD YOU DEAL WITH A CUSTOMER COMPLAINT?

Whenever dealing with a customer complaint, I would always remember that I was representing your company brand, and as such, I would be professional, courteous, and act with speed. I would listen to the customer and demonstrate effective listening skills. When dealing with a complaint it is important to show the customer you are listening, and that they have your full attention. I would ask the customer questions to get to the bottom of their complaint before outlining what I was going to do to put things right. I would apologize if the company was at fault, keep the customer updated with progress, and make sure we learned from the complaint so that it never happened again.

# HOW WOULD YOU EXPLAIN SOMETHING TECHNICAL TO A NON-TECHNICAL PERSON

Whenever explaining something technical to someone that didn’t understand, I would take my time, I would teach them things slowly and in bite-sized lessons, and I would use a technique that they could relate to. For example, I would use analogies, and also drawings and diagrams they could easily understand. I would check with the person to make sure they fully understood my explanations, I would ask them questions to clarify the learning, and I would also give them plenty of opportunities to ask me questions.

# HOW WOULD YOU DELIVER BAD NEWS TO A CUSTOMER

I would always deliver bad news to a customer by first thinking what I wanted to tell them, how I was going to deliver my message, and also any potential questions or concerns the customer might raise. I would always be truthful and deliver the bad news quickly. I would be honest with the customer, transparent, and acknowledge their feelings. A customer is understandably going to be frustrated with any type of bad news, and it’s important to demonstrate empathy and understanding whilst communicating with them. After I had delivered the bad news, I would focus on any positives, and I would explain what I was going to do to follow up with the customer at a later date. At all times I would maintain control of the conversation, and do all I could to turn things around and make the customer feel valued.

# WHAT’S YOUR IDEAL BOSS

My ideal boss would be decisive, confident, fair, consistent, and supportive. They would empower me to contribute to the team, and listen to my ideas. They would tell me when I was wrong and how I could improve, and share their expertise and experience to allow me and the rest of the team to continually improve, adapt to change, and also grow professionally.

# What is Code Review?

Code review is a software quality assurance process in which software’s source code is analyzed manually by a team or by using an automated code review tool. The motive is purely, to **find bugs**, **resolve errors**, and for most times, improving **code quality**. Reviewing the codebase makes sure that every software or new feature developed within the company is of high quality.

**Top 9 Practices for Reviewing Code**

1. Set goals and standards

2. Communicate Goals and Expectations

3. Define a Code Review Process

4. Use a code review checklist

5. Authors should annotate source code before the review

6. Review for no more than 60 minutes at a time

7. Establish a process for fixing defects found

8. Foster a positive code review culture

9. Automate to Save Time

# ****Best Code Review Technique****

**1. Instant Code Reviewing Technique**

The most direct form of reviewing code is the Instant Code review technique. In this, the developer is writing code while the reviewer sits beside reading the code simultaneously and correcting it on the go. Also known as pair programming, this process is best suited for highly complex programs where two minds can solve the problem much quicker and efficiently.

While this process looks favorable for companies but in reality, the time and workforce needed by this technique make it unfavorable. Two or more people working on code together means fewer average lines per developer. Interruption for corrections also halts the flow of work for the author of the code and the learning curve for a developer hinders if constant support or solution is presented right away by a reviewer for a complex problem.

**2. Ad-hoc (synchronous) Code Reviewing Technique**

Also known as the “Over the Shoulder” code-review process. It is the most commonly used process with around 75% of companies participating in ad-hoc reviews. In this type of synchronous method, the coder produces the code and then asks the reviewer to review the code. The reviewer joins the coder at the screen and reviews the code while discussing it, over the shoulder. It is implemented wisely because it is informal and spontaneous. The process is successful only if the reviewer is available at the time or it disrupts the coder’s speed.

**3. Meeting-Based Code Reviewing Technique**

This is the least commonly used process with only 44% using it once a month. In meeting-based code review, coders complete their work, and a meeting is called. The whole tech team sits, commenting, and attempting to improve the code together. It is a temporary process as it is highly unlikely to perform constantly considering the amount of time, loss of workforce for the time, decreased efficiency, and inability to get the whole team together.

**4. Tool-Based Code Reviewing Technique**

This process is not done by a team together, at least not on the same screen. It is also called an asynchronous code review. In this, once the code gets finished, the coder makes it available for others to review. The reviewer will review the code on their screen commenting, or even amending the errors in the codes. Then notifying the coder who on her agenda will improve it. When there are no changes, the code is marked with no comments for improvements, and the software gets approved.

# Tell us about a time you had to make a quick choice using limited information. Please describe the situation, what actions you took, and the result.

When I first started college, I used to have difficulty making choices about what to prioritize because I tend to want to learn and do everything. And as a computer science major, a lot of my classes involved group projects. During one semester, I had two large projects that I was working on simultaneously for my cybersecurity and web development courses. Each project had sections to them that would have been more challenging for me and required me to learn new things, along with one or two sections that wouldn’t be as much of a time commitment for me since I’d be honing skills I already had. My natural urge is to try to learn something new whenever possible. However, that just wasn’t possible to do with both projects at once and I needed to decide which project I should volunteer to do a more challenging section of (and spend more time on) and which one I should take on a more familiar section of.

I first checked both syllabi and found that each project was worth the same percentage of my grade and confirmed I was on track to get roughly the same grade in both classes.

So at my first meeting with each group, I asked my teammates what aspects they were interested in and most skilled at and what else they had going. I found that each group had someone who could dedicate more time than I’d be able to if I chose to take on a more challenging part of the other project. Plus, there were group members on each team who had more experience and familiarity with the portions of the projects that were new to me and who might be able to complete them without a lot of strain.

So I had to ask myself which project was going to be the most useful for my education and career goals, and the answer there was definitely the web development class—as you can probably tell from me applying to this job—so I took on the section of that project that I considered most challenging and that would allow me to learn something new. For my cybersecurity class, I volunteered to take on a portion of the project that was similar to things I’d done in the past.

Both projects turned out well—we got As for both. But the knowledge I gained from the web development project really solidified my desire to focus on front-end web dev for the long term. I also learned that I can’t do it all and applied similar methods for prioritizing future projects, and I can now make these choices much more quickly

# Tell us what you would do if you were asked to join a group of people to help complete a project.

1. Have a kickoff meeting

The first meeting is an opportunity to get to know your fellow team members, brainstorm ideas and create a plan for the project.

It’s important to set these expectations up front so that you are all on the same page before getting to work. As you discuss each of these questions, be sure to find solutions that work well for everyone.

2. Establish deadlines and responsibilities

After you’ve introduced yourselves and discussed project parameters, determine responsibilities for each member of your group. Make sure that each member has an important role in each phase of the project, from research to writing to the final presentation. Let your team members volunteer for the parts of the assignment that they find most interesting. This way, they will be more engaged in the project and likely more productive as a result.

3. Schedule regular meetups

These meetings could be virtual, via a video conferencing app, or in-person, depending on what works best for your group. Have an agenda, or a list of items to discuss, for each meeting. This helps your team stay focused and ensures that no one is left out of the loop of communication. This is also a good time to check in on project status and see if you’re on track to meet the deadlines you set in the initial meeting.

4. Be a team player

You must keep in mind that this is a group project and an effort on behalf of everyone, not just you. That means everyone should have an important role to play and an effort should be made to make sure all members are working together.

5. Hold everyone accountable

Perhaps one of the more difficult aspects of working with a team is the possibility of a confrontation, especially when people aren’t pulling their weight or there are disagreements about how to approach the project.

# The rest of the group has been working together for weeks and you have new ideas on how to approach the project.

Create a feedback-friendly atmosphere

Encourage idea-sharing day-to-day

Create multiple avenues for idea sharing

Offer anonymous tools

Broaden your idea pool

Lead by example

Reward employee creativity

Use brainstorming programs

# About My Team project building process

* **Agile Development**

There were team leader, QA engineer, data engineers, UX designers, backend developers in my team.

Team leader had broad responsibilities that might include estimating stories and planning implementation details with the team.

The analysts created wireframes, documented user stories, and reviewed test results

Our team decided how much work it can complete during the sprint where the product owner shared priorities.

I could escalate blocks that impede the team’s progress and discussed the status of user stories, teammates shared our daily goals.

We used Jira for managing our issues for our project and decided on how to assign the task to someone

I resolved the issue that the Project Manager assigned me and send Pull Request to him.

Then he checked my code and merge my updated code to Github.

In this way, we built the product according to the requirements of every step.

We showed the functionality to the product owner to gain acceptance of the completed work.

After that, we discussed what went well and what needs improvement in our process in Retrospective meetings.

# What is the biggest challenge you’ve faced in work?

In my most recent role, a former coworker left her job unexpectedly.

During this time, he was the team leader on an important new project.

When he left, my manager asked me to take it over.

While I had never performed his job duties in the past, I had previous leadership experience.

Using my leadership and problem-solving skills, I pushed the project forward, encouraged the rest of the team and completed his responsibilities as well as mine.

We were able to complete that project successfully and on deadline

# What Are Your Career Goals

My current short-term goal is to develop and use my frontend and communication skills in a job like this one.

However, I eventually want to develop into a position that allows me to continue to use these skills while also managing a team group.

I will prepare for this goal by taking on leadership positions in team projects, and by developing my professional career through attending leadership conferences such as the one put on annually by your company.

# What motivates you

What motivating me to choose a developer career was time to have an opportunity to read the book "How to become a Genius".

I wanted to become Genius like others, and I have a role model in deep mind.

It was versatile according to specific events or stories that I have ever encountered.

While reading this book, I was moved about the expression, "Where I am today is my starting point. Who I am today is my starting point. My failures and successes of the past, my fears and hopes of the future are all shadows. Today is my reality, and I’ll use it to create my world"

Based on this query, I reminded myself what my strength is and what to do.

That lead me to here, over 7 years of developer career.

# What did you like most and least about your last job?

-Least

The company was growing quickly, so it was a bit disorganized and I often had multiple supervisors and senior team members giving me different instructions.

However, I think this was a great opportunity to improve my ability to work under pressure and my communication skills. For example, I had to learn to communicate clearly and quickly to clarify with my managers and team if there was a conflict in what I had been asked to do.

I enjoyed most aspects of the job and I ended up like working in such a fast-paced environment in the end.

It just took some adjustment. It was a great learning experience for me.

-Most

I love how the company offered learning opportunities and seemed to care about helping their employees develop and grow professionally.

They had weekly lunch meetings where we’d learn a new topic, they offered tuition reimbursement, and all of the senior leadership team was very approachable and supportive.

# What is my strength and weakness?

- Strength

As mentioned, I have experience working as a frontend, backend side and full-stack developer.

Of course, I don’t know exactly if this experience will be helpful or not for your work.

However, in this period I have so many experiences in Python, NodeJS, their frameworks or libraries including Nuxt, Next, React, Vue.

If you accept me, I will try to give you great satisfaction and success.

You probably will be satisfied.

My strength is creativity and tenacity. I love failure, learn from failure and how to deal with failure next time.

Thus being optimistic, never give up to realize dream.

- Weakness

I always try to avoid confrontation, in both my personal and professional life.

This caused me to compromise sometimes on the quality of my work or what I needed to complete a project just to keep the peace.

This became a real problem when I became a manager.

One of the most critical aspects of managing people is telling them what they need to hear and not what they want to hear.

I recognized this weakness and had been actively working to voice my opinions constructively and helpfully for the betterment of the team.

# Why Are You the Best Person for the Job?

I am a self-motivated person who is willing to go above and beyond on any project, and to learn valuable skills on my own time. For example, I taught myself four computer programming languages in college, simply out of a passion to learn how to code.

My previous job working as a frontend expert provided me with the ideal experience for this position.

For 9 years, I developed many of the skills required for this job, including performance optimization, data visualizing, Api integration, good communication skill, increasing site speed.

# Are you comfortable working with a team of engineers and programmers to develop new products or features?

I am very comfortable working with a team of engineers and programmers to develop new products or features.

I have always enjoyed learning from my colleagues and using their expertise to help me solve problems.

At my last job, we had a tight deadline to create a new feature for our software program.

I was able to use the knowledge of my teammates to come up with a solution quickly

# What are some of the most important qualities that a technical lead should have

I believe that one of the most important qualities of a technical lead is communication.

As a technical lead, I would need to communicate with my team members about their projects and how we can best meet our clients’ needs.

Another quality that I think is important is problem-solving.

In this role, I would need to solve any issues that arise during the project development process.

Finally, I think leadership is another essential quality because I would need to motivate my team members and ensure that they feel comfortable asking me questions

# Provide an example of a time when you had to manage a team of engineers and programmers and how you handled the situation

In my last role as a senior engineer, I had to oversee a large-scale project involving many different departments.

We had to create an entirely new system for our company’s website, which meant we needed to collaborate with other engineers, designers and programmers.

I met with each member of my team individually to discuss their roles and responsibilities on the project.

Then, I held weekly meetings with my entire team to check in on everyone’s progress and offer help if anyone was struggling.

# What is your greatest achievement?

In my most recent role, I was responsible for managing the orientation and training programs for our new hires. Unfortunately, the content was not so good.

While it was necessary information for our new hires to have, we found that only 40% of new hires did not complete the training.

We were also receiving poor feedback on the course evaluation forms.

I decided to rework the training program to make it more relevant and interesting, based on industry best practices and feedback on the evaluation forms.

Today, 90% of participants complete the training and provide positive feedback about their experience.

My manager was so pleased with the improvements that he asked me to lead a training seminar.

# If hired for the position, what would be your priorities and goals as a technical lead?

My first priority as technical lead would be to ensure all projects I oversee meet their deadlines and stay within budget.

To do this, I would create a plan for monitoring project progress and communicate regularly with my team members about any challenges they’re facing so we can find solutions together.

Another priority would be to develop new training programs for employees who want to advance in their careers.

I believe it’s important to provide opportunities for growth and development to everyone on staff.

# What would you do if you and your team members had different ideas about how to implement a new feature

I would first ask my team members why they think the feature should be implemented in a certain way.

I’d also explain why I think it’s best to do things a different way.

If we still couldn’t agree on one solution, I would take some time to research other ways of implementing the feature.

After researching, I would present both ideas to my team again and see if anyone had any additional thoughts or feedback.

# When reviewing code written by other engineers, how do you provide constructive feedback

I find that providing constructive feedback is one of my favorite parts of being a technical lead.

I enjoy helping others learn new coding languages or update existing ones.

In my last role as a software engineer, I worked with an engineer who was learning Python for the first time.

He struggled at first but eventually learned it well enough to apply it to his work.

When reviewing his code, I provided him with specific feedback on what he did well and areas he could improve.

# Describe your experience with Agile development

I have extensive experience using Agile development.

I find this method of software development to be highly beneficial because it allows me to collaborate with my team members while also testing new features as they’re created.

When working on projects using Agile development, I typically start by creating user stories for each feature we want to add to our product.

Then, I break down those stories into smaller tasks so my team can complete them more easily.

# How often do you make mistakes when programming and how do you go about fixing them

I make mistakes all the time when programming because it’s an iterative process where I’m constantly making changes and testing my code.

However, I try to minimize the number of mistakes I make by thoroughly reading through the requirements before starting any project.

If I do make a mistake, I immediately stop what I’m doing and go back to the original coding to find out what went wrong. Then, I fix the problem and test my code again

# There is a bug in the code and the deadline is fast approaching. What do you do?

If there was a bug in the code and the deadline was fast approaching, I would first try to find out what caused the bug.

If I could fix the bug myself, I would do so immediately.

If not, I would have my team members help me look for the cause of the bug.

Once we found the cause, we would work together to fix the bug before the deadline.

# Describe the environment in which you work best.

I enjoy working as part of a team to achieve goals.

My previous jobs have had open offices without cubicle walls, which encouraged us to get feedback from others about our ideas and equally share in the workload

# What Is Your Ideal Work Environment

My ideal work environment is one where I'm able to work as part of a team and that allows everyone's talents to grow.

As I researched your company, I noticed its devotion to cultivating each employee's skills and abilities.

I've found that this type of environment is most conducive to my productivity, especially in a position that requires me to constantly improve my full stack skills.

It allows me to remain passionate about my job and helps me express my creativity to the best of my ability

# Describe your ideal boss or supervisor

My ideal boss is someone who allows employees to have freedom in daily operations, but is always available to answer questions and help when needed.

# Mentoring

I tell juniors how to accomplish what they are trying to do, but do not do it for them.

I regularly hold code reviews and use code reviews as an opportunity to point out best practices.

I get start early with planning and documentation instead of letting junior developers jump right into code.

I help a junior developer learn from his or her mistakes. There will be mistakes. I show them that mistakes are part of improving.

# What are the 3 most important things in your next role?

* Company vision/product

The company’s product should be innovative and promising.

That’s why I applied for this particular position in your company.

To me, your company’s vision and product is really promising and has good future plans.

* Culture

I look for companies who have a positive and adaptive culture, and trust.

It should be part of a positive culture where contributions are appreciated

The team should encourage each employee to suggest any good idea or solution for improving the product or innovating the team management.

And if it sounds good to the team’s vision, then the team should have a mind to accept it.

* Career

Growth potential is the most important thing to me with the company’s overall growth. I’d love to work for a company that is innovative and always looking for new opportunities to expand.

Able to learn new things and develop your skill set

# Why should we not update the state directly?

One should never update the state directly because of the following reasons.

* If you update it directly, calling the setState() afterward may just replace the update you made.
* When you directly update the state, it does not change this.state immediately. Instead, it creates a pending state transition, and accessing it after calling this method will only return the present value.
* You will lose control of the state across all components.

# Higher order component

The higher-order function is one that takes a function as an input parameter or returns a new function as output. In the context of React, higher-order components (HOCs) are the components that take a component as input and return another component as output.

A higher-order component acts as a container for other components.

This helps to keep components simple and enables re-usability.

They are generally used when multiple components have to use a common logic.

Higher-Order Components are still useful these days, even though many React developers take them as legacy, because they are from a time when React Class Components where used. Especially when they are used to render conditional JSX. However, if not using any conditional JSX, using a Hook instead of a HOC is often a better design choice in modern React.

# How about pure component

A React component is considered pure if it renders the same output for the same state and props. For this type of class component, React provides the PureComponent base class. Class components that extend the React.PureComponent class are treated as pure components.

# How do you pass data from one component to one other component and what are those

For passing the data from the child component to the parent component, we have to create a callback function in the parent component and then pass the callback function to the child component as a prop.

This callback function will retrieve the data from the child component. The child component calls the parent callback function using props and passes the data to the parent component.

# Bundling

Bundling is the process of following imported files and merging them into a single file

# Code Splitting

Code-Splitting is a feature supported by bundlers like [Webpack](https://webpack.js.org/guides/code-splitting/), [Rollup](https://rollupjs.org/guide/en/#code-splitting) and Browserify (via [factor-bundle](https://github.com/browserify/factor-bundle)) which can create multiple bundles that can be dynamically loaded at runtime.

# Context

Context is designed to share data that can be considered “global” for a tree of React components, such as the current authenticated user, theme, or preferred language

Context is primarily used when some data needs to be accessible by many components at different nesting levels. Apply it sparingly because it makes component reuse more difficult.

# Error Boundaries

Error boundaries are React components that catch JavaScript errors anywhere in their child component tree, log those errors, and display a fallback UI instead of the component tree that crashed. Error boundaries catch errors during rendering, in lifecycle methods, and in constructors of the whole tree below them

# Refs and the DOM

Refs provide a way to access DOM nodes or React elements created in the render method.

# Profiler API

The Profiler measures how often a React application renders and what the “cost” of rendering is. Its purpose is to help identify parts of an application that are slow and may benefit from optimizations such as memorization

# Asynchronization

React-async provides a declarative API to perform any REST API calls using a single React component, allowing declarative programming to be used throughout the application. It takes care of handling errors, promise resolution, and retrying promises, and deals with local asynchronous state.

you can write the very same component as a function with the useEffect hook.

Also we can implement aynchronization by using async/await

# How to improve the performance of the application?

1. Using Immutable Data Structures

2. Function/Stateless Components and React.PureComponent

3. Multiple Chunk Files

4. Using Production Mode Flag in Webpack

5. Dependency optimization

6. Use React.Fragments to Avoid Additional HTML Element Wrappers

7. Avoid Inline Function Definition in the Render Function

8. Throttling and Debouncing Event Action in JavaScript

9. Avoid using Index as Key for map

10. Avoiding Props in Initial States

11. Spreading props on DOM elements

12. Use Reselect in Redux to Avoid Frequent Re-render

13. Avoid Async Initialization in componentWillMount()

14. Memoize React Components

15. CSS Animations Instead of JS Animations

16. Using a CDN

17. Using Web Workers for CPU Extensive Tasks

18. Analyzing and Optimizing Your Webpack Bundle Bloat

19. Consider Server-side Rendering

20. Enable Gzip Compression on Web Server

# Did you write any API Services? What type of Authentication did you use?

1. I have ever used REST API and Fetch API.

A REST API is an API that follows what is structured in accordance with the REST Structure for APIs. REST stands for “Representational State Transfer”. It consists of various rules that developers follow when creating APIs.

The way a REST API is structured depends on the product it’s been made for — but the rules of REST must be followed.

The fetch() API is an inbuilt JavaScript method for getting resources from a server or an API endpoint. It’s similar to XMLHttpRequest, but the fetch API provides a more powerful and flexible feature set.

The fetch() API method always takes in a compulsory argument, which is the path or URL to the resource you want to fetch. It returns a promise that points to the response from the request, whether the request is successful or not. You can also optionally pass in an init options object as the second argument.

Once a response has been fetched, there are several inbuilt methods available to define what the body content is and how it should be handled.

# JWT? how it works?

JWT is JSON Web Token

The React app requests a JWT from the authentication server whenever the user wants to sign on.

The authentication server generates a JWT using a private key and then sends the JWT back to React app.

My React app stores this JWT and sends it to your backend server whenever user needs to make a request.

The backend server verifies the JWT using a public key and then reads the payload to determine which user is making the request

# JScript - Debouncing or throttling?

Throttling

In a nutshell, throttling means delaying function execution. So instead of executing the event handler/function immediately, you’ll be adding a few milliseconds of delay when an event is triggered. This can be used when implementing infinite scrolling, for example. Rather than fetching the next result set as the user is scrolling, you can delay the XHR call.

Another good example of this is Ajax-based instant search. You might not want to hit the server for every key press, so it’s better to throttle until the input field is dormant for a few milliseconds

Throttling can be implemented a number of ways. I can throttle by the number of events triggered or by the delay event handler being executed.

Debouncing

Unlike throttling, debouncing is a technique to prevent the event trigger from being fired too often. If you are using lodash, I can wrap the function you want to call in lodash’s debounce function

# Closures in application?

Closures are inner functions that have access to the outer function’s variables and parameters. Even after the outer function’s execution is finished, the inner functions have access to the variables in the outer function. Closures are everywhere in JavaScript

# What is JSX?

JSX is a syntax extension of JavaScript. It is used with React to describe what the user interface should look like. By using JSX, we can write HTML structures in the same file that contains JavaScript code

# What is the virtual DOM?

DOM stands for Document Object Model. The DOM represents an HTML document with a logical tree structure. Each branch of the tree ends in a node, and each node contains objects.

React keeps a lightweight representation of the real DOM in the memory, and that is known as the virtual DOM. When the state of an object changes, the virtual DOM changes only that object in the real DOM, rather than updating all the objects.

# Why use React instead of other frameworks, like Angular?

Easy creation of dynamic applications: React makes it easier to create dynamic web applications because it provides less coding and provides more functionality, whereas, with JavaScript applications, code tends to get complex very quickly.

Improved performance: React uses virtual DOM, which makes web applications perform faster. Virtual DOM compares its previous state and updates only those components in the real DOM, whose states have changed, rather than updating all the components — like conventional web applications.

Reusable components: Components are the building blocks of any React application, and a single app usually consists of multiple components. These components have their own logic and controls, and they can be reused through the application, which, in turn, dramatically reduces the development time of an application.

Unidirectional data flow: React follows a unidirectional data flow. This means that when designing a React app, we often nest child components within parent components. And since the data flows in a single direction, it becomes easier to debug errors and know where the problem occurs in an application at the moment.

Dedicated tools for easy debugging: Facebook has released a chrome extension that we can use to debug React applications. This makes the process of debugging React to web applications faster and easier.

# What is an event in React?

An event is an action that a user or system may trigger, such as pressing a key, a mouse click, etc.

React events are named using camelCase, rather than lowercase in HTML.

With JSX, you pass a function as the event handler, rather than a string in HTML.

# What is an arrow function and how is it used in React?

An arrow function is a short way of writing a function to React.

It is unnecessary to bind ‘this’ inside the constructor when using an arrow function. This prevents bugs caused by the use of ‘this’ in React callbacks.

# What are the components in React?

Components are the building blocks of any React application, and a single app usually consists of multiple components. A component is essentially a piece of the user interface. It splits the user interface into independent, reusable parts that can be processed separately.

There are two types of components in React:

**Functional Component:** A functional component is just a plain JavaScript pure function that accepts props as an argument and returns a React element(JSX).

There is no render method used in functional components.

Functional component run from top to bottom and once the function is returned it can’t be kept alive.

Also known as Stateless components as they simply accept data and display them in some form, that they are mainly responsible for rendering UI.

React lifecycle methods (for example, componentDidMount) cannot be used in functional components.

Hooks can be easily used in functional components to make them Stateful.

**Class Component:** A class component requires you to extend from React. Component and create a render function which returns a React element

It must have the render() method returning JSX (which is syntactically similar to HTML)

Class component is instantiated and different life cycle method is kept alive and being run and invoked depending on phase of class component.

Also known as Stateful components because they implement logic and state.

React lifecycle methods can be used inside class components (for example, componentDidMount).

# What is a state in React?

The state is a built-in React object that is used to contain data or information about the component. The state in a component can change over time, and whenever it changes, the component re-renders.

The change in state can happen as a response to user action or system-generated events. It determines the behavior of the component and how it will render.

# What are props in React?

Props are short for Properties. It is a React built-in object that stores the value of attributes of a tag and works similarly to HTML attributes.

Props provide a way to pass data from one component to another component. Props are passed to the component in the same way as arguments are passed in a function.

# Explain the lifecycle methods of components

getInitialState(): This is executed before the creation of the component.

componentDidMount(): Is executed when the component gets rendered and placed on the DOM.

shouldComponentUpdate(): Is invoked when a component determines changes to the DOM and returns a “true” or “false” value based on certain conditions.

componentDidUpdate(): Is invoked immediately after rendering takes place.

componentWillUnmount(): Is invoked immediately before a component is destroyed and unmounted permanently.

So far, if you have any doubts about the above React interview questions and answers, please ask your questions in the section below

# What is Redux?

Redux is an open-source, JavaScript library used to manage the application state. React uses Redux to build the user interface. It is a predictable state container for JavaScript applications and is used for the entire application’s state management.

# What are the components of Redux?

Store: Holds the state of the application.

Action: The source information for the store.

Reducer: Specifies how the application's state changes in response to actions sent to the store.

# Why do we need to React Router?

It maintains consistent structure and behavior and is used to develop single-page web applications.

Enables multiple views in a single application by defining multiple routes in the React application.

# List some of the major advantages of React.

Some of the major advantages of React are:

It increases the application’s performance

It can be conveniently used on the client as well as server side

Because of JSX, code’s readability increases

React is easy to integrate with other frameworks like Meteor, Angular, etc

Using React, writing UI test cases become extremely easy

# What are the limitations of React?

Limitations of React are listed below:

React is just a library, not a full-blown framework

Its library is very large and takes time to understand

It can be little difficult for the novice programmers to understand

Coding gets complex as it uses inline templating and JSX

# What is Provider Component?

The <Provider> components makes the Redux store available to any nested components that need to access the Redux store.

Since any React component in a React Redux app can be connected to the store, most applications will render a <Provider> at the top level, with the entire app’s component tree inside of it.

The Hooks and connect APIs can then access the provided store instance via React's Context mechanism.

# Callbacks

In javascript, a callback is simply a function that is passed to another function as a parameter and is invoked or executed inside the other function. Here a function needs to wait for another function to execute or return value and this makes the chain of the functionalities (when X is completed, then Y executed, and it goes on.). This is the reason callback is generally used in the asynchronous operation of javascript to provide the synchronous capability.

- Promises

Promises are useful in asynchronous javascript operation when we need to execute two or more back to back operations (or chaining callback), where each subsequent function starts when the previous one is completed. A promise is an object that may produce a single value sometime in the future, either a resolved value or a reason that it’s not resolved (rejected).

- Async & Await

Stop and wait until something is resolved. Async & await just syntactic sugar on top of Promises and like promises it also provides a way to maintain asynchronous operation more synchronously. So in javascript asynchronous operation can be handled in various versions…

# PureFunction

Pure Function is a function (a block of code ) that always returns the same result if the same arguments are passed.

function pureFunction(param) {

return param + 1

}

# TypeScript

TypeScript is a super set of JavaScript.

TypeScript builds on top of JavaScript. First, you write the TypeScript code. Then, you compile the TypeScript code into plain JavaScript code using a TypeScript compiler

There are two main reasons to use TypeScript:

TypeScript adds a type system to help you avoid many problems with dynamic types in JavaScript.

TypeScript implements the future features of JavaScript a.k.a ES Next so that you can use them today.

# What is JavaScript?

JavaScript is a client-side and server-side scripting language inserted into HTML pages and is understood by web browsers. JavaScript is also an Object-based Programming language

# What are JavaScript Data Types?

Following are the JavaScript Data types:

Number

String

Boolean

Object

Undefined

# Generator Function In Javascript

A generator is a function that can stop midway and then continue from where it stopped.

Generators are functions that can be exited and later re-entered.

Their context (variable bindings) will be saved across re-entrances

Calling a generator function does not execute its body immediately;

an iterator object for the function is returned instead.

When the iterator's next() method is called, the generator function's body is executed until the first yield expression, which specifies the value to be returned from the iterator or, with yield\*, delegates to another generator function.

The next() method returns an object with a value property containing the yielded value and a done property which indicates whether the generator has yielded its last value, as a boolean.

Calling the next() method with an argument will resume the generator function execution, replacing the yield expression where an execution was paused with the argument from next().

# What is Memoization in Javascript

Memoization is a top-down, depth-first, optimization technique of storing previously executed computations. Whenever the program needs the result of these computations, the program will not have to execute that computation again. Instead, it will reuse the result of the previously executed computation. This way the program will not have to repeat expensive computations. An expensive function is a function that takes some time to execute.

# Hoisting

is JavaScript's default behavior of moving all declarations to the top of the current scope

# PUT vs PATCH

PUT is a method of modifying resource where the client sends data that updates the entire resource. It is used to set an entity’s information completely. PUT is similar to POST in that it can create resources, but it does so when there is a defined URI. PUT overwrites the entire entity if it already exists, and creates a new resource if it doesn’t exist.

For example, when you want to change the first name of a person in a database, you need to send the entire resource when making a PUT request.

{“first": "John", "last": "Stone”}

To make a PUT request, you need to send the two parameters; the first and the last name.

Unlike PUT, PATCH applies a partial update to the resource.

This means that you are only required to send the data that you want to update, and it won’t affect or change anything else. So if you want to update the first name on a database, you will only be required to send the first parameter; the first name.

# What are the most significant advantages of using GraphQL over REST?

A list of the most significant advantages of using GraphQL over REST:

* There is only one endpoint in GraphQL, but REST has multiple endpoints. That's why GraphQL is more cost-effective than REST. You don't have to use your resources for various endpoints.
* In REST API, you have to use multiple requests to retrieve a complex data-set, but in GraphQL, you can execute a complex query easily in just a single request.
* You can change the request data format in GraphQL, but it is not possible to do the same in REST.
* The development speed in GraphQL is faster than REST.
* GraphQL provides high consistency across all platforms, while In REST, it is hard to get consistency across all platforms.
* GraphQL doesn't support an automatic caching system, while REST uses caching automatically.

# What is Apollo in GraphQL?

Apollo is a platform for the implementation of GraphQL. As we know that GraphQL is a query language, so in order to use this query language easily, we need a platform, Apollo provides that platform.

Apollo provides two open-sourced libraries to create client and server. Here, the client is used to fetch data from a GraphQL server, and the server is used to create an API for GraphQL client.

# What do you know by Mutation in GraphQL?

Mutation is one of the most important operations in GraphQL. It is used for write operation when you want to add delete and edit data.

# What is subscription in GraphQL?

In GraphQL, the subscription is used for listening for any data changes. The server sends a notification message to the client after any data changes, if the client is subscribed to that event.

# What do you mean by a Query in GraphQL?

A GraphQL query is used to read data. It is similar to the GET request we use in REST APIs. The GraphQL queries are used to retrieve data from the GraphQL server.

# What does useEffect do?

By using this Hook, you tell React that your component needs to do something after render. React will remember the function you passed (we’ll refer to it as our “effect”), and call it later after performing the DOM updates. In this effect, we set the document title, but we could also perform data fetching or call some other imperative API.

# Why is useEffect called inside a component?

Placing useEffect inside the component lets us access the count state variable (or any props) right from the effect. We don’t need a special API to read it — it’s already in the function scope. Hooks embrace JavaScript closures and avoid introducing React-specific APIs where JavaScript already provides a solution.

# Does useEffect run after every render?

Yes! By default, it runs both after the first render and after every update. Instead of thinking in terms of “mounting” and “updating”, you might find it easier to think that effects happen “after render”. React guarantees the DOM has been updated by the time it runs the effects.

# **What Is a REST API?**

REST is basically an architectural style of the web services that work as a channel of communication between different computers or systems on the internet. The term REST API is something else.

Those application programming interfaces that are backed by the architectural style of REST architectural system are called REST APIs. REST API compliant web services, database systems, and computer systems permit requesting systems to get robust access and redefine representations of web based resources by deploying a predefined set of stateless protocols and standard operations.

By these protocols and operations and redeploying the manageable and updatable components without causing the effect on the system, REST API systems deliver fast performance, reliability, and more progression.

# **What Is a SOAP API?**

SOAP is a standard communication protocol system that permits processes using different operating systems like Linux and Windows to communicate via [HTTP](https://dzone.com/articles/the-http-series-part-1-overview-of-the-basic-conce) and its [XML](https://dzone.com/articles/writing-and-reading-xml-file). SOAP based APIs are designed to create, recover, update and delete records like accounts, passwords, leads, and custom objects.

These offers over twenty different kinds of calls that make it easy for the API developers to maintain their accounts, perform accurate searches and much more. These can then be used with all those languages that support web services.

SOAP APIs take the advantages of making web based protocols such as HTTP and its XML that are already operating the all operating systems that are why its developers can easily manipulate web services and get responses without caring about language and platforms at all.

**Differences:**

* REST API has no official standard at all because it is an architectural style. SOAP API, on the other hand, has an official standard because it is a protocol.
* REST APIs uses multiple standards like HTTP, JSON, URL, and XML while SOAP APIs is largely based on HTTP and XML.
* As REST API deploys multiple standards, so it takes fewer resources and bandwidth as compared to SOAP that uses XML for the creation of Payload and results in the large sized file.
* The ways both APIs exposes the business logics are also different. REST API takes advantage of URL exposure like @path("/WeatherService") while SOAP API use of services interfaces like @WebService.
* SOAP API defines too many standards, and its implementer implements the things in a standard way only. In the case of miscommunication from service, the result will be the error. REST API, on the other hand, don't make emphasis on too many standards and results in corrupt API in the end.
* REST API uses Web Application Description Language, and SOAP API used Web Services Description language for describing the functionalities being offered by web services.
* REST APIs are more convenient with JavaScript and can be implemented easily as well. SOAP APIs are also convenient with JavaScript but don't support for greater implementation.

# What is Destructuring in Javascript?

The destructuring assignment syntax is a JavaScript expression that makes it possible to unpack values from arrays, or properties from objects, into distinct variables.

# Smart component vs Dumb component

Smart components are app level components that perform functions and manage data while dumb components focus solely on the UI.

# CSS BEM

BEM stands for **Block, Element,**and**Modifier**. It’s a CSS naming convention for writing cleaner and more readable CSS classes.

BEM also aims to write **independent CSS blocks** in order to reuse them later in your project.

**how BEM class namings are:**

// Blocks are named as standard CSS classes  
**.block {}**// Elements declared with 2 underscores, after block  
**.block\_\_element {}**// Modifiers declared with 2 dashes, after block or after element  
**.block--modifier {}//** element and modifier together  
**.block\_\_element--modifier {}**

**What is a Block?**

Blocks are independent, reusable and usually bigger components of a webpage. They can have modifiers and contain elements.

**Elements**

Elements are children of blocks. An element can only have 1 parent Block, and can’t be used independently outside of that block.

**Modifiers**

Modifiers represent different states or styles of classes. They can be used both for blocks or elements.

# Git rebase vs merge difference

Merging is a safe option that preserves the entire history of your repository, while rebasing creates a linear history by moving your feature branch onto the tip of main

# Flex box

The Flexible Box Layout Module, makes it easier to design flexible responsive layout structure without using float or positioning

flex-container, flex-direction, flex-wrap, flex-shrink, flex-grow, align-item, justify-content, gap

# Dependency Injection

Dependencies are services or objects that a class needs to perform its function. Dependency injection, or DI, is a design pattern in which a class requests dependencies from external sources rather than creating them.

# Difference between var, let and const keywords in Javascript

**var** keyword in JavaScript: The var is the oldest keyword to declare a variable in JavaScript.

Scope: Global scoped or function scoped.

The scope of the var keyword is the global or function scope. It means variables defined outside the function can be accessed globally, and variables defined inside a particular function can be accessed within the function.

**let** keyword in JavaScript: The let keyword is an improved version of the var keyword.

Scope: block scoped:

The scope of a let variable is only block scoped. It can’t be accessible outside the particular block ({block}). Let’s see the below example.

**const** keyword in JavaScript: The const keyword has all the properties that are the same as the let keyword, except the user cannot update it.

Scope: block scoped:

When users declare a const variable, they need to initialize it, otherwise, it returns an error. The user cannot update the const variable once it is declared.

# Prototype in Javascript

JavaScript is a prototype based language, so, whenever we create a function using JavaScript, JavaScript engine adds a prototype property inside a function,

Prototype property is basically an object (also known as Prototype object), where we can attach methods and properties in a prototype object, which enables all the other objects to inherit these methods and properties.

In JavaScript, we achieve Inheritance with the help of prototype chaining.

Prototype chain

The prototype chain mechanism is simple: When you access a property p on object obj, the JavaScript engine will search this property inside obj object. If the engine fails to search, it continues searching in the prototype of obj object and so on until reaching Object.prototype. If after the search has finished, and nothing has been found the result will be undefined. For example:

# How to Implement User Tracking System such as login, register

User behavior is an umbrella term for all the activities a user performs on your app or website.

User behavior analytics (UBA) involves collecting, grouping, and analyzing users’ behavior data to make decisions that enhance your overall UX.

Important user behavior metrics to track include trial signups and conversions, activation, and complete adoption.

You can understand user behavior by measuring the following: product and feature engagement analytics, session recordings and scroll maps, product and user experience analytics.

Tracking user behavior in SaaS helps you understand what brings value to users, identify friction points in the user journey, remove blockers and shorten time to value for your tool.

The best tools for tracking user behavior are Userpilot for tracking in-app behavior and how users progress through the journey, Amplitude for a wide range of analytics data and Hotjar for heatmaps and session recording.

# How to scale up server

Overloading the web server

With an increase in the number of clients simultaneously connecting to one web server, the server will eventually run out of CPU and RAM and cease performing. So what can be done to remedy this? Quite simply, server resources need to be increased in order to accommodate more clients. This can be done in a number of ways.

Vertical scaling: this is either the addition of resources to the existing server, or its replacement with another more powerful server. In this case, the architecture remains the same, but it's important to note that this isn’t a permanent fix. Why? Because even these resources will eventually run out and the vertical scaling would need to go on forever. This is why you need a better long-term fix. Enter horizontal scaling.

Horizontal hybrid scaling: This involves the addition of more servers which serve the same purpose as the first one. With an application’s continued popularity, the current servers runout of resources, thus we need to add more servers to serve other incoming clients.

A combination of both horizontal and vertical scaling: The above-mentioned scaling approaches are not mutually exclusive and can certainly be used in combination. Any application is capable of vertically scaling up, horizontally scaling out, neither, or both. You may well have a scenario in which parts of your application only vertically scale up, while at the same time other parts might horizontally scale out.

# UNIT TEST #

In this type of test, individual units or components of the software are tested. A unit might be an individual function, method, procedure, module, or object. A unit test isolates a section of code and verifies its correctness, in order to validate that each unit of the software’s code performs as expected.

In unit testing, individual procedures or functions are tested to guarantee that they are operating properly, and all components are tested individually. For instance, testing a function or whether a statement or loop in a program is functioning properly would fall under the scope of unit testing.

# COMPONENT TEST #

Component testing verifies the functionality of an individual part of an application. Tests are performed on each component in isolation from other components. Generally, React applications are made up of several components, so component testing deals with testing these components individually.

For example, consider a website that has different web pages with many components. Every component will have its own subcomponents. Testing each module without considering integration with other components is referred to as component testing.

Testing like this in React requires more sophisticated tools. So, we would need Jest and sometimes more sophisticated tools, like Enzyme, which we will discuss briefly later.

# SNAPSHOT TEST #

A snapshot test makes sure that the user interface (UI) of a web application does not change unexpectedly. It captures the code of a component at a moment in time, so that we can compare the component in one state with any other possible state it might take.

Jest is a delightful JavaScript testing framework with a focus on simplicity. It can be installed with npm or Yarn. Jest fits into a broader category of utilities known as test runners. It works great for React applications, but it also works great outside of React applications.

Enzyme is a library that is used to test React applications. It’s designed to test components, and it makes it possible to write assertions that simulate actions that confirm whether the UI is working correctly.

Jest and Enzyme complement each other so well, so in this article we will be using both.

# What is Decorator in Angular

Decorators are a design pattern that is used to separate modification or decoration of a class without modifying the original source code. In AngularJS, decorators are functions that allow a service, directive or filter to be modified prior to its usage.

# Angular Hook Lifecycle

| HOOK METHOD | PURPOSE | TIMING |
| --- | --- | --- |
| ngOnChanges() | Respond when Angular sets or resets data-bound input properties. The method receives a [SimpleChanges](https://angular.io/api/core/SimpleChanges) object of current and previous property values. | Called before ngOnInit() (if the component has bound inputs) and whenever one or more data-bound input properties change.  **NOTE**: If your component has no inputs or you use it without providing any inputs, the framework will not call ngOnChanges(). |
| ngOnInit() | Initialize the directive or component after Angular first displays the data-bound properties and sets the directive or component's input properties. | Called once, after the first ngOnChanges(). ngOnInit() is still called even when ngOnChanges() is not (which is the case when there are no template-bound inputs). |
| ngDoCheck() | Detect and act upon changes that Angular can't or won't detect on its own. | Called immediately after ngOnChanges() on every change detection run, and immediately after ngOnInit() on the first run. |
| ngAfterContentInit() | Respond after Angular projects external content into the component's view, or into the view that a directive is in. | Called once after the first ngDoCheck(). |
| ngAfterContentChecked() | Respond after Angular checks the content projected into the directive or component. | Called after ngAfterContentInit() and every subsequent ngDoCheck(). |
| ngAfterViewInit() | Respond after Angular initializes the component's views and child views, or the view that contains the directive. | Called once after the first ngAfterContentChecked(). |
| ngAfterViewChecked() | Respond after Angular checks the component's views and child views, or the view that contains the directive. | Called after the ngAfterViewInit() and every subsequent ngAfterContentChecked(). |
| ngOnDestroy() | Cleanup just before Angular destroys the directive or component. Unsubscribe Observables and detach event handlers to avoid memory leaks. See details | Called immediately before Angular destroys the directive or component |

# What Is A Pull Request In GitHub?

A pull request is a notification that a developer has made code changes locally that need to be reviewed, and if good, merged with the master branch. With some git branching strategies, the merge is with a feature, develop, or release branch. And, in Continuous Integration and Continuous Delivery/Deployment, pull requests are automated, though code reviews are still conducted.

# What’s The Difference Between A Pull Request And A Push?

The difference is actually pretty simple:

A push equates to assigning a task to a developer who must then start by making a clone of the “main branch” where changes can be freely done.

A pull takes the recent changes or commits in a developer’s local or cloned branch and, if they pass review and testing, merges them with your main branch.

# Same Criteria For Writing And Reviewing Code

There’s a lot to be considered when writing and reviewing code, including but not limited to:

Does the code comply with your team’s coding standards?

Is the code modular, logically structured and easy to read?

Are variables and functions named suitably and meaningfully?

Are functions, methods, behaviors, etc. sufficiently documented?

Does the code work?

How complex is the code and can it be simplified or is it “good enough”?

Is there any duplicate or unnecessary code?

How much of the code is covered with meaningful tests?

Are there any “untested hot spots” that should be scrutinized in greater detail?Does the code cover all relevant failure states?

Does the code meet performance requirements or have performance issues?

# Does Every Pull Request Involve A Code Review?

Yes, but remember there are several types of code review. Code reviews for pull requests can be (formal) technical reviews or (informal) inspections. Additionally, in Continuous Integration, Delivery, and Deployment, pull requests are automated though code reviews are still done but usually at a later stage.

In formal code reviews, the author and reviewer sit down together to discuss the changes and make revisions, as needed. Formal code reviews are best for when the PR involves significant amounts of code, code with many changes, or very complex code. A general rule of thumb is that a formal code review can cover 400 lines of code in an hour and catch 70-90% of defects.

Informal reviews can be handled via emails or comments, and are best for handling many small changes in code.

# How GitHub Pull Requests Fit The Development Process

* Create a Branch – This is the push. The actual branch you’re pushing depends on your team’s branching strategy. This should always be the master branch in GitHub Flow and Trunk Based Development – It could be different (Feature, Develop, Release) if you use Git Flow, One Flow or GitLab Flow.
* Add Commits – Commits are the files and changes to files that you make. Each commit should have a clear description of the reason for the commit. Each commit is a distinct change applied to your local branch first.
* Open a Pull Request – This is a formal request for your commit to be reviewed by another developer.
* Code Review – A developer agrees to perform a code review on your commit, which may be done formally or informally. If approved, your commit takes a step forward (to #5 or #6 depending on your team’s process). If not approved, you return to #2 with comments advising you of issues that need to be fixed.
* Deploy – If your commit passes the code review, most teams will deploy it for further, more complex testing in a simulated production environment. If the changes don’t pass tests, you return to #2 with feedback from the test.
* Merge – Finally, after your code is reviewed, approved, and passes all tests, your commit is merged with the master branch – which, in GitHub Flow and TBD should always be deployment-ready.

# What is Prop Drilling?

Prop drilling is a situation where data is passed from one component through multiple interdependent components until you get to the component where the data is needed.

# What is Reusable Component?

In React, a reusable component is a piece of UI that can be used in various parts of an application to build more than one UI instance.

For instance, we can have a Button component that displays different texts on different pages.

# object.assign() vs spread operator

The difference is that Object.assign changes the object in-place, while the spread operator (...) creates a brand new object, and this will break object reference equality.

# LEFT JOIN

The LEFT JOIN command returns all rows from the left table,

and the matching rows from the right table. The result is NULL from the right side, if there is no match.

# INNER JOIN

The INNER JOIN command returns rows that have matching values in both tables.

# RIGHT JOIN

The RIGHT JOIN command returns all rows from the right table,

and the matching records from the left table. The result is NULL from the left side, when there is no match.

# What is Primary Key?

A primary key constrain is a column or group of columns that uniquely identifies every row in the table of the relational database management system. It cannot be a duplicate, meaning the same value should not appear more than once in the table.

# What is Foreign Key?

Foreign key is a column that creates a relationship between two tables. The purpose of the Foreign key is to maintain data integrity and allow navigation between two different instances of an entity. It acts as a cross-reference between two tables as it references the primary key of another table. Every relationship in the database should be supported by a foreign key.

# Indexing

A SQL index is a quick lookup table for finding records users need to search frequently. An index is small, fast, and optimized for quick lookups. It is very useful for connecting the relational tables and searching large tables.

# What is CI / CD experience while working with team?

**Explain your experience using continuous integration and continuous deployment pipelines (CI/CD).**

I built and deployed web applications on CI/CD.

We have 3 stages in development, develop, staging, and production.

By Using CircleCI and defining github actions, I built a pipleline of each step that automatically deploy updates to the live server.

Continuous integration is mainly focused on automatically building and testing code, as compared to continuous delivery, which automates the entire software release process up to production

I also implemented fully automated CI/CD

A fully automated pipeline is one, that provides the capability to deploy new developer changes to production or any non-production environment with seamless experience and without human intervention. A fully automated pipeline may consist of many stages. Once developer commits the code changes then no human intervention is required.

When the developer commits new code changes to source code manager, latest changes are build and artifacts are created on server. And then Unit Tests or Integration testing is performed on server. When the test is successful, the latest build version is deployed.

I also used Jenkins, Github/Gitlab actions, Azure DevOps to create CI/CD pipelines

(These are advanages of CI/CD)

CI/CD setup provides faster release rate as it merges the small code changes very quickly and make the code ready for production. CI helps to find early stage failure like merge issues, build failure, integration test failure etc. that will improve transparency within team. CD helps to make code available to end users.

# DevOps explained

DevOps can be best explained as people working together to conceive, build and deliver secure software at top speed. DevOps practices enable software developers (devs) and operations (ops) teams to accelerate delivery through automation, collaboration, fast feedback, and iterative improvement.

Stemming from an [Agile approach](https://about.gitlab.com/topics/agile-delivery/) to software development, a DevOps delivery process expands on the cross-functional approach of building and shipping applications in a faster and more iterative manner. In adopting a DevOps development process, you are making a decision to improve the flow and value delivery of your application by encouraging a more collaborative environment at all stages of the development cycle.

DevOps represents a change in mindset for IT culture. In building on top of Agile, lean practices, and systems theory, DevOps focuses on incremental development and rapid delivery of software. Success relies on the ability to create a culture of accountability, improved collaboration, empathy, and joint responsibility for business outcomes.

# What is the goal of DevOps?

DevOps represents a change in mindset for IT culture. In building on top of [Agile](https://about.gitlab.com/solutions/agile-delivery/) and lean practices, DevOps focuses on incremental development and rapid delivery of software. Success relies on the ability to create a culture of accountability, improved collaboration, empathy, and joint responsibility for business outcomes.

Adopting a DevOps strategy enables businesses to increase operational efficiencies, deliver better products faster, and reduce security and compliance risk.

# How to use UnitTest(enzyme) in React?

Install Enzyme and the Preact adapter using:

npm install --save-dev enzyme enzyme-adapter-preact-pure

In the test setup code, I'll need to configure Enzyme to use the Preact adapter:

output = mount(<Component1 />);

output.find(Component2).find(Button).simulate("click")

Enzyme uses the adapter library it has been configured

with to render a component and its children.

The adapter then converts the output

to a standardized internal representation (a "React Standard Tree").

Enzyme then wraps this with an object that has methods

to query the output and trigger updates.

The wrapper object's API uses CSS-like selectors to locate parts of the output.

This imports Enzyme and sets up the adapter to run our tests.

**it or test** You would pass a function to this method,

and the test runner would execute that function as a block of tests.

**describe** This optional method is for grouping any number of it or test statements.

**expect** This is the condition that the test needs to pass.

It compares the received parameter to the matcher.

It also gives you access to a number of matchers that let you validate different things.

**mount** This method renders the full DOM,

including the child components of the parent component,

in which we are running the tests.

**shallow** This renders only the individual components that we are testing.

It does not render child components. This enables us to test components in isolation.

# How can optimize react application?

* 1. Using Immutable Data Structures
  2. Using Function / Stateless Components and React.PureComponent
  3. Need to Keep Multiple Chunk Files
  4. Using Production Mode Flag in Webpack
  5. Dependency optimization
  6. Use React.Fragments(empty angle brackets) to Avoid Additional HTML Element Wrappers
  7. Throttling and Debouncing Event Action in JavaScript

**Throttling**

In a nutshell, throttling means delaying function execution.

So instead of executing the event handler/function immediately,

we’ll be adding a few milliseconds of delay when an event is triggered.

This can be used when implementing infinite scrolling,

for example. Rather than fetching the next result

set as the user is scrolling, we can delay the XHR call.

**Debouncing**

Unlike throttling, debouncing is a technique to

prevent the event trigger from being fired too often.

If we are using lodash, we can wrap the function we want to call

in lodash’s debounce function.

* 1. Avoid using index as Key for map
  2. Spreading props on DOM elements

We should avoid spreading properties into a DOM element as it adds unknown HTML attribute, which is unnecessary and a bad practice.

* 1. Use Reselect in Redux to Avoid Frequent Re-render

Reselect is a simple selector library for Redux, which can be used for building memorized selectors. We can define selectors as a function, retrieving snippets of the Redux state for React components.

* 1. Avoid Async Initialization in componentWillMount()
  2. Memoize React Components

Memoization is an optimization technique used primarily to speed up computer programs

by storing the results of expensive function calls and

returning the cached result when the same inputs occur again.

A memoized function is usually faster because if the function is called with the same values as the previous one then instead of executing function logic it would fetch the result from cache.

* 1. Css Animations Instead of JS Animations
  2. Consider Server-side Rendering

# Tell us about accomplishments you're proud of from your e-commerce experiences

I'm proud of two major accomplishments in my e-commerce experiences. The first one is when I increased the customer base of my online store by 17% by utilizing pay-per-click advertising campaigns. The second is when I worked on an e-commerce site for a client and raised their returning customer rate by 15%

# What do you believe are the core features of strong e-commerce platforms?

I believe there are multiple core features of strong e-commerce platforms. The first one is website branding and usability, which can ensure the site is easily identifiable with a company brand and offers users easy navigation. User interactivity is also important because customers may be more likely to continue exploring e-commerce sites that offer engaging experiences. Another essential core feature is online security to protect customer transactions and form a strong connection between a company's online and physical store experiences.

# How do you usually monitor customer experiences in E-commerce?

Usually, I monitor customer experiences by collecting feedback regularly. Whenever a customer purchases an item, I add a survey link to their automated confirmation email to ask them to provide feedback about their experience. I also find it useful to add another survey link to their automated delivery email to ensure I'm gathering feedback at different stages of their customer journey

# UI / UX design Process

Project Definition & Scope, Understanding the Problem, Ideation, Sketching Prototyping, High-Fidelity Mockups & Prototypes, Usability Testing, Design Handoff

# Explain what is an algorithm in computing?

An algorithm is a well-defined computational procedure that take some value as input and generate some value as output. In simple words, it’s a sequence of computational steps that converts input into the output.

# Explain what is Quick Sort algorithm?

Quick Sort algorithm has the ability to sort list or queries quickly. It is based on the principle of partition exchange sort or Divide and conquer. This type of algorithm occupies less space, and it segregates the list into three main parts.

* Elements less than the Pivot element
* Pivot element
* Elements greater than the Pivot element

# Explain what is time complexity of Algorithm?

Time complexity of an algorithm indicates the total time needed by the program to run to completion. It is usually expressed by using the **big O notation**

# What are the types of Notation used for Time Complexity?

The types of Notations used for Time Complexity includes

* **Big Oh:** It indicates “fewer than or the same as” <expression>iterations
* **Big Omega:** It indicates “more than or same as” <expression>iterations
* **Big Theta:** It indicates “the same as”<expression>iterations
* **Little Oh:** It indicates “fewer than” <expression>iterations
* **Little Omega:** It indicates “more than” <expression>iterations

# How binary search works?

In [binary search](https://www.guru99.com/binary-search.html), we compare the key with the item in the middle position of the array. If the key is less than the item searched then it must lie in the lower half of the array, if the key is greater than the item searched than it should be in upper half of the array.

# Whether it is possible to use binary search for linked lists?

Since random access is not acceptable in linked list, it is impossible to reach the middle element of O(1) time. Thus, binary search is not possible for linked list.

# Explain what is heap sort?

[Heap-sort](https://www.guru99.com/heap-sort.html) can be defined as a comparison based sorting algorithm. It divides its input into the unsorted and sorted region, until it shrinks the unsorted region by eliminating the smallest element and moving that to the sorted region.

# What is Skip list?

Skip list the method for data structuring, where it allows the algorithm to search, delete and insert elements in a symbol table or dictionary. In a skip list, each element is represented by a node. The search function returns the content of the value related to key. The insert operation associates a specified key with a new value, while the delete function deletes the specified key.

# Explain what is Space complexity of insertion sort algorithm?

Insertion sort is an in-place sorting algorithm which means that it requires no extra or little. storage. For insertion sort, it requires only single list elements to be stored out-side the initial data, making the space-complexity 0(1).

# Explain what a “Hash Algorithm” is and what are they used for?

“Hash Algorithm” is a hash function that takes a string of any length and decreases it to a unique fixed length string. It is used for password validity, message & data integrity and for many other cryptographic systems.

# Explain how to find whether the linked list has a loop?

To know whether the linked list has a loop, we will take two pointer approach. If we maintain two pointers, and we increase one pointer after processing two nodes and other after processing every node, we are likely to encounter a situation where both the pointer will be pointing to the same node. This will only occur if linked list has a loop.

# Explain how encryption algorithm works?

Encryption is the process of converting plaintext into a secret code format referred as “Ciphertext”. To convert the text, algorithm uses a string of bits referred as “keys” for calculations. The larger the key, the greater the number of potential patterns for creating cipher text. Most encryption algorithm use codes fixed blocks of input that have length about 64 to 128 bits, while some uses stream method.

# List out some of the commonly used cryptographic algorithms?

Some of the commonly used cryptographic algorithms are

* 3-way
* Blowfish
* CAST
* CMEA
* GOST
* DES and Triple DES
* IDEA
* LOKI and so on

# Explain what is the difference between best case scenario and worst case scenario of an algorithm?

* **Best case scenario:** Best case scenario for an algorithm is explained as the arrangement of data for which the algorithm performs best. For example, we take a binary search, for which the best case scenario would be if the target value is at the very center of the data we are searching. The best case time complexity would be 0 (1)
* **Worst case scenario:** It is referred for the worst set of input for a given algorithm. For example quicksort, which can perform worst if we select the largest or smallest element of a sublist for the pivot value. It will cause quicksort to degenerate to O (n2).

# Explain what is Radix Sort algorithm?

Radix sort puts the element in order by comparing the digits of the numbers. It is one of the linear sorting algorithms for integers.

# Explain what is a recursive algorithm?

Recursive algorithm is a method of solving a complicated problem by breaking a problem down into smaller and smaller sub-problems until we get the problem small enough that it can be solved easily. Usually, it involves a function calling itself.

# Mention what are the three laws of recursion algorithm?

All recursive algorithm must follow three laws

* It should have a base case
* A recursive algorithm must call itself
* A recursive algorithm must change its state and move towards the base case

# Explain what is bubble sort algorithm?

[Bubble sort algorithm](https://www.guru99.com/bubble-sort.html) is also referred as sinking sort. In this type of sorting, the list to be sorted out compares the pair of adjacent items. If they are organized in the wrong order, it will swap the values and arrange them in the correct order.

# What is Software Development KPI?

Key performance indicators (KPIs) are values that measure the performance of the business overall.

In the context of software development, KPIs indicate how well development efforts are aligning with business objectives.

Key performance indicators (KPIs) indicate how well development efforts are aligning with business objectives.

Below are 10 essential software KPIs:

**1. Velocity**

Velocity refers to how much work the team can complete during a sprint.

In agile development, a sprint is a set period of time where specific tasks are set out to be done.

It only takes about three sprints before I will have a good idea of the average velocity of the team.

Using velocity, I can estimate how realistic the team’s goals are.

**2. Sprint Burndown**

Sprint burndown is a more narrow metric that measures how much work is actually completed during a sprint.

Using sprint burndown as a software metrics helps teams adjust their performance when

the measurement doesn’t hold true to predictions

**3. Release Burndown**

Release burndown takes note of release progress. This metric is larger in scope than sprint burndown.

And the metric is useful because it can guide teams in managing the release of a product.

Software development teams can utilize a release burndown chart to find out whether they’re behind of, ahead of, or exactly on schedule.

**4. Cycle Time**

Cycle time is a software KPI that measures how much time is spent working on a certain task. Development teams use cycle time charts to gauge the efficiency of the software development process.

**5. Cumulative Flow**

Cumulative flow demonstrates what state the software tasks, or tickets, are in by employing a visual diagram.

Different colors on the diagram would represent different stages like ‘Approved’. ‘In Progress’, ‘Backlog’, and more. These colors are arranged in bands with the width of the band correlating to cycle time.

**6. Flow Efficiency**

Flow efficiency measures the ratio between active time and total time.

Often work in progress doesn’t really mean work in progress but really time standing still. There may be waiting periods where developers cannot immediately move from one task or project to another.

I can calculate flow efficiency by dividing the time I actively spend working by the total cycle time.

**7. Code Coverage**

Code coverage is a software KPI software development teams use to measure code quality.

Also called test coverage, this metric determines how much of source code executes when it is being tested.

Code that for some reason does not execute likely has some undetected bugs.

While we shouldn’t aim for 100% code coverage, the higher the code coverage the better. And the less debugging we’ll have to do.

**8. Code Stability**

Code stability is difficult to measure. Stable code means that there are little changes to the software product that could potentially harm the business or software.

Some developers decide to chart the frequency of code changes. Others think of stability in terms of what percentage of deployed code results in downtime.

**9. Code Simplicity**

Code simplicity is a more general software engineering KPI and there are multiple metrics we can use to measure it.

Cyclomatic complexity, for one, is a quantitative measure of the number of independent paths the code must take. Fewer paths are a good sign.

Generally, simpler code is easier to test and maintain.

**10. Code Churn**

Code churn can be a measure of code stability as it refers to how often code changes over time.

If we need to rewrite code often to accommodate a new feature, then the software system involves high maintenance and is thus high risk.

**How to improve your speed index: Optimize content efficiency**

WebPageTest is the tool for measuring the speed of site.

**Speed index**

Speed index describes the average amount of time it takes for the above-the-fold content of the page to appear on a user’s screen.

It’s not a specific point-in-time metric, but rather an aggregate of the page’s metrics that calculates how “complete” the page is throughout the various stages of loading.

Speed index depends on how the page is built. If we have a lot of heavy above-the-fold content, the score might be poor.

Interval time \* (1 - visual completeness/100)

**1. Eliminate unnecessary downloads**

We don't need to include unnecessary resources.

Sometimes, resources that are meant to engage users can slow down my whole operation, which actually results in users turning away. Before I add any such resource, I should come up with ways to measure its effectiveness. For example, if I have a photo carousel or a third party widget that displays related content at the top of my homepage, I should track how often visitors click on those features to determine if they are really worth the overhead.

**2. Compress the data**

Compressing out website's assets is an easy and efficient way to save space, reduce load time, and improve the overall speed of the site. Whether I'm using a tried and true method such as Gzip or a newer compression algorithm such as Brotli, compressing my data is a must.

**3. Create a caching hierarchy**

We need to take advantage of caching as much as possible.

We need to come up the general caching strategy to follow:

Conduct auditing to determine the optimal cache lifetime for each resource.

Make sure the server specifies a cache validator.

Implement cache busting.

Use intermediary caches such as content delivery networks.

**4. Optimize the images**

We need to keep the rules to follow:

Use favor vector formats since they are resolution and scale independent.

Use lossless or lossy compression to reduce the overall size of images

Consider the priorities and choose the most appropriate image format for each individual asset.

Don't be afraid to experiment with image quality settings as we can sometimes make website significantly faster by lowering image resolutions just a little bit.

Resize images on the server so that the display size matches the original size of the image or use srcset sizing attributes.

If possible, invest in an automated image compression tool that will keep the image assets optimized at all times.

**5. Optimize the fonts**

Webfonts can help create a unique user experience that appropriately represents the brand.

For Implementing optimize the fonts, we need to keep this rules:

Limit the number of fonts we use, and use as few variants as possible.

Subset the fonts so that we only load the characters we need.

Provide every font in every format including WOFF2, WOFF, EOT and TTF. Remember to apply Gzip compression to EOT and TTF formats since they're not compressed by default.

Since fonts are static and rarely require updates, we can optimize the caching strategy by specifying revalidation tokens.

# What Is TDD (Test Driven Development)?

TDD is a software development approach where developers write tests for the desired functionality before writing the actual implementation code. The process follows these steps:

* Write a test for a small piece of the desired functionality
* Run the test, which should fail because the implementation code doesn't exist yet.
* Write the minimum amount of code necessary to make the test pass
* Refactor the code as necessary
* Repeat the process for each small piece of desired functionality

The tests serve as a way to verify that the code is working as intended and provide a safety net for future changes. By writing tests first, developers can focus on designing the functionality and edge cases before getting bogged down in implementation details.

**Frontend Skills:**

component management using design patterns, query optimizations and caching of GraphQL

# What is Design Pattern?

A design pattern provides a general reusable solution for the common problems that occur in software design. The pattern typically shows relationships and interactions between classes or objects. The idea is to speed up the development process by providing well-tested, proven development/design paradigms. Design patterns are programming language independent strategies for solving a common problem. That means a design pattern represents an idea, not a particular implementation. By using design patterns, you can make your code more flexible, reusable, and maintainable.

It’s not mandatory to always implement design patterns in your project. Design patterns are not meant for project development. Design patterns are meant for common problem-solving. Whenever there is a need, you have to implement a suitable pattern to avoid such problems in the future. To find out which pattern to use, you just have to try to understand the design patterns and their purposes. Only by doing that, you will be able to pick the right one.

Goal:

Understand the purpose and usage of each design pattern in order to pick and implement the correct pattern as needed.

Example:

In many real-world situations, we want to create only one instance of a class. For example, there can be only one active president of a country at any given time. This pattern is called a Singleton pattern. Other software examples could be a single DB connection shared by multiple objects as creating a separate DB connection for every object is costly. Similarly, there can be a single configuration manager or error manager in an application that handles all problems instead of creating multiple managers.

Types of Design Patterns

There are mainly three types of design patterns:

**Creational**

These design patterns are all about class instantiation or object creation. These patterns can be further categorized into Class-creational patterns and object-creational patterns. While class-creation patterns use inheritance effectively in the instantiation process, object-creation patterns use delegation effectively to get the job done.

Creational design patterns are the Factory Method, Abstract Factory, Builder, Singleton, Object Pool, and Prototype.

Use case of creational design pattern-

1) Suppose a developer wants to create a simple DBConnection class to connect to a database and wants to access the database at multiple locations from code, generally what the developer will do is create an instance of DBConnection class and use it for doing database operations wherever required. This results in creating multiple connections from the database as each instance of DBConnection class will have a separate connection to the database. In order to deal with it, we create DBConnection class as a singleton class, so that only one instance of DBConnection is created and a single connection is established. Because we can manage DB Connection via one instance, we can control load balance, unnecessary connections, etc.

2) Suppose you want to create multiple instances of a similar kind and want to achieve loose coupling then you can go for Factory pattern. A class implementing factory design pattern works as a bridge between multiple classes. Consider an example of using multiple database servers like SQL Server and Oracle. If you are developing an application using SQL Server database as back end, but in the future need to change the database to the oracle, you will need to modify all your code, so as factory design patterns maintain loose coupling and easy implementation, we should go for the factory design pattern in order to achieve loose coupling and the creation of a similar kind of object.

**Structural**

These design patterns are about organizing different classes and objects to form larger structures and provide new functionality.

Structural design patterns are Adapter, Bridge, Composite, Decorator, Facade, Flyweight, Private Class Data, and Proxy.

Use Case Of Structural Design Pattern-

1) When 2 interfaces are not compatible with each other and want to establish a relationship between them through an adapter it’s called an adapter design pattern. The adapter pattern converts the interface of a class into another interface or class that the client expects, i.e adapter lets classes work together that could not otherwise because of incompatibility. so in these types of incompatible scenarios, we can go for the adapter pattern.

**Behavioral**

Behavioral patterns are about identifying common communication patterns between objects and realizing these patterns.

Behavioral patterns are Chain of responsibility, Command, Interpreter, Iterator, Mediator, Memento, Null Object, Observer, State, Strategy, Template method, Visitor

Use Case of Behavioral Design Pattern-

* 1. The template pattern defines the skeleton of an algorithm in an operation deferring some steps to sub-classes. The template method lets subclasses redefine certain steps of an algorithm without changing the algorithm structure. For example, in your project, you want the behavior of the module to be able to extend, such that we can make the module behave in new and different ways as the requirements of the application change, or to meet the needs of new applications. However, no one is allowed to make source code changes to it, i.e you can add but can’t modify the structure in those scenarios a developer can approach template design pattern.

# What is Caching of Graphql?

In an endpoint-based API, clients can use HTTP caching to easily avoid refetching resources, and for identifying when two resources are the same. The URL in these APIs is a globally unique identifier that the client can leverage to build a cache. In GraphQL, though, there's no URL-like primitive that provides this globally unique identifier for a given object. It's hence a best practice for the API to expose such an identifier for clients to use.

# What is query optimizations?

Query optimization is of great importance for the performance of a relational database, especially for the execution of complex SQL statements. A query optimizer decides the best methods for implementing each query.

The query optimizer selects, for instance, whether or not to use indexes for a given query, and which join methods to use when joining multiple tables. These decisions have a tremendous effect on SQL performance, and query optimization is a key technology for every application, from operational Systems to data warehouse and analytical systems to content management systems.

# Explain the phases in a typical SDLC process briefly.

* Requirements gathering and analysis: Understanding and documenting the requirements for the software project.
* Design: Developing a plan for how the software will meet the requirements and be built.
* Implementation or coding: Writing the code for the software.
* Testing: Verifying that the software meets the requirements and identifying and fixing any defects.
* Deployment: Installing and distributing the software to its intended users.
* Maintenance: Providing ongoing support and making changes to the software as needed.

# ****SOAP vs REST API?****

* Architectural style: SOAP uses a standards-based approach with strict guidelines, while REST has a flexible architecture that uses HTTP requests.
* Data format: SOAP uses XML, while REST can use XML, JSON, or other data formats.
* Protocol: SOAP requires a different protocol, such as HTTP or SMTP, while REST primarily uses HTTP.
* Operations: SOAP supports a wider range of operations, such as transaction management, while REST focuses on CRUD operations.
* Error handling: SOAP has a standardized way of handling errors, while REST relies on HTTP status codes.
* Complexity: SOAP has a more complex implementation, while REST is simpler and more lightweight.
* Security: SOAP has built-in security features, such as WS-Security, while REST relies on SSL/TLS for security.

# Media Query CSS

A media query in CSS allows you to apply different styles to a document based on the characteristics of the device it is being displayed on. It starts with the @media rule, followed by a media type (such as screen, print, or speech) and optional expressions that check for specific conditions, such as the height and width of the viewport.

Here's an example of a media query:

@media screen and (max-width: 600px) {

body {

font-size: 14px;

}

}

In this example, the styles within the media query will only be applied when the screen width is less than or equal to 600 pixels.

# React vs React Native

React and React Native are both JavaScript libraries for building user interfaces, but they have some important differences.

React is a JavaScript library for building user interfaces in web applications. It allows you to create reusable UI components, manage the state of your application, and efficiently update the view whenever the state changes. React uses a virtual DOM to efficiently render changes to the UI, making it fast and efficient.

React Native, on the other hand, is a framework for building native mobile applications using React. Instead of rendering to the web browser's DOM, React Native uses native components that are rendered to the native platform's UI. This allows you to build truly native apps that look and feel like the apps built with Java or Swift.

In summary, React is used for building web applications, while React Native is used for building native mobile apps. While React and React Native share many concepts and ideas, the underlying technologies and libraries are different, and the way you build apps is also different. However, if you know React, you will have a head start on learning React Native, since many of the concepts and patterns are the same.

# Can you just explain more about the technical implementation of server side fragmentation? How it is displayed on the UI very quickly?

Server-side fragmentation involves dividing a large data set into smaller parts, or fragments, and distributing them across multiple servers.

This can improve the performance of applications by reducing the amount of data that needs to be transferred over the network and processed by the client.

In order to display the fragmented data quickly on the UI, the client typically requests only the fragments it needs at a given time.

This minimizes the amount of data that needs to be transferred over the network and processed by the client, which can help improve the responsiveness of the user interface.

Additionally, to further optimize the display of fragmented data on the UI, the server can use techniques such as caching, data compression, and prefetching to reduce the latency of data transfers and minimize the impact on user experience.

# Can you just explain more about the technical implementation of client side fragmentation?

# How it is displayed on the UI very quickly?

Client-side fragmentation involves dividing a large data set into smaller parts, or fragments, on the client side before it is displayed on the user interface. This can improve the performance of applications by reducing the amount of data that needs to be transferred over the network and processed by the client.

In order to display the fragmented data quickly on the UI, the client may use techniques such as lazy loading, where only the fragments that are currently visible to the user are loaded and displayed, and dynamic loading, where new fragments are loaded and displayed as the user scrolls or interacts with the interface.

Additionally, to further optimize the display of fragmented data on the UI, the client can use techniques such as caching, data compression, and optimization algorithms to reduce the latency of data transfers and minimize the impact on user experience. The client can also use multithreading or asynchronous programming to parallelize the processing of multiple fragments and speed up the overall display of the data.

# What is bearer token?

A bearer token is a type of access token in authorization protocols that can be used to access resources or APIs. It is called a "bearer" token because it can be passed in the HTTP "Authorization" header as a simple string value, without the need for any additional authentication mechanism.

Bearer tokens are commonly used in OAuth 2.0 and OpenID Connect authentication protocols and are issued by an authorization server to a client after the client has been successfully authenticated. The bearer token can then be passed to a resource server to access protected resources or APIs.

The bearer token contains information about the client's authorization to access resources or APIs, such as the scope of access granted, the expiration time of the token, and any other relevant information. The resource server can use the information in the bearer token to determine if the client is authorized to access the requested resources or APIs.

# Let's talk the example of an online shopping website like ecommerce website. The user would need data from just two services. Let's say list service and wish list service. We might want to take a note of it because the question is a little bit long and 20% of the time the user would hit all the five different other services, including the list service and the wish list service. Tell me how you design your backend? will your client always call all these different services from UI? When you client only call one service that would aggregate from different five services and you send one payload back to UI?

In the case of an online shopping website, it's important to balance the trade-off between optimization and scalability. One way to design the backend could be to have the client call all the different services as needed. This can ensure that the client is able to receive the most update data from each service, but it can also result in a large number of network calls and higher latency.

An alternative approach could be to create a single service that aggregates data from the list service and the wish list service, as well as any other relevant services. This service would be responsible for fetching the data from each of the underlying services and returning a single payload to the client. This can reduce the number of network calls and lower latency, but it can also result in a delay in updating the client with the most recent data.

Ultimately, the design of the backend will depend on the specific requirements of the application, including the number of concurrent users, the volume of data, the frequency of updates, and the desired level of performance. A good approach could be to use a combination of these two strategies, using the aggregating service for frequently requested data and calling the individual services as needed for less frequently requested data.

# How to work as a team leader in your previous company?

* Communicate effectively: Ensure clear and consistent communication with your team to build trust and align expectations.
* Set goals and expectations: Clearly define team and individual goals, and establish expectations for performance and behavior.
* Lead by example: Demonstrate the behaviors and attitudes you expect from your team, and model the values you want to promote.
* Empower your team: Give team members the resources and authority they need to be successful and feel valued.
* Foster a positive work environment: Create a supportive and inclusive work environment that promotes collaboration, creativity, and growth.
* Provide regular feedback: Offer both positive reinforcement and constructive criticism to help team members grow and improve.
* Recognize and reward success: Acknowledge and celebrate team and individual successes to boost morale and motivation.

# How did you integrate payment gateway in your website

A payment gateway is a service that facilitates secure online transactions by processing credit card or other forms of payment information from a customer and transmitting it to the acquiring bank or payment processor. Integrating a payment gateway into a website is a crucial step in building an e-commerce website, as it allows the website to accept payments from customers.

Here are the general steps to integrate a payment gateway into a website:

Choose a payment gateway: There are many payment gateway providers available, each offering a range of services and fees. Choose a payment gateway provider that fits your needs, taking into consideration the countries you want to accept payments from, the types of payments you want to accept, and any other specific requirements you have.

Set up an account: Once you have chosen a payment gateway, you will need to set up an account with the provider. This will typically involve providing business information, such as your company name, address, and tax identification number, as well as setting up security credentials for your account.

Implement the payment gateway API: Most payment gateways provide APIs that allow developers to integrate their services into websites and web applications. The API will provide instructions on how to process payments, including how to transmit payment information to the gateway and how to handle the response from the gateway.

Test the integration: Once the payment gateway API has been integrated into your website, you should thoroughly test the integration to ensure that payments are being processed correctly and that all security measures are in place.

Launch the website: After testing is complete, you can launch your website and start accepting payments from customers.

These are the general steps to integrate a payment gateway into a website. The exact process will vary depending on the payment gateway provider and the specifics of your website, but the basic principles are the same. By following these steps, you can ensure a smooth and secure integration of a payment gateway into your website, allowing you to start accepting payments from customers.

# What is the return value when left join query is executed

A left join in SQL returns all the rows from the left table (also known as the "left-side" table), along with the matching rows from the right table (also known as the "right-side" table). If there is no match for a row in the right table, NULL values will be returned for the columns from the right table.

The result of a left join query is a new table that combines the columns from both the left and right tables. The number of rows in the result table will be equal to the number of rows in the left table. The columns from the right table will have NULL values for any rows where there is no match.

In summary, the return value of a left join query is a new table that combines the columns from the left and right tables, with NULL values for any non-matching rows in the right table.

The return value of a left join query in most programming languages (including PHP, which is commonly used with databases) is typically an object or an array, depending on the specific implementation and the programming language being used.

For example, in PHP, when using the PDO (PHP Data Objects) extension to execute a left join query, the result set is returned as a PDOStatement object, which can be iterated over to access the individual rows. The individual rows can be accessed as arrays or as objects, depending on how the query is executed.

In other programming languages, such as Python or Java, the return value of a left join query may be a list or an array of objects, depending on the specific libraries or frameworks being used to interact with the database.

So, in general, the return value of a left join query can be considered an object or an array, depending on the specific implementation and programming language.

# How did you use azure in your project?

I have over 7 years of experience working with Microsoft Azure and have developed a deep understanding of the platform and its capabilities. During this time, I have successfully implemented various cloud solutions that meet the needs of my clients and have helped them realize the full potential of Azure.

I have extensive experience in designing and deploying highly available, scalable and secure solutions using Azure services such as Azure Virtual Machines, Azure Storage, Azure SQL Database, Azure App Service, and Azure Functions. I have also worked with Azure DevOps for continuous integration and deployment and have a strong understanding of Azure networking, security, and identity management.

In addition to the technical aspects of Azure, I have a strong understanding of the business side of cloud solutions and have worked closely with clients to develop cost-effective solutions that meet their budget constraints. I have also helped clients migrate their on-premises solutions to Azure and have worked with them to develop disaster recovery and business continuity plans.

Overall, my extensive experience with Microsoft Azure and my commitment to delivering quality solutions have made me a valuable asset to my clients. I am always seeking new challenges and opportunities to further enhance my skills and knowledge of Azure.

Here are the steps to use Azure in a project:

* Define the project requirements and goals: Before using Azure, it is important to have a clear understanding of the project requirements and goals. This will help you determine which Azure services are most appropriate for your project.
* Choose the Azure services that meet your needs: Azure offers a wide range of services, including virtual machines, storage, databases, web applications, and more. Choose the services that best meet the needs of your project.
* Create an Azure account: To use Azure, you'll need to create a free or paid Azure account. This will give you access to all of the services you need to build and deploy your project.
* Set up the Azure services you need: Once you have an Azure account, you can start setting up the services you need for your project. This may include creating virtual machines, storage accounts, and databases.
* Develop and test your application: Using the Azure services you've set up, you can now develop and test your application. You can use the Azure portal, CLI, or Visual Studio to manage your services and deploy your code.
* Deploy your application: Once your application is fully tested, you can deploy it to Azure. This will make your application available to users and customers.
* Monitor and manage your application: To ensure that your application is running smoothly, you'll need to monitor and manage it regularly. Azure provides various tools and services to help you with this, such as Azure Monitor and Azure Log Analytics.

By following these steps, you can use Azure in your project and take advantage of its powerful cloud services to build and deploy high-quality, scalable, and secure applications.

# How to implement ci / cd pipe line in your project?

To implement a continuous integration and continuous delivery (CI/CD) pipeline in your project, you can follow these steps:

* Choose a version control system: The first step in implementing a CI/CD pipeline is to choose a version control system, such as Git, to manage your code.
* Set up a build system: Next, you'll need to set up a build system, such as Azure DevOps, Jenkins, or CircleCI, to automate the process of building and testing your code. This will ensure that your code is tested and validated every time you make changes.
* Automate testing: Your build system should also automate testing, including unit tests, integration tests, and acceptance tests. This will help you catch bugs and issues early in the development process, before they make it to production.
* Implement continuous integration: Continuous integration is the practice of automatically building and testing your code every time you push changes to your version control system. By integrating your build system with your version control system, you can ensure that your code is always up-to-date and ready to be deployed.
* Set up a continuous delivery pipeline: Once your build system is integrated with your version control system and is automatically testing your code, you can set up a continuous delivery pipeline. This pipeline will automatically deploy your code to production or staging environments, so you don't have to do it manually.
* Monitor and manage your pipeline: To ensure that your CI/CD pipeline is working as expected, you'll need to monitor and manage it regularly. This may include fixing any issues with your build or deployment process, or updating your pipeline as your project evolves.

By following these steps, you can implement a robust CI/CD pipeline in your project and ensure that your code is built, tested, and deployed automatically and efficiently.

# How to use docker in your project?

Here are the steps to use Docker in your project:

* Install Docker: To use Docker, you'll need to install it on your local machine. Docker can be installed on Windows, Mac, and Linux, so you'll need to choose the right installation package for your operating system.
* Create a Dockerfile: A Dockerfile is a script that contains all the instructions required to build a Docker image. You'll need to create a Dockerfile for your project and define the base image, the software and tools you need, and any configuration files or scripts required to run your application.
* Build a Docker image: Once you have a Dockerfile, you can build a Docker image by running the docker build command. This will take the instructions from your Dockerfile and create an image that you can use to run your application.
* Run a Docker container: Once you have a Docker image, you can run a Docker container by using the docker run command. This will start a new container from your image and run your application inside it.
* Publish your Docker image: If you want to share your Docker image with others, you can publish it to a Docker registry, such as Docker Hub or the Azure Container Registry. This will make it easy for others to run your application and use it in their own projects.
* Monitor and manage your Docker containers: To ensure that your Docker containers are running smoothly, you'll need to monitor and manage them regularly. You can use the docker ps command to see a list of running containers, or the docker logs command to view the logs of a specific container.

By following these steps, you can use Docker in your project and take advantage of its powerful containerization capabilities to build, run, and deploy your application.

# Write the professional GCP experience in hotel booking system with example answer

One example of using GCP services in our hotel booking system is with Cloud Functions. We use Cloud Functions to trigger a notification to our customer service team whenever a customer books a room, ensuring timely customer support. This serverless approach allows us to handle high traffic and spikes in usage without worrying about server capacity or downtime.

We also use GCP's Kubernetes Engine to deploy and manage our microservices architecture. By containerizing our services and deploying them on Kubernetes Engine, we can easily scale up or down based on demand, and we can deploy updates without any downtime. Additionally, we use GCP's Cloud SQL for database management, which ensures automatic backups and high availability.

Another example of using GCP services in our hotel booking system is with BigQuery. We use BigQuery to store and analyze large amounts of booking data, allowing us to gain insights into customer preferences and behaviors. We also use Google Cloud Pub/Sub to stream booking data in real-time to our data pipeline, which then feeds into our machine learning models. This allows us to predict customer demand and optimize pricing strategies.

In addition to these services, we also prioritize security in our hotel booking system. We use GCP's Identity and Access Management (IAM) to manage access to our resources and enforce least privilege principles. We also leverage GCP's Security Command Center for continuous security monitoring and Cloud Security Scanner to identify and fix vulnerabilities.

# Describe your experience with monitoring and system design.

In my role as a systems engineer, I have extensive experience with monitoring and system design. I have designed and implemented monitoring solutions for various systems, ranging from on-premises infrastructure to cloud-based applications.

For example, in a recent project, I was tasked with designing a monitoring solution for a cloud-based e-commerce platform. We used a combination of GCP services, including Cloud Monitoring, Cloud Logging, and Cloud Trace, to ensure the health and performance of our system. We created custom dashboards in Cloud Monitoring to visualize key metrics and set up alerting based on predefined thresholds. We also used Cloud Logging to store logs and troubleshoot issues, and Cloud Trace to trace requests and identify performance bottlenecks.

In terms of system design, I have designed and implemented solutions for various clients, ranging from small businesses to large enterprises. I always prioritize scalability, resilience, and security in my designs. For example, I designed a highly available and scalable architecture for a media streaming platform using GCP services. We used Kubernetes Engine to deploy containerized applications and implemented a multi-region deployment with automatic failover. We also used Cloud CDN to improve performance and Cloud Armor to protect against DDoS attacks.

In addition to designing and implementing monitoring and system solutions, I also prioritize continuous improvement. I regularly analyze metrics and logs to identify areas of improvement and work with development teams to implement changes that improve system health and performance.

# Describe your experience with designing and implementing back-end APIs.

As a software engineer, I have extensive experience with designing and implementing back-end APIs for various applications. I always prioritize scalability, reliability, and security in my designs.

For example, in a recent project, I was tasked with designing and implementing a back-end API for a mobile banking application. We used a microservices architecture with a RESTful API to ensure modularity and scalability. We used Google Cloud Platform's App Engine to host our API, allowing us to scale our API automatically based on demand.

We designed our API with strict security measures, implementing user authentication and authorization with JSON Web Tokens (JWT) and OAuth 2.0. We also implemented rate limiting and throttling to prevent API abuse.

In terms of implementation, we used Node.js and Express.js to build our API. We followed best practices for API design, such as using HTTP status codes and RESTful resource naming conventions. We also implemented data validation and sanitization to prevent malicious attacks.

To ensure the reliability of our API, we implemented automated testing and continuous integration and deployment (CI/CD) pipelines. We used Google Cloud Build for building and testing our API and Google Cloud Run for deploying our API to production. We also used Google Cloud Monitoring to monitor the health and performance of our API in real-time.

In addition to designing and implementing our API, we also prioritized documentation and support. We created API documentation using OpenAPI (formerly known as Swagger) and provided support to our mobile application development team as needed.

# How to use Kubernetes in your project

Here are the steps to use Kubernetes in your project:

* Familiarize yourself with Kubernetes: Before using Kubernetes, it's important to have a good understanding of what it is and how it works. You can start by reading the Kubernetes documentation and learning about its key concepts, such as pods, services, and deployment objects.
* Choose a Kubernetes environment: There are several options for running a Kubernetes cluster, including using a cloud provider like Azure Kubernetes Service (AKS) or Google Kubernetes Engine (GKE), or running your own cluster on-premises. Choose the environment that best meets your needs.
* Create a Kubernetes deployment: A deployment in Kubernetes is used to manage and deploy a set of replicas of your application. To create a deployment, you'll need to create a deployment object in YAML format and specify the number of replicas, the container image to use, and any other required information.
* Deploy your application: Once you have created a deployment object, you can deploy your application to your Kubernetes cluster by using the kubectl apply command. This will create the necessary pods, services, and other objects to run your application.
* Monitor and manage your application: To ensure that your application is running smoothly, you'll need to monitor and manage it regularly. Kubernetes provides various tools and services to help you with this, such as the Kubernetes Dashboard, kubectl, and Prometheus.
* Scale your application: If your application is growing in popularity, you may need to scale it to handle more traffic. You can do this in Kubernetes by increasing the number of replicas in your deployment object and reapplying it to your cluster.

By following these steps, you can use Kubernetes in your project and take advantage of its powerful container orchestration capabilities to manage and deploy your applications at scale.

# How to use Jest in React

Here are the steps to use Jest in a React project:

* Install Jest: To use Jest, you'll need to install it in your React project. You can do this by running the following command in your project's directory: npm install --save-dev jest.
* Configure Jest: Jest requires a configuration file to run. You can create a jest.config.js file in your project's root directory and specify your configuration options. For example, you can specify the test environment, the test file extension, and any other required options.
* Write your tests: With Jest installed and configured, you can start writing tests for your React components. Jest provides a simple and easy-to-use testing framework, so you can write tests in JavaScript using familiar syntax.
* Run your tests: Once you have written your tests, you can run them using the jest command. This will execute all of your tests and display the results in the console.
* Debug your tests: If your tests are failing, you can debug them using Jest's built-in debugger. You can set breakpoints in your tests and step through them to see why they are failing.
* Automate your tests: To ensure that your tests are run every time you make changes to your code, you can automate your tests using a continuous integration (CI) system like Jenkins or Travis CI. This will run your tests automatically every time you push changes to your code repository.

By following these steps, you can use Jest in your React project to write and run tests for your components, and ensure that your application is functioning as expected.

# How to use Karma in your project

Here are the steps to use Karma in your project:

* Install Karma: To use Karma, you'll need to install it in your project. You can do this by running the following command in your project's directory: npm install --save-dev karma.
* Install Karma plugins: Karma requires various plugins to run, depending on your project's requirements. For example, if you're using Angular, you'll need to install the karma-jasmine and karma-chrome-launcher plugins. You can install these plugins by running the following command: npm install --save-dev karma-jasmine karma-chrome-launcher.
* Configure Karma: Once you have installed Karma and its plugins, you'll need to configure Karma to run your tests. You can do this by creating a karma.conf.js file in your project's root directory and specifying your configuration options. For example, you can specify the test environment, the test file extension, and any other required options.
* Write your tests: With Karma installed and configured, you can start writing tests for your project. Karma provides a simple and easy-to-use testing framework, so you can write tests in JavaScript using familiar syntax.
* Run your tests: Once you have written your tests, you can run them using the karma start command. This will execute all of your tests and display the results in the console.
* Debug your tests: If your tests are failing, you can debug them using Karma's built-in debugger. You can set breakpoints in your tests and step through them to see why they are failing.
* Automate your tests: To ensure that your tests are run every time you make changes to your code, you can automate your tests using a continuous integration (CI) system like Jenkins or Travis CI. This will run your tests automatically every time you push changes to your code repository.

By following these steps, you can use Karma in your project to write and run tests for your application, and ensure that it is functioning as expected.

# PHP interview questions

Here are some common PHP interview questions:

**What is PHP and what is it used for?**

PHP stands for Hypertext Preprocessor and it is a server-side scripting language designed for web development. It is used for creating dynamic websites, building e-commerce platforms, and processing form data.

**What is the difference between PHP and other server-side scripting languages like ASP and JSP?**

PHP is similar to other server-side scripting languages such as ASP and JSP, but it is an open-source language that can run on multiple operating systems and has a large community of users and developers. PHP is often considered more flexible and easier to learn compared to other server-side scripting languages.

**What are the key features of PHP?**

Some of the key features of PHP include: support for various databases, easy integration with HTML, ability to create dynamic content, and easy management of cookies and sessions.

**How does PHP handle errors and exceptions?**

PHP handles errors and exceptions through the use of the error reporting functions and exceptions. Error reporting functions can be used to track and log errors in the code, while exceptions provide a structured way to handle and recover from errors.

**Can you explain the difference between include, require, and require\_once?**

The difference between include, require, and require\_once is that include allows a file to be included multiple times in a script, while require only allows it to be included once. require\_once is similar to require but also ensures that the file is not included more than once.

**What is the difference between session and cookie in PHP?**

A session in PHP is a way to store data on the server that can be accessed by multiple pages during a user's interaction with the website. A cookie is a small file stored on the client's computer that can be used to store information, such as user preferences. The main difference is that a session is stored on the server, while a cookie is stored on the client's computer.

**What are some common security risks in PHP and how can they be prevented?**

Some common security risks in PHP include SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). These risks can be prevented by using parameterized queries, validating user input, and using encryption and authentication methods.

**Can you explain the difference between GET and POST methods in PHP?**

The GET method in PHP is used to retrieve data from the server and is typically used for simple requests, such as retrieving a page or searching for information. The POST method is used to submit data to the server, such as when submitting a form, and is more secure as it does not reveal data in the URL.

**What is a PHP array and how can you sort it?**

A PHP array is a data structure used to store a collection of values or data. Arrays can be sorted using the sort() function, which sorts elements in ascending order, or the rsort() function, which sorts elements in descending order.

**Can you explain the use of regular expressions in PHP?**

Regular expressions in PHP are a powerful tool for pattern matching and can be used for tasks such as validating data, searching for patterns, and replacing text.

**How can you connect to a database in PHP?**

A database can be connected to PHP using the mysqli or PDO extension. Both extensions provide a simple and efficient way to interact with a database and perform tasks such as inserting, updating, and retrieving data.

**Can you explain the use of namespaces in PHP?**

Namespaces in PHP are used to group related classes, functions, and constants under a common name. This helps to avoid naming conflicts and makes it easier to organize code.

**What is the difference between a class and an object in PHP?**

A class in PHP is a blueprint or template used to create objects that have similar properties and behaviors. An object is an instance of a class, and it can have its own data and behavior.

**Can you explain the concept of inheritance in PHP?**

Inheritance in PHP is a mechanism that allows a class to inherit properties and methods from a parent class. This allows for the creation of new classes that are based on existing ones, making it easier to reuse and extend existing code.

**What is a trait in PHP and when is it useful?**

A trait in PHP is a way to reuse code between classes, similar to inheritance. It is useful for situations where multiple classes need to share the same behavior or functionality.

# Laravel interview questions

**What is Laravel and what problem does it solve?**

Laravel is an open-source PHP framework used for web application development. It was created to solve common problems that developers face while building web applications, such as routing, security, and database management. By providing a robust set of tools and features, Laravel makes it easier for developers to build high-quality, scalable web applications.

**What are the main features of Laravel?**

The main features of Laravel include routing, middleware, controllers, blade templating engine, migrations, security, and events and listeners. Laravel also has built-in support for common web application functionality, such as form submissions and validation, caching, and authentication and authorization.

**Can you explain the Model-View-Controller (MVC) pattern and how it is used in Laravel?**

The Model-View-Controller (MVC) pattern is a software design pattern that separates the application into three main components: the model, which represents the data and business logic of the application; the view, which displays the data to the user; and the controller, which handles user input and communicates with the model and view. In Laravel, the MVC pattern is used to structure the code and provide a clear separation of concerns.

**How does Laravel handle routing and URL generation?**

Laravel handles routing and URL generation through its routing system. Developers can define routes for different URLs and map them to specific controller actions, making it easy to handle incoming HTTP requests and generate URLs for links and redirects.

**Can you explain the use of migrations in Laravel?**

Migrations in Laravel are used to manage changes to the database schema. They allow developers to version control the schema and make changes in a controlled, reproducible manner. This makes it easier to manage database changes as the application evolves.

**How does Laravel handle database relationships, such as one-to-one, one-to-many, and many-to-many?**

Laravel provides several ways to handle database relationships, such as one-to-one, one-to-many, and many-to-many. Laravel's Eloquent ORM (Object-Relational Mapping) makes it easy to define and manipulate these relationships in a convenient, intuitive way.

**Can you explain the use of controllers in Laravel and their role in handling HTTP requests?**

Controllers in Laravel are used to handle HTTP requests and perform actions based on user input. They receive incoming requests, process data, and return a response, either by rendering a view or returning data in a specific format.

**How does Laravel handle security and preventing common attacks, such as SQL injection and cross-site scripting (XSS)?**

Laravel provides several security features to prevent common attacks, such as SQL injection and cross-site scripting (XSS). For example, Laravel's query builder and Eloquent ORM automatically escape data to prevent SQL injection, and Laravel's blade templating engine automatically escapes output to prevent XSS. Additionally, Laravel provides features for password hashing and encryption, and it implements security best practices out of the box.

**Can you explain the use of Blade templating engine in Laravel and its advantages?**

Blade is Laravel's built-in templating engine. It provides a simple syntax for creating templates and generating HTML, making it easy to separate the presentation logic from the rest of the application. Blade also supports layouts and partials, making it easy to reuse code across multiple pages.

**How does Laravel handle form submissions and validation?**

Laravel provides a convenient and flexible way to handle form submissions and validation. The framework provides built-in validation rules and error handling, making it easy to validate user input and display error messages to the user.

**Can you explain the use of middleware in Laravel and its role in the request-response cycle?**

Middleware in Laravel are used to perform actions before or after a request is processed. They can be used to perform authentication, authorization, or other tasks that need to be performed before the request is handled by the controller.

**How does Laravel handle caching and performance optimization?**

Laravel provides several caching mechanisms, such as in-memory caching, file caching, and database caching, to help improve the performance of the application. Laravel also provides a convenient cache API that makes it easy to cache data and retrieve it later.

**Can you explain the use of events and listeners in Laravel and when they are useful?**

In Laravel, events are used to trigger actions or perform specific tasks in response to certain activities taking place in the application. For example, when a user registers, you might want to send them a welcome email. This can be achieved by triggering an event that sends the email when the user is registered. Listeners, on the other hand, are used to listen for events and perform specific actions in response to those events.

Events and listeners are useful for decoupling different parts of the application, making it easier to maintain and update.

**How does Laravel handle dependencies and packages, such as through Composer?**

Laravel uses the Composer package manager to handle its dependencies and packages. Composer is a tool for dependency management in PHP. It allows developers to declare the libraries they need for their projects and manage them in a single place. In Laravel, you can add dependencies to your project by specifying them in the composer.json file, and then running the composer install command. This will download and install the required packages, making them available for use in your project.

**Can you explain how you would implement authentication and authorization in a Laravel application?**

Laravel provides built-in authentication and authorization features that make it easy to implement secure authentication and authorization in your application. To implement authentication, you can use the authentication controllers provided by Laravel and define the routes for login and logout. Laravel also provides guards and providers for authentication, which determine how users are authenticated and where their credentials are stored. For authorization, you can use gates and policies, which define the rules for who can perform certain actions in the application. Laravel also provides middleware for controlling access to specific routes, which can be used to enforce authorization rules.

**What is cross-site scripting**

Cross-Site Scripting (XSS) is a type of security vulnerability that occurs when an attacker injects malicious code into a web page that is viewed by other users. This malicious code, usually in the form of a script, is executed in the browser of the user viewing the page, and can steal sensitive information, such as user credentials or personal data. There are two main types of XSS: stored XSS and reflected XSS.

Stored XSS occurs when the malicious code is permanently stored on the server and is served to every user who views the affected page. Reflected XSS occurs when the malicious code is included in a request and is reflected back in the response, allowing the attacker to inject the malicious code into the page viewed by the user.

To prevent XSS, it is important to validate and sanitize user input before displaying it on a web page. This can be achieved by using functions provided by the programming language, such as PHP's htmlspecialchars(), or by using a framework that has built-in security features, such as Laravel's Blade templating engine, which automatically escapes user input.

**CI/CD pipeline experience (they use GitHub)**

I have extensive experience in CI/CD pipelines and have been using GitHub for source control and automation for the past 3 years. I have a strong understanding of how to automate the deployment process and have implemented several techniques to reduce downtime during deployments. I have also worked closely with cross-functional teams, such as sales, marketing, and data engineering, to ensure that all projects are delivered on time and meet the needs of all stakeholders.

I have a solid track record of leading technical projects and teams. I have led the development of several new features and have mentored junior developers, helping them to grow their skills and take on more challenging projects. I am proficient in agile methodologies and I prioritize tasks effectively, ensuring that all team members are aligned on project goals and timelines.

One thing that sets me apart from other candidates is my expertise in cloud technologies. I have a deep understanding of cloud infrastructure, including AWS and Google Cloud, and have experience designing and implementing scalable, highly available systems. I am passionate about using cloud technologies to solve real-world problems and I enjoy sharing my knowledge and experience with others.

Overall, I believe that my experience in CI/CD pipelines, my ability to collaborate with cross-functional teams, my leadership skills, and my expertise in cloud technologies make me an excellent candidate for the role.

# Write the professional selenium experience with example answer

I have been using Selenium for the past 5 years to automate functional and regression testing for web applications. I have experience working with Selenium WebDriver, Selenium Grid, and various programming languages including Java, Python, and C#. I have also worked with various test frameworks, including JUnit and TestNG, to create and maintain test suites that cover a wide range of functionality.

One of my key strengths with Selenium is my ability to design and implement efficient and scalable test frameworks. For example, on a recent project, I designed a data-driven testing framework that enabled us to test a complex system with over 10,000 test cases. This framework was highly automated and allowed us to reduce manual testing effort by over 80%.

In addition to my technical skills, I also have experience leading testing efforts and working with cross-functional teams. I have a strong understanding of Agile methodologies and have worked closely with development, QA, and project management teams to ensure that all testing requirements are met. I have also been responsible for mentoring junior testers and have helped them develop their skills and knowledge of Selenium.

Overall, I believe that my experience with Selenium, my ability to design and implement efficient testing frameworks, and my experience leading testing efforts make me an excellent candidate for any role that requires expertise in automated testing.

# Write the professional experience in aws ec2 and lamda, s3

As a professional web developer, I have experience working with AWS EC2 and Lambda, as well as S3. In one of my previous projects, I had the opportunity to work on a serverless architecture that made use of AWS services.

The project was a back-end platform for an e-commerce website that needed to handle a high volume of requests and data processing. To meet these requirements, I chose to use AWS EC2 for the application servers and AWS Lambda for the background job processing. This allowed us to quickly and easily scale the infrastructure as needed, without having to worry about managing individual servers.

We also made use of Amazon S3 for storing and retrieving large amounts of data, such as product images and user uploads. This provided a highly scalable and reliable storage solution that allowed us to serve the data to our users quickly and efficiently.

In terms of deployment and management, I used AWS CloudFormation and AWS CodeDeploy to automate the deployment process and ensure that the platform was always up-to-date. This allowed us to minimize downtime and ensure that the platform was always running smoothly.

Overall, working with AWS EC2, Lambda, and S3 provided a highly scalable and flexible infrastructure that allowed us to meet the demanding requirements of the e-commerce platform. The experience taught me the importance of choosing the right tools and services to meet the specific needs of each project, and how to effectively manage and deploy cloud-based infrastructure.

# Write the microservices experience with example answers

Microservices is an architectural style for building and deploying applications as a collection of small, independent services that communicate with each other over a network. This approach offers several benefits, including increased scalability, better fault tolerance, and improved deployment and testing efficiency.

"I have extensive experience building and deploying microservices-based applications. In my previous role as a software engineer, I worked on a large e-commerce platform that was built using microservices. The platform consisted of several small, independent services that each focused on a specific aspect of the application, such as product catalog management, order management, and customer management. This allowed us to quickly and easily make changes to individual services without affecting the entire system, and it also allowed us to scale individual services as needed to meet changing demands. I also worked closely with the DevOps team to automate the deployment process, which improved the speed and reliability of our deployments."

# Disadvantage of Auth in laravel

Performance: Laravel's built-in authentication features can be slow and resource-intensive, particularly for large-scale applications with many users. This can negatively impact the overall performance of the application.

Complexity: Laravel's authentication features can be complex to set up and maintain, particularly for developers with limited experience in web application security.

Security: Laravel's authentication features may not always be fully secure, as they can be vulnerable to common attacks such as SQL injection and cross-site scripting (XSS). Developers must take extra precautions to ensure the security of their applications, such as implementing secure password storage and using encryption for sensitive data.

Customization: Laravel's authentication features may not always be customizable to meet the specific needs of an application, which can limit the ability of developers to tailor the authentication process to their needs.

Integration: Laravel's authentication features may not always be easily integrated with other systems and technologies, such as third-party authentication providers or single sign-on (SSO) solutions.

In conclusion, while Laravel's authentication features can be convenient and provide a good starting point for many applications, developers must be aware of the potential disadvantages and take the necessary steps to ensure the security and performance of their applications.

# Authorization module in Laravel

Laravel provides a flexible and modular authorization module for controlling access to resources within an application. This module provides a simple and intuitive way to manage user permissions and ensure that users have the appropriate access to resources based on their role and responsibilities.

The authorization module in Laravel uses middleware to control access to routes and actions within the application. Developers can define middleware that checks for specific permissions and authorizes or denies access based on the user's role.

Laravel also provides a simple and intuitive syntax for defining authorization rules and managing user roles. For example, you can use the can method to check if a user has a specific permission, or the role method to check if a user is part of a specific role.

Additionally, Laravel provides several tools for managing user permissions, such as the ability to assign permissions to roles, grant permissions to specific users, or revoke permissions as needed. This makes it easy to manage user access to resources and ensure that only authorized users have access to sensitive data.

In conclusion, the authorization module in Laravel provides a comprehensive and flexible solution for controlling access to resources within an application, making it easy to manage user roles and permissions, and ensure the security and integrity of sensitive data.

# What other authorization technology in laravel

In addition to Laravel's built-in authorization module, there are several other authorization technologies that can be used in Laravel, including:

Gates and Policies: Gates and Policies provide a simple and flexible way to define authorization rules for specific actions within an application. Gates are used to define rules for granting access to certain parts of the application, while policies are used to define rules for specific models.

Casbin: Casbin is an authorization library for Laravel that is based on the concept of access control policies. It provides a powerful and flexible way to define and manage authorization rules in Laravel.

Sentry: Sentry is a user management and authorization package for Laravel that provides a simple and flexible way to manage users, permissions, and roles.

JWT: JSON Web Tokens (JWT) are a popular way to handle authentication and authorization in Laravel, as they provide a secure and scalable way to transmit information between the client and server.

OAuth: OAuth is an open standard for authorization that provides a secure way for users to grant access to their data to third-party applications. Laravel provides support for OAuth 2.0, making it easy to implement secure authorization in Laravel applications.

Overall, there are many options for authorization in Laravel, and developers should choose the one that best fits their needs and requirements. The built-in authorization module provides a solid foundation, but other options may be more suitable for specific requirements.

# How many people in your team and what is your preference

"I have previously worked in teams ranging from 5 to 20 individuals, and I have found that I work well in both small and large team environments. My primary focus is always on delivering high-quality work and collaborating effectively with my team members. I have experience with agile methodologies and enjoy working in fast-paced, dynamic environments where I can continuously learn and grow.

In terms of team preferences, I have a strong preference for teams that are collaborative, communicative, and value open dialogue. I believe that open communication and continuous feedback are key to delivering successful projects and fostering a positive work environment. Additionally, I prefer teams that prioritize work-life balance and encourage a healthy work environment."

# What is your ideal environment

As an AI language model created by OpenAI, I don't have personal preferences or physical form, I don't experience environments. I am designed to function in any environment and to assist users in providing information and answering questions. My goal is to provide accurate and helpful responses to users, regardless of the environment or platform I am running on.

"My ideal work environment is one that values innovation, collaboration, and continuous learning. I thrive in fast-paced, dynamic environments where I can work on challenging projects and be a part of a team that is constantly pushing the boundaries of what is possible.

In terms of physical environment, I prefer an open and well-lit workspace that fosters collaboration and creativity. I believe that the physical workspace plays an important role in promoting a positive and productive work environment.

Additionally, I value a strong focus on work-life balance and support for remote work options. I believe that a flexible work schedule and the ability to work from home when needed is essential for a healthy work-life balance.

Overall, my ideal work environment is one that prioritizes the well-being of its employees and provides opportunities for growth and professional development."

**Node.js, TypeScript, JavaScript, React experience**

I have extensive experience working with Node.js, TypeScript, JavaScript, and React. I have developed and maintained a variety of web applications using these technologies and have a deep understanding of their capabilities and limitations.

**CI/CD pipeline experience (they use GitHub)**

Regarding CI/CD pipeline experience, I have extensive experience working with GitHub for source control and I have also implemented CI/CD pipelines for several projects. I understand the importance of automating the build, test, and deployment processes and have worked to ensure that my projects have a reliable and repeatable pipeline.

**Experience collaborating with other teams such as sales, marketing, data engineering, etc**

I enjoy collaborating with cross-functional teams and have experience working with sales, marketing, data engineering, and other teams to ensure successful project delivery. I believe that communication and collaboration are key to the success of any project, and I strive to foster a positive and productive working relationship with all stakeholders.

**Leadership experience**

In terms of leadership experience, I have led small teams of developers on several projects and have a track record of delivering high-quality software within tight timelines. I am comfortable with taking charge of a project and ensuring that the team is working effectively and efficiently.

**Something that sets you apart from other candidates**

Finally, what sets me apart from other candidates is my passion for continuously learning and staying up-to-date with the latest technologies. I have a strong desire to always be pushing the boundaries of what is possible and to use my skills and knowledge to deliver innovative solutions for my clients. Additionally, I have experience working on projects for clients in various industries, which has given me a unique perspective and ability to approach problems from different angles.

**How to use Kubernetes for CI / CD**

Kubernetes is a powerful tool for automating deployment, scaling, and management of containerized applications. When combined with a CI/CD pipeline, it can provide an efficient and scalable way to deploy applications to production.

Here are the steps to implement Kubernetes for CI/CD:

Containerize the application: The first step is to containerize the application and make sure it runs in a container environment. This can be achieved using Docker.

Create a Docker image: Once the application is containerized, you need to create a Docker image of the application. The image will contain all the necessary dependencies, configuration files, and scripts required to run the application.

Push the Docker image to a registry: Next, you need to push the Docker image to a container registry such as Docker Hub or Google Container Registry. This will make the image accessible to the Kubernetes cluster.

Create a Kubernetes deployment: A Kubernetes deployment is a blueprint for how the application should run in the cluster. It specifies the number of replicas, resources, and update strategies for the application.

Set up a CI/CD pipeline: You can set up a CI/CD pipeline using tools such as Jenkins or GitHub Actions. The pipeline will automate the process of building, testing, and deploying the application to the Kubernetes cluster.

Automate deployment with Kubernetes: The CI/CD pipeline will trigger a deployment to the Kubernetes cluster whenever changes are pushed to the source code repository. Kubernetes will then use the deployment to update the application running in the cluster.

In summary, Kubernetes can be used for CI/CD by containerizing the application, creating a Docker image, pushing the image to a registry, creating a deployment, setting up a CI/CD pipeline, and automating deployment with Kubernetes.

**Typescript vs javascript**

TypeScript is a statically typed superset of JavaScript that adds features such as classes, interfaces, and optional type annotations. TypeScript is designed to make it easier for developers to write large scale and maintainable applications. On the other hand, JavaScript is a dynamically typed language that does not include features such as classes and interfaces and relies on implicit type coercion.

One of the main advantages of using TypeScript over JavaScript is the increased development speed and fewer runtime errors due to its static type checking. This means that TypeScript can catch type-related errors at compile time, rather than at runtime, which saves time and effort in debugging. Additionally, TypeScript's static type information also makes it easier for editors and IDEs to provide better code completion and navigation, making the development experience more efficient.

Another advantage of TypeScript is that it can be used with popular front-end libraries such as React, and it can be compiled to JavaScript code that can be run on any platform that supports JavaScript. This allows developers to leverage the strengths of TypeScript while still being able to use their existing JavaScript code.

In conclusion, TypeScript provides several benefits over JavaScript, such as improved development speed, fewer runtime errors, and better support for large scale applications. However, for smaller projects or for developers who are more comfortable with dynamic typing, JavaScript may be a better choice.

**For PHP Role**

In terms of PHP, I have a strong background working with Laravel, one of the most popular PHP frameworks. I have experience building robust, scalable applications utilizing the MVC architecture and leveraging various Laravel components such as migrations, controllers, and Eloquent ORM for database interaction.

With regards to TypeScript, I have experience working with modern frontend web frameworks such as VueJS. I have experience in building user-friendly, interactive, and responsive web applications using TypeScript, HTML, CSS, and JavaScript. I am proficient in utilizing TypeScript features such as optional typing, interfaces, and classes to write clean and maintainable code.

I have experience working with server-side rendering technologies such as Next.js, which allows me to build fast and performant web applications that provide a seamless user experience. Additionally, I am well-versed in the use of UML diagrams for system design and am comfortable transitioning from monolith to microservice architecture to provide scalable and modular solutions.

I am also experienced in working with micro-front architecture and have hands-on experience in implementing multi-step transactions to ensure the consistency and integrity of data.

In terms of my leadership experience, I have served as a technical lead on several projects and have worked collaboratively with sales, marketing, and data engineering teams to ensure project success. I am a strong communicator and believe in leading by example, encouraging teamwork and fostering a positive work environment.

What sets me apart from other candidates is my passion for continuously learning and keeping up-to-date with the latest technologies and best practices in the software development industry. I am constantly seeking new challenges and opportunities to expand my skillset and bring value to my team and organization.

**For Getting hired to Frontend Position**

As a web developer with over 8 years of experience and expertise in JavaScript frameworks, I am confident that I can bring a unique skill set and valuable contributions to your team. I have a strong background in front-end development, with 8+ years of experience using React and CSS to build complex layouts and meet pixel-perfect design requirements. My experience with back-end development, using Perl or NodeJS and GraphQL APIs, has equipped me with the technical knowledge necessary to bring your projects from concept to completion.

In my previous roles, I have honed my ability to mentor other engineers, work with product teams to define and drive solutions, and build performant, responsive front-end components. I am a self-starter with a proactive attitude and a passion for writing clean, modern, maintainable, and highly performant code. I have a strong understanding of engineering principles and design patterns and am committed to continuously improving my skills.

In terms of collaboration and communication, I have a positive, solution-oriented approach and am experienced in working within Agile teams and using Jira. I am also known for my excellent interpersonal effectiveness, both in one-on-one interactions and in presenting to a room. I have a strong desire for self-awareness and growth, and am always seeking new opportunities to expand my knowledge and capabilities.

I believe that I have the right combination of skills, experience, and drive to make a positive impact at your company. I am eager to put my expertise and passion to work and look forward to the opportunity to contribute to your team's success.

**Write the professional ecommerce experience with example answer**

As a professional with e-commerce experience, I have gained hands-on experience in delivering effective and efficient e-commerce solutions for various clients. My experience has allowed me to gain a deep understanding of the entire e-commerce development process, from conceptualization to launch and post-launch support.

For example, I have worked on a project for an online fashion retailer where I was responsible for developing the front-end user interface using ReactJS, and integrating the solution with the Magento back-end for order management and payment processing. I collaborated closely with the design team to ensure that the user experience was seamless and visually appealing, and I used Agile methodologies to manage the project and deliver it on time and within budget.

In addition to this project, I have also worked on several others, including developing custom e-commerce websites using WooCommerce and Shopify, and integrating third-party payment gateways and shipping APIs. Through my e-commerce experience, I have developed a strong understanding of the unique challenges that come with e-commerce development, such as ensuring data security, optimizing page load times, and providing a seamless user experience on different devices.

I am confident that my e-commerce experience and technical skills make me an ideal candidate for any e-commerce project. I am passionate about delivering high-quality solutions that meet the needs of both businesses and customers, and I am always looking for new ways to improve the user experience and streamline processes.

**When you work on ecommerce project, what was the most difficult thing with example answer?**

When working on an e-commerce project, one of the most difficult things I encountered was ensuring the platform had the ability to handle a high volume of traffic and transactions. For example, I was part of a team that was working on an e-commerce platform for a large retail company. During peak shopping seasons, the website would receive a significant amount of traffic, leading to slow load times and even crashes. To address this issue, we implemented caching techniques and load balancing strategies to distribute the traffic evenly across multiple servers. Additionally, we optimized the database queries and implemented asynchronous processing to improve the overall performance of the platform. These efforts resulted in a more stable platform that was able to handle the increased traffic during peak seasons without any major issues.

**What is the graphql**

GraphQL is a query language and runtime for APIs. It was developed by Facebook in 2012 and released as an open-source project in 2015. GraphQL provides a more efficient, powerful, and flexible alternative to the traditional REST API. Unlike REST, which exposes a fixed set of endpoints for each resource, GraphQL allows the client to request exactly the data it needs, and nothing more. This reduces over-fetching or under-fetching of data, and enables the client to receive data in a predictable structure. GraphQL also provides a type system for the API, which can be used to validate the queries and mutations being sent to the server, and to generate client libraries.

**How can we do query optimization in graphql**

Query optimization in GraphQL is crucial for ensuring a fast and efficient performance for your application. Here are a few ways to optimize queries in GraphQL:

Batch queries: You can batch multiple queries into a single request, reducing the number of round trips between the client and the server.

Use fragments: Fragments allow you to reuse common fields in multiple queries, reducing the amount of duplicated code and improving performance.

Limit fields: Only request the fields you need to reduce the amount of data being returned and improve performance.

Use efficient resolvers: Use resolvers that are optimized for performance, such as data-fetching libraries like DataLoader or Apollo-Client.

Cache results: Implement caching to store the results of frequently used queries, reducing the number of requests to the server.

Monitor performance: Regularly monitor your GraphQL performance and analyze query execution time, data transfer size, and other metrics to identify areas for improvement.

By following these best practices, you can optimize the performance of your GraphQL queries, ensuring a smooth and efficient user experience for your application.

**Explain useeffect hook**

The useEffect hook is a React Hook that allows you to run side effects (functions) in your functional components. The useEffect hook is used to manage data and re-rendering in React components. It is a powerful tool that helps you manage state and side effects in a way that is both efficient and simple.

The useEffect hook is called after each render cycle and can be used to perform side effects such as data fetching, updating the DOM, or accessing third-party libraries. It takes two arguments: a callback function that contains the side effect logic and a dependency array. The dependency array is used to tell React when to re-run the effect. If the dependency array is empty, the effect will only run once, on mount. If the dependency array includes values, the effect will re-run whenever those values change.

Here's an example of how to use the useEffect hook to fetch data from an API:

import React, { useState, useEffect } from 'react';

const ExampleComponent = () => {

const [data, setData] = useState(null);

useEffect(() => {

fetch('https://api.example.com/data')

.then(response => response.json())

.then(data => setData(data));

}, []);

return (

<div>

{data ? <p>{data.message}</p> : <p>Loading...</p>}

</div>

);

};

In this example, the useEffect hook is used to fetch data from the API after the component has mounted. The effect will only run once, because the dependency array is empty. The setData function is used to update the state with the data that was fetched from the API.

**How did use TDD in your previous project with example answer**

In my previous project, I used Test-Driven Development (TDD) as the method for development. TDD is a software development approach in which tests are written for a feature before the actual implementation of that feature.

For example, in the project, I was tasked with building a user authentication system. I started by writing a test to check whether a user could successfully log in to the system using their email and password. I then ran the test and watched it fail, as the authentication logic was not yet implemented.

Next, I wrote the minimal code required to make the test pass, in this case, it was just enough to accept the input and return a success response. I then wrote another test to check whether the user session is maintained after logging in and repeating the process of writing the minimum code required to make the test pass.

I continued this process of writing tests, running them, and making the code pass until the authentication system was fully functional. Using TDD helped me to write clean, well-structured code and made it easier to identify and fix bugs during the development process. It also ensured that the final product was thoroughly tested and met all requirements.

**Write the flattery sentences about the team at CLX Media**

I have done some research on CLX Media and I am thoroughly impressed by the team's dedication and commitment to excellence. The diverse skill sets and expertise of the team members are truly remarkable and demonstrate a deep understanding of the industry. I am particularly impressed by the innovative solutions and cutting-edge technologies that the team has successfully implemented. I believe that working with such a talented and driven team would be an incredible opportunity for my own professional growth and I am eager to contribute my own skills and experiences to help drive the company's success.

# For Hiring Dev

As you know, US company only require US citizen.

So you have to pretend to be USC.

For this You need to attend the job interview under my name and You have to use my email, birthday, address ...

Please reference this doc

https://docs.google.com/document/d/1TKiyj-qo\_4a\_GXgu2DSJtK4SlW5\_WD0EFipWqQ\_M5oI/edit#heading=h.gwsr4onzus80

When I get video meeting link, I'll send that

While you have an introduction call, you have to ask about drug test, fingerprint

If they require them, you can say "I'm not interested" and you can only work on fully remote position

After you pass all stages of job interview, you will get hired

And then you can handle the project by yourself

I will prepare all documents related to getting hired such as Contract sign, I-9 form

I can pay you $2k - $3k per month according to your skills through cypto currency, Payoneer, Paypal

How do you think?