Screening Test/Interview Questions for Software Engineer (Remote)

Position. Instructions: Please precede all your answers with the question you

are answering. Use acronyms only after you've explained them.

Use correct spelling and grammar.

Email back your answers using either doc or pdf format.

Candidate's Name: Manish Yegurla

1. Are you currently unemployed?

A) Yes, my contract is completed last week with wells Fargo.

2. Can you explain the difference between procedural programming and object-oriented programming?

A) Procedural/Functional programming depends on functions to manage data. It depends on global state to hold all or some of the application data and supply it to other parts of the application which mostly takes a top-down approach, which can be hard to maintain as the application grows. object oriented programming (OOP) on the other hand will follow class-

based approach having objects to hold data. OOP has better data integrity and code reuse through concepts like inheritance, polymorphism, and encapsulation, making it more suitable for larger and more complex applications.

3. What is the importance of version control systems like Git in software development?

A) Version control systems like Git in software development are important because they allow multiple developers to work on the same project simultaneously without overwriting each other's changes. It will ease developers work to merge new code to the main branch by checking for conflicts before the merge. It can keep track of all the changes ever made to the branch which allows developers to revert the code if something fails. Each developer can create their own feature branch out of the main branch and work on them first before making changes directly to the main branch. Finally, Git is crucial for managing code efficiently, maintaining project integrity.

4. Describe the difference between front-end and back-end development. Which do you prefer and why?

A) Front-end development focuses on the client side of web applications, where in users can interact with directly. It is built using technologies such as HTML, CSS, JavaScript and frameworks like React. For designing of the front-end components, we can use libraries like bootstrap.

Back-end development focuses on the server side, managing the server and database to supply data to and from the client side of the application. It is built using languages and frameworks such as Node.js, Java, Python and databases like MySQL, MongoDB. Developer here writes code to handle API calls from the front-end and respond back with the information accessed from the database.

I prefer front-end development because I enjoy creating interactive and visually appealing user interfaces that enhances user experience. Working with technologies like React and Next.js feels even better as their functional approach is easy to understand and see immediate results of my work, which is very satisfying. Additionally, front-end development involves more frequent interactions with end-users and designers, which I find rewarding.

5. How do you ensure the security of a web application you're developing?

A) It involves some steps to be taken like validating the user inputs to prevent common vulnerabilities like SQL injection and cross-site scripting (XSS). Using multi-faction authentication to ensure the access to right user, this can be don't my using third part libraries like OAuth. Using HTTPS to encrypt data transmitted between the client and server. Implement proper error handling to avoid exposing sensitive information through error messages. Also, while making an API call ensure to have correct headers by configuring with Content Security Policy (CSP), Strict-Transport-Security (HSTS) to protect against common attacks like XSS, and man-in-the-middle attacks. By following these few measures, we can enhance the security of the application.

6. Can you explain the concept of Big O notation and why it's important in algorithm analysis?

A) It is used to calculate the algorithms runtime and space complexities for the given input data. It is important to measure the algorithms efficiency when we have large data. For an example we have an array of data to be sorted for which we have multiple sorting algorithms like bubble sort, quick sort and merge sort. We use merge sort because it is more efficient. O (1) means constant time where an algorithms runtime does not change with input size. O(n) means linear size, runtime of algorithm grows with input size and O(n^2) means quadratic time, algorithms runtime will grow quadratically with the input size.

7. What are some common data structures, and when would you use each one?

A) Arrays, Linked Lists, Stacks and Queues are some of the data structures.

Arrays is a collection of elements by index which can be latter accessed quickly.

Linked Lists are sequence of nodes where each node contains data and a reference to the next node. It can be used for dynamic data sets where we can insert or delete the data by traversing through the list.

Stack is a collection of elements where it follows last in first out structure (LIFO), where new elements are added at the top using push method and delete the first or recent element using pop method.

Queue is a collection of elements where it follows first in first out structure (FIFO), where new elements are added at the end using enqueue method and delete the first or recent element using dequeue method.

8. How do you approach debugging and troubleshooting when you encounter a problem in your code?

A) When I encounter a problem in my code, I start by isolating the issue, often by adding print statements or using a debugger to understand where the problem occurs. I check for common syntax errors, incorrect logic, or data type mismatches. I also use chrome debugger tools to place break points to check the passed data for a front-end application.

9. Can you discuss the advantages and disadvantages of using a relational database versus a NoSQL database?

A) Relational databases (RDBMS) deals with data that are organized in the tables and tables are connected to one another by a common key such as ID. Now the ID will be used to grab the data from all the tables using that ID, making them ideal for applications requiring complex queries and transactions, like financial systems.

NoSQL on the other hand has data that is unstructured or semi-structured. Here with the ID, we can grab all the data that is linked to it, we wont be using tables, it could be JSON

structure making them suitable for large-scale applications like social networks and real-time analytics.

10. Explain the concept of RESTful APIs and why they are commonly used in web development.

A) RESTful APIs are also called as REST API's, they use standard HTTP methods like GET, POST, PUT, and DELETE to perform operations on resources identified by URLs. These are stateless, i.e., each request which is sent from client to server should have all the information it requires for the server to process it, some of it can be in header of the API. RESTful APIs are commonly used in web development due to their simplicity, scalability, and the ease with which they can be integrated with various clients like web or mobile.

11. What is continuous integration and continuous deployment (CI/CD), and how does it benefit software development projects?

A) Continuous Integration (CI) is the practice of frequently integrating code changes into a shared repository, where automated builds and tests run to detect issues early. Continuous Deployment (CD) extends this by automatically deploying every code change that passes the tests to production. CI/CD benefits software development projects by enabling faster and more reliable delivery of software, reducing the risk of integration issues, and ensuring that code changes are tested and deployed consistently. This leads to quicker feedback, improved collaboration, and higher-quality software with fewer bugs and shorter release cycles.

12. How do you approach debugging issues in both frontend and backend code?

A) For front-end I mostly use browser developer tools to inspect elements, review console logs, and set breakpoints to step through the code. At the breakpoints we can see what data we are getting for the variables declared inn the application. I use error boundaries to catch network errors. For backend I use Qmetry tool which will hold all the loggings we placed through testing, and I utilize debugging tools available in the development environment to set breakpoints and examine variable states. In both the field, I divide the code into smaller parts where I can catch any errors and easily resolve it. I also look up online to find resources related to the bug I am dealing with.

13. What is your familiarity with cloud platforms like AWS, Azure, or Google Cloud?

A) I have familiarity with cloud platforms like AWS, Azure, and Google Cloud, having used them to deploy and manage web applications. My experience includes working with AWS services such as EC2, S3, and Lambda for scalable computing, storage, and serverless functions. I have utilized Azure for its integrated development environment and comprehensive DevOps solutions. Additionally, I have explored Google Cloud for its robust data processing and machine learning capabilities. These platforms have helped me ensure high availability, scalability, and efficient resource management in my projects.

14. What is your experience with automated testing?

A) For frontend I used Jest to write test cases for unit testing and integration testing. In backend development, I've used frameworks like JUnit for Java to test server-side logic and

APIs. Additionally, I've worked with continuous integration tools like Jenkins and GitHub Actions to automate the testing process, ensuring that tests run automatically with every code commit. This approach has helped me catch bugs early.

15. How do you ensure the scalability of a software system you're designing?

A) I use microservices architecture to break the system into smaller, manageable services that can be linked to one another, this helps me to fix bugs easily. I also use some react functionalities like useMemo, lazy loading to optimize the application. I use load balancing to distribute traffic evenly across servers and use caching mechanisms to reduce load on the database and improve response times.

16. Can you describe the agile software development methodology and its key principles?

A) It is an approach to software delivery that give flexibility, collaboration, and customer satisfaction. It focuses on tasks broken down into parts and assigning them across a team of developers with a specified time which helps in completing the work faster. Some of its key features are customer collaboration, where customer throughout the development process to ensure the product meets their needs and expectations. Continuous Improvement, where the team conducts retrospectives at the end of each sprint (Time to complete all the tasks assigned to a developer) to identify what went well, what didn't, and how to improve.

17. Have you worked with any particular programming languages or frameworks? If so, which ones and what projects did you use them for?

A) I have worked with a variety of programming languages and frameworks on several projects, leveraging my skills in both front-end and back-end development. For front-end development, I have extensively used JavaScript, ReactJS, NextJS, and TypeScript to create scalable web applications and single-page applications (SPAs). For instance, at Wells Fargo, I developed a SPA with ReactJS and Redux, significantly improving page load times and user experience.

On the back end, I have utilized Java EE technologies, NodeJS, and ExpressJS to build robust server-side applications. I also have experience with databases such as MongoDB and MySQL for efficient data management. For example, during my time at Broadridge, I worked on optimizing existing API calls and deprecating unnecessary ones to enhance the efficiency of the application. Additionally, I have deployed solutions on cloud platforms like AWS, utilizing services such as EC2 and S3 for scalable and reliable infrastructure.

18. Describe a challenging problem you faced in a previous project and how you approached solving it.

A) I have been working on a functionality called Quick-Navigation, where a customer previously is allowed to navigate through all the pages in application by filling in all the required field in every page and then was able to navigate to the next page and after completing all the pages with customer data he will then be able to submit it to next level where the decision is taken by higher officials. So the task is to allow user to navigate to any page without restricting him/her to fill out the required information first. I used hyper

links for the page names. The main problem here is some pages are dependent on the information that is passed from the previous pages, that I handled by making data available from the initial render at a global state. Next was the APIs that we were making at every page were attached to clicking next button functionalities, which was taken care by replacing the next button functionality with DOM change with the current page. There were many other small problems that I have encountered and tackled them. But during this process I have learned a valuable lesson when I reached out to my lead on one such problem is the re-rendering of the page in react will happen only when there is a state change inside the page, or the props changed (Global State).

19. How do you stay updated with the latest trends and technologies in the field of software engineering?

A) I stay updated with the latest trends and technologies in software engineering through several strategies. First, I regularly follow industry news, blogs, and forums such as Hacker News, Medium, and Stack Overflow to check on new developments and best practices. I also participate in online courses and certifications on platforms like Coursera and Udemy to continuously expand my skill set. I also work on problem solving on platforms like leetcode.

20. How do you contribute to a positive team culture, especially in a remote or distributed team environment?

A) Contributing to a positive team culture in a remote or distributed environment involves several key practices. First, I prioritize clear and frequent communication, using tools like Slack, Zoom, and email to ensure everyone is informed and aligned. I make an effort to be approachable and responsive, fostering an environment where team members feel comfortable sharing ideas and asking for help. Additionally, I participate in regular virtual meetings and team-building activities to build rapport and a sense of camaraderie. I also recognize and celebrate the achievements of my colleagues, providing positive feedback and encouragement. By being proactive, empathetic, and collaborative, I help create a supportive and productive team atmosphere, regardless of physical location.

21. What are your professional development goals?

A) My professional development goals include advancing my expertise in full-stack development, particularly in mastering emerging front-end and back-end frameworks and technologies. I aim to become proficient in cloud-native development and DevOps practices to enhance my ability to deploy and manage scalable, resilient applications. Additionally, I plan to pursue advanced certifications in cloud platforms like AWS and Azure to deepen my knowledge in cloud infrastructure and services. Another goal is to develop my leadership skills by taking on more project management responsibilities and mentoring junior developers. Ultimately, I strive to stay at the forefront of technological advancements and continuously improve my skills to contribute effectively to innovative software solutions.

22. How could your skills and experience add value to our organization?

A) My skills and experience can add value to your organization in several ways. With over three years of experience in both front-end and back-end development, I bring a comprehensive understanding of building scalable and efficient web applications. My expertise in JavaScript, ReactJS, NextJS, and Java EE technologies allows me to create responsive, user-friendly interfaces and robust server-side functionalities. Additionally, my experience with databases like MongoDB and MySQL ensures effective data management.

I have a proven track record of working in agile environments, which means I can quickly adapt to changing project requirements and collaborate effectively with cross-functional teams. My background in optimizing application performance and implementing CI/CD pipelines will contribute to faster, more reliable software delivery. Furthermore, my commitment to continuous learning ensures that I stay updated with the latest trends and technologies, bringing innovative solutions to your organization. Overall, my technical proficiency, adaptability, and proactive approach to problem-solving will help drive the success of your projects and contribute to achieving your organizational goals.

23. How much will you request per hour if you are hired?

A) Given the details of the position at Pixalate, including the responsibilities, desired attributes, and the competitive salary range, I would request an hourly rate at the higher end of the provided range, specifically $85 per hour. This rate reflects my experience and expertise in software development, my proficiency with the technologies mentioned, and my ability to contribute significantly to your projects and team.

Full Job Description

Department: Software Development.
Reports to: Senior Developer.
Location: Remote.
Schedule: 35 hours weekly.
Salary: $70.33 - $85 per hour.
Job Type: Full-time.
Schedule: Monday to Friday.

Company Description:

At **Pixalate,** our mission is to ignite innovation, inspire transformation, and implement digital solutions for a healthier nation. We believe that passion drives great ideas, collaboration builds better solutions, and valuing integrity creates a culture of positive change.

We have supported critical and high visibility IT projects, built award-winning applications, kick-started an in-house innovation lab, and developed new tools that save you time and money through a project life cycle. We aren't about "business as usual".

Our Engineering team gets to innovate and experiment daily with some of the latest technologies in our industry for a product that is paving the way in our space. We are always looking for opportunities to learn, grow, and have fun with each other. Our team culture encourages individuality, collaboration, and creative problem solving.

The Software Engineer appreciates the complexities of building a product (SAAS) and understands software architecture so that solutions are clever enough to solve the immediate problem, flexible enough to adapt to our changing market, and simple enough to maintain so that it can stand the test of time.

Responsibilities:

- Collaborate with product management to create product roadmap and validate user stories
- Develop technical specifications for new products and services
- Drive prototype development to ensure feasibility of designs or validate use of emerging technologies
- Understand the broader context of the technology vision, and advise management and product about tradeoffs that may impact schedules
- Lead projects and provide technical guidance and mentorship to scrum team members
- Perform design and code reviews, validate that designs are fulfilling user story criteria
- Align solutions with architectural direction and standards across teams
- Tackle complex technical problems; perform root cause analysis; and implement effective solutions.
- Identify areas of technical debt, outline what needs to be done and communicate ROI urgency of addressing.
- Assist in recruiting initiatives.

- Maintaining high-quality technical documentation for code, systems, and processes.

The Desired Attributes:

- 3 years of software development disciplines and deep technical skills
- Mastery in at least one of: Golang, Javascript (React or Angular), Ruby on Rails, C#.NET, Java
- Expertise in a technical area such as: Video Streaming, Security, Machine Learning, Front-end development, Event-Driven Design, Microservices
- Strong understanding of software design patterns, algorithms, and data structures
- Experience with microservices, containerization, cloud platforms (AWS preferred)
- Experience in profiling and optimizing the performance of complex systems
- Experience in optimizing tables and queries for Postgres
- Knowledge of security best practices, encryption techniques and compliance standards
- Strong follower of agile/scrum methodologies

Benefits:

- Competitive Salary • Exceptional Benefits:
- 401(k) plan with company matching & immediate vesting schedule (100% of the 1st 3% and 50% of the next 2%)
- BCBS Health, Dental, & Vision Insurance with generous employer funding for employees and dependents
- Generous flexible time off approach
- Professional development
- Remote and hybrid first organization
- Great working environment with a team of exceptional people

At **Pixalate**, Our core values of integrity, collaboration, and client-centricity are at the heart of everything we do. We believe in building long-term partnerships based on trust, transparency, and mutual success. Our dedicated team works closely with our clients to gain a deep understanding of their business needs and challenges, enabling us to deliver digital health solutions that truly make a difference.