1. **Customer Obsession**

Leaders start with the customer and work backwards. They work vigorously to earn and keep customer trust. Although leaders pay attention to competitors, they obsess over customers.

* Tell me about a time you solved a pain point for customers.

(S)When I joined my team, there was a background job which used Alteryx as a tool for ETL. And when there was any bug related to it, it was troublesome for the team to debug the issue because no one has much expertise on Alteryx and adding new functionality was more painful task. Plus, they were paying a lot for license of that software near $40k for a year. I also got some bug to fix. Meantime I notice that i3

t can be converted to dotnet application which will do all ETL process. I propose this to team manager and he agreed to it. (T)As a POC he asked me to convert one of the ETL process that job used to handle into dotnet core application. (A) I went through the whole process of that job functionality. I listed all the functionalities. I structured the code in such a way that other ETL process of the job could be easily integrated, it worked well and later I converted remaining ETL workflow. This was helpful as any team member is comfortable working to fix any bug or add new feature related to this ETL process as app is totally converted to dotnet core. on top of that it helped to save thousands of dollars that was being paid for license.

* Tell me about a time when you had to deal with a very difficult customer. *(We all deal with difficult customers from time to time. Tell me about a challenging client-facing situation and how you handled it. Describe a situation where you negotiated a win-win*.)

S: This about the free-lance project I handled; I took a project to develop software for a restaurant back in Nepal. I made a team of seven but we all will be working part time as we all had our full-time job. After we started working, they started adding new requirements, not only that they started comparing the contract price that we had with them with other software company. Their concern was we were charging them more than other software company do. Long story, short I asked them to let me attend their internal demo meeting from another company about their product, and me as software consultant from restaurant side. On that meeting I asked about the additional feature with the third company which they didn’t support and ask their price estimation if they must implement those features. Later restaurant got the estimation from another company about, and of course their charge was way above ours. Because they were only selling the general product applicable to all the restaurant, and we were building all the specific feature that this restaurant requires. (Online order, inventory management, table booking, print order in different locations of restaurant buildings). I explained this to the shareholders of restaurant. They took it well and agreed to increase the budget for the project.

* Tell me about time you exceeded customer expectations and what was the outcome of it: I was working for a Software product that helps retailer like Macys, Gap to predict allocation they need to have in their stores for upcoming month, seasons, or quarter. The problem was its performance. It nearly used to take 14 minutes to complete the whole execution. This product was handover to my team with still some feature left to complete from another offshore team in India. So, I started debugging the code, I see part where I can improve the performance. For that I had to move some of the logic to back database side. For some part I had to run some tasks in Parallel. Then outcome was good, I was able to bring the execution time to between 2-3 minutes. Of Business team was happy as they had to demo the product to the client.

1. **Ownership**

Leaders are owners. They think long term and don’t sacrifice long-term value for short-term results. They act on behalf of the entire company, beyond just their own team. They never say, “that’s not my job."

* Tell me about a time you stepped up into a leadership role,

I was working in a team of 38 who must provide maintenance to 6 applications. We must make documents of each bug we fixed and put it on the network folder according to application by month and by date. But it was hard for the other developer, team lead and for the manger also to keep track of those document. We had to check manually whether we made documents for that bugs we worked as some bugs need days to fixed and some only take couple of hours. This was more important at the audit time. So, I made the application voluntarily which will crawl through that network folder and find out whether developer had made document by comparing it again the list of bugs that were solved for that week and send email to the respective developer if it is not able to find the document for the bugs they were assigned for. This saved lots of time of the team member. It saved at least 15 minutes time of developer per week which means about 10 hours per week and 40 hours for month. I think which is great hep for the team.

* Describe a time when you sacrificed short term goals for long term success,

1. Alteryx example from 1.
2. Sure, let me share a task, that time I was working parttime freelancer. Only one guy working form India, code so coupled, convoluted, no separation of concern, He was able to do the demo in time I am sure in future if he has to add new feature it will be easier for him.

* Tell me about a time when you had to push back to HQ or challenged a decision

1. Not writing unit test case, but I wrote the utility helper class to test the main flow of the example.
2. **Invent and Simplify**

Leaders expect and require innovation and invention from their teams and always find ways to simplify. They are externally aware, look for new ideas from everywhere, and are not limited by “not invented here." As we do new things, we accept that we may be misunderstood for long periods of time.

* Tell me about a time when you were 75% through a project, & you had to pivot strategy to ensure successful delivery.

When you are working using azile methodology there is rare chances of this kind of condition. If that happens, I believe is because of very poor communication. We had to rewrite the application to. netcore to host in another cloud service provider. New to. netcore (2017) but I learned the whole process and converted it to. netcore api.

* Tell me about a time you had to deal with ambiguity

At the middle of the project, had azure subscription

* Tell me about a complex problem you worked on.

Explain the BTE.

* Describe a long-term project that you managed. How did you keep everything moving along in a timely manner?

Explain the BTE.

1. **Are Right, A Lot**

Leaders are right a lot. They have strong judgment and good instincts. They seek diverse perspectives and work to disconfirm their beliefs.

* When you had to pick btw different technologies for a project
* Changing to dot net core and angular.
* Tell me about a time when you were wrong
* Developing the background job as library style.

1. **Learn and Be Curious**

Leaders are never done learning and always seek to improve themselves. They are curious about new possibilities and act to explore them.

Think about when the last time was you really had to learn something completely new, for work or personal ventures, how you applied those learnings, what you used them for and what came out of it.

* What’s the most innovative new idea that you have implemented?
* Imagine cup
* Tell me something interesting you've learned recently
* Signal r and implemented to notify the the file import process.
* "Tell me about a time you built out a process."
* Wrote debug class
* Tell me about a time you exceeded expectations.

1. **Hire and Develop the Best**

Leaders raise the performance bar with every hire and promotion. They recognize exceptional talent, and willingly move them throughout the organization. Leaders develop leaders and take seriously their role in coaching others. We work on behalf of our people to invent mechanisms for development like Career Choice.

Tell me about a time you had a conflict with someone. How did you resolve it and what did you learn?

When I joined the team, team was using TFS for source control and it was obvious that they were very comfortable using TFS as they were using it for very long time. And after two months of time, I joined the team, it was decided that they will shift to use git for source control using VSTS as tool. But most team member were not comfortable using git as they were completely new on this. Especially, one of my managers (we have two in the team lead by director) was very reluctant using it and he was always in support of going back to TFS. And using git was very troublesome for him. I noticed that because he was overriding the push and commit done by other. I have experience using git from very long time back and I was easily able to track who was doing that.

Being a new team member, it was very difficult for me to go and say to senior team member directly that you are doing something wrong and could not say to other as I though it would hamper on his reputation. So, I talked to my director that I want to conduct one session on how to use git for all team members. Director agreed to that point. And I conducted that session to the team. And during that process I show how to use git step by step and explained how things could go wrong if we do not push our changes to repo. I explained the team the advantage of using git over TFS also. This gave the inner sight of how git was more efficient then using the TFS.

After that session, team member started to ask me when they have difficulty using git. And my manager was also one of them. At that time, I explained him more on detail. Later also he coordinates with me if he has any confusion when he must resolve issue on git.

1. **Insist on the Highest Standards**

Leaders have relentlessly high standards — many people may think these standards are unreasonably high. Leaders are continually raising the bar and drive their teams to deliver high quality products, services, and processes. Leaders ensure that defects do not get sent down the line and that problems are fixed so they stay fixed.

* What's your most successful project - see your standard?
* Import service, Scalable, maintainable, cost went down.
* Tell me about a time when you raised the bar. Watch an expert answer this "Insist on the Highest Standards" interview question.
* Wrote unit testcase for the new project and fixed the broken unit test case on library.
* Tell me about a time you had to make a decision to make short-term sacrifices for long-term gains.
* Wrote unit testcase for the new project and fixed the broken unit test case on library.
* Tell me about a time when you made a decision based on data and you were ultimately wrong.

1. **Think Big**

Thinking small is a self-fulfilling prophecy. Leaders create and communicate a bold direction that inspires results. They think differently and look around corners for ways to serve customers.

* Tell me about a time where you came up with a simple solution to a complex problem." Watch an expert answer this "Think Big" interview question.
* I wrote a stored procedure, which was quite complex to import data from stage tables to master tables. And there was a validation step to filter duplicate data based on business keys which were supposed to be static. But at the end client told us that it will be dynamic. So, I came up with new idea of having different configuration table which will store the keys and generate dynamic query to join table and validate the data. That worked and I was able to complete the task on time.

* Tell me about your greatest success.  
   Imagine cup. Convert whole ETL process to dotnet
* Tell me about a time you were creative.
* Tell me about a time when you solved a problem innovatively.
* Data used to come with Division-Group-Department-Category-SubCategory-style-color-sku. Has to validate against master table for each code. Reverse the logic

1. **Bias for Action**

Speed matters in business. Many decisions and actions are reversible and do not need extensive study. We value calculated risk taking.

* Tell me about a time you had to quickly adjust your work priorities to meet changing demands.
* If you have to work on 5 different projects, how do you prioritize?
* I will prioritize projects based on how critical the projects are based on business impacts/goals which in turn are aligned with the product roadmap. I would also have a consensus with the stake holders to ensure that everyone's on same page wrt. priority.
* How have you convinced others to take action?
* How have you managed risk in a project?
* Managing risk is a regular part of any Program/Project i have spearheaded both in Waterfall /Agile Methodology. As a leader, encouraged my team to constantly raise Risks and reviewed the risk in a regular cadence with key stakeholders. It was weekly or once in two weeks. In Agile, used Scum of Scrum and in waterfall used weekly or bi-weekly RAG status meetings. Any time a Risk is created, assign the priority and expected ETA for Risk to be closed. Make sure all relevant members are included to plan for mitigation.

Ensure alignment from all team members involved on a mitigation plan. If the Risk cannot be addressed in expected ETA, then convert it as an Issue and send communication with an explanation and how it is planned to address in the next cycle.

There is no specific example, but I always try to estimate the time of the task I am working. Constantly working with the stack holder. Talk about TRD matching an levestien algorithm stuffs.

1. **Frugality**

Accomplish more with less. Constraints breed resourcefulness, self-sufficiency, and invention. There are no extra points for growing headcount, budget size, or fixed expense.

* Tell me about a time you turned down more resources to complete a project.
* Refactoring the version code for 2022 apps on work papers.
* Tell me about a time you faced technical, and people challenges at the same time.
* Tell me about a time you had to accomplish big results with very little budget.

1. **Earn Trust**

Leaders listen attentively, speak candidly, and treat others respectfully. They are vocally self-critical, even when doing so is awkward or embarrassing. Leaders do not believe their or their team’s body odor smells of perfume. They benchmark themselves and their teams against the best.

* Tell me about a time when you received negative feedback from your manager. How did you respond?
* Tell me about a conflict with colleague
* Tell me about a disagreement w/ your boss?
* Give me an example of a time you faced a conflict while working on a team. How did you handle that?
* When I joined the team, team was using TFS for source control and it was obvious that they were very comfortable using TFS as they were using it for very long time. And after two months of time, I joined the team, it was decided that they will shift to use git for source control using VSTS as tool. But most team member were not comfortable using git as they were completely new on this. Especially, one of my managers (we have two in the team lead by director) was very reluctant using it and he was always in support of going back to TFS. And using git was very troublesome for him. I noticed that because he was overriding the push and commit done by other. I have experience using git from very long time back and I was easily able to track who was doing that.
* Being a new team member, it was very difficult for me to go and say to senior team member directly that you are doing something wrong and could not say to other as I thought it would hamper on his reputation. So, I talked to my director that I want to conduct one session on how to use git for all team members. Director agreed to that point. And I conducted that session to the team. And during that process I show how to use git step by step and explained how things could go wrong if we do not push our changes to repo. I explained the team the advantage of using git over TFS also. This gave the inner sight of how git was more efficient then using the TFS.
* After that session, team member started to ask me when they have difficulty using git. And my manager was also one of them. At that time, I explained him more on detail. Later also he coordinates with me if he has any confusion when he has to resolve issue on git. Now I think team is more comfortable on using git. And I don’t see such mistake that used to happen at the start of using git.
* The last time you had to apologize to someone,
* To saju when I synced wrong data to cloud environment, overridden user roles, user sync job, time stamp.
* Tell me about a time you made a mistake.
* Can you give me an example of how you manage conflict?
* How do you earn the trust of your team members?

1. **Dive Deep**

Leaders operate at all levels, stay connected to the details, audit frequently, and are skeptical when metrics and anecdote differ. No task is beneath them. when was the last time you dealt with a complex and interesting problem, which you had to dissect, collect data, divide into pieces, consult people, research, combine several ideas together, debug, log, deploy…? I’m sure you get the idea.

* Tell me about a time when you missed an obvious solution to a problem => Left debug code on and deployed to the dev server; pointed to the local file on constructor.
* A time when you faced a problem that had multiple possible solutions => Had to map the tax return line description on to accounts description, Fuzzy sharp has different configuration, had to take different output compare with other output match.
* Tell me about a time when you had a problem and you had to go through several hoops to discover the root cause. => Had to map the tax return line description on to accounts description, Fuzzy sharp has different configuration, had to take different output compare with other output match.
* Tell me about the most complex project you've worked on. => Business tax engine, Project, Many microservices, multiyear version, sharding of database, messages, machine learning reports, iframe.
* How have you changed an opinion or direction using data? =>
* Tell me about a time when you were able to make a decision without having much data metrics in hand.

1. **Have Backbone; Disagree and Commit**

Leaders are obligated to respectfully challenge decisions when they disagree, even when doing so is uncomfortable or exhausting. Leaders have conviction and are tenacious. They do not compromise for the sake of social cohesion. Once a decision is determined, they commit wholly.

* Tell me about a time you disagreed with someone and how you resolved it.
* Tell me about a time when you had a disagreement with your manager.
* How you manage difficult conversations?

1. **Deliver Results**

Leaders focus on the key inputs for their business and deliver them with the right quality and in a timely fashion. Despite setbacks, they rise to the occasion and never settle.

* Tell me about a tough deadline - your success story about how u overcome struggle,
* Alteryx if next year client must again pay form license
* Tell me about a time when you were not able to meet a time commitment. What prevented you from meeting it? What was the outcome and what did you learn from it?
* Tell me about a time you failed and what you learned from it

1. **Strive to be Earth's Best Employer**

Leaders work every day to create a safer, more productive, higher performing, more diverse, and more just work environment. They lead with empathy, have fun at work, and make it easy for others to have fun. Leaders ask themselves: Are my fellow employees growing? Are they empowered? Are they ready for what's next? Leaders have a vision for and commitment to their employees' personal success, whether that be at Amazon or elsewhere.

* What did you do when you needed to motivate a group of individuals?

1. **Success and Scale Bring Broad Responsibility**

We started in a garage, but we're not there anymore. We are big, we impact the world, and we are far from perfect. We must be humble and thoughtful about even the secondary effects of our actions. Our local communities, planet, and future generations need us to be better every day. We must begin each day with a determination to make better, do better, and be better for our customers, our employees, our partners, and the world at large. And we must end every day knowing we can do even more tomorrow. Leaders create more than they consume and always leave things better than how they found them.

## General

1. Why Amazon - prev video

-- innovative, chance to work with smart people, amazon website how good is user experience, can tell how much the team cares about its users, guided by leadership principles, add value to my career. Can contribute to the team from the day one.

1. Tell me about yourself: [what value can bring to your company?]

Hi, I am working for KPMG, where I have been working for 3 years as a full stack developer. My background is computer science. I did my undergrad on Computer Engineering, and I did my master’s in computer science from Maharishi International University. Before My master I did work for two of the company back in Nepal, first one was sewasoft, Then I moved with softvision. In between I was working as freelancer also that time. Working for 4 years back in Now here in KPMG, My responsibility is to work on web applications and background services for enhancement, adding new features and also sometime fix the production bugs, Apart from regular job, I like to participate on events like hackathon and tech meetups, in 2015 I was selected for world finale of Imagine cup competition which is organized by Microsoft in all over the world. That was a great experience to take part with the brightest people all over the world. I am looking for the fulltime roles which is more challenging and help me to grow further.

1. What's your strength / weakness

When writing a code, sometimes I am not able to pay a good attention to detail in a rush to complete the task. Though I give a second look I still sometimes miss my mistake. For that before merging the code I always ask my peer to review my code before merging it.

Mostly after this remote work started, I think I have been not able to manage my time efficiently. Mostly I have been working longer than regular time. Now I have made a commitment that I will learn something new after my regular job hour, it could be anything, algorithms, cooking skill, reading a books.

I always welcomes the feedbacks on the work I did, from my peers of from my seniors and I am ready to brainstorm on the idea solve it on better ways. This always helps me on the team to improve my quality of work that I am doing, and teammates are open to discuss the problem they are facing too. It helps to create a open environment in the team to discuss on the pending issues.

Jobs handled:

#### SPI Buyer:Allocation

Smoothing data: Stores get all the SKU based on the metrices they performed in last season. But there is some capacity that stores can still get when doing smoothing.

Stores:

MaxLimit:

MinLimit:

3000

* Converted the Alteryx to C# background services
* Scalable Apps
* Making team comfortable working with Git
* Help the intern.
* Sharding of database.
* Helping neetu
* Making utility tool to track down the bug reports
* Finding out the full requirement
* Tell me about a time when you were faced with a problem that had a number of possible solutions. When did you take a risk, make a mistake, or fail?
* How did you respond, and how did you grow from that experience? Describe a time you took the lead on a project.
* What did you do when you needed to motivate a group of individuals or promote collaboration on a particular project?
* How have you leveraged data to develop a strategy?
* Tell me a challenge you had where the best way forward was not clear-cut. How did you decide what to do?
* Give me an example of something you tried to accomplish but failed.
* Give me an example of a time when you showed initiative. Tell me about a time when you delegated a project effectively.
* Tell me about a time when you coached someone. When have you used your fact-finding skills to solve a problem?

Sorting the array takes \mathcal{O}(n\log{n})O(*n*log*n*), so overall complexity is \mathcal{O}(n\log{n} + n^2)O(*n*log*n*+*n*2). This is asymptotically equivalent to \mathcal{O}(n^2)O(*n*2).

Text

Description automatically generated

Timeline

Description automatically generated