# 1) How do you work under pressure or how do you handle challenge?

Problem: I recall a situation from my internship with the Epsilon team. It was my second month in the internship, and I was completely new to the technologies used there, with little understanding of the code-flow. There was a planned release in 2 weeks, but the mechanism to resend transactions to the host when it was offline for 30 seconds wasn't working as expected. This task was assigned to me, but since everyone was busy, I received little support from my team lead.

Action: Firstly, I didn't panic, as I enjoy solving problems and saw this as an opportunity to prove myself to the team. I planned how to approach the task, starting by seeking some guidance from senior engineers to better understand the code flow and functionalities. Afterward, I worked hard, putting in extra hours to figure out how the classes and functions were designed. I discovered that the status of the offline transaction in the queue wasn't resetting correctly when there was no response from the host, and I fixed the issue.

Result: My manager and team lead were pleased with my performance. They appreciated my ability to figure things out mostly on my own with little help from the team. This experience boosted my confidence and showed that I can solve problems efficiently, acting in days rather than weeks.

# 2) Describe a time when you failed?

Problem: During the first semester of my master’s program at SJSU, I worked with a team of 4 on a project for a course called Enterprise Application Development. Our task was to clone all the features of Cinemark, a movie ticket booking website, and rebuild it. As part of the project, we had to deliver a presentation discussing the design and architecture at a high level. We divided the work among ourselves, with each member responsible for completing their tasks. However, one team member stopped responding just a day before the project deadline, which resulted in us failing to deliver the presentation properly, and the professor was not satisfied with our performance.

Action: Later, I communicated with the team member who didn’t respond and understood his situation. He came from an electrical engineering background and lacked knowledge about the architecture. I shared my knowledge and resources regarding the MERN stack with him and motivated him to give his best effort. Together, we worked hard to make a positive impression on the professor. We added extra features to the project, such as an analytics dashboard for administrators to view app metrics and a QR code feature to share ticket booking details with users. Additionally, we deployed the application on the cloud, which no other team in the class had done.

Result: As a result, the professor was very pleased with our final presentation, and our team received the Best Project Award. This experience taught me the importance of putting in extra effort and overcoming challenges to achieve success, rather than dwelling on setbacks.

# 3) Demonstrate leadership skills / colleague not responsive?

Problem: During my time at Hexaware Technologies, the company organized its annual hackathon. I participated with a team of three others, and we began work on our project. I assigned tasks to each team member and scheduled hourly update calls to track our progress. However, after a few hours of development, we encountered an issue when attempting to integrate our components. Upon examining the code of a specific component, I found that it was well-written but did not align to the project requirements. It became evident that the colleague responsible had a strong coding ability but had misunderstood the requirements. I explained the requirements to him and requested an estimated time for resolving the issue during our group call. Unfortunately, he did not respond.

Action: I decided to conclude the group call and initiate a separate one-on-one conversation with him. I inquired about his lack of response in the group call, and he admitted to feeling nervous as a new member of the team. To ease his discomfort, I engaged him in a discussion about his impressions of our company and team. Once he felt more comfortable, I provided a thorough explanation of our hackathon project, including the requirements of the problematic component and its dependencies. I then asked him to articulate his understanding of the task. His explanation reassured me that he had grasped the requirements correctly, and I offered him additional support to complete the coding for the component.

Result: Consequently, we successfully integrated all our components, and the project functioned as intended. Our team's performance earned us a spot among the top 15 out of more than 300 teams in the company's hackathon. Both my manager and our team members commended our approach to the challenge and its resolution, setting a high standard for our team within the organization.

# 4) How do you prioritize your work?

Action: Usually, after finishing the standup call in the morning, I list all the tasks I need to complete and break these tasks into sub-tasks when possible. Then, I analyze the impact of each task, considering various constraints such as the frequency of customer usage of a particular feature, whether any other tasks depend on the current task being worked on by another developer or even if it is present in my list. I assign all such tasks as Priority 1. Next, I identify tasks that require communication with other developers, the product owner, or my manager, and assign them as Priority 2 to resolve dependencies on others. Finally, I assign all remaining tasks with Priority 3.

Result: As a result of this planning, I can effectively deliver tasks to both the client and the organization.

# 5) Describe a situation you were proud of?

Problem: During my tenure at NCR, I was assigned a critical task where our software was crashing after a certain number of days at pilot sites. This issue arose just before a major release scheduled for the following day, which required deployment to thousands of sites. Our senior manager was closely monitoring the situation, seeking continuous updates to decide whether to proceed with the release or hold it.

Action: I thoroughly reviewed the logs from all instances and identified a pattern indicating excessive memory usage in the system. I concluded that the crashes were caused by memory exhaustion. Despite limited information in the logs, I diligently investigated the root cause by examining all recent code changes. With guidance from a senior engineer, I pinpointed a memory leak in our code, which was a recurring issue leading to frequent memory leaks. I promptly addressed the issue, and subsequent testing by our Quality Assurance (QA) team confirmed that the memory usage remained stable. Consequently, the code was released, and no further crashing issues occurred.

Result: Given the critical nature of the problem and its resolution, my efforts were acknowledged by upper leadership and customer representatives. They appreciated my contribution and requested updates to the coding standards to include more information on memory leaks and guidelines for writing code to prevent them in the future.

# 6) What are your strengths?

- "I consider my leadership skills to be one of my greatest strengths. During my tenure at NCR, I successfully led a team of three junior engineers within 1.5 years of joining and received significant appreciation from management."

- "I prefer to address problems from the ground up. My approach involves understanding the client's requirements, taking ownership, and delving deeply to find solutions. I honor decisions and always aim to come up with the smartest solutions.

- "I am an eager learner and do not hesitate to experiment with new technologies, which contributes to my continuous growth."

- "I consistently think analytically, viewing situations from an analytical perspective, which helps me in finding the best possible solutions to any problem."

- "I dive into any topic that I come across, which leads to creative and challenging ideas.""

- "I value honesty in my workload management. If I feel that my workload is too heavy to accept another task, or if there is something I do not understand, I always inform my manager."

- ""Lastly, one of my natural talents is my ability to connect with people. I easily make connections with almost anyone and understand how to show compassion to them in the right way.""

# 7) what are your weakness?

"During my college days, shyness was something I struggled with, especially in large groups. I found it intimidating to ask questions or raise points, which often led me to remain quiet. However, I have since overcome this challenge by engaging and communicating with others. After starting my professional career, I began to actively participate in stand-up meetings and present demos to clients for my tasks. My initiative was recognized during both my internship and full-time positions, with my manager frequently acknowledging my capabilities in both development and management."

# 8) What do you expect from the role?

"My expectation of the company is to be part of a collaborative and open-minded environment. I aim to learn new technologies and gain insights into different business perspectives. I am eager to tackle challenging problems and collaborate with cross-functional teams to benefit from diverse viewpoints. Most importantly, I am excited to contribute to the development of cutting-edge solutions that the company is working on. I am confident that the company will continue to grow significantly. Along this journey, I look forward to networking with some of the brilliant minds within the organization."

# 9) What would you bring to the team?

"As a master’s student and a professional with 2.5 years of experience, I can contribute a decent amount of skills and extensive experience in high-value development projects. I have direct expertise in programming languages such as C++, Python, managing and scaling the infrastructure as well as in design, web technologies, and SQL. I excel under pressure and possess strong leadership qualities. I am committed to contributing 100% to a friendly and supportive environment and understand that the quality of work is crucial in transforming requirements into solutions. I assure you that I will bring strong development skills, leadership, and a sense of ownership to the team. Last but not least, I will also bring energy and fun to foster a friendly workplace atmosphere."

• What makes you the most qualified candidate for the role?

• What is a difficult challenge you have faced?

• What has been your proudest achievement?

• When have you solved a problem with limited resources?

• Why do you want to work at Tesla?

# Tell me about yourself

I am Surya Teja Boganatham, currently pursuing my master’s in Software Engineering at San Jose State University. I completed my bachelor’s degree in computer science in 2020 in India. During my undergraduate studies, I interned at MAQ Software for 8 months, working on a project that involved migrating SQL databases to Azure cloud. Subsequently, I joined NCR Corporation as an intern for 10 months before advancing to the roles of Software Engineer I and II over a span of nearly 3 years.

At NCR Corporation, I was part of a team responsible for a payment interface that serves as an intermediary between POS systems and payment hosts. My contributions included leading the software development of an EMV solution for Menards card, addressing challenges in our module during the pilot stage for Couche-Tard, improving the Interac debit transaction process to enhance efficiency, and working on C++ socket programming and HTTP requests for report generation. Additionally, I contributed to a project aimed at eliminating hardware dependency by developing a Python-based POS application capable of generating QR codes with transaction details and creating a website with React.js to assist customers in completing transactions. This project also involved hosting our payment interface in the cloud.

I am passionate about utilizing my skills in C++, Java, and Python to solve complex problems. I enjoy continuously learning and challenging myself with coding problems on platforms like Leetcode. I am eager to apply my love for problem-solving to a dynamic team environment.

# Why do you want to work at Tesla:

"I want to work at Tesla because I really admire the company's work in transforming transportation and promoting clean energy. especially with their Full Self-Driving Technology that makes driving easier and safer for customers. This innovation is a clear reflection of Tesla's core values, which I hold in high regard — striving to do the best, taking bold chances, treating people with respect, committing to ongoing learning, and prioritizing environmental responsibility. I'm excited about the chance to be part of a team that values innovation and is making a difference in the world. I'm also drawn to Tesla's commitment to learning and using the latest technologies. It's a place where I think I could both contribute and grow.

Beyond automotive and renewable energy solutions, I'm fascinated by Tesla's projects like Starlink, which aims to provide internet access across the globe using satellites. This idea of using advanced technology to connect people everywhere is inspiring to me. Working on such forward-thinking initiatives is exactly where I see myself. I'm eager to bring my skills to the table and be part of something that's pushing boundaries and creating new possibilities."

**Salesforce:**  
Working at Salesforce represents a unique and exciting opportunity for me, aligning perfectly with both my professional aspirations and personal values. Here's why I am eager to join this innovative and forward-thinking team because:

1. \*\*Leadership in CRM\*\*: Salesforce stands as the world's largest CRM company, leading the way in how businesses connect with their customers. This leadership position speaks volumes about the company's ability to innovate and adapt, qualities that are essential in the fast-paced world of technology.

2. \*\*Commitment to Social Responsibility\*\*: The company's 1-1-1 model of philanthropy is truly inspiring, pledging 1% of equity, time, profits, or product to help communities. This commitment to making a positive impact on society resonates deeply with my personal ethos of giving back and contributing to a greater good.

3. \*\*Core Values\*\*: Salesforce's core values of Trust, Innovation, Equality, and Customer Satisfaction are pillars that I personally admire and strive towards in my professional life. These values create a culture where I believe I can thrive, innovate, and make meaningful contributions.

4. \*\*Innovative Products\*\*: The focus on products like the Sales Cloud and Heroku demonstrates Salesforce's commitment to providing powerful, scalable solutions for businesses. My interest in these platforms stems from a desire to work on cutting-edge technology that drives business success and innovation.

5. \*\*Bright Future and Growth Opportunities\*\*: Salesforce's future scope, including cloud computing growth, AI integration, and expansion into various industries, promises a dynamic and evolving workplace. The company's focus on providing a unified customer view through its services is particularly compelling, as it aligns with my interest in how technology can enhance customer relationships.

6. \*\*Employee Well-being and Professional Growth\*\*: Salesforce is known for its emphasis on employee well-being and professional development. This supportive environment is ideal for someone like me, who values continuous learning and seeks to grow both personally and professionally.

7. \*\*Impactful Work\*\*: Joining Salesforce means being part of a team that's not just leading in technology, but also making a positive impact on the world. I am enthusiastic about using my skills to contribute to the company's success and to initiatives that benefit society at large.

In conclusion, working at Salesforce is more than just a job opportunity; it's a chance to be part of a community that values innovation, supports growth, and contributes positively to the world. I am eager to bring my skills to Salesforce and work together to create impactful solutions.  
  
  
  
  
  
**Prioritizing the tasks**  
  
**Situation**:  
In a recent project, I was managing multiple high-priority tasks simultaneously. The most pressing was fixing a **critical bug** (SLA calculation bug) in the production environment that was directly affecting users. This bug had caused significant disruptions and required immediate attention. Alongside this, I was also working on **developing a new feature (Widget for Customer support)**  for an upcoming product release, which had a strict sprint deadline. The third task involved collaborating with a teammate to ensure smooth **system integration – because needs to integrate with Five9 systems)** as both of our tasks were interdependent. Our systems needed to work together without issues for the product to function correctly, so timely and coordinated efforts were essential. Each of these tasks had overlapping deadlines, and I had to manage them effectively without compromising quality or delays. The bug fix was the most urgent, but the new feature and integration were equally important for meeting the project’s objectives. Balancing these tasks while ensuring high-quality execution and maintaining clear communication with stakeholders and my teammate was a significant challenge. It required proper planning, flexibility, and efficiency to manage the competing priorities and meet deadlines.

**Task**:  
The main objective was to resolve the **critical bug** primarily to prevent any further disruption to users, ensuring minimal downtime and customer dissatisfaction. After that, I needed to keep **feature development** on track, ensuring that it was completed on schedule to meet the sprint deadline. Finally, working with my teammates for **integration** was key to ensuring that our systems functioned as expected when merged. The bug fix had to take priority because of its immediate impact on the user experience, but I couldn’t neglect the feature development or integration. Each task was time-sensitive, and I had to ensure that the overall project remained on schedule. Each task had a distinct focus, and I needed to address them without affecting the others, which required strong time management and decision-making skills.

**Action**:  
To tackle this, I first addressed the **bug fix** as **Priority 1**. I dedicated uninterrupted time to thoroughly investigate , work with teammates and fix the issue as quickly as possible, while maintaining clear communication with stakeholders about the progress. Once the bug was resolved, I shifted my focus to the **feature development**. I broke down the tasks into smaller, actionable steps and set clear milestones to ensure steady progress while avoiding distractions. In parallel, I proactively reached out to cross functional (Five9) team to discuss the **integration task, scheduled calls for thrice a week**. Early collaboration helped identify potential dependencies and set clear expectations for both of us. We worked together to define a testing plan, By setting up regular check-ins and progress updates, we ensured that the integration would proceed smoothly once both of us completed our respective tasks. Throughout this process, I regularly reassessed priorities to ensure I stayed on track. I didn’t spend on single task for too long and adjusted my schedule if any task required extra attention. This structured and flexible approach allowed me to manage all three tasks simultaneously without neglecting any of them.

**Result**:  
- The approach practically worked well for me. I fixed the **bug fix within the same day**, preventing further disruptions to the user experience.   
- By addressing the most urgent issue first, I ensured minimal downtime and avoided significant user frustration.   
- The **feature development** was completed on time and met all the requirements set by stakeholders.   
- we avoided any potential roadblocks or delays.   
- The systems integrated successfully without any major setbacks, and testing confirmed everything was working as planned.

### **Learning**:

From this experience, I learned several lessons. First, the importance of **prioritization** need to be taken care of. Flexible prioritization is the key. Even though tasks may have set priorities, the ability to quickly reassess and adjust priorities based on new developments is crucial to success. - - **Clear communication**. By keeping stakeholders and teammates informed at each step, I ensured that expectations were aligned, which avoided misunderstandings or delays.   
- **Proactive collaboration** with my teammate, especially in planning for integration, was critical to preventing issues.  
Additionally, **breaking down large tasks into smaller**, helped me not to feel overwhelmed with tasks.  
Adaptability to all these really helped me and eager to use the same process in the future as well.  
  
  
**Conflict Resolution:**  
 **Situation** : "While working at Hexaware Technologies, I encountered a challenging situation during our monolithic to microservices migration project. We had different teams with different priorities - some developers were focused on maintaining the existing system, while others wanted to move quickly with the new microservices architecture. This created tension because we needed to balance speed with stability. Each team had valid concerns about how to proceed."

**Task** : "My task was to help implement the microservices while making sure we didn't disrupt our current system. I needed to work with multiple teams - the backend team working with Golang, our AWS infrastructure team, and our testing team. The main challenge was keeping everyone aligned and ensuring we met all our performance requirements without causing system downtime."

**Action** : "I took several steps to handle this situation. First, I made sure we had regular Agile meetings where everyone could voice their concerns openly. I used JIRA to document all our decisions and track our progress, which helped keep everything transparent. When there were technical disagreements, I would suggest running tests to get actual data rather than arguing based on opinions. Since I had experience with both the old system and new technologies, I could often suggest compromises that worked for everyone. I also made sure to listen to each team's concerns and took advices and explain technical decisions in ways that made sense to different team members."

**Result** : "These approaches worked really well. We successfully completed the migration and actually reduced our deployment time by 40%. The project was so successful that I received the Star of the Month Award for delivering three critical sprint releases on time. Most importantly, we maintained good relationships between all teams, which helped in future projects. Everyone felt heard and respected throughout the process."

**Learning** : " I learned that it's crucial to listen to all perspectives before making decisions. Having data to back up technical choices helps reduce disagreements. Regular communication through structured meetings prevents misunderstandings from escalating. I also realized how valuable it is to understand both the technical and people aspects of a problem. These lessons have helped me handle similar situations in other projects more effectively."

**Proudest Moment**:

The main challenge I faced was integrating the DGI portal with Ricoh's existing Mindshift portal. This wasn't a simple integration - we were dealing with millions of device records that needed precise mapping. The core complexity came from establishing correct foreign key relationships between DGI reference numbers and our existing records. We needed to ensure that every single device in the DGI system matched correctly with its corresponding record in Mindshift.

What made this particularly challenging was the sheer scale - millions of records - and the requirement that all asset information needed to be location-aware. This meant we couldn't just map the records; we had to implement a sophisticated filtering system based on location hierarchies. Plus, all of this had to happen without any system downtime since these portals were actively used by our clients."

Action (300 words): "I approached this systematically. First, I spent time analyzing both data structures to understand exactly how the DGI reference numbers could map to our existing system. I created an efficient indexing system specifically optimized for these reference numbers, knowing we'd need fast query performance with such large data volumes.

The most intense part was the actual integration deployment. I worked through an 8-hour continuous deployment window, handling ROTA calls throughout. I set up automated health checks that would alert us immediately if any mapping issues occurred. I also implemented rollback procedures at various stages - this was crucial because with millions of records, we needed to be able to reverse any step that didn't go as planned.

Throughout the deployment, I was actively monitoring system performance metrics and handling support calls in real-time. I had created a detailed deployment runbook that helped me track each step and ensure nothing was missed during this critical period."

Result (300 words): "The integration was a complete success. We successfully mapped all millions of records without any data loss or system downtime. The location-based filtering system worked perfectly, allowing users to easily view and manage assets across both portals based on their location permissions.

The most significant outcomes were:

* Zero data loss during the migration
* Improved query response times despite handling more complex relationships
* Seamless user experience across both portals
* Successfully maintained system uptime throughout the integration
* All device information properly mapped with their DGI reference numbers

We actually saw an improvement in system performance after the integration, which wasn't initially expected. Users could now access detailed device information more quickly than before, even though we were processing more data."

Learning (300 words): "This project taught me several valuable lessons about handling large-scale integrations. First, I learned the importance of thorough planning - having detailed mapping strategies and rollback procedures in place before starting the deployment saved us from potential issues.

I also gained deep insights into performance optimization when dealing with millions of records. The indexing strategies I implemented were crucial, and I learned how to balance query performance with data integrity. The experience of handling live support calls during a critical deployment was invaluable - it taught me how to prioritize issues and make quick, informed decisions under pressure.

Most importantly, I learned that in large-scale integrations, monitoring and alerting are just as important as the actual implementation. The automated health checks we set up helped us catch and address potential issues before they could impact users. This experience has made me a much stronger engineer, especially when it comes to handling complex system integrations and large-scale data migrations."

**Why META ?**  
  
 I always aim to work on high-performance distributed systems which supports large-scale optimization, and real-time processing. Meta, being one of the pioneers in this distributed industries, this opportunity provides me to work on some of the largest and most sophisticated distributed systems in the world.

What excites me the most is the scale and complexity of Meta’s infrastructure. Handling billions of users, real-time data streaming, with at most consistency that only very few companies do. I am particularly interested in contributing to the infrastructure that supports massive AI pipelines scaling across the globally distributed data centers.

Beyond just the technical challenges, I am deeply interested in Meta’s research and innovation in infrastructure engineering. I actively follow the technical papers and blog posts published by Meta, particularly around scalable storage, and AI infrastructure optimizations. The opportunity to **gain hands-on experience and work on these infrastructures** is incredibly exciting.  
  
Moreover, being in META is like, surrounded by some of the best engineers and researchers in the industry, which eventually provides an ideal environment for continuous learning and growth. The opportunity to collaborate with and learn from the brightest will ultimately help shape my career in the best way possible.

Finally, This role at Meta is not just a job for me—it’s an opportunity to level up my career, to make a big jump in my career to work on cutting-edge problems at a global scale, push my boundaries in terms of distributed computing, and contribute to infrastructure that powers the future of AI and large-scale applications.

**Where you see yourself in the future ?**

In the future, I see myself deeply involved in **Meta’s large-scale distributed infrastructure**, gaining **hands-on experience** in designing, building, and optimizing **high-performance systems** that support billions of users. Having studied Meta’s **technical blogs and research**, I am eager to **dive deeper into its infrastructure, understand its architecture, and work on solving complex real-world challenges** that come with scaling such a massive system.

As I build my expertise, my goal is to **contribute to Meta’s research efforts**, focusing on **optimizing distributed computing, improving large-scale data processing, and enhancing AI infrastructure scalability**. I want to **continuously learn, adapt, and solve critical engineering challenges**, ensuring that the infrastructure remains efficient and reliable.

Ultimately, I aim to **grow into an expert in infrastructure engineering**, playing a key role in building **scalable, high-performance systems** that push the boundaries of **AI and distributed computing**.  
  
  
  
  
  
  
**Tell me which areas you feel need improvement ?**  
  
 - While I've gained a solid foundation in backend development, microservices, and performance optimization over the past years, I'm most eager to expand my knowledge in large-scale distributed systems, particularly as they relate to machine learning. My experience at Hexaware, especially with the microservices, gave me a taste of the challenges in distributed scaling. I'm comfortable with cloud technologies and understand the foundations of distributed architectures.

However, I recognize that handling the scale and complexity of modern data-driven applications requires a deeper knowledge. Projects like the high-performance computing system for airflow prediction, where I used multi-threading, gRPC, and parallel processing, sparked my interest in advanced distributed computing paradigms. I see a huge opportunity to learn more about these in this role.

Essentially, I'm looking to bridge the gap between my current backend and microservices experience and the specialized skills needed to build and manage large-scale, intelligent applications. I believe this is where I can contribute the most and where I have the greatest potential for growth. I'm a quick learner and eager to tackle new challenges in this exciting field.  
and learn at each and every step in this process.  
  
  
**How well you embrace ambiguity ?**  
 - Early in my time at KPMG, I faced a significant amount of ambiguity. I was working on the in-house client portal, and as a relatively new member of the team, I was unfamiliar with many aspects of the system. The codebase was large and complex, written in C++, which I was still developing my proficiency in, and the existing documentation was limited. Furthermore, the team's processes and workflows weren't fully clear to me initially. There was a lot to learn, and it felt overwhelming at times.

One specific area of ambiguity was the testing process. While I knew testing was crucial, the existing test suites were not well-documented, and it wasn't immediately apparent how they covered the portal's various functionalities. I also wasn't familiar with all the tools and libraries the team used, such as Boost. This lack of clarity made it challenging for me to contribute effectively to the testing efforts.

Instead of being discouraged by this ambiguity, I saw it as an opportunity to learn and grow. I started by immersing myself in the codebase, gradually familiarizing myself with its structure and key components. I asked questions, even if they seemed basic, and sought guidance from more experienced team members. They were incredibly helpful and willing to share their knowledge. I also spent time learning the tools and libraries the team used, including Boost, and practiced writing C++ code to improve my skills.

To tackle the ambiguity around the testing process, I took the initiative to map out the existing test suites and document their coverage. This involved reading through the test code, analyzing the portal's features, and creating a clear overview of what was being tested and what wasn't. This not only helped me understand the testing process but also proved valuable to the team as a whole, as it provided a clearer picture of the portal's test coverage. From this, I was able to identify gaps in the testing and create additional test suites, ultimately saving the team 3 hours per week.

This experience at KPMG taught me to embrace ambiguity as a chance to learn and contribute. I realized that it's okay not to have all the answers upfront. What's important is to be proactive, ask questions, and be willing to learn and adapt. I also learned the value of documentation and process improvement. By clarifying the testing process, I not only helped myself but also made the team more efficient. I'm now much more comfortable working in ambiguous situations and I see it as an opportunity to make a real impact.

**Memory Leak issue:**  
 -  **Context:** "At KPMG, while working on the client portal, we encountered a frustrating and intermittent issue: random system crashes. These crashes were difficult to reproduce consistently, making diagnosis a real challenge. The system logs provided some clues, but they weren't very specific, pointing only to general memory-related errors. Adding to the complexity, the codebase was quite large, and as a relatively new team member, I was still getting familiar with its intricacies. We also had the added pressure of these crashes impacting users, so a quick resolution was critical."

 **Action:** "Given the limited information, I took a multi-pronged approach. First, I realized we needed more detailed logs. I collaborated with the team to implement more verbose logging, capturing memory usage, function calls, and other relevant system states. This involved understanding the existing logging framework and adding custom logging statements in key areas of the code. Second, I knew we needed to reproduce the crash more reliably. I worked with the customer support team to gather more information from affected users, like the specific actions they were taking when the crashes occurred. This helped us narrow down potential triggers. Third, I started systematically reviewing the codebase, focusing on areas related to memory management, multithreading, and resource allocation. Because the codebase was large, I used tools like GDB to attach to running processes and analyze memory usage in real-time. I also reviewed code changes made recently, as these were more likely to introduce bugs. Finally, I collaborated closely with senior engineers, sharing my findings and seeking their guidance. We brainstormed potential causes and discussed different debugging strategies."

 **Result:** "After several days of investigation, we were finally able to pinpoint the issue. It turned out to be a memory leak in a multithreaded component of the portal responsible for handling user sessions. The leak was relatively small, but over time, it would exhaust available memory, eventually leading to the system crashes. The extended logging we implemented was key in identifying this, as it showed a gradual increase in memory consumption over time. We were able to reproduce the crash more reliably after gathering specific user workflows. Once we identified the root cause, I worked on fixing the memory leak, carefully testing the fix to ensure it resolved the issue without introducing new problems. After deploying the fix, the system crashes stopped, and we were able to restore stability to the client portal. This saved the team from spending further time on an intermittent, hard-to-reproduce problem and, most importantly, restored the reliability of the system for our users."

 **Learning:** "This experience taught me the importance of systematic debugging and the value of detailed logging in complex systems. It also reinforced the need for collaboration and communication within the team. I learned how to effectively use debugging tools like GDB and how to analyze large codebases. Most importantly, I learned that even with limited information, a structured approach, persistence, and teamwork can lead to successful problem resolution. I also realized how important it is to consider multithreading and memory management issues, especially with C++ applications, and how subtle leaks can have large consequences over time."

**Stories**

some of the stories  
  
Story 1: - developed DGI ( dealers gateway portal) portal and integrated the portal with the main mindshift portal. portal for all the dealers and on field employees under them, developed widgets for asset management, dealers details integration with main portal, stock management and the customer support.   
  
Story 2: - wrote efficient SLA calculation algorithm which takes many values in consideration and dynamic manipulations based on the asset and the parts and on field employees location and also the status of the system - based on the SLA and time worked it calcultates the automatic wages ... elaborate -   
  
Story 3: wrote autmated scripts for the system check and monitoring using grafana and promethues, wrote alogoritms for memory leak issues and elaborate.....   
  
- Story 4: - Developed scheduled jobs workflows, which starts from business uploads the sheets, it extracts the data and updates the tables. elaborate....   
  
- Story 5: - Developed high end data tansmission from the high end Ricoh printers using gRPC and Cpp libraries elaborate.. for the real time transmission and parallel processing of scheduled services for concurrent batch printers ..elaborate.....

- Story 6: - Memory Leak issues, we get the logs from the system, requests the customers about the logs and check the logs, codebase is pretty large, its hard to identify it anong such vast code base and causing random system crash.  
  
 coherty scan – for scanning the memory leak issues.