# Questions for Fedex

1. What are some of the most challenging or exciting projects your team has worked on?
2. What are some of the projects this role is expected to take on?
3. What level of traffic do the applications you manage typically serve?
   1. Who do they serve?
   2. Is scalability and reliability a typical concern?
   3. Since Fedex is a global company, how do you handle deployments and ensure availability across the different regions?
4. What tech stack is typically used for this role? Are there any not really encouraged?
5. What is the onboarding process like?
6. What are the most important values in a candidate for this role?
7. What are some key goals and opportunities for your team and this role in the coming year?
8. Does Fedex offer any resources for learning and professional development? How do you and your team like to stay up-to-date on the latest technology and trends?
9. How is fedex and/or your team leveraging the recent advancements in AI technologies?
10. How would you describe the company culture and the work environment?
11. How does the company support work-life balance for its employees?
12. I’ve read a bit about the FedEx SameDay delivery bot. It was supposed to pilot in 2021? Were you involved in that development?
13. I’ve also read a bit about Fedex SameDay City delivery service and its unfortunate discontinuation. It still sounds like an incredible cross-collaborative project. I’m always interested in how these ideas move from concept to reality.
    1. Can you talk about how it came about and some of the challenges you and your team faced during development?
    2. How long did it take?

# Questions for me

1. Tell me about yourself:
   1. I am someone who is drawn to challenges, which is what led me to pursue a career in math and become an Actuary.
   2. When I first started at ICW, it was almost like a startup. My entire department was only 7 people. There was lots to do so I had to learn a lot of different things very quickly. Early on, there were a lot of repetitive tasks and manual processes that were eating up valuable time. I started picking up programming to automate a lot of my department’s tasks so that we can focus our resources on more analytical work.
   3. Over time, I learned to be a more effective programmer by learning design patterns and efficient data structures and algorithms.
   4. My love for tech and programming drew the attention of the manager for our data analytics team. He began an initiative to redesign and develop our pricing application in Java using the Spring Boot framework. He approached me and asked if I can help with the reprogramming effort. I was putting in some extra hours and split my time between Actuarial Analyses and developing software.
   5. Once the application was launched in production, my time as a developer would end and I would go back to my main duties.
   6. I decided to focus on software development and came to my current company LIDP.
2. Why do you want to work at Fedex?
   1. Rank 41 on Forbes 500 (2023) and fourth largest employer.
   2. It’s been my dream to work at a logistics company and a household name like fedex. Somewhere I can be proud to say I work. Everyone knows and loves Fedex.
   3. Great place to build a career; good reviews on Glassdoor.
   4. Large global company with so many opportunities.
3. Tight deadline:
   1. As a pricing actuary, many of my projects at my prior company had strict deadlines due to regulatory restrictions, business demands, resource availability, and other concerns.
   2. I was leading each quarterly pricing cycle, ensuring that the analysis work was all finished on time, the development and testing were completed, and the application was deployed.
   3. On top of all my usual duties, one cycle I also proposed **a challenging project of my own to build a pipeline to process new outputs and generate insightful reports from the data.**
   4. One thing I’ve learned from having worked on all these projects is that communication and proactiveness are vital to delivering successfully. To achieve this, I try to break every project into smaller deliverables and maintain clear communication with stakeholders so that the moment any deadline is in jeopardy, we may act upon it quickly and early. I use a variety of tools such as JIRA and Teams Tasks, depending on the scope and who I’m working with to make sure the projects are on track.
   5. Through effective prioritization and team coordination, I was successful in delivering on my data pipeline project, completing my analysis work to support our pricing release, and leading and deploying the pricing release.
4. Collaboration:
   1. Implement new state NY – required cross collab with everyone
   2. I’m pretty fortunate in that because I’m so familiar with our pricing application, databases, and business knowledge, I get to work with so many talented folks from all areas such as our Underwriting teams, Product teams, Development teams, and IT.
   3. At my prior company, we had a cross-functional project to implement a new pricing feature for our rating application. This required collaboration with multiple teams including Reserving, Pricing, Underwriting, IT, and Product. To ensure effective communication and cooperation, we had frequent standups to align on goals, expectation, concerns, and deliverables. We used tools such as JIRA and Microsoft Teams for content sharing and progress tracking. By listening to everyone’s input and addressing their concerns, we were able to release a successful product that adjusted our pricing more accurately and fairly.
5. Disagreement:
   1. **Python – Comparison file generator**
   2. I had a project at my prior company, which involved creating a data pipeline to process batches of outputs from our rating application. I went over it with my actuarial team who were initially skeptical of my design choices.   
        
      Most of the code that is maintained by this team has been straightforward scripts. Though this design worked well for a variety of tasks, I saw that things were starting to be inefficient due to the need for code reuse and slightly varied uses. My approach was to utilize the factory pattern in order to provide flexibility and reuse as much code as possible, minimizing mistakes made in the code.  
        
      To make sure everyone was on board (since ultimately we’d all be using and maintaining this code), I started with a few introductory training meetings on object oriented programming and the need for design patterns (as much as I understood them anyways). Armed with the shared knowledge, we then held several meetings discussing our concerns and collaborated on different approaches to solve them.   
        
      We ended up with a mixture of tactics that resulted in code that was not only easy to maintain, but was flexible and extremely efficient.
6. Learn new tech:
   1. I’m not going to lie.. When I first started my current position, I didn’t know very much about programming. My coursework included a few required computer science courses, but I was mainly interested in solving mathematical and statistical problems at the time.   
        
      Once I actually started working at ICW however, it became quickly apparent that I needed to up my programming game. There were a lot of processes that involved manual work with low tolerances for errors. The most important of which involved pulling data from various sources, joining them together, and performing complex calculations to form a report for management to consume. These processes would take weeks to complete due to resource bottlenecking and data validation.  
        
      Because of our small team and limited resources, I decided to automate and distribute the work as much as I could. I taught myself Microsoft VBA in my freetime after work hours and on weekends and built automated workflows and tools for my team to use. (I chose VBA due to its approachability and integration with our excel data files).  
        
      Eventually, over the course of a few months of just working there, I automated the entire workflow and brought down the report generation to just a few hours.
7. Setback/Failure in project:
   1. Earlier on in my position, I was tasked with **removing a feature** from our pricing application. A few months later after its removal, it was discovered that we were non-compliant and needed to add the feature back in. Due to the application, its vcs, and the time lag, this was a much more complicated task. I ended up missing a key value, and caused mispricing in specific circumstances.  
        
      From this experience, I’ve learned valuable lessons about the importance of testing meticulously and implementing a thorough review system. I spoke to my manager about this incident and together, we put together a process of peer review and having thorough validations.
8. Ambiguous project:
   1. CEO vague about request to conduct a territory analysis and suggest pricing adjustment factors based on industry data. Left a lot of details out. Territory Analysis.  
      Went back and forth with manager. Researched topic and read about how others might approach it.   
      Eventually created our own methodology. CEO loved it.
9. Communicate technical concepts:
   1. Dept was very much into cross training. We encouraged lunch and learns. I put together series on oop. Used relatable ideas and comparisons. Created hw to help reinforce. Asked open ended questions.
   2. If no time, then start with big concepts. Drill down into details as necessary.
10. Took initiative:
    1. At current company, within two weeks of starting, I was told that the dev team used to have a trace program that they and senior team members relied on heavily.
    2. I looked into it and found that it had since been deprecated and removed in the next release. Furthermore, it was no longer working in our current branch.
    3. I brought it up in our bi-weekly standup with other teams. Everyone agreed that it would be useful to have so I volunteered to fix it.
    4. I had already started looking into it and had an idea why it broke.
    5. I fixed a few of the lines involving regex patterns, tested it, then pushed my code for others to pull and use.
11. Weakness:
    1. One area I'm working on is recognizing when to let things be. Sometimes, when I come across something I don't fully understand, I tend to dive deep into it and end up going down unrelated paths. It can be hard for me to step back and say, "Okay, that's enough for now. I'll come back to it later." This can sometimes lead to overthinking and trying to plan for every possible situation, even when it's not necessary.
12. Why hire:
    1. I’m an enthusiastic hardworking person who’s open to all perspectives and ideas and embraces tech in everything.
    2. Fedex should hire me for my extensive technical expertise and experience in analytics and software development. I have a track record of automating tasks and developing efficient solutions. Additionally, my experience in cross-functional collaboration and problem-solving skills make me a valuable asset for successful project outcomes.