* **Strength-based (General) Questions**

1. **Tell me about yourself.**

First, I would like to thank you for this opportunity to meet with you. I graduated from the Full Stack Web Development program at Red River College in 2025. However, I have more than 5 years of experience as a Senior Software Engineer in the IT industry. In this role, I was responsible for structuring and developing cloud-native and microservices-based distributed systems. As for my strengths, I have significant experience developing high-performance and scalable lottery systems. On top of that, I have excellent teamwork and time management skills, which enable me to deliver projects smoothly and on time. I’m excited about the opportunity at Pollard Digital Solutions to combine my experience in the lottery industry with my passion for building a secure, scalable SaaS platform.

1. **What are your greatest weaknesses?**

One area I’ve been working on is **my English speaking fluency**, especially in fast-paced conversations. Sometimes I need a bit more time to find the right words or express my ideas clearly.

To improve, I’ve been practicing daily, joining English discussion groups, listening to podcasts, and preparing key phrases before meetings or presentations to feel more confidence. Over time, I’ve noticed that I can communicate more clearly and naturally.

I see this as a weakness I’m actively improving, and it motivates me to keep learning so I can work effectively in English-speaking professional environments.

1. **What are your greatest strengths?**

One of my greatest strengths is designing and implementing scalable and reliable systems. For example, in my past experience, I was responsible for creating a microservice-based distributed system that could handle thousands of simultaneous requests. I carefully structured the database, optimized queries, and implemented caching strategies, which improved performance by over 40%.

Another strength is problem-solving under pressure. During a system outage, I quickly identified the root cause, coordinated with the team, and deployed a fix with minimal downtime.

Finally, I am highly adaptable and enjoy learning new technologies. This allows me to pick up new frameworks or tools quickly, which is essential in the fast-changing tech environment.

1. **What motivates you at work?**

What motivates me most at work is solving challenging problems and seeing the impact of my work. For example, in my past experience, I was tasked with optimizing a search engine that had slow performance. I enjoyed analyzing the service, identifying bottlenecks, and implementing solutions that improved response time by 40%.

I’m also motivated by learning new technologies and improving my skills. When I get the chance to explore new technologies or tools, it keeps me engaged and helps me contribute more effectively to the team.

Finally, I value collaboration. Working with colleagues to share ideas and find solutions motivates me because it creates a sense of accomplishment and continuous improvement.

1. **Why should we hire you?**

I believe you should hire me because I bring a combination of relevant technical skills, problem-solving experience, and a strong commitment to teamwork. For example, in my previous projects, I’ve successfully designed and implemented backend systems using Java, ensuring they were reliable and scalable.

Beyond technical skills, I pay attention to communication and collaboration. I’ve worked closely with teammates to resolve conflicts, prioritize tasks, and deliver projects on time, which I know is important for this role.

Finally, I’m eager to learn and grow in this position. I’m confident that my skills, experience, and attitude will allow me to contribute effectively to your team and help achieve the company’s goals.

1. **How do you handle stress or pressure?**

I believe we would benefit from stress or pressure if we face them in an appropriate way. I handle stress by staying organized and breaking tasks into smaller, manageable steps. For example, during a past project, we faced a tight deadline for delivering a new feature. I prioritized the most critical and urgent tasks, coordinated closely with my team, and focused on one issue at a time.

I also make sure to take short breaks when needed to stay focused and maintain clarity. This approach helped us deliver the project on time without sacrificing quality.

Overall, I see pressure as an opportunity to stay focused, be efficient and improve my problem-solving skills.

* **Competency-based (Behavioural/Descriptive) Questions**

1. **Tell me about a time when you worked in a team.**

In the previous project, our team needed to develop a new feature for a web application within a tight deadline.

My role was to handle the backend API development and ensure it integrated smoothly with the frontend.

I coordinated closely with the front-end developers, attended daily stand-ups to discuss progress and issues, and suggested solutions when integration problems arose. I also shared API documentation to help the team work more productively.

As a result, we completed the feature on time, with no major bugs reported. The team received positive feedback from the project manager for smooth collaboration and efficient workflow.

1. **Describe a time when you solved a difficult problem.**

During an endurance test before system launch, our system started slowing down after running continuously for several hours.

At first, we suspected database inefficiency because queries were getting slower over time. I analyzed the SQL queries and identified inefficient joins and missing indexes. I optimized the queries, added proper indexing, and adjusted the connection pool settings. However, the issue still wasn’t fully resolved.

I then reviewed the performance reports in detail and discovered that the database write speed was dropping. After deeper investigation, I found the root cause – the database log file was writing too frequently to disk, which was slowing down the entire system. I optimized the log configuration and storage setting to reduce disk I/O pressure.

After the fix, the system maintained stable performance for 48 hours without slowdown. This experience taught me the importance of data-driven analysis and verifying every assumption when solving complex problems.

1. **Tell me about a time you had to learn something quickly.**

As containerized services became more widely used, our company decided to gradually migrate existing systems to a container-based environment to improve scalability and deployment efficiency.

The migration schedule was tight, but I didn’t have much hands-on experience with Docker or container orchestration tools.

I quickly created a learning plan. I studied Docker documentation, watched short tutorials, and practiced by containerizing a small internal service first. I also collaborated with DevOps engineers to understand image building, networking, and multi-container setups using Docker Compose. This hands-on approach helped me apply new knowledge right away.

Within a short time, I successfully containerized server components of our system. The new setup reduced development time and simplified maintenance. This experience taught me how to adapt quickly to new technologies and support the team during fast-paced development.

* **Situational / Hypothetical Questions**

1. **What would you do if you had a conflict with a teammate during a project?**  
   If I had a conflict with a teammate during a project, I would first focus on understanding the cause of the disagreement. In most cases, conflicts come from different perspectives or communication gaps. My goal would be to resolve the issue quickly while keeping the team’s productivity and morale intact.

First, I would talk to my teammate privately to understand their point of view and share mine calmly. I would listen carefully without interrupting and focus on finding common ground. If the issue affected the whole project, I would suggest bringing in the team lead or using objective data to guide our decision, rather than personal opinions. Throughout the process, I’d keep communication professional and solution-oriented.

In a previous project, a teammate and I disagreed about using a microservice versus a monolithic design. I proposed we list the pros and cons of each approach and review them with the team lead. We eventually agreed on a modular monolith that balanced both ideas. This experience taught me that managing conflict through calm discussion and data-based reasoning helps build stronger teamwork and better results.

1. **What would you do if you disagreed with a teammate’s idea?**

If I disagreed with a teammate’s idea, I would first focus on understanding their perspective before giving feedback. I’d frame the situation as a shared goal rather than a personal conflict.

Then, I’d take a few steps: first, I’d ask clarifying questions to understand their reasoning. Second, I’d explain my concerns respectfully, using data or examples to support my point. If we still can’t be on the same page, I’d suggest testing both approaches or bringing it to the team for discussion.

In fact, I faced a similar situation while choosing a frontend framework on a project. A teammate wanted to use React as our frontend framework because of the ease of learning and development, but I worried it would not be suited to the large and complex project. So, I prepared my reasoning and data to propose an alternative framework. Then, we discussed it together with the product manger during a planning meeting. By reviewing the future product roadmap and understanding the expected complexity, we chose the framework that I recommended, which better supported the long-term system. That experience taught me the importance of respectful discussion using data to support decisions and collaborating effectively with teammates.

1. **What would you do if you had to meet a tight deadline?**

If I had to meet a tight deadline, I would first make sure I fully understand the scope of the task and the most critical deliverables. My goal would be to complete the project on time while maintaining quality.

First, I would break the project into smaller, manageable tasks and prioritize the ones that have the biggest impact. Then, I’d create a clear schedule with milestones and check-ins, so progress can be monitored. I’d also communicate proactively with teammates to see if tasks can be delegated or if I need support on specific parts.  
Finally, I’d focus on efficiency, using existing tools, automation, or templates wherever possible to save time.

For example, in a previous project, we had to deliver a new feature within a week instead of the planned two weeks. I divided the work into daily tasks, coordinated with the team to handle parallel development, and used automated scripts to speed up testing. As a result, we completed the feature on time, and the client was satisfied with the quality. That experience showed me how careful planning, clear communication, and focused execution help meet tight deadlines.

1. **How would you handle a situation where a client is unhappy with your work?**  
   If a client was unhappy with my work, the problem could be a technical issue, a misunderstanding, or unmet expectations. My approach would be to address the issue professionally, ensuring the client feels heard and that we find a practical solution.

First, I’d listen carefully to understand their concerns fully. Second, I’d analyze the issue to identify the root cause, whether it’s code-related, a process gap, or a requirement misunderstanding. Third, I’d explain my plan to resolve the problem, coordinate with the team to implement it efficiently, and keep the client updated throughout.

For example, in a previous project, a client reported performance issues in a web feature.  
I reviewed the code, optimized database queries, and improved caching, while keeping the client informed of progress. After these improvements, the feature ran smoothly, and the client was satisfied. This taught me that listening carefully, analyzing problems, and proactive communication are key to handling client concerns effectively.”