Section 1 - Design & Architecture

Q1: Could you outline what you would consider to be the important considerations If you were asked to develop a web application that could potentially be used by thousands of users?

- Following would be important considerations for me if I had to develop an application that could be used by thousands of users:

- Ease of use – Should be easy to access, use and understand on any screen size and by all types of users
- Efficient – Application should solve it's purpose in an acceptable time
- Availability – Since the application would be used by thousands of users, I would want it to be available to everyone at all times
- Secure – Application would be at exposure of a large user base and thus it should be very secure and also handle data securely
- Reliability
- Scalability

Q2: In your own words, explain what you understand microservices to be. When would you considerusing them? What are the pros and cons of using them?

- Microservices are single business functioned focusing services or software that solves the need of a single purpose or a single business function. They can be imagined as breaking down large software applications to loosely coupled modules. Many microservices can be used together for in an application. I would consider using them in large systems that perform various functions. In this scenario, one microservice can be dedicated to each function of the application.

Example:
Application–Webbanking
Microservices – Credit, Debit, Balance check, Email notifications, etc.

Pros:

- Easily to develop, deploy, test and scale. Since they are individualized, developing, deploying and testing them becomes easier. They can also be scaled easily as we can supply more resources to one microservice instead of the complete system
- Learning curve is not very steep
- Language independent. Different microservices on the same application can have different technology stack and communicate with each other using APIs.

Cons:

- Increased network communication due to microservices communicating to one-another using APIs.
- If the number of microservices are very large, they are difficult to manage and check for dependencies.

- Understanding when to have a separate microservice for a business function.
- Managing versioning needs and monitoring of various microservices in production might become a challenge

Q3: How would you safeguard your application, preventing fault-prone microservices from making your application unresponsive?

- Thorough testing of microservices is the best way to safeguard an application from becoming unresponsive because of that service. But there can still be scenarios where microservices might crash. Programming fail-safes for the microservices can help in proper handling of the application so that even one microservice goes down, it has minimal effects on the other microservices that depend on it and no effect on the microservices that don't depend on it and keep those parts of applications running. Having backup microservices coupled with appropriate load balancers will also help in saving the complete application from unresponsiveness.

Section 2 - Development & Deployment

Q1: Describe the approach you take in testing your applications.

- The first thing would be to clearly identify and define the requirements that the application should fulfill, the business logic and the needs. I would then test the application by the following approaches against the defined requirements:

- Black Box
- White Box
- Smoke testing for crashes
- Sanity testing on modules
- Functional testing for business logic
- Regression testing for reliability

Q2: If you were working on a new feature along with several other distributed team members and you became blocked and unable to continue with your work in a productive manner, how would you go about resolving the problem so that you could continue on?

- I would first identify what is the reason behind my blockage to production. If the reason is me being unable to figure out how to do something in particular, I would seek help from other team members or seniors. I would consult or refer all the possible resources be it a person, a documentation or any material on web. If I am blocked because of waiting for input or work from some other team member, I would reach out to her/him to ask about the reason of him being unable to provide me with the resources needed. If there's something that I could help them with, I would work it out with them get things done. Completing task, meeting all the requirements, would be of utmost importance to me.

Q3: What do you value in a code base?

- I value the following things in code:
    - Readability
    - Reusability
    - Efficiency
    - Maintainability
    - Structure
    - Extendability

Section 3 - General

Q1: Tell us about a Red Hat product or open source project you really like, and why.

- One of the open-source projects that I really like from Red Hat is Ansible. It simplifies and automates deployment and configuration. I got to know about Ansible back in 2015 when I read the news of Red Hat acquiring it and have been looking into it since then. Another open source project that I really like is what I developed at Red Hat last summer. It was a real-time collaborative web-based ArchiMate drawing tool. More information can be found on my resume or on the brochure in the GitHub repository.

Q2: What do you find most exciting or interesting about working in the software development field?

- Challenges and problem solving. I have always been a problem solver. This directly refers from my motivation to develop PingCampus right after my high school and before starting my Bachelors in Science. I wanted to solve a problem and thus learnt the needed technologies and software development on my own. I love to solve problems and challenges that people face and that is what motivates and interests me as on the software development field there are always new problems and challenges that needs to be solved.

Q3: In your career to date, what are you most proud of and why?

- I have proud of 2 things in my career till date. First would be PingCampus. I developed PingCampus right after high school by learning everything on my own using resources that were openly available to me. I solved a problem faced by thousands of students, which makes me feel proud of myself. Having achieved recognition at that young age made me proud. Secondly, I am proud of the work I was able to do at Red Hat during the short period of time till I was there. Developing a prototype of a tool ready to use from scratch in that less time made me proud as I was working single-handedly on it. Being able to deliver the MVP and recommendations to take the project further in the time given to me for a completely different domain in which I was working at made me feel proud of myself.