1. Tell me about your proudest professional achievement. It can also be a personal or school project.

When I was working with Crown Equipment Corporation as an Intern, I have got a chance to upgrade their Warehouse Management SAAS application (Infolink-4). It uses the following Java web services

1. Java Server Page (JSP)
2. Apache Struts and Spring Framework for the backend
3. Gradle as a build tool
4. WildFly application server with Undertow as a web server

I have upgraded the Java version from 8 to 11, the Gradle version from 2.5 to 5.3 available at that point, and WildFly from 8.0.2 to 16. It took me entire 2 weeks of Sprint to complete and test. Typically, before upgrades were done by senior member staff, I got a chance to work on them because of my can-do attitude.

Later, I got a chance to work on their cloud platform event-driven microservice architecture application (InfoLink-6). Because of my previous experience working with Gradle, I have created new Gradle plugins and written unit tests that were previously missed for the entire plugin project. This helps in reducing code duplication as the plugin can be used in multiple backend microservices Gradle projects. After completion, I was labeled as Gradle master within the department by chief Architect Aaron Harshbarger. These were the achievements while I was going to school taking 16-19 credit hours workload and working simultaneously.

While working as a Research Assistant, I designed an iOS application written in SwiftUI, a UWP application in C# to capture wearables data (Microsoft Band 2 using UWP and Empatica E4 using iOS). For that, I have also created a simple iOS single-page version of the Nasa Task Load Index to measure demand. I have used the AWS OpenSearch service (previously Elasticsearch service) to store sensor data from both applications. AWS API gateway WebSocket to route through AWS lambda function written in Java to get messages back and forth from both applications. An event-driven LMAX disrupter is used in iOS to transfer sensor logs to Elasticsearch whereas a Serilog HTTP sink is used in the UWP application. MVVM design pattern along with thread-safe singleton design pattern is used throughout the project.

I have several moments that I can keep on writing, but these are a few of many of my proudest professional achievements.

2. Tell me about something you have read recently that you would recommend and why. (Can be a GitHub Repo, Article, Blog, Book, etc.)

I have been following Dr. Venket Subramanium for a long time. He is my go-to guy for Java conferences. Although I have never gotten a chance to see his conference talk live, I watched all his conference talks on YouTube. Recently, I got a chance to read his book **Functional Programming in Java: Harnessing the power of the Java 8 Lambda Expressions**. This book helped me learn the functional paradigm of programming using Java. As we know Java uses imperative programming patterns, I have learned Java in the same pattern as other people do. This book has helped me learn how powerful Java streams are. And, I have to say that I have learned about Java streams through his talk, and I have been using it since. But this book helped me mark out the designs that can be implemented using Java streams. This book has been outstanding for making me implement error-less, optimized, and flawless parallel code. Moreover, this book has made me fall in love with Java streams again.

If you want to implement beautiful, elegant Java code, and concise language semantics using Java, and if you want to turn your code easily into parallel, with less ceremony and effort I will highly recommend this book. This book will help you to have a paradigm change in the functional style of code not just by doing things but by also understanding the concept properly.

3. How would you explain to your grandmother what Availity does?

I think my Grandmother is a little too old to understand the concept of technology involved in Availity. However, I can help her understand what Availity does in her own understanding of language. She understands what it means to go to the hospital. She also understands that there can be an exchange of words between the front desk staff and the patients. Front desk staff can't remember all the information at once using spreadsheets. She also understands that there can be transactions between patients and hospitals as patients must pay for the services used. But she doesn't know what the patient-related information is that hospitals can operate or allow to transfer as EDI transaction. That information is easily available to the trusted parties in real-time so that healthcare services can focus on providing more health-related services. Just like my grandmother gives her insight to my parents and me, Availity does give to their users. She does understand that her healthcare claims, eligibility, and requirements are points of focus during service transactions. So, I will tell her that Availity is making service providers work easier by providing a point of service solution which helps reduce patient time in paying for their services.

However, she still might not get why her knitting business is not doing well. Maybe I will tell her that you have a record for everything so she can look at the report. Now she will understand that reporting is necessary for businesses to look at how they are performing. So, Availity will provide report snapshots of the provider's state so administrators can look at how they are performing in the market. She might be wondering why I am giving healthcare information to her. I will tell her that Availity provides services to health care providers. Just like every people need services, service providers do also need services to run, and so do health care providers.

She might not understand the complexity of HIPAA transactions, revenue cycle, EDI transactions, and many more. I will tell her that there are times when she must file for claims to get something done. She must hassle with forms just like she files her taxes. She might have to recheck again. She might wish if she can get help so that she can file sooner and get her tax credits back faster. Just like it, Availity does prevent the hassle of the revenue cycle in health care providers with less effort. When she is filing her taxes, she might not remember which form to use. So, health care offices can be confused with which form to use. Availity essentials payer portal will manage all these things so that they can run smoother. Just like when she needs information, she will try to find the source of truth. So, Availity will be the source of truth to transfer data between the providers.
Lastly, I will tell her that just like she took care of us, Availity will take care of providers when using their service with no hassle.

Question #4, #5, #6

The solution can be found in this **GitHub link**

A working example of a registration form can be found in **React Application** which is hosted as static content in **AWS S3.**

CI/CD pipeline build **[GitHub Actions Pipeline]**

Following tools, programming language, and testing frameworks are used in my code.

- **Gradle(Groovy)** and **npm** as the build tool
- **Java** for writing API in questions **#4** and **#6.**
- **Java** unit tests were written in **Groovy** using **Spock** and **JUnit-5**
- **TypeScript** with **React** with support for **material UI**
- **Jest** for running **React** test

The source code demonstrates the multi-project build using **Gradle**. Using the Gradle node plugin, **npm** scripts can run as **Gradle tasks**. As a result, I created all subprojects to be Gradle projects with support for **npm.**

Java projects are modular projects meaning, the project has **module-info.java** which can be used as modules as mentioned in **JSR-376**

The code in the mentioned **GitHub** link has support for **CI/CD** pipeline using **GitHub Actions**

It will run all the tests, build, and upload the created production build for react application is automatically uploaded into **AWS S3**. The pipeline will run when there is a push into the **main** branch or pull-requests again the **main** branch.

If there is any problem, please let me know.

Best,
Prakash Khadka

Attached here is my response to the homework assignment. Also, the response document can be found in this