How does your experience relate to other experiences you have had?

Problem-solving skills: My experience working with the team to determine runtime environments for a software development project demonstrates my problem-solving skills. We were able to identify the needs of different stakeholders, such as developers, managers, and users, and then develop a solution that met those needs.

Communication and collaboration skills: My experience working with a team to make decisions about the project demonstrates my communication and collaboration skills. I was able to communicate my ideas clearly and concisely, and you were able to work effectively with others to reach a consensus.

## How does your experience relate to other classmates’ experiences?

I am excited to learn from my classmates' experiences. I know that many of them have experience with different programming languages and technologies than I do. I am particularly interested in learning more about React.js and Node.js, as these are technologies that I am interested in using in my future projects. I am confident that I can contribute to the group by sharing my own experiences and by learning from my classmates.

## Tell me about what you taught someone? And what did they learn that they didn’t know before?

During out discussions this week about different types of database structures, I explained the following about PostgreSQL as being a hybrid of SQL and NoSQL:

* SQL databases are well-suited for storing and querying structured data, such as customer records, product catalogs, and financial transactions.
* NoSQL databases are well-suited for storing and querying unstructured data, such as user-generated content, social media data, and sensor data.
* PostgreSQL is a relational database management system that also supports some NoSQL features, such as JSON and document storage. This makes PostgreSQL a hybrid of SQL and NoSQL.
* This hybrid design gives PostgreSQL a broader range of applications than traditional SQL or NoSQL databases. For example, PostgreSQL can be used to store and query both structured and unstructured data, and it can be used for both OLTP and analytical workloads.

With this understanding we know that PostgreSQL can be used to store and query customer records in a traditional SQL table, and it can also be used to store and query JSON documents in a NoSQL collection.

Before this conversation, they didn't know that PostgreSQL was a hybrid database before. They thought that it was just a SQL database. After I explained it to them, they realized that PostgreSQL is a more versatile database that can be used for a wider range of applications.

## If you were to write your experience as STAR story, how would you phrase it?

I was tasked with working with a team to come up with a technology stack for a new web application. We needed to choose a front-end framework, back-end framework, database, and version control system. We researched different options and discussed our findings with the development team lead. We also considered our budget and timeline constraints. We came up with a few potential technology stacks: Front-end: Blazor/React/Angular, Back-end: ASP.NET/Spring/Scala, Database: MongoDB/ScyllaDB, Version control: GitHub Enterprise Cloud (preferred) or GitLab. We also decided to use GitHub Actions for CI/CD. Our chosen technology stack is well-suited for developing a web application. It is scalable, secure, and easy to maintain. We also chose technologies that are popular and have a large community, which means that it will be relatively easy to find help and support if we need it.

It is important to carefully consider all the factors involved when choosing a technology stack. This includes the type of application being developed, the budget and timeline constraints, and the skills and experience of the development team. It is also important to choose technologies that are well-suited for each other and that can work together seamlessly.

## If this was a religion class, how would you relate this week’s topic to the gospel?

In this week's class, we have been discussing RunTime Requirements Elicitation. Requirements elicitation is the process of gathering and understanding the requirements of a software project. This is an important process, because it ensures that the software project will meet the needs of the users and the business.

In the gospel, we are taught that we should work together to build the kingdom of God. We are also taught that we should be selfless and put the needs of others before our own. When we are eliciting requirements for a software project, we can apply these principles by considering the needs of all the stakeholders and by working together to elicit the best requirements possible.

We are also taught in the gospel that we should love our neighbors as ourselves. We are also taught that we should be compassionate and understanding. When we are eliciting requirements for a software project, we can apply these principles by seeking to understand the needs of the users and the business.

Finally, we are taught in the gospel that we should use our talents and abilities to serve others. When we elicit requirements for a software project, we are using our talents and abilities to create something that will benefit others.

I believe that we can learn a lot from the gospel about how to elicit requirements for a software project. By applying the principles of teamwork, cooperation, love, compassion, and service, we can elicit requirements that will help us to create successful projects that will benefit the users and the business.