**Assignment 1**

1. **What is Microservices?**

Microservices are small, autonomous and loosely coupled services that work together. Here, each service is in separate codebase which deployed independently. They communicate with each other by using well defined APIs and internal implementation of each services are hidden from other services.

1. **Challenges with monolithic oriented architecture.**
2. The design and development of applications using monolithic architecture is easy. Though, it may lead to problems when the application size and complexity increases. Because, it becomes difficult to understand the application that is too large and made changes fast and correctly.
3. When there are many services which are running in one application and have single startup of application, it makes the entire application to slow down the startup-time.
4. It is required to redeploy the entire application on each update.
5. Fault Tolerance is not met in Monolithic Architecture.

Consider a scenario when one of the important services of an application gets down and other services which are dependent on that service also gets down. This leads to failure of business and compromises the quality of experience.

1. Continuous deployment is difficult.
2. Monolithic Applications are difficult to scale when different modules have conflicting resource requirements.
3. Monolithic Applications do not adapt newer technologies. Because changes in frameworks or languages will affect entire application. It is also extremely expensive in both time and cost.
4. **Any three advantages and disadvantages.­­**

Advantages

1. Microservices have a small codebase. This makes it easy to add new features.
2. Microservices are deployed independently, makes it easier to manage bug fixes and feature releases. This requires a small team size because a small team can build, test, and deploy which promotes greater agility.
3. Microservices can be build using mix of technology stacks which best fits the requirement of the service.
4. Fault isolation becomes easier. If an individual microservice becomes unavailable, it won’t disrupt entire application because microservices are independent to each other.
5. Microservices can be scaled independently which can scale sub-units without scaling out the entire application.
6. Schema access and updates is much easier because only single microservice is affecting the schema.

Disadvantages

1. Entire application containing microservices is much more complex. The database entities which are dependent on each other through foreign key makes it difficult to design the schemas which are independent for each microservice.
2. Microservices can create network congestion and latency on server. This can lead to performance issues.

If an application contains small granular services, then it results into more interservice communication. Also, if one service calls chain of services to get the data then it becomes the problem of latency.

1. The decentralized approach of microservices has advantages of using mix technologies stack, but also there are disadvantages. There can be use of so many different languages and frameworks that makes difficult to maintain application.
2. Development and testing of dependent services over a distributed is complex when services are evolving.
3. Sometimes, it may lead to data redundancy.

To make microservices independent, some of the data other schemas are temporary stored in other services schemas, which creates the problem of data redundancy.