**REPORT-5**

**INTERSHIP IN CS**

Hello professor,

The past three weeks went very interesting I came across new things to learn. My concept for this week is WebServices. As we know that there are many web services which are used right now in the software field like SOAP (Simple Object Access Protocol) which follows SOA (Service Oriented Architecture). Soap is nothing but an xml- based messaging protocol. Next one RESTful WebServices it is nothing but Representational State Transfer. Whereas Soap is stateful and RESTful is stateless. The other one which came into existence is the AWS (Amazon Web services). In this report I won’t be explaining you much about these WebServices much but I’m pretty sure that I will cover about these topics in my final report. This report is mainly on MICROSERVICES. It is nothing but another new term, or a new way to build living software. When it comes to microservices it would be better for us to know about the previous architecture that is Monolithic System.

**Monolithic System**

* Monolithic systems are traditionally built as one big system, and they are getting harder to manage in the fast world of today’s customer expectations.
* These are problematic to scale and deploy and time-consuming as well.
* These Monoliths are inseparable and tightly coupled.
* Entire Monolith must be rebuilt and redeployed even if we want to make a small change.

So to overcome these problems at present we are going with the Microservices.

**Microservices Architecture**

* Developing a single application as a suite of small services.
* Communicates with lightweight mechanisms in which each runs in their own process.
* It is an Event-driven Architecture.
* It can be written in different programming languages.
* Easy to keep updated.
* It will not disrupt the whole system.
* It can be implemented with legacy systems still in place.
* Developers can easily collaborate.
* It can be autonomously developed, launched and die on their own.
* Microservices does one thing, and does it well. It owns each service (“you build it, you run it”).

Thank you