Name: Yash Lakhtariya
En no: 21162101012
Branch: CBA Batch: 51

Microservices Assignment-1

1) Microservices architecture will be my preference over monolithic and SOA for the given application scenario. Unlike monolithic architecture in which contains single base for whole application and single development style for all features, microservices architecture can be more helpful for choosing different programming preferences, containerized and then integrated to get different features of cart as required and further modify them if required independently without affecting other features. Also, custom features and preferences are present. Hence, unlike SOA, I can make build different services as per custom preferences. Therefore, microservices architecture is preferred over other two.

2] For the first system to keep records of hospital, there may be custom data models and preferences as per the necessities. Hence, it can be built using microservices architecture, unlike buying commercial software designed for general goals.

But analysing the other system for managing hospital parking lot, it doesn't need to be designed manually due to absence of custom preferences. Managing parking lot is a general goal and getting a nice commercial software for the same would be more preferable.

3] No, absolutely not! It is the major advantage of microservices architecture of being independent of programming preferences. Different microservices can be implemented using different programming languages as per the requirements. As Java is popularly known for its ecosystem, OOPs features, it can be used to do so in case of a web service or platform independent services, but like Python, NodeJs, C#, etc. can also be used in specific needs for different features. Also, a single app may contain microservices,

implemented using different programming languages. But, there is no general rule of using Java only for service implementation in microservices architecture.

4.1
```
function display() {
    var a = b = 10;
}
console.log('b', typeof b === 'undefined');
console.log('a', typeof a === 'undefined');
```

Output: b false
a true

Reason: Here, inside function named display, var a = b = 10 will declare variables a = 10, b = 10 but a is treated as var as it is declared with var keyword and hence has a local scope limited to function only, which is undefined outside it. But b is treated as global variable due to not being declared with let, var, const, etc. keywords which has value 10 and is not undefined in whole program, even outside the function.

4.2
```
'use strict';
function display() {
    var a = b = 10;
}
```

```
display();
console.log('b', typeof b === 'undefined');
console.log('a', typeof a === 'undefined');
```

Output: (Reference error, because b is not defined)

Reason: Unlike previous one, the usage of 'use strict' informs the compiler to run program in strict mode, which enforces stricter parsing and error handling and prevents implicit creation of variables without type i.e. let, var, const, etc. Therefore, as 'b' is not declared with any of those keywords error will be shown.