**Q1 a) What do you know about client server architecture?**

A client server architecture also known as client server model ,is a network application that breaks down tasks and workloads between clients and servers that reside on the same system or are linked by a computer network. The client send request and based on their request server send them response.

There are 3 type of client server architecture

**1-tier architecture :** - In this type of architecture , userinterface ,marketing logic and data logic are present in the same system. Drawback of these kind of system are hard to manage .

**2-tier architecture :-** In this type of architecture, the user-interface is at client side, data is store at server side and marketing logic is either client or server side but need to be maintaine . Online bus ticketing system is consider as 2-tier architecture.

**3-tier architecture :** - A three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms.

Presentation Layer- Manage by Client system

### Application layer- Application Server take-care of this layer

### Data layer – Data server look into this layer

### Any ecommerce website is the example of 3-tier architecture

### b)What is the difference between DNS and IP address?

### IP address - An IP address is a unique address that identifies a device on the internet or a local network. IP stand for Internet Protocol . Internet Protocol used in TCP/IP model for sending packets from source to destination based on IP addresses available in the header of the packet.

### An IP address is a string of numbers separated by periods.They are expressed as a set of four numbers.

### There are two version of IP addresses –

### IPv4 : - Pv4 is a version 4 of IP. It is a current version and the most commonly used IP address. It is a 32-bit address written in four numbers separated by dot or periods. This address is unique for each device.

### IPv6: - IPv6 is the next generation of IP addresses. The main difference between IPv4 and IPv6 is the address size of IP addresses. The IPv4 is a 32-bit address, whereas IPv6 is a 128-bit hexadecimal address.IPv6 provides a large address space, and it contains a simple header as compared to IPv4.

### DNS :- DNS stand for Domain Name system. It is the phonebook of the internet. People get any kind of information on internet with the help of domain name . It is the physical address of any website and is used to memorize IP addresses in simplier version. Whenever we hit any website url , Domain name translator translate domain name into Ip address and our request will get to that Ip address to get the information from that site.

### Q2. What is CAP theorem, discuss a scenario where CAP theorem helps you choose a technology over the other or helps you make a design decision.

The CAP theorem states that a [distributed system](https://www.educative.io/blog/distributed-systems-considerations-tradeoffs) can only provide two of three properties simultaneously: consistency, availability, and partition tolerance. The theorem formalizes the tradeoff between consistency and availability when there’s a partition.

A distributed system is a collection of computers that work together to form a single computer for end users. All of the distributed machines have one shared state and operate concurrently. With distributed systems, users must be able to communicate with any of the distributed machines without knowing it’s only one machine. The distributed system network stores its data on more than just a single node, using multiple physical or virtual machines at the same time.

Consistency :- In a consistent system, all nodes see the same data simultaneously. If we perform a read operation on a consistent system, it should return the value of the most recent write operation.

### Availability : - When availability is present in a distributed system, it means that the system remains operational all of the time. Every request will get a response regardless of the individual state of the nodes.

### Partition tolerance : - When a distributed system encounters a partition, it means that there’s a break in communication between nodes. If a system is partition-tolerant, the system does not fail, regardless of whether messages are dropped or delayed between nodes within the system.

### Q3 What are SOLID principles? Explain with examples wherever possible

### Solid stand for : -

### S means - Single Responsibility Principle

### O means -  Open/Closed Principle

### L means - Liskov’s Substitution Principle

### I means - Interface Segregation Principle

### D dency Inversion Principle

### Single Responsibility Principle : This principle state that if there is a class then that class should be responsible for one and only responsibility.

### Example if a class is performacing taske like –

### ->Open a database connection

### ->Fetch data from database

### ->Write the data in an external file

### The issue with this class is that it handles lot of operations. Suppose any of the following change happens in future.

### ->New database

### ->Adopt ORM to manage queries on database

### ->Change in the output structure

### So in all the cases the above class would be changed. Which might affect the implementation of the other two operations as well. So ideally according to SRP there should be three classes each having the single responsibility.

### Open-closed principle : This principle state that classes should be open for extension but closed for modification . This means that if the class A is written by the developer person1, and if the developer person2 wants some modification on that then developer person2 should be easily do that by extending class A, but not by modifying class A.

### The easy example would be the RecyclerView.Adapter class. Developers can easily extend this class and create their own custom adapter with custom behaviour without modifying the existing RecyclerView.Adapter class.

### Liskov’s Substitution Principle : This principle suggests that parent classes should be easily substituted with their child classes without blowing up the application.

### Interface segregation principle : This principle suggests that “many client specific interfaces are better than one general interface”. This is the first principle which is applied on interface, all the above three principles applies on classes.

### Dependency Inversion Principle : This principle state that classes should depend on abstraction but not on concretion. This means that we should be having object of interface which helps us to communicate with the concrete classes. What do we gain from this is, we hide the actual implementation of class A from the class B. So if class A changes the class B doesn’t need to care or know about the changes.

### Q4 : - Which of the design principles: solid principles or clean code pattern is being violated in this scenario.

### 1st - > All classes violated “ Single Responsibility Principle”