



## 1) In the scenario of encountering 200,000 concurrent users on your web or mobile project, how would you manage the load, and what technologies would you employ to handle such traffic efficiently?

First of all I can use load balancer to divide the load across multiple servers. I can Caching server to lower the number of requests made on the server to reduce the load.

## 2) While your mobile or web application is live, you need to make updates without causing any downtime. How would you achieve this goal and ensure the site remains operational throughout the update process?

I will use CI/CD technique to handle this issue. I will use continuous integration to make changes to the code and after testing got completed it will be deployed to green deployment. Now I change the user connection from blue deployment to green deployment. It will be smooth transmission that no one will notice what just happened.

## 3) If your website faces a Distributed Denial of Service (DDoS) attack, describe your strategy for detecting and preventing it effectively. What tools or techniques would you employ to safeguard your site?

First of all I will use Rate Limiter to lower the number of request made from a single IP address as it can be a brute force attack. Integrating a firewall to lower security risks such as password security, Data Leak risks.

## 4) When dealing with unstructured data in your project that needs to be stored in a database, how would you resolve this issue? Which database system would you choose, and what are the reasons behind your selection?

I will use mongodb to store unstructured data as It provides a lot of tools and query like language to manage the data.

## 5) If your website handles critical transactional data that requires robust security measures, what technology stack would you use, and why? Explain your choice in detail.

I would use aws to store data as It provides secure data management and security. Further more, I will use sql as database because handling a large amount of data in noSQL will be a problem. For frontend will .net as it provides strong connectivity with sql database.

## 6) Given an audience spanning Pakistan and the USA, how would you determine the optimal server locations and implement traffic redirection to ensure users are directed to the nearest server based on their location?

I will make the main server near location where most of the users are located for the rest of the users I will create CDN servers to their nearest location so that response time is minimum.

## 7) When dealing with static data like images, icons, and fonts on your website, how would you optimize the storage and delivery of such content? What technologies or methods would you employ for efficient management?

For the first time this data is loaded it will be stored in cache so that the number of requests can be reduced.

## 8) To implement a real-time notification system within your application, what technology stack or method would you choose, and why? Provide a comprehensive explanation of the chosen approach and its advantages.

To implify realtime notification I will use web sockets to manage the realtime notifications within users.