1.

Firstly, CORS Stands for Cross-Origin Resource Sharing

Secondly it is an HTTP-header based mechanism that allows a server to indicate any origins (Like the port on which the website is served from) other than its own from which a browser should permit loading of resources (This could be the API request).

An example of where CORS is used: I have a frontend app made with react and is served using node JS on port 3000 and then to connect this app to my Database (for API request) which is served on port let say 4000 we need a mechanism to allow sharing of information between the two ports and that is exactly where CORS comes in.

2.

In synchronous operations functions/methods are executed one at a time and only when the method that is currently been executed is completed, the following is executed. In other words, you need to wait for a task to finish to move to the next one (Like a queue).

In asynchronous operations, you can move to another task before the previous one finishes. This usually happen when the compiling engine sees that a function could be executed faster than the others (All functions are thrown in a pool, and which ever gets executed faster is sent back to the user).

An asynchronous execution is when fetching data from an API request.

3.

REST stands for representational state transfer.

A REST API (RESTful API) is an application programming interface (API) that conforms to the constraints of REST architectural style and allows for interaction with RESTful web services.

4.

Mostly is when the app gets frequent updates

And most importantly if the performance of an application is of high concern, then we would use Redux because it skips unnecessary renders. Therefore, if we have a lot of components, then only the once that are called will get renders and thus saving memory which will lead to speed (THE FLASH 'superhero')