**MVC:**

MVC stands for Model, View, and Controller.

**Model**: Model represents the shape of the data. A class in C# is used to describe a model. Model objects store data retrieved from the database.

**View**: View in MVC is a user interface. View display model data to the user and also enables them to modify them. View in ASP.NET MVC is HTML, CSS, and some special syntax (Razor syntax) that makes it easy to communicate with the model and the controller.

**Controller**: The controller handles the user request. Typically, the user uses the view and raises an HTTP request, which will be handled by the controller. The controller processes the request and returns the appropriate view as a response.

JSON:

JSON stands for **JavaScript Object Notation.** It is a **text format** for storing and transporting data. It is self-describing and easy to understand.

It is easy for humans to read and write. It is easy for machines to parse and generate.

JSON is built on two structures:

A collection of name/value pairs. In various languages, this is realized as an object, record, struct, dictionary, hash table, keyed list, or associative array.

An ordered list of values. In most languages, this is realized as an array, vector, list, or sequence.

AJAX:

It is a set of [web development](https://en.wikipedia.org/wiki/Web_development) techniques that uses various web technologies on the [client-side](https://en.wikipedia.org/wiki/Client-side) to create asynchronous [web applications](https://en.wikipedia.org/wiki/Web_application). With Ajax, web applications can send and retrieve data from a [server](https://en.wikipedia.org/wiki/Web_server) asynchronously (in the background) without interfering with the display and behaviour of the existing page. By decoupling the [data interchange](https://en.wikipedia.org/wiki/Data_exchange) layer from the presentation layer, Ajax allows web pages and, by extension, web applications, to change content dynamically without the need to reload the entire page.

Ajax is not a technology, but rather a programming concept.

TRIGGERS:

A SQL trigger is a special type of stored procedure in a database which gets executed automatically whenever special events like any data manipulation (INSERT, UPDATE or DELETE), definition (CREATE, ALTER and DROP) or logon event occurs in the database, usually to protect the integrity of the database or fulfil some business requirements, more like a programmed SQL constraint.

SQL triggers are scheduled procedures that get triggered or executed when a specified event occurs. They are primarily used to enhance the security and integrity of databases.

VIEWS:

A View in SQL is simply a virtual table created based on a result set of another SQL statement. Views were introduced to reduce the complexity of multiple tables and deliver data in a simple manner. Views help us maintain [data integrity](https://www.scaler.com/topics/data-integrity-in-sql/) and provide security to the data, thus acting as a security mechanism. Views were introduced to reduce the complexity of the multiple tables and deliver data in a simple manner. Views hide the complexity of the data in the database as they join and simplify multiple tables into a single virtual table, which is easier for a user to understand.

POST:

Post is used for sending data to the server such as uploading a file or transferring some data or adding a new row to the back end table to any kind of web form. In a simple sentence, we can say that the post method is used for inserting new items in the backend server. In REST CRUD operation it performs the create operation.

PUT:

We use post to create a new resource. A POST request requires a body in which you define the data of the entity to be created.

GET:

We use GET to read or retrieve a resource. A successful GET returns a response containing the information you requested.

The difference between POST and PUT is that PUT requests are idempotent. That is, calling the same PUT request multiple times will always produce the same result. In contrast, calling a POST request repeatedly have side effects of creating the same resource multiple times.