**HTML Fundamentals**

1. What are the basic Features of HTTP?

* It is the protocol that allows web servers and browsers to exchange data over the web.
* It is a request response protocol.
* It uses the reliable TCP connections by default on TCP port 80.
* It is stateless means each request is considered as the new request.

1. What are request methods in HTTP?
   1. GET: to retrieve information
   2. POST: used to send data to the server
   3. PUT: To replaces all current representations with the uploaded content.
   4. DELETE: Removes all current representations of the target which is given by a URI.
   5. HEAD: transfers the status line and header section only
   6. TRACE: Performs a message loop-back test along the path to the target
2. What are the differences between GET and POST methods?

Both GET and POST method is used to transfer data from client to server in HTTP protocol but Main difference between POST and GET method is that GET carries request parameter appended in URL string while POST carries request parameter in message body which makes it more secure way of transferring data from client to server in http protocol.

1. What is status code in HTTP?

Status codes are issued by a server in response to a client's request made to the server. It includes codes from IETF Request for Comments (RFCs), other specifications, and some additional codes used in some common applications of the HTTP. The first digit of the status code specifies one of five standard classes of responses. The message phrases shown are typical, but any human-readable alternative may be provided. Unless otherwise stated, the status code is part of the HTTP/1.1 standard.

* 1xx informational response – the request was received, continuing process
* 2xx successful – the request was successfully received, understood, and accepted
* 3xx redirection – further action needs to be taken in order to complete the request
* 4xx client error – the request contains bad syntax or cannot be fulfilled
* 5xx server error – the server failed to fulfil an apparently valid request

1. What are the header fields in HTTP?

There are four types of HTTP message headers:

* General header: for both request and response messages.
* Client Request-header: for request messages only.
* Server Response-header: for response messages.
* Entity-header: define meta information about the entity-body

1. What is URI?

Uniform Resource Identifier is a string of characters that unambiguously identifies a resource.

1. What are Idempotent methods and why do we call them?

An idempotent method is a method that can be called many times without different outcomes. It would not matter if the method is called only once, or ten times over.

1. What is Session State in HTTP?

Session state is a method keep track of a user session during a series

of HTTP requests. It allows a developer to store data about a user as he/she navigates through ASP.NET web pages in a .NET web application.

1. What is HTTPS?

Hypertext transfer protocol secure (HTTPS) is the secure version of HTTP, which is the primary protocol used to send data between a web browser and a website. HTTPS is encrypted in order to increase security of data transfer. This is particularly important when users transmit sensitive data, such as by logging into a bank account, email service, or health insurance provider.

**Introduction to API**

1. Explain REST and RESTFUL?

Representational state transfer (REST) is a style of software architecture. REST is an "architectural style" that basically exploits the existing technology and protocols of the Web. RESTful is typically used to refer to web services implementing such an architecture.

1. Mention what are the HTTP methods supported by REST?

The following subset of HTTP methods are supported by the REST:

* **GET**

The GET method retrieves specific information from the server as identified by the request URI.

* **PUT**

The PUT method requests that the message body sent with the request be stored under the location provided in the HTTP message.

* **DELETE**

The DELETE method deletes the specified resources.

* **POST**

The POST method modifies data on the server from which a request was sent.

* **HEAD**

The HEAD method is like the GET method except the message body is not returned in the response. The response only includes metainformation, such as a response code or corresponding headers.

1. Explain the architectural style for creating web API?

REST is one of the most popular architecture to implement Web API. It can be coupled by JSONP or RPC or individual endpoints based on the need of the application.

1. Explain the RESTFUL Web Service?

Restful Web Service is a lightweight, maintainable, and scalable service that is built on the REST architecture. Restful Web Service exposes API from your application in a secure, uniform, stateless manner to the calling client. The calling client can perform predefined operations using the Restful service. The underlying protocol for REST is HTTP. REST stands for Representational State Transfer.

1. Explain what is a “Resource” in REST?

The fundamental concept in any RESTful API is the resource. A resource is an object with a type, associated data, relationships to other resources, and a set of methods that operate on it. In this case, we refer to these resources as singleton resources. Collections are themselves resources as well.

1. Which protocol is used by RESTful web services?

RESTful web services use a famous web protocol i.e. HTTP protocol. This serves as a medium of data communication between client and server. HTTP standard methods are used to access resources in RESTful web service architecture.

1. What is messaging in RESTful web services?

RESTful Web Services make use of HTTP protocols as a medium of communication between client and server. A client sends a message in form of a HTTP Request and the server responds in the form of an HTTP Response. This technique is termed as Messaging.

1. State the core components of an HTTP Request?
   1. Verb − Indicates the HTTP methods such as GET, POST, DELETE, PUT, etc.
   2. URI – to identify the resource on the server.
   3. HTTP Version − Indicates the HTTP version. For example, HTTP v1.1.
   4. Request Header − Contains metadata for the HTTP Request message as key-value pairs. For example, client (or browser) type, format supported by the client, format of the message body, cache settings, etc.
   5. Request Body − Message content
2. State the core components of an HTTP response?
   1. Status/Response Code − Indicates the Server status for the requested resource. For example, 404 means resource not found and 200 means response is ok.
   2. HTTP Version – Same as request component
   3. Response Header − Contains metadata for the HTTP Response message as keyvalue pairs. For example, content length, content type, response date, server type, etc.
   4. Response Body − Response message content
3. What do you understand about payload in RESTFUL web service?

The payload is the part of that response that is communicating directly to you. In REST APIs this is usually some JSON formatted data.

1. Explain the caching mechanism?

When a caching mechanism is in place, it helps improve delivery speed by storing a copy of the asset you requested and later accessing the cached copy instead of the original. Otherwise, if the file is stored in cache, this results in a cache hit response and the asset is delivered from cache.

1. List the main differences between SOAP and REST?

The key differences. SOAP is a protocol. REST is an architectural style. An API is designed to expose certain aspects of an application's business logic on a server, and SOAP uses a service interface to do this while REST uses URIs.

1. Enlist advantages and disadvantages of ‘Statelessness’.

Advantages:

* Web services can treat each method request independently.
* Web services need not maintain the client's previous interactions. It simplifies the application design.
* As HTTP is itself a statelessness protocol, RESTful Web Services work seamlessly with the HTTP protocols.

Disadvantages:

* Web services need to get extra information in each request and then interpret to get the client's state in case the client interactions are to be taken care of.