# 1. Explain the concept of RESTful web service, Web API & Microservice

- RESTful Web Service: A RESTful web service is an API that follows the principles of REST (Representational State Transfer). It uses standard HTTP methods and is stateless, meaning each request from a client to server must contain all the necessary information to understand and process the request.

- Web API: Web API is a broader term referring to any API that can be accessed over the web using HTTP. It can follow REST, SOAP, GraphQL, or other protocols. A RESTful web service is a specific type of Web API.

- Microservice: A microservice is a small, self-contained service that performs a single business function. Microservices communicate with each other using lightweight mechanisms like HTTP APIs. They can be developed, deployed, and scaled independently.

# 2. Features of REST architecture

- Representational State Transfer (REST): REST is an architectural style for designing networked applications. It relies on stateless communication and resources identified by URIs.

- Stateless: Each client request to the server must contain all the information needed to understand and process the request. The server does not store any session information.

- Messages: Communication between client and server is done via HTTP messages, which include methods, headers, and body.

- Concept of Microservice: A microservice architecture structures an application as a collection of loosely coupled services, each of which implements a specific business capability.

- Difference between WebService & WebAPI:  
 - WebService: Typically uses SOAP, communicates over HTTP, returns XML.  
 - WebAPI: Can use REST, supports multiple formats (JSON, XML), uses HTTP methods.

- Not restricted to send XML as response: RESTful APIs can return data in multiple formats, such as JSON, XML, HTML, etc., depending on what the client requests.

# 3. Explain what is HttpRequest & HttpResponse

- HttpRequest: An HttpRequest is a message sent by a client to request data or perform an operation on a server. It includes the HTTP method (GET, POST, etc.), URI, headers, and optional body.

- HttpResponse: An HttpResponse is the message sent by the server in response to an HttpRequest. It contains a status code, headers, and optional body (e.g., JSON data).

# 4. List the types of Action Verbs

- HttpGet: Used to retrieve data from the server. Declared using the [HttpGet] attribute.  
- HttpPost: Used to send data to the server to create a resource. Declared using the [HttpPost] attribute.  
- HttpPut: Used to update an existing resource. Declared using the [HttpPut] attribute.  
- HttpDelete: Used to delete a resource. Declared using the [HttpDelete] attribute.

# 5. List the types of HttpStatusCodes used in WebAPI

- Ok (200): The request was successful.  
- BadRequest (400): The request could not be understood or was missing required parameters.  
- Unauthorized (401): Authentication is required and has failed or has not yet been provided.  
- InternalServerError (500): A generic error occurred on the server.

These are returned using ActionResult types in controller methods, such as:  
return Ok();  
return BadRequest();  
return Unauthorized();  
return StatusCode(500);

# 6. Explain the types of Configuration files of WebAPI

- Startup.cs: Used in .NET Core. Configures services and the app's request pipeline. It supports dependency injection using ConfigureServices method.

- appSettings.json: Stores application configuration settings such as connection strings, API keys, etc.

- launchSettings.json: Contains settings used by the local development environment to launch the application with specific profiles.

- Route.config & WebAPI.config (in .NET 4.5 framework):  
 - Route.config: Defines custom routing for ASP.NET applications.  
 - WebAPI.config: Configures routes and other settings specific to Web API such as formatters and message handlers.