WebApi Handson 1 Question and Answers

1. Concept of RESTful Web Service, Web API, and Microservice

RESTful Web Service: An API that follows REST principles:

- Stateless: No session state stored on the server
- Resource-based: Accessed via URLs (e.g., `/api/employees`)
- Uses standard HTTP verbs: GET, POST, PUT, DELETE
- Message-based: Uses request/response model with JSON or XML

Web API:

- A programming interface for web services in .NET Core/.NET Framework.
- Exposes functionality over HTTP.
- Not restricted to XML; it supports JSON, plain text, etc.
- Lightweight and supports HTTP-based communication.

Microservice:

- A small, independent service focused on a single business capability.
- Communicates via APIs.
- Decentralized development and deployment (unlike monolithic apps).

2. What is HttpRequest & HttpResponse?

HttpRequest: Represents the incoming HTTP request from the client. Contains:

- Method (GET, POST, etc.)
- Headers
- Body (payload)
- Query strings and parameters

HttpResponse: Represents the outgoing response sent from server to client. Includes:

- Status Code (200, 404, etc.)
- Headers
- Response Body (e.g., JSON)

3. Types of Action Verbs in Web API

Action Verb	Description	Attribute in Controller
GET	Read or retrieve data	[HttpGet]
POST	Create new data/resource	[HttpPost]
PUT	Update entire resource	[HttpPut("{id}")]
DELETE	Remove a resource	[HttpDelete("{id}")]

4. HttpStatusCodes Used in Web API

Code	Meaning	Use Case
200 OK	Success	Request processed successfully
400 BadRequest	Client error	Missing or invalid input data
401 Unauthorized	No Auth	JWT token missing/invalid
500 InternalServerError	Server crash	Unhandled exception in server code

5. Simple Web API Structure (Read, Write)

A Web API is a project containing:

- Controller class (inherits from `ApiController` or `ControllerBase`

- Methods with `[HttpGet]`, `[HttpPost]`, etc.

Code Example:

```
[Route("api/[controller]")]
[ApiController]
public class EmployeeController : ControllerBase
{
   [HttpGet]
   public ActionResult<List<Employee>> Get() => employees;
   [HttpPost]
   public ActionResult<Employee> Post(Employee e)
   {
      employees.Add(e);
      return CreatedAtAction(nameof(Get), new { id = e.Id }, e);
   }
}
```

6. Configuration Files in Web API

Program.cs / Startup.cs

- This is the main entry point of the application.
- It is used to configure services (like Swagger, JWT, Dependency Injection) and middleware (like authentication, routing, and error handling).
- In .NET 6 and later, Program.cs replaces the traditional Startup.cs.

appsettings.json

- Stores configuration values such as database connection strings, JWT secrets, settings.
- Allows structured, environment-specific configuration management.
- Frequently used for reading application-level constants at runtime.

launchSettings.json

- Used during development to define how the application launches.
- Specifies environment variables, application URLs (e.g., https://localhost:7218), and whether to open a browser.
- This is only used locally and not deployed to production.

Web.config and Route.config (for .NET Framework 4.5)

- Used in older .NET Framework-based Web API projects.
- Web.config holds configuration for connection strings, routes, handlers, authentication, and app settings.
- Route.config (if used separately) defines custom route templates for API endpoints.