**1.WEB API HANDS-ON**

**WEEK-4**

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

**REST (Representational State Transfer)**

* An architectural style for designing networked applications.
* **Features:**
  + **Stateless**: Each request contains all information; no session state stored on a server.
  + **Client-Server**: Separation of UI from backend logic.
  + **Cacheable**: Responses can be cached to improve performance.
  + **Uniform Interface**: Standardized communication between components.
  + **Message Format**: Not restricted to XML—can be JSON, YAML, etc.

### **Web API**

* A framework in .NET for building RESTful services using HTTP.
* Lightweight, easy to build and consume.
* Supports **JSON by default** but can return other formats like XML.

### **Web Service vs Web API**

|  |  |  |
| --- | --- | --- |
| **Feature** | **Web Service (SOAP)** | **Web API (REST)** |
| Protocol | Uses SOAP | Uses HTTP |
| Message Format | XML only | JSON, XML, etc. |
| Lightweight | Less | More (faster, smaller payload) |
| Browser support | No direct browser support | Easily accessed via browser |

### 

### 

### **Microservices**

* Architecture where applications are composed of small, independent services.
* Each service is loosely coupled, independently deployable.
* Web API is a **perfect tool** to implement microservices.

Question 2: Explain what is HttpRequest & HttpResponse.

**HttpRequest**: Carries data from client to server (URL, headers, body, method).  
**HttpResponse**: Carries data from server to client (status code, headers, response body).

Question 3: List the types of Action Verbs.

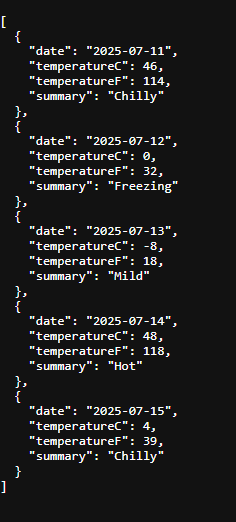
|  |  |  |
| --- | --- | --- |
| **Verb** | **Purpose** | **Attribute in WebAPI** |
| GET | Read data | [HttpGet] |
| POST | Create new data | [HttpPost] |
| PUT | Update existing data | [HttpPut] |
| DELETE | Remove data | [HttpDelete] |

Question 4: List the types of HttpStatusCodes used in WebAPI.

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Name | Description | WebAPI Usage |
| 200 | OK | Request succeeded | return Ok(data); |
| 400 | BadRequest | Invalid input | return BadRequest("message"); |
| 401 | Unauthorized | Access denied without credentials | return Unauthorized(); |
| 500 | InternalServerError | Unexpected error on the server | return StatusCode(500); |

Question 5: Demonstrate creation of a simple WebAPI - With Read, Write actions.

[CODE UPLOADED IN THE SAME DIRECTORY]



Question 6: Explain the types of Configuration files of WebAPI.

### Startup.cs

* Configures services (like DI, authentication, etc.) and middleware.
* Contains ConfigureServices and Configure methods.

### appsettings.json

* Stores app-specific settings (e.g., connection strings, logging).

### launchSettings.json

* Defines how the app is launched (e.g., environment, URLs).

### Web.config

* Used in older .NET Framework apps for app settings, connection strings, and system configs.

### RouteConfig.cs

* Used to define routing for MVC/Web API in older projects: