**1. Why do we use IActionResult instead of ActionResult?**

We use IActionResult because it provides more flexibility by allowing different types of responses (e.g., JsonResult, ViewResult, ObjectResult).

**Scenario:**

If you use ActionResult<ViewResult>, you can only return a view. However, with IActionResult, you can return a **view**, **JSON response**, **redirect**, or even **status codes** based on conditions.

**Problem:**

If you need to return both a ViewResult and JsonResult based on logic, ActionResult<T> restricts you, but IActionResult allows you to return any response type.

**2. What do request and response messages consist of?**

* **Request Message:**
  + **Method:** GET, POST, PUT, DELETE
  + **Headers:** Metadata (e.g., Authorization, Content-Type)
  + **Body:** Data sent to the server (in POST/PUT requests)
* **Response Message:**
  + **Status Code:** (e.g., 200 OK, 404 Not Found)
  + **Headers:** Information like Content-Type, Cache-Control
  + **Body:** Actual data returned (HTML, JSON, XML, etc.)

**3. Difference between HTTP and HTTPS**

* **HTTP (HyperText Transfer Protocol)** is **not secure**, meaning data is transferred in plain text.
* **HTTPS (HyperText Transfer Protocol Secure)** encrypts data using **SSL/TLS**, protecting it from attackers.

**Example:**

* http://example.com → Data is exposed.
* https://example.com → Data is encrypted and secure.

**Benefit:** HTTPS ensures **data integrity, authentication, and security** (e.g., login pages, banking transactions).

**4. What are URL segments and fragments?**

* **Segments:** Parts of the URL separated by /, defining paths.
  + Example: https://example.com/products/electronics/laptops
  + **Segments:** products, electronics, laptops
* **Fragments:** Identified by #, used for navigating within a page.
  + Example: https://example.com/help#faq
  + **Fragment:** faq (scrolls to the FAQ section)

**5. What are Builder and Dependency Injection?**

* **Builder Pattern:** Used to construct complex objects step by step.
* **Dependency Injection (DI):** Manages object dependencies by injecting them instead of creating them manually.

**Real-life Example (DI):**

* Instead of a car directly creating an **Engine** (new Engine()), it receives an **Engine** instance via a constructor.
* Benefit: **Loose coupling** → easier to test and maintain.

**6. Difference between Web Pages (Razor) and MVC**

| **Feature** | **Razor Web Pages** | **MVC** |
| --- | --- | --- |
| Architecture | Single-file, page-based | Separated (Model, View, Controller) |
| Complexity | Simple, lightweight | Suitable for complex apps |
| Business Case 1 | Blog website (small project) | Large-scale e-commerce platform |
| Business Case 2 | Simple admin panel | Multi-user enterprise app |

**7. What is Content-Type in a response message?**

**Content-Type** tells the client **what format the response data is in** (e.g., JSON, XML, HTML).

**Usage:**

* Content-Type: application/json → API returns JSON.
* Content-Type: text/html → Response is an HTML page.

**Why?**  
Ensures the browser or API client properly interprets the response.

**8. Minification, Web Bundling, WebPack, and Lazy Loading**

* **Minification:** Removes unnecessary characters from CSS/JS files to reduce size.
* **Web Bundling:** Combines multiple CSS/JS files into one to reduce HTTP requests.
* **WebPack:** A bundler that optimizes JavaScript and assets.
* **Lazy Loading:** Loads content only when needed (e.g., images load when scrolled into view).

**Performance Boost:**  
✅ Reduces file size  
✅ Fewer network requests  
✅ Faster page load times