**ASSIGNMENT-2**

1) **Difference between xml and json**

**1. Syntax and Structure**

- JSON: Uses a key-value pair format with a simpler and more concise syntax.

```json

{

"name": "John",

"age": 30,

"city": "New York"

}

```

- XML: Uses a nested element structure with opening and closing tags, which can be more verbose.

```xml

<person>

<name>John</name>

<age>30</age>

<city>New York</city>

</person>

```

**2. Data Types**

- JSON Supports a limited set of data types including strings, numbers, booleans, arrays, objects, and null.

- XML: Treats all data as text by default and requires additional parsing to interpret different data types.

**3. Readability and Writeability**

- JSON: Generally easier to read and write manually due to its straightforward and compact syntax.

XML: Can be more difficult to read and write by hand because of its verbosity and nested structure.

**4. Schema and Validation**

- JSON: Uses JSON Schema for defining the structure and validating data, which is less formal but flexible.

- XML: Supports DTD (Document Type Definition) and XSD (XML Schema Definition) for rigorous and complex schema definitions and validation.

**5. Use Cases**

- JSON: Primarily used for web services and APIs, especially in web development due to its efficiency and native support in JavaScript.

- XML: Used in a variety of applications, including configuration files, data interchange in enterprise systems, and document storage where strict validation is required.

**6. Performance**

- JSON: Typically faster to parse and generate because of its simpler and smaller format.

- XML: Parsing and processing can be slower and more resource-intensive due to its complexity and verbosity.

--------------------------------------------------------------------------------------------------------------------

2**) Diff between authorization and authentication**

**1. Purpose**

**- Authentication:** Verifies the identity of a user or entity to ensure they are who they claim to be.

**- Authorization:** Determines the permissions and access levels a verified user or entity has within a system.

**2. Process**

**- Authentication:** Typically involves checking credentials like passwords, biometric data, or tokens.

**-Authorization:** Involves checking access rights and privileges, often using access control lists or role-based access controls.

**3. Order**

**- Authentication:** Always performed first to establish identity.

**- Authorization:** Performed after authentication to decide what the authenticated user is allowed to do.

**4. Data Involvement**

**Authenticatio:** Deals with credentials such as usernames, passwords, and biometric data.

**- Authorization:** Deals with permissions and rules that define what resources or functions the user can access.

**5. Visibility**

**Authentication:** Often visible to the user, as it requires their input to verify identity.

**- Authorization:** Typically transparent to the user; it works in the background once authentication is completed.