**Final Exam**

Data Pre-Processing

Q1: What is XML?

A1: XML stands for eXtensible Markup Language, XML is a markup language used to store and transport data designed to be both human and machine readable. XML doesn’t depend on platform nor software nor programming language. This means it is possible to write a program in any language on any platform to send, receive or store data using XML.

Q2: What is XLST?

A2: XLST stands for XSL Transformations, XLTS is the most important part of XLS allowing to transform a given XML document into another XML document or into another type of document readable by a browser such as HTML and XHTML usually by transforming each XML element into an XHTML element. In order to navigate within an XML document, XPath will be used.

Basically, XSLT transforms an XML source-tree into an XML result-tree.

Q3: What is JSON?

A3: JSON stands for JavaScript Object Notation. It is an open standard file format, and data interchange format, that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and array data types.

Q4: Give an example of JSON?

A4: Example of JSON:

{

"employee": {

"name": "sonoo",

"salary": 56000,

"married": true

}

}

Q5: What is API?

A5: API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you’re using an API.

Basically, an API specifies how software components should interact. Additionally, APIs are used when programming graphical user interface (GUI) components.

Q6: Define browser API.

A6: The HTML Browser API is an extension of the HTML <iframe> element that allows web apps to implement browsers or browser-like applications. It currently works in (privileged) chrome code on Firefox desktop.

Q7: Define third party API.

A7: Third party APIs are APIs provided by third parties, generally companies such as Facebook, Twitter, or Google to allow you to access their functionality via JavaScript and use it on your site. One of the most obvious examples is using mapping APIs to display custom maps on your page.

Q8: Example of XML.

A8: XML examples:

<booklist> // Table

<book> // Entry

<title>Harry Potter</title> // Field

<author>J K. Rowling</author>

<year>2005</year>

<price>29.99</price>

</book>

<book>

<title>The flowers of evil</title>

<author>Charles Baudelaire</author>

<year>1857</year>

<price>39,95</price>

</book>

</booklist>

Q9: Difference between XML and JSON.

A9: A3: Difference between JSON and XML:

|  |  |
| --- | --- |
| JSON | XML |
| 1. JSON object has a type | 1. XML data is typeless |
| 2. JSON types: string, number, array, Boolean | 2. All XML data should be string |
| 3. JSON supports only text and number data type. | 3. XML support various data types such as number, text, images, charts, graphs, etc. It also provides options for transferring the structure or format of the data with actual data. |
| 4. JSON files are easy to read as compared to XML. | 4. XML documents are relatively more difficult to read and interpret. |
| 5. It is less secured. | 5. It is more secure than JSON. |

Q10: Why XML is outdated?

A10: XML was the default data serialization format for about ten years. But JSON’s simpler: it’s more concise, easier to read and edit, and of course its integration into JavaScript is better. (Which means, among other things, that any of the zillions of tools built in node.js use JSON for configuration and serialization.) So you see JSON and JSON serialization everywhere today. But the same things that make XML bloated and hard to read are also things that will keep XML alive indefinitely: namespaces, attributes, text entities, schema validation, etc. These features aren’t useful to people who just want to write configuration files or ship data from one place to another.