LAB EXERCISE - 5

Summary Report of transforming XML to XSL and validation.

1. Purpose of the XSL Stylesheet and XSD Schema

● XSL Stylesheet (restaurant.xsl): The XSL stylesheet is designed to transform the XML data of information about books into a user-friendly HTML format. The transformation presents the data in a structured table, making it easy for users to view and understand nutritional information for various menu items.

● XSD Schema(restaurant.xsd): The XSD schema defines the structure, rules, and constraints for the XML data. It ensures that the data conforms to specific types (e.g., integers, decimals) and meets the required structure, such as the presence of all necessary elements (e.g., Book ID ,Title, Price, Author name, reviews, etc.) within each item.

2. Transformation Process

● Tools Used: The transformation was tested using a standard web browser (Microsoft edge) with built-in XSLT support. The browser automatically applies the XSL stylesheet to the XML file, generating the HTML output.

● Steps: The XML file was loaded with an associated XSL stylesheet reference. o The XSL transformation was applied, converting the XML data into an HTML table that displayed each information about the Books clearly.

● Result: The transformation successfully converted the XML data into a structured HTML table, which displayed without issues in the browser. The HTML table correctly showed all data fields such as Book ID, Title, Author, Genre, Price, Year, Publisher, Reviews.

3. Validation Process

● Tools Used: XML validation was performed against XSD by using XSL and XSL tools extension in VS code.

● Steps: The validation process checked that all data types (e.g., integers, decimals) were correctly assigned, and that all required elements were present.

● Result: When the XML data adhered to the schema, the validation process completed successfully without errors. In scenarios where the XML data violated the schema rules, validation errors were reported.

4. Testing with Various Scenarios

● Scenario 1:

Valid Data - The XML data contained correctly typed values for all fields (e.g., integer values for Book ID, decimal values for Price), and all required elements were present.

Result - No validation errors were encountered, and the transformation process yielded the expected HTML output.

● Scenario 2:

Invalid Data (String in Numeric Field) - The Price field contained a string value (e.g., <Price>Twenty</Price>).

Result - The validation process failed, returning an error indicating that the Price field must be an integer.

● Scenario 3:

User defined type - User-Defined type Positive decimal was not declared and was defined in the code.

Result - The validation process failed and it indicated an error

● Scenario 4:

Missing Required Element - The Author element was omitted from one of the book entries.

Result - The validation process failed, reporting a missing required element error. 5. Errors or Issues Encountered

● Validation Errors : String values in numeric fields, missing required elements, and incorrectly typed values were the most common issues during validation. No errors were encountered during the XSL transformation, indicating that the XSLT code was correct.

6. Documentation of the Solution

● XSL Stylesheet: The XSL stylesheet (transform.xsl) is designed to read the XML data and output an HTML table. It iterates through each Item element and extracts the relevant fields to display in a structured manner. CSS styling is used to enhance readability.

● XSD Schema: The XSD schema defines the structure and data types of the XML elements. It ensures that each Item contains all required fields and that fields such as Book ID , title, reviews, publishers, author name, price have the correct data types (e.g., integer, decimal).

● Script or Program Used: o Transformation: XSL transformation was performed using a web browser and was validated against XSD.