**Documentation for Malnad Stores Website**

**1. Overview**

The project aims to create a web-based product catalog for **Malnad Stores**. The data for the catalog is stored in an XML file, validated against an XSD (XML Schema Definition) to ensure correctness, and transformed into an HTML webpage using XSLT (Extensible Stylesheet Language Transformations). This documentation provides an overview of the website, the data structure, and the validation process.

**2. Website Overview**

**Malnad Stores** is a virtual storefront showcasing various grocery items. The website displays products in a card format with relevant details such as price, quantity, discount, and more. The design is simple, focusing on usability, with a clean layout and responsive cards that highlight individual products.

**Website Components**:

* **Product Cards**: Each product is displayed in a card format with an image, name, price, category, quantity, discount, product code, description, and origin.
* **Styling**: The cards are styled with CSS for a modern look and feel, including hover effects to enhance interactivity.

**3. Data Structure**

**Key Elements**:

* **category (attribute)**: The category of the product (e.g., rice, wheat).
* **name**: The name of the product.
* **price**: The price of the product.
* **quantity**: The available quantity.
* **product\_image**: The image file associated with the product.
* **discount**: The discount offered on the product.
* **product\_code**: A unique code for the product.
* **description**: A brief description of the product.
* **product\_from**: The origin of the product.

**XSD Validation**:

* **Data Types**: Ensures that elements like price, quantity, and discount have the correct data types (e.g., integers, strings).
* **Constraints**: Implements restrictions such as minimum and maximum values for certain elements (e.g., price must be a positive integer).
* **Element Structure**: Enforces the correct nesting and occurrence of elements (e.g., every product must have a name, price, and image).

**5. Transformation Process**

The transformation of the XML data into HTML is achieved using XSLT. The XSLT file (product.xsl) defines how each product's data is displayed on the webpage. The transformation process involves iterating over each product element and converting it into an HTML card.

**XSLT Process**:

* **Template Matching**: The root template matches the XML document and generates the overall HTML structure.
* **Data Binding**: The xsl:value-of element is used to bind XML data to the HTML elements.
* **Styling**: The HTML output is styled with embedded CSS for a polished appearance.

**6. Conclusion**

* The project successfully integrates XML, XSD, and XSLT to create a dynamic and validated product catalog for Malnad Stores. The structured approach ensures data integrity and provides a seamless user experience on the website. The combination of XML for data storage, XSD for validation, and XSLT for transformation showcases a powerful way to manage and display complex data on the web.

Error Handling and Troubleshooting

**Error 1: Browser Cache Issue**

**Symptom**:

* Changes made to the XML or XSL files were not reflected when viewing the output in the browser. Despite multiple attempts to refresh the page, the old content continued to display.

**Cause**:

* The browser cache was storing a previous version of the files, leading to outdated content being displayed. Browsers often cache static files like XML and XSL to improve load times, but this can cause issues during development when frequent changes are made.

**Resolution**:

* The cache was cleared by performing a hard refresh of the browser, which forces the browser to reload the latest versions of all files. This was done using the following methods:



**Misspelling of the Link Tag for XSL:**

* **Issue:** The link tag referencing the XSL stylesheet in the XML file was misspelled, which caused the transformation to not occur as expected.
* **Resolution:** The link tag was corrected to properly reference the XSL file, ensuring that the XML data could be transformed into the desired HTML format successfully.

