**Report: XML Data Extraction and CSV Report Generation Using Python**

**Title:**

**Automated Extraction of CPP\_ERICSSON Configuration Data from XML Using Python**

**1. Objective**

The primary objective of this project is to automate the extraction of 5G site configuration data from a structured XML file and convert the information into a readable and analysis-ready format like CSV. The focus is on enhancing operational efficiency, reducing manual errors, and enabling data-driven decision-making in telecom network planning and optimization.

**2. Tools and Technologies Used**

| **Tool/Technology** | **Purpose** |
| --- | --- |
| Python 3.x | Core programming language |
| xml.etree.ElementTree | Parsing XML data |
| pandas | Data manipulation and DataFrame creation |
| CSV | Report format |
| Excel | Visual representation and formatting |

**3. Input Description**

The input to the program is an **XML file** that contains detailed configuration data of 5G New Radio (NR) base stations. The XML includes multiple nested elements under which information like site ID, cell information, geographical location, and frequency details are stored.

**4. Data Fields Extracted**

The following fields are extracted for each cell configuration:

* site\_id – Unique identifier of the site
* nCI – NR Cell Identity (cell-specific ID)
* cellLocalId – Identifier for local cell within site
* frequency – Transmission frequency
* cellRange – Cell range in meters
* nRCellDUId – DU identifier within the NR site
* nRTAC – Tracking Area Code
* arfcnUL / arfcnDL – Uplink and Downlink Absolute Radio Frequency Channel Number
* latitude / longitude – Geographical location of the cell
* gNBId – Global NodeB Identifier

**5. Methodology**

**a. XML Parsing**

* Used Python’s built-in xml.etree.ElementTree library to load and parse the XML.
* Navigated through nested tags using find() and iter() methods.
* Extracted required elements and attributes.

**b. Data Structuring**

* Stored each record as a dictionary in a list.
* Used pandas to convert the list of dictionaries into a tabular DataFrame for better readability.

**c. CSV Report Generation**

* Exported the DataFrame to a CSV file using df.to\_csv() method.
* File saved as 5G\_site\_data.csv.

**d. Data Verification**

* Verified extracted values using conditional logic and null checks.
* Cross-checked random entries with original XML to ensure accuracy.

**6. Challenges Faced**

* Parsing deep XML hierarchies and handling missing tags.
* Ensuring correct type casting (e.g., float vs string).
* Maintaining consistent column ordering and field validation.
* Handling scientific notation in fields like nCI.

**7. Benefits of This Approach**

* **Automation**: Removes manual effort in reading and copying configurations.
* **Reusability**: The script can be reused for future XML data.
* **Flexibility**: Easily adapted for JSON or Excel output.
* **Scalability**: Can handle large XML files efficiently.

**8. Conclusion**

The solution successfully automated the extraction and tabular representation of 5G NR site configuration data. It serves as a scalable tool for telecom data engineers, planners, and analysts to convert XML-based configurations into standard CSV/Excel formats for analysis, reporting, and planning.