Technical Documentation:

Recommendation System Model

1. **Introduction:**

* This technical documentation outlines the development process of a recommendation system model for Amazon electronics products. The entire process is divided into several key steps, including data transformation, pre-processing, machine learning model creation using the Surprise library, and the implementation of a backend system using FastAPI python and front-end templates using HTML, CSS and JavaScript.

1. **Tools/Libraries used and Purpose:**

* Python: Programming language to build the backend and machine learning model.
* Pandas: For performing ETL operations on the dataset and process the data for the data storage for training.
* NumPy: For transformation during the training in one of the used approaches.
* Surprise: To train the different machine learning models and evaluate them and later used to re-train the model based on new data.
* MySQL: To store user and product details for training of model based on the user selections and other processes.
* FastAPI: Framework to create APIs for the system built on top of Python.
* HTML: To create front-end template.
* CSS: To design the template.
* JavaScript: To make template dynamic and populate data based on the response of APIs.

1. **Data Transformation:**

2.1 Transformation of Amazon Electronics Product Data:

* The Amazon electronics product data, initially in JSON format, is transformed into a data frame using the Pandas library in Python. The process involves reading the JSON file and converting it into a structured data frame for further analysis and manipulation.

1. **Pre-processing of Data:** 
   1. Selection of Camera and Television Products:

* The data frame is pre-processed to filter and select camera and television products. This is achieved by applying Pandas filtering based on product categories.
  1. Handling Null Values and Punctuation Removal:
* Null values are handled using Pandas methods like dropna(), and punctuation is removed from relevant columns.
  1. Custom Processing with Regular Expressions:
* Custom processing, including regular expression-based transformations, is implemented to further clean and preprocess the data.
  1. Save Pre-processed Data to CSV:
* The pre-processed data is saved into a CSV file for future use and easy access.

1. **Machine Learning Model Creation:**
   1. Surprise Library:

* The Surprise library is utilized for building and comparing different machine learning models. Various collaborative filtering algorithms such as SVD, NMF, NormalPredictor, CoClustering, BaselineOnly, and SlopeOne are employed.
* The selected model (SVD) is trained on the entire dataset to learn patterns and relationships between users, products, and ratings. This trained model will be utilized for making recommendations in the production environment.
  1. Model Training and Selection:
* The model with the lowest Root Mean Squared Error (RMSE) is selected as the best-performing model for the recommendation system. In this case, SVD is identified as the optimal model.

1. **Backend System Development:**
   1. Framework:

* FastAPI, a modern web framework for building APIs with Python, is chosen for the backend system development. Its asynchronous capabilities and automatic API documentation generation contribute to the efficiency and maintainability of the system.
  1. Database Setup:
* A database is designed to store essential information, including user details, product details, recommended products, user selections, and feedback. The database schema is crafted to accommodate the specific data requirements of the recommendation system.
  1. Front-end Integration:
* Multiple APIs exposed by the backend system facilitate seamless communication between the front-end template and the recommendation system. The front-end, built using HTML, CSS, and JavaScript, interacts with these APIs to provide users with a responsive and interactive experience.