# Documentation for Matching Logic

This document provides a detailed explanation of the rules and logic applied to determine name match, address match, and overall match for the given application.

## ****Name Matching Rules****

The name\_match function evaluates whether the input name matches the extracted name based on the following rules:

### ****1. Exact Letter Match****

* All letters in the name must match exactly. Any mismatch returns False.
  + **Example**: "gyanesh ray" vs "gyanesh rar" → False.

### ****2. Abbreviated Names****

* If the name is abbreviated (e.g., "J Smith" vs "John Smith"), it will match if the abbreviated part is correct.
* Any spelling mistakes in the unabridged part will result in False.
  + **Example**: "gyanesh ray" vs "gyanesh r" → True.
  + **Example**: "Phani Vishwanath" vs "P Vishwanath" → True.
  + **Example**: "Phani Vishanath" vs "P Vishwanath" → False.

### ****3. Ignoring Middle Names****

* Middle names can be ignored. If a person uses only two names instead of three or more, the names will match.
  + **Example**: "Pankaj Lochan Panda" vs "Pankaj Panda" → True.
  + **Example**: "Pankaja Panda" vs "Pankaj Lochan Panda" → False.

### ****4. Matching Any Part of the Name****

* If a person uses only a first name or last name, it should match with any part of the other name.
  + **Example**: "Devesh" vs "Devesh Tripathi" → True.
  + **Example**: "Devesh Tripathi" vs "Devesh" → True.

### ****5. Circular Matching****

* If the order of name parts changes but all parts are present, it will match.
  + **Example**: "Ray Shushi" vs "Shushi Ray" → True.

### ****6. Single-Letter Abbreviation****

* If one part of the name is abbreviated to a single letter, it will match if the rest of the name matches completely.
  + **Example**: "Ray Shushi" vs "Shushi R" → True.
  + **Example**: "Subramani M" vs "Suamani M" → False.

## ****Address Matching Rules****

The address matching logic evaluates whether the input address matches the extracted address based on the following rules:

### ****1. Normalization****

* Certain common terms (e.g., "Marg," "Lane," "Township") are ignored during matching.
* Non-alphanumeric characters are removed, and extra spaces are normalized.

### ****2. Pincode Matching****

* If the input pincode matches the extracted pincode exactly (ignoring spaces), the score is 100. Otherwise, it is 0.
  + **Example**: "560 001" vs "560001" → 100.
  + **Example**: "560002" vs "560001" → 0.

### ****3. Field-Specific Matching****

* Each address component (e.g., street name, house number) is compared individually.
  + If the component matches completely within the extracted address, the score is 100.
  + Partial matches are scored using a similarity ratio.

### ****4. Final Address Match****

* The final match score is a weighted sum of all field scores, normalized by the total weight.
* If the final score is greater than or equal to a cutoff (e.g., 70), and the pincode matches, the address match is True. Otherwise, it is False.

## ****Overall Match Rules****

The overall match evaluates whether both the name and address match. The rules are as follows:

### ****1. True Cases****

* **Name Match** is True.
* **Final Address Match** is True.

### ****2. False Cases****

* Either the **Name Match** or the **Final Address Match** is False.