NAME: SANJITH S ROLL NO: 23AD083

#### Problem Statement:

You are tasked with creating an agent that can validate HSN codes. HSN codes are an internationally standardized system of names and numbers to classify traded products. They vary in length, typically from 2 to 8 digits, where each level of digits represents a more specific classification. For instance:

- 01 might represent "LIVE ANIMALS."
- 0101 might represent "LIVE HORSES, ASSES, MULES AND HINNIES."
- 01011010 might represent a very specific type of horse under the "LIVE HORSES..." category.

The agent will be provided with a master Excel file containing known HSN codes and their corresponding descriptions. The agent should be able to take an HSN code (or a list of HSN codes) as input and determine its validity based on this master data.

### Flow Of Execution:

## 1. User Input:

User provides a single or multiple HSN codes (via chatbot, API, or interface).

#### 2. Input Validation (Pre-processing):

- Check format: numeric and valid length (2, 4, 6, or 8 digits).
- Sanitize input (strip whitespace, ensure string format).

#### 3. Data Access Layer:

- Load and cache the master HSN Excel data (HSNCode + Description).
- Use optimized read (e.g., pandas.read\_excel) and convert to dictionary for quick lookup.

#### 4. HSN Validation Logic:

- Check if the HSN exists in the master list.
- Optional: Check for parent hierarchy (e.g., for 01011010, check if 01, 0101, 010110 exist).

#### 5. Agent Fulfillment:

- For each HSN, generate a structured response:
  - Valid  $\rightarrow$  Return HSN + description.
  - Invalid  $\rightarrow$  Return error reason ("Not found", "Invalid format").

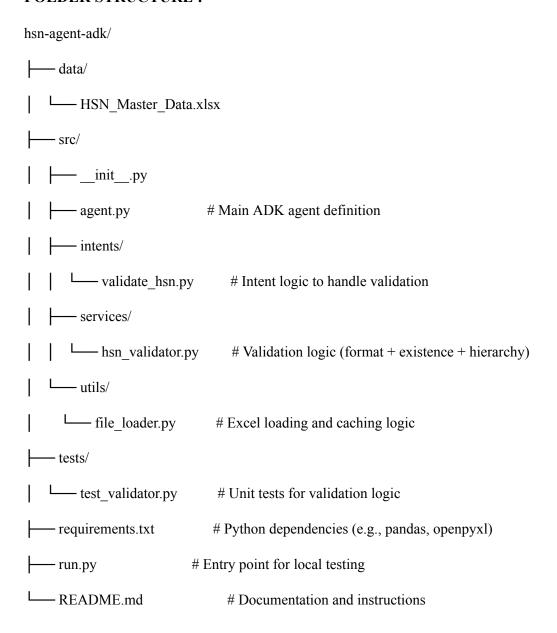
### 6. Response Generation:

ADK formats the output based on fulfillment response (text message, rich response, etc.).

## 7. Optional Enhancements:

- o Log invalid patterns.
- o Provide suggestions or auto-corrections.
- o Allow uploading updated Excel file.

### **FOLDER STRUCTURE:**



#### **CODE:**

## **Step 1: Set Up Environment:**

python -m venv venv source venv/bin/activate pip install -r requirements.txt

# **Step 2: Load Master Excel File**

```
# file_loader.py
import pandas as pd

def load_hsn_data(file_path):
    df = pd.read_excel(file_path, dtype=str)

return dict(zip(df]'HSNCode'].str.strip(), df['Description'].str.strip()))
```

# **Step 3: Implement Validation Logic**

```
# hsn_validator.py

def is_valid_format(code):
    return code.isdigit() and len(code) in [2, 4, 6, 8]

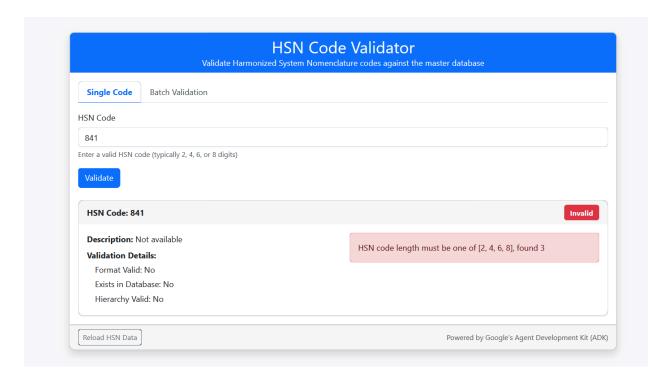
def validate_hsn_code(code, hsn_dict):
    if not is_valid_format(code):
        return {"code": code, "status": "invalid", "reason": "Invalid format"}
    if code in hsn_dict:
        return {"code": code, "status": "valid", "description": hsn_dict[code]}
    return {"code": code, "status": "invalid", "reason": "Not found"}
```

### **Step 4: Define Intent & Fulfillment in ADK**

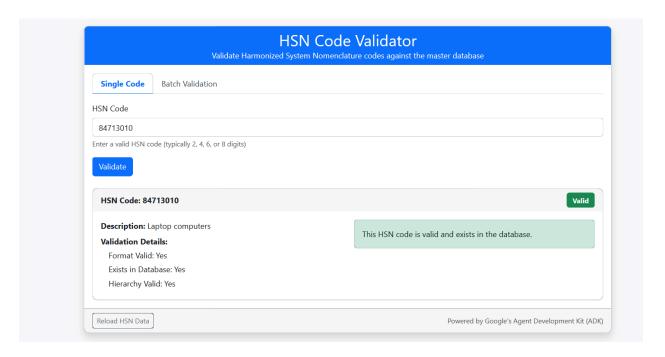
```
from services.hsn_validator import validate_hsn_code
from utils.file loader import load hsn data
```

```
hsn data = load hsn data("data/HSN Master Data.xlsx")
def handle validate hsn(intent input):
  codes = intent input.get("hsn codes", [])
  result = [validate hsn code(code.strip(), hsn data) for code in codes]
  return {"validation result": result}
Step 5: Hook to Agent in agent.py
from intents.validate_hsn import handle_validate_hsn
def route intent(intent name, user input):
  if intent name == "validate hsn":
    return handle validate hsn(user input)
  return {"error": "Unknown intent"}
Step 6: Test the Agent (Optional Local Execution):
# run.py
from agent import route intent
user input = {"hsn codes": ["01012100", "999999"]}
response = route intent("validate hsn", user input)
print(response)
```

## **OUTPUT:**



# a) Invalid HSN code



b) Valid HSN Code