

README

1. Agent Design

1.1 Overall architecture and structure of agent using ADK framework

The agent has the following files

```
MULTI_TOOL_AGENT/  
├── __init__.py  
├── agent.py  
└── HSN.csv
```

`agent.py` contains the tools the agent requires. Also it is where instantiation of the agent takes place.

1.2 Key components of your agent

Intents: it represents what the user wants, the goal behind their prompt, after the LLM recognizes the intent, we can use fulfillment logic. ex. Is 010110 valid?

Entities: it is the key pieces of information we need to pull out from a user's prompt to fulfill the intent. For example in the prompt "Is 010110 valid?", the entity is 010110.

Fulfillment logic: it is the code that actually does the work and produces a response.

```
code_str = code_str.strip()  
  
if not (MIN_HSN_CODE_LENGTH <= len(code_str) <= MAX_HSN_CODE_LENGTH):  
  
    return {  
  
        "status": "error",  
  
        "message": (  
  
            f"HSN code '{code_str}' must be between "  
  
            f"{MIN_HSN_CODE_LENGTH} and {MAX_HSN_CODE_LENGTH} digits."  
  
        )
```

```

}

if not code_str.isdigit():

    return {

        "status": "error",

        "message": f"HSN code '{code_str}' contains non-numeric characters."

    }

if code_str in _hsn_map:

    return {

        "status": "success",

        "message": f"HSN code {code_str} is valid.",

        "description": _hsn_map[code_str]

    }

else:

    return {

        "status": "error",

        "message": f"HSN code {code_str} not found."


    }

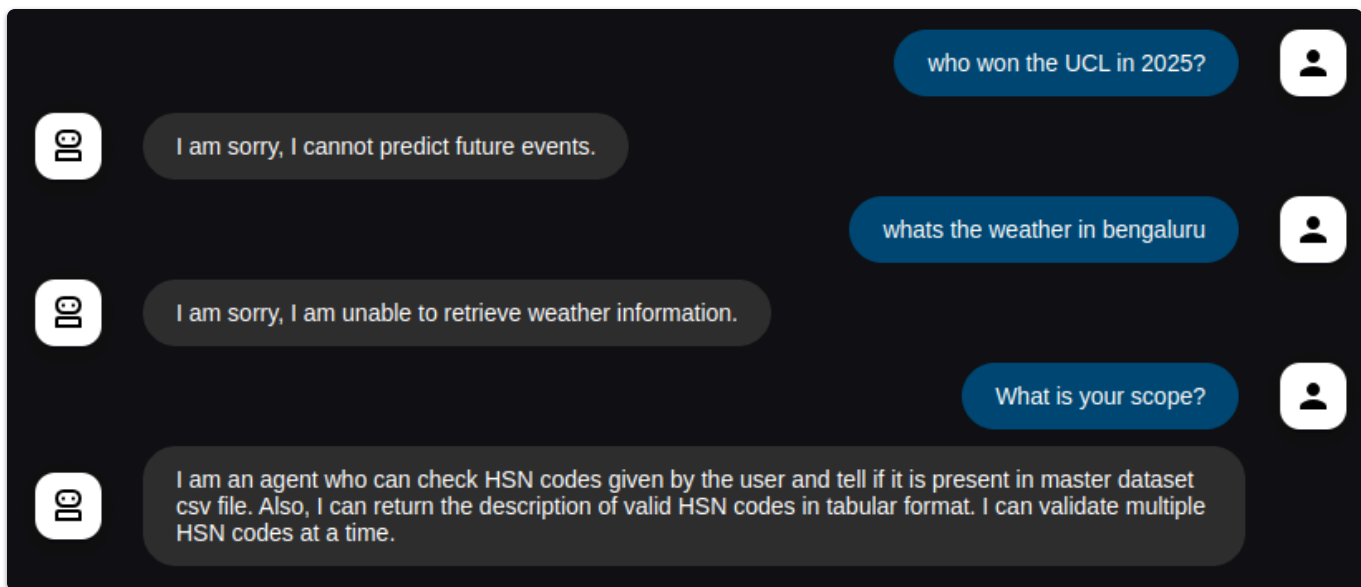
```

1.3 Agent handling user input (eg. a single HSN code, multiple HSN codes)

The agent can handle multiple user inputs if separated by commas. Other separation methods can also be used.

Agent	User
	<div>is 09, 10, 01, 02, 04, 11, 12 valid?</div> 

Agent	User																
 <table> <thead> <tr> <th>HSN Code</th><th>Description</th></tr> </thead> <tbody> <tr> <td>09</td><td>Coffee, tea, mate and spices</td></tr> <tr> <td>10</td><td>Cereals</td></tr> <tr> <td>01</td><td>LIVE ANIMALS</td></tr> <tr> <td>02</td><td>Meat and edible meat offal</td></tr> <tr> <td>04</td><td>Dairy produce; birds' eggs; natural honey; edible products of animal origin, not elsewhere specified or included</td></tr> <tr> <td>11</td><td>Products of the milling industry; malt; starches; inulin; wheat gluten</td></tr> <tr> <td>12</td><td>Oil seeds and oleaginous fruits, miscellaneous grains, seeds and fruit; industrial or medicinal plants; straw and fodder</td></tr> </tbody> </table>	HSN Code	Description	09	Coffee, tea, mate and spices	10	Cereals	01	LIVE ANIMALS	02	Meat and edible meat offal	04	Dairy produce; birds' eggs; natural honey; edible products of animal origin, not elsewhere specified or included	11	Products of the milling industry; malt; starches; inulin; wheat gluten	12	Oil seeds and oleaginous fruits, miscellaneous grains, seeds and fruit; industrial or medicinal plants; straw and fodder	
HSN Code	Description																
09	Coffee, tea, mate and spices																
10	Cereals																
01	LIVE ANIMALS																
02	Meat and edible meat offal																
04	Dairy produce; birds' eggs; natural honey; edible products of animal origin, not elsewhere specified or included																
11	Products of the milling industry; malt; starches; inulin; wheat gluten																
12	Oil seeds and oleaginous fruits, miscellaneous grains, seeds and fruit; industrial or medicinal plants; straw and fodder																



who won the UCL in 2025?

I am sorry, I cannot predict future events.

whats the weather in bengaluru

I am sorry, I am unable to retrieve weather information.

What is your scope?

I am an agent who can check HSN codes given by the user and tell if it is present in master dataset csv file. Also, I can return the description of valid HSN codes in tabular format. I can validate multiple HSN codes at a time.

1.4 Agent providing its validation output

Based on check the agent will parse the returned json and show the output.

2. Data Handling

2.1 Agent's access to file and processing of data

1. The file is loaded and read into memory once using `pandas`.
2. The columns and column keys are normalized by removing whitespace. Other normalization like missing values etc. should be done before feeding the agent the file.
3. Build in-memory index
 - A dict mapping code → description for O(1) lookups.
4. When the agent calls tool `validate_HSNcode` only the code runs the file is not loaded again

2.2 Pre-processed vs Loaded on demand

Property	Load at startup (Pre-processing)	On-demand loading
Latency	One time startup cost	Every query will incur parsing and I/O operations
Memory usage	Entire dataset and index in memory	-
Complexity	Simple	Require chunking can become complex
Scalability	Can handle many codes if server has enough ram.	-

IMO, Since this is HSN lookup agent where the file is relatively static pre-processing is ideal if there are hundreds to thousands of queries and server has enough ram. There will be a slow startup in exchange for fast per query response.

3. Validation Logic

3.1 Format Validation

The format of the user provided string is checked, the length is compared with minimum and max length, also it is checked if the code provided are digits. See [Fulfillment logic](#).

3.2 Existence Validation

Existence validation check takes $O(1)$ time complexity since it just an key check in python's `dict`.

3.3 Hierarchical validation

For standard HSN codes there is 2-digit, 4-digit, and 6-digit and full code 8-digit. for 8 digit code like 01011010, its parents can found by:

```
code = "01011010"
parents = [ code[:L] for L in LEVEL_LENGTHS ]
# which gives us ["01", "0101", "010110"]
```

We can then check each parent in map:





```
missing = [p for p in parents if p not in _hsn_map]
```

missing will be a boolean value. if true we can tell that parent levels are missing, if false we can tell all parent levels are valid.



4. Agent Response



4.1 Valid HSN code

Note: When there are multiple HSN codes provided by the user, the agent will list them out in a tabular form.

Agent	User												
	<div><div>is 01 valid?</div><div></div></div>												
<div><div></div><div>Yes, the HSN code '01' is valid. The description for it is 'LIVE ANIMALS'.</div></div>													
	<div><div>is 01, 02, asas valid HSN codes?</div><div></div></div>												
<div><div><div></div><div><div>Here's the validation results:</div><table><tr><th>HSN Code</th><th>Description</th><th>Status</th></tr><tr><td>01</td><td>LIVE ANIMALS</td><td>Valid</td></tr><tr><td>02</td><td>Meat and edible meat offal</td><td>Valid</td></tr><tr><td>asas</td><td></td><td>Invalid</td></tr></table></div></div></div>	HSN Code	Description	Status	01	LIVE ANIMALS	Valid	02	Meat and edible meat offal	Valid	asas		Invalid	
HSN Code	Description	Status											
01	LIVE ANIMALS	Valid											
02	Meat and edible meat offal	Valid											
asas		Invalid											

4.2 Invalid HSN code

Agent	User
	<div>asas</div>
<div> The HSN code 'asas' is not valid.</div>	

Agent	User
	<div data-bbox="889 142 1258 264"><div data-bbox="902 163 1104 243">96965942</div><div data-bbox="1151 163 1235 243"></div></div>
<div data-bbox="154 296 855 403"><div data-bbox="167 306 251 386"></div><div data-bbox="337 327 799 365">The HSN code 96965942 is not valid.</div></div>	