AI Agent for Dell Electronics Product Design Agent – Assignment Summary

1. Building the Graph with LangChain and LangGraph

The overarching objective here was to design an AI agent that behaves like a Dell salesperson and provides laptop recommendations based on customer-provided specifications.

```
from langgraph.graph import Graph
def recommend_laptop(state: Dict[str, Any]) -> Dict[str,
Any]:
    query = state['query']
    response = qa_chain({'question': query})
    return {'response': response['answer']}
graph = Graph()
graph.add_node('recommendation', recommend_laptop)
graph.set_entry_point('recommendation')
app = graph.compile()
```

LangGraph was used to structure the agent as a graph-based system. Each node handles a distinct task—in this case, recommendation logic. The architecture allows for state transitions and modular reasoning.

2. Loading Dell Laptop Webpages Using WebBaseLoader

```
urls = [
    'https://www.dell.com/.../inspiron-14-2-in-1-laptop',
    'https://www.dell.com/.../latitude-5450-laptop',
    'https://www.dell.com/.../latitude-7450-laptop',
    'https://www.dell.com/.../xps-14-laptop'
]
loader = WebBaseLoader(urls)
documents = loader.load()
```

Product specs were successfully retrieved from live Dell pages using LangChain's WebBaseLoader. This ensures the agent references current and accurate product data.

3. Storing and Retrieving Product Information

```
splitter = RecursiveCharacterTextSplitter(chunk_size=1000,
chunk_overlap=200)
splits = splitter.split_documents(documents)

embeddings = OpenAIEmbeddings()
vectorstore = Chroma.from_documents(splits, embeddings)
retriever = vectorstore.as_retriever(
    search_type='mmr',  # Use Maximum Marginal Relevance
for better diversity
    search_kwargs={
        'k': 6,  # Increase number of documents returned
        'fetch_k': 10,  # Fetch more documents for MMR
selection
        'lambda_mult': 0.7  # Balance between relevance and
diversity
    }
)
```

I used OpenAI embeddings and Chroma vector storage for fast, similarity-based document retrieval. This enables contextually relevant responses based on product features.

4. Input Prompts and Agent Responses

Prompt 1: "I want a dell computer for travel that has Intel® Core™ 7 150U"
Output: Inspiron 14 2-in-1 offers the Intel Core 7 150U with portability features ideal for travel.
Commentary: Correctly recommends a travel-friendly model that matches the spec.

Prompt 2: "I want a dell computer with Intel® Core™ Ultra 5 135U vPro® and 512 GB SSD"
Output: Latitude 5450 includes Intel® Core™ Ultra 5 135U with vPro® and storage configurations up to 512 GB SSD.
Commentary: Valid and consistent. The agent aligned both processor and storage needs.

Prompt 3: "I want a dell computer that has Intel® Core™ Ultra 7 165U vPro® and 1 TB SSD"
Output: Latitude 7450 meets those requirements and includes additional security and AI acceleration.
Commentary: R esponse is detailed and matched correctly.

Prompt 4: "I want a light weight XPS computer with Intel® Core™ Ultra 7 165U vPro® and 1 TB SSD"
Output: The XPS 14 offers the specified processor and storage, with a lightweight aluminum chassis ideal for mobility.
Commentary: Strong recommendation, though partial truncation was noticed.

5. Evaluation of Agent Performance

Consistency

- Output structure and tone remained professional and clear.
- Interpreted all prompts appropriately.

Accuracy

- Accurately identified processor and SSD configurations.
- Matched queries with correct Dell models.

Gaps Noted

- Some responses were cut mid-sentence.
- Price and warranty info inconsistently mentioned.
- Not all prompts returned alternatives when no exact match was found.

Recommendations for Improvement

1. **Standardized Template for Responses**
   Include model name, specs, price, weight, warranty, and suggested alternatives.
2. **Expand Knowledge Base**
   Store structured product specs and metadata for easier matching.
3. **Improve Error Handling**
   Prevent incomplete outputs and provide fallback responses.
4. **Add Validation Layer**
   Automatically check for missing info like pricing, storage, and processor mismatch.

GitHub repo: https://github.com/ryano0oceros/msds422-assign2-dell