

Kaggle Capstone Project Ideas – Diet & Nutrition Assistant Agents evidence

Part 1:

The screenshot shows a Google Colab interface with the following details:

- Title:** multi_agent_project_notebook.ipynb
- Code Area:** The main area contains Python code for a Flask application. It defines a `run()` function that handles JSON input and calls `run_agent()`. It also handles the main entry point for the application.
- File Explorer:** On the left, there's a file tree showing a directory structure with files like `run_demo.py`, `sample_data` containing CSV files (e.g., `california_housing_test.csv`, `mnist_test.csv`), and `project.zip`.
- Runtime:** The runtime is set to Python 3.
- Timestamp:** The timestamp in the bottom right corner is 8:11PM.

Part 2 Diet & nutrition agents:

The screenshot shows a Google Colab interface with the following details:

- Title:** Kaggle Capstone Project Ideas – Diet & Nutrition Assistant Agents_notebook.ipynb
- Code Area:** The main area contains Python code for a meal planning agent. It includes logic for generating meal plans based on user input and constraints.
- File Explorer:** On the left, there's a file tree showing a directory structure with files like `work.py`, `evaluator.py`, and `memory.py`.
- Runtime:** The runtime is set to Python 3.
- Timestamp:** The timestamp in the bottom right corner is 11:42PM.

Visuval studio using of the juypter server and py

The screenshot shows a Jupyter Notebook interface within Visual Studio Code. The top bar includes File, Edit, Selection, View, Go, Run, ..., Search, and Release Notes: 1.106.1. The left sidebar has icons for Generate, Code, Markdown, Run All, Outline, and Select Kernel. The main area contains two code cells:

```
from typing import Dict, Any

class Planner:
    def __init__(self, memory=None):
        self.memory = memory

    def plan(self, user_input: str) -> Dict[str, Any]:
        """
        Simple planner that interprets user input and creates a plan dict.
        """
        plan = {
            "intent": "generate_meal_plan" if "meal" in user_input.lower() or "diet" in user_input.lower() else "general",
            "user_input": user_input,
            "steps": ["parse_requirements", "generate_candidates", "finalize"]
        }
        return plan

... Overwriting project/agents/planner.py
```



```
%writefile project/agents/worker.py
from typing import Dict, Any, List
from project.tools.tools import NutritionDB

class Worker:
    def __init__(self, memory=None):
        self.memory = memory
```

Bottom right corner: Python Cell 1 of 17