Project Plan: Brain Tumor Segmentation Pipeline

1. Project Overview & Goal

Our mission is to develop a robust Computer Vision pipeline that can automatically segment (outline) a tumor from a brain MRI scan. This is a high-impact project that demonstrates a deep understanding of multi-stage image processing, which is far more impressive than a simple classification task.

Our final goal is to create an application that:

1. Takes a brain MRI as input.
2. Processes it through a 6-module pipeline.
3. Outputs the original image with the predicted tumor region clearly highlighted.
4. Provides a quantitative score of our model's accuracy.

2. The Dataset: Figshare MRI Dataset

We will be using the publicly available dataset we found on Figshare.

- **Contents:** 3,064 T1-weighted MRI images from 233 patients.
- **Key Feature:** Crucially, it includes **ground truth tumor masks**. A mask is a perfect, pixel-by-pixel outline of the tumor, which is essential for training and, more importantly, for evaluating the accuracy of our final result.
- **Format:** The data is in .mat files, which we will read using the scipy library in Python.

3. The 6-Module Pipeline: Our Blueprint for Success

This project is perfectly suited for our six-person team structure. Each team member will own one sequential module. **The success of the entire project depends on each module owner strictly adhering to the defined Input and Output contracts.**

Module & Owner	Objective	Input from Previous Module	Output to Next Module
1. Data Handler	Load and parse the complex .mat files.	File path to a .mat file.	1. Original MRI (NumPy array)2. Ground Truth Mask (NumPy array)
2. Pre-processor	Clean the raw MRI to improve segmentation	Original MRI (NumPy array).	Cleaned MRI (NumPy array).

	accuracy.		
3. Segmenter (Core Logic)	Identify and isolate the pixels that belong to the tumor.	Cleaned MRI (NumPy array).	Predicted Tumor Mask (Binary NumPy array).
4. Post-processor	Refine the predicted mask to remove noise and errors.	Predicted Tumor Mask (NumPy array).	Final Predicted Mask (Cleaned NumPy array).
5. Analyst & Evaluator	Quantify the tumor's properties and measure our accuracy.	1. Final Predicted Mask 2. Ground Truth Mask	1. Tumor Area & Centroid 2. **Dice Score (Accuracy)**
6. UI / Visualizer	Display the results in a user-friendly interface.	1. Original MRI 2. Final Predicted Mask 3. Dice Score	A visual application for demonstration.

4. Measuring Success: The Dice Coefficient

How do we know if our model is good? We can't just say "it looks okay." We need a number.

Our primary metric for success will be the **Dice Similarity Coefficient (DSC)**.

- **What it is:** A score from 0.0 to 1.0 that measures the overlap between our predicted tumor mask and the perfect ground truth mask.
- **Our Goal:** To maximize this score. A score of **0.85 or higher** would be an excellent result for a mini-project.

The Analyst (Module 5) will be responsible for implementing the function to calculate this score.

5. Key Challenges & Our Strategy

1. **Reading .mat files:** This is not a standard image format.
   - **Strategy:** The Data Handler (Module 1) will use the scipy.io.loadmat function. This is the first technical task to solve.
2. **Getting good segmentation results:** This is the hardest part of the project.
   - **Strategy:** The Segmenter (Module 3) will start with a simple, classical algorithm like **Otsu's Thresholding**. We will test it, see the results, and then try a more complex method like **K-Means Clustering** to improve our Dice Score. This iterative approach

is key.

6. Tech Stack

- **Language:** Python
- **Core Libraries:**
  - OpenCV (for all image processing tasks)
  - NumPy (for handling image arrays)
  - SciPy (specifically for loading .mat files)
  - Streamlit (for building the UI in Module 6)

7. Immediate Next Steps

1. **Assign Owners:** Assign one team member to each of the 6 modules.
2. **Set Up Environment:** Everyone should install the required Python libraries.
3. **Module 1 Starts:** The owner of Module 1 begins work immediately on the script to load and display an MRI and its mask from a .mat file. This is our first milestone.