**Sample Submission**

**Part 1: Initial Project Ideas**

1. **Project Idea 1**: Recipe Recommender
   * **Description**: A system that recommends recipes based on ingredients the user has on hand. The user enters ingredients, and the system matches them to recipes using predefined rules.
   * **Rule-Based Approach**: The system checks for exact matches and partial matches with the ingredients required for recipes in a dataset. Missing ingredients are suggested for partial matches.
2. **Project Idea 2**: Simple Chatbot
   * **Description**: A chatbot that responds to user inputs with predefined answers. The chatbot simulates a conversation by identifying keywords and phrases in user inputs.
   * **Rule-Based Approach**: Responses are based on keywords such as “hello,” “help,” or “bye.” For example, if the user says “hello,” the system responds with “Hi there! How can I assist you?”
3. **Project Idea 3**: Travel Packing List Generator
   * **Description**: A system that generates a packing list based on the user’s destination, climate, and trip duration.
   * **Rule-Based Approach**: The system uses rules to recommend items. For example, if the destination is “beach” and the climate is “hot,” the system suggests sunscreen, swimsuits, and sunglasses.

**Chosen Idea**: Recipe Recommender  
**Justification**: I chose this project because it is practical and applicable to real-life scenarios. It allows me to work with datasets, apply conditional logic, and create a system that provides meaningful recommendations based on user input.

**Part 2: Rules/Logic for the Chosen System**

The Recipe Recommender system will follow these rules:

1. **Exact Match Rule**:
   * **IF** all ingredients in a recipe are found in the user’s ingredient list → Recommend the recipe.
2. **Partial Match Rule**:
   * **IF** 75% or more of the ingredients in a recipe match the user’s ingredient list → Recommend the recipe and suggest the missing ingredients.
3. **Common Ingredients Rule**:
   * Ingredients like salt, pepper, and water are considered optional and will not be counted as missing.
4. **No Match Rule**:
   * **IF** no recipes match → Suggest adding more ingredients for better recommendations.
5. **Low Ingredient Rule**:
   * **IF** fewer than three ingredients are provided → Notify the user and suggest adding more ingredients.

**Part 5: Reflection**

This project involved designing a practical, rule-based system to recommend recipes based on user inputs. My system uses logical conditions (e.g., exact and partial matches) to evaluate user-provided ingredients against recipes in the dataset.

**Challenges**: One challenge was handling partial matches effectively. I had to decide on a threshold (75%) that balances flexibility with accuracy. Another was ensuring common ingredients like salt and water don’t skew the results, which I solved by excluding them from the missing ingredient list.

**Comparison to Machine Learning**: Unlike machine learning models, this system relies entirely on prewritten rules rather than learning from data. While this makes it simple and transparent, it limits scalability. For example, adding more complex logic or handling ambiguous inputs (e.g., "bread" vs. "whole-grain bread") would require extensive manual rule updates, which machine learning could handle more flexibly.