

### Project Plan

- How can enhancing a chatbot with multi-modal interactions improve its ability to provide personalized recipe suggestions and answer FAQs effectively?
- To investigate the impact of multi-modal interactions on chatbot effectiveness, we will develop a web-based prototype that integrates the chatbot. The website will feature a user-friendly interface where users can interact with the chatbot via text. I will utilize existing NLP libraries for intent recognition. Additionally, we will conduct a literature review on chatbot design and user engagement to guide development and implementation. Through iterative testing and user feedback on the website, we will refine the chatbot's capabilities to enhance personalized recipe suggestions and improve FAQ responses.
- Building the chatbot and the data for the recipes are the essential elements of the project. In addition, having a website where users can interact with the chatbot and a Q&A answer option would be great.
- So far we have built a pool of resources, comprised of tutorials, blog posts and at least a dataset
- For this project, I will use the Rasa framework to build the chatbot, leveraging its natural language processing capabilities for intent recognition and dialogue management. The web interface will be developed using HTML, CSS, Javascript (with React, Bootstrap for dynamic components if we have time). We use Flask / Django to set up a backend server that interacts with your Rasa chatbot and handles API requests.

### List the resources you have found so far

- **Recipe Search Algorithms:**
  - **Ingredient Matching Logic:** [Fuzzy Matching in Python](#) – A Python package for matching words based on similarity, useful for ingredient lookups.
- **Recipe Databases & APIs**
  - [Epicurious Dataset](#)
    - Kaggle dataset with recipes by Rating and Nutrition
    - Potentially useful for the Q&A add-in
  - [Spoonacular API](#)
    - Description: Offers a wide range of features including searching recipes, meal plans, ingredient lists, and nutritional information.
    - Pricing: Limited to 150 requests per day.

- Features: Search recipes by ingredients, get recipe instructions, images, and more. Provides data on nutrition, diet, and meal plans.
- [Edamam API](#)
  - Description: Provides access to a large database of recipes and nutrition information.
  - Pricing: Limited number of API calls per month.
  - Features: Search for recipes by ingredients, dietary preferences, cuisine, and more. Analyze the nutritional content of recipes and individual ingredients.
- [TheMealDB API](#)
  - Description: A free, open-source API for recipes, allowing users to search recipes by name, ingredient, and category.
  - Free access without restrictions (community-supported).
  - Features: Search recipes by ingredients, name, or cuisine. Get recipe images, instructions, and ingredient lists. Supports meal categories and areas (e.g., Italian, Mexican).
- **Tutorials:**
  - **Machine Learning (for Recipe Recommendation):** [Recommendation System Algorithms in Python \(Kaggle\)](#) – A guide to implementing recommendation systems using collaborative filtering or content-based approaches.
  - **Frontend Integration (for Bot UI):** Flask Web Framework – A Python micro-framework for web development that is useful for creating a web interface for your chatbot.

### Design a timeline for completing the project

- Phase 2:
  - Read research articles on chatbots, NLP, and recipe recommendations.
  - Gather resources on multi-modal interactions.
  - Clean data.
- Phase 3:
  - Phase 3.1: Learning
    - + Learn Rasa framework and its features.
    - + Familiarize with Flask/Django and front-end technologies (HTML/CSS/JS).
  - Phase 3.2: Initial Design and Planning
    - + Define user personas and chatbot flows.
    - + Sketch the web interface.

- Phase 4
  - Develop the chatbot using Rasa.
  - Implement the front-end interface.
  - Set up the backend with Flask/Django.
  - Write paper.
- Phase 5
  - Phase 5.1: Integration
    - + Integrate chatbot with the web interface.
    - + Connect the database for storing user data and recipes.
  - Phase 5.2: Testing and Analysis
    - + Conduct user testing to gather feedback, analyze performance and make adjustments.
  - Phase 5.3: Final Presentation
    - + Compile documentation and prepare for presentation (create poster including user guides and highlight features of the chatbot).