Summary
Snipiddy is a web application that allows users to make informed dietary decisions by leveraging AI. We extract and analyze data from menus to provide insightful information on the contents of each ingredient and help users make a decision in a user-friendly fashion.

Description
Simply enter your dietary restrictions—such as food allergies or irritating ingredients—along with any dietary preferences, diets, or caloric goals. Then, snap a photo of your menu and let our AI-powered scanner take over. Within seconds, it identifies potential allergens and offers dietary recommendations tailored to your needs, helping you make confident, informed food choices.

We will be using NextJS for the frontend. For the backend, FastAPI in Python may be useful because it will allow us to handle API requests more easily and integrate well with machine learning models like OCR. We could consider MySQL for the database system.

Potential Creative Components

- A virtual menu that provides additional educational information on targeted ingredients.
    - We can add interactive tooltips that provide scientific information about the ingredients
    - We can use NLP to summarize ingredient benefits or risks
    - We can include visualizations, like charts, to breakdown nutritional information
- A map of nearby restaurants that fit dietary preferences
    - Contains different markers above restaurants to indicate different types of restaurants, like if the restaurant is vegan, vegetarian, gluten free, etc.
    - Markers will also update based on user reviews or trends
    - Allow toggling layers, like only showing vegan restaurants or all places
    - Restaurant suggestions based on preferences and past visits

- Filters that give the user a greater control of preferences. For example, the user can find an item that costs a certain amount but also costs less. Basically multi-layered filters.
    - Show UI friendly interactive sliders for user preferences like budget, calorie, etc
    - Machine learning techniques to predict filters the user might want applied
- A history of the places the user has visited
    - Show a timeline and maybe a map to give a visual representation of history
- A Yelp-type feature that allows user to leave reviews for certain menu items
    - Use NLP to summarize positive or negative aspects of a food item based on reviews
    - Highlight frequent keywords in reviews
    - Allow users to follow other users with similar dietary preferences for better recommendations
- Create a comprehensive database of allergens
    - Add graph based visualizations to connect allergies to common dishes, cuisines, and restaurants

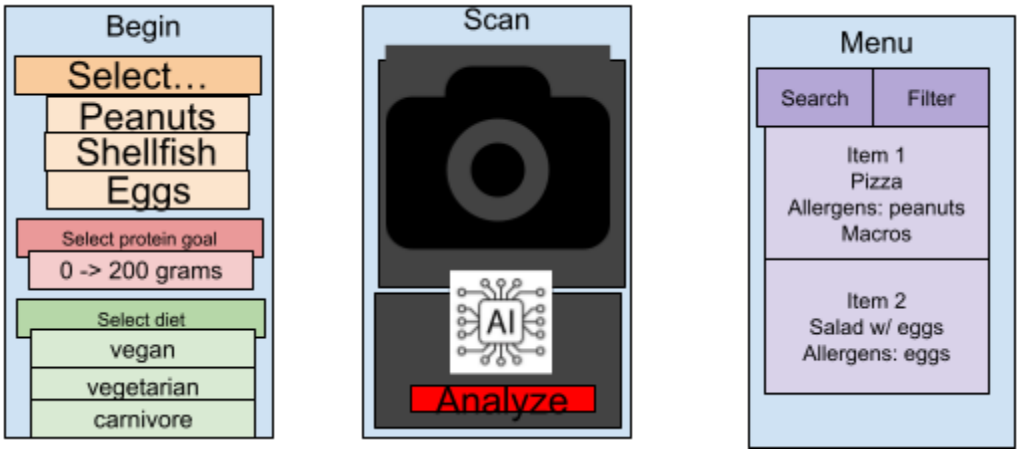
Usefulness

- The app allows users to dine-out safely by gaining a comprehensive understanding of their food options.
- The web app should be able to collect and store data from the user. The user should be able to take a picture of a menu or upload a picture. Through an API call, our app will perform optical character recognition (OCR). Then, using our local database AND the LLM (a.k.a. Augmented retrieval generation (RAG) search), we will identify and classify item contents.
- A similar app is Allergy Cam. The main difference is that it works only for allergies, but not for other food preferences, like diets and specific ingredients.

Realness

- Allergen online
    - This is allergen data from AllergenOnline in csv format. The data contains 2334 rows and 10 columns. It captures information like the species / scientific name of each allergen, the IUIS nomenclature, as well as its common name. It also contains allergenicity and length (number of amino acids), and other data that likely doesn't matter like accession number, GI number, and version.
- The University of Manchester
    - This is more allergen data from the University of Manchester. This contains a list of 84 allergens. Each allergen has a hyperlink to a page with data in text format. The data includes common and scientific names of the allergen, its occurrence (where it can be found), and other scientific information.

Mockup



Functions and Screens

On the Setup screen, we create the user account. We collect identifiable data such as name, age, email, phone number, and location. We also collect allergy information, dietary restrictions, diet, or macro goals. Some of this identifying information can also be updated by the user, such as the macro goals when they are making a dietary change.

Update functionality can be performed by the user. For example, the user updating their own allergen information, like the severity of their allergies. The user can also edit menus that they uploaded for a specific restaurant.

Delete functionality can also be performed by the user. For example, the user can remove a specific allergy from their profile if they wrote that information in by accident, or if they no longer have that allergy. Users can also remove menus from their profile if a restaurant changed their menu drastically, or they don't plan on returning to that restaurant.

On the Scan screen, the user is prompted to take a picture of a menu. We perform optical character recognition (OCR) using OpenAI-Vision to extract text from the menu's picture in a JSON format. We will collect the following data about each item: name, description, ingredients, potential allergens + confidence, potential macros + confidence, and price. We will then analyze the data, and return potential allergens cross referencing the LLM and our database.

On the Menu screen, we will show all the items scanned, as well as the results from the AI.

Project Work Distribution

1. Design how backend interacts with frontend, OCR, LLM, DB, design API routes, set up tests. User creation and authentication. - Sebastian
2. Implement API endpoints like signup, login, menu upload / process into text, create review. - Aruv
3. Choose DB system and design DB schema, integrate with API routes - Deepika
4. UI/UX, Integrate OCR with backend (setup pipeline), LLM / RAG, integrate frontend with backend - Madhumitha