PROJECT PROPOSAL

ChefAI

Team Members:

- Michael Colbert
- Omais Khan
- Sahil Jartare
- Shrey Patel
- Kevin Quiroz

GitHub Repository:

https://github.com/cis3296s25/01-ChefAI

Project Board:

https://github.com/orgs/cis3296s25/projects/58

Table of Contents

Project Proposal

Project Abstract

This document proposes a novel application of a recipe generator that creates recipes based on the ingredients that are present at your disposal. Whether you are inexperienced with cooking or want to try a new cuisine with the ingredients at your disposal, ChefAI will create the ultimate recipe for you. Additionally, you might not know what your ingredients are, or you have too many ingredients. In that case, you can take a picture of the ingredients and allow the application to generate a tasty recipe. The application will allow you to explore all cuisines and create delicious recipes in just a few clicks.

High Level Requirement

Describe the requirements – i.e., what the product does and how it does it from a user point of view – at a high level. (You can include screenshot mockup of the interface)

Conceptual Design

Recipe Cooker will be created using:

1. Backend: Python, Django
2. Frontend: HTML, CSS, JavaScript
3. Libraries:
    a. JavaScript
    b. Frameworks: Python: Django
4. APIs:
    a. Python: Clarifai, Groq.
5. Database:
    a. SQLite

We will be coding the logic of the program in Python. We will use Django framework for the backend and JavaScript for the frontend. Image detection models will be used to detect ingredients in the images that would be accessed through APIs using Python.

The User input would be converted into a dictionary of ingredient names and quantity, which would be fed to an LLM leveraged by Retrieval-Augmented Generation to produce more informed and relevant recipes.

The front end would comprise of HTML and CSS for structure and presentation, and JavaScript would be responsible for the user interaction and actions.

Proof of Concept

Here is the GitHub URL with all the project files:

https://github.com/Sahil-Jartare/Recipe-Cooker.git

Step 1 - Create a python virtual environment with version 3.8 and install openai, and other packages.

Step 2 – Run index.html in live server.

Background

The purpose of ChefAI is to generate accurate and tasty recipes based on the ingredients and their quantity. It will have an interactive UI allowing you to select the ingredients you have at your disposal or upload image/s of your ingredients along with the quantity of each ingredient.

This app is aimed to assist inexperienced cooks in making tasty food based on the ingredients they have at their disposal. There are times when you know only one recipe or have insufficient ingredients to make the recipe you know. There are also times where you want to try something new with the ingredients you have. Another vision of this app is to reduce the time of searching recipes and provide only relevant recipes with ingredients that are readily available to use at your disposal.
For times like these, all you have to do is type in the ingredients on the search bar or take a picture of all the ingredients you have and wait for the application to generate you the most optimal recipe.

This application is inspired by existing applications such as SuperCook(https://www.supercook.com/#/desktop) and DishGen(https://www.dishgen.com/), that present the user with the best recipe based on ingredients or a short description of the ingredients you would like to use.

This application will differ and improve in the following ways:

1. We will use AI to generate recipes perfect for the amount of ingredients you have.
2. We will allow the user to upload a picture of their ingredients and generate a recipe based on the ingredients.

Required Resources

- The resources required for this project are available for free. Only a laptop and IDE is required to implement the application.
- The required packages and development kit would be downloaded in their respective environments, and the group members would have access to all the files through GitHub.
- We will need to acquire knowledge on the frameworks used and learn how to integrate the various parts of the project together.
- We will have free access to the APIs for open-source models for free except for the ChatGPT API.

Project Design

Vision

FOR the inexperienced, impatient, and indecisive cooks WHO are keen on cooking food without wasting time on searching for recipes that match the availability of their ingredients, ChefAI is a Web Application powered by AI to efficiently generate the most

ideal recipe for the ingredients at your disposal THAT would take in your favored ingredients, the quantity of ingredients and the preferred cuisine UNLIKE other online free recipe providers, OUR product is powered by Artificial Intelligence to curate the perfect recipe for you at the cost of a dozen bananas.

Persona Rohan, age 20, sophomore at Dillan University

Rohan, age 20, sophomore in Dillan University studying Computer Science and Psychology. He has come for a study abroad from Mumbai, India to Philadelphia, U.S.A because he wanted to experience American culture and education. He is frugal in spending and loves to try out new food, but with his budget he cannot afford eating outside every day. Since he is a Computer Science major, he is great at using new technologies and loves to experiment with them. He has never cooked before as he never had to worry about it back home but looking at the costs of buying food in America, he is particularly keen on trying it. He does not know much about cooking and would like to search for recipes easily that provide step by step details down to the utensils that are used. Along with that he would like to try out new cuisines as he gets better. Thus, he would be interested in using ChefAI for recipes that are easy to understand and are filled with recipes of different cuisines.

Persona Michele, age 32, a housewife

Michele, age 32, is a housewife living in the suburbs of New Jersey. She does not know how to drive and relies on her husband to take her to the super-market to shop for food. Whenever she goes grocery shopping, she makes sure to buy for the whole week but sometimes over buys. She loves cooking and tries new recipes from her cookbook every day but sometimes she does not have a certain ingredient listed in the cookbook and is unable to make the recipe anymore. Then all the other ingredients that are used in the recipe are potentially wasted. Michele would like to use ChefAI that would show her a recipe that would use all the ingredients at her disposal and would provide detailed description of each step on the way.

Persona Bob, age 58, a construction worker

Bob, age 56, is a seasoned construction worker. He works 50 hours week and eats outside most of the time during the week. To relax, he makes sure to make time to cook at least once during the weekend, but since he does not cook much, he has minimal ingredients at home. He is a hard worker, but is bad with technology and very impatient. He does not like to waste time on searching recipes as most of the time he does not have all the ingredients required. He would prefer using ChefAI, as it would provide him quick, simple and tasty recipes with only ingredients that are on hand and would save the time of searching online for recipes that match the ingredients he has.

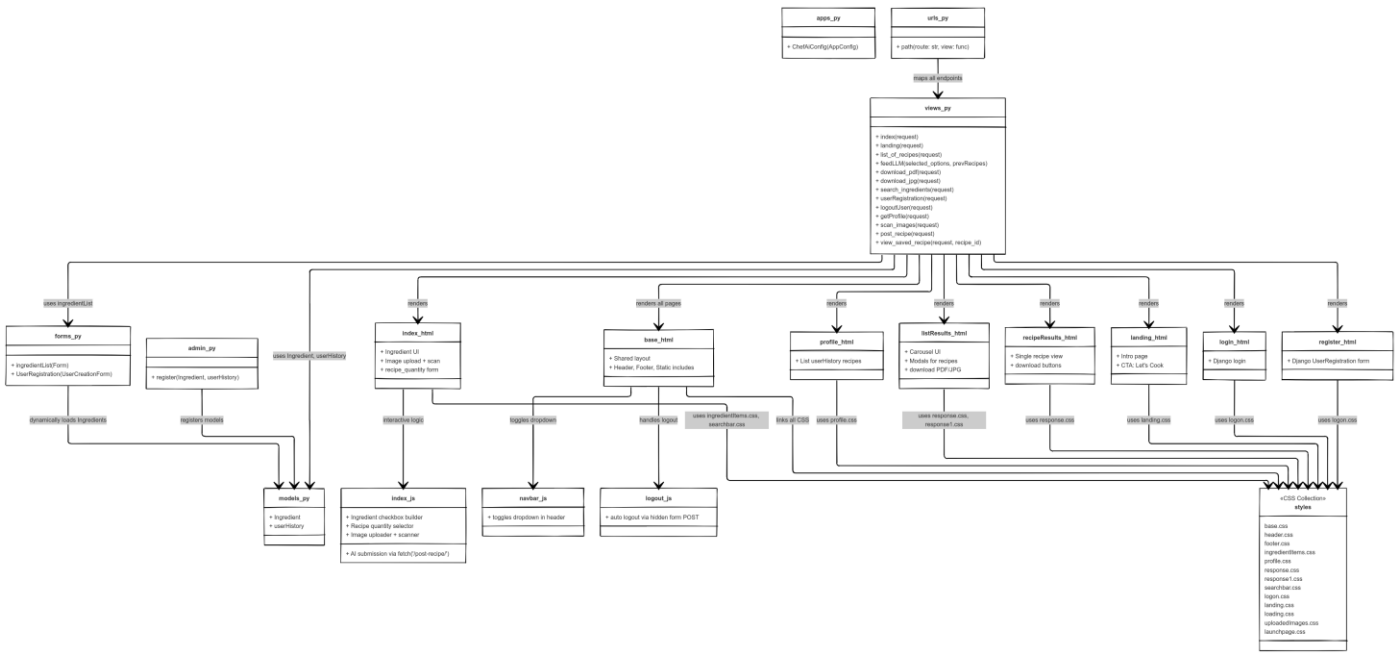
Persona George, age 82, recently widowed

George, age 82, is adjusting to life after the recent passing of his wife. He values his independence and prefers not to rely on his family for help. Cooking is new to him, and he's eager to learn how to prepare meals for himself, but he struggles with typing and navigating traditional recipe apps. ChefAI offers an ideal solution, it allows George to easily scan images of the ingredients he has at home, and then provides simple, step-by-step recipes based on what's available. The intuitive interface makes it easy for him to follow recipes without needing to type or search, helping him maintain his independence while learning to cook with confidence.

Persona Jason, age 28, junior investment banker

Jason, age 28, works as a junior investment banker in Manhattan. He's constantly drowning in 80-90 hour weeks, often stuck at the office past midnight finishing pitch decks and financial models. Even though he makes good money, he barely gets to enjoy it - his schedule is brutal. His expensive apartment might as well be a storage unit, and his fridge is a sad collection of energy drinks, an unopened bottle of champagne from his last promotion, and takeout containers in various states of decay. Growing up, his family had people who cooked for them, and in college, he just hit the dining hall. These days, he lives off expensive delivery and whatever catering shows up at the office. His doctor recently gave him grief about his cholesterol and told him to start eating better. Jason can definitely afford groceries, but he has no clue what to do in a kitchen and finds the whole thing intimidating. He's hoping ChefAI can give him some dead-simple recipes that won't take forever, don't need fancy equipment (his kitchen is basically empty), and are impossible to screw up - even for a sleep-deprived banker who's never cooked anything more complicated than toast

Class Diagram



• UML shows the main controller logic resides in views.py. It handles user input, authentication, LLM calls, image uploads, and recipe rendering

• forms.py includes ingredientList for ingredient input and UserRegistration for account creation. Both are used in views.py

• models.py defines Ingredient and userHistory to store available ingredients (database) and user-generated recipes

• urls.py maps routes to view functions such as index, list_of_recipes, and post_recipe

• index.html renders the interactive ingredient UI and image upload form, styled by ingredientItems.css and searchbar.css

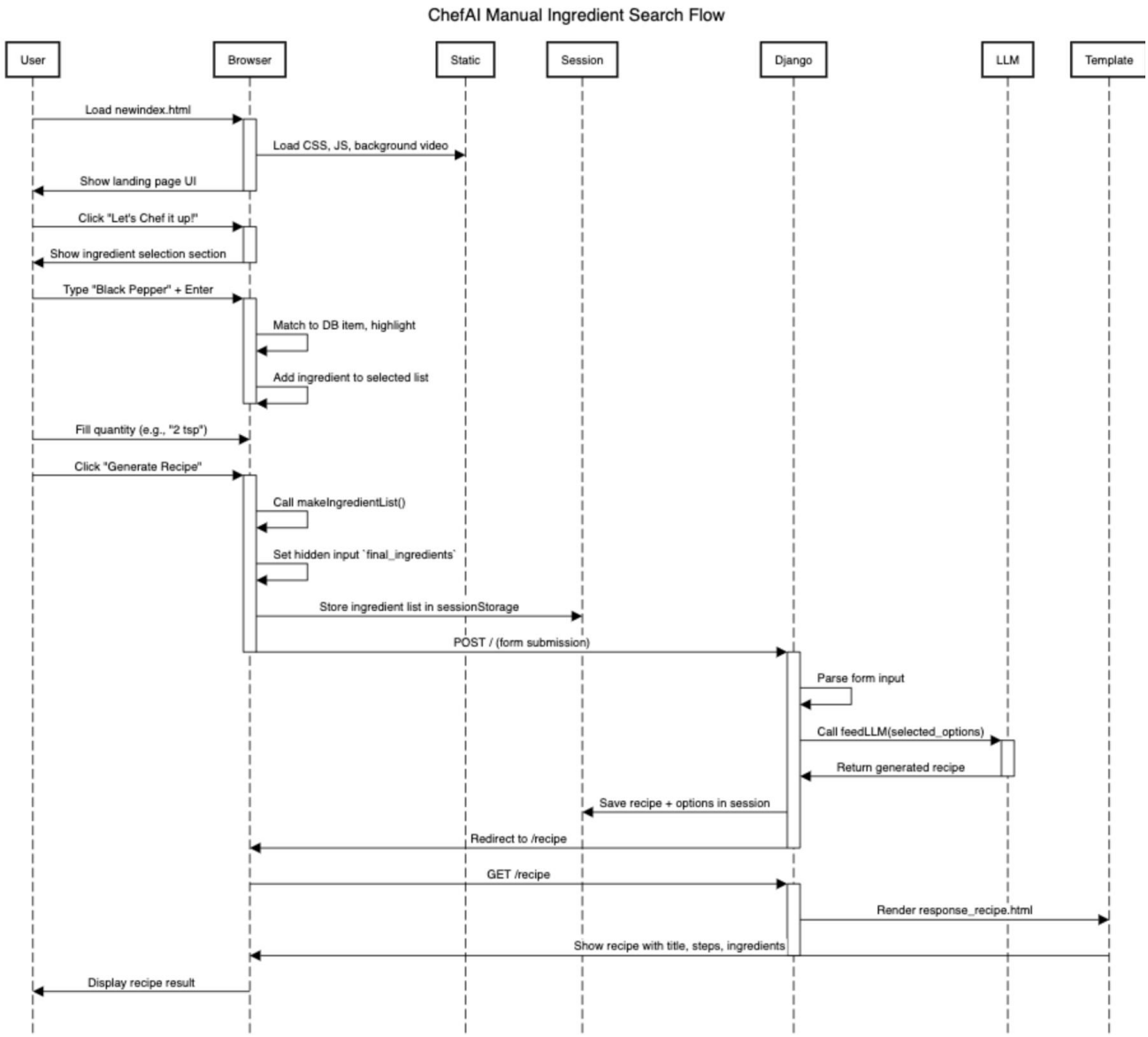
• index.js powers the frontend logic: dynamically loads ingredients, scans images, and submits to the backend with fetch()

• landing.html, listResults.html, and recipeResults.html display AI-generated recipe results with styling from response.css, response1.css, and loading.css

• register.html, login.html, and profile.html handle user auth and saved recipe history. These templates connect to forms.py, views.py, and styles like profile.css and logon.css

• base.html serves as the layout foundation, loading headers, footers, static files, and scripts like navbar.js and logout.js • admin.py registers both models (Ingredient and userHistory) for admin panel access

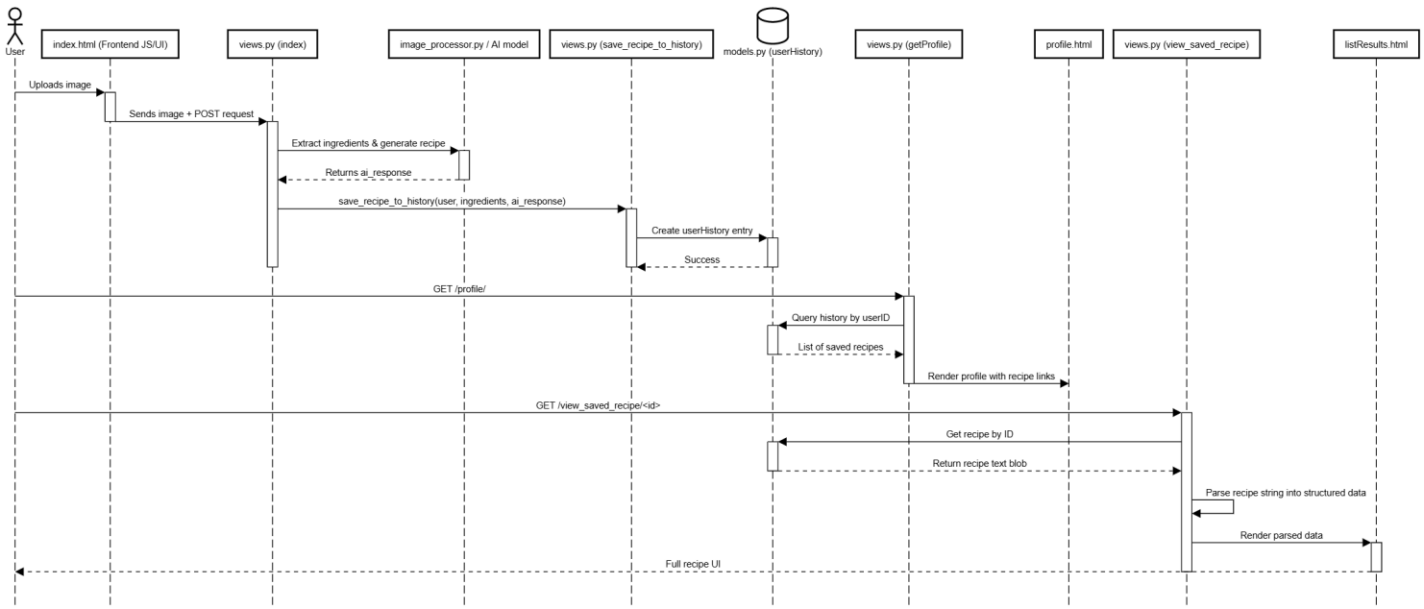
UML Sequence Diagram #1



The diagram illustrates the process from when a user visits the landing page to when a personalized recipe is displayed. Upon loading the html, the browser renders visual assets like style.css, index.js, and background video. The user initiates interaction by clicking the “Let’s Cook…” button, which transitions the UI from the landing screen to the ingredient input interface. When the user types an ingredient (e.g., “Sugar”) and presses Enter, the JavaScript script (index.js) checks for a match in the ingredient database or predefined categories. If matched, it automatically selects the item, adds it to the “Selected Ingredients” list, and allows the user to input a quantity (e.g., “2 tsp”).

As the user adds ingredients, the script compiles each name and quantity into a structured array. This array is converted to a string and assigned to a hidden form input (final_ingredients) while also being saved in sessionStorage. When the "Generate Recipe" button is clicked, the form is submitted via POST to the Django backend. The view function (views.py) retrieves the ingredients from the request, passes them to the feedLLM() function, and sends them as a prompt to the LLM (Language Model). The LLM returns a formatted recipe, which is parsed into a dictionary and stored in the session. The user is then redirected to the /recipe route, where the listResults are rendered. The final page displays the recipe title, matched ingredients, cooking time, utensils, and step-by-step instructions.

UML Sequence Diagram #2

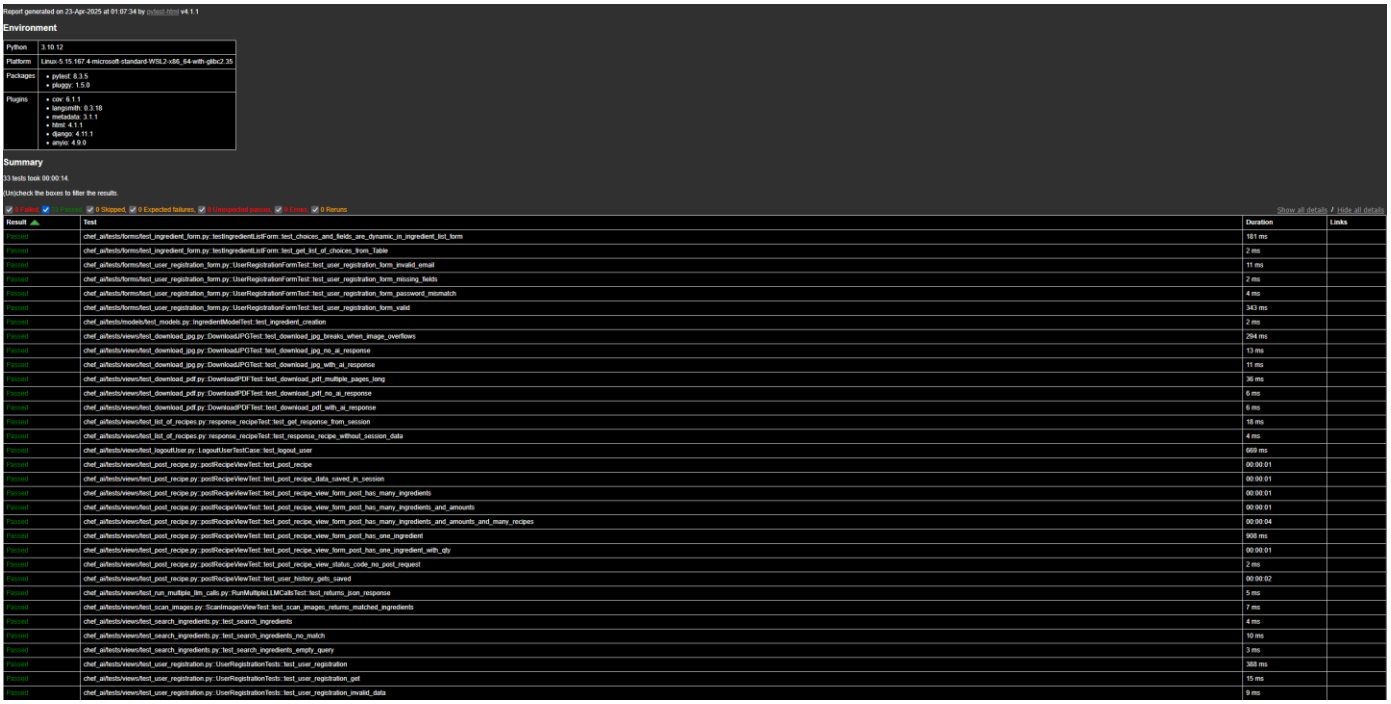


This sequence diagram shows the journey a user takes from uploading an image to viewing their saved recipes. It starts on the index.html page, where the user uploads a photo of their ingredients. That image is sent through the frontend to a Django view in views.py, which handles the image processing by passing it to an external AI model (like Hugging Face or a custom image processor). The AI returns a list of detected ingredients, which is then used to generate a recipe. This recipe, including the title, cuisine, cooking time, ingredients, utensils, and steps, is saved to the database using the save_recipe_to_history() function. Everything gets stored as a text blob in the userHistory model, tied to the current user.

When the user visits their profile.html page, the system pulls all previously saved recipes from the database for that user. These are displayed as clickable links. When the user clicks on a recipe, the view_saved_recipe function (also in views.py) retrieves the full recipe from the database and parses the saved text back into structured pieces like steps and ingredients. It then passes this data

to listResults.html, which presents the full recipe in a clean, readable layout. This flow allows the user to not only generate recipes from images but also revisit any of their past creations anytime..

Automated Test Results



Report generated on 23-Apr-2025 at 01:07:34 by pytest-html v4.1.1

Environment

Python	3.10.12
Platform	Linux-5.15.167.4-microsoft-standard-WSL2-x86_64-with-glibc2.35
Packages	• pytest: 8.3.5 • pluggy: 1.5.0
Plugins	• cov: 6.1.1 • langsmith: 0.3.18 • metadata: 3.1.1 • html: 4.1.1 • django: 4.11.1 • anyio: 4.9.0

Summary

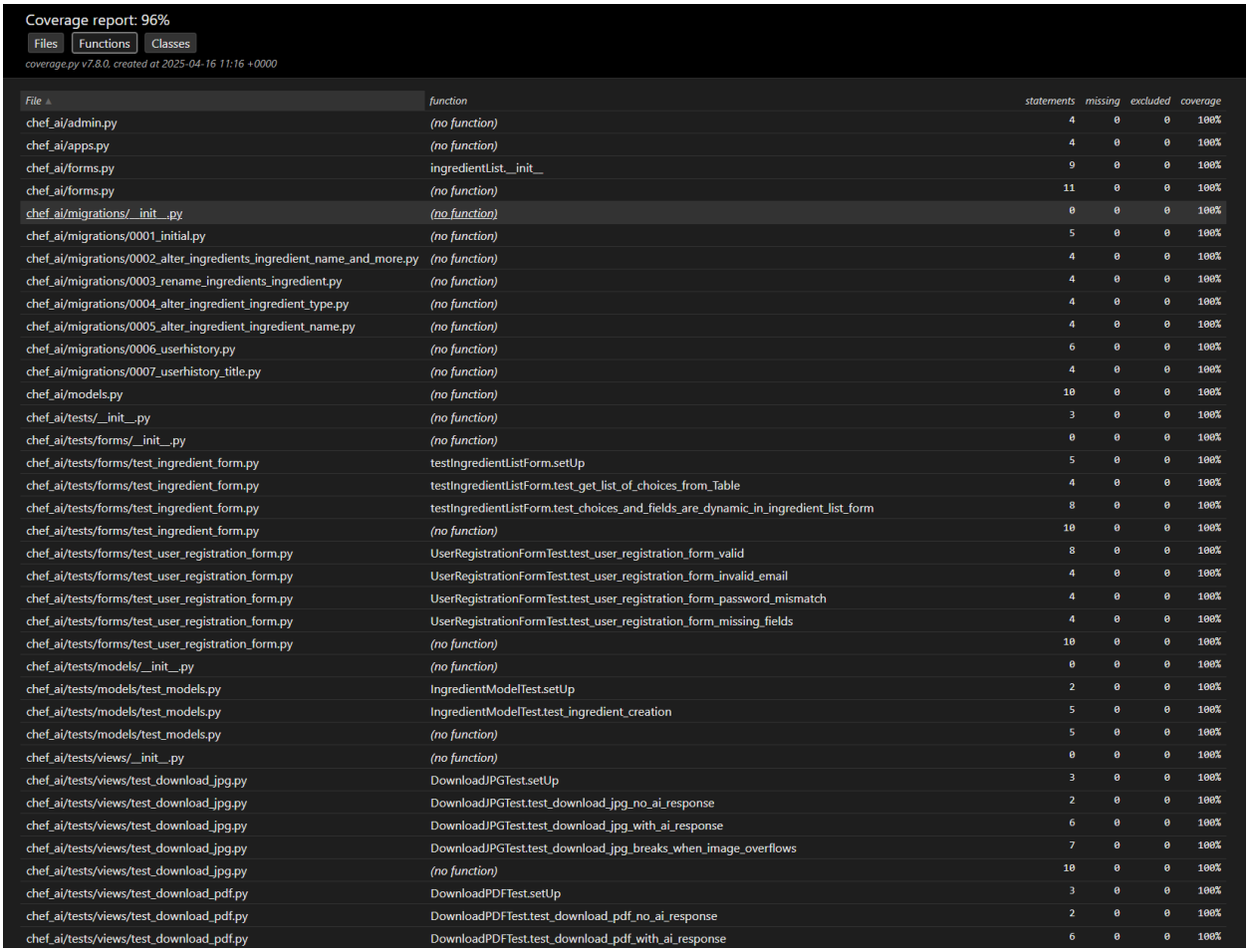
33 tests took 00:00:14.

(Un)check the boxes to filter the results.

0 Passed, 0 Skipped, 0 Expected failures, 0 Unexpected passes, 0 Errors, 0 Reruns

Show all details / Hide all details

Result	Test	Duration	Links
Passed	chef_ai/tests/forms/test_ingredient_form.py::testIngredientListForm::test_choices_and_fields_are_dynamic_in_ingredient_list_form	181 ms	
Passed	chef_ai/tests/forms/test_ingredient_form.py::testIngredientListForm::test_get_list_of_choices_from_Table	2 ms	
Passed	chef_ai/tests/forms/test_user_registration_form.py::UserRegistrationFormTest::test_user_registration_form_invalid_email	11 ms	
Passed	chef_ai/tests/forms/test_user_registration_form.py::UserRegistrationFormTest::test_user_registration_form_missing_fields	2 ms	
Passed	chef_ai/tests/forms/test_user_registration_form.py::UserRegistrationFormTest::test_user_registration_form_password_mismatch	4 ms	
Passed	chef_ai/tests/forms/test_user_registration_form.py::UserRegistrationFormTest::test_user_registration_form_valid	343 ms	
Passed	chef_ai/tests/models/test_models.py::IngredientModelTest::test_ingredient_creation	2 ms	
Passed	chef_ai/tests/views/test_download_jpg.py::DownloadJPGTest::test_download_jpg_breaks_when_image_overflows	294 ms	
Passed	chef_ai/tests/views/test_download_jpg.py::DownloadJPGTest::test_download_jpg_no_ai_response	13 ms	
Passed	chef_ai/tests/views/test_download_jpg.py::DownloadJPGTest::test_download_jpg_with_ai_response	11 ms	
Passed	chef_ai/tests/views/test_download_pdf.py::DownloadPDFTest::test_download_pdf_multiple_pages_long	36 ms	
Passed	chef_ai/tests/views/test_download_pdf.py::DownloadPDFTest::test_download_pdf_no_ai_response	6 ms	
Passed	chef_ai/tests/views/test_download_pdf.py::DownloadPDFTest::test_download_pdf_with_ai_response	6 ms	
Passed	chef_ai/tests/views/test_list_of_recipes.py::response_recipeTest::test_get_response_from_session	18 ms	
Passed	chef_ai/tests/views/test_list_of_recipes.py::response_recipeTest::test_response_recipe_without_session_data	4 ms	
Passed	chef_ai/tests/views/test_logoutUser.py::LogoutUserTestCase::test_logout_user	669 ms	
Passed	chef_ai/tests/views/test_post_recipe.py::postRecipeViewTest::test_post_recipe	00:00:01	
Passed	chef_ai/tests/views/test_post_recipe.py::postRecipeViewTest::test_post_recipe_data_saved_in_session	00:00:01	
Passed	chef_ai/tests/views/test_post_recipe.py::postRecipeViewTest::test_post_recipe_view_form_post_has_many_ingredients	00:00:01	
Passed	chef_ai/tests/views/test_post_recipe.py::postRecipeViewTest::test_post_recipe_view_form_post_has_many_ingredients_and_amounts	00:00:01	
Passed	chef_ai/tests/views/test_post_recipe.py::postRecipeViewTest::test_post_recipe_view_form_post_has_many_ingredients_and_amounts_and_many_recipes	00:00:04	
Passed	chef_ai/tests/views/test_post_recipe.py::postRecipeViewTest::test_post_recipe_view_form_post_has_one_ingredient	908 ms	
Passed	chef_ai/tests/views/test_post_recipe.py::postRecipeViewTest::test_post_recipe_view_form_post_has_one_ingredient_with_qty	00:00:01	
Passed	chef_ai/tests/views/test_post_recipe.py::postRecipeViewTest::test_post_recipe_view_status_code_no_post_request	2 ms	
Passed	chef_ai/tests/views/test_post_recipe.py::postRecipeViewTest::test_user_history_gets_saved	00:00:02	
Passed	chef_ai/tests/views/test_run_multiple_llm_calls.py::RunMultipleLLMCallsTest::test_returns_json_response	5 ms	
Passed	chef_ai/tests/views/test_scan_images.py::ScanImagesViewTest::test_scan_images_returns_matched_ingredients	7 ms	
Passed	chef_ai/tests/views/test_search_ingredients.py::test_search_ingredients	4 ms	
Passed	chef_ai/tests/views/test_search_ingredients.py::test_search_ingredients_no_match	10 ms	
Passed	chef_ai/tests/views/test_search_ingredients.py::test_search_ingredients_empty_query	3 ms	
Passed	chef_ai/tests/views/test_user_registration.py::UserRegistrationTests::test_user_registration	388 ms	
Passed	chef_ai/tests/views/test_user_registration.py::UserRegistrationTests::test_user_registration_get	15 ms	
Passed	chef_ai/tests/views/test_user_registration.py::UserRegistrationTests::test_user_registration_invalid_data	9 ms	



Coverage report: 96%

Files | Functions | Classes

coverage.py v7.8.0, created at 2025-04-16 11:16 +0000

File ▲	function	statements	missing	excluded	coverage
chef_ai/admin.py	*(no function)*	4	0	0	100%
chef_ai/apps.py	*(no function)*	4	0	0	100%
chef_ai/forms.py	ingredientList.__init__	9	0	0	100%
chef_ai/forms.py	*(no function)*	11	0	0	100%
chef_ai/migrations/__init__.py	*(no function)*	0	0	0	100%
chef_ai/migrations/0001_initial.py	*(no function)*	5	0	0	100%
chef_ai/migrations/0002_alter_ingredients_ingredient_name_and_more.py	*(no function)*	4	0	0	100%
chef_ai/migrations/0003_rename_ingredients_ingredient.py	*(no function)*	4	0	0	100%
chef_ai/migrations/0004_alter_ingredient_ingredient_type.py	*(no function)*	4	0	0	100%
chef_ai/migrations/0005_alter_ingredient_ingredient_name.py	*(no function)*	4	0	0	100%
chef_ai/migrations/0006_userhistory.py	*(no function)*	6	0	0	100%
chef_ai/migrations/0007_userhistory_title.py	*(no function)*	4	0	0	100%
chef_ai/models.py	*(no function)*	10	0	0	100%
chef_ai/tests/__init__.py	*(no function)*	3	0	0	100%
chef_ai/tests/forms/__init__.py	*(no function)*	0	0	0	100%
chef_ai/tests/forms/test_ingredient_form.py	testIngredientListForm.setUp	5	0	0	100%
chef_ai/tests/forms/test_ingredient_form.py	testIngredientListForm.test_get_list_of_choices_from_Table	4	0	0	100%
chef_ai/tests/forms/test_ingredient_form.py	testIngredientListForm.test_choices_and_fields_are_dynamic_in_ingredient_list_form	8	0	0	100%
chef_ai/tests/forms/test_ingredient_form.py	*(no function)*	10	0	0	100%
chef_ai/tests/forms/test_user_registration_form.py	UserRegistrationFormTest.test_user_registration_form_valid	8	0	0	100%
chef_ai/tests/forms/test_user_registration_form.py	UserRegistrationFormTest.test_user_registration_form_invalid_email	4	0	0	100%
chef_ai/tests/forms/test_user_registration_form.py	UserRegistrationFormTest.test_user_registration_form_password_mismatch	4	0	0	100%
chef_ai/tests/forms/test_user_registration_form.py	UserRegistrationFormTest.test_user_registration_form_missing_fields	4	0	0	100%
chef_ai/tests/forms/test_user_registration_form.py	*(no function)*	10	0	0	100%
chef_ai/tests/models/__init__.py	*(no function)*	0	0	0	100%
chef_ai/tests/models/test_models.py	IngredientModelTest.setUp	2	0	0	100%
chef_ai/tests/models/test_models.py	IngredientModelTest.test_ingredient_creation	5	0	0	100%
chef_ai/tests/models/test_models.py	*(no function)*	5	0	0	100%
chef_ai/tests/views/__init__.py	*(no function)*	0	0	0	100%
chef_ai/tests/views/test_download_jpg.py	DownloadJPGTest.setUp	3	0	0	100%
chef_ai/tests/views/test_download_jpg.py	DownloadJPGTest.test_download_jpg_no_ai_response	2	0	0	100%
chef_ai/tests/views/test_download_jpg.py	DownloadJPGTest.test_download_jpg_with_ai_response	6	0	0	100%
chef_ai/tests/views/test_download_jpg.py	DownloadJPGTest.test_download_jpg_breaks_when_image_overflows	7	0	0	100%
chef_ai/tests/views/test_download_jpg.py	*(no function)*	10	0	0	100%
chef_ai/tests/views/test_download_pdf.py	DownloadPDFTest.setUp	3	0	0	100%
chef_ai/tests/views/test_download_pdf.py	DownloadPDFTest.test_download_pdf_no_ai_response	2	0	0	100%
chef_ai/tests/views/test_download_pdf.py	DownloadPDFTest.test_download_pdf_with_ai_response	6	0	0	100%

chef_ai/tests/views/test_download_pdf.py	DownloadPDFTest.test_download_pdf_multiple_pages_long	11	0	0	100%
chef_ai/tests/views/test_download_pdf.py	*(no function)*	11	0	0	100%
chef_ai/tests/views/test_index_view.py	*(no function)*	7	0	0	100%
chef_ai/tests/views/test_list_of_recipes.py	response_recipeTest.test_get_response_from_session	8	0	0	100%
chef_ai/tests/views/test_list_of_recipes.py	response_recipeTest.test_response_recipe_without_session_data	4	0	0	100%
chef_ai/tests/views/test_list_of_recipes.py	*(no function)*	5	0	0	100%
chef_ai/tests/views/test_logoutUser.py	LogoutUserTestCase.setUp	1	0	0	100%
chef_ai/tests/views/test_logoutUser.py	LogoutUserTestCase.test_logout_user	7	0	0	100%
chef_ai/tests/views/test_logoutUser.py	*(no function)*	6	0	0	100%
chef_ai/tests/views/test_post_recipe.py	postRecipeViewTest.test_post_recipe_view_status_code_no_post_request	2	0	0	100%
chef_ai/tests/views/test_post_recipe.py	postRecipeViewTest.test_post_recipe	8	0	0	100%
chef_ai/tests/views/test_post_recipe.py	postRecipeViewTest.test_post_recipe_view_form_post_has_one_ingredient	8	0	0	100%
chef_ai/tests/views/test_post_recipe.py	postRecipeViewTest.test_post_recipe_view_form_post_has_one_ingredient_with_qty	8	0	0	100%
chef_ai/tests/views/test_post_recipe.py	postRecipeViewTest.test_post_recipe_view_form_post_has_many_ingredients	8	0	0	100%
chef_ai/tests/views/test_post_recipe.py	postRecipeViewTest.test_post_recipe_view_form_post_has_many_ingredients_and_amounts	8	0	0	100%
chef_ai/tests/views/test_post_recipe.py	postRecipeViewTest.test_post_recipe_view_form_post_has_many_ingredients_and_amounts_and_many_recipes	8	0	0	100%
chef_ai/tests/views/test_post_recipe.py	postRecipeViewTest.test_post_recipe_data_saved_in_session	7	0	0	100%
chef_ai/tests/views/test_post_recipe.py	postRecipeViewTest.test_user_history_gets_saved	12	0	0	100%
chef_ai/tests/views/test_post_recipe.py	*(no function)*	18	0	0	100%
chef_ai/tests/views/test_run_multiple_llm_calls.py	RunMultipleLLMCallsTest.test_returns_json_response	14	0	0	100%
chef_ai/tests/views/test_run_multiple_llm_calls.py	*(no function)*	11	0	0	100%
chef_ai/tests/views/test_scan_images.py	ScanImagesViewTest.test_scan_images_returns_matched_ingredients	20	0	0	100%
chef_ai/tests/views/test_scan_images.py	*(no function)*	10	0	0	100%
chef_ai/tests/views/test_search_ingredients.py	test_search_ingredients	10	0	0	100%
chef_ai/tests/views/test_search_ingredients.py	test_search_ingredients_no_match	9	0	0	100%
chef_ai/tests/views/test_search_ingredients.py	test_search_ingredients_empty_query	10	0	0	100%
chef_ai/tests/views/test_search_ingredients.py	*(no function)*	10	0	0	100%
chef_ai/tests/views/test_user_registration.py	UserRegistrationTests.test_user_registration	7	0	0	100%
chef_ai/tests/views/test_user_registration.py	UserRegistrationTests.test_user_registration_invalid_data	4	0	0	100%
chef_ai/tests/views/test_user_registration.py	UserRegistrationTests.test_user_registration_get	4	0	0	100%
chef_ai/tests/views/test_user_registration.py	*(no function)*	11	0	0	100%
chef_ai/urls.py	*(no function)*	3	0	0	100%
chef_ai/views.py	landing	1	0	0	100%
chef_ai/views.py	index	3	0	0	100%
chef_ai/views.py	list_of_recipes	4	0	0	100%
chef_ai/views.py	run_multiple_llm_calls	3	0	0	100%
chef_ai/views.py	feedLLM	14	3	0	79%
chef_ai/views.py	feedLLM.clean_section_list	1	0	0	100%
chef_ai/views.py	download_pdf	56	0	0	100%
chef_ai/views.py	download_jpg	37	1	0	97%
chef_ai/views.py	download_jpg.draw_wrapped	6	0	0	100%
chef_ai/views.py	search_ingredients	5	0	0	100%
chef_ai/views.py	userRegistration	8	0	0	100%
chef_ai/views.py	logoutUser	2	2	0	0%
chef_ai/views.py	getProfile	1	1	0	0%
chef_ai/views.py	save_recipe_to_history	2	0	0	100%
chef_ai/views.py	scan_images	18	0	0	100%
chef_ai/views.py	view_saved_recipe	15	15	0	0%
chef_ai/views.py	getProfile	4	4	0	0%
chef_ai/views.py	post_recipe	18	2	0	89%
chef_ai/views.py	*(no function)*	46	0	0	100%
Total		**694**	**28**	**0**	**96%**

coverage.py v7.8.0, created at 2025-04-16 11:16 +0000

How To Run:

1) Ensure Python3 is installed on the development computer
2) Obtain API key from GroqCloud: https://console.groq.com/keys
3) Obtain your Image Detection Model API Key: https://clarifai.com/clarifai/main/models/food-item-recognition
4) Fork repository
5) Git clone repository
6) Create ".env" file in the project root.
   a) Inside the file, create a variable:
      - `GROQ_API_KEY = "YOUR_API_KEY"`
      - `PAT = "YOUR_IMAGE_MODEL_KEY"`
7) Create a virtual environment using the command:
   a) python -m venv myenv
8) Activate the virtual environment with the command:
   a) source myenv/bin/activate (Linux system)
   b) myenv/Scripts/activate (windows system)
9) Navigate to the folder that has the file requirements.txt and type the command:
   a) pip install -r requirements.txt
10) Now cd into the directory that has the manage.py file & run the following command:
   a) python manage.py runserver

Visit http://127.0.0.1:8000

Project Progress

Week 2 Progress

Sprint Goal: The goal was to implement "Select ingredient from predefined list for getting a recipe from the list of ingredients chosen," feature.

Backlog:

1. Feature to allow users to search and select from a dropdown list that helps them automatically fill the rest of the ingredient name
2. Feature to download recipes.
3. Input cuisine, utensils, and devices that you have such as sauté-pan, oven, and microwave.
4. Image upload features, for all your ingredients.
5. Feature to scan bar codes of images
6. Feature to have more accurate recipe
7. Link to ingredients, pans, and steps during the cooking process that when hovered over would show an image describing them.

Features Implemented:

- Implemented ingredient selection from a predefined list to retrieve recipes.
- Allow users to click "Generate Recipe" after selecting ingredients, redirecting them to the response.html page.
- Display AI-generated recipe on the response.html page, including step-by-step instructions and a list of necessary ingredients.
- Modularize the ingredient selection form.
- Set up basic environment configuration
- Deploy the website to EC2 for testing and accessibility.

Task in Sprint	Velocity	Task Status at the end of Sprint	Assigned to
Create a form in Django for users to select from a list of items	8	Completed	Michael
Populate a list of items to populate the form	5	Completed	Kevin
Obtain API key for LLM	3	Completed	Shrey
Create a template for the recipe	5	Completed	Kevin
Create a template for the checkbox	3	Completed	Kevin
Create an env file	3	Completed	Michael, Shrey, Kevin, Omais, & Sahil
Take a look at the response from LLM and suggest how we should present the response to the client (response.html)	5	Completed	Omais
Create git tag and version release	3	Completed	Kevin
Deploy to EC2	5	Completed	Michael
Add prompt to code to generate response	3	Completed	Sahil
Create optimal prompt for LLM to take in value of checked boxes and generate a recipe	3	Completed	Sahil
Set up the database	3	Completed	Michael
Create a admin account to view tables	3	Completed	Michael
UML Diagram	8	Completed	Omais
Update project report with new project progress section	3	Completed	Shrey
TOTAL	**63**	**Pred. Before Sprint**	**36**

Week 3 Progress

Sprint Goal: The goal was to implement the "Allow users to type ingredients, including quantities for items. Then, allow users to easily return to the home page to ask for a new recipe." feature.

Backlog:

1. Input cuisine, utensils, and devices that you have such as sauté-pan, oven, and microwave.
2. Image upload features, for all your ingredients.
3. Feature to scan bar codes of images
4. Feature to have a more accurate recipe
5. Link to ingredients, pans, and steps during the cooking process that when hovered over would show an image describing them.

Features Implemented:

- Implemented ingredient selection from a predefined list to retrieve recipes.
- Allow users to click "Generate Recipe" after selecting ingredients, redirecting them to the response.html page.
- Display AI-generated recipe on the response.html page, including step-by-step instructions and a list of necessary ingredients.
- Modularize the ingredient selection form.
- Set up basic environment configuration
- Deploy the website to EC2 for testing and accessibility.
- Implemented a back-to-the-home screen button as the logo.
- Implemented a feature that allows users to search and select from a drop-down menu in the search bar.
- Implemented a feature to let the users download a jpeg or pdf file of the generated recipe.

Task in Sprint	Velocity	Task Status at the end of Sprint	Assigned to
Populate the database	5	Completed	Michael
Setup JavaScript to send the user input to the backend	5	Completed	Kevin
Work on CSS for search bar	5	Completed	Shrey
Have search bar perform a db query on keyup event	8	Completed	Omais
Modify the landing page and user input UI	5	Completed	Shrey
Setup the loading page	3	Completed	Shrey
Modify the UI for response.html	5	Completed	Shrey
Upload the modification done to project in Sprint 3 to EC2	5	Completed	Michael
Option to download JPG or PDF of recipe	5	Completed	Sahil
Reengineer the LLM prompt to take in the ingredient item and quantity	1	Completed	Sahil
Fix Feature names to be client friendly	3	Completed	Sahil
Modify Omais's code to have database items become a selected ingredient	5	Completed	Michael
UML Sequence Diagram #1	3	Completed	Kevin
UML Sequence Diagram #2	3	Completed	Shrey
Add Velocity column	1	Completed	Shrey
Fix backlog section in document	1	Completed	Shrey

Fix feature lables within the document	3	Completed	Shrey
Submit to canvas	1	Completed	Shrey
TOTAL (END OF SPRINT 3)	**68**	**Pred. Before Sprint**	**44**

Week 4 Progress

Sprint Goal: The goal was to implement the "Allow users to login, logoff, and look back on recipes" feature. Additionally, allow users to upload images from which ingredients would be detected feature

Backlog:

1. Feature to have more accurate recipe
2. Feature to scan bar codes
3. Feature to generate multiple recipes

Features Implemented:

- Improved search selection accuracy
- Integrated keyboard navigation support for dropdown
- Added an image detection model
- Implemented a feature to upload an image for ingredient detection
- Added auto-detection and selection of ingredients from uploaded image
- Set up registration and profile pages for users
- Created login/logout functionality
- Implemented feature to save successfully generated recipes by user
- Allow users to view any previous recipe they have searched in their profiles page
- Improved UI readability and design

Task in Sprint	Velocity	Task Status at the end of Sprint	Assigned to
Using “enter” to select recipe + other UI bugs	8	Completed	Shrey
Any successful recipes that a user searches gets saved to the database	5	Completed	Kevin
Create a login page form	5	Completed	Michael
Create a table that saves the userId and recipe	3	Completed	Shrey
Fix bug reported by users	5	Completed	Omais
Create a log out button	5	Completed	Michael
Create a profile page for viewers to see their account details	5	Completed	Michael
Using arrow keys for navigating dropdown	3	Completed	Omais
Create registration page	3	Completed	Michael
Improve recipe template readability	5	Completed	Kevin
Search detections in database and add selected ingredients in container	5	Completed	Sahil
Pass the image user inputs into detection model, and parse the detections	3	Completed	Sahil
Write unit tests for all code written	8	Completed	Michael
Add button to add image	3	Completed	Sahil
Import Open Source to Project	1	Completed	Sahil
Within the profile page, add history section to display the generated recipes for that user	5	Completed	Sahil

Find Image Detection Model	3	Completed	Sahil
Update document with new features and backlog items along with velocity	3	Completed	Omais
Create a git tag and release	1	Completed	Michael
TOTAL (END OF SPRINT 4)	**79**	**Pred. Before Sprint**	**68**

Week 5 Progress

Sprint Goal: The goal was to "Fix bugs reported by users, allow users to manage images uploaded, generate multiple recipes, and add a contact us page." Additionally, we wanted to modify some UI/UX details.

Backlog:

1. Feature to have more accurate recipe
2. Feature to scan bar codes
3. Feature to allow ingredients to be marked as important so that the recipe generated uses that ingredient.
4. Feature to allow users to upload images with text instead of having to type it themselves.

Features Implemented:

- Improved search selection accuracy
- Integrated keyboard navigation support for dropdown
- Added an image detection model
- Implemented a feature to upload an image for ingredient detection
- Added auto-detection and selection of ingredients from uploaded image
- Set up registration and profile pages for users
- Created login/logout functionality
- Implemented feature to save successfully generated recipes by user
- Allow users to view any previous recipe they have searched in their profiles page
- Improved UI readability and design

- Feature to generate multiple recipes
- Feature to add or remove ingredients after recipe generated
- Feature for user to manage uploaded images

Task in Sprint	Velocity	Task Status at the end of Sprint	Assigned to
Fix the logo margin	1	Completed	Michael
Create a button that has a hyperlink that takes the user back to the session where they selected the ingredients for the recipe	5	Completed	Michael
Allow the ability for users to choose if they want their cookies saved	5	Completed	Michael
Add a prompt for user to ask how many recipes to generate	3	Completed	Michael
Create an intermediate page to display cards for each recipe generated	5	Completed	Shrey
Remove the landing page video and make the landing page into its own page	3	Completed	Shrey
Fix background image so its covering whole background even when scrolling	3	Completed	Shrey
Ensure CSS code doesn't mess with other sections of the code	5	Completed	Shrey
Replace the LLM model back to the llama one	1	Completed	Shrey
Add a loading screen	5	Completed	Shrey
Remake FeedLLM so that it will have a number of recipes	5	Completed	Michael

parameter that the LLM will be instructed to make			
Fix pdf and jpg download, text going beyond boundaries	3	Completed	Sahil
Right now the multiple recipe cards is showing the same recipe, so we need to re-engineer the prompt or look into why it's showing the same recipe multiple times	5	Completed	Michael
Have a list of filenames they uploaded	3	Completed	Omais
Add add/delete button for files uploaded	5	Completed	Omais
Implement selective scanning	5	Completed	Omais
Fix the download button functionality. It's not working..might be a js issue	5	Completed	Sahil
Use an icon library or find icons for INstagram, X(twitter), and Facebook	3	Completed	Kevin
Make Instagram, Twitter, and Facebook accounts for our social media page	3	Completed	Kevin
Create a footer to appear at the bottom of the webpage, this should include the copyright icon and name of our product. Also make sure this is in base.html	5	Completed	Kevin
	78	**Pred. Before Sprint**	**66**

Week 6 Progress

Sprint Goal: The goal of this sprint is to have all documents revised to their final product, including the readme. Also make any necessary bug fixes to make the product a shippable product.

Backlog:

1. Feature to have more accurate recipe
2. Feature to scan bar codes
3. Feature to allow ingredients to be marked as important so that the recipe generated uses that ingredient.
4. Feature to allow users to upload images with text instead of having to type it themselves.

Features Implemented:

- Bug Fixes
- UI changes

Task in Sprint	Velocity	Task Status at the end of Sprint	Assigned to
Update UML Class diagram	3	Completed	Omais
Whats Next Mockup Slide	3	Completed	Michael
Whats Next Slide	1	Completed	Michael
ChefAI Demo Slide	3	Completed	Michael
Individual Slides-Sahil	3	Completed	Sahil
Individual Slides-Omais	3	Completed	Omais
Individual Slides-Shrey	3	Completed	Shrey
Individual Slides-Michael	3	Completed	Michael
Individual Slides - Kevin	3	Completed	Kevin
Team Scheduling Slide	1	Completed	Michael
How we brought ChefAI to life	1	Completed	Shrey
Personas/Goals Slide	1	Completed	Michael

Small bug fix rearding the divs changing color when the user returns to ingredient page	3	Completed	Michael
Update UML Sequence diagram #1	3	Completed	Kevin
Update UML Sequence diagram #2	3	Completed	Shrey
Remove gradient blur from the header and make it transparent blur	1	Completed	Shrey
Make history section into a scrollable div so that it doesn't run past the background image	3	Completed	Shrey
Fix the footer so that it appears at the bottom of the page	3	Completed	Shrey
Make git tag and release	1	Completed	Michael
Delploy to EC2	3	Completed	Michael
TOTAL (END OF SPRINT 6)	**51**	**Pred. Before Sprint**	**41**

Remaining Backlog

Feature	Velocity	Task Status at the end of Sprint	Assigned to
Feature to have more accurate recipe	13	Not Started	N/A
Feature to scan bar codes of images	19	Not Started	N/A
Feature to allow ingredients to be marked as important so that the recipe generated uses that ingredient	20	Not Started	N/A

Feature to allow users to upload images with text instead of having to type it themselves	21	Not Started	N/A