CSS 497 Design Specification

**Created by:** Ayush Mehta

**Quarter:** Summer 2025

Part 1▾Basic Info

Bringing Recipes to life: Developing an AI Powered Cooking Assistant Web App

The AI Powered Cooking Assistant Web App is a mobile and web friendly application that is designed to help busy or young individuals to discover recipes based on the ingredients they have available in the fridge or pantry. This app uses image detection and AI APIs, allowing users to upload a photo of their fridge and instantly receive recipe suggestions based on the ingredients detected.

This application will benefit the following users:

* Busy individuals or families who want quick meal ideas based on what ingredients are available.
* College students and young professionals who have limited grocery budget
* Home cooks who are looking for creative recipe ideas.

Goals:

* **Goal 1**: Develop a minimal functional web application using React for frontend and flask for backend by the end of summer quarter.

Linked Competencies:

* + Software architecture – structuring both frontend and backend components for functionality and scalability.
  + Technology standards and procedures – adhering to best practices in React and Flask development.
  + Learn by doing – gaining experience through practical implementation.
* **Goal 2**: Integrate AI-Powered image recognition for detecting ingredients through images and third-party AI API for recipe suggestions.

Linked Competencies:

* + Information gathering – researching and comparing different AI APIs
  + Interface design principles – displaying results from AI processing in a user-friendly manner.
  + Technology evaluation and selection – choosing and integrating the most appropriate AI APIs
* **Goal 3**: Learn about the Google cloud database and how to connect it to flask backend.

Linked Competencies:

* + Software architecture – designing and implementing backend to support persistent storage and data flow.
  + Systematic thinking – designing a complete data pipeline from frontend to cloud
  + Database design – structuring recipe and user data for retrieval and scalability
* **Goal 4**: Incorporate project management principles, including task tracking, and milestone planning to make sure deliverables are completed on time.

Linked Competencies:

* + Project management – planning, scheduling and executing work using tools like GitHub
  + Requirements definition and analysis – refining scope based on early stakeholder input.
  + User orientation – designing with user needs and feedback in mind.

The result of my capstone will be a minimal viable product which lets users upload an image of their ingredients or enter them manually and receive tailored recipe suggestions. The backend will process images through Google’s image recognition API and store the ingredients detected. Then it would make API request to a third-party AI API to receive recipe suggestions. To demonstrate the application, I will present a live walkthrough of the app, including core features such as image upload, ingredient detection, and recipe suggestions generation. I will test the app through user testing. I will create a google form that will let users test the website and provide feedback.

This capstone will help me learn new skills and prepare to become a full-stack software engineer with a focus on AI-driven applications. Through this project, I will gain hands on experience with industry relevant technologies like React, Flask and cloud-based services like Google Vision. This project will be a great piece to add to my portfolio that shows my ability to build end-to-end solutions that solve real world problems using modern tech stacks.

Part 2▾Resouces

I will be meeting with Aashima Mehta, a Software Development Engineer 2 from Expedia.

I am planning to meet with her twice during the quarter.

* The first meeting would be to talk about the project scope, design approach and stakeholder needs.
* The second meeting would be to showcase the working prototype of my web application and get some feedback on what could be improved.

I will identify 1-2 users who care about meal planning, and cooking with what they already have. I will collect their responses through early surveys.

When I am planning to meet with stakeholders:

* The first survey would be collecting feedback on needs, features and pain points on current recipe apps.
* The second survey would be about a live demo and discussion to confirm the app solves the original user problems.

Software that I would be using is

**React**

The purpose of this framework is to help me build the frontend of my web application like custom UI. My experience level with React is beginner, and I plan to deepen my understanding through hands-on development and documentation.

**Flask**

The purpose of this full-stack web application framework is to help me build the backend logic of the application. It will handle routing, business logic, and the development of RESTful APIs that connect the frontend with the image recognition and recipe suggestion services. My experience level with this framework is intermediate.

**Google Vision API**

This API will be used to analyze uploaded images and detect visible ingredients through image labeling. My experience level with Google Vision API is beginner, and I will be learning to use it through Google’s official documentation and tutorials.

**Git/GitHub**

The purpose of Git and GitHub is to help me with version control, tracking changes, and maintaining a clean and collaborative development workflow. My experience level with this Git and GitHub is intermediate.

**Third-party API (Groq)**

I plan to use an external API such as Groq/OpenAI to generate recipe suggestions based on the ingredients detected. My experience level with these API is beginner, and I will be exploring their documentation to understand their integration and limitations.

**Visual Studio Code**

VS Code will be my primary integrated development environment (IDE) for writing, testing, and debugging code. My experience level with this IDE is intermediate.

I will be using my personal laptop for this capstone project. I have windows Dell 13th Gen Intel(R) Core (TM) i7-1360P laptop.

Part 3▾ Detailed Specifications

**Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Field Name** | **Type** | **Description** |
| Users | user\_id | String | Unique identifier |
|  | email | String | User email address |
|  | password | String | User password |
|  | favorites | Array | List of saved recipe id’s |
| Recipes | recipe\_id | String | Unique identifier |
|  | title | String | Name of the recipe |
|  | ingredients | Array | List of ingredients name |
|  | instructions | String | Step-by-step cooking instructions |

The database design above outlines the structure of my database tables for storing user and recipe information. The database will contain two tables: users and recipes. The users table will store user’s login and saved recipes. The recipes table will store recipe information such as title, ingredients and instructions. This design provides basic user functions, such as saving recipes and retrieving them later, while also allowing for scalability when new features are added in future.

**Data Flow Diagram**

A screenshot of a computer

AI-generated content may be incorrect.

The data flow diagram above shows my AI-Powered cooking assistance app's minimal viable product (MVP). It focuses on essential functionality while excluding optional features like user login and saved recipes.

The starts off by user uploading an image from the frontend, which then gets sent to the back for processing. Once the image is processed, it is passed to an external image recognition API to detect food ingredients. The API returns the list of ingredients detected from an image and is stored in the database.

Next, the backend uses this list of ingredients and pass it to another external AI API such as Groq or spoonacular for recipe suggestions. The API returns the list of suggested recipe to backend, which sends these recipes back to frontend, where they are displayed to the user in a readable format.

To make sure the application works as expected, I will test the following use cases:

Normal Use Cases

* A user uploads a clear image of items in their fridge 🡪 app detects ingredients 🡪app displays recipes using detected ingredients
* A user manually enters ingredients instead of uploading an image 🡪 app generates relevant recipes.
* A user accesses the app on different devices (desktop, phone, tablet) 🡪 app displays responsively.

Edge Cases and Invalid Inputs

* Uploading an unclear/blurry image 🡪test how gracefully the system handles low-confidence ingredient detection.
* Uploading an image with non-food items 🡪 ensures the API does not falsely interpret unrelated items as food.
* Very large or very small image file sizes 🡪 test performance
* Entering unlikely combinations of ingredients (like pickles + chocolate syrup) 🡪 test fallback logic with creative suggestions.
* Attempting to save a recipe without being logged in 🡪 test error handling or prompt to log in
* Invalid text input (like numbers, special characters) in manual entry 🡪 validate input on frontend and backend

Testing tools

* Manual testing across browsers and devices.
* Usability testing with real users

Success Metrics

* All core features (upload image, ingredient detection, recipe generation) works as expected
* Recipe suggestions are returned within 5 seconds of request
* App runs without crash during testing
* Conducting a user survey after trying the app and aim for 80% satisfaction score from users.

Part 4 ▾ Schedule

First Quarter

| **Wk** | **ACTIVITY** | **RESULT** | **EST TIME** |
| --- | --- | --- | --- |
| 1 | * Work on weekly Status Report * Review project requirements * Brainstorm design * Research about Tech stack * Complete Project Proposal | * Project goals, tools, and scope clearly defined. * Project Proposal submitted | 20 |
| 2 | * Work on weekly Status Report * Begin learning React and setup local environment * Work on design specs * Research image recognition and recipe generation API’s * Meet with instructor #1 * Meet with expert #1 * Send a survey to stakeholder * Work on Slide Show1 | * React dev environment set up correctly * Early feedback incorporated into design * Design specs draft done * Slide show draft done * Met with instructor and expert * Finished initial stakeholders survey | 20 |
| 3 | * Work on weekly Status Report * Continue learning React * Complete design spec * Meet with Librarian * Work on career assignments * Present Slide Show 1 | * Design spec finalized * Slideshow delivered * React understanding improved * Met with librarian | 20 |
| 4 | * Work on weekly Status Report * Begin learning Flask framework and do local setup * Setup project with Flask framework with React * Setup GitHub repo for version control * Try out API’s locally * Work on Slide Show 2 * Finish career assignments | * Project repo initialized * Flask and React set up done * API tested with sample data * 3 Career Assignments done | 25 |
| 5 | * Work on weekly Status Report * Meet with instructor #2 * Present Slide Show 2 * Begin prototyping front-end UI * Begin writing backend logic like basic routing and connection to frontend | * App structure in place * Working on UI wireframes * Early backend functionality working * Met with instructor | 25 |
| 6 | * Work on weekly Status Report * Work on Slide Show 3 * Continue working on applications UI and backend | * Core Ui pages and backend routes under development | 25 |
| 7 | * Work on weekly Status Report * Meet with instructor #3 * Work on weekly Status Report * Present Slide Show 3 * Work on integrating the API’s in the application * Keep working UI and make sure it is accessible on all devices | * Slide show delivered * Working on integrating APIs * Basic app flow works end-to-end * Mobile compatibility checked | 25 |
| 8 | * Work on weekly Status Report * Work on Slide Show 3 * Final code and design cleanup for the quarter * Code review with professor or expert * Work on capstone portfolio | * Quarter 1 features stable * UI refined * Backend cleaned up * Capstone portfolio work in progress | 20 |
| 9 | * Work on weekly Status Report * Meeting with expert #2 * Work on Capstone portfolio * Send last survey to stakeholder * Present Slide Show 3 and show the project to class * Bug testing and patching * Finish the capstone portfolio | * Slide show delivered * Stakeholder feedback received * Major bugs resolved * Capstone portfolio submitted * Demo presented | 20 |

Second Quarter

| **Wk** | **ACTIVITY** | **RESULT** | **EST TIME** |
| --- | --- | --- | --- |
| 1 | * Resume development after break * Review codebase and notes * Plan out full sprint roadmap * Implement image upload, ingredient detection with Google Vision API | * Sprint goals set * Image upload and detection working | 20 |
| 2 | * Integrate recipe generation using third party API * Display results with filtering like vegetarian/non-vegetarian * Meet with instructor for feedback * Meet with expert | * Full recipe generation flow functional * Filtering options available * Feedback collected | 25 |
| 3 | * Implement user login/authentication in flask * Add Save favorite recipes functionality * Conduct small round of user testing | * User accounts working * Users able to save their favorite recipes * Initial usability testing done | 25 |
| 4 | * Set up a database for login and saving recipes * Connect backend routes to database * Fix bugs from user testing | * Database connected * User data stored * Bugs resolved | 25 |
| 5 | * Improve frontend Ui styling for clean look * Add mobile optimizations and accessibility improvements | * Web App UI improved * App polished for cross-device use | 20 |
| 6 | * Conduct second round of user testing * Meet with instructor * Add edge case handling like blurry image and non-food objects * Validate inputs on frontend/backend | * App handles errors * User feedback collected | 15 |
| 7 | * Write unit tests for backend endpoints like image analysis or recipe routes | * API reliability confirmed through tests * Code quality improved | 20 |
| 8 | * Begin working on poster presentation * Add visuals like architecture diagram, data flow and screenshots | * Draft poster with visuals prepared | 15 |
| 9 | * Get the poster reviewed by professor * Make necessary changes * Conduct final round of app testing | * Final poster approved by professor * Get app ready for public demo | 15 |
| 10 | * Conduct final bug fixes and polish codebase * Rehearse final demo | * Fix any final bugs * Final version stable and code is well documented * App ready for demo | 15 |
| Finals week | * Demo the final capstone to class * Present the poster at colloquium * Submit final code | * Capstone project completed and submitted * Presented the poster at colloquium | 6 |

Part 5 ▾ Contingency Plans

I have planned out both of my quarters in Part 4 carefully to meet the 200 hour requirement but unexpected events could occur like illness or burnout which could delay the progress. To address these possibilities, I have outlined the following contingency plans:

1. Utilize the summer gap: Although, I originally planned to not work on the project during the summer break, I will use that time as a fallback buffer.
2. Scale down extra features: I have identified few extra features like implementing filtering, database to store favorite recipes and making the web app cross-device friendly that are nice to have but not core to the MVP. If I fall behind schedule, I will prioritize the minimal viable product.