

# What-Can-We-Cook?

## The Team

### Pseudocode Warriors

- Ben Oz Elhadad - UX / UI
- Darell Isaac Sam – Backend development
- David Lucero – PMO + Doc/QA
- Nicholas Smit – Backend development / Infrastructure
- Sebastian Lopez – Backend development
- Stanley Lu – Database management

## Project Description

### Overview

Our project is a web application that helps users manage their pantry and discover recipes based on available ingredients. The application enables users to enter what items they currently have, view recipes they can cook immediately, and generate a grocery list for any missing ingredients. Each recipe provides step-by-step instructions and may include an embedded video for additional guidance.

### Core Functions

1. Recipe Management: Browse and search recipes. Each recipe includes ingredients and preparation steps.
2. Instructional Support: Recipes provide clear, step-by-step written instructions. If available, a video can be embedded for visual guidance.
3. Pantry & Smart Grocery List – Each user maintains a personal pantry stored in the database. The system compares a chosen recipe to the pantry and generates a list of missing items.

### Future Features

- User Submissions & Approvals: Users may submit their own recipes, which can be added after approval.
- Servings Scaling: Automatically adjust ingredient quantities for different serving sizes.
- Tags & Filters: Support filtering by cuisine type, dietary restrictions, or cooking time.

- Favorites/Bookmarks: Users can save preferred recipes for easy access.
- Cooking Mode: Display instructions one step at a time with a button.

## Development Environment

The application will be built using modern web technologies:

- **Backend:** FastAPI (Python) will handle business logic and provide REST API endpoints.
- **Frontend:** HTML and CSS will be used to develop a clean and responsive interface. Bootstrap templates may be used to ensure a consistent and modern design.
- **Database:** SQLite will serve as the initial SQL database for storing recipes, user accounts, ingredients, and pantry data. SQLAlchemy will be used as the ORM.
- **Authentication & Security:** Basic user authentication will be implemented for recipe submissions and pantry tracking. Credentials will be securely stored, and SSL/TLS will be used for encrypted communication.
- **Collaboration Tools:** GitHub will be used for version control and project organization, while Trello will track progress and weekly milestones. Discord will be used by team members to communicate.

## Risk Management

Our team identified three primary risks:

### 1. Scope Creep

- Risk: Attempting to add extra features beyond the core functionality could overwhelm the team and delay the project.
- Impacted Areas: Schedule, Workload.
- Mitigation: Focus on the three core functions (recipes, instructions, grocery list). Additional features will be logged as “future work.”

### 2. Schedule Delays

- Risk: Tasks may take longer than expected, leading to missed deadlines.
- Impact Areas: Deliverables, Team Coordination.

- Mitigation: Use Trello and weekly milestones to track progress. Adjust timelines early if delays are detected.

### **3. Team Availability**

- Risk: Member availability may change due to personal obligations or unforeseen events.
- Impact Areas: Productivity, Task Completion.
- Mitigation: Maintain clear communication and defined responsibilities. Reassign tasks quickly if someone becomes unavailable.

#### **Ongoing Monitoring:**

Risks will be reassessed in weekly check-ins, and contingency plans will be updated throughout the semester.

## **Project Schedule**

### **Milestones**

#### **Deliverable 1 (Proposal, Plan & Risk Management)**

**Due 9/1**

Initial project proposition, feature planning, deadline establishment, risk calculation.

#### **Deliverable 2 (System Specifications & Requirements)**

**Due 9/21**

System specifications and environment requirements will be established. Infrastructure needs will be determined and any necessary tools installed and set up.

#### **Deliverable 3 (Phase I)**

**Due 11/2**

The project is expected to be almost entirely functional by this phase. Core requirements must be met and each feature listed in the project description will be present. Trello board must be significantly completed.

#### **Deliverable 4 (Phase II)**

**Due 11/16**

Test data will be increased significantly and QA processes utilized to polish the project.  
Documentation will be completed and checked for accuracy against the completed features.

## **Deliverable 5.1 (Final Presentation)**

**Due 11/16 (First Group), 11/23 (Second Group)**

Presentation materials will be completed and utilized to showcase the project.

## **Deliverable 5.2 (Phase III Final Product)**

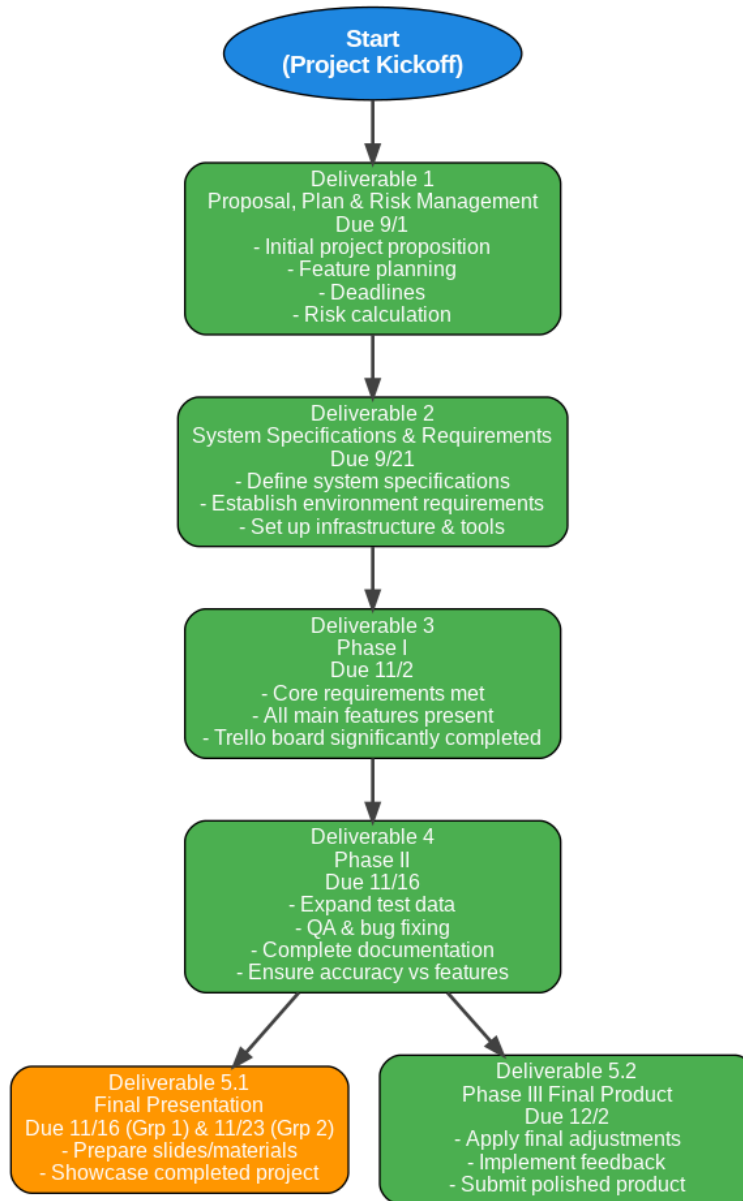
**Due 12/2**

Any final changes or adjustments will be made by this point based on incoming feedback.

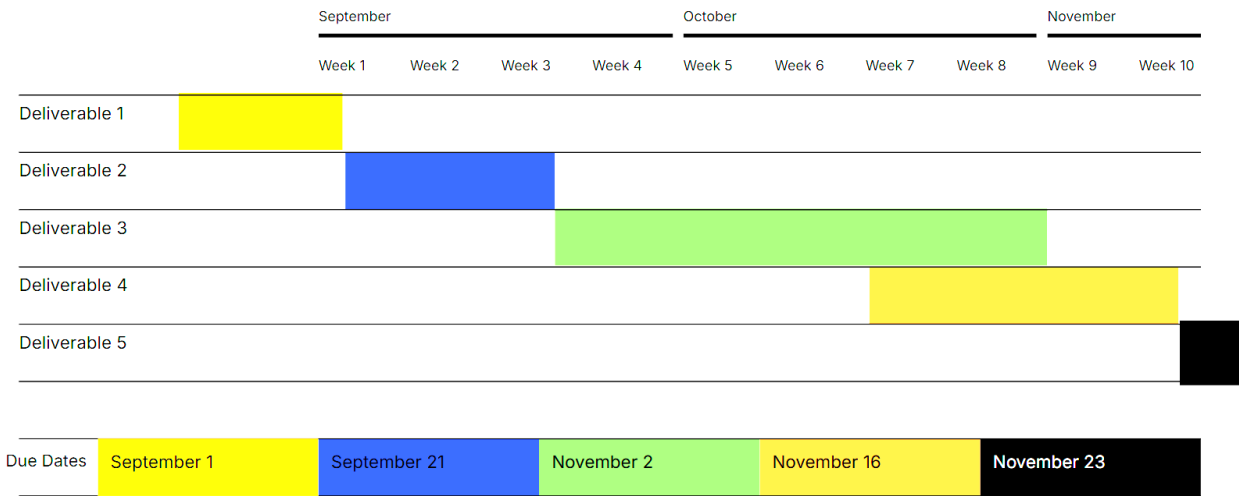
# **Team Contributions**

<b>Member name</b>	<b>Roles</b>	<b>Overall Contribution (%)</b>	<b>Notes</b>
Ben Oz Elhadad	UI/UX	16.66%	
Darell Isaac Sam	Backend Dev	16.66%	
David Lucero	PMO + Doc/QA	16.66%	
Nicholas Smit	Backend Dev / Infrastructure	16.66%	
Sebastian Lopez	Backend Dev	16.66%	
Stanley Lu	Database Management	16.66%	

# PERT Chart



# Gantt Chart



## RACI matrix project management

This is a RACI diagram. It shows which roles are responsible, accountable, consulted, or informed at each stage of the project.

	Backend	UI/UX	SQL Database	Docs/QA
Deliverable 1	C	C	C	C
Deliverable 2	R	C	R	A
Deliverable 3	A	R	R	A
Deliverable 4	R	R	I	A
Deliverable 5	R	R	R	R