Table of Contents

Contents

1. Introduction

1.1 Overview

The system is an online food ordering meal prep website that will facilitate people who have desired body goals depending on what kind of shape they are actively seeking to reach. The system will provide an opportunity for people to be able to order on a 24/7 basis and allow them to make orders for the meals they have chosen or that have been recommended.

To find the correct type of bodily goal the user should choose we have incorporated a BMI calculator to understand the user's weight and we will advise the type of body type they should go for based on their results. Users will need to provide regular check-ins to input their weight so we can aid them in visualizing their goals by using a graph to see this.
A menu section will be made available to see the range of meals available along with the ingredients and nutritional value of the meal chosen.
A recommended section will be made available for the user to suggest meals we believe would be ideal for them based on the body type they are going for when they are in one of the categories (weight loss, lean or bulk). These categories will have their own respective meals and information about the body type and the most popular meals bought section will be included.

1.2 Business Context

The website can be used by anyone who is interested in purchasing meals to help assist with their fitness goals.

1.3 Glossary

HTML – Hyper Text Markup Language, the language is used to create documents on the World Wide Web. HTML defines the structure and layout of a web document by using a variety of tags and attributes.

Django – is a high-level Python web framework that enables rapid development of secure and maintainable websites.

JS – JavaScript, the language is a scripting language that enables you to create dynamically update content. It can update HTML.

2. General Description

2.1 Product/ System Functions

User functionality system

Firstly, the user will be introduced to the homepage of the website. They will have to register to be able to order. When they are creating their account, they will be asked for their name, email address, and password. This will be done using HTML, CSS, and Django

One the home page they will be able to use our BMI calculator to see what body type plan we recommend they go for based on their results. This will be done using JavaScript and HTML.

They will pick one of the 3 categories (weight loss, lean, bulk). HTML will be required for these categories. When they pick a category, they will get to know more about the body type they are going for. They will be presented with a menu where they can view the ingredients and nutritional value of the food selected. This will be done using Django to collect the information needed to show the contents. An option to add the food items to their basket will be made possible. Within this category selected there will also be recommended food items we think the user may like and the most popular foods bought will be presented to reduce buyer indecision.

Once the user has picked out the foods, they would like to purchase they can proceed to the checkout page and process their order.

A daily weight goal check in page with graphical interface will be shown to the user. It will require the user to put in their measurements for us to track their progress over time. And they can see their previous goals and check-ins. NodeJS will be used for this.

2.2 User Characteristics and Objectives

The user will need to login or register to be able to order the meals they would like. This enables them to also be able to do check-ins, where we will be able to track the progress of that user. Otherwise, they can have a look around the website.

Guest
These are unregistered or not signed in users, they cannot order without making an account, this is to ensure we have real users and not bots.

User login
A user would have to enter their own email and password to be able to access features that would allow them to order their meals. If they enter their details correctly, they will be logged in. If they fail to get logged in a message will appear on the page telling them they have entered the wrong details and try again.

User able to order
With successful entry onto the website by logging in or registering the user can order the meals they want.

2.3 Operational Scenarios

Login
Users can login to the website.

Register
Users can register for an account on the website.

BMI calculator
Users will be able to input their measurements and we can assist in providing the right body type category to follow

Recommended system
Shows foods we think you would perhaps like

Most popular
Shows most popular meals purchased

Basket
User can view their items before proceeding to checkout

Order
User will need to put in their details to order their meal

Weight progress graph
Tracks users weight overtime to see how close they are to their goal.

2.4 Constraints

Time constraints
In this project we are going to be using a lot of technologies unfamiliar to us so we will need to allow for some time to research within our allotted time for a task. To manage our time efficiently, we will employ weekly sprints and agile methodology.

Language

We will be coding the weight progress graph using a Django chart.js this is the first time we will be using this language to create something like this but from research and viewing code we believe it will be possible to complete this operation.

Accessibility
We are going to ensure that the website is readable by screen readers, and that the page can adjust to accommodate other visual impairments.

3. Functional Requirements

3.1 Login
Description: The user needs to be able to login to be able to order meals.

Criticality: This requirement is of importance for the end product however it is not critical to the essential function of the application.

Technical issues: We must ensure user data is stored securely. For this we must ensure our database is secure.

Dependencies: The function depends on the user having registered previously.

3.2 Register
Description: For a user to log in, they must first register an account. They can do so by navigating to the user icon. Where they will see a link that will send them to the register page.

Criticality: This requirement is of importance for the end product however it is not critical to the essential function of the application.

Technical issues: We must ensure user data is stored securely. For this we must ensure our database is secure.

Dependencies: None

3.3 BMI calculator
Description: Users will need to put in their measurements to find out what category will be best fit to them to follow.

Criticality: This requirement is of moderate importance; this allows users to pick the most suitable category for them currently.

Technical issues: We must ensure user data is provided is accurate and the results reflect the result of the BMI.

Dependencies: None

3.4 Recommended System

Description: This system will bring up meals we think the user may like depending on the category chosen as a body type. It has foods we will suggest to the user.

Criticality: This requirement is of moderate importance; this allows users to have a look at possible meals offered to them that they may like.

Technical issues: We must ensure the correct meals are catered to the correct category to prevent the user from being confused or misinformed.

Dependencies: None

3.5 Most Popular

Description: This system will bring up meals that have been purchased the most.

Criticality: This requirement is of moderate importance; this allows users to have a look at popular meals picked by people in the same body type category as them.

Technical issues: We must ensure the correct meals are catered to the correct category to prevent the user from being confused or misinformed.

Dependencies: None

3.6 Basket

Description: The user can view the meals they have selected in their basket.

Criticality: This requirement is a critical part of the application. It will be one of the first things implemented along with the login and register.

Technical issues: We must ensure that the user can remove meals they don't want from their basket.

Dependencies: Will need the user to be logged in.

3.7 Order

Description: This system will bring up foods we think the user may like depending on the category chosen as a body type. It has foods we will suggest to the user.

Criticality: This requirement is of moderate importance; this allows users to have a look at their products before ordering.

Technical issues: We must ensure the correct meals selected by the user are present in this section and give them an option to remove items if they change their mind.

Dependencies: Will need the user to be logged in.

3.8 Weight Progress Graph

Description: This system displays a graph that shows the users progression with their weight and their desired goal they want to reach

Criticality: This requirement is of moderate importance; this allows users to be able to check-in about their weight and monitor their progress.

Technical issues: We must ensure the data submitted by the user is reflected accurately on the chart.

Dependencies: Will need the user to be logged in.

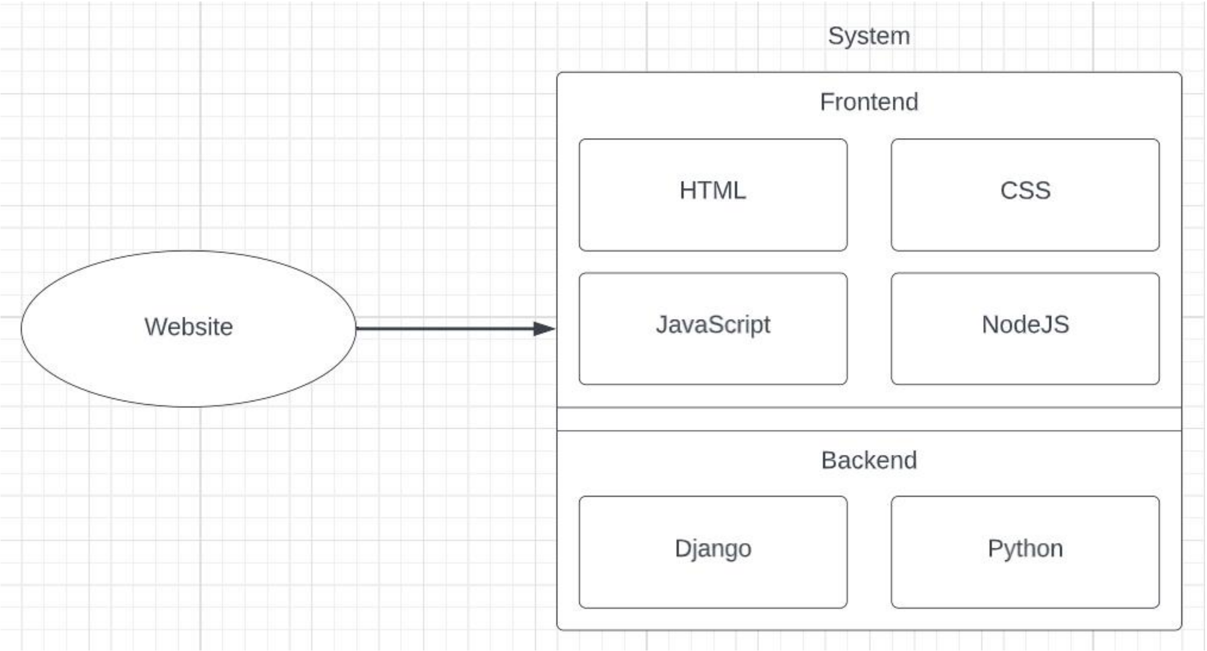
4. System Architecture

Frontend

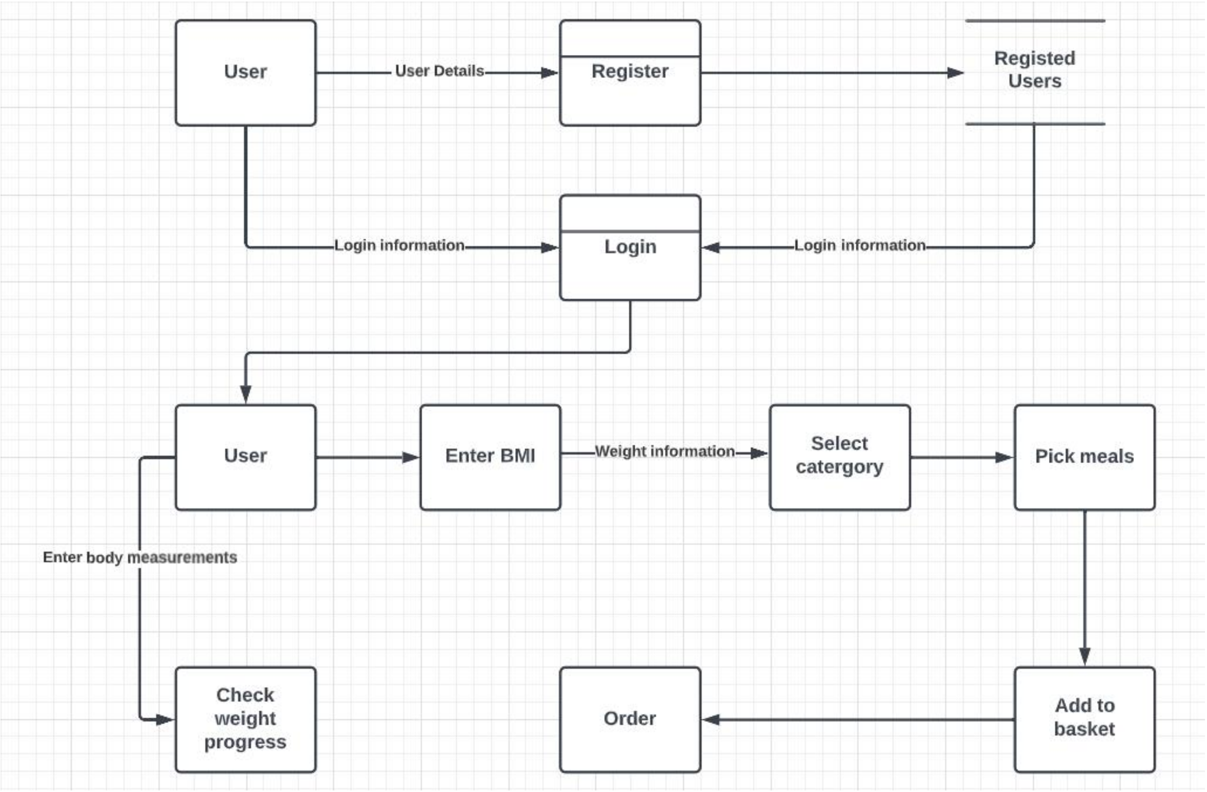
The Frontend will be using JavaScript, HTML and CSS. To present the website in an aesthetically pleasing form. We are aiming to use NodeJS as our framework in python as we have had experience using this before. Django is an option that will be considered if it's necessary or more efficient. We will use JavaScript to make an interactive and engaging website. The UI will be simple and elegant so the user will always be aware of the options that are available to them at any given moment.

Backend

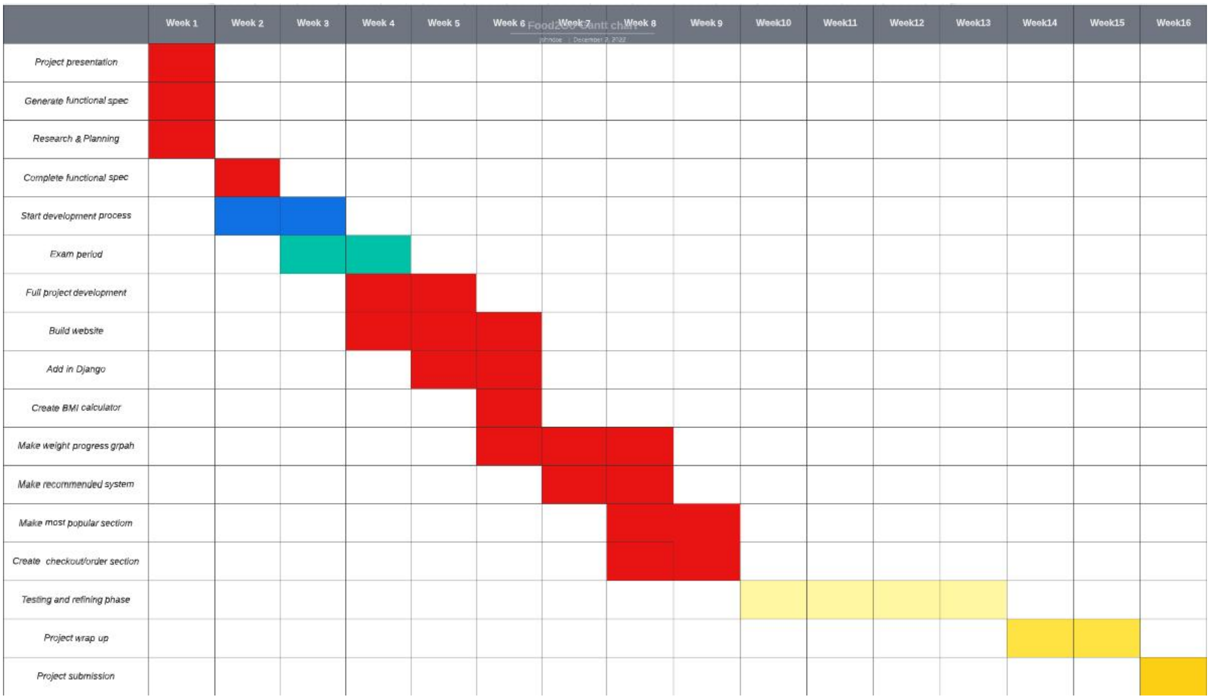
The backend database will be run using Django and Python. The function of the database will be to store any data that is related to a user. This will be their information such as their username, email address, and password. It will also be used to store nutritional information and the ingredients of the meals.



5. High Level Design



6. Preliminary Schedule



	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week10	Week11	Week12	Week13	Week14	Week15	Week16
Project presentation	🟥															
Generate functional spec	🟥															
Research & Planning	🟥															
Complete functional spec		🟥														
Start development process		🟦	🟦													
Exam period			🟩	🟩												
Full project development				🟥	🟥											
Build website				🟥	🟥	🟥										
Add in Django					🟥	🟥										
Create BMI calculator						🟥										
Make weight progress grpah						🟥	🟥	🟥								
Make recommended system							🟥	🟥								
Make most popular sectiom								🟥	🟥							
Create checkout/order section								🟥	🟥							
Testing and refining phase										🟨	🟨	🟨	🟨			
Project wrap up														🟨	🟨	
Project submission																🟨

7. Appendices

A. Django - https://www.djangoproject.com/
B. Nodejs - https://nodejs.dev/en/learn/introduction-to-nodejs/
C. Python - https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django
D. JavaScript - https://www.w3schools.com/js/
E. Weight graph - https://muschealth.org/medical-services/weight-management/resources/graph
F. BMI – calculator https://www.cdc.gov/healthyweight/assessing/bmi/index.html#:~:text=Body%20Mass%20Index%20(BMI)%20is,or%20health%20of%20an%20individual.