Software Requirements Specification

for

MyPantry Application

**Version 1.0 approved**

**Prepared by:**

**Adam Tan Jun Xiang(U2120309H)**

**Sandhiya Sukumaran(U2120682B)**

**Thomas Emmanuel Mathew(U2123204K)**

**Khoo Jia Xuan Mavis(U2121376B)**

**Ie Jia Sheng(U2122557E)**

**Nguyen Gia Khanh(U2120680J)**

**Lab Group: Z35**

**7 April 2023**

**Table of Contents**

[**1. Introduction 1**](#_1fob9te)

[1.1 Purpose 1](#_3znysh7)

1.2 Document Conventions 1

[1.3 Intended Audience and Reading Suggestions 1](#_md3qlbcr06x)

1.4 Product Scope 2

**2. Overall Description** 3

[2.1 Product Perspective 3](#_2s8eyo1)

2.2 Product Functions 3

2.3 User Classes and Characteristics 3

2.4 Operating Environment 4

2.5 Design and Implementation Constraints 4

2.6 User Documentation 4

2.7 Assumptions and Dependencies 5

**3. External Interface Requirements** 6

[3.1 User Interfaces 6](#_2jxsxqh)

3.2 Hardware Interfaces 13

3.3 Software Interfaces 13

3.4 Communications Interfaces 15

**4. System Features** 16

4.1 Login and Account creation page 16

[4.2 MyPantry 18](#_booqjqmvyxe1)

[4.3 Nearby Grocery Stores 21](#_gyh2cb1p75ek)

[4.4 My Profile Page 22](#_bk4m7jkpmnp)

[4.5 Saved Recipes Page 23](#_ljl3ery7r6hq)

[**5. Other Nonfunctional Requirements 25**](#_3whwml4)

[5.1 Performance Requirements 25](#_2bn6wsx)

5.2 Usability Requirements 25

5.3 Security Requirements 25

5.4 Reliability Requirements 26

5.5 Supportability Requirements 26

**Appendix A:** [**Data Dictionary**](#_g8qnrcg3e6et) **27**

**Appendix B: Analysis Models 27**

**Appendix C: Additional Information** [**27**](#_g8qnrcg3e6et)

# Introduction

## Purpose

The purpose of the Software Requirement Specification (SRS) document is to provide a comprehensive and in-depth description of MyPantry Application such as its functional and non-functional requirements, design constraints, purpose and features, as well as its interfaces. It serves as a tool for different parties to understand the software system requirements of the application.

## Document Conventions

Title: Arial, Font size 18, Bold

Subtitle: Arial, Font size 14, Bold

Body: Arial, Font size 12, Normal

## Intended Audience and Reading Suggestions

This document is intended for developers, testers and general users.

Developers may use this document to analyze the functionalities and features of the application through the documentation of the functional and non-functional requirements. This gives them a better understanding of how the application functions, thus allowing them to further develop and implement additional features as well as provide general maintenance for the application.

Testers may use this document to reference the previous unit testing that was already done on the application. This allows them to follow a standardised format for future testings and ensure the performance of the application is constantly held at a similar standard.

General users can use this document to get a better understanding of how the application works. This allows the users to have a more comprehensive user experience as they would better utilize the features and functionalities of the application.

## Product Scope

MyPantry is a web application that allows users to add ingredients into a virtual pantry and generate recipes from selected ingredients. This feature aims to offer convenience to users by providing fast and creative meal ideas for users who would like to utilize the readily-available ingredients in their pantry. The application will also locate and display nearby grocery stores in the user’s proximity, thereby making it convenient for users who want to shop for additional ingredients.

# Overall Description

## Product Perspective

MyPantry is a standalone web application that is developed using the ReactJS framework and is hosted and deployed on the cloud through Firebase. The web application utilizes several Application Programming Interfaces(APIs) to enable its in-app functionalities. Such APIs include:

* Google Sign In API that allows users to sign in to the application using their google account credentials.
* Google Maps API and Google Places API to locate nearby grocery marts.
* Spoonacular Recipes APIs to generate recipes from selected ingredients.
* Built-in Firestore APIs for data storage and real-time updates.
  + onSnapshot
  + getDocs
  + getFirestore
  + deleteDoc

## Product Functions

The application offers the main functionalities as follows:

* User login and registration of account.
* Adding and removing ingredients into the virtual pantry within the application.
* Generating recipes from selected ingredients.
* Saving selected recipes into a “Saved Recipes” page.
* Locating nearby grocery marts.
* Editing user profile.

## User Classes and Characteristics

The MyPantry application is targeted at two user classes:

This first user class are users who are interested in using the application to get recipe ideas that they can cook with the ingredients they have in their pantry.

The second user class are users who are interested to use the application to locate the nearby grocery marts in their proximity.

## Operating Environment

MyPantry application is designed to be used through a website browser. Users will be allowed to access its features and functionalities from any device with internet access and a compatible web browser.

The application’s frontend is built using React JS, a JavaScript library that allows developers to create reusable UI components which can be composed to form complex user interfaces that are highly interactive and user-friendly.

MyPantry utilizes Firebase for its backend system. It provides the necessary services for the application’s functionalities including user authentication, data storage and real-time database updates.

NodeJS is used to create the development environment to run the application locally. The developer must have NodeJS installed on their system. To run the application locally, follow these steps:

1. Enter the MyPantry directory.
2. Enter “npm i” in the terminal to install the relevant modules.
3. Enter “npm start” to run the project on the localhost.

## Design and Implementation Constraints

Firebase limits: -Authentication limit of 100 new account creations/ IP address/ hour

-Address verification emails of 1000 emails/ day

-Address change emails of 1000 emails/day

-Password reset email of 150 emails/day

-Firestore free quota data storage of 1GiB

Spoonacular: Daily limit of 150 points/day where calling a fetch request endpoint costs 1 point and 0.01points per recipe returned.

GooglePlaces API: 0.032USD per call of nearbySearch(), used to find the nearby stores.

## User Documentation

A simple user manual explaining the features and functionalities of the application will be accessible to the user. With the guide provided, it makes it easier for first-time users to navigate across the application and utilize the features available.

## Assumptions and Dependencies

**Assumptions**

Users should have a stable internet connection to run the application.

Users should have an electronic device that is capable of opening a web browser.

Users should have a basic understanding on how to use the device to open the web application.

**Dependencies**

1. Firebase: The application uses Firebase and built in Firestore APIs to store data and authenticate users.

1. ReactJs: MyPantry’s user interface is built on React’s component-based system. It uses the Material UI library (MUI) that provides pre-built UI components and the React Router library that handles the navigation between multiple pages in the application.

1. FortAwesome: Font Awesome is used to add icons to the UI components in the application.

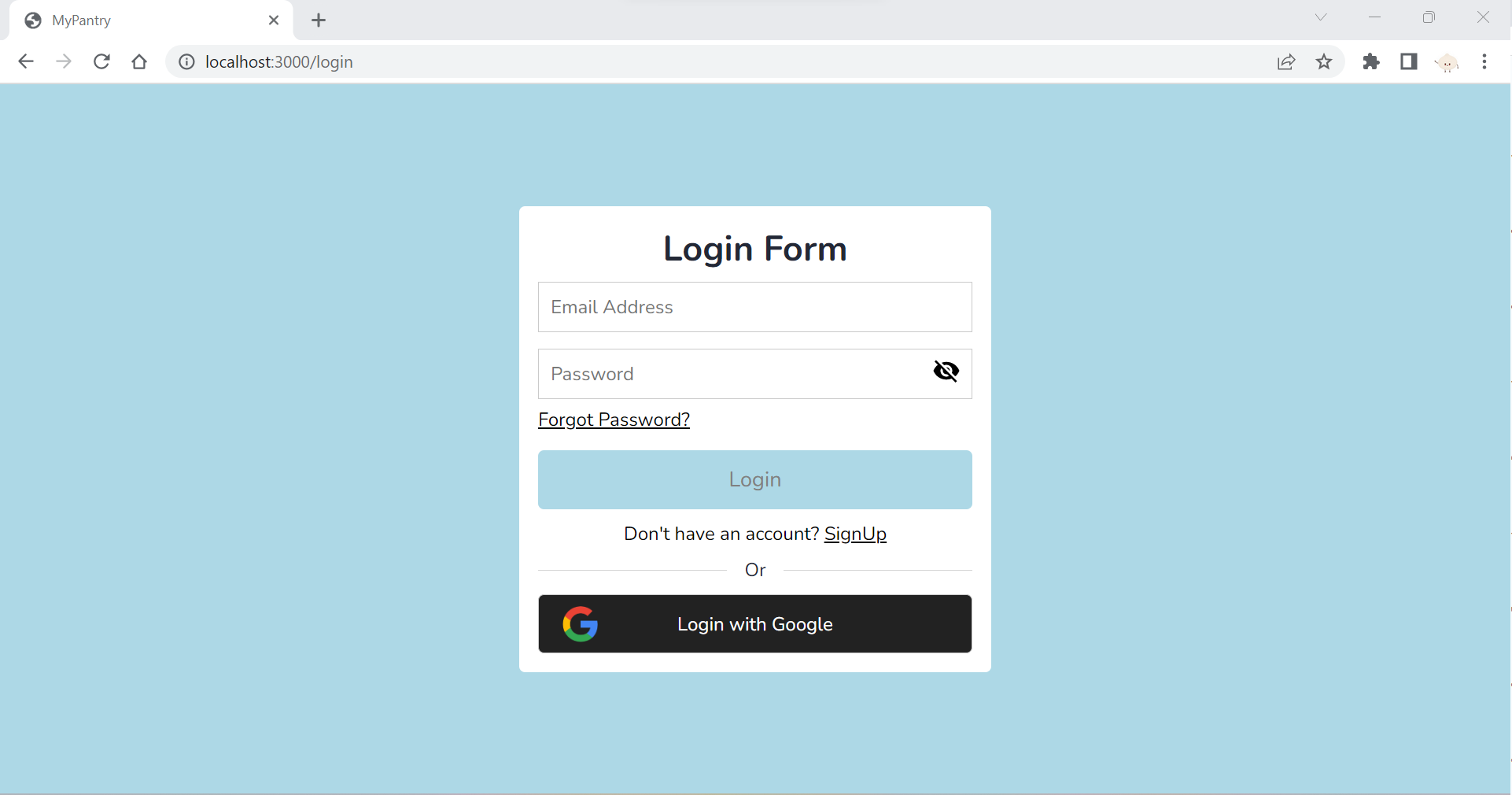
1. External APIs: APIs such as Google Sign In API, Google Maps API, Google Places API, Spoonacular Recipes APIs and built-in Firestore APIs are used in the application to implement its functional requirements.

# External Interface Requirements

## User Interfaces

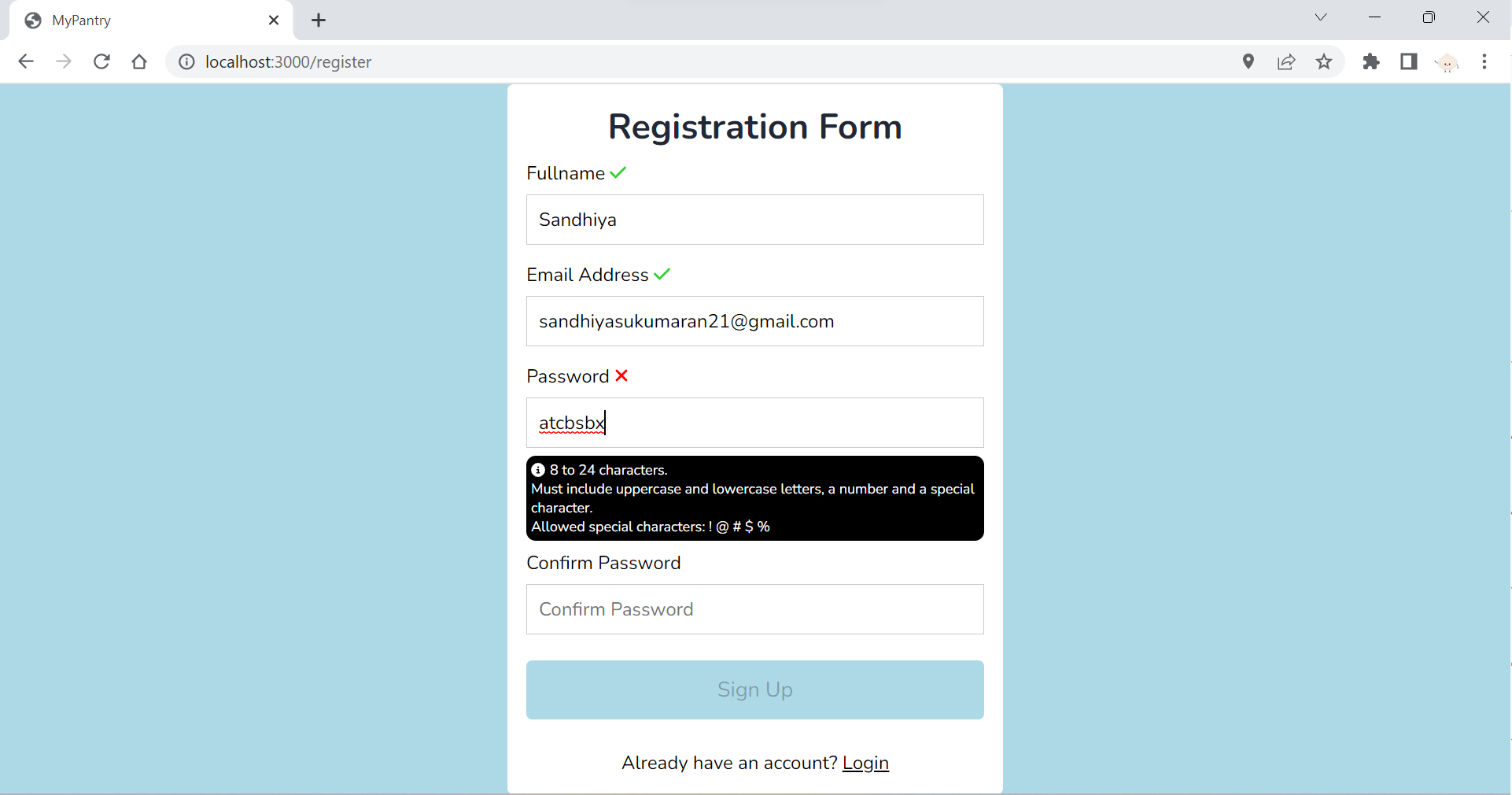
### 3.1.1. Login Page

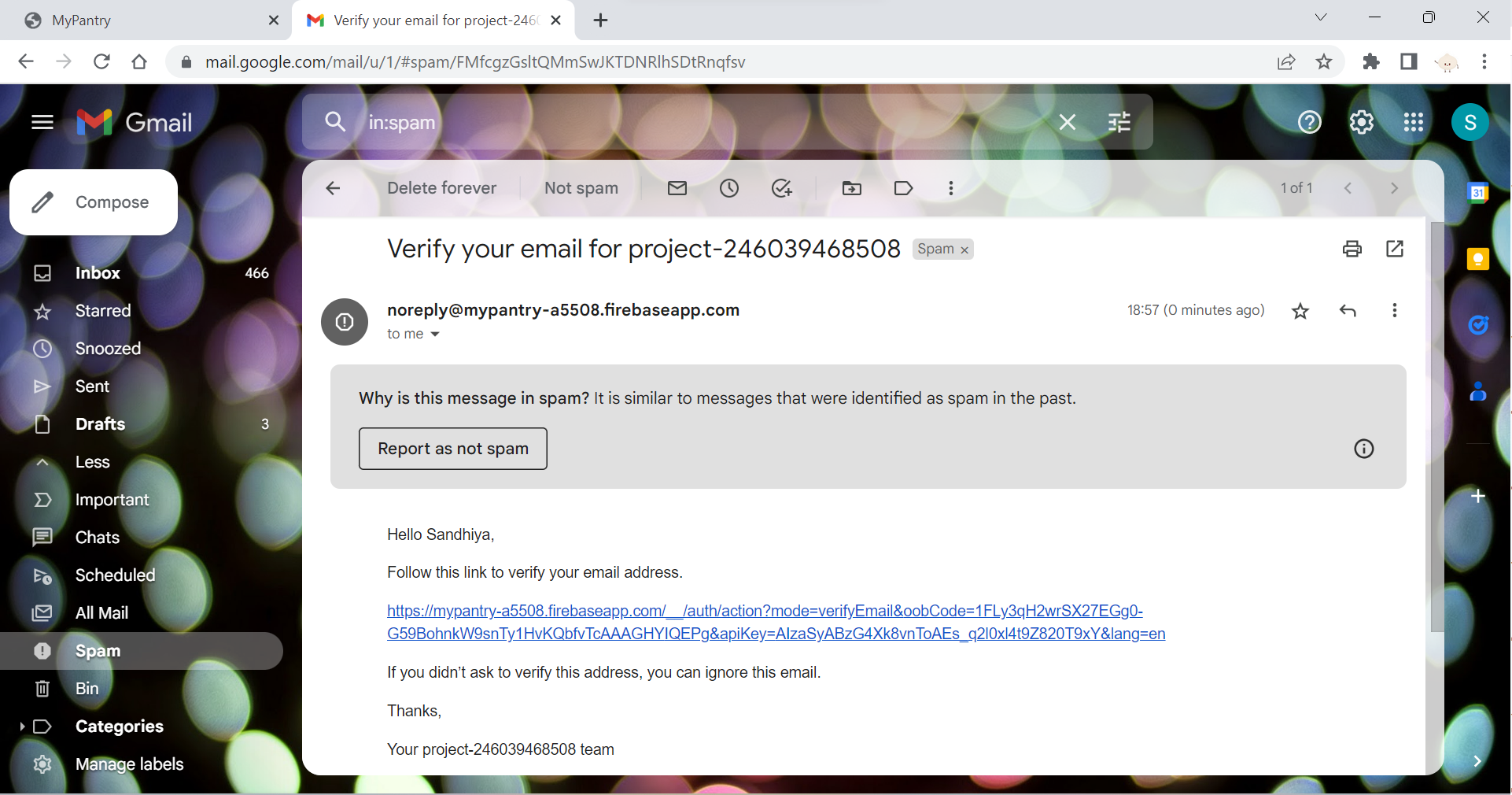
For existing users of the application, the users are given 2 options: either to log in using email address and password or to log in using their google account.

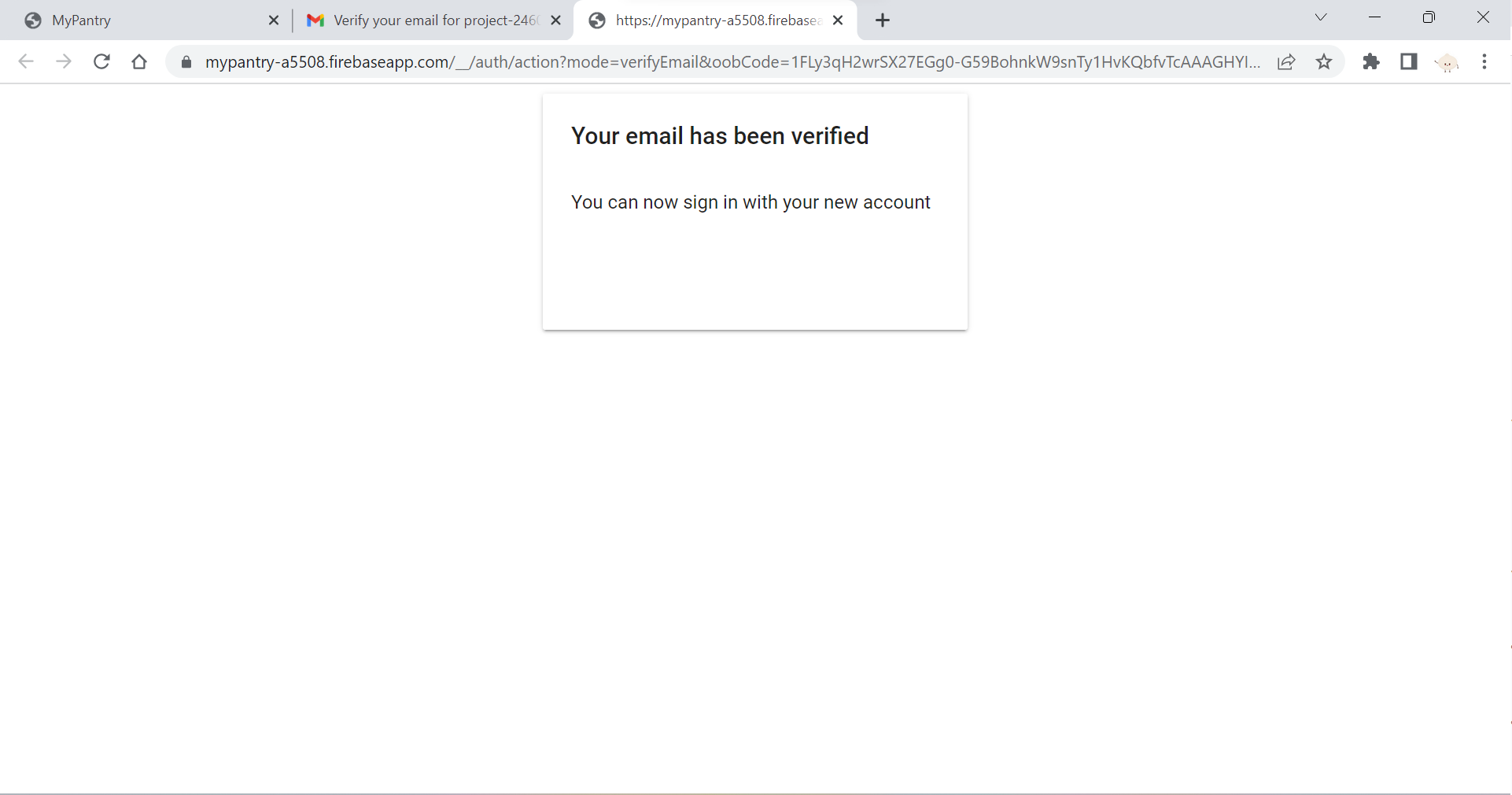


### 3.1.2. Registration Page

For new users, users will have to register for a new account by clicking on the “SignUp” button. This will prompt them to the Registration Form. Users will have to fill in the input fields such as fullname, email address and password according to its respective validation rule. Upon successful creation of an account, a verification email will be sent to the email address provided. Users will have to click on the verification link sent in order to successfully log in to the account that has been created.

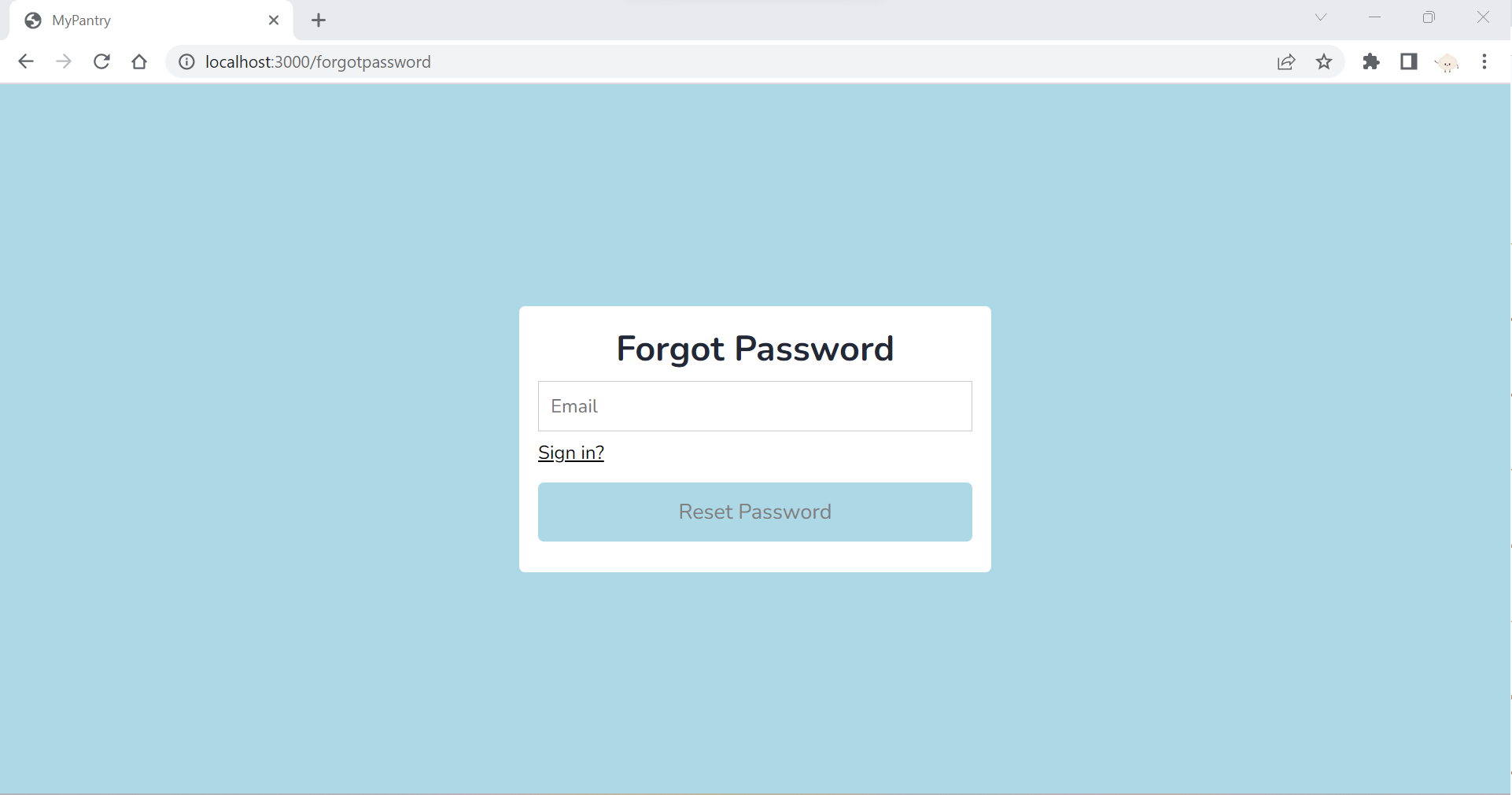






3.1.3 Forgot Password Page

Users who forgot their password can click on the “Forgot Password?” button which will navigate them to the Forgot Password page. Here, they will be required to input their email address in which a password reset link will be sent so users can reset their password.



3.1.4. MyPantry Page

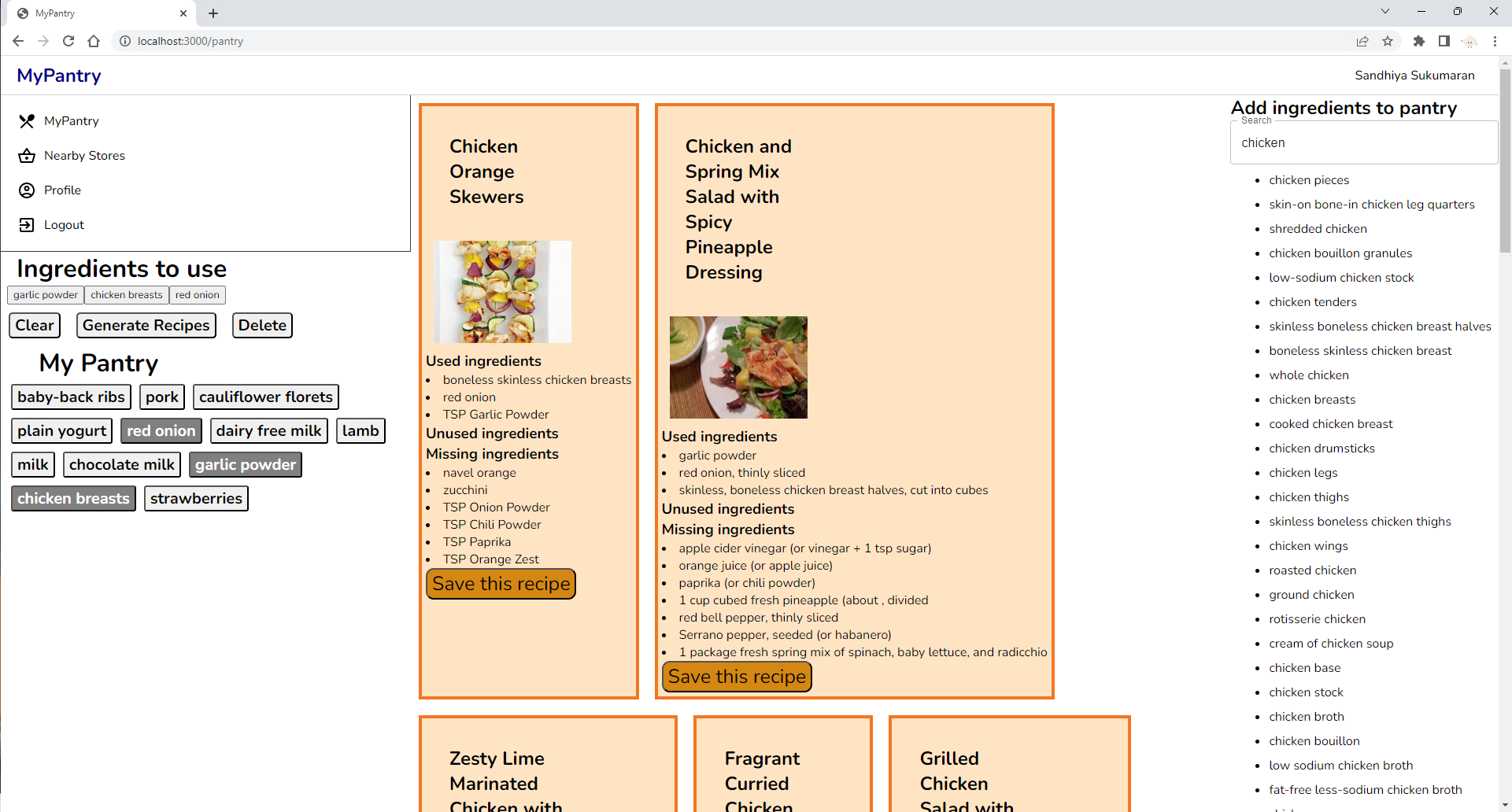
This is the main landing page of the MyPantry application.

The right side of the page lists all the possible ingredients that users can add to their pantry. Users may also use the search bar on the right to search for their desired ingredients. Clicking on the ingredients name will then add it to the “My Pantry” section on the left side of the page.

The “My Pantry” section represents the ingredients that the users already have in their pantry. Clicking on the ingredients in “My Pantry” adds it to the “Ingredients to use” list which is the list of ingredients that will be used to generate recipes. When the “Generate Recipes” button is clicked, a list of recipes utilizing the selected ingredients will be displayed. The “Clear” button removes all the ingredients in the “Ingredients to use” list whereas the “Delete” button allows users to remove selected ingredients from “My Pantry”.

For each recipe, the used and unused ingredients from the list and the additional ingredients needed for the recipe will be displayed. By clicking on the picture of the recipe, the application will direct the users to the recipe website for that particular recipe where the full recipe information such as the nutritional information and the steps to make the recipe will be shown.

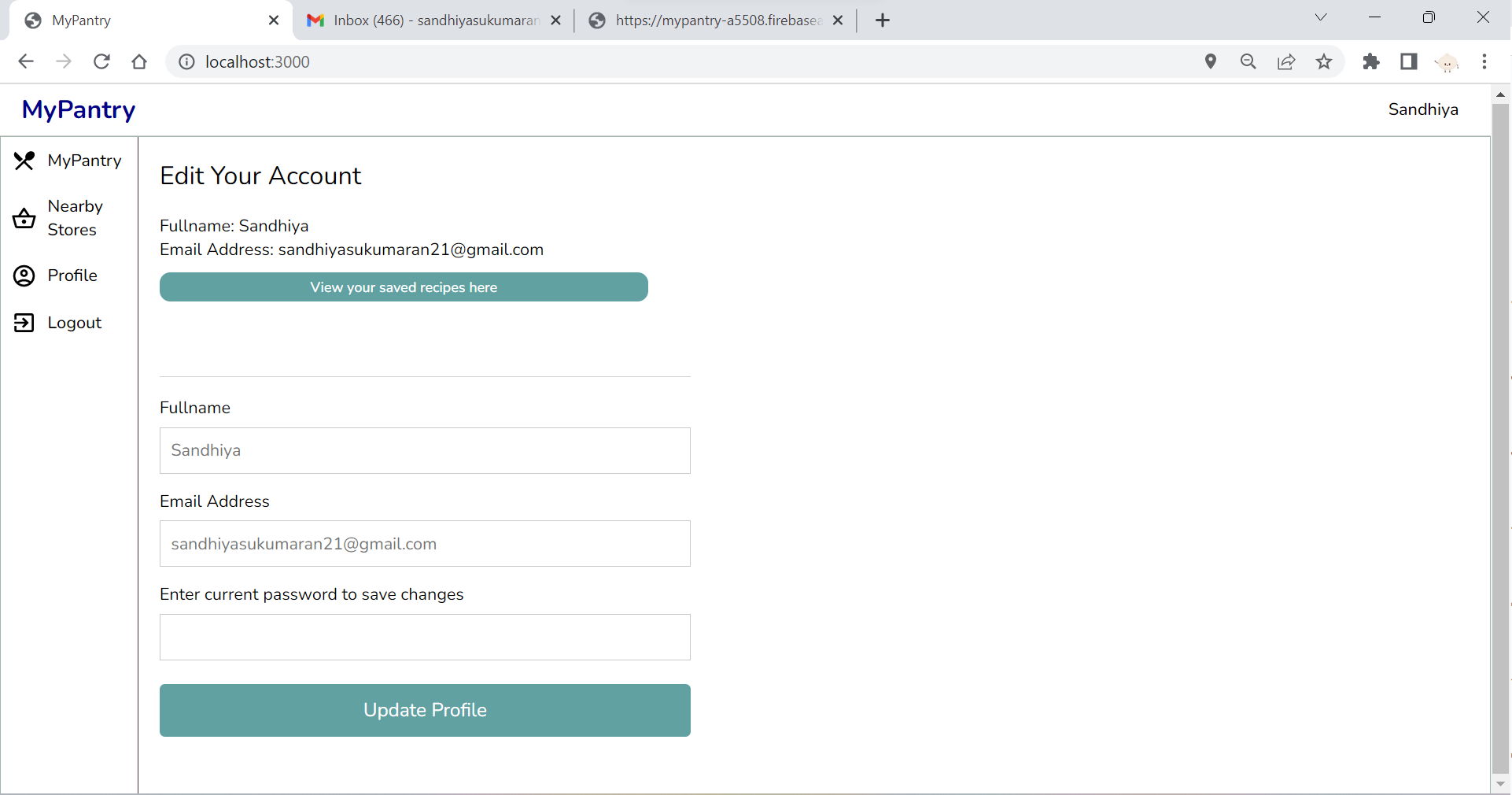
There is a “Save this recipe” button for each recipe that allows users to save their favorite recipes into the “Saved Recipes” page for easier access.



3.1.5. User Profile Page (normal sign in)

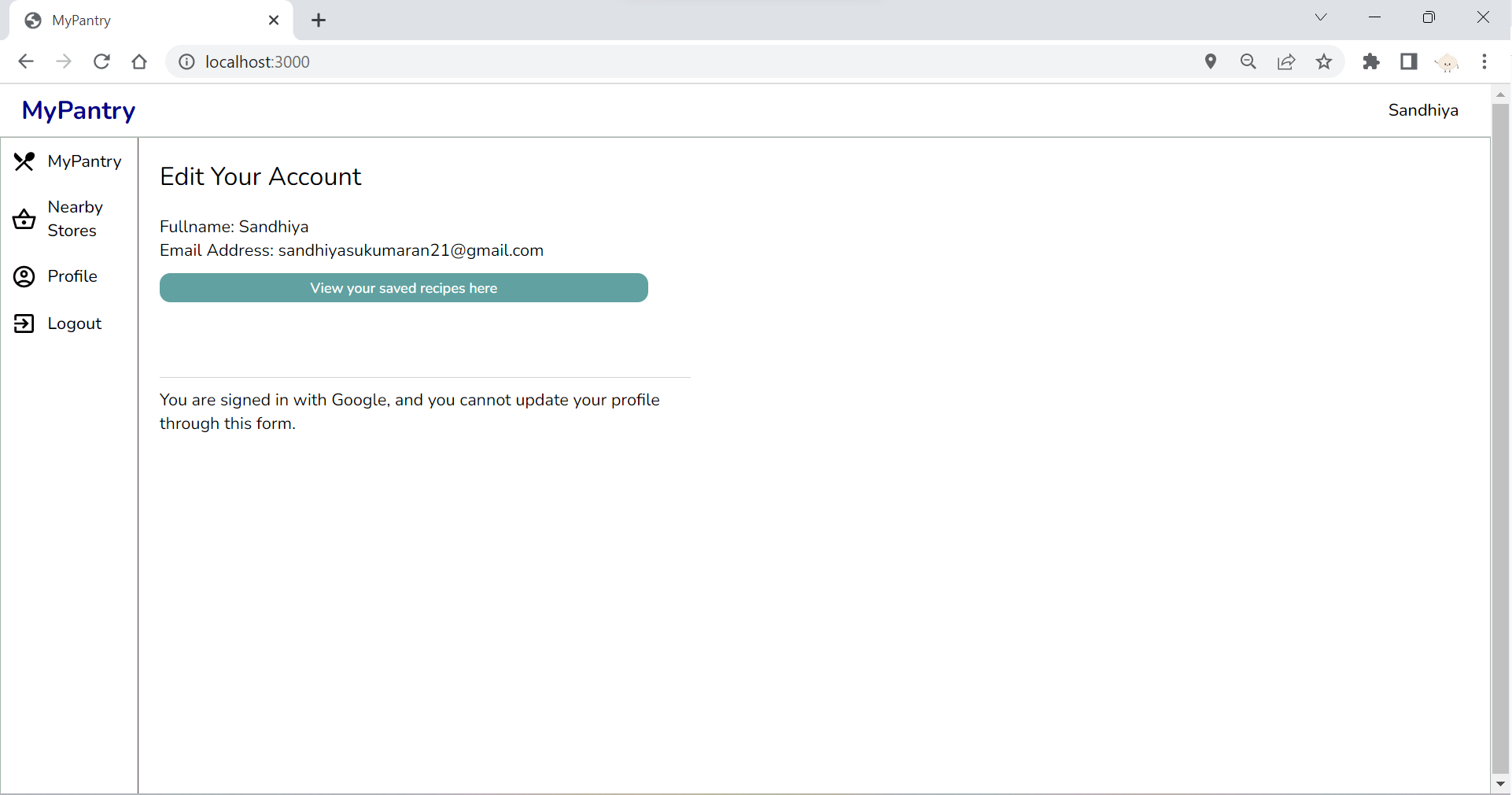
The User Profile page displays the user’s particulars such as the full name and email address. If the user logged in to the application using their email address and password, they would be able to edit these information in the user profile’s page. The same validation criteria are placed when editing these fields. Once the information is updated, a verification email will be sent to the user once again to verify these changes. The users will then be logged out and asked to re login using the new credentials.

When users click on the “View your saved recipes here” button, they will be directed to the Saved Recipes page.



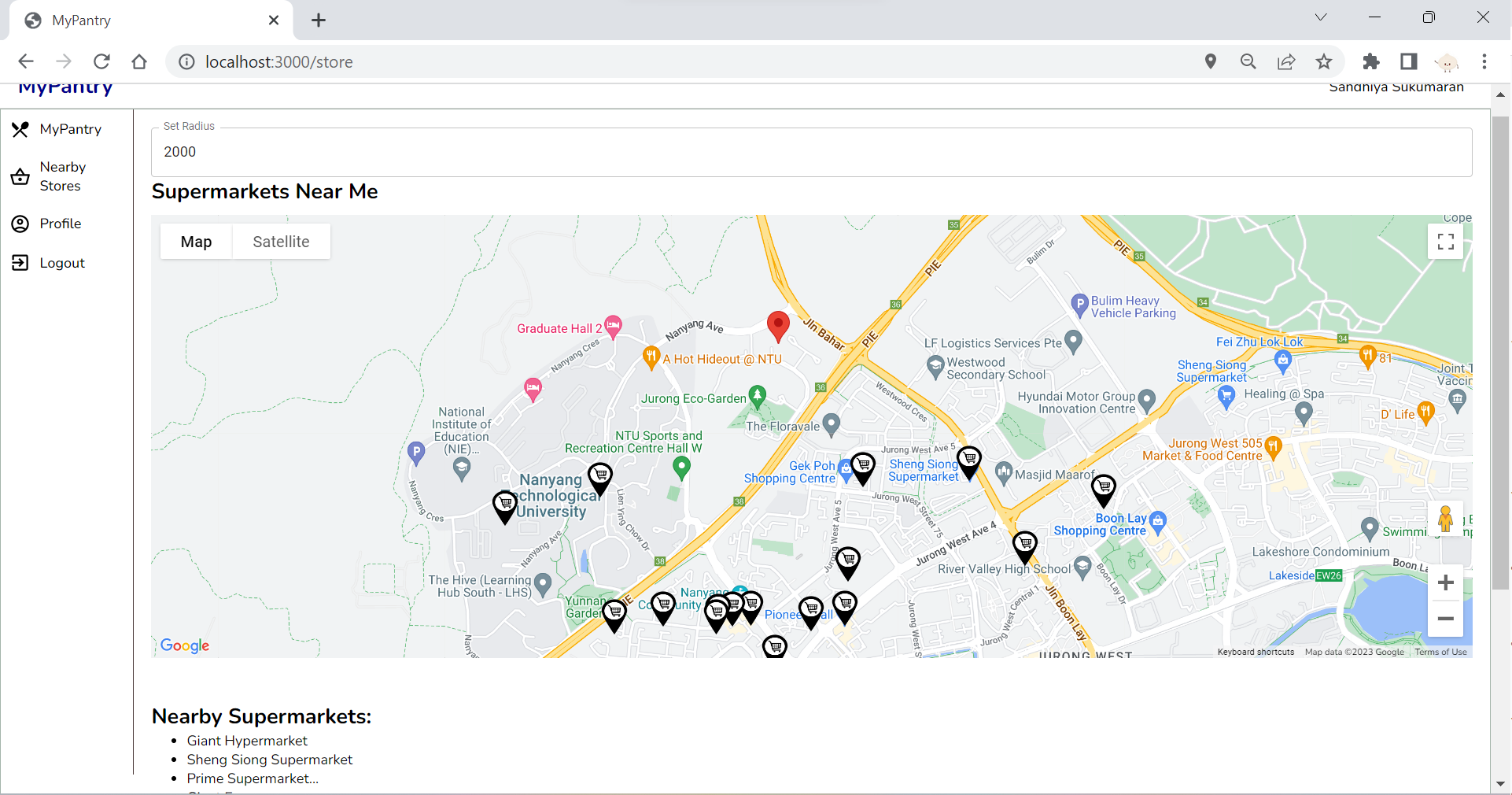
3.1.6 User Profile Page (Google sign in)

Users who logged in using their google account will not be allowed to edit their personal particulars in the User Profile page. When users click on the “Logout” button, they will be navigated to the login page once again.



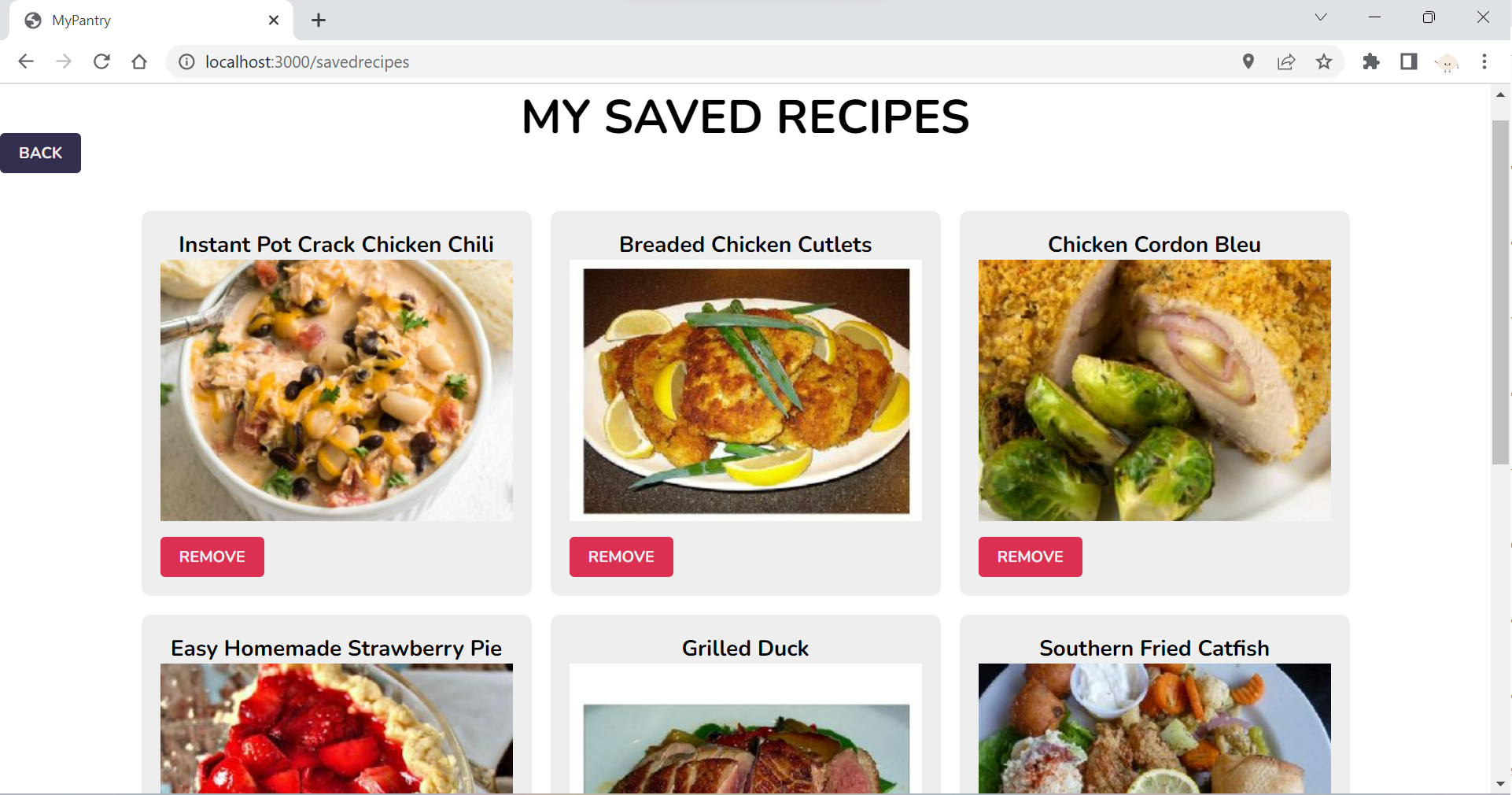
3.1.7. Nearby Supermarkets Page

The Nearby Supermarkets Page displays all the nearby grocery stores within a specific radius from the user’s location. The users may change the radius by inputting a valid number in the “Set Radius” text field at the top of the page. The list of the grocery stores that are within the radius set will also be displayed at the bottom of the screen. The red marker in the map represents the user’s current location whereas the trolley markers represent the grocery stores in the area. Clicking on the google map will direct the users to the Google Maps page. Users may also toggle between map and satellite view from within the page itself.



3.1.8. Saved Recipes Page

The Saved Recipes Page displays the list of recipes that have been saved by the users from the My Pantry Page. It can be accessed through the “View your saved recipes here” button from the profile page. Users may click on the recipe’s picture to direct them straight to the recipe’s website. Users may also remove any recipe from the Saved Recipes page by clicking on the “Remove” button.



## Hardware Interfaces

MyPantry Application can be accessed using a variety of devices such as laptops, tablets and smartphones, as long as they have a modern website browser and an internet connection.

## Software Interfaces

The MyPantry Application is built using React JS for its front-end development and Firebase for its back-end services. MyPantry also relies on external APIs to enable the functionalities in the application.

|  |  |
| --- | --- |
| **React JS libraries** | **Description** |
| Material UI Library | Provides customisable UI components such as buttons and icons to create responsive and attractive user interfaces. |
| React Router Library | Provides the routing functionality for the application. It handles the navigation between multiple pages in the application to ensure that the UI is updated correctly based on the user’s action. |

|  |  |
| --- | --- |
| **Firebase services** | **Description** |
| Firebase Authentication | Used for user authentication which enables the account sign in and registration, email verification and password reset functionality. |
| Firestore built-in APIs   * onSnapshot * getDocs * getFirestore * deleteDoc | Used for data storage and real-time data synchronization. It is used to store the list of ingredients in “My Pantry” as well as the Saved Recipes. |

|  |  |
| --- | --- |
| **External APIs** | **Description** |
| Google Sign In API | Enable users to sign in to the MyPantry application using their Google account credentials. |
| Google Map API | Used to display the google map that shows the user’s location and the location of nearby grocery marts. |
| Google Places API | Used to provide information about these grocery marts such as their names and addresses. |
| Spoonacular recipe information by ID API | Used to retrieve detailed information about a recipe using its ID. |
| Spoonacular search recipe by ingredients API | Used to generate recipes based on a list of ingredients. |

## Communications Interfaces

**3.4.1 The Web browser and network protocols are as follows:**

* Modern web browsers like Chrome, Firefox, Safari, and Edge should be compatible with the application.
* Client and server communications should be secured using HTTPS. Using HTTPS ensures the security of data in transit and eliminates the risk of man-in-the-middle attacks.
* Using WebSockets or Firebase's Realtime Database will allow real-time communication between server and client. Communication over WebSockets is full-duplex, allowing simultaneous bi-directional communication.

**3.4.2 Firebase API Communications:**

* Leverage Firebase Authentication for secure user authentication. It supports various authentication methods, such as email/password, social media logins, and single sign-on (SSO) providers.
* Utilize Firebase Firestore, a NoSQL cloud database, for storing and retrieving application data. Firestore offers real-time data synchronization and offline support for web and mobile clients.

**3.4.3 Communication via email:**

* Use Firebase Extensions, such as the "Authentication" extension to send transactional emails (e.g., account verification, password reset, email update).

**3.4.4 Standards and security in communication:**

* Follow the OAuth 2.0 standard for user authentication and authorization when integrating third-party services.
* Store API keys, secrets, and other sensitive information in secure environments such as environment variables or configuration files. Use Firebase Functions config to store and manage sensitive data for cloud functions.
* Implement Cross-Origin Resource Sharing (CORS) to restrict unauthorized domains from making requests to your server.

# System Features

## Login and Account creation page

**4.1.1 Description and Priority**

This feature contains the login UI for the user to input their email and password. It also allows the user to sign in with their Google account. If the user does not have an account, the user can sign up for an account with an email and password.

Priority: High

**4.1.2 Stimulus/Response Sequences**

1. User clicks on sign up.
   1. System will bring the user to the account registration page.
   2. User input email, name, password, and confirm password.
   3. System will check if full name is valid, i.e it is 4-20 characters long with no numerical or special characters.
      1. If full name is invalid, system will disable Sign up button.
   4. System will check if the email is valid, i.e it consists of a username, the "@" symbol, and a domain name followed by a period “.”, and then a top-level domain. “[username]@[domainName].[topLevelDomain]”
      1. If email is invalid, system will disable Sign Up button.
   5. System will check if the password has at least
      1. 1 uppercase letter
      2. 1 lowercase letter
      3. 1 special character
      4. is between 8-24 characters
      5. If the password is invalid, the system will disable the Sign Up button.
   6. System will check if input in the confirm password matches the input in password.
      1. If the Confirm Password does not match with Password, Sign Up button is disabled.
   7. User clicks on Sign Up button, and if it is not disabled
      1. System will check if the email has already been used before in account creation.
      2. If it has, “This email is already in use, please use another email.” will be displayed.
      3. If it has not been used before, "Account created successfully! Please check your email inbox for a verification email” will be displayed and account will be registered.
2. User wants to login, the user can login via Google login, or by using email and password that was registered when signing up.
   1. If user uses Google login, he will click Login with Google, and select a google account to login from. The user will be prompted to enter his google account details if his google account is not on auto-login, else the user would be logged in instantly after selecting the google account to be used.
   2. If user uses email and password to login, the system will validate email and password.
      1. If correct, the system will log the user in.
      2. If wrong, the system will display “Error signing in. Please check your email and password.”, and allows the user to input the login details again.

**4.1.3 Functional Requirements**

REQ-1:

1. The system must let the user sign up for an account.
   1. System must create an account for the user once a valid email and password is fulfilled.
   2. System must allow the user to sign up with an email, and password.
      1. System must disable the sign up button if the password didn’t adhere to requirements: at least 1 special character, 1 uppercase letter, 1 lower case letter, between 8-24 characters.
      2. System must warn the user that email has already been taken to create an account.
   3. System must allow the user to sign up with a Google account.

REQ-2:

1. If not logged in, the system must let the user log in.
   1. If not logged in, the system must let the user log in with a Google Account.
   2. If not logged in, the system must let the user login with email and password

REQ-3:

1. After the user logs in, the system must validate the user’s login details.
   1. If user’s login details are correct, system must log into the user’s account
   2. If user’s login details are incorrect, system will generate a pop-up “Error signing in. Please check your email and password”.

REQ-4:

1. If logged in, the system must let the user logout.

## MyPantry

**4.2.1 Description and Priority**

This page displays the list of possible ingredients the user can add to his pantry.The user can also search for ingredients from the list. This page also displays the user's pantry and what ingredients are in the pantry. Users can select what ingredients he wants to use from his pantry to generate a recipe. The system allows users to generate recipes based on what ingredients they want to use in their pantry. Generated recipes will show what ingredients have been used, what ingredients were not used, and what ingredients are missing. The user can also click on the image of the generated recipe to navigate to a website that will display the complete recipe information.

Priority: High

**4.2.2 Stimulus/Response Sequences**

1. User clicks on the search box.
2. System allows user to enter text to search for an ingredient
3. User enters an ingredient name as the input.
4. System displays all the ingredients that are a superset of the input.
   1. e.g If input is “garlic”, the ingredients displayed will be “garlic”,”garlic powder”,”garlic paste”, “granulated garlic” etc.
5. User clicks on a displayed ingredient to add to his pantry.
6. The system adds the clicked ingredient to the pantry, and the pantry will display the newly added ingredient.
7. User can click on multiple ingredients in the pantry to add them in the “ingredients to use” section to generate a recipe.
8. User can deselect and remove the ingredient from the “ingredients to use” section by clicking on the ingredient again in the pantry.
9. User can deselect and remove all the ingredients from the “ingredients to use” section by clicking on the “clear” button.
10. User can delete ingredients from his pantry by clicking on the “delete” button to toggle to delete mode, and when the user clicks on an ingredient from his pantry, the ingredient is deleted.
11. User can click on “Generate Recipes” to generate recipes that will use the ingredients in the “Ingredients to use” section.
12. The system will then generate recipes based on the ingredients selected, and display the recipe image, a “save recipe button” and based off the ingredients in the “ingredients to use” section, the system must display
    1. ingredients used in the recipe, i.e ingredients in the “ingredients to use” section that are used in the recipe
    2. ingredients not used in the recipe, i.e ingredients in the “ingredients to use” section that are not needed in the recipe
    3. ingredients missing from the recipe, i.e ingredients not in the “ingredients to use” section but are needed in the recipe
13. User clicks on “save recipe”, the system must save the recipe into the “Saved Recipes” page.
14. User clicks on the image, the system must redirect the user to another website that contains a full detailed information on the recipe that contains the steps in making the recipe, the cooking duration, and the nutritional information.

**4.2.3 Functional Requirements**

REQ-5:

1. The system must let the user add ingredients to the pantry.
   1. The system must let the user search for ingredients in the search bar.
   2. The system must display all the ingredients that are supersets of the input text in the search bar.
   3. If no ingredients are supersets of the input text, the system will display no ingredients.
   4. If there is no input in the search bar, the system must display all the ingredients in the list.
   5. The system must add the ingredients to the “My Pantry” section when the user clicks on an ingredient from the list.

REQ-6

1. The system must let the user generate recipes.
   1. The system must let the user select 1 or more ingredients in the pantry to use in generating a recipe.
   2. If user clicks on an ingredient from the “My Pantry” section,
      1. If “Delete” button is shown
         1. The system must add the ingredient to the “ingredients to use” section if the ingredient is not in the “ingredients to use” section.
         2. The system must remove the ingredient from the “ingredients to use” section if the ingredient is in the “ingredients to use” section.
      2. If “Deleting from pantry” button is shown
         1. The system must remove the ingredient from the “My Pantry” section.
   3. If the user clicks on “Clear”, the system must remove all the ingredients from the “ingredients to use” section.
   4. If the user clicks on “Delete”, the system must change the button text to “Deleting from pantry” and change the button color to red.
   5. If the user clicks on “Deleting from pantry”, the system must change the button text to “Delete” and change the button color to light gray
   6. The system must generate recipes when “Generate Recipes” is clicked.

REQ-7

1. For all the displayed suggested recipes, the system must display the recipes’
   1. Ingredients that are in the “Ingredients to use section” and are used in the recipe
   2. Ingredients that are in the “Ingredients to use section” but are not used in the recipe
   3. Ingredients that are not in the “Ingredients to use section” but are used in the recipe.
   4. Image of the food that will be cooked by following the recipe
   5. “Save recipe” button that saves the recipe to the user’s saved recipe page.

REQ-8

1. The system must navigate to the user to the source URL of the recipe when the user clicks on the recipe’s image.
   1. The webpage must display instructions on preparing the dish
   2. The webpage must display the full nutritional information of the dish
   3. The webpage must display the ingredients list
   4. The webpage must display the equipments list
   5. The webpage must display the estimated price breakdown.

REQ-9:

1. The system must allow the user to click on “save recipe” for a recipe the user wants.
   1. The system must add the recipe the user saved to the “saved recipes” collection.
   2. The “saved recipes” collection can be viewed in the user’s profile page.

## Nearby Grocery Stores

* + 1. **Description and Priority**

This page displays grocery stores that are near the user. It uses live location to determine which stores are near given a certain input radius. The default radius is set to 1500 meters.

Priority: Medium

* + 1. **Stimulus/Response Sequences**

1. User click on the nearby groceries store page.
2. The system requests for permission to access the user’s live location.
3. The user clicks on accept.
4. The system displays the groceries stores within a 1.5km radius of the user’s location on a google maps window.
5. The user sets the radius to 2000m using the “Set Radius” bar above.
6. The google maps window re-renders and shows all grocery stores within a 2km radius.
7. There is a list of nearby supermarkets displayed below the google maps window.
8. The user can click on a location on the google map, and click on view in google maps, and the user will be redirected to the google maps page.
   * 1. **Functional Requirements**

REQ-10:

1. The system must use the user’s location to display nearby grocery stores.

REQ-11:

1. If the user clicks on “Find on Google Maps”, the system must open the Google Maps page.

REQ-12:

1. The system must display all grocery stores within a 1.5km radius within the user’s location

REQ-13:

1. The system must allow the user to set a new radius by inputting a number to represent the new radius in terms of meters in the radius bar, and then pressing enter.

REQ-14:

1. The system must display a warning to the user that the input has to be integers only, if the user did not input an integer.

REQ-15:

1. The system must re-render the google maps window to display all grocery stores within the user’s newly set radius within the user’s location when the user inputs a valid integer and presses enter.

REQ-16:

1. The system must display the names of the supermarkets in a list within the radius below the Google maps window.

## My Profile Page

* + 1. **Description and Priority**

This page shows the user's personal information, such as full name, email used. There is a “View my saved recipes button” that allows users to view all the recipes they have saved.

Priority: Medium

* + 1. **Stimulus/Response Sequences**

1. If users sign in using google sign in
   1. When users click on their profile page, their full name, email address used is displayed.
   2. The user clicks on “View my saved recipes”.
   3. The system redirects the user to the saved recipes page.
2. If users sign in using email address and password
   1. When users click on their profile page, their full name, email address used is displayed.
   2. The user clicks on the Name text box.
   3. The user enters a new name
   4. The user enters his used email address.
   5. The user clicks on Update Profile.
   6. The system changes the user’s name.
   7. The user clicks on “View my saved recipes”.
   8. The system redirects the user to the saved recipes page.
      1. **Functional Requirements**

REQ-17:

1. If the user logged in using an email address and a password
   1. The system must allow the user to type in a new name in the Fullname text box
   2. The system must allow the user to type in the Email Address text box.
   3. The system must allow the user to type in the Password text box.
   4. The system must allow update the account’s full name if the conditions are met
      1. The user enters the correct email address used in the Email Address textbox
      2. The user enters the correct password in the Password textbox
      3. The user clicks on Update Profile button after the previous 2 conditions are met.

REC-18:

1. If the user logged in using google sign in
   1. The system must display the Fullname as the google email name and google email address used.
   2. The system must not let the user make any edits since it is tied to the google account and is fixed.

REC-19:

1. The system must allow redirect the user to the saved recipes page when the user clicks on “view your saved recipes here”.

## Saved Recipes Page

* + 1. **Description and Priority**

This page displays all the recipes that the user has saved. The user can click on any of the recipes to directly access the recipe page. This page allows a user to revisit a recipe conveniently that the user really enjoyed without having to generate the recipe again.

Priority: Medium

* + 1. **Stimulus/Response Sequences**

1. The user can view all the saved recipes
2. The system displays a red “remove” button on the bottom left corner of saved recipes.
3. The user clicks on “remove”.
4. The system removes the recipe from the saved recipes page.
5. The user clicks on a recipe.
6. The system redirects the user to a webpage that displays the recipe, showing the nutritional information and the preparation time.
   * 1. **Functional Requirements**

REQ-20:

1. The system must display all the saved recipes when in the saved recipes page

REQ-21:

1. The system must display the image of the saved recipe

REQ-22:

1. The system must display the title of the saved recipe

REQ-23:

1. The system must remove the recipe from the saved recipes page when the user clicks on “remove” button.

REQ-24:

1. The system must redirect the user to the webpage when they click on a recipe’s image.

# Other Nonfunctional Requirements

## Performance Requirements

* + 1. The web application should be able to boot up within 10 seconds when the number of users using the app is more than 10,000.
    2. The web application should have a response time of less than 2 seconds.
    3. The web application should be able display recipe outputs within 3 seconds after the user selects the ingredients.
    4. The web application should be able to update the database and display it within 1 second after the user adds/removes ingredients.
    5. The web application must display Nearby Stores within 5 seconds of the user providing location access.
    6. The web application should be able to send a reset password email within 2 minutes after the user selects the “Reset Password” button.
    7. The web application should take less than 3 seconds to load after the user signs in.

## Usability Requirements

* + 1. The UI should be simple, organized and intuitive enough for users to navigate.
    2. Text fields requiring input from the user should have example or hidden text as much as possible, to guide the users on what to input in the corresponding text field.

## Security Requirements

* + 1. The web application’s back-end can only be accessed by authorized personnel from the developer team.
    2. Each time the user uses the web application they will be prompted to log into their myPantry account.
    3. The account password must be a minimum of 8 characters and maximum of 24 characters with at least one uppercase letter, one lowercase letter, one number and one special character.
    4. The data collected from the user is to be stored securely away from public access.

## Reliability Requirements

* + 1. The web application must run without failure 95% of the time.
    2. The web application must back up its data immediately after it receives new data.

## Supportability Requirements

* + 1. The web application must be able to support 10,000 users simultaneously.
    2. The web application needs to be compatible with web browsers like Google Chrome, Safari, Mozilla Firefox, etc.
    3. The web application’s backend database must be replaceable with any commercial product supporting standard SQL queries.

**Appendix A: Glossary**

# Data Dictionary

|  |  |
| --- | --- |
| Term | Definition |
| User | A person with an account in the system and uses the services the application provides. |
| Profile | A list of user profile details in the system containing (Name, Email, Date-Of-Birth) associated with them. |
| Password | A string of characters (letters, numbers, special symbols) that protects an account from unwanted access. |
| Pantry | The list of food ingredients that the user currently has or can have access to. |
| Ingredient | An edible food or substance used to combine to make a particular dish. |
| Recipe | A set of instructions for preparing a particular dish, including a list of the ingredients required. |
| SavedRecipes | An interface that displays the list of recipes that users saves. |
| NearestMart | A list of nearest marts generated from user’s geographical location. |

**Appendix B: Analysis Models**

All the relevant diagrams, use case description and test cases and results are uploaded as separate files in the SVN repository.

**Appendix C: Additional Information**

For clearer images of our various diagrams as well as the source code of the MyPantry Application. Please refer to our Github (https://github.com/admtn/MyPantry-SC2006/tree/main/Final-Documentation) or the SVN repository.