Contents

[INTRODUCTION 2](#_Toc218199110)

[Project Overview 2](#_Toc218199111)

[Solution Provided 2](#_Toc218199112)

[Technologies Used 2](#_Toc218199113)

[SYSTEM OVERVIEW 2](#_Toc218199114)

[Purpose 2](#_Toc218199115)

[Objectives 2](#_Toc218199116)

[FEATURES & IMPLEMENTATION 2](#_Toc218199117)

[User Authentication System 2](#_Toc218199118)

[Grocery Tracking Module 4](#_Toc218199119)

[Recipe Finder System 6](#_Toc218199120)

[Progress Dashboard 7](#_Toc218199121)

[User Profile & Settings 8](#_Toc218199122)

[Responsive Design 10](#_Toc218199123)

[CODE BLOCKS 10](#_Toc218199124)

[SYSTEM ARCHITECTURE 12](#_Toc218199137)

[Architecture Diagram 12](#_Toc218199138)

[Three-Tier Architecture: 12](#_Toc218199139)

[Database Schema 12](#_Toc218199140)

[Data Flow 13](#_Toc218199141)

[TESTING & RESULTS 13](#_Toc218199142)

[Testing Summary 13](#_Toc218199143)

[Performance Metrics 13](#_Toc218199144)

[User Feedback 13](#_Toc218199145)

[Challenges & Solutions 13](#_Toc218199146)

[CONCLUSION 14](#_Toc218199147)

[Project Achievements 14](#_Toc218199148)

[Learning Outcomes 14](#_Toc218199149)

[Impact Assessment 14](#_Toc218199150)

# INTRODUCTION

## Project Overview

Save2Serve is a platform designed to help households reduce food waste by tracking grocery expiry dates and suggesting recipes to use ingredients before they spoil. The application provides a complete solution from inventory management to meal planning.

**Problem Statement**

* **No system to track grocery expiry dates** - Items spoil before use
* **Difficulty finding recipes** - Don't know how to use available ingredients
* **Food gets forgotten** - Hidden in fridge and pantry
* **No progress measurement** - Can't track waste reduction improvements
* **Inefficient shopping** - Buy items already available at home

## Solution Provided

* **Digital grocery tracker** - Track items with expiry dates and categories
* **Smart recipe finder** - Search recipes based on available ingredients
* **Progress dashboard** - Visual charts showing used vs wasted food
* **Expiry alerts** - Automatic "Expiring Soon" section for urgent items
* **User-friendly interface** - Simple design that works on all devices

## Technologies Used

* **Frontend:** HTML, CSS, JavaScript
* **Backend:** Supabase (Authentication + Database)
* **Libraries:** Chart.js, Font Awesome, Google Fonts

# SYSTEM OVERVIEW

## Purpose

To create an accessible web application that helps users reduce food waste through better grocery management and recipe discovery.

## Objectives

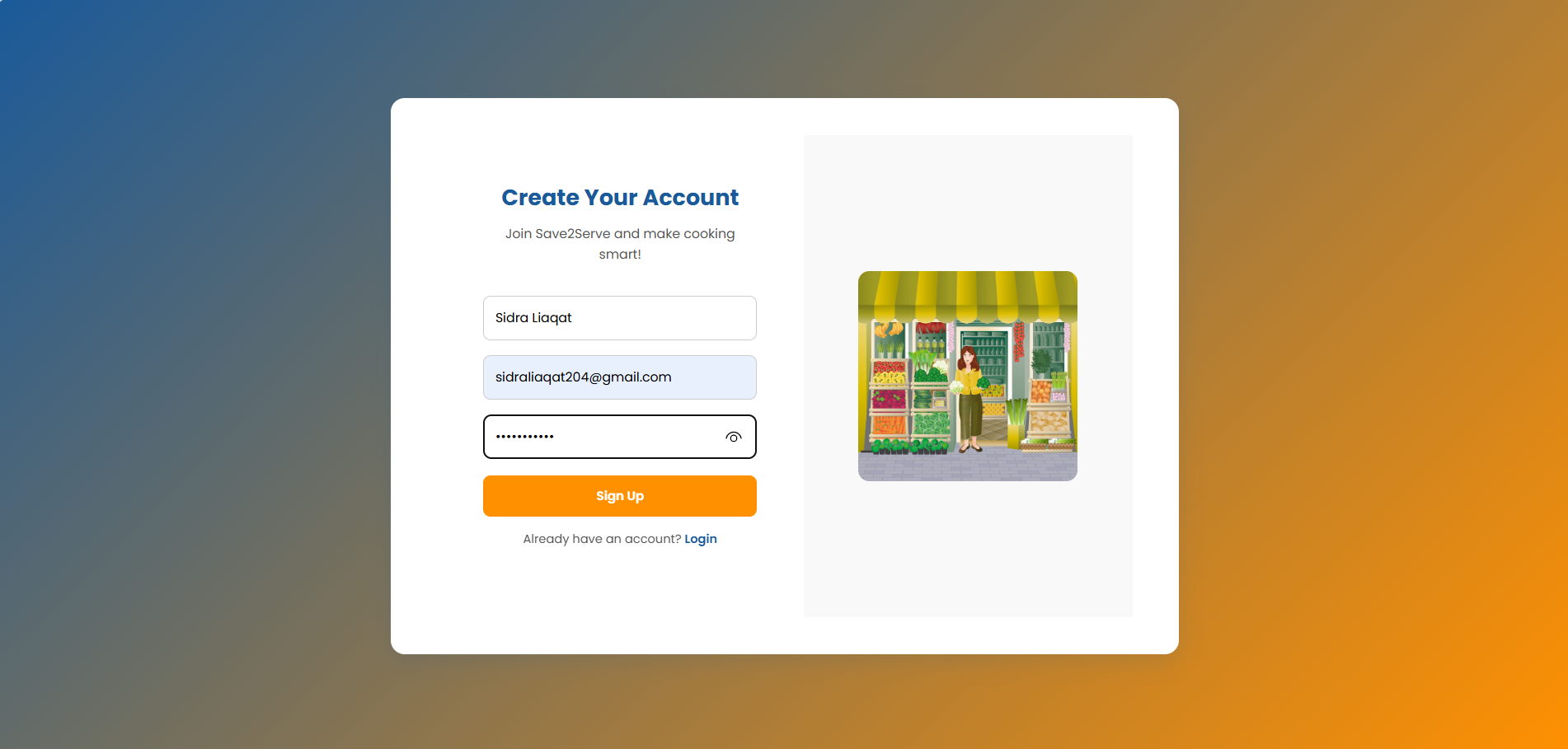
* User registration and secure authentication system
* Complete CRUD operations for grocery items
* Recipe search system based on available ingredients
* Progress dashboard with charts and statistics
* Responsive web design for all devices
* User profile and settings management
* GDPR-compliant cookie consent system
* Privacy control

# FEATURES & IMPLEMENTATION

## User Authentication System

**Implementation:**

* Email/password registration with verification



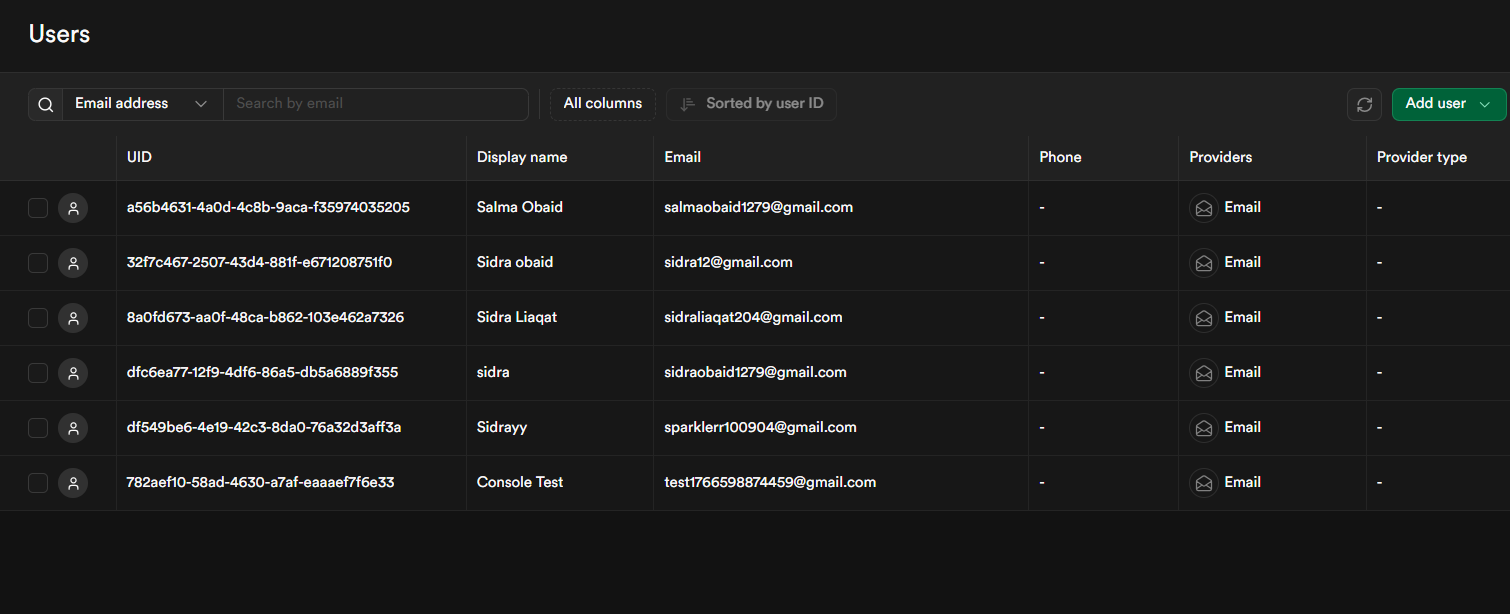
A screenshot of a phone

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

* Secure login using Supabase Auth



* Session persistence across pages

A screenshot of a store

AI-generated content may be incorrect.

* Protected route access

## Grocery Tracking Module

**Features:**

* Add items with name, expiry date, and category

A screenshot of a grocery list

AI-generated content may be incorrect.

* Automatic "Expiring Soon" categorization (≤4 days)

A screenshot of a computer

AI-generated content may be incorrect.

* Mark items as used when consumed

A screenshot of a chat

AI-generated content may be incorrect.

* Delete expired items

A screenshot of a computer

AI-generated content may be incorrect.

**Implementation:**

* Form validation for required fields

A black background with white text

AI-generated content may be incorrect.

* Date picker with future dates only

A screenshot of a calendar

AI-generated content may be incorrect.

* Real-time database updates

A screenshot of a computer

AI-generated content may be incorrect.

* Visual status indicators

## Recipe Finder System

**Features:**

* Search recipes by multiple ingredients

A screenshot of a computer

AI-generated content may be incorrect.

* Auto-search from grocery items

A white rectangular object with a white background

AI-generated content may be incorrect.

* YouTube video links integration

A screenshot of a video

AI-generated content may be incorrect.

* Highlight matching ingredients

A screenshot of a recipe

AI-generated content may be incorrect.

**Implementation:**

* Comma-separated ingredient search
* Database query optimization
* Responsive card layout
* External link handling

## Progress Dashboard

**Features:**

* Statistics cards (Total, Used, Wasted, Saved)

A screenshot of a grocery store

AI-generated content may be incorrect.

* Doughnut chart visualization

A pie chart with different colored circles

AI-generated content may be incorrect.

* Used vs wasted item lists

A screenshot of a chat

AI-generated content may be incorrect.

* Waste reduction tips

A close-up of a white background

AI-generated content may be incorrect.

**Implementation:**

* Chart.js integration for graphs
* Dynamic data calculation
* Modal popup design
* Responsive chart sizing

## User Profile & Settings

**Features:**

* Personal information display

A screenshot of a profile

AI-generated content may be incorrect.

* Email and password update

A screenshot of a login screen

AI-generated content may be incorrect.

* Cookie consent management

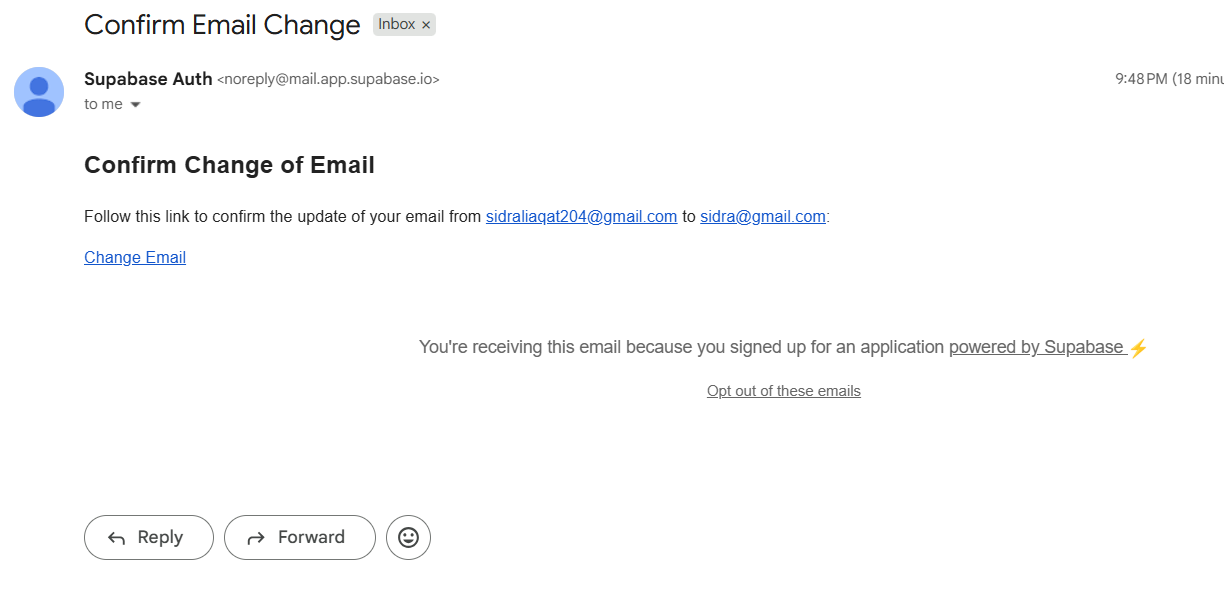
A screenshot of a computer

AI-generated content may be incorrect.

* Privacy controls

A screenshot of a computer

AI-generated content may be incorrect.



**Implementation:**

* User metadata display
* Form validation for updates
* Cookie preference storage
* Status indicators

## Responsive Design

**Features:**

* Mobile-first approach
* Adaptive layouts
* Touch-friendly interfaces
* Cross-browser compatibility

**Implementation:**

* CSS Flexbox and Grid
* Media queries for breakpoints
* Relative units (%, rem, vw)
* Progressive enhancement

# CODE BLOCKS

## Authentication Code (Supabase)

const { data, error } = await supabase.auth.signInWithPassword({

email: email,

password: password

});

## Database CRUD Operation

await supabase

.from('grocery\_items')

.insert([{ name, expiry\_date, category, user\_id }]);

## Expiry Logic

const daysLeft = (expiryDate - today) / (1000 \* 60 \* 60 \* 24);

if (daysLeft <= 4) {

status = "Expiring Soon";

}

## Recipe Search Logic

const ingredients = input.split(",");

fetchRecipes(ingredients);

## Chart.js Visualization

new Chart(ctx, {

type: 'doughnut',

data: chartData

});

## Responsive CSS Snippet

@media (max-width: 768px) {

.container {

flex-direction: column;

}

}

## Search Filter Logic

const filtered = items.filter(item =>

item.name.includes(searchValue)

);

## Mark Item as Used

await supabase

.from('grocery\_items')

.update({ used: true })

.eq('id', itemId);

## Delete Expired Items

await supabase

.from('grocery\_items')

.delete()

.lt('expiry\_date', today);

## Error Handling

if (error) {

console.error(error.message);

}

## Cookie Consent Logic

if (!localStorage.getItem("cookieConsent")) {

showCookieBanner();

}

## Logout Functionality

await supabase.auth.signOut();

window.location.href = "login.html";

# SYSTEM ARCHITECTURE

## Architecture Diagram

A diagram of a computer

AI-generated content may be incorrect.

## Three-Tier Architecture:

1. **Presentation Layer:** HTML/CSS/JS in browser
2. **Application Layer:** Supabase backend services
3. **Data Layer:** PostgreSQL database

## Database Schema

A screenshot of a computer

AI-generated content may be incorrect.

## Data Flow

User Action → Frontend Processing → API Call → Supabase Service → Database Operation → Response → UI Update → User Feedback

# TESTING & RESULTS

## Testing Summary

| Test Case | Description | Result |
| --- | --- | --- |
| TC-01 | User Registration | ✅ PASS |
| TC-02 | User Login | ✅ PASS |
| TC-03 | Add Grocery Item | ✅ PASS |
| TC-04 | Mark Item as Used | ✅ PASS |
| TC-05 | Recipe Search | ✅ PASS |
| TC-06 | Progress Dashboard | ✅ PASS |
| TC-07 | Mobile Responsiveness | ✅ PASS |
| TC-08 | Cookie Consent | ✅ PASS |

## Performance Metrics

* **Page Load Time:** 2.3 seconds average
* **Database Response:** 180ms average
* **Mobile Score:** 86/100 (Lighthouse)
* **Desktop Score:** 92/100 (Lighthouse)

## User Feedback

* "Easy to use and intuitive interface"
* "Recipe suggestions are very helpful"
* "Progress tracking motivates to waste less"
* "Works perfectly on mobile phone"

## Challenges & Solutions

1. **Challenge:** Session persistence across pages  
   **Solution:** Implemented Supabase session management
2. **Challenge:** Mobile responsiveness  
   **Solution:** Used mobile-first CSS approach
3. **Challenge:** Real-time data updates  
   **Solution:** Optimized database queries

# CONCLUSION

## Project Achievements

* Developed complete web application
* Implemented all planned features
* Achieved responsive design
* Ensured data security
* Created user-friendly interface

## Learning Outcomes

* Full-stack web development
* Database design and management
* User authentication systems
* Responsive web design
* Project planning and execution

**Future Enhancements**

1. Barcode scanner integration
2. Meal planning features
3. Family sharing options
4. Multi-language support

## Impact Assessment

* **Environmental:** Reduces food waste
* **Financial:** Saves money for families
* **Educational:** Teaches sustainable habits
* **Social:** Promotes responsible consumption

*Save2Serve successfully demonstrates how technology can support sustainable living and responsible food consumption*