

# **GROCERYZ - GROCERY LIST ORGANIZER**

## **WEB APP**



### **A Project Report**

Submitted in partial fulfillment of the  
Requirements for the award of the Degree of

**Bachelor of Computer Applications (BCA)**  
of  
**Kavikulaguru Kalidas Sanskrit University**

Submitted by

**Kalpesh Govind Talesha**

Under the esteemed guidance of  
**Prof. Sneha Shashikant Lokhande**



Kavikulaguru Kalidas Sanskrit University's

**Bakliwal Foundation College of Arts, Commerce & Science**

Vashi

**BATCH: 2022-2025**

Kavikulaguru Kalidas Sanskrit University's  
**Bakliwal Foundation College of Arts, Commerce & Science**  
Vashi



**CERTIFICATE**

This is to certify that the project entitled **Groceryz- Grocery List Organizer Web App** undertaken at the PCP Center: Bakliwal Foundation College of Arts, Commerce & science, Vashi, Navi Mumbai by **MR. KALPESH GOVIND TALESHA** holding **Seat No. (PRN: 2022018100095264)** Studying **Bachelor of Computer Applications** Semester – VI has been satisfactorily completed as prescribed by the Kavikulaguru Kalidas Sanskrit University, during the year 2024 - 2025.

**Project Incharge**

**Co-ordinator**

**External Examiner**

**Internal Examiner**

**Principal**

# ABSTRACT

Groceryz is a comprehensive web application developed to simplify and enhance grocery management for users through a user-friendly platform. The application requires users to register and log in, ensuring data security and a personalized experience. It supports two user roles: Admin and Normal Users. Admins can manage user accounts and permissions, while normal users can create, edit, and share grocery lists.

The application features intuitive list management capabilities, allowing users to add items, edit entries, mark tasks as completed, and share lists with others, enabling real-time collaboration with editing rights. To enhance accessibility, the system maintains the ten most recent grocery lists in the database, facilitating quick retrieval and continuity.

Groceryz is built using core web technologies including HTML, CSS, Vanilla JavaScript, and Modern JavaScript (Fetch API) for the frontend, and PHP with MySQL for the backend. This stack ensures a lightweight, responsive, and efficient user interface with robust server-side processing and secure data management.

Future developments aim to incorporate real-time notifications, mobile application integration, and analytics-based insights to further personalize and enrich the user experience.

# **ACKNOWLEDGEMENT**

I would like to take this opportunity to express my deepest gratitude to **Professor Sneha Lokhande** for her invaluable guidance, support, and encouragement throughout the development of my project, **Groceryz – Grocery List Organizer Web App**. Her extensive knowledge and expertise in web development played a crucial role in shaping the direction and success of this project.

Professor Lokhande consistently provided thoughtful feedback, constructive criticism, and insightful suggestions that helped me refine my ideas and enhance the application's functionality at every stage—from initial concept to final implementation. Her ability to simplify complex concepts allowed me to gain a deeper understanding of the development process and strengthen the technical skills essential for my future professional growth.

I am sincerely grateful for her patience and support during the challenging phases of this project, which kept me motivated and focused on achieving my goals. Her mentorship has been instrumental in the successful completion of the Groceryz application and has inspired me to further pursue my interest in web development.

I would also like to extend my heartfelt thanks to, **Prof. Ankit Srivastava** for his helpful suggestions and support during the course of this project. His input was valuable in overcoming technical challenges and improving key components of the application.

I extend my sincere thanks to **Dr. Sharadkumar Shah, H.O.D Prof. Sneha Shashikant Lokhande, Prof. Shaikh Mohammed Umar, Prof. Divya Patil, Prof. Kalyani Kulkarni**.

Lastly, I would like to thank my friends for their constant encouragement, collaboration, and moral support. Their presence and insights contributed significantly to maintaining a positive mindset and staying on track throughout the project journey.

Once again, my sincere thanks to everyone who contributed in any way to the successful completion of this project.

Thanking you  
Kalpesh Talesha

## **DECLARATION**

I hereby declare that the project entitled, "**Groceryz- Grocery List Organizer Web App**" done at **Bakliwal Foundation College of Arts, Commerce and Science, Vashi, Navi Mumbai**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university.

The project is done in partial fulfillment of the requirements for the award of degree Of **BACHELOR OF COMPUTER APPLICATION** to be submitted as final semester project as part of our curriculum.

Kalpesh Talesha

# TABLE OF CONTENTS

<b>Sr. No.</b>	<b>Title</b>	<b>Page No.</b>
<hr/>		
<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background	2
1.2	Objectives	3
1.3	Purpose, Scope, and Applicability	4
1.4	Achievements	6
<hr/>		
<b>2</b>	<b>System Analysis</b>	<b>7</b>
2.1	Existing System	8
2.1.1	Current Grocery Management Solutions	8
2.2	Proposed System	9
2.2.1	Features	9
2.2.2	Benefits	10
2.3	Requirement Analysis	10
2.3.1	Functional Requirements	10
2.3.2	Non-Functional Requirements	11
2.4	Hardware Requirements	11
2.4.1	Development Environment	11
2.4.2	Deployment Environment (for Hosting the Application)	12
2.5	Software Requirements	12
2.5.1	Development Software	12
2.5.2	Backend	13
2.5.3	Frontend	13
2.6	Justification of Selection of Technology	13
2.6.1	Frontend	13
2.6.2	Backend	14
2.6.3	Database	15
2.6.4	Security	16
<hr/>		
<b>3</b>	<b>Research Methodologies</b>	<b>18</b>
3.1	Questionnaire	19
3.2	Sampling	19

3.3	Data Analysis	20
3.4	Pie Charts	20
3.5	Bar Charts	21
<b>4</b>	<b>Estimation &amp; Planning</b>	<b>22</b>
4.1	Time Estimation	23
4.2	Gantt Chart	24
4.3	Resource Estimation	24
4.4	Planning Strategy	25
<b>5</b>	<b>Methodology</b>	<b>26</b>
5.1	Iterative Development Approach	27
5.2	Development Phases	27
5.2.1	Requirement Analysis	27
5.2.2	System Design	27
5.2.3	Frontend Development	27
5.2.4	Backend Development	28
5.2.5	Integration	28
5.2.6	Testing and Debugging	28
5.2.7	Finalization	28
5.3	Tools & Technologies Used	28
<b>6</b>	<b>System Design</b>	<b>29</b>
6.1	Module Division	30
6.2	Data Dictionary	32
6.3	E-R Diagram	34
6.3.1	Entities and Relationships	34
6.3.2	Diagram Structure	35
6.3.3	E-R Diagram Representation(Textual)	35
6.4	Data Flow Diagrams(DFDs)	36
6.4.1	DFD Level 0 - Context Diagram	37
6.4.2	DFD Level 1 – Main Functional Decomposition	38
<b>7</b>	<b>Implementation and Testing</b>	<b>40</b>
7.1	Code (Place Core segments)	41
7.1.1	Frontend Development	41

7.1.2	Backend Development	42
7.1.3	Core Functionalities Implemented	43
7.1.4	Session Management	44
7.2	Testing Approach	45
7.2.1	Unit Testing	45
7.2.1.1	What is Unit Testing?	45
7.2.1.1.1	Purpose of Unit Testing	45
7.2.1.1.2	How Unit Testing Works	45
7.2.1.1.3	Advantages of Unit Testing	45
7.2.1.2	How it was Applied	46
7.2.2	Integration Testing	46
7.2.2.1	What is Integration Testing?	46
7.2.2.1.1	Purpose of Integration Testing	46
7.2.2.1.2	Techniques Used	46
7.2.2.1.3	Benefits in your Project	47
7.2.2.2	How it was Applied	47
7.2.3	Test Cases	47
<b>8</b>	<b>Results and Discussions</b>	<b>50</b>
8.1	Landing Page	51
8.2	Login Page	51
8.3	SignUp Form	52
8.4	Grocery List Creation	52
8.5	Display Grocery List	53
8.6	Adding List item	53
8.7	Editing and Deleting List Items	54
8.8	Marking as Done	54
8.9	Downloading Grocery List	55
8.10	All Users Details Page	55
8.11	About Us Page	56
8.12	Contact Page	58
8.13	Dashboard	58
8.14	Change User Details	59
8.15	User Confirmation	60
8.16	Profile Pic Update	60
8.17	Forgot Password	61

8.18	OTP Verification	61
<b>9</b>	<b>Conclusion and Future Work</b>	<b>62</b>
9.1	Conclusion	63
9.2	Limitations	63
9.3	Future Work	64
<b>10</b>	<b>References</b>	<b>65</b>
10.1	Frameworks & Libraries	66
10.2	Tools & Platforms	66
10.3	Learning Resources	67

## List of Tables

Sr. No.	Title	Page No.
2.1	Justification of Technology	17
4.1	Time Estimation	23
6.1	user_data	32
6.2	grocery_lists	32
6.3	list_items	33
6.4	shared_lists	33
7.1	Test Cases	47

# List of Figures

Sr. No.	Title	Page No.
4.1	Gantt Chart	24
6.1	E-R Diagram	34
6.2	E-R Diagram Representation (Textual)	36
6.3	Data Flow Diagram	37
6.4	DFD Level 0	38
6.5	DFD Level 1	39
8.1	Landing Page	51
8.2	Login Page	51
8.3	SignUp Form	52
8.4	Grocery List Creation	52
8.5	Display Grocery List	53
8.6	Adding List item	53
8.7	Editing and Deleting List Items	54
8.8	Marking as Done	54
8.9	Downloading Grocery List	55
8.10	All Users Details Page	55
8.11.A	About Us Page	56
8.11.B		56
8.11.C		57
8.11.D		57
8.11.E		57
8.12	Contact Page	58
8.13.A	Dashboard	58
8.13.B		59
8.14	Change User Details	59
8.15	User Confirmation	60
8.16	Profile Pic Update	60
8.17	Forgot Password	61
8.18	OTP Verification	61

# **Chapter 1**

## **Introduction**

## Chapter 1

### Introduction

The **Groceryz Web Application** is a modern and intuitive platform designed to streamline and simplify the process of grocery shopping. It addresses the common challenges faced by individuals and families in managing grocery lists, ensuring a more efficient, organized, and collaborative shopping experience. Grocery shopping is often a task that leads to confusion and inefficiency, whether due to forgotten items, duplicate purchases, or lack of coordination among household members. The **Groceryz** app aims to resolve these issues by offering a centralized, digital solution for managing grocery lists in a way that enhances productivity and reduces the chances of errors.

In today's fast-paced world, people are increasingly relying on technology to manage daily tasks. While grocery shopping is a routine activity, it is often chaotic, especially when multiple people are involved in the process. Forgetting essentials, purchasing duplicates, or failing to collaborate on shared lists are common problems that individuals face. **Groceryz** provides a simple, user-friendly digital platform where users can create, organize, share, and track their grocery lists easily, all while maintaining security and user access control.

The application is designed to serve a broad audience—from individual users looking for a better way to manage their grocery shopping to families or groups who need to collaborate and coordinate their grocery shopping effectively. It helps users save time, reduce waste, and maintain organized shopping habits. By integrating features such as list sharing, collaborative editing, and secure logins, **Groceryz** makes grocery shopping more efficient and accessible than ever before.

#### 1.1 Background

The development of the **Groceryz Web Application** was driven by the recognition of a gap in the market for a collaborative and efficient tool to manage grocery shopping. Traditional methods of managing grocery lists, such as using paper or basic mobile note apps, often fall short in terms of real-time collaboration, ease of use, and organization. In many households or shared living situations, the responsibility for grocery shopping is divided among multiple people, leading to confusion and inefficiency when coordinating tasks.

## GROCERYZ WEB APP

When multiple people use different devices or systems to manage shopping lists, it becomes easy for items to be duplicated or missed. Additionally, users often struggle with having to recreate grocery lists from scratch every time, even if they frequently buy the same items. There was a need for a digital solution that would not only allow individuals to manage their lists efficiently but also facilitate collaboration among family members, housemates, or other groups.

**Groceryz** was developed to solve these challenges. By providing a cloud-based platform that allows for real-time collaboration, list sharing, and secure user authentication, the application offers a streamlined solution for users to manage their grocery lists efficiently. The app eliminates the inefficiencies associated with traditional methods and addresses the need for a shared system that can be accessed from multiple devices.

### 1.2 Objectives

The **Groceryz Web Application** was developed with the following core objectives:

1. **Streamlining Grocery Shopping:** The primary goal of the **Groceryz** application is to make grocery shopping easier by providing a centralized platform where users can create, manage, and share grocery lists. By transitioning from paper lists or basic note apps to a digital platform, **Groceryz** ensures that users can easily organize and update their lists, reducing time spent on shopping preparation.
2. **Enabling Collaborative Shopping:** One of the main features of **Groceryz** is its ability to allow multiple users to collaborate on a shared list. This is particularly beneficial for families, housemates, or groups of friends who need to coordinate their grocery shopping. Users can share their lists and even grant permission for others to edit or update the list in real-time, making collaboration seamless and reducing the risk of duplication or forgotten items.
3. **Saving Time and Reducing Waste:** The ability to save and reuse grocery lists is a key feature of **Groceryz**, allowing users to reference and reuse their last five grocery lists for future shopping trips. This functionality not only saves time but also helps users avoid unnecessary duplicate purchases, reducing food waste and making grocery shopping more efficient.
4. **Cross-Device Accessibility:** To cater to a wide range of users, **Groceryz** is designed to be accessible across different devices such as smartphones, tablets, and desktops. This ensures that

# GROCERYZ WEB APP

users can manage their grocery lists anytime, anywhere, whether they are at home, in the store, or on the go.

## 1.3 Purpose, Scope, and Applicability

### 1.3.1 Purpose

The **purpose** of the **Groceryz Web Application** is to provide a comprehensive, user-friendly solution for managing grocery lists that addresses the challenges of traditional methods. The app aims to:

1. **Simplify Grocery Shopping:** By providing a digital platform that consolidates all aspects of grocery list management, **Groceryz** reduces the complexity of organizing and tracking shopping needs.
2. **Facilitate Collaboration:** The ability to share and collaborate on lists in real-time ensures that all participants are on the same page, making shopping a more cooperative and efficient experience.
3. **Offer Accessibility:** The application's multi-device compatibility ensures that users can access and update their lists from anywhere, making grocery management convenient and flexible.
4. **Enhance User Experience:** Through intuitive design and features like list saving, editing, and sharing, the app is designed to enhance the overall user experience, making grocery shopping less stressful and more organized.

### 1.3.2 Scope

The **scope** of the **Groceryz Web Application** defines the primary functionalities, features, and limitations of the platform. It includes:

1. **User Registration and Authentication:** Users will be able to sign up, log in securely, and access their accounts. Role-based access will be provided, with **admin** users having full control over list management and normal users being able to create and share lists.

## GROCERYZ WEB APP

2. **List Management:** Users can create, edit, and delete grocery lists, as well as add and remove items. Lists can be marked as complete, and the application will save the last five lists for easy access in future shopping trips.
3. **List Sharing and Collaboration:** Users can share grocery lists with others, granting read-only or full editing access. This feature supports collaboration in real-time, ensuring that all participants can contribute and stay updated.
4. **Cross-Device Compatibility:** The application will be accessible across different devices, ensuring that users can update and manage their grocery lists on smartphones, tablets, or desktops.
5. **Limitations:** The application will focus primarily on grocery list management and will not include advanced features like online grocery ordering or inventory tracking in its initial version.

### 1.3.3 Applicability

The **Groceryz Web Application** is applicable to a wide range of users who need to manage and organize grocery shopping. Its primary target audiences include:

1. **Individuals:** Users who live alone can benefit from the application's easy-to-use interface to track their grocery needs and prevent forgetfulness or buying unnecessary items.
2. **Families:** Families can collaborate on shopping lists, assigning tasks and sharing lists in real-time, ensuring that everyone is on the same page and reducing the chances of mistakes.
3. **Roommates/Shared Households:** In situations where multiple people live together but shop individually, the app can ensure that everyone's needs are met without duplication or missed items.
4. **Small Groups and Friends:** Groups who need to coordinate on shared shopping tasks, such as meal prepping or event planning, can use the app to easily collaborate and share lists.
5. **Busy Professionals:** Individuals with limited time can benefit from the convenience of having their grocery lists organized, shared, and easily accessible, making grocery shopping faster and less stressful.

## GROCERYZ WEB APP

### 1.4 Achievements

**Groceryz** has successfully met several key milestones since its inception:

1. **User-friendly Interface:** The app provides an intuitive design that simplifies grocery list creation and management, making it accessible to users of all skill levels.
2. **Real-time Collaboration:** The list-sharing feature has been well-received, allowing multiple users to update and view lists in real-time.
3. **Cross-Device Accessibility:** The app is accessible on smartphones, tablets, and desktops, offering users flexibility in managing their grocery lists from anywhere.
4. **Improved Efficiency:** Users report a more efficient and organized grocery shopping experience, with fewer mistakes, less duplication, and the ability to reuse saved lists for future trips.

# **Chapter 2**

# **System Analysis**

## Chapter 2

### System Analysis

System analysis is an essential step in the software development process, where the system requirements and components are understood, evaluated, and specified. The goal of system analysis for the **Groceryz Web Application** is to develop an efficient and user-friendly platform for grocery list management, supporting multiple user roles (admin and normal users), data persistence, list sharing, and enhanced security. This section provides a detailed analysis, focusing on the existing system, proposed system, requirements, hardware and software needs, and the justification for selecting technologies.

#### 2.1 Existing System

##### 2.1.1 Current Grocery Management Solutions

Currently, the market offers several grocery list management tools, but they come with significant limitations that hinder user experience and functionality. These include:

- **Basic Functionality:** Most applications provide basic list creation, item addition, and task completion features. However, they often lack advanced functionalities like role-based user management or list sharing with editable permissions.
- **Single-User Systems:** Many existing applications are designed for personal use, meaning only one person can manage the lists. There's no collaborative feature that allows multiple users to interact with the same list, which is especially useful for family or group grocery shopping.
- **Limited Data Persistence:** Many applications do not retain a user's past grocery lists or provide an easy way to access previous lists. Users are required to recreate lists, which can be frustrating and inefficient.
- **Security Concerns:** A number of grocery list applications do not provide secure login mechanisms or encrypted password storage, making users' data vulnerable to attacks.

These limitations highlight the need for a more robust, feature-rich solution with secure login, role-based access, collaborative features, data persistence, and responsive design.

# GROCERYZ WEB APP

## 2.2 Proposed System

### 2.2.1 Features

The **Groceryz Web Application** will introduce a set of advanced features designed to address the issues present in existing systems. The proposed system will have the following core features:

- **User Registration & Authentication:**
  - Users will sign up with personal details such as name, phone number, email ID, and password.
  - Two types of users will be supported:
    - **Admin Users:** Admins will have full control over the application, including editing and managing normal users' details.
    - **Normal Users:** Regular users will have access to their own grocery lists, can create new lists, mark items as completed, and share lists with others.
- **Role-Based Access Control:**
  - **Admins** can access and edit details of normal users, while normal users can only manage their own lists and items.
- **Grocery List Management:**
  - Users will be able to create grocery lists, add items, update the list, and mark items as done.
  - Users will also be able to reorder, delete, and edit individual items.
- **List Sharing:**
  - Users can share their lists with others. Shared lists can be either **view-only** or **editable** depending on the permissions set by the user.
- **Last 5 Lists Storage:**
  - The system will store the last 5 grocery lists for each user, allowing them to quickly access and reuse previous lists.
- **Responsive Design:**
  - The application will be built to be responsive, meaning it will adapt to different screen sizes, from desktop monitors to mobile devices.
- **Security:**

## GROCERYZ WEB APP

- The application will ensure the protection of user data with secure login, password encryption
- **Collaboration:**
  - Users can collaborate in real-time on shared lists, allowing multiple people to add, edit, and mark items.

### 2.2.2 Benefits

- **Enhanced User Experience:** The application will provide a clean, easy-to-use interface that allows users to quickly manage their grocery lists across devices.
- **Collaboration:** By enabling list sharing and collaborative features, multiple users can work together on a shared grocery list.
- **Security:** The use of secure authentication and encrypted passwords will protect user data and prevent unauthorized access.
- **Data Persistence:** Users can always access their last five grocery lists, providing continuity and convenience.
- **Scalability:** The system is designed to scale, allowing more features to be added in the future as the user base grows.

## 2.3 Requirement Analysis

### 2.3.1 Functional Requirements

- **User Authentication:**
  - The system must support user registration with a unique email ID and password.
  - Both normal and admin users should be able to log in securely to access the system.
- **Role Management:**
  - Admin users must be able to view and modify the information of normal users.
  - Normal users should only have access to their personal data and grocery lists.
- **Grocery List Management:**
  - Users must be able to create new lists, add grocery items to the list, and mark items as done.

## **GROCERYZ WEB APP**

- Users should be able to edit the list, reorder items, and delete items.
- **List Sharing:**
  - Users should be able to share their grocery lists with others, with options for editing or viewing only.
- **Data Persistence:**
  - The system must store the last five grocery lists in the database, enabling easy access and reuse.
- **Security:**
  - User passwords should be encrypted using **BCrypt**.
  - The system should implement JWT for secure session management and authentication.

### **2.3.2 Non-Functional Requirements**

- **Performance:**
  - The application should have fast response times, especially for user authentication and CRUD operations (Create, Read, Update, Delete).
- **Scalability:**
  - The system should be able to handle an increasing number of users and grocery lists.
- **Responsiveness:**
  - The web application should be fully responsive and work seamlessly on both desktop and mobile devices.
- **Security:**
  - The application must ensure the protection of user data, particularly personal information and grocery list contents.

## **2.4 Hardware Requirements**

The hardware requirements for the development and deployment of the **Groceryz Web Application** are as follows:

### **2.4.1 Development Environment**

## **GROCERYZ WEB APP**

- **Processor:** Intel Core i5 (or equivalent) or higher to handle development processes effectively.
- **RAM:** 8 GB or more to support IDEs, multiple applications, and testing environments.
- **Storage:** 500 GB of hard disk space (preferably SSD for faster read and write speeds).
- **Operating System:** Windows 10 or higher, macOS, or Ubuntu Linux for a stable development environment.
- **Display:** Full HD (1920x1080) display or higher to facilitate smooth coding and debugging.

### **2.4.2 Deployment Environment (for Hosting the Application)**

- **Processor:** 2-4 CPU cores to handle user requests and background tasks efficiently.
- **RAM:** At least 4 GB to support the operation of the web server, database, and other services.
- **Storage:** 20 GB or more of disk space for storing user data, backups, and logs.
- **Network:** High-speed internet connection (recommended 1 Gbps) for handling incoming and outgoing data efficiently.

## **2.5 Software Requirements**

### **2.5.1 Development Software**

- **IDE / Editor**
  - **Visual Studio Code:** Preferred code editor for development due to its support for PHP, HTML, CSS, and JavaScript, along with its wide range of extensions.
- **Version Control**
  - **Git:** Used for managing source code and tracking changes during development.
  - **GitHub / GitLab:** Platforms for hosting repositories and enabling collaboration.
- **Browser**
  - **Google Chrome, Mozilla Firefox, Microsoft Edge:** Utilized for testing and ensuring cross-browser compatibility.
- **Local Server Environment**

## GROCERYZ WEB APP

- **XAMPP:** An open-source PHP development environment that includes Apache server, MySQL database, and PHP interpreter. It is used for running and testing the application locally.

### 2.5.1.1 Backend

- **PHP:**

Core server-side scripting language used to manage business logic, handle user sessions, process form data, and interact with the MySQL database.

- **MySQL:**

Relational database used to store and manage user information, grocery lists, shared permissions, and other application data.

- **BCrypt:**

Integrated via PHP's `password_hash()` and `password_verify()` functions for secure password encryption and verification.

- **JWT (JSON Web Tokens) (Optional):**

May be implemented for secure, token-based authentication and stateless session handling in future enhancements.

### 2.5.1.2 Frontend

- **HTML5 & CSS3:**

Used to create the structure and design of the web pages.

- **JavaScript (Vanilla + Modern):**

Utilized for client-side interactivity, form validations, dynamic list management, and asynchronous communication using the **Fetch API**.

## 2.6 Justification of Selection of Technology

### 2.6.1 Frontend

For the frontend development of **Groceryz**, the following technologies were chosen due to their reliability, ease of use, and strong community support:

## GROCERYZ WEB APP

- **HTML5**

HTML5 was used to create the structural layout of the web pages. It provides semantic elements and modern features that help in creating clean and accessible page structures, making the application both user- and developer-friendly.

- **CSS3**

CSS3 was selected for styling and visual enhancement of the application. It enables responsive design, improved layout control, and modern styling features, allowing the interface to be both visually appealing and mobile-friendly.

- **Vanilla JavaScript**

Vanilla JavaScript (pure JavaScript without any libraries or frameworks) was used to handle all client-side logic and interactivity. It enables features like form validation, dynamic list updates, and user interaction without the overhead of external libraries.

- **Modern JavaScript (ES6+)**

Features like let/const, arrow functions, and especially the **Fetch API** were used to write cleaner, more maintainable code and handle asynchronous data exchange with the backend efficiently.

### 2.6.2 Backend

The backend technologies for **Groceryz** were selected based on their simplicity, performance, and compatibility with the project's requirements:

- **PHP**

PHP was chosen as the core backend scripting language due to its ease of integration with HTML, support for session management, and strong compatibility with MySQL. It efficiently handles business logic, processes user inputs, manages authentication, and communicates with the database.

- **MySQL**

MySQL, a widely-used relational database management system, was selected to store user accounts, grocery lists, shared lists, and item details. It offers excellent support for structured data, reliable performance, and seamless integration with PHP.

## GROCERYZ WEB APP

- **XAMPP**

XAMPP was used as the local development environment because it bundles Apache, PHP, and MySQL into a single easy-to-use package. It simplifies the setup process and enables testing the full application locally before deployment.

- **BCrypt (via PHP's password\_hash() and password\_verify())**

These built-in PHP functions were used to securely hash and verify user passwords, ensuring safe storage and protection against unauthorized access.

- **JSON & Fetch API Integration**

Backend scripts communicate with the frontend using JSON responses, allowing asynchronous data updates via the Fetch API. This improves user experience by eliminating the need to reload the page after every action.

### 2.6.3 Database

For the **Groceryz** application, **MySQL** was selected as the database management system due to its reliability, scalability, and ease of use in web-based applications. The decision was based on the following factors:

- **Structured Data Storage**

MySQL is a relational database, making it ideal for storing structured data such as users, grocery lists, and individual items. Relationships between tables can be easily defined using primary and foreign keys, which is essential for maintaining data integrity in a multi-user system.

- **Seamless Integration with PHP**

MySQL works seamlessly with PHP, enabling efficient data manipulation and retrieval through SQL queries within backend scripts. This integration simplifies operations such as user authentication, list management, and item updates.

- **Security Features**

MySQL supports user access control, allowing specific permissions to be set for different operations. Combined with PHP's password hashing, this helps protect sensitive user information.

## GROCERYZ WEB APP

- **Performance and Scalability**

MySQL is known for its fast query processing and ability to handle large datasets. This makes it suitable for applications like Groceryz that store and retrieve multiple grocery lists, items, and shared user access.

- **Community Support and Tools**

Being one of the most popular open-source databases, MySQL offers rich documentation and a wide range of tools for administration, debugging, and performance monitoring.

### 2.6.4 Security

Security was a key consideration in the development of the **Groceryz** web application to ensure safe handling of user data and prevent unauthorized access. The following technologies and practices were implemented:

- **Password Hashing (BCrypt)**

User passwords are never stored in plain text. Instead, they are encrypted using PHP's built-in `password_hash()` function, which applies the **BCrypt** algorithm. During login, `password_verify()` is used to securely validate the user's credentials. This protects sensitive data even in the event of a database breach.

- **Session Management**

PHP sessions are used to securely maintain user login states. Session variables ensure that only authenticated users can access certain features, such as creating or editing grocery lists.

- **Input Validation and Sanitization**

All user inputs are validated and sanitized using PHP functions like `htmlspecialchars()` and `mysqli_real_escape_string()` to prevent **SQL Injection**, **Cross-Site Scripting (XSS)**, and other common attacks.

- **Secure Communication with Backend (Fetch API)**

Modern JavaScript (Fetch API) is used for sending and receiving data between the frontend and backend. Sensitive operations are handled via secure POST requests rather than exposing them in URLs.

# GROCERYZ WEB APP

- **User Access Control**

The system includes two roles: **Admin** and **Normal User**. Access to specific features is role-based, preventing unauthorized users from accessing or modifying restricted data.

- **Future Scope: JSON Web Tokens (JWT)**

JWT can be integrated in the future to support stateless and scalable authentication, especially for mobile or API-based versions of the app.

**Table 2.1: Justification of Technology**

CATEGORY	TECHNOLOGY	PURPOSE & JUSTIFICATION
FRONTEND	<b>HTML5</b>	Provides the structure of the web pages using semantic and accessible markup.
	<b>CSS3</b>	Used for styling the UI, enabling responsive and visually appealing design
	<b>JavaScript (Vanilla)</b>	Adds interactivity, handles DOM manipulation and client-side validation without external libraries.
	<b>Modern JS (Fetch API)</b>	Enables asynchronous data transfer between frontend and backend, improving user experience by avoiding full page reloads
BACKEND	<b>PHP</b>	Handles business logic, form processing, user authentication, and communicates with the MySQL database.
DATABASE	<b>MySQL</b>	Stores user accounts, grocery lists, items, and shared access data using structured and relational formats.
DEVELOPMENT TOOL	<b>XAMPP</b>	Local server environment that includes Apache, PHP, and MySQL—ideal for development and testing
SECURITY	<b>BCrypt (password_hash())</b>	Used for hashing user passwords to ensure secure storage and prevent unauthorized access
	<b>PHP Sessions</b>	Manages user sessions to maintain authentication states securely across pages.
VER	<b>Git &amp; GitHub</b>	Tracks code changes, supports collaboration, and maintains version history of the project.
Testing/Debugging	<b>Google Chrome, Firefox, Edge</b>	Used to ensure cross-browser compatibility and consistent behavior across different platforms.

# **Chapter 3**

# **Research Methodologies**

## Chapter 3

### Research Methodologies

For the **Groceryz** grocery list organizer application, implementing targeted research methodologies provides valuable insights into user preferences, feature needs, and usability. Here's a breakdown of specific methodologies suited to the development and refinement of Groceryz:

#### 3.1 Questionnaire

- **Purpose:** To gather input on shopping habits, feature priorities, and pain points users face in managing grocery lists.
- **Design:**
  - Create questions focused on users' current grocery management techniques, their interest in features like list sharing, item marking, and accessibility of recent lists.
  - Use a mix of multiple-choice questions, Likert scale ratings (e.g., for ease of use or feature importance), and open-ended questions for suggestions.
- **Sample Questions:**
  - “How frequently do you use grocery list apps?”
  - “Which features are essential for you in a grocery organizer app? (e.g., list sharing, multiple lists, reminders)”
  - “How important is it for you to access past grocery lists?”
- **Outcome:** Insights into the most desired features and user habits, which help refine Groceryz's design and feature set.

#### 3.2 Sampling

- **Purpose:** To choose a representative group from the target audience to gather relevant, scalable feedback.
- **Approach:**
  - **Stratified Sampling:** Organize participants by demographics such as shopping frequency, household size, and age to capture a diverse user base.
  - **Convenience Sampling:** Reach users through social media, local communities, or

## GROCERYZ WEB APP

online grocery forums to access potential users.

- **Sample Size:** Aim for a group of 100-200 participants to ensure diversity and actionable insights while maintaining a manageable data set.
- **Outcome:** A representative sample that reflects diverse shopping habits and app preferences, providing balanced input for feature prioritization.

### 3.3 Data Analysis

- **Purpose:** To process and interpret data from surveys and feedback sessions, helping identify user preferences.
- **Methods:**
  - **Quantitative Analysis:** Calculate statistics such as feature prioritization percentages or average ratings for app features.
  - **Qualitative Analysis:** Group open-ended responses into common themes, like challenges in managing lists or desired sharing features.
- **Tools:** Utilize Excel or Google Sheets for basic data analysis or software like Tableau for advanced visualizations.
- **Outcome:** Data-driven insights that highlight feature priorities and common user challenges, guiding development focus.

### 3.4 Pie Charts

- **Purpose:** To visually represent proportions of user feedback, making it easier to see dominant trends and preferences.
- **Application:**
  - **Feature Preferences:** Show the breakdown of features users find most important (e.g., list sharing, multi-user access).
  - **Demographic Insights:** Visualize user demographics, such as age or shopping frequency, to understand primary user groups.
- **Outcome:** Clear, visual representation of user preferences, simplifying the decision-making process.

### 3.5 Bar Charts

- **Purpose:** To compare specific data points, making it easy to see differences between user needs or feature importance.
- **Application:**
  - Compare the popularity of different functionalities (e.g., item marking, multi-list management) across different user demographics.
  - Show trends in desired user permissions, such as sharing lists with edit capabilities.
- **Outcome:** Quick visual comparisons that help prioritize features for development.

# **Chapter 4**

## **Estimation & Planning**

## Chapter 4

### Estimation & Planning

The development of **Groceryz – Grocery List Organiser Web App** was strategically planned and executed over a period of **9 months** from **15th July to 18th April**. The project was divided into multiple phases, ensuring smooth progress and timely completion of tasks.

#### 4.1 Time Estimation

Phase	Duration	Timeline	Key Activities
Requirement Analysis	2 weeks	15th Jul – 29th Jul	Identifying goals, core features, target users
System Design & Planning	2 weeks	30th Jul – 12th Aug	E-R diagram, DB schema, UI wireframes
Frontend Development	6 weeks	13th Aug – 23rd Sep	HTML, CSS, JS for all main pages
Backend Development	8 weeks	24th Sep – 18th Nov	PHP, MySQL, list/item/user logic
Core Feature Implementation	4 weeks	19th Nov – 17th Dec	Sharing, rename, edit, download
Integration & Debugging	6 weeks	18th Dec – 28th Jan	Linking frontend with backend, form validation
Testing & Enhancements	8 weeks	29th Jan – 24th Mar	Bug fixes, feature tuning, UI adjustments
Finalization & Review	3 weeks	25th Mar – 15th Apr	Report writing, documentation, screenshots

Table 4.1: Time Estimation

# GROCERYZ WEB APP

## 4.2 Gantt Chart

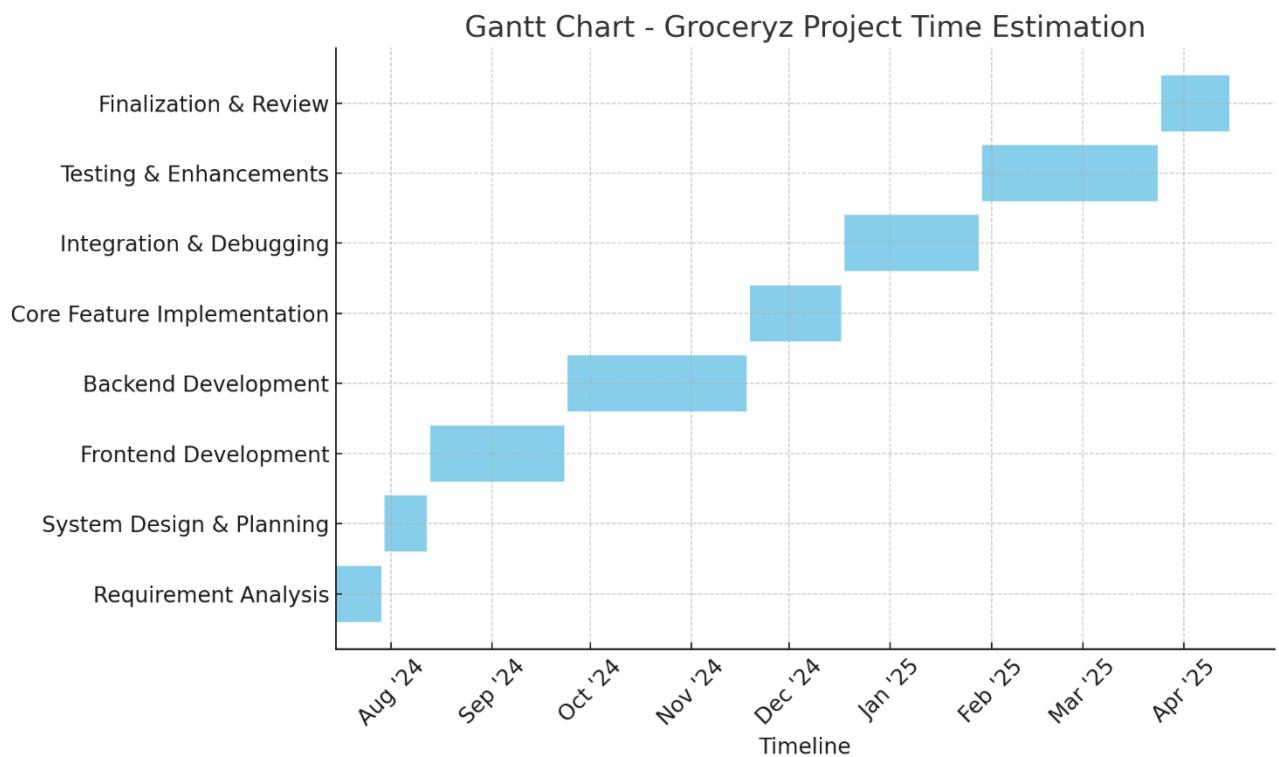


Fig. 4.1: Gantt Chart

## 4.3 Resource Estimation

- **Team Members:** 1 (Individual project)
- **Technologies Used:**
  - HTML, CSS, JavaScript
  - PHP (Procedural)
  - MySQL
- **Tools:**
  - Visual Studio Code, XAMPP, DB Browser, Canva
- **Hardware Requirements:**
  - Laptop with 8GB+ RAM, local server setup
- **Other Resources:**
  - Online references, GitHub backups, university mentor guidance

### 4.4 Planning Strategy

- Adopted a **phased development model** with weekly reviews
- **Manual version control** and backups for each module
- **Milestone tracking** using personal Kanban board (To-Do, Doing, Done)
- Testing conducted **after every major feature** to ensure stability
- Final rounds included **user feedback & real-use simulations**

# **Chapter 5**

# **Methodology**

## Chapter 5

### Methodology

To ensure a structured approach and successful project delivery, the **Iterative Development Model** was adopted for the Groceryz web application. This model allowed the project to be developed and improved through multiple cycles of planning, coding, testing, and refining.

#### 5.1 Iterative Development Approach

This model was chosen due to its flexibility in incorporating changes and feedback during development. Each iteration focused on building a part of the application, which was reviewed and refined before moving on to the next.

#### 5.2 Development Phases

##### 5.2.1 Requirement Analysis

- Identified the core problems users face while managing grocery lists.
- Finalized the must-have features like item addition, list sharing, editing, and download options.

##### 5.2.2 System Design

- Designed database schema and relationships using an E-R Diagram.
- Planned UI wireframes for both user and admin interfaces.
- Decided on technology stack: HTML, CSS, JavaScript, PHP (procedural), and MySQL.

##### 5.2.3 Frontend Development

- Developed responsive interfaces for user login/signup, list creation, item management, and admin dashboard.
- Used JavaScript to add dynamic list management features.

## GROCERYZ WEB APP

### 5.2.4 Backend Development

- Wrote PHP scripts for login authentication, session management, item CRUD, list sharing, and user-specific list handling.
- Connected frontend forms with MySQL database using procedural PHP.

### 5.2.5 Integration

- Combined all modules and ensured smooth data flow between frontend and backend.
- Ensured shared lists were editable and secure.

### 5.2.6 Testing and Debugging

- Performed unit testing on all major functions.
- Resolved issues related to form validation, item display, and session handling.
- Ensured secure data operations and proper access control.

### 5.2.7 Finalization

- Polished UI, added confirmation messages, download functionality (TXT, PDF, Excel), and ensured mobile compatibility.
- Documented all the features and prepared for report submission.

## 5.3 Tools & Technologies Used

- **Frontend:** HTML, CSS, JavaScript
- **Backend:** PHP (procedural)
- **Database:** MySQL
- **Development Tools:** Visual Studio Code, XAMPP, DB Browser
- **Design Tools:** Canva, Figma (for UI design and diagrams)

# **Chapter 6**

# **System Design**

## Chapter 6

### System Design

The system design of the **Groceryz Web Application** outlines the structure, components, data storage, and flow of information within the application. This includes dividing the system into functional modules, defining data storage elements (Data Dictionary), illustrating relationships (E-R Diagram), and modeling the flow of data (Data Flow Diagrams) and system structure (UML Diagrams).

#### 6.1 Module Division

To achieve a well-organized, maintainable application, the Groceryz Web Application is divided into core modules, each responsible for distinct functionalities. Here's a detailed breakdown of each module and its role:

##### Modules and Responsibilities

###### 1. User Management Module:

- **Description:** Manages user registration, authentication, and profile management.
- **Primary Functions:**
  - **Register:** Captures new users' details (name, email, phone, password) and stores them securely.
  - **Login:** Validates user credentials and establishes a session upon successful authentication.
  - **Profile Management:** Allows users to update profile information and manage their passwords.
- **Special Considerations:** Includes password hashing and encryption to secure sensitive data.

###### 2. Grocery List Management Module:

- **Description:** Handles the creation, update, and deletion of grocery lists.
- **Primary Functions:**
  - **Create List:** Allows users to create a new grocery list with a title and optional description.

# GROCERYZ WEB APP

- **Update List:** Permits users to edit list details such as the title and description.
- **Delete List:** Enables users to delete lists they no longer need.
- **Retrieve Recent Lists:** Stores and retrieves the last five grocery lists per user.
- **Special Considerations:** Manages access to each list so that only authorized users can view or edit.

## 3. Item Management Module:

- **Description:** Manages grocery items within lists, including item addition, updates, and completion status.
- **Primary Functions:**
  - **Add Item:** Allows users to add new items with attributes like item name, quantity, and optional notes.
  - **Edit Item:** Permits users to update item details.
  - **Mark Item as Done:** Enables users to mark items as completed.
  - **Delete Item:** Allows users to remove items from a list.
- **Special Considerations:** Tracks the status of each item to differentiate between completed and pending items.

## 4. Sharing and Permissions Module:

- **Description:** Manages the sharing of lists between users and defines access permissions.
- **Primary Functions:**
  - **Share List:** Allows users to share lists with other users by setting view or edit permissions.
  - **Edit Permissions:** Enables list owners to adjust sharing permissions.
- **Special Considerations:** Role-based access control ensures that permissions are enforced for shared users.

## 5. Database Management Module:

- **Description:** Stores and retrieves data for users, lists, items, and permissions.
- **Primary Functions:**
  - **Data Storage:** Central repository for user and list data, item details, and permissions.
  - **Data Integrity:** Enforces relational constraints to maintain data accuracy and consistency.

# GROCERYZ WEB APP

## 6.2 Data Dictionary

The **Data Dictionary** defines the structure, data types, and purpose of each field within the database tables used by the Groceryz Web Application. This application consists of several main tables. Each table stores data critical to the application's functionality, including user details, grocery lists, individual list items, and permissions for shared access.

**Table 6.1: user\_data Table**

Field	Type	Description
<b>id</b>	int(255)	Unique user ID (Primary Key)
<b>name</b>	varchar(100)	Full name of the user
<b>phone_no</b>	bigint(10)	Phone number (unique)
<b>email_id</b>	varchar(150)	Email ID (unique)
<b>role</b>	varchar(50)	Role of the user (admin/user)
<b>password</b>	varchar(255)	Hashed password
<b>profile_pic</b>	varchar(255)	URL or file path to profile picture
<b>created_at</b>	timestamp	User registration timestamp

**Table 6.2: grocery\_lists Table**

Field	Type	Description
<b>list_id</b>	int(11)	Unique ID for each list (Primary Key)
<b>user_id</b>	int(11)	Foreign Key from user_data
<b>list_name</b>	varchar(255)	Name of the grocery list
<b>shared</b>	tinyint(1)	1 if list is shared, 0 otherwise
<b>shared_with</b>	varchar(255)	(Deprecated by shared_lists)
<b>shared_permissions</b>	varchar(255)	Permissions (deprecated)
<b>created_at</b>	timestamp	List creation timestamp
<b>updated_at</b>	timestamp	Last updated timestamp

## GROCERYZ WEB APP

**Table 6.3: list\_items Table**

<b>Field</b>	<b>Type</b>	<b>Description</b>
<b>item_id</b>	int(11)	Unique ID for item (Primary Key)
<b>user_id</b>	int(11)	Foreign Key from user_data
<b>list_id</b>	int(11)	Foreign Key from grocery_lists
<b>item_name</b>	varchar(255)	Name of the item
<b>item_type</b>	varchar(100)	Type/category (e.g., Dairy, Fruits)
<b>item_qty</b>	int(11)	Quantity
<b>is_done</b>	tinyint(1)	1 if completed, 0 otherwise
<b>created_at</b>	timestamp	Item addition timestamp
<b>updated_at</b>	timestamp	Last modification timestamp

**Table 6.4: shared\_lists Table**

<b>Field</b>	<b>Type</b>	<b>Description</b>
<b>id</b>	int(11)	Unique record ID (Primary Key)
<b>list_id</b>	int(11)	Foreign Key from grocery_lists
<b>shared_with_user_id</b>	int(11)	Foreign Key from user_data (receiver)
<b>shared_by_user_id</b>	int(11)	Foreign Key from user_data (owner)
<b>permission</b>	varchar(10)	view or edit access

# GROCERYZ WEB APP

## 6.3 E-R Diagram

The Entity-Relationship (E-R) Diagram represents the logical structure of the database used in the Groceryz web application. It helps in understanding how data is connected and organized within the system.

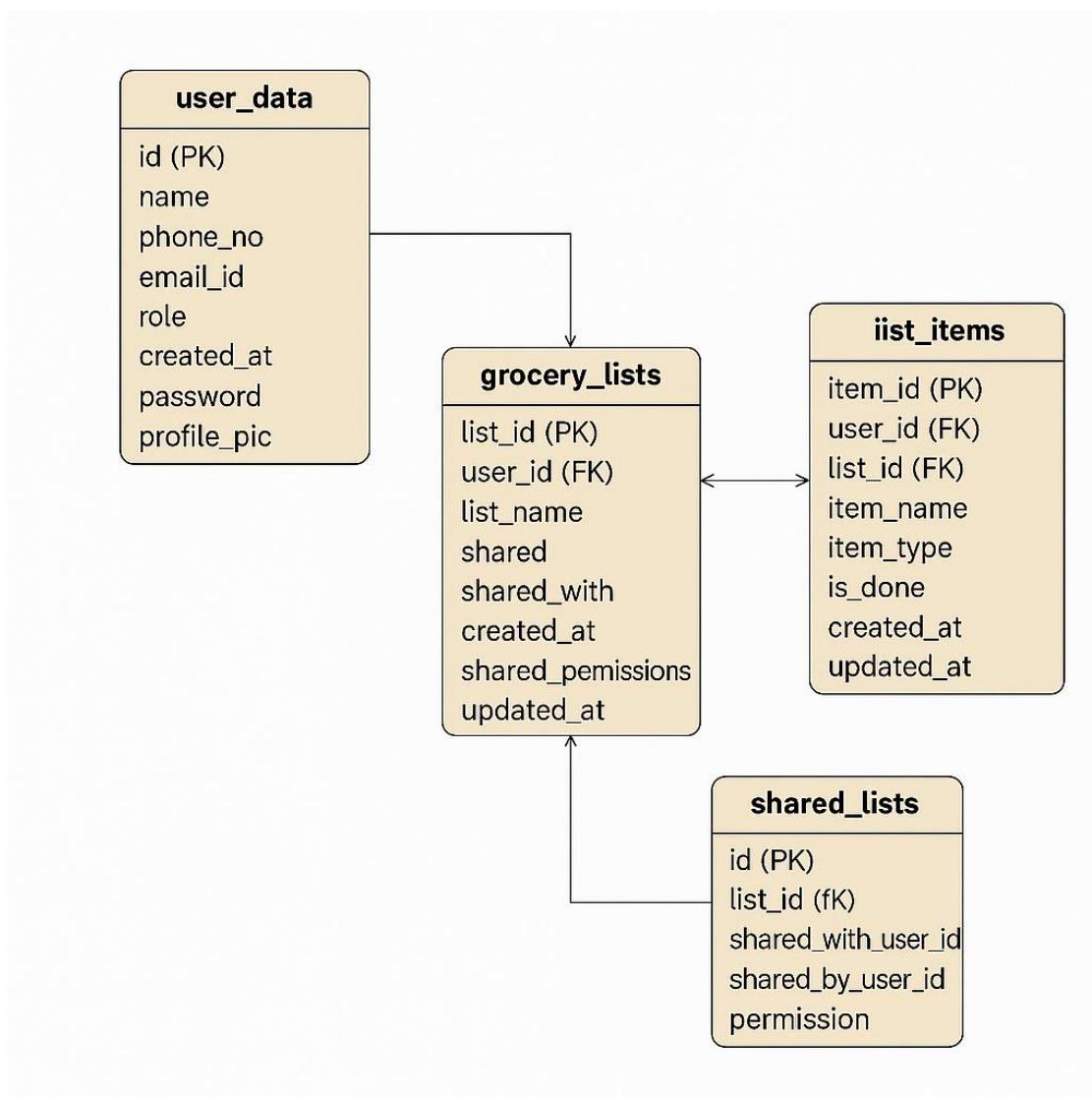


Fig. 6.1: E-R Diagram

### 6.3.1 Entities and Relationships

#### Entities:

##### 1. User

- Attributes: id, name, phone\_no, email\_id, password, profile\_pic, role, created\_at

# GROCERYZ WEB APP

## 2. Grocery List

- Attributes: list\_id, user\_id, list\_name, shared, created\_at, updated\_at

## 3. List Item

- Attributes: item\_id, list\_id, user\_id, item\_name, item\_type, item\_qty, is\_done, created\_at, updated\_at

## 4. Shared List

- Attributes: id, list\_id, shared\_by\_user\_id, shared\_with\_user\_id, permission

### Relationships:

- A **User** can create multiple **Grocery Lists** (One-to-Many)
- A **Grocery List** contains multiple **List Items** (One-to-Many)
- A **User** can share a **Grocery List** with another **User** (Many-to-Many via shared\_lists)
- Each entry in **Shared Lists** maps a **Grocery List** shared between two users with defined **permissions**

### 6.3.2 Diagram Structure

The structure of the E-R diagram follows these key connections:

- **user\_data.id** → **grocery\_lists.user\_id**
- **grocery\_lists.list\_id** → **list\_items.list\_id**
- **user\_data.id** → **list\_items.user\_id**
- **shared\_lists.list\_id** → **grocery\_lists.list\_id**
- **shared\_lists.shared\_by\_user\_id** and **shared\_with\_user\_id** → **user\_data.id**

### 6.3.3 E-R Diagram Representation (Textual)

## GROCERYZ WEB APP

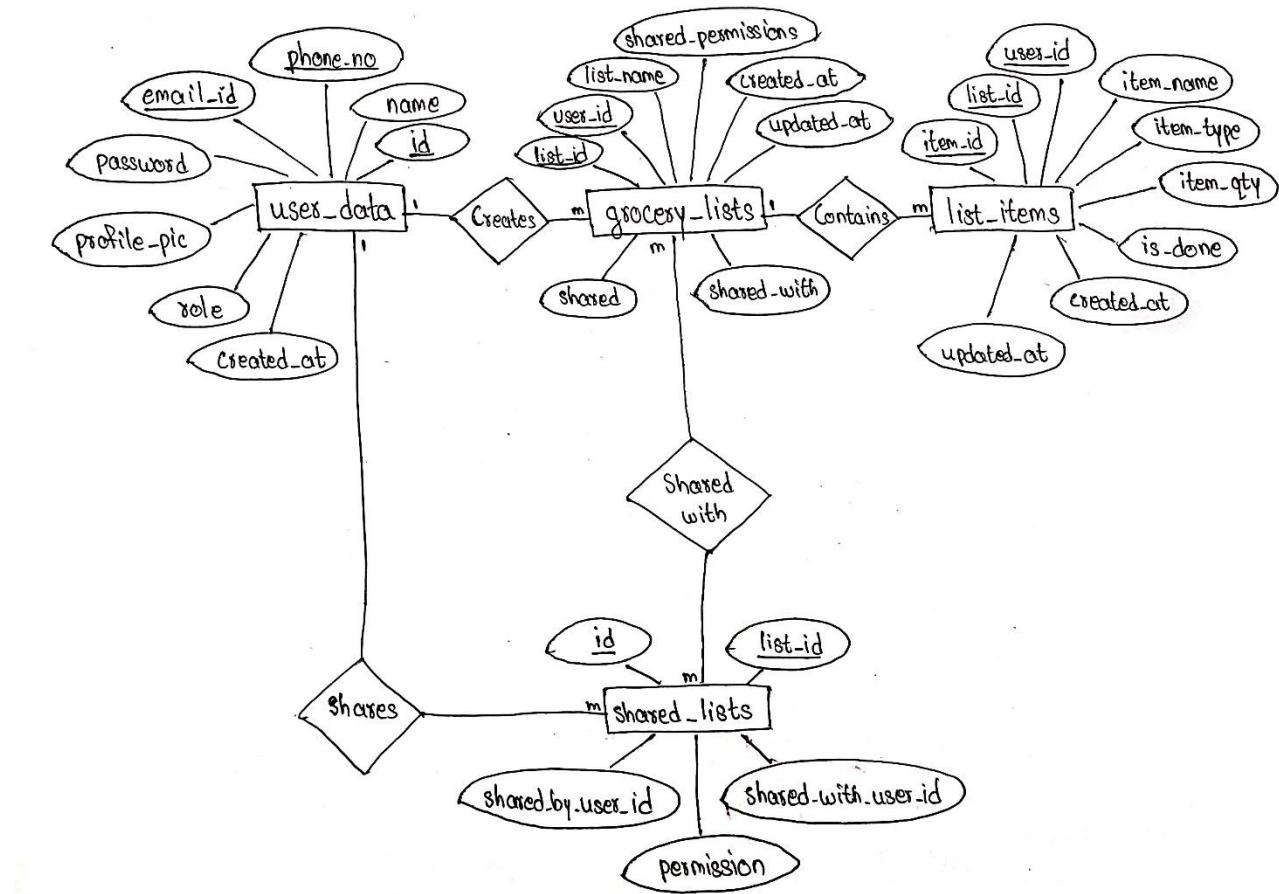


Fig. 6.2: E-R Diagram Representation (Textual)

### Notes:

- **Primary Key (PK):** Uniquely identifies a record in the table.
- **Foreign Key (FK):** Establishes a link between two tables.
- **One-to-Many (1 —————< m):** One user can create many grocery lists, and one list can have many items.
- **Many-to-Many (m —————< m):** Lists shared between multiple users using the shared\_lists bridge table.

## 6.4 Data Flow Diagrams (DFDs)

Data Flow Diagrams (DFDs) represent how data moves within the system. They focus on the processes, external entities, data stores, and the flow of data between them. The DFDs help in visualizing the functional flow of the application.

# GROCERYZ WEB APP

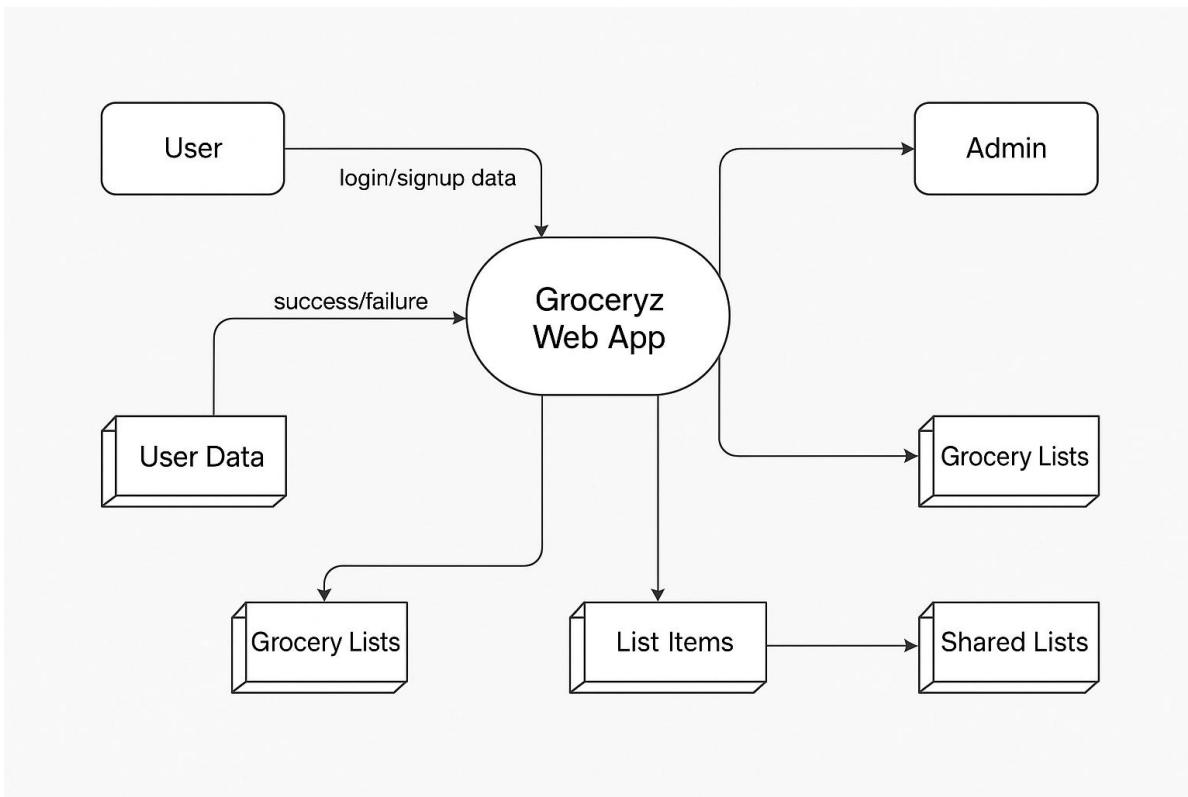


Fig. 6.3: Data Flow Diagram

## 6.4.1 DFD Level 0 – Context Level Diagram

### Purpose:

Provides a high-level overview of the entire system and its interaction with external entities.

### External Entities:

- **User**
- **Admin**

### Main Process:

- **Groceryz Web App**

### Data Stores:

- User Data
- Grocery Lists
- List Items
- Shared Lists

# GROCERYZ WEB APP

## Data Flows:

- User provides login/signup data → Groceryz System
- Groceryz returns success/failure → User
- User performs actions like create/edit/delete lists and items
- System updates corresponding data stores

## Diagram Representation (Textual):

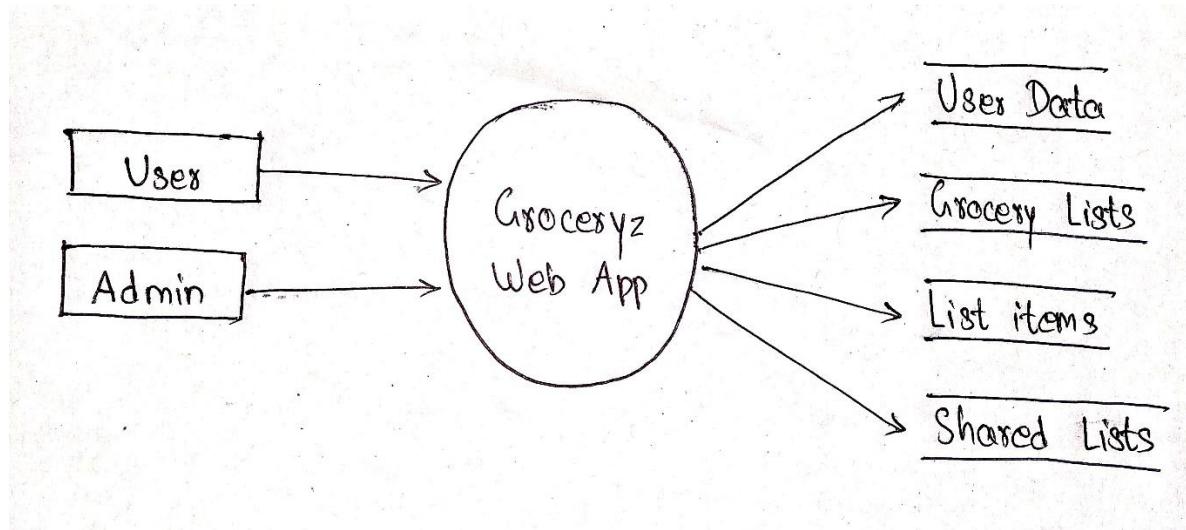


Fig. 6.4: DFD Level 0

## 6.4.2 DFD Level 1 – Main Functional Decomposition

This level breaks down the main system into sub-processes.

### Processes:

1. **User Authentication**
2. **Manage Grocery Lists**
3. **Manage List Items**
4. **Share Lists**

### Data Stores:

- D1: User Data
- D2: Grocery Lists
- D3: List Items
- D4: Shared Lists

# GROCERYZ WEB APP

## External Entities:

- User
- Admin

## Process Breakdown:

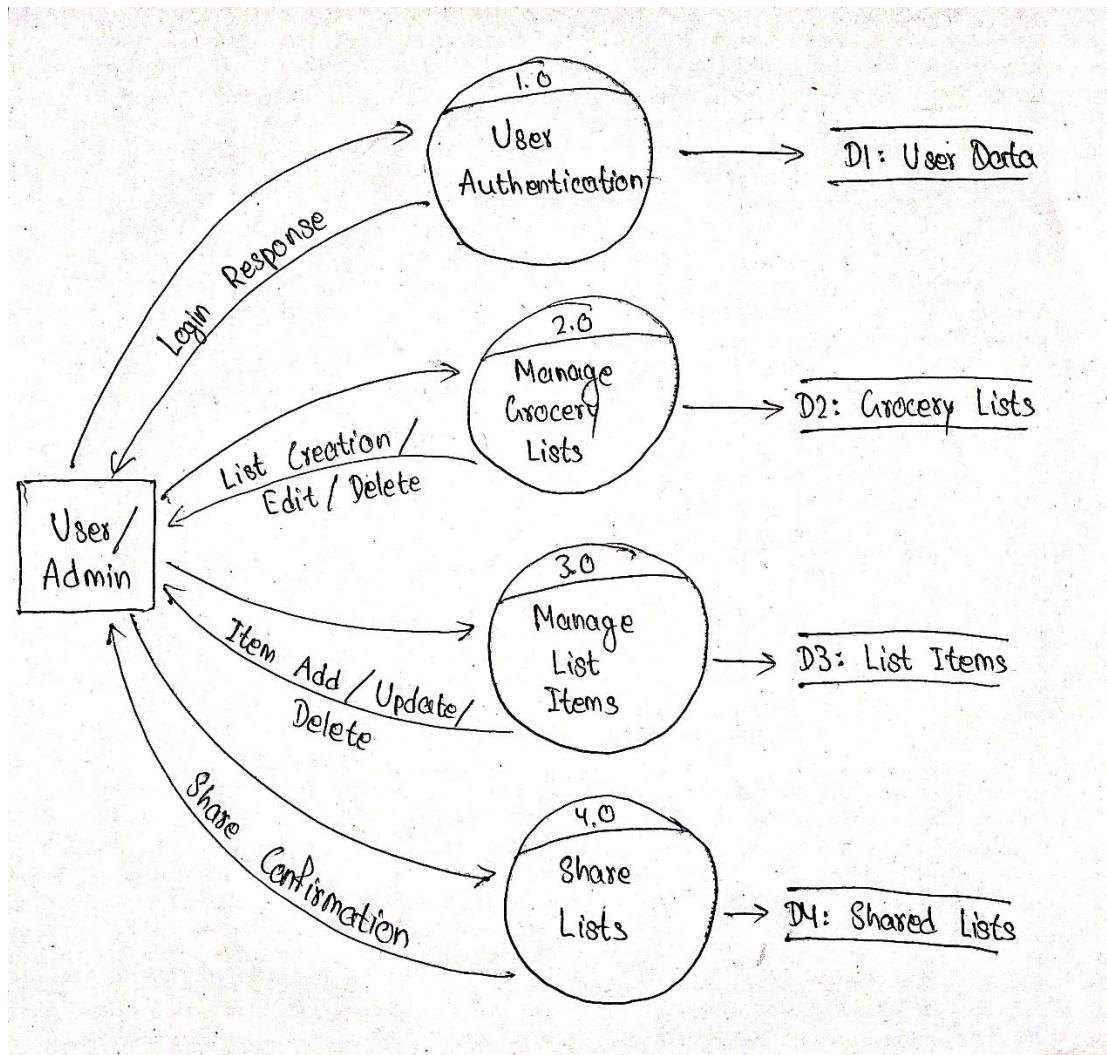


Fig. 6.5: DFD Level 1

## **Chapter 7**

# **Implementation and Testing**

## Chapter 7

### Implementation and Testing

#### 7.1 Code (Place Core segments)

In the **coding phase** of the **Groceryz** project, the focus was on developing both the **frontend** and **backend** components that would bring the web application to life. The objective was to ensure that all the required functionalities, from user authentication to item management, were implemented efficiently and effectively.

##### 7.1.1 Frontend Development

The frontend development of the **Groceryz** web application was centered on creating a user-friendly interface and providing smooth interactivity for managing grocery lists. The frontend was built using **HTML**, **CSS**, and **JavaScript** to deliver a seamless experience.

- **HTML (Structure):**

- Used to create the basic structure of each page. This includes the **Admin Home Page**, where admins can manage grocery lists and items, and the **User Dashboard**, where users can access their personal grocery lists.
- Forms were created for adding new items, renaming lists, and creating new lists. Modals and pop-ups were used to simplify these interactions without navigating away from the page.
- A dynamic table was implemented to display the grocery items in lists, providing options for item management such as edit, delete, and mark as done.

- **CSS (Styling):**

- Designed with a green theme to keep the app visually appealing and consistent with the nature of grocery shopping. The green theme was applied throughout the app, with color choices indicating different actions (e.g., green for adding items, red for deleting).
- The pages were designed to be responsive, adapting to different screen sizes and

## GROCERYZ WEB APP

devices. Flexbox and grid layouts were used to create flexible page layouts.

- Hover and focus effects were added for better interaction, such as highlighting menu items when hovered over or clicked.

- **JavaScript (Interactivity):**

- JavaScript was crucial for adding interactive elements to the app. For instance, when users add or delete items from the list, JavaScript dynamically updates the page without needing a page reload.
- **AJAX** was used to make asynchronous requests to the backend to fetch and update data, such as displaying the list of items without refreshing the page.
- JavaScript was also used to handle **list downloads** in various formats (TXT, PDF, Excel). Users could export their grocery lists using a simple button click, which would trigger a script to generate the downloadable file.

### 7.1.2 Backend Development

The backend of the **Groceryz** web application was built using **PHP** for the server-side logic and **MySQL** for database management. The backend was responsible for handling the application's core functionalities, including user authentication, item management, and list sharing.

- **PHP (Server-Side Logic):**

- **Procedural PHP** was employed for all server-side scripting. This was consistent with the project's design requirements to avoid object-oriented programming (OOP). Functions were written to handle specific tasks such as adding new grocery lists, adding items to lists, updating item status, and deleting items.
- PHP handled **user authentication** by validating login credentials against the database and establishing secure sessions. For added security, password hashing was used, and sessions were managed to restrict access to specific areas of the application based on user roles (user or admin).
- Each action, such as creating a list, adding an item, or renaming a list, was connected to an appropriate PHP file that processed the request, communicated with the database,

## GROCERYZ WEB APP

and returned a response (such as success or failure).

- PHP was also used to process **email OTPs** for password recovery. When users requested a password reset, an OTP was sent to their registered email, and they could use this OTP to reset their password.

- **MySQL (Database Management):**

- The database was structured using **MySQL** with multiple interconnected tables. These tables were used to store different types of data:
  - **user\_data:** Contains user information such as usernames, email addresses, and password hashes.
  - **grocery\_lists:** Stores information about each user's grocery lists, including the list name, creation date, and list creator.
  - **list\_items:** Each item added to a list is stored here, with fields for item name, quantity, status (e.g., done or pending), and its associated list ID.
  - **shared\_lists:** Tracks shared access for grocery lists, allowing users to collaborate on lists. This table stores which users have shared access to a specific list.
- PHP interacted with the MySQL database by sending SQL queries to perform actions such as creating new lists, retrieving list data, updating item status, and deleting items. These operations were wrapped in functions, making them reusable and modular.

### 7.1.3 Core Functionalities Implemented

The core functionalities were crucial to the success of the **Groceryz** app and were developed step by step to ensure seamless integration.

- **User Authentication:**

- **Login/Signup:** Users could create accounts and log in securely. PHP was used to process user data, validate credentials, and create sessions. Passwords were hashed using `password_hash()` for security.

## GROCERYZ WEB APP

- **Password Recovery:** When users forgot their passwords, an OTP was sent to their email address. The user could then enter the OTP to reset their password. This feature was implemented using PHP mail functions to send OTPs.

- **Grocery List Management:**

- **List Creation:** Admins could create new lists and assign them to users. The `createList.php` script handled this action by inserting a new record into the **grocery\_lists** table.
- **Adding and Managing Items:** Users could add items to their lists using the `addItem.php` script, and these items were stored in the **list\_items** table. Admins had the ability to rename lists and delete or modify items.
- **Marking Items as Done:** The system allowed users to mark items as "done" once purchased. This was managed using a status field in the **list\_items** table, and updates were sent to the database via PHP scripts.

- **List Sharing:**

- **Sharing Lists:** Users could share grocery lists with others. This was implemented via the **shared\_lists** table, where entries were created linking a list with the user who had shared access to it.
- **Permissions:** Admins had full control over the lists, while regular users had limited permissions, such as viewing and editing items within the lists shared with them.

### 7.1.4 Session Management

The system ensured that only authenticated users and admins could access certain pages. PHP sessions were used to validate user login status, and access was restricted to specific pages like the admin dashboard or user profile if the user was not logged in.

- **Session Validation:** Before accessing any restricted page, PHP checked whether the session variable (representing the logged-in user) was set. If not, the user was redirected to the login page.

## 7.2 Testing Approach

Testing ensures that the system functions correctly and meets the desired requirements. For the *Groceryz Web App*, two major testing strategies are applied: Unit Testing and Integration Testing.

### 7.2.1 Unit Testing

#### 7.2.1.1 What is Unit Testing?

**Unit Testing** is the process of individually testing the **smallest testable parts of a software application**, known as *units*, to ensure that each part functions correctly on its own. A unit can be a function, method, procedure, or a module in the code.

It is typically performed by developers during the development phase, **before integration** with other parts of the system. The main goal of unit testing is to **verify the correctness of isolated pieces of logic**.

#### 7.2.1.1.1 Purpose of Unit Testing

- To ensure that **each unit of the software** performs as expected.
- To catch bugs early in the development process.
- To **reduce the cost of fixing errors** later in the software lifecycle.
- To improve **code reliability, maintainability, and quality**.

#### 7.2.1.1.2 How Unit Testing Works

Each unit is tested separately by:

1. Providing **known input**.
2. Checking if the **actual output matches the expected output**.
3. If there's a mismatch, the unit fails and the issue is resolved before proceeding.

#### 7.2.1.1.3 Advantages of Unit Testing

- **Improved Accuracy:** Helped validate each core function independently.
- **Faster Debugging:** Easy to isolate issues within small blocks of code.
- **Database Reliability:** Ensured correct data was stored or updated.
- **Frontend Validation:** Helped ensure a smooth user experience through JS testing.

## GROCERYZ WEB APP

### 7.2.1.2 How It Was Applied

- **Frontend:**
  - JavaScript functions like addItemToTable(), validateForm(), etc. were tested independently.
  - Input validations (e.g., checking if all fields are filled before submission).
- **Backend:**
  - Scripts like createList.php, addItem.php, deleteItem.php were tested separately with different inputs and scenarios.

### 7.2.2 Integration Testing

#### 7.2.2.1 What is Integration Testing?

**Integration Testing** is the process of combining and testing multiple individual units or modules **as a group** to verify that they work together as expected. The focus here is not just on whether each component works on its own, but whether they **interact correctly** when integrated.

After all units pass **Unit Testing**, Integration Testing ensures that:

- Data flows smoothly between modules
- There are **no communication or logic errors** between systems
- The application behaves as expected when modules are combined

#### 7.2.2.1.1 Purpose of Integration Testing

- To detect interface defects between modules
- To test **real-world user workflows** that span multiple components
- To identify **data mismatches, improper logic, or API failures**
- To validate how **frontend and backend** communicate

#### 7.2.2.1.2 Techniques Used

- **Top-Down Approach:** Starting from login → home → list creation → item addition → sharing
- **Incremental Testing:** Modules were tested one-by-one, then combined
- **Database State Checking:** Verified updates via phpMyAdmin after operations
- **Error Injection:** Purposely sent bad inputs to test system handling

# GROCERYZ WEB APP

## 7.2.2.1.3 Benefits in Your Project

- Ensured **data consistency** between frontend inputs and backend processing
- Validated session management and role-based access (admin/user)
- Helped simulate **real user journeys** like adding/editing/sharing lists
- Caught bugs that unit testing alone could not (e.g., wrong list\_id, failed permissions)

## 7.2.2.2 How It Was Applied

- Testing **user login**: Verifying if user credentials are validated using user\_data table.
- Testing **item addition**: From UI → JS → AJAX → PHP → MySQL → Update table dynamically.
- Testing **list sharing**: Checking permission logic between shared users.
- Testing **session handling**: Making sure unauthorized users can't access protected pages.

## 7.2.3 Test Cases

Test Case ID	Module	Test Description	Input Data	Expected Result	Actual Result	Status
TC001	Login	Login with valid credentials	Email: test@abc.com Password: 123456	Redirect to homepage	As expected	Pass
TC002	Login	Login with invalid credentials	Email: wrong@abc.com Password: wrongpass	Show error: "Incorrect Password"	As expected	Pass
TC003	Signup	Register new user with unique email and phone	Name, Email, Phone, Password	User created successfully	As expected	Pass
TC004	Signup	Register with duplicate email	Existing Email ID	Show error: "User already exists with this email id."	As expected	Pass
TC005	Create List	Create a grocery list with a valid name	List name: "Weekly Groceries"	List saved to grocery_lists table	As expected	Pass
TC006	Create List	Try creating a list with empty name	List name: ""	Show error: "List name is required"	As expected	Pass

## GROCERYZ WEB APP

TC007	Add Item	Add a new item with valid data	Item name, type, quantity	Item added to list_items	As expected	Pass
TC008	Add Item	Add item with empty name	Empty item field	Show error	As expected	Pass
TC009	Edit Item	Update item name and quantity	Change name from "Milk" to "Almond Milk"	Item updated in database	As expected	Pass
TC010	Delete Item	Delete an item from list	Click delete on item row	Item removed from list and DB	As expected	Pass
TC011	Mark as Done	Check off item as completed	Click check icon on item	is_done updated in database	As expected	Pass
TC012	Share List	Share a list with another user	Email of another user	Entry added to shared_lists table	As expected	Pass
TC013	Access Shared List	Login as shared user and access list	Shared user login	List visible in shared section	As expected	Pass
TC014	Rename List	Rename an existing list	Click "Rename", new name: "Monthly Stock"	Updated list name reflected in UI and DB	As expected	Pass
TC015	Download List	Download list in PDF format	Click "Download PDF"	PDF file generated and downloaded	As expected	Pass
TC016	Forgot Password	Request OTP to reset password	Registered email	OTP sent to email and verified	As expected	Pass
TC017	Profile Update	Change user profile picture	Upload valid image file	Profile picture updated successfully	As expected	Pass
TC018	Profile Update	Change user name and phone number	New name and number	Details updated in user_data table	As expected	Pass
TC019	Logout	Log out from the system	Click logout button	Session cleared, redirected to login page	As expected	Pass
TC020	Session Management	Try accessing dashboard without login	Open dashboard link directly	Redirected to login page	As expected	Pass
TC021	Item Quantity Validation	Add item with negative quantity	Quantity: -5	Show error: "Quantity must be positive"	As expected	Pass
TC022	Duplicate Items	Try adding the same item twice in same list	Item: "Milk" added twice	Second entry not allowed or quantity updated	As expected	Pass

## GROCERYZ WEB APP

TC023	Shared List Permission	Shared user tries to edit with "edit" permission	Logged in shared user	Editing allowed	As expected	Pass
TC024	OTP Verification	Submit correct OTP for password reset	Correct 6-digit OTP	Redirected to reset password screen	As expected	Pass
TC025	OTP Verification	Submit invalid OTP	Incorrect OTP	Error message: "Invalid OTP"	As expected	Pass

Table 7.1: Test Cases

## **Chapter 8**

### **Results and Discussions**

## Chapter 8

### Results and Discussions

#### 8.1 Landing Page



Fig. 8.1: Landing Page

The screen is the **landing or welcome page** for web application **Groceryz** – a **Grocery List Organizer**. It displays application logo and on clicking anywhere or on ‘CLICK’ it navigates to login page.

#### 8.2 Login Page

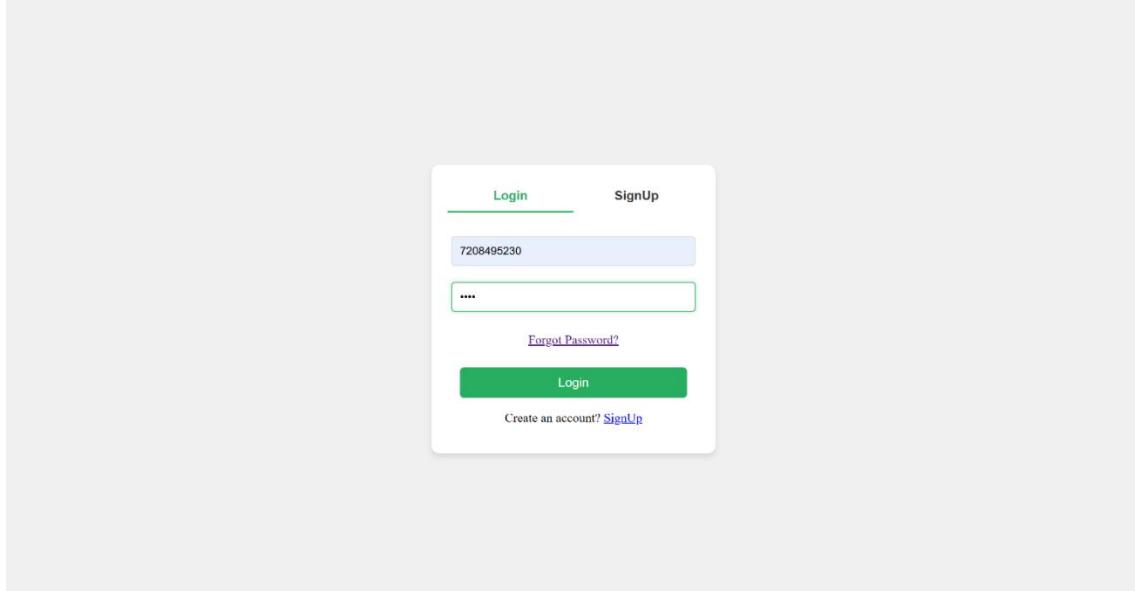
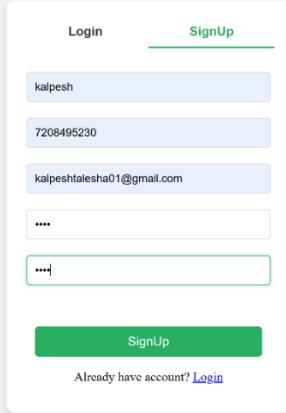


Fig. 8.2: Login Page

# GROCERYZ WEB APP

The screen shows the form to login. On accessing using valid credentials user will be logged in and directed to Home page. If the credentials are wrong it will show error.

## 8.3 SignUp Form



A screenshot of a web-based sign-up form. At the top, there are two tabs: "Login" and "SignUp", with "SignUp" being the active tab. Below the tabs are five input fields: a first name field containing "kalpesh", a middle name field containing "7208495230", an email field containing "kalpeshtalesha01@gmail.com", a password field containing "....", and a confirmation password field also containing "....". At the bottom of the form is a green "SignUp" button. Below the button, a small link says "Already have account? [Login](#)".

Fig. 8.3: SignUp Form

If user don't have a account he/she have to register first to access Groceryz they can create one through this signup form.

## 8.4 Grocery List Creation

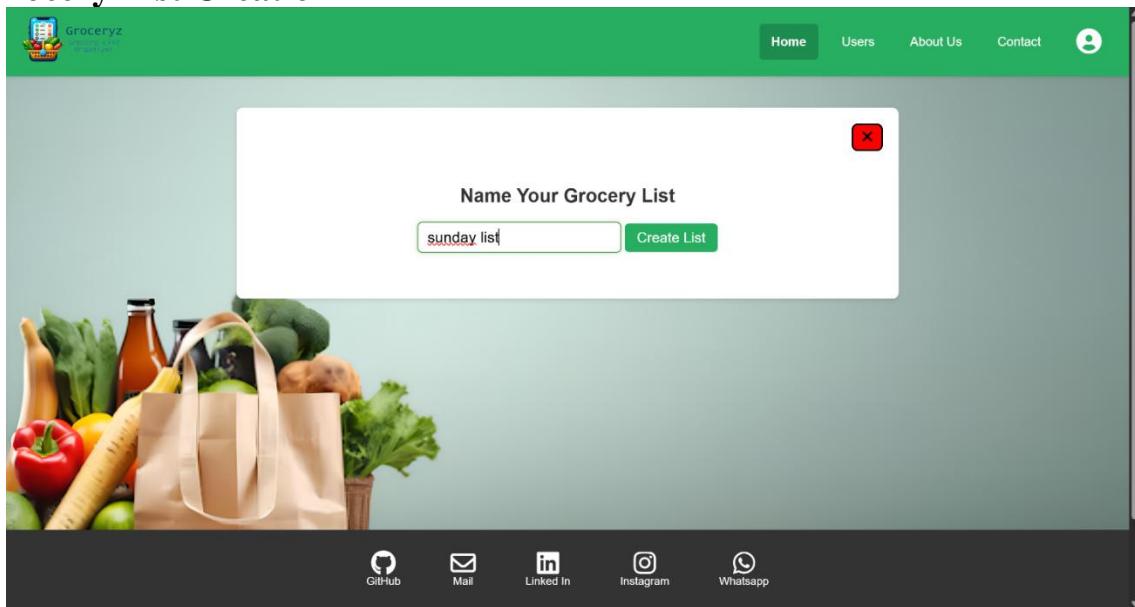


Fig. 8.4: Grocery List Creation

This screen shows the Home page of the site when have a input field to name the grocery list to be created and it created using create list button.

# GROCERYZ WEB APP

## 8.5 Display Grocery List

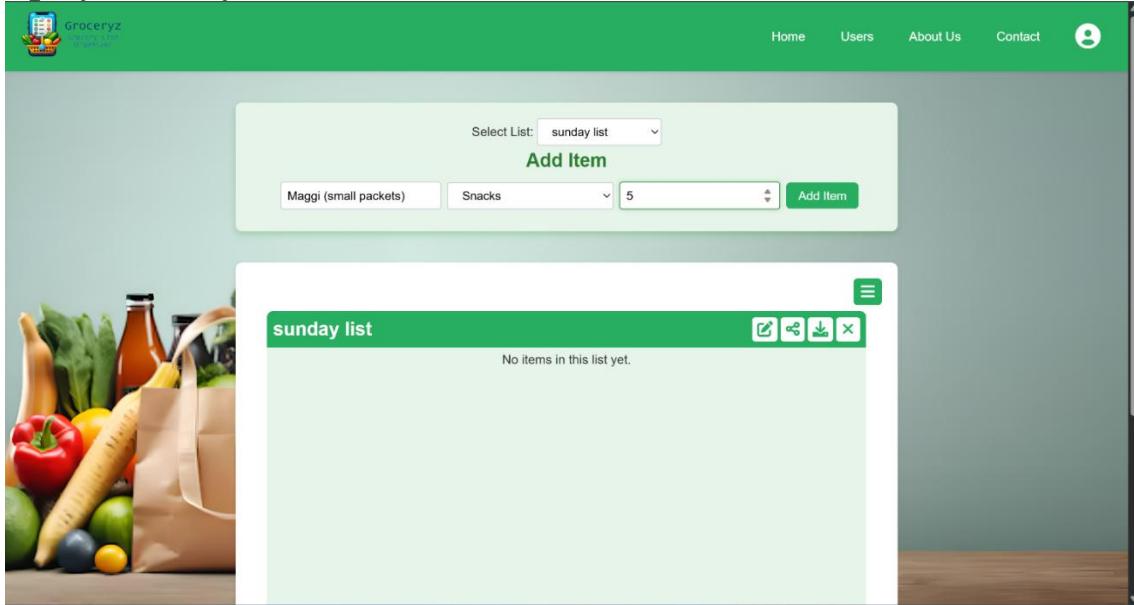


Fig. 8.5: Display Grocery List

This screen shows the grocery list along with the items in the list. It have form to add item by mentioning item name, type and quantity. Also have options to switch the list.

## 8.6 Adding List Item

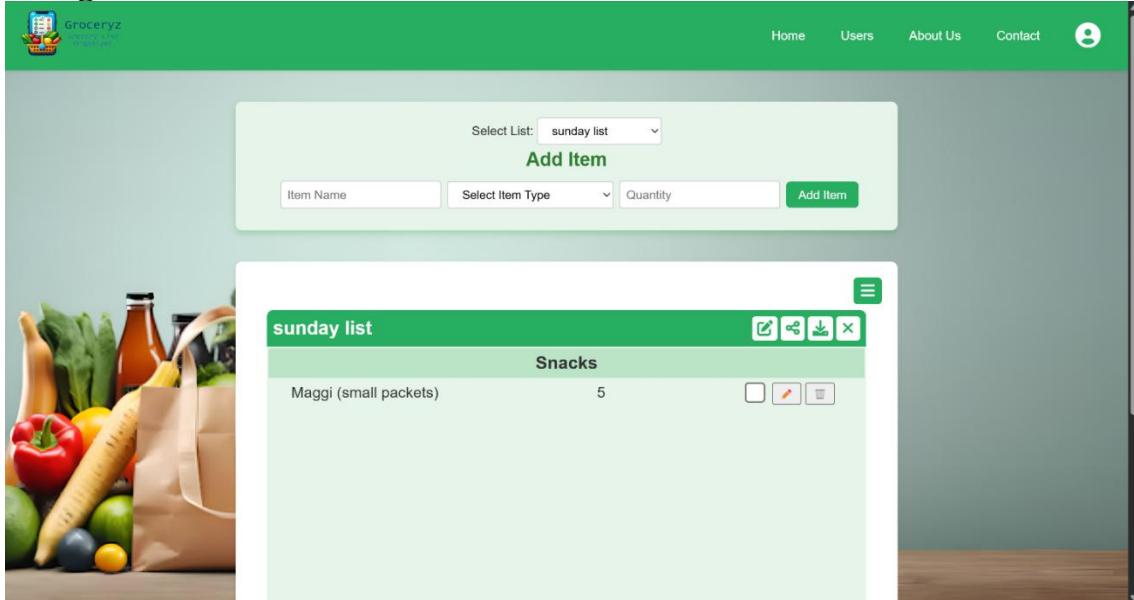


Fig. 8.6: Adding List Item

This screen shows the addition of a item (maggi) in the list and is displayed along with the options to mark as done, edit and delete item.

## GROCERYZ WEB APP

### 8.7 Editing and Deleting List Items

sunday list			
Fruits			
Apple	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Banana (dozen)	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Snacks			
Maggi (small packets)	5	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Potato Chips (10 Rs)	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Vegetables			
Cabbage (medium)	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Chilli (100g)	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Fig. 8.7: Editing and deleting List Items

This screen shows multiple items added in the list and also user can edit them or delete the unwanted items from the list.

### 8.8 Marking as Done

sunday list			
Fruits			
Apple	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Banana (dozen)	1	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Snacks			
Maggi (small packets)	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Potato Chips (10 Rs)	2	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Vegetables			
Cabbage (medium)	2	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Chilli (100g)	1	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Fig. 8.8: Marking as Done

Screen shows some of the items in list are marked as done which means the items are bought.

## GROCERYZ WEB APP

### 8.9 Downloading Grocery List

The screenshot shows a grocery list titled "sunday list". The list is organized into three categories: Fruits, Snacks, and Vegetables. Each item has a quantity listed next to it. On the right side of the list, there is a context menu with options to "Download Excel" and "Download PDF". The "Download Excel" option is highlighted with a checkmark. The "Add Item" button is visible at the top of the list.

sunday list	
Fruits	
Apple	2
Banana (dozen)	1
Snacks	
Maggi (small packets)	5
Potato Chips (10 Rs)	2
Vegetables	
Cabbage (medium)	2
Chilli (100g)	4

Fig. 8.9: Downloading Grocery List

This screen shows download options to download grocery list as an excel sheet or a pdf. According to user needs he can download any.

### 8.10 All Users Details Page

The screenshot shows a table of user details. The columns are ID, NAME, PHONE NO., EMAIL ID, and ROLE. There are three users listed: Kalpesh Talesha (admin), Kamlesh (admin), and Kush (user). Each user row has "Edit" and "Delete" buttons. A search bar and a sort dropdown are also present at the top of the table.

ID	NAME	PHONE NO.	EMAIL ID	ROLE
1	Kalpesh Talesha	7208495230	kalpeshalesha01@gmail.com	admin
2	Kamlesh	8356959907	kamleshtalesha28@gmail.com	admin
3	Kush	8928977389	kushsmoda04@gmail.com	user

Fig. 8.10: All Users Details Page

This screen shows user details page which is accessible to admin only a normal user can't see it. The page contains details of all the users of this web app and also provide options to sort or search for a user.

# GROCERYZ WEB APP

## 8.11 About Us Page

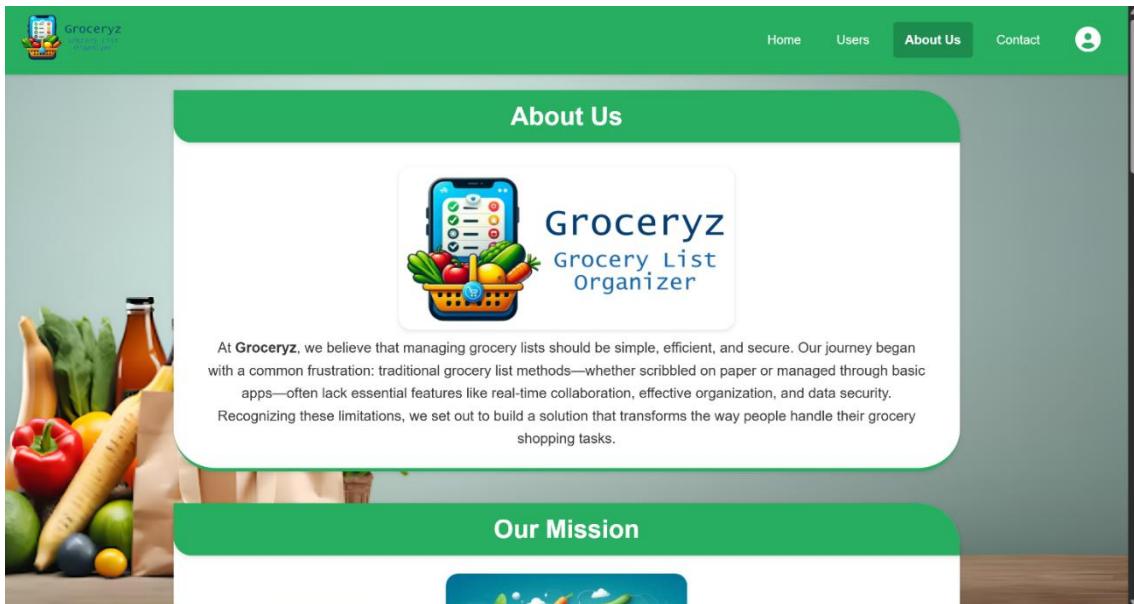


Fig. 8.11.A

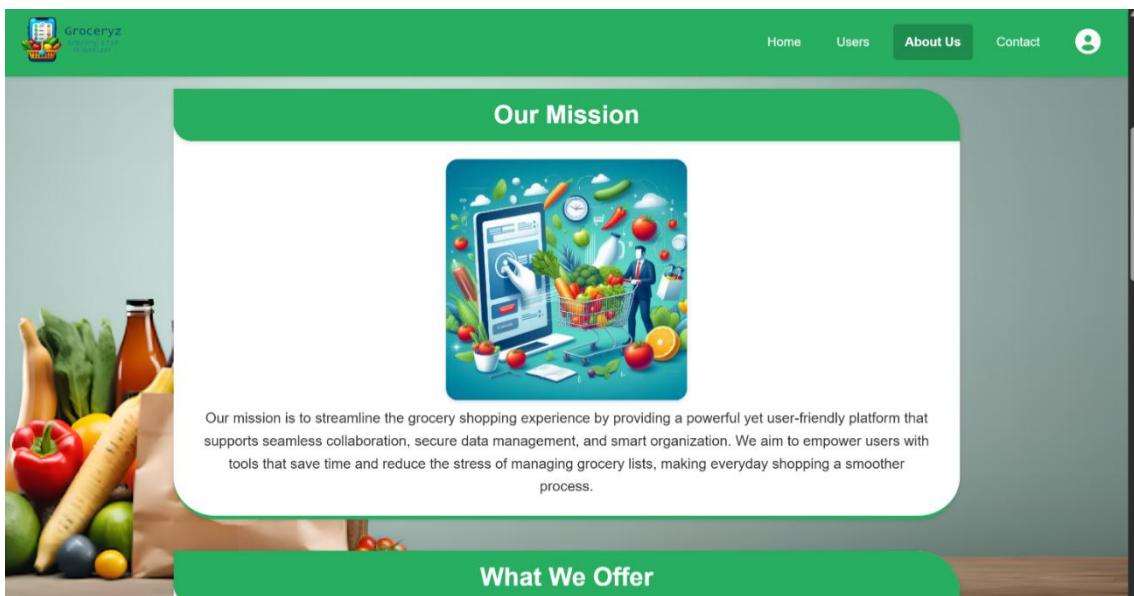


Fig. 8.11.B

This screens show the About us page which contains details of about web app such as mission, vision , what we offer and why to choose us. This is just the informative page.

# GROCERYZ WEB APP

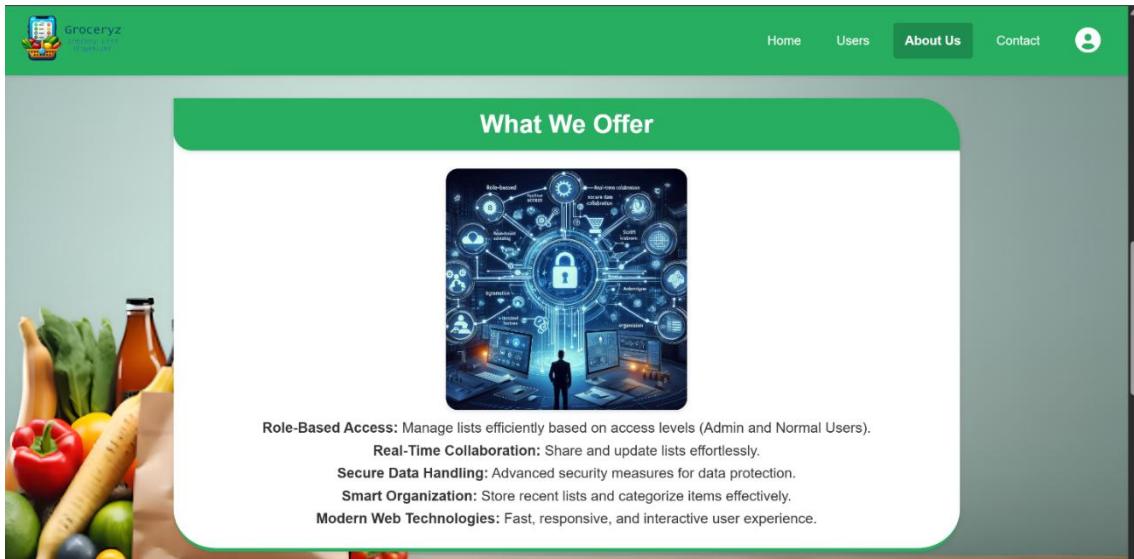


Fig. 8.11.C

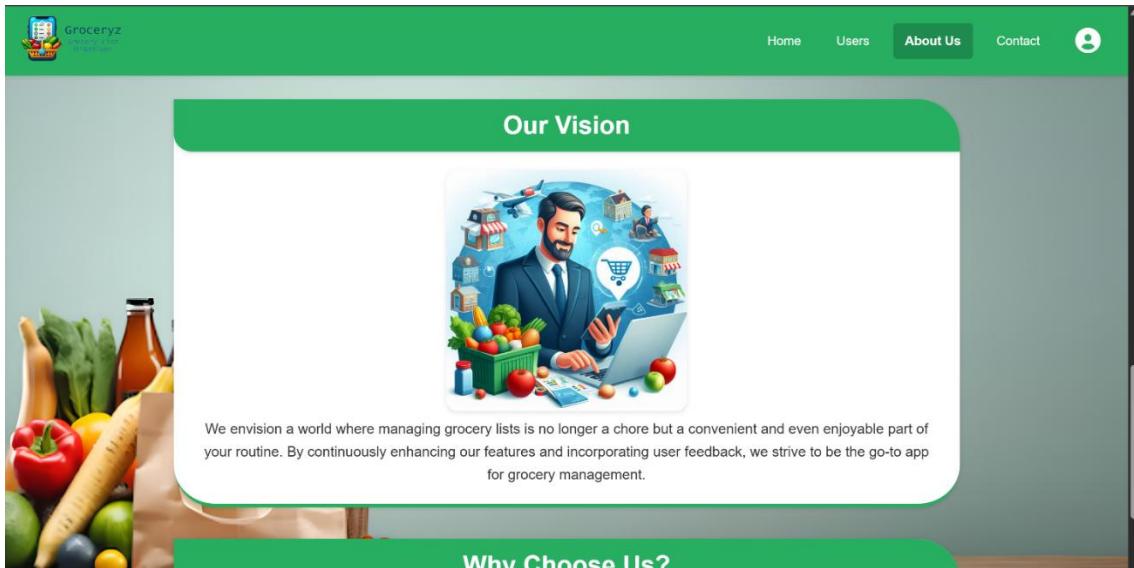


Fig. 8.11.D

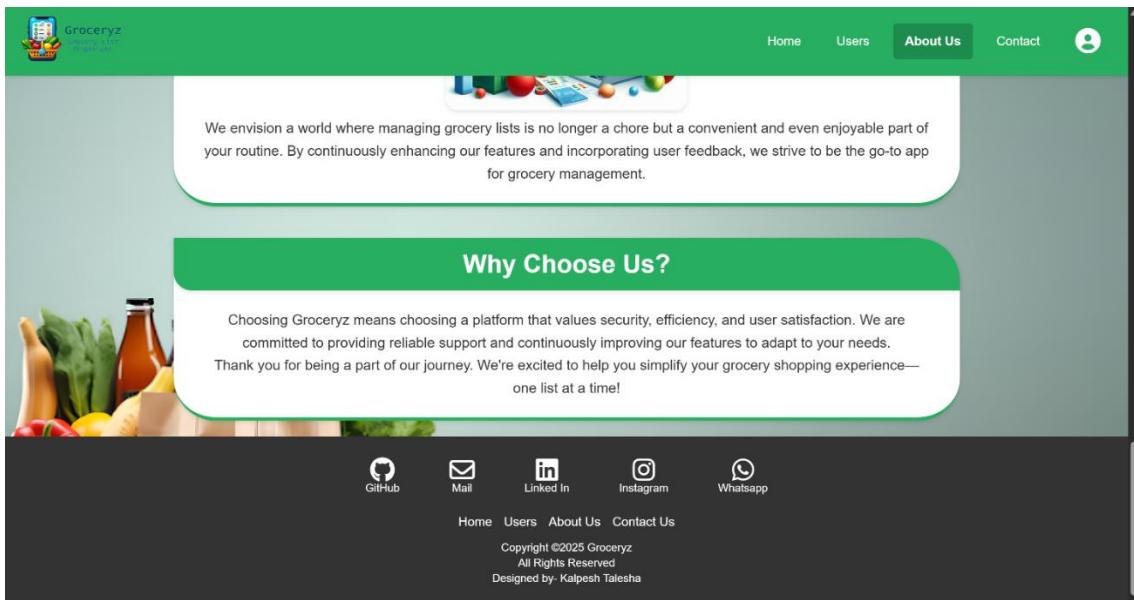


Fig. 8.11.E

## GROCERYZ WEB APP

### 8.12 Contact Page

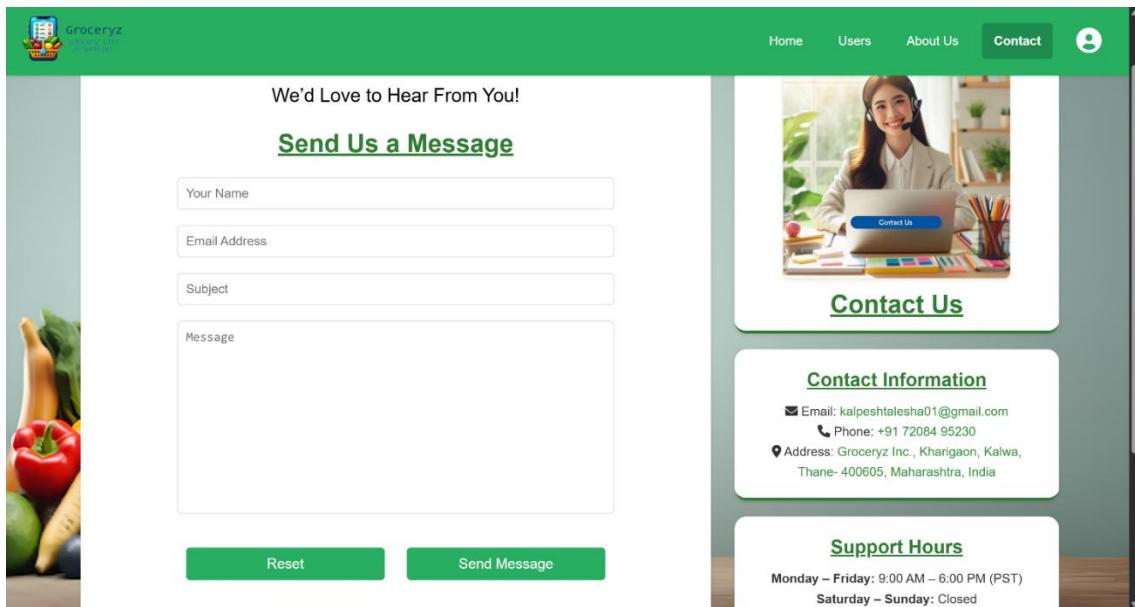


Fig. 8.12: Contact Page

This page contains contact details such as email, phone number, address and support hours. It also have a form to send the message to Groceryz.

### 8.13 Dashboard

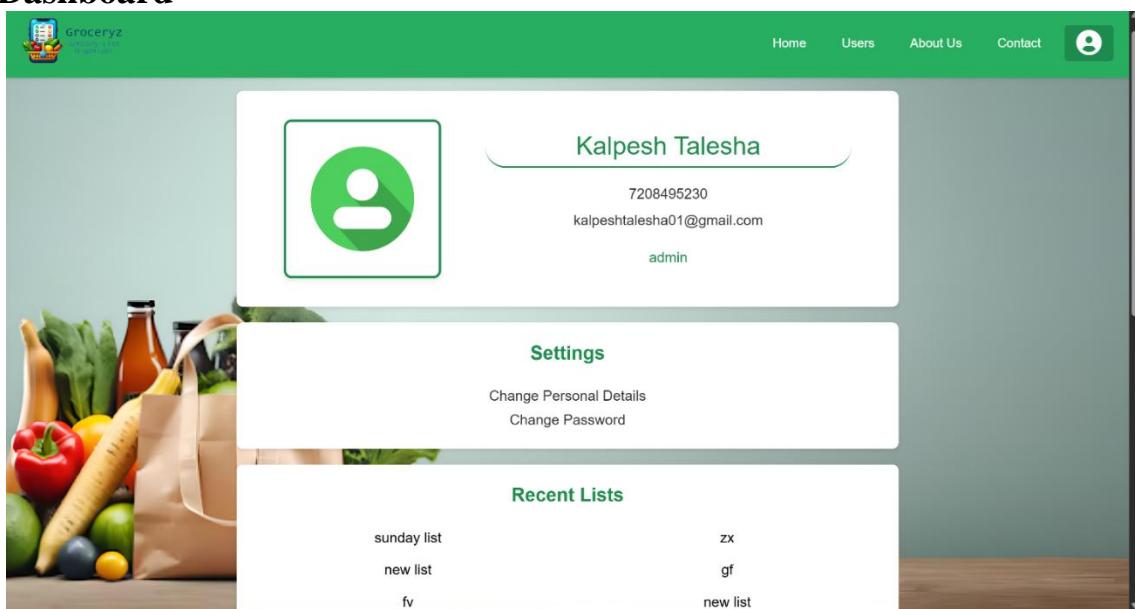


Fig. 8.13.A

This screen shows the dashboard page where user details are showed. It contains details like email, phone number, user type, profile picture, recent lists, shared lists, lists you have shared. It also have settings to update user details and password. And at the end have logout option.

## GROCERYZ WEB APP

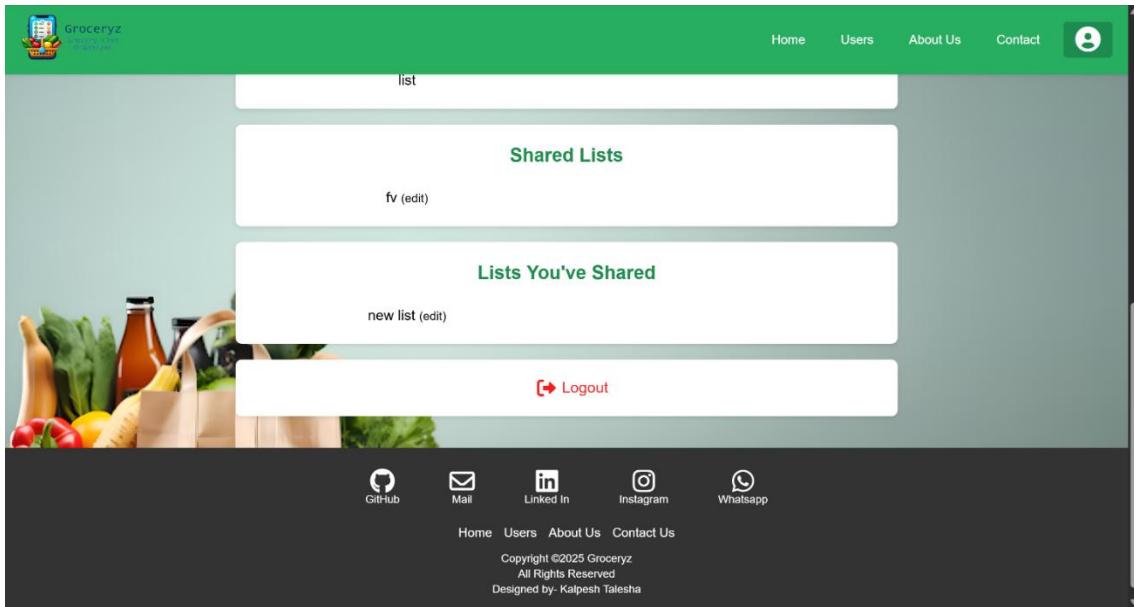


Fig. 8.13.B

### 8.14 Change User Details

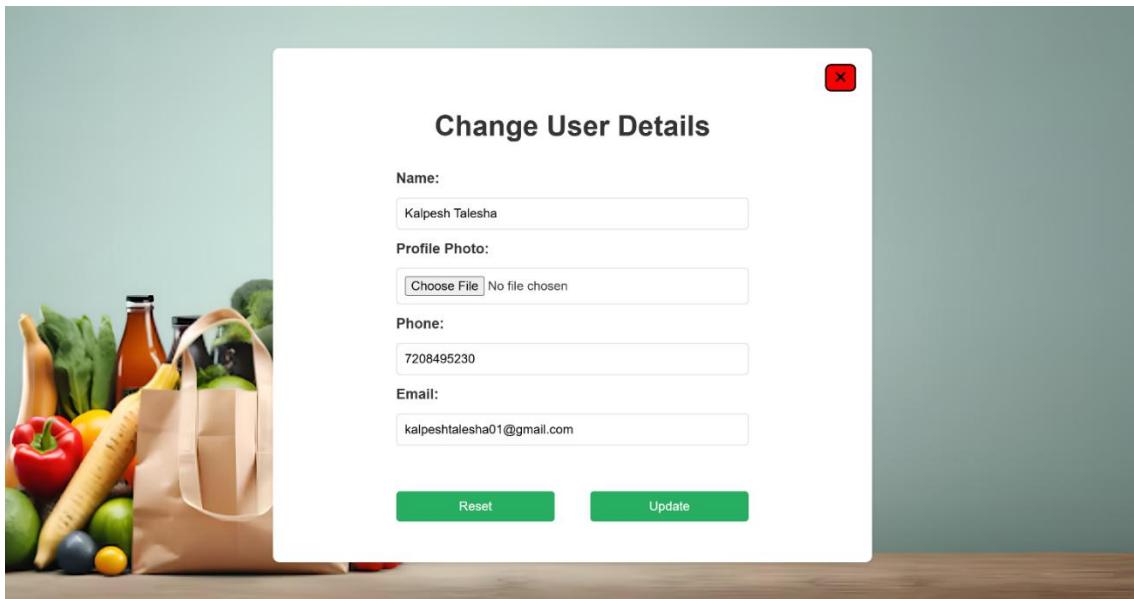


Fig. 8.14: Change User Details

This screen shows the form to change or update user details. It takes name, profile photo, phone number and email as input.

## GROCERYZ WEB APP

### 8.15 User Confirmation

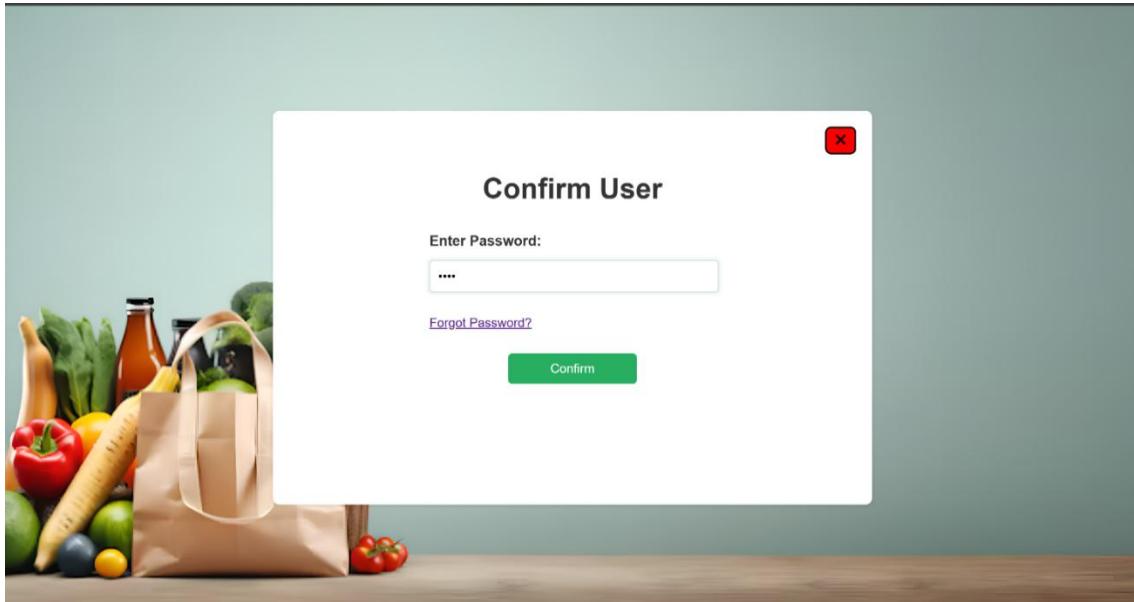


Fig. 8.15: User Confirmation

This screen displays user confirmation form which ask for password.

### 8.16 Profile Pic Update

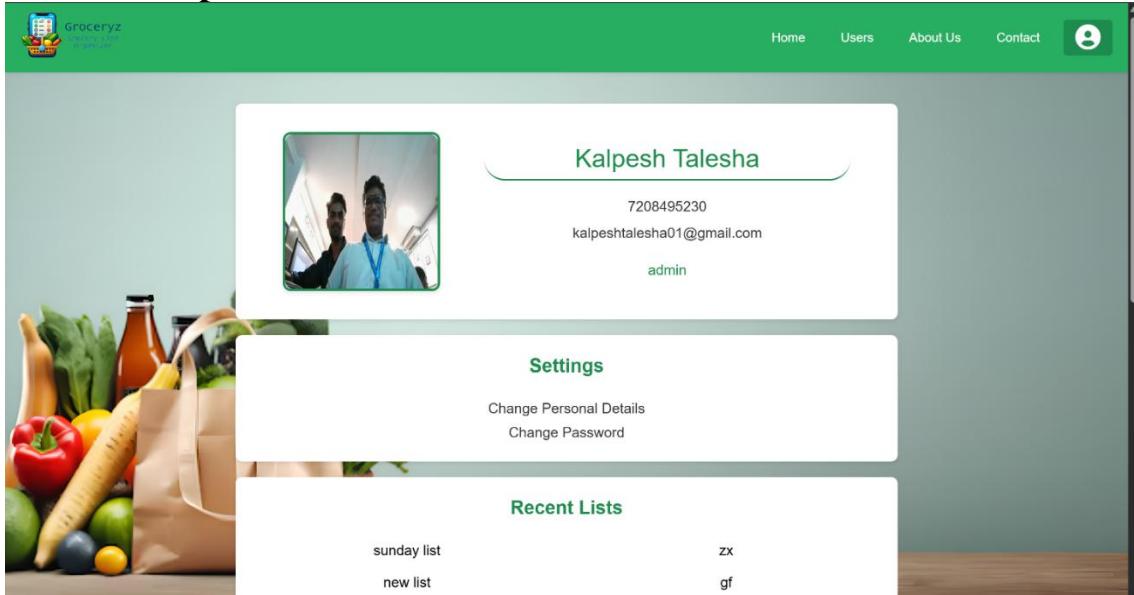
A screenshot of the groceryz web app dashboard after profile picture update. The header includes a logo, navigation links (Home, Users, About Us, Contact), and a user icon. The main area shows a profile card for "Kalpesh Talesha" with a photo, contact number (7208495230), email (kalpeshtalesha01@gmail.com), and role (admin). Below the profile card are two cards: "Settings" (with links to Change Personal Details and Change Password) and "Recent Lists" (listing "sunday list", "new list", "zx", and "gf"). The background features a blurred image of fresh vegetables and a brown paper bag.

Fig. 8.16: Profile Pic Update

This is the updated dashboard after profile picture change.

## GROCERYZ WEB APP

### 8.17 Forgot Password

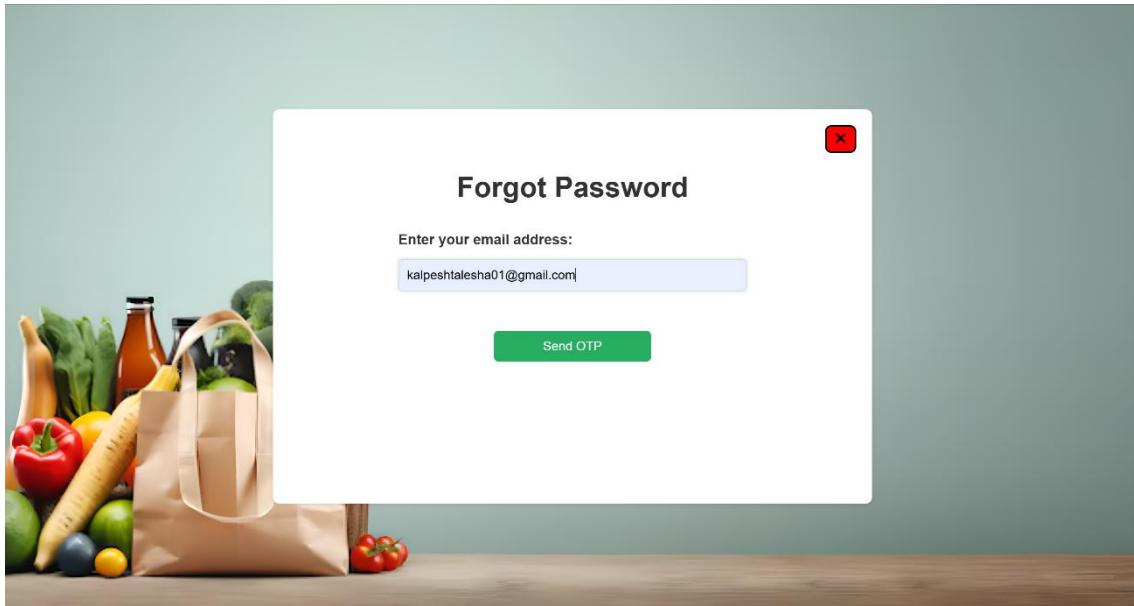


Fig. 8.17: Forgot Password

This screen displays a forgot password form which takes email as input and sends the OTP on that mail when send OTP button is clicked.

### 8.18 OTP Verification

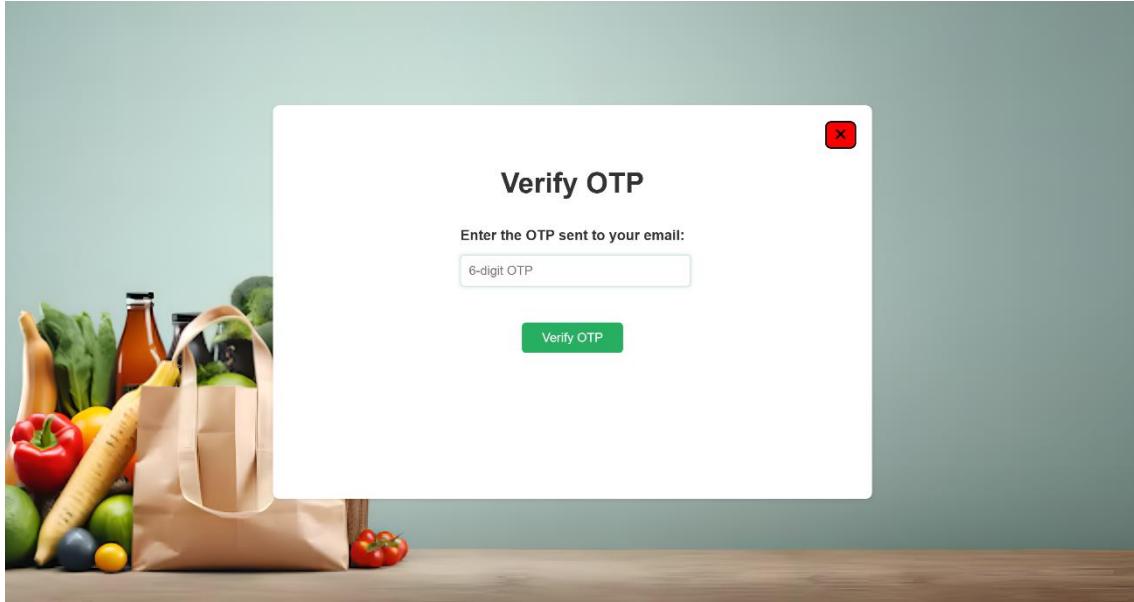


Fig. 8.18: OTP Verification

The OTP sent on the email are to be filled in this input container to verify the OTP.

## **Chapter 9**

### **Conclusion and Future Work**

## Chapter 9

# Conclusion and Future Work

### 9.1 Conclusion

The **Groceryz – Grocery List Organiser Web App** is a functional and efficient tool aimed at simplifying the process of managing grocery shopping through a digital platform. It enables users to easily create, manage, and share grocery lists online. The system includes basic and advanced features such as user authentication, list creation, item addition/removal, list sharing, and admin oversight, all wrapped in a simple, green-themed interface. Users can also download their grocery lists in various formats, including TXT, PDF, and Excel, offering flexibility in usage.

Developed using HTML, CSS, JavaScript, procedural PHP, and MySQL, the application provides secure session validation, user role management, and seamless interaction with the database. The backend handles user accounts, grocery lists, list items, and sharing permissions, making it suitable for both personal use and collaboration.

The project fulfills its objective by offering a platform that digitizes grocery management and reduces the manual effort typically involved. It also serves as a learning model for implementing full-stack web development using standard technologies. The division between user and admin roles ensures that the application remains manageable and scalable as user data grows.

### 9.2 Limitations

While Groceryz achieves its intended functionality, the system does have certain limitations that can impact the overall experience for users:

1. **Lack of Real-Time Collaboration:** Shared lists do not reflect live updates if multiple users are editing simultaneously.
2. **Basic Permission Control:** Users cannot assign custom view/edit permissions to collaborators on a list.
3. **Limited Mobile Responsiveness:** The current UI does not fully adapt to all screen sizes, making mobile use slightly inconvenient.

## GROCERYZ WEB APP

4. **Missing Advanced Features:** Features like item quantity tracking, price estimation, or expense calculation are not yet included.
5. **No Offline Mode:** The application requires an internet connection to function, with no offline access to lists.
6. **No Native App:** There is no mobile app version, which limits accessibility for users who prefer mobile platforms.
7. **Basic Security:** The system lacks multi-factor authentication, data encryption, and other advanced security practices.
8. **No API Integration:** The system does not currently integrate with external APIs for price comparison or product recommendations.

### 9.3 Future Work

There is significant scope to enhance the Groceryz application to meet modern expectations and user needs. The following upgrades and enhancements are proposed:

1. **Real-Time Updates:** Implement WebSockets or AJAX to support live updates in shared lists for better collaboration.
2. **Mobile Application:** Develop a native mobile app for Android and iOS to enhance portability and usability.
3. **Item Price and Budget Features:** Add fields for item quantity, unit prices, and auto-calculated totals to help users manage their budgets.
4. **Advanced Permission Model:** Allow list creators to assign view-only or edit rights to specific users.
5. **Offline Support:** Enable offline access and syncing when the internet connection is restored.
6. **Security Enhancements:** Introduce multi-factor authentication, encrypted storage, and secure session handling.
7. **Analytics and Suggestions:** Provide users with insights into their most bought items or spending habits and suggest frequently purchased items.
8. **API Integration:** Use third-party APIs for price comparison, product suggestions, or locating nearby stores with offers.

# **Chapter 10**

## **References**

## Chapter 10

### References

#### 10.1 Frameworks & Libraries

- **HTML5 & CSS3**

Used to structure and design the frontend of the web application.

<https://developer.mozilla.org/en-US/docs/Web/HTML>

<https://developer.mozilla.org/en-US/docs/Web/CSS>

- **JavaScript (Vanilla)**

Used for dynamic client-side interactions like item handling and UI responses.

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

- **PHP (Procedural)**

For server-side logic, handling sessions, form processing, and database communication.

<https://www.php.net/manual/en/>

<https://www.w3schools.com/php/>

- **MySQL**

A relational database system used to manage user data, grocery lists, and shared permissions.

<https://dev.mysql.com/doc/>

- **jsPDF / SheetJS (*Optional*)**

Used for generating downloadable formats of the grocery lists (PDF, Excel).

<https://github.com/parallax/jsPDF>

<https://sheetjs.com/>

#### 10.2 Tools & Platforms

- **XAMPP**

A local development environment bundling Apache, PHP, and MySQL.

<https://www.apachefriends.org/index.html>

## GROCERYZ WEB APP

- **phpMyAdmin**

Used for graphical management of the project's MySQL database.

<https://docs.phpmyadmin.net/>

- **Visual Studio Code**

Code editor used throughout the project for development and debugging.

<https://code.visualstudio.com/docs>

- **Font Awesome**

Used to enhance UI/UX with recognizable icons.

<https://fontawesome.com/docs>

- **Microsoft Copilot**

Used for AI-assisted image generation and project documentation formatting.

<https://copilot.microsoft.com>

### 10.3 Learning Resources

- **MDN Web Docs**

The primary reference for in-depth web standards and best practices.

🌐 <https://developer.mozilla.org/>

- **W3Schools**

Referenced for quick tutorials and code examples in HTML, PHP, SQL, and JS.

🌐 <https://www.w3schools.com/>

- **GeeksforGeeks**

Helped with backend scripting, SQL operations, and web logic explanations.

🌐 <https://www.geeksforgeeks.org/>

- **Stack Overflow**

Community forum for debugging, feature implementation, and issue resolution.

🌐 <https://stackoverflow.com/>