UNIVERSITY OF ENGINERRING AND TECHNOLOGY,TAXILA.

Semester: 6th

BSC COMPUTER ENGINERRING

FINAL PROJECT REPORT

GROCERY LIST HOME APPLICATION

- **Submitted By:** ATTIA BATOOL
- **Course Instructor:** ENGR.SANA
- **Lab Instructor:** ENGR.SHAHID BUTTA

DATABASE MANAGEMENT SYSTEM (DBMS)

Grocery List Home Application

Abstract

The **Grocery List Home Application** (GLHA) helps manage grocery items efficiently, reducing errors and missing stock information. It stores item details like name, quantity, and category in a database. The system allows users to add, view, and delete items through a simple interface. Using **PHP** and **MySQL**, it performs SQL queries to manage inventory data. This project demonstrates the practical use of **Database Management System** (DBMS) concepts for efficient inventory management.

Problem Statement

Managing grocery items manually often leads to errors, missing stock information, and a lack of proper records. Shopkeepers or households may forget which items are low or which category they buy the most. To solve this, I developed a **Grocery Inventory Management System** that stores, manages, and displays grocery item records using a database-driven web application.

Objectives

- To design a database that stores grocery item details including name, quantity, and category.
- To provide a user-friendly interface for adding, viewing, and deleting items.
- To implement backend logic using **PHP** and **MySQL**.
- To perform **SQL queries** to manage the inventory efficiently.
- To apply DBMS concepts practically in a small but meaningful project.

Business Rules

1. **A user can have multiple grocery items** — One-to-Many relationship.
2. **Each item belongs to one category** — Many-to-One relationship.
3. **Each category can contain multiple items** — One-to-Many.
4. **Each item is added by a specific user** — Many-to-One.
5. **Users can generate multiple expense reports** — One-to-Many.
6. **Each expense report records details of one item** — Many-to-One.
7. **Expense reports are also linked to the user who generated them** — Many-to-One.

Technologies Used

Component	**Tool/Language**
Frontend	HTML, CSS
Backend	PHP
Database	MySQL
Server	XAMPP (Apache + MySQL)

System Modules

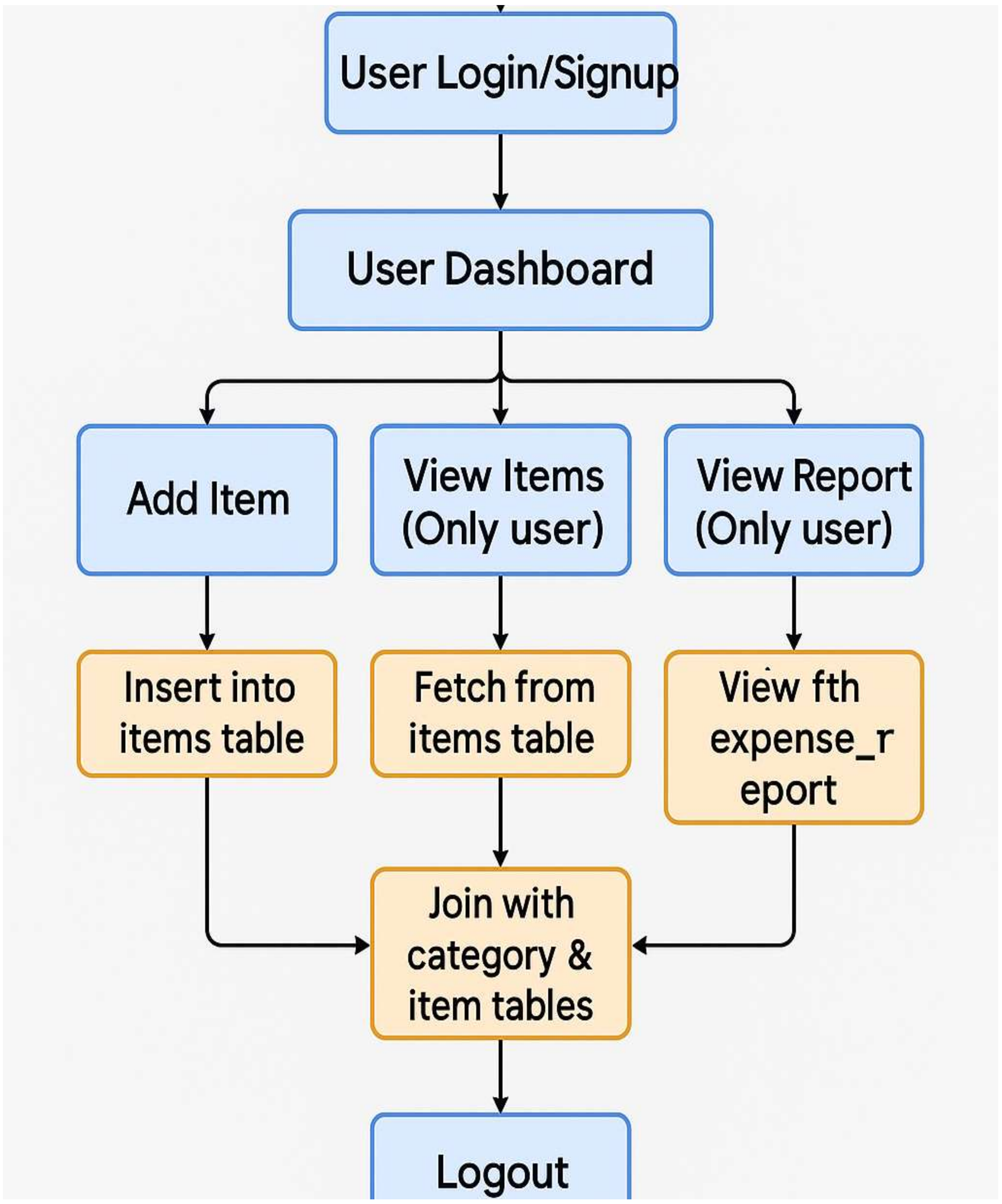
1. **Login System**
   - Simple login with username (for session control).
   - Used $_SESSION to maintain active users.
2. **Add Item Page**
   - Form to add item name, category, and quantity.
   - Data inserted into MySQL database.
3. **Item Dashboard**
   - Displays all items from the database.
   - Shows total items and per-category counts.
4. **Logout Page**
   - Session destruction with username confirmation.
5. **Category Analysis**
   - Counts how many items exist in each category (e.g., Fruit, Dairy).
   - Shows percentages using simple PHP logic.

System Architecture

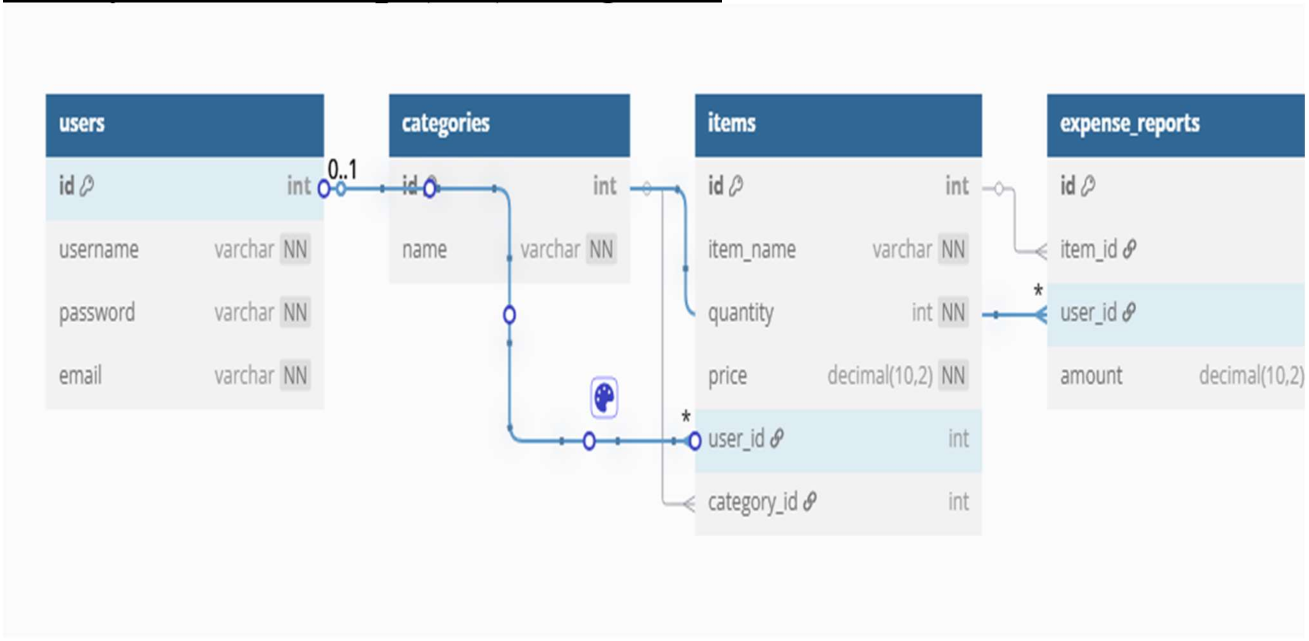
This project follows a **2-Tier Architecture**:

- **Presentation Tier:** The front-end of the system, developed using HTML and CSS, runs in the user's browser and handles UI/UX.
- **Data Tier (Application + Database):** The back-end is built with PHP and MySQL, where PHP handles business logic and directly communicates with the MySQL database to store, retrieve, and manipulate data.

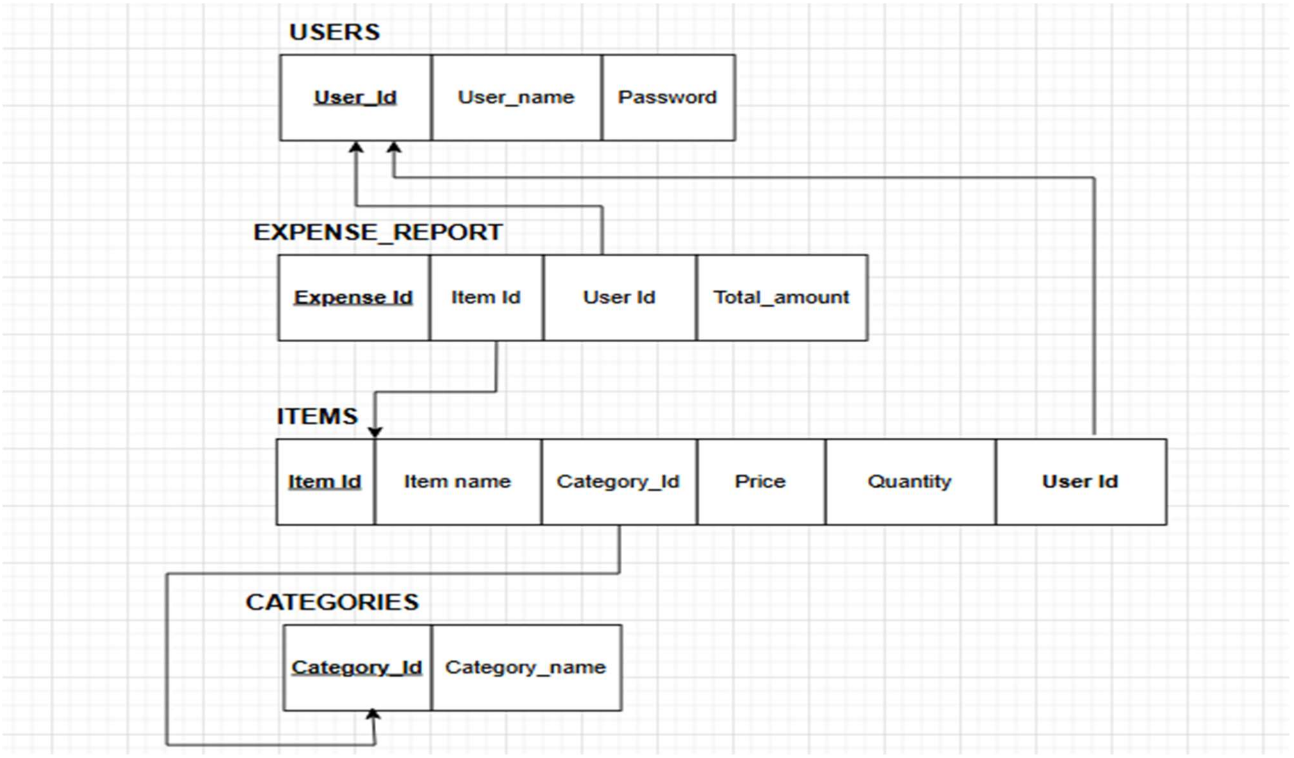
Data Flow Diagram (DFD)



Entity Relationship (ER) Diagram:



Relational Model:



Database Schema Overview

1. **Schema Name:** grocery_db
    - A single schema used to define all tables and relations.
    - Normalized to 3NF to avoid redundancy.
2. **Views Implemented:**
    - **view_all_items:** Displays joined data from items and categories.
    - **view_category_summary:** Shows item count per category.
    - **view_expense_report:** Retrieves expense info with item names and dates.

Database Design

Database Name: grocery_db
The system uses a normalized schema to manage grocery inventory, user sessions, categories, and expenses efficiently.

Tables Overview

1. **users**
    - Stores login information.
    - Fields: id, username
    - Used for session control and authentication.
2. **items**
    - Manages grocery items.
    - Fields: id, name, category, quantity,price
    - Core table for inventory operations.
3. **categories**
    - Organizes items into types (e.g., Fruits, Snacks).
    - Fields: id, name
    - Supports filtering and grouping.
4. **expense_reports**
    - Tracks purchase records and costs.
    - Fields: id, item_id, amount, date
    - Helps analyze monthly expenses.

Working Steps

1. **User opens the login page**, enters username.
2. On successful login, **user is redirected to item dashboard**.
3. **User can add new items** via form → Item is stored in items table.
4. All added items are displayed in a **table format** on dashboard.
5. A summary section shows **total items and per-category counts**.
6. If user clicks **Delete**, item is removed from the database.
7. When logout is clicked, user enters credentials again.

Results

- Items are properly inserted, displayed, and deleted from the MySQL database.
- I successfully applied **Insert, Select, Delete, and Count** queries.
- Implemented session-based login/logout system.
- Displayed grocery data dynamically from the database.
- Category analysis helps visualize which items are most used.

login_signup_page

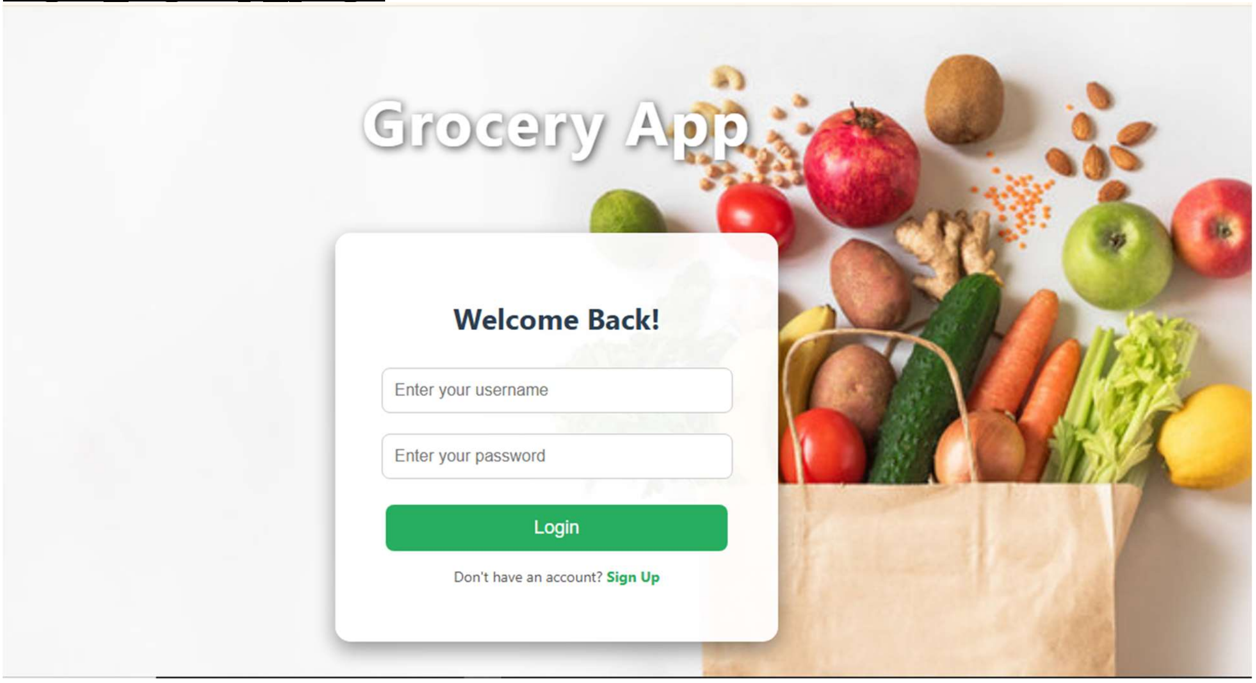
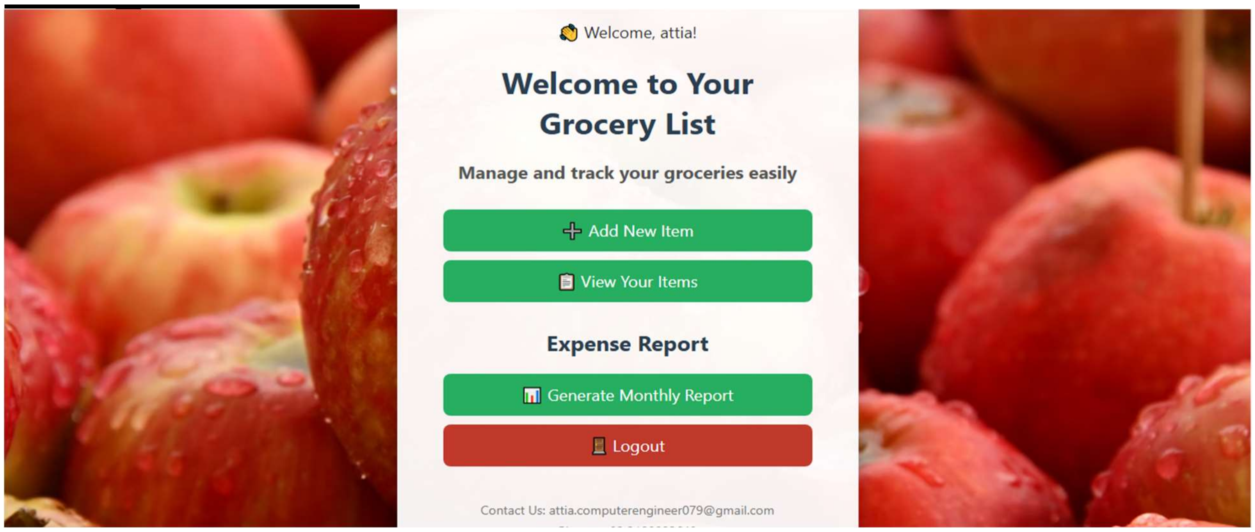


Figure 01: This is the main entry point of my project.Users can log in or sign up to access the system securely.

home_dashboard


Welcome, attia!

Welcome to Your Grocery List

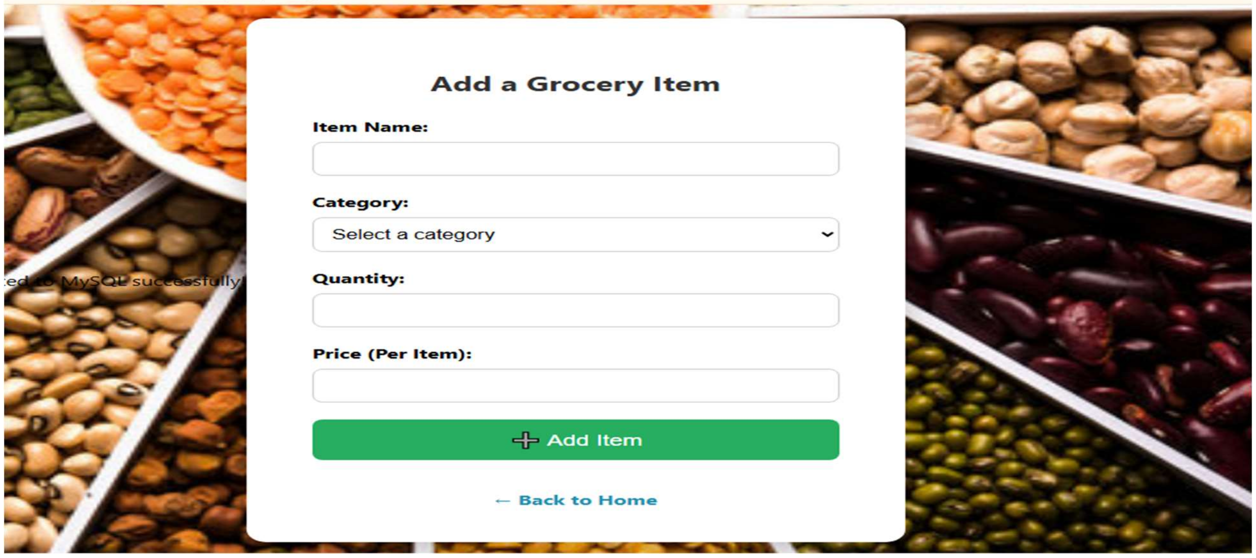
Manage and track your groceries easily

Expense Report

Contact Us: attia.computerengineer079@gmail.com

Figure 02: This is the home page after login.It has buttons to add items, view items, generate reports, and log out.

Add_items_page


ed to MySQL successfully

Add a Grocery Item

Item Name:

Category:

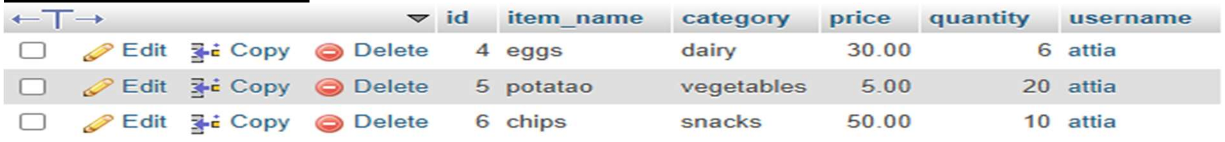
Select a category

Quantity:

Price (Per Item):

Figure 03:I created this form for adding new grocery items.Users can input item name, category, and quantity.

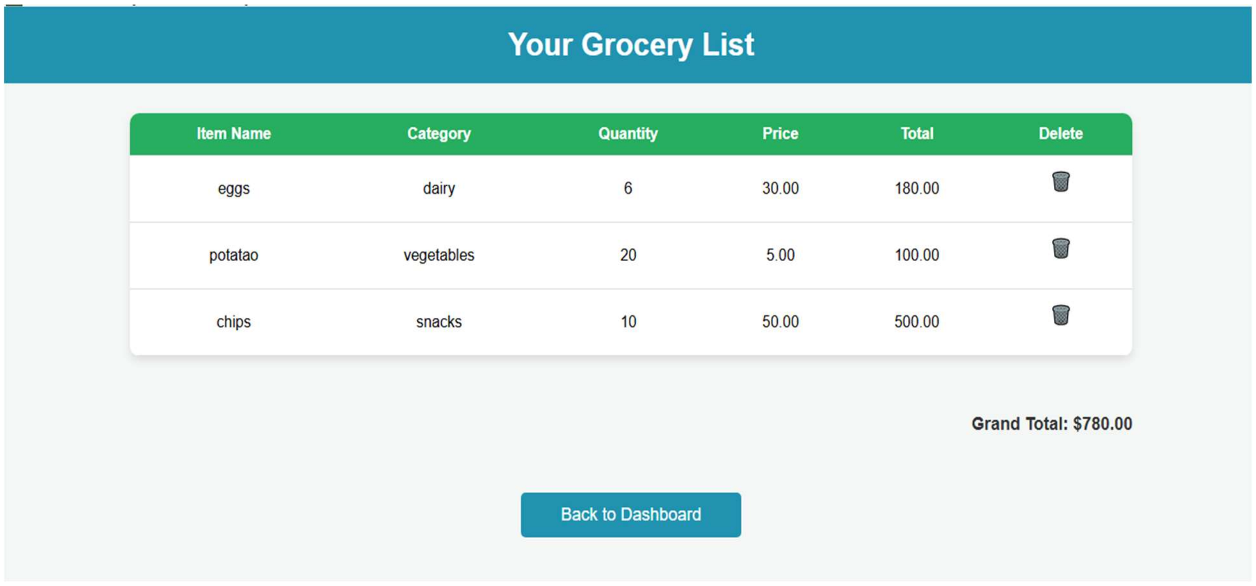
Database View



←T→				id	item_name	category	price	quantity	username
☐	Edit	Copy	Delete	4	eggs	dairy	30.00	6	attia
☐	Edit	Copy	Delete	5	potatao	vegetables	5.00	20	attia
☐	Edit	Copy	Delete	6	chips	snacks	50.00	10	attia

Figure 04: Item details stored in a database.

view_items_page



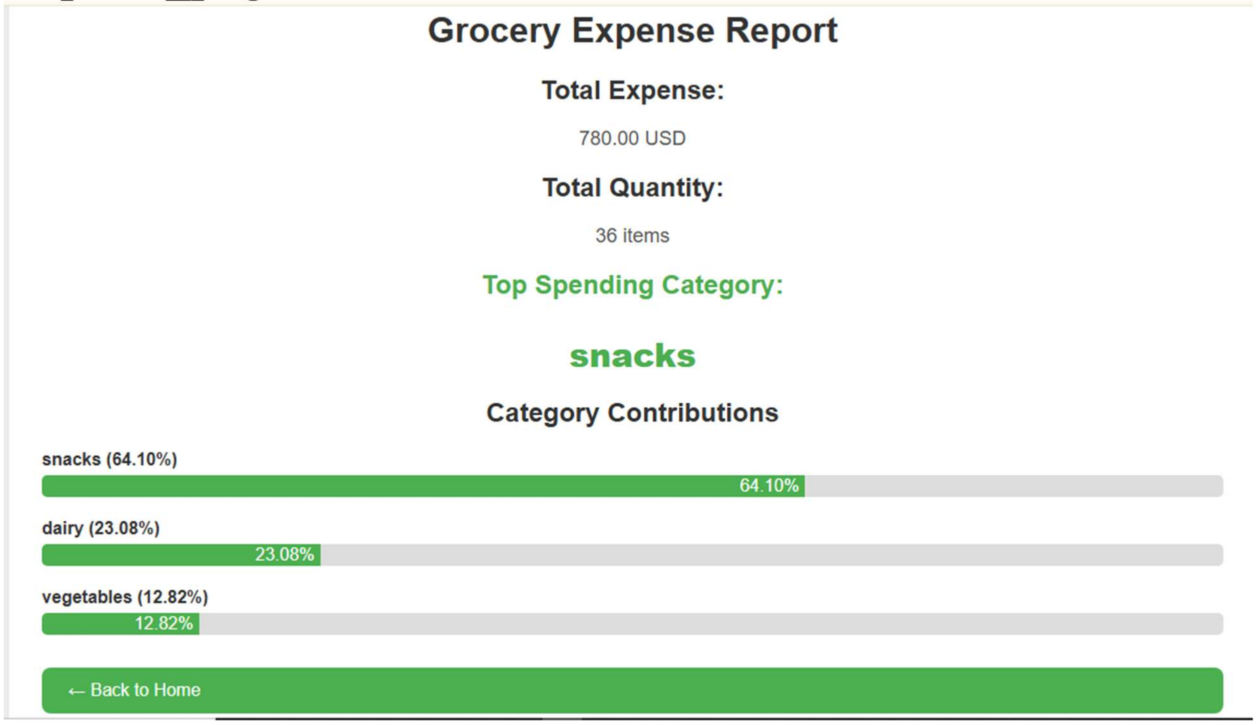
Your Grocery List

Item Name	Category	Quantity	Price	Total	Delete
eggs	dairy	6	30.00	180.00	🗑
potatao	vegetables	20	5.00	100.00	🗑
chips	snacks	10	50.00	500.00	🗑

Grand Total: $780.00

Figure 05:This page displays all added grocery items.Items are shown in a table with category and quantity.

Reports_page



Grocery Expense Report

Total Expense:

780.00 USD

Total Quantity:

36 items

Top Spending Category:

snacks

Category Contributions

snacks (64.10%)

64.10%

dairy (23.08%)

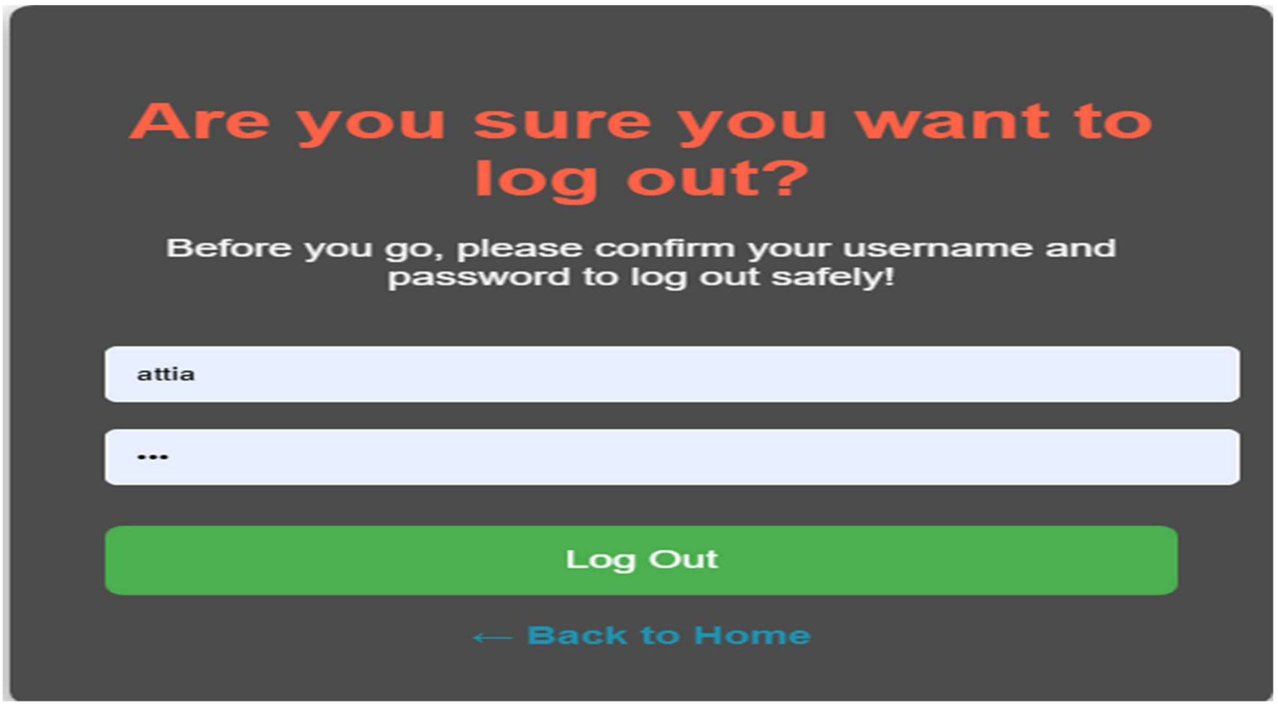
23.08%

vegetables (12.82%)

12.82%

Figure 06:This page shows generated reports.It includes category-wise analysis and total item count.

logout_page


Are you sure you want to log out?

Before you go, please confirm your username and password to log out safely!

attia

•••

Figure 07: This is the logout confirmation page.It safely ends the session and redirects to the login screen.

DBMS Concepts Applied

DBMS Concept	**How It Was Used**
Table Design	Used CREATE TABLE with suitable data types
Primary Key	Applied on both users and items table
DML Commands	Used INSERT, SELECT, DELETE queries
Query Filtering	Applied WHERE condition to manage specific records
Aggregate Functions	Used COUNT(*) for category-based analysis

Learnings & Improvements

- I learned how to **connect PHP with MySQL** and perform basic DB operations.
- Understood the practical role of a database in a web application.
- Improved logic building using SQL and condition-based filters.
- I plan to add:
    - **Update item** feature
    - **User authentication with password**
    - **Category-wise charts**

Conclusion

This project helped me implement key concepts from my **Database Management System (DBMS)** course in a practical way. I handled table design, SQL queries, and user sessions effectively. It is a useful system that can help small businesses or households manage grocery items digitally.
This was a valuable hands-on experience and a strong step toward understanding backend database handling and integration with frontend web interfaces.