Project Report: Grocery List Manager

Submitted By: ADDEPALLI THANUSH SAI
Course: INTRODUCTION TO PROBLEM SOLVING

1. Cover page

Project Title: Grocery List Manager
Student ID: 25BAI10273
Date: 25-11-2025

---

2. Introduction

In today's fast-paced world, managing daily chores like grocery shopping can be inefficient and prone to errors. The **Grocery List Manager** is a digital solution designed to simplify this process. It allows users to create, manage, and track their grocery lists efficiently, ensuring they never forget an item again. This project applies core software engineering principles to build a functional, user-friendly console-based application.

3. Problem Statement

Many individuals and families struggle with:

- Forgetting to buy essential items.
- Difficulty in managing lists for different stores or occasions.
- Lack of organization in paper-based lists.
- No way to track purchased items or budget.

The Grocery List Manager addresses these issues by providing a centralized, digital platform for list management.

---

4. Functional Requirements

1. **User Management Module**
    - User registration and login.
    - Secure authentication.
2. **Grocery List Management Module**
    - Create, view, update, and delete grocery lists.

    - Add, remove, and modify items within a list.
    - Mark items as "purchased" or "pending".
3. **Reporting & Analytics Module**
    - View total items in a list.
    - Calculate estimated total cost.
    - Display purchase progress (e.g., 5/10 items purchased).

---

5. Non-Functional Requirements

1. **Performance:** The application should load and respond to user inputs within 2 seconds.
2. **Usability:** Intuitive console interface with clear menus and prompts.
3. **Reliability:** The system should handle invalid inputs gracefully without crashing.
4. **Security:** Passwords are stored in a hashed format; user data is isolated.
5. **Maintainability:** Code is modular, well-commented, and follows coding standards.
6. **Error Handling:** Clear error messages for invalid inputs or operations.

---

6. System Architecture

The application follows a **modular layered architecture**:

- **Presentation Layer:** Console-based user interface.
- **Business Logic Layer:** Handles user actions, list operations, and calculations.
- **Data Access Layer:** Manages reading and writing user and list data to JSON files.

7. Design Decisions & Rationale

- **JSON for Storage:** Chosen for simplicity and ease of use in a small-scale project.
- **Console Interface:** Focus on core logic and modularity without frontend complexity.
- **Modular Code Structure:** Ensures maintainability and scalability.

---

8. Implementation Details

- **Language:** Python
- **Key Libraries:** json, hashlib, os
- **Modules:**
    1. categories.py – Handles user registration and login.
    2. list_manager.py – Manages grocery list operations.
    3. Data store.py – Defines the Item class.
    4. Grocery item.py – Generates list summaries and progress.
    5. main.py – Entry point of the application.

9. Challenges Faced

- Handling file I/O efficiently without data corruption.
- Ensuring user sessions remained isolated.
- Designing a user-friendly console menu system.

---

10. Learnings & Key Takeaways

- Improved understanding of modular programming and file handling.
- Learned to design and document a complete software project.
- Gained experience in version control and GitHub management.

---

11. Future Enhancements

- Web-based UI using Flask or Django.
- Multi-user collaboration on lists.
- Barcode scanning for item addition.
- Mobile app version (Android/iOS).
