**Grocery To-Do (Shopping List) Application**

*SDLC Phase: Planning Document*

Individual Assignment 1 – Software Development and DevOps

Student Name: Sofia Boicenco  
Course: Software Development and DevOps  
Date: 21.09.2025

**1. Objective and SDLC Model**  
The objective of this project is to design and implement a minimal software application that can later be used in a DevOps pipeline. The use case that I chose is a **Grocery To-Do (Shopping List) Application**, where users can create, view, update, and delete items on a shopping list.

The chosen SDLC model is **Agile**, a lightweight approach with one or two short iterations because:

* It supports **incremental delivery** of features (CRUD API, then optional UI).
* Requirements are small and can be refined quickly.
* The project must be completed in a short time.

Each SDLC phase (Planning, Requirements, Design, Implementation) was followed sequentially, with small Agile iterations for improvements and testing.

**2. Project Scope**

* **In-Scope (Version 1):**
  + Add new grocery items (with name, quantity, category).
  + View list of items (all or filtered by purchased status).
  + Edit item details.
  + Mark or unmark an item as purchased.
  + Delete items.
  + Data persistence using SQLite (items saved locally, not just in memory).
* **Out-of-Scope (Version 1):**
  + User accounts and authentication.
  + Multi-user collaboration.
  + Notifications or reminders.
  + Analytics, reports, or advanced UI design.

**3. Stakeholders**

 People who frequently buy groceries, including families or those who tend to forget what they need to buy (such as myself).

 Anyone who wants to stay organized and manage daily shopping more efficiently.

**4. Constraints and Assumptions**

* The project has to be done individually and before the given deadline.
* Development will be done in **Python (FastAPI)** with **SQLite** for persistent storage.
* The app is designed for local execution but can be scaled later for cloud deployment in a DevOps environment.
* Single user environment (no multi-user management).

**5. High-level Requirements**

* The application should provide CRUD operations for grocery items.
* The application should persist data using SQLite.
* The application should provide a simple API.

**6. Feasibility Analysis**

* **Technical Feasibility:** The required technologies (Python, FastAPI, SQLite) are lightweight, open source, and easily installed on a local machine.
* **Operational Feasibility:** The workflow is straightforward and meets basic user needs (simple grocery list).
* **Economic Feasibility:** There are no costs, as all tools and libraries are free and open source.

**7. Risk Analysis**

* **Risk 1 – Time Overrun:** There is limited time to implement all features.
  + *Mitigation:* Focus on CRUD API first; UI is secondary.
* **Risk 2 – Data Model Changes:** Adding new item attributes may require database changes.
  + *Mitigation:* Keep schema minimal in Version 1 (id, name, quantity, category, purchased).

**8. SMART Goals**

* **Specific:** Implement a grocery to-do list with CRUD functionality and a minimal frontend interface.
* **Measurable:** Deliver a working prototype with at least two core features (CRUD operations and persistent storage).
* **Achievable:** Build using lightweight technologies (FastAPI, SQLite, and HTML/JS)
* **Relevant:** Helps users organize shopping habits.
* **Time-Bound:**
  + By *September 21*: Planning and Requirements finalized.
  + By *October 5*: CRUD API fully functional and documentation complete.

**9. Resource Planning**

* **Human Resources:** 1 developer
* **Technical Resources:** Python 3, FastAPI, SQLite, GitHub
* **Time Resources:** until October 5th