**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.

**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**

* **Student Developer (Me):** Responsible for design, implementation, and documentation.
* **Professor:** Evaluates deliverables and verifies alignment with SDLC.
* **Future DevOps Pipeline:** The application will later be used for CI/CD demonstrations.

**4. Constraints and Assumptions**

* The project must be done individually and before the given deadline.
* Development will be done in **Python (FastAPI)** with **SQLite** for persistent storage.
* The application will run locally; cloud deployment is outside the initial scope.
* Single user environment (no multi-user management).

**5. 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.

**6. 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).

**7. SMART Goals**

* **Specific:** Implement a grocery to-do list with CRUD functionality.
* **Measurable:** All endpoints tested successfully using FastAPI’s Swagger UI.
* **Achievable:** The scope is minimal and achievable in the timeframe.
* **Relevant:** Provides a practical base for DevOps pipeline demonstrations.
* **Time-Bound:**
  + By *September 21*: Planning and Requirements finalized.
  + By *October 5*: CRUD API fully functional, documentation complete and repo contains at least 3 commits.

**8. Timeline / Iterations**

* **Iteration 1 (Planning & Requirements):** Create SDLC documentation (Planning, SRS).
* **Iteration 2 (Development):** Build CRUD API with SQLite persistence.
* **Iteration 3 (Optional Extension):** Add minimal HTML frontend.
* **Iteration 4 (Documentation & Reflection):** Finalize report, diagrams, README, and prepare for DevOps integration.