Final Project Proposal: Inventory Management System

1. Project Title

Efficient Inventory Management System Using Dynamic Arrays and Category Search Trees

2. Team Members

1. Seiya Genda
2. Kwan Ho
3. Zach Madison

---

3. Project Overview

This project aims to design and implement an efficient inventory management system that allows users to store, search, update, and manage product information. The system will utilize data structures learned from class, specifically a dynamic array for storing items and a tree-based structure for category-based searching. Each item in the inventory will be represented as an object containing:

- Name
- ID
- Price
- Quantity

Additionally, a Category Tree will be used to optimize search operations based on product categories, improving retrieval performance compared to linear search.

Primary implementation language: **Python**

Optional performance analysis and visualization language: **JAVA**

---

4. Objectives

The main objectives of this project include:

- Implement a dynamic array that automatically resizes to support expanding inventory data.
- Create an Item class with attributes (Name, ID, Price, and Quantity).
- Develop a tree-based category search algorithm for fast query operations.
- Allow users to perform essential inventory functions such as add, remove, update, and search.
- Conduct performance analysis comparing operations with different data sizes.
- Maintain clear documentation in a shared repository.

---

5. Methodology

Our solution will incorporate multiple algorithms and data structures:

Algorithm Implementation

- Dynamic Array for storing items, including automatic resizing using doubling strategy.
- Binary Search Tree (BST) or General Tree for category indexing and fast searching.
- Standard algorithms for array insertions, deletions, and searches.

Performance Analysis

- Measure execution time for:
    - Item insertion into dynamic array
    - Category-based tree search
    - Updating item data such as quantity and price
- Evaluate memory usage differences as the dataset scales (100 → 10,000 items)

Tools & Documentation

- Code stored on GitHub for collaboration
- Time and memory benchmarking scripts included in repository
- Commented code with a detailed README and performance results

6. Expected Outcomes

By the end of the project, we expect to achieve:

- A fully functional inventory system capable of handling large data efficiently
- Quantified performance metrics that show benefits of dynamic arrays and tree-based searching
- Real-world application example demonstrating learned data structure techniques
- Improved teamwork and version control experience

7. Timeline

Phase	Task	Deadline
Week 1	Finalize design + item structure	Nov 8
Week 2	Implement dynamic array + resizable logic	Nov 15
Week 3	Build category search tree + UI commands	Nov 22
Week 4	Performance testing + data recording	Nov 29
Week 5	Final documentation + code review	Dec 6

Conclusion

This project will provide hands-on experience applying data structure concepts to a practical and scalable software solution. With measurable performance improvements and clear system functionality, our Inventory Management System will demonstrate the importance of algorithm efficiency in real-world computing.