

Instructions for Project: Applying CRUD, Searching, and Sorting Algorithms

In this project, you will create a system that addresses a real-world problem by applying the concepts of **CRUD operations** (Create, Read, Update, Delete), **searching algorithms**, and **sorting algorithms**. The goal is to identify a problem in any domain (such as order management, inventory tracking, or student records) that can be solved using these basic concepts of computer science.

Your task is to implement the solution and provide clear documentation outlining your project's purpose and how the algorithms work to solve the problem.

Steps for Completing the Project

1. **Identify a Problem:** Choose a real-world problem or scenario that can benefit from CRUD operations, searching, and sorting. Some examples of problems you could solve:
 - **Inventory Management:** Keeping track of stock items, adding new items, searching for products, and sorting inventory.
 - **Order Management System:** Handling customer orders in various states (e.g., pending, completed), updating order statuses, searching for orders, and sorting by order numbers or timestamps.
 - **Student Record System:** Managing student information, adding or updating student records, searching for specific students, and sorting students based on certain criteria (e.g., GPA, name).
2. **Develop the System:** Implement the solution using CRUD operations:
 - **Create:** Add new data to your system.
 - **Read:** Display or retrieve data from your system.
 - **Update:** Modify or update existing data in the system.
 - **Delete:** Remove data from your system.
 - **Search:** Implement an algorithm to locate specific data (e.g., find a specific order or record).
 - **Sort:** Implement an algorithm to organize the data based on certain parameters (e.g., sort orders by number or sort students by name).
3. **Documentation:** After developing the system, document the following:
 - **Introduction:** Describe the problem you are solving, why it is important, and how your system addresses this problem.
 - **Objectives:** Outline the specific objectives of your system, including how CRUD, searching, and sorting algorithms are applied in your solution.

- **Discussion:** Discuss how your system works, the algorithms you used, how they solve the problem, and whether your objectives have been achieved.

Guidelines for Writing the Documentation

1. Introduction

The **Introduction** section should provide background information on the problem you are solving. Address the following points:

- **Relevance of the system:** Explain why this problem is significant in the real world and the challenges it presents.
- **Problem to be solved:** Describe the specific issue your system is solving (e.g., managing orders, tracking inventory, updating student records).
- **Purpose of your system:** Discuss how your system helps solve this problem, highlighting the need for efficient data management (e.g., CRUD, search, and sort).

Example:

"In many fast-food restaurants, managing orders and tracking their status can become a cumbersome task. This system aims to solve that problem by creating an efficient way to manage orders through CRUD operations, searching, and sorting. The system enables staff to quickly add, update, delete, search, and sort orders, improving the overall workflow."

2. Objectives

The **Objectives** section should clearly outline the goals of your system and what the application is designed to do. Ensure that you highlight the three main algorithms you will be using:

- **CRUD operations:** Explain the CRUD functionalities (Create, Read, Update, Delete) and what they do in the context of your system.
- **Searching Algorithm:** Explain how you will implement the search functionality (e.g., linear search, binary search) to locate specific data.
- **Sorting Algorithm:** Detail how you will sort the data (e.g., using a built-in sort function or an algorithm like bubble sort, merge sort).

Example:

"The main objective of the system is to provide an efficient way to manage orders. The objectives include:

- **Create:** Allow staff to add new orders to the system.
- **Read:** Display all orders in their respective categories (e.g., Preparing, Ready).
- **Update:** Enable staff to modify order numbers if necessary.
- **Delete:** Provide a way to remove completed orders.
- **Search:** Implement a search feature to locate specific orders.
- **Sort:** Sort the orders by number or timestamp to maintain order organization."

3. Discussion

The **Discussion** section should explain how the system works and evaluate if the objectives were met. Address the following points:

- **Algorithm Explanation:** Discuss how you implemented the CRUD operations, searching, and sorting algorithms. Provide explanations of the algorithms and why you chose them.
- **Effectiveness:** Describe how each feature contributes to solving the problem. For example, does sorting the list of orders make the system more efficient? Does searching for orders help find specific items faster?
- **Objective Achievement:** Reflect on whether the system meets the goals you outlined in the Objectives section. Did you successfully apply CRUD operations, searching, and sorting? Did you face any challenges in implementing these features?

Example:

"The system achieves all the outlined objectives. The CRUD operations are implemented using simple input fields and buttons that allow staff to interact with the order lists. The search function allows quick lookup of specific orders, while the sort function keeps the orders organized and easy to manage. The system was able to successfully handle orders in real-time, and the algorithms used helped ensure efficiency and accuracy."