### 1.Project Title: Library Management System

Description:

The Library Management System is a software application designed to automate the management of a library. It allows librarians to efficiently manage the books, customer, and transactions within the library system.

**Key Features:**

**Book Management:**

Add new books to the library inventory.

Update existing book details such as title, author, genre, and availability.

Remove books from the inventory.

Search for books by title, author, genre, or ISBN.

**customer Management:**

Add new customer to the system.

Update customer information such as name, contact details, and membership status.

Remove customer from the system.

**Transaction Management:**

Check out books to customer.

Return books back to the library.

Keep track of due dates and overdue fines.

Generate transaction reports.

Object-Oriented Design:

The Library Management System is designed using object-oriented principles, with various classes representing entities such as Book, customer, Transaction, and Library. Each class encapsulates data and behavior related to its respective entity, promoting modularity, reusability, and maintainability of the code.

Additionally, inheritance, polymorphism, and encapsulation are utilized to model relationships between classes and to achieve abstraction and code reuse.

The Library Management System utilizes file storage to persistently store data, facilitating the dynamic creation and destruction of objects during runtime.

### 2.Project Title: Online Food Ordering System

Description:

The Online Food Ordering System is a comprehensive software application designed to facilitate the ordering and delivery of food from various restaurants. It provides a user-friendly interface for customers to browse menus, place orders, make payments, and track delivery status. Additionally, it offers restaurant owners a platform to manage their menus, track orders, and handle customer feedback.

**Key Features:**

**1.User Registration and Authentication:**

Customers can create accounts.

Update customer information

Remove customer from the system.

Search for customer

**Menu Management:**

Restaurant owners can add, update, and remove items from their menus.

Customers can browse through available menus, filter items, and view details such as prices and descriptions.

**Ordering and Payment:**

Customers can add items to their carts, customize orders, and proceed to checkout.

**Order Tracking:**

Customers can track the status of their orders in real-time, from preparation to delivery.

Delivery personnel can update order statuses and provide estimated delivery times.

Object-Oriented Design:

The Online Food Ordering System is built using object-oriented programming principles, with classes representing entities such as Customer, Restaurant, Order, Payment, and Feedback. Each class encapsulates relevant data and functionality, promoting code reuse and modularity.

Inheritance and polymorphism are utilized to model relationships between classes and to enable the extension of functionality without modifying existing code. For example, different payment methods can be implemented as subclasses of the Payment class.

utilizes file storage to persistently store data, facilitating the dynamic creation and destruction of objects during runtime.

### 3. Supermarket Management System

Description:

The Supermarket Management System is a software application designed to automate and streamline the operations of a supermarket or grocery store. It provides features for managing inventory, processing sales transactions, tracking customer data, and generating reports for business analysis.

**Key Features:**

**Inventory Management:**

Add, update, search and remove products from the inventory.

Track product details such as name, category, price, quantity, and expiration date.

(optional Set alerts for low stock levels and restocking reminders.)

**Sales Transaction Processing:**

add or remove product to the customer's shopping cart.

Calculate the total cost of the purchase.

**Customer Management:**

Create, delete ,search and update customer accounts with details such as name, contact information, and purchase history.

**Employee Management:**

Manage employee accounts with roles and permissions (e.g., cashier, manager).

Track employee attendance, hours worked, and performance metrics.

Object-Oriented Design:

The Supermarket Management System is developed using object-oriented programming principles. Key classes include Product, Inventory, Customer, Employee, Transaction, ReportGenerator, and UserManager. These classes encapsulate data and functionality related to products, inventory management, customer interactions, employee management, and reporting.

Inheritance and polymorphism can be used to model relationships between different types of products (e.g., perishable, non-perishable) or employees (e.g., cashier, manager).

### 4. **Project Title: Contact Book Management System**

**Description:**  
The Contact Book Management System is a software application that allows users to manage their contacts using a linked list. The system provides functionality to add, update, delete, and search contacts, along with sorting contacts by name or phone number.

**Key Features:**

* **Contact Management:**
  + Add new contacts (name, phone number, email).
  + Update existing contact details.
  + Remove contacts from the list.
  + Search contacts by name or phone number.
* **Sorting and Searching:**
  + Sort contacts alphabetically by name.
  + Search contacts using linked list traversal.

**Object-Oriented Design:**  
The Contact Book is designed using a class structure with a Contact class (to store contact details) and a ContactBook class (to manage the linked list operations). Inheritance can be used for adding additional functionalities such as filtering by categories (family, friends, business).

### 5. **Project Title: Queue-based Ticketing System**

**Description:**  
The Queue-based Ticketing System simulates a queue in a ticket service environment. Customers are added to the queue when they request a ticket, and tickets are issued in the order of customer arrival (first-come, first-served).

**Key Features:**

* **Customer Queue Management:**
  + Add new customers to the queue.
  + Process customers in the order they joined the queue.
  + Allow customers to cancel their tickets and remove them from the queue.
* **Ticket Issuance:**
  + Assign ticket numbers to customers when they reach the front of the queue.
  + Display ticket issuance reports.

**Object-Oriented Design:**  
The system uses a Customer class for storing customer data and a TicketQueue class for managing queue operations. Queue operations (enqueue, dequeue) are implemented using Python’s collections module or a custom class.

### 6. **Project Title: Stack-based Palindrome Checker**

**Description:**  
This project involves writing a Python program that checks whether a given string is a palindrome using a stack and a queue. The system will compare characters from both ends of the string.

**Key Features:**

* **Palindrome Check:**
  + Use a stack to push characters from the string.
  + Use a queue to enqueue characters from the string.
  + Compare the stack and queue to check for a palindrome.
* **User Interface:**
  + Allow users to input a string and receive feedback if it is a palindrome.

**Object-Oriented Design:**  
The program uses a Stack and Queue class for managing the string characters. The palindrome checker is handled in a PalindromeChecker class, utilizing both data structures to verify the input string.

### 7. **Project Title: Binary Tree Traversal Visualizer**

**Description:**  
This project visualizes different tree traversal techniques (pre-order, in-order, post-order) for binary trees. It allows users to interactively input data and see the tree structure along with traversal steps.

**Key Features:**

* **Binary Tree Construction:**
  + Build a binary tree from user inputs.
* **Traversal Visualization:**
  + Visualize pre-order, in-order, and post-order traversal step-by-step.
  + Show nodes being visited in real-time.

**Object-Oriented Design:**  
The system uses a BinaryTreeNode class for tree nodes and a BinaryTree class for managing the tree structure and traversals. Visualization is handled with user-friendly outputs.

### 8. **Project Title: Queue-based Print Spooler**

**Description:**  
This system simulates a print spooler where print jobs are managed in a queue. Print jobs are processed based on their priority, and completed jobs are removed from the queue.

**Key Features:**

* **Queue Management:**
  + Add print jobs to the queue with assigned priorities.
  + Process jobs in priority order.
* **Job Tracking:**
  + Track and display the status of print jobs.

**Object-Oriented Design:**  
The program uses a PrintJob class to store job details and a PrintQueue class to manage job processing. Priority queues can be implemented for handling job priorities.