**README for Ignacio's Corporation Sales System**

**Overview**

This project is a Python-based system for managing and analyzing sales data across a fictional corporation, **Ignacio's Mexican Food**, with stores distributed across all 50 U.S. states. The system includes functionality for creating stores, customers, and orders, as well as generating comprehensive sales reports.

**Features**

* **Product Management**: Easily create products with names and prices.
* **Order Management**: Customers can add multiple products to their orders, calculate total prices, and print order details.
* **Customer Management**: Customers can place multiple orders and have their purchase history tracked.
* **Store Management**: Each store serves multiple customers and records their transactions.
* **Corporation Management**: The corporation aggregates sales data from all its stores and generates a detailed sales report.
* **Sales Reporting**: Outputs a CSV report with the details of all transactions, including order date, time, store ID, customer ID, products purchased, and total price.

**System Components**

**1. Product**

Represents individual products sold by the stores. In this case it’s Mexican food items.

* **Attributes**: name, price

**2. Order**

Tracks the products purchased in a transaction.

* **Attributes**: order\_id, products (list of products)
* **Methods**:
  + add\_product(product): Adds a product to the order.
  + total\_price(): Calculates the total price of the order.
  + print\_order(): Displays the order details.

**3. Customer**

Represents a customer with a unique ID and their purchase history.

* **Attributes**: customer\_id, orders (list of orders)
* **Methods**:
  + create\_order(order): Adds a new order to the customer’s history.

**4. Store**

Manages customer transactions for a specific location.

* **Attributes**: store\_id, customers (dictionary of customers)
* **Methods**:
  + add\_customer(customer): Adds a new customer to the store.
  + record\_order(customer, order): Records an order for a customer.

**5. Corporation**

Represents the overarching corporation that manages multiple stores.

* **Attributes**: name, stores (dictionary of stores)
* **Methods**:
  + add\_store(store): Adds a new store to the corporation.
  + record\_sale(store\_id, customer, order): Records a sale at a specific store.
  + generate\_sales\_report(filename): Generates a CSV sales report.

**Example Output**

**Order Details (Printed):**

Order ID: Order\_1234

Product: Burrito, Price: 8.99

Product: Soda Pop, Price: 1.99

Total Price: 10.98

**CSV Report Example:**

| **Date** | **Time** | **StoreID** | **CustomerID** | **OrderID** | **Products (Name**  **)** | **Total Price** |
| --- | --- | --- | --- | --- | --- | --- |
| 2024-11-12 | 14:35:22 | Ignacio's Texas | Customer\_1 | Order\_1234 | Burrito:8.99, Soda Pop:1.99 | 10.98 |

PROJECT 3 READ ME

**README: Sales Data Analysis Script**

**Overview**

This script analyzes sales data to uncover insights about product popularity, customer purchasing behavior, and sales trends. Key features include:

* **Product Popularity**: Identifies and visualizes the top 10 most popular products.
* **Basket Size**: Analyzes the number of items per transaction and visualizes the distribution.
* **Product Co-occurrence**: Highlights frequently purchased product pairs with a co-occurrence matrix and heatmap.
* **Sales Trends**: Tracks monthly sales trends (if a Date column exists).

**Requirements**

* Python 3.7+
* Libraries: pandas, matplotlib, seaborn

**Usage**

1. Update the file\_path variable with the dataset path.
2. Ensure the dataset includes:
   * OrderID
   * Products (Name:Price) in ProductName:Price format
   * (Optional) Date in YYYY-MM-DD format
3. Run the script to generate insights and visualizations.

**Outputs**

* Top products and basket size stats printed in the console.
* Visualizations:
  + Bar chart for top products
  + Histogram for basket sizes
  + Heatmap for product co-occurrence
  + Line chart for monthly sales trends