APEX

Example Apex Code for Transaction Record Creation:

```
public class GasTransactionService {
// Method to create a new gas transaction
public static void createGasTransaction(Id customerId, Decimal
quantity, Decimal
pricePerUnit) {
Gas_Transaction__c newTransaction = new Gas_Transaction__c(
Customer__c = customerId,
Quantity__c = quantity,
Price_Per_Unit__c = pricePerUnit,
Total_Amount__c = quantity * pricePerUnit,
Transaction_Date__c = Date.today()
);
insert newTransaction;
}
// Method to update loyalty points for a customer after a transaction
public static void updateCustomerLoyaltyPoints(Id customerId,
Decimal amountSpent) {
Customer__c customer = [SELECT Id, Loyalty_Points__c FROM
Customer__c WHERE Id =
```

```
:customerId LIMIT 1];
if (customer != null) {
customer.Loyalty_Points__c += (amountSpent / 100); // e.g., 1 point
per 100 spent
update customer;
}
}
}
```

Example Apex Code for Sending an Email Notification:

```
public class NotificationService {
public static void sendEmailNotification(Id customerId, String subject,
String messageBody)
{
Customer__c customer = [SELECT Id, Email__c FROM Customer__c
WHERE Id =
:customerId LIMIT 1];
if (customer.Email__c != null) {
Messaging.SingleEmailMessage mail = new
Messaging.SingleEmailMessage();
mail.setToAddresses(new String[] { customer.Email__c });
mail.setSubject(subject);
mail.setPlainTextBody(messageBody);
Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });
```

```
        }
    }

    // Trigger email after new transaction
    public static void sendTransactionNotification(Id transactionId) {
        Gas_Transaction__c transaction = [SELECT Id, Customer__r.Email__c,
        Total_Amount__c
        FROM Gas_Transaction__c WHERE Id = :transactionId LIMIT 1];
        if (transaction.Customer__r.Email__c != null) {
            String subject = 'Thank you for your purchase!';
            String body = 'Dear Customer, thank you for your recent purchase of
            gas. The total
            amount is ' + transaction.Total_Amount__c + '.';
            sendEmailNotification(transaction.Customer__r.Id, subject, body);
        }
    }
}
```

Example: Predicting Next Refill Date Based on Past Purchases

```
public class PredictiveAnalyticsService {
    // Method to predict the next gas refill date based on past purchase
    behavior
    public static Date predictNextRefill(Id customerId) {
        List<Gas_Transaction__c> transactions = [SELECT
        Transaction_Date__c, Quantity__c
```

```
FROM Gas_Transaction__c
WHERE Customer__c = :customerId
ORDER BY Transaction_Date__c DESC
LIMIT 5];
if (transactions.size() > 1) {
// Calculate average time between transactions
Integer totalDaysBetweenPurchases = 0;
for (Integer i = 0; i < transactions.size() - 1; i++) {
totalDaysBetweenPurchases +=
transactions[i].Transaction_Date__c.daysBetween(transactions[i+1].Transaction_Date__c);
}
Integer avgDaysBetweenPurchases = totalDaysBetweenPurchases /
(transactions.size()
- 1);
return
transactions[0].Transaction_Date__c.addDays(avgDaysBetweenPurchases);
}
return null; // Not enough data to make a prediction
}
}
```

Example: Automatic Low Stock Alert

```apex
public class InventoryService {
public static void checkAndAlertLowStock() {
List<Product__c> lowStockItems = [SELECT Id, Name, Stock_Level__c
FROM Product__c
WHERE Stock_Level__c < 10];
if (lowStockItems.size() > 0) {
for (Product__c product : lowStockItems) {
sendLowStockAlert(product.Name);
}
}
}
// Sends an email to store owner when stock is low
private static void sendLowStockAlert(String productName) {
String emailAddress = 'storeowner@example.com'; // Store owner's
email
Messaging.SingleEmailMessage mail = new
Messaging.SingleEmailMessage();
mail.setToAddresses(new String[] { emailAddress });
mail.setSubject('Low Stock Alert: ' + productName);
mail.setPlainTextBody('The stock level of ' + productName + ' is
below the threshold.
Please restock soon.');
Messaging.sendEmail(new Messaging.SingleEmailMessage[] { mail });
```

```
}
}
```

Example: Loyalty Points Tracking

```
trigger LoyaltyPointsTrigger on Gas_Transaction__c (after insert) {
    for (Gas_Transaction__c transaction : Trigger.New) {
        GasTransactionService.updateCustomerLoyaltyPoints(transaction.Customer__c,
        transaction.Total_Amount__c);
    }
}
```