# Marketlytics Assignment Question 3 Response:

This document will propose a solution to automate the process highlighted in the question given. To break things down I will describe a system architecture and database design we can make use of to automate the current process.  
  
The first step in the current process, i.e. getting to know the vendor, making them fill in forms etcetera can become the first step of our automation. We can make use of a web based UI where the vendors can fill in their detail. This can be something as simple as a shared google sheet (based on a template with empty spaces for the vendor to fill out), which we can import into our backend and work with, to something like a customized portal.  
  
The next step would be how we process the data our vendors just sent us, this step would include data processing, managing interactions with the database, performing any validations correlated with making sure the business is legitimate. We can split it up into modules for instance, one module can work on processing vendors’ submissions and validating KYC documents and updating the data base. Another can manage agreement details and track signatures. And lastly we can also have one to handle SKU catalog submission by processing the catalog data and storing it in the database. We can ask the vendors to send us this data either using the same UI portal as mentioned earlier or can be made into a separate portal for ease).  
  
By this point we have automated the process of getting data from the vendor and how we process it, alongside the validation needed to legitimize the vendor. In addition to this we have also achieved agreement management. The next thing to consider is how we store the data that we have processed using the backend i.e. the database design. Let’s go into some detail here.  
  
Let’s start off by defining our tables, Vendor Table, Document Table, Agreement Table and SKU catalog Table:

**Vendor Table:**

Stores basic information about each vendor.

Fields:

**VendorID** (Primary Key): Unique identifier for each vendor.

**BusinessName**: Name of the vendor's business.

**ContactPerson**: Name of the contact person representing the vendor.

**ContactDetails**: Contact information for the vendor.

**KYCStatus**: Status of KYC document verification (Pending, Verified, Rejected).

**AgreementStatus**: Status of the agreement signing process (Pending, Signed).

Other relevant fields.

**Document Table:**

Stores details about KYC documents uploaded by vendors.

Fields:

**DocumentID** (Primary Key): Unique identifier for each document.

**VendorID** (Foreign Key): Relates to the Vendor table to establish a one-to-many relationship.

**DocumentType**: Type of document (e.g., ID proof, Business registration).

**FilePath**: Path to the stored document file.

**VerificationStatus**: Status of document verification (Pending, Verified, Rejected).

Other relevant fields.

**Agreement Table:**

Stores details about agreements signed by vendors.

Fields:

**AgreementID** (Primary Key): Unique identifier for each agreement.

**VendorID** (Foreign Key): Relates to the Vendor table.

**AgreementDetails**: Details of the agreement.

**Signature**: Digital signature or acknowledgment of the agreement.

**Status**: Status of the agreement (Pending, Signed).

Other relevant fields.

**SKU Catalog Table:**

* Stores information about SKU catalogs submitted by vendors.
* Fields:
  + **SKUID** (Primary Key): Unique identifier for each SKU entry.
  + **VendorID** (Foreign Key): Relates to the Vendor table.
  + **SKUName**: Name or identifier of the SKU.
  + **Price**: Price of the SKU.
  + **ImagePath**: Path to the image file associated with the SKU.
  + Other relevant fields.

As for the relationships between each of these tables:

* **Vendor-Document Relationship:**
  + Each vendor can have multiple KYC documents (one-to-many relationship).
  + Foreign key **VendorID** in the Document table relates to the **VendorID** in the Vendor table.
* **Vendor-Agreement Relationship:**
  + Each vendor can have multiple agreements (one-to-many relationship).
  + Foreign key **VendorID** in the Agreement table relates to the **VendorID** in the Vendor table.
* **Vendor-SKU Catalog Relationship:**
  + Each vendor can submit multiple SKU entries (one-to-many relationship).
  + Foreign key **VendorID** in the SKUCatalog table relates to the **VendorID** in the Vendor table.