**All MS API Endpoints**

**Note: Endpoint, Response format is for sample and can be changes accordignly**

1. **Product-Service:**

**Mandatory**:

1. **Get All Products (Paging can be included)**
   * Endpoint: GET /api/products
   * Description: Retrieve a list of all products available in the system.
   * Response: JSON representation of all products, including basic product details.
   * It will call Inventory API to get SKU and Prices
2. **Get Product by ID**
   * Endpoint: GET /api/products/{product\_id}
   * Description: Retrieve the details of a specific product by its unique identifier.
   * Response: JSON representation of the product details, including name, price, description, etc.
   * It will use Inventory API to get SKU and Prices
3. **Create Product**
   * Endpoint: POST /api/products
   * Description: Create a new product/products and add it to the system.
   * Request Body: JSON representation of the product details to be created.
4. **Delete Product**
   * Endpoint: DELETE /api/products/{product\_id}
   * Description: Delete a specific product/products from the system.
5. **Search Products**
   * Endpoint: GET /api/products/search
   * Description: Search for products based on specific criteria (e.g., name, category, price range, etc.).
   * Request Parameters: Search criteria such as name, category, price range, etc.
   * Response: JSON representation of the search results.

**Optional**:

1. **Get Products by Category**
   * Endpoint: GET /api/products/category/{category\_id}
   * Description: Retrieve a list of products belonging to a specific category.
   * Response: JSON representation of products in the given category.
2. **Get Product Reviews**
   * Endpoint: GET /api/products/{product\_id}/reviews
   * Description: Retrieve the reviews for a specific product.
   * Response: JSON representation of product reviews.

**2. Inventory-Service**

**Mandatory**:

1. **Get All Inventory**
   * Endpoint: GET /api/inventory
   * Description: Retrieve all inventory for products available in the system.
   * Response: JSON representation of all inventory, including basic SKU, pricing details.
2. **Get Inventory by ID**
   * Endpoint: GET /api/inventory/{inventory\_id}
   * Description: Retrieve the details of a specific inventory by its unique identifier.
   * Response: JSON representation of the product details, including name, price, description, etc.
3. **Verify Inventory**
   * Endpoint: GET /api/inventory/{inventory\_id}
   * Description: Verify the details and inventory in System.
4. **Delete Inventory**
   * Endpoint: DELETE /api/inventory/products/{product\_id}
   * Description: Delete all inventory for specific product/products from the system.

**Optional:**

1. **Update Inventory**
   * Endpoint: DELETE /api/inventory/{inventory\_id}/products/{product\_id}
   * Description: update inventory specific inventory and specific product

**3. Customer-Service:**

**Mandatory**:

1. **Create User**
   * Endpoint: POST /api/users
   * Description: Create a new user account.
   * Request Body: JSON representation of user details such as username, email, password, etc.
2. **Get User by ID**
   * Endpoint: GET /api/users/{user\_id}
   * Description: Retrieve the details of a specific user by their unique identifier.
   * Response: JSON representation of the user details, excluding sensitive information like password.
3. **Update User**
   * Endpoint: PUT /api/users/{user\_id}
   * Description: Update the details of a specific user.
   * Request Body: JSON representation of the updated user details.

**Optional**:

1. **Delete User**
   * Endpoint: DELETE /api/users/{user\_id}
   * Description: Delete a specific user account from the system.

**4. Cart-Service**

**Mandatory**:

1. **Add Item to Cart**
   * Endpoint: POST /api/cart/{cart\_id}/items
   * Description: Add an item to the user's shopping cart.
   * Request Body: JSON representation of the item to be added.
2. **Remove Item from Cart**
   * Endpoint: DELETE /api/cart/items/{item\_id}
   * Description: Remove an item from the user's shopping cart.
3. **Get Cart Size**
   * Endpoint: GET /api/cart/{cart\_id}/size
   * Description: Retrieve the number of items in the user's cart.
4. **View Cart**
   * Endpoint: GET /api/cart/{cart\_id}
   * Description: Retrieve the user's shopping cart with all the items currently added.
5. **Clear Cart**
   * Endpoint: DELETE /api/cart/{cart\_id}
   * Description: Clear the user's shopping cart, removing all items.

**Optional**:

1. **Update Item in Cart**
   * Endpoint: PUT /api/cart/items/{item\_id}
   * Description: Update the quantity or any other attribute of an item in the user's shopping cart.
   * Request Body: JSON representation of the updated item details.

**5. Order-Service**

**Mandatory**:

1. **Place Order**
   * Endpoint: POST /api/orders
   * Description: Create a new order for the items in the user's shopping cart.
   * Request Body: JSON representation of the order details, including cart items, shipping address, payment information, etc.
2. **Get Order by ID**
   * Endpoint: GET /api/orders/{order\_id}
   * Description: Retrieve the details of a specific order by its unique identifier.
   * Response: JSON representation of the order details, including order status, items, shipping information, etc.
3. **Get User Orders**
   * Endpoint: GET /api/orders/user/{user\_id}
   * Description: Retrieve a list of orders for a specific user.
   * Response: JSON representation of the user's order history, including order details for each order.
4. **Update Order Status**
   * Endpoint: PUT /api/orders/{order\_id}/status
   * Description: Update the status of a specific order (e.g., "processing," "shipped," "delivered," etc.).
   * Request Body: JSON representation of the updated order status.
5. **Get All Orders**
   * Endpoint: GET /api/orders
   * Description: Retrieve a list of all orders in the system (for admin purposes).
   * Response: JSON representation of all orders, including basic order details.

**Optional:**

1. **Cancel Order**
   * Endpoint: DELETE /api/orders/{order\_id}
   * Description: Cancel a specific order.
   * Response: JSON representation confirming the order cancellation.
2. **Get Order Payment Status**
   * Endpoint: GET /api/orders/{order\_id}/payment-status
   * Description: Retrieve the payment status of a specific order.
   * Response: JSON representation of the payment status.
3. **Get Order Shipping Status**
   * Endpoint: GET /api/orders/{order\_id}/shipping-status
   * Description: Retrieve the shipping status of a specific order.
   * Response: JSON representation of the shipping status.

**6. IAM-Service**

**Mandatory**:

1. **Create User**
   * Endpoint: POST /api/user/register
   * Description: Create a new user account with customer\_id and mobile number
   * Request Body: JSON representation of user details such as customer\_id and mobile number
2. **Login User**
   * Endpoint: POST /oauth/token
   * Description: Authenticate user and Generate Token
   * Request Body: JSON representation of user details such as mobile number
3. **Refresh/Generate Token**
   * Endpoint: POST /oauth/token
   * Description: Authenticate user and ReGenerate Token, by sending refresh token- auth\_type
   * Request Body: JSON representation of user details such as customer\_id and mobile number
4. **Validate Token**
   * Endpoint: POST /oauth/check\_token
   * Description: Validate Token
   * Request Body: JSON representation of user details such as customer\_id and mobile number

**Optional**:

**7. Shipment-Service**

**Mandatory:**

1. **Create Shipment**
   * Endpoint: POST /api/logistics/shipments
   * Description: Create a new shipment for an order.
   * Request Body: JSON representation of the shipment details, including order ID, shipping address, etc.
2. **Get Delivery Estimates**
   * Endpoint: GET /api/logistics/delivery-estimates
   * Description: Retrieve estimated delivery times based on the shipping address and selected shipping method.
   * Request Parameters: Shipping address details, shipping method, etc.
   * Response: JSON representation of delivery estimates.

**Optional:**

1. **Get Shipment Status**
   * Endpoint: GET /api/logistics/shipments/{shipment\_id}/status
   * Description: Retrieve the current status of a shipment.
   * Response: JSON representation of the shipment status, including tracking information.
2. **Update Shipment Status**
   * Endpoint: PUT /api/logistics/shipments/{shipment\_id}/status
   * Description: Update the status of a shipment (e.g., "shipped," "out for delivery," "delivered," etc.).
   * Request Body: JSON representation of the updated shipment status.
3. **Cancel Shipment**
   * Endpoint: DELETE /api/logistics/shipments/{shipment\_id}
   * Description: Cancel a shipment and update the order status accordingly.
4. **Get Shipping Methods**
   * Endpoint: GET /api/logistics/shipping-methods
   * Description: Retrieve available shipping methods with their details and costs.
   * Response: JSON representation of available shipping methods.
5. **Calculate Shipping Cost**
   * Endpoint: POST /api/logistics/shipping-cost
   * Description: Calculate the shipping cost for an order based on the selected shipping method and address.
   * Request Body: JSON representation of the shipping method and address details.
   * Response: JSON representation of the calculated shipping cost.
6. **Get Order Details**
   * Endpoint: GET /api/logistics/orders/{order\_id}
   * Description: Retrieve the details of a specific order, including shipment information.
   * Response: JSON representation of the order details, including shipment status.

**8. Payment-Service**

**Mandatory:**

1. **Get Payment Methods**
   * Endpoint: GET /api/payment-methods
   * Description: Retrieve a list of available payment methods supported by the system.
   * Response: JSON representation of the available payment methods.
2. **Get Payment Status for Order**
   * Endpoint: GET /api/payments/order/{order\_id}/status
   * Description: Retrieve the payment status for a specific order.
   * Response: JSON representation of the payment status for the order.
3. **Refund Payment**
   * Endpoint: POST /api/payments/{payment\_id}/refund
   * Description: Initiate a refund for a specific payment.
   * Request Body: JSON representation of the refund details, including the refund amount and reason.
4. **Update Payment Status**
   * Endpoint: PUT /api/payments/{payment\_id}/status
   * Description: Update the status of a specific payment (e.g., "pending," "completed," "failed," etc.).
   * Request Body: JSON representation of the updated payment status.

**Optional:**

1. **Get Payment by ID**
   * Endpoint: GET /api/payments/{payment\_id}
   * Description: Retrieve the details of a specific payment by its unique identifier.
   * Response: JSON representation of the payment details, including status, payment method, amount, etc.
2. **Get Order Payments**
   * Endpoint: GET /api/payments/order/{order\_id}
   * Description: Retrieve a list of payments made for a specific order.
   * Response: JSON representation of the payments associated with the order.