List of APIs

Serial No.	**Entity Name**	**API Name**	**API Purpose**
1	USERS	registerUser	Create a new **user**.
2		updateUser	Update any field(s) of the **users** entity
3		getOneUser	Get information on a specific user based on **user id** or **first name** and **last name** or **email**.
4		getAllUsers	Get all users using any field except **password**.
5		deleteUser	Delete a user after deleting all respective **orders, reviews, shipping addresses and cart items**.
6	CartItems	createCartItem	Use **user_id** to select the user. Check the **product** and **product item** for the respective **product id** and **sku** and set the **cart_item_price** according to **product_price** and **cart_item_quantity** according to **current_stock** before creating the cart item. *Set the **cart_item_image_uri** based on the **main_image** in ProductImages in the respective product.*
7		getCartItemById	Get a particular cart item by **cart_item_id** and

			user_id.
8		getCartItemByIdSku	Get a cart item using **product_id, sku** and **user_id**.
9		searchCartItem	Get all cart items using **cart_item_price** or **cart_item_quantity**.
10		updateCartItem	Check the **cart_item_price** with **product_price** and **cart_item_quantity** with **current_stock** before updating it. Update the **cart_item_image, cart_item_price** or **cart_item_quantity** of a CartItem, selected using **cart_item_id** and **user_id**. *When updating the **cart_item_image**, do so by checking the **main_image** in ProductImages in the respective product.*
11		deleteCartItem	Delete a CartItem by **cart_item_id** and **user_id**.
12	ShippingAddresses	createShippingAddress	Create a new shipping address in the user, selected using **user_id**.
13		getShippingAddressById	Get a shipping address by **shipping_address_id** and **user_id**.
14		searchShippingAddress	Search shipping addresses by **city** OR **country** OR

			postal_area OR **building_name**. Select the user using **user_id**.
15		updateShippingAddress	Update any field of a ShippingAddress except **shipping_address_id**. Use **user_id** to select the user.
16		deleteShippingAddress	Delete a shipping address selected using **shipping_address_id** and **user_id**.
17	ORDERS	createOrder	Create an order after checking if the respective user exists and **invoking createOrderItem to insert the respective user's cart items in the order.**
18		updateOrderStatus	Update the **order_status** of an order using **user_id** and **order_id**.
19		updateOrderDateOfArrival	Update the **date_of_arrival** of an order using **user_id** and **order_id**.
20		updateOrderShippingAddress	This can be done only if the **order_status** is **Pre-Dispatch**. Update the shipping address of an order (1 or more fields) using **user_id** and **order_id**.
20a		updateOrderTotalPrice	This can be done only if the

			order_status is **Pre-Dispatch**. Updates the **total_price** by adding the **order_item_price** fields of all the order items. This API is invoked by **createOrderItem** or **deleteOrderItem**.
21		deleteOrder	Check if the **order_status** is **“Pre-Dispatch”.** Deletes an order and all its order items using **user_id** and **order_id**.
22		getOrder	Get a particular order using **user_id** and **order_id**.
23		searchOrders	Gets all orders of a particular user using **user_id**. Can filter orders using **order_status**, **total_price**, **date_created_at** or **date_of_arrival** or **shipping_address**.
24		searchOrdersUniversal	Gets all orders of all users using **user_id**. Can filter orders using **order_status**, **total_price**, **date_created_at** or **date_of_arrival** or **shipping_address**.
When inserting order items, we do not need to check with product and product item as those verifications are already done while creating the respective cart items.			
25	OrderItems	createOrderItem	Check that the **order_status** of the order is “Pre-Dispatch”.

			Check that a cart item with the respective **product_id** and **sku** is present in the respective user’s cart through **user_id**. Then, create the order item and delete the respective cart item by calling **deleteCartItem**. Then, call the **updateOrderTotalPrice** API to update the **total_price** of the respective order. Then, update the **total_stock**, **quantity_sold** and **current_stock** fields in the respective product item.
26		updateOrderItem	Check whether the **order_status** is “Pre-Dispatch”. If yes,then, check the field to be updated (**order_item_quantity**) with its equivalent in the respective product item using **product_id**, **sku**, **product_price** and **current_stock** and the respective user’s **user_id**. Then, update the order item. Then, call the **updateOrderTotalPrice** API to update

			the **total_price** of the respective order. Then, update the **total_stock**, **quantity_sold** and **current_stock** fields in the respective product item.
27		getOrderItem	Gets a particular order item of a user using **user_id** + **order_item_id** OR **user_id** + **product_id** + **sku** instead.
28		searchOrderItems	Get all order items of a particular user using **user_id** and **order_id**. Can filter them using **order_item_price**, **order_item_qty** or **order_item_image_uri**.
29		deleteOrderItem	Check whether the **order_status** is “Pre-Dispatch”. Then, using **user_id** of the selected user, delete the respective order item identified by its **order_item_id** / **product_id** and **sku**. Then, call the **updateOrderTotalPrice** API to update the **total_price** of the respective order. Then, update the **total_stock**, **quantity_sold** and **current_stock** fields in the respective product

			item.
30	PRODUCTS	createProduct	Create a product and call createProductItem and createProductImage to create at least 1 product image and 1 product item.
31		updateProductPrice	When updating the **product_price,** update the **cartitem_price** of the respective user. The product is selected using **product_id**. The cart item of the respective user (**user_id**) is selected using **product_id** and **sku**.
32		updateProduct	Used to update any product field except **product_price**, **product_items** or **product_images**.
33		getProduct	Get a particular product using **product_id**.
34		searchProducts	Search products using any field except **product_images** or **product_items**.
35		deleteProduct	Delete a product using **product_id. Then call the respective APIs to delete all cart items, product items,** *order items* **and reviews.**
36	ProductItems	createProductItem	Create a product

			item using **product_id** in the respective product.
37		updateProductItem	When updating a product item, update the corresponding cart item and *order item* using **product_id** and **sku**. Update corresponding CartItem in USER. (**cart_item_quantity**) if the **current_stock** falls below the **cart_item_quantity**. Do this globally.
38		deleteProductItem	When deleting a product item, call the corresponding APIs to delete the respective cart item and *order item*.
39		getProductItem	Get a particular product item using **product_id** and **sku**.
40		searchProductItem	Search and filter product items through any field except **sku**.
41	ProductImages	createProductImage	Select the product for which the image is to be created through **product_id**. Create the product image URI and set the **main_image** indicator if appropriate.
42		updateProductImage	If the **image_ur**i being updated is that of a main image, then the

			corresponding **order_item_image_uri** and **cart_item_image** should also be updated.
43		searchProductImage	Search product images on the basis of the **image_uri**, **main_image** and **image_id**.
44		getProductImage	Get a product image using **image_id** or **image_uri**.
45		deleteProductImage	If the **image_ur**i being deleted is that of a main image, then the corresponding **order_item_image_uri** and **cart_item_image** should also be deleted.
46	REVIEWS	createReview	Create a review for a given product and user, using **product_id** and **user_id.**
47		updateReview	A review can have its **review_comment** and **review_rating** fields updated using **product_id** and **user_id**.
48		searchReview	Reviews can be searched using the **review_comment** or **review_rating** fields.
49		getReview	Get a single review using its **review_id** and **user_id**.
50		deleteReview	A review is deleted using its **review_id** & **user_id.**