

Phase Name	Start Date	End Date	Requirements
Phase 1	2025-03-25	2025-03-28	Overview, High Level Description, Domain Design, Application Wide Requirements
Phase 2	2025-03-31	2025-04-07	Register APIs, Login/Logout APIs, Forgot Password APIs
Security Session	2025-04-04	2025-04-04	
Phase 3	2025-04-08	2025-04-14	Admin APIs, Seller APIs, Customer APIs
Phase 4	2025-04-15	2025-04-21	Category APIs, Product APIs

User
A user can register into the system.
A user can reset his password using the forgot your password functionality
A user can login/logout in the system
Admin
An admin can edit his profile such as, name image, password etc.
An admin can list all the registered customers and sellers in the system.
An admin can change the activation status, roles of any customer or seller.
An admin can list all the products which are created in the system
An admin can change the activation status of any product which is listed in the system
An admin can change the status of the order which is placed.
Customer
A customer can edit his profile such as, name, image, addresses, password etc.
A customer can list all the products which are made available by the sellers
A customer can view in-dept detail of a product such as, images, reviews, product-information, price, seller details etc
A customer can add the product either in its cart or in his wish-list or move the product in between the two
A customer can place an order for the products available in his cart
A customer can view all of his orders.
A customer can cancel a placed order or make a request to return a delivered order
Seller
A seller can edit his profile such as, company details, password etc.
A Seller will only have one address associated
A seller can list all the products he has added in his catalogue
A seller can add new product, make changes in his existing product or delete a product.
A seller can change the status of the order for his products as placed by any customer
Product
A product can have product-level metadata along with variations-level metadata defined, where variation's metadata takes priority over product's.
A product should have all its variation's primary images displayed when is listed.
Product price is by default in INR

Feature		
Validations	Password	8-15 Characters with atleast 1 Lower case, 1 Upper case, 1 Special Character, 1 Number
Emails	Asynchronous Emails	All emails triggered from the application should be done asynchronously, thus not pausing the API flow
Response Codes	404	If Url is not found If no object is found for given Id
	401	If there is no logged-in user found while accessing the APIs
	403	If the logged-in user does not have the required authorization to access the APIs
	500	For any exception/error occuring within the Services
	405	If wrong method type is used to access the API
Database	Data deletion	Do a soft delete to preserve the data in the tables using a IS_DELETED flag
	Primary Key	Use Long ID or UUID. Use hibernate_sequence table to generate an unique Id across all the tables if using Long ID.
	Auditing Information	Add the auditing information in all the necessary tables such as, date_created last_updated created_by updated_by
	User Security	Add the security (spring security) related fields in the user and role table such as, account_locked, password_expired etc
	Password	Should be stored in encoded form. No plain text password
User tokens	access_token	It should be present in all the APIs which requires an authenticated logged-in user and is sent as the request header Time based tokens. Configurable. In case if access_token expires then get the new access_token with the help of refresh_token (Valid for 24 hours)
	reset_password_token	Time based tokens. Configurable. (Valid for 15 Minutes)
	activation_token	Time based tokens. Configurable. Sent in a mail. For new customer registrations. Valid for 3 hours
User accounts	Admin Account	One Admin Account. Cannot be disabled, deleted or inactivated. Created using bootstrap.
	Seller Account	User when registering as a seller is granted the SELLER role by default. New Seller accounts needs to be activated by an admin user, before using the system.
	Customer Account	User when registering as a customer is granted the CUSTOMER role by default. New Customer accounts needs to be activated using the activation token, before accessing the system
Internationalization		Use of message.properties pertaining to different languages
Roles		Currently, the app supports 3 types of roles, CUSTOMER SELLER ADMIN

		<p>Images will be saved in a common file system directory.</p> <p>No reference needed to be saved in the database. The path can be computed.</p> <p>Example:</p> <p>For users, <BASE_PATH>/users/<user_id>.extention</p> <p>For products, <BASE_PATH>/products/<product_id>/<product_id>.extention</p> <p>For product variations, <BASE_PATH>/products/<product_id>/variations/<variation_id>.extention</p> <p>etc.</p>	
Images	File system based		
	Formats	jpg, jpeg, png, bmp	
Order Status		<p>Kept as Enum.</p> <p>An order can be cancelled as long as it is not delivered.</p> <p>An order can be returned or requested for refund once it is delivered</p>	
Cart/Wishlist		<p>A product can be added in the cart only if it has available quantities.</p> <p>A product can be added in the wish-list even if the product is unavailable</p>	
Pagination & Sorting	max offset sort order query	<p>Default page size is of 10</p> <p>Default page offset is of 0</p> <p>Default sorting on Id, or date_created, or email (wherever applicable)</p> <p>Default order is ascending</p> <p>Default query is empty. If present, the result needs to be filtered out on the basis of it. It contains pairs of field and values</p>	
Scheduling		A scheduler will run every day at 00:00 and send email to seller for all the orders that are not accepted/rejected.	

USER	ROLE	USER_ROLE	SELLER	CUSTOMER	ADDRESS	
ID	ID	USER_ID	USER_ID	USER_ID	ID	
EMAIL	AUTHORITY	ROLE_ID	GST	CONTACT	CITY	
FIRST_NAME			COMPANY_CONTACT		STATE	
MIDDLE_NAME			COMPANY_NAME		COUNTRY	
LAST_NAME					ADDRESS_LINE	
PASSWORD					ZIP_CODE	
IS_DELETED					LABEL (Ex. office/home)	
IS_ACTIVE					USER_ID	
IS_EXPIRED						
IS_LOCKED						
INVALID_ATTEMPT_COUNT						
PASSWORD_UPDATE_DATE						
CATEGORY	CATEGORY_METADATA_FIELD	CATEGORY_METADATA_FIELD_VALUES	PRODUCT	PRODUCT_VARIATION	PRODUCT_REVIEW	
ID	ID	CATEGORY_METADATA_FIELD_ID	ID	ID	CUSTOMER_USER_ID	
NAME	NAME	CATEGORY_ID	SELLER_USER_ID	PRODUCT_ID	REVIEW	
PARENT_CATEGORY_ID (This is category directly preceding this category)		VALUES (Comma separated string saved in database)	NAME	QUANTITY_AVAILABLE	RATING	
	(NEW TABLE)	(NEW TABLE)	DESCRIPTION	PRICE	PRODUCT_ID	
			CATEGORY_ID * Only the leaf category node would be associated with a product	"METADATA (Type: JSON - available in mysql to store a JSON as it is.) (Note: will contain all the information regarding variations in flat JSON format) (All variations of same category will have a fixed similar JSON structure. The fields for the json should be picked from CATEGORY table)"		
	1 - size		IS_CANCELABLE	PRIMARY_IMAGE_NAME		
	2 - colour					
4- shirt		1 (size)				
5- shoe		4 (shirt)				
	3 - length	Values M, L, XL	IS_RETURNABLE	IS_ACTIVE (NEW FIELD)		
			BRAND			
			IS_ACTIVE	{size - 28, color -red, length xcm} , quant- 5 {size - 28, color -red, length xcm} , quant- 5 {size - 28, color -blue, length xcm}, quant -10		
			IS_DELETED			
e.g Mobile, Electronics, Grocery		e.g. Shirt: size-XS,S,M,L,XL Shirt: color-Blue,red,White Tshirt: Size-26,28,30,32 Shoes: Size-8,9,10,11	e.g Redme, One Pluss	e.g 32 GB dual cam, 64 GB dual cam etc e.g P1-Shirt-Blue-M, P1-Shirt-White-M, P1-Blue-L		
Eg. Fashion -> Shoes Fashion -> Shirt Fashion - Tshirt	e.g. Size, Colour, Length, Fabric, Heel Size, Sleeve Length					
			1 --- 1	1 ---- 1 (thsirt) - 2		
				1 ----- 2 (jeans) - 1		
CART	ORDER	ORDER_PRODUCT	ORDER_STATUS			
CUSTOMER_USER_ID	ID	ID	ORDER_PRODUCT_ID			
QUANTITY	CUSTOMER_USER_ID	ORDER_ID	FROM_STATUS			
IS_WISHLIST_ITEM	AMOUNT_PAID	QUANTITY	TO_STATUS		JAVA ENUM	JAVA ENUM
PRODUCT_VARIATION_ID	DATE_CREATED	PRICE	TRANSITION_NOTES_COMMENTS		From Status	To Status
	PAYMENT_METHOD	PRODUCT_VARIATION_ID	TRANSITION_DATE		ORDER_PLACED	CANCELLED
	CUSTOMER_ADDRESS_CITY					ORDER_CONFIRMED
	CUSTOMER_ADDRESS_STATE					ORDER_REJECTED
	CUSTOMER_ADDRESS_COUNTRY				CANCELLED	REFUND_INITIATED
	CUSTOMER_ADDRESS_ADDRESS_LINE					CLOSED
	CUSTOMER_ADDRESS_ZIP_CODE				ORDER_REJECTED	REFUND_INITIATED
	CUSTOMER_ADDRESS_LABEL (Ex. office/home)					CLOSED
					ORDER_CONFIRMED	CANCELLED
						ORDER_SHIPPED
	This is Consolidated Bill	This contains individual items for consolidated bill			ORDER_SHIPPED	DELIVERED
					DELIVERED	RETURN_REQUESTED
						CLOSED
					RETURN_REQUESTED	RETURN_REJECTED
						RETURN_APPROVED
					RETURN_REJECTED	CLOSED
					RETURN_APPROVED	PICK_UP_INITIATED
					PICK_UP_INITIATED	PICK_UP_COMPLETED
					PICK_UP_COMPLETED	REFUND_INITIATED
					REFUND_INITIATED	REFUND_COMPLETED
					REFUND_COMPLETED	CLOSED

FEATURE 1: As a customer, I should be able to register a new account								
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
1	API to register a customer	POST	1) User hits an API with the mentioned manadatory fields. 2) Validation fails with appropriate response containing the validation error(s)	Email should be unique Email should be valid Phone number should be valid Password should be valid Confirm Password should match Password	Email Phone Number Password Confirm Password First Name Last Name	List of validation erros	Should validate Email, phone number and password for correctness and email for uniqueness	
			1) User hits an API with the following manadatory fields. 2) Customer is successfully registered with an inactive account and a success response message should be sent. 3) A unique time based token should be generated for account activation. 4) An async email is triggered to the customer containing the link with a unqiue activation token.			Success message		
2	API to activate the customer	PUT	1) User hits an API with an invalid token. 2) The response should be sent with an appropriate error message. 3) The user should not be activated.	Activation Token should be valid	Activation Token			An async mail is triggered informing the customer about account activation
			1) User hits an API with an expired token. 2) Generate new token and send another email to user with new token 3) Delete this token 4) The response should be sent with an appropriate error message. 5) The user should not be activated.			Error message		
			1) User hits an API with a valid activation token. 2) The customer account gets activated. 3) The token should be deleted from the system. 4) The response sent should contain a success message			Success message		
3	API to re-send activation link	POST	1) User hits the API with his email address. 2) The validation fails. 3) Response should obtain the appropriate validation errors	Account should be de-activated Email should be valid Email should exist in the database	Email Address	Error Message		
			1) User hits the API with his email address. 2) Delete already generated token for the account. 2) An unique time based token is generated for account activation. 3) Response obtained contains success message 4) Email is triggered specifying the token based link and the next steps			Success message		
FEATURE 2: As a seller, I should be able to register a new account								
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
1	API to register a seller	POST	1) User hits an API with the mentioned manadatory fields. 2) Validation fails with appropriate response containing the validation error(s)	Email should be unique Email should be valid Phone number should be valid Password should be valid GST should be unique GST should be valid as per Govt. norms Company Name should be unique (Case insensitive)	Email Password Confirm Password GST Company Name Company Address Company Contact First Name Last Name	List of validation erros		
			1) User hits an API with the following manadatory fields. 2) Seller is successfully registered with an inactive account and a success response message should be sent. 3) An async email is triggered to the seller specifying account has been created, waiting for approval			Success message		

FEATURE 1: As a customer user, I should be able to login into the system

Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
1	API to login into the system	POST	1) User hits the API with the required fields. 2) The validation fails. 3) Response obtained should contain the validation errors.	Email should be valid Password should be valid Password should not be expired Email should exist in the database Password should exist in the database Account should be activated Account should not be locked.	username (Email) password	List of errors		
			1) User hits the API with the required fields. 2) Email entered is found to be valid and exists in the db. 3) Entered password is found to be wrong for 3 successive attempts. 4) User account gets locked. 5) Response obtained should contain the appropriate error message (To be done using Custom Spring Security Filters. This is a bonus feature)					Email is triggered to the user when account gets locked
			1) User hits the API with the required fields. 2) Email is found to be valid and exists in db. 3) Account is either locked or not activated. (Spring Security UserDetails already have the isLocked field please use that for this use case) 4) Response should contain the appropriate error message			Error message		
			1) User hits the API with the required fields. 2) User gets logged-into the system. 3) Response should contain an uniquely created access-token			Access token	User should have the authority to access only the customer specific data and the APIs	

FEATURE 2: As a seller user, I should be able to login into the system

Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
1	API to login into the system	POST	1) User hits the API with the required fields. 2) The validation fails. 3) Response obtained should contain the validation errors.	Email should be valid Password should be valid Password should not be expired Email should exist in the database Password should exist in the database Account should be activated Account should not be locked.	username (Email) password	List of errors		
			1) User hits the API with the required fields. 2) Email entered is found to be valid and exists in the db. 3) Entered password is found to be wrong for 3 successive attempts. 4) User account gets locked. 5) Response obtained should contain the appropriate error message (To be done using Custom Spring Security Filters. This is a bonus feature)					Email is triggered to the user when account gets locked
			1) User hits the API with the required fields. 2) Email is found to be valid and exists in db. 3) Account is either locked or not activated. 4) Response should contain the appropriate error message			Error message		
			1) User hits the API with the required fields. 2) User gets logged-into the system. 3) Response should contain an uniquely created access-token			Access token	User should have the authority to access only the seller specific data and the APIs	

FEATURE 3: As an admin user, I should be able to login into the system

Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
			1) User hits the API with the required fields. 2) The validation fails. 3) Response obtained should contain the validation errors.			List of errors		

			1) User hits the API with the required fields. 2) Email entered is found to be valid and exists in the db. 3) Entered password is found to be wrong for 3 successive attempts. 4) User account gets locked. 5) Response obtained should contain the appropriate error message (To be done using Custom Spring Security Filters. This is a bonus feature)					Email is triggered to the user when account gets locked
			1) User hits the API with the required fields. 2) Email is found to be valid and exists in db. 3) Account is either locked or not activated. 4) Response should contain the appropriate error message	Email should be valid Password should be valid Password should not be expired Email should exist in the database Password should exist in the database		Error message		
1	API to login into the system	POST	1) User hits the API with the required fields. 2) User gets logged-into the system. 3) Response should contain an uniquely created access-token	Account should be activated Account should not be locked.	username (Email) password	Access token	User should have the authority to access only the admin specific data and the APIs	
FEATURE 4: As a logged-in customer user, I should be able to logout of the system								
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
			1) User hits the API with the required fields. 2) The validation fails. 3) Response obtained should contain the validation errors.			List of errors		
1	API to logout of the system	POST	1) User hits the API with the required fields. 2) Access token is deleted from the system. 2) User gets logged-out of the system. 3) Response should contain the success message	Access token should be valid Access token should not be expired	Access token	Message	Only the users having customer specific roles and authorities should be able to access this API The access-token should not be able to use again to make authenticated requests	
FEATURE 5: As a logged-in seller user, I should be able to logout of the system								
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
			1) User hits the API with the required fields. 2) The validation fails. 3) Response obtained should contain the validation errors.			List of errors		
1	API to logout of the system	POST	1) User hits the API with the required fields. 2) Access token is deleted from the system. 2) User gets logged-out of the system. 3) Response should contain the success message	Access token should be valid Access token should not be expired	Access token	Message	Only the user having seller specific roles and authorities should be able to access this API The access-token should not be able to use again to make authenticated requests	
FEATURE 6: As a logged-in admin user, I should be able to logout of the system								
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
			1) User hits the API with the required fields. 2) The validation fails. 3) Response obtained should contain the validation errors.			List of errors		
1	API to logout of the system	POST	1) User hits the API with the required fields. 2) Access token is deleted from the system. 2) User gets logged-out of the system. 3) Response should contain the success message	Access token should be valid Access token should not be expired	Access token	Message	Only the user having admin specific roles and authorities should be able to access this API The access-token should not be able to use again to make authenticated requests	

FEATURE 1: As an user I should be able to reset my password using forgot my password								
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
1	API to receive a token based url	POST	1) User hits the API with his email address. 2) The validation fails. 3) Response should obtain the appropriate validation errors	1. Account should be activated 2. Email should be valid 3. Email should exist in the database	email	List of validation errors		
			1) User hits the API with his email address. 2) Already generated token should be deleted for account. 2) An unique time based token is generated for forgot password. 3) Response obtained contains success message 4) Email is triggered specifying the token based link and the next steps			Success message		
2	API to reset the password using the token	PUT	1) User hits an API with an invalid token and the required fields 2) The response should be sent with an appropriate error message. 3) Password is not updated.	Token is valid Token is not expired Password and confirm password should be valid and same	Forgot password token Password Confirm Password			
			1) User hits an API with an expired token and the required fields. 2) Delete this token 3) The response should be sent with an appropriate error message. 4) Password is not updated.			Error message		
			1) User hits an API with a valid token and the required fields. 2) Password is successfully updated. 3) The token should be deleted from the system. 4) The response sent should contain a success message			Success message		An async mail is triggered informing the user about password update Instead of PUT, if PATCH is used

Common Validation for all the below mentioned APIs		1. User should have Admin rights, respond with appropriate status code and response message otherwise and restrict from accessing							
FEATURE 1: As an admin, I should be able to view all users									
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Optional Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feature/Points
	1 API to list all the registered customers	GET	1) User hits the api to obtain all registered customers			PageSize (default 10) PageOffset (default 0) Sort (default id) Email (For filtering customer)	List of customers: Id Full Name Email Is_Active		
	2 API to list all the registered sellers	GET	1) User hits the api to obtain all registered seller			PageSize (default 10) PageOffset (default 0) Sort (default id) Email (For filtering seller)	List of sellers: Id Full Name Email Is_Active Company Name Company Address Company Contact		
FEATURE 2: As an admin, I should be able to activate/de-activate a user									
Task	Description	Type	Flow	Validations	Mandatory Request Parameters		Mandatory Response Parameters	Acceptance Criteria	Bonus Feature/Points
	1 API to activate a customer	PUT	1) User hits the api with a customer user ID 2) User is not found 3) Error response is obtained 1) User hits the api with a customer user ID for a de-activated user 2) Customer user is activated. 3) Success message response is sent 4) Mail is triggered to customer specifying account activation	Customer user ID should be valid User should be in deactivated state	Customer User ID		Error message Success message		Endpoint is created as a PATCH rather than PUT
	2 API to de-activate a customer	PUT	1) User hits the api with a customer user ID 2) User is not found 3) Error response is obtained 1) User hits the api with a customer user ID for an existing activated user 2) Customer user is de-activated. 3) Success message response is sent 4) Mail is triggered to customer specifying account deactivation	Customer user ID should be valid User should be in activated state	Customer User ID		Error message Success message		Endpoint is created as a PATCH rather than PUT
	3 API to activate a seller	PUT	1) User hits the api with a seller user ID 2) User is not found 3) Error response is obtained 1) User hits the api with a seller user ID for a de-activated user 2) Seller user is activated. 3) Success message response is sent 4) Mail is triggered to seller specifying account activation	Seller user ID should be valid User should be in deactivated state	Seller User ID		Error message Success message		Endpoint is created as a PATCH rather than PUT

Common Validation for all the below mentioned APIs		1. User should have Admin rights, respond with appropriate status code and response message otherwise and restrict from accessing							
			1) User hits the api with a seller user ID 2) User is not found 3) Error response is obtained				Error message		
			1) User hits the api with a seller user ID for an existing activated user 2) Seller user is de-activated. 3) Success message response is sent 4) Mail is triggered to seller specifying account deactivation						
			1) User hits the api with a seller user ID for an already deactivated user 2) No action is performed. 3) Success message response is sent with appropriate message 4) No Mail is triggered to seller						
4	API to de-activate a seller	PUT		Seller user ID should be valid User should be in activated state	Seller User ID		Success message		Endpoint is created as a PATCH rather than PUT

Common Validation and flow for all the below mentioned APIs			1. User should be logged in, have Seller rights and has an activated account, else respond with appropriate status code and error response message and restrict from accessing the APIs					
FEATURE 1: As a logged in seller, I should be able to edit my profile								
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
1	API to view my profile	GET	1) User hits the API 2) Response obtains shows the user data		access token	Id First Name Last Name Is_Active Company Contact Company Name Image GST Address: City State Country Address Line Zip Code		
2	API to update my profile	PUT/PATCH	1) User hits the API 2) Validation fails for fields being updated. 3) Response obtains shows list of validation errors					
			1) User hits the API with updated fields 2) Profile gets updated successfully 3) Response obtained shows success message		access token Fields which are updated	success message		
3	API to update my password	PUT/PATCH	1) User hits the API 2) Validation fails for fields being updated. 3) Response obtains shows list of validation errors	Password should be valid Confirm password should be valid		list of validation errors		
			1) User hits the API with the required fields 2) Password is updated. 3) Response obtained shows the success message	Password and confirm password should match	access token password confirm password	success message		Send an_email upon password change
4	API to update an address	PUT/PATCH	1) User hits the API with the required fields 2) No address is found for the address Id 3) Response shows address not found			error message		
			1) User hits the API with the required fields 2) Validation fails. 3) Response obtained shows the list of validation errors			list of validation errors		
			1) User hits the API with the required fields 2) Address is updated. 3) Response obtained contains a success message	A valid and existing address id is to be passed	access token address Id fields being updated			
					success message			

Common Validation and flow for all the below mentioned APIs			1. User should be logged in, have Customer rights and has an activated account, else respond with appropriate status code and error response message and restrict from accessing the APIs					
FEATURE 1: As a logged in customer, I should be able to edit my profile								
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
1	API to view my profile	GET	1) User hits the API 2) Response obtains shows the user data		access token	Id First Name Last Name Is_Active Contact Image		
2	API to view my addresses	GET	1) User hits the API 2) Response obtains show the list of addresses		access token	List of addresses: City State Country Address Label Zip Code		
3	API to update my profile	PUT/PATCH	1) User hits the API 2) Validation fails for fields being updated. 3) Response obtains shows list of validation errors		access token Fields which are updated	list of validation errors		Instead of sending all the fields to update, if only those fields are sent in request which are actually being updated
			1) User hits the API with updated fields 2) Profile gets updated successfully 3) Response obtained shows success message			success message		
4	API to update my password	PUT/PATCH	1) User hits the API 2) Validation fails for fields being updated. 3) Response obtains shows list of validation errors	Password should be valid Confirm password should be valid Password and confirm password should match	access token password confirm password	list of validation errors		Send an email upon password change
			1) User hits the API with the required fields 2) Password is updated. 3) Response obtained shows the success message			success message		
5	API to add a new address	POST	1) User hits the API with the required fields 2) Validation fails. 3) Response obtained shows the list of validation errors		access token Address City Status Country Zip Code Label	list of validation errors		
			1) User hits the API with the required fields 2) New address is added for the user. 3) Response obtained contains a success message			success message		
6	API to delete an address	DELETE	1) User hits the API with the required fields 2) No address is found for the address Id 3) Response shows address not found	A valid and existing address id is to be passed	access token address Id	error message		
			1) User hits the API with the required fields 2) Address is deleted for corresponding id 3) Response shows success message			success message		
7	API to update an address	PUT/PATCH	1) User hits the API with the required fields 2) No address is found for the address Id 3) Response shows address not found	A valid and existing address id is to be passed	access token address Id fields being updated	error message		
			1) User hits the API with the required fields 2) Validation fails. 3) Response obtained shows the list of validation errors			list of validation errors		
			1) User hits the API with the required fields 2) Address is updated. 3) Response obtained contains a success message			success message		

Common Validation and flow for all the below mentioned APIs		1. User should be logged in, has an activated account and valid access token passed in all requests, else respond with appropriate status code and error response message and restrict from accessing the APIs							
FEATURE 1: As a logged in admin user, I should be able to create and update product categories along with their metadata properties, to be further used while creating products.									
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Optional Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
1	API add a Metadata field	POST	1) User hits API with mentioned manadatory and optional field (s) 2) Validation fails with appropriate response containing the validation error(s) 1) User hits API with mentioned manadatory field(s) 2) Category Metadata field is saved sucefully	1. Field name should be unique	Field Name		List of validation errors Success message with created field ID		
2	API to view all Metadata fields	GET	1) User hits API with mentioned optional field(s) 2) Response returns with all metadata fields			max offset sort order query	List of all fields		
3	API add a category	POST	1) User hits API with mentioned manadatory and optional field (s) 2) Validation fails with appropriate response containing the validation error(s)	1. Category name should be unique at root level and along breadth/depth in a tree 2. Parent Category ID if passed, should not be associated with any existing (non-deleted) product	Category Name	Parent Id	List of validation errors		
			1) User hits API with mentioned manadatory field(s) and optional Parent Category ID 2) Category is saved sucefully				Success message with created category ID		
4	API to view a category	GET	1) User hits API with mentioned mandatory field(s) 2) Validation fails for the request.	1) ID should be valid	Category ID		List of validation errors Category details with parent categories upto root level and immediate children categories, and associated fields along with possible values		
			1) User hits API with mentioned mandatory field(s) 2) Response returns with category details						
5	API to view all categories	GET	1) User hits API with mentioned optional field(s) 2) Response returns with all category details			max offset sort order query	List of all categories, with each individual category's detail as stated above		
7	API to update a category	PUT	1) User hits API with mentioned mandatory field(s) 2) Validation fails for the request.	1. ID should be valid 2. Category name should be unique at root level and along breadth/depth in a tree	Category ID Category Name		List of validation errors		
			1) User hits API with mentioned mandatory field(s) 2) Response returns with success message			Success message			
8	API to add new category metadata field for a category	POST	1) User hits API with mentioned mandatory field(s) 2) Validation fails for the request.	1) Category Id should be valid 2) Metadata Field Id should be valid 3) There should be atleast one value for each field 4) Values should be unique within the list	Category ID Metadata Field Id with possible Values		List of validation errors		Multiple objects of [Field Id and Values] should be acceptable in the request for a Category Id
			1) User hits API with mentioned mandatory field(s) 2) Response returns with success message			Success message			
9	API to update values for an existing metadata field in a category	PUT	1) User hits API with mentioned mandatory field(s) 2) Response returns with success message	1) Category Id should be valid 2) Metadata Field Id should be valid and already associated with Category 3) Values should be unique for category - metadata field combination	Category ID Metadata Field ID with new values to be added		List of validation errors	Previously saved values for a category Id and metadata field Id are not deleted or updated	Multiple objects of [Field Id and Values] should be acceptable in the same request for a Category Id
			1) User hits API with mentioned mandatory field(s) 2) Response returns with success message			Success message			
FEATURE 2: As a logged in seller user, I should be able to list all catagories and its metadata-fields									
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Optional Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
1	API to list all categories	GET	1) User hits API with mentioned manadatory and optional field (s)				List of all leaf node categories along with their metadata fields, field's possible values and parent node chain details		
FEATURE 3: As a logged in customer user, I should be able to list all catagories and its metadata-fields									
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Optional Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
1	API to list all categories	GET	1) User hits API with mentioned manadatory and optional field (s) 2) Return List all root level catagories if no ID is passed, else list of all immediate child nodes of passed category ID	1) Category Id should be valid if present		Category Id	List of valid categories at same level and with same parent		

Common Validation and flow for all the below mentioned APIs			1. User should be logged in, has an activated account and valid access token passed in all requests, else respond with appropriate status code and error response message and restrict from accessing the APIs						
FEATURE 1: As a logged in admin user, I should be able to create and update product categories along with their metadata properties, to be further used while creating products.									
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Optional Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
							1. All metadata field along with possible values for that category 2. Brands list compiled from associated products of that node (or its child nodes) 3. Minimum and Maximum price possible from lowest and highest valued product under that category		
	2 API to fetch filtering details for a category	GET	1) User hits API with mentioned manadatory and optional field (s) 2) Return List all filtering data	1) Category Id should be valid	Category Id				

Common Validation and flow for all the below mentioned APIs		1. User should be logged in, have appropriate rights as per the feature and has an activated account, else respond with appropriate status code and error response message and restrict from accessing the APIs							
FEATURE 1: As a logged in seller, I should be able to create and update the products I want to sell									
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Optional Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Feture/Points
1	API add a product	POST	1) User hits API with mentioned manadatory and optional field(s) 2) Validation fails with appropriate response containing the validation error(s)	1. Product name should be unique with respect to the (brand, category and seller) combination 2. Category ID passed should be of valid leaf node category	Name Brand Category Id	Description IS_CANCELLABLE (false) IS_RETURNABLE (false)	List of validation errors	Product created is inactive by default	
			1) User hits API with mentioned mandatory field(s) and optional parameters 2) Product is saved sucefully. 3) Product should inactive by default 4) Send an email to the admin for activating the product and informing about new product details.				Success message		
2	API add a product variation	POST	1) User hits API with mentioned manadatory and optional field(s) 2) Validation fails with appropriate response containing the validation error(s)	1. Product ID should be valid 2. Quantity should be 0 or more 3. Price should be 0 or more 4. Image should be in a valid image format 5. Product should be non-deleted and active 6. Each metadata field value sent should be from within the possible field values defined for that field in the category 7. Variation should have atleast one "metadata field - value" added, and all variations of a product should have same metadata structure. 8. Ideally each variation of a product should be unique, but there's no hard requirement for this. It's a good to have constraint.	Product Id Metadata (field-value map) Primary Image Quantity Available Price	List of secondary images	List of validation errors		
			1) User hits API with mentioned manadatory field(s) and optional parameters 2) Product variation is saved sucefully. 3) By default variation will be active				Success message		
3	API to view a product	GET	1) User hits API with mentioned mandatory field (s) 2) Validation fails for the request.	1) ID should be valid 2) Logged in user should be creator of the product 3) Product should be non-deleted	Product ID		List of validation errors		
			1) User hits API with mentioned mandatory field (s) 2) Response returns with product details				Product details along with selected category details		
4	API to view a product variation	GET	1) User hits API with mentioned mandatory field (s) 2) Validation fails for the request.	1) ID should be valid 2) Logged in user should be creator of the product variation 3) Product should be non-deleted	Product Variation Id		List of validation errors		
			1) User hits API with mentioned mandatory field (s) 2) Response returns with product details				Product variation details with parent product details.		
5	API to view all products	GET	1) User hits API with mentioned optional field(s) 2) Response returns with all non-deleted products details			max offset sort order query	List of all products created by the logged in user, along with each product's category details		
6	API to view all product variations for a product	GET	1) User hits API with mentioned optional field(s) 2) Response returns with all variations for the product ID passed	1) ID should be valid 2) Logged in user should be creator of the product 3) Product should be non-deleted	Product Id	max offset sort order query			
7	API to delete a product	DELETE	1) User hits API with mentioned mandatory field (s) 2) Validation fails for the request.	1. ID should be valid product ID 2) Logged in user should be creator of the product	Product ID		List of validation errors		
			1) User hits API with mentioned mandatory field (s) 2) Response returns with success message				Success message		
8	API to update a product	PUT	1) User hits API with mentioned mandatory field (s) and optional field that needs to be updated 2) Validation fails for the request.	1. ID should be valid 2. If updated name is passed, it should be unique with respect to the (brand, category and seller) combination 3. Logged in user should be creator of the product 1. ID should be valid 2. Logged in user should be creator of the product variation 3. Product should be non-deleted and active 4. Each metadata field value sent should be from within the possible field values defined for that field in the category	Product ID	Name Description IS_CANCELLABLE IS_RETURNABLE	List of validation errors		
			1) User hits API with mentioned mandatory field (s) and optional field that needs to be updated 2) Response returns with success message				Success message		
9	API to update a product variation	PUT	1) User hits API with mentioned mandatory field (s) and optional field that needs to be updated 2) Validation fails for the request.		Product Variation ID	Quantity Available Price Metadata Primary Image Secondary Images Is Active	List of validation errors		
			1) User hits API with mentioned mandatory field (s) and optional field that needs to be updated 2) Response returns with success message				Success message		
FEATURE 2: As a logged in customer, I should be able to list and view the products I want to purchase									

Common Validation and flow for all the below mentioned APIs				1. User should be logged in, have appropriate rights as per the feature and has an activated account, else respond with appropriate status code and error response message and restrict from accessing the APIs					
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Optional Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Future/Points
1	API to view a product	GET	1) User hits API with mentioned mandatory field (s) 2) Validation fails for the request. 1) User hits API with mentioned mandatory field (s) 2) Response returns with product details 3)	1) ID should be valid 2) Product should be non-deleted and active 3) Product should have atleast one valid variations	Product ID		List of validation errors Product details along with product's selected category details and all variations details including images		
2	API to view all products	GET	1) User hits API with mentioned mandatory and optional field(s) 2) Response returns with all non-deleted active products details associated to the category passed (if leaf node category) else products associated with all the child categories.	1) Category Id should valid 2) Category Id should be of valid leaf node 3) Product should have atleast one valid variations	Category Id	max offset sort order query	List of all products, along with each product's category details, all variations primary images.		
3	API to view Similar products	GET	1) User hits API with mentioned mandatory and optional field(s) 2) Response returns with all non-deleted active products details matching currently open product (The product ID passed)	1) Product Id should valid	Product ID	max offset sort order query	List of all similar products, along with each product's category details, all variations primary images.	The products fetched should be similar to current product (The passed product ID). Similarity can be based on category or variation or combinations of multiple parameters, whatever way you want to implement	
FEATURE 3: As a logged in admin user, I should be able to list and view the products created in the system									
Task	Description	Type	Flow	Validations	Mandatory Request Parameters	Optional Request Parameters	Mandatory Response Parameters	Acceptance Criteria	Bonus Future/Points
1	API to view a product	GET	1) User hits API with mentioned mandatory field (s) 2) Validation fails for the request. 1) User hits API with mentioned mandatory field (s) 2) Response returns with product details	1) ID should be valid	Product ID		List of validation errors Product details along with product's selected category details, all variations primary images		
2	API to view all products	GET	1) User hits API with mentioned optional field(s) 2) Response returns with all non-deleted active products details			max offset sort order query (like category ID, seller ID etc)	List of all products, along with each product's category details, all variations primary images.		
4	API to deactivate a product	PUT	1) User hits API with mentioned mandatory field (s) 2) Validation fails for the request. 1) User hits API with mentioned mandatory field 2) product is deactivated 3) An email is being sent to the product owner informing deactivation and other product details	1) ID should be valid 2) Product should be active	Product ID		List of validation errors success message		
5	API to activate a product	PUT	1) User hits API with mentioned mandatory field (s) 2) Validation fails for the request. 1) User hits API with mentioned mandatory field 2) Product is activated 3) Email is triggered to the seller informing product is activated	1) ID should be valid 2) Product should be inactive	Product ID		List of validation errors success message		