Exploratory Testing by Mudra:  
  
ChatGPT Input:  
  
I have attached our Project document: Business Case, Business Requiermnt, RCT, Road Map ==> Features for Sprint 1   
03. 01 Category Management – Add, Update, Delete.   
03. 02 Product Management – Add, Update, Delete.   
03. 03 Inventory Dashboard   
03. 04 Manage Product Stock Levels   
04.01 Log In as User   
04.02 Log In as Admin   
04.03 Easy SignUp   
04.04 Delete Account   
  
==> I have already explained the context of my system and the feature under test. Now, can you suggest test ideas for testing Features for Sprint 1?

Here are some test ideas for the features in Sprint 1, focused on **Category Management, Product Management, Inventory Dashboard, Stock Levels, User/Admin Logins, Signup, and Account Deletion**:

**1. Category Management (Add, Update, Delete)**

* **Add Category**:
  + Verify that a new category can be added with valid inputs (e.g., name, description).
  + Test that adding a category without required fields displays appropriate error messages.
  + Validate that duplicate category names are not allowed.
  + Ensure that adding a category updates the system’s category list in real time.
* **Update Category**:
  + Verify that an existing category can be edited and saved.
  + Test that changes to a category’s name and description are reflected immediately.
  + Check for validation messages when invalid data is entered during the update.
* **Delete Category**:
  + Test that a category can be deleted and is removed from the system’s list.
  + Verify that deleting a category associated with existing products displays a warning or restriction.
  + Check that the system does not allow deletion of mandatory categories, if applicable.

**2. Product Management (Add, Update, Delete)**

* **Add Product**:
  + Ensure a new product can be added with complete details (e.g., name, price, category).
  + Test that the system validates required fields and displays errors for missing inputs.
  + Validate that the product is visible in the inventory after being added.
  + Verify that adding a product with duplicate names displays an appropriate error message.
* **Update Product**:
  + Confirm that product details can be modified and saved successfully.
  + Test that all updated details reflect accurately in the inventory.
  + Validate that restricted fields (e.g., product ID) cannot be edited if such constraints exist.
* **Delete Product**:
  + Verify that a product can be deleted and is removed from the inventory list.
  + Test that an error or confirmation message is displayed before deletion.
  + Ensure that deleting a product associated with active orders triggers an appropriate warning.

**3. Inventory Dashboard**

* Verify that the dashboard displays real-time inventory data accurately.
* Test that filters and sorting options (e.g., by category, stock level) work as expected.
* Check that updates in product or category data are immediately reflected on the dashboard.
* Test the dashboard’s responsiveness across devices and screen sizes.
* Validate that only authorized users can access or view the dashboard.

**4. Manage Product Stock Levels**

* Ensure that stock levels can be adjusted manually and saved.
* Test that updating stock levels affects related functionalities (e.g., low-stock notifications).
* Verify that the system does not allow stock levels to drop below zero if restricted.
* Check that stock updates reflect across related areas, like the inventory dashboard and product detail pages.

**5. Log In as User**

* Verify that valid credentials allow users to log in successfully.
* Test that incorrect credentials display an error message.
* Ensure that login attempts are limited to prevent brute-force attacks.
* Check that logged-in users are directed to the appropriate home or dashboard page.
* Test that users can log out and are redirected to the login page.

**6. Log In as Admin**

* Ensure that only admin credentials grant access to the admin panel.
* Test that invalid credentials show an appropriate error.
* Verify that different user levels (e.g., customer vs. admin) have restricted access to admin features.
* Confirm that session timeouts or inactivity logs out admins for security.
* Test admin login on different devices and browsers.

**7. Easy Signup**

* Verify that users can sign up with valid details and are notified upon successful registration.
* Test that required fields (e.g., email, password) are validated.
* Ensure that passwords meet strength requirements.
* Validate that existing emails cannot be reused for new accounts.
* Confirm that a welcome email or notification is sent post-signup, if implemented.

**8. Delete Account**

* Verify that users can delete their accounts through the settings page.
* Test that a confirmation step or warning is shown before deletion.
* Ensure that all user data is removed or anonymized upon deletion, depending on data policies.
* Check that deleted accounts cannot be recovered if permanent deletion is implemented.
* Confirm that users receive an account deletion confirmation email or message.

These test ideas should help in verifying the functionality, security, and usability of the features in Sprint 1. Let me know if you'd like more specific test cases or if there's anything else you'd like to focus on!