

TAILOR - *i*



VIRTUAL TAILOR SHOP

Tailori Plugin Integration In Shopify Store

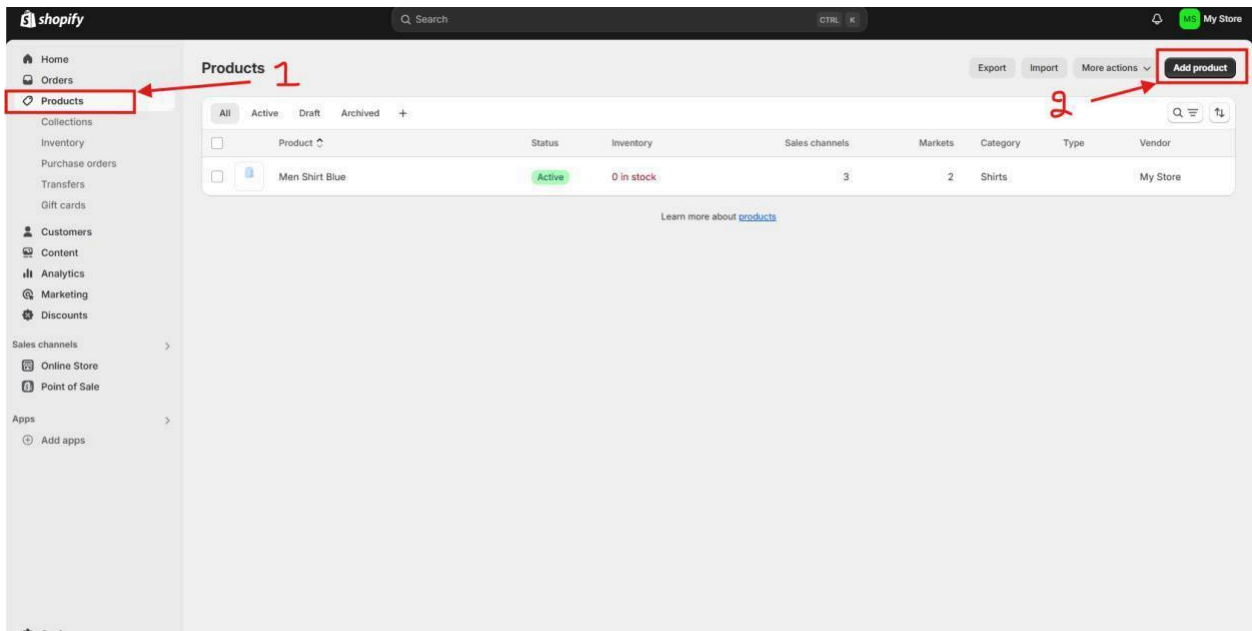
www.textronic.com

Index

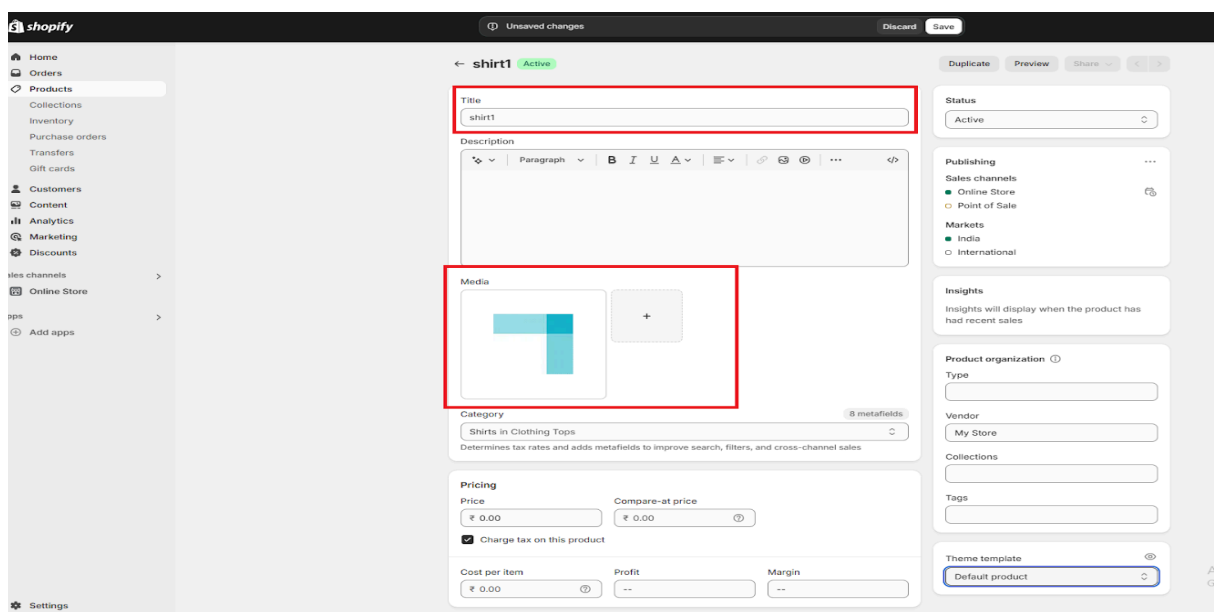
- 1. Product Upload**
- 2.Create template for product page**
- 3.Create template for Collection / Category page**
- 4.Create Meta Field for Product**
- 5.Create Collection**
- 6.Assign Product to collection**
- 7.Add meta Field to Product**
- 8.Essential Files and Plugins for Shopify Integration Files upload**
- 9.Creating API Token for Products (need read access only)**
- 10. Add Customization Page**
- 11.Add Tailori Customize Button**

I) Add Product

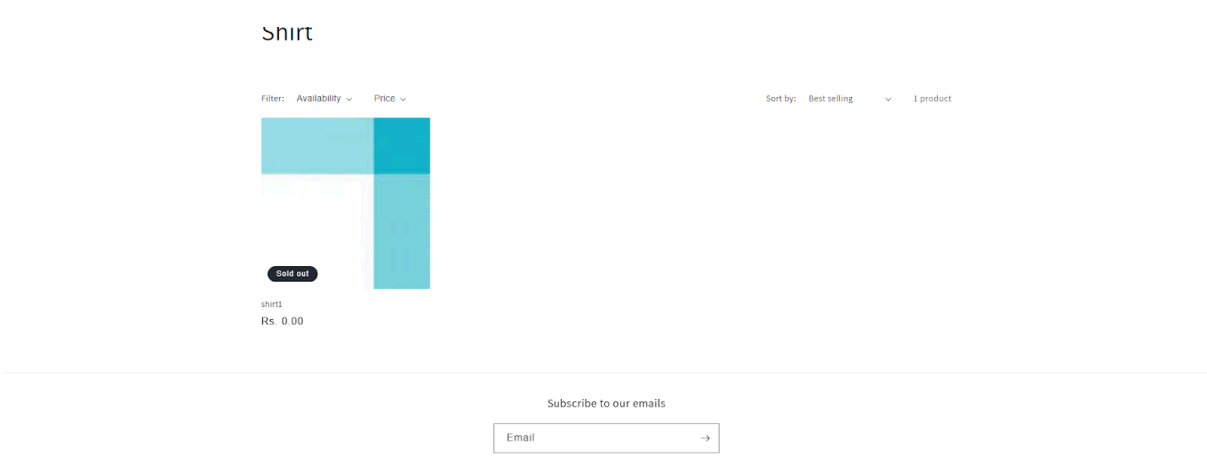
Step 1: Add products



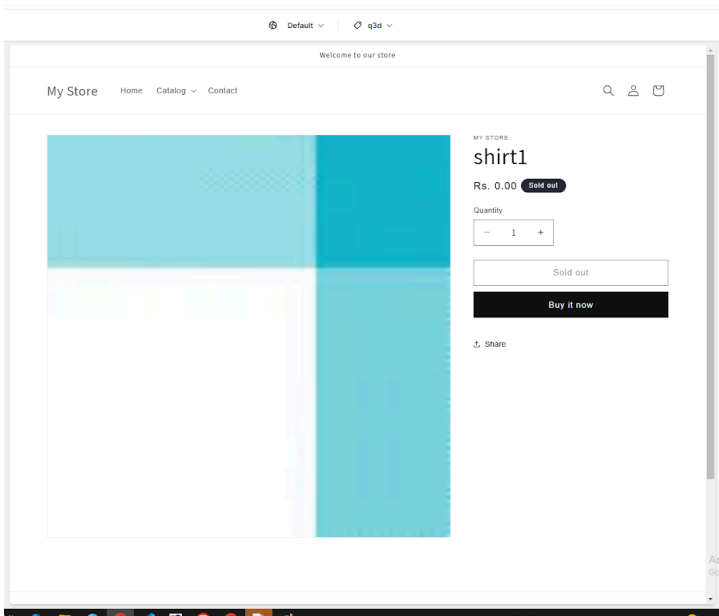
Step 2: Entering Title name (eg.Shirt,suit)



Collection / Category Page Interface



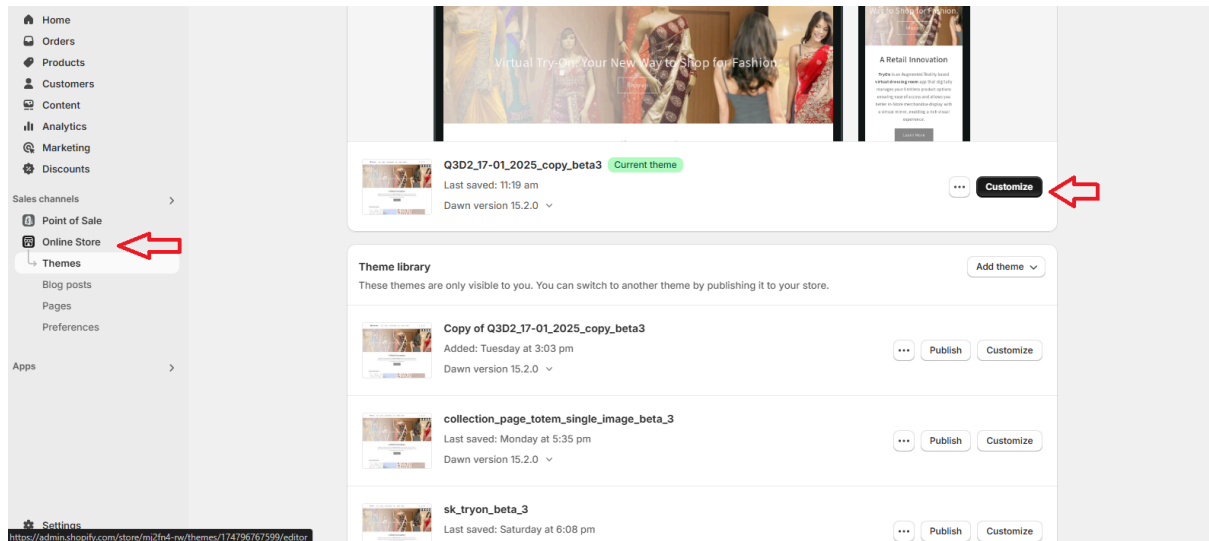
Product Page Interface



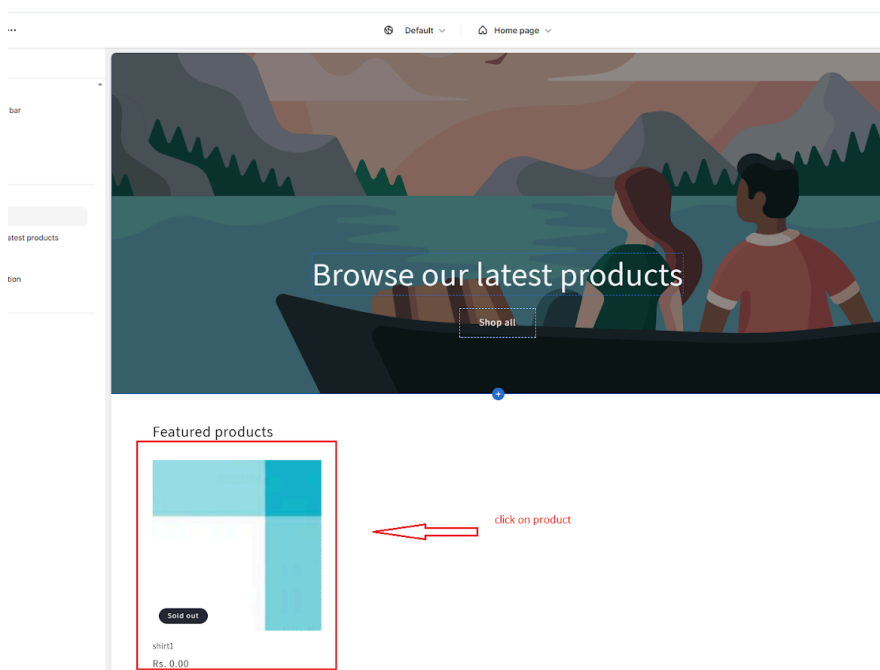
II) Create template for product page

Step 1: Go to your Shopify admin site.

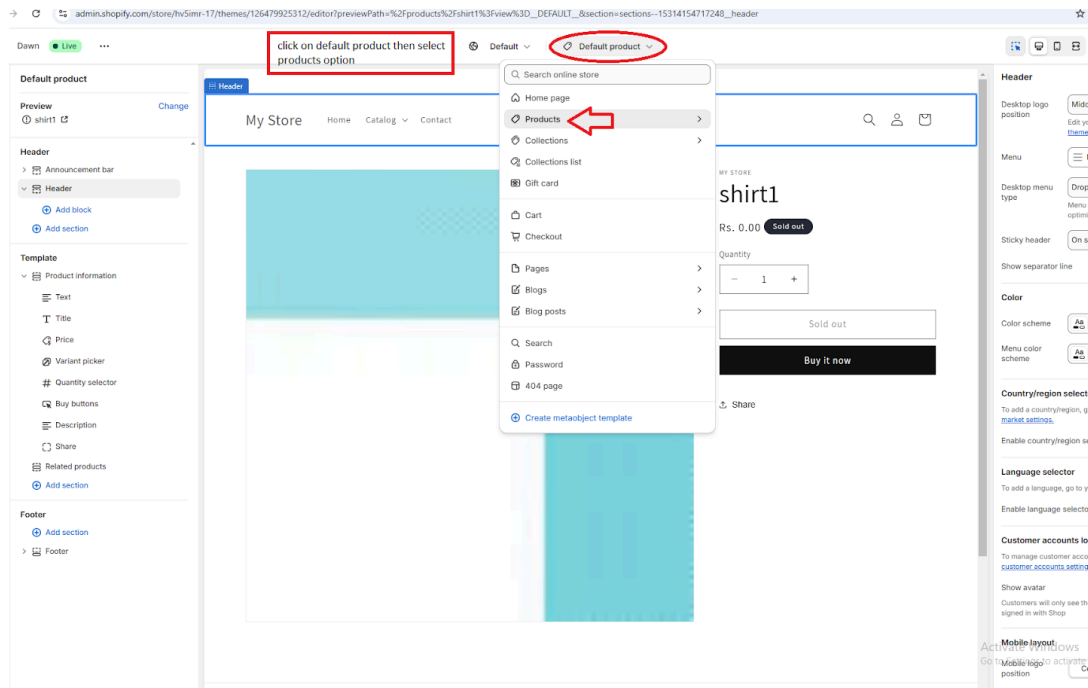
Step 2: Click on Online Store then click on “Customize”.



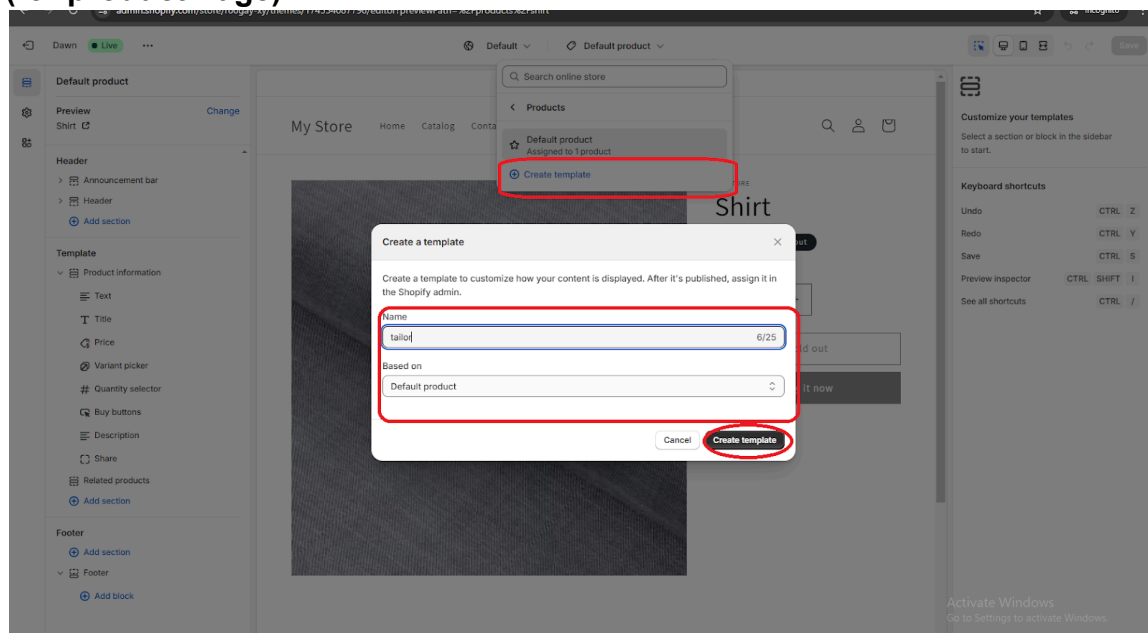
Step 3: go to the product page & click on Product



Step 4: click on default product then select products



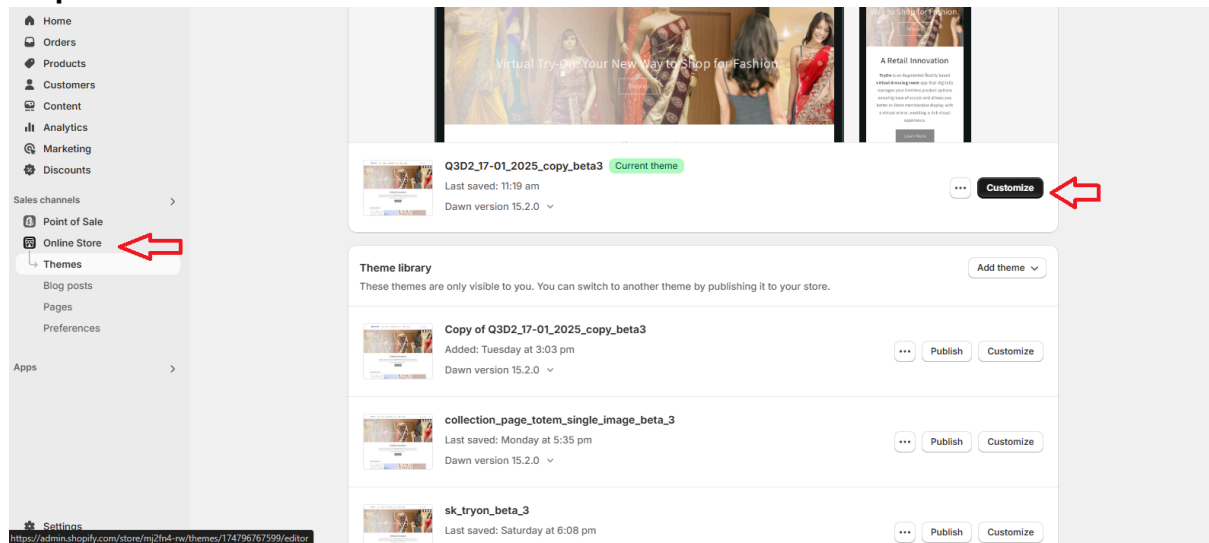
Step 5: click on the Create Template option then Add template name as “tailor” (for product Page)



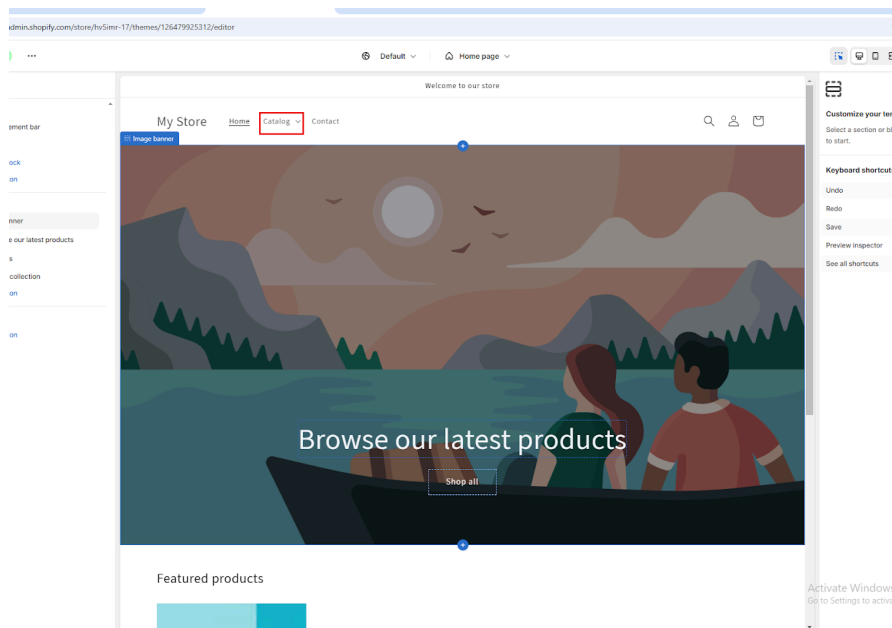
III) Create template for Category / Collection page.

Step 1: Go to your Shopify admin site.

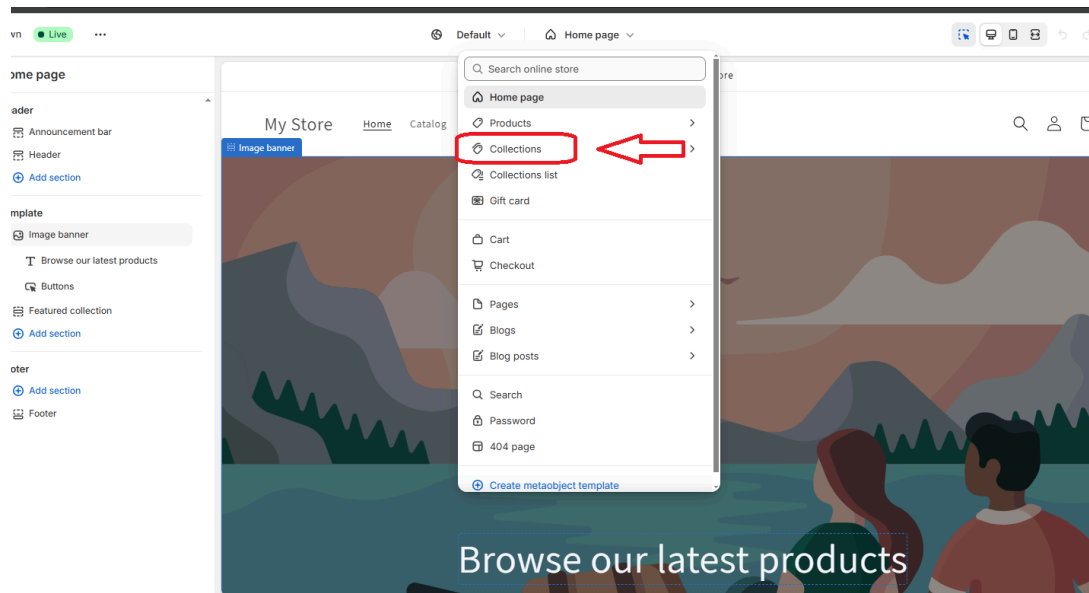
Step 2: Click on Online Store then click on “Customize”.



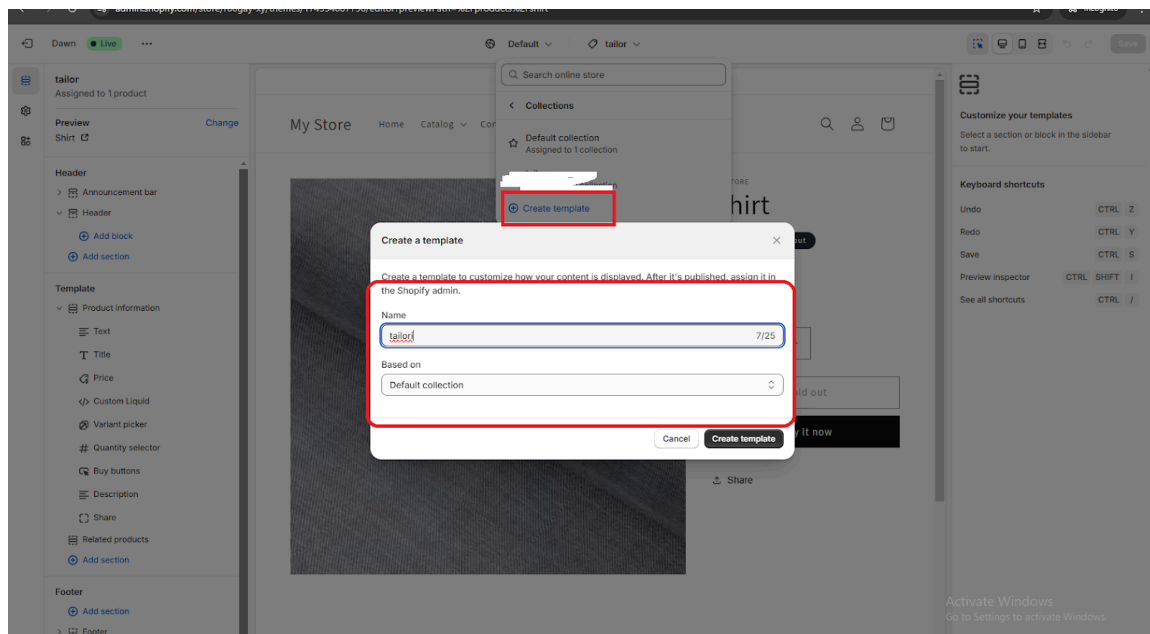
Step 3: click on catalog



Step 4: click on default product then select collection

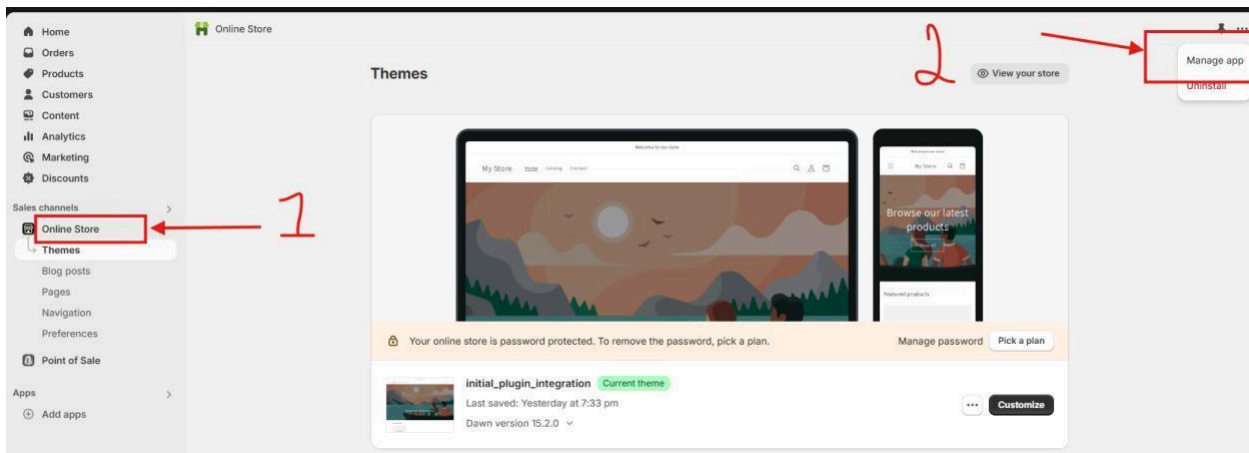


Step 5: click on the Create Template option then Add template name as “tailori” (for CategoryPage)

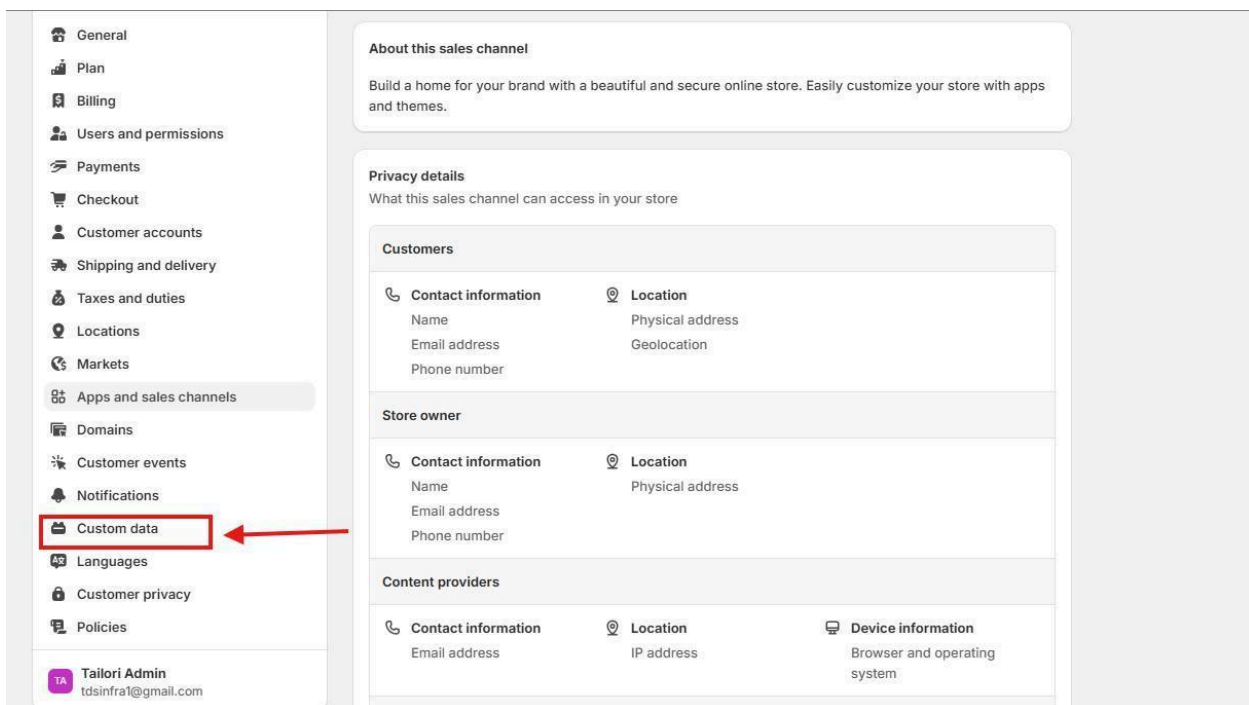


IV) Set Meta fields for products

Step 1 : click on Online store and select manage App



Step 2 : Click on Custom Data



Step 3: Click on Products

My Store
mj2fn4-rw.myshopify.com

- General
- Plan
- Billing
- Users and permissions
- Payments
- Checkout
- Customer accounts
- Shipping and delivery
- Taxes and duties
- Locations
- Markets
- Apps and sales channels
- Domains
- Customer events
- Notifications
- Custom data**
- Languages
- Customer privacy
- Policies

Custom data

Metafield definitions ⓘ
Add a custom piece of data to a specific part of your store.

🔗 Products	4 >
🔗 Variants	0 >
🔗 Collections	0 >
👤 Customers	0 >
📦 Orders	0 >
📄 Draft orders	0 >
📍 Locations	0 >
📄 Pages	0 >
📄 Blogs	0 >
📄 Blog posts	0 >
🌐 Markets	0 >

Define your first metaobject ⓘ
Metaobjects allow you to group fields and connect them to different parts of your store. Use them to create custom content or data structures.

Step 4. Enter Name and select type & Save.

My Store
pno0kd-tm.myshopify.com

Add product metafield definition ⓘ Definition pinned

Name
product

Namespace and key
custom.product
The system uses this to identify the definition.

Description
0/100

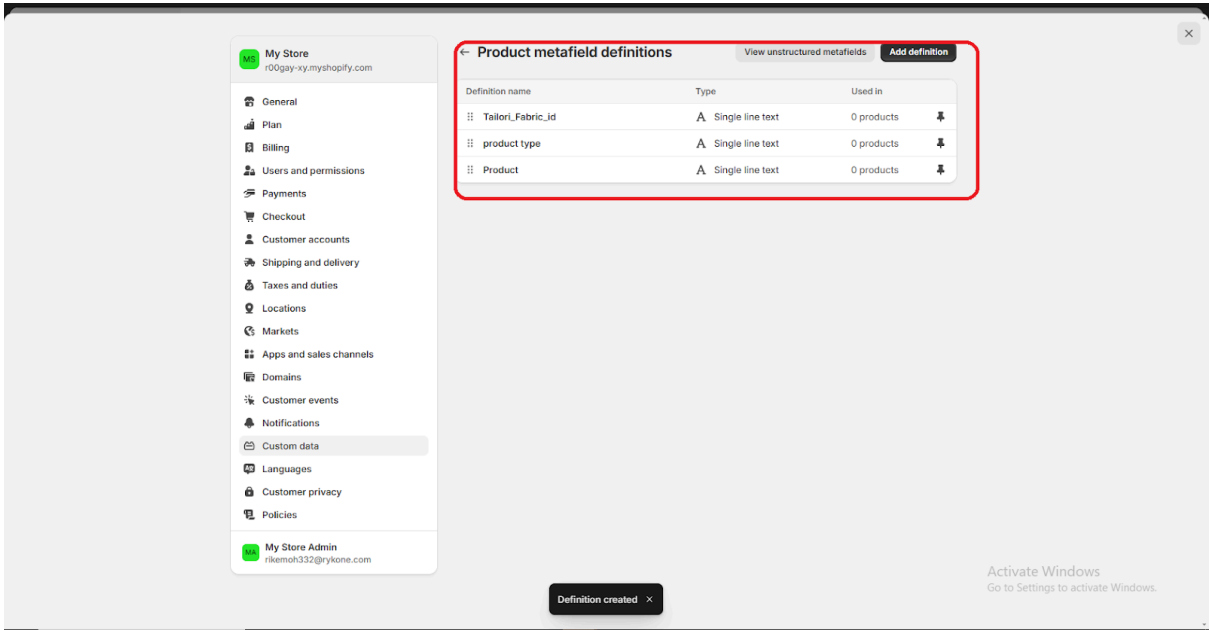
Select type

Most Used

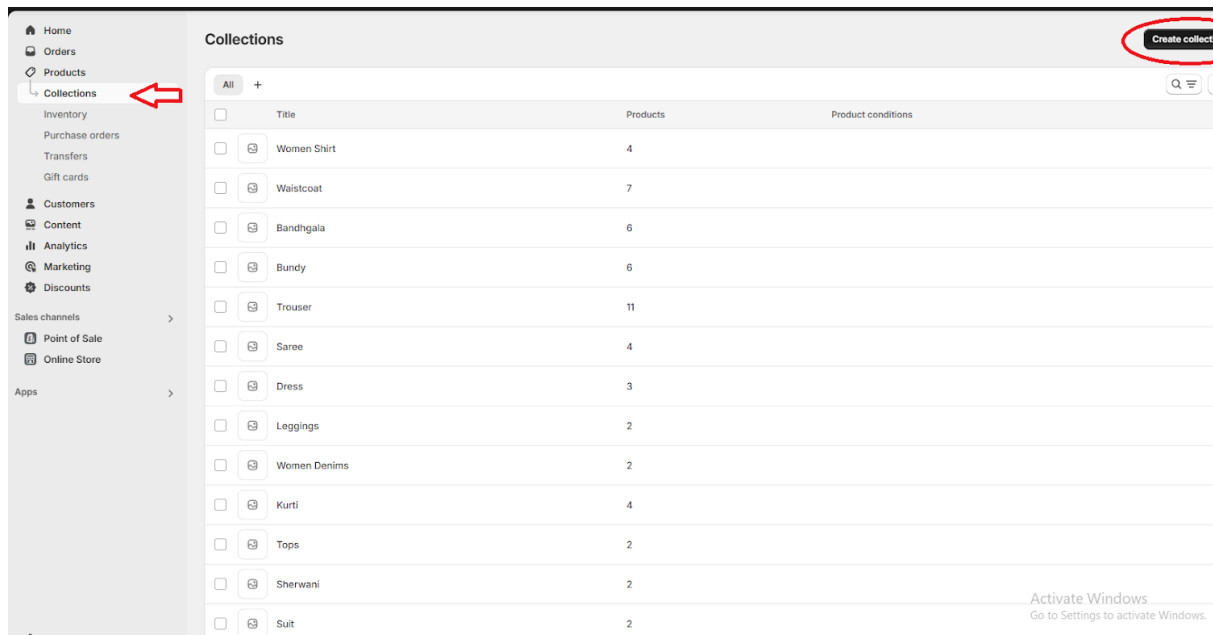
- Single line text**
- Multi-line text
- Integer
- Decimal
- Product
- File
- Date and time
- Date and time
- Date
- Measurement

Activate Windows
Go to Settings to activate Windows.

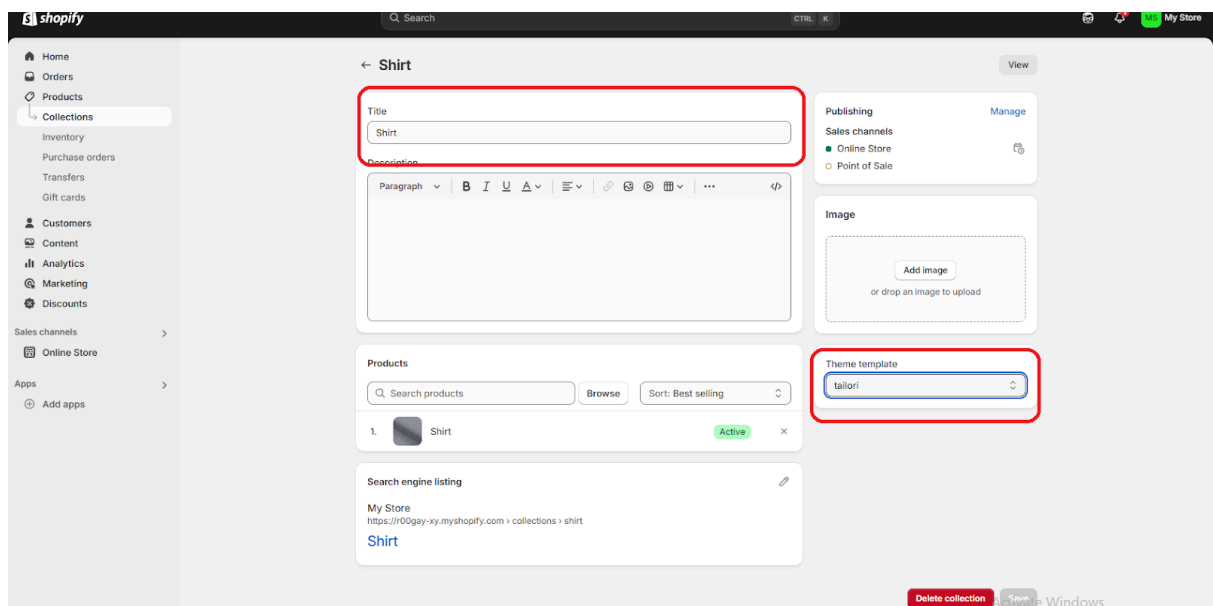
Step 5. Click on Add definition & this Three definitions are Mandatory



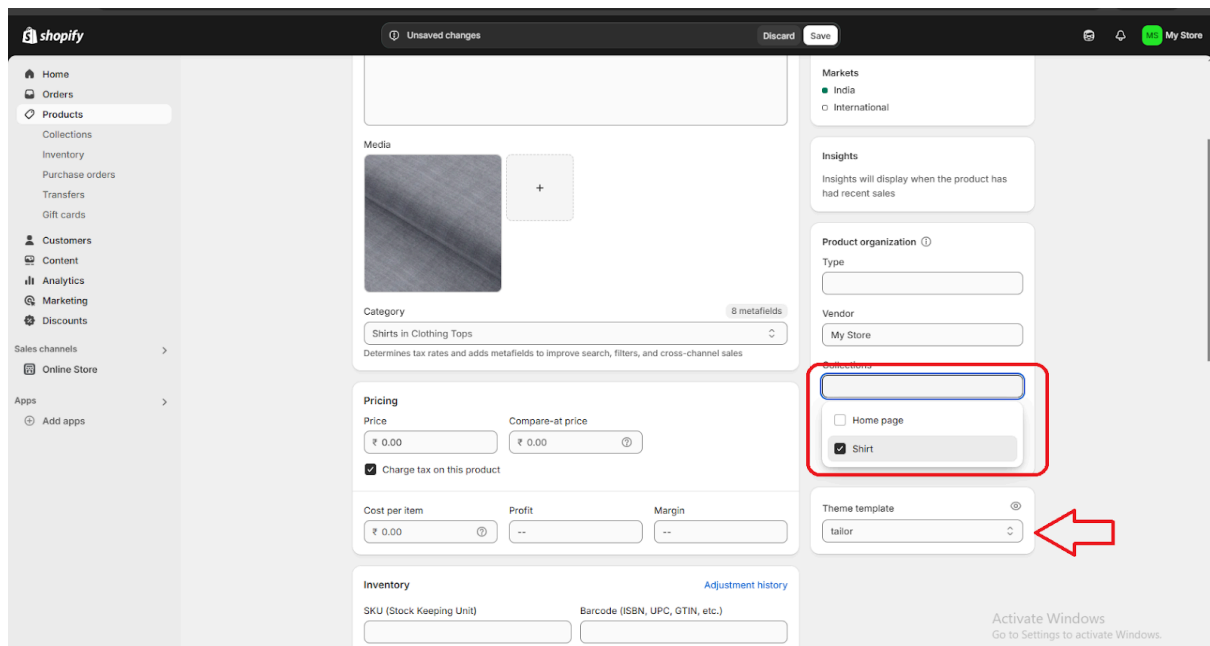
V) Add Collection



Step 1.Entering Title name (eg.shirt ,suit) then select tailori collection & Save it.



VI) Assign Product To Collection & Theme Template assign to tailor



VII) Add Metafield to Product

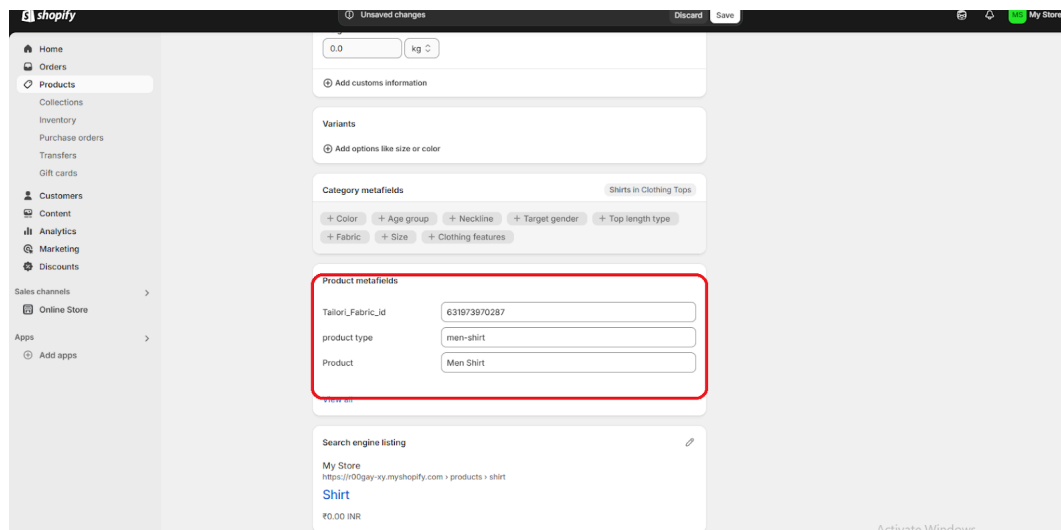
Step 1: Enter meta Details

Note: -

Tailor_Fabric_id: 1D7114533 Etc (the id of the fabric which has been uploaded must be entered) **Refere Demo Fabric Folder Tailor_fabric_id : (image name)**

Product type: men-shirt , men-jacket Etc (first letter should be small)

Product :Men Shirt, Men Suit Etc (first letter should be capital)

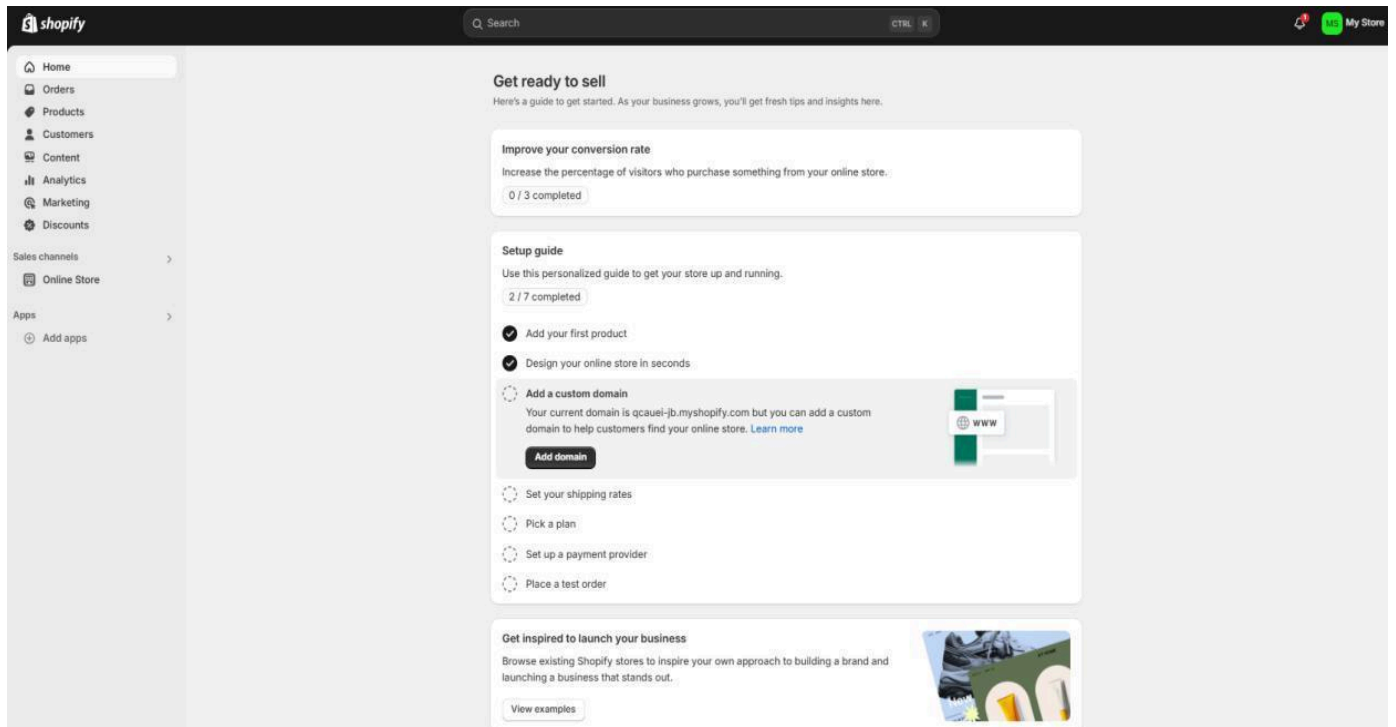


VIII) Essential Files and Plugins for Shopify Integration Files Upload

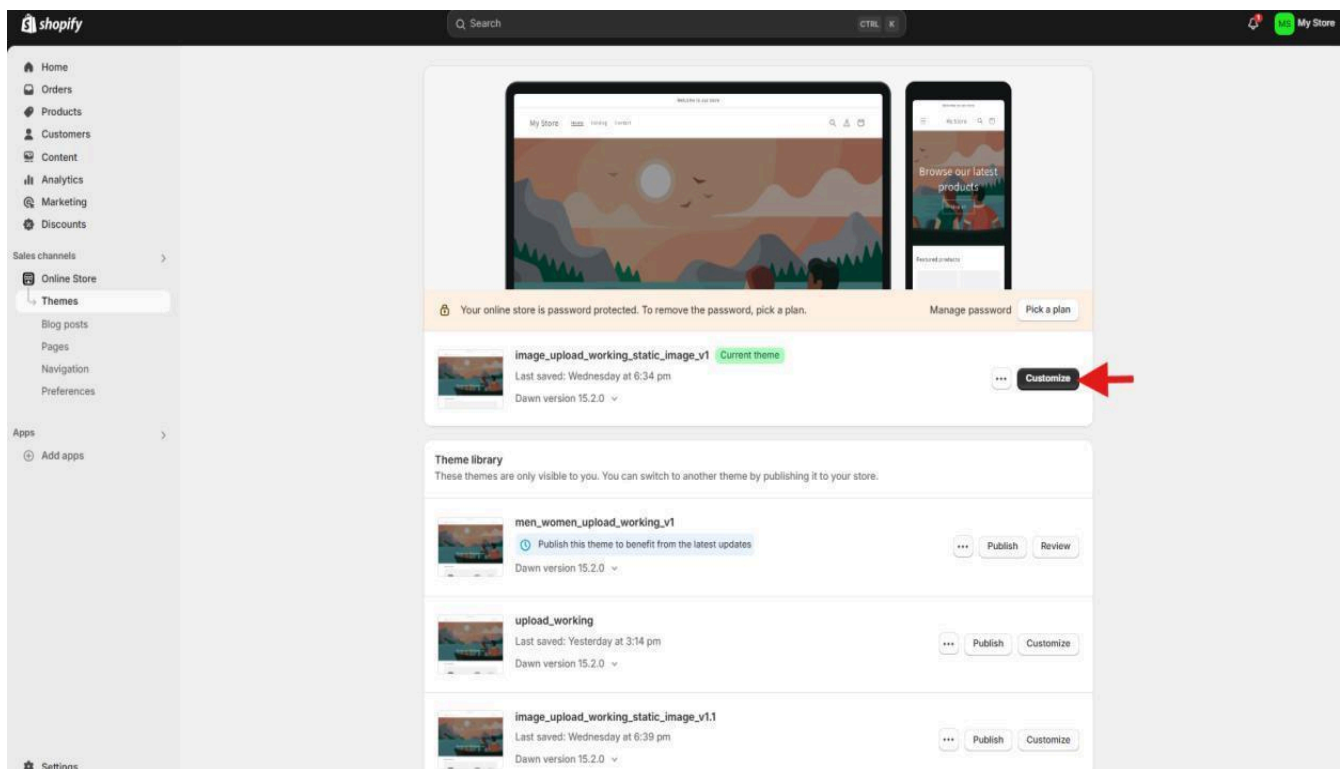
To successfully integrate the Tailori feature with Shopify, the following files and plugins are required -

1.	Theme.liquid (provided code add in body)
2.	TailoriApp.min.js
3.	taior_plugin_integration.js
4.	custom_tailor.css
5.	jQuery3.6.js

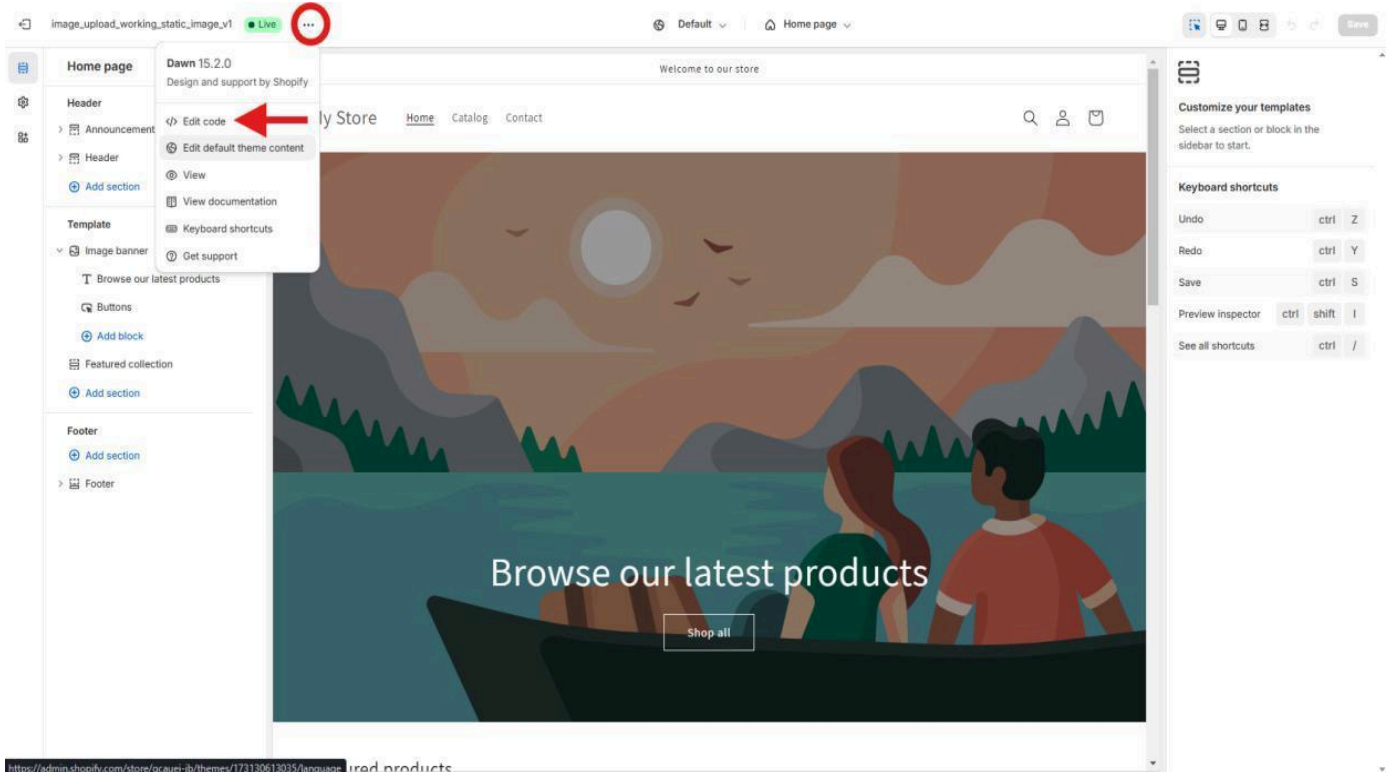
Step 1: Go to your Shopify admin site.



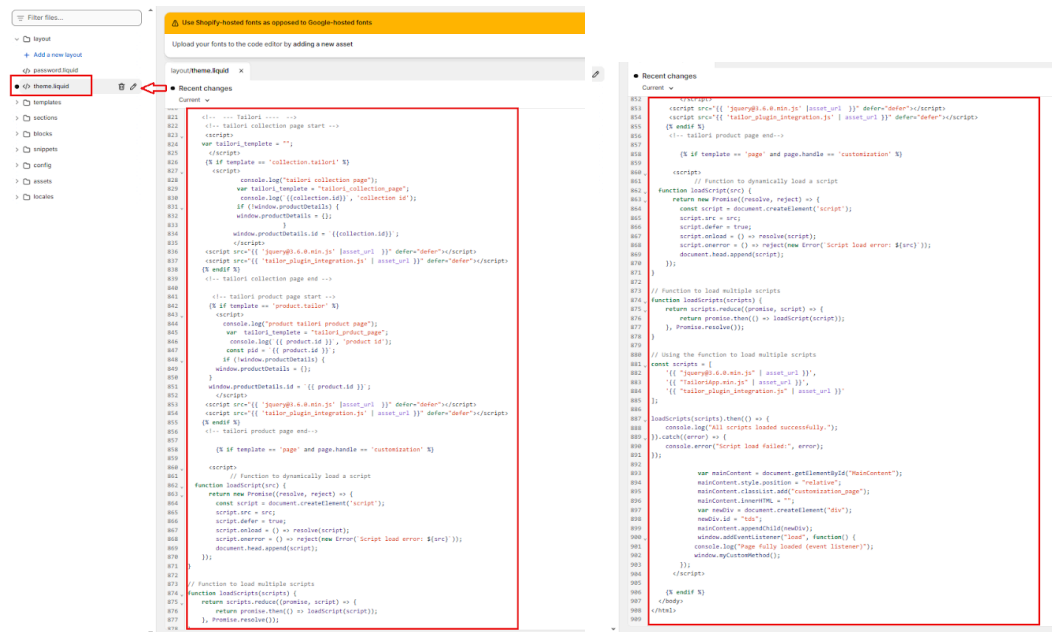
Step 2: Click on Online Store then click on “Customize”.



Step 3: Click on “Edit code”

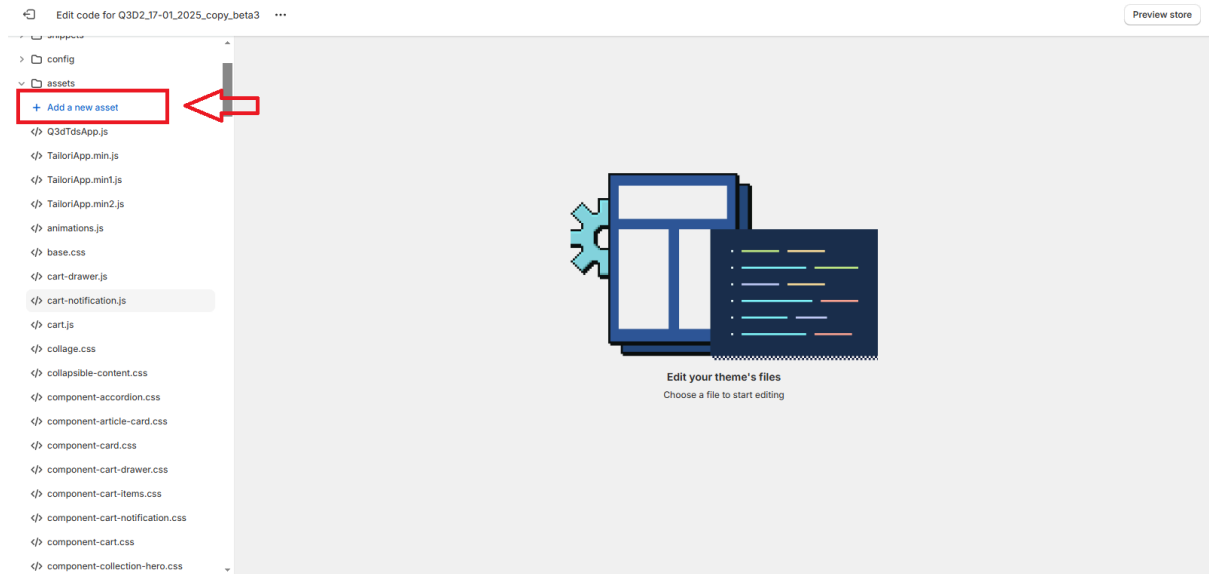


Step 4: Click on “{ } theme. Liquid Copy code from provided file and paste in body section at the end of body.



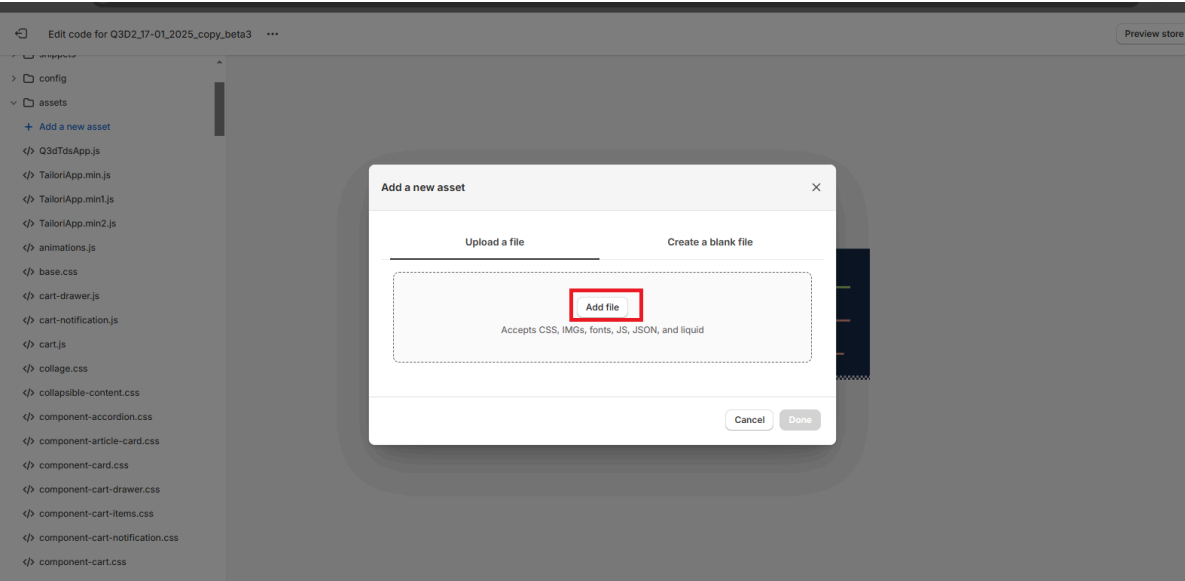
Step 4. Add Provided File JS (required)

- Click on “Add new assets”



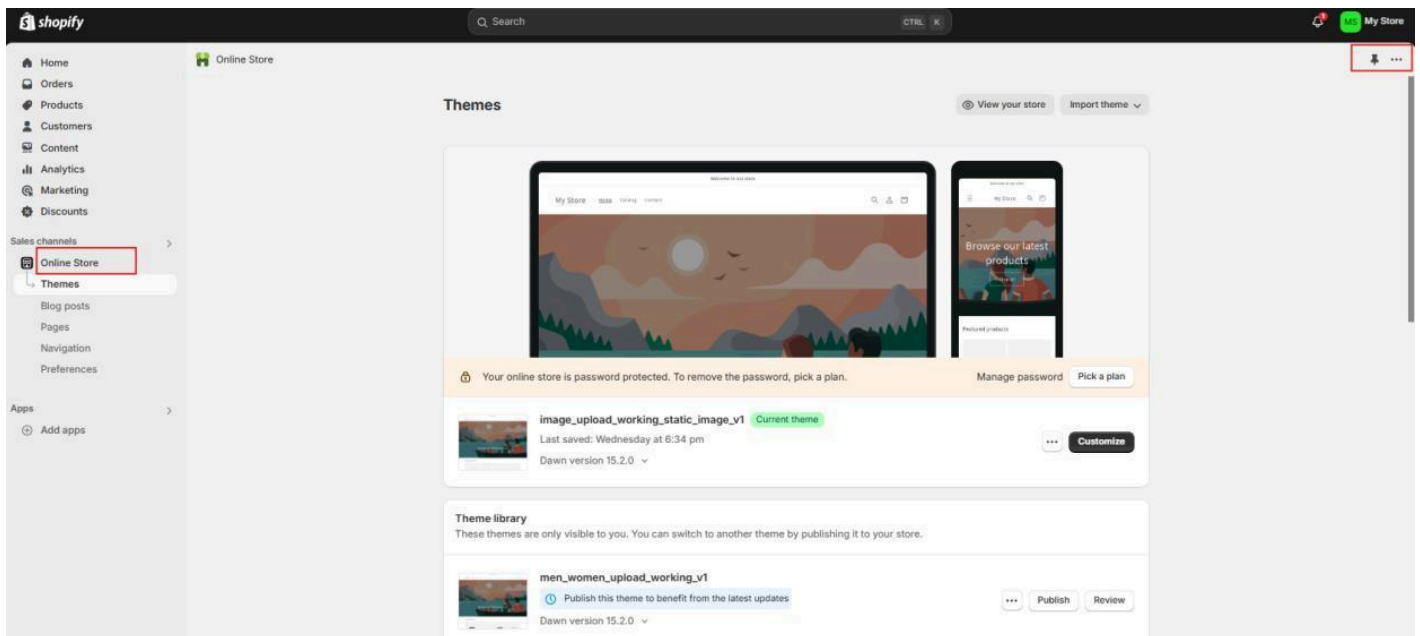
Upload Files in new assets

- custom_tailor.css
- JQuery3.6
- tailor_plugin_integration.js
- TailorApp.min.js

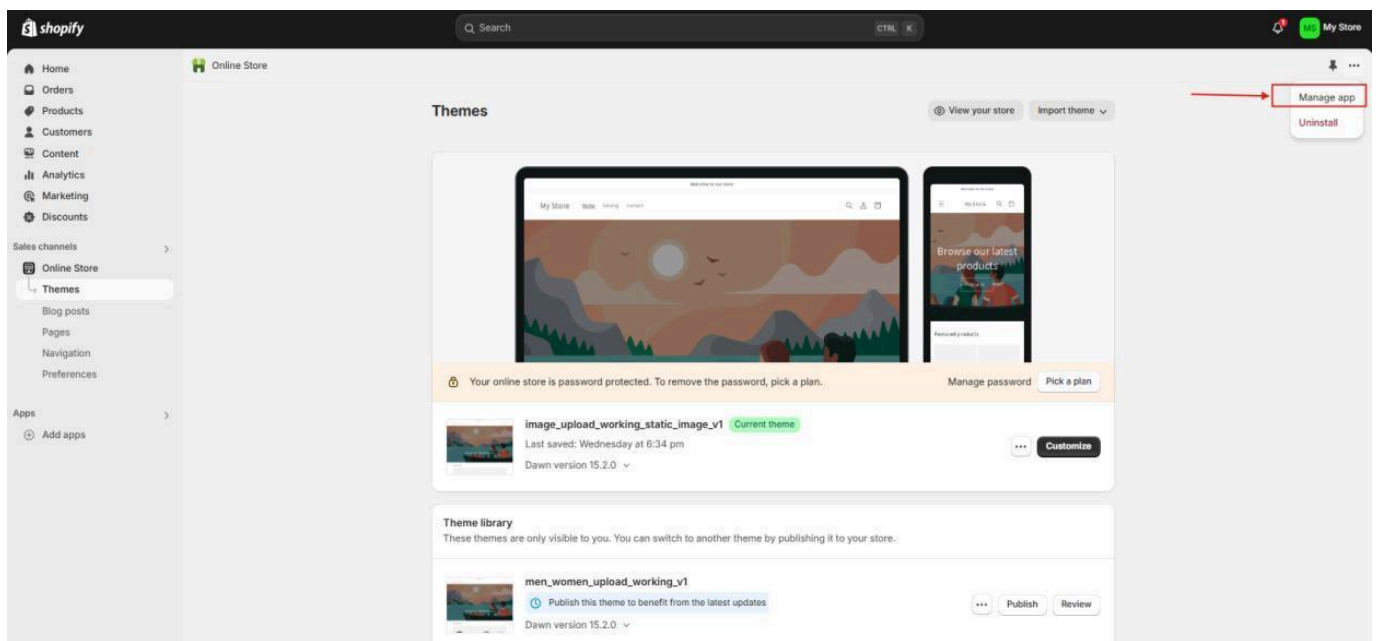


IX) Creating API Token for Products (need read access only)

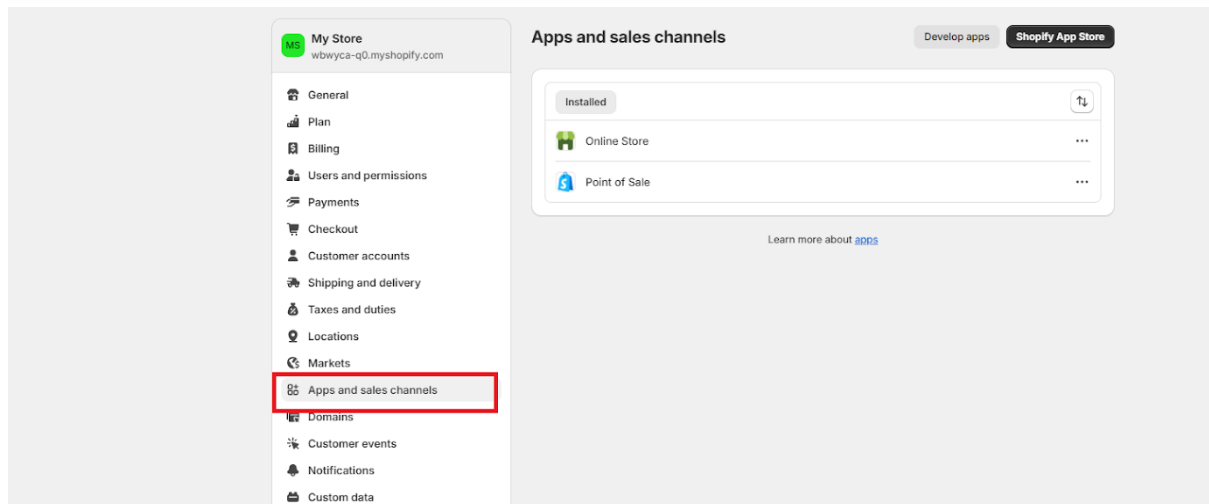
Step 1 : Click on Online Store and click on three dots symbol on the top right corner(...)



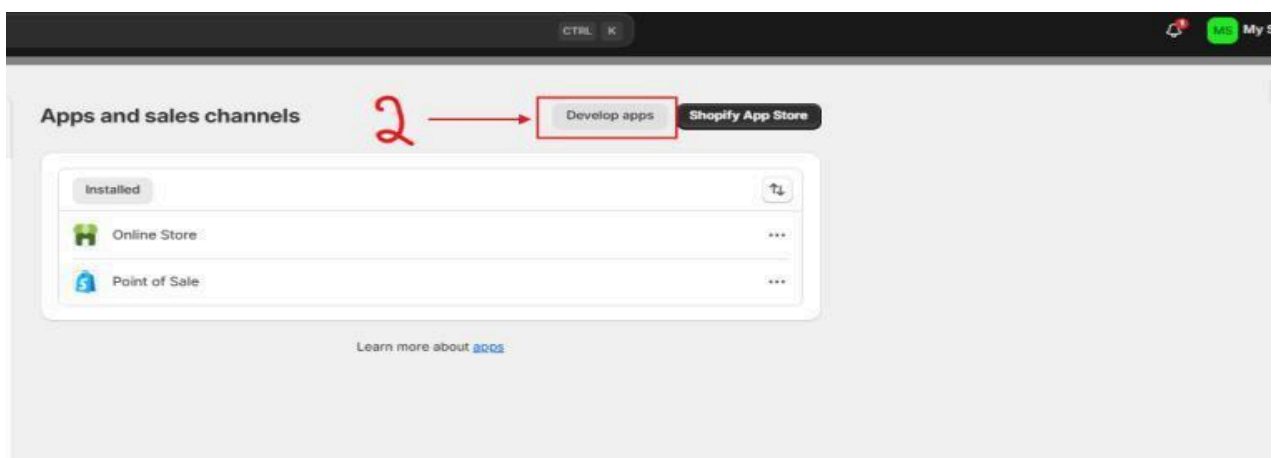
Step 2 : Click on Manage App



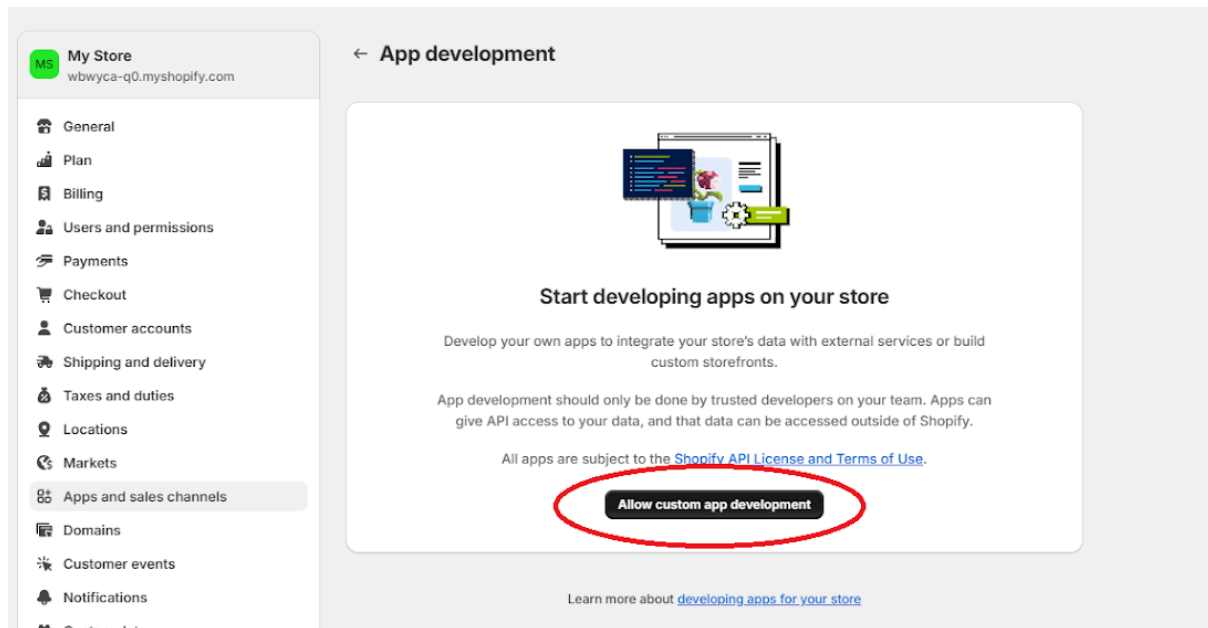
Step 3: Click On Apps and sales channels



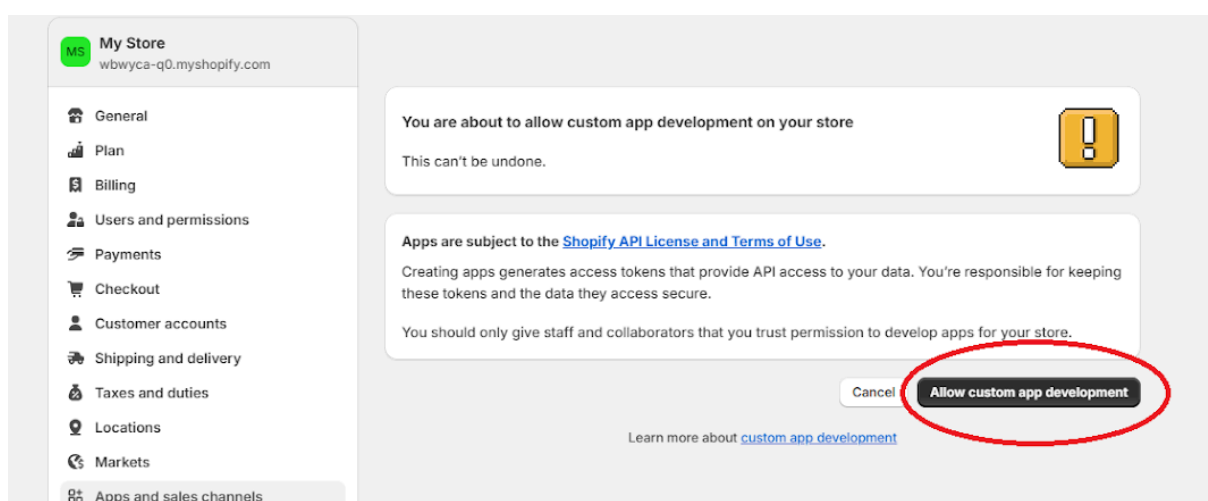
a) Click on Develop Apps



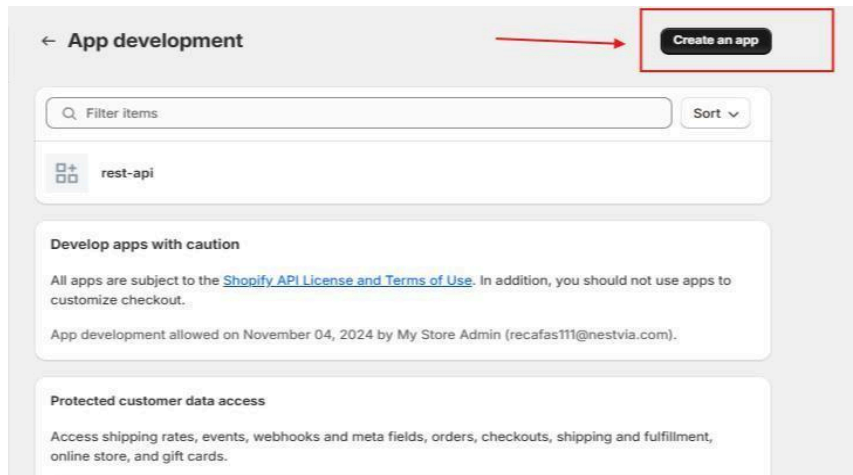
b) Click on allow custom app development



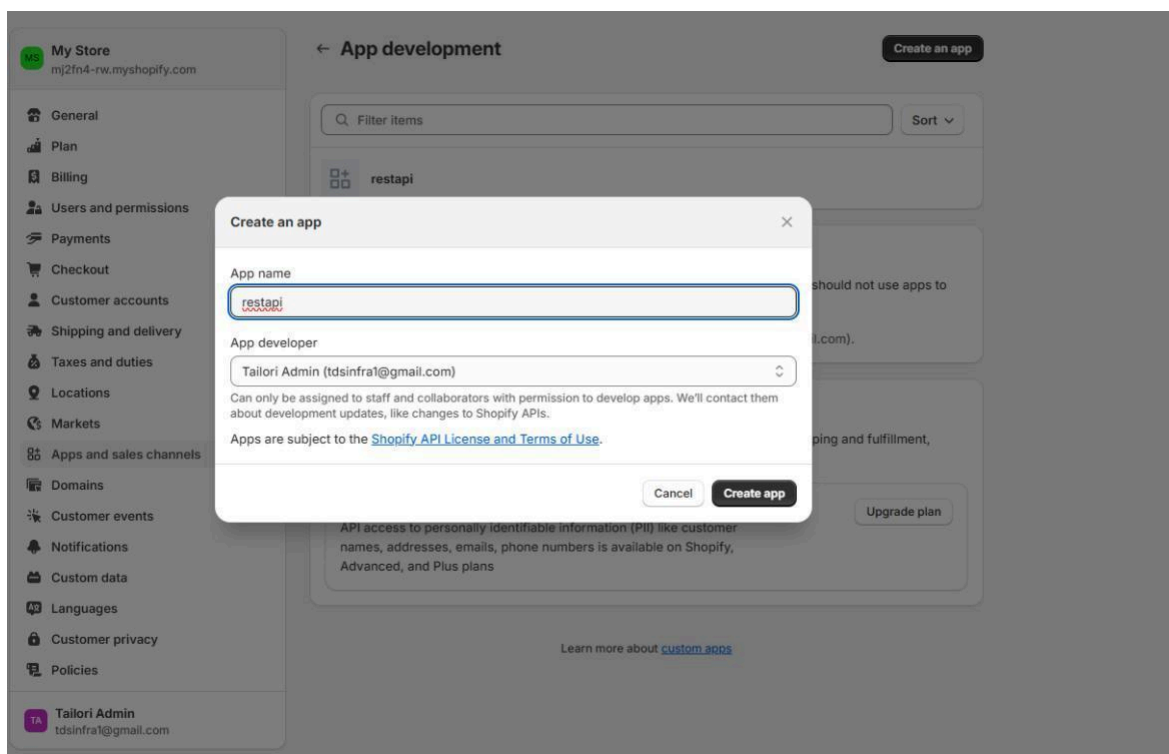
c) Click on allow custom app development



Step 4: Click on Create App



Step 5 : Enter app name example “restapi” or “testapi” (any name) & click on create app.



Step 6: Click on configuration and click configure

The screenshot shows the Shopify App development interface for an app named 'testapi'. The left sidebar contains a menu with various store settings, with 'Apps and sales channels' highlighted. The main content area is titled 'App development > testapi' and includes an 'install app' button. Below the title are tabs for 'Overview', 'Configuration', 'API credentials', and 'App settings'. The 'Configuration' tab is selected and highlighted with a red box and a red number '1'. Under the 'Configuration' tab, there are three sections: 'Admin API integration' with a 'Configure' button circled in red and labeled with a red number '2'; 'Storefront API integration' with a 'Configure' button; and 'Protected customer data access' with an 'Upgrade plan' button. The bottom of the sidebar shows the user 'Tailori Admin' with the email 'tdsinfra1@gmail.com'.

Step 7: Select necessary permissions & click on Save button (below selected elements for reference only)

The screenshot shows the Shopify permissions selection interface. The left sidebar contains a menu with various store settings, with 'Apps and sales channels' highlighted. The main content area is titled 'All Selected 8' and includes a search bar labeled 'Filter access scopes'. Below the search bar, there are several sections of permissions, each with a list of scopes and checkboxes. The 'Files' section has 'read_files' checked. The 'Fulfillment services' section has 'write_fulfillments' and 'read_fulfillments' checked. The 'Gift card transactions' section has 'write_gift_card_transactions' and 'read_gift_card_transactions' checked. The 'Gift cards' section has 'write_gift_cards' and 'read_gift_cards' checked. The 'Inventory' section has 'write_inventory' and 'read_inventory' checked. The 'Legal policies' section is partially visible at the bottom. The bottom of the sidebar shows the user 'My Store Admin' with the email 'vaviya2737@fundapk.com'.

Customer accounts

Shipping and delivery

Taxes and duties

Locations

Markets

Apps and sales channels

Domains

Customer events

Notifications

Custom data

Languages

Customer privacy

Policies

My Store Admin
vaviya2737@fundapk.com

AllSelected 8

Filter access scopes

Metaobject definitions

View or manage definitions

write_metaobject_definitions

read_metaobject_definitions

Metaobject entries

View or manage entries

write_metaobjects

read_metaobjects

Online Store navigation

View menus for display on the storefront

write_online_store_navigation

read_online_store_navigation

Online Store pages

View or manage Online Store pages

write_online_store_pages

read_online_store_pages

Order editing

View or manage edits to orders

write_order_edits

read_order_edits

Orders

Includes personally identifiable information

View or manage orders, transactions, fulfillments, and abandoned checkouts

write_orders

read_orders

Get access to store data with the [Admin API](#). Only select the scopes your app needs. Don't access APIs in ways that violate the [Shopify API License and Terms of Use](#).

Customer accounts

Shipping and delivery

Taxes and duties

Locations

Markets

Apps and sales channels

Domains

Customer events

Notifications

Custom data

Languages

Customer privacy

Policies

My Store Admin
vaviya2737@fundapk.com

AllSelected 8

Filter access scopes

read_price_rules

Product feeds

View or manage product feeds

write_product_feeds

read_product_feeds

Product listings

View or manage product or collection listings

write_product_listings

read_product_listings

Products

View or manage products, variants, and collections

write_products

read_products

Publications

View or manage groups of products that have been published to an app

write_publications

read_publications

Purchase options

View or manage purchase options owned by this app

write_purchase_options

read_purchase_options

Reports

View or manage reports on the Reports page in the Shopify admin

write_reports

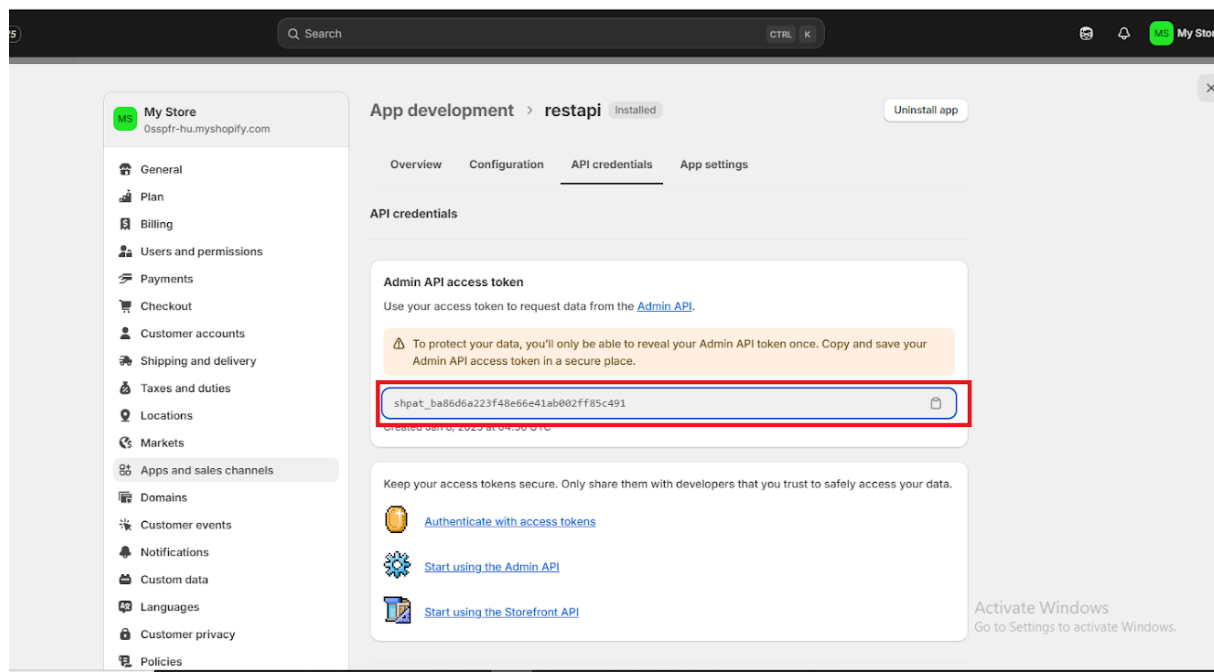
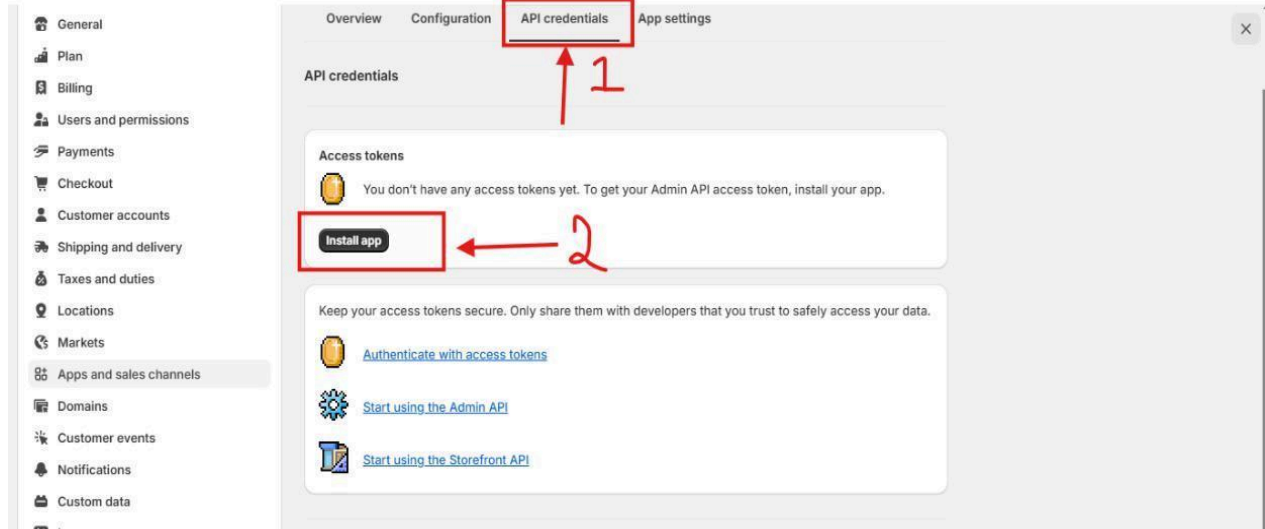
read_reports

Get access to store data with the [Admin API](#). Only select the scopes your app needs. Don't access APIs in ways that violate the [Shopify API License and Terms of Use](#).

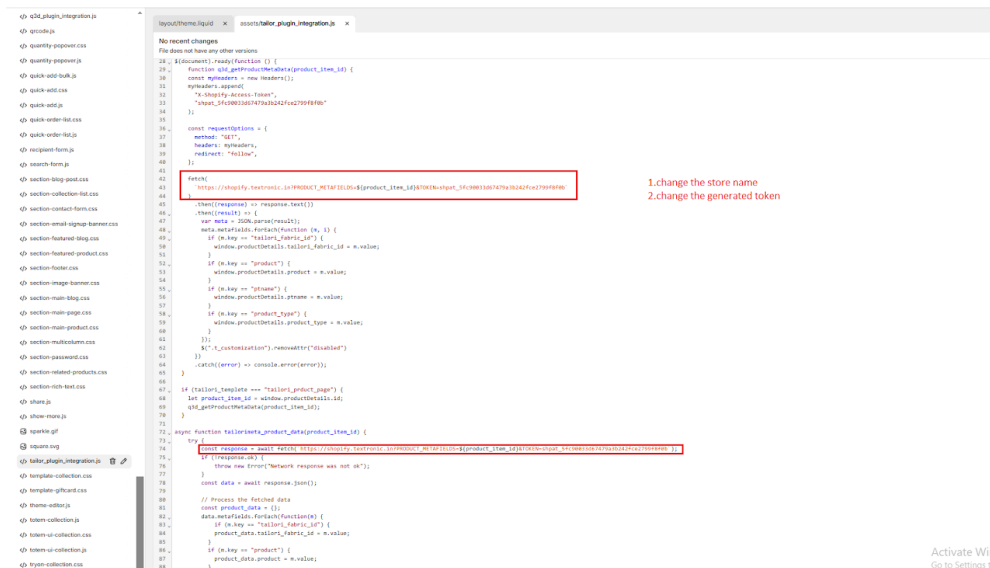
Active
Go to Settings

Step 8: Click on Api Credentials and Install App It will Generate Admin Access Token & Copy and keep Save

Token (Admin Access token required to fetch meta data details in javascript files)

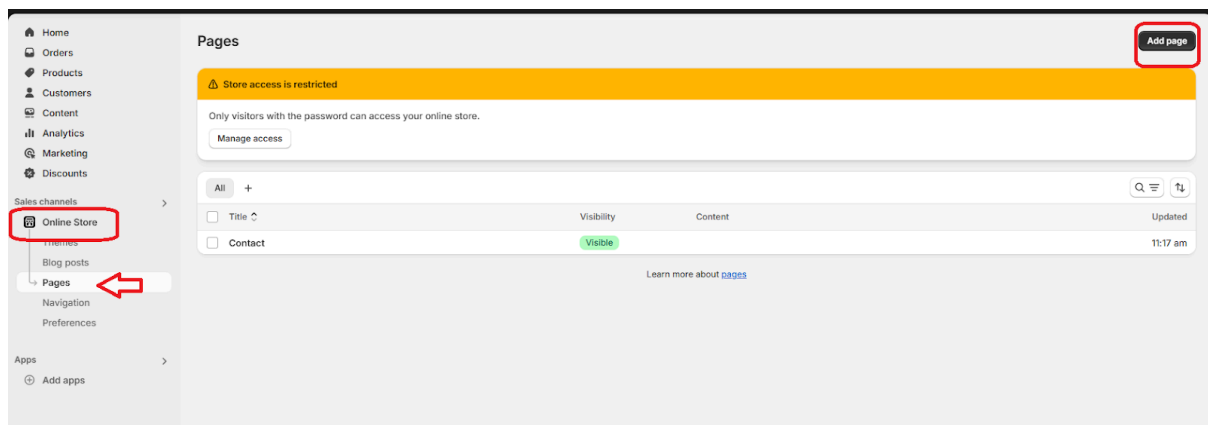


Step 9: Enter your access token in “tailor_plugin_integration.js” file which is already generated & Enter your token in Three places.



X) Add CUSTOMIZATION Page

Step 1: Click on online store > Click on Pages>Click on add page



Step 2:entering title “customization” > Select the visible option and save it.

The screenshot displays the 'Add Page' interface in the Shopify Admin. On the left, a sidebar contains navigation links for Home, Orders, Products, Customers, Content, Analytics, Marketing, and Discounts. Below this are 'Sales channels' (Online Store, Themes, Blog posts, Pages, Navigation, Preferences) and 'Apps' (Add apps). The main content area is titled '← Add Page'. It features a 'Title' field with the value 'customization', a 'Content' field with a rich text editor, and a 'Visibility' section with 'Visible' selected. The 'Template' is set to 'Default page'. At the bottom right, a red box highlights the 'Save' button.

Step 3: Replace Domain URL in file “**`tailor_plugin_integration.js`**” in below Provided screenshot underline url.

The screenshot shows a code editor with a file named 'tailor_plugin_integration.js'. The code is as follows:

```

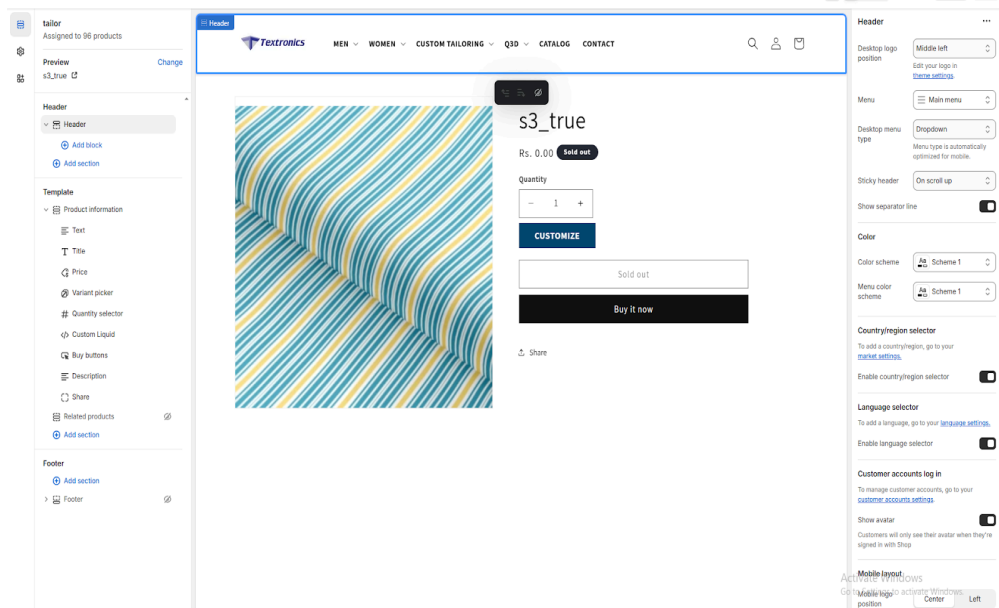
186     if (m.key == "product") {
187       product_data.product = m.value;
188     }
189     if (m.key == "ptname") {
190       product_data.ptname = m.value;
191     }
192     if (m.key == "product_type") {
193       product_data.product_type = m.value;
194     }
195   });
196   // Log the data
197   console.log(data);
198
199   return product_data;
200 } catch (error) {
201   console.error("There has been a problem with your fetch operation:", error);
202 }
203
204 // if (tailor_i_template == "tailor_i_collection_page") {
205 //   // tailor_i_getCollectionProductdata();
206 // }
207
208 $(body).on("click", ".tailor_i_collection_btn", async function (event) {
209   let product_id = $(this).attr("pid");
210   console.log(product_id);
211   let product_data = await tailorImeta_product_data(product_id)
212   window.location.href = "https://r00gay-xy.myshopify.com/pages/customization/&f_id=${product_data.tailor_i_fabric_id}&p_type=${product_data.product_type}&p_id=${window.product_id}";
213
214   $(body).on("click", ".t_customization", async function (event) {
215     window.location.href = "https://r00gay-xy.myshopify.com/pages/customization/&f_id=${window.productDetails.tailor_i_fabric_id}&p_type=${window.productDetails.product_type}&p_id=${window.productDetails.product_id}";
216   });
217 });

```

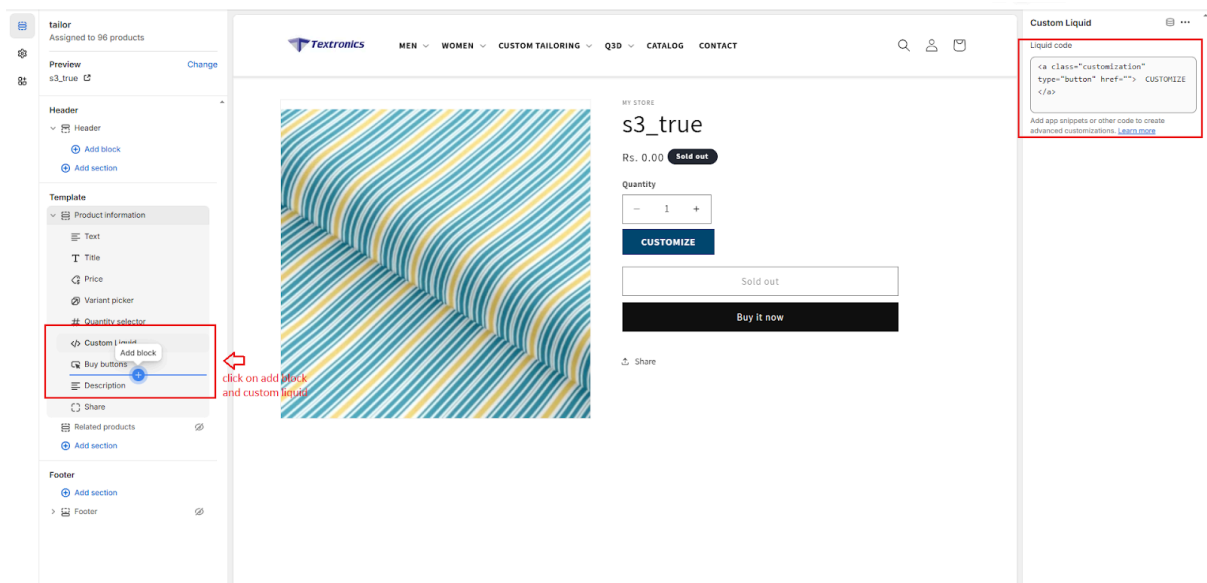
A red annotation points to the URL in line 212: "Replace the highlighted URL Change with domain URL eg. fab3d.in".

XI) Add CUSTOMIZE Button in Shopify [product Detail page]

- a) Click on customize > click on catalog product > Double click on product/image > hover on product information sub menu e.g(text or Title) > Add block name as “ custom liquid ”.

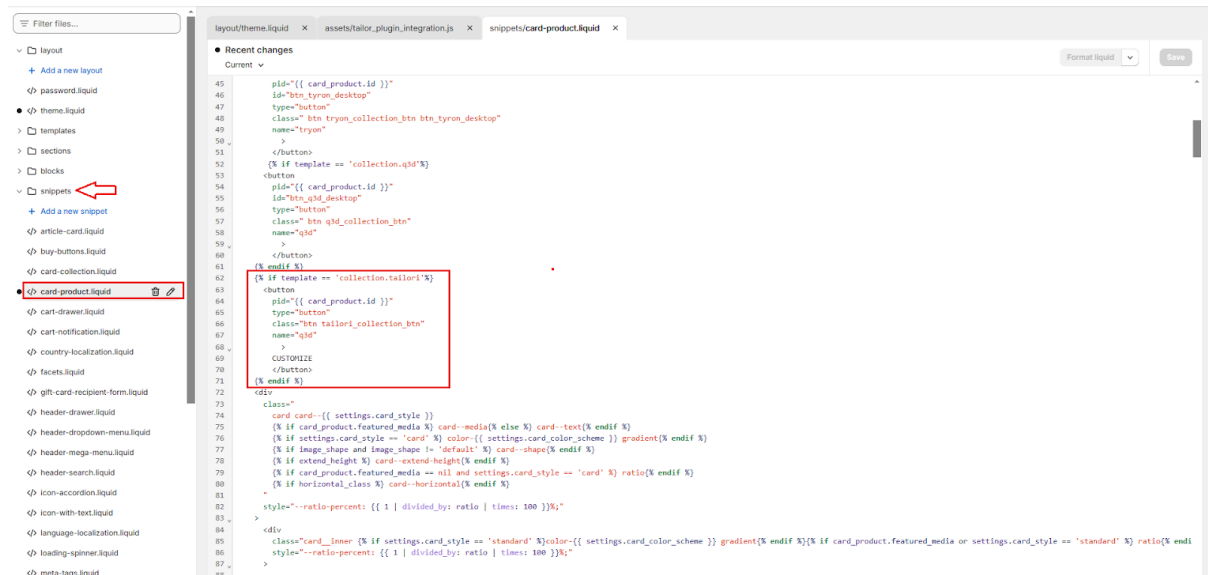


- b) Click on custom liquid & Button code is present in read.md text file, add here.



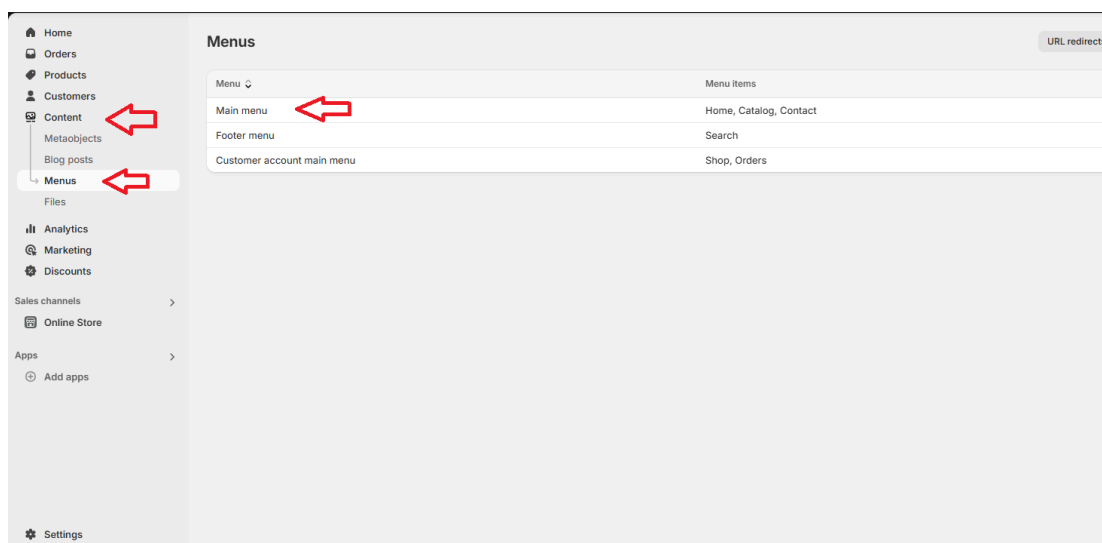
i) Add CUSTOMIZE Button in Shopify [category page]

Add this button code in file name path :- snippets> ["card-product.liquid"] Replace provided file.



ii) Add collection in catalog

Note : To show Customize button on collection page



Step1:Add menu item

Step 2:entering name select collections and select in create collection name

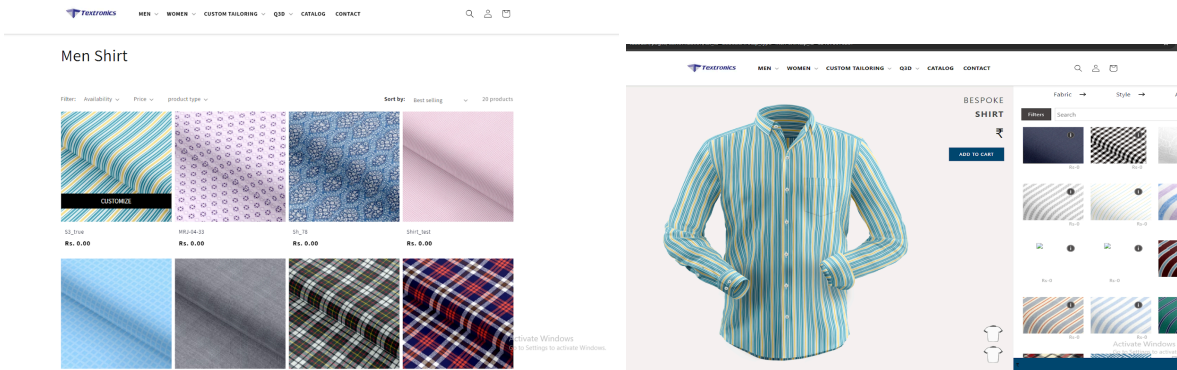
The screenshot shows the 'Main menu' configuration interface. At the top, there's a 'Name' field with 'Main menu' and a 'Handle' field with 'main-menu'. Below this is a 'Menu items' section with a list containing 'Home', 'Catalog', and 'Contact'. A red box labeled '1' highlights the 'Add menu item' button at the bottom. Another red box labeled '2' highlights the 'Label' field with 'Shirt' and the 'Link' dropdown menu. The 'Link' dropdown is open, showing options like 'Home page', 'Search', 'Collections' (highlighted with a red box), 'Products', 'Pages', 'Blogs', 'Blog posts', 'Policies', and 'Customer accounts'. A third red box labeled '3' highlights the 'Collections' option in the dropdown. Below the 'Link' dropdown, there's a 'Filter collection with tags (optional)' field with 'Hats, blue, summer'.

This screenshot shows the 'Main menu' configuration page with the 'Link' dropdown menu open. The 'Link' field now shows 'Shirt'. A red box highlights the 'Shirt' option in the dropdown menu. The 'Filter collection with tags (optional)' field still contains 'Hats, blue, summer'. The 'Add menu item' button is still visible at the bottom.

Step 3: Create Shirt drag nd drop in catalog menu & save it.

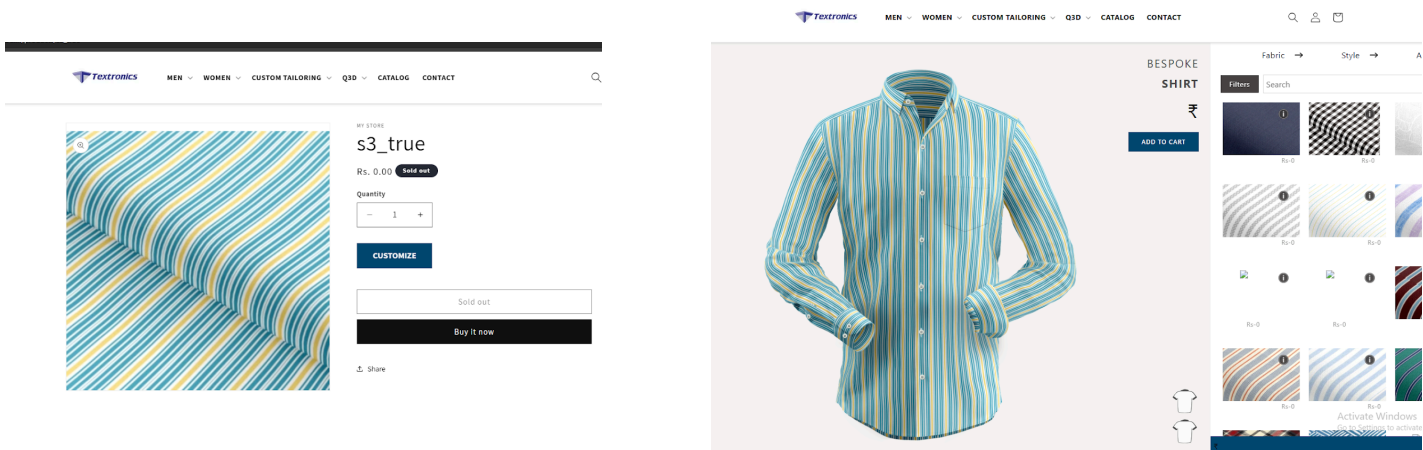
The screenshot shows the 'Main menu' configuration page. The 'Menu items' list now includes 'Home', 'Catalog', and 'Contact'. The 'Catalog' item is expanded, showing a sub-item 'Shirt'. A red box highlights the 'Catalog' item and its sub-item 'Shirt'. Below the 'Catalog' item, there's a button labeled 'Add menu item to Catalog'. At the bottom right, a red box highlights the 'Save' button.

Note :-Customize button on the collection page result



iii) Product Details Page Work

Clicking on Customize button will pop up Customize plugin right bottom corner



Thank you

Textronics Design Systems

**Plot. No. EL-109, 2nd Floor, TTC Electronic Zone, MIDC Mahape, Navi Mumbai 400 710,
INDIA**

CALL: +91 - 9867604882 / 8451942077

Email: info@textronic.net / sales@textronic.net