

1. Shopify Account Creation and Admin Dashboard Exploration

A Shopify account was created using the free trial option. After successful login, the Shopify Admin Dashboard was explored to understand store management features.

1.1 Admin Dashboard Overview

The admin panel provides access to orders, products, customers, analytics, marketing, and store settings.

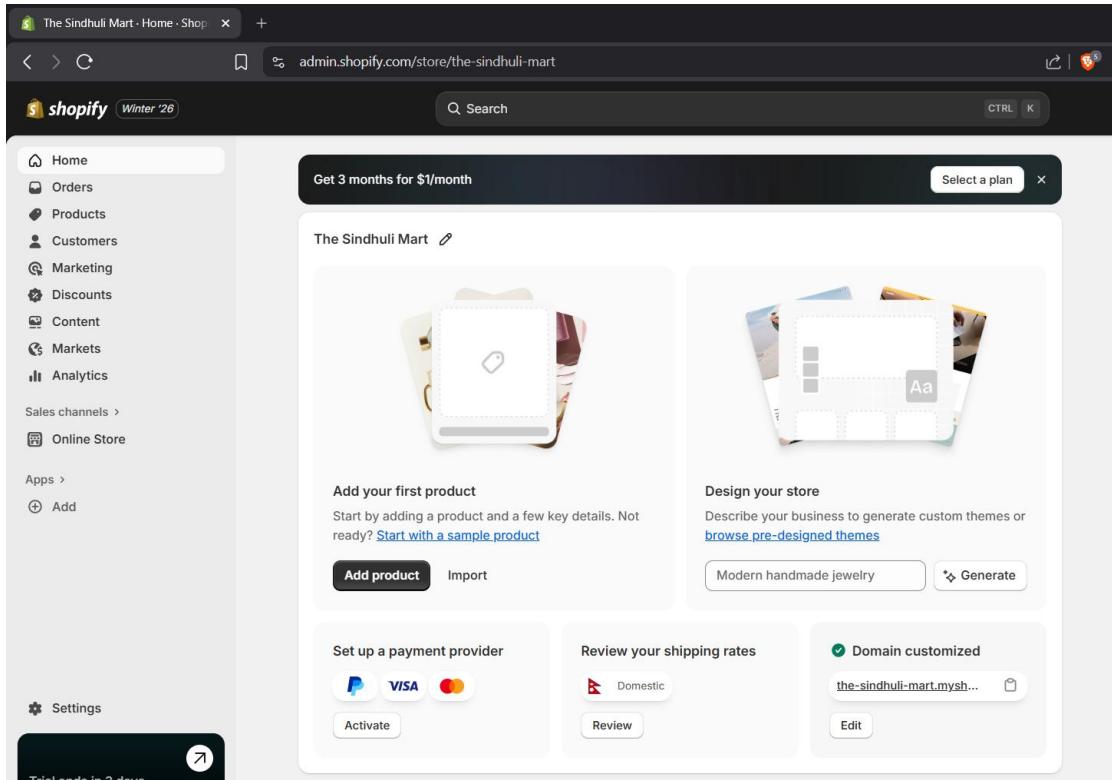


Figure 1: Admin Dashboard

Key Observations:

Order Confirmation Notifications: Automatic notifications are generated when a customer places an order, ensuring real-time order tracking.

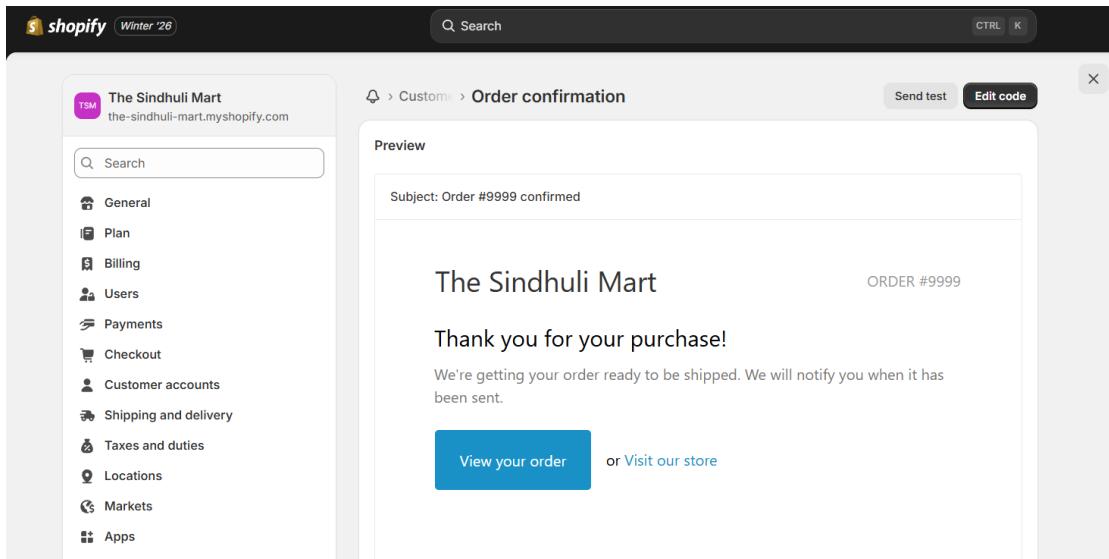


Figure 2: Order Confirmation Style

Delivery Rate: The dashboard displays delivery and fulfillment status, helping track order completion efficiency.

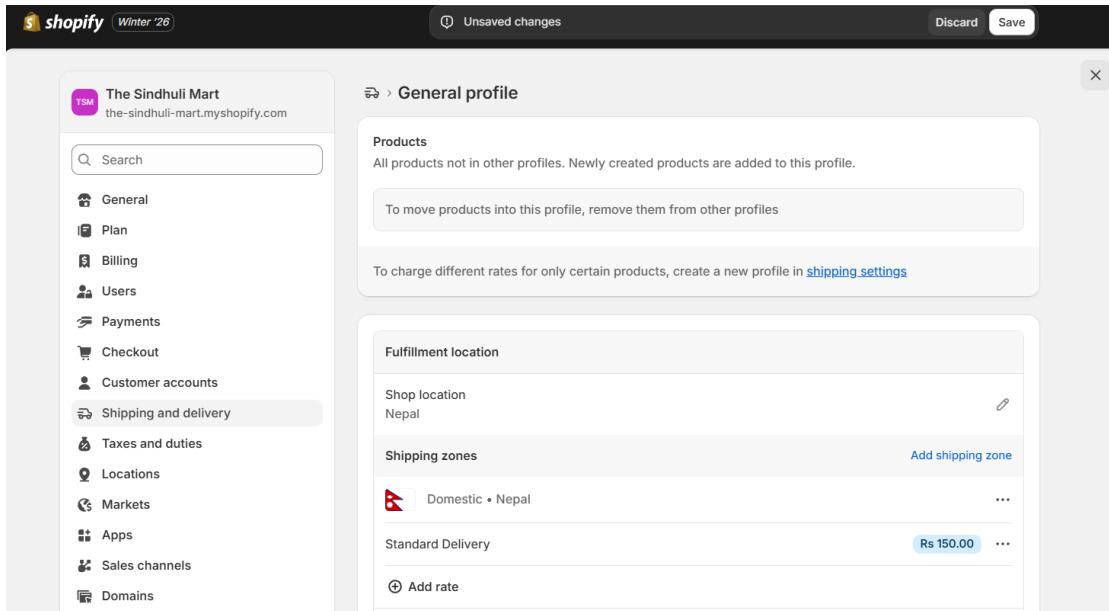


Figure 3: Delivery Rate

2. Theme Selection and Storefront Customization

2.1. Theme Selection

A theme was selected from the Shopify theme library and applied to the store.

Customization Process

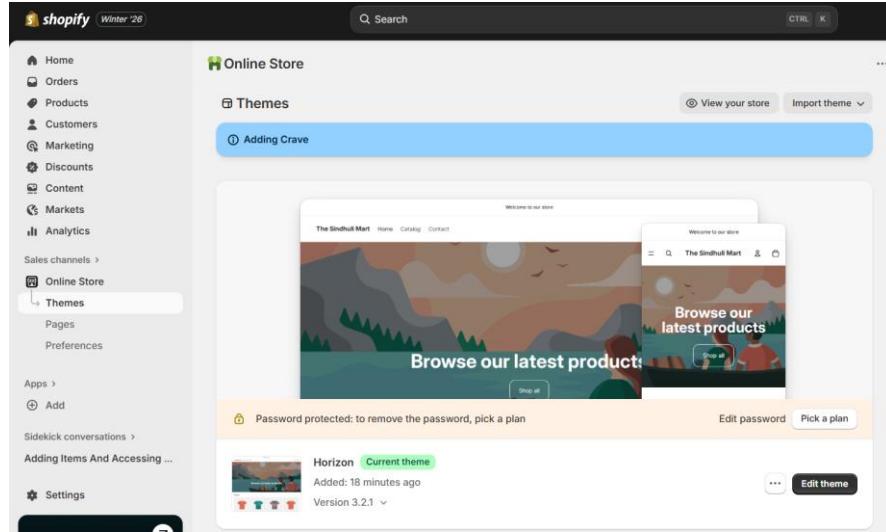


Figure 4: Themes in Shopify

Before Customization

1. Default theme layout
2. Standard color scheme
3. No logo
4. Default background and typography
5. A screenshot of the store before customization was taken.

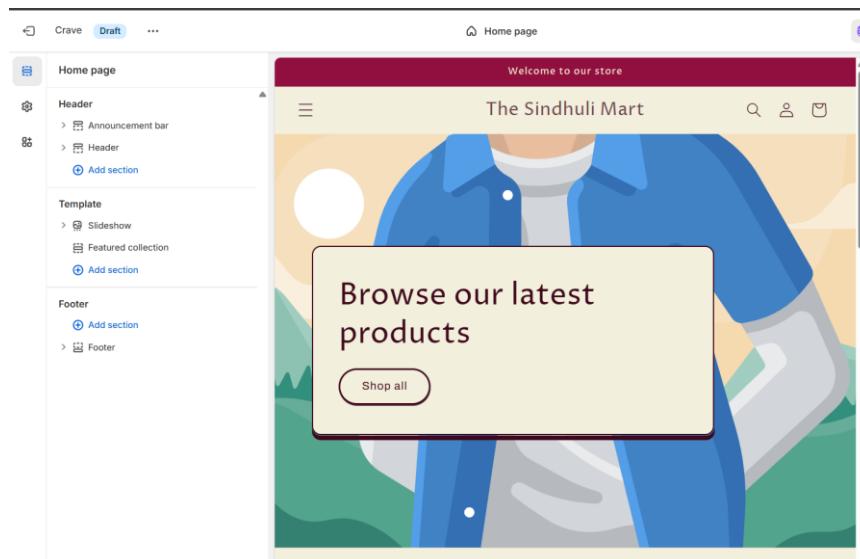


Figure 5: Before Customization

After Customization

1. Customization was applied to improve branding and visual appeal.
2. Changes Made:
 - a. Updated background images and positioning
 - b. Modified color themes
 - c. Added store logo
 - d. Adjusted layout for better responsiveness

Desktop View

The homepage layout was optimized for desktop screens, ensuring proper alignment and spacing.

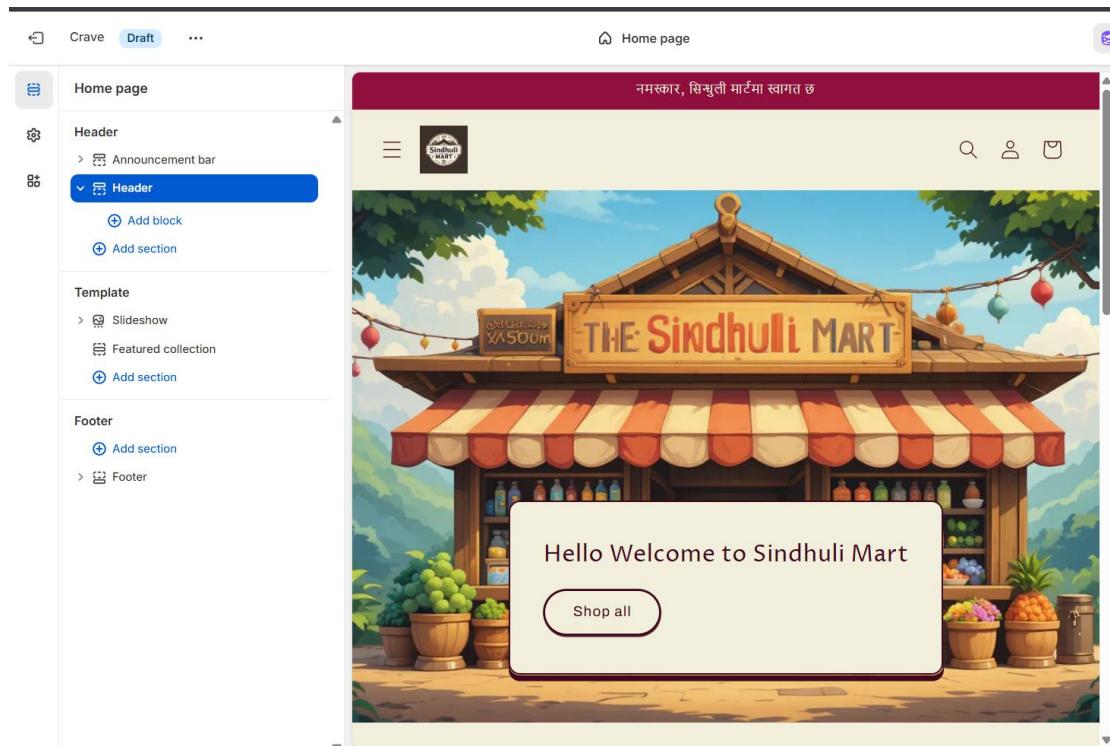


Figure 6: Desktop View After Customization

Mobile View

The mobile layout was adjusted to maintain readability and usability on smaller screens.

- a. Theme Customization (Before)
- b. Customization (After)
 - i. Desktop View
 - ii. Mobile View

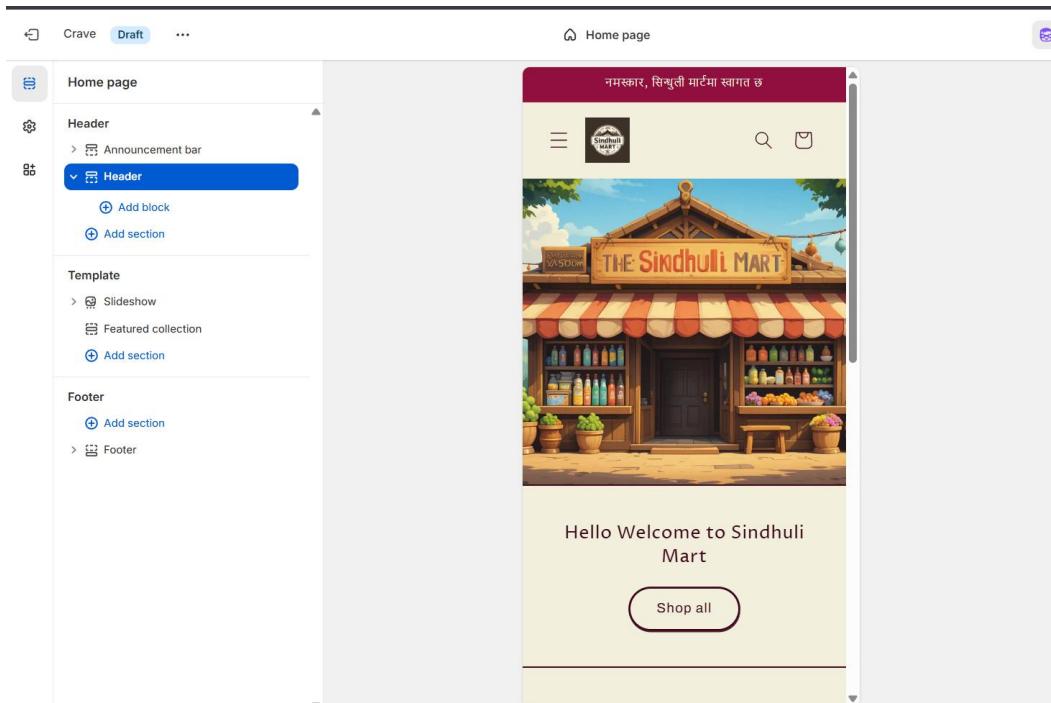


Figure 7: Mobile View After Customization

3. Adding Products to the Store

A minimum of five products were added to the Shopify store.

Each product includes the following details:

1. Product title
2. Product image
3. Description
4. Price
5. Stock quantity

These details allow proper display and inventory management within the store.

The screenshot shows the 'Add product' page in the Shopify admin. The product is titled 'EZ Oil Bottle'. The description is: 'A sleek, spill-proof oil bottle designed for easy and controlled pouring. Ideal for everyday cooking and food preparation, it helps keep your kitchen clean while storing oil safely and conveniently.' The media section shows a thumbnail of the oil bottle and a plus sign to add more. The category is 'Oil Dispensers in Oil & Vinegar Dispensers'. The status is 'Active' and publishing is set for 'Online Store'. In the 'Product organization' section, 'Type' is 'Oil Bottle'. The vendor is 'The Sindhuli Mart'. Under 'Collections', there is a search bar and two checked boxes: 'Food and Consumption' and 'Home page'. The theme template is 'Default product'.

Figure 8: Adding Product

Price

Rs 350.00

Compare at Unit price Charge tax Yes Cost per item

Inventory

Inventory tracked

Quantity	Quantity
Shop location	150

More details

SKU (Stock Keeping Unit) Barcode (ISBN, UPC, GTIN, etc.)

EZ-OB-4921 735291846027

Continue selling when out of stock

Shipping

Physical product

Package i

Sample box • 22 × 13.7 × 4.2 cm, 0 kg

Product weight kg

Country of origin HS Code

Figure 9: Product Details while Adding Product

Products					
	All	Active	Draft	Archived	+ Add product
<input type="checkbox"/>	Product	Status	Inventory	Category	Channels
<input type="checkbox"/>	EZ Oil Bottle	Active	150 in stock	Oil Dispensers	2
<input type="checkbox"/>	Peanut Butter	Active	101 in stock	Nut Butters	2
<input type="checkbox"/>	Frozen Momo	Active	500 in stock	Prepared Meals & Entrées	2
<input type="checkbox"/>	Cheese	Active	76 in stock	Cheese	2
<input type="checkbox"/>	Cumin Seed Jeera	Active	200 in stock	Herbs & Spices	2
<input type="checkbox"/>	Basmati Long Grain Rice	Active	100 in stock	Rice	2

Learn more about products

Figure 10: List of Products

3.1. Product Collections

Creating Collections

1. Product collections were created to organize related products.

Shopify Winter '26

Unsaved changes

Discard Save

Home Orders Products Collections Inventory Purchase orders Transfers Gift cards Customers Marketing Discounts Content Markets Analytics Sales channels > Online Store Apps > Add Settings

Add collection

Title: Food and Consumtion

Description: Rice, Momo and basic essentials

Publishing:

- Sales channels
 - Online Store
 - Point of Sale

Manage

Image: Add image or drop an image to upload

Collection type:

- Manual: Add products to this collection one by one. Learn more about [manual collections](#).
- Smart: Existing and future products that match the conditions you set will automatically be added to this collection. Learn more about [smart collections](#).

Products

Theme template: Default collection

Figure 11: Creation of Collection

2. Adding Products to Collections

Products were assigned to appropriate collections for easier navigation and better customer experience.

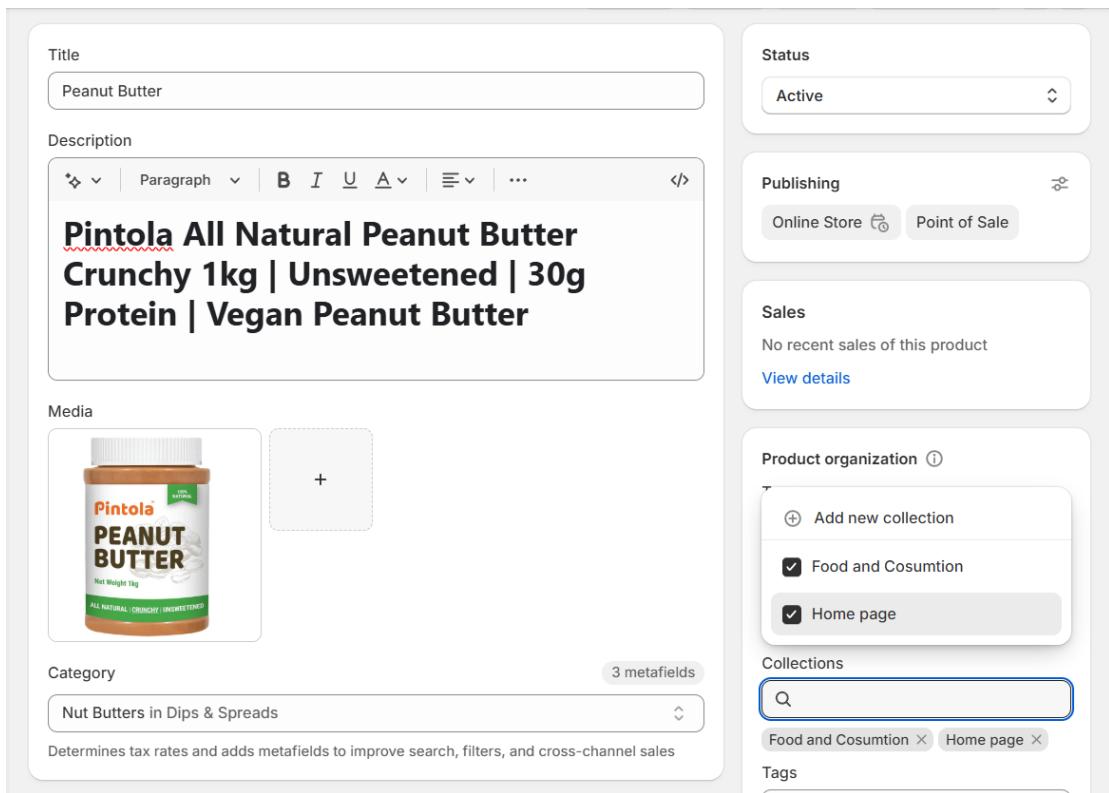


Figure 12: Addition Products to Collection

Collections			
All	Rice	+	Add collection
<input type="checkbox"/>	Title	Products	Product conditions
<input type="checkbox"/>	 Home page	3	
<input type="checkbox"/>	 Food and Cosumption	2	
Learn more about collections			

Figure 13: Product Collections

4. Navigation Menu Configuration

- a. Navigation menus were configured from the Shopify admin panel.
- b. Menu Links Created:
 1. Home
 2. Products
 3. About
 4. Contact
 5. Menu Display

The screenshot shows the Shopify admin interface for configuring a navigation menu. The left sidebar has 'Menus' selected under 'Content'. The main area is titled 'Main menu' with a 'Name' field set to 'Main menu' and a 'Handle' field set to 'main-menu'. Below this is a table titled 'Menu items' containing five entries:

Label	Link
Home	Home page
View All Collections	All collections
Shop	All products
Contact	Profile
About	Profile

At the bottom of the menu items table is a button labeled '+ Add menu item'.

Figure 14: Navigation Menu

- c. Menus were displayed correctly on the storefront



Figure 15: Navigation in Desktop View

d. Mobile-friendly navigation was verified



Figure 16: Mobile View

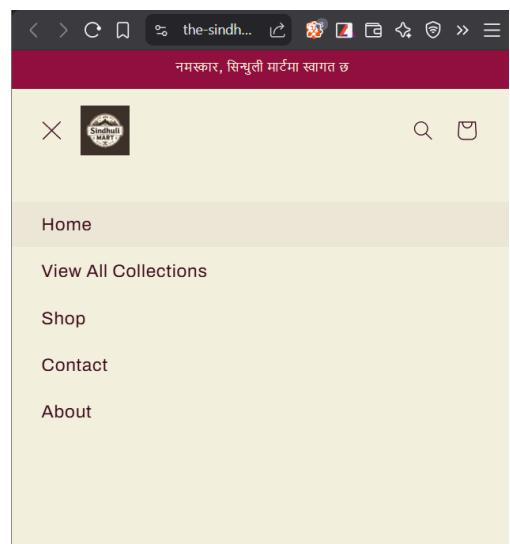


Figure 17: Navigation for Mobile View

5. Create Discount Code

1. Opened the Discounts section from the Shopify admin panel and initiated the option to add a new discount.
2. Selected a suitable discount category, specifically an order-level amount reduction.
3. Created the discount code either by using Shopify's auto-generation feature or by entering a custom code.
4. Defined the discount rules by setting customer eligibility, minimum order value, and usage restrictions.
5. Specified the activation period by assigning start and expiration dates.
6. Finalized and saved the settings to make the discount active.

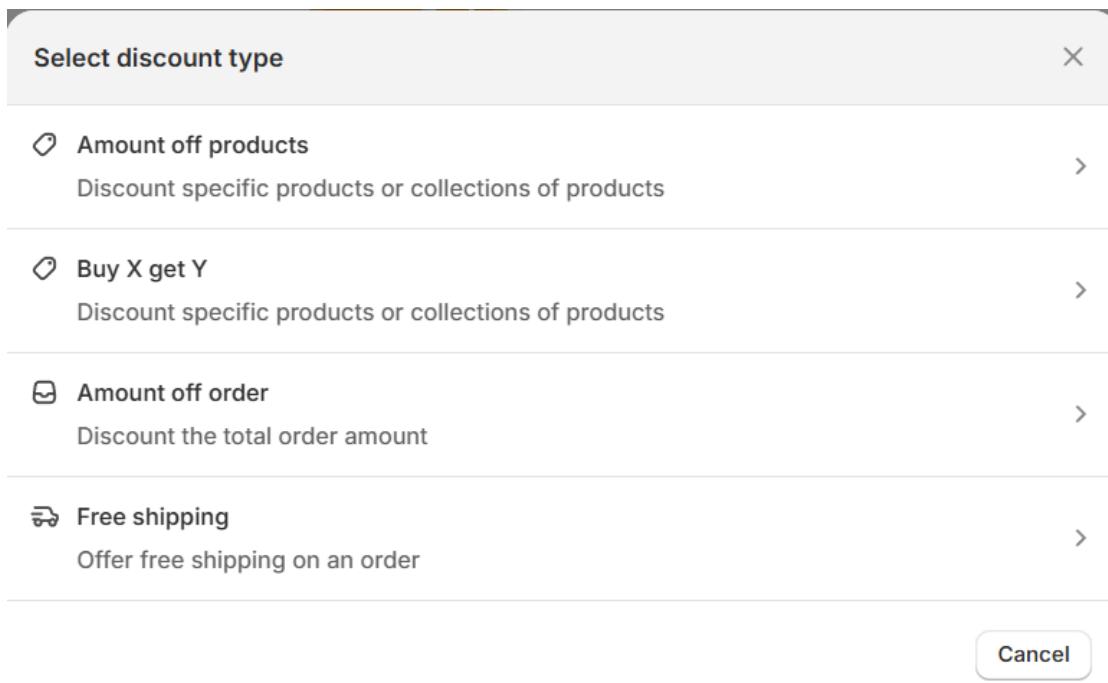


Figure 18: Discount Type Selection

Amount off products

Method

Discount code Automatic discount

Discount code [Generate random code](#)

SINDHULI

Customers must enter this code at checkout.

Discount value

Percentage %

Applies to

Specific collections [▼](#)

[Browse](#)

 Food and Cosumption
2 products [X](#)

Eligibility

Available on all sales channels

All customers
 Specific customer segments
 Specific customers

SINDHULI [Code](#)

Type

Amount off products
 Product discount

Details

- All customers
- For Online Store
- 15% off Food and Cosumption
- No minimum purchase requirement
- No usage limits
- Can't combine with other discounts
- Active from Jan 6

Sales channel access

Allow discount to be featured on selected channels

Figure 19: Discount Upon Products

6. Install and Explore any App from Shopify App Store

6.1. Installation of Shopify Inbox

1. Logged into the Shopify Admin Dashboard.
2. Navigated to the Shopify App Store from the dashboard menu.
3. Searched for “Shopify Inbox” using the search bar.
4. Selected the app from the search results and clicked Add App.
5. Confirmed the installation by clicking Install App, which automatically integrated Shopify Inbox with the store.
6. After installation, the app was accessible directly from the Shopify dashboard, ready for use.

The installation process was straightforward and required no additional configuration. Shopify Inbox connected automatically to the store, allowing immediate access to basic chat features such as live messaging and automated greetings. This demonstrates how Shopify apps can be quickly added to a store to extend functionality with minimal effort.

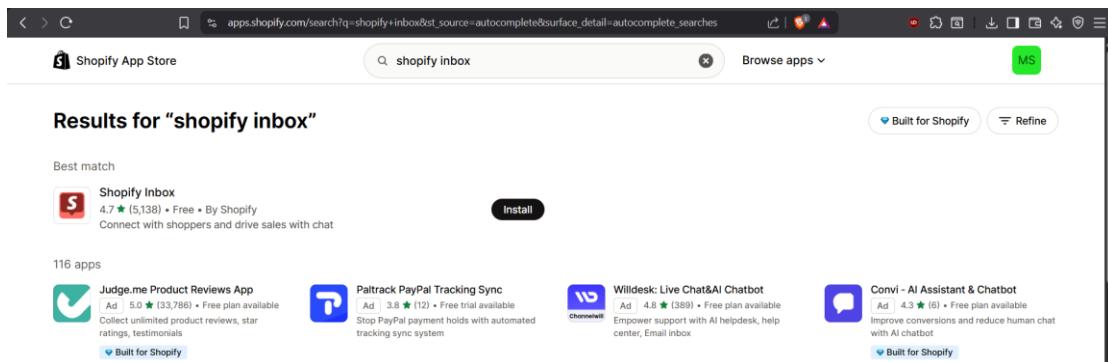


Figure 20: Shopify Store

A free application, Shopify Inbox, was searched and installed from the Shopify App Store to explore its customer communication functionality. After installation, the app was automatically connected to the Shopify dashboard, allowing access to basic customer interaction features.

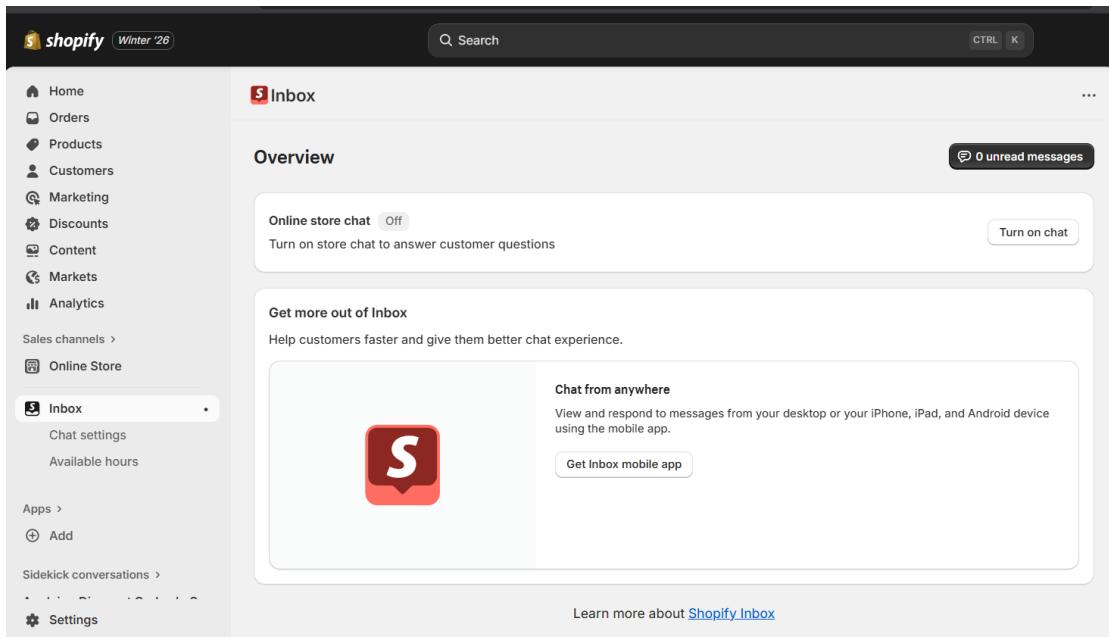


Figure 21: Shopify Inbox

The app interface was briefly explored to understand its core features, including live chat, automated greetings, and message tracking. A test conversation was initiated to demonstrate the app's communication capability. This simple exploration illustrated how Shopify Inbox can be used to interact with customers and support basic store communication

7. Set Up a Blog on Shopify Store

1. Navigated to the Apps section in the Shopify admin dashboard and selected Shopify Inbox to manage customer communication.
2. Clicked on Add App to install Shopify Inbox from the Shopify App Store.
3. Confirmed installation and accessed the Inbox dashboard directly from the Shopify admin panel.
4. Explored the interface to understand core features, including live chat, automated greetings, and message tracking.
5. Tested sending a simple message to verify the app's communication functionality.
6. Configured basic visibility and notification settings to ensure messages would appear to customers in real time.
7. Saved the settings and confirmed that Shopify Inbox was active and ready for use on the storefront.

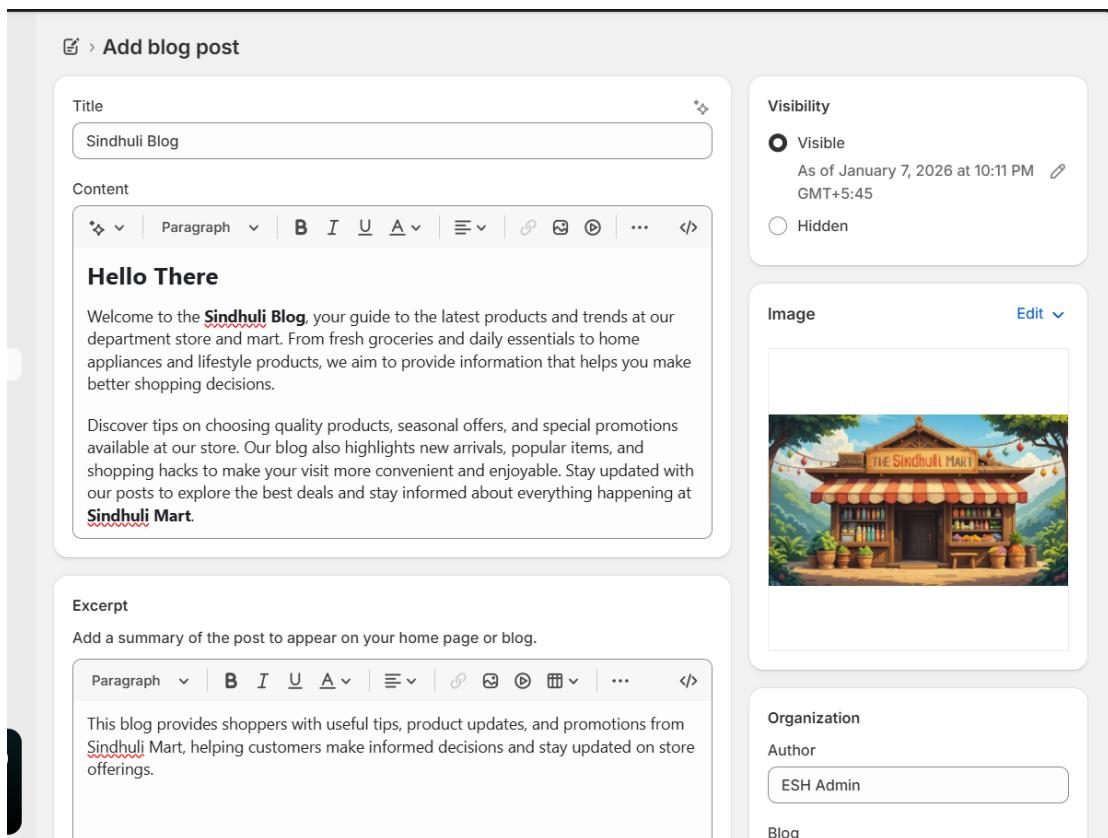


Figure 22: Blog Creation

Shopify Inbox enhances the store's communication capabilities by allowing real-time chat with customers. This app helps answer queries, provide support, and engage

visitors immediately, which improves customer experience. Even minimal use demonstrates how store owners can interact with customers beyond product listings, support trust-building, and encourage sales.

The screenshot shows the Shopify blog management interface. At the top, there's a header with 'Blog posts' and buttons for 'Manage blogs', 'Manage comments', and 'Add blog post'. A yellow banner at the top says '⚠ Store access is restricted' with the subtext 'Only visitors with the password can access your online store.' Below this is a 'Manage access' button. The main area has a table with columns: All (+), Title, Visibility, Author, Blog, Updated (dropdown), and Published. One post is listed: 'Sindhuli Blog' by 'ESH Admin' from 'News' updated 'Just now' on 'Jan 7, 2026'. The visibility is set to 'Visible'. There's also a link 'Learn more about blog posts'.

Figure 23: Blog List

Additionally, the app contributes to better store management by keeping track of customer messages and providing automated greetings, making communication more efficient and professional.

The screenshot shows the 'Sindhuli Blog' page on the Shopify storefront. The top navigation bar includes links for Home, View All Collections, Shop, Contact, and Blog. The main content features a decorative banner with a floral pattern. The title 'Sindhuli Blog' is displayed, along with the date 'JANUARY 7, 2026' and author 'ESH ADMIN'. Below the title is a 'Share' button. The blog post content starts with 'Hello There'. It welcomes visitors to the 'Sindhuli Blog' and describes its purpose: providing information on products and trends. It encourages visitors to stay updated with posts and explore deals. The footer of the page includes a 'Crave' logo, a 'Password protected' note, and buttons for 'View as', 'Nepal', 'English', and 'Edit theme'.

Figure 24: Viewing Blog

8. LAB 4.1: Create a google analytics account

1. Opened the Google Analytics website and signed in with a Google account.
2. Clicked Start measuring to create a new analytics account.
3. Entered a name for the account and proceeded to configure a new property.
4. Selected Google Analytics 4 (GA4) as the property type.
5. Provided the e-commerce store name, reporting time zone, and currency.
6. Chose Web as the data stream type and entered the store's website URL.
7. Completed the setup and saved the property settings to enable analytics tracking.

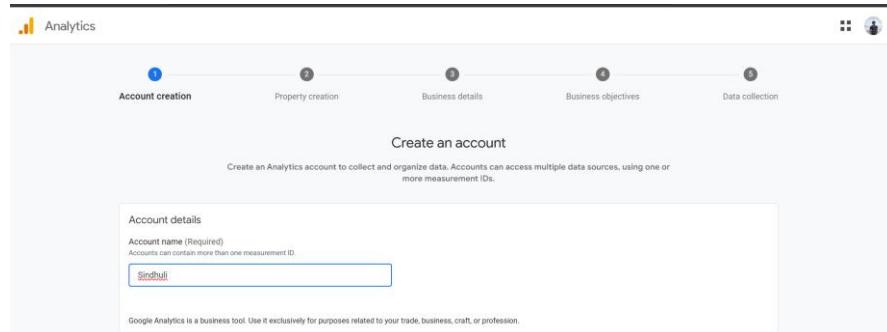


Figure 25: Account Creation For Google Analytics

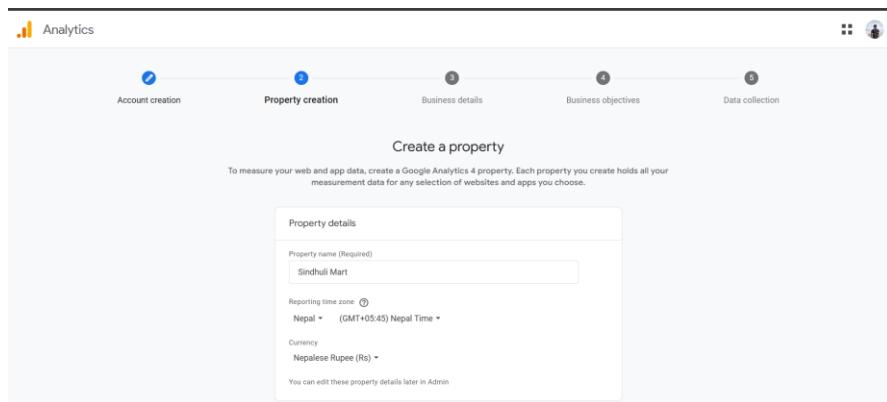


Figure 26: Property Creation

Adding Google Analytics to the Shopify store provides valuable insights into customer behavior and website performance. It allows tracking of visitors in real time, identifies traffic sources, and highlights popular products and pages. By analyzing this data, store owners can evaluate marketing effectiveness, make informed decisions to improve the shopping experience, and identify trends that support long-term growth and better customer engagement.

9. Add Google Analytics Tracking Code to Website

1. Accessed the Shopify Admin Dashboard by logging in.
2. Went to Online Store → Themes and clicked Edit code for the current active theme.
3. Located the theme.liquid file under the Layout section.
4. Copied the Global Site Tag (gtag.js) from the GA4 property settings.
5. Inserted the tracking code into the <head> section of the theme.liquid file.
6. Saved the changes to ensure Google Analytics tracking was applied to all store pages.
7. Confirmed that data was being collected by reviewing Real-time reports in Google Analytics.

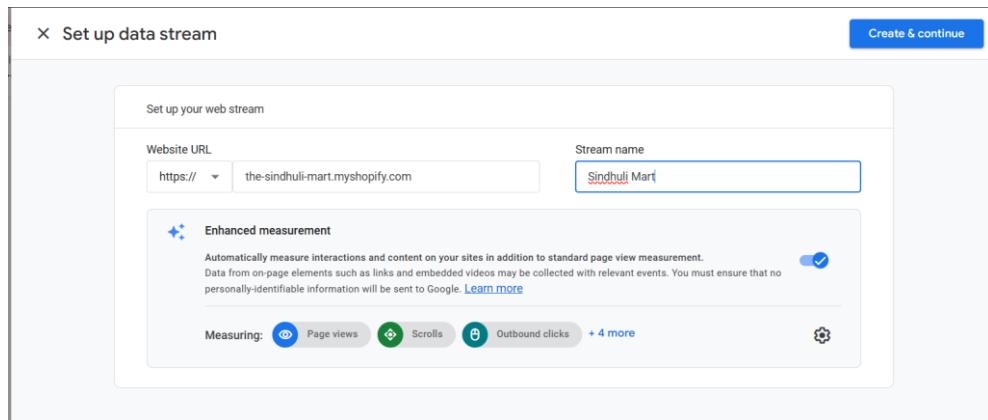


Figure 27: Data Stream Under Google Analytics

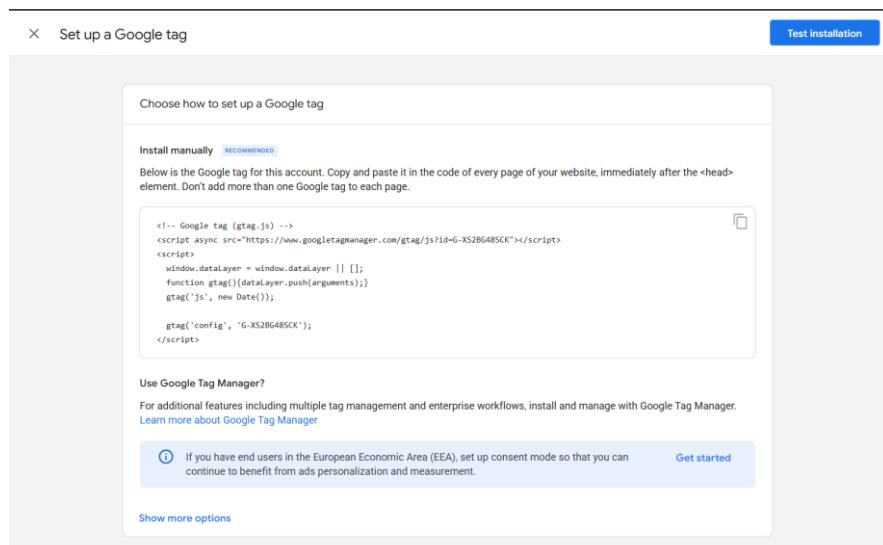
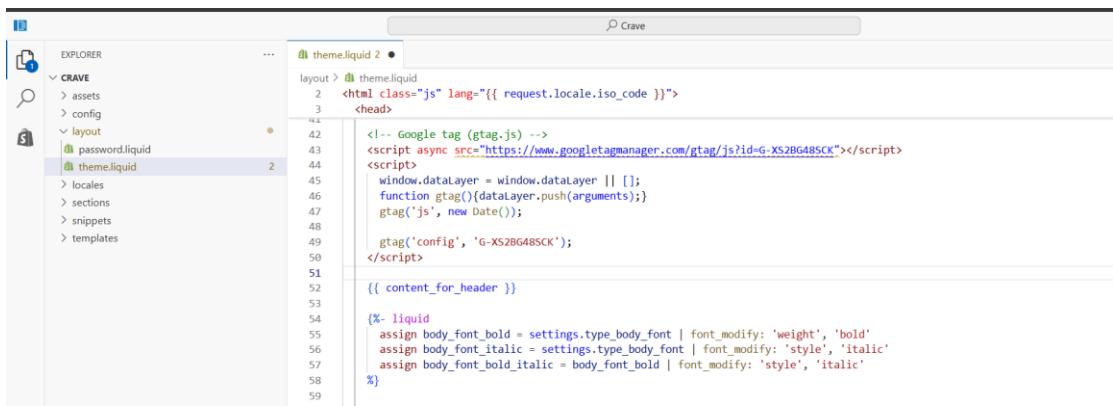


Figure 28: GA4 gtag setup



The screenshot shows a code editor interface with the following details:

- EXPLORER** sidebar on the left containing the following structure:
 - CRAVE
 - > assets
 - > config
 - < layout
 - password.liquid
 - theme.liquid**
 - > locales
 - > sections
 - > snippets
 - > templates
- theme.liquid 2** tab in the top right.
- Code Editor Content:**

```
layout > theme.liquid
  2  <html class="js" lang="{{ request.locale.iso_code }}>
  3    <head>
  4.
  42      <!-- Google tag (gtag.js) -->
  43      <script async src="https://www.googletagmanager.com/gtag/js?id=G-XS2BG485CK"></script>
  44      window.dataLayer = window.dataLayer || [];
  45      function gtag(){dataLayer.push(arguments);}
  46      gtag('js', new Date());
  47
  48      gtag('config', 'G-XS2BG485CK');
  49
  50    </head>
  51
  52
  53
  54  {{< liquid
  55    assign body_font_bold = settings.type_body_font | font_modify: 'weight', 'bold'
  56    assign body_font_italic = settings.type_body_font | font_modify: 'style', 'italic'
  57    assign body_font_bold_italic = body_font_bold | font_modify: 'style', 'italic'
  58  %}}
  59
```

Figure 29: Added code in the head section of theme.liquid