

Technical **Documentation**

**Employee ID:5454**

**2305**

**Document Version:1.0**

**Author: Nishan B Poojary**

**Swarna T**

**Razor Payment Method:**

* Install the extension on your Magento store using the Composer Package Manager.Composer provides a format for managing dependencies of PHP and requied libraries.
* Next go to your installation root directory of magento and execute the following command:

→ composer require razorpay/magento

→ bin/magento module:enable Razorpay\_Magento

* You can check if the installation was successful by executing the following command in the Magento root directory.

→ bin/magento module:status

* Then run the setup upgrade command and deploy the Razorpay module.

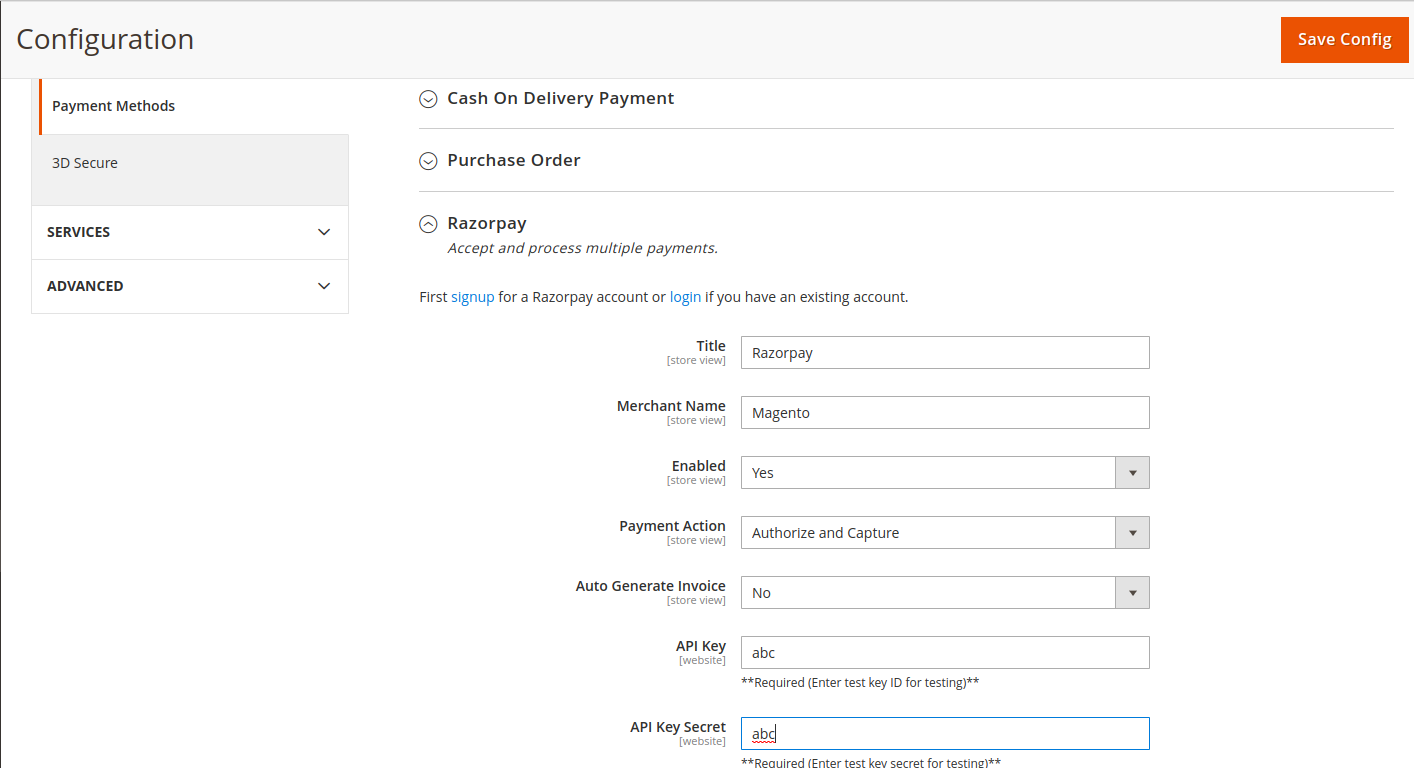
→ bin/magento setup:upgrade

→ bin/magento setup:static-content:deploy

Now your Razorpay extension will be added to the Payment Methods in the Sales section of Stores in the Magento Admin.

Next in the Payment Methods select Razorpay drop down and enable the store view as Yes, change the Payment Action to Authorize and Capture.

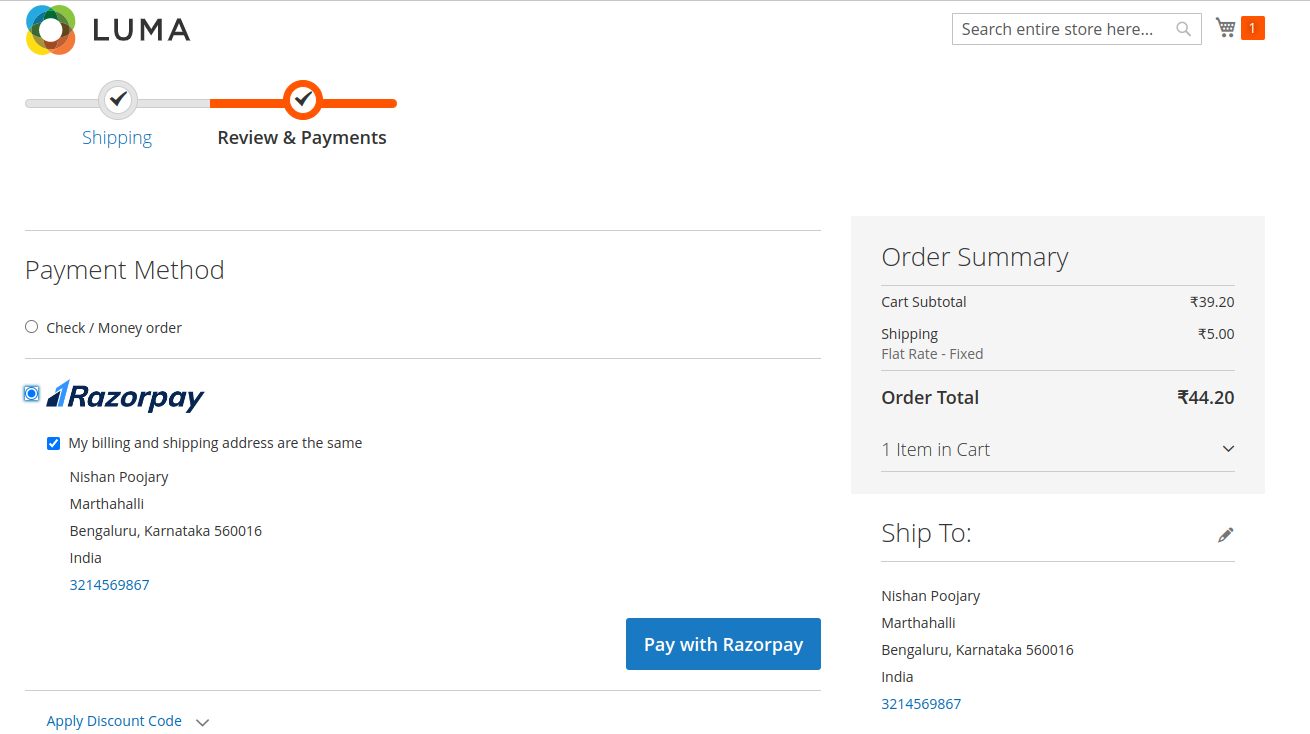
Generate the API Key and API Key Secret from the Razorpay account and enter the same, and click on Save Config.



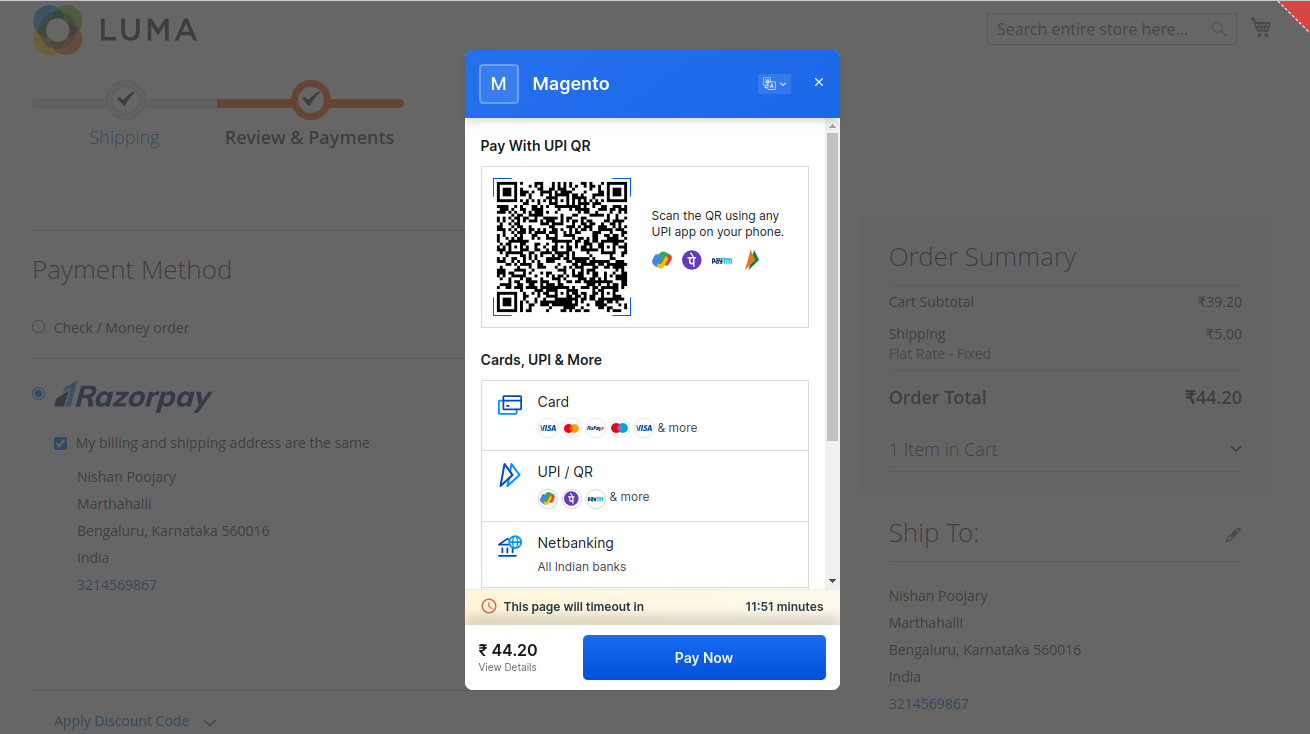
Now run the cache flush and chmod -R 777 var/ pub/ generated/ command in the terminal.

In the Luma after you add a product to cart and checkout and fill the shipping details, then you go to the Review and Payments page, where you will find the Razorpay in the Payment Methods.

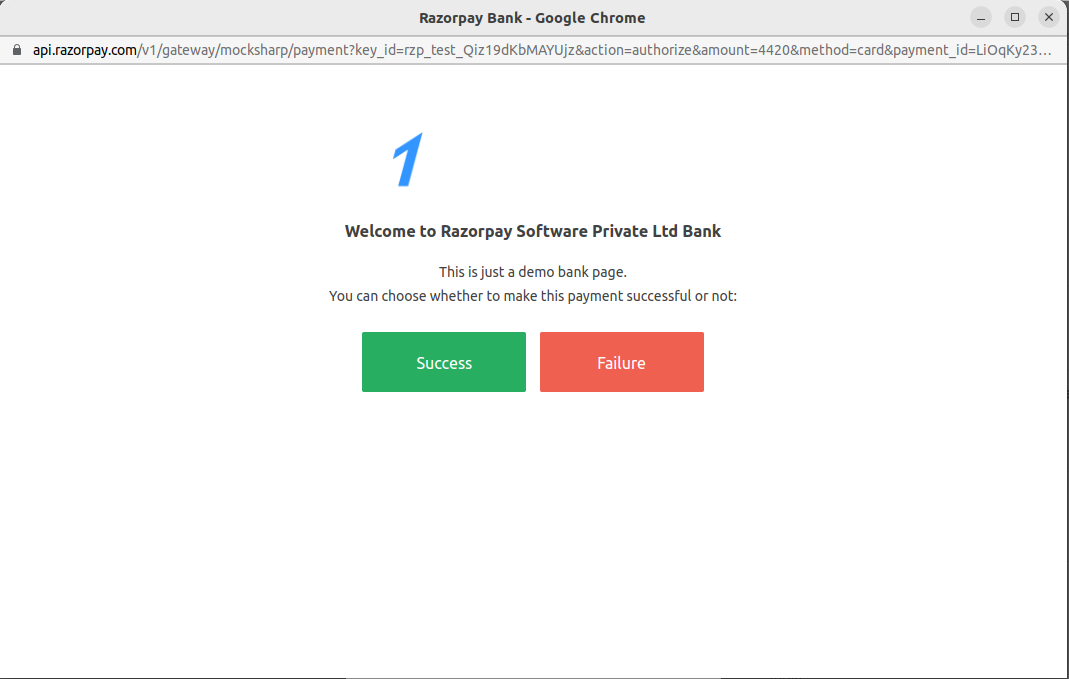
Select on the Razorpay method and click on the Pay with Razorpay checkbox.



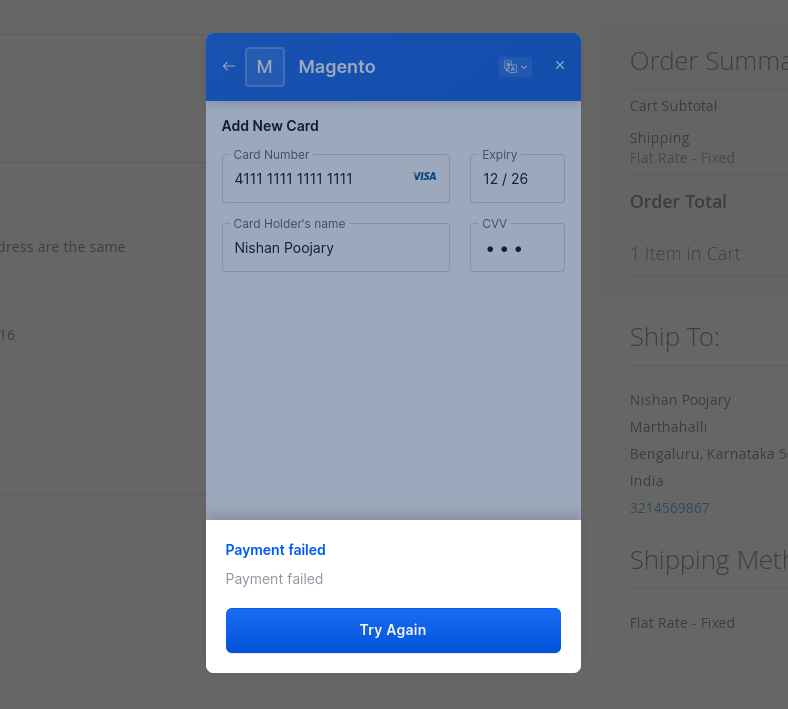
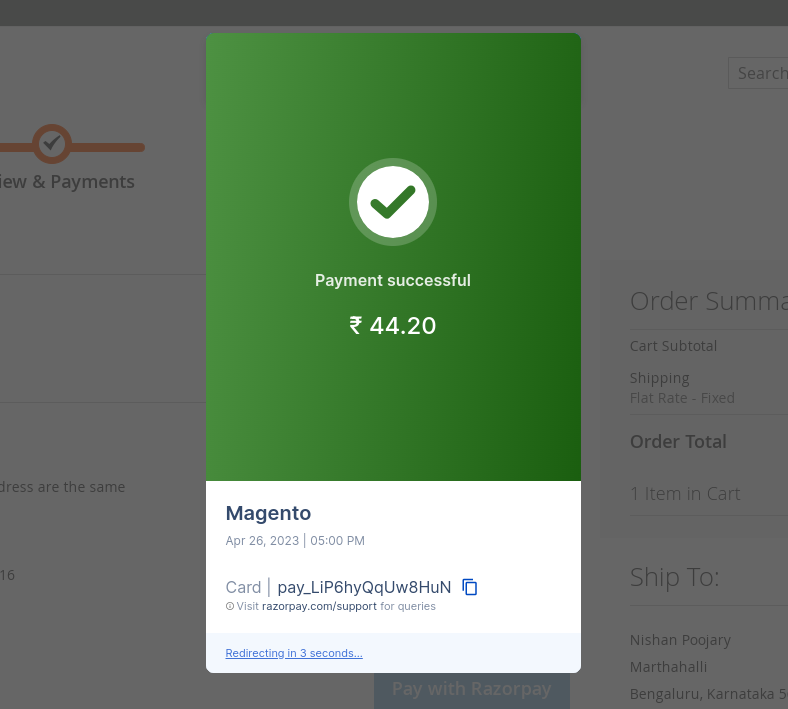
Razorpay Interface will appear to fill the card details, fill the card details and proceed to Pay Now.



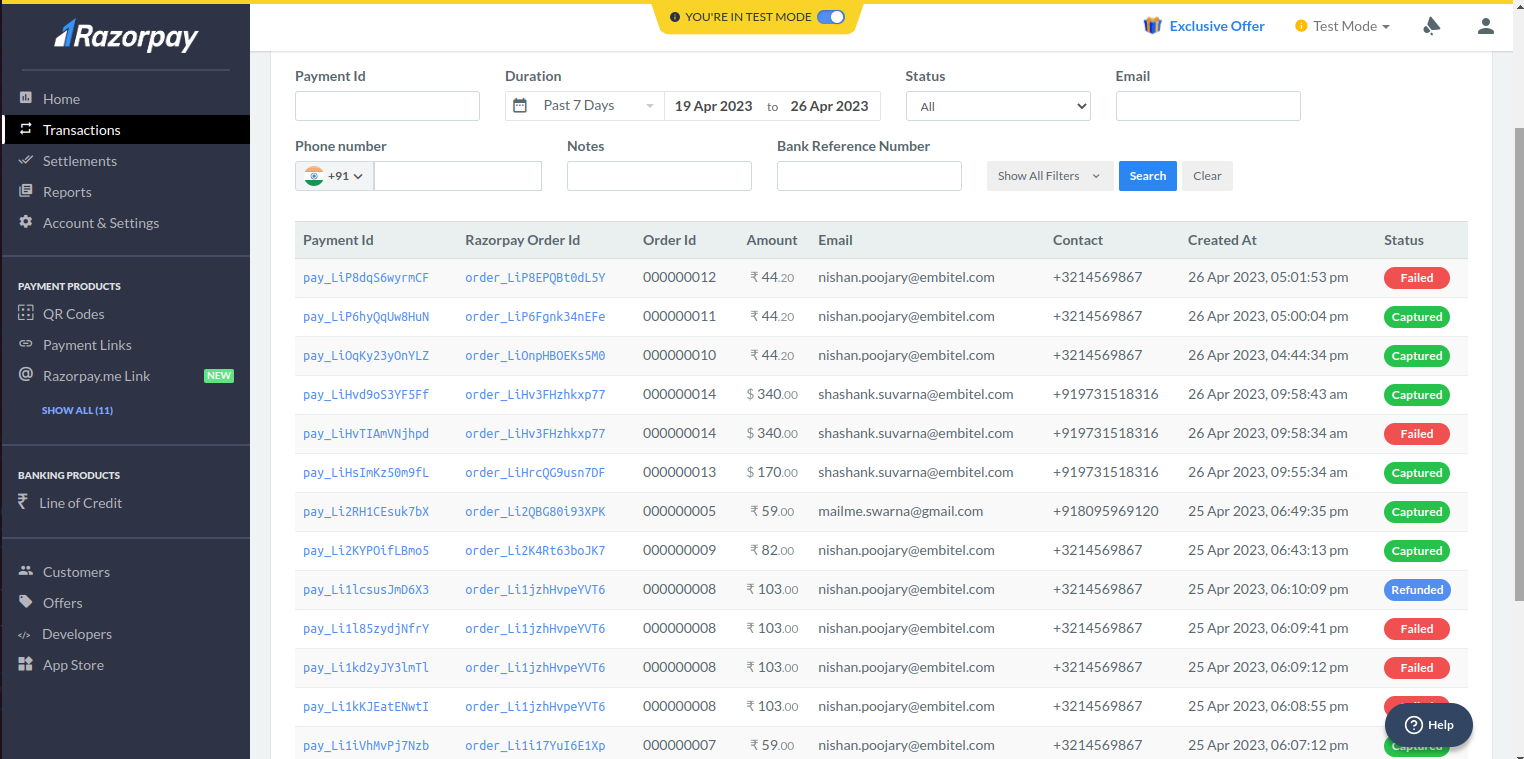
After proceeding to Pay Now, a confirmation page is displayed with two options Success and Failure.



If you select Success then the payment will be successful and stored in the transaction menu in Razorpay account, if you select Failure then it will redirect to Try Again page.



The Success and Failure payments will be reflected on the Transaction page in Razorpay.

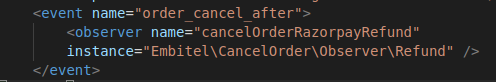
**4. Refund for Razorpay:**

### Steps for implementing Razorpay Refund Method:

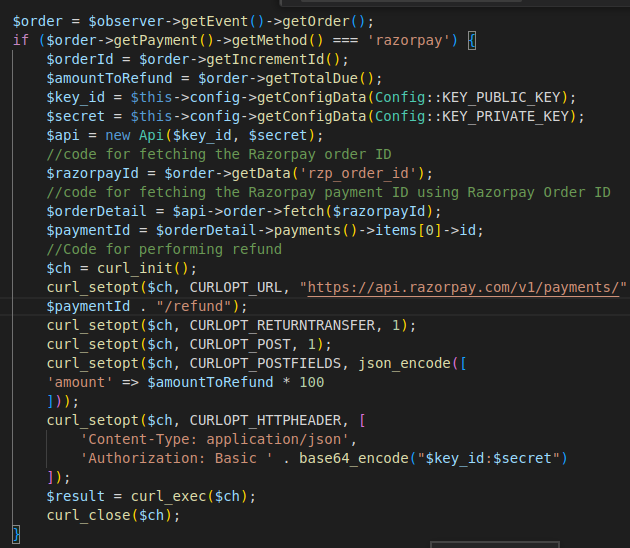
For implementing the refund of the Razorpay we created an observer in the path- app/code/Embitel/CancelOrder/Observer/Refund.php

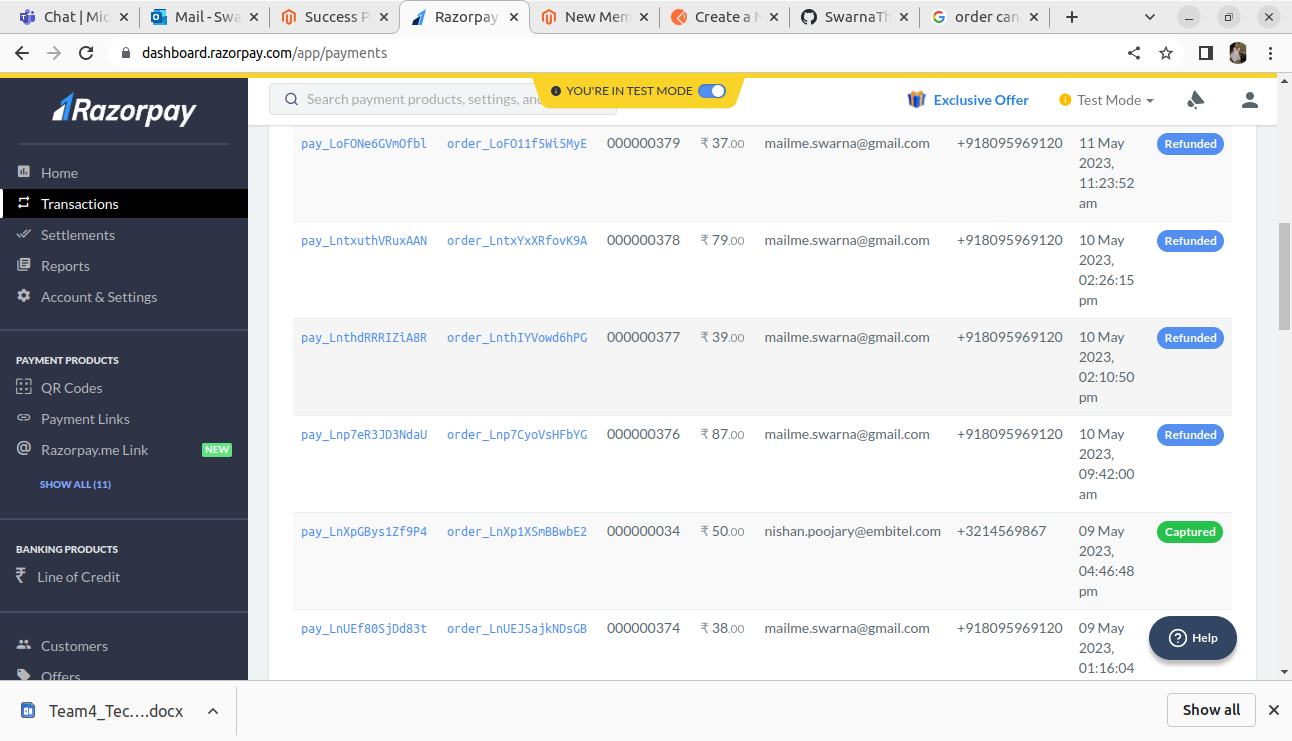
And for that observer we created a events.xml in the path-

app/code/Embitel/CancelOrder/etc/frontend/events.xml



In that observer with the order id we fetch the Razorpay payment ID with that by calling the api https://api.razorpay.com/v1/payments/{pay\_id}/refund





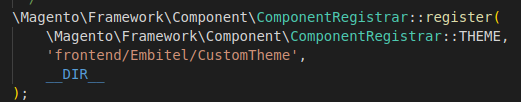
**Creating Theme:**

We have created custom Theme which inherit the properties of parent theme Luma. The path of the theme created is - /var/www/html/magento/app/design/frontend/Embitel/CustomTheme

Steps for creating custom Theme:

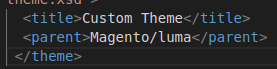
Created registration.php and composer.json for the theme and the path is –

/var/www/html/magento/app/design/frontend/Embitel/CustomTheme/registration.php



Then created theme.xml for both the theme and subtheme.

/var/www/html/magento/app/design/frontend/Embitel/CustomTheme/theme.xml



Then, enabled the custom created theme in magento admin panel.

**Override the Order Success page:**

The existing order success page shows only the order number for the particular order placed. We have to show the details of the order placed like Shipping address, Billing address, Payment Method, Payment Type and other product details like name, SKU, price and quantity. And also total, subtotal and shipping charges.

Steps for overriding existing phtml page –

First, we identified the phtml page which is in the path-

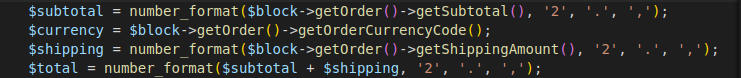
vendor/magento/module-inventory-in-store-pickup-frontend/view/frontend/templates/success.phtml

Then override the same in the custom theme created with the path- app/design/frontend/Embitel/CustomTheme/Magento\_InventoryInStorePickupFrontend/success.phtml

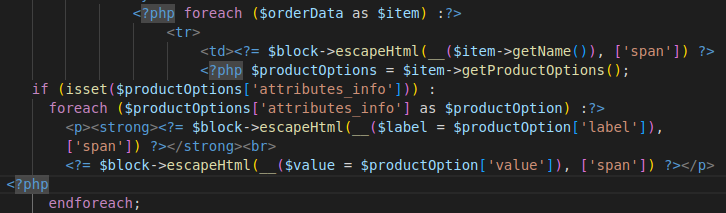
For fetching the order details like shipping address, billing address



## For fetching the currency code, subtotal and shipping charges



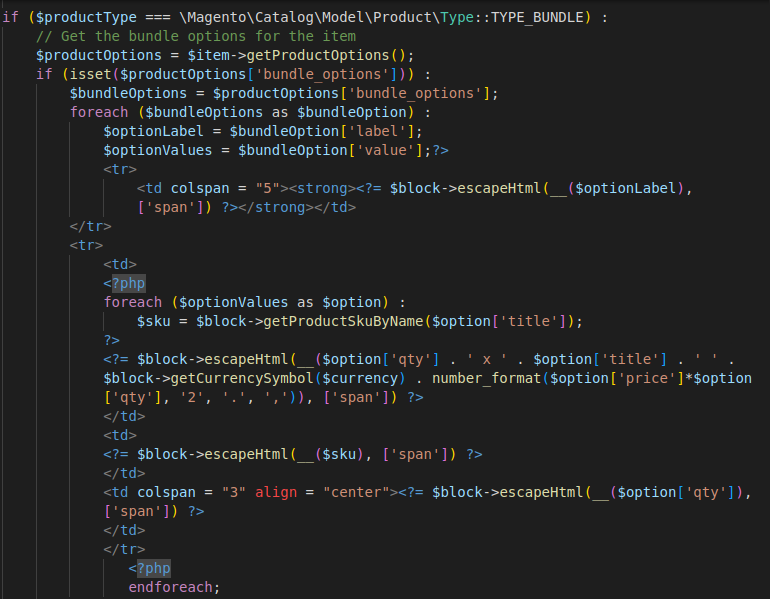
## For displaying configurable product details



## For displaying currency symbol



## For displaying bundle product details



To achieve that we created a module named OrderSuccess by creating registration.php, module.xml and enable the module by running the command bin/magento setup:upgrade

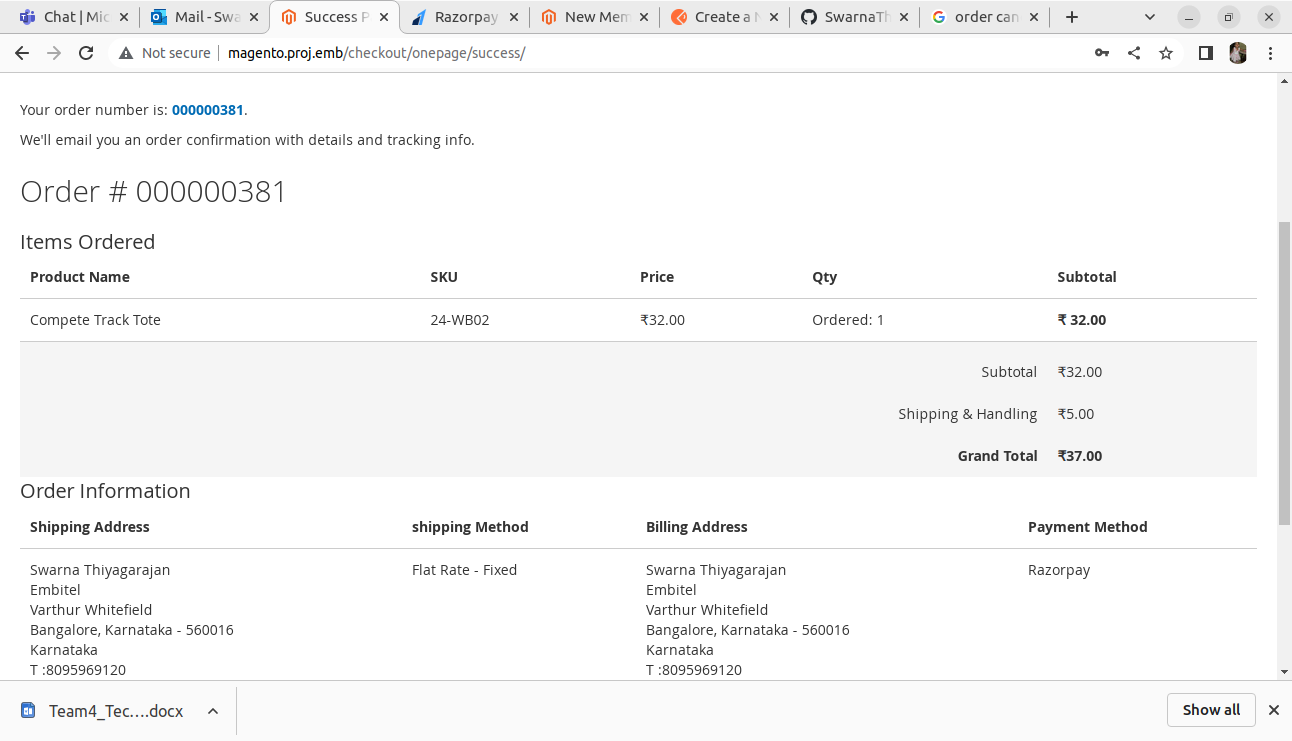
Inorder to have order details we created block file. in app/code/Embitel/OrderSuccess/Block/Success.php



To link this block to the override phtml file we created a di.xml in



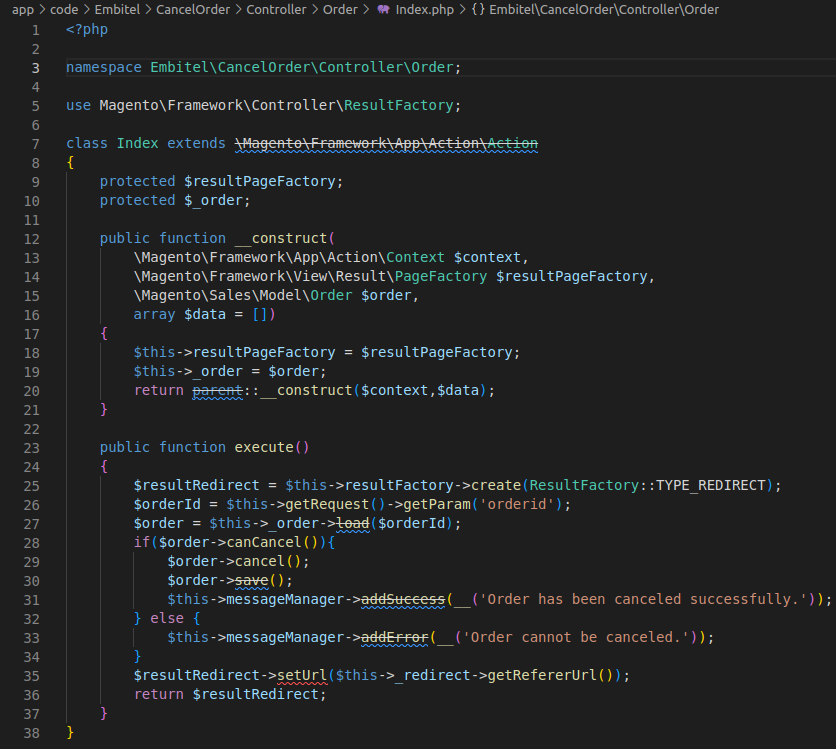
## Then we get the product details for the success.phtml file by calling the methods in the Success.php file



**Cancel Order from Frontend**

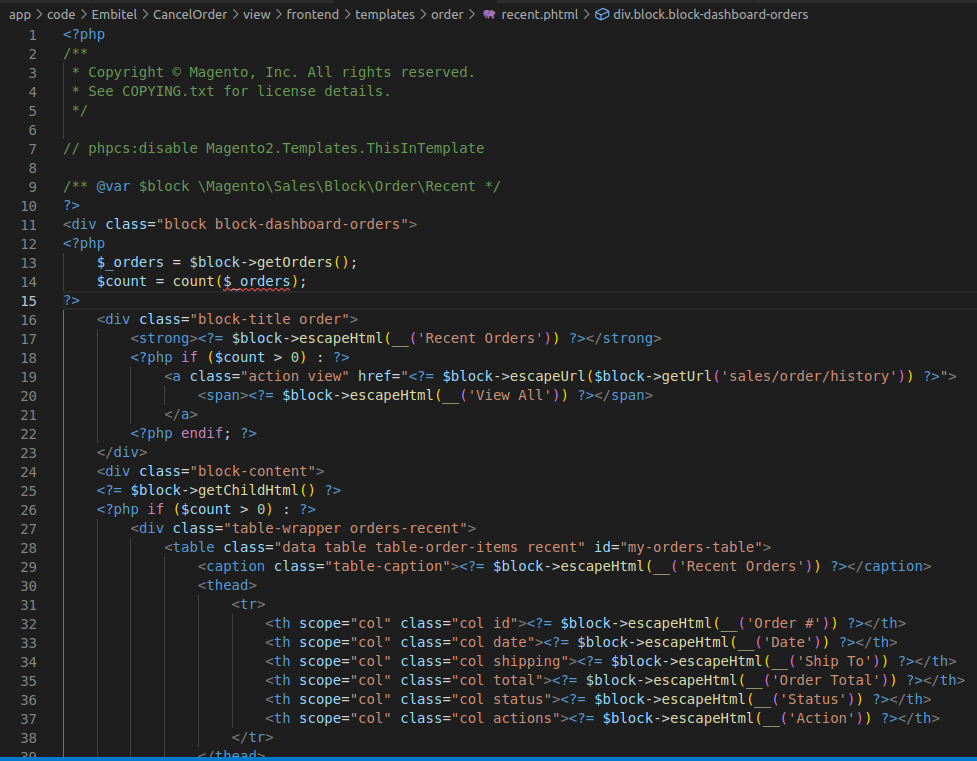
1. Create Index.php block file

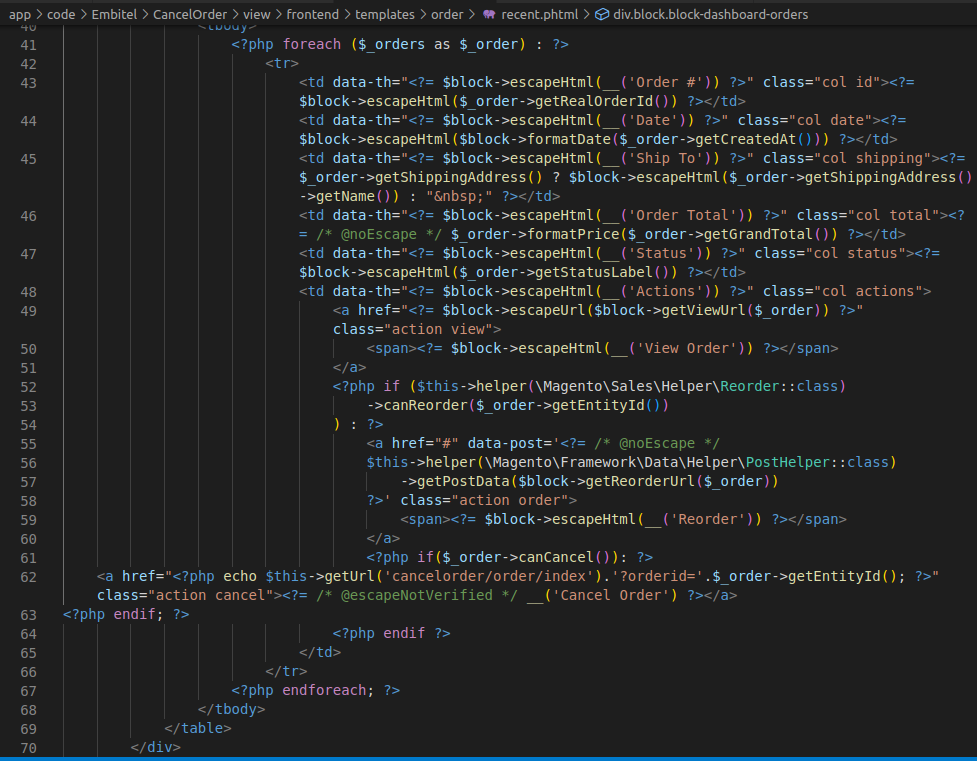
app/code/Embitel/CancelOrder/Controller/Order/Index.php



*2. Create history.phtml and recent.phtml template file to show the cancel button in My Account page*

Now override vendor/magento/module-sales/view/frontend/templates /order/recent.phtmlfile at /var/www/html/magento/app/design/frontend/Embitel/customtheme/Magento\_Sales/templates/order/recent.phtmland add below code in this file.





Override vendor/magento/module-sales/view/frontend/templates/order/ history.phtml file at /var/www/html/magento/app/design/frontend/Embitel/customtheme/Magento\_Sales/templates/order/history.phtml and add below code in this file.

