Introduction

Splitit Flex Fields Implementation

This project is created using HTML, CSS, JavaScript, and PHP. This project consists of three pages. the first page allows the buyer the select the product and displays the rounded-off monthly installment amount and first installment amount. On the second page, the buyer is asked to enter the billing details and complete the payment via Splitit Flex Fields. On the third page, the buyer is displayed with the transaction success response and a refund button to refund the transactions.

Installments Calculation Logic

Monthly Installment Amount Round-off

The monthly installment amount is rounded-off using the jQuery math.round() function. The rounded-off amount will be either higher or lower than the actual monthly amount split.

For example: if the actual monthly amount split is $16.66 it will be rounded off to the higher value $17 and if the actual monthly amount split is $83.33 then it is rounded off to a lower value $83.

First installment amount

First installment amount: the first installment amount is calculated using the below logic.

Total Amount - (rounded-off amount * Number of monthly installments)

For Example :
Total Amount = 100 .
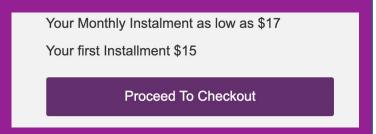Monthly Rounded-off Amount = $17
No. of installments = 6

Calculation : 100 - (17 * 5) = $15

Product and Checkout Page Content

Messaging Placement Element and Checkout Button

Upstream messaging placement element is designed using HTML and CSS. Monthly Installment amount and First installment amount change based on the product selected by the user. Values on the element changes with the help of jQuery. When the checkout button is clicked the user is redirected to the checkout page and passes the total amount and the first installment amount to the checkout page.

Your Monthly Instalment as low as $17

Your first Installment $15

Proceed To Checkout

Billing Details and Pay with Splitit

Billing details are hardcoded in this example. When the checkout page loaded the following Splitit APIs are triggered.
1) Login – to get the session ID
2) Initiate – Initiate API is triggered with 'autoCapture' set to "false" and explicitly pass 'amount' & 'firstInstallmentAmount' in the Plan Data Object to adjust the monthly installments.

When the user clicks on the "Pay with Splitit" button, Flexi fields are initialized using the public token received from the "Initiate" API response. Card details fields are displayed along with installment schedule from Splitit to the user to complete the transaction.

When the "Pay" button is clicked payment is processed by Spliitit and onSucess() / onError() function user is redirected to the appropriate response page.
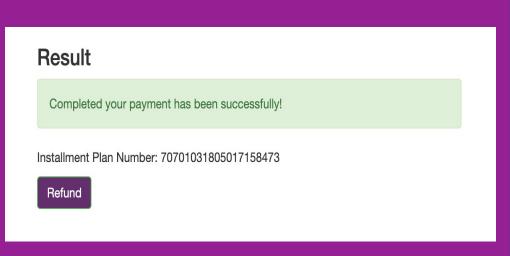
Billing Address
Full Name
John M. Doe

Email
john@example.com

Address
542 W. 15th Street

City
New York

State
NY

Zip
10001

Payment
Accepted Cards

Grant Total: $100
Your Monthly Instalment as low as $17
Your first Installment $15

Pay with Splitit

Card Number

MM/YY   CVV

6 payments, $15 + $17 per month

Payment schedule ⌄   How Splitit Works

☐ I accept Splitit Terms & Conditions and Privacy Policy

Pay

Payment Confirmation Page Content

Success Page

Result : Transaction success message is displayed to the user along with Installment Plan Number.

Refund : Login API is triggered to get the session and ID and initiate the refund API using the session ID and Installment Plan Number to refund the transaction.

Result

Completed your payment has been successfully!

Installment Plan Number: 70701031805017158473

Refund

My Integration Experience

Worked Well

Installment Amount Customization : Splitit **i**ntegration allows the merchant to customize the monthly installment amount by adjusting the first payment.

Customer Payment Experience : Splitit flex fields allow the merchant to keep the customer within the checkout page and complete the transaction and provide a better customer experience and faster checkout.

Developer Documentation: Easy and simple developer documentation helps the merchants to easily integrate with Splitit and go-live quickly.

It would have been good

Issuer Information: It would have been good if the card issuer information was provided in the "Get" response. That would have helped merchants' marketing team to quickly identify the percentage of card splits from each bank. It is a slightly tedious activity to manually search for the issuer details with the BIN number provided in the "Get" response currently.

Postman Collection: it would have been good if the Postman collections were available to download to quickly test with Postman

Schedule Information: It would have been good if the schedule information was available in the "Get" response. Currently, if the merchant has to get the schedule details they have to make an additional API call (GetSchedules) to get the complete details.

Thank You !