



# Payments UX Best Practices

## How to Build Top-Tier Flexible Payment Flows

# Table of Contents

Overview	3
Objectives	3
Payment Methods	4
Checkout Flow Fields	4
Card Number Field Best Practices	5
Challenges	5
Best Practices	5
Expiration Date Field Best Practices	6
Challenges	6
Best Practices	7
Security Code Field Best Practices	7
Challenges	7
Best Practices	8
Button Best Practices	8
Challenges	8
Best Practices	9
Putting It All Together	9
Errors	9
Style and Layout Variations	10
Security	10
One Solution for Flexibility and Compliance	10

# Overview

In crafting this guide to best practices for designing a payments UX, we embarked on an extensive exploration of the digital payments landscape, delving deep into an array of payment and non-payment sources. Our mission? To distill the essence of what makes a checkout experience truly outstanding and to unlock the secrets of seamless payment processing. These insights are not just academic—we've created a practical toolkit designed for merchants and digital platforms that are eager to reduce customer drop-off and supercharge their conversion metrics.

Consider this startling statistic from the Baymard Institute's 2023 research: poor user experience can account for up to 17% of all abandoned online shopping carts. Additionally, a 2022 study by Forrester Research amplifies the urgency of this issue, finding that poor UX can cost businesses up to \$2.8 trillion per year in lost sales due to checkout abandonment.

The foundation for this guide is in real-world success stories and proven strategies. We've gleaned insights from leaders in the digital payments world—Stripe, Adyen, Klarna, PayPal, and Shopify—to understand the “magic” behind the user experiences they provide to their customers. We've dissected the strategies of top-performing e-commerce sites to uncover what elevates them above the rest. And, we've identified the consistent patterns and industry-wide best practices that are the keys to simplifying payment acceptance and boosting conversion rates.

Transform your understanding of payments UX and get on the path to checkout flow success.

## Objectives

Getting back to the basics, payment flows should strive to meet several objectives that improve user experience, payment processing, and conversion. To build a flexible but optimized payment experience, your checkout should offer:

- **Simplicity & Familiarity:** Use intuitive interfaces, leveraging recognized UI patterns for user comfort and quick adoption.
- **Speed & Efficiency:** Customers want quick and seamless checkouts with minimal steps and few distractions.

- **Security:** Build trust and confidence with users and ensure their personal information is kept secure.
- **Flexibility:** Payment flows aren't one-size-fits-all. Customers expect their preferred payment methods will be accepted when paying.
- **Transparency:** Build transparent, honest flows that give a clear price and path for the customer.

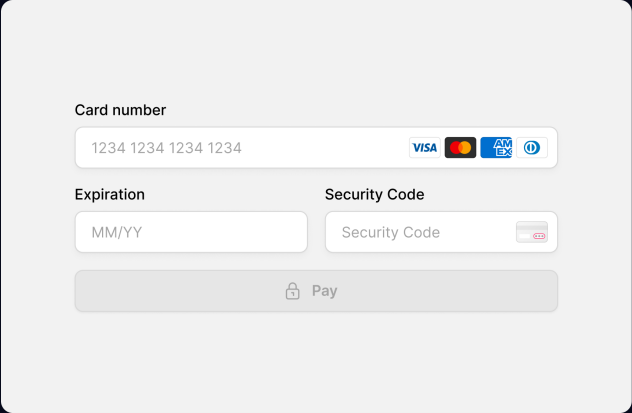
This guide provides actionable tips that will help all merchants meet these objectives.

## Payment Methods

The first potential point of friction in the payment flow is in the payment method choices. The process a user must take in selecting a payment method can significantly impact overall satisfaction with the checkout flow.

A few tips to consider when building payment method selection into a checkout flow:

- **Payment diversity matters:** Offer a variety of payment options, and be sure to vary these options based on user location to ensure you offer the user-preferred methods.
- **Make selection easy:** The process of selecting a method should be clear, easy to understand, and easy to move forward with.
- **Continuous evolution:** As new payment methods come into mass acceptance, checkout flows should offer these to customers.



The image shows a payment form with the following fields and elements:

- Card number:** A text input field containing "1234 1234 1234 1234". To the right of the field are logos for VISA, Mastercard, AMEX, and Discover.
- Expiration:** A text input field containing "MM/YY".
- Security Code:** A text input field containing "Security Code". To the right of the field is a small icon of a credit card.
- Pay Button:** A wide, light gray button at the bottom with a lock icon and the text "Pay".

## Checkout Flow Fields

Checkout flows are often more complex than meets the eye, with many moving pieces and elements. The following fields are commonly included in checkout flows, and will be the focus of this guide:

- Card Number
- Expiration Date
- Security Code
- Submit Payment Button

Additionally, merchants should consider the form's responsiveness and styling options that can best support a successful customer payment.

In the following sections, we will break down best practices by field type.

## Card Number Field Best Practices

The card number is often one of the first fields presented, and it can set the tone for the remainder of the checkout process.

### Challenges

- ✗ **Card number input difficulty:** Users struggle to enter long card numbers, leading to potential checkout abandonment.
- ✗ **Common input behavior:** Many users type card numbers in 4-digit blocks and fear card validation errors.
- ✗ **Not knowing:** Users struggle to know if their card is valid without visual cues that let them see progress as they move through the process.

### Best Practices

#### Visual Cues

- ✓ **Reinforce security with visuals:** Add visual cues to the card input fields to boost the perceived security. A simple addition of a lock icon can boost a user's trust in the checkout process.
- ✓ **Incorporate card icons:** Position card icons inside the input field for easy identification, and highlight the card type as a user inputs the card number.
- ✓ **Enhance user verification:** Display the card numbers (unmasked) so that users can verify their input prior to submission.
- ✓ **Match physical card format:** Ensure the input format closely resembles the physical card's look and that the field is restricted to number entry only.
- ✓ **Space between number blocks:** If there is a blank space between every four numbers, create that visual space in the input field for ease of entering numbers.

## User Experience Improvements

- ✓ **Automatic card identification:** Use Luhn validation to determine if card numbers are potentially valid. Even better, identify the Brand based on the BIN entered.
- ✓ **Simplify data transfer:** Allow copy and paste functionality in the fields for convenient data entry.
- ✓ **Compatibility with card managers:** Ensure smooth integration with password and card managers.

Ensure input format closely resembles the physical card's look.

Show card numbers (unmasked) allowing users to verify their input.

Allow users to space out numbers, but restrict input to only numerical values.

Placeholder with the proper format.

Showing payment options on resting state and right-aligned.

Card number

1234 1234 1234 1234

VISA

Expiration

MM/YY

Security Code

Security Code

Pay

## Expiration Date Field Best Practices

A simple date field can go a long way to making the checkout experience smooth, or not.

### Challenges

- ✗ **Mismatched expiry dates:** A mismatch between site's date fields and physical card formats cause errors, hindering checkout.
- ✗ **Proper format:** Aligning date fields with card formats improves UX, but 72% of sites are still not adhering.
- ✗ **Difficulty using drop-downs:** Drop-down menus for date selection slow down input and frustrate users.

## Best Practices

- ✓ **Card formatting consistency:** Use the same format as the physical card for input fields and as placeholders (MM/YY).
- ✓ **Automated slash insertion:** After the user enters the month, auto-insert a slash before the year input begins.
- ✓ **Insert leading zero:** If a user omits the leading 0 for single-digit months, auto-prefix the zero for simplicity.
- ✓ **Numeric input restriction:** Restrict the input fields to accept only numerical values for date entry.
- ✓ **Front-end validation:** Validate the date field before allowing form submission.
- ✓ **Direct typing:** Let users type expiry dates directly to avoid the hassle of drop-down menus.

The diagram shows a payment form with the following fields:

- Card number:** A field containing the placeholder "1234 1234 1234 1234" and logos for VISA, Mastercard, AMEX, and Discover.
- Expiration:** A field containing the placeholder "MM/YY". A blue line connects this field to a callout box below the form.
- Security Code:** A field containing the placeholder "Security Code" and a small card icon.
- Pay button:** A button with a lock icon and the text "Pay".

Below the form, there are four callout boxes with the following text:

- Use the same format as the physical card for input fields and as a placeholder (MM/YY).
- Restrict input fields to accept only numerical values for date entry.
- After the user types the month, auto-insert a slash.
- Restrict input fields to accept only numerical values for date entry.

## Security Code Field Best Practices

While in theory this is one of the simplest fields, the security code field comes with significant variability across various checkout flows. This can make entry and understanding challenging for users.

### Challenges

- ✗ **Naming variations:** Different card brands use distinct names: MasterCard: "CVC2", Visa: "CVV2", Discover: "CID", American Express: "CID" or "unique card code", Debit Card: "CSC" or "card security code".

- ✗ **Inconsistencies:** Some checkout flows also use terms like "card verification data", "card verification number", and more.
- ✗ **Code differences:** American Express uses four digits found on the front of the card while all others use the three digits from the back of the card.

## Best Practices

- ✓ **Universal term adoption:** Opt for "security code" as the field label to avoid confusion between CVC, CVV, and other terms.
- ✓ **Customize by card brand:** Recognize the card type from the card number field and use this to ensure that if an AMEX card is used, four digits are allowed, while other brands only allow three.
- ✓ **Tooltip guidance:** Display a tooltip tailored to the card brand to help users understand where to find the security code on their card.

The image shows a payment form with the following fields: "Card number" (containing "1234 1234 1234 1234"), "Expiration" (containing "MM/YY"), and "Security Code" (containing "Security Code"). To the right of the "Card number" field is a card brand selector showing logos for VISA, Mastercard, AMEX, and Discover. Below the "Security Code" field is a "Pay" button with a lock icon. A tooltip points to the "Security Code" field with the text: "Security Code changes according to Card Branch." The form is set against a dark blue background.

## Button Best Practices

Button submission should be simple, but informative, as the last touchpoint prior to purchase.

## Challenges

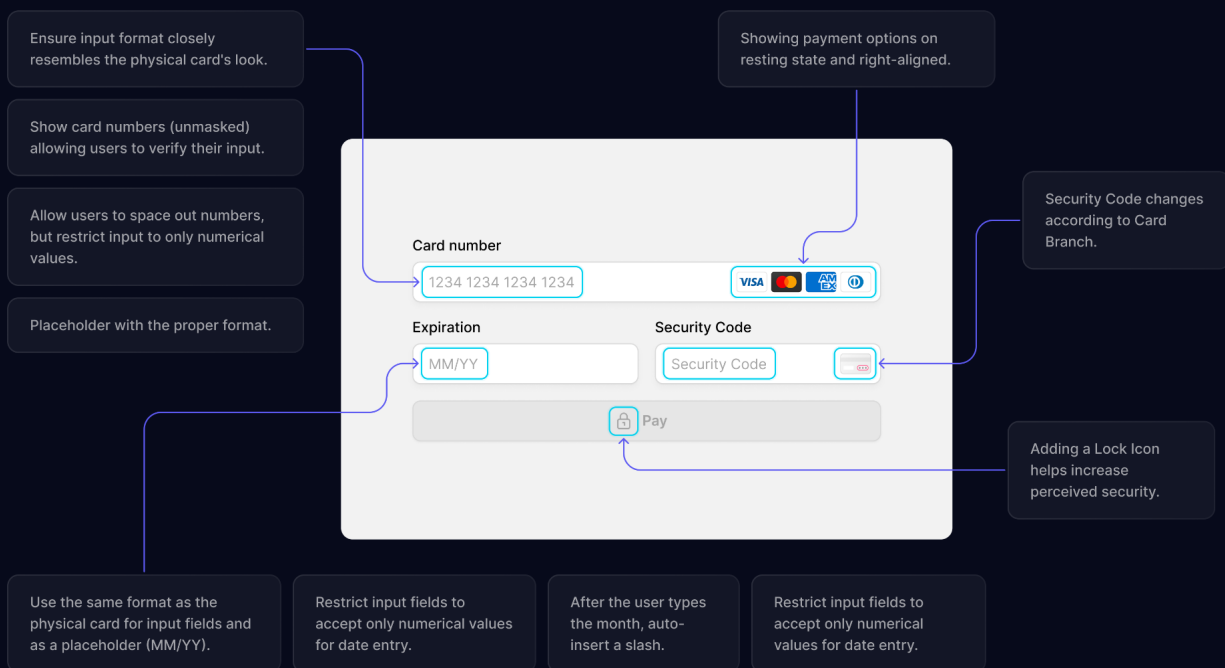
- ✗ **Lets users interact incorrectly:** When a submission button allows users to engage with it before all the fields are completed, it can give them improper hope that they've completed all required fields.
- ✗ **No feedback:** Users may find frustration if the button doesn't show an on-click event or doesn't share progress when processing.



## Best Practices

- ✓ **Grayed default state:** Gray out the button into an inactive state and only allow interaction when all required fields are completed and validated.
- ✓ **Confirm security:** Give the user reassurance by putting in visual cues that the form is secure through a lock icon or other visual element.
- ✓ **Add the payment amount:** Add the purchase amount to the button as secondary confirmation to the user of what the actual charged amount will be.

## Putting It All Together



## Errors

Should the user encounter an error when using the form, you want to provide helpful errors. If an error occurs, share what the error was, where it was, and how to fix it.

The diagram shows a payment form with three input fields: Card number, Expiration, and Security Code. The Card number field contains '4242 4242 4242 4' and has an error message 'Your card number is incomplete.' below it. The Expiration field contains '03/1' and has an error message 'Your card's expiration date is incomplete.' below it. The Security Code field contains '84' and has an error message 'Your card's security code is incomplete.' below it. A 'Pay' button is at the bottom. Callouts indicate that error messages appear at the bottom of the field and that icons change accordingly based on the card brand.

## Style and Layout Variations

Until now, the examples have included a split field view with the field label above the field. However, when low on space, a merged option meeting all best practices may be used. The field labels would be hidden and fields would sit side-by-side.

The image compares two payment form layouts: Split and Merged. The Split layout shows three separate input fields for Card number, Expiration, and Security Code, each with its own label. The Merged layout shows a single 'Card Information' section with three side-by-side input fields for Card number, Expiration, and Security Code. Both layouts include a 'Pay \$500.00' button.

## Security

While data collection security has been assumed throughout this guide, it shouldn't be overlooked. Handling and processing cardholder data directly will put your systems into PCI compliance scope. If you choose this route, it is paramount that your systems handle this sensitive data properly to meet all 12 PCI DSS Requirements.

## One Solution for Flexibility and Compliance

Choosing a secure solution like [Basis Theory Elements](#) that you can embed directly on your website and style to your specifications can cut down checkout flow go-live time significantly. Not only can you flexibly build the payment flow to your specifications, you

can also completely remove your frontend applications from PCI compliance scope, saving your team time and money.

Every example in this guide was built with Basis Theory Elements.

If you're ready to create a secure, seamless payment flow and level up your payment operations, [contact us](#) today.