



PRIMATIF BRAND STYLE GUIDE

A guide to our visual identity.







This style guide is the foundation for creating all brand communications. It ensures a consistent, professional, and innovative identity across all mediums. These are not strict rules but a framework to empower creativity while maintaining the reliability our brand stands for. Use this as a guide to make informed design decisions.

1. COLOR PALETTE

Our color palette balances professionalism with innovation. The primary colors are bold and confident, while functional tones ensure clarity and readability. Semantic colors provide intuitive user feedback without being distracting.





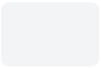

Brand Colors

The brand colors form the core of our visual identity. They are used for key highlights, important information, and to create a dynamic and recognizable presence.

COLOR	NAME	TOKEN	VALUE
	Primatif Red	PrimaryAccent	#E31937
	Primatif Red Darker	PrimaryAccentDarker	#5C0411
	Primatif Sky Blue	SecondaryAccent	#53C8ED
	Sky Blue Light	AccentSkyBlue	#A0DFF2
	Deep Blue	AccentDeepBlue	#0080A4
	Green Screen Green	TertiaryAccent	#08FF08


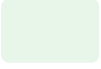

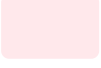

Functional Tones

The functional tones provide the neutral foundation for our documents, ensuring text is readable and the layout is clean and structured.

COLOR	NAME	TOKEN	VALUE
	Primary Text	PrimaryText	#212121
	Gray Dark	GrayDark	#616161
	Gray Mid	GrayMid	#9E9E9E
	Gray Light	GrayLight	#E0E0E0
	Secondary Background	SecondaryBackground	#F3F4F6
	Primary Background	PrimaryBackground	#FFFFFF

Semantic Colors

Recommended for use in forms, alerts, or any element meant to communicate a state. Color should never be the only means used to convey state information; it must always be accompanied by text or an icon.

COLOR	NAME	TOKEN	VALUE
	Success Green	SemanticSuccessGreen	#2E7D32
	Success Green Light	SemanticSuccessGreenLight	#E8F5E9
	Error Red	ErrorRed	#C1152E
	Error Red Light	ErrorRedLight	#FFE8EB
	Warning Yellow	WarningYellow	#FFC300

COLOR	NAME	TOKEN	VALUE
	Warning Yellow Light	WarningYellowLight	#FFF8E1
	Info Blue	InfoBlue	#53C8ED
	Info Blue Light	InfoBlueLight	#D1EEF9
	Disabled Gray	DisabledGray	#BDBDBD

2. TYPOGRAPHY

Typography is key to our brand voice. Bebas Neue provides a strong, modern feel for headlines, while Roboto ensures body text is highly legible and professional. Lato is used for subtitles and intro text to provide a clean, accessible feel that complements the primary fonts.

We utilize a combination of fonts to establish a clear visual hierarchy and maintain brand consistency.

Primary & Logo Font

PRIMATIF

Bebas Neue: Used for the "PRIMATIF" logo and major section headers.

```
font-family: 'Bebas Neue, sans-serif';
```

Headings

Heading 1 - Roboto Bold

Heading 2 - Roboto Bold

Heading 3 - Roboto Bold

```
font-family: 'Roboto, sans-serif'; font-weight: 700;
```

Body & Subtitle Copy

This is Lato, used for subtitles and introductory text.

This is Roboto, our primary font for all paragraph text, ensuring readability and a clean, professional appearance.

```
font-family: 'Lato, sans-serif'; /* For subtitles */
font-family: 'Roboto, sans-serif'; /* For body text */
```

Text Color Usage

To ensure readability and accessibility, use the following colors for text.

- **Primary Text** (`PrimaryText`) - Use for all headings and body copy.
- **Secondary Text** (`GrayDark`) - Use for secondary information or less important details.
- **Tertiary/Hint Text** (`GrayMid`) - Use for captions, hints, or placeholder text.
- **Disabled Text** (`DisabledGray`) - Use for disabled UI elements.

Links

- [This is an example link.](#)

Links use `PrimaryAccent` and maintain the same color on hover.

```
<p style={{color: brandColors.PrimaryText}}>Primary Text</p>
<p style={{color: brandColors.GrayDark}}>Secondary Text</p>
<p style={{color: brandColors.GrayMid}}>Tertiary/Hint Text</p>
<p style={{color: brandColors.DisabledGray}}>Disabled Text</p>
<a href="#" style={{color: brandColors.PrimaryAccent,
textDecoration: 'underline'}}>Example Link</a>
```

3. HEADERS & FOOTERS

These components bookend our documents, providing a consistent brand frame. The header is clean and professional, while the footer offers essential contact information in a compact, unobtrusive manner.

Headers and Footers provide consistent branding and navigation across all documents.

Standard Header

PRIMATIF

Document Title

```
<div className="p-4 flex justify-between items-center rounded-
lg" style={{border: `1px solid ${brandColors.GrayLight}`}}>
  <div className="font-bold text-2xl" style={{
...typography.headerLogo, color: brandColors.PrimaryText
}}>PRIMATIF</div>
  <div className="text-sm" style={{color:
brandColors.GrayMid}}>Document Title</div>
</div>
```

Standard Footer

PRIMATIF

© 2025 Primatif | hello@primatif.com

```
<div className="p-4 text-center rounded-lg" style=
{{backgroundColor: brandColors.GrayDark, color:
brandColors.PrimaryBackground}}>
  <div className="font-bold text-lg" style={{
...typography.footerLogo }}>PRIMATIF</div>
  <p className="text-xs mt-1" style={{color:
brandColors.GrayLight}}>© 2025 Primatif | hello@primatif.com</p>
</div>
```

4. BUTTONS

Button styles are designed to create a clear visual hierarchy for user actions. The primary button should be used for the most important action on a page, while secondary and destructive buttons offer clear alternatives.

Buttons are used for primary actions, secondary options, and destructive operations.

Primary Action

Used for the main call to action.

```
<button className="px-5 py-2 rounded-lg text-white font-semibold
shadow-md" style={{backgroundColor:
```

```
brandColors.SecondaryAccent}}>Primary Action</button>
```

Secondary Action

Used for alternative, less critical actions.

```
<button className="px-5 py-2 rounded-lg font-semibold border" style={{color: brandColors.PrimaryText, borderColor: brandColors.GrayMid}}>Secondary Action</button>
```

Destructive Action

Used for actions that cannot be undone.

```
<button className="px-5 py-2 rounded-lg text-white font-semibold shadow-md" style={{backgroundColor: brandColors.PrimaryAccent}}>Destructive Action</button>
```

5. PATTERNS & BACKGROUNDS

Subtle patterns can add texture and visual interest to otherwise plain backgrounds. The different sizes allow for flexibility, from a fine texture on a small card to a more noticeable pattern on a large hero section.

A subtle checkered pattern can be used as an overlay on light backgrounds to add texture. It is available in multiple sizes for different visual effects.

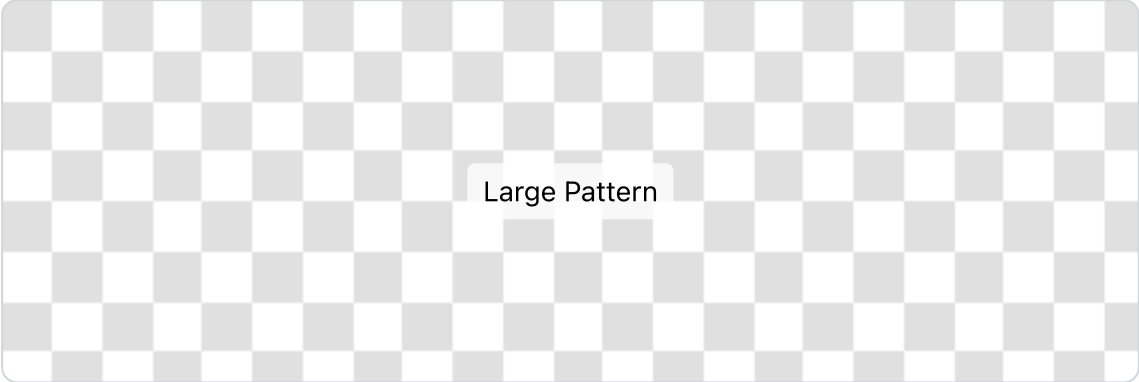
Small Pattern

```
<div className="h-48 ..." style={{ backgroundImage: checkeredPatternSmall, backgroundSize: '10px 10px' }}>...</div>
```





```
<div className="h-48 ..." style={{ backgroundImage:
checkedPatternMedium, backgroundSize: '25px 25px' }}>...</div>
```



```
<div className="h-48 ..." style={{ backgroundImage:
checkedPatternLarge, backgroundSize: '50px 50px' }}>...</div>
```

6. TABLE FORMATTING

Our table style prioritizes readability. A strong header color provides a clear starting point, while subtle row highlighting guides the eye. Semantic colors can be used within cells to draw attention to specific data points.

Tables are powerful tools for presenting structured data. Use color and formatting to create visual hierarchy and convey information clearly. Rows or cells can be styled to highlight status or importance.

ITEM	STATUS	OWNER	DUE DATE
Initial Project Setup	Complete	Tech Lead	2025-06-15
API Key Provisioning	In Progress	Client IT	2025-06-22
Database Credentials Update	Blocked	Tech Lead	2025-06-20

ITEM	STATUS	OWNER	DUE DATE
User Interface Mockups	Pending Review	Design Team	2025-06-25

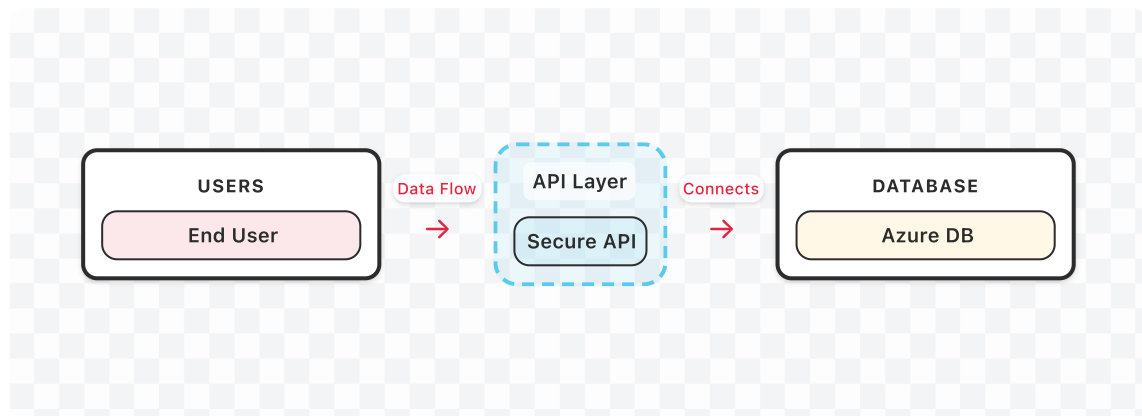
```
<div className="overflow-x-auto rounded-lg border" style={{
borderColor: brandColors.PrimaryText }}>
  <table className="w-full text-left table-auto border-collapse">
    <thead>
      <tr style={{ backgroundColor:
brandColors.AccentDeepBlue, color: brandColors.PrimaryBackground
}}>
        <th className="p-3 font-semibold text-sm uppercase">Item</th>
        <th className="p-3 font-semibold text-sm uppercase">Status</th>
        <th className="p-3 font-semibold text-sm uppercase">Owner</th>
        <th className="p-3 font-semibold text-sm uppercase">Due Date</th>
      </tr>
    </thead>
    <tbody className="text-sm">
      <tr className="border-b" style={{ borderColor:
brandColors.GrayLight, backgroundColor:
brandColors.SemanticSuccessGreenLight }}>
        /* ... cell content ... */
      </tr>
      <tr className="border-b" style={{ backgroundColor:
brandColors.WarningYellowLight }}>
        /* ... cell content ... */
      </tr>
      <tr className="border-b" style={{ backgroundColor:
brandColors.ErrorRedLight }}>
        /* ... cell content ... */
      </tr>
      <tr className="border-b" style={{ borderColor:
brandColors.GrayLight }}>
        /* ... cell content ... */
      </tr>
    </tbody>
  </table>
</div>
```


7. COMPONENT EXAMPLES

Architectural diagrams are a key part of our technical documentation. These components provide a standardized set of shapes and styles to ensure that all diagrams are consistent, professional, and easy to understand.

Diagram Elements

Key elements used to build architectural diagrams.



```
<ResponsiveDiagramContainer patternedBackground={true}>
  <div className="flex items-center justify-center ..." >
    <DemoBox title="Users" ...>
      <DemoNode ...>End User</DemoNode>
    </DemoBox>
    <DemoArrow label="Data Flow" ... />
    <DemoGroup title="API Layer" ...>
      <DemoNode ...>Secure API</DemoNode>
    </DemoGroup>
    <DemoArrow label="Connects" ... />
    <DemoBox title="Database" ...>
      <DemoNode ...>Azure DB</DemoNode>
    </DemoBox>
  </div>
</ResponsiveDiagramContainer>
```

8. SPACING & RESPONSIVE DESIGN

A consistent spacing scale creates visual rhythm and harmony. Using multiples of a base unit (4px) ensures that elements are aligned and layouts feel balanced. This is not a strict rule, but a strong guideline to avoid chaotic or inconsistent designs.

Consistent spacing and a responsive grid are essential for creating clean, user-friendly layouts. We use a 4px base unit for all spacing and margins.



2px
rem(0.125)



4px
rem(0.25)



8px
rem(0.5)



16px
rem(1)



24px
rem(1.5)



32px
rem(2)



48px
rem(3)



64px
rem(4)

```
<div className="grid grid-cols-2 md:grid-cols-4 gap-4 ...">
  {[/* ...spacing values */].map([unit, px]) => (
    <div key={px}>
      <div className="h-12 w-full ..."></div>
      <p>{px}</p>
      <p>rem({unit / 4})</p>
    </div>
  )]}
</div>
```

Responsive Breakpoints

Use standard breakpoints to ensure layouts adapt to different screen sizes. Our breakpoints are mobile-first.

Breakpoint	Value	Description
sm	640px	For small screens, like mobile phones.
md	768px	For medium screens, like tablets.
lg	1024px	For large screens, like laptops.
xl	1280px	For extra-large screens, like desktops.

```

<table className="w-full ...">
  {/* ...table headers... */}
  <tbody>
    <tr><td>sm</td><td>640px</td><td>For small screens...
  </td></tr>
    <tr><td>md</td><td>768px</td><td>For medium screens...
  </td></tr>
    <tr><td>lg</td><td>1024px</td><td>For large screens...
  </td></tr>
    <tr><td>xl</td><td>1280px</td><td>For extra-large
  screens...</td></tr>
  </tbody>
</table>

```

9. LAYOUTS & EXAMPLES

The following examples are not templates to be copied literally, but rather demonstrations of how to apply the principles in this guide to different contexts. The goal is to show how the brand can be both consistent and flexible, whether in a formal document or a modern web application.

Print Document Examples

Proposal Cover

PRIMATIF ENTERPRISE CLOUD MIGRATION STRATEGY

Prepared for: Acme Corporation

June 20, 2025

```

<div className="p-8 shadow-lg text-center">
  <div className="font-bold ...">PRIMATIF</div>
  <h1 className="text-4xl ...">Enterprise Cloud Migration
  Strategy</h1>

```

```
<p className="text-lg ...">Prepared for: Acme Corporation</p>
<div className="w-1/2 h-1 mx-auto ..."></div>
<p className="text-sm mt-8 ...">June 20, 2025</p>
</div>
```

Internal Report Page

Project Status Overview

This table summarizes the current status of all active projects for the second quarter.

ITEM	STATUS	OWNER	DUE DATE
Initial Project Setup	<div>✔ Complete</div>	Tech Lead	2025-06-15
API Key Provisioning	<div>⚠ In Progress</div>	Client IT	2025-06-22
Database Credentials Update	<div>✖ Blocked</div>	Tech Lead	2025-06-20

```
<div className="p-8 shadow-lg">
  <Header />
  <div className="py-8">
    <h2 ...>Project Status Overview</h2>
    <TableFormatting />
  </div>
  <Footer />
</div>
```

3. Implementing Secure API Endpoints

The following code demonstrates a basic secure endpoint using Python and Flask. Note the use of the `@jwt_required()` decorator to protect the route.

```
from flask import Flask
from flask_jwt_extended import jwt_required

app = Flask(__name__)

@app.route('/api/v1/data', methods=['GET'])
@jwt_required()
def get_data():
    # Your secure logic here
    return {"message": "Access to secure data granted."}
```

```
<div className="p-8 shadow-lg">
  <Header />
  <div className="py-8">
    <h2 ...>3. Implementing Secure API
Endpoints</h2>
    <p>The following code demonstrates a basic
secure endpoint...</p>
    <CodeBlock>{pythonCode}</CodeBlock>
  </div>
</div>
```

Gemini App Examples

Chat Interface

PRIMATIF AI

New Chat

Explain cloud migration benefits.

History

Settings

Cloud migration offers cost savings, scalability, and enhanced security.

Ask Primatif AI...



```
<div className="rounded-lg shadow-lg flex h-96">
  { /* Sidebar */ }
  <div className="w-1/4 ...">...</div>
  { /* Main Chat Area */ }
  <div className="w-3/4 ...">
    { /* Messages */ }
    <div className="flex-grow ...">...</div>
    { /* Input */ }
    <div className="mt-4 ...">...</div>
  </div>
</div>
```

Data Dashboard

Project Analytics

Active Projects

12

On-time
Completion

92%

Critical Alerts

1

```
<div className="rounded-lg shadow-lg p-6">
  <h3 ...>Project Analytics</h3>
  <div className="grid grid-cols-3 gap-4">
    <div className="p-4 ...">...</div>
    <div className="p-4 ...">...</div>
    <div className="p-4 ...">...</div>
  </div>
</div>
```

```
</div>
</div>
```

Settings Page

Application Settings

API Endpoint

Enable Beta Features



Dark Mode



Save Changes

```
<div className="rounded-lg shadow-lg p-6">
  <h3 ...>Application Settings</h3>
  <div className="space-y-4">
    <div><label>API Endpoint</label><input ... /></div>
    <div><span>Enable Beta Features</span><Toggle /></div>
    <div><span>Dark Mode</span><Toggle /></div>
    <div><button>Save Changes</button></div>
  </div>
</div>
```

10. CALLOUTS

Callouts are used to highlight important information to the user. They use a glassmorphism effect to stand out from the content without being distracting. Use them for tips, warnings, and confirmations.



Informational Tip

This is an informational message. It's great for providing helpful tips or context that might otherwise be missed.



Success!

The operation completed successfully. Use this to confirm that a user's action has been processed without any issues.



Warning

Please be cautious. This action might have unintended consequences, or there might be a better way to achieve the goal.



Danger Zone

This is a critical alert. This action is not reversible and may result in permanent data loss or security vulnerabilities.

```
<Callout type="info" title="Informational Tip">
```

```
  This is an informational message. It's great for providing  
  helpful tips or context.
```

```
</Callout>
```

```
<Callout type="success" title="Success!">
```

```
  The operation completed successfully. Well done!
```

```
</Callout>
```

```
<Callout type="warning" title="Warning">
```

```
  Please be cautious. This action might have unintended  
  consequences.
```

```
</Callout>
```

```
<Callout type="danger" title="Danger Zone">
```

```
  This is a critical alert. This action is not reversible and  
  may result in data loss.
```

```
</Callout>
```