

Core SaaS Pages - Designer Guide

Guide for designing and implementing the shared tenant pages



Table of Contents

1. Overview
2. Core Pages to Design
3. Critical Rules
4. File Structure
5. Theming System
6. Component Patterns
7. Asset Management
8. Testing Your Work
9. Common Pitfalls
10. Checklist

Overview

What Are Core Pages?

There are **TWO types of pages** in BudStack:

1. Platform Pages (`budstack.to`)

The main SaaS landing page and platform marketing pages:

- **Homepage** (`/app/page.tsx`) - "White-Label Medical Cannabis E-Commerce Platform"
- **About** (`/app/about/`) - Platform information
- **Pricing** (if applicable) - Pricing tiers

These pages market the **BudStack platform itself** to potential tenants.

2. Core Tenant Pages (`/store/[slug]/`)

Shared pages across ALL tenant storefronts. Unlike template-specific homepage designs, these pages must:

- Work for **every tenant** regardless of their chosen template
- Dynamically adapt to each tenant's **theme** (colors, fonts, spacing)
- Be **production-ready** with no placeholders
- Support **responsive design** (mobile, tablet, desktop)
- Follow **accessibility standards** (WCAG 2.1 AA)

The 5 Core Tenant Pages

Page	Purpose	Complexity
How It Works	Explains the consultation process	Medium
Conditions	Lists medical conditions treated	Medium
Products	Displays product catalog with filters	High
Consultation	Multi-step consultation form	High
Contact	Contact form and business info	Low

Platform Homepage (`budstack.to`)

Page	Purpose	Complexity
Platform Home	SaaS landing page	Medium

Core Pages to Design

0. Platform Homepage (`/app/page.tsx`)

Purpose: Market the BudStack SaaS platform to potential dispensary owners.

Current State: Functional but could use design improvements

Design Requirements:

- Hero section with value proposition: "White-Label Medical Cannabis E-Commerce Platform"
- Key features showcase
- NFT license requirement messaging
- Call-to-action to onboarding
- Trust indicators (compliance, speed, scale)
- Social proof (if applicable)

Key Data Points:

- Platform settings from `/api/super-admin/platform-settings`
- Business name: "BudStack" (from platform settings)
- Primary color: Dynamic (from platform settings)

Important Notes:

- This is **NOT** tenant-specific
- Does **NOT** use CSS variables like tenant pages
- Uses fixed BudStack branding
- Links to `/onboarding` for new tenant signup

1. How It Works (/app/store/[slug]/how-it-works/page.tsx)

Purpose: Explain the 3-step consultation process to patients.

Current State: Basic structure exists, needs visual polish

Design Requirements:

- Clear step-by-step process visualization
- Icons for each step (Book → Consult → Receive)
- Trust indicators (certifications, compliance)
- FAQ section
- CTA to start consultation

Key Data Points:

- Steps are **hardcoded** (same for all tenants)
- But styling must be **dynamic** (uses tenant theme)

2. Conditions (/app/store/[slug]/conditions/page.tsx)

Purpose: Showcase medical conditions that can be treated.

Current State: Basic grid exists, needs better UX

Design Requirements:

- Grid/card layout for conditions
- Icons for each condition type
- Filter/search functionality
- Links to consultation page
- Medical disclaimer

Key Data Points:

```
const conditions = [
  { name: 'Chronic Pain', icon: 'Zap', description: '...' },
  { name: 'Anxiety', icon: 'Brain', description: '...' },
  // etc.
];
```

3. Products (/app/store/[slug]/products/page.tsx)

Purpose: Display product catalog fetched from Doctor Green API.

Current State: Functional but needs UI improvements

Design Requirements:

- Product grid with images
- Filter by strain type (Indica/Sativa/Hybrid/CBD)
- Sort by price, THC%, CBD%
- Quick view modal

- Add to cart functionality
- Loading states and error handling

Key Data Points:

- Products fetched from `/api/store/[slug]/products`
 - Dynamic data (different for each tenant)
 - Uses Doctor Green API integration
-

4. Consultation (`/app/store/[slug]/consultation/page.tsx`)

Purpose: Multi-step form to collect patient information.

Current State:  Fully functional, just needs styling consistency

Design Requirements:

- 5-step wizard (Contact → Address → Conditions → Medical History)
- Progress indicator
- Form validation
- Back/Next navigation
- Success page after submission

Key Data Points:

- Form managed by `ConsultationForm` component
 - Submits to `/api/consultation/submit`
 - Integrates with Doctor Green API
-

5. Contact (`/app/store/[slug]/contact/page.tsx`)

Purpose: Allow users to reach the business.

Current State:  Basic form exists

Design Requirements:

- Contact form (name, email, message)
- Business contact info (dynamic per tenant)
- Map/location (if applicable)
- Social media links
- Office hours

Key Data Points:

```
// Contact info comes from tenant settings
const contact = {
  email: tenant.settings.contactEmail,
  phone: tenant.settings.phone,
  address: tenant.settings.address
};
```

Critical Rules

DO

1. Use CSS Variables for ALL colors and fonts

```tsx

``` 2. \*\*Make content tenant-aware\*\* ```tsx

{tenant.businessName}'s Product Catalog

``` 3. \*\*Handle loading and error states\*\* ```tsx {loading && } {error && {error}} {data && } ```

4. \*\*Test on multiple screen sizes\*\* - Mobile: 375px - Tablet: 768px - Desktop: 1280px 5. \*\*Use semantic HTML\*\* - `

`  
`  
`  
`

- Proper heading hierarchy (h1 → h2 → h3) --- #####  DON'T 1. \*\*Hardcode colors or fonts\*\*

```tsx 



``` 2. \*\*Hardcode business names\*\* ```tsx 

## Welcome to HealingBuds

---



## Welcome to {tenant.businessName}

---

``` 3. \*\*Import template-specific components\*\* ```tsx  import GTAHero from '@/templates/gta-cannabis/components/Hero'  // Core pages don't use template components ``` 4. \*\*Use absolute positioning without fallbacks\*\* ```tsx 



``` 5. \*\*Forget accessibility\*\* ```tsx 

Click me

## ## File Structure

### ### Core Pages Location

```
/home/ubuntu/healing-
buds_website/nextjs_space/app/s
tore/[slug]/
 └── how-it-works/
 └── page.tsx
 it this
 └── conditions/
 └── page.tsx
 it this
 └── products/
 └── page.tsx
 it this
 └── [id]/
 └── page.tsx
 oduct detail page
 └── consultation/
 └── page.tsx
 it this
 └── contact/
 └── page.tsx
 it this
 └── page.tsx
 it this
```

### ### Shared Components

```
graph TD; consultation[consultation] --> form[form.tsx]; consultation --> steps[steps/]; consultation --> contactStep[contact-step.tsx]; consultation --> addressStep[address-step.tsx]; consultation --> conditionsStep[conditions-step.tsx]; consultation --> medicalHistory[medical-history-*_.tsx]; ui[ui/] --> shadcn[shadcn/ui components]; shadcn --> button[button.tsx]; shadcn --> card[card.tsx]; shadcn --> input[input.tsx]; shadcn --> ...[...]; shadcn --> cart[cart- dropdown.tsx]; shadcn --> hopping[hopping cart]
```

The diagram shows a file tree structure for a 'consultation' component. The 'consultation' folder contains 'form.tsx', 'steps/' (which includes 'contact-step.tsx', 'address-step.tsx', 'conditions-step.tsx', and 'medical-history-\*\_.tsx'), and a 'ui' folder. The 'ui' folder contains 'shadcn/ui components' (which includes 'button.tsx', 'card.tsx', 'input.tsx', and '...') and 'dropdown.tsx'. A separate 'hopping cart' file is also shown.

### ### Assets Directory

