BudStack Template Design Guide

Overview

This guide defines the standard architecture for building multi-tenant templates in BudStack. It combines best practices from the hbgpt sample template with BudStack's multi-tenant requirements.

Core Principles

1. Template as a Function

Every template is a React component that receives standardized props and renders a complete homepage.

```
interface TemplateProps {
  tenant: Tenant;              // Full tenant object from database
  consultationUrl: string;     // /store/{slug}/consultation
  productsUrl: string;         // /store/{slug}/products
  contactUrl: string;          // /store/{slug}/contact
  aboutUrl: string;            // /store/{slug}/about
  heroImageUrl?: string | null; // S3 URL or null
  logoUrl?: string | null;     // S3 URL or null
}
```

2. Component-Based Architecture

Break templates into reusable sections:

```
templates/
└── {template-name}/
    ├── index.tsx              # Main template component
    ├── components/
    │   ├── Hero.tsx           # Hero section
    │   ├── About.tsx          # About section
    │   ├── ValueProps.tsx     # Features/Benefits
    │   ├── CTA.tsx            # Call-to-action
    │   ├── Footer.tsx         # Custom footer
    │   └── Header.tsx         # Custom header (optional)
    ├── styles.css             # Template-specific styles
    ├── defaults.json          # Default settings
    ├── template.config.json   # Metadata
    └── README.md              # Documentation
```

3. Data Flow Pattern

```
Database (Tenant)
  ↓
API Route (/store/[slug]/page.tsx)
  ↓
Template Component (index.tsx)
  ↓
Section Components (Hero.tsx, About.tsx, etc.)
  ↓
Rendered HTML
```

Template Structure

Main Template File (`index.tsx`)

```
'use client';

import { useState, useEffect } from 'react';
import { Tenant } from '@prisma/client';

// Import section components
import { Hero } from './components/Hero';
import { About } from './components/About';
import { ValueProps } from './components/ValueProps';
import { Footer } from './components/Footer';

import './styles.css';

interface TemplateProps {
  tenant: Tenant;
  consultationUrl: string;
  productsUrl: string;
  contactUrl: string;
  aboutUrl: string;
  heroImageUrl?: string | null;
  logoUrl?: string | null;
}

export default function MyTemplate({
  tenant,
  consultationUrl,
  productsUrl,
  contactUrl,
  aboutUrl,
  heroImageUrl,
  logoUrl
}: TemplateProps) {
  // Extract settings from tenant.settings JSON field
  const settings = (tenant.settings as any) || {};

  return (
    <>
      <Hero
        businessName={tenant.businessName}
        tagline={settings.tagline || 'Welcome'}
        heroImageUrl={heroImageUrl}
        logoUrl={logoUrl}
        consultationUrl={consultationUrl}
      />

      <About
        businessName={tenant.businessName}
        description={settings.aboutContent}
      />

      <ValueProps />

      <Footer
        businessName={tenant.businessName}
        contactEmail={tenant.contactEmail}
        contactPhone={settings.contactPhone}
        socialLinks={settings.socialLinks}
      />
    </>
  );
}
```

Section Component Pattern

```
// components/Hero.tsx
'use client';

import Image from 'next/image';
import Link from 'next/link';
import { motion } from 'framer-motion';

interface HeroProps {
  businessName: string;
  tagline?: string | null;
  heroImageUrl?: string | null;
  logoUrl?: string | null;
  consultationUrl: string;
}

export function Hero({
  businessName,
  tagline,
  heroImageUrl,
  logoUrl,
  consultationUrl
}: HeroProps) {
  return (
    <section className="relative min-h-screen">
      {/* Background */}
      {heroImageUrl && (
        <div className="absolute inset-0">
          <Image
            src={heroImageUrl}
            alt={businessName}
            fill
            className="object-cover"
            priority
          />
        </div>
      )}

      {/* Content */}
      <div className="relative z-10 container mx-auto px-4 py-32">
        <motion.h1
          initial={{ opacity: 0, y: 20 }}
          animate={{ opacity: 1, y: 0 }}
          className="text-6xl font-bold mb-4"
        >
          {businessName}
        </motion.h1>

        {tagline && (
          <p className="text-xl mb-8">{tagline}</p>
        )}

        <Link
          href={consultationUrl}
          className="px-8 py-4 bg-primary text-white rounded-lg"
        >
          Book Consultation
        </Link>
      </div>
    </section>
  );
}
```

Footer Component Standard

Props Interface

```
interface FooterProps {
  businessName: string;
  contactEmail?: string | null;
  contactPhone?: string | null;
  address?: string | null;
  socialLinks?: {
    facebook?: string;
    instagram?: string;
    twitter?: string;
    linkedin?: string;
  };
  // Optional navigation links
  aboutUrl?: string;
  productsUrl?: string;
  contactUrl?: string;
}
```

Implementation Example

```
// components/Footer.tsx
'use client';

import Link from 'next/link';
import Image from 'next/image';

export function Footer({
  businessName,
  contactEmail,
  contactPhone,
  address,
  socialLinks,
  aboutUrl,
  productsUrl,
  contactUrl
}: FooterProps) {
  return (
    <footer className="bg-foreground text-background py-12">
      <div className="container mx-auto px-4">
        <div className="grid md:grid-cols-4 gap-8">
          {/* Column 1: Branding */}
          <div>
            <h3 className="font-bold text-xl mb-4">{businessName}</h3>
            <p className="text-sm">Your trusted partner in medical cannabis</p>
          </div>

          {/* Column 2: Quick Links */}
          <div>
            <h4 className="font-semibold mb-4">Quick Links</h4>
            <ul className="space-y-2">
              {aboutUrl && (
                <li><Link href={aboutUrl}>About Us</Link></li>
              )}
              {productsUrl && (
                <li><Link href={productsUrl}>Products</Link></li>
              )}
              {contactUrl && (
                <li><Link href={contactUrl}>Contact</Link></li>
              )}
            </ul>
          </div>

          {/* Column 3: Contact */}
          <div>
            <h4 className="font-semibold mb-4">Contact</h4>
            {contactEmail && (
              <p className="text-sm mb-2">
                <a href={`mailto:${contactEmail}`}>{contactEmail}</a>
              </p>
            )}
            {contactPhone && (
              <p className="text-sm mb-2">{contactPhone}</p>
            )}
            {address && (
              <p className="text-sm">{address}</p>
            )}
          </div>

          {/* Column 4: Social */}
          {socialLinks && (
            <div>
              <h4 className="font-semibold mb-4">Follow Us</h4>
              <div className="flex gap-4">
                {/* Social icons */}
              </div>
            </div>
          )}
        </div>

        {/* Copyright */}
        <div className="border-t border-white/20 mt-8 pt-8 text-center text-sm">
          <p>&copy; {new Date().getFullYear()} {businessName}. All rights reserved.</p>
        </div>
      </div>
    </footer>
  );
}
```

Settings Schema (`defaults.json`)

```
{
  "template": "my-template",
  "fontFamily": "inter",
  "headingFontFamily": "playfair",
  "fontSize": "medium",
  "primaryColor": "#059669",
  "secondaryColor": "#10b981",
  "accentColor": "#34d399",
  "backgroundColor": "#ffffff",
  "textColor": "#1f2937",
  "heroType": "image",
  "heroVideoUrl": null,
  "heroImagePath": null,
  "logoPath": null,
  "tagline": "Your medical cannabis partner",
  "aboutContent": "We provide premium medical cannabis products...",
  "contactPhone": "+1 234 567 8900",
  "address": "123 Main St, City, Country",
  "socialLinks": {
    "facebook": "",
    "instagram": "",
    "twitter": "",
    "linkedin": ""
  },
  "pageContent": {
    "home": {
      "heroTitle": "Welcome to Medical Cannabis",
      "heroSubtitle": "Premium products delivered with care"
    }
  }
}
```

Styling Guidelines

1. Use Tailwind CSS for Base Styles

```
<section className="bg-background py-16 px-4">
  <div className="container mx-auto">
    <h2 className="text-4xl font-bold text-foreground mb-8">
      {title}
    </h2>
  </div>
</section>
```

2. Template-Specific CSS in `styles.css`

```
/* Template-specific animations */
@keyframes fadeInUp {
  from {
    opacity: 0;
    transform: translateY(20px);
  }
  to {
    opacity: 1;
    transform: translateY(0);
  }
}

/* Custom classes */
.hover-lift {
  transition: transform 0.3s ease;
}

.hover-lift:hover {
  transform: translateY(-4px);
}
```

3. Dynamic Colors from Tenant Settings

```
<button
  style={{
    backgroundColor: tenant.primaryColor,
    color: '#ffffff'
  }}
  className="px-8 py-4 rounded-lg"
>
  Book Now
</button>
```

Animation Patterns (Framer Motion)

Scroll-Triggered Animations

```
import { motion } from 'framer-motion';

<motion.div
  initial={{ opacity: 0, y: 20 }}
  whileInView={{ opacity: 1, y: 0 }}
  transition={{ duration: 0.6 }}
  viewport={{ once: true }}
>
  {content}
</motion.div>
```

Parallax Scrolling

```
import { useScroll, useTransform, motion } from 'framer-motion';
import { useRef } from 'react';

const ref = useRef<HTMLDivElement>(null);
const { scrollYProgress } = useScroll({
  target: ref,
  offset: ["start end", "end start"]
});

const y = useTransform(scrollYProgress, [0, 1], ["-20%", "20%"]);

<motion.div ref={ref} style={{ y }}>
  {content}
</motion.div>
```

Template Configuration (`template.config.json`)

```json
{
  "id": "my-template",
  "name": "My Template Name",
  "version": "1.0.0",
  "author": "Your Name",
  "description": "A beautiful template for medical cannabis dispensaries",
  "category": "medical",
  "tags": ["modern", "professional", "video-hero"],
  "preview_image": "/previews/my-template.jpg",
  "compatibility": {
    "platform": "budstack",
    "nextjs": ">=14.0.0",
    "react": ">=18.0.0",
    "features": ["multi-tenant", "path-routing", "s3-uploads"]
  },
  "features": [
    "responsive_design",
    "dark_mode",
    "parallax_scrolling",
    "video_background",
    "scroll_animations"
  ],
  "performance": {
    "lighthouse_score": 95,
    "page_size_kb": 850,
    "load_time_ms": 1200
  },
  "accessibility": {
    "wcag_level": "AA",
    "keyboard_navigation": true,
    "screen_reader_optimized": true
  }
}
```

Testing Checklist

Functional Tests

- [ ] Template loads without errors
- [ ] All navigation links work correctly
- [ ] Forms submit successfully
- [ ] Images load from S3
- [ ] Video autoplay works (if applicable)
- [ ] Mobile responsive layout
- [ ] Cross-browser compatibility

Visual Tests

- [ ] Colors match tenant branding
- [ ] Fonts display correctly
- [ ] Logo appears in correct locations
- [ ] Hero section displays properly

- [ ] Footer renders with all data
- [ ] Animations trigger correctly

Data Tests

- [ ] Tenant name displays correctly
- [ ] Contact information shows
- [ ] Custom settings apply
- [ ] Default values work when settings missing

Best Practices

1. Always Provide Fallbacks

```
const tagline = settings.tagline || 'Welcome to our store';
const heroImage = heroImageUrl || '/default-hero.jpg';
```

2. Type Safety

```
// Define strict interfaces
interface TenantSettings {
  tagline?: string;
  aboutContent?: string;
  contactPhone?: string;
  // ...
}

const settings = (tenant.settings as TenantSettings) || {};
```

3. Performance Optimization

```
// Use Next.js Image for optimization
import Image from 'next/image';

<Image
  src={heroImageUrl}
  alt={businessName}
  fill
  className="object-cover"
  priority // For above-the-fold images
/>
```

4. Accessibility

```
<button
  onClick={handleClick}
  aria-label="Book consultation"
  className="..."
>
  <Icon className="w-6 h-6" aria-hidden="true" />
  Book Now
</button>
```

5. Error Boundaries

```
try {
  const data = JSON.parse(tenant.settings);
} catch (error) {
  console.error('Failed to parse tenant settings:', error);
  // Use defaults
}
```

Migration from Standalone to BudStack

Step 1: Identify Components

List all sections in your standalone template:
- Header/Navigation
- Hero
- About
- Features/Services
- Testimonials
- CTA
- Footer

Step 2: Convert to Component Pattern

For each section:
1. Create a new file in `components/`
2. Define prop interface
3. Accept tenant data as props
4. Remove hardcoded data

Step 3: Update Main Template

1. Import all components
2. Pass tenant data as props
3. Handle settings extraction

Step 4: Test Integration

1. Run `node apply-template-defaults.js`
2. Check database has correct settings
3. View `/store/{slug}` to verify

Example: Converting hbgpt Template

Before (Standalone)

```
// src/components/Hero.tsx
export function Hero() {
  return (
    <section>
      <h1>Healing Buds</h1>
      <p>Pioneering tomorrow's medical cannabis solutions</p>
    </section>
  );
}
```

After (BudStack)

```
// templates/healing-buds-video/components/Hero.tsx
interface HeroProps {
  businessName: string;
  tagline?: string | null;
  consultationUrl: string;
}

export function Hero({ businessName, tagline, consultationUrl }: HeroProps) {
  return (
    <section>
      <h1>{businessName}</h1>
      <p>{tagline || 'Welcome'}</p>
      <Link href={consultationUrl}>Book Now</Link>
    </section>
  );
}
```

Common Pitfalls

❌ Don’t Hardcode Data

```
// BAD
<h1>Healing Buds</h1>

// GOOD
<h1>{businessName}</h1>
```

❌ Don’t Use Absolute URLs

```
// BAD
<Link href="/consultation">

// GOOD
<Link href={consultationUrl}>
```

❌ Don’t Assume Settings Exist

```
// BAD
const tagline = settings.tagline;

// GOOD
const tagline = settings?.tagline || 'Default tagline';
```

❌ Don’t Mix Template Styles with Global Styles

```
// BAD - in globals.css
.hero-section { ... }

// GOOD - in template/styles.css
.my-template-hero { ... }
```

Resources

- Next.js Documentation (https://nextjs.org/docs)
- Framer Motion (https://www.framer.com/motion/)
- Tailwind CSS (https://tailwindcss.com/docs)
- ShadCN UI (https://ui.shadcn.com/)
- BudStack Architecture (./MULTI_TENANT_ARCHITECTURE.md)

Support

For questions or issues with template development, refer to:
- Multi-Tenant Architecture Guide (./MULTI_TENANT_ARCHITECTURE.md)
- Template Migration Guide (./template-design/MIGRATION_GUIDE.md)
- Example Templates (./nextjs_space/templates/)