

# Migrating to BudStack Template System v2.0

## Overview

This guide helps you migrate existing templates or create new ones using the flexible v2.0 system.



## From v1.0 (Section-Based) to v2.0 (Flexible)

### What Changed

#### v1.0 (Old System)

```
Template Structure:
├─ metadata.json      # Template config
├─ layout.tsx         # Fixed layout wrapper
├─ sections/          # REQUIRED sections
│   ├─ hero.tsx       # REQUIRED
│   ├─ trust.tsx      # REQUIRED
│   ├─ conditions.tsx # REQUIRED
│   ├─ process.tsx    # REQUIRED
│   ├─ education.tsx  # REQUIRED
│   ├─ testimonials.tsx # REQUIRED
│   └─ cta.tsx        # REQUIRED
└─ styles.css
```

#### v2.0 (New System)

```
Template Structure:
├─ template.config.json # Template metadata
├─ index.tsx            # Main entry (FLEXIBLE)
├─ components/          # Your custom components
│   ├─ WhateverYouWant.tsx
│   └─ AnyStructureWorks.tsx
└─ styles.css           # Optional
```

### Migration Steps

1. **Rename** `metadata.json` to `template.config.json`
2. **Merge** `layout.tsx` and `sections/*` into `index.tsx`
3. **Remove forced section constraints**
4. **Update component imports**

### Example Migration

**Before (v1.0 - layout.tsx):**

```

import { Tenant } from '@prisma/client';
import { Hero } from '../sections/hero';
import { Trust } from '../sections/trust';
import { Conditions } from '../sections/conditions';
import { Process } from '../sections/process';
import { Education } from '../sections/education';
import { Testimonials } from '../sections/testimonials';
import { CTA } from '../sections/cta';

export function ModernMinimalTemplate({ tenant, heroImageUrl, consultationUrl }) {
  return (
    <div className="template-modern-minimal">
      <Hero tenant={tenant} heroImageUrl={heroImageUrl} consultationUrl={consultation-
Url} />
      <Trust />
      <Conditions consultationUrl={consultationUrl} />
      <Process />
      <Education />
      <Testimonials />
      <CTA tenant={tenant} consultationUrl={consultationUrl} />
    </div>
  );
}

```

**After (v2.0 - index.tsx):**

```

import { Tenant } from '@prisma/client';
import { HeroWithVideo } from '../components/HeroWithVideo';
import { ProductShowcase } from '../components/ProductShowcase';
import { ContentSplit } from '../components/ContentSplit';

interface TemplateProps {
  tenant: Tenant;
  consultationUrl: string;
  productsUrl: string;
  contactUrl: string;
}

export default function FlexibleTemplate({
  tenant,
  consultationUrl,
  productsUrl,
  contactUrl
}: TemplateProps) {
  return (
    <div className="my-custom-template">
      {/* Arrange sections in ANY order */}
      <HeroWithVideo
        title={tenant.businessName}
        tagline={tenant.tagline}
        videoUrl={tenant.heroVideoUrl}
        ctaUrl={consultationUrl}
        primaryColor={tenant.primaryColor}
      />

      <ProductShowcase
        productsUrl={productsUrl}
        accentColor={tenant.accentColor}
      />

      <ContentSplit
        imageUrl={tenant.heroImageUrl}
        content={tenant.description}
        ctaText="Book Now"
        ctaUrl={consultationUrl}
      />

      {/* Add your own unique sections */}
    </div>
  );
}

```



## Converting React/Vite Templates to Next.js

If you have a React/Vite template (like the uploaded `hbgpt` template), follow these steps:

### 1. File Structure Conversion

**Vite Structure:**

```

hbgpt/
├── src/
│   ├── pages/
│   │   └── Index.tsx           # Main page
│   ├── components/
│   │   ├── Header.tsx
│   │   ├── Hero.tsx
│   │   └── Footer.tsx
│   └── main.tsx
├── public/
└── index.html

```

### Next.js Template Structure:

```

healing-buds-template/
├── template.config.json
├── index.tsx                 # src/pages/Index.tsx content
├── components/
│   ├── Hero.tsx             # src/components/Hero.tsx
│   └── ...
├── assets/
└── ...                      # public/ files

```

## 2. Code Changes

### Remove React Router

```

- import { Link } from 'react-router-dom';
+ import Link from 'next/link';

- <Link to="/about">About</Link>
+ <Link href="/about">About</Link>

```

### Update Image Imports

```

- import heroImage from '@assets/hero.png';
- <img src={heroImage} alt="Hero" />
+ import Image from 'next/image';
+ <Image src="/assets/hero.png" alt="Hero" width={800} height={600} />

```

### Replace Vite Env Variables

```

- import.meta.env.VITE_API_KEY
+ process.env.NEXT_PUBLIC_API_KEY

```

### Update CSS Imports

```

- import './styles.css';
+ import '@styles/template.css';

```

## 3. Remove Unused Files

Delete these Vite-specific files:

```

- vite.config.ts

```

- index.html
- src/main.tsx
- .eslintrc.js (if not compatible)

## 4. Update package.json Dependencies

Remove:

```
{
  "dependencies": {
    "react-router-dom": "^6.x",
    "vite": "^5.x"
  }
}
```

These are provided by Next.js platform.

## Example: Migrating HBGpt Template

Let's convert the uploaded `hbgpt` template:

### Step 1: Create Template Config

**template.config.json:**

```
{
  "id": "healing-buds-video-hero",
  "name": "Healing Buds Video Hero",
  "version": "1.0.0",
  "author": "HB Design Team",
  "description": "Full-screen video hero with parallax scrolling and modern sections",
  "preview_image": "/preview.jpg",
  "tags": ["video", "parallax", "modern", "medical"],
  "created_at": "2025-11-24",
  "compatibility": {
    "platform_version": "2.0+",
    "nextjs_version": "14.x",
    "react_version": "18.x"
  },
  "features": {
    "hero_video": true,
    "parallax_scrolling": true,
    "scroll_animations": true
  }
}
```

### Step 2: Convert Main Page

**index.tsx:**

```

import { Tenant } from '@prisma/client';
import { Hero } from '../components/Hero';
import { AboutHero } from '../components/AboutHero';
import { ValueProps } from '../components/ValueProps';
import { Cultivation } from '../components/Cultivation';
import { International } from '../components/International';
import { News } from '../components/News';

interface HealingBudsTemplateProps {
  tenant: Tenant;
  consultationUrl: string;
  productsUrl: string;
  contactUrl: string;
}

export default function HealingBudsTemplate({
  tenant,
  consultationUrl,
  productsUrl,
  contactUrl
}: HealingBudsTemplateProps) {
  return (
    <div className="template-healing-buds">
      <Hero
        businessName={tenant.businessName}
        tagline={tenant.tagline}
        videoUrl={tenant.heroVideoUrl || '/hero-video.mp4'}
        logoUrl={tenant.logoUrl}
      />

      <AboutHero description={tenant.description} />

      <ValueProps consultationUrl={consultationUrl} />

      <Cultivation />

      <International contactUrl={contactUrl} />

      <News />
    </div>
  );
}

```

## Step 3: Convert Components

components/Hero.tsx:

```

import { useRef, useEffect } from 'react';
import Image from 'next/image';
import { motion, useScroll, useTransform } from 'framer-motion';
import { ChevronDown } from 'lucide-react';

interface HeroProps {
  businessName: string;
  tagline?: string;
  videoUrl?: string;
  logoUrl?: string;
}

export function Hero({ businessName, tagline, videoUrl, logoUrl }: HeroProps) {
  const videoRef = useRef<HTMLVideoElement>(null);
  const containerRef = useRef<HTMLDivElement>(null);

  const { scrollYProgress } = useScroll({
    target: containerRef,
    offset: ["start start", "end start"]
  });

  const videoY = useTransform(scrollYProgress, [0, 1], ["0%", "30%"]);
  const contentY = useTransform(scrollYProgress, [0, 1], ["0%", "50%"]);
  const opacity = useTransform(scrollYProgress, [0, 0.5], [1, 0]);

  useEffect(() => {
    if (videoRef.current && videoUrl) {
      videoRef.current.play().catch(error => {
        console.log("Video autoplay failed:", error);
      });
    }
  }, [videoUrl]);

  const scrollToContent = () => {
    window.scrollTo({
      top: window.innerHeight,
      behavior: 'smooth'
    });
  };

  return (
    <section
      ref={containerRef}
      className="relative min-h-screen flex items-center overflow-hidden pt-28
sm:pt-36 md:pt-44"
    >
      </* Video Background with Parallax */>
      {videoUrl && (
        <motion.div
          style={{ y: videoY }}
          className="absolute left-2 right-2 sm:left-4 sm:right-4 top-24 sm:top-32
md:top-40 bottom-4 rounded-2xl sm:rounded-3xl overflow-hidden z-0 shadow-2xl"
        >
          <video
            ref={videoRef}
            autoPlay
            muted
            loop
            playsInline
            className="w-full h-full object-cover"
          >
            <source src={videoUrl} type="video/mp4" />
          </video>
        </motion.div>
      )}
    </section>
  );
}

```

```

        </video>
        <div className="absolute inset-0 bg-gradient-to-br from-[#1F2A25]/60 to-
[#13303D]/55" />
        </motion.div>
    )}

    <motion.div
    style={{ y: contentY, opacity }}

className="container mx-auto px-3 sm:px-6 lg:px-8 relative z-10 py-16 sm:py-24
md:py-32"
    >
        <div className="max-w-5xl text-left relative">
            <h1 className="font-pharma text-5xl sm:text-6xl md:text-7xl lg:text-8xl
font-semibold text-white mb-6 sm:mb-8 leading-[1.1] tracking-tight drop-shadow-lg">
                Welcome to{" "}
                <span className="block mt-3">{businessName}</span>
            </h1>

            {logoUrl && (
                <div className="hidden md:block absolute -right-8 md:right-4 lg:right-12
top-1/2 -translate-y-1/2 w-[380px] md:w-[480px] lg:w-[560px] h-auto opacity-15
pointer-events-none">
                    <Image
                        src={logoUrl}
                        alt=""
                        width={560}
                        height={560}
                        className="object-contain"
                    />
                </div>
            )}

            {tagline && (
                <p className="font-body text-lg sm:text-xl md:text-2xl text-white/90 mb-8
max-w-2xl font-light leading-relaxed drop-shadow-md">
                    {tagline}
                </p>
            )}
        </div>
    </motion.div>

    {/* Scroll Indicator */}
    <button
        onClick={scrollToContent}
        className="absolute bottom-8 left-1/2 -translate-x-1/2 z-20 text-white/80 hov-
er:text-white transition-all duration-300 animate-bounce cursor-pointer"
        aria-label="Scroll to content"
    >
        <ChevronDown className="w-8 h-8" />
    </button>
</section>
);
}

```



## Packaging for Upload

---

### Option 1: Git Repository

```
# Create Git repo
cd healing-buds-template
git init
git add .
git commit -m "Initial template"
git remote add origin https://github.com/yourname/hb-template.git
git push -u origin main

# Tag version
git tag v1.0.0
git push --tags
```

Then upload via Super Admin using Git URL.

### Option 2: ZIP File

```
# Create ZIP
cd healing-buds-template
zip -r healing-buds-template-v1.0.0.zip . -x "*.git*" "node_modules/*"
```

Then upload ZIP via Super Admin interface.

---

## ? FAQ

#### Q: Can I use external npm packages?

A: Yes, but they must be compatible with Next.js 14 and React 18.

#### Q: Do I need to include Navigation/Footer?

A: No, those are handled by the platform. Focus on page content only.

#### Q: Can I override platform styles?

A: Yes, use scoped CSS classes or CSS modules to avoid conflicts.

#### Q: How do I test my template locally?

A: Clone the BudStack repo, drop your template in `templates/`, and run dev server.

#### Q: Can I sell my templates?

A: Yes! BudStack will have a marketplace for premium templates.

---

**Need Help?** Join our Discord: <https://discord.gg/budstack>