

# Multi-Tenant Medical Cannabis SaaS Platform

## Comprehensive Architecture & Development Plan

---

### Executive Summary

**Platform Name:** MedCannabis SaaS (or your preferred name)

**First Tenant:** HealingBuds

**Core Model:** NFT-based licensing with Doctor Green API integration

**Business Model:** Sell NFTs → Transfer to customers → They launch branded dispensary

---

### System Architecture Overview

#### Three-Tier User Hierarchy

##### SUPER ADMIN LAYER

(Your Management Team - NFT & Platform Management)

↓

##### TENANT ADMIN LAYER

(HealingBuds, Future Customers - Store Management)

↓

##### END USER LAYER

(Medical Cannabis Patients - Shopping & Ordering)

---

### User Roles & Permissions Matrix

#### 1. Super Admin (Your Team)

- Manage all 200+ NFTs (view, enable/disable, assign)
- Create/suspend/delete tenant accounts
- View all tenant activity & analytics
- Manage platform templates
- Configure global Doctor Green API settings
- Platform billing & subscription management
- System-wide settings & configuration

#### 2. Tenant Admin (e.g., HealingBuds Owner)

- Customize their storefront (logo, colors, branding)
- Select & customize homepage template
- Manage their product catalog (sync'd with Doctor Green)
- Process orders & consultations
- Manage their customers
- View their analytics & reports
- Configure their domain

- Manage their team members (if multi-user)
- Cannot access other tenants' data
- Cannot manage NFTs

### 3. End User (Medical Cannabis Patients)

- Browse products on tenant's storefront
- Place orders (dropshipped via Doctor Green)
- Book consultations
- Manage their profile & orders
- No backend access

---

## Multi-Tenant Database Architecture

### Core Entities

#### 1. NFTs Table

```
model NFT {
  id                String    @id @default(cuid())
  tokenId           String    @unique // Doctor Green NFT ID
  walletAddress     String?   // Current owner's wallet
  status            NFTStatus @default(AVAILABLE) // AVAILABLE, ASSIGNED, ACTIVE, SUSPENDED
  assignedTo        String?   // Tenant ID if assigned
  tenant            Tenant?   @relation(fields: [assignedTo], references: [id])
  metadata          Json?     // NFT metadata from Doctor Green
  purchaseDate      DateTime?
  activationDate    DateTime?
  expiryDate        DateTime?
  createdAt         DateTime  @default(now())
  updatedAt         DateTime  @updatedAt
}

enum NFTStatus {
  AVAILABLE // In your inventory
  ASSIGNED  // Sold to customer, pending activation
  ACTIVE    // Live tenant using it
  SUSPENDED // Temporarily disabled
  REVOKED   // Permanently disabled
}
```

#### 2. Tenants Table

```
model Tenant {
  id                String    @id @default(cuid())
  name              String    // "HealingBuds"
  slug              String    @unique // "healingbuds" (for subdomain)
  domain            String?   @unique // "healingbuds.com" (custom domain)
  status            TenantStatus @default(PENDING_SETUP)

  // NFT Relationship
  nftId             String    @unique
  nft               NFT       @relation(fields: [nftId], references: [id])
}
```

```

// Branding
logo           String?      // S3 path
logoWhite      String?      // S3 path for dark backgrounds
primaryColor   String       @default("#10b981") // Green
secondaryColor String       @default("#059669")
accentColor    String?

// Template
templateId     String       // "modern", "minimal", "classic"
customCSS      String?      @db.Text

// Business Details
businessName    String
businessEmail   String
businessPhone   String?
businessAddress Json?       // { street, city, postal, country }
taxId           String?
licenseNumber   String?

// Doctor Green API Integration
doctorGreenApiKey String?    @db.Text // Encrypted
doctorGreenConfig Json?      // Additional API settings

// Relationships
users           TenantUser[]
products        Product[]
orders          Order[]
customers       Customer[]
consultations   Consultation[]
blogPosts       BlogPost[]

// Metadata
createdAt       DateTime     @default(now())
updatedAt       DateTime     @updatedAt
activatedAt     DateTime?
suspendedAt     DateTime?
}

enum TenantStatus {
    PENDING_SETUP // NFT assigned, awaiting onboarding
    ACTIVE        // Live and operational
    SUSPENDED     // Temporarily disabled
    CANCELLED     // Account closed
}

```

### 3. Tenant Users Table (Tenant Admin & Staff)

```

model TenantUser {
    id           String    @id @default(cuid())
    tenantId     String
    tenant       Tenant    @relation(fields: [tenantId], references: [id])

    email        String
    password     String    // Hashed
}

```

```

    firstName    String
    lastName     String
    role         TenantRole @default(ADMIN)

    createdAt    DateTime @default(now())
    updatedAt    DateTime @updatedAt

    @@unique([tenantId, email])
}

enum TenantRole {
    OWNER        // Full access
    ADMIN        // Can manage everything except billing
    STAFF        // Limited access (orders, products)
}

```

#### 4. Super Admin Table

```

model SuperAdmin {
    id            String    @id @default(cuid())
    email        String    @unique
    password     String    // Hashed
    firstName    String
    lastName     String
    role         SuperAdminRole @default(ADMIN)

    createdAt    DateTime @default(now())
    updatedAt    DateTime @updatedAt
}

enum SuperAdminRole {
    SUPER_ADMIN // Full platform access
    ADMIN       // Limited admin access
}

```

#### 5. Templates Table

```

model Template {
    id            String    @id @default(cuid())
    name          String    @unique // "Modern", "Minimal", "Classic"
    slug          String    @unique // "modern", "minimal", "classic"
    description   String?
    thumbnail     String?   // Preview image

    // Component Configuration
    heroLayout    String    // "full-screen", "split", "carousel"
    featuredSections Json    // Which sections to show
    colorScheme   Json      // Default colors

    isActive     Boolean    @default(true)
    sortOrder     Int       @default(0)

    createdAt    DateTime @default(now())
    updatedAt    DateTime @updatedAt
}

```

## 6. Customers Table (End Users - Tenant-Scoped)

```
model Customer {
  id          String    @id @default(cuid())
  tenantId    String
  tenant      Tenant    @relation(fields: [tenantId], references: [id])

  email       String
  password    String    // Hashed
  firstName   String
  lastName    String
  phone       String?

  // Medical Details
  medicalCardNumber String?
  medicalCardExpiry  DateTime?

  // Relationships
  orders      Order[]
  consultations Consultation[]

  createdAt    DateTime @default(now())
  updatedAt    DateTime @updatedAt

  @@unique([tenantId, email])
  @@index([tenantId])
}
```

## 7. Products Table (Synced from Doctor Green per Tenant)

```
model Product {
  id          String    @id @default(cuid())
  tenantId    String
  tenant      Tenant    @relation(fields: [tenantId], references: [id])

  // Doctor Green Reference
  doctorGreenId String    // Product ID in Doctor Green API
  doctorGreenData Json     // Full product data from API

  // Tenant Customization
  isVisible      Boolean @default(true)
  customDescription String? @db.Text
  displayOrder   Int?

  // Cache (updated from API periodically)
  name           String
  category       String
  thcContent     Float?
  cbdContent     Float?
  price          Float
  stockStatus    String
  imageUrl       String?

  lastSyncedAt   DateTime @default(now())
  createdAt      DateTime @default(now())
}
```

```

    updatedAt          DateTime @updatedAt

    @@unique([tenantId, doctorGreenId])
    @@index([tenantId])
}

8. Orders Table (Per Tenant)

model Order {
  id                String      @id @default(cuid())
  tenantId          String
  tenant            Tenant      @relation(fields: [tenantId], references: [id])
  customerId        String
  customer          Customer    @relation(fields: [customerId], references: [id])

  // Doctor Green Integration
  doctorGreenOrderId String?    @unique
  doctorGreenStatus  String?

  // Order Details
  orderNumber        String      @unique
  status             OrderStatus @default(PENDING)
  items              Json        // Order items
  totalAmount        Float

  // Fulfillment
  shippingAddress    Json
  trackingNumber      String?

  createdAt          DateTime    @default(now())
  updatedAt          DateTime    @updatedAt

  @@index([tenantId, customerId])
}

enum OrderStatus {
  PENDING
  CONFIRMED
  PROCESSING
  SHIPPED
  DELIVERED
  CANCELLED
}

```

---

## Template System Architecture

### Template Types (Initial Launch)

#### 1. Modern Template (Current HealingBuds Design)

- Full-width hero with overlay
- Card-based layout
- Green accent colors
- Animation-heavy

## 2. Minimal Template

- Clean, white space
- Simple typography
- Subtle colors
- Faster load times

## 3. Classic Template

- Traditional layout
- Sidebar navigation
- Professional, conservative

## Template Customization Options

Each tenant can customize: - **Colors:** Primary, secondary, accent, background - **Fonts:** Choose from 5-10 professional font pairings - **Logo:** Upload standard & white versions - **Hero Section:** Select layout style - **Content Sections:** Toggle visibility of sections - **Custom CSS:** Advanced users can add custom styling

## Template Storage

```
// Each tenant's customization stored in Tenant.customCSS
// Template components dynamically inject tenant's branding
```

---

## NFT Management System

### Super Admin NFT Dashboard

#### Features:

1. **NFT Inventory View**
  - Display all 200+ NFTs
  - Filter by status (Available, Assigned, Active, Suspended)
  - Search by token ID or wallet address
2. **NFT Assignment Flow**
  - Select available NFT
  - Create new tenant account
  - Link NFT to tenant
  - Change status to ASSIGNED
3. **NFT Verification**
  - Verify NFT ownership via Doctor Green API
  - Check NFT validity
  - Monitor expiry dates (if applicable)
4. **Bulk Operations**
  - Import NFTs from CSV/JSON
  - Bulk status updates
  - Export NFT reports

## NFT Verification Middleware

```
// Middleware to verify tenant's NFT is active before allowing access
async function verifyTenantNFT(tenantId: string) {
  const tenant = await prisma.tenant.findUnique({
    where: { id: tenantId },
    include: { nft: true }
  })
}
```

```

});

if (!tenant.nft || tenant.nft.status !== 'ACTIVE') {
  throw new Error('Invalid or inactive NFT license');
}

// Optional: Verify with Doctor Green API
const isValid = await doctorGreenAPI.verifyNFT(tenant.nft.tokenId);

return isValid;
}

```

---

## Domain & Routing Architecture

### Subdomain Strategy (Default)

```

platform.yourdomain.com → Super Admin Dashboard
healingbuds.yourdomain.com → HealingBuds Store
clientname.yourdomain.com → Client's Store

```

### Custom Domain Support

Tenants can configure: - **healingbuds.com** → Points to their storefront - DNS CNAME record:  
 healingbuds.com → proxy.yourdomain.com - SSL certificates auto-provisioned

### Implementation

```

// middleware.ts - Tenant routing
export function middleware(request: NextRequest) {
  const hostname = request.headers.get('host');

  // Super Admin
  if (hostname === 'platform.yourdomain.com') {
    return NextResponse.rewrite(new URL('/admin', request.url));
  }

  // Tenant Storefront
  const tenant = await getTenantByDomain(hostname);
  if (tenant) {
    request.headers.set('x-tenant-id', tenant.id);
    return NextResponse.rewrite(new URL('/storefront', request.url));
  }

  // 404
  return NextResponse.rewrite(new URL('/404', request.url));
}

```

---

## Tenant Onboarding Flow

### Step-by-Step Process

#### Step 1: Super Admin Assigns NFT

- Super admin selects available NFT from inventory
- Creates tenant record with status = PENDING\_SETUP
- Sends onboarding link to customer

#### Step 2: Customer Receives Onboarding Link

- Email with unique setup link
- Link expires in 7 days

#### Step 3: Account Setup

- Customer creates admin account (email/password)
- Enters business details:
  - Business name
  - Contact email & phone
  - Business address
  - License number
  - Tax ID

#### Step 4: Branding Customization

- Upload logo (standard & white versions)
- Select color scheme or customize colors
- Choose font pairing

#### Step 5: Template Selection

- Preview available templates
- Select preferred layout
- Customize homepage sections

#### Step 6: Domain Configuration

- Choose subdomain (e.g., healingbuds.yourdomain.com)
- OR configure custom domain (with instructions)

#### Step 7: Doctor Green API Setup

- Enter Doctor Green API credentials
- Test connection
- Sync initial product catalog

#### Step 8: Review & Launch

- Preview storefront
  - Click “Go Live”
  - Status changes to ACTIVE
  - NFT status changes to ACTIVE
-

## Doctor Green API Integration (Per Tenant)

### API Configuration Storage

Each tenant stores their own Doctor Green API credentials:

```
// Encrypted in database
tenant.doctorGreenApiKey = encrypt(apiKey);
tenant.doctorGreenConfig = {
  baseUrl: 'https://api.doctorgreen.com',
  webhookUrl: `https://yourdomain.com/api/webhooks/doctorgreen/${tenantId}`,
  syncInterval: '1h' // How often to sync products
};
```

### Product Synchronization

**Automated Sync Process:** 1. Cron job runs every hour (or tenant-configured interval) 2. For each active tenant: - Fetch products from Doctor Green API using tenant's credentials - Update Product table with latest data - Mark lastSyncedAt timestamp

**Manual Sync:** - Tenant admin can trigger manual sync from dashboard

### Order Flow

1. **Customer Places Order** on tenant's storefront
  2. **Create Order Record** in tenant's database
  3. **Submit to Doctor Green API** using tenant's credentials
  4. **Store doctorGreenOrderId** for tracking
  5. **Poll for Updates** or use webhooks to track order status
  6. **Update Customer** with tracking info
- 

## Super Admin Dashboard Features

### 1. Dashboard Home

- Total NFTs (Available, Assigned, Active)
- Total active tenants
- Total revenue this month
- Recent tenant activity

### 2. NFT Management

- **NFT Inventory Table**
  - Columns: Token ID, Status, Assigned To, Activation Date
  - Actions: View, Assign, Suspend, Reactivate
- **Add New NFT** (manual or bulk import)
- **NFT Analytics** (utilization rate, active vs available)

### 3. Tenant Management

- **Tenant List Table**
  - Columns: Name, Domain, NFT ID, Status, Created Date
  - Actions: View Details, Login As (impersonation), Suspend, Delete
- **Create New Tenant** (assign NFT)
- **Tenant Analytics** (orders, revenue, customers per tenant)

#### **4. Template Management**

- View all templates
- Create/edit templates
- Reorder templates
- Activate/deactivate templates

#### **5. Platform Settings**

- Global Doctor Green API settings (if applicable)
- Email configuration (SendGrid, etc.)
- Payment gateway settings (Stripe for SaaS billing)
- Domain & SSL settings

#### **6. Analytics & Reporting**

- Platform-wide metrics
  - Tenant performance comparison
  - Revenue reports
  - NFT utilization reports
- 

### **Tenant Admin Dashboard Features**

#### **1. Dashboard Home**

- Today's orders
- Total customers
- Total revenue
- Quick actions

#### **2. Storefront Customization**

- Branding settings (logo, colors)
- Template selection & preview
- Custom CSS editor (advanced)
- Domain configuration

#### **3. Product Management**

- View synced products from Doctor Green
- Toggle product visibility
- Add custom descriptions
- Reorder products
- Manual sync button

#### **4. Order Management**

- Order list with filters
- Order details view
- Update order status
- Track with Doctor Green

#### **5. Customer Management**

- Customer list

- Customer details
- Order history per customer

## 6. Consultation Management

- Consultation requests
- Approve/decline
- Schedule appointments

## 7. Blog Management (if enabled)

- Create/edit blog posts
- Manage categories

## 8. Analytics

- Sales reports
- Customer insights
- Popular products

## 9. Settings

- Business details
- Team members (if multi-user)
- Notification preferences
- API integration status

---

## Project Structure

```

healingbuds_website/
  nextjs_space/
    app/
      (admin)/                                # Super Admin Routes
        admin/
          layout.tsx
          page.tsx                            # Admin Dashboard
          nfts/                               # NFT Management
          tenants/                           # Tenant Management
          templates/                         # Template Management
          settings/
        api/
          admin/                             # Admin API routes

      (onboarding)/                          # Tenant Onboarding
        onboarding/
          layout.tsx
          [token]/
            page.tsx                          # Step 1: Account Setup
            branding/                         # Step 2: Branding
            template/                         # Step 3: Template
            domain/                           # Step 4: Domain
            api/                              # Step 5: API Setup
            launch/                           # Step 6: Review & Launch

```

```

(tenant-admin)/          # Tenant Admin Dashboard
  dashboard/
    layout.tsx
    page.tsx
    products/
    orders/
    customers/
    consultations/
    blog/
    customization/
    settings/
  api/
    tenant/              # Tenant API routes

(storefront)/            # Public Storefront (Multi-tenant)
  layout.tsx              # Dynamic layout per tenant
  page.tsx                # Home (template-driven)
  about/
  products/
  consultation/
  blog/
  contact/
  auth/                  # Customer auth

api/
  admin/                 # Super Admin APIs
  tenant/                # Tenant Admin APIs
  storefront/            # Public storefront APIs
  webhooks/
    doctorgreen/         # Doctor Green webhooks
  cron/
    sync-products/       # Product sync cron

middleware.ts            # Tenant routing & auth
...

components/
  admin/                 # Super Admin components
  tenant-admin/          # Tenant Admin components
  storefront/            # Storefront components
  templates/             # Template components
    modern/
    minimal/
    classic/
  shared/                # Shared UI components

lib/
  auth/
    super-admin-auth.ts
    tenant-auth.ts
    customer-auth.ts
  tenant-context.tsx     # React context for current tenant
  nft-service.ts         # NFT operations
  doctor-green-api.ts    # Doctor Green integration

```

```
    template-engine.ts      # Template rendering
    ...

    prisma/
      schema.prisma         # Multi-tenant schema

    SAAS_ARCHITECTURE_PLAN.md      # This document
```

---

## Development Phases

### Phase 1: Foundation & Super Admin (Weeks 1-3)

**Goal:** Build super admin platform to manage NFTs and tenants

#### Week 1: Database & Auth

- ☐ Design & implement multi-tenant Prisma schema
- ☐ Set up separate auth systems (SuperAdmin, TenantUser, Customer)
- ☐ Implement tenant context/middleware for domain routing
- ☐ Set up S3 cloud storage for logos/images

#### Week 2: Super Admin Dashboard

- ☐ Build super admin authentication
- ☐ Create NFT management interface:
  - ☐ NFT inventory table
  - ☐ Add/import NFTs
  - ☐ Assign NFT to tenant
  - ☐ Enable/disable NFTs
- ☐ Create tenant management interface:
  - ☐ Tenant list
  - ☐ Tenant details view
  - ☐ Suspend/activate tenant
  - ☐ Login as tenant (impersonation)

#### Week 3: NFT Integration & Testing

- ☐ Integrate with Doctor Green NFT verification API
  - ☐ Build NFT verification middleware
  - ☐ Create analytics dashboard for super admin
  - ☐ Seed database with initial NFTs
  - ☐ Test NFT assignment flow
- 

### Phase 2: Tenant Onboarding & Template System (Weeks 4-6)

#### Week 4: Onboarding Flow

- ☐ Build onboarding wizard (6 steps)
  - ☐ Account setup
  - ☐ Business details
  - ☐ Branding upload (logo, colors)
  - ☐ Template selection
  - ☐ Domain configuration

- ☐ API setup (Doctor Green)
- ☐ Email onboarding link generation
- ☐ Onboarding progress tracking

### **Week 5: Template System**

- ☐ Extract current HealingBuds design as “Modern” template
- ☐ Create “Minimal” template
- ☐ Create “Classic” template
- ☐ Build template preview system
- ☐ Implement dynamic theming (colors, fonts, logo injection)
- ☐ Create template management UI (super admin)

### **Week 6: Domain & Tenant Activation**

- ☐ Implement subdomain routing
  - ☐ Add custom domain configuration
  - ☐ SSL certificate automation
  - ☐ Tenant activation workflow
  - ☐ Test full onboarding → activation flow
- 

## **Phase 3: Tenant Admin Dashboard (Weeks 7-9)**

### **Week 7: Core Dashboard**

- ☐ Build tenant admin authentication
- ☐ Create dashboard home page
- ☐ Implement storefront customization UI
  - ☐ Logo upload
  - ☐ Color picker
  - ☐ Template switcher
  - ☐ Custom CSS editor
- ☐ Live preview system

### **Week 8: Product & Order Management**

- ☐ Build Doctor Green API integration layer
- ☐ Create product sync system (manual + cron)
- ☐ Build product management UI
  - ☐ Product list
  - ☐ Toggle visibility
  - ☐ Custom descriptions
- ☐ Create order management UI
  - ☐ Order list
  - ☐ Order details
  - ☐ Status updates

### **Week 9: Additional Features**

- ☐ Customer management UI
- ☐ Consultation management UI
- ☐ Blog management UI (optional)
- ☐ Analytics dashboard
- ☐ Settings page (business details, team members)

---

## Phase 4: Storefront Implementation (Weeks 10-12)

### Week 10: Core Storefront

- ☐ Convert existing HealingBuds storefront to multi-tenant
- ☐ Implement tenant detection (domain → tenant lookup)
- ☐ Apply tenant branding dynamically
- ☐ Build customer authentication
- ☐ Implement customer registration

### Week 11: Shopping & Orders

- ☐ Build product catalog (tenant-specific)
- ☐ Create product detail pages
- ☐ Implement shopping cart
- ☐ Build checkout flow
- ☐ Submit orders to Doctor Green API per tenant

### Week 12: Additional Features

- ☐ Consultation booking system
  - ☐ Blog system (tenant-specific posts)
  - ☐ Customer profile & order history
  - ☐ Contact forms
  - ☐ SEO optimization per tenant
- 

## Phase 5: Polish & Launch (Weeks 13-14)

### Week 13: Testing & Optimization

- ☐ End-to-end testing (all user roles)
- ☐ Performance optimization
- ☐ Security audit
- ☐ Mobile responsiveness
- ☐ Cross-browser testing

### Week 14: Launch Preparation

- ☐ Migrate HealingBuds to first tenant
  - ☐ Create marketing materials for SaaS
  - ☐ Documentation for tenants
  - ☐ Deploy to production
  - ☐ Monitor & fix issues
- 

## Security Considerations

### 1. Data Isolation

- Strict tenant ID filtering in all queries
- Row-level security policies
- Prevent cross-tenant data leaks

## 2. Authentication

- Separate auth systems for super admin, tenant users, customers
- JWT tokens with tenant ID claim
- Role-based access control (RBAC)

## 3. API Security

- Rate limiting per tenant
- Encrypt Doctor Green API keys at rest
- HTTPS only
- CORS policies

## 4. NFT Security

- Verify NFT ownership periodically
- Audit logs for NFT status changes
- Prevent NFT reassignment without super admin approval

---

## Pricing Model (Future Consideration)

### SaaS Subscription Options

#### Option 1: NFT Purchase Only

- One-time NFT purchase
- Full platform access forever
- Pay for Doctor Green orders separately

#### Option 2: NFT + Monthly Fee

- Lower upfront NFT cost
- Monthly SaaS fee (e.g., €49/month)
- Includes platform features, hosting, support

#### Option 3: Revenue Share

- Free or low-cost NFT
- Platform takes 5-10% of each order
- Automated commission deduction

---

## Success Metrics

### Key Performance Indicators (KPIs)

**For Super Admin:** - Total active tenants - NFT utilization rate - Platform revenue - Average tenant lifetime value - Churn rate

**For Tenants:** - Orders per month - Average order value - Customer retention rate - Product sync reliability - Page load speed

---

## Technology Stack

### Core Technologies

- **Framework:** Next.js 14 (App Router)
- **Database:** PostgreSQL (with Prisma ORM)
- **Authentication:** NextAuth.js (multi-provider)
- **File Storage:** AWS S3 (via cloud storage)
- **Styling:** Tailwind CSS + shadcn/ui
- **API Integration:** Doctor Green API
- **Deployment:** Vercel / AWS (with subdomain support)
- **Monitoring:** Sentry (errors) + Vercel Analytics

### Additional Tools

- **Email:** SendGrid / Resend
  - **Payments:** Stripe (for SaaS billing, if needed)
  - **Cron Jobs:** Vercel Cron / Inngest
  - **Domain Management:** Vercel Domains / Cloudflare
- 

## Next Steps

### Immediate Actions

1. **Review & Approve Plan**
    - Review this architecture document
    - Provide feedback on any sections
    - Confirm phased approach
  2. **Set Up Development Environment**
    - Set up multi-tenant database schema
    - Configure authentication systems
    - Initialize cloud storage
  3. **Begin Phase 1**
    - Start with super admin NFT management
    - Build foundation for multi-tenancy
- 

## Open Questions

1. **NFT Details**
  - What blockchain are the NFTs on? (Ethereum, Polygon, Solana?)
  - Do you have the NFT contract address and token IDs?
  - How do you verify NFT ownership? (via Doctor Green API or blockchain directly?)
2. **Doctor Green API**
  - Do all tenants use the same Doctor Green account, or separate accounts?
  - Does Doctor Green support per-tenant API keys?
  - Are there webhook options for order updates?
3. **Domain Strategy**
  - What will be your main domain? (e.g., medcannabis-platform.com)
  - Should subdomains be: `tenantname.yourdomain.com` or `app.yourdomain.com/tenantname`?
4. **Pricing Model**
  - What will you charge for the NFT?
  - Will there be ongoing subscription fees?
5. **Initial Launch**

- How many tenants do you plan to onboard initially?
  - Timeline for launch?
- 

## Conclusion

This SaaS platform will transform your medical cannabis NFT licenses into a scalable multi-tenant business. By treating HealingBuds as the first tenant, you'll validate the platform while building a productized solution for future customers.

**Ready to build?** Let's start with Phase 1 and create the foundation for your multi-tenant empire!