# **Dig|lit Launch Backlog**

## **Phase 0: Revenue Engine Deployment**

**Mission**: Launch revenue-ready platform in 7 days using zero-budget infrastructure.

**Tagline**: Your Vision + Our Mission = Hegemony

## **🎯 WEEK 0 OBJECTIVES**

**Primary Goal**: Live website accepting payments (USDT TRC20 + Stripe) with service listings and client acquisition funnel.

**Success Metrics**:

* ✅ Website live on custom domain
* ✅ Payment processing operational (crypto + fiat)
* ✅ 5 core services listed with pricing
* ✅ Lead capture system active
* ✅ First client acquisition capability

## **📅 DAY-BY-DAY EXECUTION**

### **DAY 1: Foundation & Infrastructure ⚙️**

#### **Morning (4 hours)**

**GitHub Repository Setup**

# Create main repository

mkdir diglit-platform && cd diglit-platform

git init

# Create 6-folder architecture

mkdir -p AI\_CORE FRONTEND BACKEND BUSINESS\_INTELLIGENCE VISUAL\_ENGINE Z

# Initialize Z-command tracking

touch Z/PROJECT\_TRACKER.md

touch Z/DAILY\_LOG.md

touch Z/CODE\_INDEX.md

**Repository Structure**:

diglit-platform/

├── AI\_CORE/ # AI engine modules

├── FRONTEND/ # React application

├── BACKEND/ # Supabase functions

├── BUSINESS\_INTELLIGENCE/# Analytics & data

├── VISUAL\_ENGINE/ # Design assets

├── Z/ # Command center

│ ├── PROJECT\_TRACKER.md

│ ├── DAILY\_LOG.md

│ └── CODE\_INDEX.md

└── README.md

#### **Afternoon (4 hours)**

**Development Environment Setup**

* Install Node.js (via nvm on Parrot OS)
* Install React + Vite: npm create vite@latest diglit-frontend -- --template react
* Configure Tailwind CSS
* Setup Supabase CLI
* Connect GitHub to Netlify

**Tools Installation**:

# Node Version Manager

curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/install.sh | bash

nvm install --lts

# Supabase CLI

npm install -g supabase

# Project initialization

cd FRONTEND

npm create vite@latest . -- --template react

npm install

npm install -D tailwindcss postcss autoprefixer

npx tailwindcss init -p

**Deliverables**:

* ✅ GitHub repo with 6-folder structure
* ✅ Development environment operational
* ✅ Netlify connected for auto-deployment

### **DAY 2: Payment Infrastructure 💰**

#### **Morning (4 hours)**

**Stripe Connect Integration**

1. **Create Stripe Account**: stripe.com/connect
2. **Setup Connect Platform**:  
   * Enable payment methods: Cards, Bank transfers, Wallets
   * Configure payout schedule: Daily automatic
   * Add bank account for deposits
3. **Install Stripe SDK**:

npm install @stripe/stripe-js @stripe/react-stripe-js

1. **Backend Payment API** (Supabase Edge Function):

// BACKEND/stripe-payment/index.ts

import Stripe from 'stripe';

const stripe = new Stripe(Deno.env.get('STRIPE\_SECRET\_KEY'));

export default async (req) => {

const { amount, currency, service } = await req.json();

const paymentIntent = await stripe.paymentIntents.create({

amount: amount \* 100, // Convert to cents

currency: currency || 'usd',

metadata: { service }

});

return new Response(JSON.stringify({

clientSecret: paymentIntent.client\_secret

}));

};

#### **Afternoon (4 hours)**

**USDT (TRC20) Integration**

1. **Setup TronWeb**:

npm install tronweb

1. **Crypto Payment Component**:

// FRONTEND/src/components/CryptoPayment.jsx

import TronWeb from 'tronweb';

const USDT\_CONTRACT = 'TR7NHqjeKQxGTCi8q8ZY4pL8otSzgjLj6t'; // TRC20 USDT

const RECEIVING\_WALLET = 'YOUR\_TRON\_WALLET\_ADDRESS';

export function CryptoPayment({ amount, orderId }) {

const payWithUSDT = async () => {

const tronWeb = window.tronWeb;

if (!tronWeb) {

alert('Please install TronLink wallet');

return;

}

const contract = await tronWeb.contract().at(USDT\_CONTRACT);

const amountSun = tronWeb.toSun(amount);

await contract.transfer(RECEIVING\_WALLET, amountSun).send();

// Verify transaction and update order status

};

return (

<button onClick={payWithUSDT}>

Pay {amount} USDT (TRC20)

</button>

);

}

1. **Transaction Verification Backend**:

// BACKEND/verify-crypto/index.ts

import TronWeb from 'tronweb';

const tronWeb = new TronWeb({

fullHost: 'https://api.trongrid.io'

});

export default async (req) => {

const { txHash, orderId } = await req.json();

const tx = await tronWeb.trx.getTransaction(txHash);

// Verify transaction details match order

const isValid = verifyTransaction(tx, orderId);

if (isValid) {

// Update order status in Supabase

// Release service/product

}

return new Response(JSON.stringify({ verified: isValid }));

};

**Deliverables**:

* ✅ Stripe Connect operational
* ✅ USDT TRC20 payment flow built
* ✅ Payment verification system active

### **DAY 3: Frontend - Landing Page 🎨**

#### **Morning (4 hours)**

**Hero Section & Navigation**

// FRONTEND/src/pages/Home.jsx

export function Home() {

return (

<div className="min-h-screen bg-gradient-to-br from-slate-900 via-purple-900 to-slate-900">

{/\* Navigation \*/}

<nav className="fixed w-full z-50 bg-black/20 backdrop-blur-lg border-b border-white/10">

<div className="max-w-7xl mx-auto px-4 py-4 flex justify-between items-center">

<div className="text-2xl font-bold text-white">

Dig<span className="text-purple-400">|</span>lit

</div>

<div className="space-x-8">

<a href="#services" className="text-white/80 hover:text-white">Services</a>

<a href="#shop" className="text-white/80 hover:text-white">Shop</a>

<a href="#contact" className="text-white/80 hover:text-white">Contact</a>

<button className="bg-purple-600 px-6 py-2 rounded-lg hover:bg-purple-500">

Get Started

</button>

</div>

</div>

</nav>

{/\* Hero Section \*/}

<section className="pt-32 pb-20 px-4">

<div className="max-w-7xl mx-auto text-center">

<h1 className="text-6xl md:text-8xl font-bold text-white mb-6">

Your Vision + Our Mission

</h1>

<p className="text-4xl md:text-5xl text-purple-400 font-light mb-8">

= Hegemony

</p>

<p className="text-xl text-white/70 max-w-3xl mx-auto mb-12">

We transform any vision into reality. From AI-powered virtual assistants

to complete digital transformation - we build empires on zero budget.

</p>

<div className="flex gap-4 justify-center">

<button className="bg-purple-600 px-8 py-4 rounded-lg text-lg hover:bg-purple-500">

Start Your Project

</button>

<button className="border border-white/30 px-8 py-4 rounded-lg text-lg hover:bg-white/10">

Explore Services

</button>

</div>

</div>

</section>

{/\* Trust Indicators \*/}

<section className="py-12 border-y border-white/10">

<div className="max-w-7xl mx-auto grid grid-cols-4 gap-8 text-center">

<div>

<div className="text-4xl font-bold text-purple-400">$0</div>

<div className="text-white/60">Upfront Cost</div>

</div>

<div>

<div className="text-4xl font-bold text-purple-400">24/7</div>

<div className="text-white/60">AI Support</div>

</div>

<div>

<div className="text-4xl font-bold text-purple-400">Global</div>

<div className="text-white/60">Operations</div>

</div>

<div>

<div className="text-4xl font-bold text-purple-400">100%</div>

<div className="text-white/60">Satisfaction</div>

</div>

</div>

</section>

</div>

);

}

#### **Afternoon (4 hours)**

**Services Showcase Section**

// FRONTEND/src/components/ServicesGrid.jsx

const services = [

{

id: 'va',

icon: '🤖',

title: 'AI Virtual Assistants',

description: 'Deploy intelligent VAs for any business function',

pricing: 'From $500/month',

features: ['24/7 Availability', 'Multi-language', 'Industry-specific']

},

{

id: 'transformation',

icon: '⚡',

title: 'Digital Transformation',

description: 'Complete enterprise modernization',

pricing: 'From $5,000/project',

features: ['AI Integration', 'Cloud Migration', 'Process Automation']

},

{

id: 'erp',

icon: '📱',

title: 'Palm ERP',

description: 'Modular ERP in your palm',

pricing: 'From $999/month',

features: ['Mobile-first', 'AI-powered', 'Quantum-secure']

},

{

id: 'consulting',

icon: '🎯',

title: 'Business Consulting',

description: 'Strategy to execution',

pricing: 'Custom pricing',

features: ['Growth Strategy', 'Market Analysis', 'Operations Design']

},

{

id: 'development',

icon: '💻',

title: 'Custom Development',

description: 'Build anything from scratch',

pricing: 'From $10,000',

features: ['Full-stack', 'AI-native', 'Scalable architecture']

},

{

id: 'security',

icon: '🔒',

title: 'Cyber Security',

description: 'Quantum-resistant protection',

pricing: 'From $2,000/month',

features: ['Zero-trust', 'Post-quantum crypto', '24/7 monitoring']

}

];

export function ServicesGrid() {

return (

<section id="services" className="py-20 px-4 bg-black">

<div className="max-w-7xl mx-auto">

<h2 className="text-5xl font-bold text-white text-center mb-16">

Our Solutions

</h2>

<div className="grid md:grid-cols-3 gap-8">

{services.map(service => (

<div key={service.id}

className="bg-gradient-to-br from-purple-900/20 to-slate-900/20 border border-purple-500/20 rounded-xl p-8 hover:border-purple-500/50 transition-all">

<div className="text-6xl mb-4">{service.icon}</div>

<h3 className="text-2xl font-bold text-white mb-2">{service.title}</h3>

<p className="text-white/60 mb-4">{service.description}</p>

<div className="text-purple-400 font-semibold mb-4">{service.pricing}</div>

<ul className="space-y-2 mb-6">

{service.features.map((feature, i) => (

<li key={i} className="text-white/70 text-sm">✓ {feature}</li>

))}

</ul>

<button className="w-full bg-purple-600 py-3 rounded-lg hover:bg-purple-500">

Get Started

</button>

</div>

))}

</div>

</div>

</section>

);

}

**Deliverables**:

* ✅ Professional landing page
* ✅ Services showcase with pricing
* ✅ Mobile responsive design

### **DAY 4: E-Commerce Shop 🛒**

#### **Morning (4 hours)**

**Product Catalog System**

// BACKEND/schema.sql (Supabase)

CREATE TABLE products (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

name TEXT NOT NULL,

description TEXT,

price DECIMAL(10,2) NOT NULL,

currency TEXT DEFAULT 'USD',

category TEXT,

image\_url TEXT,

stock\_quantity INTEGER DEFAULT 0,

is\_digital BOOLEAN DEFAULT false,

created\_at TIMESTAMP DEFAULT NOW()

);

CREATE TABLE orders (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

customer\_email TEXT NOT NULL,

total\_amount DECIMAL(10,2) NOT NULL,

payment\_method TEXT, -- 'stripe', 'usdt', 'wire'

payment\_status TEXT DEFAULT 'pending',

transaction\_hash TEXT, -- For crypto payments

created\_at TIMESTAMP DEFAULT NOW()

);

CREATE TABLE order\_items (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

order\_id UUID REFERENCES orders(id),

product\_id UUID REFERENCES products(id),

quantity INTEGER NOT NULL,

unit\_price DECIMAL(10,2) NOT NULL

);

#### **Afternoon (4 hours)**

**Shop Frontend Component**

// FRONTEND/src/pages/Shop.jsx

import { useState, useEffect } from 'react';

import { supabase } from '../lib/supabase';

export function Shop() {

const [products, setProducts] = useState([]);

const [cart, setCart] = useState([]);

useEffect(() => {

loadProducts();

}, []);

const loadProducts = async () => {

const { data } = await supabase

.from('products')

.select('\*')

.order('created\_at', { ascending: false });

setProducts(data);

};

const addToCart = (product) => {

setCart([...cart, product]);

};

return (

<div className="min-h-screen bg-slate-900 py-20 px-4">

<div className="max-w-7xl mx-auto">

<h1 className="text-5xl font-bold text-white mb-12">Shop</h1>

<div className="grid md:grid-cols-4 gap-6">

{products.map(product => (

<div key={product.id}

className="bg-slate-800 rounded-lg overflow-hidden hover:transform hover:scale-105 transition-all">

<img src={product.image\_url}

alt={product.name}

className="w-full h-48 object-cover"/>

<div className="p-4">

<h3 className="text-white font-semibold mb-2">{product.name}</h3>

<p className="text-white/60 text-sm mb-4">{product.description}</p>

<div className="flex justify-between items-center">

<span className="text-purple-400 font-bold">

${product.price}

</span>

<button

onClick={() => addToCart(product)}

className="bg-purple-600 px-4 py-2 rounded hover:bg-purple-500">

Add to Cart

</button>

</div>

</div>

</div>

))}

</div>

{/\* Cart Summary \*/}

{cart.length > 0 && (

<div className="fixed bottom-4 right-4 bg-slate-800 p-6 rounded-lg shadow-2xl">

<h3 className="text-white font-bold mb-2">Cart ({cart.length})</h3>

<div className="text-purple-400 text-2xl font-bold mb-4">

${cart.reduce((sum, item) => sum + parseFloat(item.price), 0).toFixed(2)}

</div>

<button className="w-full bg-purple-600 py-3 rounded-lg hover:bg-purple-500">

Checkout

</button>

</div>

)}

</div>

</div>

);

}

**Deliverables**:

* ✅ Product catalog database
* ✅ Shopping interface
* ✅ Cart functionality

### **DAY 5: Client Portal & Dashboard 📊**

#### **Morning (4 hours)**

**Authentication System**

// FRONTEND/src/lib/supabase.js

import { createClient } from '@supabase/supabase-js';

const supabaseUrl = import.meta.env.VITE\_SUPABASE\_URL;

const supabaseKey = import.meta.env.VITE\_SUPABASE\_ANON\_KEY;

export const supabase = createClient(supabaseUrl, supabaseKey);

// Auth helpers

export const signUp = async (email, password) => {

const { data, error } = await supabase.auth.signUp({ email, password });

return { data, error };

};

export const signIn = async (email, password) => {

const { data, error } = await supabase.auth.signInWithPassword({ email, password });

return { data, error };

};

export const signOut = async () => {

const { error } = await supabase.auth.signOut();

return { error };

};

#### **Afternoon (4 hours)**

**Client Dashboard**

// FRONTEND/src/pages/Dashboard.jsx

import { useState, useEffect } from 'react';

import { supabase } from '../lib/supabase';

export function Dashboard() {

const [user, setUser] = useState(null);

const [projects, setProjects] = useState([]);

const [orders, setOrders] = useState([]);

useEffect(() => {

checkUser();

loadProjects();

loadOrders();

}, []);

const checkUser = async () => {

const { data: { user } } = await supabase.auth.getUser();

setUser(user);

};

return (

<div className="min-h-screen bg-slate-900 py-20 px-4">

<div className="max-w-7xl mx-auto">

<div className="flex justify-between items-center mb-12">

<h1 className="text-4xl font-bold text-white">Dashboard</h1>

<button className="bg-purple-600 px-6 py-3 rounded-lg">

New Project

</button>

</div>

{/\* Stats Grid \*/}

<div className="grid md:grid-cols-4 gap-6 mb-12">

<StatCard title="Active Projects" value="3" icon="🚀" />

<StatCard title="Total Spent" value="$12,450" icon="💰" />

<StatCard title="Services Used" value="7" icon="⚡" />

<StatCard title="Support Tickets" value="2" icon="🎫" />

</div>

{/\* Projects Table \*/}

<div className="bg-slate-800 rounded-lg p-6 mb-8">

<h2 className="text-2xl font-bold text-white mb-6">Active Projects</h2>

<table className="w-full">

<thead>

<tr className="text-left text-white/60 border-b border-white/10">

<th className="pb-4">Project</th>

<th className="pb-4">Service</th>

<th className="pb-4">Status</th>

<th className="pb-4">Progress</th>

<th className="pb-4">Actions</th>

</tr>

</thead>

<tbody className="text-white">

{projects.map(project => (

<tr key={project.id} className="border-b border-white/5">

<td className="py-4">{project.name}</td>

<td className="py-4">{project.service}</td>

<td className="py-4">

<span className="bg-green-500/20 text-green-400 px-3 py-1 rounded-full text-sm">

{project.status}

</span>

</td>

<td className="py-4">

<div className="w-32 h-2 bg-slate-700 rounded-full overflow-hidden">

<div className="h-full bg-purple-500"

style={{width: `${project.progress}%`}}/>

</div>

</td>

<td className="py-4">

<button className="text-purple-400 hover:text-purple-300">

View →

</button>

</td>

</tr>

))}

</tbody>

</table>

</div>

{/\* Orders History \*/}

<div className="bg-slate-800 rounded-lg p-6">

<h2 className="text-2xl font-bold text-white mb-6">Recent Orders</h2>

{/\* Orders list implementation \*/}

</div>

</div>

</div>

);

}

function StatCard({ title, value, icon }) {

return (

<div className="bg-slate-800 p-6 rounded-lg">

<div className="text-4xl mb-2">{icon}</div>

<div className="text-white/60 text-sm mb-1">{title}</div>

<div className="text-3xl font-bold text-white">{value}</div>

</div>

);

}

**Deliverables**:

* ✅ User authentication
* ✅ Client dashboard
* ✅ Project tracking interface

### **DAY 6: AI Chat Assistant & Lead Capture 🤖**

#### **Morning (4 hours)**

**AI Assistant Integration**

// FRONTEND/src/components/AIAssistant.jsx

import { useState } from 'react';

export function AIAssistant() {

const [isOpen, setIsOpen] = useState(false);

const [messages, setMessages] = useState([

{ role: 'assistant', content: 'Hello! I\'m Dig|lit AI. How can I help transform your vision today?' }

]);

const [input, setInput] = useState('');

const sendMessage = async () => {

if (!input.trim()) return;

const userMessage = { role: 'user', content: input };

setMessages([...messages, userMessage]);

setInput('');

// Call backend AI endpoint

const response = await fetch('YOUR\_SUPABASE\_FUNCTION\_URL/ai-chat', {

method: 'POST',

headers: { 'Content-Type': 'application/json' },

body: JSON.stringify({

messages: [...messages, userMessage]

})

});

const data = await response.json();

setMessages(prev => [...prev, {

role: 'assistant',

content: data.reply

}]);

// Capture lead if email detected

if (data.email) {

await captureLeadEmail(data.email);

}

};

return (

<>

{/\* Floating Chat Button \*/}

<button

onClick={() => setIsOpen(!isOpen)}

className="fixed bottom-6 right-6 w-16 h-16 bg-purple-600 rounded-full shadow-2xl flex items-center justify-center hover:bg-purple-500 z-50">

<span className="text-3xl">💬</span>

</button>

{/\* Chat Window \*/}

{isOpen && (

<div className="fixed bottom-24 right-6 w-96 h-[500px] bg-slate-800 rounded-2xl shadow-2xl flex flex-col z-50 border border-purple-500/30">

{/\* Header \*/}

<div className="p-4 border-b border-white/10 flex justify-between items-center">

<div>

<h3 className="text-white font-bold">Dig|lit AI</h3>

<p className="text-white/60 text-sm">Online now</p>

</div>

<button onClick={() => setIsOpen(false)} className="text-white/60 hover:text-white">

✕

</button>

</div>

{/\* Messages \*/}

<div className="flex-1 overflow-y-auto p-4 space-y-4">

{messages.map((msg, i) => (

<div key={i} className={`flex ${msg.role === 'user' ? 'justify-end' : 'justify-start'}`}>

<div className={`max-w-[80%] p-3 rounded-lg ${

msg.role === 'user'

? 'bg-purple-600 text-white'

: 'bg-slate-700 text-white'

}`}>

{msg.content}

</div>

</div>

))}

</div>

{/\* Input \*/}

<div className="p-4 border-t border-white/10">

<div className="flex gap-2">

<input

type="text"

value={input}

onChange={(e) => setInput(e.target.value)}

onKeyPress={(e) => e.key === 'Enter' && sendMessage()}

placeholder="Type your message..."

className="flex-1 bg-slate-700 text-white px-4 py-2 rounded-lg focus:outline-none focus:ring-2 focus:ring-purple-500"

/>

<button

onClick={sendMessage}

className="bg-purple-600 px-6 py-2 rounded-lg hover:bg-purple-500">

Send

</button>

</div>

</div>

</div>

)}

</>

);

}

#### **Afternoon (4 hours)**

**Lead Capture Backend**

// BACKEND/ai-chat/index.ts

import { OpenAI } from 'openai';

import { createClient } from '@supabase/supabase-js';

const openai = new OpenAI({ apiKey: Deno.env.get('OPENAI\_API\_KEY') });

const supabase = createClient(

Deno.env.get('SUPABASE\_URL'),

Deno.env.get('SUPABASE\_SERVICE\_KEY')

);

export default async (req) => {

const { messages } = await req.json();

// System prompt for Dig|lit AI

const systemPrompt = `You are Dig|lit AI Assistant. Your role:

1. Understand the user's business vision or problem

2. Recommend appropriate Dig|lit services

3. Capture contact information naturally

4. Qualify leads based on project scope

5. Schedule consultations when ready

Services to recommend:

- Virtual Assistants ($500-2000/mo)

- Digital Transformation ($5K-50K)

- Palm ERP ($999/mo)

- Custom Development ($10K+)

- Business Consulting (Custom)

Be conversational, enthusiastic, and solution-focused.`;

const completion = await openai.chat.completions.create({

model: 'gpt-4',

messages: [

{ role: 'system', content: systemPrompt },

...messages

]

});

const reply = completion.choices[0].message.content;

// Detect email in conversation

const emailRegex = /[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}/;

const emailMatch = messages[messages.length - 1].content.match(emailRegex);

if (emailMatch) {

// Save lead to database

await supabase.from('leads').insert({

email: emailMatch[0],

conversation: messages,

status: 'new',

source: 'ai\_chat'

});

}

return new Response(JSON.stringify({

reply,

email: emailMatch ? emailMatch[0] : null

}));

};

**Lead Database Schema**:

CREATE TABLE leads (

id UUID PRIMARY KEY DEFAULT uuid\_generate\_v4(),

email TEXT NOT NULL,

name TEXT,

phone TEXT,

company TEXT,

conversation JSONB,

interested\_services TEXT[],

project\_budget TEXT,

status TEXT DEFAULT 'new',

source TEXT,

created\_at TIMESTAMP DEFAULT NOW()

);

**Deliverables**:

* ✅ AI chat assistant
* ✅ Lead capture system
* ✅ Conversation tracking

### **DAY 7: Testing, Deployment & Launch 🚀**

#### **Morning (4 hours)**

**Quality Assurance Checklist**

# Pre-Launch QA Checklist

## Functionality Tests

- [ ] Homepage loads correctly

- [ ] All navigation links work

- [ ] Services page displays all offerings

- [ ] Shop shows products from database

- [ ] Cart add/remove functions

- [ ] Stripe payment processes

- [ ] USDT payment shows correct wallet

- [ ] User registration works

- [ ] Login/logout functions

- [ ] Dashboard loads user data

- [ ] AI chat responds correctly

- [ ] Lead capture saves to database

## Cross-Browser Testing

- [ ] Chrome (desktop & mobile)

- [ ] Firefox

- [ ] Safari (iOS)

- [ ] Edge

## Performance

- [ ] Page load < 3 seconds

- [ ] Images optimized

- [ ] Lighthouse score > 90

## Security

- [ ] HTTPS enabled

- [ ] API keys in environment variables

- [ ] Supabase RLS policies active

- [ ] Input validation on all forms

- [ ] SQL injection protection

- [ ] XSS protection enabled

- [ ] CORS configured properly

## Mobile Responsiveness

- [ ] All pages responsive

- [ ] Touch interactions work

- [ ] Mobile navigation functional

- [ ] Payment flows mobile-optimized

## Payment Testing

- [ ] Stripe test mode transactions

- [ ] USDT wallet address verified

- [ ] Order confirmation emails

- [ ] Transaction logging works

#### **Afternoon (4 hours)**

**Final Deployment Steps**

**1. Environment Variables Setup**

# .env.production

VITE\_SUPABASE\_URL=https://your-project.supabase.co

VITE\_SUPABASE\_ANON\_KEY=your-anon-key

VITE\_STRIPE\_PUBLIC\_KEY=pk\_live\_your-key

VITE\_TRON\_WALLET=YOUR\_TRON\_WALLET\_ADDRESS

# Supabase Edge Functions

OPENAI\_API\_KEY=sk-your-key

STRIPE\_SECRET\_KEY=sk\_live\_your-key

SUPABASE\_SERVICE\_KEY=your-service-key

**2. Build & Deploy**

# Build production bundle

cd FRONTEND

npm run build

# Deploy to Netlify (auto-deploy via GitHub)

git add .

git commit -m "Launch: Dig|lit Platform v1.0"

git push origin main

# Netlify will auto-deploy from main branch

**3. Domain Configuration**

# In Netlify Dashboard:

1. Add custom domain: diglit.com

2. Configure DNS:

- A Record: @ → 75.2.60.5

- CNAME: www → diglit.netlify.app

3. Enable HTTPS (auto SSL)

4. Set up redirects:

- /api/\* → Supabase Edge Functions

**4. Supabase Edge Functions Deployment**

# Deploy all backend functions

supabase functions deploy stripe-payment

supabase functions deploy verify-crypto

supabase functions deploy ai-chat

# Set secrets

supabase secrets set OPENAI\_API\_KEY=sk-...

supabase secrets set STRIPE\_SECRET\_KEY=sk\_live\_...

**5. Database Finalization**

-- Enable Row Level Security

ALTER TABLE products ENABLE ROW LEVEL SECURITY;

ALTER TABLE orders ENABLE ROW LEVEL SECURITY;

ALTER TABLE leads ENABLE ROW LEVEL SECURITY;

-- Public read access for products

CREATE POLICY "Products are viewable by everyone"

ON products FOR SELECT

USING (true);

-- Users can only see their own orders

CREATE POLICY "Users can view own orders"

ON orders FOR SELECT

USING (auth.uid()::text = customer\_id);

-- Only authenticated users can create orders

CREATE POLICY "Authenticated users can create orders"

ON orders FOR INSERT

WITH CHECK (auth.role() = 'authenticated');

**Deliverables**:

* ✅ Full QA completed
* ✅ Production deployment live
* ✅ Domain configured with SSL
* ✅ All backend functions operational

## **🎯 POST-LAUNCH: FIRST 24 HOURS**

### **Immediate Actions**

**1. Marketing Blitz**

# Launch Announcement Template

Subject: 🚀 Dig|lit is LIVE - Your Vision, Our Mission, Total Hegemony

We just launched Dig|lit - the world's first zero-budget enterprise

transformation platform.

✅ AI Virtual Assistants from $500/mo

✅ Complete Digital Transformation from $5K

✅ Palm ERP - Mobile-first, AI-powered

✅ Custom Development & Consulting

🎁 LAUNCH SPECIAL: First 10 clients get 30% off

Visit: diglit.com

Pay with: USDT (TRC20) or Credit Card

Transform your vision into reality TODAY.

**Distribution Channels**:

* LinkedIn (personal profile + company page)
* Twitter/X thread
* Reddit (r/entrepreneur, r/startups, r/digitalnomad)
* IndieHackers
* Product Hunt (schedule for launch)
* Hacker News (Show HN post)
* Email to existing network
* WhatsApp Business groups
* Telegram crypto/business communities

**2. Content Marketing**

# Blog Post: "How We Built a $100M Company on $0 Budget"

1. The Vision (Your story)

2. The Tech Stack (Open-source power)

3. The Architecture (6-folder system)

4. The AI Core (Secret sauce)

5. First Client Success (testimonial)

6. Join the Revolution (CTA)

**3. Lead Magnet** Create free resource to capture emails:

* "Complete Guide to Zero-Budget Digital Transformation"
* "AI Virtual Assistant Implementation Playbook"
* "ERP Selection Framework for Modern Businesses"

**4. Partnership Outreach** Target organizations that need services:

* Small business associations
* Startup incubators
* NGOs and non-profits
* Digital agencies (white-label services)
* Consulting firms (subcontracting)

## **📊 SUCCESS METRICS (Week 1)**

### **Target KPIs**

* **Website Traffic**: 1,000+ visitors
* **Lead Captures**: 50+ emails
* **Demo Requests**: 10+ qualified leads
* **Paid Consultations**: 3+ bookings ($500 each)
* **First Client**: 1 signed contract ($5K+)

### **Tracking Setup**

// Google Analytics 4

gtag('config', 'GA\_MEASUREMENT\_ID', {

'custom\_map': {

'dimension1': 'service\_interest',

'dimension2': 'traffic\_source'

}

});

// Track conversions

gtag('event', 'lead\_capture', {

'event\_category': 'engagement',

'event\_label': 'email\_signup'

});

gtag('event', 'demo\_request', {

'event\_category': 'conversion',

'event\_label': 'consultation\_booking'

});

## **🔥 REVENUE ACCELERATION STRATEGIES**

### **Week 1-2: Quick Wins**

**1. Virtual Assistant Package** ($500-2000/mo)

* Easiest to sell, fastest to deploy
* Target: Solo entrepreneurs, small teams
* Delivery: Mix AI tools + offshore talent
* Margin: 70%+

**2. Website/Landing Page Service** ($1,500-5,000)

* Quick turnaround (3-5 days)
* Use our proven templates
* Upsell to monthly maintenance
* Target: Local businesses, startups

**3. Business Consultation** ($500/session)

* 90-minute strategy calls
* Deliver actionable roadmap
* Convert to transformation projects
* Target: Established businesses

### **Month 1: Scale Services**

**4. Digital Transformation Projects** ($10K-50K)

* Full enterprise modernization
* 3-6 month engagements
* Recurring revenue potential
* Target: Mid-market companies

**5. Palm ERP Subscriptions** ($999-2,999/mo)

* SaaS model, recurring revenue
* High lifetime value
* Low marginal cost
* Target: SMBs, NGOs, logistics

**6. White-Label Services** (30-50% revenue share)

* Partner with agencies
* Fulfill their client work
* Scale without marketing
* Target: Digital agencies, consultancies

## **💰 PAYMENT OPTIMIZATION**

### **Multi-Currency Strategy**

**USDT (TRC20) - Crypto Native**

// Instant settlement, near-zero fees

Advantages:

- Global accessibility

- No chargebacks

- 24/7 processing

- Privacy-friendly

Target: Crypto enthusiasts, international clients, high-value deals

**Stripe Connect - Universal Fiat**

// 135+ currencies, bank deposits

Advantages:

- Credit cards, bank transfers

- Automatic conversion

- Fraud protection

- Professional appearance

Target: Traditional businesses, US/EU clients, recurring subscriptions

**Wire Transfer - Enterprise**

// Direct bank-to-bank

Advantages:

- High-value transactions

- Enterprise credibility

- No processing fees

- Clear audit trail

Target: Fortune 500, government, $50K+ projects

### **Payment Flow Optimization**

Client Selects Service

↓

AI Recommends Payment Method

↓

┌──────────────────────────┐

│ < $1K: Stripe (instant) │

│ $1-10K: USDT or Stripe │

│ $10K+: Wire or USDT │

└──────────────────────────┘

↓

Escrow/Smart Contract (optional)

↓

Milestone-based Release

↓

Auto-convert to Local Currency

↓

Bank Account Deposit

## **🛡️ RISK MITIGATION**

### **Common Launch Issues & Solutions**

**1. Payment Disputes**

* Solution: Milestone-based payments
* Tool: Smart contract escrow
* Documentation: Every interaction logged

**2. Service Delivery**

* Solution: Pre-built templates + SOPs
* Team: Virtual assistants on standby
* Backup: Network of freelancers

**3. Technical Failures**

* Solution: Multi-region deployment
* Monitoring: UptimeRobot + Sentry
* Backup: Static fallback pages

**4. Regulatory Compliance**

* KYC/AML: Automated with Sumsub
* Taxes: Stripe Tax calculation
* Legal: Terms generated by AI + lawyer review

## **📈 GROWTH ROADMAP (30-60-90 Days)**

### **Day 30 Goals**

* **Revenue**: $10,000+
* **Clients**: 5 active projects
* **Team**: 3 virtual assistants hired
* **Products**: 20+ items in shop
* **Traffic**: 5,000+ monthly visitors

### **Day 60 Goals**

* **Revenue**: $50,000+
* **Clients**: 15 active projects
* **Team**: 10 specialists (VAs, developers)
* **Partnerships**: 3 agency partnerships
* **MRR**: $10,000 from subscriptions

### **Day 90 Goals**

* **Revenue**: $100,000+
* **Clients**: 30+ active projects
* **Team**: 25+ distributed workforce
* **Products**: Palm ERP beta launched
* **Funding**: First investment/grant secured

## **🎓 LEARNING & ITERATION**

### **Daily Tracking (Z-Folder)**

**Z/DAILY\_LOG.md Template**

# Date: 2025-10-XX

## Completed Tasks

- [ ] Task 1

- [ ] Task 2

## Revenue Today

- Source 1: $XXX

- Source 2: $XXX

- Total: $XXX

## Lessons Learned

- Insight 1

- Challenge faced

- Solution implemented

## Tomorrow's Priorities

1. Priority 1

2. Priority 2

3. Priority 3

## Blockers

- Issue needing resolution

## Wins

- Celebration item

### **Weekly Review**

* Analyze conversion rates
* Identify top-performing services
* Adjust pricing based on demand
* Optimize marketing channels
* Update AI prompts based on conversations

## **🚨 CRITICAL SUCCESS FACTORS**

### **Non-Negotiables**

1. **Speed**: Respond to leads within 15 minutes
2. **Quality**: Every delivery exceeds expectations
3. **Documentation**: Everything logged in Z-folder
4. **Cash Flow**: Never start work without payment
5. **Scalability**: Build systems, not just projects

### **The Dig|lit Promise**

Every vision deserves reality.

Every client becomes a case study.

Every project funds the next revolution.

Your hegemony is our mission.

## **🎯 NEXT STEPS AFTER LAUNCH**

**Hour 1-24**: Monitor, respond, iterate **Day 2-7**: Scale what works, kill what doesn't **Week 2-4**: Systematize delivery, hire team **Month 2-3**: Launch advanced products (ERP, AI agents) **Month 4-6**: Expand globally, raise capital **Month 7-12**: Achieve $1M revenue, prepare Series A

## **📞 EMERGENCY CONTACTS & RESOURCES**

### **Technical Support**

* Netlify Support: support.netlify.com
* Supabase Discord: supabase.com/discord
* Stripe Support: support.stripe.com

### **Community**

* Indie Hackers: indiehackers.com
* r/SaaS: reddit.com/r/SaaS
* Growth Hackers: growthhackers.com

### **Tools Cheatsheet**

# Quick commands

npm run dev # Local development

npm run build # Production build

git push origin main # Deploy to production

supabase db reset # Reset database

supabase functions deploy # Deploy all functions

# Check deployment

curl https://diglit.com/api/health

## **🏆 LAUNCH CHECKLIST (FINAL)**

### **Pre-Launch (Hour 0)**

* [ ] All code committed and pushed
* [ ] Environment variables verified
* [ ] Payment gateways tested
* [ ] Domain DNS propagated
* [ ] SSL certificate active
* [ ] Analytics tracking verified
* [ ] AI assistant responding correctly
* [ ] Email notifications working

### **Launch (Hour 1)**

* [ ] Tweet launch announcement
* [ ] Post on LinkedIn
* [ ] Submit to Product Hunt
* [ ] Email personal network
* [ ] Join relevant Discord/Slack communities
* [ ] Update GitHub README with live link
* [ ] Create Hacker News "Show HN" post

### **Post-Launch (Hour 2-24)**

* [ ] Monitor uptime
* [ ] Respond to all comments/messages
* [ ] Track first conversions
* [ ] Fix any reported bugs immediately
* [ ] Document feedback in Z-folder
* [ ] Celebrate first payment 🎉

## **💎 THE FOUNDATION IS LAID**

You now have:

* ✅ **Production website** accepting payments
* ✅ **Service catalog** with clear pricing
* ✅ **E-commerce shop** for products
* ✅ **Client portal** for project management
* ✅ **AI assistant** for lead capture
* ✅ **Payment infrastructure** (crypto + fiat)
* ✅ **Deployment pipeline** (GitHub → Netlify)
* ✅ **Backend services** (Supabase + Edge Functions)

**The machine is OPERATIONAL.**

Now execute the marketing plan, land the first clients, and let revenue fuel the expansion.

Your $100 trillion empire starts with the first $100.

**Dig|lit is LIVE. The hegemony begins NOW.** 🚀

## **📚 APPENDIX: QUICK REFERENCE**

### **Essential Commands**

# Development

npm run dev

# Build

npm run build

# Deploy functions

supabase functions deploy [function-name]

# Database migrations

supabase db push

# View logs

netlify logs

supabase functions logs [function-name]

# Environment

netlify env:set KEY value

### **Support Resources**

* Documentation: /Z/README.md
* API Reference: /Z/API\_DOCS.md
* Troubleshooting: /Z/TROUBLESHOOTING.md
* Deployment Log: /Z/DEPLOYMENT\_LOG.md

**END OF 7-DAY LAUNCH BACKLOG**

*Next Document: Payment Infrastructure & Revenue Model (Building Now...)*