# **Dig|lit Web Development Masterplan v2.0**

## **Modular Architecture: Frontend-First Approach**

"Build in modules. Each piece works independently. Frontend leads, backend follows."

## **PART I: PROJECT STRUCTURE (The Foundation)**

### **Directory Architecture (Modular by Design)**

diglit-platform/

│

├── 📁 FRONTEND/ # Priority 1: User Interface

│ ├── public/ # Static assets

│ │ ├── favicon.ico

│ │ ├── logo.svg

│ │ ├── robots.txt

│ │ └── manifest.json # PWA manifest

│ │

│ ├── src/

│ │ ├── 📁 modules/ # Feature modules (self-contained)

│ │ │ ├── home/

│ │ │ │ ├── HomePage.tsx

│ │ │ │ ├── HeroSection.tsx

│ │ │ │ ├── FeaturesGrid.tsx

│ │ │ │ ├── home.module.css

│ │ │ │ └── home.test.tsx

│ │ │ │

│ │ │ ├── services/

│ │ │ │ ├── ServicesPage.tsx

│ │ │ │ ├── ServiceCard.tsx

│ │ │ │ ├── PricingTable.tsx

│ │ │ │ └── services.module.css

│ │ │ │

│ │ │ ├── shop/

│ │ │ │ ├── ShopPage.tsx

│ │ │ │ ├── ProductGrid.tsx

│ │ │ │ ├── ProductCard.tsx

│ │ │ │ ├── Cart.tsx

│ │ │ │ ├── Checkout.tsx

│ │ │ │ └── shop.module.css

│ │ │ │

│ │ │ ├── dashboard/

│ │ │ │ ├── DashboardPage.tsx

│ │ │ │ ├── ProjectList.tsx

│ │ │ │ ├── OrderHistory.tsx

│ │ │ │ ├── Analytics.tsx

│ │ │ │ └── dashboard.module.css

│ │ │ │

│ │ │ ├── payments/

│ │ │ │ ├── PaymentPage.tsx

│ │ │ │ ├── StripeCheckout.tsx

│ │ │ │ ├── CryptoCheckout.tsx

│ │ │ │ ├── WireTransfer.tsx

│ │ │ │ └── payments.module.css

│ │ │ │

│ │ │ ├── auth/

│ │ │ │ ├── LoginPage.tsx

│ │ │ │ ├── RegisterPage.tsx

│ │ │ │ ├── ForgotPassword.tsx

│ │ │ │ ├── AuthContext.tsx

│ │ │ │ └── auth.module.css

│ │ │ │

│ │ │ └── ai-chat/

│ │ │ ├── AIChatWidget.tsx

│ │ │ ├── ChatBubble.tsx

│ │ │ ├── ChatInput.tsx

│ │ │ └── ai-chat.module.css

│ │ │

│ │ ├── 📁 components/ # Reusable UI components

│ │ │ ├── ui/ # Base components

│ │ │ │ ├── Button.tsx

│ │ │ │ ├── Input.tsx

│ │ │ │ ├── Card.tsx

│ │ │ │ ├── Modal.tsx

│ │ │ │ ├── Dropdown.tsx

│ │ │ │ ├── Spinner.tsx

│ │ │ │ └── Toast.tsx

│ │ │ │

│ │ │ ├── layout/ # Layout components

│ │ │ │ ├── Header.tsx

│ │ │ │ ├── Footer.tsx

│ │ │ │ ├── Sidebar.tsx

│ │ │ │ ├── Navigation.tsx

│ │ │ │ └── Container.tsx

│ │ │ │

│ │ │ └── forms/ # Form components

│ │ │ ├── FormInput.tsx

│ │ │ ├── FormSelect.tsx

│ │ │ ├── FormTextarea.tsx

│ │ │ └── FormCheckbox.tsx

│ │ │

│ │ ├── 📁 lib/ # Utilities & helpers

│ │ │ ├── api/ # API client

│ │ │ │ ├── client.ts

│ │ │ │ ├── auth.ts

│ │ │ │ ├── orders.ts

│ │ │ │ ├── payments.ts

│ │ │ │ └── products.ts

│ │ │ │

│ │ │ ├── utils/ # Helper functions

│ │ │ │ ├── format.ts # Date, currency formatting

│ │ │ │ ├── validation.ts # Form validation

│ │ │ │ ├── storage.ts # Local/session storage

│ │ │ │ └── analytics.ts # Tracking helpers

│ │ │ │

│ │ │ ├── hooks/ # Custom React hooks

│ │ │ │ ├── useAuth.ts

│ │ │ │ ├── useCart.ts

│ │ │ │ ├── useProducts.ts

│ │ │ │ ├── useOrders.ts

│ │ │ │ └── useLocalStorage.ts

│ │ │ │

│ │ │ └── constants/ # Constants & configs

│ │ │ ├── routes.ts

│ │ │ ├── api-endpoints.ts

│ │ │ └── theme.ts

│ │ │

│ │ ├── 📁 styles/ # Global styles

│ │ │ ├── globals.css # Global CSS

│ │ │ ├── variables.css # CSS custom properties

│ │ │ ├── animations.css # Keyframe animations

│ │ │ └── utilities.css # Utility classes

│ │ │

│ │ ├── 📁 types/ # TypeScript types

│ │ │ ├── user.ts

│ │ │ ├── product.ts

│ │ │ ├── order.ts

│ │ │ ├── payment.ts

│ │ │ └── api.ts

│ │ │

│ │ ├── App.tsx # Root component

│ │ ├── main.tsx # Entry point

│ │ ├── router.tsx # Route configuration

│ │ └── vite-env.d.ts # Vite types

│ │

│ ├── .env.example # Environment variables template

│ ├── .env.local # Local environment (gitignored)

│ ├── .gitignore

│ ├── index.html # HTML entry point

│ ├── package.json

│ ├── tsconfig.json # TypeScript config

│ ├── vite.config.ts # Vite config

│ ├── tailwind.config.js # Tailwind CSS config

│ └── README.md

│

├── 📁 BACKEND/ # Priority 2: Server Logic

│ ├── supabase/

│ │ ├── functions/ # Edge functions

│ │ │ ├── stripe-payment/

│ │ │ │ └── index.ts

│ │ │ ├── crypto-verify/

│ │ │ │ └── index.ts

│ │ │ ├── ai-chat/

│ │ │ │ └── index.ts

│ │ │ └── send-email/

│ │ │ └── index.ts

│ │ │

│ │ ├── migrations/ # Database migrations

│ │ │ ├── 20250101000000\_initial\_schema.sql

│ │ │ ├── 20250102000000\_add\_orders.sql

│ │ │ └── 20250103000000\_add\_payments.sql

│ │ │

│ │ └── config.toml # Supabase config

│ │

│ └── README.md

│

├── 📁 AI\_CORE/ # Priority 3: AI Features

│ ├── models/ # AI models

│ │ └── .gitkeep

│ ├── agents/ # AI agents

│ │ └── .gitkeep

│ └── README.md

│

├── 📁 BUSINESS\_INTELLIGENCE/ # Priority 4: Analytics

│ ├── dashboards/

│ │ └── .gitkeep

│ └── README.md

│

├── 📁 VISUAL\_ENGINE/ # Priority 5: Design Assets

│ ├── design-system/

│ │ └── .gitkeep

│ ├── 3d-assets/

│ │ └── .gitkeep

│ └── README.md

│

├── 📁 Z/ # Command Center

│ ├── PROJECT\_TRACKER.md # Master task list

│ ├── DAILY\_LOG.md # Daily progress

│ ├── CODE\_INDEX.md # Code organization map

│ ├── DEPLOYMENT\_LOG.md # Release history

│ └── TROUBLESHOOTING.md # Issue resolution

│

├── .gitignore # Git ignore rules

├── README.md # Project overview

└── package.json # Root package.json (workspaces)

## **PART II: INITIALIZATION COMMANDS**

### **Step 1: Create Directory Structure**

# Create root directory

mkdir diglit-platform && cd diglit-platform

# Initialize git

git init

# Create main folders

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

# Initialize Z-folder files

cd Z

touch PROJECT\_TRACKER.md DAILY\_LOG.md CODE\_INDEX.md DEPLOYMENT\_LOG.md TROUBLESHOOTING.md

cd ..

# Create root README

cat > README.md << 'EOF'

# Dig|lit Platform

> Your Vision + Our Mission = Hegemony

## Project Structure

- \*\*FRONTEND/\*\* - React + TypeScript web application

- \*\*BACKEND/\*\* - Supabase edge functions & database

- \*\*AI\_CORE/\*\* - AI models and agents

- \*\*BUSINESS\_INTELLIGENCE/\*\* - Analytics & reporting

- \*\*VISUAL\_ENGINE/\*\* - Design assets & 3D elements

- \*\*Z/\*\* - Command center & documentation

## Quick Start

```bash

# Install dependencies

cd FRONTEND && npm install

# Start development server

npm run dev

## **Documentation**

* [Architecture](https://claude.ai/chat/Z/CODE_INDEX.md)
* [Deployment](https://claude.ai/chat/Z/DEPLOYMENT_LOG.md)
* [Troubleshooting](https://claude.ai/chat/Z/TROUBLESHOOTING.md) EOF

---

### Step 2: Initialize Frontend (Vite + React + TypeScript)

```bash

# Navigate to FRONTEND

cd FRONTEND

# Create Vite project with React + TypeScript

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

# Install dependencies

npm install

# Install Tailwind CSS

npm install -D tailwindcss postcss autoprefixer

npx tailwindcss init -p

# Install additional dependencies

npm install react-router-dom

npm install @tanstack/react-query

npm install zustand

npm install clsx tailwind-merge

npm install lucide-react

# Install dev dependencies

npm install -D @types/node

### **Step 3: Configure Tailwind CSS**

# Update tailwind.config.js

cat > tailwind.config.js << 'EOF'

/\*\* @type {import('tailwindcss').Config} \*/

export default {

content: [

"./index.html",

"./src/\*\*/\*.{js,ts,jsx,tsx}",

],

theme: {

extend: {

colors: {

primary: {

50: '#faf5ff',

100: '#f3e8ff',

200: '#e9d5ff',

300: '#d8b4fe',

400: '#c084fc',

500: '#a855f7',

600: '#9333ea',

700: '#7e22ce',

800: '#6b21a8',

900: '#581c87',

},

dark: {

50: '#f8fafc',

100: '#f1f5f9',

200: '#e2e8f0',

300: '#cbd5e1',

400: '#94a3b8',

500: '#64748b',

600: '#475569',

700: '#334155',

800: '#1e293b',

900: '#0f172a',

},

},

fontFamily: {

sans: ['Inter', 'system-ui', 'sans-serif'],

mono: ['Fira Code', 'monospace'],

},

},

},

plugins: [],

}

EOF

# Create CSS custom properties

mkdir -p src/styles

cat > src/styles/variables.css << 'EOF'

:root {

/\* Colors \*/

--color-primary: #a855f7;

--color-primary-dark: #7e22ce;

--color-dark: #0f172a;

--color-dark-800: #1e293b;

--color-text: #ffffff;

--color-text-muted: rgba(255, 255, 255, 0.6);

/\* Spacing \*/

--space-xs: 0.25rem;

--space-sm: 0.5rem;

--space-md: 1rem;

--space-lg: 1.5rem;

--space-xl: 2rem;

--space-2xl: 3rem;

/\* Typography \*/

--font-size-xs: 0.75rem;

--font-size-sm: 0.875rem;

--font-size-base: 1rem;

--font-size-lg: 1.125rem;

--font-size-xl: 1.25rem;

--font-size-2xl: 1.5rem;

--font-size-3xl: 1.875rem;

--font-size-4xl: 2.25rem;

--font-size-5xl: 3rem;

/\* Shadows \*/

--shadow-sm: 0 1px 2px 0 rgb(0 0 0 / 0.05);

--shadow-md: 0 4px 6px -1px rgb(0 0 0 / 0.1);

--shadow-lg: 0 10px 15px -3px rgb(0 0 0 / 0.1);

--shadow-xl: 0 20px 25px -5px rgb(0 0 0 / 0.1);

/\* Border radius \*/

--radius-sm: 0.25rem;

--radius-md: 0.375rem;

--radius-lg: 0.5rem;

--radius-xl: 0.75rem;

--radius-2xl: 1rem;

/\* Transitions \*/

--transition-fast: 150ms cubic-bezier(0.4, 0, 0.2, 1);

--transition-base: 300ms cubic-bezier(0.4, 0, 0.2, 1);

--transition-slow: 500ms cubic-bezier(0.4, 0, 0.2, 1);

}

EOF

# Update main CSS

cat > src/index.css << 'EOF'

@import './styles/variables.css';

@tailwind base;

@tailwind components;

@tailwind utilities;

@layer base {

\* {

@apply border-border;

}

body {

@apply bg-dark-900 text-white font-sans antialiased;

}

h1 {

@apply text-4xl md:text-5xl lg:text-6xl font-bold;

}

h2 {

@apply text-3xl md:text-4xl font-bold;

}

h3 {

@apply text-2xl md:text-3xl font-semibold;

}

h4 {

@apply text-xl md:text-2xl font-semibold;

}

}

@layer components {

.btn {

@apply inline-flex items-center justify-center px-6 py-3 rounded-lg font-semibold transition-all duration-200;

}

.btn-primary {

@apply bg-primary-500 text-white hover:bg-primary-600 hover:shadow-lg hover:-translate-y-0.5;

}

.btn-secondary {

@apply border-2 border-white/20 text-white hover:bg-white/10 hover:border-white/30;

}

.card {

@apply bg-dark-800 border border-white/10 rounded-xl p-6;

}

.input {

@apply w-full px-4 py-3 bg-dark-800 border border-white/10 rounded-lg text-white placeholder:text-white/40 focus:outline-none focus:ring-2 focus:ring-primary-500 focus:border-transparent;

}

}

@layer utilities {

.text-balance {

text-wrap: balance;

}

}

EOF

### **Step 4: Create Folder Structure Inside Frontend**

# Create all necessary directories

cd src

mkdir -p modules/{home,services,shop,dashboard,payments,auth,ai-chat}

mkdir -p components/{ui,layout,forms}

mkdir -p lib/{api,utils,hooks,constants}

mkdir -p types

mkdir -p styles

# Create placeholder files

touch types/{user.ts,product.ts,order.ts,payment.ts,api.ts}

## **PART III: MODULAR COMPONENTS (Build Order)**

### **Module 1: Base UI Components (Foundation)**

#### **1.1 Button Component**

// src/components/ui/Button.tsx

import { ButtonHTMLAttributes, forwardRef } from 'react';

import { clsx } from 'clsx';

export interface ButtonProps extends ButtonHTMLAttributes<HTMLButtonElement> {

variant?: 'primary' | 'secondary' | 'outline' | 'ghost';

size?: 'sm' | 'md' | 'lg';

isLoading?: boolean;

}

export const Button = forwardRef<HTMLButtonElement, ButtonProps>(

({

className,

children,

variant = 'primary',

size = 'md',

isLoading = false,

disabled,

...props

}, ref) => {

return (

<button

ref={ref}

className={clsx(

'inline-flex items-center justify-center font-semibold rounded-lg transition-all duration-200',

'focus:outline-none focus:ring-2 focus:ring-primary-500 focus:ring-offset-2 focus:ring-offset-dark-900',

'disabled:opacity-50 disabled:cursor-not-allowed disabled:hover:transform-none',

{

// Variants

'bg-primary-500 text-white hover:bg-primary-600 hover:shadow-lg hover:-translate-y-0.5': variant === 'primary',

'bg-dark-800 text-white border border-white/20 hover:bg-dark-700': variant === 'secondary',

'border-2 border-white/20 text-white hover:bg-white/10 hover:border-white/30': variant === 'outline',

'text-white hover:bg-white/10': variant === 'ghost',

// Sizes

'px-4 py-2 text-sm': size === 'sm',

'px-6 py-3 text-base': size === 'md',

'px-8 py-4 text-lg': size === 'lg',

},

className

)}

disabled={disabled || isLoading}

{...props}

>

{isLoading ? (

<>

<svg

className="animate-spin -ml-1 mr-2 h-4 w-4"

xmlns="http://www.w3.org/2000/svg"

fill="none"

viewBox="0 0 24 24"

>

<circle

className="opacity-25"

cx="12"

cy="12"

r="10"

stroke="currentColor"

strokeWidth="4"

/>

<path

className="opacity-75"

fill="currentColor"

d="M4 12a8 8 0 018-8V0C5.373 0 0 5.373 0 12h4zm2 5.291A7.962 7.962 0 014 12H0c0 3.042 1.135 5.824 3 7.938l3-2.647z"

/>

</svg>

Loading...

</>

) : children}

</button>

);

}

);

Button.displayName = 'Button';

#### **1.2 Input Component**

// src/components/ui/Input.tsx

import { InputHTMLAttributes, forwardRef } from 'react';

import { clsx } from 'clsx';

export interface InputProps extends InputHTMLAttributes<HTMLInputElement> {

label?: string;

error?: string;

helperText?: string;

}

export const Input = forwardRef<HTMLInputElement, InputProps>(

({ className, label, error, helperText, ...props }, ref) => {

return (

<div className="w-full">

{label && (

<label className="block text-sm font-medium text-white/80 mb-2">

{label}

{props.required && <span className="text-red-400 ml-1">\*</span>}

</label>

)}

<input

ref={ref}

className={clsx(

'w-full px-4 py-3 bg-dark-800 border rounded-lg text-white placeholder:text-white/40',

'transition-colors duration-200',

'focus:outline-none focus:ring-2 focus:ring-primary-500 focus:border-transparent',

{

'border-white/10 hover:border-white/20': !error,

'border-red-500 focus:ring-red-500': error,

},

className

)}

{...props}

/>

{error && (

<p className="mt-2 text-sm text-red-400">{error}</p>

)}

{helperText && !error && (

<p className="mt-2 text-sm text-white/60">{helperText}</p>

)}

</div>

);

}

);

Input.displayName = 'Input';

#### **1.3 Card Component**

// src/components/ui/Card.tsx

import { HTMLAttributes, forwardRef } from 'react';

import { clsx } from 'clsx';

export interface CardProps extends HTMLAttributes<HTMLDivElement> {

hover?: boolean;

}

export const Card = forwardRef<HTMLDivElement, CardProps>(

({ className, hover = false, children, ...props }, ref) => {

return (

<div

ref={ref}

className={clsx(

'bg-dark-800 border border-white/10 rounded-xl p-6',

'transition-all duration-200',

{

'hover:border-primary-500/50 hover:shadow-lg hover:shadow-primary-500/10 hover:-translate-y-1': hover,

},

className

)}

{...props}

>

{children}

</div>

);

}

);

Card.displayName = 'Card';

#### **1.4 Modal Component**

// src/components/ui/Modal.tsx

import { ReactNode, useEffect } from 'react';

import { X } from 'lucide-react';

import { Button } from './Button';

export interface ModalProps {

isOpen: boolean;

onClose: () => void;

title?: string;

children: ReactNode;

size?: 'sm' | 'md' | 'lg' | 'xl';

}

export function Modal({ isOpen, onClose, title, children, size = 'md' }: ModalProps) {

// Close on Escape key

useEffect(() => {

const handleEscape = (e: KeyboardEvent) => {

if (e.key === 'Escape') onClose();

};

if (isOpen) {

document.addEventListener('keydown', handleEscape);

document.body.style.overflow = 'hidden';

}

return () => {

document.removeEventListener('keydown', handleEscape);

document.body.style.overflow = 'unset';

};

}, [isOpen, onClose]);

if (!isOpen) return null;

const sizeClasses = {

sm: 'max-w-md',

md: 'max-w-lg',

lg: 'max-w-2xl',

xl: 'max-w-4xl',

};

return (

<div

className="fixed inset-0 z-50 flex items-center justify-center p-4 bg-black/80 backdrop-blur-sm"

onClick={onClose}

>

<div

className={`bg-dark-800 border border-white/10 rounded-2xl shadow-2xl w-full ${sizeClasses[size]}`}

onClick={(e) => e.stopPropagation()}

>

{/\* Header \*/}

{title && (

<div className="flex items-center justify-between p-6 border-b border-white/10">

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

<button

onClick={onClose}

className="p-2 hover:bg-white/10 rounded-lg transition-colors"

>

<X className="w-5 h-5" />

</button>

</div>

)}

{/\* Content \*/}

<div className="p-6">

{children}

</div>

</div>

</div>

);

}

### **Module 2: Layout Components**

#### **2.1 Header/Navigation**

// src/components/layout/Header.tsx

import { useState } from 'react';

import { Link } from 'react-router-dom';

import { Menu, X } from 'lucide-react';

import { Button } from '../ui/Button';

export function Header() {

const [mobileMenuOpen, setMobileMenuOpen] = useState(false);

const navigation = [

{ name: 'Home', href: '/' },

{ name: 'Services', href: '/services' },

{ name: 'Shop', href: '/shop' },

{ name: 'About', href: '/about' },

{ name: 'Contact', href: '/contact' },

];

return (

<header className="fixed top-0 left-0 right-0 z-40 bg-dark-900/80 backdrop-blur-lg border-b border-white/10">

<nav className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

<div className="flex items-center justify-between h-20">

{/\* Logo \*/}

<Link to="/" className="flex items-center space-x-2">

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

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

</div>

</Link>

{/\* Desktop Navigation \*/}

<div className="hidden md:flex items-center space-x-8">

{navigation.map((item) => (

<Link

key={item.name}

to={item.href}

className="text-white/80 hover:text-white transition-colors font-medium"

>

{item.name}

</Link>

))}

</div>

{/\* CTA Buttons \*/}

<div className="hidden md:flex items-center space-x-4">

<Link to="/login">

<Button variant="ghost">Log In</Button>

</Link>

<Link to="/register">

<Button variant="primary">Get Started</Button>

</Link>

</div>

{/\* Mobile Menu Button \*/}

<button

className="md:hidden p-2 hover:bg-white/10 rounded-lg transition-colors"

onClick={() => setMobileMenuOpen(!mobileMenuOpen)}

>

{mobileMenuOpen ? (

<X className="w-6 h-6" />

) : (

<Menu className="w-6 h-6" />

)}

</button>

</div>

{/\* Mobile Menu \*/}

{mobileMenuOpen && (

<div className="md:hidden py-4 border-t border-white/10">

<div className="flex flex-col space-y-4">

{navigation.map((item) => (

<Link

key={item.name}

to={item.href}

className="text-white/80 hover:text-white transition-colors font-medium px-4 py-2"

onClick={() => setMobileMenuOpen(false)}

>

{item.name}

</Link>

))}

<div className="flex flex-col space-y-2 px-4 pt-4 border-t border-white/10">

<Link to="/login" onClick={() => setMobileMenuOpen(false)}>

<Button variant="ghost" className="w-full">Log In</Button>

</Link>

<Link to="/register" onClick={() => setMobileMenuOpen(false)}>

<Button variant="primary" className="w-full">Get Started</Button>

</Link>

</div>

</div>

</div>

)}

</nav>

</header>

);

}

#### **2.2 Footer**

// src/components/layout/Footer.tsx

import { Link } from 'react-router-dom';

import { Github, Twitter, Linkedin, Mail } from 'lucide-react';

export function Footer() {

const currentYear = new Date().getFullYear();

const footerLinks = {

product: [

{ name: 'Services', href: '/services' },

{ name: 'Shop', href: '/shop' },

{ name: 'Pricing', href: '/pricing' },

{ name: 'Roadmap', href: '/roadmap' },

],

company: [

{ name: 'About', href: '/about' },

{ name: 'Blog', href: '/blog' },

{ name: 'Careers', href: '/careers' },

{ name: 'Contact', href: '/contact' },

],

legal: [

{ name: 'Privacy', href: '/privacy' },

{ name: 'Terms', href: '/terms' },

{ name: 'Security', href: '/security' },

{ name: 'Compliance', href: '/compliance' },

],

social: [

{ name: 'GitHub', href: 'https://github.com/diglit', icon: Github },

{ name: 'Twitter', href: 'https://twitter.com/diglit', icon: Twitter },

{ name: 'LinkedIn', href: 'https://linkedin.com/company/diglit', icon: Linkedin },

{ name: 'Email', href: 'mailto:hello@diglit.com', icon: Mail },

],

};

return (

<footer className="bg-dark-900 border-t border-white/10">

<div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-12">

<div className="grid grid-cols-2 md:grid-cols-5 gap-8 mb-8">

{/\* Brand \*/}

<div className="col-span-2 md:col-span-1">

<Link to="/" className="text-2xl font-bold mb-4 block">

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

</Link>

<p className="text-white/60 text-sm mb-4">

Your Vision + Our Mission = Hegemony

</p>

<div className="flex space-x-4">

{footerLinks.social.map((item) => {

const Icon = item.icon;

return (

<a

key={item.name}

href={item.href}

target="\_blank"

rel="noopener noreferrer"

className="text-white/60 hover:text-primary-400 transition-colors"

>

<Icon className="w-5 h-5" />

</a>

);

})}

</div>

</div>

{/\* Product \*/}

<div>

<h3 className="font-semibold text-white mb-4">Product</h3>

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

{footerLinks.product.map((item) => (

<li key={item.name}>

<Link

to={item.href}

className="text-white/60 hover:text-white transition-colors text-sm"

>

{item.name}

</Link>

</li>

))}

</ul>

</div>

{/\* Company \*/}

<div>

<h3 className="font-semibold text-white mb-4">Company</h3>

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

{footerLinks.company.map((item) => (

<li key={item.name}>

<Link

to={item.href}

className="text-white/60 hover:text-white transition-colors text-sm"

>

{item.name}

</Link>

</li>

))}

</ul>

</div>

{/\* Legal \*/}

<div>

<h3 className="font-semibold text-white mb-4">Legal</h3>

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

{footerLinks.legal.map((item) => (

<li key={item.name}>

<Link

to={item.href}

className="text-white/60 hover:text-white transition-colors text-sm"

>

{item.name}

</Link>

</li>

))}

</ul>

</div>

{/\* Newsletter \*/}

<div>

<h3 className="font-semibold text-white mb-4">Newsletter</h3>

<form className="space-y-2">

<input

type="email"

placeholder="Enter your email"

className="w-full px-3 py-2 bg-dark-800 border border-white/10 rounded-lg text-sm text-white placeholder:text-white/40 focus:outline-none focus:ring-2 focus:ring-primary-500"

/>

<button

type="submit"

className="w-full px-3 py-2 bg-primary-500 text-white rounded-lg text-sm font-semibold hover:bg-primary-600 transition-colors"

>

Subscribe

</button>

</form>

</div>

</div>

{/\* Bottom Bar \*/}

<div className="pt-8 border-t border-white/10 flex flex-col md:flex-row justify-between items-center space-y-4 md:space-y-0">

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

© {currentYear} Dig|lit. All rights reserved.

</p>

<p className="text-white/40 text-xs">

Built with ❤️ using zero-budget infrastructure

</p>

</div>

</div>

</footer>

);

}

### **Module 3: Home Page (Complete MVP)**

#### **3.1 Home Page Main Component**

// src/modules/home/HomePage.tsx

import { HeroSection } from './HeroSection';

import { FeaturesGrid } from './FeaturesGrid';

import { ServicesPreview } from './ServicesPreview';

import { CTASection } from './CTASection';

import { StatsSection } from './StatsSection';

export function HomePage() {

return (

<div className="min-h-screen">

<HeroSection />

<StatsSection />

<FeaturesGrid />

<ServicesPreview />

<CTASection />

</div>

);

}

#### **3.2 Hero Section**

// src/modules/home/HeroSection.tsx

import { Link } from 'react-router-dom';

import { Button } from '../../components/ui/Button';

import { ArrowRight, Sparkles } from 'lucide-react';

export function HeroSection() {

return (

<section className="relative min-h-screen flex items-center justify-center overflow-hidden">

{/\* Animated Background \*/}

<div className="absolute inset-0 bg-gradient-to-br from-dark-900 via-primary-900/20 to-dark-900">

<div className="absolute inset-0 bg-[url('/grid.svg')] opacity-10" />

</div>

{/\* Floating Orbs \*/}

<div className="absolute top-20 left-20 w-72 h-72 bg-primary-500/30 rounded-full blur-3xl animate-pulse" />

<div className="absolute bottom-20 right-20 w-96 h-96 bg-purple-500/20 rounded-full blur-3xl animate-pulse delay-1000" />

{/\* Content \*/}

<div className="relative z-10 max-w-7xl mx-auto px-4 sm:px-6 lg:px-8 py-32 text-center">

{/\* Badge \*/}

<div className="inline-flex items-center space-x-2 bg-primary-500/10 border border-primary-500/20 rounded-full px-4 py-2 mb-8">

<Sparkles className="w-4 h-4 text-primary-400" />

<span className="text-sm text-primary-300 font-medium">

AI-Powered Business Solutions

</span>

</div>

{/\* Main Heading \*/}

<h1 className="text-5xl md:text-7xl lg:text-8xl font-bold mb-6 bg-gradient-to-r from-white via-primary-200 to-white bg-clip-text text-transparent">

Your Vision + Our Mission

</h1>

<p className="text-4xl md:text-5xl lg:text-6xl font-light text-primary-400 mb-8">

= Hegemony

</p>

{/\* Subheading \*/}

<p className="text-xl md:text-2xl text-white/70 max-w-3xl mx-auto mb-12">

Transform any vision into reality with AI-powered solutions.

From virtual assistants to complete digital transformation—

we build empires on zero budget.

</p>

{/\* CTA Buttons \*/}

<div className="flex flex-col sm:flex-row gap-4 justify-center items-center">

<Link to="/register">

<Button size="lg" className="group">

Start Your Project

<ArrowRight className="ml-2 w-5 h-5 group-hover:translate-x-1 transition-transform" />

</Button>

</Link>

<Link to="/services">

<Button variant="outline" size="lg">

Explore Services

</Button>

</Link>

</div>

{/\* Trust Indicators \*/}

<div className="mt-16 flex flex-wrap justify-center items-center gap-8 text-white/40 text-sm">

<div className="flex items-center space-x-2">

<span className="w-2 h-2 bg-green-400 rounded-full animate-pulse" />

<span>No upfront cost</span>

</div>

<div className="flex items-center space-x-2">

<span className="w-2 h-2 bg-green-400 rounded-full animate-pulse" />

<span>Pay with crypto or fiat</span>

</div>

<div className="flex items-center space-x-2">

<span className="w-2 h-2 bg-green-400 rounded-full animate-pulse" />

<span>24/7 AI support</span>

</div>

</div>

</div>

{/\* Scroll Indicator \*/}

<div className="absolute bottom-8 left-1/2 -translate-x-1/2 animate-bounce">

<div className="w-6 h-10 border-2 border-white/20 rounded-full flex items-start justify-center p-2">

<div className="w-1 h-3 bg-white/40 rounded-full" />

</div>

</div>

</section>

);

}

#### **3.3 Stats Section**

// src/modules/home/StatsSection.tsx

export function StatsSection() {

const stats = [

{ value: '$0', label: 'Upfront Cost', description: 'Start with zero investment' },

{ value: '24/7', label: 'AI Support', description: 'Always available assistance' },

{ value: 'Global', label: 'Operations', description: 'Serve clients worldwide' },

{ value: '100%', label: 'Satisfaction', description: 'Client success guarantee' },

];

return (

<section className="py-20 border-y border-white/10 bg-dark-800/50">

<div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

<div className="grid grid-cols-2 md:grid-cols-4 gap-8">

{stats.map((stat, index) => (

<div key={index} className="text-center">

<div className="text-4xl md:text-5xl font-bold text-primary-400 mb-2">

{stat.value}

</div>

<div className="text-white font-semibold mb-1">

{stat.label}

</div>

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

{stat.description}

</div>

</div>

))}

</div>

</div>

</section>

);

}

#### **3.4 Features Grid**

// src/modules/home/FeaturesGrid.tsx

import { Card } from '../../components/ui/Card';

import {

Zap, Shield, Globe, Brain, Clock, DollarSign,

Users, TrendingUp, Lock, Rocket

} from 'lucide-react';

export function FeaturesGrid() {

const features = [

{

icon: Brain,

title: 'AI-Native Solutions',

description: 'Built with artificial intelligence at the core, not bolted on as an afterthought.',

},

{

icon: Zap,

title: 'Lightning Fast',

description: 'Deploy in days, not months. Our modular architecture accelerates delivery.',

},

{

icon: DollarSign,

title: 'Zero Budget Friendly',

description: 'Start with no upfront costs. Scale as you grow with flexible payment options.',

},

{

icon: Shield,

title: 'Quantum-Resistant Security',

description: 'Future-proof encryption protecting your data from next-generation threats.',

},

{

icon: Globe,

title: 'Global Infrastructure',

description: 'Edge computing ensures <100ms latency anywhere in the world.',

},

{

icon: Clock,

title: '24/7 Availability',

description: 'AI assistants never sleep. Get support and answers anytime, anywhere.',

},

{

icon: Users,

title: 'White Glove Service',

description: 'Dedicated account managers who understand your business goals.',

},

{

icon: TrendingUp,

title: 'Scalable Architecture',

description: 'From 10 to 10 million users—infrastructure that grows with you.',

},

{

icon: Lock,

title: 'Data Sovereignty',

description: 'You own your data. Export anytime in open formats. No lock-in.',

},

{

icon: Rocket,

title: 'Continuous Innovation',

description: 'AI auto-updates your platform with latest features and optimizations.',

},

];

return (

<section className="py-20 bg-dark-900">

<div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

<div className="text-center mb-16">

<h2 className="text-4xl md:text-5xl font-bold mb-4">

Why Choose Dig|lit?

</h2>

<p className="text-xl text-white/60 max-w-3xl mx-auto">

We combine cutting-edge AI with battle-tested infrastructure to deliver

solutions that work today and scale for tomorrow.

</p>

</div>

<div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6">

{features.map((feature, index) => {

const Icon = feature.icon;

return (

<Card key={index} hover>

<div className="flex items-start space-x-4">

<div className="flex-shrink-0">

<div className="w-12 h-12 bg-primary-500/10 rounded-lg flex items-center justify-center">

<Icon className="w-6 h-6 text-primary-400" />

</div>

</div>

<div>

<h3 className="text-lg font-semibold text-white mb-2">

{feature.title}

</h3>

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

{feature.description}

</p>

</div>

</div>

</Card>

);

})}

</div>

</div>

</section>

);

}

#### **3.5 Services Preview**

// src/modules/home/ServicesPreview.tsx

import { Link } from 'react-router-dom';

import { Card } from '../../components/ui/Card';

import { Button } from '../../components/ui/Button';

import { ArrowRight } from 'lucide-react';

export function ServicesPreview() {

const services = [

{

emoji: '🤖',

title: 'AI Virtual Assistants',

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

price: 'From $500/month',

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

href: '/services/virtual-assistants',

},

{

emoji: '⚡',

title: 'Digital Transformation',

description: 'Complete enterprise modernization with AI',

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

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

href: '/services/digital-transformation',

},

{

emoji: '📱',

title: 'Palm ERP',

description: 'Modular ERP system in your palm',

price: 'From $999/month',

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

href: '/services/palm-erp',

},

{

emoji: '🎯',

title: 'Business Consulting',

description: 'Strategy to execution with AI guidance',

price: 'From $500/session',

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

href: '/services/consulting',

},

{

emoji: '💻',

title: 'Custom Development',

description: 'Build anything from scratch with AI acceleration',

price: 'From $10,000',

features: ['Full-stack Development', 'AI-native Architecture', 'Scalable Systems'],

href: '/services/custom-development',

},

{

emoji: '🔒',

title: 'Cyber Security',

description: 'Quantum-resistant protection for modern threats',

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

features: ['Zero-trust Architecture', 'Post-quantum Crypto', '24/7 Monitoring'],

href: '/services/security',

},

];

return (

<section className="py-20 bg-black">

<div className="max-w-7xl mx-auto px-4 sm:px-6 lg:px-8">

<div className="text-center mb-16">

<h2 className="text-4xl md:text-5xl font-bold mb-4">

Our Solutions

</h2>

<p className="text-xl text-white/60 max-w-3xl mx-auto">

Choose from our comprehensive suite of AI-powered services,

or combine them for a complete digital ecosystem.

</p>

</div>

<div className="grid md:grid-cols-2 lg:grid-cols-3 gap-6 mb-12">

{services.map((service, index) => (

<Card key={index} hover className="flex flex-col">

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

<h3 className="text-2xl font-bold text-white mb-2">

{service.title}

</h3>

<p className="text-white/60 mb-4 flex-grow">

{service.description}

</p>

<div className="text-primary-400 font-semibold text-lg mb-4">

{service.price}

</div>

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

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

<li key={i} className="text-white/70 text-sm flex items-start">

<span className="text-primary-400 mr-2">✓</span>

{feature}

</li>

))}

</ul>

<Link to={service.href} className="mt-auto">

<Button variant="primary" className="w-full group">

Learn More

<ArrowRight className="ml-2 w-4 h-4 group-hover:translate-x-1 transition-transform" />

</Button>

</Link>

</Card>

))}

</div>

<div className="text-center">

<Link to="/services">

<Button variant="outline" size="lg">

View All Services

</Button>

</Link>

</div>

</div>

</section>

);

}

#### **3.6 CTA Section**

// src/modules/home/CTASection.tsx

import { Link } from 'react-router-dom';

import { Button } from '../../components/ui/Button';

import { ArrowRight, CheckCircle } from 'lucide-react';

export function CTASection() {

const benefits = [

'Start with zero upfront investment',

'Pay with crypto (USDT) or credit card',

'Cancel anytime, no long-term contracts',

'Money-back guarantee if not satisfied',

];

return (

<section className="py-20 bg-gradient-to-br from-primary-900/20 via-dark-900 to-dark-900 relative overflow-hidden">

{/\* Background decoration \*/}

<div className="absolute inset-0 bg-[url('/grid.svg')] opacity-5" />

<div className="relative z-10 max-w-4xl mx-auto px-4 sm:px-6 lg:px-8 text-center">

<h2 className="text-4xl md:text-5xl font-bold mb-6">

Ready to Build Your Empire?

</h2>

<p className="text-xl text-white/60 mb-8">

Join hundreds of businesses transforming their vision into reality with AI-powered solutions.

</p>

<div className="grid md:grid-cols-2 gap-4 mb-12 text-left max-w-2xl mx-auto">

{benefits.map((benefit, index) => (

<div key={index} className="flex items-start space-x-3">

<CheckCircle className="w-5 h-5 text-primary-400 flex-shrink-0 mt-0.5" />

<span className="text-white/80">{benefit}</span>

</div>

))}

</div>

<div className="flex flex-col sm:flex-row gap-4 justify-center items-center">

<Link to="/register">

<Button size="lg" className="group">

Get Started Free

<ArrowRight className="ml-2 w-5 h-5 group-hover:translate-x-1 transition-transform" />

</Button>

</Link>

<Link to="/contact">

<Button variant="outline" size="lg">

Schedule a Demo

</Button>

</Link>

</div>

<p className="mt-8 text-sm text-white/40">

No credit card required • Free consultation • 30-day money-back guarantee

</p>

</div>

</section>

);

}

### **Module 4: Router Setup**

// src/router.tsx

import { createBrowserRouter } from 'react-router-dom';

import { RootLayout } from './components/layout/RootLayout';

import { HomePage } from './modules/home/HomePage';

// Lazy load other pages for code splitting

import { lazy } from 'react';

const ServicesPage = lazy(() => import('./modules/services/ServicesPage'));

const ShopPage = lazy(() => import('./modules/shop/ShopPage'));

const DashboardPage = lazy(() => import('./modules/dashboard/DashboardPage'));

const LoginPage = lazy(() => import('./modules/auth/LoginPage'));

const RegisterPage = lazy(() => import('./modules/auth/RegisterPage'));

export const router = createBrowserRouter([

{

path: '/',

element: <RootLayout />,

children: [

{

index: true,

element: <HomePage />,

},

{

path: 'services',

element: <ServicesPage />,

},

{

path: 'shop',

element: <ShopPage />,

},

{

path: 'dashboard',

element: <DashboardPage />,

},

{

path: 'login',

element: <LoginPage />,

},

{

path: 'register',

element: <RegisterPage />,

},

],

},

]);

// src/components/layout/RootLayout.tsx

import { Outlet } from 'react-router-dom';

import { Suspense } from 'react';

import { Header } from './Header';

import { Footer } from './Footer';

export function RootLayout() {

return (

<div className="min-h-screen flex flex-col">

<Header />

<main className="flex-grow pt-20">

<Suspense fallback={

<div className="min-h-screen flex items-center justify-center">

<div className="animate-spin rounded-full h-12 w-12 border-t-2 border-b-2 border-primary-500" />

</div>

}>

<Outlet />

</Suspense>

</main>

<Footer />

</div>

);

}

### **Module 5: App Entry Point**

// src/App.tsx

import { RouterProvider } from 'react-router-dom';

import { router } from './router';

export default function App() {

return <RouterProvider router={router} />;

}

// src/main.tsx

import React from 'react';

import ReactDOM from 'react-dom/client';

import App from './App.tsx';

import './index.css';

ReactDOM.createRoot(document.getElementById('root')!).render(

<React.StrictMode>

<App />

</React.StrictMode>,

);

## **PART IV: EXECUTION CHECKLIST**

### **Phase 1: Core Setup (Day 1)**

# ✅ TASK 1: Initialize Project Structure

cd ~/projects

mkdir diglit-platform && cd diglit-platform

git init

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

# ✅ TASK 2: Setup Frontend

cd FRONTEND

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

npm install

npm install -D tailwindcss postcss autoprefixer

npx tailwindcss init -p

# ✅ TASK 3: Install Dependencies

npm install react-router-dom @tanstack/react-query zustand

npm install clsx tailwind-merge lucide-react

npm install -D @types/node

# ✅ TASK 4: Create Directory Structure

cd src

mkdir -p modules/{home,services,shop,dashboard,payments,auth,ai-chat}

mkdir -p components/{ui,layout,forms}

mkdir -p lib/{api,utils,hooks,constants}

mkdir -p types styles

# ✅ TASK 5: Copy All Component Files

# Copy Button.tsx, Input.tsx, Card.tsx, Modal.tsx into components/ui/

# Copy Header.tsx, Footer.tsx, RootLayout.tsx into components/layout/

# Copy HomePage.tsx and sub-components into modules/home/

# Copy App.tsx, main.tsx, router.tsx to src/

# Update index.css with provided styles

# ✅ TASK 6: Start Development Server

npm run dev

# Open http://localhost:5173

### **Phase 2: Basic Pages (Day 2-3)**

// ✅ TASK 7: Services Page

// src/modules/services/ServicesPage.tsx

export default function ServicesPage() {

return (

<div className="min-h-screen py-20">

<div className="max-w-7xl mx-auto px-4">

<h1 className="text-5xl font-bold mb-8">Our Services</h1>

<p className="text-xl text-white/60">Detailed services catalog coming soon...</p>

</div>

</div>

);

}

// ✅ TASK 8: Shop Page

// src/modules/shop/ShopPage.tsx

export default function ShopPage() {

return (

<div className="min-h-screen py-20">

<div className="max-w-7xl mx-auto px-4">

<h1 className="text-5xl font-bold mb-8">Shop</h1>

<p className="text-xl text-white/60">Product catalog coming soon...</p>

</div>

</div>

);

}

// ✅ TASK 9: Auth Pages

// src/modules/auth/LoginPage.tsx

import { Input } from '../../components/ui/Input';

import { Button } from '../../components/ui/Button';

export default function LoginPage() {

return (

<div className="min-h-screen flex items-center justify-center py-20">

<div className="max-w-md w-full bg-dark-800 border border-white/10 rounded-xl p-8">

<h1 className="text-3xl font-bold mb-8 text-center">Log In</h1>

<form className="space-y-6">

<Input

label="Email"

type="email"

placeholder="you@example.com"

required

/>

<Input

label="Password"

type="password"

placeholder="••••••••"

required

/>

<Button type="submit" className="w-full">

Log In

</Button>

</form>

</div>

</div>

);

}

### **Phase 3: Deployment (Day 4)**

# ✅ TASK 10: Build for Production

npm run build

# ✅ TASK 11: Deploy to Netlify

# Option A: Netlify CLI

npm install -g netlify-cli

netlify login

netlify init

netlify deploy --prod

# Option B: Connect GitHub to Netlify

git add .

git commit -m "Initial commit: Dig|lit MVP"

git remote add origin https://github.com/yourusername/diglit-platform.git

git push -u origin main

# Then go to netlify.com:

# 1. New site from Git

# 2. Connect GitHub repo

# 3. Build command: cd FRONTEND