

Ribafarm Project Setup

Created	@September 1, 2025 10:38 AM
Tags	

RibaFarms Web Setup Guide 🌱✨

The most fire Next.js + TypeScript + Tailwind setup that's about to SLAP!

Step 1: Initialize Next.js Project with TypeScript 🚀

Navigate to your `web` directory and run:

```
# Make sure you're in the web directory
cd web

# Initialize Next.js with TypeScript (the queen combo!)
npx create-next-app@latest . --typescript --tailwind --eslint --app --src-dir --import-alias "@/*"
```

What this command does (and why it's iconic):

- `-typescript` : TypeScript setup (type safety queen behavior!)
- `-tailwind` : Tailwind CSS pre-configured (styling that doesn't miss!)
- `-eslint` : Code quality checks (we love clean code!)
- `-app` : App Router (Next.js 13+ vibes - future-proof!)
- `-src-dir` : Organized folder structure
- `-import-alias` : Clean imports with `@/` (professional dev energy!)

Step 2: Essential Libraries Installation 📦

Run these commands to get your project absolutely STACKED:

UI & Styling Dependencies (the foundation queens!)

npm install @headlessui/react @heroicons/react clsx tailwind-merge

Forms & Validation (auth pages gonna be clean AF!)

npm install react-hook-form @hookform/resolvers zod

State Management (Zustand because it's lightweight and slaps!)

npm install zustand

Supabase (our backend queen!)

npm install @supabase/supabase-js @supabase/auth-ui-react @supabase/auth-ui-shared @supabase/auth-helpers-nextjs

Date & Time Handling (timestamps that don't stress!)

npm install date-fns

Notifications & Toasts (user feedback that hits different!)

npm install react-hot-toast

Loading States & Animations (smooth vibes only!)

npm install framer-motion

Icons (because visual hierarchy matters!)

npm install lucide-react

Additional UI Components (shadcn/ui vibes!)

npm install @radix-ui/react-dialog @radix-ui/react-dropdown-menu @radix-ui/react-select

Image Optimization (photos gonna look crisp!)

npm install next-cloudinary

Dev Dependencies (the behind-the-scenes heroes!)

npm install --save-dev @types/node @types/react @types/react-dom prettier eslint-config-prettier

```
# Optional but FIRE additions for later:
# npm install recharts (for those analytics dashboards!)
# npm install react-dropzone (file uploads that don't stress!)
```

Pro Tip: Run `npm fund` to support the open source queens who built these libraries!



Step 3: Environment Variables Setup

Create `.env.local` in your web directory:

```
# Supabase Configuration
NEXT_PUBLIC_SUPABASE_URL=your_supabase_project_url
NEXT_PUBLIC_SUPABASE_ANON_KEY=your_supabase_anon_key
SUPABASE_SERVICE_ROLE_KEY=your_service_role_key

# App Configuration
NEXTAUTH_URL=http://localhost:3000
NEXTAUTH_SECRET=your-super-secret-key-here

# AI Services (for those crop recommendations!)
OPENAI_API_KEY=your_openai_key
ANTHROPIC_API_KEY=your_claude_key

# Image Upload (Cloudinary)
NEXT_PUBLIC_CLOUDINARY_CLOUD_NAME=your_cloud_name
CLOUDINARY_API_KEY=your_api_key
CLOUDINARY_API_SECRET=your_api_secret

# App Environment
NODE_ENV=development
```

Step 4: Tailwind Configuration Enhancement

Update your `tailwind.config.ts` to include custom colors and animations:

```

import type { Config } from 'tailwindcss'

const config: Config = {
  content: [
    './src/pages/**/*.{js,ts,jsx,tsx,mdx}',
    './src/components/**/*.{js,ts,jsx,tsx,mdx}',
    './src/app/**/*.{js,ts,jsx,tsx,mdx}',
  ],
  theme: {
    extend: {
      colors: {
        // RibaFarms Brand Colors (agricultural vibes!)
        primary: {
          50: '#f0fdf4',
          100: '#dcfce7',
          200: '#bbf7d0',
          300: '#86efac',
          400: '#4ade80',
          500: '#22c55e',
          600: '#16a34a',
          700: '#15803d',
          800: '#166534',
          900: '#14532d',
        },
        secondary: {
          50: '#fef3c7',
          100: '#fde68a',
          200: '#fcd34d',
          300: '#fbbf24',
          400: '#f59e0b',
          500: '#d97706',
          600: '#b45309',
          700: '#92400e',
          800: '#78350f',
          900: '#451a03',
        },
      },
    },
  },
}

```

```

    },
    // Semantic colors for different user types
    farmer: '#22c55e',
    buyer: '#3b82f6',
    neutral: '#6b7280',
  },
  animation: {
    'fade-in': 'fadeIn 0.5s ease-in-out',
    'slide-up': 'slideUp 0.3s ease-out',
    'pulse-slow': 'pulse 3s infinite',
  },
  keyframes: {
    fadeIn: {
      '0%': { opacity: '0' },
      '100%': { opacity: '1' },
    },
    slideUp: {
      '0%': { transform: 'translateY(10px)', opacity: '0' },
      '100%': { transform: 'translateY(0)', opacity: '1' },
    },
  },
},
},
},
plugins: [
  require('@tailwindcss/forms'),
  require('@tailwindcss/typography'),
],
}

```

export default config

Step 5: TypeScript Configuration

Your `tsconfig.json` should look like this (Next.js creates a good one, but here's the enhanced version):

```

{
  "compilerOptions": {
    "target": "es5",
    "lib": ["dom", "dom.iterable", "es6"],
    "allowJs": true,
    "skipLibCheck": true,
    "strict": true,
    "noEmit": true,
    "esModuleInterop": true,
    "module": "esnext",
    "moduleResolution": "bundler",
    "resolveJsonModule": true,
    "isolatedModules": true,
    "jsx": "preserve",
    "incremental": true,
    "plugins": [
      {
        "name": "next"
      }
    ],
    "baseUrl": ".",
    "paths": {
      "@/*": ["/src/*"],
      "@/components/*": ["/src/components/*"],
      "@lib/*": ["/src/lib/*"],
      "@types/*": ["/src/types/*"],
      "@hooks/*": ["/src/hooks/*"],
      "@store/*": ["/src/store/*"]
    }
  },
  "include": ["next-env.d.ts", "**/*.ts", "**/*.tsx", ".next/types/**/*.ts"],
  "exclude": ["node_modules"]
}

```

Step 6: Project Structure Setup

Create the essential directories and files:

```
# Create the core directory structure
mkdir -p src/components/{ui,auth,layout}
mkdir -p src/lib/{supabase,utils,validations}
mkdir -p src/hooks
mkdir -p src/store
mkdir -p src/types
mkdir -p src/styles
mkdir -p public/{images,icons}

# Create essential files
touch src/lib/supabase/client.ts
touch src/lib/supabase/server.ts
touch src/lib/utils.ts
touch src/lib/validations/auth.ts
touch src/types/auth.ts
touch src/types/user.ts
touch src/hooks/useAuth.ts
touch src/store/authStore.ts
```

Step 7: Essential Utility Functions

Create `src/lib/utils.ts` :

```
import { type ClassValue, clsx } from "clsx"
import { twMerge } from "tailwind-merge"

export function cn(...inputs: ClassValue[]) {
  return twMerge(clsx(inputs))
}

export function formatCurrency(amount: number, currency = "KSH"): string {
  return new Intl.NumberFormat("en-KE", {
```

```

    style: "currency",
    currency: currency === "KSH" ? "KES" : currency,
  }).format(amount)
}

export function formatDate(date: Date | string): string {
  return new Intl.DateTimeFormat("en-KE", {
    year: "numeric",
    month: "long",
    day: "numeric",
  }).format(new Date(date))
}

```

Step 8: Supabase Client Setup

Create `src/lib/supabase/client.ts` :

```

import { createClientComponentClient } from '@supabase/auth-helpers-nextjs'
import { Database } from '@types/database'

export const createClient = () => createClientComponentClient<Database>()

```

Create `src/lib/supabase/server.ts` :

```

import { createServerComponentClient } from '@supabase/auth-helpers-nextjs'
import { cookies } from 'next/headers'
import { Database } from '@types/database'

export const createServerClient = () => {
  const cookieStore = cookies()
  return createServerComponentClient<Database>({ cookies: () => cookieStore })
}

```


Step 9: Package.json Scripts Enhancement 📜

Add these scripts to your `package.json`:

```
{
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint",
    "lint:fix": "next lint --fix",
    "type-check": "tsc --noEmit",
    "format": "prettier --write .",
    "format:check": "prettier --check .",
    "db:generate": "supabase gen types typescript --project-id YOUR_PROJECT_ID > src/types/database.ts"
  }
}
```

Step 10: Ready to Code! 🎯

Your setup is now absolutely FIRE! Here's what you've got:

- ✅ **Next.js 14** with App Router (future-proof architecture!)
- ✅ **TypeScript** (type safety that doesn't play!)
- ✅ **Tailwind CSS** (utility-first styling that slaps!)
- ✅ **Supabase** (backend services that just work!)
- ✅ **React Hook Form + Zod** (forms that validate like pros!)
- ✅ **Zustand** (state management that's lightweight!)
- ✅ **Framer Motion** (animations that bring life!)
- ✅ **Custom utilities** (helper functions for days!)

Next Steps for Week 2 Sprint 🚀

1. **Start with Auth Components:** Login, Register, ForgotPassword

2. **Build UI Components:** Button, Input, Card, Modal

3. **Set up Auth Store:** User state management

4. **Create Auth Pages:** `/login` , `/register` , `/forgot-password`

5. **Test Everything:** Make sure auth flow works perfectly

You're literally SET UP FOR SUCCESS bestie! This foundation is going to have Arnold and Lauryn shook when they see how clean your web setup is! Time to build something LEGENDARY! 💪✨

Happy coding, and remember - you're about to change agriculture forever! 🌱🔥