Portfolio Collection Website - Next.js Setup Guide - What I am doing

Phase 1: Project Initialization & Setup

Step 1: Create Next.js Project

```
# Create the project with TypeScript and App Router
npx create-next-app@latest portfolio-collection --typescript --tailwind --
eslint --app --src-dir --import-alias "@/*"

# Navigate to project directory
cd portfolio-collection
```

Step 2: Install Required Dependencies

```
# Core dependencies
npm install firebase @reduxjs/toolkit react-redux redux-persist

# UI and Form handling
npm install react-hook-form @hookform/resolvers zod lucide-react

# PDF generation
npm install jspdf html2pdf.js @types/jspdf

# Additional utilities
npm install clsx tailwind-merge date-fns

# Development dependencies
npm install -D @types/node @types/react @types/react-dom
```

Step 3: Project Folder Structure

```
portfolio-collection/
├── public/
│   ├── icons/
│   └── images/
├── src/
│   ├── app/
│   │   ├── (auth)/
│   │   │   ├── login/
│   │   │   │   └── page.tsx                   # DONE
│   │   │   ├── register/
│   │   │   │   └── page.tsx                   # DONE
│   │   │   └── layout.tsx                     # DONE
│   │   ├── (dashboard)/
│   │   │   ├── dashboard/
│   │   │   │   └── page.tsx                   # DONE
│   │   │   ├── portfolio/
│   │   │   │   ├── create/
│   │   │   │   │   └── page.tsx               # DONE
│   │   │   │   ├── edit/
│   │   │   │   │   └── page.tsx               # DONE
│   │   │   │   └── [id]/
│   │   │   │       └── page.tsx               # DONE
│   │   │   ├── directory/
│   │   │   │   └── page.tsx                   # DONE
│   │   │   ├── admin/
│   │   │   │   ├── categories/
│   │   │   │   │   └── page.tsx               # DONE
│   │   │   │   ├── requests/
│   │   │   │   │   └── page.tsx               # DONE
│   │   │   │   └── page.tsx                   # DONE
│   │   │   └── layout.tsx                     # DONE
│   │   ├── api/
│   │   │   ├── auth/
│   │   │   ├── portfolio/
│   │   │   └── categories/
│   │   ├── globals.css
│   │   ├── layout.tsx
│   │   ├── page.tsx                            # DONE
│   │   ├── loading.tsx                         # DONE
│   │   └── not-found.tsx                       # DONE
│   ├── components/
│   │   ├── ui/
│   │   │   ├── button.tsx                      # DONE
│   │   │   ├── input.tsx                       # DONE
│   │   │   ├── card.tsx                        # DONE
│   │   │   ├── modal.tsx                       # DONE
│   │   │   ├── toast.tsx                       # DONE
│   │   │   └── index.ts                        # DONE
│   │   ├── auth/
│   │   │   ├── LoginForm.tsx                   # DONE
│   │   │   ├── RegisterForm.tsx                # DONE
│   │   │   ├── PasswordReset.tsx               # DONE
│   │   │   └── AuthProvider.tsx                # DONE
│   │   ├── portfolio/
│   │   │   ├── MultiStepForm/
│   │   │   │   ├── Step1PersonalInfo.tsx       # DONE
│   │   │   │   ├── Step2Education.tsx          # DONE
│   │   │   │   ├── Step3SkillsExperience.tsx   # DONE
│   │   │   │   ├── Step4Projects.tsx           # DONE
│   │   │   │   ├── FormNavigation.tsx          # DONE
│   │   │   │   └── index.tsx                   # DONE
│   │   │   ├── PortfolioCard.tsx               # DONE
│   │   │   ├── PortfolioView.tsx               # DONE
│   │   │   ├── PortfolioDirectory.tsx          # DONE
│   │   │   ├── SkillCategorySelector.tsx       # DONE
│   │   │   └── PDFExport.tsx                   # DONE
│   │   ├── admin/
│   │   │   ├── CategoryManagement.tsx          # DONE
│   │   │   ├── RequestApproval.tsx             # DONE
│   │   │   └── AdminPanel.tsx                  # DONE
│   │   ├── layout/
│   │   │   ├── Header.tsx                      # DONE
│   │   │   ├── Sidebar.tsx                     # DONE
│   │   │   ├── Footer.tsx                      # DONE
│   │   │   └── Navigation.tsx                  # DONE
│   │   └── common/
│   │       ├── LoadingSpinner.tsx              # DONE
│   │       ├── ErrorBoundary.tsx               # DONE
│   │       ├── ProtectedRoute.tsx              # DONE
│   │       └── SEO.tsx                         # DONE
│   ├── lib/
│   │   ├── firebase/
│   │   │   ├── config.ts                       # DONE
│   │   │   ├── auth.ts                         # DONE
│   │   │   ├── firestore.ts                    # DONE
│   │   │   └── storage.ts                      # DONE
│   │   ├── redux/
│   │   │   ├── store.ts                        # DONE
│   │   │   ├── slices/
│   │   │   │   ├── authSlice.ts                # DONE
│   │   │   │   ├── portfolioSlice.ts           # DONE
│   │   │   │   ├── skillCategoriesSlice.ts     # DONE
│   │   │   │   ├── categoryRequestsSlice.ts    # DONE
│   │   │   │   └── uiSlice.ts                  # DONE
│   │   │   └── providers.tsx                   # DONE
│   │   ├── hooks/
│   │   │   ├── useAuth.ts                      # DONE
│   │   │   ├── useDebounce.ts                  # DONE
│   │   │   ├── usePortfolio.ts                 # DONE
│   │   │   ├── useSkillCategories.ts           # DONE
│   │   │   ├── useCategoryRequests.ts          # DONE
│   │   │   └── useLocalStorage.ts              # DONE
│   │   ├── utils/
│   │   │   ├── validation.ts                   # DONE
│   │   │   ├── formatters.ts                   # DONE
│   │   │   ├── constants.ts                    # DONE
│   │   │   ├── pdf-generator.ts                # DONE
│   │   │   └── helpers.ts                      # DONE
│   │   └── types/
│   │       ├── auth.ts                         # DONE
│   │       ├── portfolio.ts                    # DONE
│   │       ├── skillCategories.ts              # DONE
│   │       └── index.ts                        # DONE
│   │       └── ui.ts                           # DONE
│   └── styles/
│       └── components.css
├── .env.local
├── .env.example
├── .gitignore
├── next.config.js
├── tailwind.config.js
├── tsconfig.json
├── package.json
└── README.md
```

Phase 2: Configuration Files Setup

Environment Variables (.env.local)

```
# Firebase Configuration
NEXT_PUBLIC_FIREBASE_API_KEY=your_api_key
NEXT_PUBLIC_FIREBASE_AUTH_DOMAIN=your_project.firebaseapp.com
NEXT_PUBLIC_FIREBASE_PROJECT_ID=your_project_id
NEXT_PUBLIC_FIREBASE_STORAGE_BUCKET=your_project.appspot.com
NEXT_PUBLIC_FIREBASE_MESSAGING_SENDER_ID=your_sender_id
NEXT_PUBLIC_FIREBASE_APP_ID=your_app_id

# Admin Configuration
NEXT_PUBLIC_ADMIN_EMAILS=admin1@example.com,admin2@example.com

# App Configuration
NEXT_PUBLIC_APP_URL=http://localhost:3000
NEXT_PUBLIC_APP_NAME=Portfolio Collection
```

Next.js Configuration (next.config.js)

```js
/** @type {import('next').NextConfig} */
const nextConfig = {
  experimental: {
    appDir: true,
  },
  images: {
    domains: ["firebasestorage.googleapis.com"],
    formats: ["image/webp", "image/avif"],
  },
  env: {
    CUSTOM_KEY: process.env.CUSTOM_KEY,
  },
};

module.exports = nextConfig;
```

Tailwind Configuration (tailwind.config.js)

```
/** @type {import('tailwindcss').Config} */
module.exports = {
  darkMode: ["class"],
  content: [
    "./pages/**/*.{ts,tsx}",
    "./components/**/*.{ts,tsx}",
    "./app/**/*.{ts,tsx}",
    "./src/**/*.{ts,tsx}",
  ],
  theme: {
    container: {
      center: true,
      padding: "2rem",
      screens: {
        "2xl": "1400px",
      },
    },
    extend: {
      colors: {
        border: "hsl(var(--border))",
        input: "hsl(var(--input))",
        ring: "hsl(var(--ring))",
        background: "hsl(var(--background))",
        foreground: "hsl(var(--foreground))",
        primary: {
          DEFAULT: "hsl(var(--primary))",
          foreground: "hsl(var(--primary-foreground))",
        },
        secondary: {
          DEFAULT: "hsl(var(--secondary))",
          foreground: "hsl(var(--secondary-foreground))",
        },
        destructive: {
          DEFAULT: "hsl(var(--destructive))",
          foreground: "hsl(var(--destructive-foreground))",
        },
        muted: {
          DEFAULT: "hsl(var(--muted))",
          foreground: "hsl(var(--muted-foreground))",
        },
        accent: {
          DEFAULT: "hsl(var(--accent))",
          foreground: "hsl(var(--accent-foreground))",
        },
        popover: {
          DEFAULT: "hsl(var(--popover))",
          foreground: "hsl(var(--popover-foreground))",
        },
        card: {
          DEFAULT: "hsl(var(--card))",
          foreground: "hsl(var(--card-foreground))",
        },
      },
      borderRadius: {
        lg: "var(--radius)",
        md: "calc(var(--radius) - 2px)",
        sm: "calc(var(--radius) - 4px)",
      },
      keyframes: {
        "accordion-down": {
          from: { height: 0 },
          to: { height: "var(--radix-accordion-content-height)" },
        },
        "accordion-up": {
          from: { height: "var(--radix-accordion-content-height)" },
          to: { height: 0 },
        },
      },
      animation: {
        "accordion-down": "accordion-down 0.2s ease-out",
        "accordion-up": "accordion-up 0.2s ease-out",
      },
    },
  },
  plugins: [require("tailwindcss-animate")],
};
```

Phase 3: Initial Development Steps

Step 1: Setup Firebase Configuration

1. Create a Firebase project at https://console.firebase.google.com
2. Enable Authentication (Email/Password and Google)
3. Create Firestore database
4. Set up Firestore security rules
5. Copy configuration to `.env.local`

Step 2: Setup TypeScript Types

Create comprehensive type definitions for the entire application.

Step 3: Setup Redux Store

Configure Redux Toolkit with proper slices for state management.

Step 4: Create Base Components

Start with UI components, then authentication, then portfolio components.

Step 5: Implement Authentication

Set up Firebase authentication with protected routes.

Next Steps

1. **Phase 1**: Complete project setup and Firebase configuration
2. **Phase 2**: Implement authentication system
3. **Phase 3**: Build portfolio creation multi-step form
4. **Phase 4**: Add portfolio viewing and directory features
5. **Phase 5**: Implement admin panel and category management
6. **Phase 6**: Add PDF export functionality
7. **Phase 7**: Testing and deployment

Would you like me to proceed with detailed implementation of any specific phase?