**Setup NextJS Application**

**Langkah 1 - Create NextJs Application**

Next.js menyediakan cara yang cepat dan mudah untuk memulai pengembangan aplikasi React full-stack modern. Metode paling sederhana adalah menggunakan tool resmi create-next-app, yang akan mengatur semuanya secara otomatis sesuai preferensi anda.

**🚀 Instalasi Otomatis dengan create-next-app**

Untuk membuat aplikasi Next.js baru, buka terminal Anda dan jalankan perintah berikut:

npx create-next-app@latest

Perintah ini akan menjalankan CLI interaktif yang membantu Anda membangun proyek dengan konfigurasi sesuai kebutuhan.

**📋 Penjelasan Opsi CLI**

Setelah menjalankan perintah di atas, Anda akan melihat beberapa pertanyaan. Berikut adalah penjelasan setiap pilihan beserta rekomendasinya:

|  |  |  |
| --- | --- | --- |
| **Prompt** | **Description** | **Recommendation** |
| What is your project named? | Set the name of your project folder. | Choose a meaningful name like my-portfolio or dashboard-app. |
| Would you like to use TypeScript? | Adds TypeScript support. | IMG_256Recommended for type safety and scalability. |
| Would you like to use ESLint? | Enables linting with ESLint. | IMG_257Recommended for maintaining code quality. |
| Would you like to use Tailwind CSS? | Adds Tailwind CSS setup for styling. | IMG_258Optional but great for utility-first styling. |
| Would you like your code inside a src/ directory? | Organizes app files inside a src/ folder. | IMG_259Recommended for better structure. |
| Would you like to use App Router (recommended)? | Enables the new App Router instead of the traditional Pages Router. | IMG_260Recommended for modern routing features. |
| Would you like to use Turbopack for next dev? | Uses Turbopack, a faster dev server alternative to Webpack. | IMG_261Still experimental but promising. Try it if you're curious. |
| Would you like to customize the import alias (@/\* by default)? | Sets up path aliases for cleaner imports. | IMG_262Skip this — default @/\* works fine for most cases. |

Langkah Selanjutnya

Buka project di vscode atau editor favorit kalian.

**Langkah 2 - Update ESLint Configuration Pada NextJS Project**

ESLint adalah alat penting untuk memastikan kualitas dan konsistensi kode dalam proyek JavaScript/TypeScript. Next.js sudah menyertakan konfigurasi dasar, tetapi dalam banyak kasus, kita ingin menyesuaikan konfigurasi tersebut untuk mendukung struktur proyek, plugin tambahan, dan aturan penulisan kode yang lebih baik.

Dokumentasi ini akan memandu Anda dalam memperbarui ESLint ke konfigurasi modern menggunakan format ESLint Flat Config, serta menambahkan aturan pengurutan impor otomatis dengan eslint-plugin-import.

**📦 Install Plugin Tambahan**

Pertama, kita perlu menginstal plugin ESLint untuk TypeScript agar bisa menulis aturan khusus untuk file .ts dan .tsx.

Jalankan perintah berikut di terminal:

npm install --save-dev prettier eslint-plugin-import eslint-plugin-prettier eslint-config-prettier

**Pengetahuan plugin**

📚 eslint-config-prettier vs. eslint-plugin-prettier

🔹 eslint-config-prettier

Ini adalah konfigurasi yang saat ini Anda gunakan di proyek Anda.

Tujuannya adalah untuk menonaktifkan semua rule ESLint yang bisa berkonflik dengan format kode dari Prettier.

Bersifat pasif, artinya hanya mematikan rule ESLint tertentu agar tidak bentrok dengan Prettier.

Pendekatan ini umumnya direkomendasikan saat mengintegrasikan ESLint dengan Prettier.

🔹 eslint-plugin-prettier

Paket ini memiliki pendekatan yang berbeda: menjalankan Prettier sebagai rule ESLint.

Artinya, format kode dari Prettier akan dianggap sebagai rule ESLint, dan jika ada yang tidak sesuai, akan muncul sebagai error atau warning dari ESLint.

Memberikan kemampuan untuk menegakkan aturan format Prettier lewat mekanisme pelaporan milik ESLint.

🔹prettier

Prettier adalah opinionated code formatter, yaitu alat untuk secara otomatis memformat kode agar rapi dan konsisten sesuai standar yang telah ditentukan.

🔧 Fungsi Utama:

Menata penulisan kode: indentasi, tanda kutip, titik koma, panjang baris, dll.

Menghilangkan perbedaan gaya antar developer.

Bekerja di berbagai bahasa: JavaScript, TypeScript, JSON, HTML, CSS, dan lainnya.

✅ Kelebihan:

Otomatis dan cepat.

Minim konfigurasi.

Bisa diintegrasikan dengan editor, Git hook, maupun ESLint.

🎯 Contoh Format yang Diatur:

// Sebelum

function sayHello(name){console.log("Hello, "+name);}

// Sesudah (dengan Prettier)

function sayHello(name) {

console.log("Hello, " + name);

}

🔹eslint-plugin-import

eslint-plugin-import adalah plugin ESLint yang membantu mengelola dan mengoptimalkan penggunaan import statement dalam proyek.

🔧 Fungsi Utama:

Memastikan semua import valid (modul tersedia).

Menangani urutan pengimporan (import order) supaya lebih rapi dan terstruktur.

Mendeteksi duplikat atau import yang tidak digunakan.

Membantu developer menjaga konsistensi dan kebersihan struktur impor.

✅ Kelebihan:

Sangat berguna di proyek besar dengan banyak modul/komponen.

Mempermudah navigasi dan pemeliharaan kode.

Bisa dikustomisasi: misalnya mengelompokkan import berdasarkan jenisnya.

🎯 Contoh Rule Populer: import/order

Aturan ini memungkinkan kita untuk menyusun import seperti ini:

// Built-in

import fs from 'fs';

// External

import React from 'react';

import axios from 'axios';

// Internal alias

import Button from '@/components/Button';

import { getData } from '@app/api';

// Relative

import Header from '../Header';

**Perbarui File Konfigurasi ESLint**

Pada project lama Next.js menggunakan file .eslintrc.json. Namun pada saat ini NextJS sudah menggunakan eslint.config.mjs. Jika anda mendapati kalau anda masih menggunakan .eslintrc.json , anda perlu menggantinya menjadi file JavaScript bernama eslint.config.mjs di root folder proyek Anda.

Buat atau ubah file eslint.config.js menjadi seperti berikut:

import { dirname } from 'path';

import { fileURLToPath } from 'url';

import { FlatCompat } from '@eslint/eslintrc';

import importPlugin from 'eslint-plugin-import';

const \_\_filename = fileURLToPath(import.meta.url);

const \_\_dirname = dirname(\_\_filename);

const compat = new FlatCompat({

baseDirectory: \_\_dirname,

});

const eslintConfig = [

...compat.extends('next/core-web-vitals', 'next/typescript', 'prettier'),

{

plugins: {

import: importPlugin,

},

rules: {

'import/order': [

'error',

{

groups: [

'builtin',

'external',

'internal',

['parent', 'sibling'],

'index',

'object',

],

'newlines-between': 'always',

pathGroups: [

{

pattern: '@/components/\*\*',

group: 'internal',

position: 'before',

},

{

pattern: '@app/\*\*',

group: 'external',

position: 'after',

},

],

pathGroupsExcludedImportTypes: ['builtin'],

alphabetize: {

order: 'asc',

caseInsensitive: true,

},

},

],

},

ignores: ['components/ui/\*\*'],

},

{

files: ['\*\*/\*.ts', '\*\*/\*.tsx'],

rules: {

'@typescript-eslint/no-explicit-any': 'off',

'@typescript-eslint/ban-ts-comment': 'off',

},

},

];

export default eslintConfig;

**Penjelasan Singkat Konfigurasi**

**FlatCompat:** Digunakan agar kita bisa tetap memakai konfigurasi lama (extends) di format konfigurasi baru (Flat Config).

**Plugin eslint-plugin-import:** Menambahkan kemampuan untuk mengatur dan mengelola urutan import secara otomatis.

**Aturan import/order:**

Mengelompokkan import berdasarkan tipe (builtin, external, internal, dsb).

Memasukkan pathGroups untuk mendeteksi alias seperti @/components/\*\* dan @app/\*\*.

Mengurutkan import secara alfabetis dan memisahkannya dengan newline.

**Aturan khusus untuk TypeScript:**

Menonaktifkan rule @typescript-eslint/no-explicit-any untuk mengizinkan penggunaan any, memberikan fleksibilitas saat prototyping. Ini bukan berarti kita akan menggunakan any semau kita, tapi justru akan memberikan fleksibilitas seperti membuat higher order function dan semacamnya. Kekurangannya adalah programmer junior di team anda harus diperingati karena penggunaan any tetaplah adalah hal yang terlarang jika digunakan untuk keperluan yang tidak seharusnya.

Menonaktifkan rule @typescript-eslint/ban-ts-comment agar anotasi seperti @ts-ignore dapat digunakan tanpa peringatan.

**Langkah 3 - Konfigurasi VS Code dengan .vscode/settings.json**

Agar pengalaman development di Visual Studio Code menjadi lebih konsisten, efisien, dan sesuai standar tim/proyek, kita dapat menambahkan konfigurasi khusus untuk editor dengan membuat folder .vscode dan file settings.json.

File ini digunakan untuk menyimpan pengaturan khusus proyek seperti format kode otomatis, integrasi dengan ESLint, Prettier, dan TypeScript.

**Membuat Folder dan File**

Di root folder proyek Anda, buat folder bernama .vscode

Di dalam folder tersebut, buat file baru bernama settings.json

Struktur file akan terlihat seperti ini:

my-app/

├── .vscode/

│ └── settings.json

├── src/

├── package.json

└── …

**Tujuan dan Manfaat settings.json**

File settings.json bertujuan untuk:

Menentukan formatter default (misalnya Prettier)

Mengaktifkan format otomatis saat menyimpan file

Mengaktifkan perbaikan otomatis oleh ESLint

Menyesuaikan gaya penulisan kode seperti penggunaan tanda kutip, koma akhir, tab/spasi, dsb.

Mengatur path TypeScript agar menggunakan versi lokal dari node\_modules

Konfigurasi ini hanya berlaku pada proyek saat ini, sehingga tidak mengubah pengaturan global VS Code Anda.

Isi settings.json

Salin dan tempel konfigurasi berikut ke dalam file .vscode/settings.json:

{

"editor.defaultFormatter": "esbenp.prettier-vscode",

"editor.formatOnSave": true,

"editor.codeActionsOnSave": {

"source.fixAll.eslint": "explicit",

"source.addMissingImports": "explicit"

},

"prettier.tabWidth": 2,

"prettier.useTabs": false,

"prettier.semi": true,

"prettier.singleQuote": true,

"prettier.jsxSingleQuote": true,

"prettier.trailingComma": "es5",

"prettier.arrowParens": "always",

"[json]": {

"editor.defaultFormatter": "esbenp.prettier-vscode"

},

"[typescript]": {

"editor.defaultFormatter": "esbenp.prettier-vscode"

},

"[typescriptreact]": {

"editor.defaultFormatter": "esbenp.prettier-vscode"

},

"[javascriptreact]": {

"editor.defaultFormatter": "esbenp.prettier-vscode"

},

"typescript.tsdk": "node\_modules/typescript/lib",

"files.exclude": {

"\*\*/node\_modules": true,

"next-env.d.ts": true,

"\*\*/.vscode": true

}

}

**Penjelasan Singkat Beberapa Baris Penting**

| **Baris** | **Penjelasan** |
| --- | --- |
| "editor.defaultFormatter" | Mengatur Prettier sebagai formatter default. |
| "editor.formatOnSave": true | Otomatis format file saat disimpan. |
| "source.fixAll.eslint" | Memperbaiki semua error ESLint secara otomatis saat save. |
| "prettier.\*" | Aturan format yang digunakan Prettier. |
| "typescript.tsdk" | Mengarahkan VS Code untuk menggunakan TypeScript lokal dari node\_modules agar sesuai versi proyek. |

Dengan konfigurasi ini, Anda akan mendapatkan workflow yang rapi dan otomatis: setiap kali menyimpan file, VS Code akan mengatur format kode dan memperbaiki lint error sesuai aturan yang telah ditentukan. Karena kita menyimpannya di project, aturan ini berlaku untuk setiap programmer yang bekerja dengan kalian di project tersebut.

**Langkah 4 – Instalasi prettier-plugin-tailwindcss & Konfigurasi .prettierrc**

🎯 Mengapa Langkah Ini Penting?

Langkah ini sangat penting untuk memastikan formatting kode tetap konsisten, khususnya dalam proyek yang menggunakan Tailwind CSS.

Dengan bantuan prettier-plugin-tailwindcss, Prettier akan:

Secara otomatis mengurutkan class Tailwind berdasarkan urutan standar yang direkomendasikan.

Membantu Anda menghindari class duplikat atau tidak urut, yang bisa mempersulit pemeliharaan UI.

Memastikan kode HTML, JSX, dan komponen tetap rapi dan mudah dibaca.

Instalasi Plugin

Jalankan perintah berikut di terminal:

npm install --save-dev prettier-plugin-tailwindcss

Pastikan prettier sudah terpasang sebagai dependensi proyek Anda. Jika belum, perintah di atas sudah sekaligus menginstalnya.

Membuat File .prettierrc

Buat file konfigurasi Prettier bernama .prettierrc di root folder proyek Anda, lalu isi dengan konfigurasi berikut:

{

"plugins": ["prettier-plugin-tailwindcss"],

"singleQuote": true,

"jsxSingleQuote": true,

"semi": true,

"tabWidth": 2,

"bracketSpacing": true,

"jsxBracketSameLine": false,

"arrowParens": "always",

"trailingComma": "es5",

"printWidth": 80

}

📖 Penjelasan Singkat Konfigurasi

| Properti | Fungsi |
| --- | --- |
| "plugins" | Menambahkan plugin eksternal ke Prettier, dalam hal ini untuk Tailwind CSS |
| "singleQuote" | Gunakan tanda kutip satu ('') untuk string |
| "jsxSingleQuote" | Gunakan tanda kutip satu di atribut JSX |
| "semi" | Tambahkan titik koma di akhir baris |
| "tabWidth" | Lebar indentasi: 2 spasi |
| "bracketSpacing" | Tambah spasi di dalam objek ({ foo: bar }) |
| "jsxBracketSameLine" | Penutup tag JSX di baris baru |
| "arrowParens" | Selalu pakai tanda kurung di parameter arrow function |
| "trailingComma" | Tambahkan koma di akhir objek/array bila memungkinkan |
| "printWidth" | Maksimal panjang baris adalah 80 karakter |

✅ Hasil yang Didapat

Tanpa plugin ini:

<div class="p-4 flex flex-col text-center justify-between bg-white shadow-md">

Dengan plugin prettier-plugin-tailwindcss, class akan otomatis diurutkan:

<div class="flex flex-col justify-between p-4 text-center bg-white shadow-md">

**Langkah 5 – Mengganti Font Family Sesuai dengan Design System**

Dalam sebuah proyek yang berbasis pada desain (terutama dari Figma), sangat penting untuk menggunakan font yang konsisten dengan design system. Font bukan hanya soal estetika, tapi juga bagian dari identitas merek dan pengalaman pengguna.

Secara default, beberapa proyek Next.js mungkin menggunakan font seperti Geist dari Vercel. Namun, jika desainer Anda menggunakan font lain seperti Poppins, maka Anda harus menyesuaikan implementasi font-nya di proyek Anda.

Tujuan Langkah Ini

Menyesuaikan font family proyek agar sesuai dengan desain dari Figma.

Menggunakan pendekatan next/font/google untuk performa optimal.

Menyediakan CSS variable yang bisa digunakan di seluruh project.

1. Instalasi Font dengan next/font/google

Next.js menyediakan cara optimal dan efisien untuk memuat font Google melalui modul next/font/google.

Font akan di-inject secara otomatis ke halaman, dengan dukungan preload dan subset yang efisien. Anda tidak perlu mengatur CDN atau mengatur font secara manual.

Contoh Sebelumnya: Font Geist

Contoh ini menggunakan dua jenis font: Geist dan Geist\_Mono.

import { Geist, Geist\_Mono } from "next/font/google";

const geistSans = Geist({

variable: "--font-geist-sans",

subsets: ["latin"],

});

const geistMono = Geist\_Mono({

variable: "--font-geist-mono",

subsets: ["latin"],

});

Font ini mungkin tidak sesuai dengan brand atau desain sistem Anda.

Solusi Baru: Menggunakan Font Poppins

Misalnya, desain dari Figma menggunakan Poppins sebagai font utama. Maka kita perlu menggantinya dengan cara berikut:

🔧 Update File app/layout.tsx atau app/layout.jsx

import clsx from 'clsx';

import type { Metadata } from 'next';

import { Poppins } from 'next/font/google';

import './globals.css';

export const metadata: Metadata = {

title: 'Create Next App',

description: 'Generated by create next app',

};

const poppins = Poppins({

subsets: ['latin'],

weight: ['400', '500', '600', '700', '800'],

style: ['normal', 'italic'],

variable: '--font-poppins',

});

export default function RootLayout({

children,

}: Readonly<{

children: React.ReactNode;

}>) {

return (

<html lang='en'>

<body className={clsx(poppins.variable, 'antialiased')}>

{children}

</body>

</html>

);

}

Tambahan: npm install clsx

**Langkah 6 - Install ShadCN**

📌 Apa Itu ShadCN?

ShadCN adalah sebuah kumpulan komponen UI berbasis Tailwind CSS, yang dibangun di atas prinsip reusability, accessibility, dan customizability.

Komponen-komponen ShadCN dirancang agar mudah dimodifikasi, terintegrasi langsung dengan Tailwind, dan cocok digunakan dalam proyek modern seperti Next.js.

🔍 Karakteristik Utama:

Terintegrasi dengan Radix UI (untuk fungsionalitas interaktif dan aksesibilitas).

Bergaya Tailwind-first dan tidak mengunci Anda dalam desain tertentu.

Komponennya dibuat langsung ke dalam proyek Anda, jadi sepenuhnya bisa dikustomisasi.

Mengapa Harus Install ShadCN Sebelum Konfigurasi Tailwind?

Ketika Anda menjalankan shadcn init, proses tersebut akan:

Mengupdate file globals.css

Menambahkan utility/helper ke dalam folder lib/utils

Jika Anda sudah lebih dulu melakukan konfigurasi Tailwind, maka perubahan yang dilakukan oleh ShadCN akan mengacaukan struktur CSS global Anda, membuat file globals.css menjadi:

Tidak konsisten

Berantakan

Sulit diatur kembali

🛑 Kesimpulan:

Selalu install ShadCN lebih dulu sebelum mulai mengatur atau memodifikasi konfigurasi Tailwind Anda.

Cara Instalasi ShadCN

Jalankan perintah berikut di terminal Anda:

npx shadcn@latest init

Apa yang Akan Ditambahkan?

1. Folder lib/utils

Folder ini akan berisi file-file utilitas, seperti:

// Contoh: lib/utils.ts

export function cn(...inputs: ClassValue[]) {

return twMerge(clsx(...inputs));

}

Kegunaan:

Fungsi cn() digunakan untuk menggabungkan class Tailwind secara dinamis.

Dipakai oleh hampir semua komponen ShadCN.

Bisa juga Anda gunakan di komponen buatan sendiri untuk handling class.

2. Update pada app/globals.css

ShadCN akan menambahkan base styling, font, dan beberapa variabel CSS agar komponen UI-nya bisa bekerja dengan baik.

**Langkah 7 – Install Tailwind & Konfigurasi globals.css dan tailwind.config.ts**

📌 Tujuan Langkah Ini

Langkah ini bertujuan untuk:

Menginstal dan mengatur Tailwind CSS agar siap digunakan.

Mengimplementasikan design system dari desainer (misalnya dari Figma) ke dalam globals.css.

Menggunakan konfigurasi tambahan di tailwind.config.ts untuk mendukung utilitas kustom, animasi, dan auto-complete dari Tailwind IntelliSense.

1. **Buat File tailwind.config.ts**

Di root proyek Anda, buat file baru:

touch tailwind.config.ts

File ini akan menampung:

Konfigurasi tambahan dari design system.

Plugin utilitas kustom.

Keyframe animasi untuk kebutuhan seperti marquee.

💡 Mengapa butuh tailwind.config.ts jika sudah ada globals.css?

Utility class yang dibuat di globals.css tidak terdeteksi oleh Tailwind IntelliSense, sehingga:

Tidak ada auto-complete di VS Code.

Tidak bisa melihat isi class langsung.

Tidak bisa menggunakan ekspresi JavaScript.

Jadi, untuk class yang kompleks dan dinamis, selalu lebih baik ditulis di tailwind.config.ts.

1. **Update globals.css**

Tambahkan file globals.css di dalam src/app/ atau src/styles/, lalu isi dengan struktur design system yang Anda terima dari desainer Figma.

Contoh globals.css sudah diberikan sebelumnya dan dapat disesuaikan dengan kebutuhan design system Anda. Contoh tersebut mencakup:

Font size dan line height dengan variable CSS

Color tokens untuk tema warna (primary, neutral, accent, dsb)

Border radius, spacing, dan font weight

Custom utility seperti:

.custom-container → mempermudah layouting

.flex-center, .flex-between, .flex-start → mempercepat pembuatan layout berbasis flex

Tema dari ShadCN agar sinkron dengan komponen UI

Berikut isi dari globals.css

(Important: globals.css ini menggunakan design system dari design yang digunakan pada project untuk keperluan portfolio development workshop, silahkan update sesuai keperluan)

@import 'tailwindcss';

@config '../../tailwind.config.ts';

@import 'tw-animate-css';

@custom-variant dark (&:is(.dark \*));

@theme {

/\* Font Family \*/

/\* Text Sizes and Line Heights \*/

--text-display-3xl: 3.75rem;

--text-display-3xl--line-height: 4.5rem;

--text-display-2xl: 3rem;

--text-display-2xl--line-height: 3.75rem;

--text-display-2xl--letter-spacing: -2%;

--text-display-xl: 2.5rem;

--text-display-xl--line-height: 3rem;

--text-display-xl--letter-spacing: -2%;

--text-display-lg: 2.25rem;

--text-display-lg--line-height: 2.75rem;

--text-display-lg--letter-spacing: -2%;

--text-display-md: 2rem;

--text-display-md--line-height: 2.5rem;

--text-display-sm: 1.75rem;

--text-display-sm--line-height: 2.375rem;

--text-display-xs: 1.5rem;

--text-display-xs--line-height: 2.25rem;

--text-xl: 1.25rem;

--text-xl--line-height: 2.125rem;

--text-lg: 1.125rem;

--text-lg--line-height: 2rem;

--text-md: 1rem;

--text-md--line-height: 1.875rem;

--text-sm: 0.875rem;

--text-sm--line-height: 1.75rem;

--text-xs: 0.75rem;

--text-xs--line-height: 1.5rem;

/\* Font Weights \*/

--font-weight-regular: 400;

--font-weight-medium: 500;

--font-weight-semibold: 600;

--font-weight-bold: 700;

--font-weight-extrabold: 800;

/\* Color Theme \*/

--color-neutral-25: #fdfdfd;

--color-neutral-50: #fafafa;

--color-neutral-100: #f5f5f5;

--color-neutral-200: #e9eaeb;

--color-neutral-300: #d5d7da;

--color-neutral-400: #a4a7ae;

--color-neutral-500: #717680;

--color-neutral-600: #535862;

--color-neutral-700: #414651;

--color-neutral-800: #252b37;

--color-neutral-900: #181d27;

--color-neutral-950: #0a0d12;

--color-primary-100: #e5f4fb;

--color-primary-200: #6fcfff;

--color-primary-300: #0093dd;

--color-primary-400: #050a1b;

--color-base-white: #ffffff;

--color-base-background: #0c0d0d;

/\* Radius \*/

--radius-xxs: 0.125rem;

--radius-xs: 0.25rem;

--radius-sm: 0.375rem;

--radius-md: 0.5rem;

--radius-lg: 0.625rem;

--radius-xl: 0.75rem;

--radius-2xl: 1rem;

--radius-3xl: 1.25rem;

--radius-4xl: 1.5rem;

/\* Spacing \*/

--spacing-xxs: 0.125rem;

--spacing-xs: 0.25rem;

--spacing-sm: 0.375rem;

--spacing-md: 0.5rem;

--spacing-lg: 0.75rem;

--spacing-xl: 1rem;

--spacing-2xl: 1.25rem;

--spacing-3xl: 1.5rem;

--spacing-4xl: 2rem;

--spacing-5xl: 2.5rem;

--spacing-6xl: 3rem;

--spacing-7xl: 4rem;

--spacing-8xl: 5rem;

--spacing-9xl: 6rem;

--spacing-10xl: 8rem;

--spacing-11xl: 8.75rem;

}

/\* ---------------------------------------- \*/

/\* shadcn theme \*/

/\* ---------------------------------------- \*/

:root {

--radius: 0.625rem;

--background: oklch(1 0 0);

--foreground: oklch(0.129 0.042 264.695);

--card: oklch(1 0 0);

--card-foreground: oklch(0.129 0.042 264.695);

--popover: oklch(1 0 0);

--popover-foreground: oklch(0.129 0.042 264.695);

--primary: oklch(0.208 0.042 265.755);

--primary-foreground: oklch(0.984 0.003 247.858);

--secondary: oklch(0.968 0.007 247.896);

--secondary-foreground: oklch(0.208 0.042 265.755);

--muted: oklch(0.968 0.007 247.896);

--muted-foreground: oklch(0.554 0.046 257.417);

--accent: oklch(0.968 0.007 247.896);

--accent-foreground: oklch(0.208 0.042 265.755);

--destructive: oklch(0.577 0.245 27.325);

--border: oklch(0.929 0.013 255.508);

--input: oklch(0.929 0.013 255.508);

--ring: oklch(0.704 0.04 256.788);

--chart-1: oklch(0.646 0.222 41.116);

--chart-2: oklch(0.6 0.118 184.704);

--chart-3: oklch(0.398 0.07 227.392);

--chart-4: oklch(0.828 0.189 84.429);

--chart-5: oklch(0.769 0.188 70.08);

--sidebar: oklch(0.984 0.003 247.858);

--sidebar-foreground: oklch(0.129 0.042 264.695);

--sidebar-primary: oklch(0.208 0.042 265.755);

--sidebar-primary-foreground: oklch(0.984 0.003 247.858);

--sidebar-accent: oklch(0.968 0.007 247.896);

--sidebar-accent-foreground: oklch(0.208 0.042 265.755);

--sidebar-border: oklch(0.929 0.013 255.508);

--sidebar-ring: oklch(0.704 0.04 256.788);

}

@theme inline {

--color-background: var(--background);

--color-foreground: var(--foreground);

--font-sans: var(--font-geist-sans);

--font-mono: var(--font-geist-mono);

--color-sidebar-ring: var(--sidebar-ring);

--color-sidebar-border: var(--sidebar-border);

--color-sidebar-accent-foreground: var(--sidebar-accent-foreground);

--color-sidebar-accent: var(--sidebar-accent);

--color-sidebar-primary-foreground: var(--sidebar-primary-foreground);

--color-sidebar-primary: var(--sidebar-primary);

--color-sidebar-foreground: var(--sidebar-foreground);

--color-sidebar: var(--sidebar);

--color-chart-5: var(--chart-5);

--color-chart-4: var(--chart-4);

--color-chart-3: var(--chart-3);

--color-chart-2: var(--chart-2);

--color-chart-1: var(--chart-1);

--color-ring: var(--ring);

--color-input: var(--input);

--color-border: var(--border);

--color-destructive: var(--destructive);

--color-accent-foreground: var(--accent-foreground);

--color-accent: var(--accent);

--color-muted-foreground: var(--muted-foreground);

--color-muted: var(--muted);

--color-secondary-foreground: var(--secondary-foreground);

--color-secondary: var(--secondary);

--color-primary-foreground: var(--primary-foreground);

--color-primary: var(--primary);

--color-popover-foreground: var(--popover-foreground);

--color-popover: var(--popover);

--color-card-foreground: var(--card-foreground);

--color-card: var(--card);

}

/\* ---------------------------------------- \*/

/\* end of shadcn theme \*/

/\* ---------------------------------------- \*/

body {

@apply bg-base-background text-sm font-normal text-white;

font-family: var(--font-poppins);

}

@layer components {

.custom-container {

margin: 0 auto;

@apply max-w-302 pr-6 pl-6;

}

}

@layer utilities {

/\* flex utilities \*/

.flex-center {

@apply flex items-center justify-center;

}

.flex-between {

@apply flex items-center justify-between;

}

.flex-start {

@apply flex items-center justify-start;

}

}

img {

user-drag: none;

-webkit-user-drag: none; /\* For Safari \*/

user-select: none; /\* Prevent selection \*/

touch-action: none; /\* Disable touch gestures \*/

-webkit-user-select: none; /\* For Safari \*/

}

3. Konfigurasi tailwind.config.ts

Berikut isi file tailwind.config.ts lengkap dengan plugin kustom dan animasi marquee:

import type { Config } from 'tailwindcss';

import plugin from 'tailwindcss/plugin';

const textSizes = {

'display-3xl': '--text-display-3xl',

'display-2xl': '--text-display-2xl',

'display-xl': '--text-display-xl',

'display-lg': '--text-display-lg',

'display-md': '--text-display-md',

'display-sm': '--text-display-sm',

'display-xs': '--text-display-xs',

'text-xl': '--text-xl',

'text-lg': '--text-lg',

'text-md': '--text-md',

'text-sm': '--text-sm',

'text-xs': '--text-xs',

};

const fontWeights = {

regular: '--font-weight-regular',

medium: '--font-weight-medium',

semibold: '--font-weight-semibold',

bold: '--font-weight-bold',

extrabold: '--font-weight-extrabold',

};

const customTextPlugin = plugin(({ addUtilities }) => {

const newUtilities: Record<string, any> = {};

for (const [sizeName, sizeVar] of Object.entries(textSizes)) {

for (const [weightName, weightVar] of Object.entries(fontWeights)) {

const className = `.${sizeName}-${weightName}`;

newUtilities[className] = {

fontSize: `var(${sizeVar})`,

lineHeight: `var(${sizeVar}--line-height)`,

fontWeight: `var(${weightVar})`,

};

}

}

addUtilities(newUtilities);

});

export default {

theme: {

extend: {

animation: {

marquee: 'marquee var(--duration) linear infinite',

'marquee-vertical': 'marquee-vertical var(--duration) linear infinite',

},

keyframes: {

marquee: {

from: { transform: 'translateX(0)' },

to: { transform: 'translateX(calc(-100% - var(--gap)))' },

},

'marquee-vertical': {

from: { transform: 'translateY(0)' },

to: { transform: 'translateY(calc(-100% - var(--gap)))' },

},

},

},

},

plugins: [customTextPlugin],

} satisfies Config;

**Langkah 8 – Clear The Boilerplate**

Setelah melakukan setup awal proyek Next.js menggunakan create-next-app, secara default akan ada banyak file boilerplate (template) yang sebenarnya tidak Anda butuhkan, terutama jika Anda ingin memulai dengan struktur bersih dan desain sendiri.

Langkah ini bertujuan untuk membersihkan file dan kode default, agar proyek Anda lebih ringan, rapi, dan siap dibangun dari nol.

🗂️ 1. Bersihkan app/page.tsx

Secara default, file app/page.tsx akan berisi konten dan style bawaan dari Next.js yang menampilkan logo dan link ke dokumentasi. Kita akan menghapus semua isi file ini dan menggantinya dengan komponen kosong sebagai titik awal.

Ganti dengan kode berikut:

const Home = () => {

return <div></div>;

};

export default Home;

Dengan ini, halaman utama (/) akan kosong dan siap diisi konten sesuai kebutuhan proyek.

🗑️ 2. Kosongkan Folder public

Folder public/ biasanya berisi file gambar dan ikon bawaan dari Next.js dan Vercel. Untuk proyek yang menggunakan desain kustom dari Figma, file-file ini tidak relevan dan sebaiknya dihapus.

🧾 File yang biasanya ada di folder public/:

file.svg

globe.svg

next.svg

vercel.svg

window.svg

✅ Tindakan:

Hapus semua file tersebut, sehingga struktur folder public/ menjadi kosong.

Hasil Akhir

Setelah membersihkan boilerplate:

Halaman utama kosong dan siap diisi komponen baru.

Tidak ada aset gambar bawaan yang memenuhi folder public.

**Langkah 9 – Ambil Aset dari Figma & Atur Folder Untuk Aset**

Untuk membangun antarmuka pengguna yang sesuai dengan desain, langkah penting berikutnya adalah mengambil semua aset (ikon & gambar) dari file Figma dan mengaturnya ke dalam proyek Anda. Aset ini mencakup: logo, ikon brand, gambar profil, gambar hero, dan elemen visual lainnya.

**📝 1. Buka Desain di Figma**

Masuk ke file desain Figma yang digunakan untuk proyek Anda.

Pastikan Anda sudah memiliki akses "Can View" atau "Can Edit" agar bisa mengekspor aset.

1. **Download Aset dari Figma**

Untuk setiap aset (ikon atau gambar):

Klik elemen (ikon/gambar) yang ingin di download.

Di sidebar kanan, scroll ke bagian Export.

Klik tombol + jika belum ada opsi ekspor.

Pilih format ekspor sesuai kebutuhan:

SVG untuk ikon vektor (misalnya logo, ikon brand).

JPG/PNG untuk gambar raster (misalnya foto profil, background).

Klik tombol Export.

🎯 Gunakan format SVG untuk ikon agar tetap tajam di semua resolusi, dan JPG/PNG untuk gambar visual seperti ilustrasi atau foto.

1. **Buat Struktur Folder Aset**

Setelah mendownload semua aset, susun folder di dalam direktori public/ proyek Anda seperti berikut:

public/

├── icons/

│ ├── icon-adobe.svg

│ ├── icon-airbnb.svg

│ ├── icon-databricks.svg

│ ├── icon-dropbox.svg

│ ├── icon-netflix.svg

│ ├── icon-paypal.svg

│ ├── icon-postman.svg

│ ├── icon-upwork.svg

│ └── icon-zoom.svg

└── images/

├── hero-image.png

├── impact-image.jpg

├── logo.svg

├── profile1.jpg

├── profile2.jpg

├── profile3.jpg

├── project1.jpg

├── project2.jpg

├── project3.jpg

├── tab1.jpg

├── tab2.jpg

└── tab3.jpg

🎉 Project siap untuk digunakan!

GLOBAL.CSS (Porto 2)

@import 'tailwindcss';

@config '../../tailwind.config.ts';

@import 'tw-animate-css';

@custom-variant dark (&:is(.dark \*));

@theme {

  /\* Font Family \*/

  /\* Text Sizes and Line Heights \*/

  --text-display-3xl: 3.75rem;

  --text-display-3xl--line-height: 4.5rem;

  --text-display-2xl: 3rem;

  --text-display-2xl--line-height: 3.75rem;

  --text-display-2xl--letter-spacing: -2%;

  --text-display-xl: 2.5rem;

  --text-display-xl--line-height: 3rem;

  --text-display-xl--letter-spacing: -2%;

  --text-display-lg: 2.25rem;

  --text-display-lg--line-height: 2.75rem;

  --text-display-lg--letter-spacing: -2%;

  --text-display-md: 2rem;

  --text-display-md--line-height: 2.5rem;

  --text-display-sm: 1.75rem;

  --text-display-sm--line-height: 2.375rem;

  --text-display-xs: 1.5rem;

  --text-display-xs--line-height: 2.25rem;

  --text-xl: 1.25rem;

  --text-xl--line-height: 2.125rem;

  --text-lg: 1.125rem;

  --text-lg--line-height: 2rem;

  --text-md: 1rem;

  --text-md--line-height: 1.875rem;

  --text-sm: 0.875rem;

  --text-sm--line-height: 1.75rem;

  --text-xs: 0.75rem;

  --text-xs--line-height: 1.5rem;

  /\* Font Weights \*/

  --font-weight-regular: 400;

  --font-weight-medium: 500;

  --font-weight-semibold: 600;

  --font-weight-bold: 700;

  --font-weight-extrabold: 800;

  /\* Color Theme \*/

  --color-neutral-100: #ffffff;

  --color-neutral-200: #b5b5b5;

  --color-neutral-300: #545357;

  --color-neutral-400: #242424;

  --color-neutral-500: #0f0f0f;

  --color-neutral-600: #020203;

  --color-primary-100: #d5bbfa;

  --color-primary-200: #8144df;

  --color-primary-300: #632abb;

  --color-secondary-100: rgba(220, 73, 166, 0.4);

  --color-secondary-200: #dc49a6;

  --color-secondary-300: #b83387;

  --color-base-white: #ffffff;

  --color-base-background: #0c0d0d;

  /\* Radius \*/

  --radius-xxs: 0.125rem;

  --radius-xs: 0.25rem;

  --radius-sm: 0.375rem;

  --radius-md: 0.5rem;

  --radius-lg: 0.625rem;

  --radius-xl: 0.75rem;

  --radius-2xl: 1rem;

  --radius-3xl: 1.25rem;

  --radius-4xl: 1.5rem;

  /\* Spacing \*/

  --spacing-xxs: 0.125rem;

  --spacing-xs: 0.25rem;

  --spacing-sm: 0.375rem;

  --spacing-md: 0.5rem;

  --spacing-lg: 0.75rem;

  --spacing-xl: 1rem;

  --spacing-2xl: 1.25rem;

  --spacing-3xl: 1.5rem;

  --spacing-4xl: 2rem;

  --spacing-5xl: 2.5rem;

  --spacing-6xl: 3rem;

  --spacing-7xl: 4rem;

  --spacing-8xl: 5rem;

  --spacing-9xl: 6rem;

  --spacing-10xl: 8rem;

  --spacing-11xl: 8.75rem;

}

/\* ---------------------------------------- \*/

/\* shadcn theme \*/

/\* ---------------------------------------- \*/

:root {

  --radius: 0.625rem;

  --background: oklch(1 0 0);

  --foreground: oklch(0.129 0.042 264.695);

  --card: oklch(1 0 0);

  --card-foreground: oklch(0.129 0.042 264.695);

  --popover: oklch(1 0 0);

  --popover-foreground: oklch(0.129 0.042 264.695);

  --primary: oklch(0.208 0.042 265.755);

  --primary-foreground: oklch(0.984 0.003 247.858);

  --secondary: oklch(0.968 0.007 247.896);

  --secondary-foreground: oklch(0.208 0.042 265.755);

  --muted: oklch(0.968 0.007 247.896);

  --muted-foreground: oklch(0.554 0.046 257.417);

  --accent: oklch(0.968 0.007 247.896);

  --accent-foreground: oklch(0.208 0.042 265.755);

  --destructive: oklch(0.577 0.245 27.325);

  --border: oklch(0.929 0.013 255.508);

  --input: oklch(0.929 0.013 255.508);

  --ring: oklch(0.704 0.04 256.788);

  --chart-1: oklch(0.646 0.222 41.116);

  --chart-2: oklch(0.6 0.118 184.704);

  --chart-3: oklch(0.398 0.07 227.392);

  --chart-4: oklch(0.828 0.189 84.429);

  --chart-5: oklch(0.769 0.188 70.08);

  --sidebar: oklch(0.984 0.003 247.858);

  --sidebar-foreground: oklch(0.129 0.042 264.695);

  --sidebar-primary: oklch(0.208 0.042 265.755);

  --sidebar-primary-foreground: oklch(0.984 0.003 247.858);

  --sidebar-accent: oklch(0.968 0.007 247.896);

  --sidebar-accent-foreground: oklch(0.208 0.042 265.755);

  --sidebar-border: oklch(0.929 0.013 255.508);

  --sidebar-ring: oklch(0.704 0.04 256.788);

}

@theme inline {

  --color-background: var(--background);

  --color-foreground: var(--foreground);

  --font-sans: var(--font-geist-sans);

  --font-mono: var(--font-geist-mono);

  --color-sidebar-ring: var(--sidebar-ring);

  --color-sidebar-border: var(--sidebar-border);

  --color-sidebar-accent-foreground: var(--sidebar-accent-foreground);

  --color-sidebar-accent: var(--sidebar-accent);

  --color-sidebar-primary-foreground: var(--sidebar-primary-foreground);

  --color-sidebar-primary: var(--sidebar-primary);

  --color-sidebar-foreground: var(--sidebar-foreground);

  --color-sidebar: var(--sidebar);

  --color-chart-5: var(--chart-5);

  --color-chart-4: var(--chart-4);

  --color-chart-3: var(--chart-3);

  --color-chart-2: var(--chart-2);

  --color-chart-1: var(--chart-1);

  --color-ring: var(--ring);

  --color-input: var(--input);

  --color-border: var(--border);

  --color-destructive: var(--destructive);

  --color-accent-foreground: var(--accent-foreground);

  --color-accent: var(--accent);

  --color-muted-foreground: var(--muted-foreground);

  --color-muted: var(--muted);

  --color-secondary-foreground: var(--secondary-foreground);

  --color-secondary: var(--secondary);

  --color-primary-foreground: var(--primary-foreground);

  --color-primary: var(--primary);

  --color-popover-foreground: var(--popover-foreground);

  --color-popover: var(--popover);

  --color-card-foreground: var(--card-foreground);

  --color-card: var(--card);

  /\* --radius-sm: calc(var(--radius) - 4px);

  --radius-md: calc(var(--radius) - 2px);

  --radius-lg: var(--radius);

  --radius-xl: calc(var(--radius) + 4px); \*/

}

/\* ---------------------------------------- \*/

/\* end of shadcn theme \*/

/\* ---------------------------------------- \*/

html {

  scroll-behavior: smooth;

}

body {

  @apply bg-base-background text-sm font-normal text-white;

  font-family: var(--font-poppins);

}

@layer components {

  .custom-container {

    margin: 0 auto;

    @apply max-w-302 pr-6 pl-6;

  }

}

@layer utilities {

  /\* flex utilities \*/

  .flex-center {

    @apply flex items-center justify-center;

  }

  .flex-between {

    @apply flex items-center justify-between;

  }

  .flex-start {

    @apply flex items-center justify-start;

  }

}

img {

  user-drag: none;

  -webkit-user-drag: none; /\* For Safari \*/

  user-select: none; /\* Prevent selection \*/

  touch-action: none; /\* Disable touch gestures \*/

  -webkit-user-select: none; /\* For Safari \*/

}

**Tailwind.config.ts (Porto 2)**

import type { Config } from 'tailwindcss';

import plugin from 'tailwindcss/plugin';

const textSizes = {

  'display-3xl': '--text-display-3xl',

  'display-2xl': '--text-display-2xl',

  'display-xl': '--text-display-xl',

  'display-lg': '--text-display-lg',

  'display-md': '--text-display-md',

  'display-sm': '--text-display-sm',

  'display-xs': '--text-display-xs',

  'text-xl': '--text-xl',

  'text-lg': '--text-lg',

  'text-md': '--text-md',

  'text-sm': '--text-sm',

  'text-xs': '--text-xs',

};

const fontWeights = {

  regular: '--font-weight-regular',

  medium: '--font-weight-medium',

  semibold: '--font-weight-semibold',

  bold: '--font-weight-bold',

  extrabold: '--font-weight-extrabold',

};

const customTextPlugin = plugin(({ addUtilities }) => {

  const newUtilities: Record<string, any> = {};

  for (const [sizeName, sizeVar] of Object.entries(textSizes)) {

    for (const [weightName, weightVar] of Object.entries(fontWeights)) {

      const className = `.${sizeName}-${weightName}`;

      newUtilities[className] = {

        fontSize: `var(${sizeVar})`,

        lineHeight: `var(${sizeVar}--line-height)`,

        fontWeight: `var(${weightVar})`,

      };

    }

  }

  addUtilities(newUtilities);

});

export default {

  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: {

        pink: {

          50: '#fff7f9',

          100: '#ffe4e9',

          200: '#ffc8d3',

          300: '#ffabbd',

          400: '#ff89a7',

          500: '#ff6691',

          600: '#ff4e7a',

          700: '#ff3562',

          800: '#ff1d4a',

          900: '#ff0032',

        },

      },

      animation: {

        'accordion-down': 'accordion-down 0.2s ease-out',

        'accordion-up': 'accordion-up 0.2s ease-out',

        marquee: 'marquee var(--duration) linear infinite',

        'marquee-vertical': 'marquee-vertical var(--duration) linear infinite',

      },

      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' },

        },

        marquee: {

          from: { transform: 'translateX(0)' },

          to: { transform: 'translateX(calc(-100% - var(--gap)))' },

        },

        'marquee-vertical': {

          from: { transform: 'translateY(0)' },

          to: { transform: 'translateY(calc(-100% - var(--gap)))' },

        },

      },

    },

  },

  plugins: [customTextPlugin],

} satisfies Config;

Layout.tsx

import clsx from 'clsx';

import type { Metadata } from 'next';

import { Poppins } from 'next/font/google';

import './globals.css';

export const metadata: Metadata = {

  title: 'Create Next App',

  description: 'Generated by create next app',

};

const poppins = Poppins({

  subsets: ['latin'],

  weight: ['400', '500', '600', '700', '800'],

  style: ['normal', 'italic'],

  variable: '--font-poppins',

});

export default function RootLayout({

  children,

}: Readonly<{

  children: React.ReactNode;

}>) {

  return (

    <html lang='en'>

      <body className={clsx(poppins.variable, 'antialiased')}>{children}</body>

    </html>

  );

}

**Page.tsx**

'use client';

import React from 'react';

import About from './home/partials/about';

import Careers from './home/partials/careers';

import Comparison from './home/partials/comparison';

import ContactForm from './home/partials/contact-form';

import FAQ from './home/partials/faq';

import Footer from './home/partials/footer';

import Hero from './home/partials/hero';

import Navbar from './home/partials/navbar';

import Projects from './home/partials/projects';

import Skills from './home/partials/skills';

import Teksaja from './home/partials/teksaja';

import Testimonials from './home/partials/testimonials';

const Home = () => {

  return (

    <div>

      <Navbar />

      <Hero />

      <About />

      <Careers />

      <Skills />

      <FAQ />

      <Comparison />

      <Teksaja />

      <Projects />

      <Testimonials />

      <ContactForm />

      <Footer />

    </div>

  );

};

export default Home;

**Src > layouts > section.tsx**

import React from 'react';

type SectionProps = {

  children: React.ReactNode;

  title: string;

  subtitle: string;

  id?: string;

};

export const Section: React.FC<SectionProps> = ({

  children,

  title,

  subtitle,

  id,

}) => {

  return (

    <div className='custom-container py-10 md:py-20' id={id}>

      {/\* heading \*/}

      <div className='text-center'>

        <h2 className='display-sm-bold md:display-lg-bold text-neutral-25'>

          {title}

        </h2>

        <p className='text-sm-regular md:text-md-regular mt-4 text-white'>

          {subtitle}

        </p>

      </div>

      {/\* content \*/}

      <div className='mt-6 flex items-center justify-center md:mt-16'>

        {children}

      </div>

    </div>

  );

};

**Components > ui**

**Accordion.tsx**

'use client';

import \* as AccordionPrimitive from '@radix-ui/react-accordion';

import { PlusIcon, MinusIcon } from 'lucide-react';

import \* as React from 'react';

import { cn } from '@/lib/utils';

function Accordion({

  ...props

}: React.ComponentProps<typeof AccordionPrimitive.Root>) {

  return <AccordionPrimitive.Root data-slot='accordion' {...props} />;

}

function AccordionItem({

  className,

  ...props

}: React.ComponentProps<typeof AccordionPrimitive.Item>) {

  return (

    <AccordionPrimitive.Item

      data-slot='accordion-item'

      className={cn(

        'mb-4 rounded-2xl bg-neutral-900 p-4 last:mb-0 md:mb-8 md:rounded-4xl md:p-6',

        className

      )}

      {...props}

    />

  );

}

function AccordionTrigger({

  className,

  children,

  ...props

}: React.ComponentProps<typeof AccordionPrimitive.Trigger>) {

  return (

    <AccordionPrimitive.Header className='flex'>

      <AccordionPrimitive.Trigger

        data-slot='accordion-trigger'

        className={cn(

          '!text-md-semibold md:text-lg-semibold text-neutral-25 hover:text-primary-200 data-[state=open]:text-primary-200 group/trigger flex flex-1 cursor-pointer items-start justify-between gap-4 text-left transition-all',

          className

        )}

        {...props}

      >

        <div className='flex-1 text-left'>{children}</div>

        <div className='shrink-0'>

          <PlusIcon

            color='#fdfdfd'

            className='group-data-[state=open]/trigger:hidden'

          />

          <MinusIcon

            color='#fdfdfd'

            className='group-data-[state=closed]/trigger:hidden'

          />

        </div>

      </AccordionPrimitive.Trigger>

    </AccordionPrimitive.Header>

  );

}

function AccordionContent({

  className,

  children,

  ...props

}: React.ComponentProps<typeof AccordionPrimitive.Content>) {

  return (

    <AccordionPrimitive.Content

      data-slot='accordion-content'

      className='data-[state=closed]:animate-accordion-up data-[state=open]:animate-accordion-down text-sm-regular overflow-hidden'

      {...props}

    >

      <div className={cn('text-sm-regular mt-4 text-neutral-400', className)}>

        {children}

      </div>

    </AccordionPrimitive.Content>

  );

}

export { Accordion, AccordionItem, AccordionTrigger, AccordionContent };

**UI > Button.tsx**

import { Slot } from '@radix-ui/react-slot';

import { cva, type VariantProps } from 'class-variance-authority';

import \* as React from 'react';

import { cn } from '@/lib/utils';

const buttonVariants = cva(

  'flex justify-center items-center shrink-0 disabled:pointer-events-none disabled:opacity-50 cursor-pointer outline-none',

  {

    variants: {

      variant: {

        default:

          'bg-gradient-to-r from-[#8746eb] to-[#dc49ab] text-sm-medium text-neutral-25 rounded-full hover:shadow-[0\_0\_17px\_rgba(135,70,235,0.6)]',

        outline:

          'border border-neutral-800 rounded-full hover:border-neutral-700',

      },

      size: {

        default: 'h-11 px-14.5',

        icon: 'size-10 md:size-12',

      },

    },

    defaultVariants: {

      variant: 'default',

      size: 'default',

    },

  }

);

function Button({

  className,

  variant,

  size,

  asChild = false,

  ...props

}: React.ComponentProps<'button'> &

  VariantProps<typeof buttonVariants> & {

    asChild?: boolean;

  }) {

  const Comp = asChild ? Slot : 'button';

  return (

    <Comp

      data-slot='button'

      className={cn(buttonVariants({ variant, size, className }))}

      {...props}

    />

  );

}

export { Button, buttonVariants };

Ui > Carousel.tsx

'use client';

import useEmblaCarousel, {

  type UseEmblaCarouselType,

} from 'embla-carousel-react';

import { ChevronLeft, ChevronRight } from 'lucide-react';

import \* as React from 'react';

import { Button } from '@/components/ui/button';

import { cn } from '@/lib/utils';

type CarouselApi = UseEmblaCarouselType[1];

type UseCarouselParameters = Parameters<typeof useEmblaCarousel>;

type CarouselOptions = UseCarouselParameters[0];

type CarouselPlugin = UseCarouselParameters[1];

type CarouselProps = {

  opts?: CarouselOptions;

  plugins?: CarouselPlugin;

  orientation?: 'horizontal' | 'vertical';

  setApi?: (api: CarouselApi) => void;

};

type CarouselContextProps = {

  carouselRef: ReturnType<typeof useEmblaCarousel>[0];

  api: ReturnType<typeof useEmblaCarousel>[1];

  scrollPrev: () => void;

  scrollNext: () => void;

  canScrollPrev: boolean;

  canScrollNext: boolean;

} & CarouselProps;

const CarouselContext = React.createContext<CarouselContextProps | null>(null);

function useCarousel() {

  const context = React.useContext(CarouselContext);

  if (!context) {

    throw new Error('useCarousel must be used within a <Carousel />');

  }

  return context;

}

function Carousel({

  orientation = 'horizontal',

  opts,

  setApi,

  plugins,

  className,

  children,

  ...props

}: React.ComponentProps<'div'> & CarouselProps) {

  const [carouselRef, api] = useEmblaCarousel(

    {

      ...opts,

      axis: orientation === 'horizontal' ? 'x' : 'y',

    },

    plugins

  );

  const [canScrollPrev, setCanScrollPrev] = React.useState(false);

  const [canScrollNext, setCanScrollNext] = React.useState(false);

  const onSelect = React.useCallback((api: CarouselApi) => {

    if (!api) return;

    setCanScrollPrev(api.canScrollPrev());

    setCanScrollNext(api.canScrollNext());

  }, []);

  const scrollPrev = React.useCallback(() => {

    api?.scrollPrev();

  }, [api]);

  const scrollNext = React.useCallback(() => {

    api?.scrollNext();

  }, [api]);

  const handleKeyDown = React.useCallback(

    (event: React.KeyboardEvent<HTMLDivElement>) => {

      if (event.key === 'ArrowLeft') {

        event.preventDefault();

        scrollPrev();

      } else if (event.key === 'ArrowRight') {

        event.preventDefault();

        scrollNext();

      }

    },

    [scrollPrev, scrollNext]

  );

  React.useEffect(() => {

    if (!api || !setApi) return;

    setApi(api);

  }, [api, setApi]);

  React.useEffect(() => {

    if (!api) return;

    onSelect(api);

    api.on('reInit', onSelect);

    api.on('select', onSelect);

    return () => {

      api?.off('select', onSelect);

    };

  }, [api, onSelect]);

  return (

    <CarouselContext.Provider

      value={{

        carouselRef,

        api: api,

        opts,

        orientation:

          orientation || (opts?.axis === 'y' ? 'vertical' : 'horizontal'),

        scrollPrev,

        scrollNext,

        canScrollPrev,

        canScrollNext,

      }}

    >

      <div

        onKeyDownCapture={handleKeyDown}

        className={cn('relative', className)}

        role='region'

        aria-roledescription='carousel'

        data-slot='carousel'

        {...props}

      >

        {children}

      </div>

    </CarouselContext.Provider>

  );

}

function CarouselContent({ className, ...props }: React.ComponentProps<'div'>) {

  const { carouselRef, orientation } = useCarousel();

  return (

    <div

      ref={carouselRef}

      className='overflow-hidden'

      data-slot='carousel-content'

    >

      <div

        className={cn(

          'flex',

          orientation === 'horizontal' ? '-ml-5' : '-mt-5 flex-col',

          className

        )}

        {...props}

      />

    </div>

  );

}

function CarouselItem({ className, ...props }: React.ComponentProps<'div'>) {

  const { orientation } = useCarousel();

  return (

    <div

      role='group'

      aria-roledescription='slide'

      data-slot='carousel-item'

      className={cn(

        'min-w-0 shrink-0 grow-0 basis-full md:basis-1/2 lg:basis-1/3',

        orientation === 'horizontal' ? 'pl-5' : 'pt-5',

        className

      )}

      {...props}

    />

  );

}

function CarouselPrevious(props: React.ComponentProps<typeof Button>) {

  const { scrollPrev, canScrollPrev } = useCarousel();

  return (

    <Button

      data-slot='carousel-previous'

      variant='outline'

      size='icon'

      disabled={!canScrollPrev}

      onClick={scrollPrev}

      {...props}

    >

      <ChevronLeft />

      <span className='sr-only'>Previous slide</span>

    </Button>

  );

}

function CarouselNext(props: React.ComponentProps<typeof Button>) {

  const { scrollNext, canScrollNext } = useCarousel();

  return (

    <Button

      data-slot='carousel-next'

      variant='outline'

      size='icon'

      disabled={!canScrollNext}

      onClick={scrollNext}

      {...props}

    >

      <ChevronRight />

      <span className='sr-only'>Next slide</span>

    </Button>

  );

}

function CarouselNavigation({

  className,

  ...props

}: React.ComponentProps<'div'>) {

  return (

    <div

      className={cn(

        'mt-6 flex items-center justify-center gap-3 md:mt-10',

        className

      )}

      {...props}

    >

      <CarouselPrevious />

      <CarouselNext />

    </div>

  );

}

export {

  type CarouselApi,

  Carousel,

  CarouselContent,

  CarouselItem,

  CarouselNavigation,

};

**Ui > Checkbox.tsx**

'use client';

import \* as CheckboxPrimitive from '@radix-ui/react-checkbox';

import { Check } from 'lucide-react';

import \* as React from 'react';

import { cn } from '@/lib/utils';

function Checkbox({

  className,

  ...props

}: React.ComponentProps<typeof CheckboxPrimitive.Root>) {

  return (

    <CheckboxPrimitive.Root

      data-slot='checkbox'

      className={cn(

        'peer data-[state=checked]:bg-primary-300 data-[state=checked]:text-neutral-25 data-[state=checked]:border-primary-300 focus-visible:border-ring focus-visible:ring-ring/50 size-5 shrink-0 rounded-sm border border-neutral-800 outline-none focus-visible:ring-[3px] disabled:cursor-not-allowed disabled:opacity-50',

        className

      )}

      {...props}

    >

      <CheckboxPrimitive.Indicator

        data-slot='checkbox-indicator'

        className='flex items-center justify-center text-current transition-none'

      >

        <Check className='size-3.5' />

      </CheckboxPrimitive.Indicator>

    </CheckboxPrimitive.Root>

  );

}

export { Checkbox };

**Ui > Dialog.tsx**

'use client';

import \* as DialogPrimitive from '@radix-ui/react-dialog';

import \* as React from 'react';

import { cn } from '@/lib/utils';

function Dialog({

  ...props

}: React.ComponentProps<typeof DialogPrimitive.Root>) {

  return <DialogPrimitive.Root data-slot='dialog' {...props} />;

}

function DialogTrigger({

  ...props

}: React.ComponentProps<typeof DialogPrimitive.Trigger>) {

  return <DialogPrimitive.Trigger data-slot='dialog-trigger' {...props} />;

}

function DialogPortal({

  ...props

}: React.ComponentProps<typeof DialogPrimitive.Portal>) {

  return <DialogPrimitive.Portal data-slot='dialog-portal' {...props} />;

}

function DialogClose({

  ...props

}: React.ComponentProps<typeof DialogPrimitive.Close>) {

  return <DialogPrimitive.Close data-slot='dialog-close' {...props} />;

}

function DialogOverlay({

  className,

  ...props

}: React.ComponentProps<typeof DialogPrimitive.Overlay>) {

  return (

    <DialogPrimitive.Overlay

      data-slot='dialog-overlay'

      className={cn(

        'data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0 fixed inset-0 z-50 bg-black/80',

        className

      )}

      {...props}

    />

  );

}

function DialogContent({

  className,

  ...props

}: React.ComponentProps<typeof DialogPrimitive.Content>) {

  return (

    <DialogPortal data-slot='dialog-portal'>

      <DialogOverlay />

      <DialogPrimitive.Content

        data-slot='dialog-content'

        className={cn(

          'data-[state=open]:animate-in data-[state=closed]:animate-out data-[state=closed]:fade-out-0 data-[state=open]:fade-in-0 data-[state=closed]:zoom-out-95 data-[state=open]:zoom-in-95 fixed top-[50%] left-[50%] z-50 grid w-full max-w-139.5 translate-x-[-50%] translate-y-[-50%] rounded-lg px-5 shadow-lg duration-200 outline-none',

          className

        )}

        {...props}

      />

    </DialogPortal>

  );

}

function DialogHeader({ className, ...props }: React.ComponentProps<'div'>) {

  return (

    <div

      data-slot='dialog-header'

      className={cn('rounded-t-2xl', className)}

      {...props}

    />

  );

}

function DialogBody({ className, ...props }: React.ComponentProps<'div'>) {

  return (

    <div

      data-slot='dialog-body'

      className={cn(

        'z-50 rounded-2xl bg-neutral-950 pt-6 pb-8 md:pt-8 md:pb-10',

        className

      )}

      {...props}

    />

  );

}

function DialogFooter({ className, ...props }: React.ComponentProps<'div'>) {

  return (

    <div

      data-slot='dialog-footer'

      className={cn(

        'flex flex-col-reverse gap-2 sm:flex-row sm:justify-end',

        className

      )}

      {...props}

    />

  );

}

function DialogTitle({

  className,

  ...props

}: React.ComponentProps<typeof DialogPrimitive.Title>) {

  return (

    <DialogPrimitive.Title

      data-slot='dialog-title'

      className={cn(

        'text-lg-semibold md:text-xl-semibold text-neutral-25 text-center',

        className

      )}

      {...props}

    />

  );

}

function DialogDescription({

  className,

  ...props

}: React.ComponentProps<typeof DialogPrimitive.Description>) {

  return (

    <DialogPrimitive.Description

      data-slot='dialog-description'

      className={cn('text-muted-foreground text-center text-sm', className)}

      {...props}

    />

  );

}

export {

  Dialog,

  DialogClose,

  DialogContent,

  DialogDescription,

  DialogFooter,

  DialogHeader,

  DialogBody,

  DialogOverlay,

  DialogPortal,

  DialogTitle,

  DialogTrigger,

};

**UI > Featured-card.tsx**

import React from 'react';

import { cn } from '@/lib/utils';

import SkewMotion from './skew-motion';

type FeatureCardProps = {

  icon: React.ReactNode;

  title: string;

  description: string;

  className?: string;

  animationDelay?: number;

};

const FeatureCard: React.FC<FeatureCardProps> = ({

  icon,

  title,

  description,

  className,

  animationDelay,

}) => {

  return (

    <SkewMotion asChild delay={animationDelay}>

      <div

        className={cn(

          'rounded-md bg-gradient-to-b from-[#0093DD] to-transparent p-0.25',

          className

        )}

        style={{

          width: 'clamp(6.375rem, 15.89vw, 12rem)',

          borderRadius: 'clamp(0.25rem, 0.66vw, 0.5rem)',

        }}

      >

        <div

          className='rounded-md bg-neutral-900'

          style={{

            borderRadius: 'clamp(0.25rem, 0.66vw, 0.5rem)',

            padding: 'clamp(0.5rem, 1.25vw, 1rem)',

          }}

        >

          <div

            className='bg-primary-300 flex-center aspect-square rounded-full shadow-[0\_0\_17px\_rgba(0,147,221,0.6)] [&>\*]:h-auto [&>\*]:w-full'

            style={{

              width: 'clamp(1.38rem, 3.31vw, 2.5rem)',

              padding: 'clamp(0.31rem, 0.83vw, 0.63rem)',

            }}

          >

            {icon}

          </div>

          <h3

            className='text-neutral-25 font-semibold'

            style={{

              fontSize: 'clamp(0.47rem, 1.16vw, 0.88rem)',

              lineHeight: 'clamp(0.94rem, 2.32vw, 1.75rem)',

              marginTop: 'clamp(0.25rem, 0.66vw, 0.5rem)',

            }}

          >

            {title}

          </h3>

          <p

            className='font-regular text-neutral-400'

            style={{

              fontSize: 'clamp(0.5rem, calc((14 / 1208) \* 100vw), 0.875rem)',

              lineHeight:

                'clamp(0.9375rem, calc((28 / 1208) \* 100vw), 1.75rem)',

            }}

          >

            {description}

          </p>

        </div>

      </div>

    </SkewMotion>

  );

};

export default FeatureCard;

**Ui > form-status-dialog.tsx**

import Image from 'next/image';

import React from 'react';

import { Button } from './button';

import {

  Dialog,

  DialogBody,

  DialogClose,

  DialogContent,

  DialogDescription,

  DialogHeader,

  DialogTitle,

} from './dialog';

interface FormStatusDialogProps extends React.ComponentProps<typeof Dialog> {

  variant: 'success' | 'error';

  loading?: boolean;

}

const FormStatusDialog: React.FC<FormStatusDialogProps> = ({

  variant,

  loading,

  ...props

}) => {

  return (

    <Dialog {...props}>

      <DialogContent className='max-w-md w-full'>

        <DialogHeader

          className='relative h-48 bg-gradient-to-br from-[#0093DD]/90 to-[#46AEE2]/90 rounded-t-2xl shadow-lg flex items-center justify-center'

        >

          <Image

            src={

              variant === 'success'

                ? '/icons/icon-form-message-success.svg'

                : '/icons/icon-form-message-error.svg'

            }

            alt={variant === 'success' ? 'success' : 'error'}

            width={220}

            height={242}

            className='animate-bounce-slow scale-125 md:scale-150'

          />

        </DialogHeader>

        <DialogBody className='bg-neutral-900 rounded-b-2xl p-6 text-center'>

          <DialogTitle className='text-2xl font-bold text-white mb-2'>

            {variant === 'success'

              ? 'Message Sent Successfully!'

              : 'Failed to Send Message'}

          </DialogTitle>

          <DialogDescription className='text-neutral-400 mb-6'>

            {variant === 'success'

              ? 'Thank you for reaching out. Our team will get back to you soon.'

              : 'Please try again later. If the issue persists, check your internet connection.'}

          </DialogDescription>

          <DialogClose asChild>

            <Button

              className='w-full bg-gradient-to-r from-[#0093DD] to-[#46AEE2] hover:from-[#0085C9] hover:to-[#3F9ED0] transition-all duration-300'

            >

              {loading ? (

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

                  <span className="mr-2">Loading...</span>

                  <svg className="animate-spin h-5 w-5 text-white" 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"></circle>

                    <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"></path>

                  </svg>

                </div>

              ) : (

                'Return Home'

              )}

            </Button>

          </DialogClose>

        </DialogBody>

      </DialogContent>

    </Dialog>

  );

};

export default FormStatusDialog;

**Ui > Form.tsx**

'use client';

import \* as LabelPrimitive from '@radix-ui/react-label';

import { Slot } from '@radix-ui/react-slot';

import \* as React from 'react';

import {

  Controller,

  FormProvider,

  useFormContext,

  useFormState,

  type ControllerProps,

  type FieldPath,

  type FieldValues,

} from 'react-hook-form';

import { Label } from '@/components/ui/label';

import { cn } from '@/lib/utils';

const Form = FormProvider;

type FormFieldContextValue<

  TFieldValues extends FieldValues = FieldValues,

  TName extends FieldPath<TFieldValues> = FieldPath<TFieldValues>,

> = {

  name: TName;

};

const FormFieldContext = React.createContext<FormFieldContextValue>(

  {} as FormFieldContextValue

);

const FormField = <

  TFieldValues extends FieldValues = FieldValues,

  TName extends FieldPath<TFieldValues> = FieldPath<TFieldValues>,

>({

  ...props

}: ControllerProps<TFieldValues, TName>) => {

  return (

    <FormFieldContext.Provider value={{ name: props.name }}>

      <Controller {...props} />

    </FormFieldContext.Provider>

  );

};

const useFormField = () => {

  const fieldContext = React.useContext(FormFieldContext);

  const itemContext = React.useContext(FormItemContext);

  const { getFieldState } = useFormContext();

  const formState = useFormState({ name: fieldContext.name });

  const fieldState = getFieldState(fieldContext.name, formState);

  if (!fieldContext) {

    throw new Error('useFormField should be used within <FormField>');

  }

  const { id } = itemContext;

  return {

    id,

    name: fieldContext.name,

    formItemId: `${id}-form-item`,

    formDescriptionId: `${id}-form-item-description`,

    formMessageId: `${id}-form-item-message`,

    ...fieldState,

  };

};

type FormItemContextValue = {

  id: string;

};

const FormItemContext = React.createContext<FormItemContextValue>(

  {} as FormItemContextValue

);

function FormItem({ className, ...props }: React.ComponentProps<'div'>) {

  const id = React.useId();

  return (

    <FormItemContext.Provider value={{ id }}>

      <div

        data-slot='form-item'

        className={cn('grid gap-2', className)}

        {...props}

      />

    </FormItemContext.Provider>

  );

}

function FormLabel({

  className,

  ...props

}: React.ComponentProps<typeof LabelPrimitive.Root>) {

  const { error, formItemId } = useFormField();

  return (

    <Label

      data-slot='form-label'

      data-error={!!error}

      className={cn(

        'text-xs-regular md:text-sm-regular text-neutral-25 leading-6 md:leading-7',

        className

      )}

      htmlFor={formItemId}

      {...props}

    />

  );

}

function FormControl({ ...props }: React.ComponentProps<typeof Slot>) {

  const { error, formItemId, formDescriptionId, formMessageId } =

    useFormField();

  return (

    <Slot

      data-slot='form-control'

      id={formItemId}

      aria-describedby={

        !error

          ? `${formDescriptionId}`

          : `${formDescriptionId} ${formMessageId}`

      }

      aria-invalid={!!error}

      {...props}

    />

  );

}

function FormDescription({ className, ...props }: React.ComponentProps<'p'>) {

  const { formDescriptionId } = useFormField();

  return (

    <p

      data-slot='form-description'

      id={formDescriptionId}

      className={cn('text-muted-foreground text-sm', className)}

      {...props}

    />

  );

}

function FormMessage({ className, ...props }: React.ComponentProps<'p'>) {

  const { error, formMessageId } = useFormField();

  const body = error ? String(error?.message ?? '') : props.children;

  if (!body) {

    return null;

  }

  return (

    <p

      data-slot='form-message'

      id={formMessageId}

      className={cn('text-destructive text-sm', className)}

      {...props}

    >

      {body}

    </p>

  );

}

export {

  useFormField,

  Form,

  FormItem,

  FormLabel,

  FormControl,

  FormDescription,

  FormMessage,

  FormField,

};

**Ui > Input.tsx**

import \* as React from 'react';

import { cn } from '@/lib/utils';

function Input({ className, type, ...props }: React.ComponentProps<'input'>) {

  return (

    <input

      type={type}

      data-slot='input'

      className={cn(

        'text-sm-regular md:text-md-regular h-11 w-full rounded-md bg-neutral-900 px-3 py-2 outline-none placeholder:text-neutral-500 md:h-12 md:rounded-xl md:px-4 md:py-2.25',

        'focus:ring-[1px] focus:ring-neutral-500',

        className

      )}

      {...props}

    />

  );

}

export { Input };

**Ui > Label.tsx**

'use client';

import \* as LabelPrimitive from '@radix-ui/react-label';

import \* as React from 'react';

import { cn } from '@/lib/utils';

function Label({

  className,

  ...props

}: React.ComponentProps<typeof LabelPrimitive.Root>) {

  return (

    <LabelPrimitive.Root

      data-slot='label'

      className={cn(

        'flex items-center gap-2 text-sm leading-none font-medium select-none group-data-[disabled=true]:pointer-events-none group-data-[disabled=true]:opacity-50 peer-disabled:cursor-not-allowed peer-disabled:opacity-50',

        className

      )}

      {...props}

    />

  );

}

export { Label };

**Ui > marquee.tsx**

import { ComponentPropsWithRef } from 'react';

import { cn } from '@/lib/utils';

interface MarqueeProps extends ComponentPropsWithRef<'div'> {

  className?: string;

  reverse?: boolean;

  pauseOnHover?: boolean;

  children: React.ReactNode;

  vertical?: boolean;

  repeat?: number;

}

export const Marquee: React.FC<MarqueeProps> = ({

  className,

  reverse = false,

  pauseOnHover = true,

  children,

  vertical = false,

  repeat = 4,

  ...props

}) => {

  return (

    <div

      className={cn(

        'group flex [gap:var(--gap)] overflow-hidden p-2 [--duration:40s] [--gap:3rem]',

        className

      )}

      {...props}

    >

      {Array(repeat)

        .fill(null)

        .map((\_, i) => (

          <div

            key={i}

            className={cn('flex shrink-0 justify-around [gap:var(--gap)]', {

              'animate-marquee flex-row': !vertical,

              'animate-marquee-vertical flex-col': vertical,

              'group-hover:[animation-play-state:paused]': pauseOnHover,

              '[animation-direction:reverse]': reverse,

            })}

          >

            {children}

          </div>

        ))}

    </div>

  );

};

<div className='parent'>

  <div className='child'></div>

</div>;

**Ui > phases.acordion.tsx**

import { ComponentPropsWithRef } from 'react';

import { cn } from '@/lib/utils';

interface MarqueeProps extends ComponentPropsWithRef<'div'> {

  className?: string;

  reverse?: boolean;

  pauseOnHover?: boolean;

  children: React.ReactNode;

  vertical?: boolean;

  repeat?: number;

}

export const Marquee: React.FC<MarqueeProps> = ({

  className,

  reverse = false,

  pauseOnHover = true,

  children,

  vertical = false,

  repeat = 4,

  ...props

}) => {

  return (

    <div

      className={cn(

        'group flex [gap:var(--gap)] overflow-hidden p-2 [--duration:40s] [--gap:3rem]',

        className

      )}

      {...props}

    >

      {Array(repeat)

        .fill(null)

        .map((\_, i) => (

          <div

            key={i}

            className={cn('flex shrink-0 justify-around [gap:var(--gap)]', {

              'animate-marquee flex-row': !vertical,

              'animate-marquee-vertical flex-col': vertical,

              'group-hover:[animation-play-state:paused]': pauseOnHover,

              '[animation-direction:reverse]': reverse,

            })}

          >

            {children}

          </div>

        ))}

    </div>

  );

};

<div className='parent'>

  <div className='child'></div>

</div>;

Ui > sheet.tsx

import { ComponentPropsWithRef } from 'react';

import { cn } from '@/lib/utils';

interface MarqueeProps extends ComponentPropsWithRef<'div'> {

  className?: string;

  reverse?: boolean;

  pauseOnHover?: boolean;

  children: React.ReactNode;

  vertical?: boolean;

  repeat?: number;

}

export const Marquee: React.FC<MarqueeProps> = ({

  className,

  reverse = false,

  pauseOnHover = true,

  children,

  vertical = false,

  repeat = 4,

  ...props

}) => {

  return (

    <div

      className={cn(

        'group flex [gap:var(--gap)] overflow-hidden p-2 [--duration:40s] [--gap:3rem]',

        className

      )}

      {...props}

    >

      {Array(repeat)

        .fill(null)

        .map((\_, i) => (

          <div

            key={i}

            className={cn('flex shrink-0 justify-around [gap:var(--gap)]', {

              'animate-marquee flex-row': !vertical,

              'animate-marquee-vertical flex-col': vertical,

              'group-hover:[animation-play-state:paused]': pauseOnHover,

              '[animation-direction:reverse]': reverse,

            })}

          >

            {children}

          </div>

        ))}

    </div>

  );

};

<div className='parent'>

  <div className='child'></div>

</div>;

**Ui > skew-motion.tsx**

'use client';

import { Slot } from '@radix-ui/react-slot';

import { HTMLMotionProps, motion } from 'framer-motion';

import React from 'react';

interface SkewMotionProps extends HTMLMotionProps<'div'> {

  children: React.ReactNode;

  duration?: number;

  delay?: number;

  asChild?: boolean;

}

const MotionSlot = motion.create(Slot);

const MotionDiv = motion.div;

const SkewMotion: React.FC<SkewMotionProps> = ({

  children,

  duration = 0.5,

  delay = 0,

  asChild = false,

  ...motionProps

}) => {

  const Comp = asChild ? MotionSlot : MotionDiv;

  return (

    <Comp

      animate={{ skewX: -7 }}

      transition={{ duration, delay }}

      {...motionProps}

    >

      {children}

    </Comp>

  );

};

export default SkewMotion;

**Ui > tabs.tsx**

'use client';

import { Slot } from '@radix-ui/react-slot';

import { HTMLMotionProps, motion } from 'framer-motion';

import React from 'react';

interface SkewMotionProps extends HTMLMotionProps<'div'> {

  children: React.ReactNode;

  duration?: number;

  delay?: number;

  asChild?: boolean;

}

const MotionSlot = motion.create(Slot);

const MotionDiv = motion.div;

const SkewMotion: React.FC<SkewMotionProps> = ({

  children,

  duration = 0.5,

  delay = 0,

  asChild = false,

  ...motionProps

}) => {

  const Comp = asChild ? MotionSlot : MotionDiv;

  return (

    <Comp

      animate={{ skewX: -7 }}

      transition={{ duration, delay }}

      {...motionProps}

    >

      {children}

    </Comp>

  );

};

export default SkewMotion;

**Ui > Textarea.tsx**

import \* as React from 'react';

import { cn } from '@/lib/utils';

function Textarea({ className, ...props }: React.ComponentProps<'textarea'>) {

  return (

    <textarea

      data-slot='textarea'

      className={cn(

        'text-sm-regular md:text-md-regular h-33.5 w-full resize-none rounded-md bg-neutral-900 px-3 py-2 outline-none placeholder:text-neutral-500 md:rounded-xl md:px-4 md:py-2.25',

        'focus:ring-[1px] focus:ring-neutral-500',

        className

      )}

      {...props}

    />

  );

}

export { Textarea };

**Component > section.tsx**

'use client';

import React from 'react';

interface SectionProps {

  children: React.ReactNode;

  title: string;

  subtitle: string;

  id: string;

}

const Section: React.FC<SectionProps> = ({ children, title, subtitle, id }) => {

  return (

    <section id={id} className="space-y-8">

      <div className="text-center mb-8">

        <h2 className="text-4xl font-bold mb-2">{title}</h2>

        <p className="text-gray-400">{subtitle}</p>

      </div>

      {children}

    </section>

  );

};

export { Section };

**Constant > comparison-data.ts (DATA)**

export interface Technology {

  name: string;

  me: string;

  other: string;

}

export const comparisonData: Technology[] = [

  {

    name: 'React Expert',

    me: '✅',

    other: '❌',

  },

  {

    name: 'UI/UX Design',

    me: '✅',

    other: '❌',

  },

  {

    name: 'TypeScript Proficiency',

    me: '✅',

    other: '❌',

  },

  {

    name: 'Performance Optimization',

    me: '✅',

    other: '❌',

  },

  {

    name: 'Responsive Design',

    me: '✅',

    other: '❌',

  },

  {

    name: 'Clean Code',

    me: '✅',

    other: '❌',

  },

  {

    name: 'Testing & QA',

    me: '✅',

    other: '❌',

  },

];

**Constant > faq-data.ts**

type FAQItem = {

  title: string;

  description: string;

};

export const FAQData: FAQItem[] = [

  {

    title: 'How do you ensure websites load quickly and efficiently?',

    description:

      'We provide end-to-end IT solutions, including web & mobile app development.',

  },

  {

    title: 'What’s your approach to front-end development??',

    description:

      'I focus on clean, maintainable code and prioritize user experience. My approach involves close collaboration with designers to ensure exact implementation and seamless interactions across all devices.',

  },

  {

    title: 'What kind of projects do you specialize in?',

    description:

      'It depends on the complexity and scope of the project. Small projects may take a few weeks, while larger enterprise solutions can take several months. We provide a detailed project timeline during the consultation phase.',

  },

  {

    title: 'How do you handle project deadlines?',

    description:

      'We work together as a team and manage the time to finish the project as soon as possible.',

  },

  {

    title: 'What technologies do you use?',

    description:

      'Tredning technologies such as next.js, react, redux, javascript, typescript, framer motion, etc.',

  },

];

**Constant > navigation-data.ts**

type Navigation = {

  label: string;

  href: string;

};

export const navigationData: Navigation[] = [

  {

    label: 'Home',

    href: '#',

  },

  {

    label: 'About',

    href: '#about',

  },

  {

    label: 'Skills',

    href: '#skills',

  },

  {

    label: 'Projects',

    href: '#projects',

  },

  {

    label: 'FAQ',

    href: '#faq',

  },

  {

    label: 'Contacts',

    href: '#contacts',

  },

];

**Constant > projects-data.ts**

import { StaticImageData } from 'next/image';

import project1Src from '../../public/images/image1.png';

import project2Src from '../../public/images/image2.png';

import project3Src from '../../public/images/image3.png';

import project4Src from '../../public/images/image4.png';

import project5Src from '../../public/images/image5.png';

import project6Src from '../../public/images/image6.png';

type Project = {

  imageSrc: StaticImageData;

  name: string;

  description: string;

  skill1: string;

  skill2: string;

  skill3: string;

  link: string;

};

export const projectsData: Project[] = [

  {

    imageSrc: project1Src,

    name: 'Landing Page',

    description:

      'The BIGGEST community event in Indonesia, which will run for 30 full days with the theme.',

    skill1: 'React',

    skill2: 'HTML',

    skill3: 'CSS',

    link: 'http://www.community.com',

  },

  {

    imageSrc: project2Src,

    name: 'Landing Page',

    description:

      'Private hospital in the city of padang that provides special surgical services',

    skill1: 'React',

    skill2: 'HTML',

    skill3: 'CSS',

    link: 'http://www.privatehospital.com',

  },

  {

    imageSrc: project3Src,

    name: 'Landing Page',

    description:

      'Product engaged in WhatsApp API services and other information technology services.',

    skill1: 'React',

    skill2: 'HTML',

    skill3: 'CSS',

    link: 'http://apiservices.com',

  },

  {

    imageSrc: project4Src,

    name: 'Landing Page',

    description:

      'nusa.id cloud is part of the Nusanet Group (PT. Media Antar Nusa) of experience as an Internusa.',

    skill1: 'React',

    skill2: 'HTML',

    skill3: 'CSS',

    link: 'http://www.nusa.id',

  },

  {

    imageSrc: project5Src,

    name: 'Landing Page',

    description: 'UKUR is a constructor and interior marketplace in Indonesia.',

    skill1: 'React',

    skill2: 'HTML',

    skill3: 'CSS',

    link: 'http://www,ukur.com',

  },

  {

    imageSrc: project6Src,

    name: 'Landing Page',

    description:

      'Product engaged in WhatsApp API Services and other information technology services.',

    skill1: 'React',

    skill2: 'HTML',

    skill3: 'CSS',

    link: 'http://www.engaged.com',

  },

];

**Constant > skills-data.ts**

export interface Skill {

  name: string;

  level: number;

}

export const skillsData: Skill[] = [

  { name: 'HTML', level: 100 },

  { name: 'Redux', level: 85 },

  { name: 'JavaScript', level: 90 },

  { name: 'React', level: 90 },

  { name: 'CSS', level: 80 },

  { name: 'TypeScript', level: 70 },

];

**Skills-data.ts**

export interface Skill {

  name: string;

  proficiency: number;

  iconPath: string;

  iconWidth: number;

  iconHeight: number;

}

export const skillsData: Skill[] = [

  {

    name: 'HTML',

    proficiency: 100,

    iconPath: '/icons/html-icon.png',

    iconWidth: 46,

    iconHeight: 52

  },

  {

    name: 'CSS',

    proficiency: 80,

    iconPath: '/icons/css-icon.png',

    iconWidth: 46,

    iconHeight: 54

  },

  {

    name: 'JavaScript',

    proficiency: 90,

    iconPath: '/icons/javascript-icon.png',

    iconWidth: 52,

    iconHeight: 52

  },

  {

    name: 'React',

    proficiency: 90,

    iconPath: '/icons/react-icon.png',

    iconWidth: 52,

    iconHeight: 52

  },

  {

    name: 'Redux',

    proficiency: 85,

    iconPath: '/icons/redux-icon.png',

    iconWidth: 57,

    iconHeight: 52

  },

  {

    name: 'TypeScript',

    proficiency: 70,

    iconPath: '/icons/typescript-icon.png',

    iconWidth: 52,

    iconHeight: 52

  }

];

**Social-media.ts**

import iconGlobal from '../../public/icons/global-icon.png';

import iconInstagram from '../../public/icons/icon-instagram.svg';

import iconLinkedIn from '../../public/icons/icon-linkedin.svg';

type Icon = {

  href: string;

  src: string;

  alt: string;

};

export const socialMediaData: Icon[] = [

  {

    href: 'https://www.facebook.com/',

    src: iconGlobal,

    alt: 'Global',

  },

  {

    href: 'https://www.instagram.com/',

    src: iconInstagram,

    alt: 'Instagram',

  },

  {

    href: 'https://www.linkedin.com/',

    src: iconLinkedIn,

    alt: 'LinkedIn',

  },

];

**Tabs-data.ts**

import iconGlobal from '../../public/icons/global-icon.png';

import iconInstagram from '../../public/icons/icon-instagram.svg';

import iconLinkedIn from '../../public/icons/icon-linkedin.svg';

type Icon = {

  href: string;

  src: string;

  alt: string;

};

export const socialMediaData: Icon[] = [

  {

    href: 'https://www.facebook.com/',

    src: iconGlobal,

    alt: 'Global',

  },

  {

    href: 'https://www.instagram.com/',

    src: iconInstagram,

    alt: 'Instagram',

  },

  {

    href: 'https://www.linkedin.com/',

    src: iconLinkedIn,

    alt: 'LinkedIn',

  },

];

**Testimonials-data.ts**

import { StaticImageData } from 'next/image';

import profile1Src from '../../public/images/profile1.jpg';

import profile2Src from '../../public/images/profile2.jpg';

import profile3Src from '../../public/images/profile3.jpg';

type Testimonial = {

  rating: number;

  description: string;

  profileSrc: StaticImageData;

  profileName: string;

  profileOccupation: string;

};

export const testimonialsData: Testimonial[] = [

  {

    rating: 4,

    description:

      'Working with Syntax has been a game-changer for our business. Their IT solutions streamlined our operations and helped us scale faster than we expected. Highly recommended!',

    profileSrc: profile1Src,

    profileName: 'Michael Anderson',

    profileOccupation: 'CEO of FinTech Solutions',

  },

  {

    rating: 5,

    description:

      'The UI/UX design provided by Syntax completely transformed our platform. Our users love the new interface, and engagement rates have significantly increased',

    profileSrc: profile2Src,

    profileName: 'Sarah Thompson',

    profileOccupation: 'Head of Product at E-Hub',

  },

  {

    rating: 4,

    description:

      'Cybersecurity was a major concern for us, but Syntax delivered a secure and scalable solution that gave us peace of mind. Their team is truly professional and reliable.',

    profileSrc: profile3Src,

    profileName: 'David Collins',

    profileOccupation: 'CTO of MedTech Innovations',

  },

  {

    rating: 5,

    description:

      'Working with Syntax has been a game-changer for our business. Their IT solutions streamlined our operations and helped us scale faster than we expected. Highly recommended!',

    profileSrc: profile1Src,

    profileName: 'Michael Anderson',

    profileOccupation: 'CEO of FinTech Solutions',

  },

  {

    rating: 4,

    description:

      'The UI/UX design provided by Syntax completely transformed our platform. Our users love the new interface, and engagement rates have significantly increased',

    profileSrc: profile2Src,

    profileName: 'Sarah Thompson',

    profileOccupation: 'Head of Product at E-Hub',

  },

  {

    rating: 5,

    description:

      'Cybersecurity was a major concern for us, but Syntax delivered a secure and scalable solution that gave us peace of mind. Their team is truly professional and reliable.',

    profileSrc: profile3Src,

    profileName: 'David Collins',

    profileOccupation: 'CTO of MedTech Innovations',

  },

  {

    rating: 4,

    description:

      'Working with Syntax has been a game-changer for our business. Their IT solutions streamlined our operations and helped us scale faster than we expected. Highly recommended!',

    profileSrc: profile1Src,

    profileName: 'Michael Anderson',

    profileOccupation: 'CEO of FinTech Solutions',

  },

  {

    rating: 5,

    description:

      'The UI/UX design provided by Syntax completely transformed our platform. Our users love the new interface, and engagement rates have significantly increased',

    profileSrc: profile2Src,

    profileName: 'Sarah Thompson',

    profileOccupation: 'Head of Product at E-Hub',

  },

  {

    rating: 4,

    description:

      'Cybersecurity was a major concern for us, but Syntax delivered a secure and scalable solution that gave us peace of mind. Their team is truly professional and reliable.',

    profileSrc: profile3Src,

    profileName: 'David Collins',

    profileOccupation: 'CTO of MedTech Innovations',

  },

];

**Trusted-company-data.ts**

import { StaticImageData } from 'next/image';

import AdobeIcon from '../../public/icons/icon-adobe.svg';

import AirbnbIcon from '../../public/icons/icon-airbnb.svg';

import DatabricksIcon from '../../public/icons/icon-databricks.svg';

import DropboxIcon from '../../public/icons/icon-dropbox.svg';

import NetflixIcon from '../../public/icons/icon-netflix.svg';

import PaypalIcon from '../../public/icons/icon-paypal.svg';

import PostmanIcon from '../../public/icons/icon-postman.svg';

import UpworkIcon from '../../public/icons/icon-upwork.svg';

import ZoomIcon from '../../public/icons/icon-zoom.svg';

type TrustedCompanyImageProps = {

  src: StaticImageData;

  alt: string;

};

export const trustedCompaniesData: TrustedCompanyImageProps[] = [

  {

    src: AdobeIcon,

    alt: 'Adobe',

  },

  {

    src: AirbnbIcon,

    alt: 'Airbnb',

  },

  {

    src: DatabricksIcon,

    alt: 'Databricks',

  },

  {

    src: DropboxIcon,

    alt: 'Dropbox',

  },

  {

    src: NetflixIcon,

    alt: 'Netflix',

  },

  {

    src: PaypalIcon,

    alt: 'PayPal',

  },

  {

    src: PostmanIcon,

    alt: 'Postman',

  },

  {

    src: UpworkIcon,

    alt: 'Upwork',

  },

  {

    src: ZoomIcon,

    alt: 'Zoom',

  },

];

**Home > partials > about.tsx (the design)**

import Image from 'next/image';

import React from 'react';

const About = () => {

  return (

    <section

      className='custom-container items-center gap-6 md:gap-12'

      id='about'

    >

      <div className='flex items-center justify-center bg-gradient-to-r from-[#8746eb] to-[#dc49ab]'>

        <Image src='/images/hero.png' alt='Hero' width={818} height={459} />

      </div>

      <div className='custom-container flex flex-wrap items-start gap-6 py-5 md:gap-12 md:py-20'>

        <div className='flex-[1.5] basis-80'>

          <h2 className='text-neutral-25 mt-0 pt-2 text-3xl font-extrabold'>

            Who am I?

          </h2>

        </div>

        <div className='flex-[8.5] basis-80'>

          <p className='text-sm-regular md:text-md-regular mt-4'>

            Hi, I am Alex, a Front-End Developer & Web Programming Instructor

            based in Jakarta. With over{' '}

            <span className='font-bold'>3 years of experiences</span>, I

            specialize in crafting interactive, responsive, and user-friendly

            websites that deliver exceptional digital experiences. My expertise

            spans across various industries, including e-commerce, education,

            and SaaS. I am proficient in modern front-end technologies, clean

            coding practices, and performance optimization.

          </p>

        </div>

      </div>

      <div className='mt-8 flex flex-wrap justify-center gap-4 pb-20 sm:gap-6 lg:gap-2'>

        {[

          { title: '3+', subtitle: 'Years Experiences', gradient: false },

          { title: '30+', subtitle: 'Projects', gradient: true },

          { title: '90+', subtitle: 'Students', gradient: false },

          { title: '20+', subtitle: 'Happy Clients', gradient: true },

        ].map((item, index) => (

          <button

            key={index}

            className={`flex aspect-square w-[45vw] max-w-[273px] flex-col items-center justify-center rounded-full text-center text-xl font-bold sm:w-[45vw] lg:w-[273px] ${

              item.gradient

                ? 'border border-transparent bg-gradient-to-r from-[#8746eb] to-[#dc49ab] text-white'

                : 'border border-neutral-100 text-neutral-100'

            }`}

          >

            <div>

              <h3 className='text-xl font-bold'>{item.title}</h3>

              <p className='text-sm'>{item.subtitle}</p>

            </div>

          </button>

        ))}

      </div>

    </section>

  );

};

export default About;

**Careers.tsx**

'use client';

import { motion } from 'framer-motion';

import React from 'react';

import { careerData } from '@/constants/careers-data';

import { Section } from '../../../components/layouts/section';

interface CareerProps {

  title: string;

  company: string;

  description: string;

  experience?: string[];

  index: number;

}

const Career: React.FC<CareerProps> = ({

  title,

  company,

  description,

  experience,

  index,

}) => {

  return (

    <motion.div

      initial={{ opacity: 0, y: 20 }}

      whileInView={{ opacity: 1, y: 0 }}

      transition={{

        delay: index \* 0.5,

        duration: 0.5,

      }}

      viewport={{ once: false }}

      className='w-full'

    >

      <div className='grid grid-cols-[80px\_80px\_1fr] gap-4'>

        {/\* Empty Column \*/}

        <div className='w-8' />

        {/\* Timeline Column \*/}

        <motion.div

          className='relative'

          initial={{ opacity: 0 }}

          whileInView={{ opacity: 1 }}

          transition={{

            delay: index \* 0.5 + 0.2,

            duration: 0.5,

          }}

          viewport={{ once: false }}

        >

          {index < careerData.length - 1 && (

            <motion.div

              className='absolute left-1/2 h-full w-1 -translate-x-1/2 transform bg-gradient-to-b from-pink-500 to-pink-700'

              initial={{ height: 0 }}

              whileInView={{ height: '100%' }}

              transition={{

                delay: index \* 0.5 + 0.2,

                duration: 0.5,

                ease: 'easeOut',

              }}

              viewport={{ once: false }}

            />

          )}

          <motion.div

            className='absolute left-1/2 h-8 w-8 -translate-x-1/2 -translate-y-1/2 transform rounded-full border-4 border-primary-100 bg-gradient-to-br from-[#8746eb] to-[#dc49ab]'

            initial={{ scale: 0.5, opacity: 0 }}

            whileInView={{ scale: 1, opacity: 1 }}

            transition={{

              delay: index \* 0.5 + 0.2,

              duration: 0.5,

            }}

            viewport={{ once: false }}

            whileHover={{ scale: 1.05, x: 6 }}

            whileTap={{ scale: 0.9 }}

          />

        </motion.div>

        {/\* Right Column - Profile Details \*/}

        <motion.div

          className='space-y-4'

          initial={{ opacity: 0 }}

          animate={{ opacity: 1 }}

          transition={{ delay: 0.3 }}

        >

          <h1 className='text-2xl font-bold'>{title}</h1>

          <div className='flex items-center gap-4'>

            <span className='text-xl font-semibold text-white'>{company}</span>

            <span className='text-white'>•</span>

            <span className='text-white'>{description}</span>

          </div>

          {experience && (

            <ul className='list-inside list-disc space-y-1'>

              {experience.map((exp, index) => (

                <motion.li

                  key={index}

                  className='text-white'

                  whileHover={{ scale: 1.05, x: 6 }}

                  transition={{ duration: 0.2 }}

                >

                  {exp}

                </motion.li>

              ))}

            </ul>

          )}

        </motion.div>

      </div>

    </motion.div>

  );

};

const Careers = () => {

  return (

    <Section

      title='Career Journey'

      subtitle='A visual timeline of key milestones and experiences from over the years.'

      id='careers'

    >

      <div className='w-full max-w-[1200px]'>

        {careerData.map((career, index) => (

          <Career

            key={index}

            title={career.title}

            company={career.company}

            description={career.description}

            experience={career.experience}

            index={index}

          />

        ))}

      </div>

    </Section>

  );

};

export default Careers;

**Comparison.tsx**

import { motion } from 'framer-motion';

import React from 'react';

import { comparisonData } from '../../../constants/comparison-data';

interface SectionProps {

  title: string;

  subtitle: string;

  id: string;

  children: React.ReactNode;

}

const Section: React.FC<SectionProps> = ({ title, subtitle, id, children }) => {

  return (

    <section id={id} className='mb-16'>

      <div className='mb-8 text-center'>

        <h2 className='mb-2 text-4xl font-bold'>{title}</h2>

        <p className='text-xl text-gray-600'>{subtitle}</p>

      </div>

      {children}

    </section>

  );

};

const Comparison: React.FC = () => {

  return (

    <Section

      title='Why We Stand?'

      subtitle='A showcase of my unique approach and skill set compared to conventional front-end-developers'

      id='comparison'

    >

      <div className='custom-container flex flex-col items-center justify-center gap-4'>

        {/\* Header \*/}

        <div className='flex w-full items-center justify-center gap-4 rounded-full bg-gradient-to-r from-blue-500 to-indigo-500 p-2'>

          <div className='flex-[5.0] basis-80 text-center font-medium'>

            Skill

          </div>

          <div className='flex-[2.5] basis-80 text-center font-medium'>Me</div>

          <div className='flex-[2.5] basis-80 text-center font-medium'>

            Other

          </div>

        </div>

        {/\* Data Rows \*/}

        {comparisonData.map((tech) => (

          <motion.div

            key={tech.name}

            initial={{ opacity: 0, y: 20 }}

            animate={{ opacity: 1, y: 0 }}

            transition={{ duration: 0.5 }}

            className='flex w-full items-center gap-4'

          >

            <div className='flex-[5.0] basis-80 text-center'>{tech.name}</div>

            <div className='flex flex-[2.5] basis-80 items-center justify-center text-center text-2xl'>

              <span

                className={`${tech.me === '✅' ? 'text-green-600' : 'text-red-500'}`}

              >

                {tech.me}

              </span>

            </div>

            <div className='flex flex-[2.5] basis-80 items-center justify-center text-center text-2xl'>

              <span

                className={`${tech.other === '✅' ? 'text-green-600' : 'text-red-500'}`}

              >

                {tech.other}

              </span>

            </div>

          </motion.div>

        ))}

      </div>

    </Section>

  );

};

export default Comparison;

**Contact-form.tsx**

import { motion } from 'framer-motion';

import React from 'react';

import { comparisonData } from '../../../constants/comparison-data';

interface SectionProps {

  title: string;

  subtitle: string;

  id: string;

  children: React.ReactNode;

}

const Section: React.FC<SectionProps> = ({ title, subtitle, id, children }) => {

  return (

    <section id={id} className='mb-16'>

      <div className='mb-8 text-center'>

        <h2 className='mb-2 text-4xl font-bold'>{title}</h2>

        <p className='text-xl text-gray-600'>{subtitle}</p>

      </div>

      {children}

    </section>

  );

};

const Comparison: React.FC = () => {

  return (

    <Section

      title='Why We Stand?'

      subtitle='A showcase of my unique approach and skill set compared to conventional front-end-developers'

      id='comparison'

    >

      <div className='custom-container flex flex-col items-center justify-center gap-4'>

        {/\* Header \*/}

        <div className='flex w-full items-center justify-center gap-4 rounded-full bg-gradient-to-r from-blue-500 to-indigo-500 p-2'>

          <div className='flex-[5.0] basis-80 text-center font-medium'>

            Skill

          </div>

          <div className='flex-[2.5] basis-80 text-center font-medium'>Me</div>

          <div className='flex-[2.5] basis-80 text-center font-medium'>

            Other

          </div>

        </div>

        {/\* Data Rows \*/}

        {comparisonData.map((tech) => (

          <motion.div

            key={tech.name}

            initial={{ opacity: 0, y: 20 }}

            animate={{ opacity: 1, y: 0 }}

            transition={{ duration: 0.5 }}

            className='flex w-full items-center gap-4'

          >

            <div className='flex-[5.0] basis-80 text-center'>{tech.name}</div>

            <div className='flex flex-[2.5] basis-80 items-center justify-center text-center text-2xl'>

              <span

                className={`${tech.me === '✅' ? 'text-green-600' : 'text-red-500'}`}

              >

                {tech.me}

              </span>

            </div>

            <div className='flex flex-[2.5] basis-80 items-center justify-center text-center text-2xl'>

              <span

                className={`${tech.other === '✅' ? 'text-green-600' : 'text-red-500'}`}

              >

                {tech.other}

              </span>

            </div>

          </motion.div>

        ))}

      </div>

    </Section>

  );

};

export default Comparison;

**Faq.tsx**

import { motion } from 'framer-motion';

import React from 'react';

import { comparisonData } from '../../../constants/comparison-data';

interface SectionProps {

  title: string;

  subtitle: string;

  id: string;

  children: React.ReactNode;

}

const Section: React.FC<SectionProps> = ({ title, subtitle, id, children }) => {

  return (

    <section id={id} className='mb-16'>

      <div className='mb-8 text-center'>

        <h2 className='mb-2 text-4xl font-bold'>{title}</h2>

        <p className='text-xl text-gray-600'>{subtitle}</p>

      </div>

      {children}

    </section>

  );

};

const Comparison: React.FC = () => {

  return (

    <Section

      title='Why We Stand?'

      subtitle='A showcase of my unique approach and skill set compared to conventional front-end-developers'

      id='comparison'

    >

      <div className='custom-container flex flex-col items-center justify-center gap-4'>

        {/\* Header \*/}

        <div className='flex w-full items-center justify-center gap-4 rounded-full bg-gradient-to-r from-blue-500 to-indigo-500 p-2'>

          <div className='flex-[5.0] basis-80 text-center font-medium'>

            Skill

          </div>

          <div className='flex-[2.5] basis-80 text-center font-medium'>Me</div>

          <div className='flex-[2.5] basis-80 text-center font-medium'>

            Other

          </div>

        </div>

        {/\* Data Rows \*/}

        {comparisonData.map((tech) => (

          <motion.div

            key={tech.name}

            initial={{ opacity: 0, y: 20 }}

            animate={{ opacity: 1, y: 0 }}

            transition={{ duration: 0.5 }}

            className='flex w-full items-center gap-4'

          >

            <div className='flex-[5.0] basis-80 text-center'>{tech.name}</div>

            <div className='flex flex-[2.5] basis-80 items-center justify-center text-center text-2xl'>

              <span

                className={`${tech.me === '✅' ? 'text-green-600' : 'text-red-500'}`}

              >

                {tech.me}

              </span>

            </div>

            <div className='flex flex-[2.5] basis-80 items-center justify-center text-center text-2xl'>

              <span

                className={`${tech.other === '✅' ? 'text-green-600' : 'text-red-500'}`}

              >

                {tech.other}

              </span>

            </div>

          </motion.div>

        ))}

      </div>

    </Section>

  );

};

export default Comparison;

**Footer.tsx**

import Image from 'next/image';

import Link from 'next/link';

import React from 'react';

import { socialMediaData } from '@/constants/social-media-data';

const Footer = () => {

  return (

    <footer className='border-t border-neutral-800'>

      <div className='custom-container flex flex-row items-center justify-between gap-4 py-10'>

        <div className='flex items-center gap-4'>

          <Image

            src='/icons/logo.svg'

            alt='logo'

            width={40}

            height={40}

            className='h-10 w-10'

          />

          <p className='text-xs-regular text-neutral-400'>

            Another masterpiece by Edwin, trained at WebProgrammingHack bootcamp

          </p>

        </div>

        <div className='flex gap-4'>

          {socialMediaData.map((icon) => (

            <Link

              key={icon.alt}

              href={icon.href}

              className='flex-center size-10 rounded-full border border-neutral-800 p-2.5'

            >

              <Image src={icon.src} alt={icon.alt} className='h-full w-auto' />

            </Link>

          ))}

        </div>

      </div>

    </footer>

  );

};

export default Footer;

**Hero.tsx**

import Image from 'next/image';

import React from 'react';

import { Button } from '@/components/ui/button';

const Hero = () => {

  return (

    <section

      id='hero'

      className='custom-container md: flex flex-wrap items-center justify-center gap-10 overflow-hidden pt-28'

    >

      <Button variant='outline'>Alex&apos;s Profile</Button>

      <div className='space-y-6'>

        <h1 className='text-center text-4xl font-bold'>

          I am a{' '}

          <span className='bg-gradient-to-r from-[#dc49ab] to-[#8746EB] bg-clip-text text-transparent'>

            Front-End Developer

          </span>

          <br />& Web Programming Instructor

        </h1>

        <p className='text-md mx-auto max-w-[930px] text-justify font-normal'>

          Hi, I&apos;m Alex, a passionate web developer with over 3 years of

          experiences in creating responsive websites. I also teach aspiring

          developers to master modern web programming and bring their ideas to

          life.

        </p>

      </div>

      <div>

        <Button>Alex&apos;s Profile</Button>

      </div>

      <div className='relative mt-8 h-[100px] w-full'>

        <Image

          src='/images/background.png'

          alt='Background'

          fill

          className='object-cover'

          priority

        />

      </div>

    </section>

  );

};

export default Hero;

**Navbar.tsx**

'use client';

import { Menu } from 'lucide-react';

import { motion, useScroll, useTransform } from 'motion/react';

import Image from 'next/image';

import Link from 'next/link';

import React from 'react';

import { Button } from '@/components/ui/button';

import {

  Sheet,

  SheetTrigger,

  SheetContent,

  SheetClose,

} from '@/components/ui/sheet';

import { navigationData } from '@/constants/navigation-data';

const Navbar = () => {

  const { scrollY } = useScroll();

  const background = useTransform(

    scrollY,

    [0, 100],

    ['rgba(12,13,13,0', 'rgba(12,13,13,0.5)']

  );

  const backdropBlur = useTransform(

    scrollY,

    [0, 100],

    ['blur(0px)', 'blur(10px)']

  );

  return (

    <motion.header

      style={{

        background,

        backdropFilter: backdropBlur,

      }}

      className='fixed top-0 z-50 w-full'

    >

      <div className='flex-between custom-container h-16 md:h-21'>

        <Image

          src='/icons/logo.svg'

          alt='logo'

          width={40}

          height={40}

          className='max-md:h-8 max-md:w-35.25'

        />

        <nav className='hidden lg:block'>

          <ul className='flex-start gap-3'>

            {navigationData.map((data) => (

              <li key={data.label}>

                <Link href={data.href} className='hover:text-primary-200 p-4'>

                  {data.label}

                </Link>

              </li>

            ))}

          </ul>

        </nav>

        <Button asChild className='hidden flex-row items-center gap-2 lg:flex'>

          <div className='flex items-center gap-2'>

            <Image src='/icons/mail.png' alt='mail' width={30} height={30} />

            <Link href='#contacts'>Hire me</Link>

          </div>

        </Button>

        <Sheet>

          <SheetTrigger asChild>

            <Menu className='cursor-pointer lg:hidden' />

          </SheetTrigger>

          <SheetContent>

            <nav className='mt-16'>

              <ul className='flex flex-col gap-4'>

                {navigationData.map((data) => (

                  <li key={data.label}>

                    <SheetClose asChild>

                      <Link

                        href={data.href}

                        className='hover:text-primary-200 py-4'

                      >

                        {data.label}

                      </Link>

                    </SheetClose>

                  </li>

                ))}

              </ul>

            </nav>

            <Button asChild className='mt-3 w-full'>

              <SheetClose asChild>

                <Link href='#contact'>Hire me</Link>

              </SheetClose>

            </Button>

          </SheetContent>

        </Sheet>

      </div>

    </motion.header>

  );

};

export default Navbar;

**Projects.tsx**

import { StaticImageData } from 'next/image';

import Image from 'next/image';

import React from 'react';

import { Section } from '@/components/layouts/section';

import { projectsData } from '@/constants/projects-data';

const Projects = () => {

  return (

    <Section id='projects' title='' subtitle='' bg-red-500>

      <Cards>

        {projectsData.map((project) => (

          <Card

            key={project.name}

            imageSrc={project.imageSrc}

            name={project.name}

            description={project.description}

            skill1={project.skill1}

            skill2={project.skill2}

            skill3={project.skill3}

            link={project.link}

          />

        ))}

      </Cards>

    </Section>

  );

};

export default Projects;

type CardsProps = {

  children: React.ReactNode;

};

const Cards: React.FC<CardsProps> = ({ children }) => {

  return <div className='flex flex-wrap gap-6 md:gap-5'>{children}</div>;

};

type CardProps = {

  imageSrc: StaticImageData;

  name: string;

  description: string;

  skill1: string;

  skill2: string;

  skill3: string;

  link: string;

};

const Card: React.FC<CardProps> = ({

  imageSrc,

  name,

  description,

  skill1,

  skill2,

  skill3,

  link,

}) => {

  return (

    <div className='flex-1 basis-70'>

      <Image

        src={imageSrc}

        alt={name}

        className='aspect-square rounded-2xl object-cover md:rounded-4xl'

      />

      <h4 className='text-lg-semibold text-neutral-25 mt-4'>{name}</h4>

      <p className='text-md-regular mt-2 text-white'>{description}</p>

      <div className='mt-4 flex gap-2'>

        <button className='rounded-lg border-2 border-white px-4 py-2 text-white transition-colors hover:bg-blue-50'>

          {skill1}

        </button>

        <button className='rounded-lg border-2 border-white px-4 py-2 text-white transition-colors hover:bg-blue-50'>

          {skill2}

        </button>

        <button className='rounded-lg border-2 border-white px-4 py-2 text-white transition-colors hover:bg-blue-50'>

          {skill3}

        </button>

      </div>

      <div className='mt-4'>

        <a

          href={link}

          className='text-blue-600 transition-colors hover:text-blue-700'

        >

          Visit Project

        </a>

      </div>

    </div>

  );

};

Scrollup.tsx

import { MoveUp } from 'lucide-react';

import Link from 'next/link';

import React from 'react';

import { Button } from '@/components/ui/button';

const ScrollUp = () => {

  return (

    <Button

      variant='outline'

      size='icon'

      className='bg-base-background fixed right-10 bottom-10'

      asChild

    >

      <Link href='#hero'>

        <MoveUp />

      </Link>

    </Button>

  );

};

export default ScrollUp;

**Skill-motion.tsx (optional)**

import { motion } from 'framer-motion';

import Image from 'next/image';

import React from 'react';

// Define types for the component

interface SkillIcon {

  src: string;

  alt: string;

  width: number;

  height: number;

}

interface SkillMotionProps {

  className?: string;

  containerHeight?: string;

  containerWidth?: string;

}

// Define animation properties as constants

const animationProps = {

  animate: {

    rotate: [0, 360],

    x: [0, 50, 0, -50, 0],

    y: [0, -50, 0, 50, 0],

  },

  transition: {

    duration: 10,

    repeat: Infinity,

    ease: 'linear',

  },

};

// Define skill icons as constants

const skillIcons: SkillIcon[] = [

  { src: '/icons/css-icon.png', alt: 'CSS', width: 46, height: 54 },

  { src: '/icons/redux-icon.png', alt: 'Redux', width: 57, height: 52 },

  {

    src: '/icons/typescript-icon.png',

    alt: 'TypeScript',

    width: 52,

    height: 52,

  },

  { src: '/icons/html-icon.png', alt: 'HTML', width: 46, height: 52 },

  {

    src: '/icons/javascript-icon.png',

    alt: 'JavaScript',

    width: 52,

    height: 52,

  },

  { src: '/icons/react-icon.png', alt: 'React', width: 52, height: 52 },

];

const Skillmotion: React.FC<SkillMotionProps> = ({ className = '', containerHeight = 'clamp(21.25rem, 52.73vw, 39.81rem)', containerWidth = '100%' }) => {

  return (

    <div

      className={`relative ${className}`}

      style={{ height: containerHeight, width: containerWidth }}

    >

        {/\* CSS Icon \*/}

        <div className='absolute top-[100px] right-[75px]'>

          <motion.div {...animationProps} className='mb-4'>

            <Image

              src={skillIcons[0].src}

              alt={skillIcons[0].alt}

              width={skillIcons[0].width}

              height={skillIcons[0].height}

            />

          </motion.div>

        </div>

        {/\* Redux Icon \*/}

        <div className='absolute top-[200px] right-[200px]'>

          <motion.div {...animationProps} className='mb-4'>

            <Image

              src={skillIcons[1].src}

              alt={skillIcons[1].alt}

              width={skillIcons[1].width}

              height={skillIcons[1].height}

            />

          </motion.div>

        </div>

        {/\* TypeScript Icon \*/}

        <div className='absolute top-[350px] right-[150px]'>

          <motion.div {...animationProps} className='mb-4'>

            <Image

              src={skillIcons[2].src}

              alt={skillIcons[2].alt}

              width={skillIcons[2].width}

              height={skillIcons[2].height}

            />

          </motion.div>

        </div>

        {/\* HTML Icon \*/}

        <div className='absolute top-[70px] left-[200px]'>

          <motion.div {...animationProps} className='mb-4'>

            <Image

              src={skillIcons[3].src}

              alt={skillIcons[3].alt}

              width={skillIcons[3].width}

              height={skillIcons[3].height}

            />

          </motion.div>

        </div>

        {/\* JavaScript Icon \*/}

        <div className='absolute top-[230px] left-[90px]'>

          <motion.div {...animationProps} className='mb-4'>

            <Image

              src={skillIcons[4].src}

              alt={skillIcons[4].alt}

              width={skillIcons[4].width}

              height={skillIcons[4].height}

            />

          </motion.div>

        </div>

        {/\* React Icon \*/}

        <div className='absolute top-[400px] left-[160px]'>

          <motion.div {...animationProps} className='mb-4'>

            <Image

              src={skillIcons[5].src}

              alt={skillIcons[5].alt}

              width={skillIcons[5].width}

              height={skillIcons[5].height}

            />

          </motion.div>

        </div>

      </div>

  );

};

export default Skillmotion;

**Skills.tsx**

import { motion } from 'framer-motion';

import Image from 'next/image';

import React from 'react';

import { Section } from '@/components/layouts/section';

import { skillsData, Skill } from '../../../constants/skills-data';

const Skills = () => {

  const skillPositions = [

    { top: 100, left: null, right: 75 },

    { top: 200, left: null, right: 200 },

    { top: 235, left: null, right: 150 },

    { top: 70, left: 200, right: null },

    { top: 155, left: 90, right: null },

    { top: 300, left: 160, right: null },

  ];

  return (

    <Section

      title='My Core Skills'

      subtitle='An overview of the key technolgies and frameworks I specialize in.'

      id='skills'

    >

      <div className='w-full'>

        <div className='custom-container flex flex-wrap items-center gap-7 overflow-hidden md:gap-10'>

          {/\* Skills icons section \*/}

          <div

            className='flex-start relative flex-[4.7] basis-80'

            style={{ height: 'clamp(10rem, 40vw, 25rem)' }}

          >

            {skillsData.map((skill, index) => {

              const position = skillPositions[index];

              return (

                <div

                  key={skill.name}

                  className={`absolute ${position.left ? `left-[${position.left}px]` : `right-[${position.right}px]`} top-[${position.top}px]`}

                >

                  <motion.div

                    animate={{

                      x: [0, 50, 0, -50, 0],

                      y: [0, -25, 0, -25, 0],

                    }}

                    transition={{

                      duration: Math.random() \* 5 + 5,

                      repeat: Infinity,

                      ease: 'linear',

                      delay: index \* 0.5,

                    }}

                    className='mb-4'

                  >

                    <Image

                      src={skill.iconPath}

                      alt={skill.name}

                      width={skill.iconWidth}

                      height={skill.iconHeight}

                    />

                  </motion.div>

                </div>

              );

            })}

          </div>

          {/\* Skill progress bars section \*/}

          <div className='flex-[5.3] basis-80 flex-col justify-start'>

            <div className='pt-2'>

              <div className='space-y-3'>

                {skillsData.map((skill: Skill, index) => (

                  <motion.div

                    key={skill.name}

                    initial={{ opacity: 0, y: 20 }}

                    whileInView={{ opacity: 1, y: 0 }}

                    transition={{

                      delay: index \* 0.5,

                      duration: 0.5,

                      ease: 'easeOut',

                    }}

                    viewport={{ once: false }}

                  >

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

                      <span className='text-sm-regular bg-gradient-to-r from-white to-gray-300 bg-clip-text text-transparent'>

                        {skill.name}

                      </span>

                      <span className='text-sm-semibold bg-gradient-to-r from-white to-gray-300 bg-clip-text text-transparent'>

                        {skill.proficiency}%

                      </span>

                    </div>

                    <motion.div

                      className='h-2 w-full rounded-full bg-black'

                      initial={{ width: '0%' }}

                      whileInView={{ width: `${skill.proficiency}%` }}

                      transition={{

                        delay: index \* 0.5,

                        duration: 1,

                        ease: 'easeOut',

                      }}

                      viewport={{ once: false }}

                    >

                      <div className='h-full rounded-full bg-gradient-to-r from-blue-500 to-purple-500'></div>

                    </motion.div>

                  </motion.div>

                ))}

              </div>

            </div>

          </div>

        </div>

      </div>

    </Section>

  );

};

export default Skills;

**Teksaja.tsx**

import React from 'react';

const teksaja = () => {

  return (

    <div className='custom-container flex pt-5'>

      <div className='w-1/2 text-left font-medium'>

        <h2 className='text-3xl font-bold text-white'>

          Experience in Front-End Development

        </h2>

      </div>

      <div className='w-1/2 text-right font-medium'>

        <p className='leading-relaxed text-white'>

          I have experience developing 30+ web projects across various

          industries, including marketplaces, health, fashion, sports, and more.

          Implemented 1000+ responsive web pages with interactive features and

          smooth animations

        </p>

      </div>

    </div>

  );

};

export default teksaja;

**Testimonial.tsx**

import { StaticImageData } from 'next/image';

import Image from 'next/image';

import React from 'react';

import { Section } from '@/components/layouts/section';

import { Marquee } from '@/components/ui/marquee';

import { testimonialsData } from '@/constants/testimonials-data';

const Testimonials = () => {

  return (

    <Section

      title='People Say About Me'

      subtitle='Hear from clients and colleagues about their experiences working with me.'

      id='testimonial'

    >

      <div className='custom-container relative flex flex-col gap-8'>

        {/\* First Row \*/}

        <div className='w-full overflow-hidden mix-blend-luminosity'>

          <Marquee className='w-full py-4 [--duration:120s] md:max-w-[calc(100%+160px)]'>

            {testimonialsData

              .filter((\_, index) => index % 2 === 0)

              .map((testimonial, index) => (

                <TestimonialCard

                  key={index}

                  description={testimonial.description}

                  profileSrc={testimonial.profileSrc}

                  profileName={testimonial.profileName}

                  profileOccupation={testimonial.profileOccupation}

                />

              ))}

          </Marquee>

        </div>

        {/\* Second Row \*/}

        <div className='w-full overflow-hidden mix-blend-luminosity'>

          <Marquee className='w-full py-4 [--duration:120s] md:max-w-[calc(100%+160px)]'>

            {testimonialsData

              .filter((\_, index) => index % 2 !== 0)

              .map((testimonial, index) => (

                <TestimonialCard

                  key={index}

                  description={testimonial.description}

                  profileSrc={testimonial.profileSrc}

                  profileName={testimonial.profileName}

                  profileOccupation={testimonial.profileOccupation}

                />

              ))}

          </Marquee>

        </div>

      </div>

    </Section>

  );

};

export default Testimonials;

type TestimonialCardProps = {

  description: string;

  profileSrc: StaticImageData;

  profileName: string;

  profileOccupation: string;

};

const TestimonialCard: React.FC<TestimonialCardProps> = ({

  description,

  profileSrc,

  profileName,

  profileOccupation,

}) => {

  return (

    <div className='bg-base-background relative overflow-hidden rounded-[10px] border-2 border-transparent p-6 transition-all duration-300 hover:border-[2px] hover:border-[var(--gradient-border)] hover:[--gradient-border:linear-gradient(to\_right,violet,pink)]'>

      <div className='mt-4 flex items-start gap-4 md:mt-6'>

        <Image

          src={profileSrc}

          alt={profileName}

          className='h-12 w-12 rounded-full'

        />

        <div>

          <p className='text-sm-bold md:text-md-semibold text-neutral-200'>

            {profileName}

          </p>

          <p className='text-sm-regular md:text-md-regular text-white'>

            {profileOccupation}

          </p>

        </div>

      </div>

      <p className='text-sm-regular md:text-md-regular text-neutral-25 mt-4 mb-2 line-clamp-6 max-h-[240px] max-w-[320px] leading-relaxed md:max-h-[260px] md:max-w-[450px]'>

        {description}

      </p>

    </div>

  );

};

**Trusted-by.tsx (margquee)**

import Image from 'next/image';

import React from 'react';

import { Marquee } from '@/components/ui/marquee';

import { trustedCompaniesData } from '@/constants/trusted-companies-data';

const TrustedBy = () => {

  return (

    <div className='mt-6.5 md:mt-20'>

      <h3 className='text-md-semibold md:text-xl-semibold text-neutral-25 text-center'>

        Trusted by Leading Companies Worldwide

      </h3>

      <CompaniesMarquee />

    </div>

  );

};

export default TrustedBy;

const CompaniesMarquee = () => {

  return (

    <div className='custom-container relative flex h-28.5 items-center md:h-50'>

      <div className='w-full overflow-hidden mix-blend-luminosity'>

        <Marquee className='py-4'>

          {trustedCompaniesData.map((data) => (

            <Image

              key={data.alt}

              src={data.src}

              alt={data.alt}

              className='h-full w-auto object-contain select-none'

              style={{

                height: 'clamp(2.13rem, 3.97vw, 3rem)',

              }}

            />

          ))}

        </Marquee>

        <div className='from-base-background absolute inset-y-0 left-0 w-[15%] bg-gradient-to-r to-transparent' />

        <div className='from-base-background absolute inset-y-0 right-0 w-[15%] bg-gradient-to-l to-transparent' />

      </div>

    </div>

  );

};

**Usecase-tabs.tsx (not in portfolio 2)**

import Image, { StaticImageData } from 'next/image';

import Link from 'next/link';

import React from 'react';

import { Section } from '@/components/layouts/section';

import { Button } from '@/components/ui/button';

import { Tabs, TabsContent, TabsList, TabsTrigger } from '@/components/ui/tabs';

import { tabsData } from '@/constants/tabs-data';

const UseCaseTabs = () => {

  return (

    <Section

      title='Tailored IT Solutions for Every Industry'

      subtitle='From finance to healthcare and retail, we provide cutting-edge technology solutions to drive efficiency, security, and growth in your industry'

    >

      <Tabs defaultValue={tabsData[0].key} orientation='horizontal'>

        <TabsList>

          {tabsData.map((tab) => (

            <TabsTrigger key={tab.key} value={tab.key}>

              {tab.label}

            </TabsTrigger>

          ))}

        </TabsList>

        {tabsData.map((tab) => {

          const { key, ...contentProps } = tab;

          return (

            <TabsContent key={key} value={key}>

              <Content {...contentProps} />

            </TabsContent>

          );

        })}

      </Tabs>

    </Section>

  );

};

export default UseCaseTabs;

type ContentProps = {

  title: string;

  description: string;

  imageSrc: StaticImageData;

};

const Content: React.FC<ContentProps> = ({ title, description, imageSrc }) => {

  return (

    <div className='flex-between mx-auto max-w-232.5 flex-wrap gap-5 pt-6 md:gap-25 md:pt-17'>

      <div className='flex-[7.4] basis-75'>

        <h3 className='display-xs-bold md:display-xl-bold text-neutral-25'>

          {title}

        </h3>

        <p className='text-sm-regular md:text-md-regular mt-3 text-neutral-400 md:mt-4'>

          {description}

        </p>

        <Button asChild className='mt-6 w-full px-19.5 md:mt-12 md:w-fit'>

          <Link href='#'>Detail</Link>

        </Button>

      </div>

      <Image

        src={imageSrc}

        alt={title}

        className='aspect-square flex-[2.6] basis-75 rounded-2xl object-cover'

      />

    </div>

  );

};

**Lib > utils.ts**

import { clsx, type ClassValue } from "clsx"

import { twMerge } from "tailwind-merge"

export function cn(...inputs: ClassValue[]) {

  return twMerge(clsx(inputs))

}

-----

**.gitignore**

# See https://help.github.com/articles/ignoring-files/ for more about ignoring files.

# dependencies

/node\_modules

/.pnp

.pnp.\*

.yarn/\*

!.yarn/patches

!.yarn/plugins

!.yarn/releases

!.yarn/versions

# testing

/coverage

# next.js

/.next/

/out/

# production

/build

# misc

.DS\_Store

\*.pem

# debug

npm-debug.log\*

yarn-debug.log\*

yarn-error.log\*

.pnpm-debug.log\*

# env files (can opt-in for committing if needed)

.env\*

# vercel

.vercel

# typescript

\*.tsbuildinfo

next-env.d.ts

.preetiierrc

{

  "plugins": ["prettier-plugin-tailwindcss"],

  "singleQuote": true,

  "jsxSingleQuote": true,

  "semi": true,

  "tabWidth": 2,

  "bracketSpacing": true,

  "jsxBracketSameLine": false,

  "arrowParens": "always",

  "trailingComma": "es5",

  "printWidth": 80

}

**Components.json**

{

  "$schema": "https://ui.shadcn.com/schema.json",

  "style": "new-york",

  "rsc": true,

  "tsx": true,

  "tailwind": {

    "config": "",

    "css": "src/app/globals.css",

    "baseColor": "slate",

    "cssVariables": true,

    "prefix": ""

  },

  "aliases": {

    "components": "@/components",

    "utils": "@/lib/utils",

    "ui": "@/components/ui",

    "lib": "@/lib",

    "hooks": "@/hooks"

  },

  "iconLibrary": "lucide"

}

**Eslint.config.mjs**

import { dirname } from 'path';

import { fileURLToPath } from 'url';

import { FlatCompat } from '@eslint/eslintrc';

import importPlugin from 'eslint-plugin-import';

const \_\_filename = fileURLToPath(import.meta.url);

const \_\_dirname = dirname(\_\_filename);

const compat = new FlatCompat({

  baseDirectory: \_\_dirname,

});

const eslintConfig = [

  ...compat.extends('next/core-web-vitals', 'next/typescript', 'prettier'),

  {

    plugins: {

      import: importPlugin,

    },

    rules: {

      'import/order': [

        'error',

        {

          groups: [

            'builtin',

            'external',

            'internal',

            ['parent', 'sibling'],

            'index',

            'object',

          ],

          'newlines-between': 'always',

          pathGroups: [

            {

              pattern: '@/components/\*\*',

              group: 'internal',

              position: 'before',

            },

            {

              pattern: '@app/\*\*',

              group: 'external',

              position: 'after',

            },

          ],

          pathGroupsExcludedImportTypes: ['builtin'],

          alphabetize: {

            order: 'asc',

            caseInsensitive: true,

          },

        },

      ],

    },

    ignores: ['components/ui/\*\*'],

  },

  {

    files: ['\*\*/\*.ts', '\*\*/\*.tsx'],

    rules: {

      '@typescript-eslint/no-explicit-any': 'off',

      '@typescript-eslint/ban-ts-comment': 'off',

    },

  },

];

export default eslintConfig;

------------------

**Next.config.ts**

import type { NextConfig } from 'next';

const nextConfig: NextConfig = {

  reactStrictMode: true,

};

export default nextConfig;

Package.json

{

  "name": "it-company-portfolio",

  "version": "0.1.0",

  "private": true,

  "scripts": {

    "dev": "next dev --turbopack",

    "build": "next build",

    "start": "next start",

    "lint": "next lint"

  },

  "dependencies": {

    "@emailjs/browser": "^4.4.1",

    "@hookform/resolvers": "^5.0.1",

    "@radix-ui/react-accordion": "^1.2.3",

    "@radix-ui/react-checkbox": "^1.1.5",

    "@radix-ui/react-dialog": "^1.1.7",

    "@radix-ui/react-label": "^2.1.3",

    "@radix-ui/react-slot": "^1.2.0",

    "@radix-ui/react-tabs": "^1.1.4",

    "class-variance-authority": "^0.7.1",

    "clsx": "^2.1.1",

    "embla-carousel-react": "^8.6.0",

    "lucide-react": "^0.487.0",

    "motion": "^12.6.3",

    "next": "15.2.4",

    "react": "^19.0.0",

    "react-dom": "^19.0.0",

    "react-hook-form": "^7.55.0",

    "react-spinners": "^0.16.1",

    "react-use": "^17.6.0",

    "tw-animate-css": "^1.2.5",

    "zod": "^3.24.3"

  },

  "devDependencies": {

    "@eslint/eslintrc": "^3",

    "@iconify/react": "^5.2.1",

    "@tailwindcss/postcss": "^4",

    "@types/node": "^20",

    "@types/react": "^19",

    "@types/react-dom": "^19",

    "eslint": "^9",

    "eslint-config-next": "15.2.4",

    "eslint-config-prettier": "^10.1.1",

    "eslint-plugin-import": "^2.31.0",

    "eslint-plugin-prettier": "^5.2.5",

    "prettier": "^3.5.3",

    "prettier-plugin-tailwindcss": "^0.6.11",

    "tailwind-merge": "^3.1.0",

    "tailwindcss": "^4",

    "tailwindcss-intersect": "^2.2.0",

    "tailwindcss-motion": "^1.1.0",

    "typescript": "^5"

  }

}

**Postcss.config.mjs**

const config = {

  plugins: ['@tailwindcss/postcss'],

};

export default config;

**Tsconfig.json**

{

  "compilerOptions": {

    "target": "ES2017",

    "lib": ["dom", "dom.iterable", "esnext"],

    "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"

      }

    ],

    "paths": {

      "@/\*": ["./src/\*"]

    }

  },

  "include": ["next-env.d.ts", "\*\*/\*.ts", "\*\*/\*.tsx", ".next/types/\*\*/\*.ts"],

  "exclude": ["node\_modules"]

}