Pulse UI

Next.js 16 + shadcn/ui Admin Dashboard Template

Developer Documentation

Installation | Running | Building | Deployment | Customization

Published by Codervent - codervent.com

Table of Contents

1. Introduction 3

2. Requirements 3

3. Installation 4

4. Running the Project 4

5. Project Structure 5

6. Routing and Layouts 5

7. Building for Production 6

8. Deployment 7

9. Customization 8

10. Troubleshooting 11

11. Support and License 11

1. Introduction

Thank you for purchasing Pulse UI. Pulse UI is a modern, responsive admin dashboard template built with Next.js 16 (App Router), React 19, TypeScript, Tailwind CSS v4 and shadcn/ui. It ships with ready-made dashboards, apps, tables, forms, charts, authentication pages, switchable UI themes and a large set of reusable components so you can start your next Next.js admin project quickly.

Key technologies

Area	Technology
Framework	Next.js 16 (App Router), React 19, TypeScript
Styling	Tailwind CSS 4 (via PostCSS), tw-animate-css
UI components	shadcn/ui, Radix UI, Lucide icons
Icon fonts	Bootstrap Icons and Boxicons (loaded from CDN in the root layout)
Font	Inter, loaded with next/font/google
Forms and validation	React Hook Form, Zod
Tables	TanStack Table
Charts	Recharts, ApexCharts
State and theming	Zustand (UI theme store), custom ThemeProvider (light, dark and system mode)

2. Requirements

Make sure the following tools are installed on your computer before you begin.

Tool	Version	Notes
Node.js	20.9 or newer	Minimum required by Next.js 16. Node.js 22 LTS is recommended. Download from nodejs.org.
Package manager	npm 10+ / pnpm / yarn	npm is installed together with Node.js.
Code editor	Any	Visual Studio Code is recommended.
Git	Optional	Useful for version control and deployment.

Check your versions in a terminal:

```
node -v
npm -v
```

3. Installation

Follow these steps to get the template running locally.

Step 1 - Extract the package

Unzip the downloaded file you received after purchase. Open the extracted folder that contains the project source (the folder with package.json) in your code editor.

Step 2 - Open a terminal in the project folder

```
cd path/to/pulse-ui-next
```

Step 3 - Install dependencies

```
# using npm
npm install

# or using pnpm
pnpm install

# or using yarn
yarn install
```

> **Peer dependency errors?**
> If npm reports a peer dependency conflict (common with React 19 packages), run `npm install --legacy-peer-deps` instead.

4. Running the Project

Start the development server:

```
npm run dev
```

Next.js starts the dev server (Turbopack is the default bundler in Next.js 16) on port 3000. Fast Refresh updates the page automatically whenever you save a file.

> **Open the correct URL**
> The template is preconfigured with a base path of `/pulse-ui-next` in `next.config.ts`. Open `http://localhost:3000/pulse-ui-next` in your browser. To serve the app from the root (`http://localhost:3000`), see the Deployment section.

Available scripts

Command	Description
`npm run dev`	Starts the Next.js development server with Fast Refresh.
`npm run build`	Creates an optimized production build.
`npm run start`	Serves the production build with the Next.js Node.js server. Run `npm run build` first.
`npm run lint`	Runs ESLint across the project to find code problems.

5. Project Structure

A typical layout of the source code is shown below. Folder names may differ slightly depending on your version of the template.

```
pulse-ui-next/
 |- public/                 Static assets (favicon, images, theme backgrounds)
 |- src/
 | |- app/
 | | |- (layout-pages)/     Pages with sidebar and header (dashboards, apps ...)
 | | |- (no-layout-pages)/  Pages without layout (login, register, errors ...)
 | | |- layout.tsx          Root layout: metadata, font, theme script, providers
 | | |- page.tsx            Home route
 | | |- globals.css         Tailwind, theme variables and UI themes
 | |- components/
 | | |- ui/                 shadcn/ui primitives (button, card, dialog ...)
 | | |- layout/             Admin layout, header and footer
 | | |- app-sidebar.tsx     Sidebar and menu configuration
 | | |- theme-customizer.tsx  UI theme switcher panel
 | | |- ...                 calendar, file-manager, data-table, forms and more
 | |- hooks/                Custom React hooks
 | |- lib/                  Helpers such as utils.ts (cn function)
 | |- providers/            theme-provider.tsx and ui-theme-provider.tsx
 | |- store/                ui-theme.store.ts (Zustand)
 |- components.json         shadcn/ui configuration
 |- next.config.ts          Next.js configuration (base path, export ...)
 |- postcss.config.js       PostCSS and Tailwind CSS 4
 |- eslint.config.mjs       ESLint configuration
 |- package.json            Dependencies and scripts
 |- tsconfig.json           TypeScript configuration
```

Path alias

The project uses the `@` alias for the `src` folder, for example `import { Button } from "@/components/ui/button"`.

6. Routing and Layouts

Pulse UI uses the Next.js App Router. Every folder inside `src/app` that contains a `page.tsx` file becomes a URL. Two route groups decide which layout a page uses. Route group names in parentheses are not part of the URL.

Route group	Use it for
`(layout-pages)`	Pages that need the sidebar and header: dashboards, apps, tables, forms, charts, account pages.
`(no-layout-pages)`	Standalone pages without the admin layout: login, register, password reset, error and coming soon pages.

Adding a new page

- Create a folder with a `page.tsx` file inside the correct route group.
- Add a menu item for it in `src/components/app-sidebar.tsx`.
- Add `"use client"` at the top of a file that uses React state, effects or browser APIs.

```
// src/app/(layout-pages)/my-page/page.tsx
export default function MyPage() {
  return (
    <div className="p-4">
      <h1 className="text-2xl font-semibold">My Page</h1>
    </div>
  )
}
```

Adding a sidebar menu item

The sidebar menu is defined in the `navMain` array inside `src/components/app-sidebar.tsx`. Follow the same url format used by the existing items.

```
// src/components/app-sidebar.tsx  (inside data.navMain)
{
  title: "My Page",
  url: "my-page",
  icon: House,            // any icon imported from "lucide-react"
},

// menu item with a submenu
{
  title: "Reports",
  url: "#",
  icon: ChartNoAxesCombined,
  items: [
    { title: "Sales", url: "reports/sales" },
    { title: "Traffic", url: "reports/traffic" },
  ],
},
```

7. Building for Production

When you are ready to publish your app, create a production build:

```
npm run build
```

The optimized build is generated in the `.next` folder. You can run it with the Next.js server to test it locally:

```
npm run start
```

Static export (optional)

Pulse UI does not need server-side features, so you can also export it as plain static files and host it on any static hosting service. Enable export in `next.config.ts`:

```
// next.config.ts
import type { NextConfig } from "next"

const nextConfig: NextConfig = {
  output: "export",              // writes static files to the out folder
  basePath: "/pulse-ui-next",    // remove to serve from the domain root
  images: { unoptimized: true }, // required for static export
}

export default nextConfig
```

Run `npm run build` and the static site is generated in the `out` folder. Upload the contents of that folder to your host.

> **Build fails?**
> The build type-checks and lints your code. Fix the file named in the error and run the build again. The Inter font is downloaded from Google Fonts during the build, so the build machine needs internet access.

8. Deployment

Pulse UI can be deployed in two ways: as a Node.js app (Option A) or as static files (Option B). Both use the same code.

Option A - Node.js hosting

Vercel (recommended)

- Push the project to GitHub, GitLab or Bitbucket and import it in Vercel.
- Framework preset: Next.js (detected automatically). Build command: `npm run build`.
- No `vercel.json` file is needed. Remove `output: "export"` from next.config.ts if you added it.

Your own server (VPS)

Install Node.js 20.9+, copy the project to the server and run:

```
npm install
npm run build
npm run start
```

Keep the process running with a process manager such as PM2:

```
npm install -g pm2
pm2 start npm --name "pulse-ui" -- start
pm2 save
```

Then put Nginx in front of it as a reverse proxy:

```
server {
    listen 80;
    server_name example.com;

    location / {
        proxy_pass http://127.0.0.1:3000;
        proxy_http_version 1.1;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection "upgrade";
    }
}
```

Option B - Static hosting (export)

Enable `output: "export"` as shown in the Building section, run `npm run build` and use the `out` folder.

Netlify

- Build command: `npm run build`. Publish directory: `out`.

Apache (shared hosting / cPanel)

Upload the contents of `out` to `public_html` and add an `.htaccess` file so URLs without the .html extension work:

```
<IfModule mod_rewrite.c>
  RewriteEngine On
  RewriteCond %{REQUEST_FILENAME} !-d
  RewriteCond %{REQUEST_FILENAME}.html -f
  RewriteRule ^(.*)$ $1.html [L]
</IfModule>
ErrorDocument 404 /404.html
```

Nginx (static files)

```
server {
    listen 80;
    server_name example.com;
    root /var/www/pulse-ui/out;
    index index.html;

    location / {
        try_files $uri $uri.html $uri/ =404;
    }
    error_page 404 /404.html;
}
```

Deploying under a sub-folder or at the domain root

The base path controls the URL prefix of every page and asset. Change it in `next.config.ts` and rebuild:

```
// next.config.ts
const nextConfig: NextConfig = {
  basePath: "/admin",   // app lives at example.com/admin
  // basePath: "",      // or remove it to serve from example.com/
}
```

> **Update these files when you change the base path**
> 1) The theme background images in `src/app/globals.css` use absolute urls that include the base path, for example `url("/pulse-ui-next/images/bg-themes/dark-blue.png")`. Change the prefix in every url(...) there. 2) Update `metadataBase`, the canonical and Open Graph urls in `src/app/layout.tsx`. 3) For static hosting, upload the files so they are reachable under the same path (for example into a `pulse-ui-next` folder).

9. Customization

Changing theme colors

Theme colors are CSS variables defined in `src/app/globals.css`. The default light theme is in `:root` and the default dark theme is in `.dark`. Edit these values to change the look of the whole template, for example the primary color:

```
/* src/app/globals.css */
:root {
  --primary: oklch(0.55 0.25 285);
  --background: oklch(1 0 0);
  --sidebar: oklch(1 0 0);
}
.dark {
  --primary: oklch(0.7 0.2 285);
  --background: oklch(0.15 0.02 285);
}
```

Light and dark mode

Light, dark and system mode are handled by `src/providers/theme-provider.tsx`. The selected mode is saved in the browser under the localStorage key `theme`. Read or change it from any client component with the `useTheme()` hook:

```
"use client"
import { useTheme } from "@/providers/theme-provider"

const { theme, resolvedTheme, setTheme } = useTheme()
setTheme("dark")   // "light" | "dark" | "system"
```

UI themes

On top of light and dark mode, Pulse UI includes switchable UI themes. Each theme is a CSS class on the `<html>` element, defined in `src/app/globals.css`.

Theme class	Description
`dark-blue`	Default theme. Dark blue glass look with a background image.
`gaussian-black`	Black glass look with a background image.
`classic-light`	Light theme with a soft pastel gradient background.
`semi-dark`	Dark sidebar with light page content.

How it works: the selected UI theme is stored with Zustand in `src/store/ui-theme.store.ts` (localStorage key `ui-theme`), applied to the `<html>` element by `src/providers/ui-theme-provider.tsx`, and switched by the panel in `src/components/theme-customizer.tsx`.

Changing the default theme

Visitors who already chose a theme keep their saved choice. For new visitors, set the default in these places and keep them in sync:

```
// 1. src/store/ui-theme.store.ts
theme: "dark-blue",                   // default UI theme

// 2. src/app/layout.tsx  (inside the themeScript string)
var ui = "dark-blue";                 // fallback before the store loads

// 3. src/app/layout.tsx  (light, dark or system mode)
<ThemeProvider defaultTheme="dark">
```

To test as a new visitor, clear the localStorage keys `theme` and `ui-theme` in your browser developer tools and reload.

No flash of the wrong theme

The server cannot read localStorage, so without help the page would first paint in the light theme and then switch. To prevent this, `src/app/layout.tsx` contains a small script in `<head>` that adds the saved classes to `<html>` before the first paint. Keep this script, and keep `suppressHydrationWarning` on the `<html>` tag. If you add a theme, nothing changes in the script because it reads the theme name from storage.

Adding your own UI theme

- Copy one of the theme blocks in `globals.css` (for example `.dark-blue`), rename the class and change the variables.
- Add the class name to the `CUSTOM_THEMES` list in `ui-theme-provider.tsx`.
- Add a choice in `theme-customizer.tsx` that calls `setTheme("ocean")` from `useUIThemeStore`.

```
/* src/app/globals.css */
.ocean {
  --background: oklch(0.2 0.03 230);
  --foreground: oklch(0.98 0 0);
  --primary: oklch(0.7 0.15 220);
  --primary-foreground: oklch(1 0 0);
  --sidebar: oklch(0.18 0.03 230);
  /* copy the remaining variables from .dark-blue */
}

// src/providers/ui-theme-provider.tsx
const CUSTOM_THEMES = [
  "dark-blue",
  "classic-light",
  "gaussian-black",
  "semi-dark",
  "ocean",
]
```

Theme background images

The background images used by the glass themes are stored in `public/images/bg-themes` and referenced from `globals.css`. Replace the image files, or change the url(...) values, to use your own backgrounds.

Sidebar

The sidebar menu, user block and project links are defined in the `data` object at the top of `src/components/app-sidebar.tsx`. The logo and brand name are in the `SidebarHeader` section of the same file. When the sidebar is collapsed to icons, hovering it expands it with a blur (glass) effect. This comes from the `sidebar-hover-glass` class applied to the `<Sidebar>` element and defined in `globals.css`.

SEO and metadata

Page title, description, keywords, Open Graph and Twitter tags are defined in the `metadata` export in `src/app/layout.tsx`. Update these values for your product:

```
// src/app/layout.tsx
export const metadata: Metadata = {
  metadataBase: new URL("https://your-domain.com"),
  title: { default: "Your App Name", template: "%s | Your App Name" },
  description: "A short description of your app.",
  // keywords, authors, openGraph, twitter ...
}
```

Individual pages can set their own title by exporting `metadata` from their `page.tsx` (server components only). Put your social image at `public/og-image.png` (1200 x 630 pixels recommended).

Adding shadcn/ui components

Additional shadcn/ui components can be added with the shadcn CLI:

```
npx shadcn@latest add dialog
```

Changing logo, name and favicon

- Replace the favicon at `src/app/favicon.ico` and the images inside the `public` folder.
- Update the title, description and icons in the `metadata` export of `layout.tsx`.
- Update the brand name and logo in the sidebar header in `app-sidebar.tsx`.

10. Troubleshooting

Problem	Solution
npm install fails with ERESOLVE	Run `npm install --legacy-peer-deps`.
Errors about the Node.js version	Upgrade Node.js to 20.9 or newer.
Blank page, or CSS and JS files return 404	The app runs under the base path. Open `/pulse-ui-next`, or change `basePath` in next.config.ts (see Deployment).
Theme background image is missing	The urls in globals.css include the base path. Update them to match your `basePath`.
Page flashes in the light theme on refresh	Keep the theme script in the `<head>` of layout.tsx and `suppressHydrationWarning` on the `<html>` tag.
Changes to globals.css or config files do not show	Stop the dev server, delete the `.next` folder and run `npm run dev` again.
Build fails while downloading the Inter font	The build needs internet access to reach Google Fonts, or switch to `next/font/local`.
404 when refreshing a page on static hosting	Add the rewrite rules for your host (see Option B in Deployment).
Port 3000 already in use	Stop the other process or run `npm run dev -- -p 4000`.
Old dependencies after updating	Delete node_modules and the lock file, then run npm install again.

11. Support and License

If you need help, please contact us through the marketplace where you purchased the template or visit codervent.com. When reporting an issue, include your Node.js version, the command you ran, and the full error message so we can help faster.

Your license allows you to use this template according to the license type you purchased (Regular or Extended). Please review the license terms on the marketplace page. Redistributing or reselling the template source code as-is is not permitted.

Thank you!

If you enjoy Pulse UI, please leave a rating on the marketplace. It helps us keep improving the template.