CODERVENT

Syndron

Shadcn UI + Next.js 16
Admin Dashboard Template

DOCUMENTATION

Installation, setup, customization and deployment guide

Version 1.0 | October 2026
codervent.com
store.codervent.com/syndron-ui-next

Table of Contents

Quick Start . . . 3

1. Introduction . . . 3

1.1 Key Features . . . 3

1.2 Technology Stack . . . 4

1.3 What You Need to Know . . . 4

2. System Requirements . . . 4

2.1 Install Node.js . . . 4

3. Package Contents and Folder Structure . . . 5

4. Installation . . . 6

5. Running the Project . . . 7

5.1 Development server . . . 7

5.2 Available scripts . . . 7

6. Building for Production . . . 7

7. Customization . . . 8

7.1 Branding: logo, favicon and metadata . . . 8

7.2 Theme colors and dark mode . . . 8

7.3 Adding a new page . . . 8

7.4 Adding shadcn/ui components . . . 9

7.5 Removing unused pages . . . 9

8. Deployment . . . 9

8.1 Deploy to Vercel . . . 9

8.2 Deploy to Netlify . . . 10

8.3 Deploy to a VPS (Ubuntu with PM2 and Nginx) . . . 10

8.4 Deploy with Docker . . . 11

8.5 Static export (shared hosting, GitHub Pages, S3) . . . 11

9. Updating and Maintenance . . . 12

10. Troubleshooting . . . 12

11. Support, License and Credits . . . 13

11.1 Getting help . . . 13

11.2 License . . . 13

11.3 Third-party credits . . . 13

Quick Start

Already comfortable with Node.js? These are the only commands you need. Each step is explained in detail in the sections that follow.

```
# 1. Extract the downloaded package and open the project folder
cd admin-template

# 2. Install dependencies
npm install

# 3. Start the development server
npm run dev

# 4. Open http://localhost:3000 in your browser
```

Note
You need **Node.js 20.9 or later**. Run `node -v` to check your version. See Section 2 for full requirements.

1. Introduction

Thank you for purchasing **Syndron**. Syndron is a modern, responsive admin dashboard template built with **Next.js 16**, **React 19**, **Tailwind CSS 4**, **shadcn/ui** and **TypeScript**. It is designed for SaaS products, CRM and analytics platforms, internal tools and eCommerce back offices, and gives you a production-ready starting point so you can focus on your own features.

1.1 Key Features

- Next.js 16 with the App Router and TypeScript
- Tailwind CSS 4 and shadcn/ui components that you fully own and can edit
- Multiple dashboard variations, charts and analytics components
- Data tables with sorting and pagination
- Forms with validation, authentication pages, invoice and profile pages
- Light and dark mode with a fully responsive layout

1.2 Technology Stack

Technology	Purpose
Next.js 16	React framework: routing, server rendering, bundling and optimization
React 19	UI library
TypeScript	Static typing for safer, easier-to-maintain code
Tailwind CSS 4	Utility-first styling, configured through PostCSS
shadcn/ui	Accessible, customizable component source code added to your project
ESLint	Code quality and consistency checks

1.3 What You Need to Know

Basic familiarity with HTML, CSS, JavaScript and React is recommended. Knowledge of Next.js, TypeScript and Tailwind CSS will help when customizing the template, but you can follow this guide to install, run and deploy it without deep expertise.

2. System Requirements

Requirement	Details
Node.js	Version 20.9 or later. The current LTS release is recommended.
Package manager	npm 10 or later (installed with Node.js). The project ships with a `package-lock.json`, so npm is the default choice. pnpm, Yarn and Bun also work.
Code editor	Visual Studio Code is recommended, with the ESLint and Tailwind CSS IntelliSense extensions.
Git (optional)	Useful for version control and for deploying to hosting platforms.
Browser	Latest version of Chrome, Edge, Firefox or Safari.
Operating system	Windows 10/11, macOS or Linux.

2.1 Install Node.js

Download the LTS installer from **nodejs.org** and run it. Afterwards, confirm the installation from a terminal (Command Prompt, PowerShell, Terminal or your editor's built-in terminal):

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

Both commands should print a version number. If Node.js is older than 20.9, install a newer version before continuing.

3. Package Contents and Folder Structure

After purchase, download the package from your account and extract the ZIP file. The Next.js project is contained in the `admin-template` folder. Open this folder in your editor; it is the root of the application.

File / Folder	Description
`src/`	Application source code: routes and pages, components, hooks, utilities and styles. This is where you will spend most of your time.
`public/`	Static assets served from the site root, such as images, logos, icons and fonts.
`components.json`	Configuration for the shadcn/ui CLI (style, aliases and paths).
`next.config.ts`	Next.js configuration (image settings, redirects, export mode and so on).
`package.json`	Project scripts and the list of dependencies.
`package-lock.json`	Locks exact dependency versions so every install is identical.
`postcss.config.*`	PostCSS setup used by Tailwind CSS 4.
`tsconfig.json`	TypeScript compiler configuration and path aliases.
`eslint.config.*`	ESLint rules for the project.
`.env.local`	Local environment variables. Keep this file private.
`.gitignore`	Files and folders excluded from Git.
`next-env.d.ts`	Auto-generated TypeScript definitions for Next.js. Do not edit.
`README.md`	Short project overview.
`node_modules/`	Installed dependencies. Created by `npm install`. Never edit or upload this folder.
`.next/`	Build cache created when you run or build the app. Safe to delete at any time.
`out/`	Static export output, created only when static export is enabled (see Section 8.5).

Note
Folders such as `node_modules`, `.next` and `out` are generated automatically. You may not see them until you install dependencies or build the project.

4. Installation

Step 1: Extract the package

Unzip the downloaded file to a location of your choice. Avoid very long folder paths and folders synced in real time by cloud drives, as these can slow down installs on Windows.

Step 2: Open the project in a terminal

```
cd path/to/admin-template
```

In Visual Studio Code you can also use **File > Open Folder**, select `admin-template`, then open the integrated terminal with `Ctrl + ``.

Step 3: Install dependencies

```
npm install
```

This downloads every package listed in `package.json` into `node_modules`. It can take a few minutes depending on your connection. Using another package manager?

```
pnpm install        # pnpm
yarn install        # Yarn
bun install         # Bun
```

> **Tip**
> If npm reports an **ERESOLVE** peer dependency conflict, run `npm install --legacy-peer-deps`. This is common with newer React versions and is generally safe.

Step 4: Configure environment variables

Environment variables hold configuration that changes between environments, such as site URLs or API keys. The project reads them from the `.env.local` file in the project root. Open it and review each value. A typical entry looks like this (example format only):

```
NEXT_PUBLIC_APP_URL=http://localhost:3000
```

Rule	Explanation
Public variables	Variables beginning with `NEXT_PUBLIC_` are embedded in the browser bundle. Never put secrets in them.
Private variables	Variables without the prefix are available only on the server.
Restart required	Restart the dev server after changing `.env.local`.
Never commit	Do not commit `.env.local` to Git or share it publicly.

5. Running the Project

5.1 Development server

```
npm run dev
```

Open **http://localhost:3000** in your browser. The development server supports Fast Refresh, so changes to your code appear instantly without a full reload. To use a different port:

```
npm run dev -- -p 3001
```

5.2 Available scripts

Command	What it does
`npm run dev`	Starts the development server with hot reloading.
`npm run build`	Creates an optimized production build.
`npm run start`	Serves the production build (run `build` first).
`npm run lint`	Checks the code with ESLint.

> **Note**
> Script names are defined in the `scripts` section of `package.json`. Check that file for the complete and current list.

6. Building for Production

Before deploying, create and test a production build locally:

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

The build step compiles TypeScript, optimizes pages and bundles assets. If it finishes without errors, `npm run start` serves the optimized app at **http://localhost:3000**. To use another port: `npm run start -- -p 8080`.

Pre-deployment checklist

- Replace the app name, logo, favicon and metadata with your own branding.
- Review `.env.local` and add the same variables to your hosting platform.
- Remove demo data and any pages you do not need.
- Run `npm run lint` and `npm run build` with no errors.
- Confirm that no secrets are exposed through `NEXT_PUBLIC_` variables.

7. Customization

7.1 Branding: logo, favicon and metadata

- Place your logo and favicon files in the `public/` folder and update the references in the layout components.
- Update the page title and description in the root layout (typically `src/app/layout.tsx`) using the Next.js metadata API.

```
export const metadata = {
  title: "Your App Name",
  description: "A short description of your application.",
};
```

7.2 Theme colors and dark mode

Tailwind CSS 4 and shadcn/ui use CSS variables for colors, radius and other design tokens. These are defined in the global stylesheet (typically `src/app/globals.css`), with separate values for light and dark mode. Change the variables to restyle the entire dashboard in one place.

```
:root {
  --primary: oklch(0.55 0.22 275);
  --radius: 0.625rem;
}
```

Tip
The exact variable names and color format are defined in your stylesheet. Edit the existing values rather than adding new ones.

7.3 Adding a new page

Next.js uses file-based routing. Create a folder inside the app directory and add a `page.tsx` file. The folder name becomes the URL.

```
// src/app/my-page/page.tsx   ->   http://localhost:3000/my-page
export default function MyPage() {
  return <h1 className="text-2xl font-semibold">My new page</h1>;
}
```

To make the page reachable from the sidebar, add an entry to the navigation configuration used by the sidebar component.

7.4 Adding shadcn/ui components

Syndron is configured for the shadcn/ui CLI through `components.json`. Add any component from the shadcn/ui library with one command:

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

The component source code is copied into your project, so you can edit it freely.

7.5 Removing unused pages

Delete the unused route folders from the app directory and remove their links from the sidebar navigation. Then run `npm run build` to confirm nothing else depended on them.

8. Deployment

Next.js can be deployed in several ways. Choose the one that fits your project:

Option	Best for	Needs a Node.js server?
Vercel	Easiest deployment, built by the Next.js team	Managed for you
Netlify	Git-based deployments with a managed platform	Managed for you
VPS / own server	Full control, custom domains and infrastructure	Yes
Docker	Containerized and portable deployments	Yes (in container)
Static export	Shared hosting, GitHub Pages, S3 and similar	No

8.1 Deploy to Vercel

- Push your project to a GitHub, GitLab or Bitbucket repository.
- Sign in to **vercel.com**, click **Add New Project** and import the repository.
- Vercel detects Next.js automatically. Keep the default build settings.
- Add your environment variables under **Settings > Environment Variables**.
- Click **Deploy**. Every future push to your main branch redeploys automatically.

Or deploy from the command line:

```
npm install -g vercel
vercel          # preview deployment
vercel --prod   # production deployment
```

8.2 Deploy to Netlify

- Push the project to a Git repository and choose **Add new site > Import an existing project** in Netlify.
- Use `npm run build` as the build command. Netlify detects Next.js and applies its Next.js runtime automatically.
- Add your environment variables in the site settings, then deploy.

8.3 Deploy to a VPS (Ubuntu with PM2 and Nginx)

Install Node.js on the server, upload your project (without `node_modules` and `.next`), then build and start it:

```
npm install
npm run build
sudo npm install -g pm2
pm2 start npm --name "syndron" -- start
pm2 save
pm2 startup
```

Use Nginx as a reverse proxy so your domain points to the app. Create a site config such as `/etc/nginx/sites-available/syndron`:

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

    location / {
        proxy_pass http://127.0.0.1:3000;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
    }
}
```

Enable the site, reload Nginx and add a free HTTPS certificate:

```
sudo ln -s /etc/nginx/sites-available/syndron /etc/nginx/sites-enabled/
sudo nginx -t && sudo systemctl reload nginx
sudo certbot --nginx -d yourdomain.com
```

8.4 Deploy with Docker

Create a file named `Dockerfile` in the project root:

```
FROM node:22-alpine
WORKDIR /app
COPY package*.json ./
RUN npm ci
COPY . .
RUN npm run build
EXPOSE 3000
CMD ["npm", "start"]
```

```
docker build -t syndron .
docker run -p 3000:3000 --env-file .env.local syndron
```

Tip
Add a `.dockerignore` file containing `node_modules` and `.next` to keep the image small and builds fast.

8.5 Static export (shared hosting, GitHub Pages, S3)

If you do not need server-side features, Next.js can generate plain HTML, CSS and JavaScript files that run on any static host. Open `next.config.ts` and enable export mode:

```
import type { NextConfig } from "next";

const nextConfig: NextConfig = {
  output: "export",
  images: { unoptimized: true },
  trailingSlash: true,
};

export default nextConfig;
```

Run `npm run build`. The finished site is created in the `out/` folder. Upload the **contents** of `out/` to your host's public web directory.

Warning
Static export does not support API routes, middleware, server actions or the built-in image optimizer. If your site lives in a sub-folder (for example *example.com/admin*), also set `basePath: "/admin"` in the configuration.

9. Updating and Maintenance

- **Template updates:** Download the latest version from your account. Compare it with your project using a diff tool or Git before copying changes, so you do not overwrite your own customizations.
- **Dependencies:** Run `npm outdated` to see available updates and `npm update` to apply minor and patch updates.
- **Major upgrades:** Upgrade one major dependency at a time and read its release notes. Run `npm run build` after each step.
- **Backups:** Keep your project in Git so you can always roll back.

10. Troubleshooting

Problem	Solution
Unsupported Node.js version, or `node` not found	Install Node.js 20.9 or later from nodejs.org and reopen your terminal.
Port 3000 is already in use	Stop the other process or run `npm run dev -- -p 3001`.
ERESOLVE / peer dependency error during install	Run `npm install --legacy-peer-deps`.
Strange errors or missing modules after moving the project	Delete `node_modules` and `.next`, then run `npm install` again. On Windows: `rmdir /s /q node_modules .next`.
Styles are missing or not updating	Restart the dev server, delete the `.next` folder and confirm that the global stylesheet is imported in the root layout.
Environment variable changes have no effect	Restart the dev server. `NEXT_PUBLIC_` values are fixed at build time, so rebuild before deploying.
Build fails with TypeScript errors	Run `npx tsc --noEmit` to list the errors, fix them, then build again.
PowerShell blocks npm scripts on Windows	Run `Set-ExecutionPolicy RemoteSigned -Scope CurrentUser` in PowerShell, or use Command Prompt instead.
Hydration warning in the browser console	Often caused by browser extensions. Test in a private window.
404 on page refresh after static export	Enable `trailingSlash: true` and make sure the host serves `index.html` files from folders.

11. Support, License and Credits

11.1 Getting help

Before contacting support, please read this guide and check the Troubleshooting section. When you ask for help, include your Node.js version, your operating system, the exact command you ran and the full error message. This lets us respond much faster.

Resource	Link
Product page	store.codervent.com/syndron-ui-next
Website	codervent.com

11.2 License

Your use of Syndron is governed by the license you purchased. Please review the license terms on the platform where you bought the template to understand how many projects you may use it in and whether the end product can be sold or redistributed.

11.3 Third-party credits

Syndron is built on excellent open-source software, including Next.js, React, Tailwind CSS and shadcn/ui. A complete list of libraries is available in the `dependencies` section of `package.json`. Each library remains the property of its respective authors and is subject to its own license.

> **Tip**
> If you enjoy Syndron, a short review on the marketplace where you purchased it helps us keep improving the template. Thank you!