About the template

Nilam is a professional and minimal looking React, Nextjs, Typescript domain selling and domain auction template made with tailwind css framework. Nilam is made with working contact forms with Google SMTP so you may use Nilam directly for parking or sell any domain without any extra functionality. It's responsive for all of the devices and designed with a clean look and feel, stylish & modern trend. It comes with six unique layout which includes tons of sections, customization options and features for multiple purposes.

Requirement:

1. Node Version: > 10.13.0
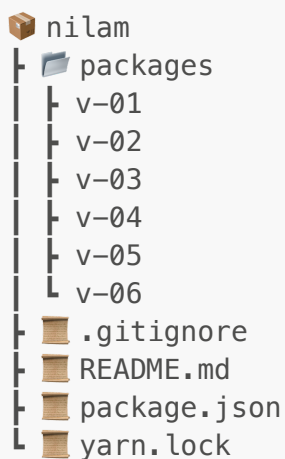2. Yarn Version: 1.22.24
3. Gmail account for SMTP configuration

Technology we use

1. React (A JavaScript library for building user interfaces)
2. NextJs (React SSR and SSG Framework)
3. Create Next App
4. yarn workspaces
5. Nodemailer (Used to sending emails through SMTP)
6. Tailwind CSS A utility-first CSS framework.

Demos

- Version one
- Version two
- Version three
- Version four
- Version five
- Version six

Folder Structure

```
📦 nilam
├ 📁 packages
│ ├ v-01
│ ├ v-02
│ ├ v-03
│ ├ v-04
│ ├ v-05
│ └ v-06
├ 📄 .gitignore
├ 📄 README.md
├ 📄 package.json
└ 📄 yarn.lock
```

```
within individual packages
📦 v-01/
├ 📁 public/
│ ├ 📁 images/
│ │ ├ 🖼 banner-bg-female.jpg
│ │ ├ 🖼 banner-bg-sale.jpg
│ │ ├ 🖼 banner-bg.jpg.jpg
│ │ ├ 🖼 banner-parallax-01.png
│ │ ├ 🖼 banner-parallax-02.png
│ │ ├ 🖼 bg-v4.jpg
│ │ ├ 🖼 domain-01.jpg
│ │ ├ 🖼 header-bg.jpg
│ │ ├ 🖼 hero-bg.jpeg
│ │ └ 🖼 hero-bg.jpg
│ ├ 🖼 favicon.ico
│ └ vercel.svg
├ 📁 src/
│ ├ 📁 components/
│ │ ├ 📁 Home/
│ │ │ ├ 📁 Cta/
│ │ │ │ └ 🖼 Cta.tsx
│ │ │ ├ 📁 Features/
│ │ │ │ └ 🖼 Features.tsx
│ │ │ ├ 📁 Hero/
│ │ │ │ └ 🖼 Hero.tsx
│ │ │ ├ 📁 Pricing/
│ │ │ │ └ 🖼 Pricing.tsx
│ │ │ └ 📁 Service/
│ │ │ └ 🖼 Service.tsx
│ │ ├ 🖼 Footer.tsx
│ │ └ 🖼 Layout.js
│ ├ 📁 pages/
│ │ ├ 📁 api/
│ │ │ └ 🖼 contact.js
│ │ ├ 🖼 _app.tsx
│ │ └ 🖼 index.tsx
│ └ 📁 styles/
│ └ 🖼 globals.css
├ 🖼 .eslintrc
├ 🖼 .gitignore
├ 🖼 README.md
├ 🖼 next-env.d.ts
├ 🖼 next.config.js
├ 🖼 package-lock.json
├ 🖼 package.json
├ 🖼 postcss.config.js
├ 🖼 tsconfig.json
├ 🖼 tailwind.config.js
└ 🖼 yarn.lock
├ 🖼 vercel.json
```

Getting Started

Go to project root like:

```
cd nilam
```

configure SMTP:

1. Go to the Google Developers Console
2. Select Other(Custom Name) option from select app
3. Give a name of your project and click generate.
4. Copy the password.

Setup Instructions

- Go to the version you want to use in nilam packages folder. create .env.local file in the root of version. And add the code like below.

```
USER_EMAIL=YOUREMAIL_ADDRESS
APP_PASSWORD=YOUR_APP_PASSWORD
```

At the end your `.evn.local` file should be look like:

```
USER_EMAIL=test@yourmail.com
APP_PASSWORD=ouase3435jowew4
```

Second, run the development server:

step-2: within the project root run

```
yarn
# or
yarn install
```

step-3: after successful installation of required packages run you required project or package like:

```
yarn dev:v-01  # to run the v-01 package or project
# or
yarn dev:v-02  # to run the v-02 package or project
# or
yarn dev:v-03  # to run the v-03 package or project
# or
```

```
yarn dev:v-04  # to run the v-04 package or project
# or
yarn dev:v-05  # to run the v-05 package or project
# or
yarn dev:v-06  # to run the v-06 package or project
```

Open http://localhost:3000 with your browser to see the result.

You can start editing the page by modifying `src/pages/index.tsx`. The page auto-updates as you edit the file. You can change the site title, description, metadata in `component/Layout.js` file. All components area in components folder.

NOTE:

please check the `package.json` file for other useful scripts like `clean`

> if you want to clean node_modules and build's folder run

```
yarn clean
```

Learn More

To learn more about Next.js, take a look at the following resources:

- Next.js Documentation - learn about Next.js features and API.
- Learn Next.js - an interactive Next.js tutorial.

Deployment

Deploy on Vercel(previously known as ZEIT Now)

The easiest way to deploy your Next.js app is to use the Vercel Now Platform step-1: go to your project root like:

```
cd nilam
```

step-2: run deployment command like:

```
deploy-vercel:v-01
```

which is promoted for some information fill your information or just stay with the defaults as

```
$ vercel
? Set up and deploy "~/your-directory-path"? [Y/n] y
```

```
? Which scope do you want to deploy to? your-scope
? Link to existing project? [y/N] n
? What's your project's name? your-project-name
? In which directory is your code located? ./
Auto-detected project settings (Next.js):
- Build Command: `next build` or `build` from `package.json`
- Output Directory: Next.js default
- Development Command: next dev --port $PORT
? Want to override the settings? [y/N] N
```

IMPORTANT:

before deployment you need to setup your env variable on vercel go to your project root and set your env values within the `vercel.json` file. (you can also setup env value by others method supported by verecl for more details [follow the link](#))

NOTE:

make sure you had the [vercel-cli](#) install and logged in your account.

Thank you for choosing the product. If you need any other support please create a support ticket. If you love the item we will appreciate if you provide us a 5* rating.