

Mosaic Documentation

Unzip the file that you have downloaded. Here is what you will find inside the main folder:

```
[template folder]/
├─ mosaic-sketch.zip
├─ mosaic-html.zip
├─ mosaic-react.zip
└─ mosaic-vue.zip
```

- **mosaic-sketch.zip** contains the design files for Sketch
- **mosaic-html.zip** contains the HTML template
- **mosaic-react.zip** contains the React template
- **mosaic-vue.zip** contains the Vue template

HTML template

Unzip and open the folder of the HTML template with your favourite editor (I am using [VSCode](#)).

The HTML template is shipped with Webpack just to let you quickly get into the development process. We don't discourage you from using any other build tool!

The first thing to do is to **install NPM dependencies**. To do that, open the terminal, type `npm install` and press the `Enter` key. This command will create a `node_modules` folder in the root of your template, which contains all required dependencies.

Feel confused? 🤔 [Watch this video!](#)

Start a development server

Run `npm start` in the terminal to run a development server and open <http://localhost:8080/> to view it in the browser. The page will reload every time you make edits.

Any kind of change should be made inside the `src` folder. Inside that, you will find the HTML template pages, alongside with styles ([src/css](#)), images ([src/images](#)), and some JavaScript ([src/js](#)).

In order to make the content easier to read and maintain, we have sliced the HTML pages content into partials by taking advantages of `html-loader` for Webpack (<https://webpack.js.org/loaders/html-loader/>). Partial HTML files are located in the [src/partials](#) folder. No extra configuration is required.

CSS & JS customisations

There isn't much to say about the CSS when it comes to templates that are built on the top of Tailwind CSS, so I would recommend to check out the [Tailwind documentation](#).

As for JavaScript, we have started adopting third-party libraries (e.g. [Alpine.js](#), [AOS](#), and more) instead of custom code. Check out the [src/js/main.js](#) file to see how to import the JS dependencies.

Compile and minify for production

When you have done with changes, run `npm run build` for compiling and minify for production. This command will generate a `dist` folder in the root of your template that you can upload via FTP to your server.

Deploy with Netlify

You can alternatively host your site on [Netlify](#): our templates come with a ready-made `netlify.toml` file to allow you deploying with Netlify and go live in a few clicks. You just need to create a repository from the HTML template, create a new site from Git in Netlify, and deploy the app. For more information, refer to Netlify documentation: <https://docs.netlify.com/>

TL;DR

[Check out the video](#) to see how to start a development server, make changes, and deploy your site via FTP or with Netlify.

React template

Unzip and open the folder of the React template with your favourite editor (I am using [VSCode](#)).

These React templates are bootstrapped with [Create React App](#), and [Craco](#) for overriding the default CRA Webpack configuration.

The first thing to do is to **install NPM dependencies**. To do that, open the terminal, type `npm install` and press the [Enter](#) key.

Feel confused? 🤔 [Watch this video!](#)

- ! We've been adopting CRA for demo purposes only. In case you want to use a framework built on the top of React (like Gatsby or Next.JS), you would probably need to import components into your project, and make a few adjustments (for example: import styles and images, configure routing, et cetera).

Unfortunately we do not provide support with product customisation, tutorials/guides for non-technical people, neither do we guarantee that our products will be compatible with third party software or services.

Start a development server

Run `npm start` or `yarn start` in the terminal to run a development server and open `http://localhost:3000/` to view it in the browser. The page will reload every time you make edits.

Any kind of change should be made inside the `src` folder. Inside that, you will find components, alongside with styles (`src/css`), images (`src/images`), and some JavaScript (`src/js`).

CSS customisations

There isn't much to say about the CSS when it comes to templates that are built on the top of Tailwind CSS, so I would recommend to check out the [Tailwind documentation](#).

Files structure

All React component files are stored in the `src/` folder 📌

```
src/
|
├─ pages/
|   (contains site pages)
|
├─ partials/
|   (e.g. header, footer, page sections, etc)
|
└─ utils/
    (contains more units, like modal, dropdown, etc)
```

Add or remove pages

Every template comes with a set of ready-made pages, provided in form of components and stored into the `src/pages/` directory.

Router settings are defined in [src/App.js](#) file so, in case you want to add or remove pages, you should need to add or remove routes in that file. Check out the [React Router documentation](#) to know more.

Build the app for production

When you have done with changes, run `npm run build` or `yarn build` for compiling and minify for production. You can learn more on the [Create React App documentation](#).

Vue template

Unzip and open the folder of the Vue template with your favourite editor (I am using [VSCode](#)).

The first thing to do is to install NPM dependencies. To do that, open the terminal, type `npm install` and press the [Enter](#) key.

Feel confused? 🤔 [Watch this video!](#)

- ! We are shipping our templates with a very basic Vue.js configuration for demonstration purposes only, and to let you quickly get into the development process. In case you want to use a framework built on the top of React (like Nuxt.js), you would probably need to import components into your project, and make a few adjustments (for example: import styles and images, configure routing, et cetera).

Unfortunately we do not provide support with product customisation, tutorials/guides for non-technical people, neither do we guarantee that our products will be compatible with third party software or services.

Start a development server

Run `npm run dev` in the terminal to run a development server and open <http://localhost:8080/> to view it in the browser. The page will reload every time you make edits.

Any kind of change should be made inside the `src` folder. Inside that, you will find components, alongside with styles ([src/css](#)), images ([src/images](#)), and some JavaScript ([src/js](#)).

CSS customisations

There isn't much to say about the CSS when it comes to templates that are built on the top of Tailwind CSS, so I would recommend to check out the [Tailwind documentation](#).

Files structure

All React component files are stored in the [src/](#) folder 📌

```
src/
|
├─ pages/
|   (contains site pages)
|
├─ partials/
|   (e.g. header, footer, page sections, etc)
|
└─ utils/
    (contains more units, like modal, dropdown, etc)
```

Add or remove pages

Every template comes with a set of ready-made pages, provided in form of components and stored into the [src/pages/](#) directory.

Router settings are defined in [src/main.js](#) file so, in case you want to add or remove pages, you should need to add or remove routes in that file. Check out the [Vue Router documentation](#) to know more.

Build the app for production

When you have done with changes, run `npm run build` for compiling and minify for production. You can learn more on the [Vite documentation](#).