Mosaic Documentation

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

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

- **Mosaic.fig** contains the design files for Figma
- **Mosaic.sketch** 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 a pre-compiled CSS file. You can use the Tailwind CLI tool if you need to make changes to the template, and need to re-compile the the CSS. To do that, just follow the stems below:

- First, ensure that node.js & npm are both installed. If not, choose your OS and installation method from this page and follow the instructions.
- Next, use your command line to enter your project directory.
- This template comes with a ready-to-use package file called `package.json`. You just need to run `npm install` to install all of the dependencies into your project.
- When `npm` has finished with the install, run `npm run build` to recompile the `style.css` file in the root directory.

You're ready to go! The most useful task for rapid development is `npm run dev`, which rebuild the CSS every time you make a change in the HTML or JS files.

React template

Unzip and open the folder of the React 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 React + Vite 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 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 run dev` 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, dropddown, 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` for compiling and minify for production. You can learn more on the Vite 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: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, dropddown, 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.

Laravel template

Unzip and open the folder of the Laravel template with your favourite editor (I am using VSCode).

Setup your .env config file

Make sure to add the database configuration in your .env file such as database name, username, password and port.

Install Laravel dependencies

In the root of your Laravel application, run the `php composer.phar install` (or `composer install`) command to install all of the framework's dependencies.

Migrate the tables

In order to migrate the tables and setup the bare minimum structure for this app to display some data you shoud open your terminal, locate and enter this project directory and run `php artisan migrate`.

Database seeding

Once you have all your database tables setup you can then generate some test data which will come from our pre-made database table seeders. In order to do so, in your terminal run `php artisan db:seed`.

> Note: If you run this command twice, all the test data will be duplicated and added to the existing table data, if you want to avoid having duplicate test data please make sure to `truncate` the following tables in your database:

- `campaign_marketer`
- `campaigns`
- `customers`
- `datafeeds`
- `invoices`
- `jobs`
- `marketers`
- `members`
- `orders`
- `transactions`

Launch the back-end

In order to make this Laravel installation work properly on your local machine you can run `php artisan serve`

You should receive a message like this:

`Starting Laravel development server: http://127.0.0.1:8000` simply copy the URL in your browser and you'll be ready to test out your new Mosaic Laravel app.

Compile the front-end

In order to compile all the CSS and JS assets for the front-end of this site you need to **install NPM dependencies**. To do that, open the terminal, type `npm install` and press the `Enter` key.

Then run `npm run dev` in the terminal to run a development server to re-compile static assets when making changes to the template.

When you have done with changes, run `npm run build` for compiling and minify for production.