**How to Configure / How to Install **

After downloading the file from themeforest, You will find Apat.zip file.Then unzip the Apat.zip and run the following commands ,

If you want to run these template on React Js server , then

1. yarn on Apat folder.

2. yarn web Apat folder.

Then , please go to address localhost:3000 on your browser and You will find Apat landing page.

If you want to Build these template on Next Js server , then

1. yarn on Apat folder.
2. 'yarn landing-build' on Apat folder.
3.'yarn landing-start' on Apat folder.

Stack We Used

1. Lerna (A tool for managing JavaScript projects with multiple packages. https://lernajs.io)

2. Yarn Workspace.

3. React Js and Next Js.
4. Gatsby Js.

5. Styled System and Styled Components

6. Firebase Deployment.

Folder Structure

Go to appart - > src folder . After entering to src folder, You will find some other folders.

1. assets

2. component

3. images

common

The whole work is in src -> component folder . After entering to component folder

1. Banner

2. Faq

3. Footer

4. Testimonial

Now, we will discuss about each and every folders and their tasks .

assets

In assets folder, you will find the css folder. In main.css folder, you will find all of the common styles needed for the template. You will also find the image folder, where all of the images are kept on the basis of the specific landing page.

components

If you are familiar with react or create react app architecture, then we are familiar with components. Components are reusable codes that you will use throughout your project . Here in components folder , we wrote some custom components which are used in our landing pages . We have done some basic style with the styled components (https://www.styled-components.com/).

Under the component folder, you will find some basic components like Text, Heading, Image, Input etc. We have written these components to make the developer's life easy. By using these basic components, you can write custom components according to your need.

Landing

In this folder , you will find the pages folder of next js and next configuration files. If you want to change something in next js folder , change here .

pages

As we have used next.js , we have a script to your package.json like this:

```
{
"scripts": {
"dev": "next",
"build": "next build",
"start": "next start"
}
}
```

After that, the file-system is the main API. Every .js file becomes a route that gets automatically processed and rendered.

Please have a look at this link https://nextjs.org/docs/ for a quick look. You will understand the basic things so quickly.

In these main js files, we have imported all the codes written in the common -> containers folder step by step. Every .js file becomes a route that gets automatically processed and rendered in next.js,so you will find out,

next.config.js

We have used some plugins for better performance and optimization .

1. next-optimized-images

2. next-fonts

3. next-css

Landing-Gatsby

In this folder , you will find the pages folder of gatsby js and gatsby configuration files. If you want to change something in gatsby js, change here.

gatsby-config.js

According to gatsby configurations , we gave support for styled components and web fonts for our project. The other settings are as same as gatsby default configurations . You can follow the config documentations of gatsby Js[
]()[https://www.gatsbyjs.org/docs/customization/#customization]() .

Icons

We have used custom flat icon. If you want to see our icon list then you need to go to pages->icons.js folder. After running yarn web command then go to your browser and write http://localhost:3000/icons and hit enter.

We have also used react icon kits to support large variety of icons. You can check out through this link https://wmira.github.io/react-icons-kit/

Firebase Deployment

You can follow this doc for firebase hosting https://firebase.google.com/docs/hosting/quickstart. We followed the documentation and integrated with our settings.

Step 1 : Install the Firebase CLI

1.Install Node.js using one of the following options. Installing Node.js automatically installs npm. (Note: The Firebase CLI requires Node.js v6.0.0 or later.)

2.Install the Firebase CLI using npm by running:

```
‘npm install -g firebase-tools’
```

3.Sign into Firebase using your Google account by running:

```
‘firebase login’
```

4.To test that authentication worked (and to list all of your Firebase projects), run the following command:

```
`firebase list`
```

Step 2 : Initialize your project

To connect your local project to your Firebase project, run the following command from the root of your local project directory:

```
`firebase init`
```

Select Firebase Hosting and Firebase function from the list. Then you have to answer couple of questions like:
What language would you like to use to write Cloud Functions? JavaScript
Do you want to use ESLint to catch probable bugs and enforce style?
No
Do you want to install dependencies with npm now?
No

What do you want to use as your public directory?
public

Then, an index.html file will created under the public directory.

Step 3 : Deploy your project

To deploy your project, you have to do couple of things. At first, go to firebase.json file. Then delete all codes.

Then, if you want to host the next js project, then go to landing.firebase.json file, copy all of the codes and paste it in firebase.json file.

If you want to host the Gatsby js project, then go to gatsby-landing.firebase.json file, copy all of the codes and paste it in firebase.json file.

Before running the below deploy command, you have to delete the public->index.html file. and go to landing->package.json, find engines and do the following

```
"engines": {
                "node": "8"
        }
```

Then, To deploy to your site, run the following command from the root of your local project directory:

`yarn firebase-deploy`

Now Deployment

Step 1 : Install the Now CLI

Install Now Desktop(https://zeit.co/download) or Now CLI(https://zeit.co/download#now-cli).

Step 1 : Create now.json file

If you want to host the next js project, Create a file named now.json in the root folder of your project. Then go to landing.now.json file, copy all of the codes and paste it in now.json file.

If you want to host the gatsby js project, Create a file named now.json in the root folder of your project. Then go to gatsby-landing.now.json file, copy all of the codes and paste it in now.json file.

You can change the name according to your project in now.json file like:

```
{
  "name": "<name-for-your-project>"
}
```

Step 3 : Deploy your project

Finally, simply run the following command to deploy it:

`now`

You may have to confirm your email address for first time. Please provide your email address in your command line and a confirmation email will be sent in your email address. Verify the email address and then run ‘now’ command again.

Netlify Deployment

At first, open an account on netlify and go to ‘sites’ tab.

If you want to host the next js project, go to your command line and run this command.

`yarn netlify-deploy`

After running this command, please go to ‘landing’ folder. You will find a ‘out’ folder there. Drag and drop this ‘out’ folder on netlify ‘sites’ tab.

If you want to host the gatsby js project, go to your command line and run this command.

`yarn gatsby-build`

After running this command, please go to ‘landing-gatsby’ folder. You will find a ‘public’ folder there. Drag and drop this ‘public’ folder on netlify ‘sites’ tab.