**Keen Super Admin & Keen Admin**

Tools Required:

* Node JS
* Visual Studio Code for React JS
* Git for Visual Studio Code

1. **Node JS**

Since Javascript is used for React Js development, there is no way around Node.js and NPM. Node.js is a JavaScript runtime which makes it possible to run JavaScript outside of the browser. The underlying node package manager (NPM) is used to install frameworks and libraries, such as React.js, to your project on the command line. They are called node packages. The node package manager is used to install libraries/frameworks (node packages), such as React.

Installation: https://nodejs.org/

1. **Visual Studio Code**

There are plenty of editors and IDEs out there to develop web applications. Depending on your personal, project or company requirements, you can choose from a range of lightweight editors to full-blown IDEs. Visual Studio Code is a solution in between. Many developers, operating on MacOS but also Windows, enjoy using it. That’s why my recommendation would be Visual Studio Code to start developing React applications on MacOS. I have used Visual Studio code for the complete development of the project from the very beginning hence I would recommend Visual Studio Code, it is lightweight and easy to use and the best part is you can choose from a variety of themes which makes looking at the code relaxing. Just a suggestion you can use any other IDEs you want but this is recommended.

The installation on a MacOS machine is simple: Navigate to the official [Visual Studio Code](https://code.visualstudio.com/) website and download the recent version of VS Code. Afterward, install it and check if everything is working when you open it. That’s it for the installation of an editor/IDE on your machine.

Installation:https://code.visualstudio.com/

1. **Git for Visual Studio Code**

All the projects that I do are always collaborated on github, it is the most convenient platform to share and work on the projects in a collaborative environment, you can easily install github through their official website.

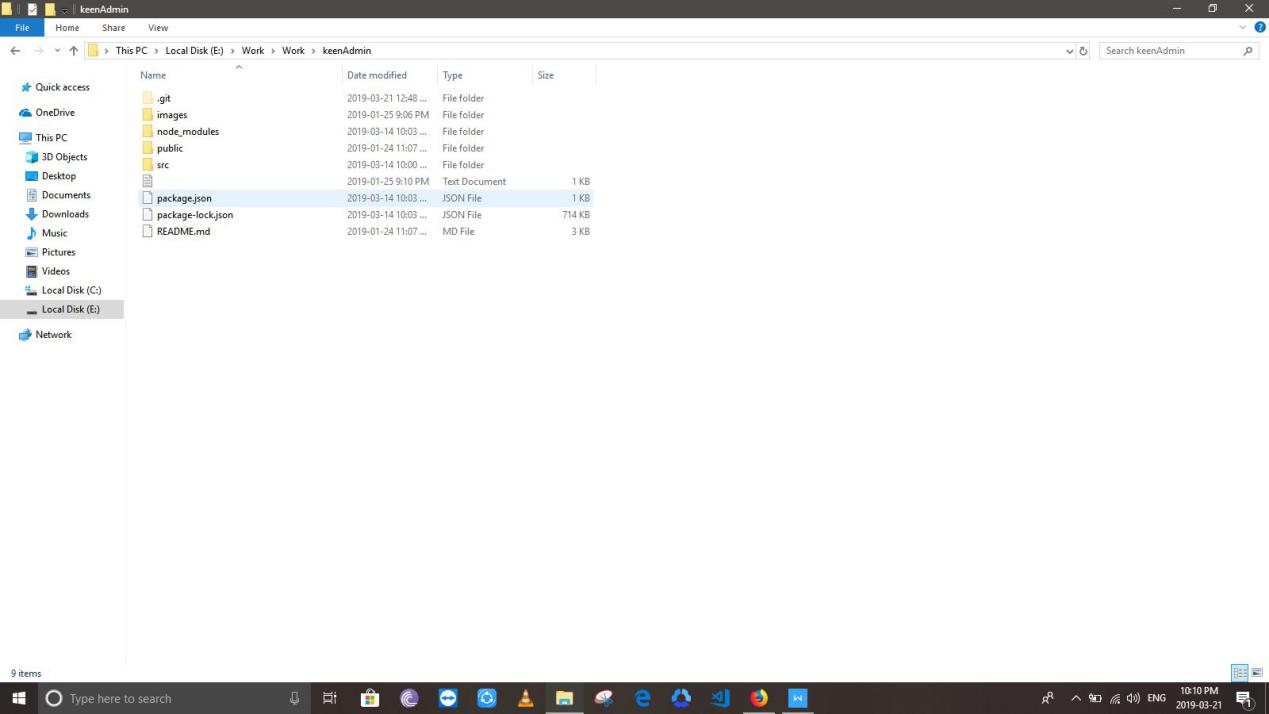
Installation:https://git-scm.com

How to setup the Project:

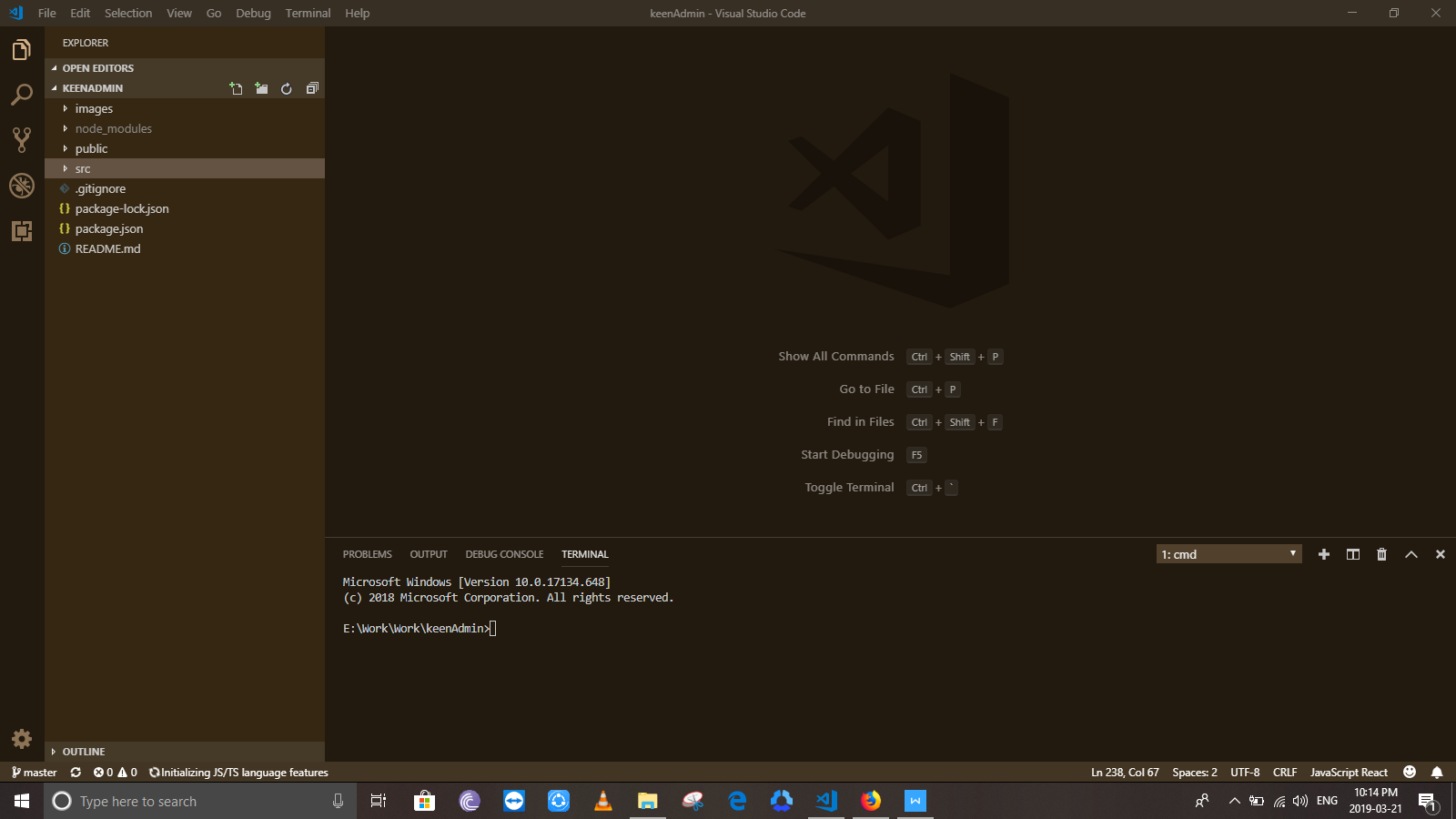
Now with all the required software installed, you can go to the github repositories and clone the project in your directory or just download the zip file. It is better if you clone the repositories from the bitbucket.

After cloning the apps successfully you can open the any one of the app directly from Visual Code or use command line to open the app. You can open command line from the root folder and type “code .” and it will open your project. Root folder will look like the picture below and you will not have the node\_modules folder.

There are two different apps so you will have to open them separately.



Once you open the required project you will see a screen like below, just open the terminal from visual studio code and type “npm install” this command will automatically install all the dependencies and npm packages required to run the project.

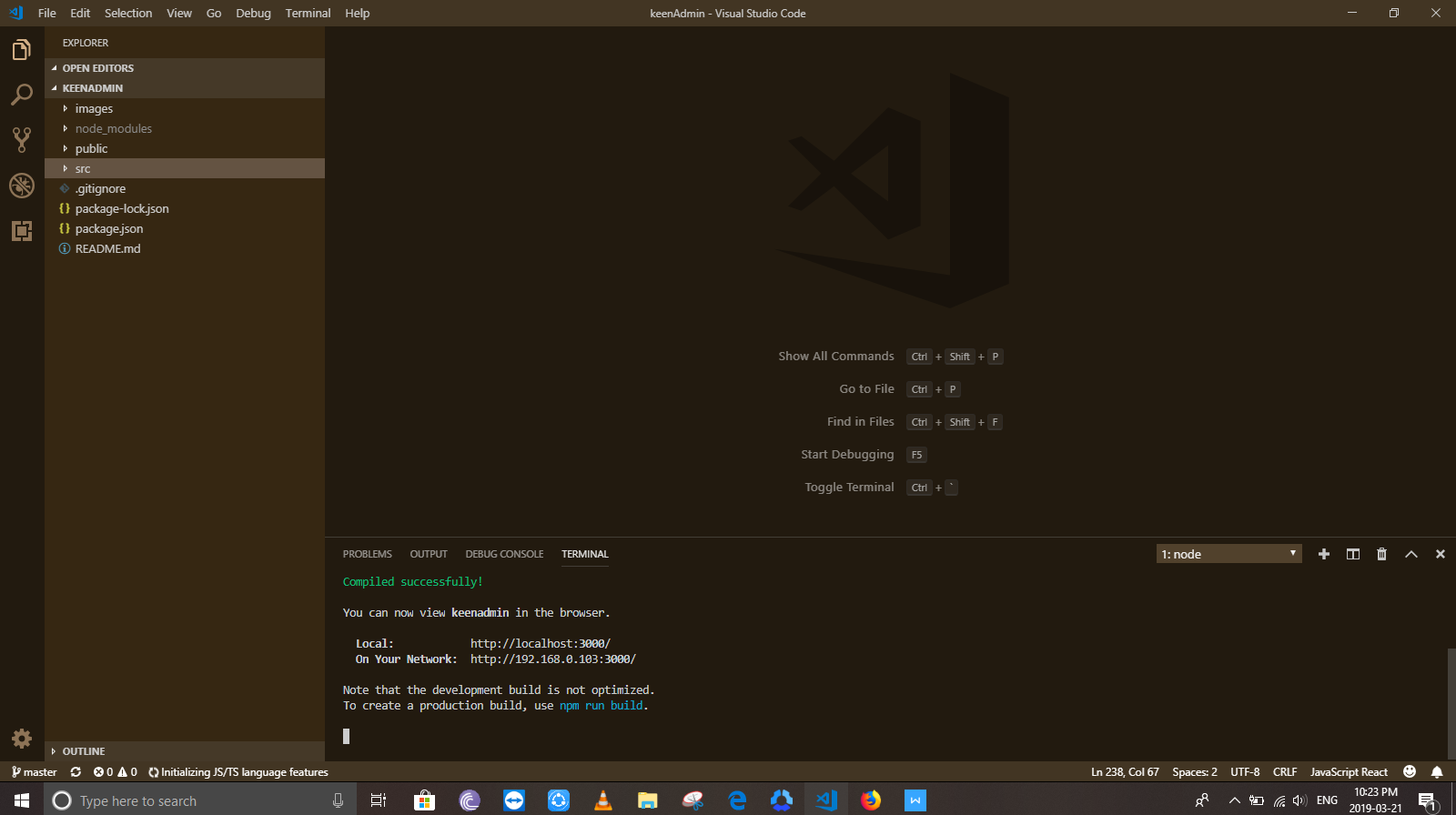


After all the dependencies are installed your project will be ready to run. You can start your project by typing “npm start” It will run on localhost:3000.

When you start the project Node will automatically run the required scripts to start your project.

Keen Admin and Keen Super Admin are different projects so you will have to open both projects separately and run the “npm install” command.

After the successful compilation of your project your project will be running in your in your default browser and your terminal will look like this.



Steps:

1. Install the software
2. Clone the repository
3. Open the app in Visual Code
4. Open Terminal
5. “npm install”
6. “npm start”

Whenever you make any changes to the files and hit save your app will automatically compile and restart.

How to Edit the UI:

* **Knowledge**

To make changes in the UI you will firstly need to have very good knowledge of React, how it works, rendering of elements, states and life cycles etc.

The best place to start is: <https://reactjs.org/docs/getting-started.html>

You will also to need to have good concepts of Javascript and ES6. All the new React is written in ES6, it can also be written in ES5 but ES6 is most preferable and also considered as best practice. Right now all the code is in ES6.

Then, I used “Material-Ui” as the framework for components. Materail UI is a React Framework used to create Material Designs. You should also have good knowledge of Material UI as well.

You can learn it from here: https://material-ui.com/

Now to make the edits anywhere in the project you will just have to understand the file structure. The file structure being used is also the best practice and I am sure you will not find it hard to understand.

**Following is the explanation for Keen Admin, both Keen Super Admin and Keen use the same file structure.**

* **File Structure**

The “node\_modules” folder contains all the node packages and it should be left as it is. It is not for editing at all.

“public” folder contains the html file which is basically the main file that runs. This fill is also not for editing unless you just want to add any kind of links. (for example adding bootstrap)

App.js is the entry point of your application. It only has one component “CustomRoutes” which contains all the routes of your application.

Now, there are two main folders here:

1. “src”
2. “firebase”

* “src”

This folder contains all the views of your application. Each and everything that you see in your application is somewhere here in this folder. Src has sub folders depending on what you wan to edit or see which will be explained further.

All of the sub folders contain a root file and sub folders, and all sub folders contain further files and sub folders, all the sub files and folders are eventually called in the root folder, which makes it possible for App.js to have only one component.

In order to make any any kind of edits a very good knowledge of React.JS and Material UI is a must.

* “firebase”

This folder contains the code for firebase integration and nothing more. This should not be edited unless you want to replace it with your database.

**VIEWS**

Views folder contains 4 root folders each of which has a root file. The folders are as follows

1. Home
2. Login
3. Navbar
4. routes

All the folders are named exactly as to what their purpose is I will explain all the folders in detail one by one.

In order to know from where a component is coming from you will just have to see the path from where it is imported. All the components are imported at the start of the files. All the Custom Components are being imported form files that start with “../” or “./”



Each folder serves for its purpose for example the “routes” folder contains a “routes.jsx” file which contains all the routes of the application.

The Drawer Folder contains the “Drawer.jsx” file which is the left drawer shown on the screen.

Views for Conversations and Users are complex and hence have there own folders inside the Home folder.

Similarly the Login folder contains the Login.jsx file which contains the code for the Login Page you see.

If you know how the components system works in React it will be much easier for you to edit any where in the project, even create your own components and integrate them with the existing ones.

The file structure being used is the best practice and most recommended for React JS it, knowing how React works will make it easier to work with the project.

**Conclusion:**

In order to make any kinds of Edits you will just have to learn React and once you learn React I am sure you would not want to work with any other front-end framework for web. Cheers! :)